

РЕФЕРАТ

Магистерская диссертация содержит 42 страницы, 2 таблицы, 15 рисунков. Список использованных источников содержит 9 позиций.

АЛГОРИТМИЧЕСКАЯ ТОРГОВЛЯ, ОБУЧЕНИЕ С
ПОДКРЕПЛЕНИЕМ, МЕТОД Q-ОБУЧЕНИЯ, РЕКУРСИВНЫЙ
АЛГОРИТМ, ОБРАБОТКА ДАННЫХ, ТЕХНИЧЕСКИЕ
ИНДИКАТОРЫ.

Магистерская диссертация посвящена проектированию рекурсивного алгоритма для обучения торговых систем в алгоритмической торговле. В качестве данных используются открытые источники с котировками в разные моменты времени. Системы стараются найти оптимальные параметры для торговли в определенные временные ряды.

Для решения поставленной задачи используется метод Q-обучения торговых систем на основе технических индикаторов. Модель спроектирована на языке Python с использованием пакетного решения Anaconda. При построении модели учтены особенности рыночной торговли на фондовых рынках и определены показатели, которые помогают приблизить работы алгоритма к торгам в реальном времени. Построенная модель показывает хорошие результаты на исторических данных и в будущем может быть использована в работе профессиональных участников рынка.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ОСНОВНАЯ ЧАСТЬ	6
1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	7
1.1 Машинное обучение в алгоритмической торговле	7
1.2 Цель и задачи работы.....	10
1.3 Обзор литературы	10
1.4 Практическая ценность и новизна работы.....	11
1.5 Актуальность работы.....	12
1.7 Деревья решений.....	15
1.8 Рекуррентные нейронные сети.....	17
1.9 Свёрточные нейронный сети	20
1.10 Обучение с подкреплением.....	22
1.11 Модель ARIMA	25
1.12 Многослойный персептрон (MPC).....	26
1.13 Технические индикаторы	27
2. ПРАКТИЧЕСКАЯ ЧАСТЬ	31
2.1 Симуляция работы биржевого робота	31
2.2 Алгоритм работы биржевого робота	32
2.3 Описание структуры данных.....	33
2.4 Реализация алгоритма	34
2.5 Результаты работы	34
2.6 Выводы по работе алгоритма	39
ЗАКЛЮЧЕНИЕ	41
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	42

ВВЕДЕНИЕ

Данная работа будет посвящена применению алгоритмов машинного обучения и обработки данных в обучение торговых систем. На данном этапе биржевого дела, алгоритмическая торговля является одним из главных инструментов институциональные маркетмейкеров, а также для спекулятивных участников рынка. Многие физические и юридические лица заинтересованы в получении выгоды от финансовых рынков, поэтому постоянно усовершенствуют свой арсенал алгоритмов торговли и ищут новые решения.

В данной работе мы будем рассматривать алгоритмическую торговлю со стороны моделирования работы биржевого робота. Основная цель биржевого робота – это получение прибыли и сохранение её на определенном промежутке времени.

Главным объектом нашего исследования будет биржа. Предметом нашего исследования будет стратегия рекурсивного обучения торговой системы. Под рекурсивным обучением мы подразумеваем нахождение оптимальных параметров торговой системы, которые помогут нам достичь максимальной прибыли и минимизировать убытки. Поведение торговой системы будет моделироваться с помощью алгоритмов машинного обучения, с помощью машинного обучения мы будем определять наилучшие стратегии и набор параметров.

Для построения деревьев решений используются генетические алгоритмы, сами деревья используют для классификации данных. Алгоритм получает набор входных признаков и предсказывает дальнейших ход, который нужно совершить системе, чтобы получить прибыль.

Генетические алгоритмы применяются для построения деревьев решений и ансамблей деревьев решений. Деревья решений используются как классификаторы, по набору входных признаков они предсказывают действие, которое необходимо совершить роботу в каждый момент времени.

Технические индикаторы являются входными признаками для торговой системы, это очень важный параметр для алгоритма, так как на фоне выбранного индикатора или пары индикатор, строится алгоритм работы робота.

Обычно в алгоритмической торговле используется метод машинного обучения с учителем. Во многих алгоритмах применяются генетические алгоритмы для создания сложных деревьев с различными параметрами в вершинах, что приводит к переобучению модели.

Главной идеей нашей работы будет использования метода рекурсивного обучения торговой системы – на каждом историческом промежутке мы будем запускать несколько торговых систем и находить наилучшие n систем с наилучшими параметрами, которые показывают максимальную доходность и минимальные просадки внутри торгового промежутка. После мы будем использовать найденные параметры на данных, которые торговая система еще не видела и заново отбирать наилучшие алгоритмы и настройки для торговых роботов.

Основной ценностью данной работы является возможность создать самообучаемую систему, которая будет являться продуктом для биржи или инвестиционных фондов.

ОСНОВНАЯ ЧАСТЬ

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Машинное обучение в алгоритмической торговле

Прогнозирование будущих цен активов является сложной задачей и обуславливается это непредсказуемыми изменениями в экономике. Одним из главных параметров прогнозирования цен является временной ряд. Высокая волатильность рынка является главным фактором для тщательного изучения результатов работы регрессионных моделей.

В последнее время алгоритмическая торговля развивается с высокой скоростью, так как разрабатываются новые алгоритмы машинного обучения, обработки данных и их анализа. Участники рынка активно конкурирует за технологии и пытаются использовать автоматизированные системы для своей работы.

Фондовый рынок является открытой торговой площадкой для совершения операций с ценными бумагами различных эмитентов. Реализация прогнозирования будущей цены актива является важной задачей для широкого круга компаний.

С зарождения фондовых рынков участники торгов искали разные способы для получения прибыли. Основными из них были – анализ котировок и графиков того или иного инструмента, создание и использование индикаторов и алгоритмов предсказания будущей цены актива. В конце концов, сформировались основные подходы к анализу рынка: технический анализ, фундаментальный анализ и технологические методы. В совокупности эти подходы давали очень хорошие результаты для предсказания будущего движения цены. При этом всегда остается актуальной задача повышения точности прогноза, для её решения в данный момент большинство компаний используют методы машинного обучения, включая алгоритмы глубокого обучение.

В торговле на фондовом рынке нейронные сети могут имитировать действия агента, выполняющего те или иные задачи на рынке акций. Важной

задачей при построении нейронных сетей является предобработка данных, и многие исследователи решают ее по-своему, предлагая новые методы, так как не существует устоявшихся практик однозначной предобработки входных массивов данных ввиду большого разнообразия как самих данных, так и ее типов. Выходные данные в рамках конкретной задачи, напротив, могут быть однозначно определены, принимая вид финансовых или экономических показателей. Что касается оценки результатов, то наиболее часто употребляемым методом является сравнение результатов работы алгоритма со значениями из тестовой выборки.

Традиционным способом применения машинного обучения является обучение с учителем. В качестве входных данных используется цена актива за определенный промежуток времени. Целевая переменная алгоритма это изменение цены актива. В прогнозировании цены часто используются алгоритмы изучения деревьев решений машины опорных векторов.

В таких алгоритмах входными признаками являются технические индикаторы рынка. Обилие технических индикаторов дает возможность манипулировать конфигурациями параметров и получать наилучшие результаты для каждого промежутка времени. Алгоритмы позволяют извлекать тренды рынка из неструктурированных данных. Обучение с подкреплением является одним из быстроразвивающихся методов исследования котировок на финансовых рынках.

Генетические алгоритмы успешно применяются в исследования показателей рынка. Их можно использовать для построения ансамблей алгоритмов машинного обучения для повышения точности работы отдельных алгоритмов.

Анализ временных рядов представляет собой интересную область исследования с большим количеством прикладных задач в бизнесе, экономике, финансах и компьютерных науках. Целью анализа временного ряда является изучение его динамики, построение модели, описывающей структуру данных и, наконец, прогноз будущих значений ряда. Крайне важно

построение эффективной модели с получением максимально возможной точности.

Если разобрать любую биржу на группу параметров, то можно выделить основные фрагменты - биржевой стакан [1] – структура, описывающая выставленные участниками рынка заявки. Рыночные свечи характеризуются ценой актива в начале периода (open price), ценой актива в конце периода (close price), максимальной ценой за период (high price) и минимальной ценой (low price). К этим параметрам можно добавить объемы покупок и продаж отдельных инструментов за данный период.

На рис. 1.1 показан график цен на акции компании Apple за 2019 год.



Рис. 1.1 – График цен на акции компании Apple за 2019 год.

В работе мы будем рассматривать свечные график, так как они содержат расширенные параметры торгов. Технические индикаторы используются для численного описания состояние биржи в каждый момент времени, они рассчитываются на основе данных свечных графиков. Для алгоритмов машинного обучения технические индикаторы выполняют функцию входных признаков.

1.2 Цель и задачи работы

Главной целью работы является создание рекурсивного алгоритма обучения торговых систем. Данный алгоритм поможет максимизировать полученную прибыль на определенной промежуток времени и минимизировать потери.

В данной работе мы будем решать следующие задачи:

- Изучения предметной области алгоритмической торговли.
- Создание рекурсивного алгоритма обучения торговой системы.
- Сравнение результатов стандартных алгоритмов машинного обучения в алгоритмической торговле с рекурсивным алгоритмом обучения торговой системы.
- Анализ работы алгоритма на реальных биржевых исторических данных и оценка его качества.

1.3 Обзор литературы

Одним из традиционных методов машинного обучения к задачам алгоритмической торговли является обучение с учителем. В качестве целой переменной, то есть переменной, значение которой предсказывается алгоритмом, выступает изменение цены актива за некоторый период времени [2] или, альтернативно, волатильность рынка [3].

За входные признаки принимают технические индикаторы рынка. В самом начале исследования применения машинного обучения к прогнозированию событий на финансовых рынках наиболее часто применяются простые и хорошо изученные алгоритмы - деревья решений машины [4] опорных векторов [5]. С успешным применением нейронных сетей в задачах обработки естественного языка, они стали применяться и в области алгоритмической торговли для предсказания движения цен на рынке под влиянием появляющихся политических или экономических новостей [6],

то есть появляется возможность извлекать сигналы рынка из неструктурированной информации.

В последние годы обучение с подкреплением является одним из быстроразвивающихся разделов машинного обучения, с помощью этого метода достигнуты такие результаты, как, например, способность компьютера играть в различные игры на симуляторе Atari на уровне, превосходящем уровень человека, ничего изначально не зная о правилах игры [7]. Обучение с подкреплением применяются к задачам алгоритмической торговли, но более распространённым методом было применения обучения с учителем, так как это позволяло достичь лучших результатов, однако в последнее время в связи с успехами, достигнутыми с помощью обучения с подкреплением в других предметных областях, сильно возрос интерес к исследованию возможности применения этого класса алгоритмов к задачам алгоритмической торговли [8]. К широкому спектру прикладных задач применяются генетические алгоритмы, которые также используются в алгоритмической торговле. Существует и применение генетических алгоритмов к обобщенным задачам, например таким, как построение ансамблей алгоритмов машинного обучения для повышения точности работы отдельных алгоритмов [9] и построения деревьев решений [10]. Генетические алгоритмы показывают наилучшую эффективность в анализе временных интервалов.

Метод генерации торговых правил заключается в том, что с помощью генетических алгоритмов создаются функции от технических индикаторов рынка, которые предсказывают изменение цены актива наиболее точно.

1.4 Практическая ценность и новизна работы

В работе будет рассматриваться рекурсивный алгоритм обучения торговой системы. Это нестандартный подход к применению машинного обучения, так как каждый алгоритм будет рассматривать как отдельная торговая система. Для каждой системы мы можем использовать разные

алгоритмы машинного обучения, и из всей выборки алгоритмов выбрать наилучшие параметры для «прогонки» их на следующем временном интервале.

В данной работе мы будем применять линейную регрессию для предсказания финансовых возвратов. Для предсказания цен мы будем использовать обучение с подкрепление Q-обучение.

Теоретической ценностью данной работы является развитие исследования по применению методов обработки данных и алгоритмов машинного обучения к алгоритмической торговле. Исследование метода рекурсивного обучения может хорошо помочь в получении прибыли и уменьшение издержек на финансовых рынках.

Также данный метод может быть использован в других предметных областях, исследования в которых похожи на задачи стратегического поведения.

Одна из ценностей данной работы – конструирование программных продуктов для торговли на бирже с использованием исследуемых алгоритмов в работе.

1.5 Актуальность работы

Развитие методов для алгоритмической торговли активно развивается в последние годы. Это происходит благодаря следующим факторам: развитию технологий обработки данных и анализа данных, изучения применения методов машинного обучения к фондовым рынкам и близким предметным областям, структурированию больших объемов информации и использование новых методов хранения данных.

Конкуренция является ключевым двигателем развития прогресса, так как участники рынка состязаются в битве автоматизированных систем, которые используют разные алгоритмы с применением машинного обучения.

В связи с этим у участника рынка есть очень большой интерес к использованию новых методов получения прибыли. Участниками рынка являются не только частные компании и хедж-фонды, цель которых увеличить свой капитал, но и научное сообщество, так как развитие методов машинного обучения и обработки данных может поспособствовать новым открытиям и в других предметных областях.

1.6 Линейные модели

Линейные модели являются одними из наиболее полезных классов гипотез. Обучающие алгоритмы, которые широко применяются в алгоритмической торговле, строятся на линейных предсказательных переменных, потому что в большинстве случаев они могут быть очень хорошо натренированы. Данные модели справляются в условиях шумных финансовых данных и имеют сильную связь с финансовым миром. Данные алгоритмы отлично интерпретируются и хорошо укладываются в данные, обеспечивая им хорошую опорную линию.

Уже более 200 лет нам известна линейная регрессия, которую применяли к астрономии и с помощью неё анализировали статистические свойства. С того времени для усвоения параметров был разработан целый ряд расширений, который адаптировал линейную регрессию и базовый метод обычный наименьших квадратов (ordinary least squares).

Среди таких расширений есть обобщенные линейные модели (generalized linear models, GLM), которые расширяют область применения, допуская результирующие отклики, переменные, которые обычно приводят к распределению ошибок, отличному от нормального распределения. Такие модели включают в себя логистические модели для категориальных переменных, которые появляются в задачах классификации.

Робастные методы оценивания позволяют поддержать статистический вывод только в тех случаях, когда данные нарушают исходные допущения, в

основном, из-за корреляции во времени или между наблюдениями. Это происходит в панельных данных, которые содержат повторяющиеся наблюдения по таким единицам измерения, как исторические финансовые возвраты в активах.

Усадочные методы нужны для улучшения предсказательный результатов линейных моделей. Используя штраф за сложность, который смещает коэффициенты модели, они снижают дисперсию и улучшают вне выборочную предсказательную результативность.

Линейные модели применяются к регрессионным задачам, а также к задачам классификации в целях статистического вывода и предсказания. Данные модели активно используются для оценивания стоимости активов в определенные временные интервалы. С помощью линейной регрессии выявляют факторы, которые стимулируют финансовые возвраты, особенно в области риск-менеджмента, а также для предсказания возвратов на разных временных рядах. Классификационные задачи направлены на анализ ценовых прогнозов.

Множественная регрессионная модель задается линейную функциональную связь между непрерывным входным параметром и p входными параметрами, которые имеют структуру любого типа, но могут требовать предобработки.

Линейная регрессионная модель содержит следующую форму для одного единственного экземпляра выхода y и входного вектора $x^T = [x_1 \dots x_p]$ и ошибки ε , которая показана в формуле 1.1:

$$y = f(x) + \varepsilon = \beta + \sum_{j=1}^p \beta_j x_j + \varepsilon \quad (1.1)$$

Значение коэффициента β является частичным, средним эффектом переменной x_j на выходе при неизменности всех остальных переменных.

Матрица регрессоров обычно обозначается x это матрица значений объясняющих переменных множества объектов. Каждая строка в таком случае

представляет собой отдельный объект, а последующие столбцы соответствуют переменным и их конкретным значениям для этого объекта.

1.7 Деревья решений

Одним из классических алгоритмов машинного обучения являются деревья решений. Существует большое количество разновидностей деревьев, которые применяются в задачах классификации, регрессии. В алгоритмической торговле деревья решений обычно рассматриваются как классификаторы для численных признаков.

Дерево решений представляет из себя связный граф без петель и кратных рёбер. Вершины дерева выполняют разные функции, в них записаны метки возможных классов, по которым ведут классификацию.

В отличие от линейных моделей деревья решений используют подход усваивания и последовательного применения набора правил, которые разделяют точки данных на подмножества, а затем делают одно предсказание их каждого подмножества. Выходные значения являются основой для подмножества тренировочных образцов, которые возникают в результате применения заданной последовательности правил.

Деревья решений предсказывают вероятность по относительным частотам, либо по значению преобладающих классов. Это их главное отличие от регрессионных моделей, так как там вычисляется предсказание по среднему значению результатов для имеющихся точек данных.

Бинарные деревья представлены логикой, где корень является отправной точкой для всех образцов, а узлы — это места применения ряда правил принятия решений. Данные распределяются по ребрам дерева по мере их разбиения на меньшие подмножества до прибытия в узлы, где модель делает предсказание. В дереве решений основной путь от корня к листьям создает четкое понимание того, какие признаки и их значения приводят к конкретным модельным решениям.

При тренировочном запуске алгоритм сканирует признаки и для каждого признака пытается найти отсечку. Данная отсечка разбивает данные и минимизирует потери, возникающие в результате работы алгоритма.

На рис. 1.2 можно увидеть дерево решений, построенное слева-направо.

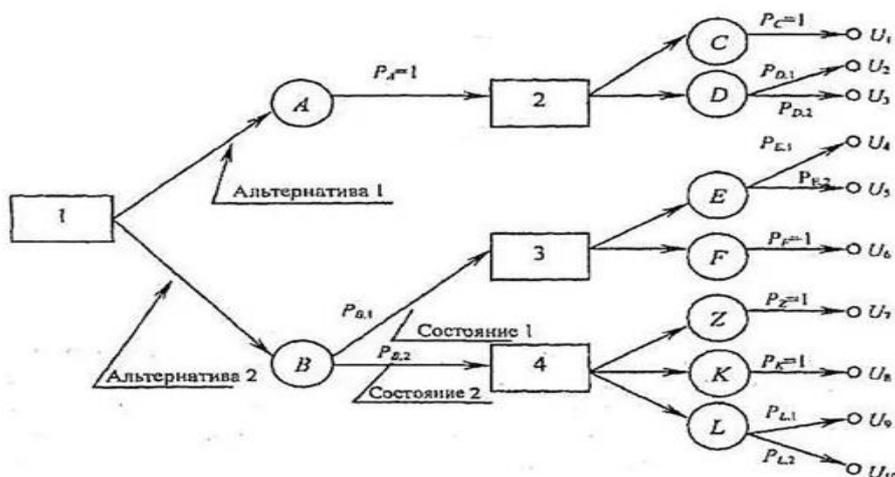


Рис. 1.2 – Дерево решений

Для построения дерева решений во время тренировочного процесса обучающий алгоритм повторяет процесс разделения признакового пространства на взаимоисключающие и коллективно исчерпывающие участки, каждый из которых представлен листовым узлом.

Минус данного алгоритма в том, что он не может оценивать все возможные разбиения признакового пространства, так как количество возможных комбинации может быть огромным. Для того, чтобы преодолеть данный подход используется рекурсивное бинарное разбиение. Данный процесс является рекурсивным, поскольку он использует группу подмножеств данных, которые были получены в результате предыдущих разбинок. Данный алгоритм является жадным, потому что он не оценивает потери на несколько шагов вперед, а вместо этого подбирает наилучшее правило в форме комбинации признака и порога.

С добавлением все новых и новых узлов в дерево с помощью рекурсивных разбинок, число тренировочных данных сокращается. Для модели дерева работает правило 2^n узлов на каждом уровне, потому что

образцы разбиваются равномерно и дают сбалансированное дерево с равным числом дочерних элементов в каждом узле. Но на практике это является маловероятным, так как число образцов вдоль некоторых ветвей может уменьшаться, а деревья расти до разных уровней глубины вдоль разных путей.

Для предсказания нового наблюдения модель использует правила, полученные ею во время тренировки, о том какому листовому узлу должна быть назначена точка данных, а затем использует моду или среднее тренировочных наблюдений в точке признакового пространства. Чем меньше число тренировочных образцов в заданной точке, тем больше снижается качество предсказания.

Рекурсивное разбиение продолжается до тех пор, пока каждый узел не будет содержать единственный образец и погрешность не будет сведена к нулю.

1.8 Рекуррентные нейронные сети

Рекуррентная нейронная сеть – это последовательная модель, предназначенная для преобразования входной последовательности одной предметной области в выходную последовательность другой. RNN модели отлично подходят для распознавания рукописного текста, речи и машинного перевода.

Данная модель создавалась с возможностью обработки больших последовательных данных и решением задач с распределением их во времени. Модель может обрабатывать только один элемент последовательности на одном временном шаге. Посчитанные параметры передаются на следующий временной шаг, чтобы облегчить вычисления. На рис. 1.3 можно увидеть схему работы рекуррентной нейронной сети.

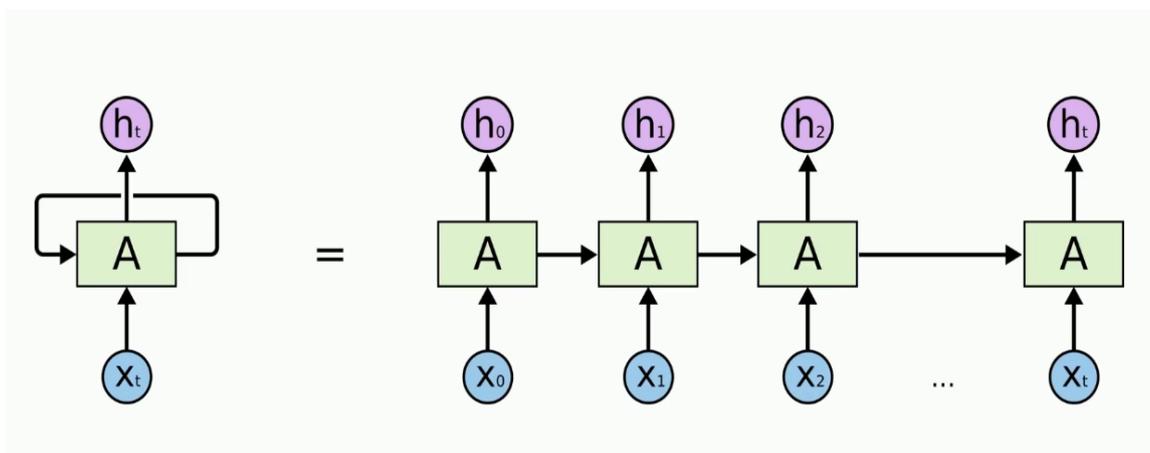


Рис. 1.3 – Модель рекуррентной нейронной сети

Сложность рекуррентной сети заключается в том, что при учёте каждого шага времени становится необходимым создавать новый слой нейронов, что порождает лишние вычислительные сложности модели. Кроме этого, многослойные реализации вычислительно неустойчивы, так как в них зашкаливают веса. При ограничении расчёта фиксированным неустойчивым окном, то полученная модель не будет отображать долгосрочных трендов.

Чтобы решить эту проблему создали специальный нейрон со сложной внутренней структурой для запоминания долгосрочной информации и назвали его Long-short Term Memory (LSTM) (рис. 1.4). Данный нейрон хорошо понимает, насколько долго ему нужно хранить старые данные, когда использовать новые и объединять старую информацию с новой.

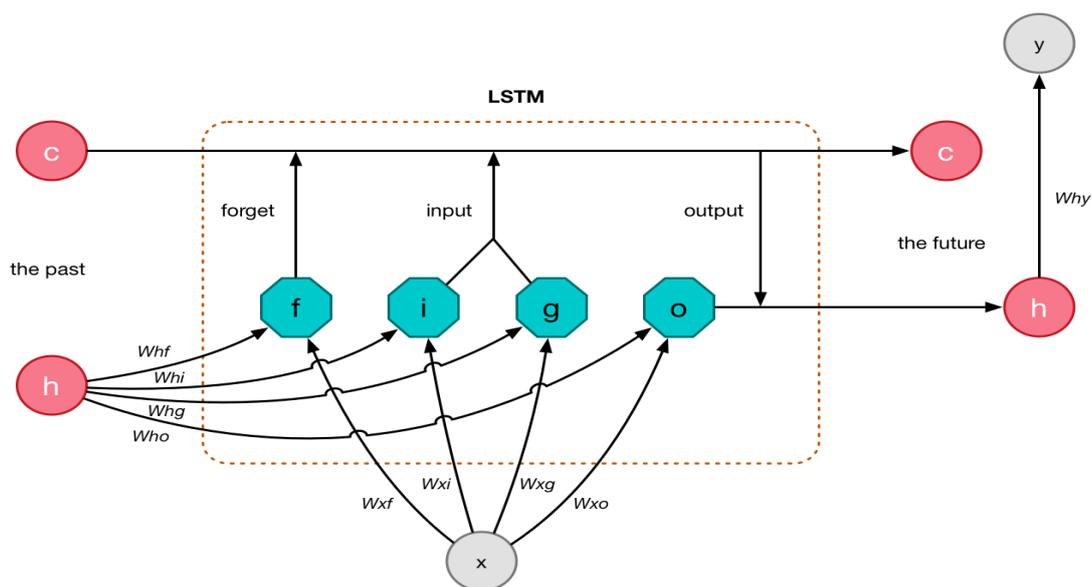


Рис. 1.4 – LSTM

Типичный блок LSTM объединяет четыре параметризованных слоя, которые взаимодействуют друг с другом и состояниями ячейки путем преобразования и передачи вдоль вектора. Обычно эти слои включают входной, выходной и «забывающий» канал, но есть варианты, когда модель может иметь дополнительные механизмы или не иметь некоторые из этих механизмов.

При тестировании связки RNN с ячейками LSTM были достигнуты впечатляющие результаты в области обработки слов и составления предложений. Модель сама может изучить взаимосвязь между символами, чтобы сформировать слова, а позже сформировать предложения, что очень хорошо подойдет для стратегии торговли по новостям источникам.

Существует и расширенная модель RNN, которая называется Sequence-to-Sequence (рис. 1.5). Она состоит из двух фрагментов RNN, кодировщика и декодера. Модель используется в работе с последовательными данными в чат-ботах или персональных помощниках. В данной модели кодировщик изучает контекст поступающих данных, а затем передает полученные знания на сторону декодера через «вектор контекста». Декодер начинает изучать вектор контекста и создает правильные ответы.

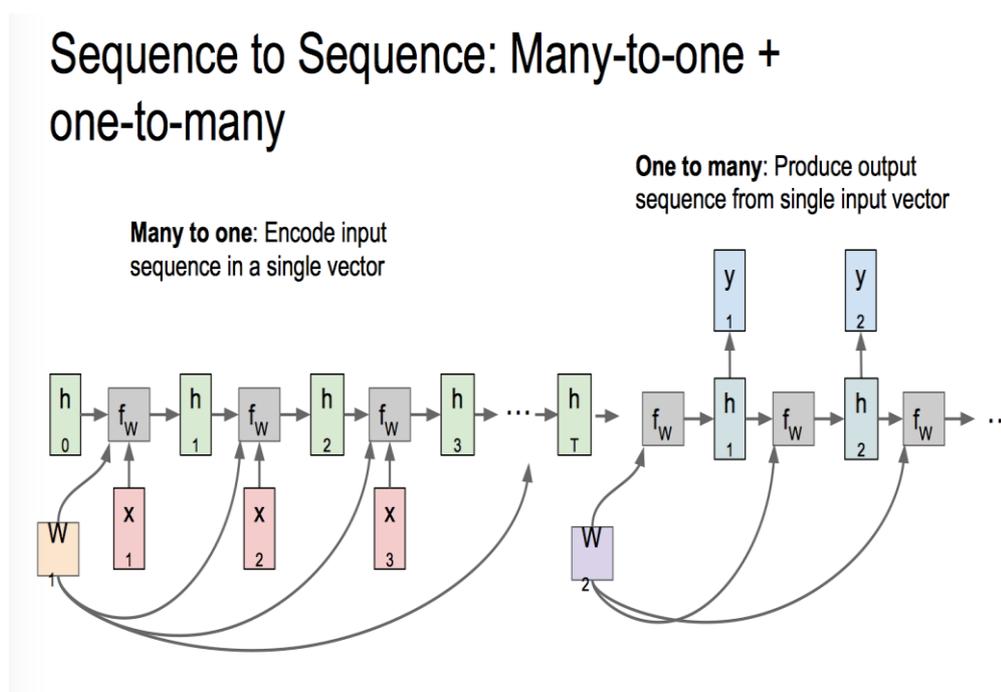


Рис. 1.5 – RNN Sequence-to-Sequence

В алгоритмической торговле прогнозирование RNN больше похоже на модель скользящей средней, так как она не может анализировать и предсказывать все колебания. Модель многослойного персептрон (MLP) намного лучше работает для прогнозирования временного интервала.

1.9 Сверточные нейронный сети

Сверточные нейронные сети (рис. 1.6) названы в честь операции линейной алгебры, называемой сверткой, которая заменяет общее матричное умножение, типичное для сетей обратной связи. Исследования в области архитектуры CNN продолжались очень быстро, и новые архитектуры, которые улучшают производительность на некоторых эталонных продолжат появляться часто. CNN предназначены для изучения иерархических представлений объектов из сеточных данных. Одним из их недостатков является то, что они не изучают пространственные отношения, то есть относительное положение этих объектов.

Convolutional Neural Networks

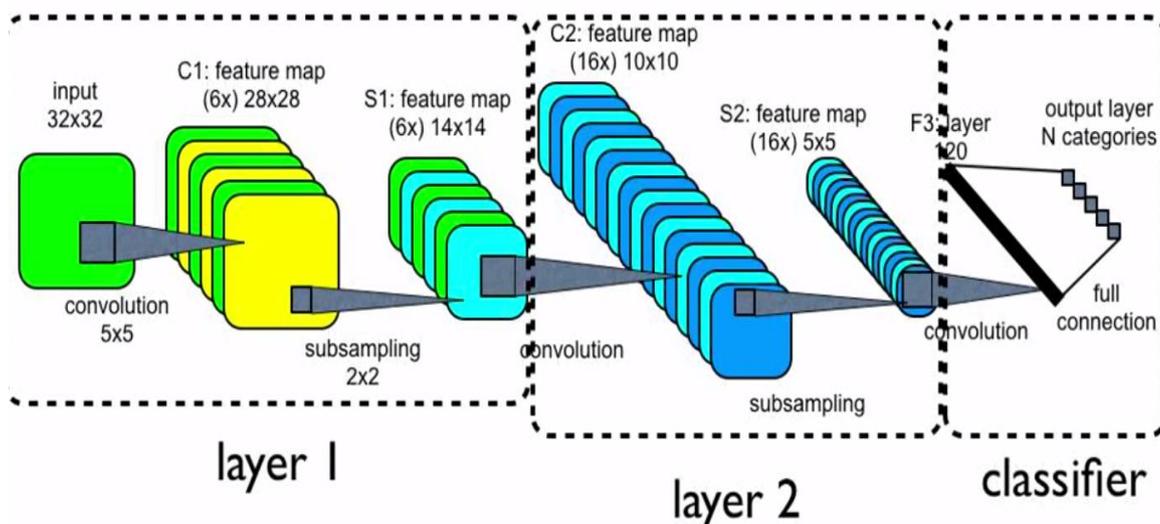


Рис. 1.6 – CNN

CNN очень похожи на RNN, они состоят из блоков, содержащих параметры, называемые весами и смещениями. Сам процесс обучения регулирует эти параметры для оптимизации входных данных сети для данного входного сигнала. Также каждый блок применяет свои параметры к линейной операции над входными данными или активациями, полученным от других блоков, возможно, с последующим нелинейным преобразованием.

Главное отличие CNN сетей от RNN в том, что они кодируют предположение о том, что входной сигнал имеет структуру, наиболее встречающуюся в данных изображениях, где пиксели образуют двумерную сетку, как правило, с несколькими каналами.

Важным элементом кодирования является операция свертки, которая дает CNN возможность объединять структуру в группы. CNN нужно гораздо меньше параметров, чтобы вычисления происходили наиболее эффективно, это подразумевает функциональная связь между входными и выходными данными.

В задачах алгоритмической торговли временные ряды являются одними из главных источников информации, поэтому для их анализа можно использовать CNN модель. Регулярные изменения временных рядов приводят к сеточной структуре данных, как и для структуры изображений. Мы можем использовать архитектуру CNN для одномерных и многомерных временных рядов. Обычно на практике не хватает данных для обучения CNN с нуля со случайно инициализацией.

Подход обучения CNN основан на предварительном обучении на очень большом наборе данных и его цель состоит в том, чтобы свёрточные фильтры извлекали представление об объекте. Второй этап подразумевает использование результатов для переобучения нового CNN или в качестве входных данных для новой сети, которая решает поставленную задачу.

Архитектура CNN использует последовательность свёрточных слоёв для обнаружения структурированных шаблонов, добавляя один или несколько

полностью связанные слоёв для отображения свёрточной активации в классы результатов или значений.

Современные CNN может тренироваться несколько недель на нескольких графических процессорах.

1.10 Обучение с подкреплением

Обучение с подкреплением (RL) (рис. 1.7) это вычислительный подход к целенаправленному обучению, выполняемый агентом, который взаимодействует с типичной стохастической средой, о которой агент имеет неполную информацию. Обучение с подкреплением направлено на автоматизацию процесса принятия решений и достижения долгосрочной цели путем изучения значений состояний и действий по сигналу вознаграждения. Конечная цель состоит в том, чтобы получить политику связей, которая кодирует поведенческие правила и сопоставляет состояния и действия.



Рис. 1.7 – Принцип обучения с подкреплением

Задачи обучения с подкреплением направлены на оптимизацию решений агента, основанных на целевой функции по отношению к окружающей среде. Агент получает информации о среде, ему назначается вознаграждение за действие, и он переводится в новое состояние с учётом вероятностных распределений, о которых сам агент может знать или не знать. Проектированию такой среды требует значительных предварительных усилий, чтобы облегчить целенаправленное обучение агента.

Методы RL отличаются сложностью их состояний и пространств действий, которые могут быть дискретным или непрерывными. Требования от машинного обучения аппроксимации функциональных связей между состояниями, действиями и их значениями является сложность данного алгоритма обучения. Они также направлены на то, чтобы учиться на опыте о том, как принимать меры, которые достигают долгосрочные цели. Агент и окружающая среда взаимодействуют в течение последовательности дискретных временных шагов через интерфейс действий, наблюдений за состоянием и вознаграждений.

Существует несколько фундаментальных подходов к решению задач обучения с подкреплением. Динамическое программирование делает часто нереалистичным предположение о полном знании окружающей среды, но является концептуальной основой для большинства других подходов. Метод Монте-Карло исследует окружающую среду и затраты, и преимущества различных решений путем выборки всей последовательности состояний-действий-вознаграждений. Обучение по временной разнице значительно повышает эффективность самой выборки за счёт обнуления более коротких последовательностей.

Данные подходы для непрерывного пространства состояний или действий очень часто используются в машинном обучении для аппроксимации значений или функции политики. Они позволяют интегрировать контролируемое обучение в модели.

Алгоритм Q-обучения (рис. 1.8) был прорывом в области обучения с подкреплением. Данный алгоритм оптимизирует функцию действия-значения, чтобы приблизить q . Алгоритму не нужно выбирать действия на основе политики, которая подразумевается функцией значений. Однако понятие конвергенции требует от алгоритма, чтобы все пары действий-значений продолжали обновляться на протяжении всего учебного процесса.

Данный алгоритм продолжает улучшать функцию значений действий состояния после случайно инициализации для заданного числа эпизодов. Он

выбирает действие на каждом временном шаге, основанное на жадной политике, и использует скорость обучения, что обновить функции стоимости на основе вознаграждения и её текущей оценки функции стоимости для следующего состояния.

Аппроксимация функций значений вызывает ряд проблем RL алгоритмов, такие как:

- выборки могут быть сильно коррелированы, когда последовательные состояния похожи и распределения поведения состояниям и действиям не является стационарным, а изменяется в результате обучения агента
- при непрерывном состоянии агент не может использовать больше значений, так образом он может потерять нужные параметры для точности результата
- при непрерывном обучении на каждом временном шаге существует только одна выборка, что ограничивает систему для принятия решений



Рис. 1.8 – Алгоритм Q-обучения

1.11 Модель ARIMA

ARIMA является обобщенной моделью ARMA, которая соединяет в себе moving average и autoregressive процессы. В то время как модели экспоненциального сглаживания основаны на описании тренда и сезонности в данных, модели ARIMA направлены на описание автокорреляций в данных. Сама модель состоит из следующих частей: AR – регрессионная модель, в которой используются зависимости между наблюдениями и числом интегрированных наблюдений (p), I – обеспечение стационарности путем взятия разностей (d), MA – анализ зависимостей между наблюдениями и остатками при применении модели к интегрированным наблюдениям.

Математическое описание модели показано на в формуле (1.2):

$$X_i = c + \sum_{i=1}^p \phi_i x_{i-1} + \epsilon + \sum_{i=0}^q \theta_i \epsilon_{i-1} \quad (1.2)$$

Параметры q и p называются порядками MA и AR. Метод ARIMA позволяет сделать прогноз на неструктурированных данных из-за введения в модель интегрированности. Достигается это путем взятия разностей уровней временного ряда друг из друга.

Стоит учитывать и сезонность временного ряда, большой вклад в модель внесут краткосрочные компоненты. Важным этапом анализа является оценка коэффициентов модели. Нужно следить за дисперсией, если она растёт с течением времени, необходимо использовать стабилизирующую дисперсию трансформации и взятие разностей.

В алгоритмической торговле важной областью является применение одномерных моделей временных рядов в прогнозирование волатильности. Волатильность финансовых временных рядов обычно не постоянна во времени, а изменяется, причем периоды волатильности группируются вместе.

Так как AR и MA между собой взаимодействуют, информация, предоставляемая функциями автокорреляции и частичной автокорреляции, больше не является надежной и может использоваться только в качестве начальной точки.

Члены AR и MA могут нейтрализовать эффекты друг друга, поэтому всегда следует уменьшать число членов AR и MA на 1.

1.12 Многослойный персептрон (MPC)

Нейронные сети прямого распространения называются многослойными персептронами (рис. 1.9). Входной сигнал в таких сетях передается от слоя к слою. Сама сеть состоит из множества узлов, которые образуют входной слой; одного или несколько скрытых слоёв вычислительных нейронов; одного выходного слоя нейронов.

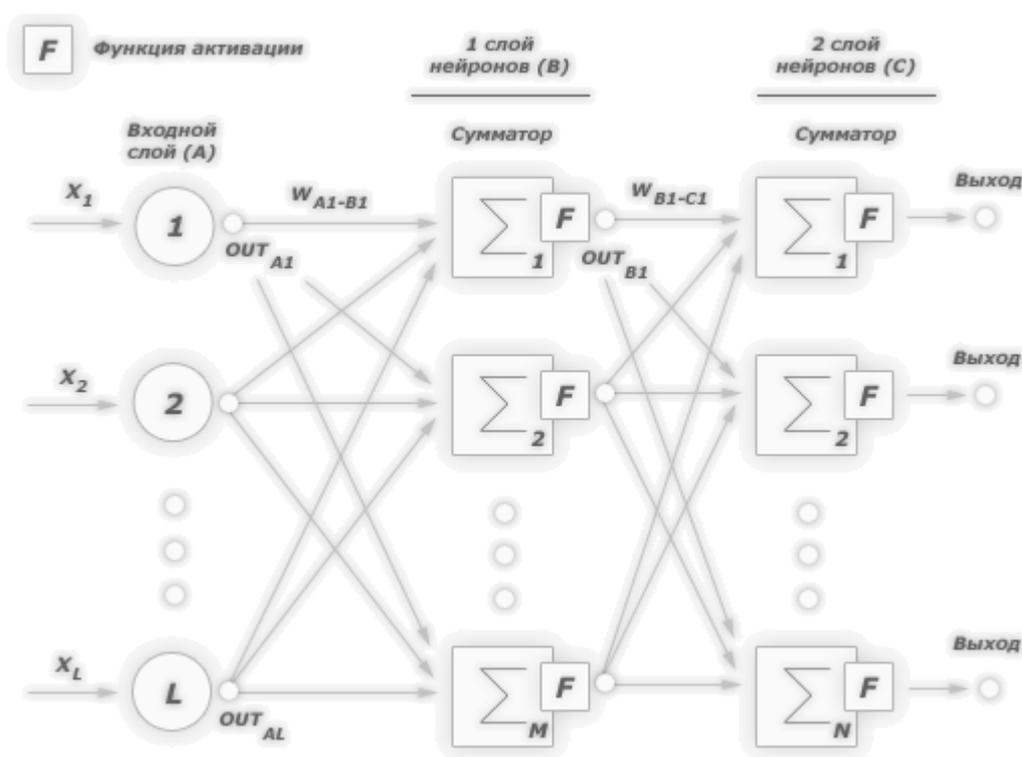


Рис. 1.9 – Многослойный персептрон

У модели MPC количество входных и выходных элементов определяется условиями задачи. Главное сомнение, которое может возникнуть – какие входные значения использовать, а какие нет. Сколько использовать промежуточных слоёв и элементов в них, пока совершенно неясен. В качестве начального параметра можно взять один промежуточный слой, а число

элементов в нём положить равным полусумме числа входных и выходных элементов.

МРС используются в решении разнообразных задач и имеет три отличительных признака:

- каждый нейрон сети имеет нелинейную функцию активации.
- несколько скрытых слоёв
- высокую связность

Многослойный персептрон обычно содержит один или несколько скрытых нейронов, которые не являются частями входа или выхода сети. Эти нейроны позволяют обучаться решению сложных задач и последовательно извлекать наиболее важные признаки.

Комбинация всех свойств в МРС вместе со способностью обучения на собственном опыте, дает огромную вычислительную мощь. Однако плюсы иногда становятся минусами, так как существует неполнота современных знаний о поведении такого рода сетей, усложняется теоретический анализ многослойного персептрона.

1.13 Технические индикаторы

Техническими индикаторами называют различного рода математические функции, применяемые к историческим значениям (ценам, объемам торгов и прочим), которые показывают участнику рынка дополнительную информацию об истории ценового движения. По предназначению технические индикаторы принято делить на следующие подгруппы:

- индикаторы тренда (скользящие средние, SMA, EMA)
- осцилляторы – предсказывающие развороты тенденции (ROC, RSI)
- каналные индикаторы (полосы Боллинджера, канал Кельтнера)

Стоит выделить самые популярные индикаторы, которые применяются трейдерами на финансовых рынках. Простое скользящее среднее (SMA) (рис. 1.10) – это индикатор направления тренда, который вычисляет арифметическое среднее для истории цен инструмента за определенный промежуток времени (формула 1.3).

$$SMA_t(N) = \frac{\sum_{p=t-N}^t close_p}{N} \quad (1.3)$$

Чаще всего рассчитывается по официальным ценам закрытия, но может быть применён к максимальной, минимальной, цене открытия и средней цене.

Чтобы построить 200-дневное скользящее среднее суммируются цены закрытия выбранного инструмента за последние 200 дней, и полученная сумма делится на 200. Каждый день добавляется новое значение, старое отбрасывается. Таким образом, среднее значение меняется изо дня в день. 50-дневное скользящее среднее строится на основе данных за последние 50 дней, а 10-дневное — на основе данных за последние 10 дней.

Дневные скользящие можно перенести на недельный график и корректировать период. Инвесторы обычно используют данные индикатор как индикатор тренда.



Рис. 1.10 – Технически индикатор SMA

У данного индикатора есть свои плюсы и минусы. Главный минус данного индикатора, что он не предсказывает скорое изменение тренда и обычно запаздывает. Плюс индикатора в помощи определения развития будущего тренда.

Индикатор ROC (Rate of Change) (формула 1.4) – это осциллятор скорости изменения цены между его периодами в процентном соотношении.

$$\text{RateOfChange} = \frac{P}{P_n} \quad (1.4)$$

Нулевая линия является ключевым элементом этого типа осцилляторов. Её пересечение снизу вверх считается положительным сигналом для покупки, а пересечение сверху вниз сигналом для продажи. Индикатор скорости изменения цены хорошо показывает дивергенции. Если, к примеру, максимальное значение индикатора ROC было ниже предыдущего, а цена сформировала новый максимум, то наблюдается медвежье расхождение показаний индикатора и цены, поэтому вскоре следует ожидать коррекцию вниз или разворот цены (рис. 1.11).



Рис. 1.11 – Индикатор ROC

В данной работе каждой акции будут рассчитаны несколько технических индикаторов, включая простую скользящую среднюю (более 5, 10, 15, 20, 25, 30, и 40 дней), волатильность (более 5, 10 и 20 дней), полосы Боллинджера (с использованием 5-дневной простой скользящей средней),

конвергенция скользящей средней, дивергенция (через 12 и 26 дней), импульс (через 2, 5 и 10 дней) и индекс относительной силы (более 14 дней).

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Симуляция работы биржевого робота

В работе будет использоваться биржевой робот, который будет торговать на бирже на одном финансовом инструменте. Основной целью работы робота будет получение максимальной прибыли за выбранный промежуток времени в результате совершенных им действий на бирже. Робот может совершать три действия: покупка инструмента, продажа инструмента и удержание инструмента.

Данная задача будет реализована алгоритмом обучения с подкреплением, так как данный робот будет совершать действия в зависимости от состояния среды, в которой он находится (финансовый рынок), робот будет узнавать о результате своей работы выраженной в условных единицах прибыли или убытка от совершенных торговых операций.

Для данных алгоритмов достаточно узнавать результат торговых операций только в конце периода, в этом состоит разница с традиционными формулировками задачи обучения с подкреплением, в которой вознаграждение используется по мере выполнения задачи.

Перед началом работы стоит ввести несколько обозначений, который мы будем использовать:

- Количество сделок – количество совершенных покупок или продаж инструментов.
- Позиция – количество, которым владеет робот в определенный момент времени.
- Объем – цена актива, умноженная на количество в момент совершения операции с инструментом.
- Результат работы робота – его конечный результат, полученный после проведения всех торговых операций.

Мы будем стимулировать нашего робота совершать операции на краткосрочных промежутках, вместо того чтобы заработать на длительных

периодах, так как нам выгодно большое количество сделок для увеличения показателя портфеля.

В качестве главной стратегии для тестирования будет «buy-sell-hold», робот умеет покупать, удерживать инструмент или продавать в шорт. Наши расчёты будут проводиться с учётом стандартной комиссии за сделки, организованную брокерской компаний, в нашей работе она составляет 5 центов за сделку, также мы будем учитывать задержку поступления данных, которая будет равна 0.00015 секундам. Данное добавление приблизит к реальному процессу торговли на финансовых рынках.

2.2 Алгоритм работы биржевого робота

В нашей работе мы будем использовать рекурсивных подход обучения торговых систем. Стоит отметить, что данный алгоритм был изобретён в организации ПАО «Санкт-Петербургская биржа», одни из сотрудников алгоритмического отдела, но ранее в нём не практиковались методы машинного обучения. Под торговой системой мы подразумеваем работу робота на каких-либо инструментах. Мы последовательно будем обучать наши торговые системы с разными параметрами и методами на определенном промежутке времени. В качестве параметров мы будем использовать технические индикаторы. Обучив и получив наилучшие индикаторы каждой системы, мы выделим наиболее успешные системы и будем их использовать на данных, которые система ещё не видела и так далее. При больших убытках торговой системы мы будем отключать её или заменять её параметры на другие, тем самым сможем минимизировать наши убытки. Стартовая сумма нашего портфеля будет 10000 долларов, это оптимальное значение для входа в рынок.

Мы будем покупать и продавать по 50 лотов определенного инструмента, но в дальнейшем этот параметр можно настроить в зависимости от размера портфеля и приоритетности работы.

В качестве тестовых выборок мы будем использовать реальные исторические данные разных инструментов за разные промежутки времени из открытых источников.

Стоит уточнить, что наши торговые системы будут иметь ряд ограничений. В реальных условиях торгов может устанавливать максимальный и минимальный лот на сделку, но так как мы будем использовать инструменты американского фондового рынка, то большинство инструментов на данном рынке имеет лот кратный 1. В реальных торгах при выборе инструмента стоит учесть его ликвидность при торговле на часовых, минутных или секундных фреймах, так как ликвидность является ключевым параметром для возможности увеличения прибыли.

В нашей модели действия происходят мгновенно, но в реальном режиме времени может происходить задержка между принятием решений роботом о покупке или продаже и исполнением его заявки на бирже. Но в нашем случае данными задержками можно пренебречь, так как в нашей работе будем рассматривать дневные периоды и будем торговать на дневном фрейме.

Мы будем использовать нашего робота для торговли на одном инструменте, но мы также можем научить его торговать на нескольких инструментах одновременно при использовании асинхронного программирования.

В ходе экспериментирования с разными параметрами стоит учитывать, что нет особо смысла сравнивать результат работы нашего алгоритма на разных инструментах. Для оценки качества работы наших алгоритмов будем анализировать конечный результат работы нашей стратегии.

2.3 Описание структуры данных

Для наших торговых систем мы будем использовать открытые данные с сайта yahoo.com, которые будут включать в себя дату, цену открытия, максимальную цену, минимальную цену, цены закрытия, цену торгов, объемы

торгов. Временной промежутком, который мы будем использовать 01-01-2019 по 31-12-2019.

2.4 Реализация алгоритма

Наш алгоритм будет состоять из нескольких частей, полученные данные с открытого источника будут использоваться в разных торговых системах с разными параметрами, каждый отдельный робот не будет знать о результатах торговли другого. За основной параметр мы возьмем расчёт средних цен окном, вторичными параметрами будут являться технические индикаторы и разным методы машинного обучения. Наилучшие торговые системы мы будем заносить в таблицу и анализировать их результаты на новых данных. Также будем строить графики для наилучших и наихудших систем для понимания разбора логики работы робота

Сам алгоритм будет реализован с помощью языка Python 3.8 в виртуальной среде с использованием библиотеки Anaconda, Dask и IDE PyCharm 2020.1.

2.5 Результаты работы

При тестировании нашего алгоритма мы взяли за основу акции Microsoft (MSFT), так как данная бумага является высоколиквидной, и цена её колеблется в районе 120 – 180 долларов за 1 лот.

Первый промежуток, который мы использовали для обучения наших торговых систем с 2019-01-01 по 2019-02-01. При обучении алгоритма на обучающих данных мы выяснили, что индикаторы SMA60, SMA5, vol30, momentum2, vol40, SMA20, SMA10, momentum5, SMA25 показали наилучшие результаты (таблица 2.1).

Мы будем использовать данные индикаторы для проверки их точности на тестовом торговом промежутке. Столбчатый график результатов обучения показан на рис. 2.1.

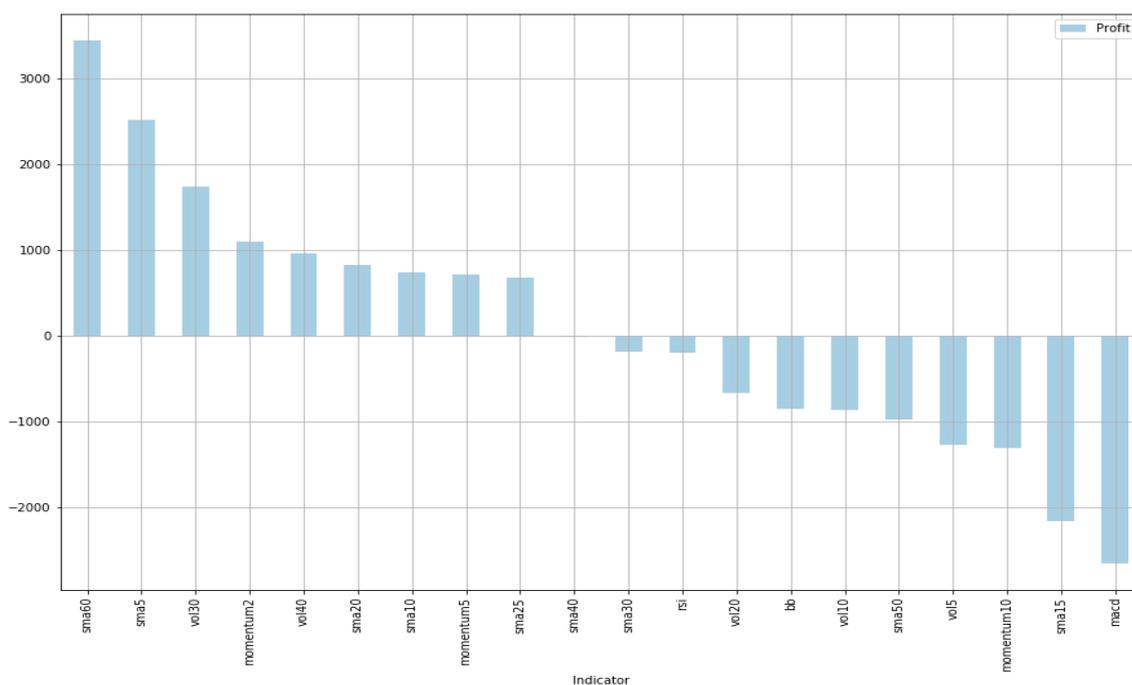


Рис. 2.1 – Столбчатый график результатов обучения

Таблица 2.1 – Наилучшие индикаторы на тестовом обучении

Индикатор	Прибыль
SMA60	3449.39
SMA5	2513.39
vol30	1735.79
momentum2	1102.39
vol40	963.3922
SMA20	825.3922
SMA10	735.3922
momentum5	715.7974
SMA25	674.3922

Следующим торговым промежутком 2019-02-02 по 2019-03-01, на котором мы применим нашу обученную систему. Стоит уточнить, что мы будем использовать все имеющиеся индикаторы для анализа обучения торговых систем, на реальных торгах стоило бы просчитать все точки для

тейк-профитов и стоп-лоссов, чтобы алгоритмы, показавшие наихудшие результаты на тестовых обученных, не создавали нам убытки. Наилучшим решением было бы создание инструмента отключения неприбыльных торговых систем при определенном показателе просадки.

При тестировании нашего алгоритма обучения с подкреплением на данных, которых он еще не видел, мы получили схожий результат (таблица 2.2).

Таблица 2.2 – Наилучшие индикаторы на новых данных

Индикатор	Прибыль
SMA60	1422.56
vol5	1053.56
SMA40	881.70
SMA5	863.56
vol10	771.56
SMA30	756.56
SMA50	727.70
MACD	639.56
momentum10	545.56
momentum5	416.56
RSI	377.70

При сравнении двух таблиц можно увидеть, что выбранные индикаторы уверенно показывают хорошие результаты на обучающей данных и на тестовых, которые алгоритм обучения с подкреплением ещё не видел. Но также добавились и индикаторы, которые приносили нам убытки, это обосновывается тем, что рыночный тренд постоянно меняется, и чтобы предугадать движения котировок, надо тщательно анализировать поток информации.

Столбчатый график результатов работы Q-алгоритма можно увидеть на рис. 2.2.

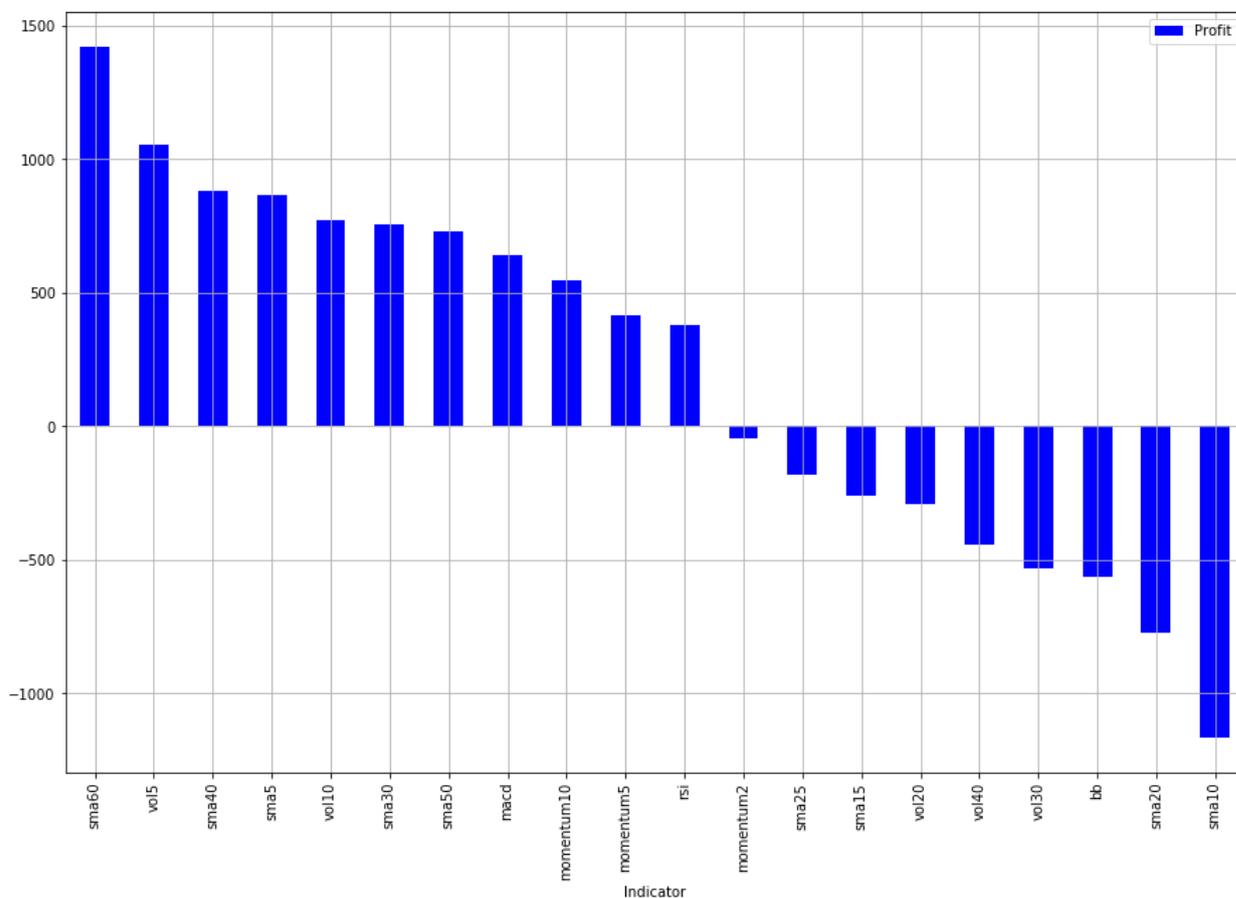


Рис. 2.2 – Столбчатый график результатов работы алгоритма на новых данных

Попробуем улучшить нашу модель рекурсивной подачи параметров, автоматизировав нахождение наилучших индикаторов на каждом промежутке. Мы будем использовать топ 3 индикатора, которые хорошо взаимосвязаны между собой. Применим статистическую взаимосвязь двух или более случайных величин или по-другому - корреляцию.

Зададим более внушительный промежуток для нашей модели – будем обучать её на данных от 2019-01-01 до 2019-06-01 и проанализируем взаимосвязь технических индикаторов между собой.

На рис. 2.3 показана корреляционная матрица, показывающей величину корреляции между несколькими техническими.

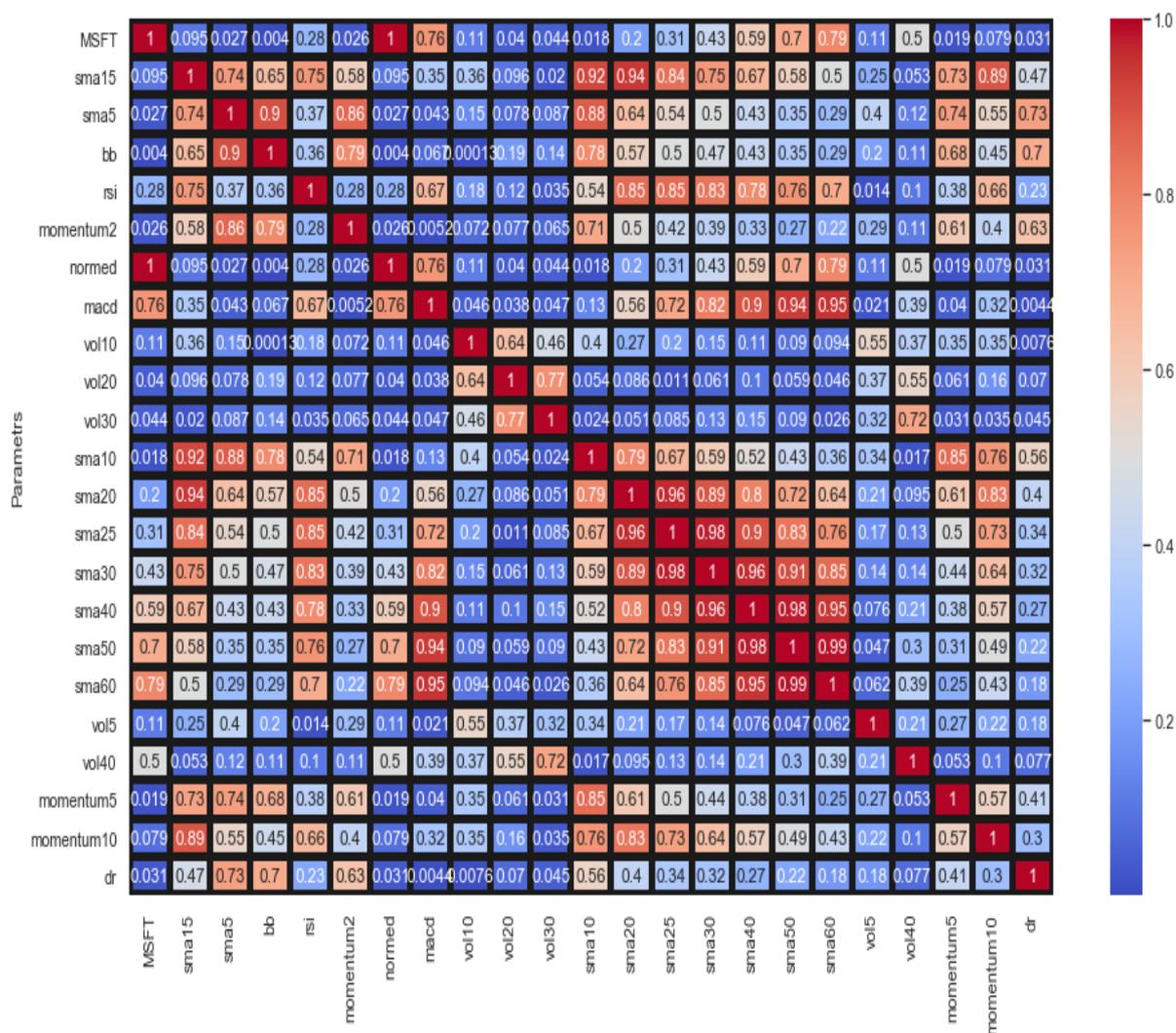


Рис. 2.3 – Тепловая карта корреляции технических индикаторов

На тепловой карте видно, что наилучшая взаимосвязь у SMA5, bb, momentum2, наш алгоритм во время обучения использовал их в большинстве случаев, поэтому мы проанализируем данную связку на новых данных.

Возьмем промежуток 2019-06-02 по 2019-12-31 и оценим работу нашего алгоритма. Как оказалось, выбранные индикаторы показали очень хороший результат (рис. 2.4). На обучающей данных максимальная просадка достигала величины -1000 долларов в самом начале периода. С помощью рассчитанных технических индикаторов нам удалось заработать \$1,662.69 (16.63 %) на обучающей выборке.

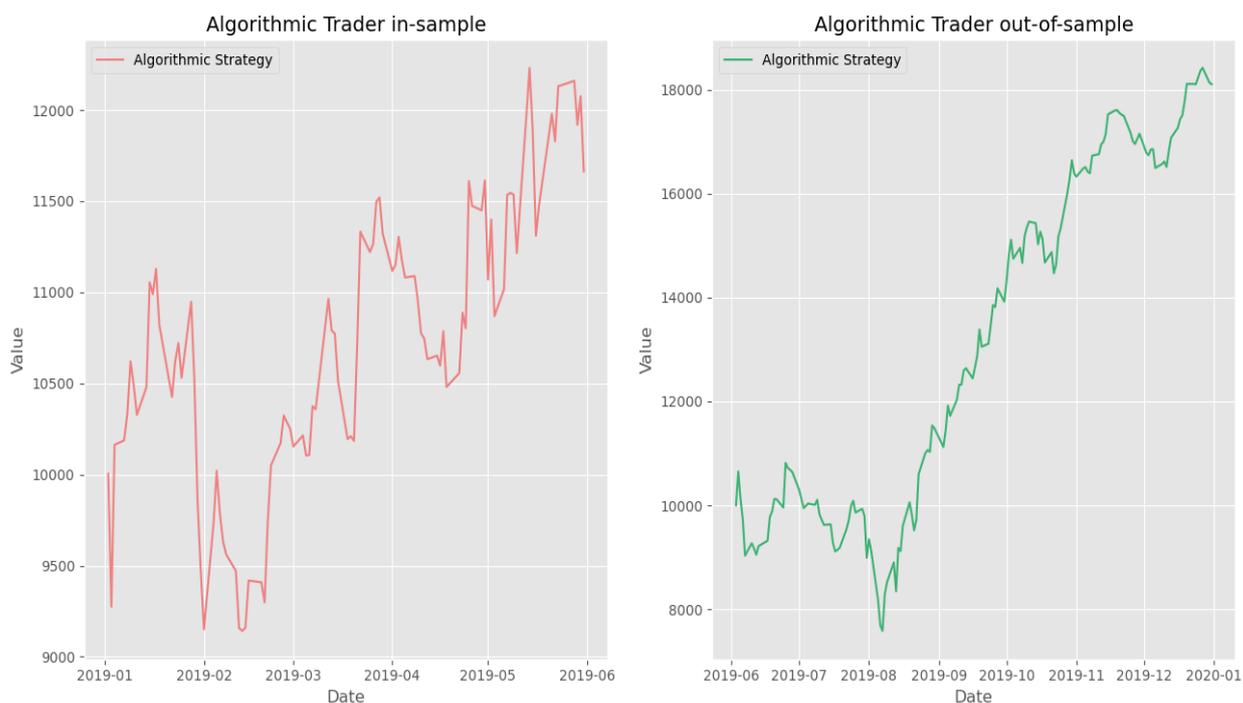


Рис. 2.4 – Результат работы алгоритма на обучающей и тестовой выборках

На тестовой выборке максимальная просадка была чуть больше, но сам алгоритм заработал \$8,102.82 (81.03 %), что является очень хорошим результатом за такой промежуток времени. Стоит отметить, что в данный промежуток не было значимых новостей для компании Microsoft, которые могли бы повлиять на результат торговли наших систем. Но стоит понимать, что рынок очень быстро реагирует на внешние изменения и это может повлиять на конечный результат работы наших торговых систем.

2.6 Выводы по работе алгоритма

В данной главе мы проанализировали работу нашего алгоритма обучения с подкреплением. Рекурсивной ввод параметров и автоматизация этого процесса даёт хорошие результаты. Данная алгоритм торговли стоит развивать, и он может показать отличные результаты на реальных торгах, так как в нём есть уникальность параметризации.

Корреляция всех возможных индикаторов внутри рынка позволит инвесторам или компаниями получать прибыль и минимизировать убытки. К этому алгоритму стоит добавить рисковые модели, что блокировать неправильные решения роботов.

Лучшие алгоритмы были реализованы с помощью популярных технических индикаторов, что подтверждает их значимость. Алгоритм с подкреплением является одним из лучших для задач алгоритмической, так как он подстраивается по тренды рынка и может самостоятельно принимать решение.

ЗАКЛЮЧЕНИЕ

В ходе данной работы была рассмотрена предметная область алгоритмической торговли с точки зрения применения к её задачам алгоритмов машинного обучения. Основной задачей, решаемой в работе, было создание автоматизированных торговых систем, совершающие заданные действия на бирже и способные зарабатывать на совершение сделок по покупке и продаже финансовых инструментов. В работе были рассмотрены основные алгоритмы машинного обучения и обработки данных для алгоритмической торговли. Изучен и проанализирован алгоритм обучения с подкреплением, а также и другие алгоритмы. В результате данной работы был выявлена сложность решения задач в алгоритмическую торговле – основная из них это непредсказуемость рынка. Для решения задачи был представлен алгоритм рекурсивного обучения торговой системы с помощью Q-обучения с подкреплением.

В ходе приведённых экспериментов на реальных исторических биржевых данных по финансовым инструментам, показана возможность успешного применения алгоритма для решаемой задачи. Совокупность алгоритмов дала положительный финансовый результат практически во всех экспериментах с помощью стратегии «buy-sell-hold».

Дальнейшим возможным направлением работы являются создание программного продукта для торговли на бирже, работающего на основе предложенного алгоритма, а также дальнейшие исследования применимости различных алгоритмов машинного обучения к этой задаче в рамках рассмотрения её как задачи обучения с подкреплением.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Gould M. D., Porter M. A., Williams S., McDonald M., Fenn D. J., Howison S. D. Limit order books. *Quantitative Finance*, 2013. С. 1709–1742.
2. Tun L. , Shen L. G. . Stock Price's Prediction with Decision Tree. *Applied Mechanics and Materials*, 2011. С. 1116-1121.
3. Seethalakshmi R. , Saavithri V. , Badrinath V. , Vijayabanu C.. PCA based support vector machine technique for volatility forecasting. *IJRET: International Journal of Research in Engineering and Technology*, 2014.
4. Mitchell T. M. . *Machine Learning*, 1997.
5. Burges C. J. C. . A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 1998. С. 121-167.
6. Ding X. , Zhang Y. , Liu T. , Duan J. . Deep Learning for Event-Driven Stock Prediction. *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
7. Mnih V. , Kavukcuoglu K. , Silver D. , Graves A. , Antonoglou I. , Wierstra D. , Riedmiller M. . Playing Atari with Deep Reinforcement Learning. *NIPS Deep 49 Learning Workshop*, 2013. 12. Moody J., Saffell M. . Reinforcement Learning for Trading. *Lecture Notes in Computer Science*, № 5821, 1999. С. 18-23. 13. Fernandez-Tapia J. . High-Frequency Trading meets Reinforcement Learning. 2015.
8. Sylvester J. , Chawla N. V. . *Evolutionary Ensembles: Combining Learning Agents using Genetic Algorithms*, 2005.
9. Barros R. C. , Basgalupp M. P. , de Carvalho A. C. P. L. F. , Freitas A. A. . A Survey of Evolutionary Algorithms for DecisionTree Induction. *IEEE transactions on systems, man and cybernetics - part C: applications & reviews*, 2012.