КАК СЧИТАТЬ Q-ФУНКЦИЮ

Д. И. Качков

Белорусский государственный университет, г. Минск;
fpm.kachkovDI@bsu.by;
науч. рук. – М. К. Буза, д-р техн. наук, проф.

В работе рассматривается Q-learning – один из алгоритмов обучения с подкреплением. Ключевым в алгоритме Q-learning является итеративное построение Q-функции, ставящей в соответствие каждой паре (состояние среды, действие) действительное число – долгосрочный выигрыш, который может быть получен агентом после совершения данного действия в данном состоянии. В статье предложен ряд механизмов, идей и подходов, позволяющий ускорить процесс построения Q-функции.

Ключевые слова: обучение с подкреплением; Q-обучение; Q-функция; оптимизация процесса обучения

ВВЕДЕНИЕ

Большинство алгоритмов машинного обучения ориентировано на автоматизацию и усовершенствование известных человеку стратегий. При разработке автоматического шахматного игрока используются обширные базы партий, дебютов и стратегий; при создании алгоритма распознавания изображений необходима обучающая выборка; алгоритмы классификации, хоть и относятся к обучению без учителя, требуют глубоких знаний о структуре сравниваемых объектов.

В случае, когда условия неизвестны либо изменяются непредсказуемым образом, подобный подход не применим. Возникает проблема разработки алгоритма, способного находить оптимальную стратегию действий с нуля, то есть в отсутствие каких бы то ни было исходных данных. Можно переформулировать эту задачу следующим образом: разработка искусственного игрока, способного найти эффективную стратегию действий в игре, правила которой наперёд неизвестны. «Игру» здесь следуют понимать в обобщённом смысле: управляемое взаимодействие со средой, ограниченное некоторыми законами, нацеленное на максимизацию заданного числового параметра.

Один из возможных подходов – представление игры в виде марковского процесса принятия решений. Для поиска успешных стратегий в рамках марковского процесса принятия решений можно использовать обучение с подкреплением. Среди представленных в этой области алгоритмов выберем Q-обучение. Идея заключается в итеративном построе-

нии Q-функции, заданной на пространстве пар (состояние, действие) и возвращающей ожидаемый долгосрочный выигрыш.

Данный подход действительно позволяет найти оптимальную стратегию, однако требует значительных затрат по времени и по памяти. Так, например, для шахмат существует оценка в 1043 возможных игровых позиций. Очевидно, современные технологии не позволяют рассчитать значения Q-функции для всевозможных состояний.

ЗАДАЧИ ИССЛЕДОВАНИЯ

На основании вышесказанного получаем необходимость разработать оптимизации алгоритма Q-обучения. Среди потенциальных направлений оптимизации можно выделить следующие:

- повышение точности оценок, получаемых в ходе одной итерации обучения;
- повышение интенсивности исследования наиболее частых и достижимых состояний за счёт меньшей интенсивности исследования редких и практически недостижимых состояний;
- разработка вспомогательных алгоритмов, позволяющих предсказывать значение Q-функции для тех аргументов, для которых она не была уточнена достаточное количество раз.

Соответственно, задача данного исследования состоит в рассмотрении потенциальных усовершенствований алгоритма Q-обучения в рамках обозначенных направлений.

ОПТИМИЗАЦИЯ ВЫЧИСЛЕНИЯ Q-ФУНКЦИИ

Данный раздел содержит ряд идей, который позволят ускорить процесс Q-обучения.

В базовом Q-обучении введён фиксированный параметр – фактор обучения, определяющий, насколько сильно агент доверяет новой информации.

Уместно сделать его изменяющимся. Очевидно, что исходное случайное значение Q-функции на некотором аргументе не является адекватным. Каждое итеративное обновление значения Q-функции на этапе обучения повышает его достоверность. Следовательно, каждый следующий прецедент в рамках итоговой оценки должен учитываться с меньшим весом, чем предыдущие. Тем не менее, не следует использовать обратную пропорциональность в чистом виде: очередное новое значение рассчитывается в условиях, когда Q-функция более точно оп-

ределена на всей области определения, и соответственно, является более достоверным, нежели все предыдущие.

Следует понимать, что если в рамках очередной итерации был получен крупный выигрыш, то с большой долей вероятности успешным было не только последнее действие, но последовательность предшествующих действий. Соответственно, имеет смысл обновлять не только Q-функцию для последнего совершённого действия, но для последовательности пар позиция-действие, пройденных накануне. Подобная идея используется в алгоритме, названном Q(λ)-обучение.

Пусть имеется возможность моделировать среду в ходе обучения. Например, в случае последовательных дискретных игр с полной информацией подразумевается возможность на этапе обучения выбирать не только свой ход, но и ход противника. В этом случае имеет смысл параллельно обучать игроков за обе роли. Благодаря постоянно возрастающему уровню противника уменьшится количество иррациональных ходов, совершаемых им, из-за которых ошибочные действия обучающегося игрока могут получать неверную высокую оценку.

Кроме того, в этом случае имеется возможность блуждать по дереву возможных сценариев и, соответственно, использовать более сложную формулу для пересчёта значения Q-функции, учитывающую все потенциальные пути развития игры.

Ещё одна идея заключается в обновлении на каждой итерации значения Q-функции не только для текущей пары позиция-действие, но и для всех похожих аргументов. Например, в случае игры в крестики-нолики можно пользоваться симметрией игры и на каждом этапе обновлять значение функции для всех пар позиция-действие, получаемых из текущей пары с помощью операций поворота и отражения. Тем не менее, такая стратегия требует качественного знания об условиях игры: понятия о «похожих» ситуациях.

Пусть текущее состояние и каждое допустимое действие считываются в виде набора осмысленных числовых параметров (значение которых, тем не менее, неизвестно). В этом после обучения можно выбрать набор наиболее достоверных значений Q-функции (то есть просчитанных наибольшее количество раз). Среди выбранных аргументов можно выбрать те, на которых получен низкий результат, и те, на которых высокий. Соответственно, используя эти аргументы в качестве тестовой выборки, можно запустить некоторый алгоритм классификации, который научится различать «сильные», «нейтральные» и «слабые» ходы. В частности, для создания подобного классифицирующего алгоритма можно случайно сгенерировать большое количество случайных линейных классификаторов и применить бустинг-алгоритмы.

Кроме того, числовое представление текущей позиции позволяет хранить Q-функцию не в виде таблицы, а, например, в виде свёрточной нейронной сети. Это позволит не только сэкономить память, но и предсказывать значение Q-функции для аргументов, которые не были затронуты в ходе обучения.

ПРАКТИЧЕСКИЕ РЕЗУЛЬТАТЫ

Некоторые из предложенных оптимизаций были проверены на практике. В качестве игры рассматривалась стандартная игра в крестики-нолики – небольшие размеры дерева вариантов игры позволяли определить, придерживается ли обученный игрок абсолютно оптимальной стратегии (то есть выигрывает, если соперник предоставляет такую возможность, и сводит партию к ничьёй в противном случае). Цикл обучения на N партиях повторялся 25 раз. Утверждалось, что выбранный алгоритм позволяет обучиться крестикам-ноликам за N партий, если все 25 раз обученный игрок действовал оптимально. Для базового Q-обучения требовалось более 100000 партий. Использование гибких факторов обучения понизило этот показатель до 80000 партий. После дополнительного внедрения сохранения историй алгоритм обучался крестикам-ноликам за 60000 партий. Алгоритм, который учитывал симметричность игры в крестики-нолики, обучился идеальной стратегии за 30000 партий.

ЗАКЛЮЧЕНИЕ

Таким образом, в настоящей работе предложен ряд подходов, которые позволяют ускорить процесс обучения с подкреплением. Некоторые из них были проверены на практике и продемонстрировали свою эффективность.