

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра дискретной математики и алгоритмики

ПОЛИЩУК Александр Михайлович

**ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ
ТОРГОВЛИ НА РЫНКЕ ФОРЕКС**

Магистерская диссертация
специальность 1-31 80 09 «Прикладная математика и информатика»

Научный руководитель
Марков Сергей Викторович,
кандидат физико-математических наук,
доцент

Допущена к защите

«___» _____ 2021 г.

Заведующий кафедрой дискретной математики и алгоритмики

Котов Владимир Михайлович

доктор физико-математических наук, профессор

Минск, 2021

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ	4
ВВЕДЕНИЕ	5
ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ.....	7
ГЛАВА 1 ОСНОВНЫЕ ПОНЯТИЯ И ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ	10
1.1 Основные положения.....	10
1.2 Выбор валютной пары и периода для наблюдения	12
1.3 Основные цели стратегии.....	13
1.4 Применение машинного обучения для торговли на Форексе	13
ГЛАВА 2 ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ.....	15
2.1 Описание метода	15
2.2 Основные компоненты	15
2.3 Глубокое обучение с подкреплением	18
2.4 Особенности обучения с подкреплением	20
2.5 Итоги	21
ГЛАВА 3 РАЗРАБОТКА СТРАТЕГИИ.....	22
3.1 Описание стратегии	22
3.2 Торговый агент.....	23
3.3 Модель нейронной сети.....	25
3.4 Дополнительные функции	26
ГЛАВА 4 ОБУЧЕНИЕ И ТЕСТИРОВАНИЕ СТРАТЕГИИ.....	28
4.1 Загрузка данных	28
4.2 Этапы обучения торговой стратегии	29
4.3 Выбор технических индикаторов.....	31
4.4 Обучение и валидация торговой стратегии.....	32
4.5 Тестирование торговой стратегии.....	36
ГЛАВА 5 ТОРГОВЛЯ НА ФОРЕКС.....	40
5.1 Открытие демо-счета.....	40

5.2 Использование метатрейдера.....	41
5.3 Оценка успешности стратегии.....	43
ЗАКЛЮЧЕНИЕ	44
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	45

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

1. CPU - central processing unit, центральный процессор
2. DQN – deep Q-learning, deep Q-network, глубокое обучение с подкреплением
3. GPU - graphics processing unit, графический процессор
4. MA – moving average, скользящее среднее
5. MOM – momentum indicator, индикатор импульса
6. ROC – price rate of change indicator, индикатор скорости изменения цены
7. ИНС – искусственная нейронная сеть

ВВЕДЕНИЕ

«Каждый наш день — это счет в банке, а деньги на нем — это наше время. Здесь нет бедных и богатых, у каждого есть по 24 часа».

Кристофер Райс

Рынок Форекс - совокупность учреждений, которые предоставляют возможность обменивать и торговать валютой [1]. Правильно используя инструменты для торговли, опыт и чутье, трейдер (человек, торгующий на рынке Форекс) может очень сильно улучшить свое материальное положение. Однако стоит понимать, что «хорошим» трейдером может стать далеко не каждый, так как Форекс – это игра, а в игре всегда есть победители и проигравшие. Человек, не имеющий четкой стратегии и не знающий тонкостей рынка, в скором времени осознает, что зря тратит свое время и средства. Поэтому разработка торговой стратегии – это очень важный и долгий процесс с большим количеством итераций подбора параметров, сменой подходов и алгоритмов и тестирования полученной в результате стратегии перед применением в реальной торговле.

Существует три главных анализа финансовых и торговых рынков:

1. Фундаментальный анализ;
2. Технический анализ;
3. Интуитивный подход к анализу.

Также последнее время во многих областях применяются методы машинного обучения, и торговля на Форекс не стала исключением: одни только ИНС обучают и используют как в техническом, так и в фундаментальном подходе при анализе рынка валют. Методы машинного обучения не заменяют оригинальные стратегии торговли, а используются в совокупности с традиционными подходами анализа и разработки стратегий для Форекса. Это обусловлено тем, что есть много алгоритмов, которые, используя какие-то параметры системы и индикаторы из технического анализа, могут предсказывать курсы обмена валют не хуже, а временами и лучше, чем подобные алгоритмы, основанные на более традиционных подходах.

Одним из подходов машинного обучения, который применяется в торговле, является обучение с подкреплением. Я решил выбрать данный способ предсказания курса валют, так как если стоит задача – максимизировать прибыль и минимизировать убытки, коей и является торговля на Форексе, то обучение с подкреплением отлично для того подходит. Помимо этого, очень

важно то, что можно применять данную концепцию и для того, чтобы распределять свои активы, то есть управлять портфелем инвестора.

Таким образом, дальше будет описан процесс изучения и разработки торговой стратегии на основе обучения с подкреплением.

Целью данной работы является анализ изменения курсов валют и построение и тестирование прибыльной торговой стратегии с применением методов машинного обучения для дальнейшего использования на Форексе.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Магистерская диссертация, 45 страниц, 14 рисунков, 4 таблицы, 7 источников.

КУРСЫ ВАЛЮТ, ГЛУБОКОЕ ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ, НЕЙРОННЫЕ СЕТИ, ОБУЧЕНИЕ МОДЕЛИ, ИНДИКАТОРЫ, ТРЕЙДЕР, ТРЕНД, СДЕЛКА

Объект исследования: закономерности и шаблоны изменения курсов валют.

Цель работы: анализ изменения курсов валют и разработка торговой стратегии на основе глубокого обучения с подкреплением.

Результаты: на основе полученных знаний и в ходе практического эксперимента разработана стратегия, запрограммирован и протестирован алгоритм прогнозирования курсов валют. У полученного алгоритма есть потенциал для применения при торговле в метатрейдере и он может быть основой последующих улучшенных алгоритмических стратегий.

Область применения: торговля на Форекс.

АГУЛЬНАЯ ХАРАКТЕРЫСТЫКА ПРАЦЫ

Магістарская праца, 45 старонак, 14 ілюстрацый, 4 табліцы, 7 крыніц.

КУРСЫ ВАЛЮТ, ГЛЫБОКАЕ НАВУЧАННЕ З ПАДМАЦАВАНЬНЕМ, НЕЙРОНАВЫЯ СЕТКІ, НАВУЧАННЕ МАДЭЛІ, ІНДЫКАТАРЫ, ТРЭЙДАР, ТРЭНД, ЗДЗЕЛКА

Аб'ект даследавання: заканамернасці і шаблоны змены курсаў валют.

Мэта работы: аналіз змены курсаў валют і распрацоўка гандлёвай стратэгіі на аснове глыбокага навучання з падмацаваннем.

Вынікі: на аснове атрыманых ведаў і ў ходзе практычнага эксперыменту распрацавана стратэгія, запраграмаваны і пратэставаны алгарытм прагназавання курсаў валют. У атрыманага алгарытму ёсць патэнцыял для прымянення пры гандлі ў метатрэйдары і ён можа быць асновай наступных палепшаных алгарытмічных стратэгий.

Вобласць прымянення: гандаль на Форекс.

ABSTRACT

Master's thesis, 45 pages, 14 pictures, 4 tables, 7 sources.

CURRENCY EXCHANGE RATES, DEEP REINFORCEMENT LEARNING,
NEURAL NETWORKS, MODEL TRAINING, INDICATORS, TRADER, TREND,
DEAL

The object of research: regularities and patterns of change of exchange rates.

The aim of the work: analysis of changes in exchange rates and development of a trading strategy based on deep reinforcement learning.

The results: on the basis of the obtained knowledge and practical experiment, an algorithm for the prediction of currency exchange rates has been developed, programmed and tested. The resulting algorithm has the potential to be used for trading in the metatrader and it can be the basis for further improved algorithmic strategies.

Scope: Forex trading.

ГЛАВА 1

ОСНОВНЫЕ ПОНЯТИЯ И ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Основные положения

Форекс (валютный рынок, от англ. *Foreign Exchange* — «зарубежный обмен») — это система устойчивых экономических и организационных отношений, возникающих при осуществлении операций по покупке или продаже иностранной валюты, платежных документов в иностранных валютах, а также операций по движению капитала иностранных инвесторов.

Основными участниками валютного рынка являются:

1. Центральные банки;
2. Коммерческие банки;
3. Инвестиционные и пенсионные фонды;
4. Предприятия, осуществляющие экспортно-импортные операции;
5. Валютные биржи;
6. Валютные брокерские фирмы;
7. Физические лица. [2]

Когда речь идет о том, чтобы торговать конкретному человеку на рынке Форекс, то это невозможно делать без торгового брокера (брокерской фирмы), которые и осуществляют посредничество. За свою работу они берут некоторую комиссию или фиксированную сумму, а взамен трейдер может открывать реальный счет или демо-счет, пополнять его и выводить средства. Также эти фирмы предоставляют различные инструменты и программы, такие как метатрейдеры (например, всем известная программа MetaTrader от Alpari).

Прежде чем начать торговать на Форексе, требуется выбрать подход, с учетом которого будет производиться анализ изменения курсов валютной пары и разрабатываться торговая стратегия. Существует 3 главных подхода к анализу и торговле на валютном рынке:

1. *Фундаментальный анализ.* Изучает движение цен на макроэкономическом уровне. Он может способствовать определению главного рыночного *тренда* (основное направление динамики рыночной цены), однако для определения конкретного момента совершения сделки фундаментального анализа зачастую бывает недостаточно, и тогда используют технический анализ. Основными индикаторами в фундаментальном анализе являются мировые новости, экономические новости, объем проданной валюты, исторические события, добыча полезных ископаемых и др.;

2. *Технический анализ*. Основан на следующей гипотезе — рыночные цены являются отражением желаний и действий всех участников рынка. В результате и цена и объем отражают каждую сделку, совершенную всеми трейдерами в конкретный момент времени. Важнейшую роль в техническом анализе составляют технические индикаторы и осцилляторы, функции которых используются для определения специфических показателей и закономерностей в изменении курса валютной пары. Данный тип анализа не так сильно зависит от валютной пары, из-за чего большая часть индикаторов может применяться сразу для нескольких валютных пар;

3. *Интуитивный подход к анализу*. Мало кем используется, не приводит к долговременному успеху.

На выбор того или иного подхода влияет множество показателей, в том числе валютная пара, объем сделок, количество свободного времени трейдера, уровень осведомленности и изучения политики той или иной страны, период наблюдения и др.

Виды торговли по времени открытия сделки:

1. *Краткосрочная торговля*. Сделки совершаются в пределах дня и обычно занимают от нескольких минут до нескольких часов или дней;

2. *Среднесрочная торговля*. Сделки совершаются не так часто, как при краткосрочной торговле, занимают от нескольких дней до нескольких недель;

3. *Долгосрочная торговля*. Сделки могут быть открыты на протяжении недель или даже месяцев.

Краткосрочная и среднесрочная торговли являются типичными для технического анализа, а вот долгосрочная чаще всего используется теми трейдерами, что владеют глубокими знаниями в фундаментальном анализе.

Вне зависимости от выбора подхода и периода для наблюдения, чтобы провести анализ изменения курса валютных пар, а также разработать и реализовать торговую стратегию, которая будет прогнозировать скачки курса валютной пары, необходимо проделать определенную последовательность действий:

1. Выбрать валютную пару и период для построения стратегии;
2. Загрузить котировки и данные;
3. Разработать стратегию;
4. Протестировать стратегию;
5. Применить стратегию в приложении онлайн-трейдер.

Таким образом, рассмотрены основные моменты, с учетом которых и будет разработана торговая стратегия для рынка Форекс.

В данной работе я выбрал технический анализ и краткосрочную торговлю, как самый популярный подход в торговле на Форексе. Конечно, еще учтен тот факт, что методы машинного обучения, о которых будет написано далее и которые планируется использовать (а именно обучение с подкреплением), эффективны и в большинстве своем применяются именно для технического анализа (хотя и для фундаментального анализа тоже, например, при разборе содержания экономических новостей).

Целью данной работы в первую очередь является научный интерес и возможность построить некоторую закономерность в курсах валют, эффективно применяя методы машинного обучения, а также быть уверенными, что мы знаем, как работает алгоритм прогнозирования, поскольку данная информация в обычных приложениях «трейдерах» скрыта либо может быть предоставлена неверной. Также эта работа заложит основу для написания впоследствии уже более серьезных алгоритмов анализа курсов на основе разработанного, чтобы впоследствии использовать для торговли на реальном счете на Форексе.

1.2 Выбор валютной пары и периода для наблюдения

Выбор валютной пары и периода для наблюдения – одни из ключевых моментов, когда речь заходит о разработке стратегии торговли на Форексе.

Если трейдер выбрал фундаментальный анализ и его ключевым показателем является выход новости и ее содержание, то если экономики стран зависимы, новость будет влиять примерно одинаково на обе валюты в валютной паре и из-за этого торговая стратегия может быть неэффективна.

Когда же речь идет про технический анализ, который я и выбрал, то здесь уже, как правило, закономерности и основные шаблоны тренда могут подходить под разные валютные пары. Самые популярные валютные пары в мире - доллар США, евро, фунт стерлингов, японская иена и швейцарский франк входят в группу основных валютных пар: EUR/USD, GBP/USD, USD/JPY и USD/CHF. Существуют три другие валюты, которые являются общими для торговли на рынке Форекс: новозеландский доллар, канадский доллар и австралийский доллар. Если они связаны с долларом США, мы получаем группу пар валют: NZD/USD, USD/CAD и AUD/USD [3].

В своей работе выбрал валютную пару **USD/JPY** как одну из основных валютных пар. В дальнейшем с целью улучшения и тестирования стратегии можно сравнить показатели, полученные на данной валютной паре, и скажем, на паре EUR/USD.

Что же касается периода для наблюдения, то тут взяты данные за 1 календарный год (2019). Так как торговля ведется краткосрочная, нужно выбрать минимальный интервал, по истечении которого будет анализироваться новое значение курса. Таким оптимально является 1 час, потому что:

1. Так как торговля краткосрочная, брать интервал, равный 1 день, бессмысленно, такой подходит для среднесрочной торговли;
2. Слишком маленький интервал тоже плох, так как учитывает содержит в себе слишком много шума и колебаний рынка, сопоставимых с погрешностью.

В итоге выбрана валютная пара USD/JPY с интервалом в 1 час, при этом период наблюдения взят равным 1 год.

1.3 Основные цели стратегии

У торговой стратегии есть две основные цели:

1. Уменьшить время, за которое алгоритм сможет выдать ответ, нужно ли открывать сделку или нет;
2. Уменьшить время и потери, которые будут после завершения тренда (то есть, когда все пойдет в обратную сторону).

Оба этих пункта крайне важны, поскольку как раз-таки и определяют то, насколько эффективна будет стратегия. Разумеется, немаловажным является и то, например, на что открывается сделка (повышение/понижение). Однако это уже будет рассмотрено и проанализировано далее.

Теперь, сформулировав ряд целей и обосновав выбор валютной пары и период для анализа, можно приступать непосредственно к выбору инструментов для разработки из арсенала машинного обучения для разработки собственной стратегии и проведения экспериментов.

1.4 Применение машинного обучения для торговли на Форексе

Машинное обучение – обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться. Влияние и значение машинного обучения на разные сферы деятельности и разделы науки все больше увеличивается, и финансы тому не исключение.

Основные типы задач машинного обучения в контексте торговли на рынке Форекс (рисунок 1.1):

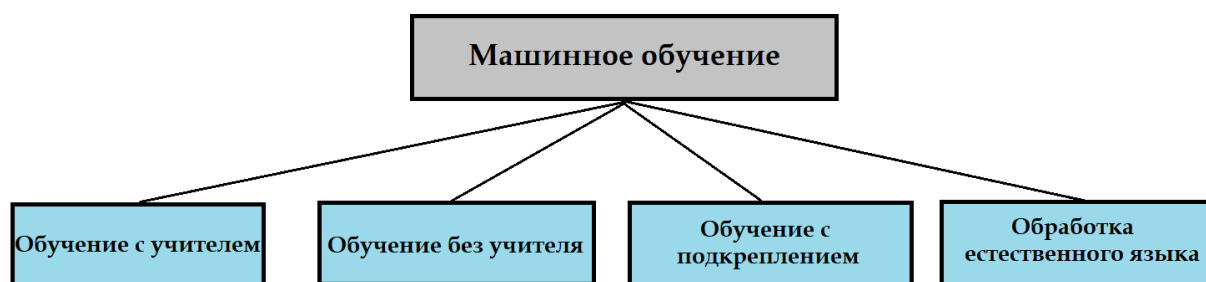


Рисунок 1.1 – Основные типы задач машинного обучения в контексте торговли на рынке Форекс

1. *Обучение с учителем.* В данном разделе есть алгоритмы регрессии и классификации. Алгоритмы регрессии, такие как линейная регрессия, метод k ближайших соседей, дерево принятия решений и другие, используются для предсказания значения непрерывной величины, а именно курса валют в следующий момент времени [4]. Алгоритмы классификации, такие как логистическая регрессия, метод опорных векторов и другие, используются для того, чтобы предсказать по значениям курса, пойдет цена вверх или нет;

2. *Обучение без учителя.* В данном случае у нас нет размеченных данных, как это было в обучении с учителем. Однако обычно при торговле на Форексе очень много входных данных, и чтобы уменьшать размерность без потери информации, используются алгоритмы из данного раздела, такие как метод главных компонент и другие;

3. *Обучение с подкреплением.* Так как обучение на основе опыта и вознаграждений лежит в его основе, а при торговле на Форексе нужно минимизировать убытки и максимизировать прибыль, то целесообразно использовать в алгоритмической торговле и при управлении финансовым портфелем;

4. *Обработка естественного языка.* В отличие от предыдущих трех, может эффективно использоваться в фундаментальном анализе при обработке экономических и мировых новостей и выявлении на основании их содержания закономерностей и трендов в изменении курсов валют.

В своей работе я выбрал в качестве метода машинного обучения – обучение с подкреплением.

ГЛАВА 2

ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ

2.1 Описание метода

Обучение с подкреплением – один из методов машинного обучения, в основе которого лежит взаимодействие разработанной системы (далее *агента*) с окружающей средой [5]. Данный подход используется во многих задачах и играх, когда нужно определенным образом обучить алгоритм, чтобы выполнялись следующие требования:

1. Максимизировать награду, которую можно получить в результате выполнения некоторых действий;
2. Минимизировать убытки, которые случаются при неверно выбранном варианте действия.

Как ни странно, если говорить про трейдера, который торгует на котировках на Форексе или чем-то другим (например, акциями), то именно описанные задачи выше он и должен решать. Награда для него – это разница в курсе продаж и покупки, когда с учетом уплаченных процентов системе остается прибыль от сделки. Если же сделка была совершена неверно либо просто была неприбыльной, но закрытой, то это не что иное, как убыток.

Торговая стратегия и использующий ее торговый робот, в основе которой лежит метод обучения с подкреплением и которая может использоваться для совершения реальных сделок на Форексе – это отличный инструмент и помощник для трейдера. Однако стоит учесть тот факт, что с первой попытки придумать стратегию и тут же использовать в реальной торговле не выйдет: это кропотливый процесс, который требует много времени уделить не только построению самого алгоритма, но и его обучению и тестированию.

В контексте торговли на Форексе, агент представляет собой аналог трейдера, который открывает либо закрывает сделки. Сама же среда представляет собой котировки и индикаторы, которые и характеризуют среду в разные моменты времени.

Далее подробнее будут описаны основные компоненты, которые требуется разработать для обучения с подкреплением.

2.2 Основные компоненты

Чтобы разработать стратегию торговли на основе обучения с подкреплением, нужно понимать, какие компоненты будут входить в торговую

систему и формировать ее. Это общие понятия для любой подобной системы, но в данном случае описание представлено в контексте торговли на Форексе.

Основные компоненты обучения с подкреплением:

1. Торговый агент;
2. Действие;
3. Функция награды (reward function);
4. Среда;
5. Состояние среды.

Остановимся на каждом компоненте подробнее.

Торговый агент – это сущность, которая представляет собой компонент для торговли на Форексе.

Действие – это некоторая операция, которая выполняется торговым агентом. Действий может быть 3:

1. Покупка валюты (*Buy*);
2. Продажа валюты (*Sell*);
3. Нет действия (*No action*).

Именно из-за третьего действия мне и нравится данный метод. Все дело в том, что покупка и продажа совершаются не так часто, так как приходится выжидать лучшего момента для их реализации, и во время этого ничего кроме накопления знаний о системе не происходит. Однако для разметки данных для обучения, например, на задаче классификации, это очень важно. И вот проблема: мало того, что есть некоторая сложность в пометке множества моментов времени и котировок как *не активные*, так это еще и составляет большую часть пометок в данных. То есть получается, что нужно решать задачу балансировки выборки из-за того, что агент большую часть времени не выжидает, а не открывает новые сделки. Как дальше будет показано, метод обучения с подкреплением решает эту задачу за нас.

Функция награды (reward function) – функция, которая дает ответ агенту в виде некоторой величины (в нашем случае прибыльность текущего действия). Подробнее о ней будет написано далее.

Среда – изменение курса котировок во времени, ее отслеживает и на нее направлена стратегия торгового агента.

Состояние среды – параметры и индикаторы, которые описывают текущее состояние системы в данный момент времени. В зависимости от того, насколько полно и какими параметрами и индикаторами описывается состояние, торговый агент может более или менее полно обучаться, так как именно сведения о текущем состоянии будут передаваться для обучения.

Все описанное выше проиллюстрировано на рисунке 2.1 в виде простой блок-схемы.

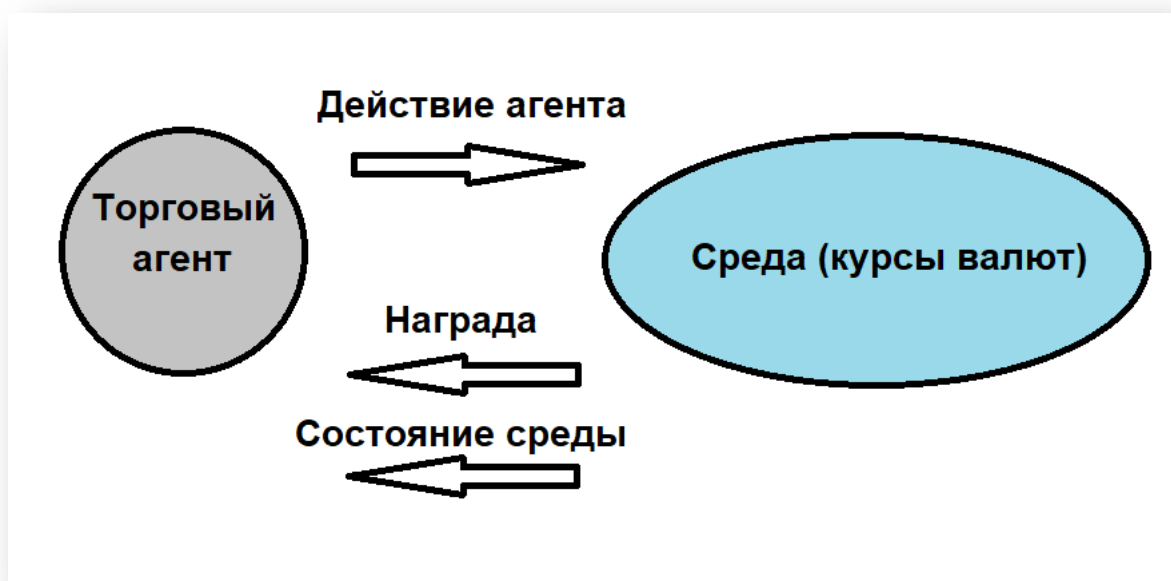


Рисунок 2.1 – Основные компоненты обучения с подкреплением

Цель торгового агента – максимизировать прибыль и минимизировать убытки. Он производит некоторые действия, а в результате получает награду, в результате чего и происходит шаг за шагом обучение.

Но как же описать самого агента и среду. Это можно делать по-разному. Например, среда может быть описана моделью, с которой взаимодействует агент. Модель – это дескриптор среды. С помощью модели мы можем узнать, как окружающая среда будет взаимодействовать с торговым агентом и давать сигналы ему. Модель среды описывается функцией награды (*reward function*) и функцией вероятности перехода (*P*).

Функция награды (R_t , *reward function*) – награда, которую мы можем получить в t -ый момент времени.

Функции награды используются для определения функции будущей награды.

Функция будущей награды (G_t , *future reward*) – награда, которую можно получить, начиная с момента времени t :

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k-1} \quad \#(2.1)$$

Переменная γ в пределе от 0 до 1, и это параметр недоверия будущим значениям, так как в будущем все не так определено и из-за этого получается, что те значения должны быть с меньшим весом, чем ближайшие к нам по временной оси [6].

Q-функция (Q-value) – функция от пары действие-состояние, которая говорит, какое мы ожидаем получить вознаграждение в будущем, если есть такое состояние s_t системы и торговый агент выполняет действие a_t :

$$Q(s, a) = E[G_t | s = s_t, a = a_t] \quad (2.2)$$

Функция вероятности перехода (P) – функция, описывающая вероятность перехода из одного состояния в другое после совершения некоторого действия.

Однако если у нас есть функция вероятности перехода и ограниченное количество действий, то уместно говорить об алгоритмах, основанных на модели окружающей среды (*Model-based algorithms*).

Если же информации недостаточно (у нас нет функции вероятности переходов или действий слишком много, которые можно выполнить, то есть нет достаточных предварительных знаний, то речь идет об алгоритмах обучения с подкреплением, не зависящих от моделей (*Model-free algorithms*). Одним из таких алгоритмов и является *глубокое обучение с подкреплением*. Дальше о нем и пойдет речь.

2.3 Глубокое обучение с подкреплением

Глубокое обучение с подкреплением (Deep Q-Network) – подход машинного обучения, при котором используются нейронные сети для обновления и аппроксимации Q-функции. Ведь Q-функция – это по факту таблица, по которой мы можем сказать, какое действие нужно выполнить, если на вход пришло определенное состояние. И наступает такой момент, когда если это делать без нейронной сети, то возникают следующие проблемы:

1. Когда число состояний и число действий велико, таблица Q-значений становится ужасно сложной;
2. Появляется нестабильность, так как при огромном количестве параметров Q-значения могут меняться в обратную сторону, но все еще выдавать хорошие результаты все на новых и новых данных.

Эти проблемы решает глубокое обучение с подкреплением. Используя нейронную сеть, задача сводится к оптимизации весов многоуровневой

нейронной сети, и это уже выглядит не как таблица, а как множество слоев. Кроме того, следующие два подхода значительно улучшают обучение:

1. Воспроизведение опыта (*experience replay*) – возможность алгоритма сохранять в памяти набор объектов, хранящих в себе действия, награды, следующие состояния, которые во время обучения можно извлекать и на них обучаться. Этот прием уменьшает корреляцию в последовательности и сглаживает изменения в данных;

2. Периодическое обновление целевых значений – возможность не без перерыва обновлять значения, а с каким-то интервалом. Тогда получается, что можно добавить гиперпараметр *step* (шаг), который будет влиять на то, насколько важны нам мелкие колебания и стоит ли обучаться на них (например, если речь идет о краткосрочной торговле, то задав такой параметр, можно не обучаться на каждом мелком промежутке времени, а брать лишь более важные и объективные значения курса).

На рисунке 2.2 схематически изображена Q-нейронная сеть для торговли на рынке Форекс.

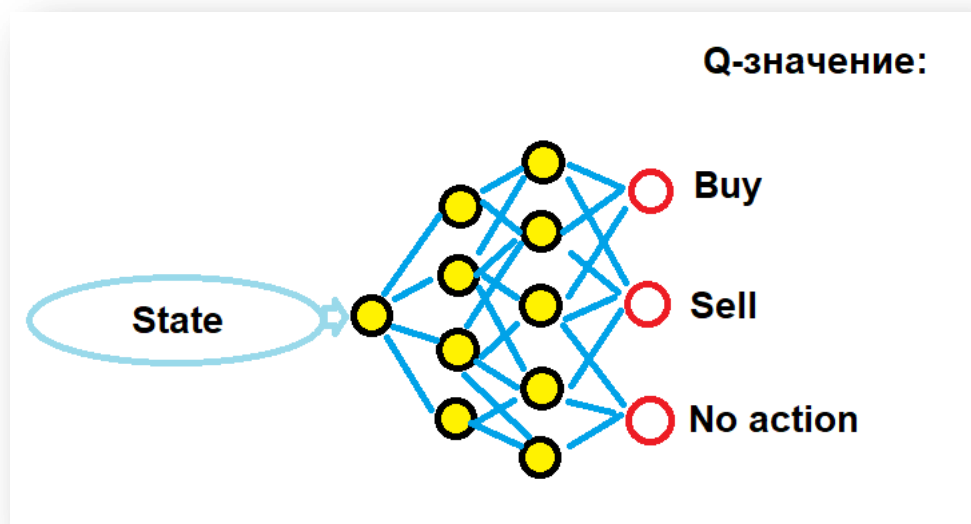


Рисунок 2.2 – Глубокая Q-нейронная сеть

Модель подобной нейронной сети и будет использована в торговом агенте, который описан в следующей главе.

2.4 Особенности обучения с подкреплением

Перед тем как приступить непосредственно к разработке торгового агента и нейронной сети как его модели, стоит упомянуть о некоторых особенностях и проблемах, с которыми можно столкнуться в ходе работы над такой стратегией:

1. Эффективное использование ресурсов. Все дело в том, что чтобы обучить нейронную сеть хорошо предсказывать и оценивать то или иное действие торгового агента, нужно как и обычно при обучении нейронных сетей потратить на это время и CPU или GPU. Обучение может занять достаточно много времени, прежде чем удастся получить значения весов, удовлетворяющих требованиям;

2. При задаче торговли на Форексе может так получиться, что сигналы (индикаторы, состояние) могут ассоциироваться с наградой, которая должна быть получена гораздо позже, чем та, на которой в данный момент обучается нейронная сеть. То есть, возможно, стоит обращать внимание не только на награду в данный момент времени, но и на ту, которая может быть получена через некоторое количество шагов после текущего момента времени;

3. Также довольно сложно на первый взгляд понять взаимосвязь между входными данными (в нашем случае состоянием системы) и ее результатами (выходными данными). Хотя это и задача нейронной сети, результат обучения не всегда получается интерпретировать. Именно поэтому добавление большого количества параметров и индикаторов с одной стороны, может привести к лучшим результатам, а с другой будет еще сложнее позже их модифицировать, так как неясно будет, что и как повлияло на результат обучения нейронной сети;

4. В данный момент я еще не придумал, как правильно разделять сумму, на которой торговать. Это тоже одна из задач (задача разделения портфеля), которые может решить обучение с подкреплением (то есть решать, какой процент от имеющейся суммы стоит отдать под конкретную сделку). Однако в данный момент я решил, что достаточно взять сумму сделки 1 у.е (1 USD для валютной пары USD/JPY, 1 EUR для валютной пары EUR/USD), и тогда наградой будет именно разница на курсе. А с учетом того, что на Форексе можно использовать торговое плечо, в зависимости от его значения можно получать нужную нам прибыль;

5. Процент успешных сделок, как при задаче классификации также сложно посчитать, так как неясно в данном случае, что считать успешной сделкой. Однако если считать, что сделка в минус – ошибочная сделка, а сделка в плюс – успешная, то выбрав какое-то пороговое значение (чтобы избежать

совсем уж маленьких величин) можно все же посчитать некоторую величину успешности. Вообще в данный момент мне интересна итоговая сумма как основной критерий успешности стратегии.

Это те особенности, на которые имеет смысл обратить внимание при дальнейшем изучении и разработке торгового агента и модели нейронной сети.

2.5 Итоги

В основу разрабатываемой стратегии положен метод глубокого обучения с подкреплением из-за плюсов, описанных выше. В таком случае удастся применить знания, полученные в ходе изучения нейронных сетей и разобраться в построении торгового агента для открытия/закрытия сделок на Форексе на основе обучения с подкреплением.

Поскольку необходимо реализовать не только агента, но и нейронную сеть, которую он использует, можно будет разделить разработку на некоторые этапы.

Операций у агента будет 3 – основные операции при торговле – купля, продажа, выжидание (нет действий). Таким образом, он максимально будет походить на трейдера, который торгует.

В дальнейшем разработку можно применять не только для интеграции с теми же программами-метатрейдерами, но и для изучения и улучшения разработанной стратегии.

Описание разработки торгового агента, а также обучения и тестирования (выбран для реализации язык Python) будет представлено в следующих главах.

ГЛАВА 3 РАЗРАБОТКА СТРАТЕГИИ

3.1 Описание стратегии

Основная цель торговой стратегии – максимизация прибыли и минимизация убытков. В прошлой главе был описан метод обучения с подкреплением, который и ляжет в основу реализованной далее стратегии. Подробнее остановлюсь на некоторых моментах, которые я учитывал при разработке и из-за которых я принимал те или иные решения.

Основные положения, которые будут применяться в данной стратегии:

1. По-прежнему рассматривается лишь одна валютная пара – американский доллар/японская йена (USD/JPY). Возможно, в дальнейшем есть смысл сравнить с другими валютными парами, такими как EUR/USD или USD/RUB. Однако сейчас для первой реализации и обучения будет достаточно и одной валютной пары;

2. Ведется краткосрочная торговля, то есть интересны интервалы в 1 час в пределах выбранного периода наблюдения. Такой подход позволяет реализовать высокочастотную торговую стратегию, которая для данной задачи и фич подходит лучше, а также интересна с практической точки зрения. Было принято решение отказаться от долгосрочной торговли с интервалом в 1 день, так как хоть в долгосрочной торговле тоже есть закономерности и шаблоны изменения курса, распознать их достаточно сложно в силу таких экономических явлений, как новости, которые сильно влияют на долгосрочный тренд. Плюс, в таком случае, когда не берутся очень маленькие интервалы, такие как 5 или 15 минут, нельзя ссылаться на колебания курса, которые очень мешают при краткосрочной торговле, из-за чего приходится выбирать постоянно разные минимальные интервалы времени от 5 минут до часа (1 час и взят за основной интервал);

3. Все основные действия по обучению и выбору необходимого действия будут выполняться торговым агентом, который, в свою очередь, будет содержать и модель нейронной сети с обученными весами. Модель является его важнейшей характеристикой, так как именно ее предсказания, основанные на состоянии системы – предсказания торгового агента;

4. Для описания состояния системы будут использоваться как различия в курсе в пределах торгового окна (количество предшествующих значений котировок), так и некоторые индикаторы из технического анализа. Однако в дальнейшем планируется использовать и более сложные индикаторы,

рассчитанные не только на основе курса (например, вышла ли новость или какой объем валюты был продан). Понятно, что некоторые параметры могут быть несущественными, однако иные в состоянии достаточно сильно влиять на обучение нейронной сети.

Основная цель стратегии – получения максимальной прибыли. Это задача торгового агента, который получает состояние среды и выдает некоторое действие (купить, продать, ничего не делать). Обучение алгоритма происходит с помощью глубокого Q-обучения, в основе которого лежит модель глубокой нейронной сети, которая обучается и аппроксимирует функцию Q-значений, обучая веса нейронной сети. То есть в конечном итоге, обучив веса, мы можем передавать состояние и получать ответ – у какого действия больше шансов реализоваться либо какое действие принесет максимальную выгоду.

Чтобы реализовать глубокое Q-обучение, необходимо подготовить код торгового агента.

3.2 Торговый агент

Торговый агент представлен классом `Agent`, который содержит 3 основных метода:

1. `act` – нужен для того, чтобы в ответ на переданное состояние `state` выдавать одно из трех действий: *buy*, *sell* или *no action*. Данный метод также в случае, если торговля лишь началась и `epsilon` (один из параметров агента) еще достаточно велик (близок к единице) выдает случайное действие (1, 2 или 0 соответственно), то есть вначале расчет больше на изучение среды агентом, а не применение полученных знаний (которых, очевидно, изначально нет);
2. `model` – этот метод возвращает модель глубокого обучения, используемую в торговом агенте, важнейшую его характеристику;
3. `replay` – метод непосредственного обучения, который будет вызываться во время обучения на тестовых данных и постепенно с помощью которого будут улучшаться веса нейронной сети.

Помимо этих трех методов, есть еще конструктор, в который можно передавать самые разные параметры (в данном случае передаю размерность (количество) параметров, описывающих среду `stateSize`, а также название модели нейронной сети `modelName` для загрузки в торгового агента и флаги `isEvaluate` и `isLoadModel`, которые показывает, производится ли сейчас обучение или тестирование и нужно ли загружать уже обученную модель соответственно).

Кроме этих переменных, в самом агенте хранятся переменные `epsilon`, которые будут использоваться в начале обучения, когда нужно выбирать случайным образом действия. Со временем действия должны браться исключительно из тех знаний, что были получены в ходе обучения, поэтому постепенно значение `epsilon` будет становиться все меньше и меньше и в итоге почти всегда случайное значение в промежутке от 0 до 1 (а `epsilon` именно в этих пределах) будет больше (а по условию должно быть меньше) и в итоге все чаще будет браться именно значение из *Q-value* (то есть нейронная сеть уже будет предсказывать, а не угадывать нужное действие).

Рассмотрим подробнее метод `replay`, так как именно в нем происходит основное обучение нейронной сети, то есть нашего торгового агента.

Основные шаги в методе `replay`:

1. Поскольку в агенте хранится определенное количество состояний (в переменной `memory`), извлекаем некоторое количество последних, равное переданному значению переменной `batchSize`. Этот набор будет использован для обучения;

2. Далее для всего набора, представляющего собой кортеж из значений текущего состояния, следующего состояния, награды и действия, выполняем следующие пункты (3 - 5);

3. Во всех случаях, за исключением конца эпизода, выполняется обновление награды согласно следующему уравнению (уравнение Беллмана):

$$Q_t(s, a) = R_t + \gamma * \operatorname{argmax}(Q_{t+1}(s, a)) \quad \#(3.1)$$

где Q_t – значение Q-функции в момент времени t ;

s – состояние системы;

a – действие торгового агента;

R_t – значение функции награды в момент времени t ;

γ – параметр недоверия будущим значениям в отрезке $[0; 1]$;

Q_{t+1} – значение Q-функции в следующий момент времени $t + 1$.

4. Обновляем *Q-value* для текущего действия в данный момент, для которого уже можем с помощью нейронной сети предсказать значения для всех действий;

5. Обучаем нейронную сеть после обновления целевой переменной (новое *Q-value*).

6. После этого, если необходимо, то уменьшаем `epsilon`.

Таким образом, в пунктах 1-6 описана схема, по которой работает метод `replay`. Вызываться он будет лишь во время обучения и тогда, когда уже будет достаточно объектов в памяти (пункт 1), на которых можно будет обучаться и обновлять веса нейронной сети. Сама модель нейронной сети, которую я использую в агенте, описана в следующем пункте.

3.3 Модель нейронной сети

В методе `model` может быть задана любая модель нейронной сети, которая подходит и удобная в данной задаче. Рассмотрим подробнее структуру модели, которая используется сейчас:

1. На вход модель ожидает строку размера `stateSize`. По факту в данный момент это и есть количество параметров, которые приходят на входной слой (это значение зависит от того, какие из критериев и индикаторов в данном обучении используются, то есть этот размер ничем не ограничен, если все значение индикаторов, описывающих обстановку на рынке за предыдущий период, важны);

2. Далее используется 3 внутренних слоя с функцией активации *ReLU* и количеством нейронов 128, 128, 128 на каждом из слоев (можно было взять и 64, 32, 8, но так я лишь расширил сетку, так что она включает по факту в себя и сетку меньшей размерности);

3. В выходном слое я использую линейную функцию активации. Количество нейронов равно количеству действий, которое может совершить торговый агент (в данном случае 3, так как кроме *открытия*, *закрытия* и *никакого действия* больше модель ничего и не должна выдавать).

4. Также как `loss` используется *Mean Squared Error* и *Adam* оптимизатор.

Далее приведено общее описание модели и количество параметров в ней (рисунок 3.1)

Model: "sequential"		
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 128)	256
dense_1 (Dense)	(None, 128)	16512
dense_2 (Dense)	(None, 128)	16512
dense_3 (Dense)	(None, 3)	387
Total params: 33,667		
Trainable params: 33,667		
Non-trainable params: 0		

Рисунок 3.1 – Описание модели

Таким образом, пока что я решил использовать данную модель как исходную, в дальнейшем конечно планирую протестировать ее с разными параметрами и возможно она претерпит некоторые изменения, так как при использовании нейронной сети для прогнозирования сделок, у нас есть много параметров, с которыми можно экспериментировать:

1. Количество скрытых слоев;
2. Количество нейронов в скрытом слое;
3. Количество циклов обратного распространения;
4. Скорость обучения;
5. Импульс, коэффициент инерции для адаптации весов;
6. Функция активации.

Но это лишь параметры в пределах модели нейронной сети. Как минимум стоит еще учитывать, что и формат строки, которая передается нейронной сети на вход, а также параметры в самом торговом агенте и при обучении могут очень сильно влиять на то, какие в итоге значения весов будут в обученной модели и насколько они будут хорошо подходить под критерии успешной модели.

3.4 Дополнительные функции

В пункте 3.2 описывалось, что методы `act` и `replay` торгового агента работают с переменной `state`, которая представляет собой описание состояния в текущий момент времени. Это может быть список из значений последних `n` котировок либо же набор технических индикаторов. Как пример, можно использовать разницу курсов за промежуток времени, равный торговому окну (например, 5 часов).

Все действия, которые подготавливают состояния для дальнейшей передачи в методы трейдера, реализованы в методе `getState`. В дальнейшем именно в нем я буду наполнять состояние различными индикаторами и параметрами.

Важно, чтобы параметры были отмасштабированы (для этого можно применить, например, функцию сигмоиды либо использовать различные классы из готовой библиотеки *sklearn*), так как данные должны быть стандартизированы и, например, разница в курсе и скользящее среднее — величины совершенно разного порядка.

Помимо этого, были реализованы методы для отображения данных после обучения и тестирования, чтобы лучше понимать, какие сделки (открытие или

закрытие) в какое время были открыты, а также какова была общая прибыль за выбранный интервал.

Процесс и реализация обучения, а также тестирование будут описаны в следующей главе.

ГЛАВА 4

ОБУЧЕНИЕ И ТЕСТИРОВАНИЕ СТРАТЕГИИ

4.1 Загрузка данных

Данные о курсах валют в прогнозировании очень важны, а именно их согласованность и правдивость, поэтому подходить к этому делу нужно основательно и очень аккуратно. Данные о курсах валют, то есть котировки валютной пары USD/JPY, скачаны с сайта www.histdata.com и www.investing.com [7]. Investing.com это платформа финансовых рынков, предоставляющая данные в режиме реального времени, котировки, графики, финансовые инструменты, последние новости и анализ на 250 биржах по всему миру в 44 языковых изданиях. Выбрав нужный период и валютную пару, можно без проблем скачать историю котировок.

Ранее я использовал сайт www.exness.com, однако ныне на нем можно скачать лишь данные о котировках в очень мелком интервале (там бывает даже по секундам), из-за чего данные будут требовать дополнительной обработки. Мне же нужны данные о котировках по часам, поэтому легче всего сразу получить их в нужном формате. Итого были скачаны файлы и преобразованы в формат csv за 2017, 2018, 2019 и 2020 годы (как будет показано позже, все варианты нужны). В результате после небольшой обработки сами данные представляют собой такую таблицу, как показано на рисунке 4.1.

Date	Open	High	Low	Close
2019-01-01 17:00	109.648	109.650	109.648	109.650
2019-01-01 18:00	109.676	109.676	109.657	109.675
2019-01-01 19:00	109.669	109.679	109.651	109.652
2019-01-01 20:00	109.625	109.629	109.625	109.628
2019-01-01 21:00	109.597	109.611	109.589	109.590

Рисунок 4.1 – Почасовые данные о котировках

Нас будет интересовать лишь колонка *Date* и *Close*. Обычно еще используется вместо *Close* название *Price*, но в данном случае это никакой роли

не играет. Впоследствии можно будет использовать помимо этих колонок и другие, и высчитывать от их значений иные индикаторы, но это позднее.

4.2 Этапы обучения торговой стратегии

Перед тем, как я приступлю к описанию обучения торговой стратегии, акцентирую внимание на переменных, которые в ней фигурируют.

`windowSize` – количество параметров, описывающих состояние (*state*). Название закрепилось после того, как я использовал окно в несколько часов для торговли описание системы в конкретный момент времени. Нужно для того, чтобы правильно инициировать входную размерность в нейронной сети, а также для передачи нужного интервала в функцию `getState`.

`agent` – переменная класса `Agent`, в которой есть методы `act` и `replay`, а также сама модель нейронной сети.

`buyInventory`, `sellInventory` – поля из класса `Agent`, список открытых и список закрытых позиций торговым агентом. В данные переменные сохраняются в момент открытия и извлекаются в момент закрытия значения котировок.

`memory` – поле из класса `Agent`, хранит в себе предыдущие состояния системы.

`state` – переменная, которая представляет собой список величин, которые вернула функция `getState` в t -ый период времени.

`nextState` – аналогично предыдущей переменной, но вычисляется для момента времени $t+1$.

`statesBuyTime` – список моментов времени, в которые было выполнено действие покупки.

`statesSellTime` – список моментов времени, в которые было выполнено действие продажи.

`totalProfit` – общая прибыль после завершения одного эпизода.

`episodeCount` – количество эпизодов или эпох обучения. Один проход по всем тестовым данным есть один эпизод.

`batchSize` – количество состояний (в данном случае последних), на которых будет обучаться нейронная сеть в методе `replay` класса `Agent`.

`done` – флаг, который переводится в значение *true*, если эпизод закончен.

Таким образом, я перечислил основные переменные, которые будут использоваться в написании обучения на всей тестовой выборке.

Итак, поскольку я выбрал торговую стратегию на основе Q-обучения, то этапы обучения представляют следующую последовательность действий:

1. Для каждого момента времени получаем состояние системы. Для этого вызываем функцию `getState` и записываем значение в переменную `state`;
2. Вызываем функцию `act` в классе `Agent`, которой и передаем состояние из предыдущего пункта;
3. Если значение действия 0, то это значит, что действие *no action*; если значение равно 1, то это действие *buy*. Нужно добавить текущее значение котировки в список позиций `buyInventory`, из которого на этапе продажи нужно извлечь цену покупки и сравнить для вычисления выгоды; если значение действия равно 2, это *sell*, в `sellInventory` нужно добавить текущее значение, а на этапе покупки далее извлечь для подсчета выгоды. Итог суммируем в переменную `totalProfit`;
4. Если вся тестовая выборка была просмотрена, то выставаем флаг `done` в `true`;
5. Текущее состояние `state` добавляем в список `memory`, и заменяем его на значение `nextState` (так как на следующем временном шаге мы снова получим следующее состояние, уже новое);
6. Если у торгового агента в памяти достаточно данных, чтобы обучаться, а именно их больше, чем `batchSize`, то вызываем метод `replay` у агента;
7. Также во время обучения после каждого эпизода (прохода по всем тестовым данным) отображаем график и сумму для более наглядного процесса обучения. Стоит отметить, что лучше также хранить все уже полученные данные для построения графиков и отчетности;
8. После каждого эпизода я сохраняю модель в *Google Colab* или на свой локальный диск, чтобы позже можно было ее использовать для тестирования и продолжения обучения в случае, когда что-то пошло не так и система дала сбой.

Методы, реализованные в классе `Agent`, были описаны в предыдущей главе, поэтому здесь лишь напомним, что и для чего используется:

1. метод `act` используется для того, чтобы получить одно из трех значений действия, а именно *buy*, *sell*, или *no action*, которые возвращаются в виде чисел 1, 2 или 0;
2. Метод `replay` нужен непосредственно для обучения нашей модели и обновления весов нейронной сети, и именно этот метод – ключевое отличие этапа обучения;
3. Метод `model` используется для реализации модели, которая используется непосредственно в торговом агенте для обучения.

4.3 Выбор технических индикаторов

Как уже было упомянуто ранее, состояние системы (*state*) описывается различным количеством параметров, или индикаторов. Поскольку в торговой стратегии я опираюсь лишь на значения изменения курсов, то я использую технические индикаторы для обучения и тестирования, на основании которых нейронная сеть и будет принимать решение, как поступить в данном конкретном состоянии среды.

Технический индикатор (технический индикатор рынка) – это функция, построенная на значениях изменения курсов валют или объема продаж, анализ поведения которой помогает выявить тенденции в изменении или сохранении текущего тренда.

В своей работе я использую следующие технические индикаторы: MOM, MA, ROC.

MOM (momentum indicator, индикатор импульса) – индикатор, который сравнивает значение текущего курса со значением курса в прошлом. Он чем-то похож на ROC, но за счет того, что нет нормализации значения, может использоваться совершенно по-другому.

$$MOM(t, n) = Price_t - Price_{t-n} \#(4.1)$$

где $MOM(t, n)$ – значение индикатора в момент времени t ;

$Price_t, Price_{t-n}$ – значения курсов в момент времени t и $t-n$.

MA – moving average, скользящее среднее – индикатор, значением которого является усредненное значение курсов за некоторый интервал времени. Достаточно распространенный среди трейдеров, также есть его разновидности и модификации (например, ЕМА, экспоненциальное скользящее среднее).

$$MA(t, n) = \frac{\sum_{i=t-n+1}^t Price_i}{n} \#(4.2)$$

где $MA(t, n)$ – значение индикатора в момент времени t ;

$Price_i$ – значения курсов в моменты времени i в интервале $[t-n; t]$.

ROC – price rate of change indicator, индикатор скорости изменения цены – осциллятор импульса, вычисляет отношение изменения курса за определенный период времени. Используется для идентификации тренда, а также для определения «перекупленности» или «перепроданности» валюты.

$$ROC(t, n) = \frac{Price_t - Price_{t-n}}{Price_{t-n}} * 100\% \#(4.3)$$

где $ROC(t, n)$ – значение индикатора в момент времени t ;

$Price_t, Price_{t-n}$ – значения курсов в момент времени t и $t-n$.

В итоге я использовал при обучении и тестировании стратегии индикаторы, описанные выше.

Примечание – Ко всем значениям индикаторов применяется функция сигмоиды для стандартизации значений, для скользящего среднего (МА) считается разность с текущим значением курса (именно в таком случае можно понять, какой сейчас тренд), а также для ROC нет умножения на 100.

4.4 Обучение и валидация торговой стратегии

Для обучения я выбрал данные о курсах котировок USD/JPY за 2019 год. Для обучения я взял 80% данных выборки, а для валидации и дополнительной проверки 20% (данные идут после данных для обучения, то есть это конец 2019 года, чуть больше двух месяцев). Уже на основании результатов валидации будут выбираться лучшие значения, на которых и будет проходить тестирование.

Обучение для всех параметров индикаторов производилось на протяжении 11 эпизодов, где эпизод – это одна итерация по всей обучающей выборке. Во всех случаях фигурируют два основных параметра, которые я менял:

1. *Размер батча* (количество состояний, на которых будет происходить обучение через каждый фиксированный интервал времени);
2. *Количество часов*, которые передаются параметром в функции индикаторов для вычисления значений, а после используются в описании состояния среды.

После этого, чтобы проверить, насколько хорошо подобраны параметры, проводится валидация.

Индикатор МА. Для МА я взял размер batch равный 32 и варьировал количество часов, за которые вычислялось значение индикатора. В результате на рисунке 4.2 отображен график значения валидации в зависимости от выбранного интервала и эпизода обучения. Видно, что с каждым эпизодом есть постепенное увеличение прибыли. Это обосновывается тем, что значение среднего на большем интервале может дать необходимую информацию о том, какой тренд в течение дня.



Рисунок 4.2 – График валидации МА для batch 32

Индикатор MOM. Для индикатора MOM я варьировал размер батча (8, 16, 32) и количество часов с интервалом в 2 часа. В итоге на рисунке 4.3 – данные для batch 32, рисунке 4.4 – данные для batch 16 и на рисунке 4.5 – данные для batch 8. Сразу видно, что прибыль на данном индикаторе удастся получить большую, чем на МА, и что лучше выбирать от 2 до 6 часов, так как дальше уже результаты гораздо хуже.



Рисунок 4.3 – График валидации MOM для batch 32



Рисунок 4.4 – График валидации MOM для batch 16

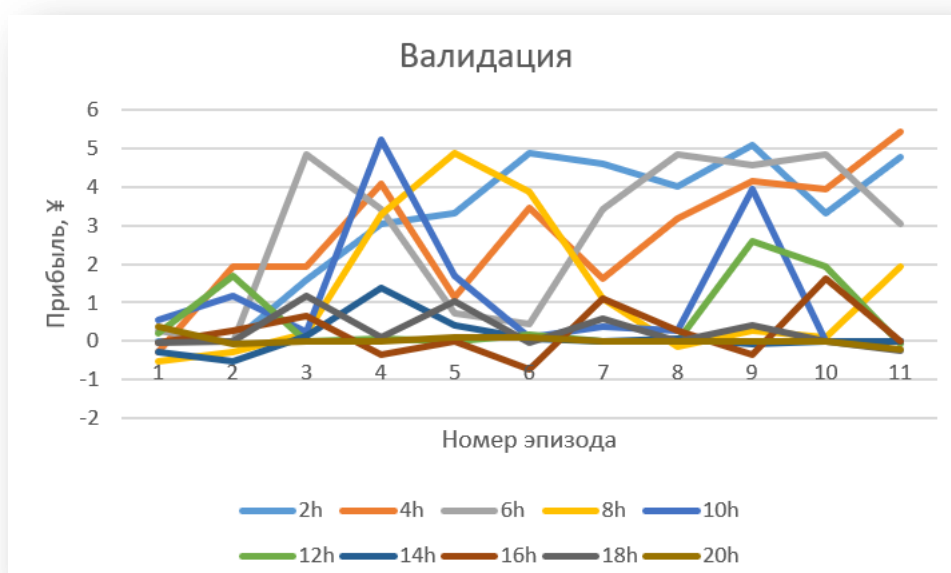


Рисунок 4.5 – График валидации MOM для batch 8

Индикаторы MA, MOM, ROC. В данном случае я варьировал размер батча (16, 32) и количество часов с интервалом, равным 6 часам. Поскольку индикаторов уже используется несколько, есть ожидания, что они будут учитывать разные параметры изменения курса и тем самым удастся улучшить показатели стратегии. В итоге на рисунке 4.6 – данные для batch 32, рисунке 4.7 – данные для batch 16. Видно, что есть колебания в значениях валидации и что если батч 32, то лучше взять интервал 24 часа, а если батч 16, то лучше брать 12 часов, так как нет отрицательных значений.



Рисунок 4.6 – График валидации МА, MOM, ROC для batch 32

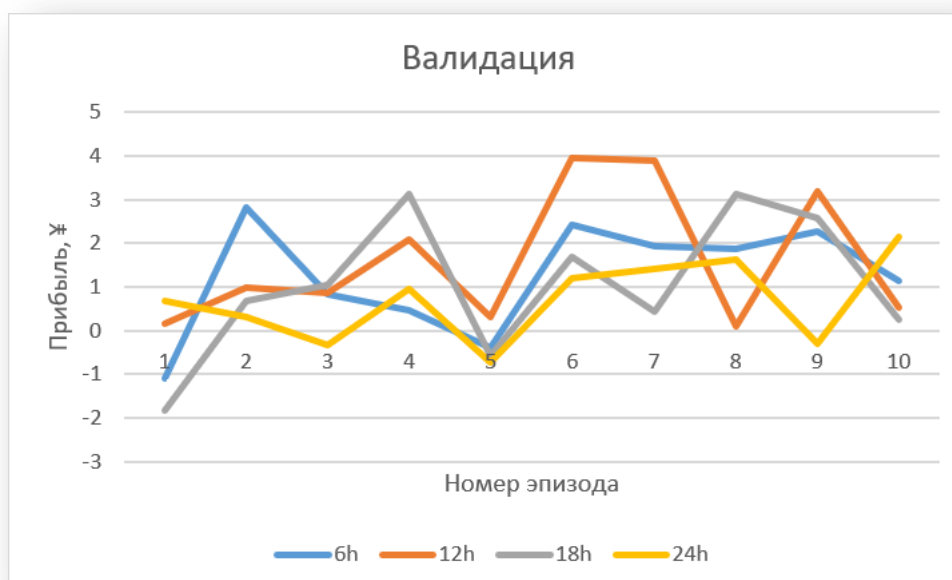


Рисунок 4.7 – График валидации МА, MOM, ROC для batch 16

Далее в таблице 4.1 приводятся результаты общей прибыли для лучших значений, полученных при валидации. Результаты идут в порядке уменьшения прибыли, то есть в первой строке идет наибольшее значение, а во второй и третьей – следующие за ним. Также я взял срез в виде пятого, седьмого и девятого эпизодов, так как именно на них в большинстве случаев достигались наибольшие значения. Конкретные параметры индикаторов, на которых достигаются наибольшие значения, буду интересны уже на этапе тестирования, поэтому здесь они никак не описываются.

Таблица 4.1 – Результаты валидации

	A	B	C	D	E	F	G
1		MA	MOM			MA, ROC, MOM	
2		batch 32	batch 32	batch 16	batch 8	batch 32	batch 16
3	episode 5						
4	validation (1st)	1.074	4.938	5.776	4.874	1.866	0.312
5	validation (2nd)	0.556	4.106	3.75	3.322	1.294	-0.391
6	validation (3d)	0.159	3.874	3.366	1.696	1.232	-0.56
7	episode 7						
8	validation (1st)	2.438	4.858	4.78	4.6	2.752	3.878
9	validation (2nd)	0.668	4.714	4.626	3.44	0.158	1.928
10	validation (3d)	0.469	4.622	4.622	1.616	-0.407	1.406
11	episode 9						
12	validation (1st)	1.788	5.524	5.222	5.104	2.628	3.2
13	validation (2nd)	1.487	5.104	4.956	4.564	1.122	2.588
14	validation (3d)	1.042	4.412	4.832	4.156	0.924	2.254

Как видно из таблицы, лучшие значения получены для индикатора MOM. На втором месте идет МА, если брать ранние эпизоды. Если же брать эпизоды позже, то есть седьмой или девятый, то лучше в данном случае комбинирование индикаторов МА, ROC, MOM. Исходя из этих данных о лучших стратегиях на основании валидации и будет осуществлено тестирование.

4.5 Тестирование торговой стратегии

Для тестирования я сперва выбрал данные о курсах котировок USD/JPY за 2020 год, а также еще за 2017 и 2018 годы для статистической достоверности. Модели стратегии и параметры выбраны на основании описанной выше валидации, поэтому теперь необходимо их протестировать на большом интервале времени. Результаты тестирования за 2020 год представлены в таблице 4.2.

Как видно, лучше всего по-прежнему индикатор MOM, причем лучше всего брать батч 32 или 16, так как на них удастся достичь максимальных значений. Например, 12.367 ¥ достигается если брать параметр индикатора 10 часов (аналогично и для валидации). Также видно, что на более поздних эпизодах результаты в среднем лучше, чем на более ранних, что говорит о том, что нейронная сеть еще доучилась. В целом индикатор хорошо себя показал и стратегия, основанная на нем, жизнеспособна.

Что касается остальных индикаторов (отдельно МА, комбинирования индикаторов МА, MOM, ROC), то хоть с параметром 24 часа на седьмом

эпизоде и есть результат 4.398 ¥ для MA, MOM, ROC, в остальных случаях стратегия выдает достаточно большой минус, который говорит о том, что еще нужно улучшать и по-другому комбинировать данные индикаторы.

Таким образом, по результатам тестирования на данных за 2020 год, лучшей себя показала стратегия, в которой используется индикатор MOM.

Таблица 4.2 – Результаты валидации и тестирования за 2020 год

	A	B	C	D	E	F	G
1		MA	MOM			MA, ROC, MOM	
2		batch 32	batch 32	batch 16	batch 8	batch 32	batch 16
3	episode 5						
4	validation (1st)	1.074	4.938	5.776	4.874	1.866	0.312
5	validation (2nd)	0.556	4.106	3.75	3.322	1.294	-0.391
6	validation (3d)	0.159	3.874	3.366	1.696	1.232	-0.56
7	test (1st)	-8.72	8.793	11.361	10.359	-2.74	-7.853
8	test (2nd)	-6.254	6.429	6.221	6.739	-7.615	-12.157
9	test (3d)	-7.13	7.299	5.311	3.031	-12.772	-1.519
10	episode 7						
11	validation (1st)	2.438	4.858	4.78	4.6	2.752	3.878
12	validation (2nd)	0.668	4.714	4.626	3.44	0.158	1.928
13	validation (3d)	0.469	4.622	4.622	1.616	-0.407	1.406
14	test (1st)	-5.62	10.193	9.425	8.257	-10.358	-9.748
15	test (2nd)	-1.194	8.095	9.331	5.621	4.398	-7.373
16	test (3d)	-2.054	6.609	-0.824	4.155	-4.34	-7.906
17	episode 9						
18	validation (1st)	1.788	5.524	5.222	5.104	2.628	3.2
19	validation (2nd)	1.487	5.104	4.956	4.564	1.122	2.588
20	validation (3d)	1.042	4.412	4.832	4.156	0.924	2.254
21	test (1st)	-6.735	12.367	8.227	8.159	-6.424	-9.802
22	test (2nd)	-4.746	8.159	10.751	9.661	-12.017	-2.804
23	test (3d)	-7.544	8.859	6.262	7.315	-6.161	-6.113

Далее для того, чтобы убедиться в том, что стратегия действительно работает, было принято решение провести тестирование на более ранних периодах, так как стратегия должна определять закономерности вне зависимости от конкретного периода времени. Решено было взять 2 предыдущих года до 2019 года, на котором происходило обучение и валидация. Результаты для 2017 года приведены в таблице 4.3, а для 2018 года – в таблице 4.4.

Таблица 4.3 – Результаты валидации и тестирования за 2017 год

	A	B	C	D	E	F	G
1		MA	MOM			MA, ROC, MOM	
2		batch 32	batch 32	batch 16	batch 8	batch 32	batch 16
3	episode 5						
4	validation	1.074	4.938	5.776	4.874	1.866	0.312
5	validation	0.556	4.106	3.75	3.322	1.294	-0.391
6	validation	0.159	3.874	3.366	1.696	1.232	-0.56
7	test (1st)	-4.96	19.35	22.82	24.01	1.44	3.54
8	test (2nd)	-2.14	21.65	22.18	23.9	-0.25	14.4
9	test (3d)	6.05	24.6	23.83	29.7	17.18	-1.42
10	episode 7						
11	validation	2.438	4.858	4.78	4.6	2.752	3.878
12	validation	0.668	4.714	4.626	3.44	0.158	1.928
13	validation	0.469	4.622	4.622	1.616	-0.407	1.406
14	test (1st)	-1.15	20.52	24.62	24.4	21.89	15.91
15	test (2nd)	2.38	21.8	19.38	21.08	0.77	10.14
16	test (3d)	4.52	24.56	24.56	21.21	12.72	11.47
17	episode 9						
18	validation	1.788	5.524	5.222	5.104	2.628	3.2
19	validation	1.487	5.104	4.956	4.564	1.122	2.588
20	validation	1.042	4.412	4.832	4.156	0.924	2.254
21	test (1st)	4.7	20.17	17.73	18.52	6.67	16.22
22	test (2nd)	1.77	18.52	25.48	22.46	13.12	1.07
23	test (3d)	3.93	17.88	26.3	21.05	1.13	21.33

Как видно по результатам за 2017 год, прибыль для всех типов индикаторов в стратегиях выросла, а это значит, что в это году шаблонов изменения тренда, как в 2019 году, было больше. Тенденция сохранилась – индикатор MOM выдает лучшие значения и прибыль уже выходит на многих параметрах около 21 – 24 ¥, что почти в 2 раза больше, чем на данных за 2020 год.

Что касается индикатора MA и совместного использования MA, MOM, ROC, то MA выдает почти всегда положительные, но относительно MOM очень малые значения прибыли, а комбинация индикаторов имеет достаточно сильный разброс значений, чтобы говорить о ее успешности даже на 2017 году.

По результатам тестирования на данных за 2018 год видно, что прибыль для всех параметров хоть и уменьшилась по сравнению с 2017 годом, однако значения зато все положительные и колеблются в интервале от 5 до 15 ¥.

Никакой корреляции результатов тестирования с порядком в валидации нет. У индикатора MOM наименьший интервал разброса значений прибыли.

Таблица 4.4 – Результаты валидации и тестирования за 2018 год

	A	B	C	D	E	F	G
1		MA	MOM			MA, ROC, MOM	
2		batch 32	batch 32	batch 16	batch 8	batch 32	batch 16
3	episode 5						
4	validation	1.074	4.938	5.776	4.874	1.866	0.312
5	validation	0.556	4.106	3.75	3.322	1.294	-0.391
6	validation	0.159	3.874	3.366	1.696	1.232	-0.56
7	test (1st)	12.78	12.48	12.89	8.51	14.25	8.98
8	test (2nd)	13.06	15.69	14.37	12.42	10.28	10.64
9	test (3d)	7.48	12.24	11.82	10.12	8.16	15.26
10	episode 7						
11	validation	2.438	4.858	4.78	4.6	2.752	3.878
12	validation	0.668	4.714	4.626	3.44	0.158	1.928
13	validation	0.469	4.622	4.622	1.616	-0.407	1.406
14	test (1st)	10.52	16.41	8.9	13.84	10.56	12.16
15	test (2nd)	15.08	11.99	12.79	12.12	7.87	15.43
16	test (3d)	4.74	14.63	14.63	12.83	5.57	12.09
17	episode 9						
18	validation	1.788	5.524	5.222	5.104	2.628	3.2
19	validation	1.487	5.104	4.956	4.564	1.122	2.588
20	validation	1.042	4.412	4.832	4.156	0.924	2.254
21	test (1st)	12.35	15.65	11.38	11.7	8.76	12.38
22	test (2nd)	9.04	11.7	8.85	9.31	6.5	15.81
23	test (3d)	9.08	13.2	13.68	16.5	5.83	14.06

После дополнительного и необходимого тестирования на данных за 2017 и 2018 год можно с уверенностью сказать, что тенденция сохранилась – на индикаторе MOM достигаются лучшие значения прибыли, тем самым подтверждая, что стратегия, основанная на нем, работает. В остальных же случаях еще нужно дорабатывать и искать иные комбинации индикаторов с подбором лучших параметров (количество часов, батчей, коэффициенты в формулах и другое).

Таким образом, в итоге есть разработанная и протестированная торговая стратегия на основе глубокого обучения с подкреплением, основным индикатором в которой для описания состояния среды является индикатор MOM.

ГЛАВА 5

ТОРГОВЛЯ НА ФОРЕКС

5.1 Открытие демо-счета

Разработанная стратегия может быть в дальнейшем применена для торговли в реальном времени, так как реализованный торговый агент должен лишь иметь историю котировок для корректного предсказания на основании состояния среды, которое рассчитывается именно на основании предыдущих данных. Поэтому далее алгоритм можно будет использовать в качестве «консультанта» при торговле на рынке Форекс. Но для интернет-трейдинга необходимо открыть счет, на котором будут средства для совершения сделок.

Я решил открыть демо-счет (именно на нем я ранее тестировал разные подходы и стратегии) на сервере Альпари. Данная компания – мировой лидер рынка Форекс. Она предоставляет на своем сайте (<https://alpari.com>) самые разные возможности для интернет-трейдинга, в том числе и открытие демо-счета.

Сперва необходимо пройти регистрацию на сайте, а уже после, авторизовавшись в своем личном кабинете, можно открывать счета как с возможностью пополнять реальными средствами, так и учебные.

На рисунке 5.1 представлены основные характеристики открытого мной демо-счета.

Тип счета:	demo.ecn.mt5	Баланс:	1 006.75 USD
Торговый сервер:	Alpari-MT5-Demo	Средства:	1 006.75 USD
Пароль для входа в торговую платформу:	***** Изменить	Пополнить счет	
Торговая платформа:	MetaTrader 5		
Кредитное плечо:	1:100 Изменить		
Показывать счет на главной странице:	Да Нет		
Удаление счета:	Удалить счет		

Рисунок 5.1 – Настройки торгового демо-счета

Основной интерес здесь представляет характеристика «кредитное плечо», которая показывает, во сколько раз больше средств будет использовано при проведении сделки по сравнению с количеством средств на счете. В данном случае, так как кредитное плечо указано 1:100, то если со счета для совершения

сделки будет взиматься 1000 USD, то в сделке будет участвовать сумма, равная $1000 \cdot 100 = 100000$ USD.

Также к данному счету выдается уникальный логин и пароль для торговой платформы *MetaTrader 5* от Альпари, которые необходимы для подключения к счету в мета трейдере.

5.2 Использование метатрейдера

После создания торгового демо-счета необходимо как-то к нему подключиться для торговли на Форекс. Для этого существуют как мобильные приложения с ограниченным функционалом, так и полноценные платформы вроде *MetaTrader 4* и *MetaTrader 5*. Именно эти полноценные многофункциональные системы позволяют подключаться к торговым счетам, торговать и тестировать свои стратегии.

Я решил использовать *MetaTrader 5* от Альпари. После установки приложения с сайта Альпари и подключения к демо-счету, отображается окно, которое изображено на рисунке 5.2.

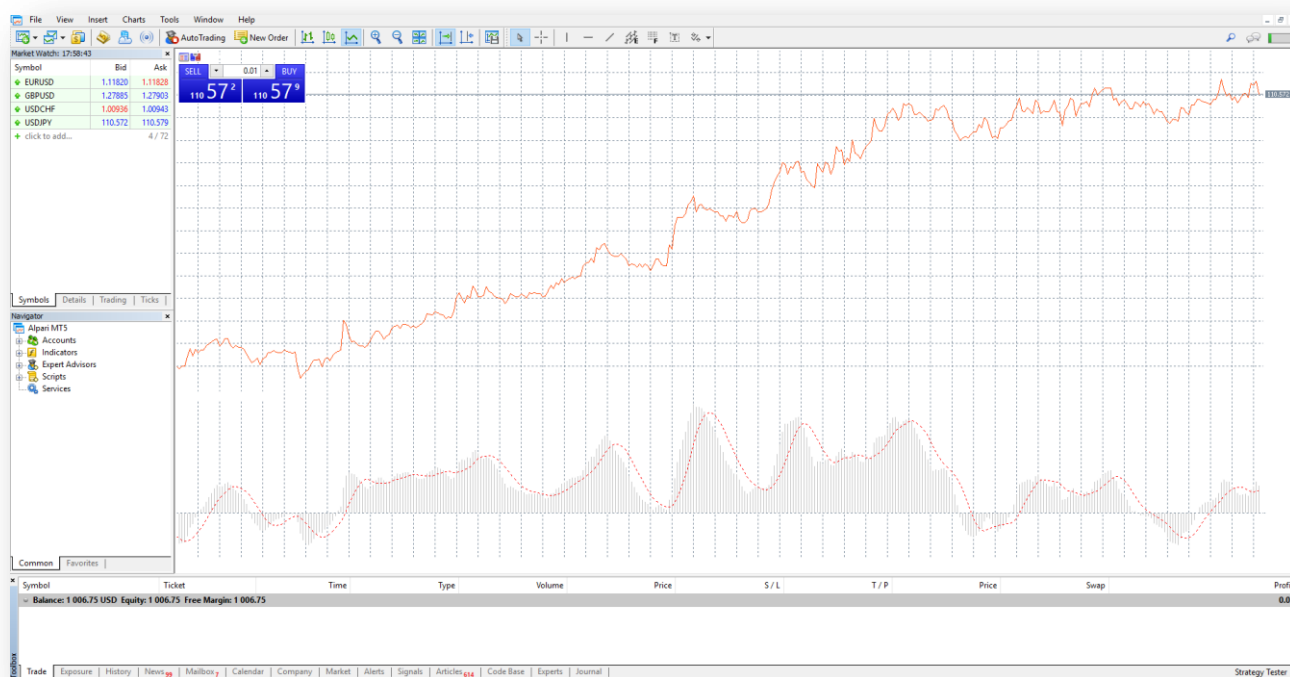


Рисунок 5.2 – Вид мета трейдера с сайта Альпари

Самыми важными элементами тут являются кнопки открытия и закрытия сделки, график изменения курсов, тестер стратегий. По умолчанию Volume (объем) стоит $0.01 = 1000$ USD, так как единица объема составляет 100000 USD. Так можно регулировать, на какой сумме будет происходить торговля в

конкретной сделке. Например, если кредитное плечо 1:100, а объем выставлен 0.1, то со счета точно будет списано для открытия сделки 100 USD, а также комиссия, которая высчитывается по формуле:

$$\text{Комиссия} = \frac{\text{Volume} * \text{Contract} * \text{RateToUSD}}{1\,000\,000} * 16\,USD * 2 \#(5.1)$$

где *Volume* – объем сделки;

Contract – размер контракт;

RateToUSD – курс базовой валюты к USD.

Например:

- объем сделки – 0.01;
- размер контракта — 100000 USD;
- курс USD/JPY на момент открытия позиции — 110 (но так как базовая валюта – доллар, то учитываться он не будет).

$$\text{Комиссия} = (0.01 \times 100000 \times 1 / 1\,000\,000) \times 16\,USD \times 2 = 0.032\,USD.$$

В дальнейшем в планах использование и синхронизация работы метатрейдера и разработанного алгоритма, чтобы данные, которые обновляются каждый час, обрабатывались и отображались сразу в двух местах.

Есть минимум 2 варианта работы:

1. Синхронизация работы метатрейдера и алгоритма торговли, как приведено на рисунке 5.3;
2. Использование API для встраивания своих скриптов для тестирования и торговли внутри платформы.

В обоих случаях принятие решения за открытие или закрытие сделки лежит на моем торговом агенте, разработанном в предыдущей главе, о взаимодействие с данными разное. В любом случае, это как вариант для дальнейшего развития проекта.

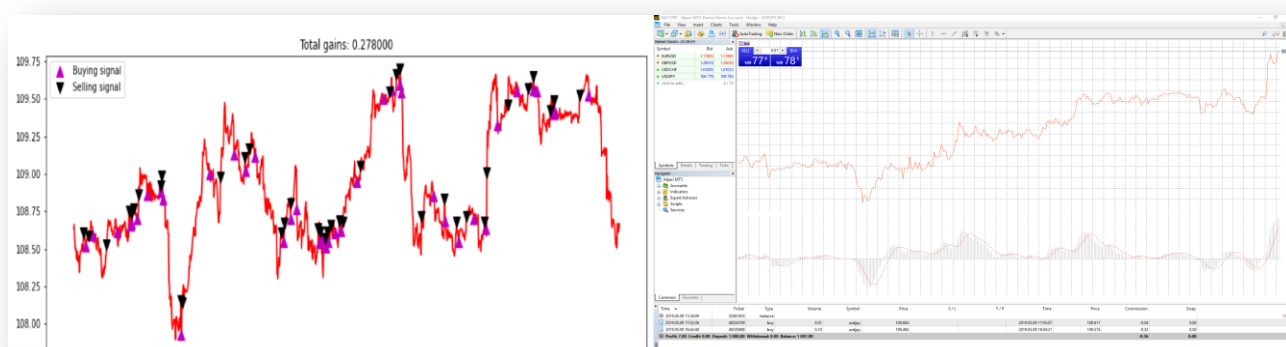


Рисунок 5.3 – Работа алгоритма и метатрейдера

5.3 Оценка успешности стратегии

Исходя из данных тестирования, полученных в предыдущей главе, можно сделать предположение, что в год на 1 USD удастся получить общую прибыль минимум 11 йен. При курсе USD/JPY примерно 110 получается, что это примерно 1/10 доллара. То есть выходит, что прибыль 10 процентов годовых при торговле лишь на 1 USD. Но нужно учитывать еще следующие факты:

1. Двойная комиссия при открытии сделки в большинстве метатрейдеров. Если отталкиваться от того, что комиссия в целом мала 0.00032 доллара на 1 долларе, равном объему контракта, то это еще не так ощутимо;

2. Ставка рефинансирования, которая меняется ежедневно и из-за этого некоторые сделки закрываются вовсе не по тому курсу, по которому ожидалось (тут речь идет про своп). Однако с учетом того, что большая часть сделок открывается и закрывается в пределах дня, я пренебрег данным фактом и не учитывал в данный момент при расчете прибыли;

3. Несмотря на то, что используя кредитное плечо можно увеличить суммы (везде выше использовался объем равный 1 доллару) в 100 раз, нужно иметь в виду, что и в случае потерь затраты будут увеличены в соответствующее количество раз.

4. В целом я рассчитываю на 5% процентов годовых с учетом всех пунктов выше, а используя кредитное плечо, которое и является ключевым моментом при торговле, можно увеличить это значение в разы, что явно лучше ставок по депозитам на текущий момент в нашей стране в валюте (максимум 2.5 %).

Таким образом, хоть риски и присутствуют, разработанная стратегия после некоторых улучшений и дополнительного тестирования подходит для того, чтобы быть использованной для торговли на рынке Форекс и как основа для улучшенных стратегий.

ЗАКЛЮЧЕНИЕ

В процессе выполнения данной работы были получены следующие результаты:

- изучены основы обучения с подкреплением;
- изучены и определены инструменты для достижения поставленной цели;
- написан программный код для реализации стратегии торговли;
- выполнено обучение и тестирование стратегии;
- изучены способы возможного улучшения модели и торгового агента с целью максимизации прибыли;
- получен практический опыт в анализе экономических данных, таких как курс валютной пары USD/JPY.

В результате проделанной работы можно сказать, что торговая стратегия жизнеспособна, но требует дополнительного тестирования перед применением в реальной торговле на Форексе.

Также есть возможность синхронизировать действия торгового агента с реальной торговой платформой, но это уже будет сделано после дополнительного обучения и тестирования модели и подбора лучших параметров.

На примере разработанной торговой стратегии продемонстрировано, что глубокое обучение с подкреплением может быть эффективно применено как один из подходов для разработки торговой стратегии на Форексе.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Лин, К. Дейтрейдинг на рынке Forex. Стратегии извлечения прибыли /Кетти Лин; Пер. с англ. - М.: Альпина Бизнес Букс, – 2007. - 240 с.
2. Найман, Э. Л. Малая Энциклопедия Трейдера /Эрик Л. Найман - К. ВИРА-Р Альфа Капитал, – 1999. - 236 с.
3. Торговля на Форекс // Сайт брокера [Электронный ресурс]. – 2021. – Режим доступа <https://admiralmarkets.com/ru/education/articles/forex-basics>. - Дата доступа: 28.03.2021.
4. Машинное обучение // Экономический портал [Электронный ресурс]. – 2019. –Режим доступа <https://financial-hacker.com/build-better-strategies-part-4-machine-learning/>. - Дата доступа: 15.11.2020.
5. Richard S. Sutton Reinforcement Learning: An Introduction / Richard S. Sutton, Andrew G. Barto; The MIT Press, – 2017. - 427 с.
6. Tatsat, H. Machine Learning and Data Science Blueprints for Finance / Hariom Tatsat, Sahil Puri, and Brad Lookabaugh; O'Reilly, – 2021. - 412 с.
7. Экономический календарь // Новостной экономический портал [Электронный ресурс]. – 2020. –Режим доступа <https://www.investing.com/>. - Дата доступа: 10.10.2020.