

Министерство образования Республики Беларусь
Белорусский государственный университет
Механико-математический факультет

М.А. Журавков

Технологии искусственного интеллекта и
интеллектуальные системы компьютерного моделирования
и инженерных расчетов. Вводный курс

Учебное пособие

Минск
БГУ
2024

УДК 004.8(075.8)

Ж 91

Решение о депонировании вынес:
Совет механико-математического факультета
30 января 2024 г., протокол № 6

Автор:

Журавков Михаил Анатольевич, доктор физико-математических наук,
профессор, заведующий кафедрой теоретической и прикладной механики
ММФ БГУ.

Рецензенты:

Профессор кафедры веб-технологий и компьютерного моделирования
ММФ БГУ, академик, доктор тех. наук, профессор Абламейко С.В.;
Профессор кафедры теоретической механики и механики материалов,
машиностроительный факультет, БНТУ, доктор физ.-мат. наук, профессор
Василевич Ю.В.

Журавков, М.А. Технологии искусственного интеллекта и интеллектуальные системы компьютерного моделирования и инженерных расчетов. Вводный курс : учеб. пособие / М.А. Журавков ; БГУ, Механико-математический фак. – Минск : БГУ, 2024. – 177 с. : ил. – Библиогр.: с. 170–173.

Учебное пособие представляет собой вводный курс в такое актуальное современное направление, как технологии искусственного интеллекта (ИИ) и их применение в интеллектуальных системах компьютерного моделирования и инженерных расчетов.

В пособии рассматриваются основные определения, понятия технологий ИИ, составные части ИИ через собственное восприятие и интерпретацию автора применительно к разработке интеллектуальных систем компьютерного моделирования и инженерных расчетов для различных классов задач механики.

Данное пособие используется, в частности, при чтении таких специальных курсов для студентов отделения «Механика. Математическое моделирование» механико-математического факультета БГУ, как ««Основы САД и САЕ технологий в механике. Основы технологий ИИ в механике», «Компьютерное моделирование».

Материал данного курса лекций представляет интерес как специалистам в области теоретической и прикладной механики, математического моделирования, численных методов, так и может быть рекомендован широкому кругу читателей, интересующихся применением современных методов численного и компьютерного моделирования в прикладных науках.

ВВЕДЕНИЕ

ИИ способен изменить наш мир, то, как мы живем и взаимодействуем: повысить производительность и вовлеченность в работу, стимулировать творчество.

Однако эти технологии развиваются экспоненциально быстро и выводят мир на неизведанную территорию, наполненную помимо положительного эффекта и потенциальными рисками, включая неправильное использование технологий, нарушения безопасности и другие риски.

(Давос-2024. Из итогов Всемирного экономического форума)

2 ноября 2023 года Великобритания, США, ЕС, Австралия, Китай и ряд других стран на саммите, проходившем в Великобритании, подписали первую в истории международную декларацию, в которой говорится, что **искусственный интеллект (ИИ)** представляет потенциально катастрофический риск для человечества.

В британском саммите по безопасному использованию искусственного интеллекта приняли участие более 100 деятелей политики и бизнеса, в том числе генеральный директор OpenAI Сэм Олтмен и владелец SpaceX, Tesla Илон Маск. Республика Корея согласилась провести еще один такой саммит через шесть месяцев, а Франция проведет его через год.

28 стран подписали так называемую **декларацию Блетчли** в первый день саммита по безопасному использованию ИИ. Декларация получила свое название от места проведения саммита – особняка в Блетчли. Страны договорились вместе работать над исследованиями в области безопасности ИИ, даже несмотря на признаки того, что США и Великобритания конкурируют за лидерство в разработке новых правил.

Британский премьер-министр Риши Сунак отметил: «Нет ничего более преобразующего для будущего наших детей и внуков, чем такие технологические достижения, как искусственный интеллект. Однако мы обязаны обеспечить им безопасное и ответственное развитие ИИ, осознавая риски, которые он представляет, на ранних стадиях процесса». Заместитель министра науки и технологий КНР У. Чжаохуэй сказал коллегам-делегатам: «Мы поддерживаем принципы взаимного уважения, равенства и взаимной выгоды. Страны, независимо от их размера и масштаба, имеют равные права на разработку и использование ИИ». В кулуарах саммита Илон Маск предупредил: «Впервые мы столкнулись с ситуацией, когда происходит что-то, что будет значительно умнее самого умного человека. Мне не ясно, можем ли мы на самом деле контролировать такую вещь».

До сих пор в мире не существует единого мнения относительно того, как может выглядеть глобальный набор правил ИИ или кто должен их разрабатывать.

В настоящее время наблюдается всплеск повышенного интереса к ИИ и даже необычного ажиотажа вокруг данной тематики. По сути дела,

ситуация с ИИ напоминает то, что было во время начала массового проникновения персональных компьютеров во все сферы жизнедеятельности человека.

Количество публикаций, специальных репортажей, интервью, где затрагиваются различные аспекты ИИ растет в геометрической прогрессии. Создается впечатление, что все, что сегодня производит человек, начиная от технологий и до готовых изделий, уже обладает в том либо иной мере ИИ. Даже специалисты-эксперты зачастую сомневаются, когда необходимо выдать заключение о том, имеет ли рассматриваемый объект ИИ или нет.

Внедрение информационно-коммуникационных технологий привело к изменениям всех сфер жизни. С момента появления первого компьютера до полной роботизации целых производств, а не только отдельных технологических процессов, прошло не так уж много времени. В настоящее время вопрос уже заключается не только в том, каким образом автоматизировать рутинную работу человека, роботизировать большое количество операций (т.е. разработка эффективных алгоритмов, программируемых определенных действия), но и как наделять робота (систему) «интеллектом» (научить систему определенной «мыслительной деятельности»). Все это возможно с помощью технологий ИИ. Данные технологии сейчас стремительно развиваются и получают всё больше внимания со стороны не только исследователей, но и практиков во многих отраслях народного хозяйства.

В процессе развития ИИ исследователи, работающие в этой области, столкнулись с проблемами, которые выходят за границы традиционной информатики. Как оказалось, прежде всего, необходимо понять механизмы процессов обучения, чувственного восприятия, понять природу языка и многое другое. Для того, чтобы имитировать работу мозга человека необходимо понять механизм действия миллиардов взаимосвязанных нейронов.

Одна из основных проблем, которая стоит перед исследователями и разработчиками ИИ – ***познание процесса функционирования человеческого разума, а не просто имитация его работы*** [1]. Однако это не единственная глобальная проблема ИИ как направления науки. Так, ещё одна заключается в том, что ***учёные никак не могут сойтись во мнениях относительно самого предмета исследования – интеллекта.***

На данный момент существует несколько определений интеллекта, базирующихся на различном взгляде на проблему [1]. Для некоторых интеллект – это умение решать сложные задачи. Другие рассматривают его как способность обучаться, обобщать, анализировать, третьи – как способность воспринимать информацию и сохранять её в качестве знания для построения адаптивного поведения в среде или контексте. Для

четвёртых – это способность выполнять функции творческого характера, которые, как правило, относят исключительно к человеческим способностям. Для некоторых же интеллект может быть только биологическим феноменом.

По причине разницы в определениях и желании достичь более специфичных результатов, основанных на данных взглядах на проблему определения интеллекта, *различаются и подходы к его исследованию*. При этом, из всех подходов можно выделить два основных:

- *нисходящий*, он же *семиотический* – создание экспертных систем, имитирующих высокоуровневые психологические процессы;
- и *восходящий* или *биологический* – изучение нейронных сетей и эволюционных вычислений, моделирующих интеллектуальное поведение на основе биологических элементов.

ИИ можно рассматривать как область знаний, то есть ИИ как наука.

С этой точки зрения Искусственный Интеллект – это наука и технологии создания интеллектуальных машин и систем, которые могут выполнять творческие функции, традиционно считающиеся прерогативой человека.

Но, так как ИИ непосредственно связан с информационными системами, поэтому **ИИ** рассматривают и как **интеллектуальные компьютерные системы**, умеющие интерпретировать сложные задачи и самостоятельно разрабатывать подходящие решения на уровне, достаточно близком к мыслительным психологическим способностям человека.

Чтобы более четко понять суть ИИ, все же следует обратиться к его базовому определению и основным понятиям.

1. ОБЩИЕ ПОНЯТИЯ, ОПРЕДЕЛЕНИЯ

1.1. Происхождение термина «искусственный интеллект».

Дартмутский семинар

Термин «искусственный интеллект» был введён Джоном Маккарти в 1956 году на Дартмутском семинаре и *сам термин не связан напрямую с пониманием интеллекта человека.*

Здесь следует уточнить что в английском языке словосочетание *artificial intelligence* не имеет антропоморфной окраски, которую оно приобрело в традиционном русском переводе: слово **intelligence** в используемом контексте **скорее означает «умение рассуждать разумно».**

Поясняя своё определение, Джон Маккарти указывал: *«Проблема состоит в том, что пока мы не можем в целом определить, какие вычислительные процедуры мы хотим называть интеллектуальными. Мы понимаем некоторые механизмы интеллекта и не понимаем остальные. Поэтому под интеллектом в пределах этой науки понимается только вычислительная составляющая способности достигать целей в мире».*

Говоря про саму конференцию в Дартмутском университете это был семинар, посвящённый искусственному интеллекту, на который пригласили всех видных американских исследователей, так или иначе связанных с вопросами теории управления, теории автоматов, нейронных сетей, теории игр и исследованием интеллекта.

В заявке на проведение мероприятия Маккарти указал: *«Мы предлагаем исследование искусственного интеллекта сроком в 2 месяца с участием 10 человек. Исследование основано на предположении, что любое свойство интеллекта может быть столь точно описано, что машина сможет его симулировать. Мы попытаемся понять, как обучить машины использовать естественные языки, формировать абстракции и концепции, решать задачи, сейчас подвластные только людям, и улучшать самих себя. Мы считаем, что существенное продвижение в одной или более из этих проблем вполне возможно, если специально подобранная группа учёных будет работать над этим в течение лета».*

И всё же основной целью данного собрания было рассмотрение вопроса: *можно ли моделировать рассуждения, интеллект и творческие процессы с помощью вычислительных машин.*

В ходе семинара, в итоге был сформулирован **один из основных принципов создания искусственного интеллекта – меняющиеся ответы на переменную среду.**

В заявке *этот тезис был сформулирован Марвином Минским:* «Нужно разработать машину, которая бы демонстрировала определённый вид обучения. Такая машина должна быть снабжена входным и выходным

каналами, то есть средствами обеспечения разнообразных выходных ответов на входящие вопросы. Такой метод обучения можно назвать «метод проб и ошибок», то есть процесс приобретения диапазона вариантов ввода-вывода функций. Машина, спрограммированная таким образом, при помещении в соответствующую среду и с учётом критериев «провал/успешное достижение цели» может быть обучена проявлять целенаправленное поведение».

Дартмутский семинар не стал местом каких-либо новых крупных открытий, но именно он позволил сойтись вместе и познакомиться всем наиболее важным деятелям в этой научной области. В результате Дартмутская конференция стала ежегодным событием, посвящённым достижениям в области искусственного интеллекта.

1.2. Что такое ИИ?

После Дартмутского семинара Джоном Маккарти в 1959 г. в статье «Программы со здравым смыслом» [2], подробно описал новое понятие «искусственный интеллект», где ИИ рассматривался именно как вычислительная система, способная создавать подпрограммы. Позже Джон Маккарти уточнил, что «искусственный интеллект – это вычислительная часть способности достигать целей в мире». Все технические системы являются целенаправленными, и это демонстрирует применимость методологии ИИ в инженерной деятельности. Джон Маккарти подчеркивал, что введенное им определение «Intelligence» означает «сообразительность», «понимание», «способности», «проницательность», «распознавание» «сбор и обработку информации». **Именно с этих позиций, на наш взгляд, необходимо рассматривать ИИ на текущий момент развития науки и технологий.**

Следует сказать, что в русскоязычном переводе смысл термина «artificial intelligence» несколько исказился и сегодня имеет место некоторая путаница и завышенные неоправданные ожидания в отношении ИИ. Но, стоит подчеркнуть, что **если бы Дж. Маккарти имел в виду именно «интеллект», то скорее бы применил слово «intellect», а не «intelligence».**

Замечание. Так, например, британская разведка Intelligence Service означает «Службу по сбору информации».

Некоторая двусмысленность в понятии ИИ возникла, наверное, из-за того, что под ИИ в первую очередь понимают системы решений, позволяющие имитировать мыслительные (когнитивные) функции человека и на этой основе получать выводы и результаты, сопоставимые с результатами интеллектуальной деятельности человека. Важным обстоятельством при этом является наличие в имитационном процессе этапов самообучения системы и поиска решений (зачастую без заранее

заданного алгоритма).

Поэтому, говоря об ИИ будем отталкиваться именно от такой **интерпретации термина «artificial intelligence»**: наличие в системе алгоритмов с реализацией таких качеств, как «обработка, распознавание и интерпретация», «анализ и предсказание», «понимание и проницательность», базирующиеся на законах, методах и подходах современной науки и, прежде всего, математики, физики и информатики. При этом, указанные качества имеют весьма широкий диапазон приложений.

С этих позиций можно считать, что **ИИ – это средства автоматизации интеллектуальной деятельности человека**. Исходя из такой интерпретации следует, что в ИИ можно выделить несколько основных направлений [3]. Одно из основных направлений ИИ связано с *автоматизацией рациональной деятельности* человека. Примером таковой является решение задач по классическим курсам «Теоретическая механика» или «Математический анализ». Следующее важное направление ИИ – *автоматизация интуитивной деятельности* человека. Например, умение распознавать лица на фотографии.

Один из основных принципов создания искусственного интеллекта — меняющиеся ответы на переменную среду.

Повышенное внимание к ИИ в средствах массовой информации, помимо специализированных изданий, когда демонстрируются уникальные возможности антропоморфных и бионических роботов, специализированных игровых систем и комплексов, создает у людей впечатление, что технологии ИИ имеют значительное развитие в широком диапазоне приложений. В реальности, дела обстоят совершенно не так и по многим направлениям внедрение ИИ находится еще только на начальной стадии.

Так, например, в горнодобывающей промышленности, в машиностроительной отрасли, где потенциал и эффективность использования технологий ИИ просто безграничны, примеров разработки систем с элементами ИИ еще совершенно мало.

В целом, сегодня ИИ всё ещё остаётся относительно новым научным направлением, исследующим важные и пока не решённые фундаментальные проблемы. К ним относятся аспекты сознания, мышления, принятия решений, оптимизации систем, обработки больших данных и «машинного обучения», нечёткой логики и генетических алгоритмов. Интересно подчеркнуть, что методология ИИ используется в различных отраслях науки и техники. Такая широта проникновения в различные отрасли науки делает ИИ своего рода современной философией, особенно если принять во внимание необходимость рассмотрения проблем

сознания, мышления и их взаимодействие с материей (материальными носителями) при решении задач ИИ.

Для использования и развития искусственного интеллекта необходимо наличие как минимум трех составляющих [1]:

- ✓ значительные вычислительные мощности,
- ✓ большие объемы данных и знаний,
- ✓ развитые интеллектуальные алгоритмы.

В 21-м веке существенно выросли вычислительные мощности, математиками и программистами разработаны новые эффективные методы и алгоритмы в области ИИ (в, частности, методы «глубокого обучения»). Это в совокупности и обусловило значимый прогресс в области создания современных технологий ИИ и, что является важным, стимулировало правительства многих стран серьезно заняться вопросами поддержки развития ИИ в своих странах.

Область ИИ как наука взаимосвязана со многими другими областями научных исследований. Среди них, в первую очередь, разделы фундаментальной и прикладной математики, физика, обработка сигналов, машинное обучение, компьютерное зрение, психология, лингвистика, биология, наука о мозге и др.

Необходимое требование к «системам с интеллектом» в настоящее время заключается в том, что элементы ИИ не должны «работать» как «чёрные ящики», выдающие решение. Они должны представлять собой не только вызывающий доверие инструментарий для решения задачи, но и демонстрировать понятный и эффективный путь получения решения. В особенности это проявляется при разработке автоматизированных систем поддержки принятия решений как одного из наиболее перспективных направлений развития ИИ, в особенности при разработке интеллектуальных систем моделирования и прикладных расчетов. Отметим, что *на современном этапе развития технологий ИИ речь идет именно об автоматизированных системах, т.е. системах с участием человека в управление процессом, осуществляемым при поддержке ИИ*. С этой точки зрения, одними из стратегических целей активного развития систем ИИ является [4, 5]:

- разработка математических основ методов обработки и интеллектуального анализа данных для различных прикладных областей и направлений;
- разработка математических основ систем компьютерного моделирования, расчетов и анализа разнообразных физических процессов;
- переход к новым интеллектуальным CAD-, CAE- и CAM-технологиям. К перечисленным технологиям в плане использования ИИ в системах

механико-математического моделирования и инженерных расчетов (САЕ-системах) считаем необходимым добавить еще следующие [5]:

- рекомендации к постановке модельных математических и компьютерных задач» для реальных исследуемых процессов и явлений;
- интеллектуальная интерпретация и анализ результатов моделирования.

Принцип работы ИИ заключается в сочетании большого объема данных с возможностями быстрой, итеративной обработки этих данных интеллектуальными алгоритмами, что позволяет программам автоматически обучаться на базе закономерностей и признаков, содержащихся в данных [1].

При создании систем ИИ в настоящее время разработчики в большинстве своем ориентируются на *технологии машинного обучения*: компьютерная программа обрабатывает данные и предлагает решение вне зависимости от строго обозначенных схем (алгоритмов). Программа «находит» закономерности, зависимости, формулирует ответы и выдает прогнозы в заданиях с большим набором параметров, что не под силу человеку.

Для обработки данных в системах ИИ используются *искусственные нейронные сети (ИНС)*, созданные по подобию биологических нейронных сетей. ИНС завоевывают все большую популярность и по существу являются на сегодня главным «инструментарием» базовых математических моделей ИИ. Помимо ИНС применяются также методы статистики, исследования операций и др.

Расширение сферы приложений ИИ, усложнение класса решаемых задач, спектра и объема данных, используемых для создания прикладных ИИ-моделей и интеллектуальных систем на базе ИИ, требует существенного расширения теоретической и алгоритмической базы ИИ, включая необходимость развития методов машинного обучения с использованием математических и физических моделей объектов и явлений предметных областей, методов консолидации мультимодальных данных, методов создания геометрических и топологических компонентов нейронных глубоких сетей, методов моделирования изучаемых 3D-объектов и др. [6].

1.3. Основные направления исследований в области ИИ в настоящее время

1.3.1. Символьное моделирование мыслительных процессов

Анализируя историю ИИ, такое направление исследований, как *моделирование рассуждений*, можно выделить, наверное, в качестве одного из первых основных направлений. Долгие годы развитие ИИ как науки продвигалось именно по этому пути, и теперь это одна из самых

развитых областей в современных технологиях ИИ. Моделирование рассуждений подразумевает создание символьных систем, на входе которых поставлена некая задача, а на выходе требуется её решение. *Как правило, предлагаемая задача уже формализована, то есть переведена в математическую форму, но, либо не имеет алгоритма решения, либо он слишком сложен и трудоёмок.* Это направление включает, например: доказательство теорем, принятие решений и теория игр, планирование и диспетчеризация, прогнозирование и др.

1.3.2. Работа с естественными языками

Важным направлением ИИ является обработка естественных языков. Данное направление изучает проблемы компьютерного анализа и синтеза естественных языков. Применительно к ИИ анализ означает понимание языка, а синтез – генерацию грамотного текста. Решение этих двух проблем означает создание более удобной формы взаимодействия компьютера и человека.

Теоретически, построение естественно-языкового интерфейса для компьютеров – очень привлекательная цель. Ранние системы, работая с ограниченным «миром кубиков» и используя ограниченный словарный запас, все же выглядели достаточно хорошо, вдохновляя этим своих создателей. Однако оптимизм быстро иссяк, когда эти системы столкнулись со сложностью и неоднозначностью реального мира.

Понимание естественного языка считается полной задачей ИИ, потому как распознавание живого языка требует огромных знаний системы об окружающем мире и возможности с ним взаимодействовать. Само определение смысла слова «понимать» – одна из главных задач искусственного интеллекта.

На данный момент проблема восприятия текста машинами активно изучается. Множество разработчиков пытаются создать алгоритм, благодаря которому программа бы воспринимала и обрабатывала семантику текста, находила и исправляла ошибки не грамматические и пунктуационные, а фактические и логические, которые сейчас может найти лишь человек.

Например, в научной работе автор необоснованно использовал выражение, которое (специалисты это знают) в литературе по этой тематике является ошибкой, или противоречит общепринятому определению. Ошибки подобного типа легко определяются человеком, но не машиной. *Это и есть основная проблема, с которой работает данное направление исследований искусственного интеллекта.*

Замечание. В связи с актуальностью этой проблемой она признана одной из основных проблем столетия в сфере искусственного интеллекта и за решение проблемы восприятия смысла текста на русском языке можно

получить премию в размере 100 000 000Р, и столько же за решение для английского языка.

1.3.3. Представление, обработка и использование знаний

Направление ***инженерия знаний*** объединяет задачи получения знаний из простой информации, их систематизации и использования. Это направление исторически связано с созданием ***экспертных систем*** – программ, использующих специализированные базы знаний для получения достоверных заключений по какой-либо проблеме, для решения сложных задач, обычно требующих богатого человеческого опыта. Помимо этого, данное направление так же изучает методы и средства извлечения, представления, структурирования и использования знаний.

Извлечение информации – это задача автоматического извлечения (построения) структурированных данных из неструктурированных или слабоструктурированных машиночитаемых документов. Извлечение информации является разновидностью информационного поиска, связанного с обработкой текста на естественном языке.

К типичным задачам извлечения информации относится, например, следующая: просканировать набор документов, написанных на естественном языке, и наполнить базу данных выделенной полезной информацией. К сожалению, современные подходы извлечения информации используют методы обработки естественного языка, направленные лишь на очень ограниченный набор тем (вопросов, проблем), часто только на одну тему.

Представление знаний – вопрос, возникающий в когнитологии (науке о мышлении), в информатике и в исследованиях искусственного интеллекта. В когнитологии он связан с тем, как люди хранят и обрабатывают информацию. В информатике – с подбором представления конкретных и обобщённых знаний, сведений и фактов для накопления и обработки информации в компьютерах. ***Главная задача в искусственном интеллекте – научиться хранить знания таким образом, чтобы программы могли осмысленно обрабатывать их и достигнуть тем самым подобия человеческого интеллекта.***

Под термином «представление знаний» чаще всего подразумеваются способы представления знаний, ориентированные на автоматическую обработку современными компьютерами, и, в частности, представления, состоящие из явных объектов и из суждений или утверждений о них. Представление знаний в подобной явной форме позволяет компьютерам делать дедуктивные выводы из ранее сохранённого знания.

Некоторые вопросы, которые возникают в представлении знаний с точки зрения ИИ [7]:

- Какова природа знаний и как мы представляем знания?

- Должна ли схема представления связываться с частной областью знаний, или она должна быть общецелевой?
- Насколько выразительна данная схема представления?
- Должна ли быть схема декларативной или процедурной?
- *Пример действия системы, базирующейся на инженерии знаний:*
- Рассмотрение задачи,
- Запрос к базам данных по задаче,
- Внесение и структурирование полученной информации,
- Создание базы данных по структурированной информации,
- Тестирование полученной информации,
- Внесение корректировок и доработка системы.

Инженерия знаний сегодня уже имеет широкое практическое применение. В США до 90% кредитных решений по клиентам розничного банковского бизнеса принимается с использованием экспертных систем на основе баз знаний FICO. Подразделом инженерии знаний является **метаинженерия знаний**, пригодная для разработки ИИ.

Производство знаний из данных – одна из базовых проблем интеллектуального анализа данных. Существуют различные подходы к решению этой проблемы, в том числе – на основе нейросетевых технологий, использующих процедуры вербализации нейронных сетей.

Замечание. Вербализация нейронных сетей – минимизирование описания работы синтезированной и уже обученной нейронной сети в виде нескольких взаимозависимых алгебраических или логических функций.

1.3.4. Машинное обучение

Проблематика машинного обучения касается **процесса самостоятельного получения знаний интеллектуальной системой в процессе её работы**. Это направление было центральным с самого начала развития ИИ. В 1956 году, на Дартмутской летней конференции, Рей Соломонофф написал отчёт о вероятностной машине, обучающейся без учителя, назвав её: «Индуктивная машина вывода».

Машинное обучение – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, методов оптимизации, теории вероятностей, теории графов, различные техники работы с данными в цифровой форме.

Различают два подхода к обучению:

1. Обучение по прецедентам, или индуктивное обучение, основано на выявлении эмпирических закономерностей в данных.

2. Дедуктивное обучение предполагает формализацию знаний экспертов и их перенос в компьютер в виде Базы Знаний.

Дедуктивное обучение в большинстве своем принято относить к области экспертных систем. Поэтому по своей сути термины «машинное обучение» и «обучение по прецедентам» можно считать синонимами.

Задачу для дедуктивного обучения (машинного обучения) можно сформулировать следующим образом:

Имеется множество объектов (ситуаций) и множество возможных ответов (откликов, реакций). Существует некоторая зависимость между ответами и объектами, но она неизвестна. Известна только конечная совокупность прецедентов – пар «объект – ответ», называемая обучающей выборкой. На основе этих данных требуется восстановить неявную зависимость, то есть построить алгоритм, способный для любого возможного входного объекта выдать достаточно точный классифицирующий ответ. Эта зависимость не обязательно выражается аналитически и здесь реализуют принцип эмпирически формируемого решения. Важной особенностью при этом является способность обучаемой системы к обобщению, то есть к адекватному отклику на данные, выходящие за пределы имеющейся обучающей выборки. Для измерения точности ответов вводится оценочный функционал качества.

Данная постановка является обобщением классических задач аппроксимации функций. В классических задачах аппроксимации объектами являются действительные числа или векторы. В реальных прикладных задачах входные данные об объектах могут быть неполными, неточными, нечисловыми, разнородными. Эти особенности приводят к большому разнообразию *методов машинного обучения*.

Целью машинного обучения является частичная или полная автоматизация решения сложных профессиональных задач в самых разных областях человеческой деятельности.

Сфера применений машинного обучения постоянно расширяется. Повсеместная информатизация приводит к накоплению огромных объёмов данных в науке, производстве, бизнесе, транспорте, здравоохранении. Возникающие при этом задачи прогнозирования, управления и принятия решений часто сводятся к обучению по прецедентам. Раньше, когда таких данных не было, эти задачи либо вообще не ставились, либо решались совершенно другими методами.

1.3.5. «Биологическое моделирование» в системах ИИ

Данный подход отличается от понимания искусственного интеллекта по Джону Маккарти, при котором исходят из положения о том, что искусственные системы не обязаны повторять в своей структуре и

функционировании структуру и протекающие в ней процессы, присущие биологическим системам.

Сторонники «биологического моделирования» считают, что феномены человеческого поведения, его способность к обучению и адаптации – это следствие именно биологической структуры и особенностей её функционирования.

К технологиям «биологического моделирования» можно отнести несколько направлений. Нейронные сети используются для решения нечётких и сложных проблем, таких как распознавание геометрических фигур или кластеризация объектов. Генетический подход основан на идее, что некий алгоритм может стать более эффективным, если позаимствует лучшие характеристики у других алгоритмов («родителей»). Относительно новый подход, где ставится задача создания автономной программы – агента, взаимодействующего с внешней средой, называется агентным подходом.

Биологическое направление, называемое так же квазибиологическая парадигма, сосредоточено на разработке и использовании компьютеров, которые функционируют как живые организмы или содержат биологические компоненты, так называемые биокомпьютеры.

Биокомпьютинг позволяет решать сложные вычислительные задачи, организуя вычисления при помощи живых тканей, клеток, вирусов и биомолекул. Часто используют молекулы дезоксирибонуклеиновой кислоты, на основе которых создают ДНК-компьютер. Кроме ДНК, в качестве биопрототипа могут использоваться также белковые молекулы и биологические мембраны. Например, на основе бактериородопсин-содержащих плёнок создают молекулярные модели перцептрона.

1.3.6. Интеллектуальная робототехника

Области робототехники и искусственного интеллекта тесно связаны друг с другом. Интегрирование этих двух наук, создание *интеллектуальных роботов* составляют ещё одно направление ИИ.

Интеллектуальность требуется роботам, чтобы манипулировать объектами, выполнять навигацию с проблемами локализации (определять местонахождение, изучать ближайшие области) и планировать движение (как добраться до цели).

Интеллектуальная робототехника – прикладная наука, занимающаяся разработкой автоматизированных робототехнических систем с использованием наработок искусственного интеллекта.

В 1832 году Семён Николаевич Корсаков представил серию из пяти «интеллектуальных машин» — механических прообразов современных экспертных систем, в конструкции которых впервые в истории информатики были применены перфорированные карты. С формальной

точки зрения машины Корсакова реализовывали основные операции над множествами, представляющие тот базис, который лежит в основе современной дискретной математики и информатики.

Примером интеллектуальной робототехники могут служить многочисленные современные игрушки-роботы (Pleo, AIBO, QRIO и др.).

1.3.7. Машинное творчество

Природа человеческого творчества ещё менее изучена, чем природа интеллекта. Тем не менее, эта область ИИ существует. Основные задачи данного направления ИИ – написание компьютером музыки, литературных произведений (часто стихов или сказок), художественное творчество. Так, создание реалистичных образов широко используется в кино и индустрии игр.

К отдельному направлению можно отнести решение проблемы технического творчества в системах искусственного интеллекта. Началом исследований в этом направлении можно считать разработку в 1946 году Г.С. Альтшуллером теории решения изобретательских задач.

Конечно, помимо перечисленных существуют, конечно, и другие направления использования и развития ИИ, которые в свою очередь объединяют или вообще не пересекаются с приведёнными выше примерами. Например, программирование интеллекта в компьютерных играх, нелинейное управление, интеллектуальные системы информационной безопасности и т.д. В перспективе предполагается тесная связь развития искусственного интеллекта с разработкой квантового компьютера, так как некоторые свойства искусственного интеллекта имеют схожие принципы действия с квантовыми компьютерами.

1.4. Современный ИИ. Эра «сверхразума» – общий ИИ

В настоящее время можно выделить два крупных стратегических направления развития искусственного интеллекта:

- решение проблем, связанных с приближением специализированных систем ИИ к возможностям человека, и их интеграции, которая реализована природой человека;
- создание искусственного разума, представляющего интеграцию уже созданных систем ИИ в единую систему, способную решать проблемы человечества.

Сегодня наблюдается вовлечение большого количества предметных областей в сферу ИИ, имеющих скорее практическое отношение к ИИ, а не фундаментальное. Многие подходы были опробованы, но к созданию искусственного разума ни одна исследовательская группа пока так и не подошла.

Одна из основных проблем ИИ, как уже было сказано ранее, заключается в отсутствии контекста. *Важным отличием человеческого интеллекта от машинного является контекст того или иного действия.* У нас, поскольку мы разумные люди, есть лучшее понимание окружающего мира, а вот у ИИ такого понимания нет.

Облачная робототехника, машинное обучение, компьютерное зрение, распознавание речи – все эти аспекты ИИ развиваются и продвигаются, порой делая значительные шаги в конкретных областях. Однако у ИИ по-прежнему нет ничего человеческого.

Если роботы, используя ИИ и человеческую инженерию, в ближайшем будущем смогут приблизиться к «ловкости человека», но еще не скоро они смогут по-настоящему понять находящийся вокруг них мир во всей совокупности всех составляющих. Контекст и изобретательность все еще находятся в сфере деятельности людей. Технология сама по себе не является ни плохой, ни хорошей – все зависит лишь от того, как мы ее используем. Хочется верить, что с помощью искусственного интеллекта, робототехники и у нас есть огромный потенциал для их использования именно для блага людей.

В настоящее время имеет место новый качественный «скачок» в области развития технологий ИИ. Как говорят уже многие, наступает эра «сверхразума» – искусственный интеллект не только «уже лучше человека» во многих областях, но и «готов взять на себя» функции творчества, традиционно считающиеся прерогативой человека.

Другими словами, в последние годы наметился поворот от исследований *частного ИИ (когда система очень эффективно может решать достаточно узкую задачу, например, поставить диагноз какого-либо вида рака по изображению больного органа)* к работам по созданию *общего (сильного) искусственного интеллекта* (в принятой международной номенклатуре – AGI, Artificial General Intelligence) [8]. Под **общим (сильным) интеллектом** понимают человеческий интеллект, наделенный сознанием, т.е. практически естественный интеллект. Теория *сильного искусственного интеллекта* предполагает, что компьютеры смогут приобрести способность мыслить и осознавать себя, в то время как *теория слабого искусственного интеллекта* исключает такую возможность. Все разработанные и используемые в настоящее время системы ИИ относятся по сути дела к слабому интеллекту [9].

Под *сильным ИИ в логике* [10] понимается алгоритмическое воспроизведение высших когнитивных функций человека при решении творческих задач, связанных с извлечением и оперированием смыслами для широкого класса приложений на основе перспективных технологий ИИ. Это отличает его от традиционного («слабого») ИИ, работающего с базовыми

когнитивными функциями (распознавание речи, машинное зрение, вывод на приорных знаниях конкретной области и пр.).

Для сильного искусственного интеллекта есть ряд необходимых свойств:

- Принятие решений, использование стратегий, решение головоломок и действия в условиях неопределенности;
- Представление знаний, включая общее представление о реальности;
- Планирование;
- Обучение;
- Общение на естественном языке;
- Сила воли;
- И объединение всех этих способностей для достижения общих целей.

На данный момент «сильный» ИИ еще не создан, однако «слабый» ИИ, как мы показали, уже повсюду применяется в различных сферах деятельности человека. «Сильный» ИИ, действительно «мыслящий как человек», – дело весьма отдаленного будущего [11]. «Слабый», или узкий ИИ (Narrow AI), – это ИИ, предназначенный для решения только определённого класса задач, разрабатывается уже сегодня. Такой ИИ опирается на численные процедуры и сложные математические модели только определённой области знаний и не является универсальным («сильным»).

Таким образом, реально действующим сегодня и в среднесрочной перспективе является «слабый» (узкий) ИИ, который предназначен для решения узкоспециальных задач. Это позволяет прогнозировать сегодня социальный заказ для подготовки инженеров к использованию «слабого» ИИ на протяжении ближайших 15–25 лет.

Современный уровень развития технической базы вычислительной техники является фундаментальным препятствием для создания интеллекта, поскольку на нынешнем уровне невозможно воспроизвести психические процессы на искусственных носителях.

Если у ученых получится создать общий ИИ, такой продукт может стать копией человека: он сможет учиться выполнять абсолютно разные задачи и переносить свои знания из одной сферы в другую, будет мыслить критически, а еще кооперироваться с людьми или другими искусственными интеллектами, чтобы достичь своих целей. Удастся ли в ближайшей перспективе создать такую технологию, пока неизвестно, но, когда это произойдет, наш мир круто изменится [12].

Современные интеллектуальные системы используют целый стек технологий, который состоит из методов и алгоритмов искусственного интеллекта, фреймворков машинного обучения, а также инфраструктурных решений для их поддержки (облачные системы, специализированные аппаратные системы и др.). Обеспечение доверия к таким системам является

долгосрочным вызовом, активным поиском ответа на который мировое сообщество занимается уже несколько лет. Однако на текущий момент не существует научно-технологической базы для разработки высоконадежных доверенных и одновременно эффективных систем, использующих технологии искусственного интеллекта (интеллектуальных систем), в том числе отсутствуют инструменты для поиска новых видов уязвимостей и противодействия новым типам угроз, специфичным для этих технологий [13]. Работа над ответом на этот вызов ведется в двух встречных направлениях. Первое направление – выработка требований к интеллектуальным системам и разработка стандартов. Второе направление – создание научно-технологической базы, поддерживающей разрабатываемые стандарты.

В современном мире ИИ используется практически повсеместно, это создаёт предпосылки для нового толчка прогресса. Искусственный разум позволяет автоматизировать производство, а значит и продуктивность труда.

Информационные технологии, кибернетика имеют огромное число плюсов, но также имеют свои минусы, которые требуют очень пристального внимания человечества. *Эти минусы связаны с опасностью, возникающей при работе с искусственным интеллектом.*

■ Одни из проблем связаны с возможностью потери стимула людей к творческому труду. Всеми виной всеобщая компьютеризация и использование машин в сфере искусств. Но всё же стало ясно, что люди добровольно не отдадут самый квалифицированный творческий труд, так как он является привлекательным для самих людей.

■ Вторая группа проблем более серьёзна и состоит она в следующем. Уже в настоящее время существуют программы и машины, которые в процессе своей работы способны обучаться, то есть приспосабливаться к внешним факторам. В ближайшем будущем могут появиться машины, которые будут обладать таким уровнем надёжности и приспособляемости, что человеку не нужно будет вмешиваться в процесс. В таком случае человек перестанет выполнять свою функцию – функцию поиска решений.

Возможно, человек станет не способен адекватно реагировать на изменения внешних условий, а также возможно перестанет быть способен принять управление на себя в случае ЧС. Необходимо будет ввести некоторые пределы в автоматизации процессов, которые связаны с тяжёлыми аварийными ситуациями, тогда у человека, производящего надзор за управляющей машиной, всегда хватит реакции и умения правильно действовать в непредвиденной ситуации.

Подобные ситуации возможны в ядерной энергетике и транспорте. Стоит особенно отметить такую опасность в ракетных войсках, так как

ошибка может иметь ужасные последствия.

Оказывается, что, даже перепроверя и многократно дублируя, вероятность ошибок очень высока. Отсутствие контролирующего оператора может привести к фатальной ошибке.

Проблемы ИИ будут решаться людьми постоянно. Будут появляться всё новые и новые проблемы и, похоже, что этот процесс является бесконечным.

Таким образом, искусственный интеллект в дальнейшем должен сыграть большую роль в развитие человечества. В будущем искусственный интеллект будет использоваться не только в науке и производстве, но и станет неотъемлемой частью жизни каждого цивилизованного человека.

2. ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ И ИНЖЕНЕРНЫХ РАСЧЕТОВ

2.1. Общие сведения

В начале двадцать первого века человечество вступило в эру петафлопсных суперЭВМ, в ближайшее время ожидается появление эксафлопсных суперЭВМ. Эффект от ожидаемой в ближайшем десятилетии разработки «промышленного» квантового компьютера, прототипы которого уже существуют, даже трудно на сегодня оценить. Все это кардинальным образом изменит возможности и подходы к компьютерным технологиям моделирования физических процессов и явлений. Но, следует отметить такой важный факт: темпы развития прикладного программного обеспечения существенно отстают от роста суперкомпьютерных мощностей. Поэтому, актуальной задачей является разработка «философии» и собственно создание прикладного программного обеспечения нового поколения. А, вследствие еще и бурного развития технологий ИИ, *важную роль в разработке компьютерных технологий математического моделирования состояния и поведения физических и инженерных объектов различной природы приобретают технологии ИИ.*

В настоящее время в связи с активным внедрением средств вычислительной техники во все сферы научно-исследовательской и производственной деятельности человека коренным образом меняются подходы и идеология выполнения научных исследований и инженерной работы специалистов. *Компьютер сегодня представляет собой не просто инструмент для работы, а выполняет функцию интеллектуального помощника.*

Применение подходов и методов численного (компьютерного) анализа при выполнении исследований и решении прикладных задач механики и инженерии имеет уже внушительную историю развития. Исследователи еще на стадии развития компьютерной техники и технологий поняли, что компьютеры должны дать новый мощный импульс развитию научных технологий и инженерии. Но, до появления современных компьютеров и компьютерных технологий использование реалистичных математических моделей при выполнении сложных фундаментальных и прикладных исследований и расчетов было возможным лишь в достаточно ограниченных случаях.

Практически одновременно с появлением компьютеров начался и *процесс разработки и внедрения специальных компьютерных технологий для проведения сложных трудоемких расчетов, связанных с исследованиями разнообразных механических процессов и явлений,*

изучением механических систем и решением многочисленных задач прикладной механики. Одним из наиболее важных условий, которому еще на первоначальном этапе создания таких компьютерных технологий необходимо было неукоснительно следовать, являлось требование, чтобы компьютеры использовались не только как калькуляторы (т.е. вычислительные машины для быстрого и более-менее автоматизированного расчета). Поэтому необходимо было разработать новые методы и методики, использующие все преимущества нового инструментария – компьютерных технологий. Кроме того, благодаря своему главному качеству – обеспечению выполнения математических операций с огромным быстродействием – создавались возможности применения более адекватных реальным процессам математических моделей.

Наряду с развитием компьютерных технологий численного моделирования физических процессов и явлений, возрастают и требования к ним со стороны «заказчиков» (ученых, инженеров, конструкторов, технологов). Требуется увеличение адекватности и точности модельных исследований. А это означает постоянное совершенствование и усложнение методов численного моделирования и соответствующего программного обеспечения.

Следует отметить, что требования возрастают не только к компьютерным технологиям численного моделирования, но и к специалистам, которые решают на основе компьютерного моделирования прикладные задачи из различных областей механики и инженерии. Значение имеет не только большое разнообразие численных методов и алгоритмов и необходимость выбора наиболее эффективного для рассматриваемой задачи метода, технологий и собственно базовой математической модели. Важным аспектом является тот факт, что требования и пожелания «заказчиков» опережают возможности как численных методов, так и компьютеров. Поэтому квалификация «расчетчика» должна позволять ему выполнить «оптимальный» выбор для удовлетворения требованиям «заказчика» от постановки модельной механико-математической задачи, выбора численного метода и соответствующего программного обеспечения для этапа вычислений и до анализа и интерпретации результатов расчетов.

Одной из центральных в общей задаче механико-математического моделирования механических процессов является проблема построения определяющих уравнений, корректно и адекватно описывающих рассматриваемые процессы и явления. Очевидно, что в настоящее время уже нельзя пользоваться простейшими моделями описания механического состояния объекта. На сегодняшний день можно уже утверждать, что, используя специализированное программное обеспечение, специалист в

состоянии решить модельную математически сложную задачу из очень широкого диапазона прикладных процессов и явлений, но при условии, что ***модельная задача правильно сформулирована и корректно построена расчетная схема численного моделирования.*** Математическое моделирование на базе компьютерных технологий стало многофункциональным средством для решения практически любых задач теоретической и прикладной механики.

Итак, задачи современных фундаментальных и прикладных исследований, инженерных расчетов, связанных с изучением поведения и состояния сложных механических объектов (систем, конструкций, явлений) невозможно решать без активного использования компьютерных технологий. Причем, последние охватывают различные аспекты решения задач: от формулировки и постановки модельных механико-математических задач, построения расчетных численных схем до выполнения непосредственных вычислений, интерпретации и анализа результатов расчетов.

Сегодня возможности компьютерного моделирования качественно изменяются и одно из существенных новых требований к технологиям компьютерного моделирования – **наличие в таких технологиях элементов искусственного интеллекта (ИИ)**, или, что представляется более точным, присутствие таких «навыков», как «сообразительность», «понимание», «проницательность», «умение адекватного прогнозирования на основе обработки больших массивов информации и данных» (что и составляет суть понятия «artificial intelligence»).

Современные системы компьютерного моделирования должны «уметь» давать рекомендации к постановке модельных задач, уметь корректировать вычислительные алгоритмы, интерпретировать результаты вычислений и выдавать человеку возможный план действий.

Таким образом, в настоящее время одно из главных требований к технологиям компьютерного моделирования – такие системы должны «уметь» давать рекомендации на всех стадиях рассмотрения физического или технологического процесса.

Можно констатировать, что сегодня ***компьютерные технологии моделирования физических и производственных процессов это: высокопроизводительные вычисления + знания и «большие данные» + математические модели + искусственный интеллект.***

Необходимость внедрения элементов технологий искусственного интеллекта особенно остро стоит в настоящее время как при компьютерном моделировании технических систем ответственного назначения с целью создания их *цифровых двойников*, так и при управлении данных систем в процессе их эксплуатации.

Тенденции развития систем ИИ убедительно показывают, что дальнейшее развитие и эффективность использования технологий ИИ существенным образом определяется совместным использованием различных подходов и методов, таких, например, как: методы обработки информации и изображений, описания присутствующих на них объектов; методы анализа данных (обнаружение математических закономерностей, различающих классы объектов); методы и технологии компьютерного моделирования различных объектов и др. Для прогресса и развития систем ИИ необходимым условием является совместное согласованное развитие различных научных направлений.

ИИ обладает несомненным преимуществом при обработке больших объемов данных, но, как указывалось ранее, и считаем необходимым это еще раз подчеркнуть, не учитывает сопутствующие обстоятельства – так называемый контекст принятия решения. Во многих ситуациях это представляет собой критический фактор. В особенности это важно при принятии решений в нестандартных ситуациях по результатам выполненных расчетов, моделирования. Здесь весьма важным является интуиция и личный опыт эксперта. В этом направлении системы ИИ тесным образом связаны с системами приобретения знаний и получения решений.

2.2. Разработка прикладных систем на базе CAD&CAE-технологий

Практическое применение методов ИИ в *системах компьютерного моделирования* физических и технологических процессов (в широком смысле этого понятия), решения разнообразных инженерных задач все еще находится на стадии начального развития. Вместе с тем, проектирование, расчет и создание сложных технических систем остро требует методов анализа и поддержки принятия решений. Такие методы необходимы для решения междисциплинарных проблем техники, анализа больших данных и других задач.

Как указывалось уже ранее, среди направлений, определяющих разработку систем с элементами интеллектуального анализа, особую роль имеет такое направление как:

- Разработка компьютерных прикладных/инженерных систем/методик на базе современных CAD&CAE-технологий для реализации широкого диапазона прикладных моделей и расчетных методик для решения различных типов и классов инженерно-технических задач как на этапе проектирования технических объектов, так и их текущей эксплуатации.

Несколько замечаний, касающихся данного направления.

В настоящее время в технологические процессы разработки и производства промышленной продукции различного назначения активно

внедряются **CALS-технологии** (Continuous Acquisition and Life cycle Support – непрерывная информационная поддержка поставок и жизненного цикла изделий – информационные технологии, используемые в управлении процессами жизненного цикла изделия или системы, в основном для сложных (высокотехнологичных и наукоёмких) образцов продукции машиностроения и иных объектов техники). В CALS-системах предусмотрены подсистемы, поддерживающие хранение, обработку и передачу информации, оперативный доступ к данным. Эффективность CALS-технологий существенным образом базируется на использовании различных CAD/CAE/CAM/PDM-систем.

Несколько слов об указанных системах.

CAD-системы – Computer-aided design, системы автоматизированного проектирования. Автоматизированные системы, реализующие информационную технологию выполнения функций проектирования. Представляют собой организационно-технические системы, предназначенные для автоматизации процесса проектирования, состоящие из персонала и комплекса технических, программных и других средств автоматизации его деятельности для создания конструкторской и технологической документации, 2D- и 3D-моделей и чертежей. Для обозначения подобных систем широко используется аббревиатура САПР.

CAE-системы – Computer-aided engineering, системы инженерного анализа, автоматизации инженерных расчётов. CAE – общее название для программ и программных пакетов, предназначенных для решения различных инженерных задач: расчётов, анализа и симуляции физических процессов. Современные системы инженерного анализа (или системы автоматизации инженерных расчётов) (CAE) применяются совместно с CAD-системами (зачастую интегрируются в них, в этом случае получают гибридные CAD/CAE-системы). CAE-системы – это разнообразные программные продукты, позволяющие при помощи расчётных методов (метод конечных элементов, метод конечных разностей, метод конечных объёмов и др.) оценить, как поведёт себя компьютерная модель изделия в реальных условиях эксплуатации, помогают убедиться в работоспособности изделия, без привлечения больших затрат времени и средств. Расчётная часть CAE-пакетов основана главным образом на численных методах решения краевых задач, описывающих физические процессы (МКЭ, МКР, МКО и т.д.).

CAM-системы – Computer Aided Manufacturing, системы автоматизации технологических процессов при производстве изделия. CAM-система – автоматизированная система, либо модуль автоматизированной системы, предназначенный для подготовки управляющих программ при изготовлении изделия. Под термином

понимаются как сам процесс компьютеризированной подготовки производства, так и программно-вычислительные комплексы, используемые инженерами-технологами.

PDM-система – Product Data Management, система управления данными об изделии – организационно-техническая система, обеспечивающая управление всей информацией об изделии. При этом в качестве изделий могут рассматриваться различные сложные технические объекты (корабли и автомобили, самолёты и ракеты, компьютерные сети и др.).

Входящие в CALS-системы модули позволяют осуществлять управление проектом (PDM-система), выполнять двух- и трёхмерное проектирование инженерно-технических изделий (деталей, сборочных единиц и т.п.) (CAD-системы), производить инженерные расчёты, анализ, моделирование и оптимизацию проектных решений (CAE-системы), разработку технологических процессов, синтез управляющих программ для робототехнических комплексов и технологического оборудования с ЧПУ, моделирование процессов обработки, в том числе построение траекторий относительного движения инструмента и заготовки в процессе обработки, расчёт норм времени обработки и т.п. (CAM-системы).

CAD/CAE-системы представляют собой один из базовых элементов современных CALS-технологий и в общем систем автоматизированной интеллектуальной поддержки производственных и технологических процессов.

2.3. Анализ видеоинформации («Компьютерное зрение»)

В 21-м веке с развитием мощности компьютеров, появления интеллектуальных алгоритмов глубокого обучения системы анализа и синтеза информации, компьютерного зрения стали интенсивно развиваться и внедряться практически во всех сферах жизнедеятельности человека и общества.

Компьютерное зрение является важным и очень большим разделом ИИ, включающим распознавание образов и обработку изображений. Она включает в себя набор методов, которые наделяют компьютер способностью «видеть» и извлекать информацию из увиденного. Говоря более математическим языком, компьютерное зрение – это раздел информатики и искусственного интеллекта, разрабатывающий принципы, методы, технологии идентификации и классификации предметов, явлений, процессов, сигналов, изображений и ситуаций.

Распознавание любого объекта на изображении основывается на отличающих его признаках. Выбор набора признаков для распознавания часто определяется интуицией и опытом эксперта. Однако при

автоматизированном решении многих задач трудно использовать те признаки, которые визуально являются наиболее информативными для распознавания. Часто довольно сложно определить, какие именно признаки используются человеком при классификации тех или иных образов, а также гарантировать, что их использование будет эффективнее набора признаков, полученных математически. Решения данной проблемы «в лоб» не существует. Большое количество признаков увеличивает вычислительные затраты и часто приводит к ухудшению результатов распознавания. Дополнительные сложности возникают в связи с тем, что выбор признаков, т.е. настройка системы распознавания, производится на ограниченном числе образов, а в процессе работы системе приходится оперировать гораздо большим количеством образов, различающие признаки для которых, могут отличаться.

Как и другие типы ИИ, компьютерное зрение ориентируется на выполнение и автоматизацию задач, имитирующих человеческие возможности. В этом случае компьютерное зрение старается имитировать зрение и восприятие человека. Спектр практического применения технологий компьютерного зрения обуславливает то, что оно стало центральным компонентом множества современных инноваций и решений [12, 14].

2.4. Интеллектуальные системы компьютерного моделирования

Важное направление развития ИИ – *системы моделирования, анализа и поддержки принятия решений*. Для создания эффективных систем моделирования и поддержки принятия решений необходимо использовать математический аппарат, отражающий в детерминированной форме законы механики, методы вариационного исчисления и дедукции.

Один и тот же объект механики может быть описан различными механико-математическими моделями, исходя из цели, поставленной исследователем. Любая модель точна для одних классов/типов задач и не точна для других. К тому же, нужно знать на каких этапах наиболее эффективно использовать и какую именно модель, чтобы не допускать ошибок. Целесообразно отметить, что при построении математической модели рассматриваемой задачи механики и выборе метода ее решения необходимо руководствоваться не только принятой механической моделью поведения объекта, но и конечной целью решения задачи.

На сегодня компьютерное моделирование стало неотъемлемым элементом систем ИИ и успехи их очевидны. Вместе с тем, с задачей количественного соответствия результатов моделирования и данных по реальным процессам моделируемого процесса дело обстоит не столь успешно, так как достичь высокой количественной точности при

рассмотрении математических моделей крайне сложно. Причин такого положения достаточно много. Поэтому на сегодня *одной из важных задач является разработка, развитие и адаптация современных продвинутых подходов и методов математического и компьютерного моделирования для различных классов задач с элементами интеллектуального анализа.*

Использование технологий ИИ является эффективным практически на всех этапах механико-математического моделирования процессов и явлений механики. Среди них, например, интеллектуальная обработка, интерпретация и анализ больших объемов знаний и данных на основе математических моделей изучаемых объектов, процессов или явлений; разработка баз знаний по различным направлениям и предметным областям механики; разработка систем анализа результатов моделирования и принятия решений и др.

Главное назначение таких новых инноваций в технологии моделирования – существенное повышение эффективности (адекватность, точность, время выполнения собственно вычислений и т.д.) процесса моделирования от осмысления реальной задачи, формулировки модельной задачи и до анализа и интерпретации результатов расчетов, выдачи рекомендаций и предложений [15].

2.5. ИИ и инженерное дело

Практическое применение методов ИИ в инженерном деле на сегодня все же еще находится на «начальном» этапе. Но, очевидно, что прогресс в развитии технологий автоматизации процессов проектирования сложных технических систем остро требует использования *методов поддержки принятия инженерных решений.* Такие методы нужны для решения широкого спектра задач, возникающих при проектировании технических систем (междисциплинарность требуемых знаний, анализ больших объемов данных и целый комплекс других задач). Сегодня, для системной цифровой трансформации производств необходимо решение задач, связанных с поддержкой и даже автоматизацией воспроизведения высших когнитивных функций специалистов, связанных с креативным принятием решений [10].

Например, горное дело является благодарной областью приложения технологий ИИ, поскольку не только опирается на эффективные средства сбора и первичной интерпретации данных с контролируемым качеством, но и имеет альтернативные источники априорных знаний (причинно-следственные математические модели, инженерное ПО, экспертные обобщения, справочники, руководства, регламенты и пр.). Это позволяет гибко сочетать подходы *ИИ на данных* с технологиями *ИИ на знаниях*, что в ряде случаев является ключевым для приближения к возможностям *сильного ИИ.*

В горном деле практически все классы задач, относящиеся к деятельности предметных специалистов (геолог, маркшейдер, проектировщик, конструктор, технолог и др.), являются перспективными с точки зрения активного использования технологий ИИ.

Специфика горнодобывающего производства обуславливает большую вариативность технологических и сопутствующих процессов даже в пределах одного горнодобывающего предприятия. Как следствие, прямое тиражирование прикладных систем ИИ даже между разными рудниками/шахтами одного предприятия требует значительных усилий *специалистов в области машинного обучения*. Поэтому применение технологий машинного обучения (МО), позволяющих существенно сократить временные и ресурсные затраты, в данном случае является приоритетным. Особенно это важно для решения задач автоматического моделирования и проектирования, когда модели ИИ на данных комбинируются с объектами прикладной математики, реализуемыми классическими численными методами моделирования и оптимизации.

Несмотря на существенный задел в области специализированных предметных систем автоматизированного проектирования, в подавляющем большинстве на текущий момент времени такие решения основаны на воспроизведении лучших практик специалистов-конструкторов. Однако переход к *парадигме генеративного дизайна, имитирующей логику конструктора* не на основе априорных знаний, а за счет использования альтернативных процессов рационального выбора (например, эволюционных вычислений), позволяет формировать нестандартные конструкторские решения, часто превышающие когнитивные возможности человека.

Цифровая трансформация производств неизбежно приводит к усложнению условий формирования и принятия решений за счет увеличения объема учитываемых факторов. В этих условиях даже решения, принимаемые квалифицированным специалистом и даже группами специалистов могут считаться лишь условно адекватными и достоверными. Потому принципиальным является развитие алгоритмов ИИ, которые обеспечивают *автоматическую эффективную валидацию и верификацию проектных, технических и управленческих решений*. Это дает основу для создания «самообъясняющихся» систем ИИ, способных интерпретировать решения, как принятые человеком, так и рекомендованные иными системами ИИ.

Процессы цифровой трансформации крупных предприятий, как правило, ставят своей целью объединение возможностей различных (в том числе, уже имеющихся) цифровых систем, формирующих процессы подготовки и принятия решений. Это включает в себя совместное

использование традиционного моделирования, методов оптимизации и прикладных систем ИИ, в том числе построенных на совершенно разных принципах. Как следствие, это требует развития технологий композитного ИИ, позволяющих комбинировать различные системы ИИ, а также строить на их основе новые решения с заданным функциональным назначением.

3. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ОБРАЗОВАНИИ

3.1. Общие сведения

При всём многообразии мнений об ИИ, на текущем этапе его развития все же актуальной представляется задача достаточно четкого определения, «какой ИИ возможен сегодня и в ближайшей перспективе, и чему именно можно и нужно обучать студентов уже сейчас» [16].

Технологии ИИ изменяют в настоящее время образовательную реальность может быть даже в большей степени, чем это имело место при активном внедрении Интернета в повседневную жизнь.

Существенный толчок к внедрению технологий ИИ в процессы обучения в вузах произошел во время пандемии Коронавируса и повсеместного внедрения дистанционных форм обучения.

Одна из наиболее обсуждаемых в обществе тем в настоящее время, связана с дискуссией о том, представляют ли системы ИИ реальную угрозу развитию и существованию человечества, какие угрозы и риски связаны с таким интенсивным прогрессом в разработке технологий ИИ? Как внедрение ИИ в «реальную жизнь человека» повлияет на его существование? Следует отметить, что весьма сильный акцент на поисках «отрицательных сторон» ИИ, к сожалению, в некоторой степени даже преобладает над положительными возможностями и потенциалом, который открывается перед человечеством при логически выверенном, базирующемся на научном подходе использовании технологий ИИ.

Сегодня наблюдается удивительный феномен создания и массового распространения чат-ботов ИИ, использующихся в том числе и в образовании.

Наиболее ярким представителем таких чат-ботов в настоящее время является ChatGPT. ChatGPT представляет собой систему, построенную на основе архитектуры глубоких нейронных сетей. Она обучена на огромном количестве текстов, поэтому может отвечать на разные вопросы и помогать людям в поисках необходимой информации. ChatGPT представляет собой бесплатный сервис, реализованный в виде классического чат-бота. В основе платформы лежит нейросеть, разработанная специалистами компании OpenAI, которая принадлежит обширному семейству моделей GPT (сокращение от Generative Pre-trained Transformer - генеративный предварительно обученный трансформер). Основное направление данных нейросетей – генерация текстов. ChatGPT сегодня – одна из самых крупных и продвинутых систем обработки естественного языка в мире. По заявлению разработчиков она «может быть использована для решения любых задач на английском языке» (очевидно, что «любых задач», достаточно преувеличенно, но, очевидно, что спектр применения нейросетей будет

только возрастать).

Отметим, что речь идет об эффективности и «(не)полезности» использования систем, подобных ChatGPT, при обучении студентов – в процессе приобретения знаний; обучению умению самостоятельно мыслить, анализировать исходную информацию и результаты расчетов; приобретать навыки к творчеству, изобретательству и т.п.

Подчеркнем, что мы считаем, что **потенциальные возможности, которые открываются при внедрении нейросетевых систем в систему обучения студентов и высшему образованию в целом, сегодня уже невозможно игнорировать.** Нейросетевые системы, чат-боты представляют собой «своеобразные триггеры», которые заставляют с новых позиций «посмотреть на подходы к системе образования», инициируют разработку и внедрение новых технологий в образовании.

Интересным представляется «собственный» ответ ChatGPT на вопрос «какое негативное воздействие на обучение может оказывать ChatGPT?». Ответ был такой: «Хотя *инструмент* может дать быстрые и простые ответы на вопросы, он *не развивает навыки критического мышления и решения проблем, которые необходимы для успеха в учёбе и на протяжении всей жизни*» [17].

Ответ ChatGPT достаточно содержателен и содержит ключевую фразу «не развивает навыков критического мышления и решения проблем», что представляет собой одну из основных целей высшего образования – научить студентов мышлению, логики, самостоятельности при решении сложных профессиональных задач и проблем.

Весьма показателен в этом плане критический анализ процесса подготовки своей выпускной работы с помощью чат-бота ChatGPT1, выполненный собственно студентом РГГУ [18]. В статье [17] весьма четко выделено несколько важных моментов, которые мы интерпретируем здесь.

Во-первых, в процессе подготовки работы принимают участие три «субъекта»: студент, его научный руководитель и чат-бот ChatGPT. То есть это не процесс «постановка задачи – передача данных чат-боту – получение результата», а своеобразный процесс творчества и создания выпускной работы.

Во-вторых, необходимо «правильно формулировать задачи для ИИ и задавать правильные вопросы на всех этапах подготовки работы». Важное обстоятельство здесь состоит в том, что, если человек не обладает хотя бы минимальным запасом знаний в предметной области, ему вряд ли получится корректно сформулировать постановку задач для ИИ и оценить правильность модельных задач, которые предлагает к решению ИИ.

В-третьих, на этапе решения задачи необходимо следить за процессом решения (выполняемыми операциями, действиями) и обязательно

проверять источники, которые использует нейросеть при получении решения.

Очень важный факт: нейросеть может содержать и реальную проверенную информацию, и ненадежную «сомнительную» информацию, кроме того, она может «выдумывать» собственную. В этом случае ChatGPT будет с полной уверенностью рассказывать несуществующие факты и ссылаться на ею же выдуманные источники или сомнительные, ненадежные.

Замечание. Такое свойство ChatGPT и подобных систем ИИ называется «галлюцинации» [17].

Очевидно, чтобы оценить процесс решения задачи и использованные при этом методы и подходы, необходимо опять же самому обладать определенными знаниями в данной предметной области. В противном случае алгоритм и процесс решения может выглядеть весьма «убедительным», но быть неправильным.

В качестве показательных примеров можно сослаться на выполненные нами со студентами отделения «Механика. Математическое моделирование», механико-математического факультета БГУ эксперименты по решению задач аналитической механики для систем со многими степенями свободы с использованием уравнений Лагранжа второго рода и задач механики материалов для статически неопределимых балок. Во многих случаях чат-бот ChatGPT решал задачи весьма правдоподобно, но неправильно, если его «вовремя» не подкорректировать.

В приложении изложены результаты некоторых наших экспериментов по использованию ChatGPT для решения задач механики. Здесь приведем некоторые общие выводы, из выполненных экспериментов, связанных с решением задач аналитической механики и сопромата.

Наиболее важное заключение состоит в том, что анализ решения рассмотренных задач сопротивления материалов и аналитической механики показал, что эффективным подходом к решению сложных задач в области аналитической механики и сопротивления материалов с помощью ИИ является «диалог оператора с ИИ» с разбиением задач на более простые подзадачи и формулировке ИИ конкретных четких вопросов.

ChatGPT является языковой моделью, обученной на большом объеме текстовых данных. Его главными преимуществами при решении задач сопромата и аналитической механики являются:

1. Большой объем знаний: ChatGPT имеет доступ к огромному объему информации, включая технические справочники, учебники, научные статьи и другие источники знаний, в частности, по сопромату и аналитической механике. Благодаря этому, модель может предоставить подробные и точные ответы на широкий спектр вопросов в этих областях.

2. Быстрая обработка информации: ChatGPT может анализировать и синтезировать большие объемы текста за короткое время. Это позволяет ему быстро извлекать

необходимую информацию из сформулированных вопросов и предоставлять ответы с минимальными задержками. Модель также способна обрабатывать большое количество вариантов решения задач и предоставлять сравнительные анализы для выбора оптимального подхода к решению.

3. Гибкость и адаптивность: ChatGPT может адаптироваться к различным стилям формулировки вопросов и умеет работать с различными форматами данных. Он может понимать текстовые описания задач, формулы, графики и другие визуальные материалы. Благодаря этому, модель может предоставлять пошаговые решения задач, а также объяснять основные принципы и концепции, лежащие в основе данных задач.

Вместе с тем, у ChatGPT есть определенные существенные недостатки:

1. Ограниченная практическая экспертиза: Хотя ChatGPT обладает широкими теоретическими знаниями в области сопромата и аналитической механики, у него отсутствует практический опыт и интуитивное понимание, которое характерно для опытных инженеров и специалистов. Это означает, что модель может предоставлять теоретически верные ответы, но они могут быть не всегда применимы или оптимальны в реальных практических ситуациях.

2. Недостаточное понимание контекста задачи: ChatGPT не всегда может полностью понять сложность и особенности конкретной задачи, в частности по сопромату. Он рассматривает вопросы изолированно и может упускать некоторые важные аспекты, которые могут быть существенными для правильного решения задачи. Это требует более точной формулировки вопроса и уточнений для достижения более точных результатов.

3. Ограниченный доступ к реальным данным и случаям из практики: ChatGPT основан на текстовых данных, используемых в его обучении, и не имеет прямого доступа к реальным данным или случаям из практики. Это может ограничить его способность адекватно учитывать различные факторы, такие как материалы, геометрия конструкции, условия нагрузки и другие реальные ограничения, которые могут влиять на решение задачи.

4. Неполнота и возможность ошибок: Хотя модель стремится предоставить точные и полезные ответы, она все равно может допускать некоторую неполноту или содержать ошибки. Это может быть связано с ограниченными знаниями или ошибками в данных, на которых была обучена модель. Поэтому всегда рекомендуется использовать полученную информацию вместе с проверенными источниками и применять собственное критическое мышление.

И, *в-четвертых*, необходимо четко представлять себе, что серьезного аналитического внутреннего понимания моделируемых процессов у ИИ нет. Поэтому, выводы и анализ полученных решений должен выполнять человек!

Основной вывод в работе [18], который подчеркивается и в [17], заключается в том, что *идеальная схема работы с системами, подобными ChatGPT, состоит в том, что ИИ следует доверять выполнять конкретные задачи, а функции постановки задач, формулировки целей, анализа, обобщения, корректировки и редактирования промежуточных и конечных результатов выполнять самостоятельно исследователю.*

По сути работа нейросетей состоит в обработке больших (огромных) массивов данных на основе различных методов, подходов, технологий.

Говорить о генерации знаний и информации собственно самим компьютером можно только в переносном смысле. Поэтому, когда, например, мы формулируем системе ИИ задачу о выдаче «правила, закона на основе некоторых данных», то речь идет об обработке, на базе заданных алгоритмов, данных, которые в наибольшей степени релевантны поставленному вопросу. То есть важным является правильная формулировка задания системе человеком.

Мнения ученых по поводу использования систем ИИ типа ChatGPT в научных исследованиях достаточно резко расходятся.

Так, к «положительным» возможностям систем ИИ можно отнести [19]:

- помощь в анализе и интерпретации больших массивов данных;
- выдача рекомендаций и предоставление соответствующей информации для принятия решений;
- разработка «стандарта» для описания и предоставления научно-технической и инженерно-технической информации, для доступа и корректного понимания информации широким кругом специалистов из различных стран мира.

Но, системам, подобным *ChatGPT*, на сегодня присущ еще целый ряд «изъянов», которые не позволяют говорить о высокой эффективности их использования в серьезных научных исследованиях. К таковым можно отнести, например [17]:

- *Ненадёжность данных.*

Так, *ChatGPT* обучен на огромном массиве данных, которые сами по себе не проверены на достоверность и актуальность. Кроме того, данные в системы *ChatGPT* «достаточно быстро устаревают», то есть требуется постоянное обновление базы данных, удаление устаревших и сомнительных и обучение системы на новых знаниях. Так как алгоритмы обучения системы *ChatGPT* разрабатываются человеком, поэтому, как и человеку, системе присущи такие черты, как субъективность, предвзятость, неточность и ошибочность. Будучи обученным на текстах, написанных людьми, *ChatGPT* с неизбежностью воспроизводит человеческие изъяны. Так как алгоритмы носят общий характер и «не привязаны» к конкретному контексту, поэтому рекомендации *ChatGPT* могут оказаться неадекватными данному контексту.

- *Ложный контент.*

ChatGPT не только опирается в своей работе на ненадёжные источники, но и способен создавать собственные ложные содержания.

- *Ограничения.*

Можно сказать, что *ChatGPT* особенно силён в том, чтобы создавать обманчивое впечатление о своих возможностях.

В некоторых научных дисциплинах передача рутинных операций искусственному интеллекту, сокращение в этой связи времени реализации исследований и подготовки публикации, позволит учёным сосредоточиться на самой научной работе. Однако кажущаяся скорость представления результатов работы ChatGPT нивелируется необходимостью проверки и обоснования того, что «написал» ИИ. *Для серьёзной научной работы все данные, рекомендации и прочие результаты, представленные ChatGPT, требуют тщательной перепроверки, что может занять значительное время.* Кроме этого, стиль ответов ИИ затрудняет их интерпретацию из-за отсутствия обоснования, доказательств, а часто и логики.

– Авторство.

Весьма непросто представляется вопрос о том, является ли статья, подготовленная при помощи ИИ, оригинальным текстом? Что считать «помощью ИИ» при написании статьи? Очевидно, что использование ИИ вовсе не означает, что эта система обязательно была использована недобросовестно. Так, многие системы ИИ хорошо зарекомендовали себя с редактированием текста, переводом научно-технического текста на иностранный язык. Исходя из «здоровой логики», наверное, необходимо открытое заявление авторов об использовании систем ИИ.

3.2. О подготовке кадров в области ИИ

Подготовка кадров представляет один из ключевых вопросов развития ИИ, поскольку именно подготовка кадров к использованию ИИ в профессиональной деятельности позволит избавиться от разочаровывающих «зим» и направить развитие ИИ в результативном направлении.

Замечание. «Зима ИИ» – длительный период разочарования из-за неоправданных ожиданий и отсутствия достигнутых целей.

Для реализации поставленных целей и ускоренного внедрения технологических решений на основе ИИ в различные отрасли экономики и сферы общественных отношений важнейшей задачей является обеспечение «рынка технологий» искусственного интеллекта квалифицированными кадрами, спрос на которых все время возрастает. Особая роль в этих процессах принадлежит университетам, которые ведут образовательную деятельность по подготовке высококвалифицированных кадров в области технологий ИИ. Именно кадры и образование становятся ключевыми факторами развития цифровой экономики.

Важно, что будущий инженер должен применять уже освоенные знания и навыки в рамках своей специальности, используя систему поддержки принятия решений, которая является одним из направлений развития ИИ.

Широко применяемая ныне в одном из направлений ИИ, а именно в так называемых «рекомендательных системах», методология машинного обучения, основанная на выявлении закономерностей в больших данных и перенесении этих закономерностей на другие данные (индукция), больше подходит к социальным приложениям (реклама, маркетинг, социология и т.д.). В таких приложениях значительную роль играют трудно учитываемые субъективные предпочтения, которые невозможно обобщить в виде детерминированных закономерностей. Адекватным подходом в таких случаях представляется применение стохастических моделей, статистики и теории вероятности. Здесь и заложена основная идея машинного обучения – систематизация информации о статистических и случайных связях. В точных науках, где накоплен огромный массив знаний в виде законов, т.е. математических выражений, возможно создание систем поддержки принятия решений на основе законов природы.

Студент (специалист) должен принимать содержательные решения в конкретной предметной области при поддержке ИИ, который будет выявлять неочевидные зависимости в сложных системах, помогая тем самым обучающемуся в принятии решений, осознании сложности многообразных связей в сложных технических системах. Таким образом обучающийся и приобретает навыки взаимодействия с ИИ.

В процессе обучения ИИ у обучающихся должны не только закрепиться известные компетенции, но и сформироваться новые компетенции, позволяющие реализовать междисциплинарный подход к решению инженерных проблем оптимизации решений.

Обучение ИИ может осуществляться как в рамках отдельной дисциплины, так и в традиционных дисциплинах. Подготовка к использованию ИИ обеспечит современный уровень высшего технического образования. Студенты благодаря ИИ приобретают системное видение инженерной деятельности в её целостности, междисциплинарных взаимосвязях. Им демонстрируется процесс трансформации проектных решений для соответствия нормам и оптимальности.

Для того чтобы реализовать национальные стратегические задачи в сфере ИИ и стать одним из международных лидеров в развитии ИИ, важно не только увеличивать объёмы подготовки ИИ-специалистов, но и ориентировать образовательные программы вузов на актуальные запросы рынка технологических решений в сфере ИИ, а также отраслей экономики и социальной сферы, где эти решения применяются.

3.3. Некоторые выводы

Система образования является достаточно инерционной системой. Образовательный процесс в своей сущности и содержательной форме

претерпевает изменения достаточно медленно. Кроме того, «восторженные и фантастические» проекты на существенные и даже радикальные изменения в высшем образовании, которые обязательно появляются при активном развитии и внедрении новых прорывных инженерно-технических, технологических, ИКТ инноваций в большинстве своем остаются на уровне проектов.

Но, как мы уже неоднократно подчеркивали ранее (см., например, [20]), сейчас все же представляется весьма важным рассмотрение вопроса о целях высшего образования и средствах их достижения на современном этапе развития общества. С прогрессом в области разработки систем ИИ и все более глубоким проникновением систем ИИ в образовательный процесс, с расширением форм дистанционного обучения необходимо «уточнить» или дать новую интерпретацию, определение таким понятиям, как знание, образование, университет и т.п.

Запретить искусственный интеллект, тем более в образовании, невозможно, а системы прокторинга (*прокторинг* – система контроля на онлайн-экзамене/тестировании; контроль осуществляет проктор – администратор, который наблюдает за процессом через веб-камеру) всегда будут отставать от инноваций, новых технологий. Кроме запретов и контроля, следует серьезно задуматься о легитимном использовании систем ИИ в учебном процессе.

Перспективным направлением является использование систем ИИ в технологиях «персонализации образования», персональных помощников в образовательном процессе. Системы, подобные ChatGPT открывают новые перспективы в «самообразовании», при овладении новыми знаниями. Для преподавателей такие системы – эффективный инструмент для подготовки к лекциям, разработки нового курса.

Облачная робототехника, машинное обучение, компьютерное зрение, распознавание речи — все эти аспекты ИИ развиваются и продвигаются, порой делая значительные шаги в конкретных областях. Однако у ИИ по-прежнему пока нет ничего человеческого.

Даже если роботы, используя ИИ и человеческую инженерию, когда-нибудь смогут приблизиться к нашей ловкости, они никогда не смогут по-настоящему понять находящийся вокруг них мир во всей его хрупкости и потенциале. Контекст и изобретательность останутся в сфере людей. Технология сама по себе не является ни плохой, ни хорошей — все зависит лишь от того, как мы ее используем. Хочется верить, что с помощью искусственного интеллекта, робототехники и у нас есть огромный потенциал для их использования именно для блага людей.

4. МАШИННОЕ ОБУЧЕНИЕ

4.1. Введение

В 1959 году специалист по вычислительной технике из компании IBM Артур Самуэль написал компьютерную программу для игры в шашки. Каждому положению на доске присваивался определенный вес, базирующийся на вероятности выигрыша. Изначально вероятность определялась по формуле, в которой учитывались такие факторы, как количество шашек на каждой стороне и количество дамк. Подход работал, но Самуэль придумал, каким образом можно повысить его эффективность. Сыграв с программой тысячи партий, он использовал их результаты для уточнения позиционных весов. К середине 1970-х годов программа достигла уровня хорошо подготовленного непрофессионального игрока. Таким образом, Артур Самуэль написал компьютерную программу, которая могла по мере накопления опыта улучшать свои собственные результаты. То есть **программа училась**. Этот этап можно считать зарождением **машинного обучения (ML - machine learning)**.

Как отмечает академик РАН Семенов А.Л. [3], уже в 1960-е годы была, в основном, решена проблема построения систем компьютерной алгебры, но принципиальный прогресс в интуитивном решении широкого круга задач распознавания был достигнут только в XXI веке благодаря **алгоритмам машинного обучения**.

Машинное обучение называют самой перспективной и сложной областью ИИ.

4.2. О понятии машинного обучения

Что понимается под термином «машинное обучение» и как устроены алгоритмы обучения?

Единого определения для **машинного обучения (МО)** пока нет. Но большинство исследователей *формально определяют машинное обучение* примерно так: Машинное обучение — это наука о том, как «заставить» ИИ учиться и действовать как человек, а также сделать так, чтобы он сам постоянно улучшал свое обучение и способности на основе предоставленных нами данных о реальном мире.

Изначально компьютеры использовались для задач, алгоритм решения которых известен человеку. И только в последние годы пришло понимание, что они могут находить способ решать задачи, для которых алгоритма решения нет или он не известен человеку. Так появился искусственный интеллект в широком смысле и технологии машинного обучения, в частности.

Поэтому, достаточно *«классическое» определение машинного обучения* звучит так: МО – подраздел искусственного интеллекта, изучающий методы построения алгоритмов, которые приобретают способность эффективно решать задачи в процессе систематического обучения на множестве сходных задач.

Определение машинного обучения от Tom Mitchell: говорят, что программа обучается на опыте E для решения задачи T с эффективностью P , если ее эффективность решения задачи T , измеряемая как P , увеличивается с опытом E .

Замечание. Том Митчелл (Tom Michael Mitchell) – американский учёный, профессор Университета Карнеги-Меллон, основатель первой в мире кафедры машинного обучения и автор первого учебника по этому предмету.

В машинном обучении используются алгоритмы, которые обучаются на основе ранее полученного опыта. Такая компьютерная программа улучшает свои результаты по мере того, как получает все новые и новые примеры. Тогда можно надеяться, что, если вы «закинете» в этот механизм достаточное количество данных, он научится распознавать образы и выдавать разумные результаты уже для новых входных данных.

«Простыми словами», смысл машинного обучения заключается в том, что базовые алгоритмы могут предоставить информацию о наборе данных без написания кода для решения этой проблемы вручную. Вместо написания кода, вы предоставляете данные для базового алгоритма, а он формирует собственные выводы на основе этих данных.

Если говорить кратко, машинное обучение заключается в извлечении знаний из данных.

С помощью машинного обучения ИИ может анализировать данные, запоминать информацию, строить прогнозы, воспроизводить готовые модели и выбирать наиболее подходящий вариант из предложенных.

Так, сегодня такие системы в особенности распространены там, где необходимо выполнять огромные объемы вычислений: банковский расчет кредитного рейтинга, аналитика в области маркетинговых и статистических исследований, бизнес-планирование, демографические исследования, инвестиции, поиск фейковых новостей и мошеннических сайтов и др.

Сегодня ключевые исследования сфокусированы на разработке машинного обучения с эффективным использованием данных – то есть систем глубокого обучения, которые могут обучаться более эффективно, с той же производительностью, за меньшее время и с меньшими объемами данных. Такие системы востребованы в персонализированном здравоохранении, обучении роботов с подкреплением, анализе эмоций и др.

4.3. Как связаны машинное обучение, нейросети и глубокое обучение

Машинное обучение представляет собой одну из областей искусственного интеллекта (ИИ). Его цель заключается в том, чтобы *научить компьютеры обучаться самостоятельно*. С помощью алгоритма обучения машина может определять закономерности в указанных данных, выполнять построение модели и предсказывать вещи без явно запрограммированных правил и моделей.

Напомним, что *искусственный интеллект* – это название всей области (как биология, химия, геология). ИИ – способность компьютерной системы имитировать такие когнитивные функции человека, как обучение и решение задач. Посредством ИИ, компьютерная система использует математические функции и логику для имитации процессов мышления, которые позволяют людям обучаться на новой информации и принимать решения.

Машинное обучение — это раздел искусственного интеллекта. Важный, но не единственный (рис.4.1).

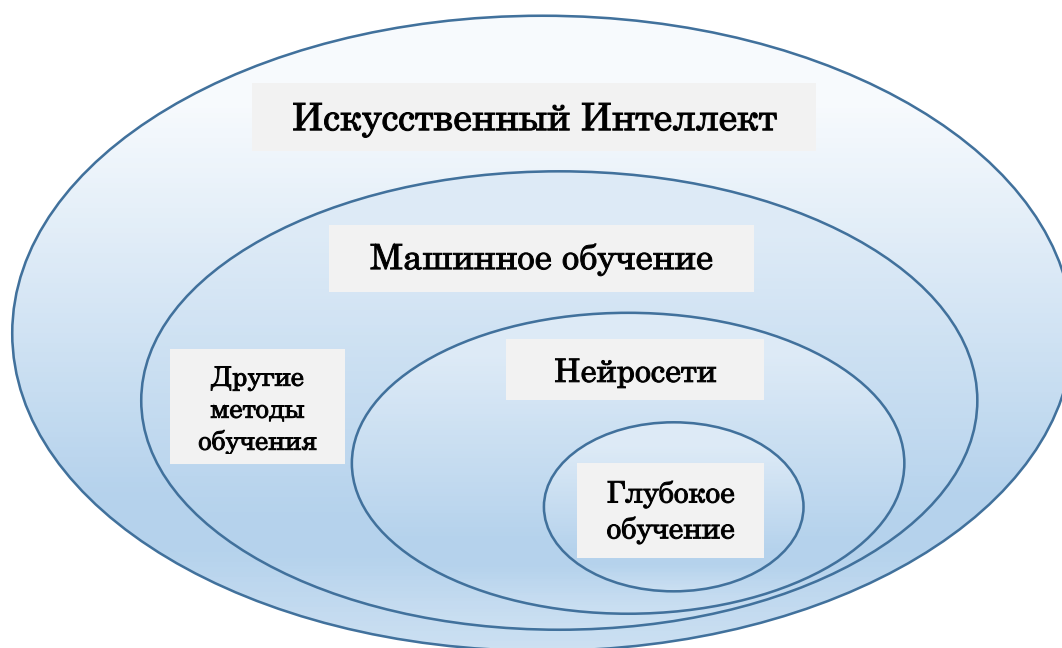


Рис.4.1. Взаимосвязь МО, нейросетей и глубокого обучения

Так как машинное обучение – одно из направлений применения ИИ, поэтому можно сказать, что *Нейросети* – один из видов машинного обучения. Популярный, но есть и другие.

Что касается понятия «глубокое обучение», то *Глубокое обучение* – это один из видов архитектуры нейросетей, один из подходов к их построению и обучению.

На практике сегодня мало кто отличает, где глубокие нейросети, а где «не очень». Сегодня говорят обычно уже только о названии конкретной сети

и всё. Глубокое обучение также включает в себя исследование и разработку алгоритмов для машинного обучения. В частности – обучения правильному представлению данных на нескольких уровнях абстракции. Системы глубокого обучения за последние десять лет добились особенных успехов в таких областях как обнаружение и распознавание объектов, преобразование текста в речь, поиск информации.

Поэтому, сравнивать можно только вещи одного уровня, иначе получается ситуация типа «что лучше: машина или колесо?».

4.4. Три составляющие МО

Цель машинного обучения – предсказать результат по входным данным. Чем разнообразнее входные данные, тем проще машине найти закономерности и тем точнее результат.

Для «обучения машины» необходимы три компонента (рис.4.2) [21]:

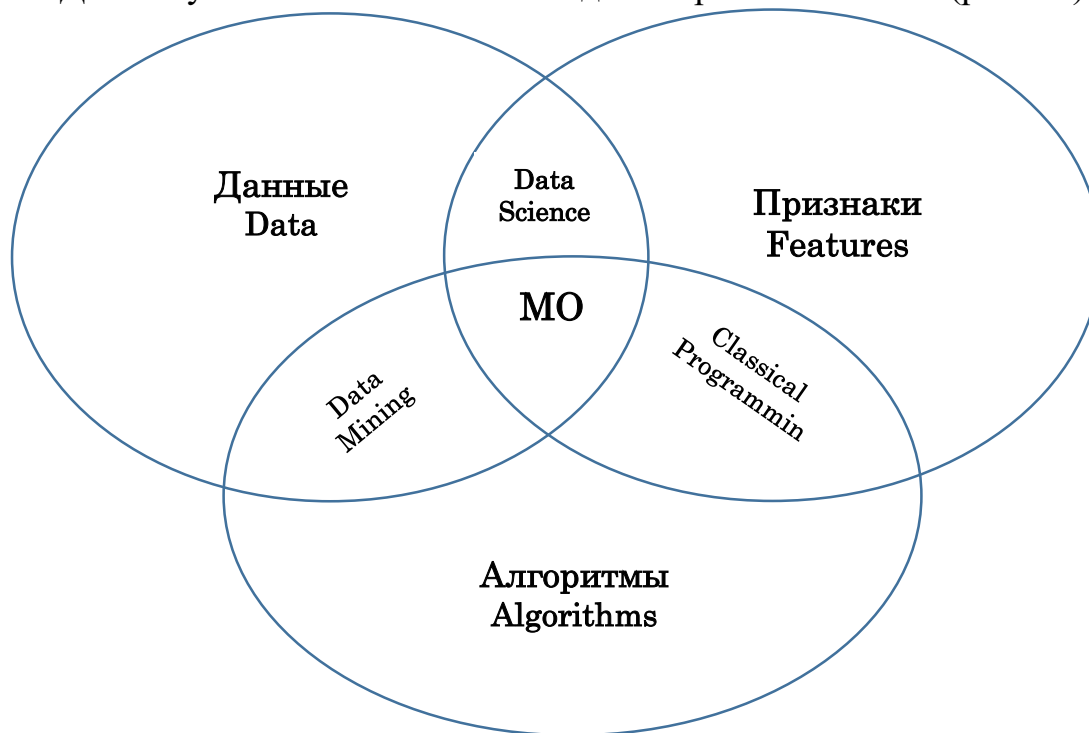


Рис. 4.2. Основные компоненты МО

1. Данные

Примеры: Если необходимо определять спам – нужны примеры спам-писем; если предсказывать курс акций – нужна история цен; если необходимо узнать интересы пользователя, то надо иметь его «лайки», посты.

Данных нужно как можно больше. Десятки тысяч примеров – самый минимальный набор данных.

Данные собираются различными способами:

- «вручную» (долго, зато без ошибок);
- автоматически (все данные, «без предварительной разборки», передаются машине для обработки);
- самые «хитрые» системы (типа «гугла») используют своих же пользователей для бесплатной разметки (например, ReCaptcha, которая требует «найти на фотографии все дорожные знаки»).

Замечание. За хорошими наборами данных (датасетами) идёт большая охота. Датасеты раскрывают свои алгоритмы крайне редко.

2. Признаки («фичи», features)

Примеры «фичей», свойств, характеристик, признаков: пробег автомобиля, пол пользователя, цена акций и т.д.

Но, компьютер должна знать, на что ему конкретно «смотреть». Оптимально, когда данные структурированы. Например, собраны в табличках. В этом случае названия колонок и есть «фичи».

В случае, когда информация не систематизирована (например, имеется большое множество картинок) и признаков много, то модель работает медленно и неэффективно. Отбор правильных признаков занимает много времени, зачастую даже больше, чем всё остальное обучение.

Опасной является ситуация, когда алгоритм «сам решает», какие признаки следует отобрать как «правильные». В этом случае в модель вносится субъективность и «она начинает дико врать».

3. Алгоритм

Практически всегда задачу можно решить разными методами. От выбора метода зависит точность, скорость работы и размер готовой модели.

Но если данные ненадежные, ошибочные, недостоверные, то даже самый лучший алгоритм не поможет. Поэтому в приоритете сбор надежных и проверенных данных, причем большого объема.

Отметим, что, как следует из рис.4.2, на «пересечении» двух пар выделенных компонент располагаются, в свою очередь, важные разделы.

4.5. Машинное обучение и «традиционное» программирование

Подходы к решению задачи в области машинного обучения и традиционного программирования разные [22].

А. Покажем это на задаче реализации спам-фильтра.

Каким образом можно *реализовать фильтр спама с применением приемов традиционного программирования?*

1 шаг. Сначала необходимо изучить задачу. Изучая спам, можно заметить, что в теме сообщения в большом количестве встречаются определенные слова и фразы.

2 шаг. Возможно, мы бы обратили внимание на наличие нескольких шаблонов (паттернов) в имени отправителя, в теле сообщения и в остальных

частях письма. В связи с этим, следует написать алгоритм обнаружения для каждого замеченного шаблона (паттерна) (рис. 4.3).

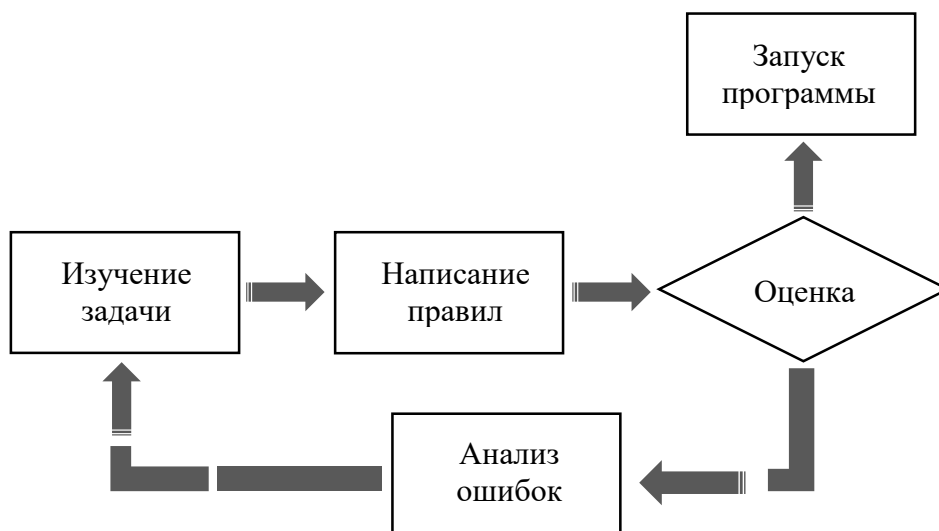


Рис. 4.3. Блок-схема алгоритма фильтр спама с применением приемов традиционного программирования

3 шаг. Тогда программа, написанная в соответствии с алгоритмом, будет маркировать (помечать) сообщения как спам в случае выявления некоторого числа таких паттернов.

4 шаг. Программу необходимо протестировать и повторять шаги 1 и 2 до тех пор, пока программа не станет достаточно хорошей и эффективной.

Поскольку задача нетривиальна, поэтому в программе с высокой вероятностью появится длинный список достаточно сложных правил, который довольно трудно сопровождать.

В противоположность этому подходу, *фильтр спама, основанный на приемах машинного обучения, автоматически узнаёт, какие слова и фразы представляют собой хорошие маркеры (прогнозаторы) спама, обнаруживая необычно часто встречающиеся шаблоны слов внутри примеров спам-сообщений в сравнении с примерами нормальных сообщений. При этом программа оказывается гораздо более короткой, легкой в сопровождении и более точной (рис.4.4).*

Теперь предположим, что «спамеры» выяснят, что все их сообщения, которые содержат, например, текст «4U» блокируются. Вместо этого они могут начать писать текст «For U». В этом случае фильтр спама, построенный с использованием приемов традиционного программирования, потребуется обновить, чтобы он маркировал сообщения с текстом «For U». Таким образом, если «спамеры» продолжают обходить написанный фильтр спама, то придется постоянно писать новые правила. Напротив, фильтр спама, основанный на приемах МО, автоматически заметит, что фраза «For U» стала необычно часто встречаться в сообщениях,

маркированных пользователями как спам, и начнет маркировать их без вмешательства с нашей стороны (рис. 4.5).

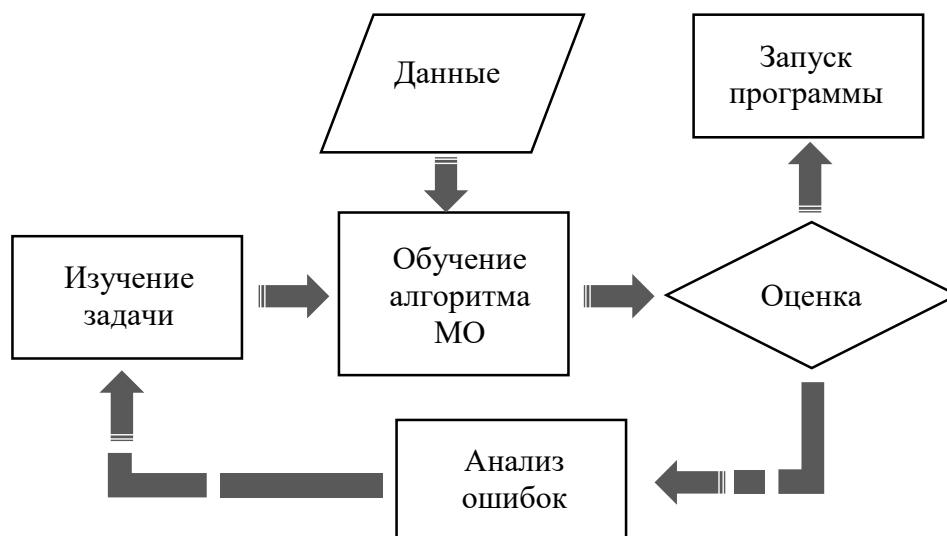


Рис. 4.4. Блок-схема алгоритма фильтр спама, основанного на приемах машинного обучения

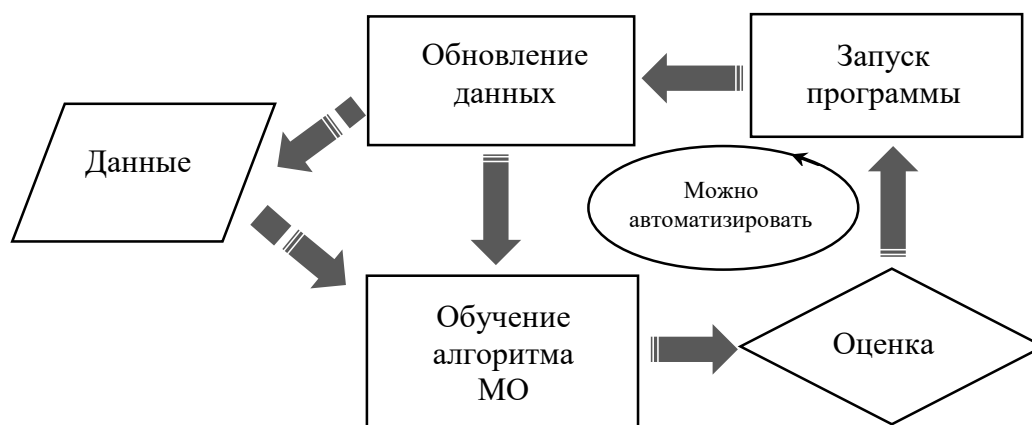


Рис. 4.5. Блок-схема алгоритм фильтр спама для обнаружения модернизированной фразы на приемах машинного обучения

В. Еще одна область, где МО показывает блестящие результаты по сравнению с подходами «традиционного программирования», охватывает задачи, которые либо слишком сложно решать с помощью традиционных подходов, либо для их решения нет известных алгоритмов. Например, это распознавание речи.

Предположим, что мы хотим начать с простого и написать программу, способную различать слова «один» и «два». Можно обратить внимание, что слово «два» начинается со звонкого звука «Д». Поэтому можно закодировать алгоритм, который измеряет интенсивность высокого звука и применяет это для проведения различий между упомянутыми словами. Очевидно, такой прием не будет масштабироваться на тысячи слов,

произносимых миллионами очень разных людей в шумных окружениях и на десятках языков.

В связи с этим, лучшее решение будет основываться на алгоритме МО, который учится самостоятельно, располагая множеством звукозаписей с примерами произношения каждого слова. Кроме того, машинное обучение способствует более глубокому пониманию проблемы (рис. 4.6).

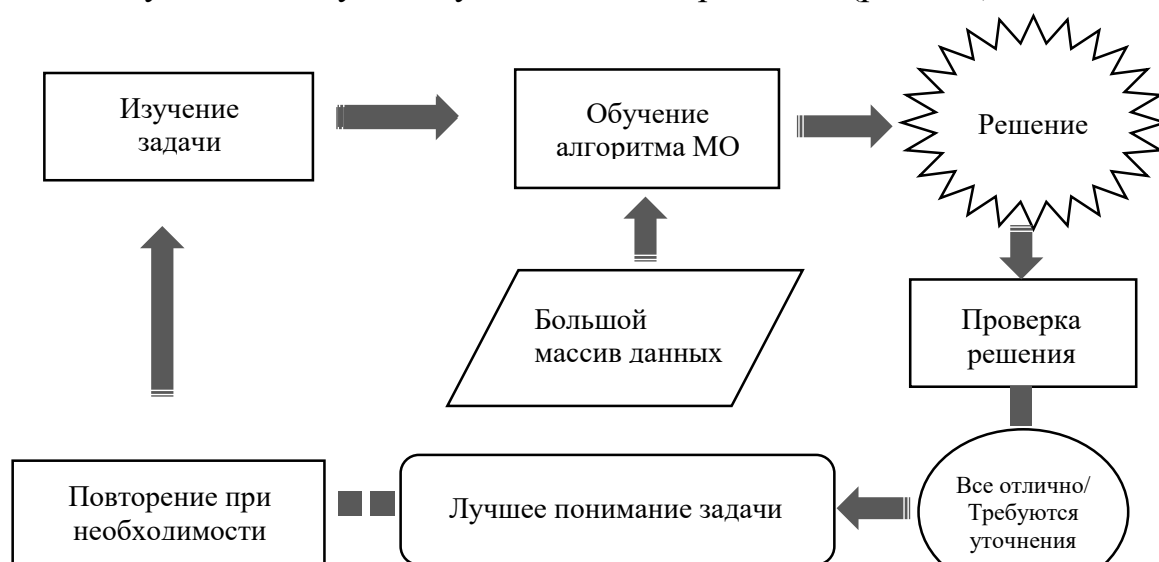


Рис. 4.6. Блок-схема алгоритма МО, который учится самостоятельно, располагая множеством данных с примерами

С. Существенным моментом является то, что *алгоритмы МО можно инспектировать с целью выяснения, чему они научились*. Например, после того, как фильтр спама был обучен на достаточном объеме спам-сообщений, его легко обследовать для выявления списка слов и словосочетаний, которые он считает лучшими маркерами (прогнозаторами) спама. Иногда удается найти неожиданные взаимосвязи или новые тенденции, приводящие к лучшему пониманию задачи.

Д. Применение приемов МО для исследования крупных объемов данных может помочь в обнаружении паттернов, которые не были замечены сразу. Это называется *интеллектуальным или глубинным анализом данных (Data mining)*.

Разница в подходах к решению задач в области МО и традиционного программирования весьма хорошо отображена на рис. 4.7.

Таким образом, машинное обучение весьма эффективно подходит для:

- задач, для которых существующие решения требуют большого объема тонкой настройки или длинных списков правил – один алгоритм МО часто способен упростить код и выполняться лучше, чем традиционный подход;
- сложных задач, для которых традиционный подход не дает пригодного решения – лучшие приемы МО вполне вероятно сумеют найти решение;

- изменяющихся сред – система МО способна адаптироваться к новым данным;
- получения сведений о сложных задачах и очень больших объемах данных.

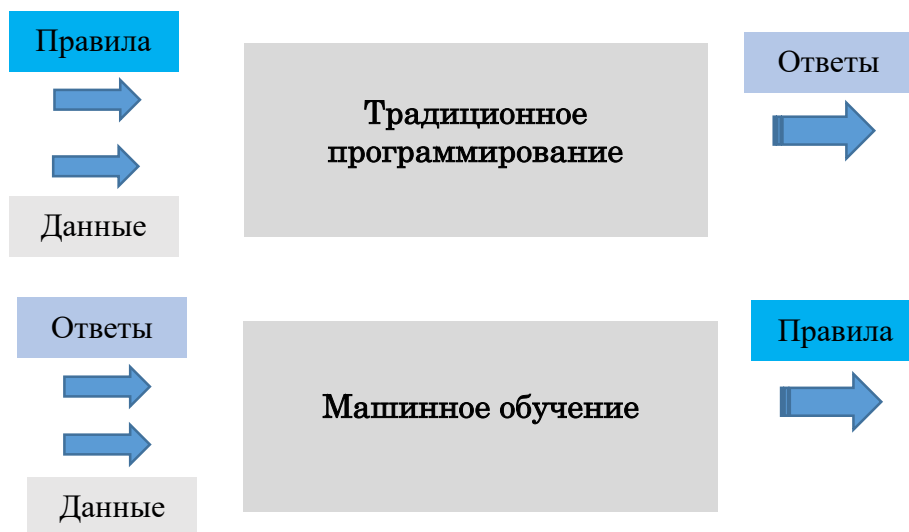


Рис. 4.7. Разница в подходах к решению задач в области МО и традиционного программирования

4.6. Основные виды (типы) машинного обучения

Общее «дерево» МО, включающее виды МО, типизацию этих видов и алгоритмы может быть представлено в виде следующей схемы (рис.4.8).

В выделении типов машинного обучения существует несколько подходов.

Так, еще недавно *распространенной классификацией* было выделение *следующих трех типов МО*:

- обучение с учителем или контролируемое обучение (supervised learning);
- обучение без учителя (unsupervised learning);
- обучение с подкреплением (reinforced learning).

Сегодня в машинном обучении в качестве базовых имеется четыре основных направления [21] (рис. 4.9).

4.6.1. Вид 1. Классическое обучение

Этот вид машинного обучения включает простейшие алгоритмы, которые являются прямыми наследниками алгоритмов из чистой статистики для вычислительных машин 1950-х годов. Такие алгоритмы используются для решения задач, легко поддающихся формализации. Например, такие, как поиск закономерностей в расчетах, вычисление траектории объектов.

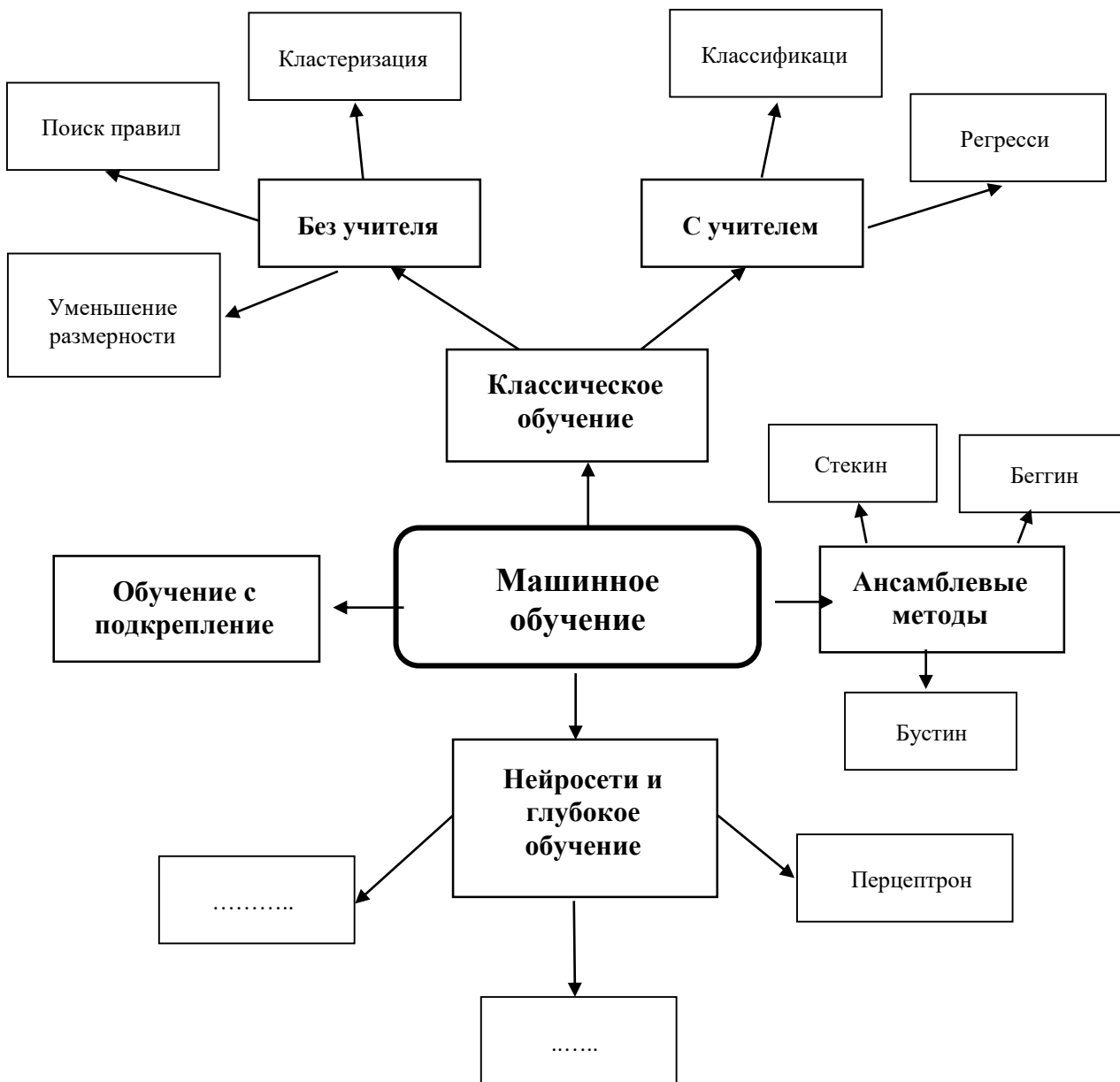


Рис. 4.8. Общее «дерево» МО, включающее виды МО, типизацию этих видов и алгоритмы

Сегодня алгоритмы МО, построенные на базе классического обучения – самые распространенные, на которых держится большая половина интернета. Именно они формируют блок рекомендаций на многих платформах.

Так, например, когда имеем дело с блоком «Рекомендованные статьи» на сайте, или в случае, когда банк блокирует деньги на карточке после первой же осуществленной покупки за границей – это почти всегда «дело рук» одного из этих алгоритмов.

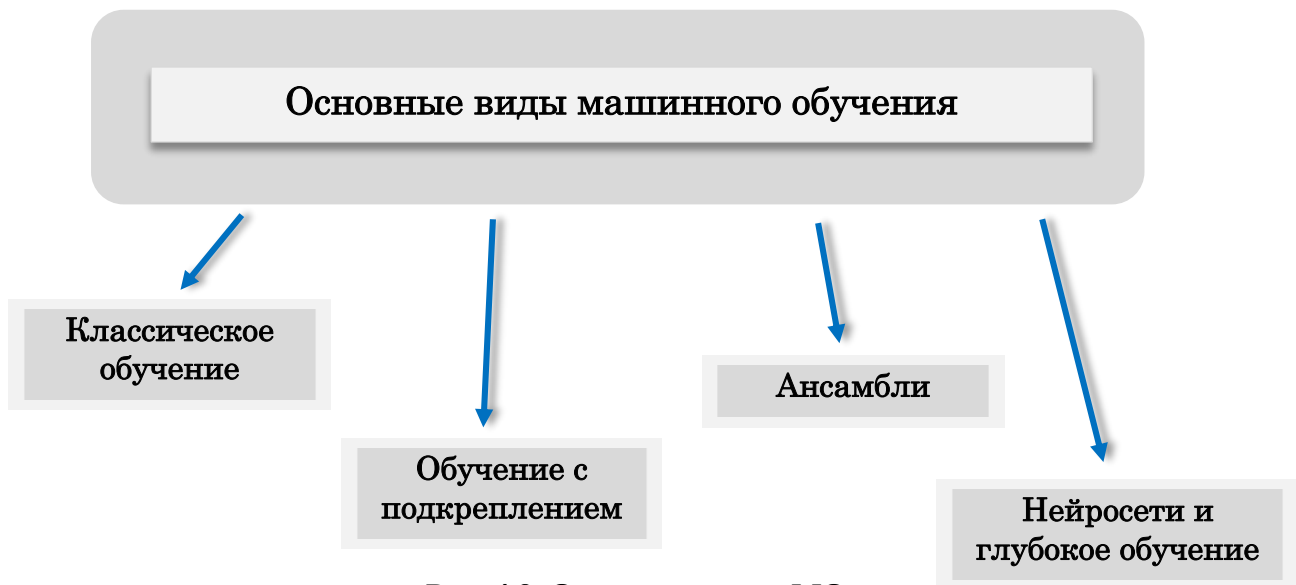


Рис. 4.9. Основные виды МО

Следует отметить, что крупные и «солидные» корпорации предпочитают в своих задачах повсеместно использовать нейросети. Но «обычным» разработчикам необходимо проводить предварительные исследования: когда задача может быть решена классическими методами, то «дешевле» реализовать алгоритм на них, а только потом думать об улучшениях.

Кроме того, при всей своей популярности, классические алгоритмы весьма просты, что их легко усвоить практически не подготовленному пользователю.

Классическое обучение, в свою очередь, тоже бывает разным (рис. 4.8).

■ **Обучение с учителем или контролируемое обучение (*supervised learning*).**

Обучение с учителем представляет собой вариант, когда у машины есть некий учитель, который знает, какой ответ правильный. Это значит, что исходные данные уже размечены (отсортированы) нужным образом и машине остается лишь определить объект с нужным признаком или вычислить результат.

Например, «учитель» рассказывает, что на этой картинке кошка, а на этой собака. То есть «учитель» уже заранее разделил (разметил) все данные на кошек и собак, а машина учится на конкретных примерах.

Соответственно, в *обучении без учителя*, машине просто предоставляется «куча» фотографий животных и говорят «разберись, кто здесь на кого похож». Данные не размечены, у машины нет учителя, и она пытается сама найти любые закономерности.

Очевидно, что с учителем машина обучится быстрее и точнее. Поэтому,

например, в задачах оперативного управления этот алгоритм используется намного чаще.

Алгоритм обучения с учителем принимает маркированные данные и создает модель, которая выполняет предсказания, предоставляя новые данные.

Главная цель обучения с учителем – обучить модель на помеченных обучающих данных, что позволит вырабатывать прогнозы на не встречающихся ранее или будущих данных. Здесь понятие «с учителем» относится к набору обучающих образцов (входных данных), где желаемые выходные сигналы (метки) уже известны. На рисунке 4.10 представлена типичная последовательность действий при обучении с учителем, где помеченные обучающие данные передаются алгоритму машинного обучения для подгонки к прогнозирующей модели, которая может вырабатывать прогнозы на новых непомеченных входных данных.

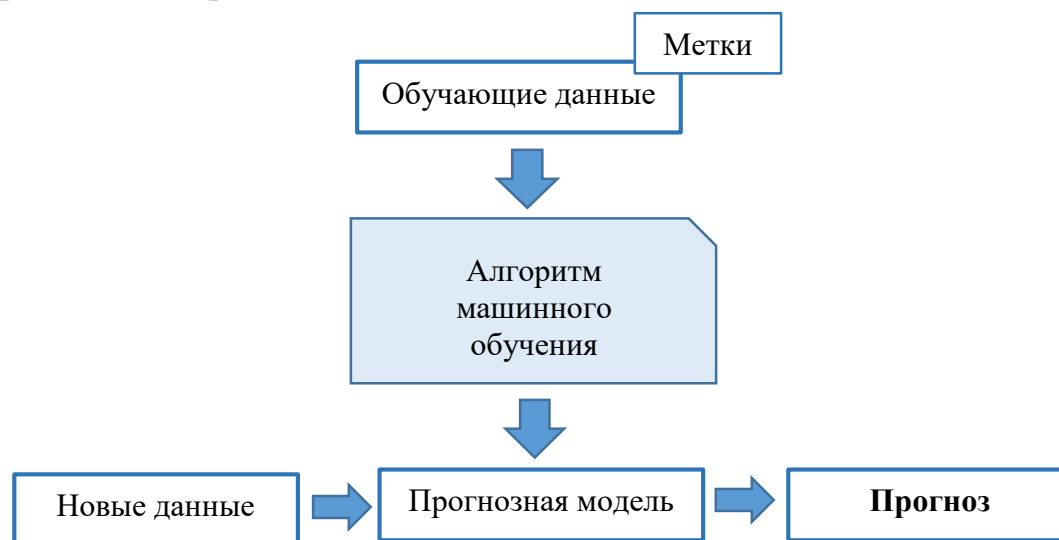


Рис. 4.10. Типичная последовательность действий при обучении с учителем

Задачи обучения с учителем делятся на два типа: *классификация* – предсказание категории объекта, и *регрессия* – предсказание места на числовой прямой.

Классификация – это подкатегория обучения с учителем, где целью является прогнозирование, к какому классу относится тот или иной образец (наблюдение, событие, явление, вещь и т.д.).

Классификация разделяет объекты по заранее известному признаку (носки по цветам, документы по языкам, музыку по жанрам и т.д.), рис. 4.11.

Сегодня классификация широко используется, например, для построения спам-фильтров; определения языка; поиска похожих документов; анализа тональности; распознавания рукописных букв и цифр; определения подозрительных транзакций и т.д.

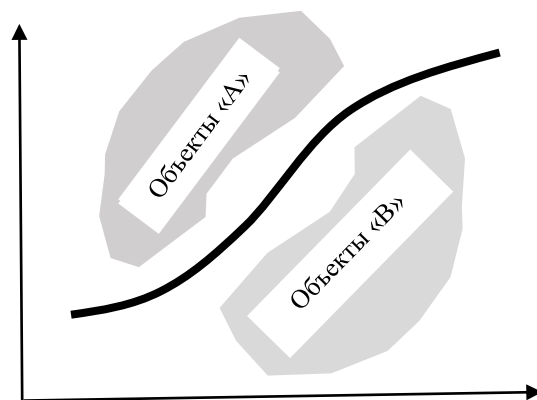


Рис. 4.11. Тип обучения с учителем - классификация

Принцип работы классификации состоит в следующем. Имеется множество объектов (ситуаций), разделенных некоторым образом на классы. Задано конечное количество объектов, для которых известно, к каким классам они относятся. Это множество называется *обучающей выборкой*. Классовая принадлежность остальных объектов неизвестна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества.

Классификация вещей представляет собой, наверное, самую популярную задачу во всем машинном обучении. Машина при использовании алгоритмов классификации подобна ребёнку, который учится раскладывать игрушки. Например, роботов в один ящик, танки в другой. Но, что делать, если среди игрушек имеется робот-танк? В этом случае ситуация «неразрешима».

Классифицировать объект – значит, указать номер (или наименование класса), к которому относится данный объект.

Классификация объекта – номер или наименование класса, выдаваемого алгоритмом классификации в результате его применения к данному конкретному объекту.

Типы классов:

- двухклассовая (бинарная) классификация – когда число классов равно двум;
- многоклассовая классификация – когда число классов больше двух.

Для классификации всегда нужен учитель, в качестве которого выступают размеченные данные с признаками и категориями, которые машина будет учиться определять по этим признакам.

Популярные алгоритмы: Наивный Байес, Деревья Решений, Логистическая Регрессия, К-ближайших соседей, Метод Опорных Векторов.

Например, в предыдущие годы все спам-фильтры работали на алгоритме Наивного Байеса. Машина считала сколько раз определенное

«подозрительное» слово встречается в спаме, а сколько раз в нормальных письмах. Перемножала эти две вероятности по формуле Байеса, складывала результаты всех слов и делала соответствующие выводы.

Позже «спамеры» научились обходить фильтр Байеса, просто вставляя в конец письма много слов с «хорошими» рейтингами. Такой метод получил ироничное название *Отравление Байеса*, а фильтровать спам стали другими алгоритмами. Но метод, использующий фильтр Байеса, навсегда остался в учебниках как самый простой, красивый и один из первых практически полезных.

Рассмотрим *другой пример* полезной классификации. Вы берёте кредит в банке. Как банку удостовериться, вернёте вы его или нет? Точно никак, но у банка есть тысячи профилей других людей, которые уже брали кредит до вас. Там указан их возраст, образование, должность, уровень зарплаты и главное – кто из них вернул кредит, а с кем возникли проблемы.

В классификации такого типа понятно, где здесь данные и какой надо предсказать результат. Поэтому можно обучить «машину», найти закономерности и получить ответ. Но в данном случае основной вопрос заключается в том, что *банк (потребитель информации) не может слепо доверять ответу машины, без дополнительных объяснений*. Вдруг какой-то собой, действия хакеров, и т.п.

Для такого типа задач существует подход, называемый *Деревья Решений*. Машина автоматически разделяет все данные по вопросам, ответы на которые «да» или «нет». Вопросы могут быть не совсем адекватными с точки зрения человека, например, в рассматриваемом примере: «зарплата заёмщика больше, чем 25999 рубля?», но машина придумывает их так, чтобы на каждом шаге разбиение было самым точным (рис. 4.12).

Так образом получается дерево вопросов. Чем выше уровень, тем более общий вопрос. Обычно постановка вопросов и их количество выполняется аналитиками.

Деревья нашли свою нишу в областях с высокой ответственностью: в диагностике, медицине, финансах. В чистом виде деревья сегодня используют редко, но вот их ансамбли (о которых речь пойдет далее) лежат в основе крупных систем и зачастую даже более эффективны, чем нейросети. *Например*, когда вы задаете вопрос Яндекс, именно ансамбль деревьев ранжирует результаты.

Самым популярным алгоритмом классической классификации является *Метод Опорных Векторов (SVM)*. Им классифицируют «практически всё»: виды растений, лица на фотографиях, документы по тематикам и т.д. Много лет он был главным ответом на вопрос «какой бы мне взять классификатор». Идея SVM по своей сути проста – он ищет, как

так провести две прямые между категориями, чтобы между ними образовался наибольший зазор.



Рис.4.12. Пример построения «дерева решений»

У классификации есть полезная обратная сторона – *поиск аномалий*. Когда какой-то признак объекта сильно не вписывается в наши классы, мы ярко подсвечиваем его на экране. Сейчас так делают в медицине: компьютер подсвечивает врачу все подозрительные области МРТ или выделяет отклонения в анализах. На биржах таким же образом определяют нестандартных игроков, которые скорее всего являются инсайдерами. Научив компьютер «как правильно», мы автоматически получаем и обратный классификатор – как неправильно.

Сегодня для классификации всё чаще используют нейросети, ведь по сути их для этого и изобрели.

Рекомендуется следующее правило: сложнее данные – сложнее алгоритм. Так, для текста, цифр, табличек рекомендуется начинать с «классики». Там модели меньше, обучаются быстрее и работают понятнее. Для картинок, видео и другой информации из категории «BigData» следует все же использовать нейросети.

Регрессия по своей сущности – это та же классификация, только вместо категории предсказывается число. Например, стоимость автомобиля по его пробегу, количество пробок по времени суток, объем спроса на товар от

роста компании и т.д. На регрессию идеально «ложатся» любые задачи, где есть зависимость от времени.

Наиболее популярные сферы использования регрессии сегодня: прогноз стоимости ценных бумаг; анализ спроса, объема продаж; медицинские диагнозы; любые зависимости числа от времени. Регрессия очень популярна для задач из области финансов и аналитики, она, например, встроена даже в Excel.

Алгоритм регрессии достаточно очевиден: машина рисует линию, которая в среднем отражает зависимость между дискретным набором величин. В отличие от человека, выполняет она это математически точно, считая среднее расстояние до каждой точки и пытаясь «всем угодить», рис. 4.13.

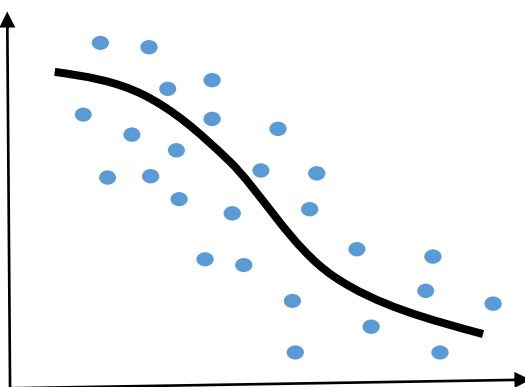


Рис. 4.13. Тип обучения с учителем - регрессия

Наиболее распространенные виды регрессии: *линейная* (регрессия «рисует» прямую линию) и *полиномиальная* («рисует» кривая линия). Помимо этого, существуют и другие виды регрессии.

Отметим, что имеется «*Логистическая Регрессия*», которая на самом деле не является регрессией, а представляет собой метод классификации. Поэтому, в связи с таким названием, существует определенная путаница.

Схожесть регрессии и классификации подтверждается еще и тем, что многие классификаторы, после небольшой модернизации, превращаются в регрессоры. Например, можно не просто смотреть к какому классу принадлежит объект, а запоминать, насколько он близок к определенному классу. А это уже регрессия.

■ Обучение без учителя

Обучение без учителя (Unsupervised Learning) представляет собой ситуацию, когда «машина сама» должна найти среди хаотичных данных верное решение и отсортировать объекты по неизвестным признакам (например, определить, где на фото собака).

Обучение без учителя было изобретено позже, чем обучение с учителем, – в начале девяностых годов. На сегодня на практике обучение

без учителя используется реже. Но имеется определенный класс задач, где «у нас просто нет выбора». Так, обучение без учителя применяют для данных, которые просто невозможно разметить из-за их колоссального объема.

Итак, главным образом, при обучении без учителя предоставляются непомеченные и неклассифицированные данные, в которых нужно найти закономерности и создать структуру данных для получения значения.

Размеченные (классифицированные) данные на практике – достаточно дорогая редкость. Но, например, что делать, если требуется написать классификатор автобусов? Выходить на улицу и фотографировать десятки тысяч автобусов и подписывать затем, где какой автобус – не самый оптимальный вариант. Наиболее распространенный вариант в настоящее время на практике – привлечь к этому делу тысячи (и даже миллионы) респондентов посредством специального сервиса (как делает, например, Яндекс). Другой вариант – попробовать использовать обучение без учителя. Хотя, в настоящее время «хороших» прикладных систем, использующих обучение без учителя практически нет.

Обучение без учителя, всё же, чаще используют как метод анализа данных, а не как основной алгоритм.

Основные формы обучения без учителя: кластеризация (clustering) и уменьшение размерности (dimensionality reduction).

Обучение без учителя разделяет объекты по неизвестному признаку. «Машина сама решает, как лучше».

Сегодня обучение без учителя используют главным образом для: сегментации рынка (типов покупателей, лояльности); объединения близких точек на карте; сжатия изображений; анализа и разметки новых данных; как детекторы аномального поведения; риск-менеджмента; прогноза акций и распродаж в ретейле; мерчендайзинга. *Например*, по такому принципу работает алгоритм iPhoto, который находит на фотографиях лица (не зная, чьи они) и объединяет их в альбомы.

Популярные алгоритмы: Метод К-средних, Mean-Shift, DBSCAN

Кластеризация – это классификация, но без заранее известных классов. Она «сама ищет» похожие объекты и объединяет их в кластеры. Количество кластеров можно задать заранее или доверить это машине. Похожесть объектов машина определяет по тем признакам, которые мы ей разметили, т.е. в один класс объединяются те объекты, у которых много схожих характеристик, рис. 4.14.

Отличный пример кластеризации – маркеры на картах в «вебе». Так, например, когда мы ищем на карте все разведывательные геоколонки, где на определенной глубине находится пласт полезного ископаемого, то с помощью движка такие геоколонки группируются в кружочки с циферкой.

Более сложные примеры кластеризации имеются в приложениях iPhoto или Google Photos, которые находят лица людей на фотографиях и группируют их в альбомы. Приложение не знает, как зовут ваших друзей, но может отличить их по характерным чертам лица. Это пример типичной кластеризации.

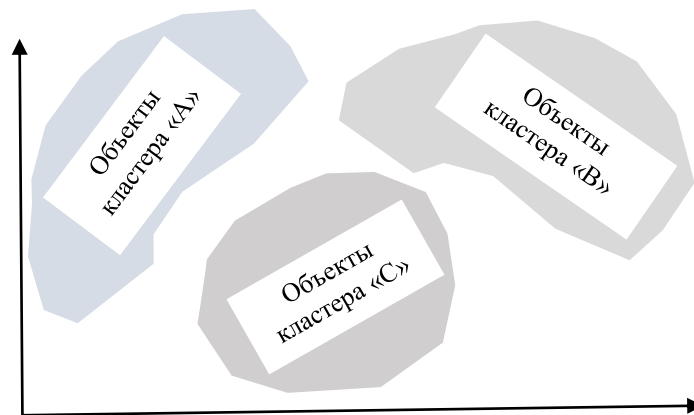


Рис. 4.14. Тип обучения без учителя - кластеризация

Сжатие изображений – еще одна популярная задача. Сохраняя картинку в PNG, можно установить палитру, скажем, в 32 цвета. Тогда кластеризация найдёт все «примерно красные» пиксели изображения, высчитает из них «средний красный» и заменит все красные на него. Меньше цветов – меньше файл. Проблема только с цветами типа Cyan: он ближе к зеленому или синему? В этом случае поможет популярный алгоритм кластеризации – *Метод K-средних (K-Means)*. Суть метода состоит в следующем: случайным образом бросаем на палитру цветов наши 32 точки, обзывая их центроидами. Все остальные точки относим к ближайшему центроиду от них. Получаются как бы созвездия из самых близких цветов. Затем двигаем центроид в центр своего созвездия и повторяем процедуру до тех пор, пока центроиды не перестанут двигаться. Считаем, что кластеры обнаружены, стабильны и их ровно 32, как и надо было. Иллюстрация метода *K-средних* показана на рисунке 4.15, где показана схема расстановки оптимальным способом трех станций отгрузки грузов со всех имеющихся в регионе пунктов (например, карьеров).

4.6.2. Вид 2. Обучение с подкреплением

Обучение с подкреплением (Reinforcement Learning) используют там, где задачей стоит не анализ данных, а «выживание в реальной среде».

При этом, средой может быть не только реальный мир, но и виртуальное пространство (видеоигра). Наиболее очевидные примеры – автопилот автомобиля, который «учится» не сбивать пешеходов, или робот-пылесос. Знания об окружающем мире таким роботам могут быть полезны,

но чисто для справки. Не важно сколько данных они соберут, у них всё равно не получится предусмотреть все ситуации. Потому их цель – минимизировать ошибки, а не рассчитать все ходы. Роботы учатся «выживать» в пространстве с максимальной заданной «выгодой» (например, временем поездки в автомобиле, максимальной площадью уборки в определенном помещении).

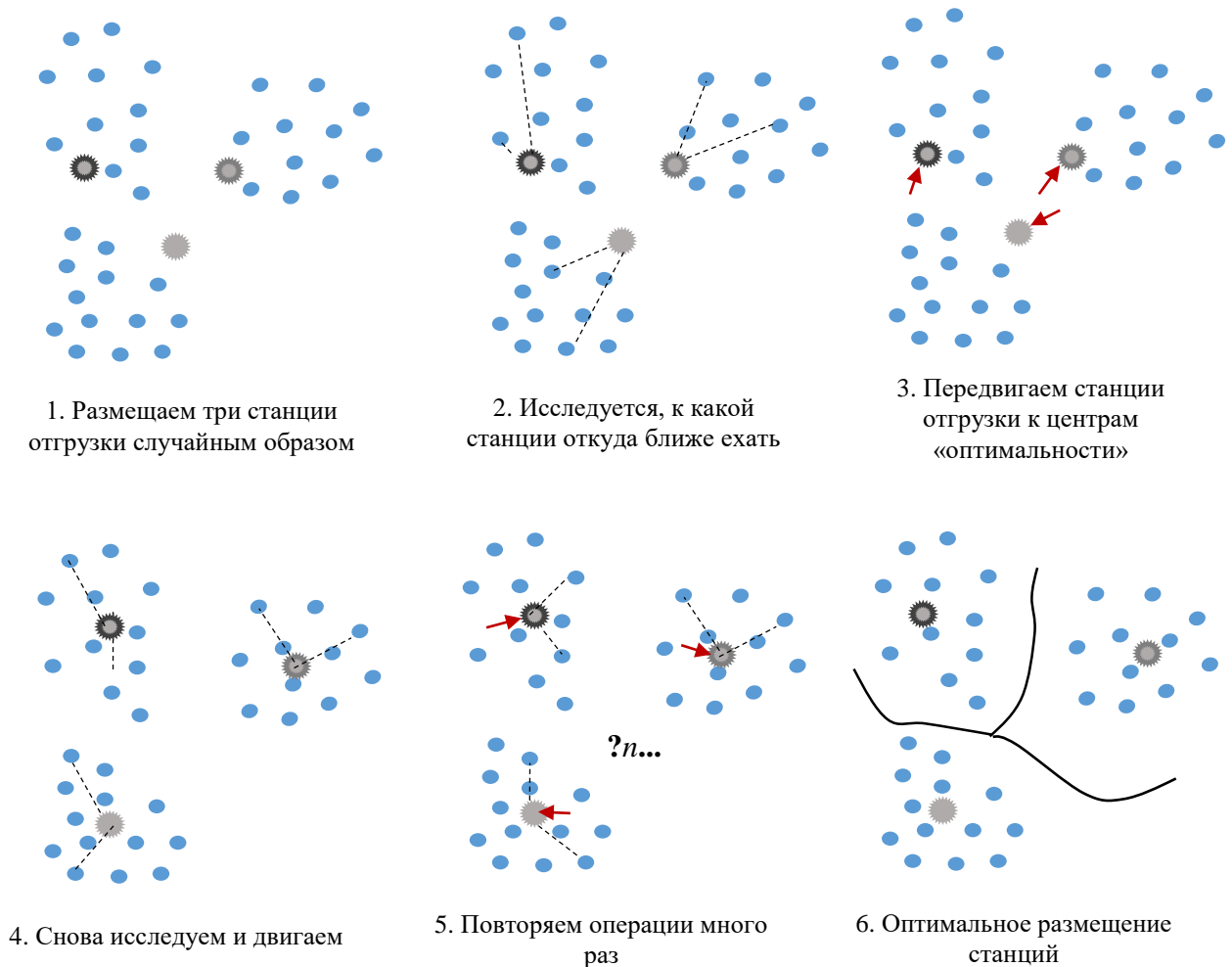


Рис. 4.15. Иллюстрация работы метода К-средних

Сегодня «обучение с подкреплением» используют главным образом для беспилотников; самоуправляемых автомобилей; роботов-пылесосов; компьютерных игр; торговли на фондовом рынке; управления ресурсами предприятий и компаний.

«Выживание в среде» и есть идея обучения с подкреплением.

Если провести аналогию с обществом людей, то ситуацию можно описать следующим образом: «Давайте бросим бедного робота в реальную жизнь и будем, как и человека, штрафовать его за ошибки и награждать за правильные поступки».

Умные модели роботов-пылесосов и самоуправляемые автомобили

обучаются именно так: им создают виртуальный город (часто на основе карт настоящих городов), населяют случайными пешеходами и отправляют учиться «никого там не убивать». Когда робот начинает хорошо себя чувствовать в искусственном пространстве, его выпускают для тестирования на реальные улицы. Запоминать сам город машине не нужно. Такой подход называется *Model-Free*. Конечно, есть и классический *Model-Based*, но в нём машине пришлось бы запоминать модель всей планеты, всех возможных ситуаций на всех перекрёстках мира. Такое просто не работает. В обучении с подкреплением машина не запоминает каждое движение, а пытается обобщить ситуации, чтобы выходить из них с максимальной выгодой. *Задача робота – свести ошибки к минимуму, за что он получает возможность продолжать работу без препятствий и сбоев.*

Таким образом, обучение с подкреплением использует систему поощрений, а также метод проб и ошибок в целях максимизации долгосрочного поощрения. Это более сложный, чем рассмотренные ранее, вид обучения. В данном случае ИИ нужно не просто анализировать данные, а действовать самостоятельно в реальной среде (улица, дом, видеоигра и др.).

Популярные алгоритмы: Q-Learning, SARSA, DQN, A3C, Генетический Алгоритм.

Используя обучение с подкреплением алгоритму AlphaGo удалось обыграть чемпиона по игре Го. Хотя незадолго до этого было доказано, что число комбинаций в этой игре физически невозможно просчитать, ведь оно превышает количество атомов во вселенной. То есть если в шахматах машина реально просчитывала все будущие комбинации и побеждала, то в случае игры Го так выполнить было невозможно. Поэтому алгоритм AlphaGo просто выбирал наилучший выход из каждой ситуации и делала это достаточно точно, чтобы обыграть.

Эта идея лежит в основе *алгоритма Q-learning* и его производных (SARSA и DQN). Буква Q в названии означает слово Quality, то есть робот учится поступать наиболее качественно в любой ситуации, а все ситуации он запоминает как простой марковский процесс.

Машина прогоняет миллионы симуляций в среде, запоминая все сложившиеся ситуации и выходы из них, которые принесли максимальное вознаграждение. Но как понять, когда имеет место известная ситуация, а когда абсолютно новая? Так, например, самоуправляемый автомобиль стоит у перекрестка и загорается зелёный — значит можно ехать? А если справа мчится скорая помощь с мигалками? Однозначного ответа, к сожалению, нет. Исследователи постоянно этим занимаются, изобретая все новые «костыли». Одни прописывают все ситуации руками, что позволяет им обрабатывать исключительные случаи типа *проблемы вагонетки*. Другие

идут глубже и отдают эту работу нейросетям, пусть сами всё найдут. Так вместо Q-learning'a появился *алгоритм Deep Q-Network (DQN)*.

Замечание. Задачу «проблема вагонетки» можно продемонстрировать следующими примерами. **1.** Предположим, что судья или магистрат столкнулись с разъярённой толпой, которая требует найти виновного в каком-то преступлении и угрожает в противном случае кровавой мезьью определенной части общества. Истинный виновник неизвестен, судья считает себя способным предотвратить кровопролитие, только подставив невиновного человека под обвинение и казнив его». **2.** Наряду с этим примером можно рассмотреть другой, в котором пилот, самолёт которого вот-вот потерпит крушение, решает, следует ли ему уклониться от более населённого района в менее населённый. Но, более известными являются следующие примеры. **3.** Водитель управляет едущим трамваем, который может только свернуть с одной колеи на другую. Пять человек работают на одной колее и один — на другой. Любой, кто окажется на колее, по которой он поедет, будет убит. **4.** Как и прежде, вагонетка несётся по рельсам, к которым привязаны пять человек. Вы находитесь на мосту, который проходит над рельсами. У вас есть возможность остановить вагонетку, бросив на пути что-нибудь тяжёлое. Рядом с вами находится толстый человек, и единственная возможность остановить вагонетку – столкнуть его с моста на пути. Каковы ваши действия?

Reinforcement Learning для простого обывателя выглядит как настоящий интеллект. Потому что машина сама принимает решения в реальных ситуациях! В настоящее время методы «обучение с подкреплением» весьма популярны. Разрабатываются алгоритмы сопряжения данной группы методов с нейросетями, чтобы они стали более точными (что бы робот-пылесос не стучался о ножку стула по двадцать раз).

4.6.3. Вид 3. Ансамбли

К такому виду машинного обучения относятся группы алгоритмов, которые используют сразу несколько методов машинного обучения и исправляют ошибки друг друга.

В настоящее время «Ансамбли» используют главным образом для:

- всего, где подходят классические алгоритмы (но ансамбли работают точнее);
- поисковых систем;
- компьютерного зрения;
- распознавания объектов.

Ансамбли и нейросети сегодня дают самые точные результаты и используются всеми крупными компаниями.

Эффективность «Ансамблей» базируется на достаточно простой идее: если взять несколько не очень эффективных методов обучения и обучить их исправлять ошибки друг друга, то в этом случае качество такой системы будет намного выше, чем каждого из методов по отдельности.

Популярные алгоритмы: Random Forest, Gradient Boosting.

Очевидно, что «Ансамбль» можно собрать любым способом. Но в этом случае нельзя «ручаться» за точность. Потому сегодня есть *три наиболее проверенных способа строить ансамбли*.

Стекинг. Разные алгоритмы обучают по отдельности, а потом передают их результаты на вход последнему, который и принимает итоговое решение (рис. 4.16).

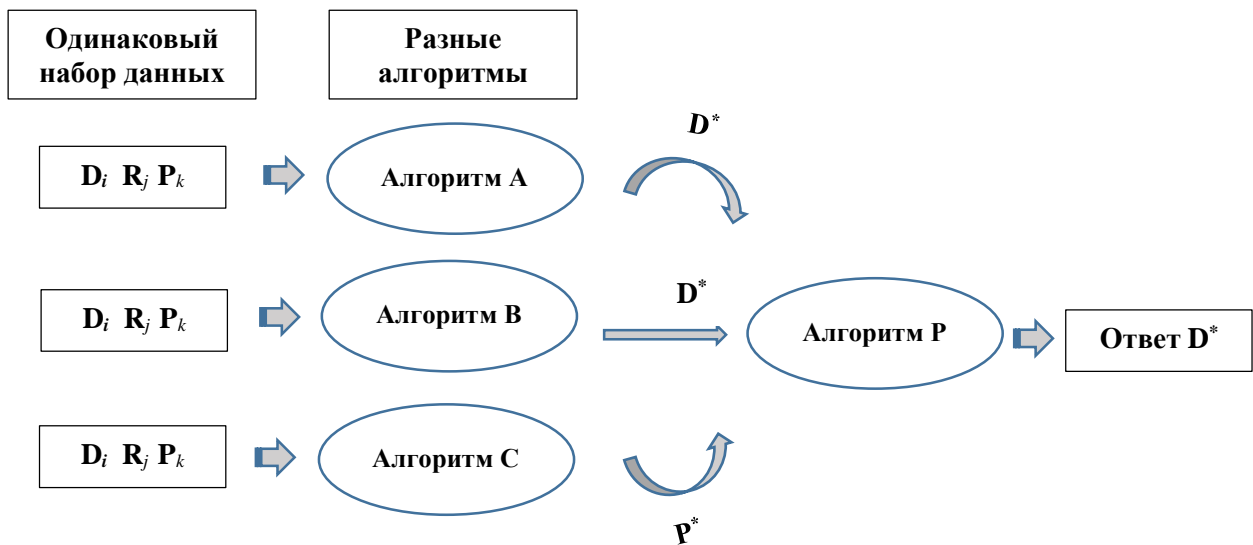


Рис. 4.16. Принцип работы Стекинга

Ключевое слово в данном случае – разные алгоритмы, ведь один и тот же алгоритм, обученный на одних и тех же данных не имеет смысла. Какие алгоритмы выбирать – дело разработчика. Но *в качестве решающего алгоритма чаще всего берут регрессию*.

На практике сегодня стекинг применяется уже достаточно редко, потому что описанные далее два других метода обычно более точные.

Беггинг (Bootstrap AGGregatING). Это подход, когда один алгоритм многократно обучают на случайных выборках из исходных данных, а потом усредняют ответы (рис. 4.17).

Данные в случайных выборках могут повторяться. То есть из набора 1-2-3 можно делать выборки 2-2-3, 1-2-2, 3-1-2 и так пока не надоест. На них обучаем один и тот же алгоритм несколько раз, а в конце вычисляем ответ простым голосованием.

Самый популярный пример беггинга – *алгоритм Random Forest*, (беггинг на деревьях), рис. 4.17. Например, когда вы открываете камеру на

телефоне и видите, как она очертила лица людей в кадре желтыми прямоугольниками – скорее всего это работ беггинга. Нейросеть будет слишком медлительна в реальном времени, а беггинг идеален, ведь он может считать свои деревья параллельно на всех шейдерах видеокарты.

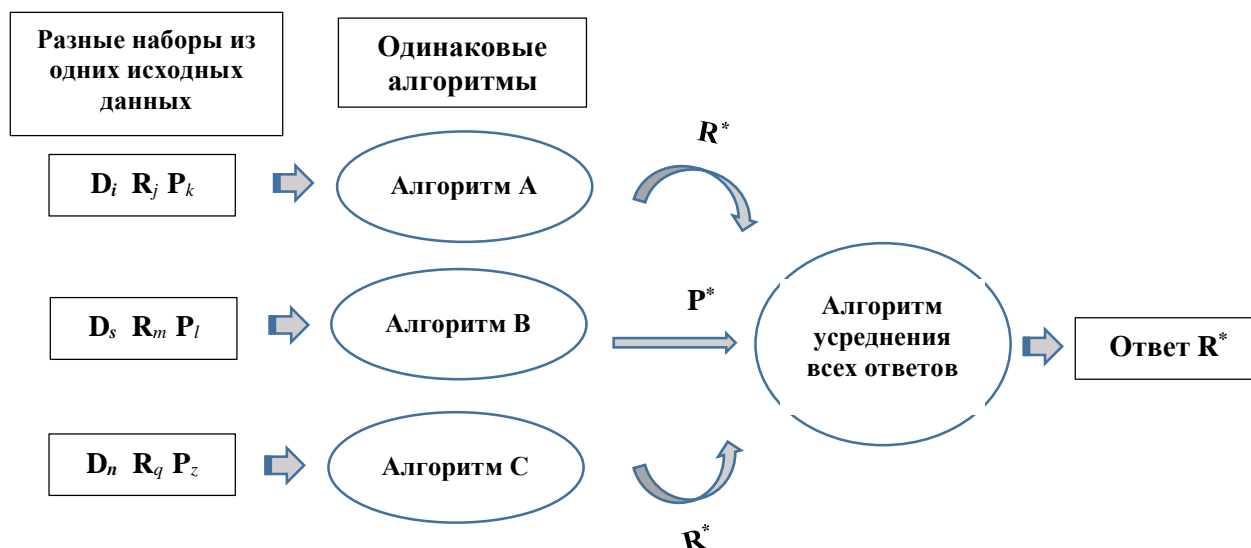


Рис. 4.17. Принцип работы Беггинга

Очень эффективная способность «параллеливаться» даёт беггингу преимущество даже над следующим методом, который работает точнее, но только в один поток. Хотя можно разбить данные на сегменты и запустить несколько алгоритмов одновременно.

Бустинг. В этом случае алгоритмы обучают последовательно, при этом каждый следующий уделяет особое внимание тем случаям, на которых ошибся предыдущий (рис. 4.18).

Как и в беггинге в бустинге делаются выборки из исходных данных, но теперь не совсем случайно. В каждую новую выборку берется часть тех данных, на которых предыдущий алгоритм отработал неправильно. То есть в этом случае новый алгоритм как бы «доучивается» на ошибках предыдущего.

Плюсы такого типа «Ансамблей» – очень высокая точность классификации. Минусы уже названы ранее и состоят в том, что бустинг «не параллелится». Хотя всё равно работает быстрее нейросетей.

Пример работы бустинга – работа Яндекс с запросами. Яндекс сейчас весь построен на бустинге.

4.6.4. Вид 4. Нейросети и глубокое обучение

Нейросети представляют собой сложный уровень обучения ИИ. Нейросети моделируют работу человеческого мозга, который состоит из

нейронов, постоянно формирующих между собой новые связи.

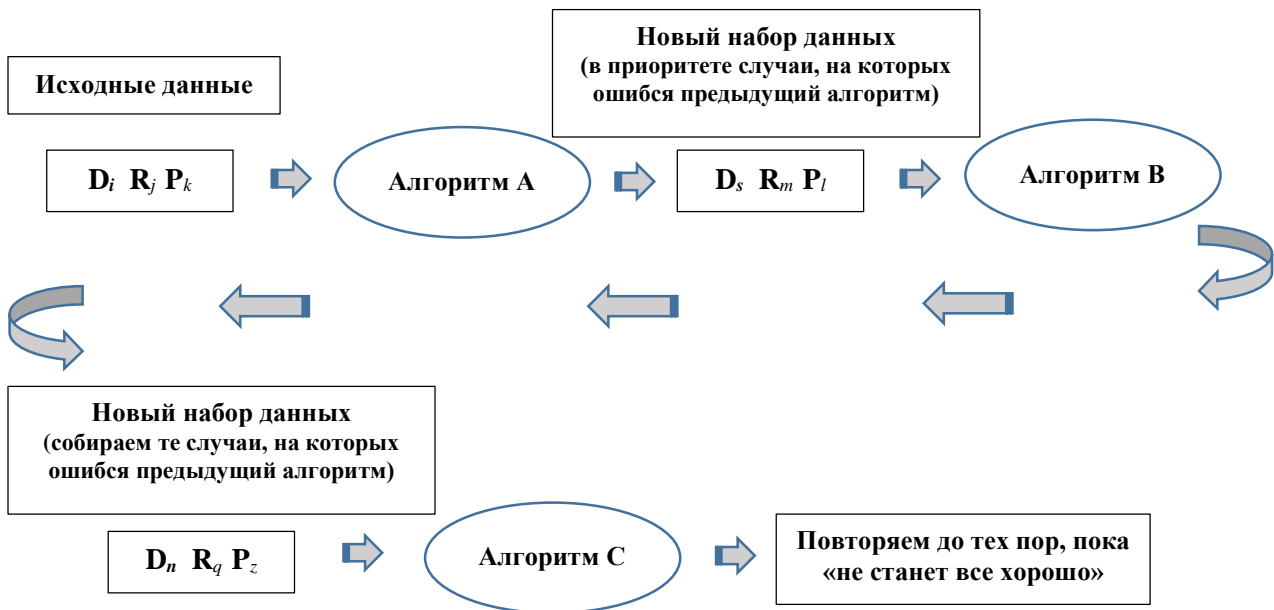


Рис. 4.18. Принцип работы Бустинга

В настоящее время нейросети используют:

- вместо всех вышеперечисленных алгоритмов вообще;
- для определения объектов на фото и видео;
- для распознавания и синтеза речи;
- при обработке изображений, переносе стиля;
- при машинном переводе.

Наиболее популярные архитектуры нейросетей: Перцептрон, Свёрточные Сети (CNN), Рекуррентные Сети (RNN), Автоэнкодеры и др.

Любая нейросеть – это набор нейронов и связей между ними. Нейрон по своей сути представляет функцию с множеством входов и одним выходом. Задача нейрона – взять числа со своих входов, выполнить над ними некую функцию и отдать результат на выход. *Пример простейшего нейрона:* просуммировать все цифры со входов, и если их сумма больше N , то выдать на выход единицу, иначе – ноль.

В свою очередь, связи – это каналы, через которые нейроны шлют друг другу цифры. У каждой связи есть свой вес, являющийся её единственным параметром, который можно условно представить как прочность связи. Так, *например*, если через связь с весом 0.5 проходит число 10, то оно превращается в 5. Сам нейрон «не разбирается», что к нему пришло и суммирует всё подряд, поэтому веса и нужны, чтобы управлять тем, на

какие входы нейрон должен реагировать, а на какие нет (рис.4.19).

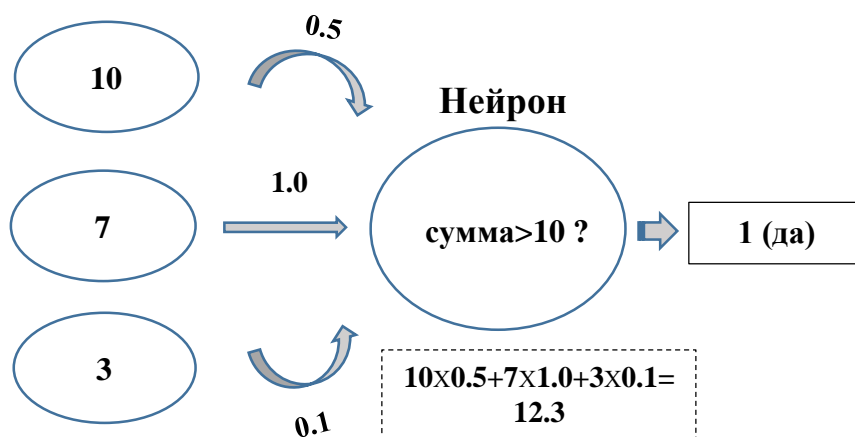


Рис. 4.19. Принцип работы искусственного нейрона

Чтобы сеть не превратилась в неупорядоченное множество, *нейроны связывают не как захочется, а по слоям*. Внутри одного слоя нейроны никак не связаны, но соединены с нейронами следующего и предыдущего слоя. Данные в такой сети идут строго в одном направлении – от входов первого слоя к выходам последнего.

ИИ собирает данные со всех входов, оценивая их вес по заданным параметрами, затем выполняет нужное действие и выдает результат. Сначала он получается случайным, но затем через множество циклов становится все более точным. Хорошо обученная нейросеть работает, как обычный алгоритм или точнее.

Например. Пусть на входе имеется изображение написанной от руки цифры 4. Если сеть содержит достаточное количество слоёв и правильно расставлены веса в такой сети, то подав на вход изображение написанной от руки цифры 4, чёрные пиксели активируют связанные с ними нейроны сети, которые активируют следующие слои, и так далее и далее, пока в итоге не на выходе активируется тот выход, который отвечает за четвёрку и результат будет достигнут (рис. 4.20).

Отметим, что в реальности при программировании, естественно, никаких нейронов и связей не пишут, алгоритм представляет собой операции с матрицами.

Когда сеть построена, важной задачей является правильно расставить веса, чтобы нейроны реагировали только на нужные сигналы. Для этого используется *процедура обучения сети* на имеющихся в наличии примерах «входов» и правильных «выходов». В рассмотренном примере нейросети «показывают» рисунок той же цифры 4 и «говорят», чтобы сеть «подстроила» свои веса так, чтобы на выходе при таком входе всегда была четвёрка.

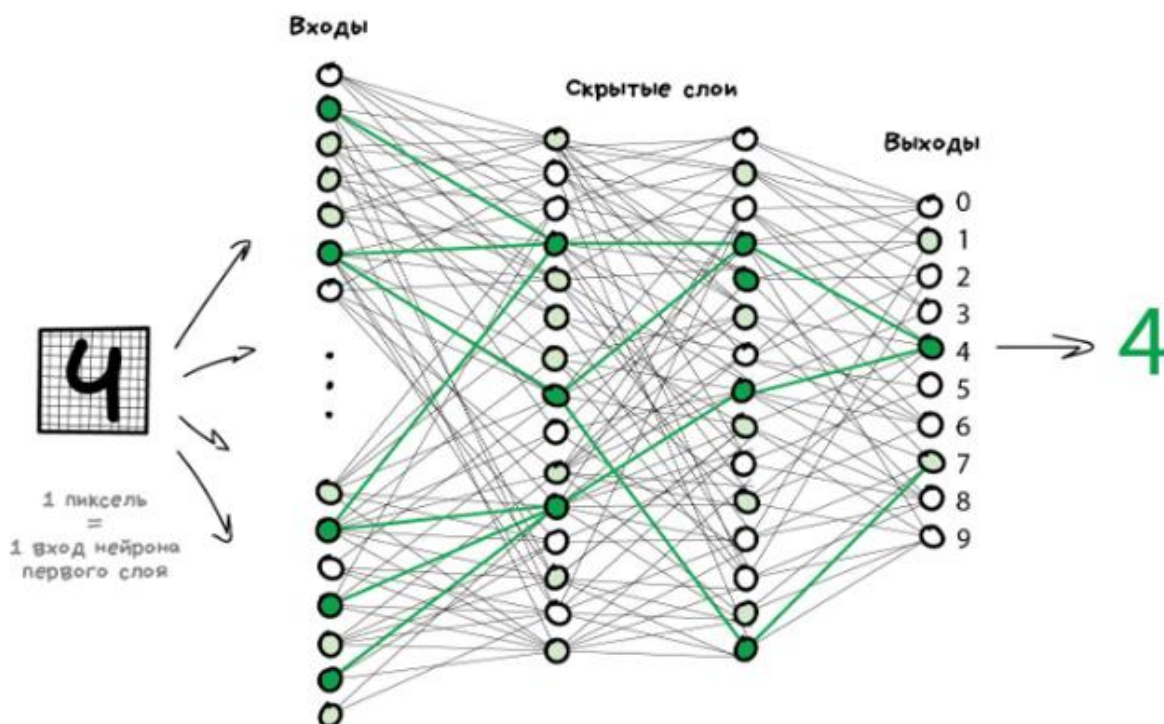


Рис. 4.20. Пример «работы» нейросети по распознаванию цифры «четыре»

Таким образом, при обучении сначала все веса просто расставлены случайно. «Показываем» сети цифру, она выдаёт какой-то случайный ответ, а мы сравниваем, насколько результат отличается от нужного нам. Затем идём по сети в обратном направлении, от выходов ко входам, и анализируем работу каждого нейрона. Активность определенных нейронов на какую-то связь необходимо уменьшить, а другую увеличить. Это выполняется присвоением новых весов. Через тысяч сто таких циклов «прогнози-проверили-подкорректировали» есть надежда, что веса в сети откорректируются так, как мы хотели. Такой подход называется Backpropagation или «Метод обратного распространения ошибки».

При таком подходе на обучение сети с большим количеством слоёв требовались огромные вычислительные мощности, которые были в те времена (двадцать лет назад) практически недоступны. Поэтому в тот период времени в нейросетях все сильно разочаровались и считали этот путь машинного обучения неперспективным.

Настоящим прорывом в области нейронных сетей стала разработка порядка десяти лет назад методов глубокого обучения, которое обучает нейросети на нескольких уровнях абстракций.

Нейросети с глубоким обучением требуют огромных массивов данных и технических ресурсов. Именно они лежат в основе машинного перевода, чат-ботов и голосовых помощников, создают музыку и дипфейки, обрабатывают фото и видео.

В глубоком обучении *используются две главных архитектуры: сверточные нейросети и рекуррентные нейросети.*

Замечание. В 2012 году свёрточная нейросеть произвела фурор в конкурсе ImageNet, из-за чего в мире внезапно вспомнили о методах глубокого обучения, описанных еще в 90-х годах.

Сверточные нейросети первыми «научились» распознавать неразмеченные изображения – самые сложные объекты для ИИ. Для этого они разбивают их на блоки, определяют в каждом доминирующие линии и сравнивают с другими изображениями нужного объекта;

Рекуррентные нейросети отвечают за распознавание текста и речи. Они выявляют в них последовательности и связывают каждую единицу (букву или звук) с остальными.

Отличие глубокого обучения от классических нейросетей состоит в новых методах обучения, которые справляются с большими размерами сетей. Однако *сегодня лишь теоретики разделяют, какое обучение можно считать глубоким, а какое «не очень»* [1].

Сегодня практически повсеместно используются популярные «глубокие» библиотеки типа Keras, TensorFlow и PyTorch даже когда надо собрать мини-сетку на пять-шесть слоёв. Просто потому что они удобнее всего того, что было раньше. И называют это просто нейросетями.

Одними из главных в методах глубокого обучения на сегодняшний момент являются сверточные нейросети.

Свёрточные Нейросети (CNN) сейчас находятся на пике популярности. Они используются для поиска объектов на фото и видео, распознавания лиц, переноса стиля, генерации и дорисовки изображений, создания эффектов типа «слоу-мо» и улучшения качества фотографий. Сегодня CNN применяют везде, где есть картинки или видео. Так, в айфоне несколько таких сетей «смотрят на фотографии», чтобы распознать объекты на них.

Проблема с изображениями всегда была в том, что непонятно, как выделять на них признаки. Текст можно разбить по предложениям, взять свойства слов из словарей. Картинки же приходилось размечать руками, объясняя машине, где «у котика на фотографии ушки, а где хвост» (рис. 4.21). Такой подход даже называли «handcrafting признаков» и раньше все так и делали.

Проблем у «ручного крафтинга» достаточно много. Во-первых, если «котик на фотографии прижал ушки или отвернулся», то в этом случае нейросеть ничего «не увидит». Во-вторых, попробуйте сами сейчас назвать хотя бы десять характерных признаков, отличающих котиков от других животных. Обычному человеку сделать это достаточно проблематично. Но, когда ночью мимо вас пробегает чёрное пятно, даже краем глаза вы сможете сказать котик это, или другое животное. Потому что человек не смотрит

только на форму ушей и количество лап – он оценивает объект по куче разных признаков, о которых сам даже не задумывается. А значит, не понимает и не может объяснить машине.



Рис. 4.21. Принцип работы подхода «handcrafting признаков»

Получается, машине надо самой учиться искать эти признаки, составляя из каких-то базовых линий. *Эту процедуру можно сделать, например, следующим образом:* для начала разделяем изображение на блоки 8x8 пикселей и выбираем, какая линия доминирует в каждом: горизонтальная [–], вертикальная [∣] или одна из диагональных [∕]. Могут и две, и три, так тоже бывает. На выходе получаем несколько массивов палочек, которые по сути являются простейшими признаками наличия очертаний объектов на картинке. По сути это тоже картинки, просто из палочек. Значит можно вновь выбрать блок 8x8 и посмотреть уже, как эти палочки сочетаются друг с другом. А потом еще и еще.

Такая операция называется *свёрткой*, откуда и пошло название метода. Свёртку можно представить, как слой нейросети, ведь нейрон – это абсолютно любая функция.

Когда через построенную нейросеть «прогоняется» куча фотографий котов, нейросеть автоматически расставляет большие веса тем сочетаниям из палочек, которые увидела чаще всего. Причём неважно, это прямая линия спины или сложный геометрический объект типа мордочки, что-то обязательно будет ярко активироваться.

На выходе же можно поставить простой перцептрон, который будет смотреть, какие сочетания активировались и говорить, кому они больше характерны – кошке или собаке.

Эффективность изложенного подхода заключается в том, что у нас получилась нейросеть, которая сама находит характерные признаки объектов. Нам больше не надо отбирать их руками. Мы можем сколько угодно «кормить» её изображениями любых объектов, просто копируя, например, миллионы картинок с объектами из интернета. Сеть сама составит карты признаков из палочек и научится определять что угодно.

4.7. Некоторые общие правила МО

Система МО разрабатывается для того, чтобы она помогла извлекать смысловую информацию из данных. Для обучения модели используется ограниченное множество примеров, на основании которых модель пытается научиться общим правилам. После того, как модель научилась этим правилам, можно использовать ее в рабочем процессе, предъявляя ей новые данные. Применяя освоенные моделью правила к новым данным, система может дать полезную информацию об этих данных.

Если модель может выдавать точные прогнозы на ранее не встречающихся данных, то говорят, что модель обладает способностью обобщать результат на тестовые данные. *Необходимо построить модель, которая будет обладать максимальной обобщающей способностью.*

Если не уделять достаточного внимания деталям в обучающих примерах, то правила будут слишком общими и при работе с новыми данными это может привести к неверным заключениям. С другой стороны, если будем уделять слишком много внимания деталям в примерах, то правила будут слишком специфическими и вновь, вероятно, при работе с новыми данными можно прийти к неправильным заключениям.

*Эти феномены соответственно называются **недостаточностью и избыточностью**.*

Более общей и неприятной проблемой из этих двух является *избыточность*, которая может, если ей не уделить необходимого внимания, сделать готовую систему бесполезной. Контролировать и управлять избыточностью можно с помощью методов, известных под общим названием регуляризация.

Системы машинного обучения очень чувствительны к избыточности. Если во входных данных присутствует какая-либо индивидуальная особенность, которая помогает системе сделать правильное предсказание, то она найдет возможность использовать ее.

Недостаточность – обычно менее существенная проблема, чем избыточность. Недостаточность часто «лечится» путем использования большего количества обучающих данных. С большим количеством обучающих примеров, систем может выработать более качественные правила для понимания каждой части из них.

В *машинном обучении с учителем* необходимо построить модель на обучающих данных, а затем получить точные прогнозы для новых, еще не встречавшихся для модели данных, которые имеют те же самые признаки, что и использованный обучающий набор. Обычно модель строится таким образом, чтобы она давала точные прогнозы на обучающем наборе данных. Если обучающий и тестовый наборы имеют много общего между собой, можно ожидать, что модель будет точной и на тестовом наборе. Однако в

некоторых случаях этого не происходит. Например, если строим очень сложные модели, необходимо помнить, что на тестовой выборке можно получить произвольную правильность.

В качестве примера рассмотрим следующую выборку, выполненную по опросу сотрудников научно-исследовательской лаборатории (табл. 4.1).

Табл. 4.1. Обучающая выборка

	Возраст	Уч.степень	Авто	Семья	Дом	Лодка
1	67	Доктор наук	да	женат	да	да
2	52	Доктор наук	да	женат	да	да
3	21	нет	да	холост	нет	нет
4	25	Канд. наук	нет	женат	нет	да
5	45	Доктор наук	да	разведен	нет	нет
6	39	Канд. наук	да	женат	да	нет
7	26	нет	да	холост	нет	да
8	40	нет	нет	женат	да	нет
9	53	Канд. наук	да	разведен	да	да
10	65	Канд. наук	да	женат	да	да
11	58	Доктор наук	да	женат	да	да
12	32	нет	нет	холост	нет	нет

Исходя из данных, размещенных в таблице, можно сформулировать такое правило: «если сотрудник научно-исследовательской лаборатории старше 50 лет, то у него есть автомобиль и у него есть дом(дача), то он обязательно имеет и лодку». На представленной обучающей выборке это правило работает со 100% правильностью.

Можно было бы сформулировать множество правил, которые хорошо работают для представленной выборки. Но в данном случае нас интересует вопрос, могут ли сотрудники других лабораторий института купить лодку. Поэтому нам нужно правило, которое будет хорошо работать для новых «клиентов». Как видно, достижение 100% правильности на обучающей выборке не поможет в этом.

Единственный показатель качества работы алгоритма на новых данных – это использование тестового набора. Однако интуитивно ожидаем, что простые модели должны лучше обобщать результат на новые данные. Если бы правило звучало «люди старше 50 лет хотят купить лодку» и оно объясняло бы поведение всех клиентов, мы доверяли бы ему больше, чем правилу, которое помимо возраста включало бы другие не совсем значимые факторы. Поэтому нам всегда нужно искать самую простую модель.

В нашем примере, построена модель, которая слишком сложна для имеющегося у нас объема информации. Такая ситуация называется «**переобучением**» (overfitting). Переобучение происходит, когда модель слишком точно подстраивается под особенности обучающей выборки и

получаем модель, которая хорошо работает на обучающей выборке, но не умеет обобщать результат на новые данные.

С другой стороны, если модель будет слишком проста (например, мы сформулируем правило «все сотрудники, у кого есть дом(дача), покупают лодку»), то она не сможет охватить всё многообразие и изменчивость данных и модель будет плохо работать даже на обучающей выборке. Выбор слишком простой модель называется «недообучением» (underfitting).

Существует оптимальная точка, которая позволяет получить наилучшую обобщающую способность. Собственно, это и есть модель, которую необходимо найти (рис. 4.22).

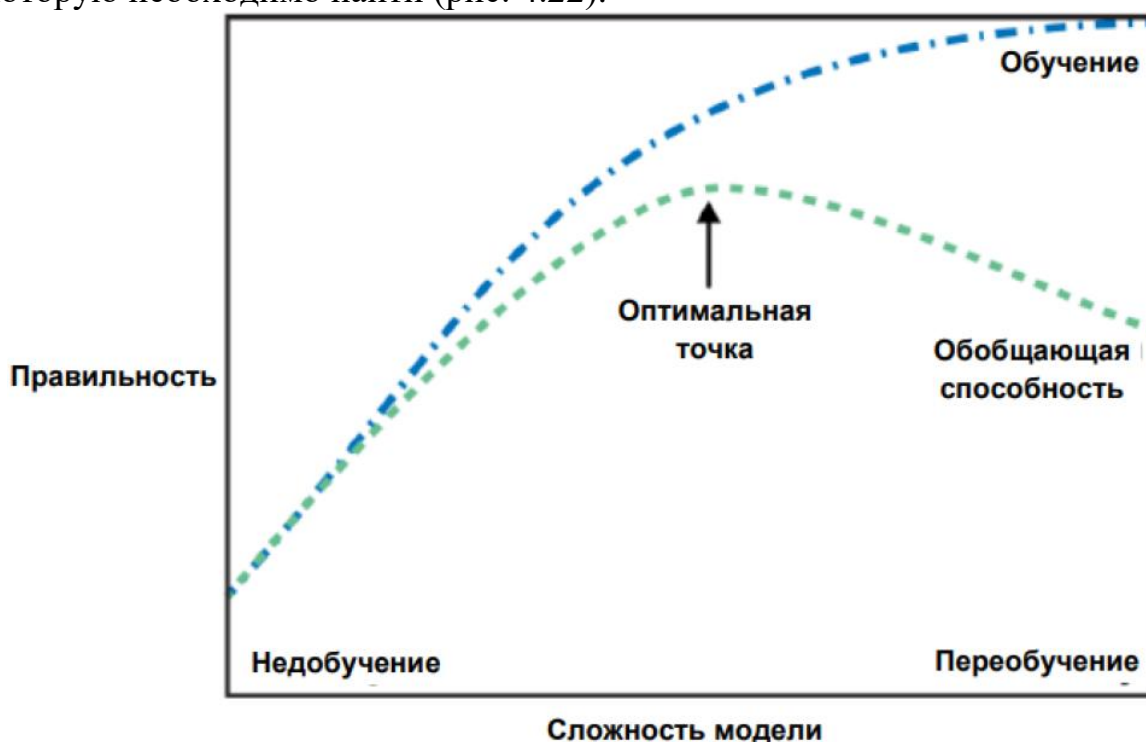


Рис. 4.22. Взаимосвязь между сложностью модели и размером набора данных

Сложность модели тесно связана с изменчивостью входных данных, которые содержатся в обучающей выборке. Чем больше разнообразие точек данных в обучающей выборке, тем более сложную модель можно использовать, не беспокоясь о переобучении. Однако простое дублирование одних и тех же точек или сбор очень похожих данных здесь не поможет.

Возвращаясь к примеру с покупкой лодки, можно сказать, что если бы у нас в наличии было более 10000 строк данных о клиентах и все они подчинялись бы правилу «если сотрудник научно-исследовательской лаборатории старше 50 лет, то у него есть автомобиль и у него есть дом(дача), то он обязательно имеет и лодку», мы бы с гораздо большей вероятностью поверили в это правило, чем если бы оно было сформулировано лишь по 12 строкам данных.

Увеличение объема данных и построение более сложных моделей позволяет существенно улучшить качество прогнозов в машинном обучении с учителем.

Для двумерных массивов данных можно показать прогнозы для всех возможных точек тестового набора, разместив их в плоскости x_1 . Для этого можно использовать следующий алгоритм. Задается цвет плоскости в соответствии с тем классом, который будет присвоен точке этой области. Это позволяет *построить границу принятия решений* (decision boundary), которая разбивает плоскость на две области: область, где алгоритм присваивает класс 0, и область, где алгоритм присваивает 1.

На рисунке 4.23 слева можно увидеть, что использование модели одного ближайшего соседа дает границу принятия решений, которая очень хорошо согласуется с обучающими данными. Увеличение числа соседей приводит к сглаживанию границы принятия решений. Более гладкая граница соответствует более простой модели. Другими словами, использование *нескольких соседей* соответствует высокой сложности модели, а использование *большого количества соседей* соответствует низкой сложности модели.



Рис.4.23. Пример построения модели при учете различного количества соседей

Если использовать максимальное количество соседей (количество соседей равно количеству точек обучающей выборки), то каждая точка тестового набора будет иметь одних и тех же соседей (соседями будут все точки обучающей выборки) и все прогнозы будут одинаковыми: будет выбран класс, который наиболее часто встречается в обучающей выборке.

Несколько слов *о взаимосвязи между сложностью модели и обобщающей способностью модели*. Рассмотрим пример с использованием реального набора данных Breast Cancer [23]. Данные разбивались на обучающую и тестовую выборки. Затем выполнялась оценка качества работы модели на обучающей и тестовой выборках с использованием разного количества соседей (рис. 4.24).

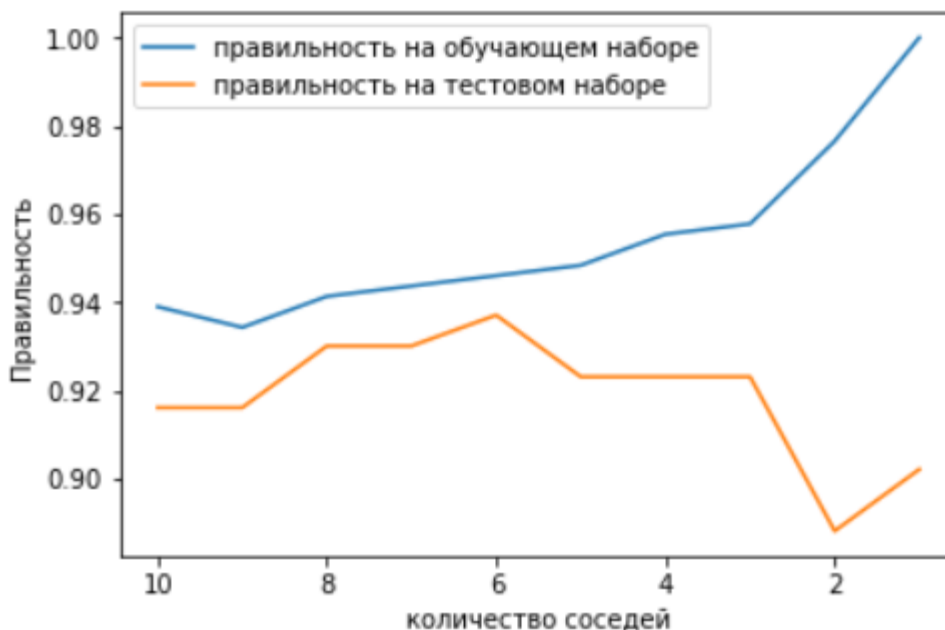


Рис.4.24. Оценка качества работы модели на обучающей и тестовой выборках с использованием разного количества соседей

При использовании модели одного ближайшего соседа правильность на обучающем наборе идеальна. Однако при использовании большего количества соседей модель становится все проще и правильность на обучающем наборе падает. Правильность на тестовом наборе в случае использования одного соседа ниже, чем при использовании нескольких соседей. Это указывает на то, что использование одного ближайшего соседа приводит к построению слишком сложной модели. С другой стороны, когда используются 10 соседей, модель становится слишком простой и она работает хуже.

Оптимальное качество работы модели наблюдается где-то посередине, когда используется шесть соседей. Худшая по качеству модель дает правильность на тестовом наборе около 88%, что по-прежнему может быть приемлемым результатом.

4.8. Предварительная обработка данных

Предварительная обработка данных – важнейший этап при построении систем машинного обучения. Грамотная и корректная предварительная обработка позволяет эффективно обучить модель МО.

Рассмотрим задачу предварительной обработки данных на примере.

Пусть имеется таблица обработки данных физико-механических характеристик пород, находящихся на определенной глубине, по керну, извлеченному из разных мест шахтного поля, таблица 4.2.

Таблица 4.2. Физико-механические характеристики пород

№п/п	Участок шахтного поля	Порода	Значение характеристики	Мощность слоя
1	север	Слой_I	A ₁	B ₁
2	юг	Слой_II	A ₂	B ₂
3	запад	Слой_III	A ₃	нет
4	юг	Слой_III	A ₄	B ₄
5	север	Слой_I	A ₅	B ₅
6	запад	Слой_III	нет	B ₆
7	III	Слой_II	A ₇	B ₇
8	II	Слой_I	A ₈	B ₈
9	I	Слой_II	A ₉	B ₉

В данном датасете имеются следующие проблемы: в элементе 6 нет значения физико-механической характеристики, а в элементе 3 нет значения мощности слоя.

В рассматриваемом датасете в колонках «участок шахтного поля» и «порода» присутствуют категориальные признаки.

Задача состоит в необходимости масштабирования признаков для увеличения эффективности работы модели МО.

Разделение датасета. Необходимо разделить выборку на *матрицу признаков* и *вектор зависимых переменных* (dependent variables). Модель МО будет использовать матрицу признаков, чтобы предсказать зависимые переменные.

В рассматриваемом случае это сделать достаточно просто. Матрица признаков – «участок шахтного поля» и «порода». Вектор зависимых переменных – «значение характеристики» и «мощность слоя».

Обработка недостающих данных. Очень часто в датасетах возникает ситуация, когда некоторые поля могут быть пустыми в силу тех или иных причин. Самое простое решение – удалить наблюдения с недостающими данными. Но, если объем выборки небольшой, то это может повлиять на эффективность машинного обучения.

В рассматриваемом случае можно воспользоваться следующим «простым» методом: заполнить недостающие значения средним всех остальных значений в этой колонке.

Кодирование категориальных признаков. Категориальные признаки называют по-разному: факторными, номинальными. Их значения определяют факт принадлежности к какой-то категории. Примеры таких признаков: пол, страна проживания, номер группы, категория товаров и так далее.

Для компьютерной обработки необходимо преобразовать эти

признаки в числовые значения. Процесс преобразования категориальных признаков в числовые значения называется «кодированием».

Например, если кодируем значения в колонке «порода», то в результате такой кодировки получим, что каждый слой имеет свой собственное число, таблица 4.3.

Таблица 4.3. Кодировка пород

Порода	Значение характеристики	Мощность слоя
1	A ₁	B ₁
2	A ₂	B ₂
3	A ₃	нет
3	A ₄	B ₄
1	A ₅	B ₅
3	нет	B ₆
2	A ₇	B ₇
1	A ₈	B ₈
2	A ₉	B ₉

Вместе с тем, такой метод имеет фатальный недостаток – он проецирует категориальный признак на вещественную прямую и теряется весь смысл категориальности. Дополнительно, появляются *ложные интерпретации*. Например, в данном случае модель попытается извлечь информацию о том, что Слой_III > Слой_I или извлечь информацию о порядке следования чисел, хотя это не свойство данных о слое, а свойство выбранной нами кодировки.

Для того, чтобы избежать этого недостатка, используют *метод наивного или глупого (dummy) кодирования или one-hot-кодирования* (Dummy Encoding или One Hot Encoding). Суть его состоит в том, что для кодируемого признака создаются N новых признаков, где N – число категорий. В результате получаем сложный код, состоящий из набора нескольких чисел (2, 3 или более) для каждого кодируемого признака. Отметим, что предварительный анализ данных, кодирование признаков – тема отдельного серьезного курса.

Разделение выборки на обучающую и тестовую. Обучающая выборка используется для обучения модели. Тестовая выборка используется для проверки эффективности работы модели. Данные тестовой выборки не должны ранее встречаться в модели, модель в процессе обучения не должна соприкасаться с данными тестовой выборки.

Масштабирование признаков. Очень часто можно увидеть споры насчет того, какую операцию нужно выполнить первой – разделение выборки или масштабирование признаков.

Преобладающим мнением считается, что сначала необходимо произвести разделение выборки, а потом проводить масштабирование признаков.

Это связано с тем, что тестовая выборка должна представлять возможность имитировать реальные данные, с которыми модель раньше не сталкивалась. Так как масштабирование признаков включает в себя вычисление среднего или среднеквадратического отклонения всего датасета, то модели «передается» информация о тестовой выборке и оценка модели на тестовой выборке не может считаться «чистой».

Диапазон значений признака в датасете может очень сильно варьироваться. Некоторые алгоритмы МО чувствительны к ситуациям, когда диапазон значений одного признака существенно отличается от диапазона значений другого признака. Например, некоторые классификаторы считают расстояние между двумя точками с помощью Евклидовой метрики. Если один из признаков имеет широкий диапазон значений, расстояние будет сильно зависеть от этого признака. В этом случае необходимо выполнить нормализацию, чтобы каждый признак вносил примерно одинаковый вклад в итоговое значение расстояния.

Масштабирование можно реализовать путем стандартизированной оценки (или z-score normalization) или нормализованной оценки (rescaling или min-max normalization):

стандартизированная оценка

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation}(x)},$$

нормализованная оценка

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}.$$

Отметим, что не следует проводить масштабирование dummy-признаков, которые кодируют категориальные данные.

Чтобы разобраться в концепциях машинного обучения, нужно обладать следующими знаниями:

- Углубленные знания начального уровня алгебры: необходимо разбираться в таких понятиях, как переменные, коэффициенты, линейные уравнения, исчисления, а также графики.
- Владение основами программирования, а также опыт написания кода на Python: опыт в машинном обучении не требуется, однако необходимо уметь читать и писать код Python с базовыми конструкциями, такими как определения функций, списки, словари, циклы и условные выражения.

- Базовые знания следующих библиотек Python: NumPy; Pandas; SciKit-Learn; SciPy; Matplotlib (и/или Seaborn).

4.9. Общие проблемы машинного обучения

Существующие на сегодня проблемы в области МО можно сгруппировать следующим образом.

- Для того чтобы эффективно обучать нейросети и любые сложные алгоритмы, необходимы огромные массивы данных и технические ресурсы: серверы, специальные помещения для них, высокоскоростной интернет без сбоев, много электроэнергии. На получение нужных данных уходят годы работы и миллионы долларов. Такие затраты может позволить себе только крупная ИТ-корпорация. Открытых датасетов совсем не много, некоторые можно купить, но стоят они очень дорого;

- С ростом мощностей для сбора и обработки датасетов растут и вредные выбросы, которые производят крупнейшие датацентры;

- Данные нужно не только собрать, но и разметить так, чтобы машина точно определила, где какой объект и какие у него признаки. Это касается числовых данных, текстов, изображений. Опять же, чтобы сделать это вручную, нужны миллионные вложения. Например, у «Яндекса» есть «Яндекс.Голока» – сервис, где неразмеченные данные вручную обрабатывают миллионы фрилансеров. Такое тоже может себе позволить далеко не каждый разработчик;

- Даже если данных много и они регулярно обновляются, в процессе обучения может выясниться, что алгоритм не работает. Проблема может быть и в данных, и в самом подходе: когда машина успешно решила задачу с одними данными, но не в состоянии масштабировать решение с новыми условиями;

- Несмотря на все прорывы в глубоком обучении нейросетей, ИИ пока что не может создавать что-то абсолютно новое, выходить за рамки предложенных условий и превзойти заложенные в него способности. Другими словами, он пока что не в состоянии превзойти человека.

4.10. Перспективы машинного обучения: не начнет ли ИИ думать за человека?

Вопрос о том, не сделает ли машинное обучение Искусственный Интеллект умнее, чем человек, изначально не совсем корректный. Дело в том, что в природе нет универсальной иерархии в плане интеллекта.

Если мы говорим «станут умнее нас», значит мы подразумеваем, что существует некая единая шкала интеллекта, наверху которой находится человек, собаки пониже, а глупые голуби находятся где-то в самом низу.

Мы по умолчанию считаем себя умнее остальных существ. Получается, что человек должен превосходить нижестоящих животных во всём. Но «не все так просто». Так, например, белка способна запоминать местонахождения тысячи тайников с запасами, что не под силу даже очень умному человеку. А у осьминогов каждое щупальце способно мыслить и действовать самостоятельно. Получается, что *интеллект – это набор разных навыков, а не единая измеримая величина?* Или просто запоминание орешков в него не входит?

Так же и с ИИ: он уже превосходит нас во всем, что касается сложных вычислений, но по-прежнему не способен сам ставить себе новые задачи и решать их, подбирая нужные данные и условия. Это ограничение в последние годы пытаются преодолеть в рамках сильного ИИ, но пока безуспешно. Надежду на решение этой проблемы внушают квантовые компьютеры, которые выходят за пределы обычных вычислений.

Зато мы в ближайшем будущем сможем заметно расширить свои возможности с помощью ИИ, передавая ему рутинные и затратные операции, общаясь и управляя техникой при помощи нейроинтерфейсов.

Еще один очень важный вопрос: почему мы заранее считаем, что возможности человеческого мозга ограничены? В интернете, журналах обожают рисовать графики, на которых технологический прогресс обозначен экспонентой, а возможности людей обозначают константой. Но так ли это?

Вот давайте, прямо сейчас в уме умножьте 1680 на 950. Да, знаю, вы даже пытаться не станете. Но дай вам калькулятор, это займёт две секунды. Значит ли это, что вы только что расширили возможности своего мозга с помощью калькулятора? Можно ли продолжать их расширять другими машинами? Я вот использую заметки на телефоне — значит ли это, что я расширяю свою память с помощью машины? Получается, мы уже успешно расширяем способности нашего мозга с помощью машин. Или нет?

4.11. Примеры использования методов и технологий МО

Пример из робототехники. До недавнего времени одним из наиболее распространенных способов к обучению манипуляторов и роботов являлся подход, основанный на построении нейронной сети с обучением роботов «захвату и манипулированию» с использованием реальных объектов. Данный процесс является достаточно трудоемким и длительным. Сегодня активно развивается подход, основанный на построении математических моделей динамики и кинематики роботов и манипуляторов, их моделей прочности, надежности и предельных нагрузок с последующей реализацией в сопряженной интерактивной компьютерной модели. Используя такую модель вместо реального физического объекта для обучения нейронной

сети, можно научить робота распознавать объекты и надежно их захватывать. Кроме того, при таком подходе создается больше возможностей наделять промышленные роботы интеллектуальной функциональностью и «научить их интеллектуальному общению с человеком», приобретая в процессе обучения новые, все более сложные навыки.

Один из эффективных подходов к *созданию интеллектуального добычного комплекса*: использование имитационного обучения на базе компьютерной модели объекта и глубокого обучения с подкреплением сигналами от среды взаимодействия (массива горных пород). При этом следует сочетать чувствительности (сенсорики) и дистанционный тип взаимодействия с роботом в системе «человек — машина». Для восприятия того, как робот видит реальность через камеру, оператор использует специализированную гарнитуру виртуальной реальности.

Самообучающаяся модель пласта [6]. Нефтегазовая отрасль остается и в XXI веке одной из самых наукоемких и цифровизированных и имеет хорошо развитые технологии и численные модели для моделирования процессов нефтедобычи и управления данными процессами. Но, обилие методов, технологий и продуктов для моделирования нефте(газо)насыщенных пластов и вмещающих породной толщи, процессов нефтедобычи не решает главной проблемы – неопределенность и недостаточность имеющейся исходной информации, вытекающей из несовершенства методов ее получения, а также вечного противоборства стратегий исследования неизученных областей и применения имеющихся знаний (exploration vs exploitation), что приводит к естественному желанию максимально эффективно использовать имеющуюся информацию.

Неопределенности информации бывают разных типов, это, как и относительно понятные неопределенности измерений, как геологических, так и промысловых, численные погрешности гидродинамических симуляций, погрешности процесса адаптации геологогидродинамической модели, так и более сложные и концептуальные источники неопределенностей, как то: некорректность полуэмпирических зависимостей и формул, закладываемых в расчеты, отсутствие или несоответствие геологического реализма моделируемых подземных структур реальным.

Модели пласта, описывающие различные аспекты его функционирования (петрофизическая модель (ПФМ), гидродинамическая модель (ГДМ), сейсмогеологическая модель (СГМ), концептуальная геологическая модель (КГМ), цифровая геологическая модель (ЦГМ)), взаимосвязаны между собой, т.е. выходные данные одной модели являются входными для другой. Итоговая гидродинамическая модель (ГДМ)

объединяет все полученные знания о месторождении, адаптируется на реальные данные из месторождения (используя иногда прокси-модели типа IsobarProxy). Далее, происходит долгий ручной процесс адаптации модели на добычу, а весь процесс построения модели месторождения может занимать до полугода и требовать большого количества ресурсов.

Необходимость использования новых методов машинного обучения и анализа данных для построения модели месторождения определяется следующими обстоятельствами:

- данные измерений имеют очень разную локальность и степень надежности, их требуется объединять процедурой, которая будет настраиваться по целевым метрикам;
- данные сейсмических измерений не имеют однозначной интерпретации, процедура интерпретации в идеале должна быть настраиваемой на каждом месторождении;
- решение обратной задачи крайне неэффективно проводить в исходном пространстве кубов гидродинамических параметров месторождения размерностью порядка миллионов или даже миллиардов, поэтому требуются алгоритмы представления месторождения в пространстве параметров меньшей размерности;
- для ускорения процесса адаптации гидродинамической модели требуется значительно ускорить симуляции с возможной потерей точности;
- для адекватного учета рисков и определения конечных коридоров неопределенности по прогнозам добычи требуется проброс неопределенности из исходных данных измерений в конечные прогнозы.

Основной задачей построения Самообучающейся Модели Пласта (СМП) [6] является одновременный учет с помощью технологий ИИ максимального числа имеющихся данных с их неопределенностями для построения цифровой модели месторождения и оценки будущих показателей добычи, извлекаемых запасов и других важных параметров с доверительными интервалами для принятия обоснованных управленческих решений и, в конечном итоге, повышении экономической эффективности разработки месторождения за счет увеличения количества извлекаемых углеводородов, уменьшения числа неоптимально пробуренных скважин и уменьшения трудозатрат специалистов для построения цифровой модели месторождения. Самообучающаяся модель пласта представляет собой иерархию моделей и методов, принимающую на вход разного типа исходные данные с их неопределенностями, алгоритмы расчета добычи по исходным данным, историю добычи, проводящую автоматическую адаптацию моделей на добычу, уточняющую оценку неопределенностей и выдающую прогнозы добычи с доверительными интервалами.

Для построения СМП необходимо решить следующие задачи [24, 25]:

Задача объединения данных о месторождении.

Задача обусловленной генерации карт параметров месторождения.

Задача ускорения и уточнения процесса проксиадаптации гидродинамической модели.

Процесс адаптации является собой наиболее сложную для реализации часть проводимых работ по построению единой модели месторождения [26, 27]. Сложность процесса адаптации состоит в решении обратной задачи коррекции модели для соответствия расчетов добычи данным измерений, при этом прямая задача расчетов добычи решается обычно с помощью вычислительно сложных недифференцируемых гидродинамических симуляторов по типу Schlumberger Eclipse или tNavigator. Поэтому для решения данной задачи применимы в основном методы на основе Монте-Карло, а также для ускорения процесса – представления в сжатом пространстве признаков и разного рода методы глобальной оптимизации.

Однако в любом случае остается проблема медленности расчета симулятора и сложности всего процесса адаптации в многомерном пространстве параметров. Одним из возможных решений является предварительная адаптация с помощью быстрых упрощенных (прокси) моделей, и затем точная адаптация с помощью симулятора. Ведутся работы по прокси-адаптации гидродинамической модели месторождения с учетом неопределенностей исходных данных. Также важным направлением исследований являются ускорение и уточнение расчетов прокси-моделей с помощью физически информированных нейронных сетей, это направление является одним из наиболее многообещающих и ключевых для работы центра.

Конечной целью программы исследований СМП является сокращение времени построения модели месторождения и ее адаптации от месяцев до нескольких недель, причем большая часть действий будет выполняться в автоматическом режиме с помощью развитых для этого технологий ИИ. Для достижения этой цели требуется не только развивать отдельные части вычислительной цепочки, но также выстраивать верхнеуровневое управление процессом построения модели месторождения с помощью автоматизированных инструментов, интегрирующее в себя как классические методы, так и разные инновационные решения, доступные для использования, и позволяющие с помощью умного планирования выбрать путь решения отдельных задач в зависимости от требуемых метрик качества финального результата и имеющихся ограничений. Для решения этой задачи в центре развиваются методы иерархического моделирования и мультиагентного взаимодействия на основе технологий общего научного и инженерного ИИ.

Решение задач механики сплошных сред [28].

В настоящее время задачи МСС решаются главным образом численно с применением различных численных методов (МКЭ, МКР, МГИУ и др.). Их применение при рассмотрении большинства современных задач сопряжено с необходимостью разработки сложных программных комплексов. Однако существует альтернативный подход к решению задач МСС, основанный на поиске экстремумов целевых функционалов. Такой подход подразумевает аппроксимации некоторых функций в области их определения. Например, компоненты НДС в твердом деформируемом теле, вектор скорости и функцию потока в области течения жидкости и т.п.). ***Минимизация целевого функционала посредством аппроксимации неизвестных функций – типичная задача для машинного обучения.***

Поэтому основной целью исследований в этом случае является поиск и реализация физически обоснованного целевого функционала для машинного обучения, позволяющего моделировать состояние сплошной среды (твердого деформируемого тела, жидкости, газа).

Следует сказать, что альтернативный подход моделирования состояния объекта МСС путем решения вариационных задач требует обоснования эквивалентности классическому подходу.

Для продуктивной разработки интеллектуальных систем используется большое количество библиотек и фреймворков машинного обучения [29]. Эти программные инструменты существенно ускоряют создание прикладных продуктов. Но, как любое программное обеспечение они могут содержать ошибки и недокументированные возможности на уровне своего исходного кода или своих программных зависимостей. Такие уязвимости могут быть использованы злоумышленником для проведения атак на целевую систему. Таким образом, *обеспечение доверия к интеллектуальным системам в целом невозможно без обеспечения доверия к этим ключевым системным компонентам.*

Одним из центральных вопросов разработки систем доверенного интеллекта является проблема неустойчивости отображений, выучиваемых моделями машинного обучения, к изменениям входных данных. Даже незначительное изменение входного объекта, например, добавление невидимого глазу шума к картинке, может существенно изменить предсказание модели на новом объекте, который как внешне, так и по метрикам схожести практически не отличается от исходного объекта. Этот феномен привел к возникновению так называемых состязательных атак (adversarial attack).

Подходы к построению моделей, устойчивых к атакам, делятся на 2 класса. В первом подходе модифицируется (сглаживается) сама модель, что приводит к более высокой эмпирической устойчивости. Возможно получить теоретические гарантии на величину атаки, при которой модель не будет

менять свои предсказания. Во втором подходе изменяется способ обучения.

Повышение интерпретируемости моделей. Нейронные сети широко используются в качестве мощных инструментов моделирования, и большинство крупных поставщиков включили их в свое программное обеспечение для интеллектуального анализа данных. Моделирование, однако, является лишь частью процесса интеллектуального анализа данных. Дополнительно необходимо проанализировать влияние входных переменных на результат модели. Результаты некачественной интерпретации применяемой модели могут быть использованы злоумышленником для проведения атак на целевую систему, например, через “отравление” данных. Таким образом, обеспечение доверия к интеллектуальным системам требует обеспечения интерпретируемости применяемых моделей.

Наличие или отсутствие злоумышленника, сценарии воздействия злоумышленника на процессы жизненного цикла прикладных интеллектуальных систем, разнообразие самих прикладных задач и интеллектуальных систем требуют тщательного анализа и систематизации этих сценариев и угроз. В результате проведения такого анализа были разработаны критерии доверия к интеллектуальным системам, которые в дальнейшем будут апробированы на реальных интеллектуальных системах.

При этом доверие к прикладным интеллектуальным системам не может рассматриваться в отрыве от эффективности таких систем. Так, тривиальной является разработка модели машинного обучения, неуязвимой, например, к состязательным атакам, если не задавать минимальные требования к точности и скорости распознавания для этой модели. Таким образом, важной задачей является создание тестовой базы из наборов данных и моделей на основе реальных задач из различных прикладных областей, и разработка методик для оценки соответствия интеллектуальных систем и их компонентов требованиям в области доверия и эффективности.

5. МЕТОДЫ НЕЙРОННЫХ СЕТЕЙ

5.1. Общие положения

Методы искусственных нейронных сетей (ИНС) входят в группу методов, теоретической базой которых являются стохастические модели, основанные на численно-экспериментальном подходе.

В методах ИНС используется «непрямое» представление. Исследуемый объект представляется косвенно через систему связанных узлов, но при этом, эти узлы далеко не всегда имеют физический смысл, как и их входные параметры, и получаемые значения. Подразумевается, что модель работает по аналогии с человеческим разумом.

Замечание. Сфера деятельности, работающая с нейросетями, в настоящее время зачастую называется Deep Learning. Специалисты, которые работают в этой сфере, называются Data Scientist. Deep Learning требует знаний во многих областях математики и программирования для реализации серьёзных задач. Здесь используется и статистика, и теория вероятностей, и линейная алгебра, а также требуются знания в областях машинного обучения и науки о данных, и, конечно, в предметной области решаемых задач.

Главное достижение ИНС-методов (или приближений) состоит в том, что *в этом случае нет необходимости вводить предположение о типе функциональных отношений между основными характеристиками, описывающими рассматриваемые процессы в исследуемой среде* (как, например, в классических подходах МСС) и, кроме того, отпадает необходимость иметь точные знания о свойствах материала в рассматриваемой области.

Искусственная нейронная сеть (или нейронная сеть) представляет собой математическую модель, построенную по принципу организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма, а также её программную или аппаратно-программную реализацию. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы. Первой такой попыткой были нейронные сети У. Маккалока и У. Питтса [30]. После разработки алгоритмов обучения получаемые модели стали использовать в практических целях: в задачах прогнозирования, для распознавания образов, в задачах управления и др.

Итак, ИНС представляет собой систему соединённых и взаимодействующих между собой простых структур (искусственных нейронов). Каждый нейрон получает на вход данные, обрабатывает их, а затем передаёт эти данные другому нейрону. Абсолютно все нейросети состоят из нейронов, а соединённые разными способами в большие цепочки

они способны решать различные по сложности задачи, и разбег по этой сложности далеко не маленький.

Нейросеть — это не всегда что-то сложное и требующее больших усилий в реализации. Это вполне упорядоченная и закономерная структура, сложность которой зависит от сложности конкретной задачи.

Сегодня уже представляется распространённым мнение о том, что, так как «Deep learning» идет семимильными шагами вперёд, поэтому люди придут к тому, что большая часть профессий будет полностью заменена нейросетями, так как в нейросетях можно реализовывать задачи абсолютно разного характера – от нахождения правильного ответа в примерах, до полностью автономного искусственного интеллекта.

ИНС-технологии интересны тем, что на их основе можно создать самокорректирующиеся искусственные интеллектуальные системы, самонастраивающиеся на конкретные условия.

Центральным моментом при использовании ИНС является выбор геометрии нейронной сети [31].

ИНС не программируются в привычном смысле этого слова, они обучаются. *Возможность обучения – одно из главных преимуществ нейронных сетей перед традиционными алгоритмами.* Технически обучение заключается в нахождении **коэффициентов связей (весов)** между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными и выходными данными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искажённых данных.

Кратко искусственные нейронные сети можно определить, как сеть простых единиц (узлов, нейронов), обладающих индивидуальными свойствами и оперирующих локально. Узлы соединены связями, которые могут изменяться (сокращаться или увеличиваться) при переходе от одного узла к другому. Каждый узел (нейрон) в соответствии со своими связями получает сигналы от другого узла, обрабатывает их и передает следующим (рис. 5.1). Важно, что каждый нейрон (узел) в слое может получать несколько входных данных от нейронов предыдущего слоя, обрабатывать их и отдавать на обработку нейронам в следующем слое (рис.5.2).

5.2. Кратко из истории появления и развития ИНС

Нейронные сети как понятие начали своё существование уже довольно давно, как только устройство человеческого мозга начало изучаться: первые описания в своих исследованиях нейронным сетям дали двое американских учёных Уоррен Маккалок и Уолтер Питтс в статье “Логическое исчисление

идей, относящихся к нервной активности” в 1943-м году. Уже тогда будет предложена и объяснена схема работы нейросетей, которая является основой всех современных ИНС.

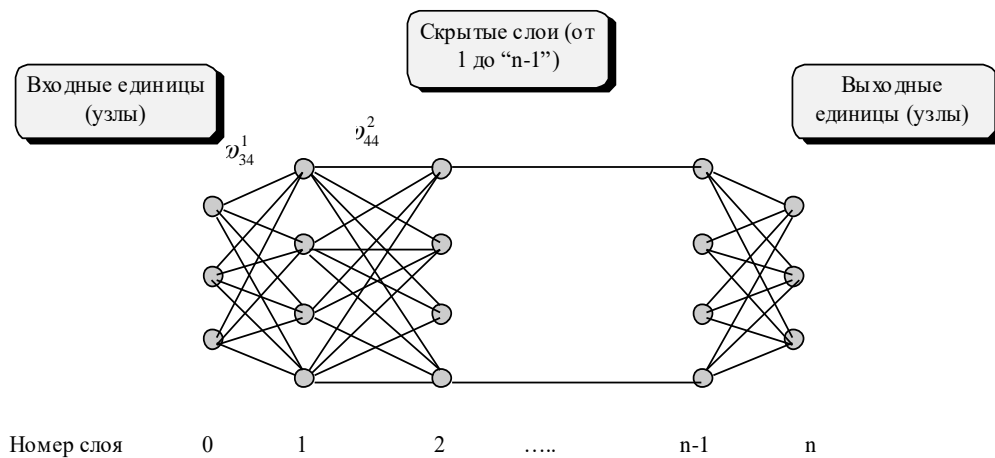


Рис. 5.1. Условная схема нейронной сети

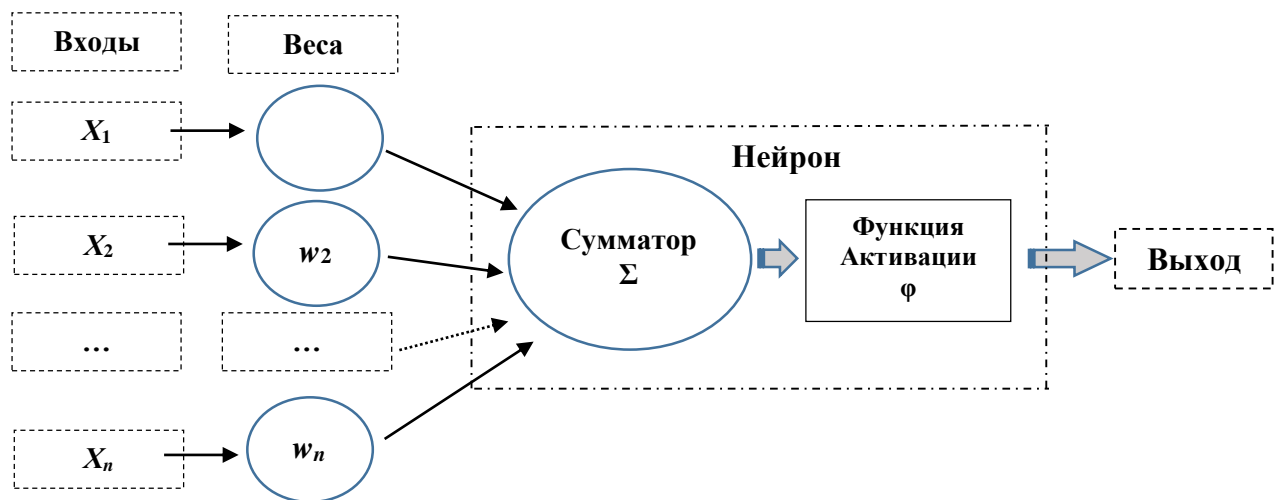


Рис. 5.2. Принцип работы искусственного нейрона

В 1948-м году выходит в свет книга Норберта Винера “Кибернетика: или Контроль и коммуникация у животных и машин”, где Н. Винер рассматривал идею обучения: программирование компьютера на анализ своих ошибок в игре в шахматы и последовательное улучшение процесса игры, тем самым сделать компьютер способным обыгрывать большинство людей, но не гроссмейстеров. Обращая внимание на техническое состояние “вычислительных машин” тех лет, реализация подобного была попросту невозможна и казалась чем-то абсолютно невероятным.

В 1949-м г. выходит книга уже другого учёного – нейробиолога Дональда Олдинга Хебба: “Организация поведения: нейропсихологическая теория”. В этой книге даётся определение синаптической пластичности –

основному механизму, который позволяет человеку запоминать информацию и обучаться чему-либо. По концепции обучения, описанной Хеббом в его исследованиях, частая стимуляция нервных клеток приводит к формированию скоординированных нейронных структур, которые развиваются посредством стимуляции нейронов между собой. Тем самым, описанный процесс преподносится как биологическая основа процесса обучения.

Через 9 лет, в 1958-м году американский учёный Фрэнк Розенблатт изобретает первую нейронную сеть и даёт ей название однослойный перцептрон, а ещё через 2 года Розенблатт построит компьютер Марк-1, основанный полностью на схеме работы перцептрона. Разработанная Розенблаттом сеть была преподнесена как возможность решения задач классификации. Однослойный перцептрон стал популярным – его начинают использовать для распознавания образов, прогнозирования погоды и т.д. Данное изобретение дало довольно сильный толчок в развитии нейронных сетей, хотя что тогда, что сейчас, однослойный перцептрон решает только несложные задачи.

В том же 1960-м году Бернард Уидроу разрабатывает ADALINE (Adaptive Linear Element) – схема нейронной сети, состоящая из одного слоя. Адалин использовался для решения задач предсказания и адаптивного управления. При создании такой нейросети, Уидроу руководствовался дельта-правилом (метод обучения перцептрона по принципу градиентного спуска по поверхности ошибки, разработал сам Уидроу), а также ему пришлось создать новый тип компонента электронной техники - мемистор. Главная особенность мемистора – регулирование сопротивления током управляющей цепи. Существует и многослойная нейросеть, состоящая из множества Адалинов – MADALINE

Несмотря на всю популярность однослойных сетей, люди быстро начали понимать, на что они способны и какие проблемы не могут быть разрешены такими схемами. Уже в 1963-м году впервые в СССР исследованием “трудных” для перцептрона задач занялся учёный Института проблем передачи информации А.П. Петров, а его труды стали основой работы другого советского учёного – Михаила Моисеевича Бонгарда. Краткая суть их трудов – выявить проблемы, с которыми сталкивается перцептрон для определённого типа задач, что делает невозможным их решение, а также сделать небольшую переделку алгоритма для возможности решения этих самых задач.

Но заставить переделку и расширить свои возможности перцептрон так и не смог – в 1969-м году американский учёный Марвин Ли Минский публикует свои исследования в области перцептрона, в которых приводит чёткие доказательства невозможности перцептроном решения задач

“чётности” и “один в блоке”. После выхода исследовательской работы интерес к ИНС резко падает.

Несмотря на спад интереса к ИНС, преданные своему делу люди всё ещё продолжали трудиться и предлагать новые решения, однако до 1982-го года на ИНС всё ещё никто не обращал внимания, хотя в 1974-м был изобретён алгоритм обратного распространения ошибки для обучения многослойных перцептронов. В то время алгоритм не заставил ИНС подняться на пик популярности, однако он ещё даст этому толчок в более позднее время.

В 1975-м японский информатик и учёный Фукушима Кунихико выводит во свет самоорганизующуюся нейросеть для распознавания образов - когнитрон. Архитектура этой сети была похожа на строение зрительной коры головного мозга и организовывалась по принципу её работы. Минус бы в том, что для распознавания состояний образа сеть уже должна была знать все его состояния, что было довольно ресурсозатратно для компьютеров того времени.

Наступает 1982-й год, интерес к ИНС снова возрастает благодаря двум людям, сделавшим открытия – американский учёный Джон Джозеф Хопфилд разрабатывает сеть с алгоритмами обучения, отличными от привычных алгоритмов обучения перцептронов (впоследствии сети такого типа будут называться сетями Хопфилда), а финский учёный Теуво Калеви Кохонен разрабатывает структуру сети, позволяющую решать задачи кластеризации и визуализации данных.

Далее, в 1986-м году, метод обратного распространения ошибки обретает вторую жизнь благодаря Дэвиду Руммельхарту и Джеффри Хинтону. Второе дыхание алгоритма дало бум популярности нейросетей (относительно предыдущих лет), и, вследствие, огромный толчок развития. Метод используется до сих пор во многих типах нейросетей.

Следующий толчок развития датируется 2012-м годом, когда пара американских учёных создала нейросеть, которая распознавала высокоуровневые сложные концепты (например, коты) только взглянув на непомеченные изображения. Также развитие компьютеров в техническом плане, развитие технологий обучения без учителя и распределённых вычислений достигло точки, когда на нескольких машинах можно было использовать довольно большие по структуре сети, в частности, для решения задач распознавания объектов на изображениях, что получило название глубокого обучения.

5.3. Архитектура (геометрия) нейросетей

Архитектура нейросетей может быть с разными принципами работы и передачи информации. Приведем некоторые из *популярных и применяемых*

сегодня типов геометрии нейросетей [31, 32].

1. Многослойный перцептрон (персептрон)

Самая известная и достаточно уже «старая» архитектура, в которой идут подряд несколько слоев нейронов – входной, один или несколько скрытых слоев и выходной слой.

Такая нейросеть почти всегда обучается методом обратного распространения ошибки (back propagation), что автоматически означает, что мы должны предоставить для обучения набор пар «входной вектор – правильный выход». Когда входной вектор «отправляется» на вход сети, последовательно рассчитываются состояния всех промежуточных нейронов и на выходе образуется выходной вектор, который и сравнивается с правильным. Расхождение дает ошибку, которую можно распространить обратно по связям сети, вычислить вклад в итоговую ошибку каждого нейрона, и скорректировать его веса, чтобы исправить сеть. Повторив эту процедуру многократно, создаются условия для обучения сети. Таким образом, обучение такой сети достигается за счёт нескольких сотен – тысяч повторений коррекции ошибки.

Сети такого типа очень эффективны при решении задач, где:

1. Ответ действительно зависит только от того, что подается на вход сети и никак не зависит от истории входов (т.е. это не динамический процесс, или, по крайней мере, на вход подается исчерпывающая информация об этом процессе в форме, пригодной для обработки сетью).

2. Ответ не зависит или слабо зависит от высоких степеней и/или произведений параметров (так как функции этого типа данная сеть строить почти не умеет).

3. В наличии имеется достаточно много примеров (желательно иметь не менее сотни примеров на каждую связь сети), или у вас есть большой опыт борьбы с эффектом специализации. Это связано с тем, что, имея много коэффициентов, сеть может банально запомнить много конкретных примеров и выдавать на них отличный результат. Но ее прогнозы не будут иметь ничего общего с реальностью в случае, если дать на вход примеры не из обучающей выборки.

«Сильные стороны» такого типа сети. Сеть типа «многослойный перцептрон» изучена достаточно глубоко. Класс задач, для которых сети такого типа эффективно использовать, хорошо известны.

Среди недостатков таких сетей: неумение работать с динамическими процессами, необходимость в наличии большой обучающей выборки.

Поэтому сети типа «многослойный перцептрон» не имеют существенных перспектив в будущем. Большинство современных задач, которые перспективно решать на базе ИНС, не входят в класс задач,

решаемых многослойным перцептроном с методом обратного распространения ошибки.

2. Рекуррентный перцептрон

Архитектура строения данного типа нейросетей очень похожа на обычный перцептрон, единственное существенное отличие состоит в том, что его выходы попадают ему же на входы и участвуют в обработке уже следующего входного вектора данных. То есть в случае рекуррентного перцептрона имеет место не набор отдельных, ничем не связанных образов, а некоторый процесс, и значение имеют не только сами входы, но и то, в какой последовательности они поступают. Из-за этого возникают отличия в методе обучения – используется то же самое обратное распространение ошибки, но для того, чтобы ошибка попала по рекуррентной связи в прошлое, используются разные «ухищрения» (если подойти к задаче «в лоб», то возникнет проблема ухода ошибки на бесконечное число циклов назад). В остальном же ситуация похожа на обычный перцептрон.

Для обучения нейросети в данном случае нужно иметь достаточно длинную последовательность пар вход–выход, которую необходимо много раз «прогнать» через сеть, чтобы ее обучить (или же иметь под рукой математическую модель искомого процесса, которую можно тестировать при различных возможных условиях и в реальном времени давать результаты сети для обучения).

Сеть такого типа хорошо «решает» задачи управления динамическими процессами (классический пример – решение задачи о стабилизации перевернутого маятника), прогноза развития динамических процессов. Перспективно использовать такой тип сетей для задач, в которых помимо явно наблюдаемого входа у системы есть некоторое внутреннее состояние, которое не совсем понятно, как использовать.

На текущий момент алгоритмы, реализующий данный тип сетей, еще недостаточно хорошо проработаны. Так, в ситуации, когда сеть «не работает», весьма затруднительно разобраться, в чем заключается проблема. В процессе обучения такой сети может возникнуть ситуация «самовозбуждение» – когда сигнал, полученный с выхода, забивает все, что приходит по входам.

В целом «на сегодня» рекуррентные перцептроны используются довольно редко, хотя их потенциал высок. Хорошие перспективы имеет подход с непрерывно адаптирующейся к объекту управления сетью, хотя для этого еще необходимо решить проблему неустойчивости обучения.

3. Ассоциативная память

«Ассоциативная память» представляет собой широкий класс сетей, состоящих из одного (реже двух) слоя нейронов. Входной слой в таком типе архитектуры является же и его выходом, поэтому количество слоёв в такой

сети немногочисленно (рис.5.3). Выходы нейрона поступают на его входы в следующий момент времени. Этот слой служит и входом сети (в начальный момент выходы нейронов принимаются равными входному вектору) и её выходом (значения на нейронах, образовавшиеся в конце работы, считаются ответом сети).

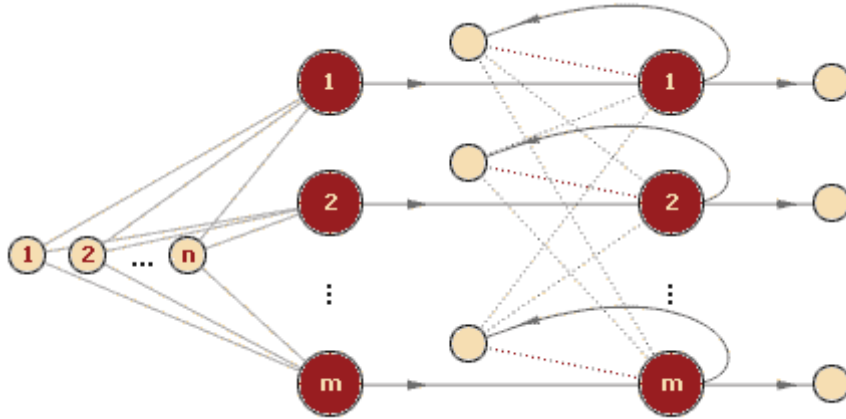


Рис. 5.3. Пример ассоциативной памяти – нейросеть Хэмминга

Эта сеть меняет свои состояния с течением времени до тех пор, пока состояние не перестанет меняться. Свойства весовой матрицы выбираются таким образом, чтобы устойчивое состояние всегда гарантированно достигалось (обычно это происходит за несколько шагов). Такая сеть «помнит» некоторое количество векторов и при подаче на вход любого нового вектора может определить на какой из ранее имеющих (запомненных) он более всего похож (отсюда и соответствующее название сети). Двухслойная модификация этой сети (*гетероассоциативная память*) может запоминать вектора не по одному, а по парам разной размерности.

Сети такого типа хорошо зарекомендовали себя применительно к задачам, где нужно определить «похожесть» вектора на один из стандартных, хранящихся в памяти сети. Собственно, это практически единственный класс задач, где такие сети эффективны.

Например, весьма перспективно использовать такой тип сети при анализе и обработке аэро- и космоснимков при построении систем мониторинга процессов и объектов.

К сильным сторонам такого типа сетей можно отнести: очень быстрое обучение (т.к. вместо градиентного спуска решается система уравнений), возможность удаления образа из памяти или добавления в память нового, не затронув остальные. Некоторые свойства такого типа памяти напоминают свойства человеческого мозга, поэтому изучение их свойств интересно и с такой позиции.

К существенным «слабостям» данного типа сетей можно отнести очень

узкий класс решаемых задач, неумение обобщать примеры, максимальный объем памяти жестко связанный с размерностью запоминаемого вектора.

Вместе с тем, нейросети типа «ассоциативная память» имеют достаточно широкое распространение вследствие:

- разработана ядерная ассоциативная память, которая способна к обобщению образов и имеет неограниченный объем памяти (сеть растет по мере ее заполнения);

- разработана динамическая ассоциативная память, которая запоминает не отдельные образы, а определенные последовательности образов и поэтому может применяться для распознавания элементов динамических процессов;

- динамическая ассоциативная память демонстрирует способность к генерации отклика, содержащего разные элементы запомненных последовательностей при подаче входного сигнала, соответствующего одновременно разным последовательностям (что, возможно, является некоторой грубой моделью творчества человека);

- гибрид ядерной и динамической ассоциативной памяти может дать новое качество в распознавании последовательностей (например, в распознавании речи).

4. Спайковые сети

Спайковые сети представляют собой особый класс сетей, в которых сигнал представлен не вещественным числом, как во всех ранее рассмотренных сетях, а набором импульсов (спайков) одинаковой амплитуды и длительности, и информация содержится не в амплитуде, а в интервалах между импульсами, в их паттерне. Спайковые нейроны на выходе генерируют одиночные импульсы (если суммарный вход не очень большой), или пакеты импульсов (если суммарный вход достаточно велик). То есть в них информация кодируется не числовыми значениями активации нейронов, как в классических ИНС, а временем появления спайков – коротких импульсов.

В спайковых сетях используются специальные нейроны – спайкин-нейроны. Они генерируют спайки при превышении входного сигнала определенного порога.

Для обучения спайковых сетей используются следующие основные подходы: обратное распространение ошибки; нейроэволюционные алгоритмы; правило Хебба; STDP (spike-timing-dependent plasticity, изменение весов в зависимости от времени прихода спайков).

Наиболее часто применяется обратное распространение ошибки, как и для классических ИНС и правило Хебба. Остальные подходы пока менее распространены, но активно исследуются.

Правило Хебба примерно формулируется следующим образом: если

второй нейрон сработал сразу после первого, то связь от первого ко второму усиливается, а если сразу перед первым, то связь ослабевает.

Рассмотрим принцип работы спайковых нейронных сетей на простом примере. Пусть имеется сеть из трех нейронов. Сигнал подается на входной нейрон, который генерирует спайки с определенной частотой. Эти спайки передаются на следующие нейроны и либо ускоряют, либо замедляют генерацию спайков в них. Таким образом происходит передача и обработка информации в спайковой сети. Главное отличие от обычных ИНС в том, что здесь нет конкретных числовых значений активации нейронов. Вся информация закодирована во временных моделях спайковой активности.

Спайковые сети могут применяться для решения многих практических задач. Особенно они полезны там, где требуется работа с потоками данных в реальном времени.

Основные области применения спайковых ИНС: распознавание и обработка изображений; анализ видеопотоков; обработка естественного языка; распознавание речи; системы управления движением и др.

Спайковые ИНС активно используются в робототехнике для обработки данных с сенсоров и управления движением. Они позволяют роботам эффективнее взаимодействовать с окружающей средой.

Также спайковые сети применяются в системах распознавания речи и изображений, работающих на мобильных устройствах. Благодаря низкому энергопотреблению, они могут эффективно использоваться в смартфонах и других гаджетах.

Использование спайковых сетей имеет хорошие перспективы при рассмотрении химико-биологическим процессов.

Применение такого типа сетей при рассмотрении процессов и явлений в механике пока весьма ограничено.

Среди достоинств спайковых сетей, помимо упомянутых: высокая скорость обучения за счет одновременной передачи множества сигналов; лучшее соответствие структуре и функционированию биологических нейронных сетей; возможность динамического добавления новых нейронов и связей; устойчивость к повреждениям отдельных нейронов.

То есть спайковые ИНС в целом ближе к работе мозга и обладают лучшей масштабируемостью и гибкостью.

В более отдаленной перспективе, после создания подходящего алгоритма обучения, такие сети по функциональности сравняются или даже превзойдут другие типы нейросетей, а еще позднее на их основе можно будет собирать структуры, пригодные для прямого подключения к биологическому мозгу, для расширения возможностей интеллекта. Сборки спайковых сетей смогут построить целый самостоятельный искусственный интеллект или расширять функции человеческого интеллекта.

5.4. Обучение и настройка нейронных сетей

Нейросети различаются не только по методам передачи информации между нейронами, но и по способам обучения нейронов. Естественно, что для разных задач и архитектур применяются разные методы.

“Обучение” нейронной сети является первым этапом построения сети. В итоге такого “обучения” нейронная сеть может полностью воспроизводить экспериментально полученные используемые для обучения результаты. При своем формировании на этапе “обучения” сеть работает в автоматическом режиме.

Существует два различных вида “обучения”: наблюдаемое и ненаблюдаемое. При “наблюдаемом обучении” правильные ответы (например, значения натуральных/экспериментальных данных) известны заранее и сеть “изучает” их.

С точки зрения математики, обучение нейронных сетей — это многопараметрическая задача нелинейной оптимизации.

Примеры распространенных методов обучения нейросетей [32, 33]:

1. Метод обратного распространения ошибки (Back Propagation)

Принцип работы данного метода заключается в реализации градиентного спуска в пространстве весов (т.е. осуществляется постоянная их корректировка для получения минимума ошибки).

В нейросети подаются входные и желаемые выходные данные, инициализация весов происходит с помощью генератора случайных чисел. Далее вычисляются выходные данные с помощью функции активатора, а также отклонение (разность) полученного выхода от желаемого выхода, которое называется ошибкой. На основе этой разности происходит коррекция весов, затем алгоритм выполняется снова, пока веса не приобретут оптимальные значения для решения задачи. Подразумевается, что в процессе обучения алгоритм проходит огромное количество циклов (от 1000 до нескольких десятков тысяч).

К основным недостаткам данного метода можно отнести следующие обстоятельства. Так как обратное распространение использует градиентный спуск, то в сложных сетях может быть найден локальный минимум вместо глобального, тем самым нейросеть может завершить обучение преждевременно и работать некорректно. Кроме того, в процессе обучения веса могут стать очень большими величинами. Это может привести к тому, что нейроны будут работать с большими значениями на областях, где ошибка будет получаться очень маленькой, тем самым коррекция весов будет несущественной. В результате сеть будет обучаться слишком долго, либо вообще перестанет обучаться.

2. Метод упругого распространения ошибки (Resilient Propagation)

Данный метод работает аналогично методу обратного распространения

ошибки за исключением того, что веса корректируются немного иным способом: если знак ошибки положительный, то весовой коэффициент уменьшается на величину коррекции, в противном случае он увеличивается на величину коррекции.

Отметим, что метод упругого распространения ошибки работает в 4-5 раз быстрее, чем метод обратного распространения ошибки.

К недостаткам данного метода, как в случае и метода обратного распространения ошибки, можно отнести повышенный шанс попасть в локальный минимум вместо глобального. В этом случае имеет место, повышенный шанс завершить обучение, когда оно еще требует продолжения.

3. Генетический алгоритм

Чаще всего генетический алгоритм используется для решения задач оптимизации и моделирования путём случайного подбора. Алгоритм использует механизмы, аналогичные естественному подбору, откуда и получил свое название.

Алгоритм принимает входные данные, которые называются начальной популяцией. Затем с помощью функции приспособленности (активатора) начальным данным даётся оценка. С учетом этой оценки отбираются наиболее удачные варианты, затем происходит их скрещивание по заданным параметрам, в результате чего появляется новое улучшенное множество решений. Новое множество оценивается повторно функцией-активатором. Если новое поколение решений удовлетворяет параметрам выхода из алгоритма, выполнение прекращается. В противном случае алгоритм заново повторяет процедуру с последними получившимися данными.

Недостатки данного метода заключаются в следующем. Решение может быть более пригодным лишь по сравнению с другими решениями, условия для остановки алгоритма могут быть неясны и некорректны. Множество решений может сойтись в процессе к локальному оптимуму, а не к глобальному. Алгоритм плохо масштабируется под сложность решаемых задач, т.е. чем больше область поиска решений, тем больше данных подвержены «мутации». Тем самым, полученные данные могут быть необъективными или вовсе непригодными к использованию. Если разделить задачу на более мелкие подзадачи, то выходные данные с каждой такой подзадачи должны быть совместимы друг с другом, чего добиться данным алгоритмом довольно сложно, т.к. конечные данные с каждой подзадачи могут быть несовместимы с последующими мутациями. Метод требует много временных ресурсов, т.к. обычно такой алгоритм работает с большими объёмами данных и один пуск функции для оценки данных может занимать от нескольких часов до нескольких дней.

5.5. Функции-активаторы

Неотъемлемой частью нейросетей являются **функции-активаторы**, которые поставляют нейронам выходные данные. Без функций-активаторов нейросеть просто не сможет обучиться, т.к. без выходных данных, которые должны быть приняты следующим нейроном, сеть просто не поймёт, от чего ей отталкиваться.

Активаторы также имеют свою классификацию и область решаемых задач.

1. Ступенчатая функция

Ступенчатая функция (рис.5.4) представляет собой простейшую функцию активации, которая может работать по одному из следующих двух условий:

- функция активирована, если переменная находится в области, большей нуля;
- функция возвращает 1, если переменная находится в области, большей нуля, иначе функция возвращает значение 0.

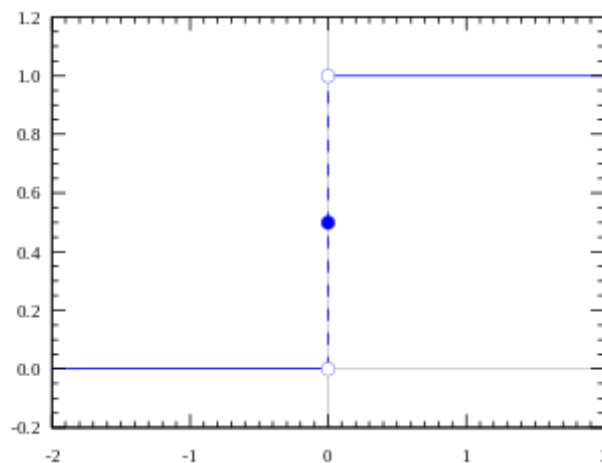


Рис. 5.4. Ступенчатая функция

Ступенчатая функция совершенно не подходит для сложных задач и при использовании нейросетей с большим количеством нейронов в одном слое, т.к. может активировать несколько нейронов сразу. Как следствие этому, сеть будет работать некорректно из-за конфликта данных.

2. Линейная функция

Линейная функция представляет собой прямую линию и пропорциональна входу (взвешенной сумме на нейроне) (рис.5.5).

В отличие от ступенчатой функции линейная функция позволяет получать спектр значений, а не только один ответ из двух.

Данная функция не подходит для работы с градиентным спуском, так как производная функции в любой точке равна одному и тому же значению, соответственно, нейросеть стоит всегда в одной точке градиента. Кроме

того, она не подходит для работы с большим количеством слоёв, так как линейная комбинация линейных функций – это тоже линейная функция. В таком случае нужно либо сокращать количество слоёв, либо использовать другой активатор.

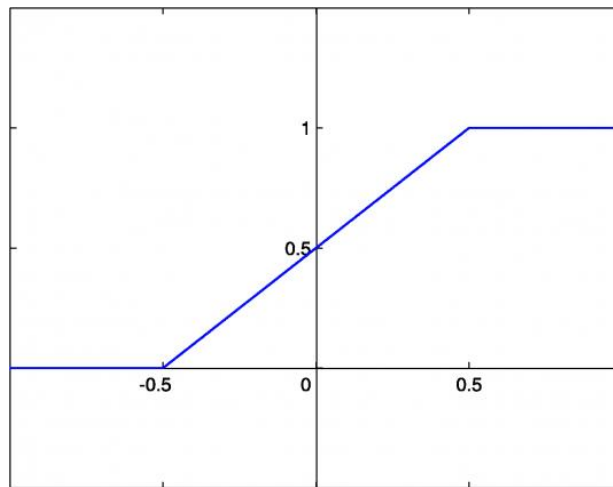


Рис. 5.5. Линейная функция $f = cx$

3. Сигмоидальная функция

Сигмоидальная функция – гладкая функция, в некотором роде подобна ступенчатой функции, но, в отличие от неё, не является бинарной, что позволяет получать спектр значений (рис.5.6). Сигмоидальная функция по своей природе нелинейна, а значит, подходит для нейросетей с большим количеством слоёв.

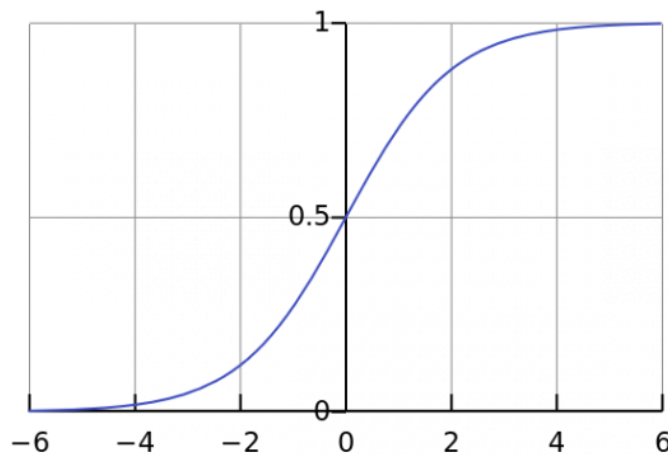


Рис. 5.6. Сигмоидальная функция $f(x) = (1 + e^{-x})^{-1}$

Сигмоидальная функция, в отличие от линейной, не приводит к ошибкам при больших значениях активации. Данная функция стремится привести значения к одной из сторон кривой (например, при $x = 2$ к верхнему и при $x = -2$ к нижнему), что позволяет получить чёткие границы при предсказании.

Вместе с тем, из-за слабой реакции значений функции по отношению к x , при больших значениях $|x|$ в нейросетях, использующих в своих алгоритмах градиент, могут появляться проблемы с обучением. Обучение в таких сетях может происходить слишком долго, либо вообще не происходить.

4. Гиперболический тангенс

Гиперболический тангенс по существу является скорректированной сигмоидальной функцией (рис.5.7).

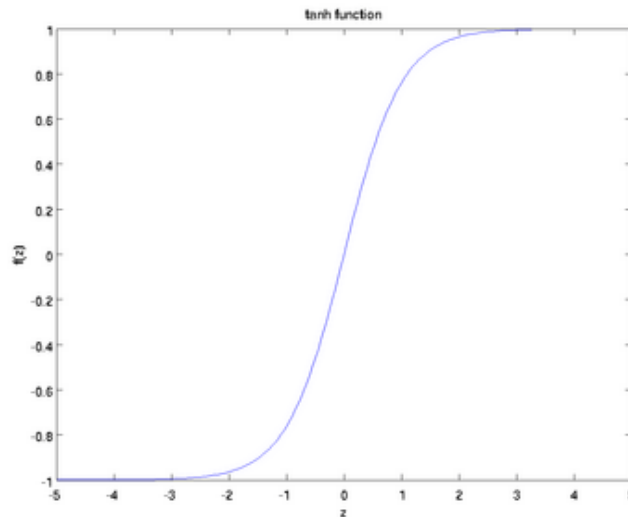


Рис. 5.7. Гиперболический тангенс $f(x) = \left(1 + e^{-2x}\right)^{-1} - 1$

Гиперболический тангенс имеет больший градиент, чем у сигмоидальной функции.

Гиперболическому тангенсу присущи те же недостатки, что и сигмоидальной функции.

5. ReLu

Функция ReLu возвращает значение x , если x положительно, и 0 в противном случае (рис.5.8).

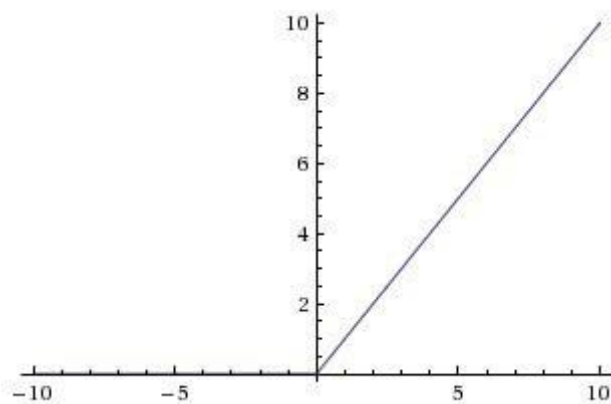


Рис. 5.8. Функция ReLu: $f = \max(0, x)$

На первый взгляд может показаться, что ReLu имеет проблемы, аналогичные линейной функции, т.к. она линейна в первом квадранте. На самом деле, она не является линейной по своей природе, и комбинация этих двух функций тоже не является линейной, что сразу отбрасывает все проблемы линейной функции. Функция ReLu является менее ресурсозатратной по сравнению с сигмоидальной функцией и гиперболическим тангенсом, т.к. не требует активации некоторых нейронов из-за возвращения 0 для отрицательных x , что позволяет использовать большие нейросети на относительно слабых машинах.

К недостаткам данной функции можно отнести то, что так как при отрицательных x функция ReLu представляет собой горизонтальную линию, поэтому градиент в этой части будет равен 0, что усложняет использование ReLu в алгоритмах с градиентным спуском, т.к. веса попросту не будут корректироваться. Однако существуют способы, помогающие избежать эту проблему. Основная идея этих способов – сделать градиент неравным нулю и постепенно восстанавливать его во время тренировки.

5.6. Использование методов ИНС для решения задач моделирования механических процессов

При построении алгоритмов решения задач механики на основе технологий ИНС удобно использовать сеть, построенную по принципу “многослойный перцептрон с распространением вперед” (МРВ), рис.5.1. На этапе “обучения” при этом эффективно использовать “наблюдаемое обучение”. Как указывалось ранее, МРВ сети обычно применяются для аппроксимации неизвестных функциональных зависимостей между входными единицами (исходные данные) и выходными (характеристики изучаемого процесса).

Опишем алгоритм работы МРВ сети. Связь между узлами сети представляется весами ω_{ij}^k , где индекс “ i ” соответствует номеру узла “ $k-1$ ” слоя, а индекс “ j ” – номеру узла, принадлежащего “ k ” слою. Номер слоя входа – “0”, выходного слоя – “ n ”. Сигналы передаются только в одном направлении: по направлению от входного слоя к выходному.

Значение в узле является композицией соответствующего веса и добавочной величины сигнала в узле предыдущего слоя:

$$y_i'^k = \sum_{j=1}^{n_{k-1}} \omega_{ij}^k y_j^{k-1}.$$

Данная величина в дальнейшем подвергается трансформации функцией активизации:

$$y_i^k = f(y_i'^k) = f\left(\sum_{j=1}^{n_{k-1}} \omega_{ij}^k y_j^{k-1}\right).$$

Введение функции активизации необходимо для наличия возможности моделирования произвольных линейных соотношений между входными и выходными узлами.

В качестве функции активизации может быть использована практически любая нелинейная функция, но рекомендуется, чтобы данная функция была дифференцируема и ограничена. В качестве таковой часто принимаются гиперболические функции или Гауссовы. Например, для большого класса задач механики одним из обычных выборов функции активизации является сигмоидальная функция, т.е.

$$f(y) = \frac{1}{1 + e^{-y}}.$$

Естественно, что конечный результат работы нейронной сети зависит от значений весовых коэффициентов ω_{ij}^k , которые определяются при выполнении процедуры “обучения сети”.

“Обучение” нейронной сети. Обучение нейронной сети осуществляется на основе известных значений входной и выходной информации, то есть известных значений входных и выходных узлов. Такое множество известных входных и выходных величин называется множеством входных–выходных пар.

“Наблюдаемое обучение” представляет собой в действительности общую оптимизационную задачу, в которой находится минимум ошибки E_p , определяемой следующим уравнением:

$$E_p = \frac{1}{2} \sum_{i=1}^{n_0} (t_{p_i} - y_{p_i}^n)^2, \quad (5.1)$$

где t_{p_i} – реальные выходные значения, а $y_{p_i}^n$ – выходные значения, полученные при работе сети, n_0 – число выходных единиц.

Задача сводится к нахождению таких весовых функций ω_{ij}^k , чтобы ошибка E_p , определяемая как решение уравнения (5.1), была минимальна.

Замечание. Данная задача в свою очередь является весьма непростой, так как функция E_p , как функция многих переменных ω_{ij}^k является нелинейной и может иметь большое число локальных минимумов.

Подготовка входных–выходных пар.

Все множество входных–выходных пар разделяется на два подмножества. Первое определяется как “подмножество для обучения”, а второе – как “подмножество для тестирования”.

Такое разделение выполняется случайным образом, чтобы исключить элемент субъективизма. Обычно “подмножество для обучения” больше, чем “подмножество для тестирования”.

Название этих подмножеств соответствует их назначению: первое

используется для “обучения” сети, а второе - для проведения процедуры тестирования и проверки адекватности работы сети.

Выбор оптимальной геометрии сети.

Самым простым подходом при выборе наиболее эффективной геометрии сети – создание нескольких сетей и взаимное их сравнение.

Хотя процесс выбора геометрии сети во многом определяется пристрастиями исследователя, но при этом все же существуют общие правила, которым все же необходимо следовать. Приведем некоторые из них.

✓ Так, если количество узлов в сети очень велико, то обучающая процедура может быть весьма медленной, при этом каждое последующее вычисление на новом слое влечет за собой вычислительную ошибку.

✓ Хотя сети большого размера и позволяют в конечном итоге найти разыскиваемые соотношения, но они могут быть плохо обусловленными. Большая сеть может легко воспроизводить подмножество обучающих входных–выходных пар, но при этом она может “плохо работать” при воспроизведении малого по объему подмножества пар для тестирования.

✓ Сети с малым количеством узлов в свою очередь могут иметь проблемы в построении надежных функциональных зависимостей при выполнении процедуры “обучения”.

Существуют искусственные приемы для выбора наиболее оптимальной геометрии сети, которые можно разделить на такие две группы:

1. Построение сети, начиная с выбора сети с малым количеством узлов, и последовательным добавлением новых скрытых слоев узлов до тех пор, пока не добиваемся наиболее оптимальной архитектуры сети.

2. Построение сети, начиная с выбора сети с большим количеством узлов, с последующим исключением “бесполезных” узлов.

Отметим, что построение нейронных сетей может быть выполнено с помощью общих математических пакетов (Matlab, Mathematica, Maple).

Используя нейронные сети и эмпирические методы, включая эмпирические критерии прочности, можно выполнять исследование НДС тел/конструкций, не прибегая к решению основных разрешающих уравнений состояния и не решая при этом какие-либо уравнения МГЭ, МКЭ или МДЭ.

Такие «непрямые» технологии моделирования имеют свои достоинства и недостатки.

К достоинствам следует отнести, например, то, что:

- геометрические и физические связи в задачах, проявляющиеся в разрешающих уравнениях и уравнениях состояния при использовании «прямого» отображения геометрии и процессов, больше не являются настолько важными;

- при решении задачи могут использоваться различные нейронные сети и эмпирические модели;
- существует возможность того, что восприятие, характерное человеческому разуму, может быть симитировано через нейронные сети таким образом, что программа сможет генерировать различные оценки, основываясь на эмпирических методах и опыте.

К недостаткам данных методов относится то, что:

- данная процедура может быть рассмотрена просто как «аппроксимация сложной кривой» (потому что программе необходимо задавать команды);
- модель не может давать надежные оценки, когда параметры лежат вне области, определенной при разработке модели;
- при разработке модели могут быть не учтены критические процессы;
- наблюдается недостаток теоретического обоснования верификации и подтверждения методов и даваемых ими результатов.

Метод нейронных сетей обладает большим потенциалом из-за «непрямого» характера представления геометрии и процессов, и поэтому в будущем, может быть, станет возможным создать такие нейронные сети, которые смогут самостоятельно генерировать методы решения, воспринимать и выносить оценки. Тем не менее, **данные методы в настоящее время не являются альтернативой обычному моделированию.**

5.7. Пример построения простой нейросети

Рассмотрим несложный пример. Пусть у нас есть некая задача со входными данными, аналитическое решение этой задачи и пример для самостоятельного решения (рис.5.9).

Задача				
	Ввод			Вывод
Пример 1	0	0	1	0
Пример 2	1	1	1	1
Пример 3	1	0	1	1
Пример 4	0	1	1	0
Пример 5	1	0	0	?

Рис. 5.9. Пример задачи для решения с использованием ИНС

Как такую задачу решает человек? Так как в данном случае имеются примеры с решениями, поэтому человек находит взаимосвязь между входными данными и выходными, а затем решает пример 5 по аналогии этой взаимосвязи. По такому же принципу действует и нейросеть, где процесс нахождения взаимосвязи и называется обучением.

Несложно догадаться, что ответ на пример 5 будет 1, так как выводной элемент представляет из себя значение первого элемента списка в каждом примере.

Теперь задачу нужно реализовать в коде. Так как задача по своей структуре несложная и не требует сложных способов реализации, воспользуемся простыми и эффективными для данного типа задач алгоритмами. В качестве способа передачи информации выберем многослойный перцептрон, алгоритма обучения – обратное распространение ошибки (Back Propagation), а функции активации – сигмоидальную функцию.

С каждого примера берётся элемент x_i из списка и для каждого из них выявляется вес ω_i . Далее эти значения попадают в **сумматор**, где производится суммирование произведения x_i на ω_i (т.к. в каждом списке по 3 элемента, суммирование в данном случае будет состоять из 3-х произведений). Затем результат суммирования подставляется в функцию активации, которая будет определять выходной сигнал с каждого нейрона (рис.5.2, 5.10).

Затем необходимо написать код со способом обучения сети, а точнее метод инициализации весов. Код метода обратного распространения ошибки выглядит следующим образом, рис.5.11.

```
1 import numpy as np # подключаем библиотеку NumPy
2
3
4 def sigmoid(x):
5     return 1 / (1 + np.exp(-x)) # инициализируем функцию - сигмоид
6
7
8 training_inputs = np.array([[0, 0, 1],
9                             [1, 1, 1],
10                            [1, 0, 1],
11                            [0, 1, 1]]) # массив входных данных (примеров)
12
13 training_outputs = np.array([[0, 1, 1, 0]]).T # транспонированный массив ответов на примеры
```

Рис. 5.10. Подготовка данных для реализации

```

15 synaptic_weights = 2 * np.random.random((3, 1)) - 1 # инициализируем массив весов с помощью генератора случайных чисел
16 # Метод обратного распространения
17 for _ in range(20000):
18     input_layer = training_inputs
19     # подставляем скалярное произведение весов и входных данных в функцию активации
20     outputs = sigmoid(np.dot(input_layer, synaptic_weights))
21     # вычисляем ошибку
22     err = training_outputs - outputs
23     # вычисляем корректировки к весам
24     adjustments = np.dot(input_layer.T, err * (outputs * (1 - outputs)))
25     # корректируем веса
26     synaptic_weights += adjustments

```

Рис. 5.11. Реализация метода обратного распространения ошибки

Алгоритм получает массив примеров в отдельную переменную, затем подставляет скалярное произведение имеющихся весов и значений примеров в сигмоидальную функцию. Получившиеся результаты отнимаются от реальных должных результатов, затем в отдельную переменную записываются корректировки к имеющимся весам, затем эти корректировки суммируются с имеющимися весами. Таким образом алгоритм будет исполняться 20000 раз (один такой цикл называется *эпохой*), чтобы корректировки весов были более точными, т.к. за одно исполнение веса не смогут скорректироваться сразу до нужных значений.

Здесь может возникнуть закономерный вопрос: а почему инициализация весов “рандомизирована”? Т.к. не существует универсальных методов для точного подбора весов, который бы гарантировал нахождение наилучшей начальной точки для любой решаемой задачи, то такие точки в большинстве реализаций получаются путём случайного подбора в каком-либо интервале с равномерным распределением на этом интервале. Также важно подобрать интервал, иначе нейросеть может обучаться слишком долго, либо вовсе обучиться неправильно.

После процесса обучения нейросеть уже даёт нам ответы, максимально приближённые к реальным выходным данным (рис.5.12):

Python Console

Данные после обучения:

```

[[0.00679838]
 [0.99445493]
 [0.99548631]
 [0.00553517]]

```

Рис. 5.12. Данные после обучения

Отсюда можно сделать вывод, что нейросеть успешно обучена и готова к выполнению задач такого типа. Дадим ей новые данные, которые раньше не встречались в процессе обучения, т.е. данные из примера 5 (рис. 5.13).

```

27 # Даём сети новые данные для решения задачи
28 new_inputs = np.array([1, 1, 0]) # новый массив для решения
29 output = sigmoid(np.dot(new_inputs, synaptic_weights))
30
31 print('Решение новой задачи: ', output)
32

```

Рис. 5.13. Новые данные для нейросети к решению

Так как веса уже подобраны и корректировки для них не требуются, можно подставить данные в функцию–активатор всего один раз. Нейросеть дает ответ (рис.5.14):

Решение новой задачи: [0.99996184]

Рис. 5.14. Построенное по новым данным решение

Как видно, значение решения очень близко к единице, что и является правильным ответом.

5.8. Библиотеки для работы с нейросетями в Python

Нейросеть, описывающая пример из предыдущего пункта, является самой простой в реализации нейросетью, а реализация алгоритмов её обучения была написана вручную без использования библиотек (кроме библиотеки NumPy, т.к. она подключалась для работы с данными для сети, а не для написания самих алгоритмов обучения). Однако существуют задачи, в которых нужно реализовать более сложные вычисления, работать с большими объёмами данных, реализовать более сложные алгоритмы, т.к. того требует задача, и чтобы не писать при проектировании все алгоритмы заново, были созданы библиотеки, в которых уже собраны различные алгоритмы для реализации архитектур, методов обучения и подготовки данных для дальнейшего анализа и использования в обучении.

В настоящее время существует большое количество доступного коммерческого программного обеспечения для моделирования нейронных сетей. Так, например, в сети Internet существуют специальные www - сервера с информацией об искусственных нейронных сетях и программным обеспечением (см., например <http://www.gamma.rug.nl> или <http://www.neuronet.ph.kcl.ac.uk/neuronet/software/software.html>).

Описание программного обеспечения приведено и в различных монографиях по нейронным сетям. Кроме того, построение нейронных сетей может быть выполнено с помощью общих математических пакетов (Matlab, Mathematica, Maple).

Краткое представление об библиотеках, использующихся при проектировании нейросетей:

1. Библиотека NumPy

Библиотека NumPy была спроектирована для работы с многомерными массивами и выполняет вычислительные операции с этими массивами. Можно задать вопрос: зачем такая библиотека, если Python сам по себе и так это умеет? Ответ достаточно прост. Python – интерпретируемый язык программирования, а это значит, что он очень медленный. Библиотека NumPy написана на C – компилируемом языке, что позволяет ускорить операции над большими массивами данных в сотни, а то и в тысячи раз, по сравнению с аналогичными алгоритмом, написанным в Python.

Библиотека имеет в себе улучшенные версии встроенных – Python алгоритмов, а также немного алгоритмов для удобных математических вычислений (например, команда Transpose (или просто T), позволяющая сделать транспонирование матрицы), которых нет в стандартной библиотеке, что, в свою очередь.

На данный момент библиотека используется как замена стандартным встроенным методам Python по вышеописанным причинам, а также существует много библиотек, которые работают именно с NumPy - типами массивов (при создании массивов данных с помощью NumPy они числятся как NumPy – объекты, а не Python – объекты, но основная цель библиотеки – работа с данными, представленными в виде многомерных массивов, а также работа с этими данными и несложные вычислительные операции над ними.

2. Pandas

Pandas (сокращённо от Panel Data Sets) – библиотека, написанная на языке Python, и содержащая в себе набор инструментов и алгоритмов для анализа данных и их обработки.

Pandas также может использовать NumPy–объекты для работы, а, в свою очередь, NumPy является более низкоуровневым инструментом по отношению к Pandas.

Чаще всего в Pandas работают с данными, представленными в виде таблиц, и вся работа строится на удобном табличном представлении этих самых данных (отсюда и название). Также в библиотеке имеются инструменты для работы над статистикой данных, преобразования типов данных, составления собственных таблиц из нескольких других и набор адаптированных для Python SQL-операций (например, inner join и т.д.).

Из-за наличия инструментов, содержащихся в Pandas, Python – разработчики имеют возможность не прибегать к использованию языков программирования, ориентированных на работу с данными и содержащие инструменты для математической статистики (например, R и Octava).

3. TensorFlow

Библиотека с исходным кодом, разработанная Google, которая

содержит в себе алгоритмы машинного обучения и инструменты для использования этих алгоритмов, а также библиотека предоставляет инструменты для обучения нейросетей.

Изначально библиотека разрабатывалась командой Google Brain в 2011-м году для использования только внутри компании, но в 2015-м была переведена в открытый доступ.

В первую очередь библиотека была написана для использования в Python, но также имеет адаптации под другие языки программирования: R, C#, Haskell, Java, Swift, Go, C++.

Основное отличие TensorFlow от других библиотек машинного обучения состоит в том, что TensorFlow может работать на многих параллельных процессорах (как на центральном процессоре, так и на графическом), что позволяет делать математические вычисления в несколько раз быстрее за счёт достижения увеличения вычислительной производительности путём параллелизации вычислительных процессов.

Нейросети, написанные на TensorFlow, используются в поиске Google, а также библиотека очень хороша для работы с изображениями.

4. Scikit-learn

Ещё одна библиотека для машинного обучения, содержащая в себе инструменты для работы с задачами классификации, регрессии и кластеризации, также содержит в себе инструменты для кросс-валидации данных, работы с метриками, изображениями, графиками и текстовыми данными.

Также в scikit-learn содержатся алгоритмы из других библиотек – NumPy, Pandas, Matplotlib (библиотека для визуализации данных), SciPy (библиотека для технических вычислений).

Помимо всего вышеописанного, библиотека умеет работать с линейными и метрическими методами, графами, нелинейными методами, выполнять препроцессинг данных.

5. Keras

Библиотека, разработанная французским программистом Франсуа Шолле в качестве надстройки над TensorFlow, для использования в Python с целью работы с *сетями глубокого обучения*.

Библиотека содержит в себе инструменты для проектирования нейронных сетей и их обучения. Сами инструменты написаны настолько просто, что создать свою рабочую нейросеть можно буквально за 10 строк кода

Так как Keras является надстройкой над TensorFlow (т.е. Keras написан на той же платформе, что TensorFlow), он также имеет возможность параллелизации процессов, и, соответственно, является высокопродуктивной библиотекой в плане математических вычислений.

6. ЦИФРОВЫЕ ДВОЙНИКИ

6.1. Цифровые двойники современного предприятия

ВМ-технологии и построение цифровых двойников (ЦД) сегодня является уже неотъемлемой частью работы современного предприятия [34-37].

Несколько слов о ВМ-технологиях и их существенных отличий от технологий ЦД.

Основным направлением использования ВМ-технологий в настоящее время является проектирование и строительство промышленных и гражданских объектов. При этом, ВМ-технологий связаны главным образом с процессом проектирования и в малой степени ориентированы на отображение режимов работы моделируемого объекта в реальном масштабе времени. Более того, они практически не имеют функциональности, позволяющей выполнять глубокий анализ процессов, определяющих жизнедеятельность объекта [38].

Замечание. Концепция информационного моделирования (Building Information Modeling – BIM) является эффективным решением проблемы обеспечения и сопровождения жизненного цикла любого крупного инфраструктурного объекта, включающего в себя несколько основных этапов: планирование, проектирование, реализация и эксплуатация. Применение концепции информационного моделирования особенно характерно для строительства, и включает все этапы создания и управления трехмерным цифровым представлением физических и функциональных характеристик здания в течение всего его жизненного цикла.

Цифровой двойник (ЦД) некоторого объекта можно определить, как **цифровую (виртуальную, компьютерную) динамическую интеллектуальную модель данного объекта с большим набором информации и данных от различных датчиков, приборов и экспертов.** При этом, под термином объект понимается любой процесс, явление, система, физический, технологический или производственный объект.

Предполагается, что ЦД воспроизводит форму, характерные черты и свойства, поведение и состояние «своего» физического прототипа (объекта) по совокупности заданных параметров, свойств, критериев. Очевидно, что тем большее количество определяющих параметров и свойств объекта учитывает его ЦД, тем более последний адекватен и близок оригиналу.

Таким образом, *идеология создания ЦД требует включения в виртуальную модель помимо исходной информации, информацию и данные от различных датчиков, приборов и экспертов.*

В соответствии с основными целями и задачами, ЦД необходим для имитационного моделирования и компьютерной симуляции состояния и

поведения физического оригинала при различных условиях воздействия на него. Это помогает существенно сэкономить время и финансовые средства. Кроме того, в некоторых случаях, если речь идет, например, об опасных и вредных процессах и производствах, позволяет обеспечить безопасность людей (персонала) и избежать причинение вреда окружающей среде, экосистеме в целом.

«Философия» создания ЦД подразумевает не только разработку виртуальной цифровой модели исследуемого объекта, но и ее непрерывное взаимодействие с оригиналом в режиме реального времени. Взаимодействие между оригиналом объекта и его ЦД обеспечивается разработкой специализированной системы «прямого и обратного отклика» виртуальной модели и ее прототипа (например, посредством автоматизированной системы мониторинга состояния физического объекта). Структура, особенности и специфика прототипов учитываются построением цифровых базовых моделей на основе САД- и САЕ-технологий [39]. САД-системы позволяют получать точную копию топологии объекта, а использование САЕ-технологий дает возможность проведения компьютерной симуляции и расчетов физических и технологических процессов, составляющих сущность объекта.

Итак, ЦД нельзя рассматривать только как цифровую (хотя и очень подробную) модель реального физического объекта. Важная характеристика ЦД – его взаимодействие и взаимосвязь со своим оригиналом в масштабе реального масштаба времени на основе данных и информации, которыми они постоянно обмениваются и на основе такого обмена определенная корректировка и совершенствование цифровой модели. Например, от ЦД к оригиналу поступают результаты моделирования (расчетов), а от оригинала цифровому двойнику – данные от датчиков о состоянии объекта. Этим свойством ЦД кардинально отличается, например, от цифровых копий, проектов, моделей объектов.

Таким образом, принцип работы ЦД заключается в постоянном взаимодействии посредством системы датчиков и специализированного аппаратного обеспечения с реальным объектом. ЦД, помимо цифровой модели объекта, содержит систему обработки и отображения информации, поступающей от мониторинга текущего состояния объекта, Базы данных.

Поскольку виртуальная копия обладает той же информацией, что и ее прототип, регламент работ по техническому обследованию и наблюдениям за состоянием отдельных элементов физического объекта существенным образом меняется. Так, в случае применения ЦД контролировать работу оборудования можно удаленно. Комбинируя данные из различных источников информации, ЦД может в режиме реального времени анализировать и прогнозировать состояние физического прототипа.

Сравнивая прогнозируемые и фактические данные обратной связи, ЦД в состоянии прогнозировать реакции системы на критически важные для безопасного функционирования объекта события, а также для выявления возможных проблем на ранней стадии, когда они еще не достигли критического состояния.

Важными составляющими компонентами ЦД являются **математические и компьютерные имитационные модели**, позволяющие моделировать различные стороны жизнедеятельности физического объекта. Знания, полученные в условиях эксплуатации, используются для изменения параметров этих моделей (для верификации, «калибровки», уточнения параметров и характеристик).

В концепцию ЦД входит также наличие целого ряда *автоматизированных систем для мониторинга состояния физического объекта*. Данные от таких систем от физического прототипа (близнеца) передаются ЦД.

Замечание. Впервые строго концепцию цифрового двойника описал, наверное, в 2002 году профессор Мичиганского университета Майкл Гривс в монографии «Происхождение цифровых двойников» [40]. По мнению Майкла Гривса в идеале необходимо стремиться к тому, чтобы « всю информацию об изделии можно было получить от его цифрового двойника ».

Повышенный интерес к технологиям цифровых двойников и мощный толчок к выполнению исследований в этом направлении в настоящее время обусловлены развитием искусственного интеллекта (ИИ) и интернета вещей. Согласно исследованию Gartner Hype Cycle, описывающему циклы зрелости технологий, это произошло в 2015 году [41]. Технологии цифровых двойников становятся все более совершенными за счёт интеграции с технологиями ИИ, разработками в области виртуальной и дополненной реальности, развитием 3D- и 4D-печати, технологий компьютерного зрения.

Следует подчеркнуть еще раз, что ЦД *не являются статичными, консервативными*, а представляют из себя целую систему сопряженных (связанных между собой) динамических цифровых двойников, моделирующих рассматриваемый объект на различных стадиях его жизнедеятельности (рождение (проектирование), рост (строительство), активный жизненный цикл (эксплуатация) и т.д.).

Итак, для создания ЦД необходимо наличие трех важных компонент:

- Средства комплексного моделирования и компьютерного симулирования;

- Средства сбора данных, объединения их, обработки, преобразования в необходимую информацию для передачи цифровой модели; средства коммуникации между различными частями ЦД;
- Средства машинного обучения.

Как показывают результаты исследований, применение технологии цифровых двойников является перспективным направлением, позволяющим минимизировать человеческий фактор, при этом максимизировав экономические показатели предприятия за счет оптимизации технологических процессов.

Так, например, создание детализированных цифровых двойников ответственных масштабных элементов горнодобывающего предприятия является основой концепции «безлюдного рудника», позволяющей значительно повысить автономность работы производственных объектов. *Применительно к горнодобывающему предприятию ЦД:*

- предоставляет возможность осуществлять в реальном масштабе времени мониторинг состояния объекта, своевременный анализ данных и выдачу рекомендаций для предотвращения проблем до их возникновения;
- позволяет планировать профилактический ремонт с целью сокращения и предотвращения простоев;
- планировать развитие горных работ на перспективу и т.д.

Очевидно, что цифровые двойники представляют собой необходимый компонент систем регионального сопряженного геомониторинга для регионов крупномасштабного ведения горных работ [42, 43].

Одним из базовых элементов ЦД рудника (шахты) является *горно-геологическая информационная система (ГГИС)*, поставляющая основной объем исходной и текущей горно-геологической и горнотехнической информации [42, 43].

Для создания ЦД сложного реального объекта необходимо построить ЦД отдельных структурных элементов объекта, которых может быть не только сотни, но и тысячи. Поэтому можно выделить ***различные уровни функционирования цифровых двойников.***

1. ЦД отдельного элемента (компонента).

Так, если успешная работа некоторого механизма (системы) определяющим образом зависит от функционального состояния конкретных деталей (модулей), то необходимо создать ЦД этих деталей (модулей), позаботившись, конечно, об организации «обратной связи» от них их виртуальным копиям.

Например, это могут быть ролики в системе конвейеров в шахте, подшипники в вентиляторах местного проветривания в вентиляционной системе рудника, трос в системе подъемного комплекса.

2. ЦД отдельных модулей структурных элементов объекта, что обеспечивает контроль за состоянием конкретного оборудования, элемента объекта.

Например, в таком структурном элементе рудника, как «система вентиляции и проветривания подземного пространства» в качестве отдельных модулей можно выделить вентиляторы, двигатели. В этом случае, при необходимости, «физические модули» могут обмениваться информацией со своими виртуальными копиями.

3. ЦД структурных элементов объекта, что позволяет контролировать состояние и работу сложных элементов, состоящих из большого количества модулей, обеспечивающих выполнение определенной функциональности.

Применительно к объекту «рудник» структурным элементом может быть горно-добычной комплекс, подъемный комплекс и т.д.

4. ЦД всего объекта в целом, позволяющий получать информацию о текущем состоянии объекта и прогнозировать ситуацию в дальнейшем. Работа ЦД данного уровня зависит от согласованной совместной работы ЦД всех предыдущих уровней.

Так, ЦД рудника представляет собой сопряженный виртуальный двойник большого количества цифровых двойников всех внутренних компонент рудника.

В связи с огромным разнообразием объектов для которых создается ЦД, существует большое множество типов и видов ЦД. **По целевому назначению применительно к такому объекту, как современное производственное предприятие можно выделить следующие типы ЦД.**

- CAD (или BIM) ЦД (CAD-DT) – это сопряженный общий CAD-проект реального объекта (предприятия), который содержит все компоненты и системы его жизнеобеспечения. Например, применительно к такому объекту, как рудник, это планы горных работ, планы транспортной системы, системы электроснабжения, подъемный комплекс и т.д.).

Замечание. Данный тип можно отнести по общей классификации к «прототипу ЦД» (DTP) – виртуальный аналог реального объекта, который содержит все исходные данные оригинала.

- CAE&Simulation ЦД (CAE&S-DT) – виртуальная сопряженная модель объекта, позволяющая выполнять математическое моделирование и компьютерное моделирование (симуляцию) физических и технологических процессов, сопровождающих весь комплекс производственных и технологических процессов (например, добычи полезного ископаемого в условиях конкретного производственного цикла рудника).

Замечание. Этот тип ЦД можно сопоставить, согласно общей классификации, с типом «экземпляр ЦД» (DTI) – содержит данные обо всех

характеристиках и процессах, сопровождающих эксплуатацию реального физического объекта. Данный ЦД действует параллельно с оригиналом.

- **САМ ЦД (САМ-DT)** – интеллектуальная автоматизированная система целого набора ЦД отдельных структурных элементов объекта (предприятия), системы мониторинга за состоянием структурных элементов объекта (датчики, наблюдения и т.д.), АРМов специалистов и служб для управления жизнедеятельностью предприятия в целом.

Замечание. В соответствии с общей классификацией данный тип относится к типу «агрегированный ЦД» (DTA) – автоматизированная система из цифровых двойников и реальных объектов, которыми можно управлять из единого центра и обмениваться данными внутри системы.

Как видно из введенной типизации цифровых двойников, наименование типов «созвучно» с имеющими широкое распространение аббревиатурами CAD, CAE, CAM. Но ЦД имеет существенные отличия от традиционных CAD-моделей или расчетных CAE-моделей [44].

Отметим, что сегодня зачастую любую цифровую (компьютерную) версию реального объекта называют ЦД. В принципе, такой подход позволителен, но тогда необходимо ввести **классификацию уровней виртуальной репрезентации физического объекта**. Очевидно, что классификация определяется базовыми критериями, на основе которой она выполняется. Отметим, что выделенная выше типизация ЦД может рассматриваться как определенная классификация. Распространенной на сегодня является **следующая классификация уровней цифровых двойников** [44]:

- **Доцифровой двойник (уровень 1)** – традиционный (классический) виртуальный прототип.

- **Цифровой двойник (уровень 2)** – ЦД, в котором виртуальная модель физического объекта способна объединять множество данных различной природы о «физическом близнеце». При этом, возможен сбор данных от физических датчиков и специализированной аппаратуры.

- **Адаптивный Цифровой двойник (уровень 3)** – ЦД с наличием адаптивного пользовательского интерфейса (элементы Искусственного Интеллекта) как для физических, так и виртуальных двойников. ЦД включают алгоритмы машинного обучения на основе технологии нейронных сетей. Модели, используемые в таком Цифровом двойнике, постоянно обновляются на основе данных, получаемых от физического близнеца в режиме реального времени.

- **Умный Цифровой двойник (уровень 4)** – ЦД, обладающий всеми возможностями ЦД 3-го уровня, но помимо этого у него имеются функции «машинного неконтролируемого обучения». На этом уровне ЦД обладает высокой степенью автономии.

Очевидно, что затраты на создание ЦД достаточно значительны, но перспективы их использования на современном предприятии окупают все вложенные средства. Так, *ЦД рудника (шахты) позволяет:*

- Выполнить компьютерную симуляцию новых технических решений для отдельных функциональных частей (структурных элементов) рудника и оценить эффективность предлагаемого решения на базе системы различных критериев еще до стадии непосредственного внедрения в производственный цикл.

Например, это может быть оценка проекта на отработку нового участка шахтного поля, исследование эффективности решений по охране выработок, модификации в системе вентиляции, конвейерной системы или системы электроснабжения и т.п.

- Спрогнозировать возможные проблемы или уязвимость отдельных структурных элементов и предложить способы решения потенциальных критических ситуаций.

- На основе моделирования, анализа и обобщения данных о реальном состоянии структурных элементов выдать предложения по повышению эффективности работы процессов и состоянию различных подсистем и элементов.

Например, если рассмотреть такой ответственный элемент как геотехническую систему «подъемный комплекс – конструкционные элементы шахтного ствола – околоствольный массив», то создание ЦД данной геотехнической системы позволит эффективно отслеживать состояние всех элементов системы и выдавать прогноз о возможных нарушениях в их функционировании, и на основе информации от датчиков и специализированной аппаратуры оптимизировать режимы работы подъемных сосудов [45].

- Выполнять оценку и планирование жизнедеятельности объекта (рудника) на долгосрочную перспективу.

Важной составляющей ЦД рудника является *модуль «Имитационное компьютерное моделирование»*.

Для того чтобы предсказать, как физический прототип будет работать в реальных условиях при внедрении предлагаемых технических решений, весьма эффективно использовать технологии имитационного моделирования на базе ЦД. При этом, для одной и той же компоненты физического объекта может быть построено несколько моделей, в зависимости от назначения, для чего она используется. Так, например, для массива горных пород в пределах горного отвода строится несколько цифровых моделей (от геологической структурной модели до геомеханической модели с подземными сооружениями). Для рудника в целом строятся различные «САПР-модели» (моделирование электросетей,

системы вентиляции, горных работ и др.), «экономико-математические модели» и др.

Такой подход намного перспективнее традиционного процесса имитационного моделирования, так как позволяет рассмотреть более широкий диапазон возможных условий функционирования исходного объекта (предприятия). ЦД дополняется данными от «физического двойника», поступающими в базы данных, которые, в свою очередь, используются при выполнении моделирования. ЦД объекта отслеживает критические ситуации и отказы различных систем физического объекта и на основе имитационного моделирования предлагает критерии предельных состояний, прочности и долговечности, оценки срока службы (в широком понятии этого термина) различных систем. Таким образом, имеется возможность выполнять «обслуживание и оценку надежности и срока службы различных систем и компонент физического объекта» по анализу текущего состояния систем и компонент. Эти знания можно использовать, чтобы заблаговременно запланировать и выполнить профилактический ремонт или замену компонент объекта до того, как они «выйдут из строя».

6.2. Этапы создания ЦД предприятия

Процедура создания ЦД включает в себя большое количество различных этапов и процедур. Их количество, состав и наполнение существенным образом *определяется исходным объектом (прототипом)*, для которого создается ЦД. Вместе с тем, имеется набор некоторых базовых этапов. Рассмотрим их на примере такого сложного объекта, как рудник (шахта) по добыче полезного ископаемого подземным способом.

■ Этап «Исследование объекта»

Этот этап предшествует разработке ЦД в том случае, если у ЦД есть реальный прототип. Применительно к руднику такой этап обязателен.

На данном этапе составляется детальная карта прототипа, описываются все процессы и характеристики объекта. При этом важно исследовать объект при различных условиях функционирования.

■ Этап «Построение цифровой модели объекта»

В случае, если реального прототипа нет, то этот этап может быть первым. Например, при проектировании рудника в новых горно-геологических условиях и при использовании пионерских технологических решений, создание цифровой модели рудника и ЦД предшествует этапу строительства рудника.

При построении цифровой модели используется большой набор методов прикладной и вычислительной математики, информатики.

■ Этап «Построение ЦД объекта»

Этап включает разработку архитектуры ЦД и перенос этой архитектуры на специальные платформы (например, фирмы Siemens, Dassault Systemes, ...). Специализированное программное обеспечение имеют в своем составе средства, позволяющие объединить математические модели, наборы данных и интерфейс для управления всеми компонентами ЦД.

■ Этап *«Тестирование основных процессов функционирования (работы) прототипа на ЦД»*

Основная цель этого этапа – тестирование работы ЦД при «стандартных» режимах работы прототипа и моделирование работы прототипа в случае «внештатных ситуаций».

Результаты деятельности ЦД на данном этапе тщательно изучаются экспертами и специалистами для внесения корректировок в архитектуру и алгоритмы работы элементов ЦД.

■ Этап *«Наладка и запуск ЦД»*

После получения заключения о том, что тестирование ЦД на предыдущем этапе выполнено в полном объеме и удовлетворяет требованиям специалистов и экспертов, переходят к этапу испытаний ЦД в реальных условиях (опытно-промышленные испытания). Каков бы не был объем тестовых испытаний на этапе «Тестирование основных процессов функционирования (работы) прототипа на ЦД», невозможно обеспечить учет всех возможных критических ситуаций.

Этап наладки и запуска в реальных условиях функционирования прототипа (рудника) требует выполнения большого объема работ, в том числе и по внесению корректировок в ЦД.

■ Этап *«Корректировка и развитие ЦД и его физического прототипа в ходе реальной эксплуатации ЦД»*

Этап подразумевает работу цифрового двойника в совокупности с его реальным физическим объектом до тех пор, пока не будут отлажены все системы и процессы.

По результатам этой работы могут вноситься изменения и собственно в оригинальный физический объект для достижения максимальной эффективности жизнедеятельности.

6.3. От пилотного проекта до масштабирования технологии ЦД

Оптимальным представляется начинать разработку и внедрение технологии Цифровых двойников на предприятии с «пилотного проекта» по разработке ЦД конкретного оборудования или технологического процесса. После успешного внедрения данного проекта в производственный процесс можно масштабировать такой подход на другие составные элементы и системы предприятия.

Процесс создания и внедрения технологии ЦД может быть таков:

1. Выполняется анализ, в каких подразделениях/службах и для каких процессов представляется наиболее «удобным» разработать и внедрить первый ЦД. Рекомендуется выбрать процесс/оборудование, являющееся одним из наиболее важных элементов общего технологического процесса жизнедеятельности предприятия. В результате, благодаря внедрению ЦД, появляется возможность «увидеть и оценить» эффективность такого подхода на работу предприятия и выдать предложения на дальнейшее развитие и внедрение ЦД на предприятии.

2. После выбора объекта для создания ЦД необходимо определиться с «генеральным разработчиком» выполнения пилотного проекта. Это важный шаг, так как «неудачный» выбор разработчика может поставить под сомнение успешную реализацию проекта в целом.

3. Следующим шагом является разработка подробных технического задания и проекта на выполнение и собственно реализация пилотного проекта. При этом необходимо «помнить» о том, что выработанное решение в дальнейшем требует масштабирования и функционирования в реальных производственных условиях без «сильной зависимости» от внешних исполнителей.

4. После успешного внедрения пилотного проекта требуется разработка плана по дальнейшему масштабированию технологии ЦД на другие объекты и процессы.

5. Этап масштабирования решений и разработки ЦД других объектов и процессов. Следует учесть, что ЦД отдельных процессов и объектов должны функционировать в интеграции друг с другом. *Это является принципиальным моментом* и требует принятия определенных решений, требований и ограничений «на входе». То есть необходим в обязательном порядке «авторский надзор» над разработками Цифровых двойников для создания в итоге *адаптивного, а в последующем и умного ЦД предприятия.*

7. ЭКСПЕРТНЫЕ СИСТЕМЫ. ИНЖЕНЕРИЯ ЗНАНИЙ

7.1. Общие понятия и определения

Важным элементом систем ИИ являются *Базы Знаний и Данных* в совокупности с *системами приобретения знаний и получения решений*.

Почему знаниям при построении интеллектуальных систем необходимо уделять повышенное внимание? Во-первых, современные методы символьных рассуждений все еще не обеспечивают четкого логически непротиворечивого представления знаний, описания задач на различных уровнях абстрагирования, распределения средств решения задач, управления кооперирующими процессами, объединения в ходе вывода различных источников знаний. Вторая причина преимущественного внимания к знаниям, а не к формальным методам рассуждения состоит в том, что люди-специалисты достигают исключительно высоких результатов вследствие своих познаний. Если эти знания аккумулировать в машинных программах и применять их, то такие программы тоже достигнут высокого уровня работы. Третья причина преимущественного внимания к знаниям связана с признанием их собственной ценности. Традиционный способ передачи знаний от специалиста к новичку связан с затратой многих лет на обучение и практическую стажировку. Извлечение знаний у специалистов и придание им формы, позволяющей использовать знания в вычислительных машинах, существенно удешевляет и их воспроизводство, и применение. В то же время процесс уточнения знаний можно ускорить, сделав личные знания доступными публичной проверке и оценке.

Система приобретения знаний строится на основе обработки данных, полученных:

- в ходе диалога с экспертом (в общем смысле этого слова);
- в процессе накопления практического опыта эксплуатации систем;
- в результате анализа новых достижений в области фундаментальных и прикладных задач рассматриваемой проблемной области и связанных с ней смежных направлений.

Система получения решения выполняет такие функции, как просмотр содержимого Баз Знаний, добавление в Базу недостаточных фактов, применение правил логического вывода для выработки решения об оценке ситуации или о выборе очередного шага в процедуре оценивания.

Знания являются ключевым фактором при решении сложных задач. Поэтому они оправдывают большие затраты, связанные с их добычей и требуют хорошо отработанной и эффективной технологии для придания им «товарного вида».

Знания обычно существуют в двух видах [46]: **общедоступные и индивидуальные знания.**

Общедоступные знания – это факты, определения и теории, которые изложены в общих фундаментальных источниках по данной области (учебники, справочники и т.д.).

Помимо общедоступных сведений специалисты должны обладать еще и *индивидуальными знаниями*, отсутствующими в общих литературных источниках. Эти индивидуальные знания в значительной степени состоят из эмпирических правил, которые называются эвристиками. Замечательным свойством эвристик является возможность специалистам данной предметной области (экспертам) на их основе при необходимости выдвигать разумные предположения, находить правильные и перспективные подходы к решению задач и эффективно работать в случае неполных или «зашумленных» исходных (первичных) данных.

Поэтому одной из центральной задач при построении и конструировании Баз Данных и Знаний, а на их основе и **экспертных систем**, является подбор специалистов, обладающими уникальными индивидуальными знаниями и с помощью этих экспертов выявление и воспроизведение таких знаний.

Главные задачи при построении подсистемы «Базы Знаний и Данных» состоят в выявлении и четкой формулировке специальных знаний, а также правил внесения и собственно внесение этих знаний в вычислительную машину.

Замечание. Необходимо различать **знание и умение**. Умелое выполнение или решение некоторой задачи характеризуется такими чертами, как большая скорость (или какой-либо иной показатель эффективности) исполнения, малое количество ошибок, оптимальная умственная напряженность, большая приспособляемость и робастность (устойчивость к сбоям и малым отклонениям в начальных условиях). То есть в этих чертах проявляются как сами знания, так и техника их использования и владения ими. Умение (или мастерство) можно определить, как обладание необходимыми знаниями и умение эффективно ими воспользоваться.

7.2. Экспертные системы

Базы знаний являются основной частью экспертных систем. Применительно к экспертной системе в определенной предметной области базы знаний представляют собой совокупность данных, фактов и правил логического вывода, позволяющих формулировать корректные, «правильные» заключения, выводы, предложения и рекомендации. Базы знаний в совокупности с *набором правил* позволяют на основании предоставляемых пользователем фактов распознавать ситуации, формулировать решения и/или выдавать рекомендации, предложения для

выбора действия.

Рассмотрим более подробно, что собой представляют экспертные системы.

Распространенным «классическим» определением экспертной системы (ЭС, expert system) является такое [47]: «Экспертная система – система, предназначенная для решения трудно формализуемых задач, у которых отсутствует алгоритм решения, алгоритм решения которых не известен, или обладает достаточно большой размерностью».

Современные экспертные системы начали разрабатываться исследователями искусственного интеллекта в 1970-х годах. Предшественники экспертных систем были предложены еще в 1832 году С. Н. Корсаковым, создавшим механические устройства, так называемые «интеллектуальные машины», позволявшие находить решения по заданным условиям, например, определять наиболее подходящие лекарства по наблюдаемым у пациента симптомам заболевания.

Главным достоинством экспертных систем является возможность накопления знаний, их сохранение и обработка. В отличие от субъективной оценки человека, экспертные системы обрабатывают информацию объективно (с точки зрения установленных правил), что улучшает качество проводимой экспертизы. При этом, при решении задач, требующих обработки большого объема знаний, возможность возникновения ошибки при переборе очень мала.

ЭС представляют собой одно из ведущих направлений в области искусственного интеллекта. При их разработке используются такие методы ИИ как: методы представления знаний, логического вывода, эвристического поиска, распознавания предложений на естественном языке и др. Можно утверждать, что именно ЭС позволили получить очень большой коммерческий эффект от применения таких мощных методов. На сегодня, наверное, трудно найти проблемную область, в которой не создано ни одной ЭС или такие попытки не предпринимались бы.

ЭС представляют собой интеллектуальную компьютерную систему, способную частично заменить специалиста-эксперта в разрешении проблемной ситуации. Так как ЭС, как и эксперт-человек, в процессе своей работы оперирует со знаниями, поэтому знания о предметной области, необходимые для работы ЭС, должны быть определенным образом формализованы и представлены в виде баз знаний, которые могут изменяться и дополняться в процессе развития системы.

ЭС ориентированы на решение задач, обычно требующих проведения экспертизы человеком-специалистом. В отличие от «классических» машинных программ, использующих процедурный анализ, ЭС решают задачи в узкой предметной области (конкретной области экспертизы) на

основе дедуктивных рассуждений. Такие системы часто оказываются способными найти решение задач, которые плохо структурированы и плохо определены. С отсутствием структурированности ЭС «справляются» путем привлечения *эвристик, т.е. правил, взятых «с потолка»*, что может быть полезным в тех системах, когда недостаток необходимых знаний или времени исключает возможность проведения полного анализа.

Одно из важных достоинств ЭС не только возможность накапливать и сохранять знания, но и обрабатывать их, согласно определенным правилам, а также обновлять знания, что позволяет «новичкам» и специалистам с недостаточной квалификацией, повышать квалификацию, принимать обоснованные и корректные решения и выводы.

Для разработки высокоэффективной ЭС необходимо наличие следующих элементов, «работающих» в тесном взаимодействии друг с другом:

■ *Пользователь* – специалист предметной области, для решения задач которой ЭС и предназначена. Обычно ориентируются на то, что квалификация «усредненного» пользователя недостаточно высока, поэтому он нуждается в помощи и поддержке.

■ *Эксперт* – специалист предметной области, имеющий высокую квалификацию, обладающий достаточным опытом и знаниями для выдачи рекомендаций по наполнению баз знаний, построению правил принятия решений и т.п.

■ *Аналитик* (инженер по знаниям, когнитолог, инженер-интерпретатор) – специалист в области искусственного интеллекта, играющий роль посредника между экспертом и базами знаний.

■ *Базы знаний* – ядро экспертной системы. Совокупность знаний предметной области, представленная в форме, понятной эксперту и пользователю.

■ *«Решатель»* – программа, моделирующая ход рассуждений эксперта на основании знаний, имеющихся в базе знаний.

■ *Подсистема объяснений* – программа, позволяющая пользователю получить ответы на вопросы, как была получена та или иная рекомендация, почему машина приняла именно такое решение. Так, например, ответ на вопрос «как?» содержит трассировку всего процесса получения решения с указанием использованных фрагментов базы знаний. Ответ на вопрос «почему?» представляет собой ссылку на умозаключение, предшествовавшее полученному решению

■ *Интеллектуальный редактор* – программа, позволяющая аналитику создавать базы знаний в диалоговом режиме.

Экспертные системы имеют две категории пользователей и два отдельных «входа», соответствующих различным целям взаимодействия

пользователей с ЭС:

1. Обычный пользователь (эксперт), которому требуется консультация ЭС (диалоговый сеанс работы с ней), в процессе которой система решает некоторую экспертную задачу. Диалог с ЭС осуществляется через диалоговый процессор – специальную компоненту ЭС. Существуют две основные формы диалога с ЭС: диалог на ограниченном подмножестве естественного языка (с использованием словаря-меню, при котором на каждом шаге диалога система предлагает выбор профессионального лексикона экспертов) и диалог на основе из нескольких возможных действий;

2. Экспертная группа инженерии знаний, состоящая из экспертов в предметной области и инженеров знаний. В функции этой группы входит заполнение баз знаний, осуществляемое с помощью специализированной диалоговой компоненты ЭС – подсистемы приобретения знаний, которая позволяет частично автоматизировать этот процесс.

Хотя системы, основанные на знаниях, при своем создании требуют в обязательном порядке наличие экспертов, в дальнейшем при использовании они имеют определенные преимущества перед человеком-экспертом. Так, например:

1. У них нет предубеждений.

2. Они не делают поспешных выводов.

3. Эти системы работают систематизировано, рассматривая все детали, часто выбирая наилучшую альтернативу из всех возможных.

4. База знаний может быть очень и очень большой. Будучи введены в машину один раз, знания сохраняются навсегда. Человек же имеет ограниченную базу знаний, и если данные долгое время не используются, то они забываются и навсегда теряются.

5. Системы, основанные на знаниях, устойчивы к «помехам». Эксперт пользуется побочными знаниями и легко поддается влиянию внешних факторов, которые непосредственно не связаны с решаемой задачей. ЭС, не обремененные знаниями из других областей, по своей природе менее подвержены «шумам». Со временем системы, основанные на знаниях, могут рассматриваться пользователями как разновидность тиражирования – новый способ записи и распространения знаний. Подобно другим видам компьютерных программ они не могут заменить человека в решении задач, а скорее напоминают орудия труда, которые дают ему возможность решать задачи быстрее и эффективнее.

6. Эти системы не заменяют специалиста, а являются инструментом в его руках.

В настоящее время «классическая» концепция экспертных систем, которая была принята в 1970–1990 годах [46–48], претерпевает

существенные изменения.

К ЭС *первого поколения* относятся системы, которые могут лишь повторить логический вывод эксперта. Очевидно, что специалисту, решающему интеллектуально сложную задачу, явно недостаточно возможностей системы, которая лишь имитирует деятельность человека. Ему необходимо, чтобы ЭС выступала в роли «полноценного помощника и советчика», способного проводить анализ нечисловых данных, выдвигать и отбраковывать гипотезы, оценивать достоверность фактов, самостоятельно пополнять свои знания, контролировать их непротиворечивость, делать заключения на основе прецедентов и, может быть, даже порождать решение новых, ранее не рассматривавшихся задач.

Наличие таких возможностей является характерным для ЭС *второго поколения*, концепция которых начала активно разрабатываться порядка 10 лет назад. Экспертные системы, относящиеся ко второму поколению, называют *партнерскими*, или усилителями интеллектуальных способностей человека. Их общими отличительными чертами является умение обучаться и развиваться, т.е. эволюционировать.

Экспертные системы первого и второго поколений отличаются методами и формами представления, обработки и использования знаний. В экспертных системах первого поколения:

- 1) знаниями системы являются только знания эксперта, опыт накопления знаний не предусматривается;
- 2) методы представления знаний позволяли описывать лишь статические предметные области;
- 3) модели представления знаний ориентированы на простые области.

В экспертных системах второго поколения:

- 1) используются не поверхностные знания, а более глубокие; возможно дополнение предметной области;
- 2) ЭС может решать задачи динамической базы данных предметной области.

По типу решаемых задач экспертные системы можно классифицировать следующим образом (таблица 7.1).

Таблица 7.1. Типы решаемых задач ЭС

Экспертные системы	
Тип	Адресуемые задачи
Интерпретация	Построение описаний ситуаций по наблюдаемым данным
Диагностика	Заключение о нарушениях в системе, исходя из наблюдений
Мониторинг	Сравнение наблюдений с критическими точками плана
Проектирование	Построение конфигурации объектов при ограничениях
Прогнозирование	Вывод вероятных следствий из заданных ситуаций

Планирование	Проектирование плана действий
Оптимизация	Выбор оптимального плана действия, мероприятий
Обучение	Диагностика, отлаживание и исправление поведения обучающегося
Контроль и Управление	Интерпретация, прогноз и мониторинг поведения системы
Ремонт	Выполнение плана применения выработанной рекомендации
Отладка	Выработка рекомендаций по устранению неисправностей
.....

✓ Интерпретация данных

Интерпретирующие системы обладают способностью получать определенные заключения на основе результатов наблюдения.

✓ Диагностирование.

В этом направлении наиболее развиты ЭС в области медицины. Диагностические системы используются для установления связи между нарушениями деятельности организма и их возможными причинами.

✓ Мониторинг, диагностика работы и функционирования ответственных систем, устройств.

В этой сфере системы, основанные на знаниях, незаменимы как, например, при диагностике работы и ремонте механических, электрических и т.п. систем и машин (автомобилей, дизельных локомотивов и т.д.), так и при устранении неисправностей и ошибок в аппаратном и программном обеспечении специализированных систем.

✓ Проектирование.

✓ Прогнозирование.

Прогнозирующие системы предсказывают возможные результаты или события на основе данных о текущем состоянии объекта.

✓ Планирование.

Планирующие системы предназначены для достижения конкретных целей при решении задач с большим числом переменных.

✓ Оптимизация.

✓ Обучение.

Экспертные системы могут являться составной частью в компьютерных системах обучения. Примером обучения может служить компьютерная игра, сложность которой увеличивается по мере возрастания степени квалификации играющего.

✓ Контроль и Управление

Системы, основанные на знаниях, могут применяться в качестве интеллектуальных систем контроля и принимать решения, анализируя данные, поступающие от нескольких источников. Такие системы работают на атомных электростанциях, управляют воздушным движением и осуществляют медицинский контроль. Они могут быть также полезны при

регулировании финансовой деятельности предприятия и оказывать помощь при выработке решений в критических ситуациях

✓ Ремонт

✓ Отладка и др.

Большинство ЭС включают знания, по содержанию которых их можно отнести одновременно к нескольким типам. Например, обучающая система может также обладать знаниями, позволяющими выполнять диагностику и планирование. Она определяет способности обучаемого по основным направлениям курса, а затем с учетом полученных данных составляет учебный план. Управляющая система может применяться для целей контроля, диагностики, прогнозирования и планирования. Так, система, обеспечивающая сохранность жилища, может следить за окружающей обстановкой, распознавать происходящие события (например, открылось окно), выдавать прогноз (вор-взломщик намеревается проникнуть в дом) и составлять план действий (вызвать полицию).

Пример. С учетом таблицы 7.1 характерные черты системы регионального геомеханического мониторинга (СРГМ) как системы с элементами ИИ [42] можно описать следующим образом:

Интерпретирующая система выводит описание ситуаций из наблюдаемых данных и анализа исходной информации. То есть СРГМ должна объяснять исходные наблюдаемые данные, описывающие ситуацию или состояние системы, обуславливающее эти данные.

Прогнозирующая система выводит вероятные следствия из заданных ситуаций, что составляет предсказание, прогноз и оценку по заданным начальным условиям в соответствии с заданными алгоритмами и методиками расчетов.

В СРГМ наиболее логично использовать параметрическую динамическую модель, в которой значения параметров «подгоняются» под конкретную ситуацию и условия. Выводимые из этой модели следствия составляют основу для прогнозов. Отметим, что необходимо делать вероятностные оценки, так как в противном случае прогнозирующие системы могут породить большое число возможных сценариев будущих событий.

Диагностическая система позволяет судить о нарушениях в работе системы в целом и ее отдельных составных частях по наблюдениям. Главным образом диагностика в СРГМ касается соотношения наблюдаемого поведения изучаемого объекта с причинами, обусловившими это поведение.

Система проектирования строит динамические конфигурации объектов с учетом ограничений и правил поведения, присущих рассматриваемой задаче проектирования.

Система планирования проектирует планы действий, что включает в себя задачи планирование при проектировании. Причем это как задачи, например, планирования коммуникаций и инженерных сооружений, так и планирования профильных линий, экспериментов и т.п.

Предупредительная система (система мониторинга) сопоставляет результаты наблюдений за поведением системы с характеристиками, являющимися эталонными в данных условиях, и определяет опасные места или критические свойства.

Система отладки дает рекомендации относительно ликвидации или смягчения вредных (опасных) последствий, состояния или функционирования. Для исправления обнаруженных затруднений используются средства отладки, планирования и выполнения.

Система обучения демонстрирует и отлаживает поведение и решения «студента» (под термином «студент» понимается специалист, не обладающий достаточным запасом знаний, опыта или не имеющий права принимать самостоятельно ответственные решения).

Управляющая система обеспечивает адаптивное управление всем поведением системы. Чтобы выполнить это, СРГМ как управляющая система должна постоянно интерпретировать текущую ситуацию, прогнозировать будущее, диагностировать причины возникающих проблем, формулировать план их ликвидации и контролировать его выполнение, обеспечивая его успех.

Качество ЭС определяется размером и качеством базы знаний (правил или эвристик). Общая схема функционирования ЭС состоит в циклическом выполнении следующих режимов: выбор (запрос) данных или результатов анализа данных; наблюдения, интерпретация результатов, усвоение новой информации; выдвижение с помощью правил временных гипотез и затем выбор следующей порции данных или результатов анализа данных. Такой процесс продолжается до тех пор, пока не будет выдано заключение, которое можно рассматривать как окончательное.

По отношению к такому параметру, как *«связь с реальным временным интервалом»*, *экспертные системы классифицируются следующим образом:*

- Статические ЭС – решающие задачи в условиях, не изменяющихся во времени исходных данных и знаний.
- Квазидинамические ЭС – интерпретируют ситуацию, которая меняется с некоторым фиксированным интервалом времени.
- Динамические – решающие задачи в условиях, изменяющихся во времени исходных данных и знаний.

Независимо от такого типа классификации ЭС, в любой ЭС присутствуют следующие три типа знаний:

✓ *Структурированные статические знания*

Статические знания о предметной области. После того как эти знания выявлены, они уже не изменяются.

✓ *Структурированные динамические знания*

Изменяемые знания о предметной области. Они обновляются по мере выявления новой информации.

✓ *Рабочие знания*

Знания, применяемые для решения конкретной задачи или проведения консультации.

В ЭС необходимо предусмотреть *два режима функционирования*:

Режим ввода знаний – в этом режиме эксперт с помощью инженера по знаниям посредством редактора базы знаний вводит известные ему сведения о предметной области в базу знаний ЭС.

Режим консультации – пользователь ведёт диалог с ЭС, сообщая ей сведения о текущей задаче и получая рекомендации ЭС. Например, на основе сведений о физическом состоянии больного ЭС ставит диагноз в виде перечня заболеваний, наиболее вероятных при данных симптомах.

Нередко в качестве маркетингового хода экспертными системами объявляются современные программные продукты, в «классическом» понимании таковыми не являющиеся (например, компьютерные справочно-правовые системы). Предпринимаемые попытки объединить «классические» подходы к разработке экспертных систем с современными подходами к построению пользовательского интерфейса (например, проекты CLIPS Java Native Interface, CLIPS.NET и др.) не находят пока поддержки среди крупных компаний-производителей программного обеспечения и по этой причине остаются пока в экспериментальной стадии.

К программам, подобным экспертным системам, можно в какой-то мере отнести и поисковые или справочные (энциклопедические) системы. По запросу пользователя они предоставляют наиболее подходящие (релевантные) разделы базы статей (представления об объектах областей знаний, их виртуальную модель).

Все же, основными отличиями ЭС от других программных продуктов являются использование данных и знаний, специальных алгоритмов принятия решений, а также формулировки новых знаний на основе имеющихся. Важно, что в ЭС известен алгоритм обработки знаний, а не алгоритм решения задачи. Поэтому применение алгоритма обработки знаний может привести к получению такого результата при решении конкретной задачи, который не был предусмотрен. Более того, алгоритм обработки знаний заранее неизвестен и строится по ходу решения задачи на основании эвристических правил.

Критерий использования ЭС для решения задач

Существует ряд прикладных задач, которые решаются с помощью систем, основанных на знаниях, более успешно, чем любыми другими средствами. При определении целесообразности применения таких систем нужно руководствоваться следующими критериями.

1. Данные и знания надежны и не меняются со временем.
2. Пространство возможных решений относительно невелико.
3. В процессе решения задачи должны использоваться формальные рассуждения.

Существуют системы, основанные на знаниях, но которые пока еще не пригодны для решения задач методами проведения аналогий или абстрагирования (человеческий мозг справляется с этим лучше). В свою очередь традиционные компьютерные программы оказываются эффективнее систем, основанных на знаниях, в тех случаях, когда решение задачи связано с применением процедурного анализа. Системы, основанные на знаниях, более подходят для решения задач, где требуются формальные рассуждения.

4. Должен быть по крайней мере один эксперт, который способен явно сформулировать свои знания и объяснить свои методы применения этих знаний для решения задач.

В таблице 7.2 приведены сравнительные свойства прикладных задач по наличию которых можно судить о целесообразности использования для их решения ЭС.

Таблица 7.2. Критерий применимости ЭС

Применимы ЭС	Не применимы ЭС
Не могут быть построены строгие алгоритмы или процедуры, но существуют эвристические методы решения.	Имеются эффективные алгоритмические методы.
Есть эксперты, которые способны решить задачу.	Отсутствуют эксперты или их число недостаточно.
По своему характеру задачи относятся к области диагностики, интерпретации или прогнозирования.	Задачи носят вычислительный характер.
Доступные данные «зашумлены».	Известны точные факты и строгие процедуры.
Задачи решаются методом формальных рассуждений.	Задачи решаются процедурными методами, с помощью аналогии или интуитивно.
Знания статичны (неизменны).	Знания динамичны (меняются со временем).

В целом ЭС не рекомендуется применять для решения следующих типов задач:

- ✓ математических, решаемых обычным путем формальных преобразований и процедурного анализа;
- ✓ задач распознавания, поскольку в общем случае они решаются численными методами;
- ✓ задач, знания о методах решения которых отсутствуют (невозможно построить базу знаний).

Ограничения в применении экспертных систем

Даже лучшие из существующих ЭС имеют определенные ограничения по сравнению с человеком-экспертом, среди них:

1. Большинство ЭС не вполне пригодны для применения конечным пользователем. Если вы не имеете некоторого опыта работы с такими системами, то у вас могут возникнуть серьезные трудности. Многие системы оказываются доступными только тем экспертам, которые создавали из базы знаний.

2. Вопросно-ответный режим, обычно принятый в таких системах, замедляет получение решений. Например, без ЭС врач может (а часто и должен) принять решение значительно быстрее, чем с ее помощью.

3. Навыки системы не возрастают после сеанса экспертизы.

4. Все еще остается проблемой приведение знаний, полученных от эксперта, к виду, обеспечивающему их эффективную машинную реализацию.

5. ЭС не способны обучаться, не обладают здравым смыслом, интуицией. Домашние кошки способны обучаться даже без специальной дрессировки, ребенок в состоянии легко уяснить, что он станет мокрым, если опрокинет на себя стакан с водой, однако если начать выливать кофе на клавиатуру компьютера, у него не хватит “ума” отодвинуть ее.

6. ЭС неприменимы в больших предметных областях. Их использование ограничивается предметными областями, в которых эксперт может принять решение за время от нескольких минут до нескольких часов.

7. В тех областях, где отсутствуют эксперты, применение ЭС оказывается невозможным.

8. Имеет смысл привлекать ЭС только для решения когнитивных задач. Теннис, езда на велосипеде не могут являться предметной областью для ЭС, однако такие системы можно использовать при формировании футбольных команд.

9. Человек-эксперт при решении задач обычно обращается к своей интуиции или здравому смыслу, если отсутствуют формальные методы решения или аналоги таких задач.

Системы, основанные на знаниях, оказываются неэффективными при необходимости проведения скрупулезного анализа, когда число “решений” зависит от тысяч различных возможностей и многих переменных, которые изменяются во времени. В таких случаях лучше использовать базы данных с интерфейсом на естественном языке.

Проблемы, возникающие при создании ЭС. Перспективы разработки ЭС.

Каталог ЭС и инструментальных программных средств для их разработки содержит тысячи систем. Во многих странах сотни специализированных учреждений, фирм занимаются их разработкой и внедрением.

Вместе с тем, имеются серьезные принципиальные трудности, препятствующие более широкому распространению ЭС и серьезно замедляющие, и осложняющие их разработку. Они вполне естественных и вытекают из самих принципов разработки ЭС.

Первая трудность связана с этапом постановки задач. Большинство заказчиков, планируя разработку ЭС, вследствие недостаточной компетентности в вопросах применения методов ИИ, склонна значительно преувеличивать ожидаемые возможности системы. Заказчик желает увидеть в ней «самостоятельно мыслящего эксперта» в исследуемой области, способного решать широкий круг задач. Отсюда и типичные первоначальные постановки задачи по созданию ЭС, такие, например, как «Разработать ЭС по полностью автоматизированной процедуре обработки изображения», «Создать медицинские ЭС по постановке диагноза и лечению заболеваний сердечно-сосудистой системы у детей младшего возраста» и др. Однако, следует четко отдавать себе отчет, что *мощность эвристических методов решения задач при увеличении общности их постановки резко уменьшается.* Поэтому наиболее целесообразно (особенно при попытке создания ЭС в области, для которой у разработчиков еще нет опыта создания подобных систем) ограничиться для начала не слишком сложной обозримой задачей в рассматриваемой области, для решения которой нет простого алгоритмического способа (то есть неочевидно, как написать программу для решения этой задачи, не используя методы обработки знаний). Кроме того, важно, чтобы уже существовала сложившаяся методика решения этой задачи «какими-то известными способами». Для успешной разработки ЭС необходимы не только четкая и конкретная постановка задач, но и разработка подробного описания метода(ов) ее решения. *Если это сделать затруднительно, дальнейшая работа по построению ЭС теряет смысл.*

Следующая и основная трудность – проблема приобретения (усвоения) знаний. Эта проблема возникает при «передаче» знаний, которыми

обладают эксперты-люди, ЭС. Очевидно, что для того, чтобы «обучить» компьютерную систему, прежде всего требуется сформулировать, систематизировать и формализовать знания. Это может показаться парадоксальным, но *большинство экспертов (за исключением, может быть, математиков), успешно используя в повседневной деятельности свои обширные знания, испытывают большие затруднения при попытке сформулировать и представить в системном виде хотя бы основную часть этих знаний: иерархию используемых понятий, эвристики, алгоритмы, связи между ними.* Оказывается, что для подобной формализации знаний необходим определенный систематический стиль мышления, более близкий математикам и программистам, чем, например, юристам и медикам. Кроме того, необходимы, с одной стороны, знания в области математической логики и методов представления знаний, с другой – знания возможностей вычислительной техники, программного обеспечения, в частности, языков и систем программирования.

На этапе приобретения знаний могут возникнуть трудности и психологического порядка: эксперт может препятствовать передаче своих знаний ЭС, полагая, что это снизит его престиж как специалиста и создаст предпосылки для замены его «машиной». Однако эти опасения лишены оснований: ЭС «уверенно» работает лишь в типовых ситуациях, а также удобна в случаях, когда человек находится в состоянии стресса, в наиболее сложных ситуациях, требующих нестандартных рассуждений и оценок, эксперт- человек незаменим.

Третья серьезная трудность – в очень большой трудоемкости создания ЭС. Создание ЭС требуется выполнения очень большого объема разноплановых работ (средства управления базой знаний, логического вывода, диалогового взаимодействия с пользователем и т.д.). Объем программирования столь велик, а программы столь сложны и нетрадиционны, что имеет смысл, как это принято сейчас при разработке больших программ, на первом этапе создать демонстрационный прототип системы – предварительный вариант, в котором в упрощенном виде реализованы лишь ее основные планируемые возможности и которая будет служить для заказчика подтверждением того, что разработка ЭС для решения данной задачи принципиально возможна, а для разработчиков – основой для последующего улучшения и развития системы.

Одной из причин неудач в создании ЭС в прикладных областях – *недооценка авторами ЭС объемов и роли неявных знаний. Системы, базы знаний которых создаются на основе справочников, зачастую так справочниками и остаются.* Большинство же таких систем оказываются даже хуже справочников, так как «сковывают исследовательскую мысль пользователя».

«Плохая» репутация экспертных систем первого поколения сыграло плохую имиджевую роль для таких систем. Когда стала очевидной слабая эффективность экспертных систем первого поколения, многие специалисты пришли к выводу, что такая технология создания ЭС была тупиковым направлением в развитии информационных технологий.

«Возрождение» ЭС произошло, когда начали создаваться ЭС в виде систем с интеллектуальными базами знаний. Их используют в здравоохранении, страховании, банковском деле и других областях, чтобы с помощью правил и объектов накапливать опыт, повысить качество принимаемых решений. Базы знаний встроены сегодня в наиболее современные крупные системы интеллектуального анализа. Так, они находятся в самой сердцевине программ- агентов, осуществляющих поиск в сети Internet, и помогают коллективам пользователей справиться с потоками информации.

В настоящее время применение ЭС позволяет существенно сократить расходы на подготовку квалифицированного персонала, дальнейшую проверку работоспособности и надежности разрабатываемых и исследовательских систем, а также уменьшить время проектирования и(или) исследований.

Объектная технология, на основе которой создаются и развиваются современные ЭС – значительный шаг вперед по сравнению с CASE-средствами, т.к. она похожа на наше восприятие окружающей действительности. Представление о моделировании меняется, то же самое происходит и с объектами, поэтому сопровождение программируемых объектов может выполняться аналогично приспособлению наших умозрительных образов к изменению окружающих условий. Данная технология прекрасно подходит аналитикам и программистам, так как очень напоминает стратегию решения проблем и соответствует мыслительным процессам людей, считающихся экспертами в своей области.

Чтобы стать экспертом, специалисту нужен инструментарий, имитирующий мышление эксперта. Разработка парадигмы превращается из задачи, чуждой мышлению человека, в знакомое, привычное и легко выполняемое задание. Как работают эксперты? Следуя принципам, заложенным в объектно-ориентированные технологии, они подразделяют проблемы на объекты или классы объектов. По мере накопления знаний в определенной области они делают обобщения, ориентируясь на выделенные объекты или классы объектов. Некоторые обобщения имеют иерархическую структуру, где свойства высших объектов наследуются объектами низшего уровня. Сущность может соответствовать нескольким классам объектов и взаимодействовать с различными объектами или классами. По мере того как знания эксперта углубляются, на их основе

формируются новые ассоциации, а отдельные уровни иерархии пропадают или расширяются.

Методика объектно-ориентированного программирования основана на модели, напоминающей образы, возникающие в мозгу аналитика, которая представляет предметы и процессы в виде объектов и связей между ними. Наблюдая событие, эксперт легко выделяет знакомые образы. Для решения проблем он применяет конкретные правила, рассматривая при этом исследуемую проблему под определенным ракурсом.

Так, например, в настоящее время при разработке систем автоматизированного проектирования (САПР) уже нельзя обойтись без ЭС, которые существенным образом повышают качество проектов, их экономическую и производственную эффективность.

Процесс проектирования и разработки прикладных экспертных систем включает в себя несколько крупных этапов.

Этап идентификации задачи. В первую очередь необходимо выполнить исследование проблемной области, изучить общую задачу. Необходимо четко определиться с основной целью разработки ЭС и с категорией пользователей. Важным моментом является выбор экспертов и коллектива разработчиков. В итоге, эта стадия заканчивается составлением подробного плана разработки ЭС.

Этап «извлечения» и структурирования знаний. Выполняется аналитический (содержательный) анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач. Исходя из этого, определяются с наборами знаний, необходимыми для решения задачи. Определяются способы представления всех видов знаний, формализуются основные понятия, определяются способы интерпретации знаний, моделируется работа системы, оценивается адекватность целям системы зафиксированных понятий, методов решений, средств представления и манипулирования знаниями.

Этап формализации. Осуществляется наполнение экспертом базы знаний. В связи с тем, что основой ЭС являются знания, данный этап является наиболее важным и наиболее трудоёмким этапом разработки ЭС. Процесс приобретения знаний разделяют на извлечение знаний из эксперта, организацию знаний, обеспечивающую эффективную работу системы, и представление знаний в виде, понятном ЭС. Процесс приобретения знаний осуществляется инженером по знаниям на основе анализа деятельности эксперта по решению реальных задач.

На следующем этапе желательно выполнить *разработку прототипа ЭС.*

Прототип системы представляет собой «усечённая» версию системы, предназначенную для проверки правильности «кодирования» фактов,

знаний и стратегий рассуждения экспертов. Стандартный объём прототипа составляет обычно от нескольких десятков до сотни правил.

Следующий этап – *доработка и развитие прототипа до полноценной экспертной системы*. Развитие прототипа до промышленной экспертной системы можно представить как итерационный процесс, включающий в себя такие стадии:

- ✓ Демонстрационный прототип. Система решает часть задач и демонстрирует жизнеспособность подхода. В неё входит несколько десятков правил, понятий и т.п.
- ✓ Исследовательский прототип. Система решает большинство задач, но неустойчива в работе и не полностью проверена. Содержит несколько сотен правил и понятий.
- ✓ Действующий прототип. Система надёжно решает все задачи на реальных примерах, но нет оптимизации ресурсов.
- ✓ Промышленная система. Система обеспечивает высокое качество решения при минимизации затрачиваемых ресурсов.
- ✓ Коммерческая система. Система, ко всему прочему, хорошо документирована, снабжена соответствующим сервисом.

7.3. Инженерия Знаний

Для разработки ЭС необходимо участие в ней особого рода специалистов, выполняющих функции «посредников» между экспертами в предметной области и экспертными системами. Такие специалисты называются **инженеры знаний** (в оригинале – knowledge engineers), а сам процесс разработки ЭС и других интеллектуальных программ, основанных на представлении и обработке знаний – **инженерией знаний** (knowledge engineering). Следует отметить, что во многих странах специальность «инженер знаний» введена в обучение в университетах, в нашей стране основы инженерии знаний изучаются пока в рамках специальных курсов.

Функции эксперта и инженера знаний редко совмещаются в одном лице. Чаще всего функции инженера знаний выполняет разработчик ЭС.

Инженерия знаний имеет целью построение базы знаний в совокупности с подсистемами приобретения знаний и обработки информации. Инженерия знаний имеет своей целью создание интеллектуальных компьютерных систем, позволяющих, в связи с такой внутренней организованностью и построением знаний специалистов, извлекать эти знания и эффективно их использовать.

Изложим более подробно **сведения о Базах знаний**.

База знаний ЭС состоит из упорядоченного множества фактов, информации (знаний) из конкретной области или по конкретному

направлению, а также правил анализа информации. ЭС на основе анализа информации «выдает» рекомендации по решению запроса от пользователя.

Таким образом, база знаний ЭС содержит факты (сведения, информацию из конкретной предметной области) и правила, представляющие собой набор «инструкций», применяя которые к известным фактам можно получать новые факты.

Для построения алгоритмов построения правил и выдачи рекомендаций необходимо построить *логическую модель базы знаний*, основой которой являются правила и методы математической логики (с помощью языка предикатов описание фактов и правил логического вывода, описание обобщённых и конкретных сведений, конкретных и обобщённых запросов к базам данных и базам знаний и др.).

Обычно факты в базе знаний описывают те явления, которые являются постоянными для данной предметной области. Характеристики, значения которых меняются и зависят от условий конкретной задачи, ЭС получает от пользователя в процессе работы, и сохраняет такие сведения в рабочей памяти.

База знаний ЭС создаётся при помощи трёх групп людей:

- эксперты той проблемной области, к которой относятся задачи, решаемые ЭС;
- инженеры знаний, являющиеся специалистами по разработке интеллектуальных информационных систем;
- программисты, осуществляющие реализацию ЭС.

Для первоначального приобретения знаний требуется активная работа всех трех перечисленных категорий специалистов. Она может длиться от нескольких недель до месяцев работы, может даже нескольких лет.

7.3.1. Методология построения системы «Базы Знаний»

Процедура получения Знаний и Данных у специалиста и занесения их в программу называется *извлечением или приобретением знаний*.

Как указывалось ранее, задачи по выявлению знаний у специалиста предметной области (эксперта) и их формализации являются приоритетом *инженера знаний*. *Группа инженеров знаний* (инженер знаний и специалист в предметной области) определяет задачи, которые необходимо решить, выявляет существенные для задач понятия и вырабатывает правила, выражающие отношения между понятиями.

Подсистема приобретения знаний предназначена для добавления в базу знаний новых правил и модификации имеющихся. В ее задачу входит приведение правила к виду, позволяющему подсистеме вывода применять это правило в процессе работы. В более сложных системах предусмотрены

еще и средства для проверки вводимых или модифицируемых правил на непротиворечивость с имеющимися правилами.

В таблице 7.3 показаны основные этапы построения (эволюции) Баз Знаний (и Данных) в процессе приобретения знаний.

Таблица 7.3. Этапы эволюции системы «Базы Знаний»

Идентификация	Определение характеристик задачи
Концептуализация	Поиск понятий для представления знаний
Формализация	Разработка структур для организации знаний
Реализация	Формулировка правил, воплощающих знания
Испытания	Оценка правил, в которых воплощено знание

В ходе *идентификации* инженер знаний и специалист работают вместе над вопросом выделения проблемной области и интересующей их темы, указывают круг участников процесса создания системы (например, дополнительных специалистов), определяют задачи и цели построения системы.

В ходе *концептуализации* специалист предметной области и инженер знаний выявляют основные понятия, отношения и характер информационных потоков, необходимые для описания процесса решения задач в данной предметной области.

Этап *формализации* связан с отражением ключевых понятий и отношений данной предметной области в некотором формальном представлении, связанном с выбором системы и языка программирования.

В ходе *реализации* формализованные знания комбинируются и реорганизуются для достижения их совместимости с характером информационных потоков задачи.

И в ходе *испытания* производится оценка работы программы-прототипа и ее пересмотр с целью приведения в соответствие с принятыми стандартами высокого качества функционирования.

7.3.2. Построение Баз Знаний на основе сведений, полученных от эксперта

Процесс построения Баз Знаний состоит из трех этапов:

- Описание предметной области;
- Выбор метода и модели представления знаний;
- Приобретение Знаний.

В свою очередь процедура описания предметной области включает в себя следующие этапы:

- определение характера решаемых задач;
- выделение объектов предметной области;

- установление связей между объектами;
- выбор модели представления Знаний;
- выявление специфических особенностей предметной области.

Главная цель начального этапа построения Базы Знаний – определить, как будет выглядеть описание предметной области на различных уровнях абстракции. Основные шаги абстрагирования при построении Баз Знаний (и Данных) можно представить в виде схемы, изображенной на рисунке 7.1.

Замечание. Так как знания в обязательном порядке требуют наличия и данных (неразрывно связаны с ними), то для того, чтобы подчеркнуть важность данных, Базы Знаний часто называют Базы Знаний и Данных.

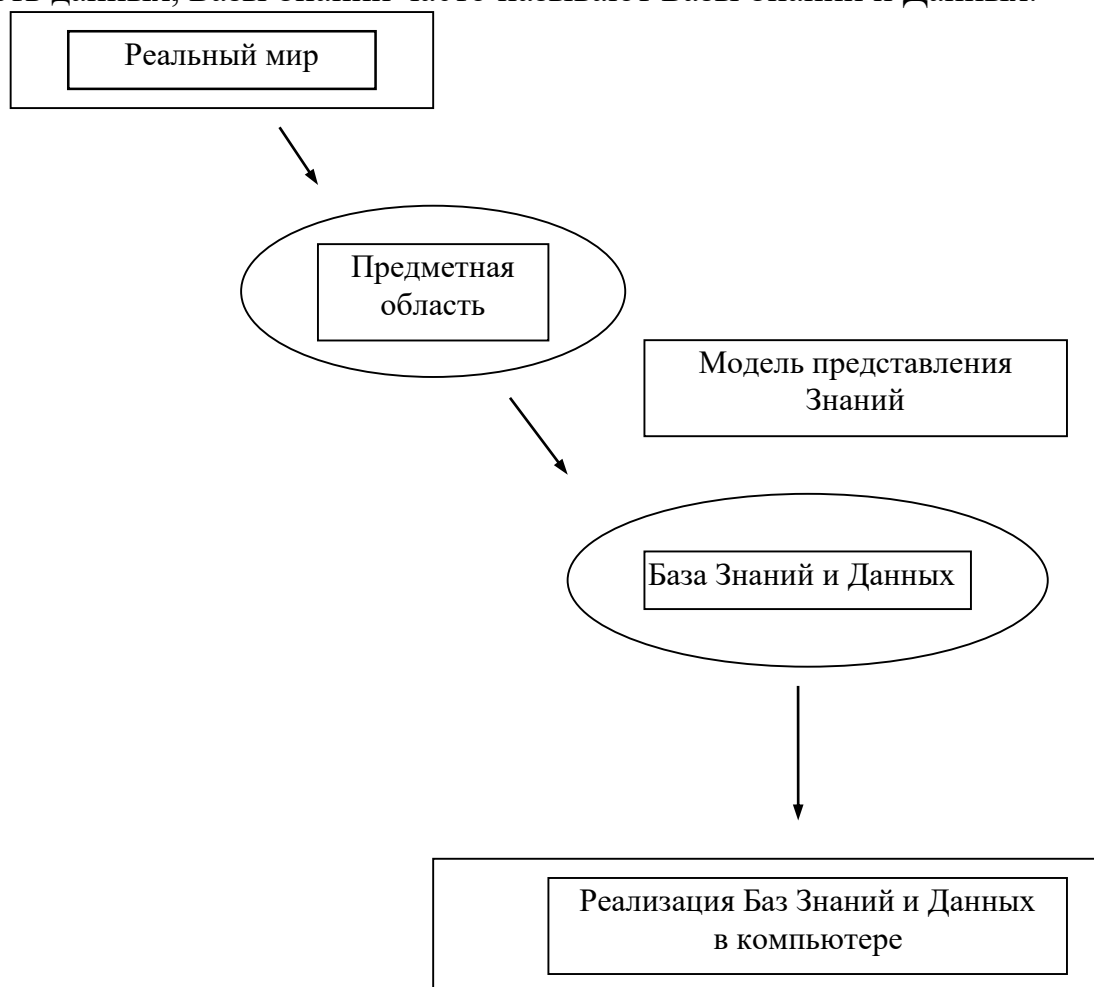


Рис. 7.1. Этапы создания Баз Знаний для предметной области

После выделения предметной области необходимо ее формально описать. Для этого выбирается какой-либо способ представления знаний о ней (**модель представления Знаний**). При этом должны воспользоваться той моделью, с помощью которой можно лучше всего отобразить специфику предметной области.

Запись содержания Баз Знаний (и Данных) специальными средствами в компьютере называется ее реализацией. Способ реализации Баз Знаний в

компьютере называют **архитектурой** системы, чтобы отличить его от модели представления Знаний.

7.3.3. Требования к Знаниям

Несколько слов о структурировании знаний предметной области.

Глубинные знания должны отражать наиболее общие принципы, в соответствие с которыми развиваются все процессы в предметной области, а также свойства и характерные черты этих процессов.

Поверхностные знания представляют собой эвристики (набор элементарных правил) и отдельные закономерности, устанавливаемые опытным путем и используемые при отсутствии общих теорий и методов.

Глубинные знания можно определить, как совокупность основных закономерностей, аксиом и фактов о конкретной предметной области.

Содержание и наполнение глубинных и поверхностных знаний является относительным и динамическим и для конкретной предметной области определяется выбором природы и характера класса рассматриваемых задач.

Необходимо подчеркнуть, что если человек в процессе мышления и анализа не различает упомянутые виды иерархий и равномерно применяет как поверхностные, так и глубинные знания при решении задач, то при построении Базы Знаний, напротив, обязательно нужно очень четко представлять, какого типа знания и какого вида иерархия включается в нее (Базу Знаний). Для получения глубинных знаний необходимо понять внутренние механизмы, действующие в предметные области и основные закономерности, которые обуславливают возможность принятия правильных решений.

Для получения четкого, однозначного и корректного ответа на сложный вопрос в системе Баз Знаний необходимо выполнение следующих требований:

1. Знания и данные должны быть надежными;
2. Знания и данные должны быть неизменными (на момент получения ответа);
3. Пространство возможных решений (ответов) должно быть небольшим.

Следует отметить, что в реальных приложениях перечисленным требованиям удовлетворяют, к сожалению, далеко не все источники знаний и исходных данных.

Рассмотрим каждое из требований более подробно.

Первое требование состоит в том, чтобы знания и данные не содержали ошибок и неточностей, были четко сформулированы и не имели пропусков. Кроме того, важным является согласованность и корректность знаний. Главным преимуществом использования надежных знаний и данных

является монотонность системы. При этом вполне достаточным оказывается построение одной линии рассуждения, т.е. нет необходимости в поиске многих аргументов, подтверждающих потенциальные заключения.

Представляется обычной ситуация, когда знания и данные ненадежные: не все данные есть в наличии, достоверность некоторых из них вызывает большие сомнения и т.д. Такая ситуация в принципе является нормальной. *Проблема построения выводов на основе неопределенных или неполных данных требует использования специальных подходов.*

Одним из наиболее простых подходов является *модель приближенной импликации*, суть которой заключается в использовании *коэффициентов уверенности*, указывающих силу (достоверность, надежность) данного правила. Такой подход позволяет отойти от точности, присущей исчислению предикатов, где утверждения просто или истины, или ложны. Рассуждения являются неточными, и эта неточность имеет численное выражение.

Замечание 1. Коэффициенты уверенности идут от вероятностей, но в общем случае между ними есть серьезные отличия.

Замечание 2. Вместо подхода «приближенной импликации» можно (и это представляется более строгим) было бы воспользоваться *правилом Байеса определения условных вероятностей*. Однако в этом случае необходимо использовать огромное количество данных, что на практике не представляется возможным.

Другим подходом, отклоняющимся от классической логики, является *нечеткая логика*, рассмотренная, например, Заде. В этом случае вводится понятие *неточности*. Нечеткое множество представляет собой множество значений величины с соответствующими им значениями возможности. Имеются свои правила вывода для нечетких множеств, т.е. при заданном множестве $A=\{a\}$ нечеткое множество F определяется путем задания функции принадлежности $\mu(a)$, определенной на A и принимающей значения из $[0;1]$.

Второе требование значительно образом упрощает процедуры работы с подсистемой и позволяет избежать необходимости следить за постоянными изменениями знаний и данных со временем.

Большинство задач требуют проведения рассуждений о ситуациях, меняющихся во времени. В этом случае удобно использовать, например, *подход «ситуационного исчисления»*, предложенный Джоном Маккарти [49, 50]. В этом случае «ситуация» включается в процессе моделирования наряду с другими объектами. В ситуационном исчислении действия представлены функциями, областью определения и областью значения которых являются ситуации.

Требование небольшого размера пространства поиска решений

(*третье требование*) позволяет не предпринимать каких-либо мер для учета ограниченности вычислительных ресурсов и нет необходимости думать о специальных структурах данных, специальной оптимизации поиска решения.

Оптимальным является требование того, чтобы пространство поиска было небольшим. Рассмотрим *методы работы в случае очень больших пространств поиска*.

В большинстве задач анализа данных желательно находить все возможные интерпретации, согласующиеся с этими данными. *Идеальным является рассмотрение всех возможных случаев и исключения тех из них, которые несовместимы с данными*. Однако, часто на практике не представляется возможным рассмотреть все возможные решения.

Метод исключения (порождения и проверки). Работоспособность метода обеспечивается отсечением ситуаций на ранних этапах цикла порождения и проверки (порождения вершин общей процедуры и их проверки). Однако с увеличением правил отсечения возрастает вероятность того, что решения могут быть пропущены и кроме того, вводимые дополнительные новые специализированные правила начинают охватывать все меньшее число случаев.

Имеются задачи (их достаточно много), для которых неприменим метод порождения и проверки: нельзя построить генератор решений, допускающий отсечения на начальном этапе поиска. Для таких задач можно использовать *метод сопоставления*, относящийся к «слабым методам» поиска. Метод сопоставления отправляется от начального состояния, проходит по промежуточным и заканчивает работу в конечном состоянии без возвращения назад в ходе поиска. Таким образом, главное требование метода сопоставления состоит в том, что нет возврата при поиске. Это значит, что в каждом промежуточном состоянии система должна определить, лежит ли состояние на пути к решению. Для этого, в свою очередь, необходимо, чтобы на множестве решений существовал частичный порядок, такой, что последствия применения некоторого оператора касались бы только идущих далее частей решения.

Другим методом поиска решения таких задач служит *метод использования абстрактного решения*. В случае, если рассматриваемое множество задач достаточно велико, то очевидно, что нет возможности каждую задачу разбить на одни и те же подзадачи. При таких условиях, например, метод сопоставления мало эффективен. Необходимо, чтобы используемый подход отражал переменную структуру планов.

Можно применять *метод последовательного уточнения сверху вниз*. В противоположность методу сопоставления последовательное уточнение сверху вниз устанавливает отношение частичного порядка лишь для

промежуточных состояний системы. Одной из основных трудностей, связанных с последовательным уточнением сверху вниз, является отсутствие обратной связи от процесса решения задач. Предполагается, что в одной и той же точке (т.е. на некотором уровне критичности) для каждой задачи из рассматриваемого раздела должны быть приняты решения одного и того же вида.

Совершенно на другом принципе управления процессом рассуждения основан подход, называемый **принципом наименьшей предвзятости**.

Основная идея принципа состоит в том, что решения не следует принимать произвольным образом или раньше времени. Их следует откладывать до того момента, когда для их принятия будет достаточно информации.

Принцип наименьшей предвзятости применим, если система может:

- знать, когда имеющейся информации достаточно для принятия решения;
- временно приостановить решение подзадачи, если необходимая информация отсутствует;
- переходить от одной подзадачи к другим, возобновляя работу по мере получения необходимой информации;
- объединять информацию от различных подзадач.

Принцип наименьшей предвзятости координирует принятие решений с наличием необходимой информации и обеспечивает переключение фокуса внимания системы между имеющимися подзадачами. Принцип наименьшей предвзятости непригоден, когда есть много альтернатив и отсутствуют веские причины для определенного выбора одной из них. В этом случае необходима та или иная форма правдоподобных рассуждений.

Выдвижение гипотез или построение правдоподобных умозаключений составляет неотъемлемую часть **эвристического поиска**.

Характерными ситуациями, в которых выдвижение гипотез играет важную роль, являются, например, следующие:

а) Система знаний является неполной. В таких случаях системы решений на какой-то стадии решения в некоторых случаях не может сделать наилучший выбор и решение задачи не может быть завершено без выдвижения гипотез.

б) Пространство поиска может содержать весьма много решений. В этой ситуации выдвижение гипотез может быть действенным средством для сокращения количества решений.

в) Существует эффективная процедура построения решений путем систематического улучшения аппроксимаций (например, уточнение сверху вниз). Выдвижение гипотез оказывается уместным в случаях, когда наблюдается быстрая сходимости процесса. Главная трудность состоит в

обнаружении неверных гипотез и разработке эффективного способа отказа от таких гипотез.

При поиске решений необходимо помнить, что единственной линии рассуждений обычно недостаточно, кроме того недостаточно и использования единственного источника знаний.

7.3.4. Приобретение знаний. Конструирование системы

Приобретение знаний – это процесс передачи и преобразования опыта по решению задач от некоторого источника знаний в программу.

Потенциальные источники знаний включают в себя экспертов, специальную литературу и базы данных. Отметим, что знания представляют собой набор специальных фактов, процедур, оценок и правил принятия решений в определенной предметной области, а не общие знания об этой области (охватить все знания из предметной области не представляется возможным) в целом.

Одним из наиболее трудных аспектов является структуризация предметных знаний, определение и формализация предметных концепций.

Основу знаний образуют: *символьные описания*, характеризующие эмпирические и дефинитивные (по определению) отношения в некоторой области и *процедуры*, предназначенные для обращения (работы) с этими описаниями.

Процесс «приобретения знаний машиной» очень сложен и трудоемок. На рис.7.2 показана условная схема передачи знаний от источника в Базу Знаний через инженера (конструктора) знаний.

База Знаний должна быть отделена от остальных программ (быть достаточно самостоятельной системой). Ее главными достоинствами должны быть прозрачность и гибкость.

Замечание. Прозрачность системы означает, что система понятна как разработчику, так и пользователю.

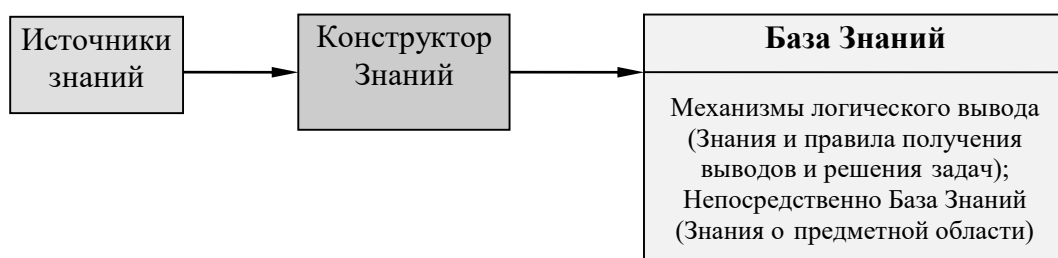


Рис. 7.2. Условная схема инженерии знаний - передача знаний от источника в базу знаний через инженера знаний

База Знаний должна быть отделена от остальных программ (быть достаточно самостоятельной системой). Ее главными достоинствами должны быть прозрачность и гибкость.

Замечание. Прозрачность системы означает, что система понятна как разработчику, так и пользователю.

Процесс приобретения знаний может быть условно представлен в виде схемы, представленной на рисунке 7.3.

На схеме рис. 7.3 в качестве конструктора знаний может выступать непосредственно инженер знаний или интеллектуальная редактирующая программа. Источником знаний может служить эксперт (человек, обладающий определенным запасом знаний), литературные источники или какие-то иные исходные базы данных.



Рис. 7.3. Стадии процесса приобретения Знаний

Обычно выделяют *три типа знаний*, приобретение которых соответственно требует применения трех различных методов:

- *операционные*, необходимые для решения задачи в текущий момент;
- *вспомогательные*, т.е. факты, которые эксперт может и не помнить, но обязан знать, где их можно найти;
- *дополнительные*, т.е. сведения о внешних условиях, в которых будет происходить решение задачи.

Простые системы, основанные на знаниях, ведут с пользователем достаточно элементарный диалог, в котором он может обойтись словами «да», «нет» и может быть «почему?».

7.3.5. *Формы и способы представления знаний*

Существует несколько способов представления знаний в ЭС, однако общим для всех них является то, что знания представляются в символической форме (элементарными компонентами представления знаний являются тексты, списки и другие символические структуры). Тем самым, в ЭС реализуется принцип символической природы рассуждений, который заключается в том, что процесс рассуждения представляется как последовательность символических преобразований.

Знания могут принимать *различную форму*. Это могут быть *эмпирические ассоциации*. Данный тип является широко распространенным среди специалистов горного профиля, которые применяют эвристические методы для сочетания вероятностных, содержащих ошибки и заранее не полностью достоверных исходных данных, построения на их основе правил вывода и получения собственно выводов.

Знания могут быть сформулированы в форме *понятий, ограничений и закономерностей*, которыми регламентируются действия в определенной конкретной предметной области, подобласти, разделе.

Еще одной формой представления знаний является использование *причинных моделей* для изучаемых систем и *схем логического вывода*, используемых эти причинные модели для предсказания, диагностики, планирования и анализа ситуаций.

Наиболее распространенный способ представления знаний – в виде конкретных фактов и правил, по которым из имеющихся фактов могут быть выведены новые. Факты представлены, например, в виде троек:

(Атрибут – Объект - Значение).

Такой факт означает, что заданный объект имеет заданный атрибут (свойства) с заданным значением. Например, тройка (Модуль упругости – материал_1 – 10^5) представляет факт «модуль упругости материала №1 составляет значение 10^5 Н/м^2 ». В более простых случаях факт выражается

неконкретным значением атрибута, а каким-либо простым утверждением, которое может быть истинным или ложным, например: «В исследуемом объекте образовалась система макротрещин». В таких случаях факт можно обозначить каким-либо кратким именем (например, макротрещины) или использовать для представления факта сам текст соответствующей фразы.

Правила в базе знаний имеют вид: «ЕСЛИ A ТО S , где A – условие, а S – действие. Действие S выполняется, если A истинно. Наиболее часто действие S , так же, как и условие, представляет собой утверждение, которое может быть выведено системой (то есть становится ей известной), если истинно условие правила A ».

Правила в базе знаний служат для представления *эвристических знаний (эвристики), то есть неформальных правил рассуждения, вырабатываемых экспертом на основе опыта его деятельности.*

Простой пример правила: «ЕСЛИ в исследуемой области твердого тела нарушаются условия предельного состояния, ТО в данной области образуется система трещин».

В качестве условия A может выступать либо факт (как в данном примере), либо несколько фактов A_1, \dots, A_N , соединенных логической операцией И: « A_1 И A_2 И ... И A_N ». Как известно, такое выражение в математической логике называется конъюнкцией. Оно считается истинным в том случае, если истинны все его компоненты. Пример предыдущего правила с более сложным условием: «ЕСЛИ в исследуемой области твердого тела нарушаются условия предельного состояния И напряженное состояние соответствует состоянию обобщенного растяжения, ТО в данной области образуется система трещин» (Правило 1).

Действия, входящие в состав правил, могут содержать новые факты. При применении таких правил, эти факты становятся известны системе, то есть включаются во множество фактов, которое называется рабочим множеством. Например, если факты «В исследуемой области твердого тела нарушаются условия предельного состояния» и «Напряженное состояние соответствует состоянию обобщенного растяжения» уже имеются в рабочем множестве, то после применения приведенного выше правила в него также включается факт «В данной области образуется система трещин».

Если система не может вывести некоторый факт, истинность или ложность которого требуется установить, то система спрашивает о нем пользователя. Например: «ВЕРНО ЛИ, ЧТО в исследуемой области твердого тела нарушаются условия предельного состояния?»

При получении положительного ответа от пользователя факт «В исследуемой области твердого тела нарушаются условия предельного состояния» включается в рабочее множество.

Как упоминалось ранее, существуют *динамические и статические*

базы знаний. Динамическая база знаний изменяется со временем. Новые факты, добавляемые в базу знаний, являются результатом вывода, который состоит в применении правил к имеющимся фактам.

В системах с монотонным выводом факты, хранимые в базе знаний, статичны, то есть не изменяются в процессе решения задачи. В системах с немонотонным выводом допускается изменение или удаление фактов из базы знаний. В качестве примера системы с немонотонным выводом можно привести ЭС, предназначенную для составления перспективного плана производства (например, плана развития горных работ). В такой системе по вашему желанию могут быть изменены даже те данные, которые после вывода уже вызвали срабатывание каких-либо правил. Иными словами, имеется возможность модифицировать значения атрибутов в составе фактов, находящихся в рабочей памяти. Изменение фактов в свою очередь приводит к необходимости удаления из базы знаний заключений, полученных с помощью упомянутых правил. Тем самым вывод выполняется повторно для того, чтобы пересмотреть те решения, которые были получены на основе подвергшихся изменению фактов.

7.3.6. Организация хранения информации

Способ хранения символьных образов практически не отличается от способа хранения числовой информации. Для целей хранения информации обоих типов удобно использовать *Базы Данных сетевого типа*.

В базах такого типа для хранения информации применяется *принцип запоминания совокупности элементов, поднаборов, наборов и моделей данных*. Элементы данных «принадлежат» поднаборам, наборам и моделям. Конкретная модель составляется из одного или нескольких наборов, поднаборов и элементов данных (рис.7.4.).

Между моделями и наборами имеется связь типа «родитель-потомок». В отличие от общепринятых родственных связей набор может являться потомком более чем одной модели.

Для указанной структуры Баз Данных необходима разработка системы указателей, что в значительной степени облегчает пользователю ведение всей совокупности информационных объектов. Представляется важным, чтобы при выборке какого-либо элемента совокупности (например, набора) руководитель имел возможность извлечь все связанные с ним данные (поднаборы, элементы и т.д.).

Пример. Важной особенностью разработки Баз Знаний для СРГМ (системы регионального геомеханического мониторинга) является хранение символьных объектов не отдельно, а хранение наборов фактов и связей между ними как единого целого. Набор фактов и связей между ними называется **чанком**. Например, рассмотрим следующее предложение: «При

подземной разработке полезных ископаемых на земной поверхности образуется мульда сдвижения». Объектами здесь могут являться буквы. Тогда целесообразно объединить буквы в чанки, в качестве которых в данном случае будут выступать слова и хранить в памяти только слова (чанки) и связи между ними. При ином рассмотрении в качестве объектов можно рассматривать слова. Тогда чанки будут представлять из себя причинно-следственные фразы и предложения (например, мульда сдвижения).

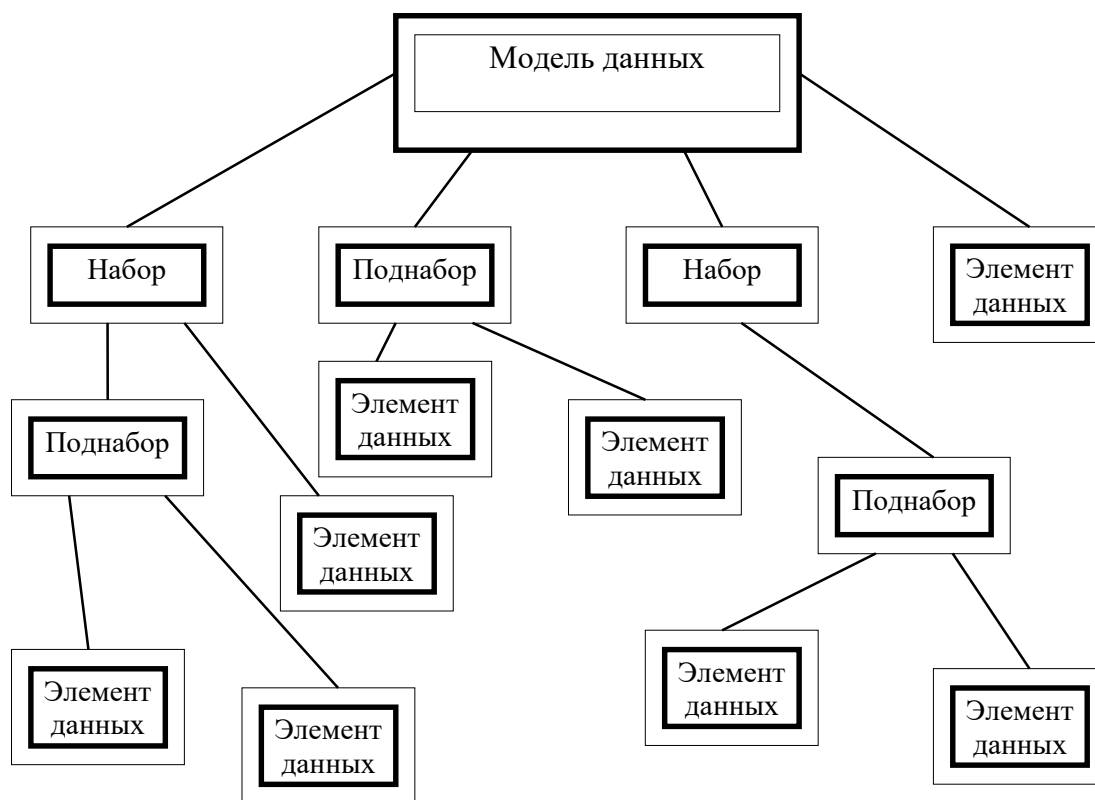


Рис. 7.4. Структура сетевой Базы данных

То есть целесообразнее всего организовать структуру Знаний так, чтобы по возможности хранить чанки и взаимосвязи между ними, а не отдельные объекты.

Замечание. Способность формировать чанки отличает эксперта в конкретной предметной области от «неэксперта». Эксперт обладает способностью объединять в чанки большие объемы данных и устанавливать иерархические связи между ними для того, чтобы быстро извлекать эти данные из памяти и с их помощью распознавать новые ситуации.

7.4. Принципы построения Баз Знаний

7.4.1. Типы иерархической структуры для построения Баз Знаний

Главной целью начального этапа построения системы Баз Знаний

является определением структуры описания системы на различных уровнях и выбором способа представления знаний (модель представления знаний).

Представление Знаний является наиболее важной областью разработки предметных интеллектуальных систем. Знания имеют форму описаний объектов, взаимосвязей и процедур. Разработка удобного метода представления знаний означает выработку способа выражения общих закономерностей предметной области и возможность накопления знаний, необходимых для ежедневного решения новых задач.

Способ представления знаний значительным образом определяется выбором главного целевого назначения системы и основных задач, стоящих перед системой. Для установления иерархической структуры системы и разбивки ее на отдельные модули далее необходимо сформулировать подцели и определить связи между отдельными фрагментами системы. Следующим шагом является выделение объектов предметной области и установление границ системы. После выявления объектов необходимо установить связи между ними и способ организации выделенных объектов. Можно выделить иерархии трех видов: функциональную, причинно-следственную (отражает поведение системы) и структурную (табл.7.4).

Таблица 7.4. Виды иерархии знаний

Знания	Иерархия	Применение	Вид логики
Поверхностные	Структурная	Описание структуры предметной области	Причинно-следственная логика
	Причинно-следственная	Описание причинно-следственных связей (модель поведения предметной области)	
Глубинные	Функциональная	Описание предметной области в терминах функций ее элементов	Логика целей

Систему Базы Знаний (и Данных) можно представить в виде некоторой иерархической структуры.

В вершине иерархии находится собственно система «Базы Знаний (и Данных)» в целом.

На следующем уровне система представляется в виде набора скомпонованных блоков-разделов.

Третий уровень образован документами, из которых состоит каждый блок-раздел.

На еще более низком уровне находится описание основных понятий, определения и т.д. и т.п.

Движение по структуре осуществляется сверху вниз, т.е. строится (определяется) вся цепочка вплоть до элементарных (базовых) понятий.

Замечание. Этот метод локализации необходимых сведений и документов не требует наличия у проверяющего (изучающего) достаточных сведений из предметной области, т.е. знания, на основе которых осуществляется рассмотренный поиск, могут быть поверхностными.

Причинно-следственная иерархия позволяет построить модель «поведения» системы «Базы Знаний», т.е. функционирования в терминах «причина-следствие».

Так, например, рассмотрение вопроса «Проблема защиты рудников от затопления» влечет за собой: во-первых, ссылку на «Руководство по предотвращению ...». Для дальнейшей конкретизации необходимо задание уточняющих характеристик и переход на следующий уровень иерархии для конкретизации сути вопроса.

Таким образом, причинно-следственная иерархия использует (опирается) на знания более глубокого уровня (чем структурная иерархия).

Функциональная иерархия представляет собой самый высокий уровень абстракции. При этом система «Базы Знаний» рассматривается как иерархия подсистем и модулей в совокупности с функциональным назначением подсистем (т.е. обязательно с определенными функциями, присущим подсистемам).

Так, например, функциональное назначение системы в целом заключается в обеспечении специалистов и служащих всех рангов и подразделений, имеющих отношение к проблемам и вопросам геомеханического обеспечения горных работ, всем любым набором необходимых сведений и документов по всем вопросам геомеханического обеспечения горных работ. На следующем уровне функциональной иерархии имеется целый спектр функций.

Обычно для выбранной предметной области строится иерархия всех трех перечисленных видов. Однако, эта процедура должна быть гибкой. То есть для системы «Базы Знаний (и Данных)» это означает, что для различных подсистем основной системы могут быть использованы различные виды иерархии или их совокупности в зависимости от типа, характера и сути непосредственной рассматриваемой системы.

Так, структурная иерархия может быть введена при рассмотрении подсистемы «Строение, состав подрабатываемой породной толщи». В нее входят следующие уровни: породная толща в целом, отдельные характерные зоны подрабатываемого породного массива, затем, пласты, слои и т.д. Следующий пример – функциональная иерархия при построении

системы «Деформационные процессы в подрабатываемых породной толще». При этом за верхний уровень можно принять сдвигения и деформации на земной поверхности, затем следует уровень «деформационные процессы в приповерхностной толще, водозащитной толще». Дробление проводится до уровня деформационных процессов на уровне отрабатываемого пласта.

Таким образом, содержание Базы Знаний определяется тем, какие объекты, взаимосвязи между ними и виды иерархий выделены в предметной области.

Для системы «Базы Знаний (и Данных)», по-видимому, наиболее эффективным и распространенным представляется использование причинно-следственной иерархии.

Причиной при этом называем такую связь между двумя событиями или проявлениями признаков, если одно из них с необходимостью влечет за собой другое. Т.е. причинная связь представляется в терминах модели поведения предметной области.

При переходе в описании предметной области с уровня поверхностных на уровень глубинных знаний обязательным образом увеличивается количество имеющейся информации о ней, а также о том, каким образом ее отдельные составляющие модули объединяются в единое целое. Для перехода от структурной иерархии к описанию поведения системы требуется использование соответствующего теоретического аппарата, в качестве которого *выступает причинно-следственная логика*. В свою очередь для получения функциональной иерархии на основе модели поведения системы необходимо использовать *логику целей*.

Замечание. Очевидно, что функциональная иерархия соответствует наиболее глубокому уровню знаний о предметной области.

7.4.2. Этапы построения системы «Базы Знаний»

Итак, создание Базы Знаний и Данных (БЗД) осуществляется в следующей последовательности:

- выявляются объекты предметной области;
- выявляются взаимосвязи между ними;
- определяются виды иерархий;
- объекты разбиваются на классы;
- структурируется База Знаний.

Создание системы БЗД состоит из двух основных фаз.

- **Первая фаза** – идентификация и концептуализация проблемы.

Идентификация включает в себя, например, отыскание эксперта, источников знаний и ресурсов, ясную формулировку проблемы.

Концептуализация предусматривает выбор основных понятий и связей, необходимых для описания проблемы.

• **Вторая фаза** имеет дело с формализацией, практической реализацией и тестированием подходящей архитектуры системы и включает постоянное переформулирование понятий, изменение форм представления и доведение до совершенства реализованной системы.

Технически приобретение знаний состоит из трех основных задач:

- ввод данных и знаний в систему,
- устранение ошибочных данных и знаний,
- выверка и расширение знаний.

Некоторые руководящие принципы построения и создания базы знаний.

- Необходимо четко следить за тем, чтобы не выходить за рамки специализированной области. Стараться предельно ясно и четко формулировать задачи и выводы. Определять важнейшие стороны общей проблемы.
- Составить список использованных терминов, подзадач.
- Отделять проблемно-ориентированные знания от универсальных знаний.
- Создать библиотеку обработанных системой случаев.
- Обеспечивать некоторую избыточность системы.

7.4.3. Построение системы Базы Знаний по принципу «Продукционной системы»

Систему БЗД можно строить по принципу «**Продукционная система**». Последняя представляют собой систему, основанную на правилах и систему с выводом, использующей сопоставление по образцу.

Продукционная система включает в себя три основных составляющих: базу правил, рабочую память и механизм вывода. В дополнение к ним в нее обычно входят еще и подсистема приобретения знаний, средства общения на естественном языке и подсистема объяснения (рис.7.5).

Конкретизируем каждую из компонент.

База правил (память для хранения правил)

Память для хранения правил представляет собой один из необходимых компонентов системы БЗД. База правил содержит набор срабатывающих в определенных ситуациях правил, имеющих общую форму «ЕСЛИ-И-ТО».

Конструкции такого типа носят название **продукционных правил**. Каждое правило складывается из двух частей. Первая из них – *посылка правил* – состоит из элементарных предложений, соединенных логическими связками «И, ИЛИ» и т.д. Вторая часть – *заключение* – состоит из одного

или нескольких предложений, которые образуют выдаваемое правилом решение либо указывают на действие, подлежащее выполнению (табл. 7.5).

Правило может содержать **атрибуты** и **значения** (так в примере 3 табл. 7.5 атрибут – горные работы, значение – особые зоны).

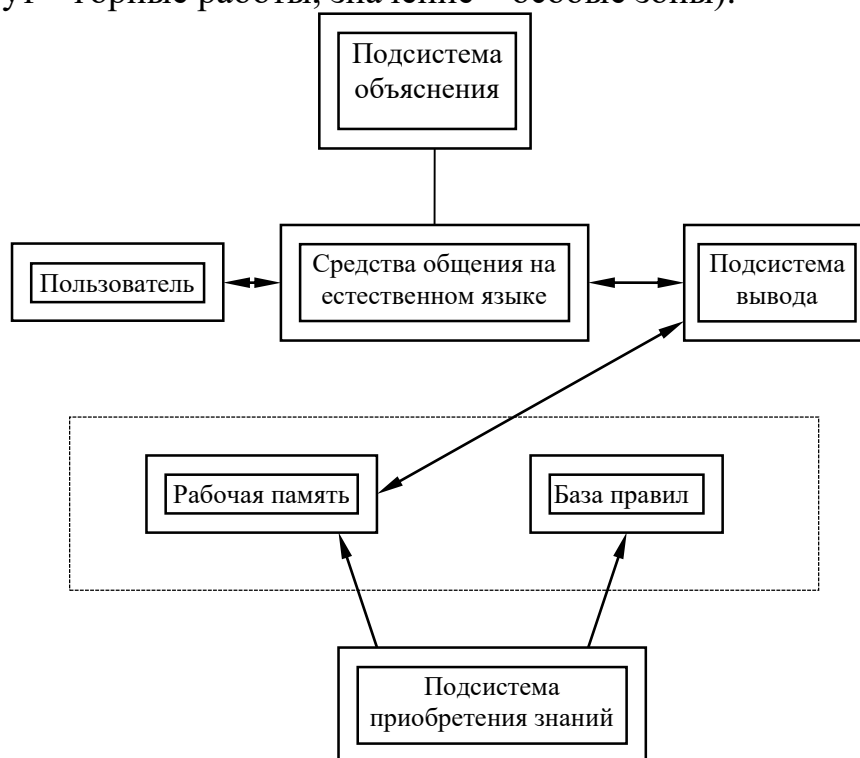


Рис. 7.5. Принципиальная структура системы БЗД как продукционной системы

Таблица 7.5. Пример продукционных правил

1.	ЕСЛИ И ТО	Рассматривается камерная система отработки Рассчитываются основные характеристики ожидаемых сдвижений и деформаций на земной поверхности Выдается «Методика прогнозирования сдвижений и деформаций земной поверхности при применении камерной системы отработки»
2.	ЕСЛИ И ТО	Рассматриваются мероприятия по охране зданий и сооружений в зоне подработки Изучается состояние жилых строений в сельской местности Выдаются «Указания по охране сооружений и природных объектов от вредного влияния подземных горных разработок»
3.	ЕСЛИ ТО	Горные работы подошли к особой зоне Изучить инструкции по ведению горных работ в особых условиях

Другая важная часть системы – *База Данных (или рабочая память)*. В этой памяти хранится все множество документов и фактов, описывающих рассматриваемую предметную область и все пары «атрибут-значение», установленные к определенному времени. Содержимое рабочей памяти со временем изменяется.

База Знаний включает в себя совокупность правил и содержимое

Базы Данных.

Содержимое *рабочей памяти* содержит и набор решений, построенных на основе применения правил к имеющимся данным в Базе.

Система БЗД представляет собой *систему с монотонным выводом фактов*, т.е. хранимая в рабочей памяти информация не изменяется в процессе принятия выбора решения.

Подсистема вывода (интерпретатор правил)

Подсистема вывода (интерпретатор правил) – программная компонента экспертных систем, реализующая процесс рассуждений на основе базы знаний и рабочего множества. Она выполняет две основные функции:

- во-первых, просмотр существующих фактов из рабочей памяти и правил из базы знаний и добавление (по мере возможности) в рабочую память новых фактов;

- во-вторых, определение порядка просмотра и применения правил.

Эта подсистема управляет ***процессом консультации***, сохраняя для пользователя информацию о полученных заключениях и запрашивает у него информацию, когда для срабатывания очередного правила в рабочей памяти оказывается недостаточно данных.

Цель ЭС – вывести некоторый заданный факт, который называется *целевым утверждением* (то есть в результате применения правил добиться того, чтобы этот факт был включен в рабочее множество), либо опровергнуть этот факт (то есть убедиться, что его вывести невозможно, следовательно, при данном уровне знаний системы он является ложным). Целевое утверждение может быть либо «заложено» заранее в базу знаний системы, либо извлекается системой из диалога с пользователем.

Работа подсистемы вывода представляет собой последовательность шагов, на каждом из которых из базы выбирается некоторое правило, которое применяется к текущему содержимому рабочего множества. Цикл заканчивается, когда выведено либо опровергнуто целевое утверждение.

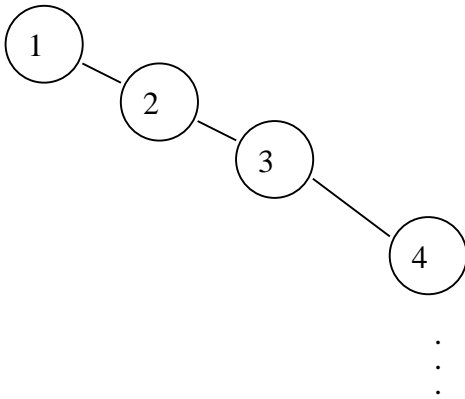
Цикл работы экспертной системы иначе называется логическим выводом. Логический вывод может происходить многими способами, из которых наиболее распространенные – ***прямой порядок вывода и обратный порядок вывода***.

Схема прямого и обратного выводов схематично представлена на рисунке 7.6.

Прямой порядок вывода – от фактов, которые находятся в рабочем множестве, к заключению. Если такое заключение удастся найти, то оно заносится в рабочее множество. Прямой вывод часто называют выводом, управляемым данными.

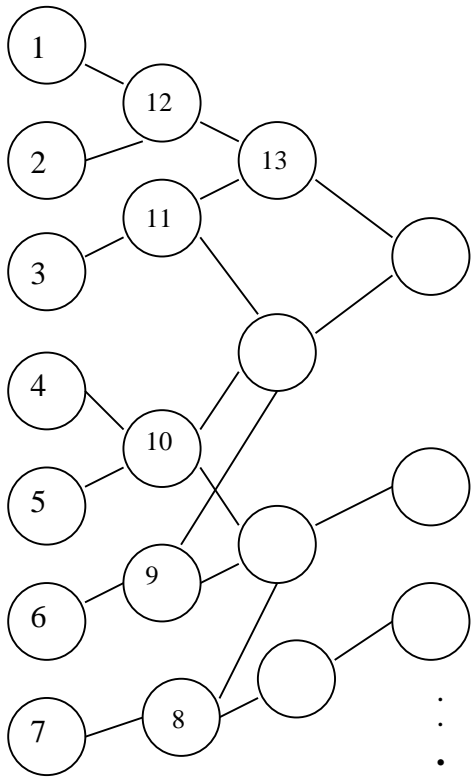
Прямой вывод

**Начало
поиска**



Заключение

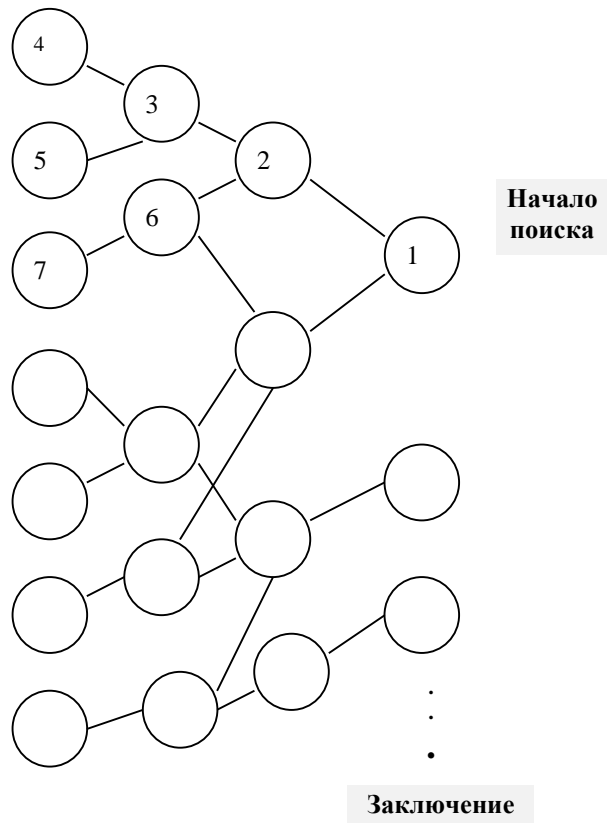
**Начало
поиска**



Заключение

Обратный вывод

Поиск в глубину



Поиск в ширину

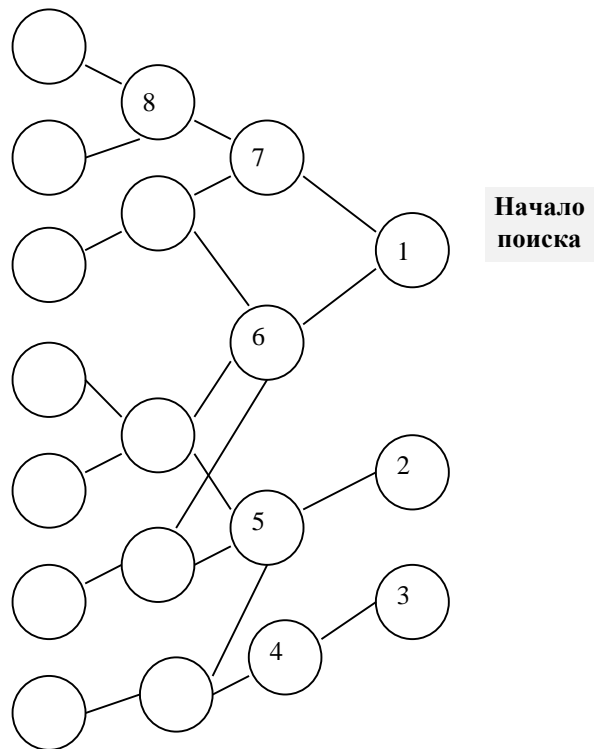


Рис.7.6. Графы, описывающие стратегию вывода

Для иллюстрации добавим к нашему примеру базы знаний о появлении в твердом деформируемом теле системы трещин еще одно правило: «ЕСЛИ в исследуемой области образуется система трещин, ТО необходимо решить вопрос об эксплуатационной пригодности объекта» (правило 2).

Предположим также, что факты «В исследуемой области твердого тела нарушаются условия предельного состояния» и «Напряженное состояние соответствует состоянию обобщенного растяжения» имеются в рабочем множестве, а целью системы является ответ на вопрос пользователя: «Необходимо решить вопрос об эксплуатационной пригодности объекта?»

При прямом выводе работа системы будет протекать следующим образом:

Шаг 1. Рассматривается правило 1. Его условие истинно, так как оба элемента конъюнкции имеются в рабочем множестве. Применяем правило 1, добавляем к рабочему множеству факт «В данной области образуется система трещин».

Шаг 2. Рассматривается правило 2. Его условие истинно, т.к. утверждение из условия имеется в рабочем множестве. Применяем правило 2, добавляем к рабочему множеству факт «Необходимо решить вопрос об эксплуатационной пригодности объекта». Целевое утверждение выведено.

Обратный порядок вывода: заключения просматриваются до тех пор, пока не будут обнаружены в рабочей памяти или получены от пользователя факты, подтверждающие одно из них. Таким образом, *в системах с обратным выводом* вначале выдвигается некоторая гипотеза, а затем механизм вывода в процессе работы как бы возвращается назад, переходя от нее к фактам и пытается найти среди них те, которые подтверждают эту гипотезу. Если она оказывается правильной, то выбирается следующая гипотеза, детализирующая первую и являющаяся по отношению к ней подцелью. После этого отыскиваются факты, подтверждающие истинность подчиненной гипотезы.

Вывод такого типа называется **управляемым целями**. Обратный поиск применяется в тех случаях, когда цели известны и их сравнительно немного.

В системах с прямым выводом по известным фактам отыскивается заключение, которое из этих фактов следует. Поэтому прямой вывод называют выводом, **управляемым данными**.

Вывод, **основанный на сочетании обратного и ограниченного прямого** называется **циклическим**.

В рассматриваемом примере вывод целевого утверждения «Необходимо решить вопрос об эксплуатационной пригодности объекта» обратной цепочкой рассуждений выполняется следующим образом:

Шаг 1. Рассматривается правило 1. Оно не содержит цели в правой части. Переходим к правилу 2.

Шаг 2. Рассматривается правило 2. Оно содержит цель в правой части правила. Переходим к правой части правила и рассматриваем в качестве текущей цели утверждения «В данной области образуется система трещин».

Шаг 3. Текущей цели нет в рабочем множестве. Рассмотрим правило 1, которое содержит цель в правой части. Обе компоненты его условия имеются в рабочем множестве, так что условие истинно. Применяем правило 1, в результате выводим утверждение «В данной области образуется система трещин», которое было нашей предыдущей целью.

Шаг 4. Применяем правило 2, условием которого является данное утверждение. Получаем вывод исходного утверждения.

Заметим, что для упрощения ситуации мы предположили, что в обоих случаях факты «В исследуемой области твердого тела нарушаются условия предельного состояния» и «Напряженное состояние соответствует состоянию обобщенного растяжения» уже известны системе. На самом деле система выясняет истинность или ложность факта, входящего в условие некоторого правила, спрашивая об этом пользователя в тот момент, когда она пытается применить правило.

Приведенный пример сознательно выбран очень простым и не отражающим многих проблем, связанных с организацией вывода в экспертной системе. В частности, из примера может создаться впечатление, что прямая цепочка рассуждений эффективнее, чем обратная, что на самом деле, вообще говоря, не так. Эффективность той или иной стратегии вывода зависит от характера задачи и содержимого базы знаний. В системах диагностики чаще применяется прямой вывод, в то время как в планирующих системах более эффективным оказывается обратный вывод. В некоторых системах вывод основывается на сочетании обратного и ограниченно-прямого. Такой комбинированный метод получил название циклического.

Выше уже отмечалось, что механизм вывода включает в себя два компонента – один из них реализует собственно вывод, другой управляет этим процессом. Компонент вывода выполняет первую задачу, рассматривая имеющиеся правила и факты из рабочего множества и добавляя в него новые факты при срабатывании какого-нибудь правила. Управляющий компонент определяет порядок применения правил. Рассмотрим каждый из этих компонентов более подробно.

Как уже отмечалось, механизм вывода включает в себя **два компонента - реализация собственно вывода и управление этим процессом**. Рассмотрим каждый из этих компонентов.

Компонент вывода. Его действие основано на использовании правила

вывода, суть которого можно сформулировать как: если истинно (задано) утверждение A_1, A_2, \dots и существует правило вида «ЕСЛИ A_i , ТО B_j », тогда утверждение B_1, B_2, \dots также истинно (выбирается).

Важным является наличие способности функционирования компонент вывода в условиях недостатка (неточности) информации, т.е. механизм вывода должен обладать возможностью продолжить рассуждения, определенные первоначально пользователем, и со временем найти решение даже при недостатке информации, т.е. система ни в коем случае не должна останавливаться из-за отсутствия какой-либо части входной информации (решение может и не быть точным).

Управляющий компонент определяет порядок применения правил и возможность изменения фактов (документов), если консультации продолжаются.

Управляющий компонент выполняет 4 функции:

1. *Сопоставление* – образец правила сопоставляется с имеющимися фактами.

2. *Выбор* – если в конкретной ситуации могут быть использованы сразу несколько правил, то из них выбирается одно-два, наиболее подходящее по заданному критерию (разрешение конфликта).

3. *Срабатывание* – если образец правила при сопоставлении совпал с какими-либо фактами из рабочей памяти, то правило срабатывает.

4. *Действие* – рабочая память подвергается изменению путем добавления в нее заключения сработавшего правила. Или же, если в правой части правила содержится указание на какое-либо действие, то оно выполняется. Интерпретатор правил работает циклически, просматривая в каждом цикле все правила, выявляя среди них те, посылки которых совпадают с заданными на данный момент фактами.

В одном цикле может сработать только одно правило. После выбора правило срабатывает, его заключение заносится в рабочую память и цикл повторяется сначала. Если несколько правил успешно сопоставлены с фактами, то интерпретатор производит выбор по определенному критерию (например, введение дополнительного факта) единственного правила, которое и срабатывает в этом цикле. Цикл работы интерпретатора можно представить в виде схемы, приведенной на рисунке 7.7.

Информация из рабочей памяти последовательно сопоставляется с посылками правил для выявления успешного сопоставления. Вся совокупность отобранных правил составляет *конфликтное множество*. Для разрешения конфликта интерпретатор имеет критерий, с помощью которого возможен выбор единственного правила, которое и срабатывает.

Управляющая процедура системы БЗД является **недетерминированной**, что означает, что траектория поиска решения в

пространстве состояний полностью определяется данными.

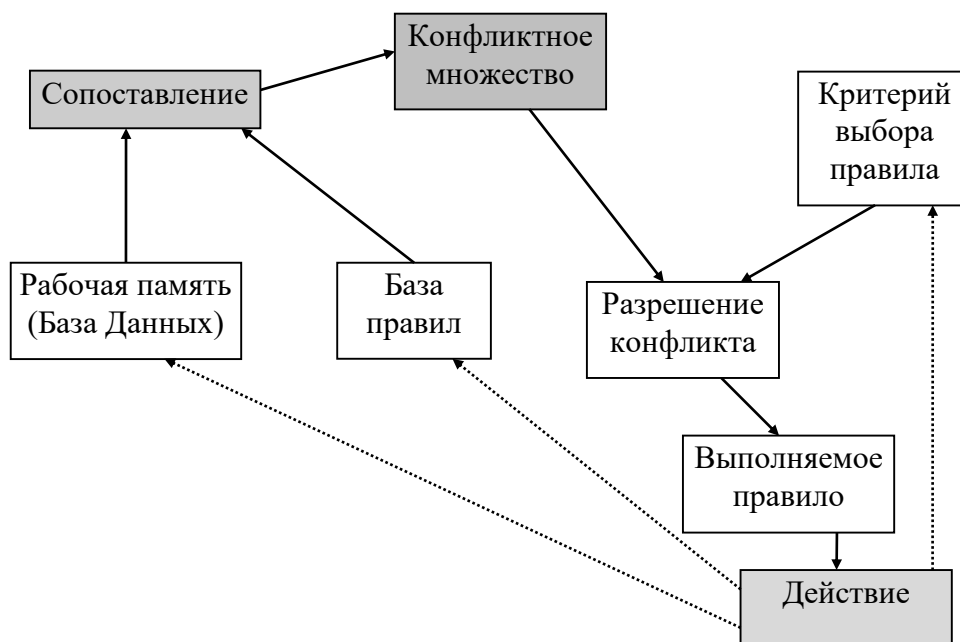


Рис. 7.7. Цикл работы интерпретатора

Стратегия управления выводом

Выбор метода поиска, т.е. стратегии вывода, представляет собой один из важных вопросов, возникающих при проектировании управляющей компоненты системы. От выбранного метода поиска существенным образом зависит порядок применения и срабатывания правил.

При разработке стратегии управления выводом необходимо руководствоваться ответом на следующие вопросы:

1. Какую точку в пространстве состояний принять в качестве исходной?

От выбора этой точки зависит и метод осуществления поиска – в прямом или обратном направлении.

2. Как повысить эффективность поиска решения?

Для повышения эффективности поиска решения, необходимо найти эвристики разрешения конфликтов, связанных с существованием нескольких возможных путей для продолжения поиска в пространстве состояний, поскольку требуется отбросить те из них, которые заведомо не ведут к искомому решению.

Повышение эффективности поиска

Система БЗД насчитывает сотни правил и документов, поэтому желательно использовать какую-либо стратегию управления выводом, позволяющей минимизировать время поиска решения. К таким стратегиям

относятся **поиск в глубину, поиск в ширину, разбиение на подзадачи, альфа-бета алгоритм.**

Суть **поиска в глубину** состоит в том, что при выборе очередной подцели в пространстве состояний предпочтение всегда, когда это возможно, отдается той, которая соответствует следующему, более детальному уровню описания задачи.

При **поиске в ширину** система вначале анализирует все состояния, находящиеся на одном уровне пространства состояний и лишь затем переходит к следующему уровню детальности.

Поиск в ширину не ограничивает анализ какой-либо заранее очерченный круг признаков. Поиск в глубину позволяет собрать воедино все признаки, связанные с выдвинутой гипотезой.

При **стратегии «разбиение на подзадачи»** в исходной задаче выделяются подзадачи, решение которых рассматривается как достижение промежуточных целей на пути к конечной цели. Такая стратегия является успешной, если задача хорошо структурирована. В этом случае задачу можно оптимально разбить на цепочку иерархически связанных целей-подцелей и добиться того, чтобы путь к ее решению в пространстве был минимальным.

Альфа-бета алгоритм состоит в том, что задача сводится к уменьшению пространства состояний путем удаления в нем ветвей, не перспективных для поиска успешного решения. Поэтому просматриваются только те вершины, в которые можно попасть в результате следующего шага, после чего неперспективные направления исключаются из дальнейшего рассмотрения.

7.4.4. Представление знаний, основанное на фреймах

Как отмечалось уже ранее, систему БЗД наиболее оптимально моделировать как производственную систему, т.к. схема функционирования последней совпадает по своей сути с формой процессов мышления человека при решении задач. Вместе с тем, так как в производственных системах базы знаний насчитывают сотни правил, то при такой сложности системы процесс обновления состава правил и контроль связей между ними становится весьма затруднительным. Добавляемые правила могут дублировать имеющиеся или вступать с ними в противоречие.

Альтернативным по отношению к производственным системам является **представление знаний, основанное на фреймах**. Такое представление дает возможность хранить родовидовую иерархию понятий в базе знаний в явной форме.

Фреймом называется структура для описания информации одного класса или стереотипной ситуации, состоящая из характеристик этого

класса или ситуаций и их значений. Характеристики называются слотами, а значения – заполнителями слотов.

Например, в СРГМ присутствует фреймовская структура «Указания по охране сооружений и природных объектов от вредного влияния подземных горных разработок в условиях Старобинского месторождения калийных солей», включающая в себя общий разветвленную сеть подфреймов: «Меры охраны зданий, сооружений и природных объектов, расположенных на земной поверхности», «Меры охраны подземных объектов и сооружений», «Руководство по прогнозированию ожидаемых сдвижений и деформаций земной поверхности» и т.д. Если в качестве фреймовской рассмотреть структуру «Математические модели поведения деформируемых твердых тел», то последняя включает в себя общий фрейм «Математические модели механики сплошных сред» и разветвленную сеть подфреймов «Модели теории упругости», «Модели ползучести деформируемых тел» «Модели пластичности» и т.д.

Совокупность фреймов, моделирующая какую-либо предметную область, представляет собой иерархическую структуру, в которой фреймы объединены с помощью родовидовых связей. На верхнем уровне иерархии находится фрейм, содержащий наиболее общую информацию, истинную для всех остальных фреймов. Фреймы должны обладать способностью наследовать значения характеристик «своих родителей», находящихся на более высоком уровне иерархии.

Замечание. Это вовсе не означает, что характеристики, передаваемые по умолчанию фреймам, находящимся ниже них в иерархии, не должны быть изменены внутри нижних фреймов. В случае, если последние содержат собственные значения данных характеристик, то в качестве истинных принимаются именно они. Это обстоятельство позволяет легко учитывать во фреймовых системах различного рода исключения.

Фреймы могут быть статические и динамические. В первом случае фреймы не могут быть изменены в процессе решения задачи, в системах второго типа это допустимо.

Обычно каждый фрейм соответствует некоторому объекту предметной области, а слоты содержат описывающие этот объект данные, т.е. в слотах находятся значения признаков объекта. Помимо конкретного значения в слоте могут храниться процедуры и правила, которые вызываются при необходимости вычисления\нахождения этого значения. Такие процедуры и правила активизируются только по мере надобности. Процедуры, располагающиеся в слоте, называются связанными процедурами.

Например, пусть имеется фрейм, связанный с различными типами разрушения твердых деформируемых тел, содержащий слот «разрушение материалов в условиях сложного объемного нагружения», в котором в свою

очередь находятся ссылки на законы, процедуры и критерии, детализирующие конкретные ситуации.

7.4.5. Представлений знаний в виде семантической сети

Семантические сети представляют собой наиболее общий способ представления знаний.

Семантическая сеть отображает совокупность объектов предметной области и отношений между ними. Объектам соответствуют узлы (или вершины) сети, а отношениями между объектами – соединяющие их дуги. Так, например, семантическую сеть «Проект на отработку участка шахтного поля» можно представить в таком виде (рис. 7.8).

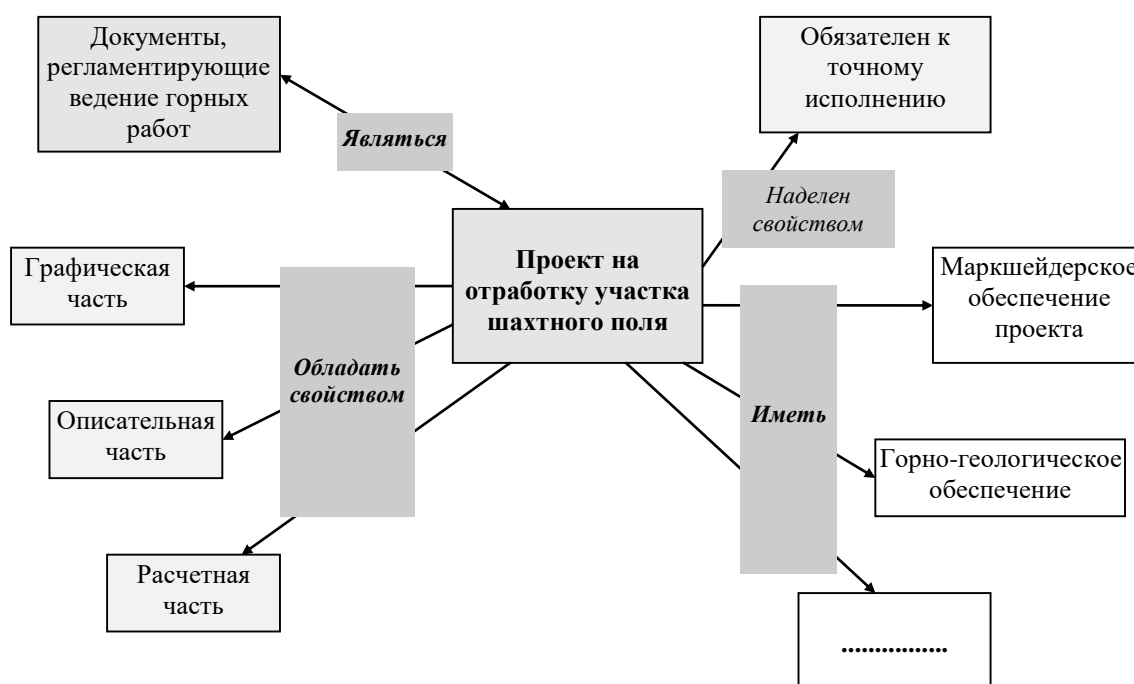


Рис. 7.8. Пример семантической сети «Проект на отработку участка шахтного поля»

В конкретную семантическую сеть включаются только те объекты (документы), которые необходимы для полного описания запрашиваемого предмета изучения. В качестве объектов в рассматриваемом случае могут выступать обобщенные и частные понятия, документы, правила, методики, действия, отдельные свойства объектов.

Вершины семантической сети соединяются дугой, если соответствующие объекты предметной области находятся между собой в каких-либо отношениях.

Наиболее распространенными являются следующие типы отношений:

Быть элементом класса\подкласса (являться) – означает, что объект

входит в состав данного класса. Например, «Геологическое обеспечение проекта» является элементом «Проект на отработку участка шахтного поля».

Иметь – задает свойства объектов. Например, «Геологическое обеспечение проекта» имеет общее описание геологии данного участка и геологические колонки по данному участку.

Являться следствием – отражает причинно-следственные связи между объектами. Например, «Проведение дегазационного профилактического бурения» является следствием «Ведение горных работ на 3 горизонте 4 шахтного поля».

Иметь значение – задает значения свойств объектов. Например, документ «Таблица критических деформаций земной поверхности» имеет несколько значений в зависимости от горно-геологических и других условий.

Основной недостаток семантических сетей и представлений в виде фреймов – сложность обработки исключений.

7.5. Использование аппарата математической логики при построении экспертных систем

Построение экспертной системы осуществляется на основе определенных правил и положений, т.е. синтаксиса. Кроме того, необходимо иметь в наличии средства для выражения связей между объектами. Для выполнения указанных требований удобно ***использовать средства математической логики: логику высказываний и исчисление предикатов.***

Напомним некоторые определения.

Предикатом называется функция, принимающая только два значения – ***истина*** и ***ложь*** и предназначенная для выражения свойств объектов или связей между ними.

Выражение, в котором утверждается или отрицается наличие каких-либо свойств у объекта называется **высказыванием**. Высказывание логически следует из заданных посылок, если оно истинно всегда, когда истинны посылки.

Правила вывода называются **непротиворечивыми**, если их заключения логически следуют из посылок.

Совокупность правил вывода является **полной**, если она позволяет вывести все высказывания, логически следующие из посылок.

Любому высказыванию можно приписать значение ***истинно*** или ***ложно***. В свою очередь отдельные высказывания могут соединяться булевыми операторами **И, ИЛИ, НЕ**.

Основу исчисления высказываний составляют правила образования сложных высказываний из атомарных (единичных). Сложные правильно построенные формулы (высказывания) формируются с помощью логических связок и кванторов (логических операций) (табл. 7.6).

Таблица 7.6. Операторы и кванторы исчисления предикатов

Логические связки	\wedge	И
	\vee	ИЛИ
	\neg	НЕ
	\rightarrow	ЛОГИЧЕСКИ СЛЕДУЕТ
	\equiv	ЭКВИВАЛЕНТНО
Кванторы	\exists	СУЩЕСТВУЕТ
	\forall	ДЛЯ ВСЕХ

База знаний считается *полной*, если любое высказывание, истинность которого вытекает из содержащихся в ней сведений, может быть получено с помощью логического вывода.

Для получения правильно построенных формул необходимо иметь базу знаний и *правила вывода*.

«Стандартное» правило вывода заключается в следующем: пусть известно, что истинно утверждение A и существует правило вида «Если A , то B », тогда утверждение B так же истинно. Правила срабатывают, когда находятся факты, удовлетворяющие их левой части: если истинен посыл, то должно быть истинно и заключение.

На первый взгляд кажется, что такой вывод легко может быть реализован, тем не менее на практике не всегда получается сформулировать правильно вывод.

Одна из причин такой ситуации заключается в том, что одно слово может иметь множество значений. Так, например, для словосочетания «data mining» при передаче его значения на русском языке наиболее часто используется сегодня словосочетание «интеллектуальный анализ данных». Более точным является словосочетание «обнаружение знаний в базах данных» (knowledge discovery in databases, KDD). Одним из значений «data mining» является «добыча данных». А если переводить выражение пословно, то слово «mining» в первую очередь относится к добыче полезных ископаемых и означает «добычу». Поэтому могут возникнуть вопросы относительно значения данного словосочетания.

В качестве следующего примера можно взять выражение «Сдвиг в горном массиве». Для словосочетания «горный массив» может быть два значения: «массив гор» и «породная толща». То есть, например, «горный массив в средней Азии» и «породная толща в области добычи полезных ископаемых».

Понять факты становится еще сложнее, если они являются составными частями более сложных фактов, положений.

В качестве подтверждения можно привести такие простые с точки зрения человеческой логики примеры.

Пример 1. «ЕСЛИ выпускник механико-математического факультета легко может вычислить производную полиномиальной функции n -го порядка»

И

«Петр Иванов – выпускник механико-математического факультета БГУ, ТО Петр Иванов легко вычислит производную полиномиальной функции третьего порядка».

Пример 2. «ЕСЛИ автомобиль белого цвета легко заметить ночью»

И

«Автомобиль Никиты белый, ТО Автомобиль Никиты легко заметить ночью».

Эти заключения легко выведет любой человек, но они оказываются очень трудными для современных экспертных систем.

Компонент вывода должен обладать способностью функционировать при любых условиях. Механизм вывода должен быть способен продолжить рассуждение и со временем найти решение даже при недостатке информации. Это решение может и не быть точным, однако система ни в коем случае не должна останавливаться из-за того, что отсутствует какая-либо часть входной информации.

Управляющий компонент

Этот компонент определяет порядок применения правил, а также устанавливает, имеются ли еще факты, которые могут быть изменены в случае продолжения консультации. Управляющий компонент выполняет четыре функции:

1. *Сопоставление* – образец правила сопоставляется с имеющимися фактами;

2. *Выбор* – если в конкретной ситуации могут быть применены сразу несколько правил, то из них выбирается одно, наиболее подходящее к заданному критерию (разрешение конфликта).

3. *Срабатывание* – если образец правила при сопоставлении совпал с какими-либо фактами из рабочего множества, то правило срабатывает.

4. *Действие* – рабочее множество подвергается изменению путем добавления в него заключения сработавшего правила. Если в правой части правила содержится указание на какое-либо действие, то оно выполняется (как, например, в системах обеспечения безопасности информации).

Интерпретатор правил работает циклически. В каждом цикле он просматривает все правила, чтобы выявить среди них те посылки, которые

совпадают с известными на данный момент фактами из рабочего множества. Интерпретатор определяет также порядок применения правил. После выбора правило срабатывает, его заключение заносится в рабочее множество, и затем цикл повторяется сначала.

В одном цикле может сработать только одно правило. Если несколько правил успешно сопоставлены с фактами, то интерпретатор производит выбор по определенному критерию единственного правила, которое и срабатывает в данном цикле.

Схематически цикл работы интерпретатора выглядит следующим образом.

Информация из рабочего множества последовательно сопоставляется с посылками правил для выявления успешного сопоставления. Совокупность отобранных правил составляет *конфликтное множество*. Для разрешения конфликта интерпретатор имеет критерий, с помощью которого он выбирает единственное правило, после чего оно срабатывает. Это выражается в занесении фактов, образующих заключение правила, в рабочее множество или в изменении критерия выбора конфликтующих правил. Если же в заключение правила входит название какого-нибудь действия, то оно выполняется (например, подается звуковой сигнал, начинает выполняться процедура и т.д.).

Новые данные, введенные в систему сработавшим правилом, в свою очередь могут изменить критерий выбора правила. В том случае, если, например, компьютерная система, предназначенная для игры в шахматы, разыгрывает партию за двух игроков, то она может принять решение придерживаться атакующей стратегии через ход, т.е. атаковать будет один из партнеров. Если вы сами играете с этой системой, то в какой-то момент она может перейти к использованию оборонительной стратегии (по крайней мере временно), а затем опять вернуться к наступательной игре. Изменение критерия основывается на заключениях, полученных после анализа положения на доске, которое представлено в рабочем множестве системы, а также правил игры (статических структурных знаний) и структурных динамических знаниях (эвристиках).

В действительности ЭС не располагают процедурами, которые могли бы построить в пространстве состояний сразу весь путь решения задачи. Более того, зачастую даже не удается определить, имеется ли вообще какое-нибудь решение задачи. Тем ни менее поиск решения выполняется, поскольку движением в пространстве состояний управляют скрытые или виртуальные процедуры.

Среди разработчиков ЭС они получили название демонов, поскольку во время работы системы находятся в «засаде» и активизируются только тогда, когда «их просят о помощи», т.е. на самом деле ведут себя как добрые

демоны [7, 51]. Свое название демоны получили от «демона Максвелла» – действующего лица одного из мысленных экспериментов, предложенного его автором для критики законов термодинамики [52]. Другим их прообразом является Пандемониум Оливера Селфриджа – первой модели человека, в котором деятельность биологической системы представлялась как работа вызываемых по образцу демонов [53, 54]. Если же воспользоваться научной терминологией, то такие управляющие процедуры получили название недетерминированных. Это означает, что траектория поиска решения в пространстве состояний полностью определяется данными.

При разработке управляющего компонента механизма (подсистемы) вывода необходимо решить вопрос о том, по какому критерию следует выбирать правило, которое будет применено в конкретном цикле.

Уже на ранней стадии разработки ЭС необходимо знать, что будет вводить конечный пользователь. Это нужно для того, чтобы убедиться, будет ли система достаточно практична и сможет ли она вжиться в среду, в которой ей предстоит работать.

Участие пользователя выражается в следующем:

- *Конкретные задачи*

Пользователь, сталкиваясь с конкретными проблемами, может объяснить возникновение проблем и предложить возможные варианты их решения.

- *Общение*

Интерфейс пользователя должен соответствовать словарю пользователя и уровню его подготовки.

- *Установление связей*

Знакомство пользователя с причинами и последствиями, вызывающими то или иное действие в процессе функционирования системы, неопределимо в определении взаимосвязей фактов в базе знаний.

- *Обратная связь*

Отличительной особенностью удобной в использовании ЭС является ее способность объяснить конечному пользователю ход своих рассуждений.

7.6. Стратегии управления выводом

7.6.1. Разработка стратегии

Одним из важных вопросов, возникающих при проектировании управляющей компоненты экспертных систем, основанных на знаниях, является *выбор метода поиска решения*, т.е. *стратегии вывода*. От выбранного метода поиска будет зависеть порядок применения и срабатывания правил. Процедура выбора сводится к определению

направления поиска и способа его осуществления. Процедуры, реализующие поиск, обычно «защиты» в механизм вывода, поэтому в большинстве систем инженеры знаний не имеют к ним доступа и, следовательно, не могут в них ничего изменять по своему желанию.

При разработке стратегии управления выводом необходимо ответить на два вопроса:

1. *Какую точку в пространстве состояний принять в качестве исходной?*

Еще до начала поиска решения система, основанная на знаниях, должна каким-то образом выбрать исходную точку поиска – в прямом или обратном направлении.

2. *Как повысить эффективность поиска решения?*

Чтобы добиться повышения эффективности поиска решения, необходимо найти эвристики разрешения конфликтов, связанных с существованием нескольких возможных путей для продолжения поиска в пространстве состояний, поскольку требуется отбросить те из них, которые заведомо не ведут к искомому решению.

7.6.2. Повышение эффективности поиска

В системах, Базы Знаний которых насчитывают сотни правил, весьма желательным является использование какой-либо стратегии управления выводом, позволяющей минимизировать время поиска решения и тем самым повысить эффективность вывода. К числу таких стратегий относятся поиск в глубину, поиск в ширину, разбиение на подзадачи и альфа- бета алгоритм.

а) Сопоставление методов поиска в глубину и ширину.

Суть поиска в глубину состоит в том, что при выборе очередной подцели в пространстве состояний предпочтение всегда, когда это возможно, отдается той, которая соответствует следующему, более детальному уровню описания задачи.

При поиске в ширину, напротив, система проанализирует все признаки, находящиеся на одном уровне пространства состояний, и лишь затем перейдет к признакам следующего уровня детальности.

Специалисты в какой-либо узкой области выше оценивают поиск в глубину, поскольку он позволяет собрать воедино все признаки, связанные с выдвинутой гипотезой. Универсалы же отдают предпочтение поиску в ширину, так как в этом случае анализ не ограничивается заранее очерченным кругом признаков. Особенности пространства поиска во многом определяют целесообразность применения той или иной стратегии. Так, например, программы для игры в шахматы строятся на основе поиска

в ширину, поскольку при использовании поиска в глубину число анализируемых ходов может быть очень большим.

б) Альфа-бета алгоритм.

Задача сводится к уменьшению пространства состояний путем удаления в нем ветвей, не перспективных для поиска успешного решения. Поэтому просматриваются только те вершины, в которые можно попасть в результате следующего шага, после чего неперспективные направления исключаются из дальнейшего рассмотрения. Например, если цвет предмета, который мы ищем, не красный, то его бессмысленно искать среди красных предметов.

Альфа-бета алгоритм нашел широкое применение в основном в системах, ориентированных на различные игры, например, в шахматных программах.

в) Разбиение на подзадачи

При такой стратегии в исходной задаче выделяются подзадачи, решение которых рассматривается как достижение промежуточных целей на пути к конечной цели. Если удастся правильно понять сущность задачи и оптимально разбить ее на систему иерархически связанных целей-подцелей, то можно добиться того, что путь к ее решению в пространстве поиска будет минимален. Однако если задача является плохо структурированной, то сделать это невозможно.

При сведении задачи к подзадачам производится исследование исходной задачи с целью выделения такого множества подзадач, чтобы решение некоторого определенного подмножества этих подзадач содержало в себе решение исходной задачи.

Рассмотрим, *например*, задачу о выборе критерия пластичности при деформировании под нагрузкой твердого деформируемого тела.

Эта задача может быть сведена, например, к следующей последовательности подзадач:

Подзадача 1. Что такое деформируемое твердое тело.

Подзадача 2. Что такое пластичность деформируемого твердого тела.

Подзадача 3. Какие критерии пластичности являются наиболее распространенными.

Подзадача 4. Как выбирать наиболее «подходящий» критерий пластичности.

Подзадача 5. Выбор критерия пластичности в рассматриваемой модельной задаче.

В данном случае решение всех пяти сформулированных подзадач обеспечивает некоторое решение первоначально сформулированной задачи.

В свою очередь, каждая из подзадач может быть решена с применением какого-либо метода. К ним могут быть применены методы, использующие

пространство состояний, или же их можно проанализировать с целью выделения для каждой своих подзадач и т.д. Если продолжить процесс разбиения возникающих подзадач на еще более мелкие, то в конце концов мы приходим к некоторым элементарным задачам, решение которых может считаться тривиальным.

На каждом из этапов может возникнуть несколько альтернативных множеств подзадач, к которым может быть сведена данная задача. Т.к. некоторые из этих множеств в конечном итоге, возможно, не приведут к окончательному решению задачи, то, как правило, для решения первоначальной задачи необходим поиск в пространстве множеств подзадач.

г) Использование формальной логики при решении задач.

Часто для решения задач либо требуется проведение логического анализа в определенном объеме, либо поиск решения существенно отличается после такого анализа. Иногда такой анализ показывает, что определенные проблемы неразрешимы, то есть может иметь место такая ситуация, что из выбранного начального состояния «попасть» в итоговое не представляется возможным. *Например*, в игре в пятнадцать можно доказать, что целевая конфигурация (2) не может быть получена из начальной конфигурации (1).

1				2			
1	2	3	4	15	14	13	12
5	6	7	8	11	10	9	8
9	10	11	12	7	6	5	4
13	14	15		3	2	15	

7.7. Диалог с ЭС. Объяснение

Для того чтобы с ЭС мог взаимодействовать любой неподготовленный пользователь, необходимо наличие в ней средств общения на естественном языке. В этом случае весьма важным представляется разработка как можно более дружественного для пользователя интерфейса на естественном языке. При этом, необходимо учитывать, что все допустимые входные сообщения пользователя ограничены набором понятий, содержащихся в Базе Знаний.

ЗАКЛЮЧЕНИЕ

Технологии ИИ все более активно становятся полноправными «соавторами» алгоритмов механико-математического и компьютерного моделирования разнообразных физических и технологических процессов и явлений. Поэтому сформулированное требование к интеллектуальному компьютерному моделированию на базе математического моделирования: компьютерные технологии механико-математического моделирования это: высокопроизводительные вычисления + знания и «большие данные» + математические модели + искусственный интеллект, является корректным и может быть принято в качестве базового.

С учетом потенциальных возможностей вычислительных технологий, сегодня представляется реальным *разработка «интеллектуальных» систем компьютерного моделирования*, что подразумевает «умение» таких систем выдавать рекомендации к постановке модельных задач, к корректировке вычислительных алгоритмов, выполнять интеллектуальную обработку и интерпретацию результатов вычислений.

Каждая из затронутых в пособии тем является весьма обширной и представляет собой тематику отдельного специального курса.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1. Рассел С., Норвиг П. Искусственный интеллект: современный подход. 2-е издание. Изд-во «Вильямс». 2016. 1408 с.

2. McCarthy, J. (1968). Programs with Common Sense. In: Minsky, M. Semantic Information Processing. Cambridge, MA: The MIT Press, pp. 403-418. Available at: <https://www.cs.cornell.edu/selman/cs672/readings/mccarthy-upd.pdf>

3. Семенов А.Л. ИИ: Почему математика? // Доклады Российской академии наук. Математика, информатика, процессы управления». 2022, том 508, с. 3–5. DOI:10.31857/S2686954322070268.

4. Лёвин Б.А., Пискунов А.А., Поляков В.Ю., Савин А.В. Искусственный интеллект в инженерном образовании // Высшее образование в России. 2022, том. 31, №.7, стр. 79-95.

5. Абламейко С.В., Журавков М.А. Технологии искусственного интеллекта: текущее состояние и направления развития // Наука и инновации. №2 (240). 2023. С. 56-61. <https://innosfera.belnauka.by>

6. Бурнаев Е.В. и др. Фундаментальные исследования и разработки в области прикладного искусственного интеллекта / Е.В. Бурнаев, А.В. Бернштейн, В.В. Вановский, А.А. Зайцев, А.М. Булкин, В.Ю. Игнатъев, Д.Г. Шадрин, С.В. Илларионова, И.В. Оселедец, А.Ю. Михалев, А.А. Осипцов, А.А. Артемов, М.Г. Шараев, И.Е. Трофимов // Доклады Российской Академии Наук. Математика, Информатика, Процессы управления, 2022, том 508, с. 19–27.

7. Гаврилова Т.А., Кудрявцев Д.В., Муромцев Д.И. Инженерия знаний. Модели и методы. Издательство "Лань". 2016. 324 с. ISBN 978-5-8114-2128-2.

8. Бруссард М. Искусственный интеллект: Пределы возможного. Пер. с англ. М.: Альпина, 2020. 362 с. ISBN: 978-5-00139-080-0.

9. Дубровский Д.И. Задача создания Общего искусственного интеллекта и проблема сознания // Философские науки. 2021. Т. 64. № 1. С. 13–44. DOI: 10.30727/0235-1188-2021-64-1-13-44.

10. Бухановский А.В. Интеллектуальные технологии цифровой трансформации промышленных производств // Доклады Российской Академии Наук. Математика, Информатика, Процессы управления, 2022, том 508, с. 33–40.

11. Ветров Д. Восемь вопросов к искусственному интеллекту // Национальный исследовательский университет «Высшая школа экономики». 2020. 23 декабря.

URL: <https://www.hse.ru/news/communication/428400557.html>

12. Абламейко С.В. Искусственный интеллект в Беларуси – состояние и развитие. Наука и инновации, №5 (231), Май 2022, с.36-40.
13. Турдаков Д.Ю. и др. Доверенный искусственный интеллект: вызовы и перспективные решения / Д.Ю. Турдаков, А.И. Аветисян, К.В. Архипенко, А.В. Анциферова, Д.С. Ватолин, С.С. Волков, А.В. Гасников, Д.А. Девяткин, М.Д. Дробышевский, А.П. Коваленко, М.И. Кривоносов, Н.В. Лукашевич, В.А. Малых, С.И. Николенко, И.В. Оселедец, А.И. Перминов, И.В. Соченков, М.М. Тихомиров, А.Н. Федотов, М.Ю. Хачай. // Доклады Российской Академии Наук. Математика, Информатика, Процессы управления, 2022, том 508, с. 13–18
14. Абламейко С., Недзведзь А., Богущ Р. Технологии искусственного интеллекта: компьютерное зрение // Наука и инновации. 2023;(3):48-58.
15. Лёвин Б.А., Пискунов А.А., Поляков В.Ю., Савин А.В. Искусственный интеллект в инженерном образовании // Высшее образование в России. 2022. Т. 31. № 7. С. 79–95. DOI: 10.31992/0869-3617-2022-31-7-79-95.
16. <https://play.google.com/store/apps/details?id=com.openai.chatgpt>
17. Ивахненко Е.Н., Никольский В.С. ChatGPT в высшем образовании и науке: угроза или ценный ресурс? // Высшее образование в России. 2023. Т. 32. № 4. С. 9–22. DOI: 10.31992/0869-3617-2023-32-4-9-22.
18. Как я написал диплом с помощью ChatGPT и оказался в центре спора о нейросетях в образовании. URL: <https://journal.tinkoff.ru/neurodiploma>.
19. Biswas, S.S. (2023). Potential Use of Chat GPT in Global Warming. Annals of Biomedical Engineering. DOI: 10.1007/s10439-023-03171-8
20. Абламейко С.В., Журавков М.А. Математика и математики БГУ и Беларуси: 100 лет развития / С. В. Абламейко, М. А. Журавков. – Минск: БГУ, 2021. – 255 с.: ISBN 978-985-881-020-7.
21. <https://utopya88.github.io/ML.html>.
22. <https://cig-rdlab.gitbook.io/ml/lectures/zanyatie-1>
23. W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994) 163-171.
24. Rout L., Korotin A., Burnaev E. Generative Modeling with Optimal Transport Maps. ICLR, 2022;
25. Asadulaev A., Korotin A., Egiazarian V., Burnaev E. Neural Optimal Transport with General Cost Functionals. arXiv:2205.15403, 2022.
26. Arnold D. et al. Uncertainty quantification in reservoir prediction: part 1 – model realism in history matching using geological prior definitions. Mathematical Geosciences. – 2019. – Vol. 51. – No. 2. – pp. 209–240.

27. Demyanov V. et al. Uncertainty quantification in reservoir prediction: part 2 – handling uncertainty in the geological scenario. *Mathematical Geosciences*. – 2019. – Vol. 51. – No. 2. – pp. 241–264.

28. Корнаев А.В., Корнаева Е.П., Стебаков И.С. Теоретические предпосылки физически обоснованного машинного обучения и его приложения к гидродинамике // Доклады РАН. Математика, Информатика, Процессы управления, 2022, том 508, С. 100–101.

29. Турдаков Д.Ю. и др. Доверенный искусственный интеллект: вызовы и перспективные решения / Д.Ю. Турдаков, А.И. Аветисян, К.В. Архипенко, А.В. Анциферова, Д.С. Ватолин, С.С. Волков, А.В. Гасников, Д.А. Девяткин, М.Д. Дробышевский, А.П. Коваленко, М.И. Кривонос, Н.В. Лукашевич, В.А. Малых, С.И. Николенко, И.В. Оселедец, А.И. Перминов, И.В. Соченков, М.М. Тихомиров, А.Н. Федотов, М.Ю. Хачай // Доклады Российской Академии Наук. Математика, Информатика, Процессы управления, 2022, том 508, С. 13–18.

30. Краткий обзор популярных нейронных сетей. Электронный ресурс. <https://habr.com/ru/post/83781>.

31. Головкин В.А., Краснопрошин В.В. Нейросетевые технологии обработки данных – Минск: БГУ, 2017. 263 с.

32. Созыкин А.В. Обзор методов обучения глубоких нейронных сетей // Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». 2017. Т.6, № 3. С. 28–59. DOI: 10.14529/cmse170303

33. Головкин В.А. Нейроинтеллект: теория и применение: в 2 кн. – Брест, 1999. – Кн.1: Организация и обучение нейронных сетей с прямыми и обратными связями.

34. Li, Wenjing & Li, Siyi & Lin, Zhiyong & Li, Qi. (2021). Information modeling of mine working based on BIM technology. *Tunnelling and Underground Space Technology*. 115. 103978. 10.1016/j.tust.2021.103978.

35. Brian Hickey, Dr Carine Gachon, Dr John Cosgrove. Digital Twin – A Tool for Project Management in Manufacturing. *Procedia Computer Science*. 217, 2023, Pages 720-727, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2022.12.268>.

36. Xin Liu, Du Jiang, Bo Tao, Feng Xiang, Guozhang Jiang, Ying Sun, Jianyi Kong, Gongfa Li. A systematic review of digital twin about physical entities, virtual models, twin data, and applications. *Advanced Engineering Informatics*. 55, 2023, 101876, ISSN 1474-0346, <https://doi.org/10.1016/j.aei.2023.101876>.

37. Mohsen Attaran, Bilge Gokhan Celik. Digital Twin: Benefits, use cases, challenges, and opportunities. *Decision Analytics Journal*. 6, 2023, 100165, ISSN 2772-6622, <https://doi.org/10.1016/j.dajour.2023.100165>.

38. Крюков К.М., Шаповалов А.В. Использование технологии цифровых двойников в строительстве // ИВД. 2022. №5 (89).
39. Анциферов С.И., Михайлов В.В., Сычев Е.А., Карачевцева А.В. Разработка цифрового двойника // Технологии аддитивного производства. Т.1. №3, 2019. С. 39–45.
40. Grieves, M. Digital Twin: Manufacturing Excellence through Virtual Factory Replication; A White Paper; Michael Grieves, LLC: Melbourne, FL, USA, 2014.
- 41.РБК:<https://trends.rbc.ru/trends/industry/6107e5339a79478125166eeb>
42. Журавков М.А. Корпоративные автоматизированные системы сопряженного геомониторинга // Наука и инновации. Научно-практический журнал. №4 (74). 2009. С.51 – 54.
43. Журавков М.А. Роль и место корпоративных автоматизированных систем сопряженного геомониторинга в основных технологических процессах современного горнодобывающего предприятия // Рудник будущего. Научно-технический журнал. №3. 2010. С.108 – 112.
44. Хитрых Д. Цифровой двойник: концепция, уровни, связь с Интернетом вещей и роль численного и системного моделирования // САПР и графика. №7. 2020.
45. Журавков М.А., Николайчик М.А. Построение цифровых двойников элементов рудника // Горная механика и машиностроение. 2023. №2. С. 5-14.
46. Нейлор К. Как построить свою экспертную систему. М.: Энергоатомиздат, 1991.
47. Элти Д., Кумбс М. Экспертные системы: концепции и примеры. М.: Финансы и статистика, 1987.
48. Уотермент Д. Руководство по экспертным системам. М.: Мир, 1980.
49. Combining narratives, John McCarthy et al. (1998),
50. H. Levesque, F. Pirri, and R. Reiter (1998). Foundations for the situation calculus. Electronic Transactions on Artificial Intelligence, 2(3–4):159-178
51. <https://ru.wikipedia.org/wiki/Демон> (программа)
52. Бриллюэн Л. Наука и теория информации. М.: Физматлит, 1960. 495 с.
53. Оливер Л. «Пандемониум». М.: "Эксмо", 2017. 400 с. ISBN: 978-5-699-56879-6.
54. Фомин Я. А. Распознавание образов: теория и применения. 2-е изд. – М.: ФАЗИС, 2012. – 429 с. ISBN 978-5-7036-0130-4.

ОГЛАВЛЕНИЕ

	ВВЕДЕНИЕ	3
1.	ОБЩИЕ ПОНЯТИЯ, ОПРЕДЕЛЕНИЯ	6
1.1	Происхождение термина «искусственный интеллект». Дартмутский семинар	6
1.2	Что такое ИИ?	7
1.3	Основные направления исследований в области ИИ в настоящее время	10
1.3.1	<i>Символьное моделирование мыслительных процессов</i>	10
1.3.2	<i>Работа с естественными языками</i>	11
1.3.3	<i>Представление, обработка и использование знаний</i>	12
1.3.4	<i>Машинное обучение</i>	13
1.3.5	<i>«Биологическое моделирование» в системах ИИ</i>	14
1.3.6	<i>Интеллектуальная робототехника</i>	15
1.3.7	<i>Машинное творчество</i>	16
1.4	Современный ИИ. Эра «сверхразума» – общий ИИ	16
2.	ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ И ИНЖЕНЕРНЫХ РАСЧЕТОВ	21
2.1	Общие сведения	21
2.2	Разработка прикладных систем на базе CAD&CAE-технологий	24
2.3	Анализ видеoinформации («Компьютерное зрение»)	26
2.4	Интеллектуальные системы компьютерного моделирования	27
2.5	ИИ и инженерное дело	28
3.	ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ОБРАЗОВАНИИ	31
3.1	Общие сведения	31
3.2	О подготовке кадров в области ИИ	36

3.3	Некоторые выводы	37
4.	МАШИННОЕ ОБУЧЕНИЕ	39
4.1	Введение	39
4.2	О понятии машинного обучения	39
4.3	Как связаны машинное обучение, нейросети и глубокое обучение	41
4.4	Три составляющие МО	42
4.5	Машинное обучение и «традиционное» программирование	43
4.6	Основные виды (типы) машинного обучения	47
4.6.1	<i>Вид 1. Классическое обучение</i>	47
4.6.2	<i>Вид 2. Обучение с подкреплением</i>	56
4.6.3	<i>Вид 3. Ансамбли</i>	59
4.6.4	<i>Вид 4. Нейросети и глубокое обучение</i>	61
4.7	Некоторые общие правила МО	67
4.8	Предварительная обработка данных	71
4.9	Общие проблемы машинного обучения	75
4.10	Перспективы машинного обучения: не начнет ли ИИ думать за человека?	75
4.11	Примеры использования методов и технологий МО	76
5.	МЕТОДЫ НЕЙРОННЫХ СЕТЕЙ	82
5.1	Общие положения	82
5.2	Кратко из истории появления и развития ИНС	83
5.3	Архитектура (геометрия) нейросетей	86
5.4	Обучение и настройка нейронных сетей	92
5.5	Функции-активаторы	94
5.6	Использование методов ИНС для решения задач моделирования механических процессов	97
5.7	Пример построения простой нейросети	100
5.8	Библиотеки для работы с нейросетями в Python	103

6.	ЦИФРОВЫЕ ДВОЙНИКИ	106
6.1	Цифровые двойники современного предприятия	106
6.2	Этапы создания ЦД предприятия	113
6.3	От пилотного проекта до масштабирования технологии ЦД	114
7	ЭКСПЕРТНЫЕ СИСТЕМЫ. ИНЖЕНЕРИЯ ЗНАНИЙ	116
7.1	Общие понятия и определения	116
7.2	Экспертные системы	117
7.3	Инженерия Знаний	132
7.3.1	<i>Методология построения системы «Базы Знаний»</i>	133
7.3.2	<i>Построение Баз Знаний на основе сведений, полученных от эксперта</i>	134
7.3.3	<i>Требования к Знаниям</i>	136
7.3.4	<i>Приобретение знаний. Конструирование системы</i>	140
7.3.5	<i>Формы и способы представления знаний</i>	142
7.3.6	<i>Организация хранения информации</i>	144
7.4	Принципы построения Баз Знаний	145
7.4.1	<i>Типы иерархической структуры для построения Баз Знаний</i>	145
7.4.2	<i>Этапы построения системы «Базы Знаний»</i>	148
7.4.3	<i>Построение системы Базы Знаний по принципу «Производственной системы»</i>	149
7.4.4	<i>Представление знаний, основанное на фреймах</i>	158
7.4.5	<i>Представлений знаний в виде семантической сети</i>	160
7.5	Использование аппарата математической логики при построении экспертных систем	161
7.6	Стратегии управления выводом	165
7.6.1	<i>Разработка стратегии</i>	165
7.6.2	<i>Повышение эффективности поиска</i>	166
7.7	Диалог с ЭС. Объяснение	168

	ЗАКЛЮЧЕНИЕ	169
	БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ	170