

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программного обеспечения

РЕКОМЕНДОВАНО К ЗАЩИТЕ
В ГЭК И ПРОВЕРЕНО НА ОБЪЕМ
ЗАИМСТВОВАНИЯ

Заведующий кафедрой

к.т.н., доцент

 М. С. Воробьева

24.06.2019 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

(магистерская диссертация)

ПРОГНОЗИРОВАНИЕ ЦЕН НА ФОНДОВОМ РЫНКЕ С ПОМОЩЬЮ
МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

02.04.03. Математическое обеспечение и администрирование
информационных систем

Магистерская программа «Разработка, администрирование и защита
вычислительных систем»

Выполнил работу
Студент 2 курса
очной формы обучения



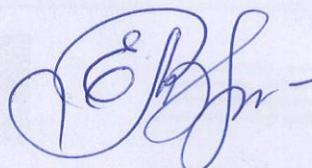
Вакарин
Дмитрий
Николаевич

Научный руководитель
к.т.н., доцент



Воробьева
Марина
Сергеевна

Рецензент
к.э.н., доцент



Мазикова
Екатерина
Владимировна

Тюмень 2019

ОГЛАВЛЕНИЕ

| | |
|--|----|
| ВВЕДЕНИЕ..... | 3 |
| ГЛАВА 1. ОБЗОР МОДЕЛЕЙ ДЛЯ ПРОГНОЗИРОВАНИЯ ЦЕН НА БИРЖЕВЫХ РЫНКАХ..... | 6 |
| 1.1 Модели и методы прогнозирования цен на биржевых рынках..... | 6 |
| 1.2 Биржевые данные и методы их получения..... | 13 |
| 1.3 Инструменты для прогнозирования значений биржевых активов..... | 19 |
| ГЛАВА 2. МОДЕЛИ НЕЙРОННЫХ СЕТЕЙ КАК СПОСОБ ПРОГНОЗИРОВАНИЯ БИРЖЕВЫХ ЗНАЧЕНИЙ..... | 23 |
| 2.1 Развитие нейронных сетей..... | 23 |
| 2.2 Математическая модель нейронных сетей, основные понятия.... | 27 |
| 2.3 Многослойный персептрон..... | 34 |
| 2.4 Рекуррентные нейронные сети..... | 37 |
| ГЛАВА 3. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ ЗНАЧЕНИЙ ИНДЕКСА S&P500 С ПОМОЩЬЮ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ..... | 41 |
| 3.1 Разработка и этапы проектирования приложения для прогнозирования значения S&P500..... | 41 |
| 3.2 Предварительная обработка данных и программная реализация моделей нейронных сетей..... | 45 |
| 3.3 Обучение моделей и прогнозирование значений индекса..... | 48 |
| ЗАКЛЮЧЕНИЕ..... | 51 |
| СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ..... | 53 |

ВВЕДЕНИЕ

Биржевой рынок – один из наиболее популярных способов инвестирования во всем мире. Низкий порог вхождения, потенциальная прибыль, техническая доступность – все это делает фондовые рынки привлекательным объектом для притока денежных средств.

Фондовые рынки являются подходящим местом для агрегирования и перераспределения денежных ресурсов, осуществления сделок между компаниями (слияние/поглощение). В следствии этого денежные обороты биржи достигают больших значений – к примеру, Нью-Йоркская фондовая биржа NYSE обслуживает суточный объем торгов в 50 млрд. долларов. В связи с этим, неудивительно, что существует множество видов анализа и моделей, реализованных с помощью вычислительных систем, пытающихся получить доход, связанный с деятельностью торговых площадок.

В XX веке, с развитием техники и появлением новых открытых данных, увеличилась доступность проведения анализа – информацию стало проще хранить и обрабатывать. Численные методы, используемые в статистике и математике, стали применяться к финансовым данным. Ранние исследования предсказания цены датируются 1960-1970 годами: работой Eugene Fama «Поведение рыночной цены на бирже», и работой «Теория случайных блужданий», которая была развита Cootner в 1964, Fama и Fisher в 1969 году. Эти ранние модели предполагают, что рыночная цена не может быть спрогнозирована с более чем 50% точностью. Со временем стали появляться исследования (см. Malkiel, 2003; Smith, 2003; Vollen, Mao & Zeng, 2011), которые подтверждают, что рынок может быть спрогнозирован. Точность предыдущих исследований по прогнозированию стоимости не превышала 83% (см. Lawrence, 1997; Vu, Chang, Ha & Collier, 2012). Эффективной системы не существуют и многие, связанные с этой темой, обсуждения касаются повышения точности предсказания. Доступность

инвестирования и потенциальные объемы прибыли, делают задачу прогнозирования цены финансовых активов актуальной.

Цена компании на фондовом рынке зависит от многих факторов. Факторы, влияющие на ценообразование фондовых активов, в зависимости от применяемого метода анализа могут варьироваться от макроэкономических показателей до технических индикаторов, таких как скользящая средняя и другие. Выбор факторов, влияющих на ценообразование зависит от самого финансового инструмента, а также ряда допущений, присутствующих в модели. К примеру, цены на облигации государственного займа, согласно исследованию Ричарда Колмена (1989) во многом зависят от макроэкономических показателей, в то время как цены на высоколиквидные фьючерсы прогнозируются путем выделения высокочастотных технических индикаторов, таких как цены за предыдущий период и пр.

Коллективные базы знаний (Wikipedia, Google Trend), машинное обучение, развитие систем баз данных, наличие крупных площадок с API, занимающихся систематическим сбором и хранением информации (Finanz, Google Finance, Yahoo finance, Nyse, Nasdaq) – все это эффективные инструменты для изучения вопроса о ценообразовании ценных бумаг.

Объектом исследования является рыночная стоимость активов на фондовом рынке.

Предметом исследования является математическая модель, позволяющая по выборке значений множества факторов прогнозировать значения показателей рыночной стоимости биржевых активов.

Цель магистерской диссертации – разработать систему, определяющую значение индекса S&P500 на основе цен акций компаний, входящих в индекс при помощи методов машинного обучения.

Для достижения цели были поставлены следующие задачи:

— изучить исследования, связанные с прогнозированием цен акций;

- определить, какие данные могут быть использованы для прогнозирования цены;
- на основе выбранных предикторов, разработать модель определяющую цену акции;
- разработать web-приложение;
- оценить точность полученных результатов;

ГЛАВА 1. ОБЗОР МОДЕЛЕЙ ДЛЯ ПРОГНОЗИРОВАНИЯ ЦЕН НА БИРЖЕВЫХ РЫНКАХ

1.1. Модели и методы прогнозирования цен на биржевых рынках

Для прогнозирования цен на биржах используют множество методов и моделей. В этом параграфе дадим описательную характеристику наиболее используемым методам и моделям, которые применяются в задачах прогнозирования цен.

Методы прогнозирования – это научный форматизированный подход к решению задачи, который также характеризуется определенной последовательностью действия. Модель прогнозирования реализуется с помощью методов.

Под моделью прогнозирования понимается формализованное математическое описание процессов или объектов. Модели прогнозирования как правило опираются на использование методов прогнозирования. В настоящее время разработано большое количество различных методов и моделей. Для задач прогнозирования подбор моделей и методов осуществляется индивидуально и во многом зависит от принадлежности к тем или иным классам моделей.

Существуют различные классификации задач прогнозирования. Одной из таких классификаций – является классификация Громова [1], которая делит все типы задач прогнозирования в экономике на несколько уровней, от простого до сверхсложного.

Согласно классификации Громова, задача прогнозирования биржевых цен относится к категории «сверхсложных». Во многом это обусловлено сложной биржевой структурой, наличием большого количества факторов, влияющих на ценообразование, а также ограниченным доступом к биржевым данным для их анализа. В связи со сложностью задачи, при ее решении приходится учитывать большое количество функциональных

зависимостей, а также осуществлять тщательный выбор модели прогнозирования.

Термины модели и методы прогнозирования – являются схожими понятиями. В исследовательских работах, посвященных задачам прогнозирования нет строго определения методам и моделям и, как правило, данные термины могут замещать друг друга.

Модели, используемые для прогнозирования на финансовых рынках можно разделить на две крупные группы – качественные и количественные.

Качественные модели в целях формализации задачи используют в своей практике экспертные оценки. Эксперты – это лица, обладающие опытом в анализе объекта прогнозирования и факторов, которые могут на него воздействовать. Так как на формирования цены биржевого актива может влиять большое количество факторов, такие модели используют, когда невозможно оценить или учесть влияния многих факторов. Наиболее известными моделями в данной категории являются:

- Экспертные оценки. В данном случае используется экспертная оценка;
- Метод Дельфи. В данном методе используется экспертная оценка нескольких аналитиков, которая агрегируется в единое мнение, на основе анкетирований и пр.;
- Консенсус-прогнозы. Суть метода заключается в усреднении прогнозных мнений экспертов;

Количественные модели основаны на том, что взаимосвязь между различными факторами на фондовом рынке может выражено в количественном или числовом значении. Данные модели являются комплексными, т.к. в данных моделях используется множество вспомогательных методов. Класс количественных моделей также включает в себя две особые подгруппы моделей – универсальные и специальные.

Универсальными моделями являются:

- Регрессионные модели;

- Модели анализа временных рядов;
- Модели равновесия и др.;

Данные модели характеризуются как универсальные, т.к. применяются в других задачах прогнозирования таких как медицинская диагностика, добывающий сектор и пр.

Специальные модели являются моделями, применяемыми для анализа определенных финансовых инструментов. Биржа - это обобщенное название различных торговых площадок, на каждой из которых торгуются различные торговые активы [2]. На крупных биржах как правило представлены следующие виды активов:

- Валюта;
- Акции;
- Облигации;
- Фьючерсы;
- Опционы;
- ETF-фонды;

Данные активы обладают специфическими особенностями, которые определяют выбор модели для их прогнозирования. Для прогнозирования значений цен определенных активов существуют специальные модели:

- Модель дисконтированных дивидендов;
- Модель опционных рынков;
- Фундаментальный анализ;
- Технический анализ и пр.;

Выбор модели прогнозирования во многом определяется представлением причинно-следственных связей, возникающих на биржевом рынке. На основе причинно-следственных связей выделяются следующие группы моделей:

- Модели, основанные на взаимодействии участников рынка;
- Модели, основанные на физико-математических законах

- Модель, основанная на случайных факторах, которые нельзя формализовать в виде каких-либо законов;

Классификация моделей прогнозирования цен на финансовых рынках также осуществляется на основе методов обработки данных. В соответствии с данной классификацией выделяют:

1. Неструктурные модели.
2. Структурные модели.
3. Интеллектуальные модели.

В группу неструктурных моделей входят методы анализа временных рядов, регрессионные методы анализа, авторегрессионные модели (AR), модели экспоненциального сглаживания (ES), системы одновременных уравнений.

Структурные модели применяются для моделирования сложных систем. Так как биржевые рынки представляют собой сложную структуру, о которой тяжело получить представление [3], формализация их представляется сложной задачей. Структурные модели способствуют получению представления о зависимостях на биржевых рынках. К данному типу методов относят нейросетевые модели, модели на базе цепей Маркова, метод опорных векторов.

Интеллектуальные модели находят применение в том случае, если объект прогнозирования рассматривается как не формализуемый. К данному типу относят модель нечеткой логики, методы экспертных оценок, а также большую группу моделей Искусственного Интеллекта.

Структурные, неструктурные и интеллектуальные модели образуют группу количественных моделей [3, с. 58].

Приведем базовую типологию количественных методов прогнозирования в таблице:

Типология моделей

| Модели | Входящие данные | Способ переработки входящих данных | Исходящие данные |
|---|--|---|---|
| Неструктурные | Экономические и/или производственные показатели прошлого периода | Статистическая обработка значений временных рядов показателей. Постоянство найденных взаимосвязей, на основе корреляционно регрессионных связей и др. | Будущие значения экономических и/или производственных показателей |
| Структурные | Положения экономической теории, функции спроса, предложения полезности, производственная функция | «Подбор», «калибровка» показателей системы (рынка) для достижения заданных условий | Значения экономических и производственных показателей в заданных условиях (условиях равновесия) |
| Интеллектуальные (комбинированные, гибридные) | I. Обучение: Широкий набор «входящих» или «обучающих» данных («прошлый опыт») II. Работа: искомые входные параметры для прогноза | I. Обучение: Применение широкого математического инструментария и технологий обработки данных | Прогноз будущего состояния |

В основе неструктурных моделей лежит предположение о том, что зависимости на рынке неизменны на горизонте прогнозирования. В связи с этим данные модели показывают приемлемый результат при анализе небольших временных отрезков. Структурные модели применяются для средних и больших периодов [4].

Группа моделей искусственного интеллекта появилась недавно относительно прочих моделей. Достижения в области математики,

биохимии, информатики позволили исследователям моделировать нейронные сети, которые анализируют информацию по принципам схожим с принципами работы мозга. Данные сети обладают возможностью обучаться и анализировать информацию с помощью нейронов, которые лежат в основе данных сетей. Множество моделей искусственного интеллекта способных обучаться образуют моделей машинного обучения

Машинное обучение определяется как свойство автоматических сетей, которые способны анализировать информация по аналогии с мыслительной способностью человека. Несмотря на то, что данные модели принято объединять в отдельную группу моделей, в их основе лежат статистические и математические модели.

Модели машинного обучения представлены следующими моделями:

- Аналитические модели;
- Искусственные нейронные систем;
- Генетические алгоритмы;
- Статистические модели (модели регрессионного анализа, ARIMA и пр.);

Модели искусственных нейронных сетей являются самыми распространенными моделями [7, с. 182], используемых для прогнозирования значения временного ряда.

Согласно исследованиям, которые посвящены анализу моделей искусственного интеллекта, нейронные сети отличает универсальность применения. Наиболее распространёнными задачами для нейронных сетей являются:

- классификация;
- прогнозирование;
- оптимизация;
- идентификация;
- сегментация;

- кластеризация;

Согласно исследованию посвященному различным моделям прогнозирования, на фондовых рынках модели нейронных сетей введены многими крупнейшими инвестиционными компаниями, занимающимися управлением инвестициями, анализом рынков и пр. [8].

В настоящее время различными типами нейронных систем пользуются такие компании как:

- Black Rock (6.3 трлн. долл. AUM);
- Vanguard Group (5.1 трлн. долл. AUM);
- Charles Schwab (3.36 трлн. долл. AUM);
- JPMorgan Chase (278 трлн. долл. AUM);

AUM – активы под управлением.

Модели нейронных сетей классифицируют по типу связи между нейронами, по типу элементов, по объему памяти.

Классификация моделей нейронных сетей по типу связи приведена в таблице:

Таблица 1.2.

Классификация нейронных сетей

| направление классификации | типы нейронных сетей |
|---------------------------------------|--|
| Прямые и обратные связи | |
| Без обратных связей | Статические NN, NN прямого распространения, NN радиальных базисных функций, Персептрон |
| С обратными связями | Динамические NN, RNN, NN обратного действия |
| Специфические связи | Самоорганизующиеся NN, NN адаптивного резонанса |
| Наличие или отсутствие боковых связей | |
| Включены боковые связи | Сложнодинамические NN с боковыми связями в скрытом слое |
| Нет боковых связей | Остальные NN |

С развитием нейронных сетей и компьютерной техники возросло количество исследований, изучающих задачу прогнозирования цен на фондовом рынке.

В исследовании IBM 1988 года, с использованием нейронных сети прямого распространения, точность прогноза составила 57,8%. Индийские исследователи А. Хуссиан и М. Мажумдер в своей работе над прогнозированием индийского индекса S&P CNX Nifty, с использованием метода обратного распространения ошибок и нейронной сети обратного распространения со скрытым слоем, добились точности 69,72%. К. Счирхолд и С. Дагли в своей работе 1996 года, посвященной прогнозированию индекса S&P500 с помощью модели многослойного персептрона, добились точности 75,4%. Д. Жанг и К. Чиян в своем исследовании 2004 года, посвященному прогнозированию индекса Shanghai Composite с использованием многослойных нейронных сетей, добились точности 81,7%.

Согласно исследованиям наилучшие результаты прогнозов получается при попытке спрогнозировать значения индексов по значениям активов, входящих в данный индекс.

1.2 Биржевые данные и методы их получения

Сбор и предварительная обработка данных является одним из первых этапов в работе над задачами прогнозирования значений [9].

В процессе торговли, биржевые активы в каждый момент времени генерируют множество данных (см. рис.). Получение биржевых данных может осуществляться с помощью двух типов источников:

- API биржи;
- API сторонних ресурсов;

Большинство российских бирж предоставляют интерфейс для получения данных, в то время как в США распространены сторонние ресурсы получения данных. [10]

| | |
|---|------------------|
| Цена предторгового периода/Цена аукциона открытия | 243 |
| Цена сделки минимальная | 237,56 |
| Цена сделки максимальная | 243,9 |
| Цена последней сделки | 238,02 |
| Цена закрытия | 238,02 |
| Количество сделок за день, штук | 64 580 |
| Объем сделок за день, штук ценных бумаг | 61 061 660 |
| Объем сделок за день | 14 650 470 642,3 |
| Объем первой сделки | - |
| Объем последней сделки | - |
| Средневзвешенная цена | 239,93 |
| Рыночная цена (2), рублей | 239,93 |
| Рыночная цена (3), рублей | 239,93 |
| Признаваемая котировка, рублей | 238,02 |
| Объем сделок для расчета рыночной цены (2), рублей | 14 650 470 642,3 |
| Объем сделок для расчета рыночной цены (3), рублей | 14 650 470 642,3 |
| Объем сделок для расчета признаваемой котировки, рублей | 14 650 470 642,3 |

Рис. 1.1 – Пример биржевых данных акций ОАО «Сбербанк».

API Московской Биржи

Московская биржа является крупнейшей биржевой группой в России, включающей в себя множество торговых площадок, где осуществляется торговля акциями, облигациями, деривативами, валютой и драгоценными металлами.

Информационно-статистический сервер Московской Биржи (ИСС / ISS) функционирует в рамках программного комплекса Интернет-представительства Московской Биржи и служит для предоставления клиентским приложениям данных с рынков Московской Биржи.

В рамках интерфейса доступны следующие типы информации: статические данные о рынках (режимы торгов и их группы, финансовые инструменты и их описание), данные для построения графиков ("свечей"),

сделки (анонимно), котировки, исторические данные, различные метаданные.

Аналогично продукту MOEX Trade INFO, который также работает через ИСС, данные могут предоставляться или по подписке в режиме реального времени или в свободном доступе (без авторизации, но с задержкой).

Взаимодействие с сервером осуществляется по протоколу http. Данные для аутентификации также возможно передавать по протоколу http.

Запросы к ИСС формируются в виде URL с параметрами. Концепция формирования ссылок базируется на архитектуре RESTful. Таким образом параметры передаются не только в конце строки, но и формируют сам URL.

Параметры, входящие в состав URL определяются синтаксисом каждого конкретного запроса. На задаваемый параметром объект рынка указывает предшествующий этому параметру элемент URL. Доступные типы рыночных объектов:

- /engines/(trade_engine_name)
- /markets/(market_name)
- /boards/(boardid)
- /boardgroups/(board_group_id)
- /securities/{secid}

Значения первых четырёх переменных могут быть получены через запрос index, последней – через securities. Параметры в конце URL могут относиться или к конкретному блоку данных, который будет получен через запрос, или ко всем блокам, или быть системными, применимыми ко всем запросам. Относящиеся к запросам параметры указаны в описании синтаксиса и могут передаваться как с явным названием блока: «блок.параметр=», так и без: «параметр=»; в последнем случае действие будет распространяться на все блоки, для которых этот параметр имеет силу.

Ответ от сервера осуществляется в следующих форматах: XML, CSV, JSON, HTML [23].

API Московской биржи не позволяет получать внутридневные исторические данные. Например, невозможно получить исторические значения цен акций Сбербанка с интервалом в минуту.

API сторонних ресурсов

IEX (iextrading.com) – электронный ресурс предоставляющий API для получения биржевых данных рынков США. Для данного API существуют неофициальные реализации на таких языках как Python, C++, C#, Java и др.

Получение данных осуществляется с помощью http запросов. Поддерживаются только запросы GET. При создании запроса множественные параметры разделяются запятыми. Все запросы поддерживают параметр filter, который возвращает подмножество данных.

```
?filter=symbol,volume,lastSalePrice
```

Рис. 1.2 – Пример указания фильтра в запросе.

Данные с IEX ресурса предоставляются в формате CSV или JSON.

Примеры допустимых http запросов:

- GET /ref-data/symbols – возвращает массив тикеров, которые поддерживаются IEX;
- GET /ref-data/daily-list/symbol-directory – вызывает массив содержащий новые выпуски ценных бумаг, изменения тикеров и названий, удаленных выпусков и прочих изменений;
- GET /ref-data/daily-list/dividends – вызывает массив содержащий информацию о выплачиваемых дивидендах;
- GET /tops?symbols=aapl – предоставляет агрегированные котировки IEX и предложение к покупке в реальном времени для всех ценных бумаг книги ордеров IEX;

IEХ не предоставляет возможность получения внутридневных исторических данных.

Alpha Vantage API (alphavantage.co) – электронный ресурс для получения биржевых данных. API данного ресурса разделен на три группы:

- Биржевые данные;
- Данные криптовалют;
- Технические индикаторы;

Биржевые данные предоставляются по четырем различным временным периодам: дневные, недельные, месячные и внутридневные. Дневные, недельные и месячные данные содержат 20+ летний диапазон значений. Внутридневные данные предоставляются только за один день.

Данные предоставляются в формате JSON и CSV. Пример запроса для получения внутридневных данных Microsoft:

```
https://www.alphavantage.co/query?function=TIME_SERIES_INTRADAY&symbol=MSFT&interval=5min&apikey=demo
```

Один из самых известных провайдеров биржевых данных – Yahoo.Finance (finance.yahoo.com). Данный ресурс позволяет загрузить исторические дневные данные практически по всем торгуемым инструментам на американских биржах. Данный сервис предоставляет интуитивный интерфейс для загрузки исторических данных в формате CSV или JSON. Сервис позволяет загружать дневные, недельные, месячные значения.

Примеры допустимы http запросов:

- market/get-summary – сводная информация в реальном времени;
- market/get-movers – отображает активы которые выросли или упали в цене с привязкой к региону;
- market/get-quotes – релевантная информация по тикеру;
- market/get-charts – данные для визуализации;

Yahoo.Finance не предоставляет исторических внутридневных данных.

Электронный ресурс Finam (finam.ru) – это российский сайт компании ООО «Финам». ООО «Финам» – это крупнейший в России розничный брокер, а также инвестиционная группа, специализирующаяся на оказании трейдерских, банковских услуг. Сайт Finam является очень популярным среди трейдеров и является самой используемой российской площадкой для загрузки исторических данных [10].

Данный ресурс предоставляет графический интерфейс для загрузки данных. Формат загрузки данных – TXT, CSV. Пример настройки интерфейса для загрузки исторических цен акций ОАО «Газпром»:

The screenshot shows the configuration interface for downloading historical data for the stock GAZP on the MosBourse website. The interface includes the following elements:

- Stock Selection:** "МосБиржа акции" (MosBourse Stocks) and "ГАЗПРОМ ао" (GAZP JSC).
- Interval and Frequency:** Start date: 30.03.2005, End date: 24.05.2019, Interval: 1 min.
- File Name:** "Имя выходного файла" (Output file name): GAZP_050330_190524, Format: .txt.
- Contract Name:** "Имя контракта" (Contract name): GAZP.
- Format:** "Формат" (Format) section with date format (гггммдд), time format (ччммсс), and "Выдавать время" (Output time) options: начала свечи (Start of candle), окончания свечи (End of candle), московское (Moscow).
- Separator:** "Разделитель" (Separator) section with field separator (запятая (,)) and number of digits (нет).
- File Format:** "Формат записи в файл" (File record format): TICKER, PER, DATE, TIME, OPEN, HIGH, LOW, CLOSE, VOL.
- Options:** "Добавить заголовок файла" (Add file header) is checked, "Заполнять периоды без сделок" (Fill periods without trades) is unchecked.

Рис. 1.3. - Настройка интерфейса получения котировок ОАО «Газпром».

Worldtradingdata (worldtradingdata.com) – американский ресурс предоставляющий API для загрузки биржевых данных. Данный ресурс отличается от других широким набором торговых площадок и инструментов, по которым можно извлечь данные.

После регистрации клиенту присваивается API токен, который используется в запросах на получение данных. API использует только http запросы, а данные предоставляются в формате JSON и CSV.

Worldtradingdata API позволяет загружать внутридневные данные с частотой 1 минута/15 минут/ 30 минут/ 1 час, исторические данные с частотой в 1 день/1 месяц/1 год. В сервис встроена система подписок и внутридневные данные предоставляются на платной основе.

Таблица 1.3.

Сравнение цен на подписки различных ресурсов

| Ресурс | Цена подписки |
|------------------|---------------|
| Kibot | 500\$ |
| QantQuote | 250\$ |
| Tickdatamarket | 730\$ |
| Worldtradingdata | 25\$ |

1.3. Инструменты для прогнозирования значений биржевых активов

В настоящее время на основе моделей машинного обучения реализованы различные библиотеки, программные продукты и интернет ресурсы, позволяющие анализировать и прогнозировать биржевые данные.

Инвестиционные компании и банки, также предлагают различные прогнозные значения по биржевым активам. В данном случае, как правило, используются комбинированные методы и модели прогнозирования. Согласно исследованию Bloomberg, большинство прогнозов банков и инвестиционных домов строится на основе экспертных оценок [26].

Торговые терминалы

Терминал Bloomberg – это компьютерная система, предоставляемая компанией Bloomberg, позволяющая специалистам в области финансов и других отраслей пользоваться сервисом Bloomberg Professional, через который пользователи в реальном времени могут контролировать и

анализировать финансовые рынки. Терминал сдается в аренду на двухлетний цикл. Стоимость подписки составляет около 2000\$ в месяц.

Данный терминал разработан в соответствии с клиент-серверной архитектурой с сервером, запущенным на Unix платформе. Клиент представляет собой Windows приложение, подключаемое к серверам через роутер, предоставленный Bloomberg. Вместе с терминалом клиенту предоставляется клавиатура, разработанная специально для Bloomberg Terminal. На данной клавиатуре расположены клавиши для быстрой навигации в терминале.

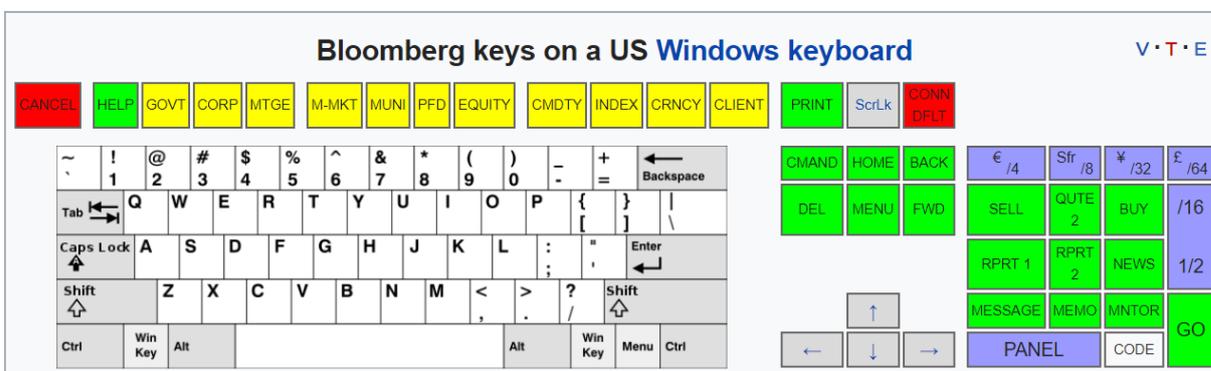


Рис. 1.4 – Клавиатура Bloomberg Terminal.

Bloomberg Terminal, использует статистические индикаторы, позволяющие пользователю анализировать рыночную ситуацию. В данном терминале реализована возможность выгрузки любых биржевых данных. Bloomberg Terminal API предоставляет возможность создавать собственное программное обеспечение, благодаря интеграции с языками C, C++, Java, Python и пр. Bloomberg Terminal является самым популярным программным продуктом для работы с биржей в мире [12].

Web ресурсы

Magic Finance (magicfinance.ap-southeast-1.elasticbeanstalk.com) – веб сервис позволяющий прогнозировать значения крупнейших мировых индексов. Данный ресурс, один из немногих, предоставляет возможность прогнозирования с помощью методов искусственного интеллекта и машинного обучения.

Magic Finance использует для прогнозирования многообразное обучение, сверточные нейронные сети и метод опорных векторов. Модель запускается в конце каждого торгового дня и прогнозирует будущий тренд. Точность прогнозирования составляет более 90% [14, с. 170].

PredictWallStreet (predictwallstreet.com) – веб ресурс агрегирующий множество экспертных оценок для прогнозирования цен на бирже. Данный веб сервис охватывает преимущественно биржевой рынок США и составляет прогноз практически по всем торгуемым активам. PredictWallStreet имеет большое сообщество пользователей. Количество успешных прогнозов составляет 76% от общего количества прогнозов на момент июня 2017 года.

Программные библиотеки, используемые для прогнозирования

Программные библиотеки представляют набор объектов и методов, разработанных для общего анализа данных. В большинстве своем такие библиотеки используют общие математические, статистические методы и методы искусственного интеллекта без привязки к специфике биржевой торговли.

Stocker является популярной библиотекой для Python, которая разработана специально для исследования биржевого рынка. С помощью данной библиотеки можно загружать исторические данные по более чем 10 000 биржевым активам. Загрузка данных в Stocker разработана на основе API веб ресурса Quandl – популярного поставщика данных. В основе Stocker используются непараметрические методы регрессии.

TensorFlow – открытая программная библиотека, разработанная компанией Google, для общего анализа данных с помощью нейронных сетей. Библиотека реализована для Python, C++, Java, Go и пр. TensorFlow является одной из самых популярных библиотек для работы с нейронными сетями [15]. API библиотеки совместим с API Keras – открытой нейросетевой библиотекой, нацеленной на работу с методами глубокого

обучения. Существует JavaScript версия библиотеки TensorFlow. Список библиотек для реализации нейронных сетей с сравнительными характеристиками приведен в приложении 1.

ГЛАВА 2. МОДЕЛИ НЕЙРОННЫХ СЕТЕЙ КАК СПОСОБ ПРОГНОЗИРОВАНИЯ БИРЖЕВЫХ ЗНАЧЕНИЙ

2.1. Развитие нейронных сетей

Первоначально дадим характеристику нейронным сетям, затем искусственным нейронным сетям. После этого перейдем к рассмотрению используемых в работе типов нейронных сетей – рекуррентные сети, многослойный персептрон.

Модели нейронных сетей – один из относительно недавних методов исследования числовых данных. Модели нейронных сетей состоят из преобразователей данных, которые называются нейроны. Нейроны моделируют биологические нейроны человеческого мозга, благодаря чему вычислительные системы, построенные в соответствии с моделями нейронных сетей, способны анализировать, адаптироваться и обучаться.

К текущему времени изучено и предложено большое количество различных моделей нейроподобных элементов и нейронных сетей [17].

Термин «нейронные сети» сформировался у исследователей, которые изучали основные принципы организации и функционирования биологических нейронных сетей. Впервые упоминания этого термина встречается в 40-х годах XX века. Главные результаты в данной области, обычно связывают с именами американских исследователей У. Маккалоха, Д. Хебба, Ф. Розенблатта, М. Минского, Дж. Хопфиллда и др.

Первые нейронные сети создала группа исследователей, объединив биологические и физиологические подходы. Первоначально они были выполнены как электронные сети. Но уже несколькими годами позднее они были перенесены в компьютерную среду, которая существует в настоящее время. Минский, Розенблатт и Уидроу разработали сети, которые состояли из одного слоя искусственных нейронов. Это вызвало большой успех в данной области. Искусственные нейроны, называемые также

персептронами, стали использоваться для широкого класса задач. Эти задачи включали в себя прогнозирование погоды, анализ электрокардиограмм, искусственное зрение.

Во многом это произошло благодаря модели Д. Хэбба. Именно он сформулировал в 1949 году закон, ставший отправной точкой для обучения нейронных сетей. Так, Хэбб предложил, что обучение заключается в изменении синоптических связей. Система, согласно данной теории самостоятельно обучается и выполняет требуемую задачу без особого вмешательства эксперта. Стоит отметить, что вклад М. Минского, благодаря его исследованию задачи «Исключающего ИЛИ» также имеет большое значение для развития нейронных систем [21, с. 417].

Персептрон – термин, введенный в 1958 году известным исследователем в данной области Ф. Розенблаттом. Данный нейрон в основном использовался для задач классификации. При обучении персептрон получал сообщение от «учителя». Благодаря данному сообщению объект классифицировался. Что немало важно, персептрон мог классифицировать ранее не использовавшиеся объекты. 1968-1985 года можно назвать затишьем в области исследования нейронных систем. С появлением машин, позволяющих проводить большое количество операций, а также параллельных машин интерес к данной области стал увеличиваться. Одной из основных работ, появившихся в то время стала работа Ф. Уоссермена «Нейрокомпьютерная техника». Далее работа Д. Хопфилда (1982) поднимает уровень работы с нейронными сетями на новый уровень. Данный исследователь, основываясь на правилах обучения Хэбба показывает, что задачи с нейронами могут быть сведены к обобщению ряда моделей, разработанных к тому времени в физике неупорядоченных систем.

В 1986 году появляется известная работа Вильямса, Румельхарта и Хинтона, которая освещает вопросы касающиеся работы и поведения слоистых нейронных сетей [18].

В целом, можно сказать, что в 80-е года сформировался прочный теоретический фундамент нейронных сетей. С развитием вычислительных систем данный фундамент позволил создать большое количество нейронных систем, которые в последующие два десятилетия стали использовать для очень широкого круга задач. Стали появляться организации, занимающиеся разработкой программного обеспечения, конструирующие нейронные сети. В финансовом секторе нейронные сети стали показывать большую эффективность при их использовании в решении задач прогнозирования спроса на продукцию, а также в анализе платежеспособности клиентов банка.

Один из примеров использования методов глубокого обучения – это исследование 2007 года канадского исследователя Джеффри Хинтона. Он создал алгоритмы глубокого обучения нейронных сетей. При обучении нижних слоев сети, Хинтон использовал ограниченную машину Больцмана. Машина Больцмана представляет собой стохастическую рекуррентную нейронную сеть. Данная обученная сеть способна распознавать в лица на фотографиях и в настоящее время используется практически во всех цифровых фотоаппаратах, а также в интернет-поисковиках при классификации изображений [18].

В настоящее время, по оценкам специалистов, имеются противоречивое понимания развития нейронных сетей. За последние годы в этой был сильный технологический скачек. Было разработано множество алгоритмов и сетей. По словам американского ученого Кита Хесбора нейронные сети переживают остановку развития. По его же словам во многом это связано с имеющимися технологическими доступными мощностями. С другой стороны, мы можем явно полагать, что закон Мура, прогнозирующий неуклонный рост транзисторов в вычислительных системах – работает до сих пор, а, следовательно, вопрос доступной мощности лишь дело времени. Нейронные сети остаются важной частью

анализа данных и используются в настоящий момент практически во всех областях человечества.

Несмотря на скачек в развитии нейронных сетей, проведение аналогии работы с человеческим мозгом и образом мышления не совсем правильная. В мозге нет областей где бы процесс обработки информации стал ясен до конца.

2.2 Математическая модель нейронных сетей

Искусственная нейронная сеть представляет собой математическую модель, а также аппаратную и программную реализацию данной модели. Особенностью является, что данная модель работает по принципу биологических нейронных сетей – клеток живого организма. Во многом данное понятие сформировалось при попытке смоделировать процессы, протекающие в человеке.

Искусственная нейронная сеть представляет систему процессоров, которые соединены между собой. Каждый из этих процессоров принимает и обрабатывает сигналы от других процессоров. Подобным образом реализуется система слоев нейронов.

С точки зрения математического анализа нейронные сети представляют собой способ решения нелинейных задач оптимизации, проблемы эффективного параллелизма [20].

Биологические нейроны – основной компонент мозга человека, а также его нервной системы. Между данными нейронами передаются электрические импульсы. Строение биологического нейрона:



Рис. 2.1 – Схема работы биологического нейрона.

Дендриты – компоненты, принимающие импульс нейрона. Далее аксон – канала предающий импульс. Синапсами являются специальные образования, влияющие на силу импульса, для контакта аксона и дендрита. Каждый раз при прохождении синапса сила импульса меняется определенное число раз. Это называется вес синапса. Когда к нейрону по нескольким дендритам поступают, они начинают суммироваться. Если в этот момент у суммарного импульса превышен порог, то нейрон переходит в состояние, когда начинает формировать свой собственный импульс, который посылается дальше по аксону. Так как веса синапса могут меняться со временем, поведение нейрона также может меняться [21, с. 118]. Математическую модель данного процесса можно представить следующим образом:

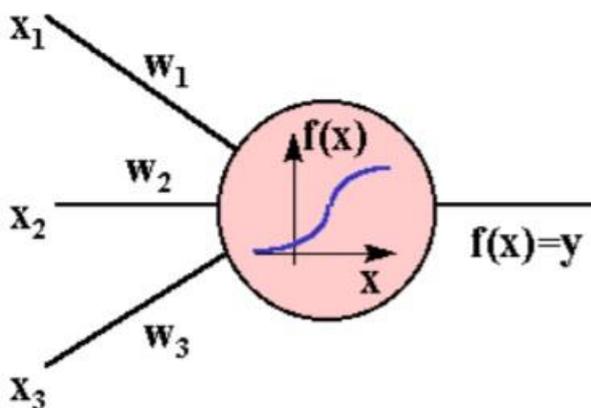


Рис. 2.2 – Математическая модель нейрона.

На модели выше видно нейрон с тремя входами или дендритами, где синапсы имеют веса w_1, w_2, w_3 , к которым поступают силы x_1, x_2, x_3 соответственно. Таким образом, после синапсов и дендритов к нейрону поступают импульсы x_1w_1, x_2w_2, x_3w_3 . В самом нейроне полученные импульсы преобразуются согласно передаточной функции $F(x)$.

Искусственный нейрон выглядит похожим способом. На вход искусственного нейрона поступают сигналы, которые одновременно являются выходами другого нейрона. Такой вход умножается на вес, затем произведения суммируются, определяя уровень активации нейрона. Данную модель можно схематично изобразить на рисунке:

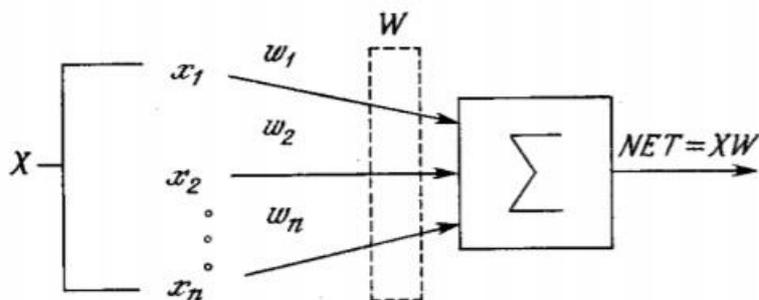


Рис. 2.3 – Математическое представление работы нейрона.

Множество сигналов x_1, x_2, \dots, x_n поступает на вход нейрона. Данные сигналы в совокупности можно обозначить вектором X . Они вполне схожи по принципу работы с биологическим нейроном. Затем сигналы также умножаются на веса w_1, w_2, \dots, w_n и суммируются в суммирующем блоке Σ . Каждый вес равен одной синаптической связи, аналогичной биологическому нейрону. Выход, который обычно именуется NET, создается из суммирующего блока, где все взвешенные элементы складываются алгебраически [23].

Выходящий сигнал преобразуется с помощью линейной функции, называемой активационной. Данная функция обозначается как F и дает выходной сигнал OUT.

Схематичную модель искусственного нейрона с активационной функцией можно представить на рисунке:

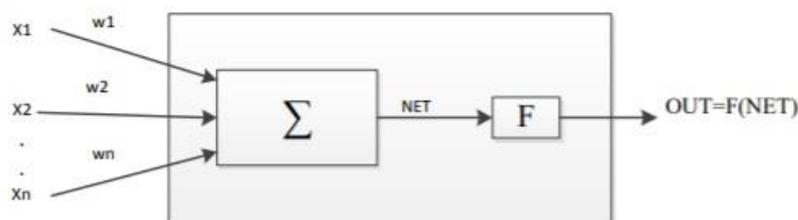


Рис. 2.4 – Модель работы нейрона с активационной функцией.

Блок, обозначенный F, принимает сигнал NET и выдает сигнал OUT. F, в данном случае – сжимающая функция, в случае если при любых значениях NET значения OUT принадлежат конечному интервалу. Несмотря на то, что модель искусственного нейрона игнорирует множество биологических процессов стоит отметить исключительное сходство живого нейрона и искусственного.

Математическая характеристика понятий: активационная функция, перцептрон, сигмоидальный нейрон.

Активационная функция характеризует собой зависимость сигнала на выходе нейрона от суммы сигналов на его входах. Обычно такая функция является либо монотонно возрастающей и находится в области значений $[-1, 1]$, что является гиперболическим тангенсом и $[0, 1]$ – сигмоида. Искусственный нейрон характеризуется своей активационной функцией. К примеру, нейрон, характеризующийся сигмодой называется сигмоидальный нейрон [24].

К основным активационным функциям относятся:

- Пороговая активационная функция или функция Хевисайда. Ее нельзя использовать для алгоритма обратного распространения ошибки;

$$f(x) = \begin{cases} 1 & \text{if } x \geq -w_0x_0 \\ 0 & \text{else} \end{cases} \quad (2.1)$$

- Сигмоидальная активационная функция;

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.2)$$

- Гиперболический тангенс;

$$\tanh(Ax) = \frac{e^{Ax} - e^{-Ax}}{e^{Ax} + e^{-Ax}} \quad (2.3)$$

Персептрон является типом искусственного нейрона, разрабатываемый Розенблаттом в 1950-ых и 1960-ых годах. В настоящее время, в последних исследованиях все чаще используют другую модель искусственного нейрона – сигмоидальный нейрон. Чтобы понять принцип его работы, необходимо рассмотреть принцип работы персептрона. Персептрон принимает на вход значения X и выдает бинарный результат. Розенблатт предложил использовать веса, числа, выражающие важность каждого входящего значения. Сумма значений сравнивается с некоторым пороговым значением threshold , и по результатам определяется выходной сигнал – 0 или 1. Данное пороговое значение также определяется параметром нейрона.

Персептроны могут быть классифицированы как искусственные сети:

- с одним скрытым слоем;
- с пороговой активационной функцией;
- с прямым распространением сигнала;

Обучение персептрона состоит в изменении матрицы весов.

Существует четыре исторически сложившихся видов персептронов:

- персептрон с один скрытым слоем;

- однослойный персептрон: входные элементы которого напрямую соединены с выходными с помощью системы весов. Данный вид обучения является простейшей сетью прямого распространения;
- многослойный персептрон по Розенблатту. В данном случае присутствуют дополнительные скрытые слои;
- многослойный персептрон по Румельхарту. Скрытые слои также присутствуют, но обучение проводится по методу обратного распространения ошибки;

Желаемое поведение весов можно было бы с легкостью осуществить, если бы небольшое изменение весов приводило бы к небольшому изменению на выходе в сети, но активационные функции работают иначе. Даже небольшое изменение весов или смещение персептрона способно кардинально изменить выходное значение персептрона, к примеру, с 0 до 1. Поэтому даже самое незначительно изменение одного из элементов может создать значительные трудности в понимании поведения сети [29].

Сигмоидальный нейрон в значительной степени похож на персептрон, однако небольшие изменения в весах не так влияют на выход нейрона. На вход подаются значения между 0 и 1. На выходе также выдается значение между 0 и 1. Это происходит потому что в качестве активационной функции выступает сигмоида, являющаяся нелинейной. График кривой сигмоиды представлен на рисунке:

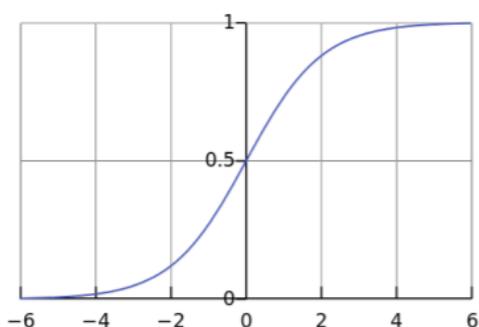


Рис. 2.5 - График логистической кривой сигмоиды.

Чем больше b (параметр наклона сигмоидальной функции активации), тем сильнее крутизна данного графика. Если b стремиться к бесконечности, то сигмоида стремиться к функции Хевисайда.

$$f(x) = \frac{1}{1 + e^{-\beta x}} \quad (2.4)$$

Здесь β – параметр наклона сигмоидальной функции активации;

Одним из важнейших свойств сигмоидальной функции является ее непрерывность. Применение непрерывной функции активации позволяет использовать при обучении градиентные методы.

К обучению нейронных сетей относят следующие термины: эпоха, размер серии, количество итераций. Эпоха представляет собой прямой и обратный проход по всем тренировочным параметрам. Размер серии – количество тренировочных примеров для одной итерации прямого и обратного проходов. Количество итераций – количество проходов. Один проход = прямой проход + обратный проход. Таким образом имея 1000 примеров, размер серии = 500, нам потребуется две итерации, чтобы завершить одну эпоху [25].

Одна из наиболее известных стратегий обучения сети – алгоритм обратного распространения ошибок. Данный алгоритм определяет стратегию подбора весов с применением градиентных методов оптимизации. Целевая функция, обычно определяется как квадратичная разность суммы между фактическими и ожидаемыми значениями, является непрерывной, то градиентные методы оптимизации являются эффективными.

Пусть имеется некий конечный набор тренировочных данных (m примеров). Для обучения применяется градиентный спуск. Квадратичная ошибка для примера будет высчитана по формуле:

$$J(W, b; x, y) = \frac{1}{2} \|h_{W,b}(x) - y\|^2 \quad (2.5)$$

В таком случае целевая функция будет выглядеть так:

$$\begin{aligned}
 J(W, b) &= \left[\frac{1}{m} \sum_{i=1}^m J(W, b; x^{(i)}, y^{(i)}) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \\
 &= \left[\frac{1}{m} \sum_{i=1}^m \left(\frac{1}{2} \|h_{W,b}(x^{(i)}) - y^{(i)}\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2
 \end{aligned} \tag{2.6}$$

Здесь W – веса модели;

b – нейрон смещения;

λ – параметр регуляризации весов;

первый член выражения – сумма квадратов ошибок;

второй члены выражения – регуляризация, позволяющий уменьшить значение весов и предотвратить переобучение;

Параметр регуляризации весов используют для проверки значимости частей данного выражения, чтобы минимизировать квадратичную ошибку $J(W, b)$. Для обучения нейронной сети необходимо инициализировать каждый параметр W и b малыми случайными величинами, близкими к нулю, а затем применить алгоритмы оптимизации, например, упоминавшийся градиентный спуск [30].

Так как $J(W, b)$ не является выпуклой функцией, то градиентный спуск восприимчив к локальным оптимумам. Каждая итерация градиентного спуска обновляет параметры следующим образом:

$$\begin{aligned}
 W_{ij}^{(l)} &= W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) \\
 b_i^{(l)} &= b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(W, b),
 \end{aligned} \tag{2.7}$$

где α – скорость обучения.

$$\begin{aligned}\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) &= \left[\frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x^{(i)}, y^{(i)}) \right] + \lambda W_{ij}^{(l)} \\ \frac{\partial}{\partial b_i^{(l)}} J(W, b) &= \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial b_i^{(l)}} J(W, b; x^{(i)}, y^{(i)})\end{aligned}\quad (2.8)$$

Основные шаги присущие алгоритму обратного распространения ошибки.

- 1) Осуществляется прямой проход по сети, вычисляются активации слоев L1, L2, L3 и так далее до выходного слоя Ln'
- 2) Для каждого входного элемента i в выходном слое n рассчитывается ошибка

$$\delta_i^{(n_i)} = \frac{\partial}{\partial z_i^{(n_i)}} \frac{1}{2} \|y - h_{W,b}(x)\|^2 = -(y_i - a_i^{(n_i)}) \cdot f'(z_i^{(n_i)})\quad (2.9)$$

- 3) Для $l = n_l-1, n_l-2, n_l-3, \dots, 2$, каждого элемента в слое l , рассчитывается

$$\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_j^{(l+1)} \right) f'(z_i^{(l)})\quad (2.10)$$

- 4) Вычисляются частные производные

$$\begin{aligned}\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x, y) &= a_j^{(l)} \delta_i^{(l+1)} \\ \frac{\partial}{\partial b_i^{(l)}} J(W, b; x, y) &= \delta_i^{(l+1)}\end{aligned}\quad (2.11)$$

2.3 Многослойный персептрон

Многослойная сеть формально определяется как совокупность элементарных обрабатывающих элементов, называемый нейронами. Нейронные сети прямого распространения имеют входной сигнал, который

передается от слоя к слою. Такие сети принято называет многослойным персептроном. Сеть состоит из входного слоя, скрытых вычислительных внутренних слоев и выходного слоя. Данная сеть является однонаправленной сетью сигмоидального типа.

Структуру схему многослойного персептрона можно представить следующим образом:

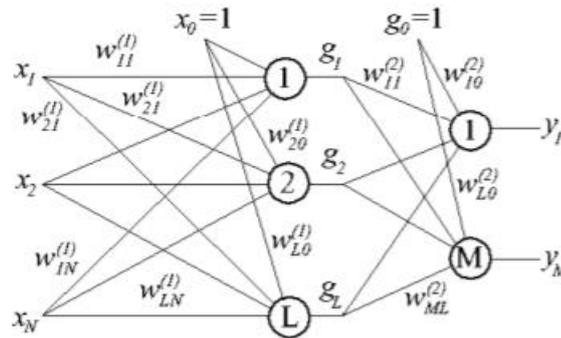


Рис 2.9 - Структурная схема многослойного персептрона.

Обозначения:

- $g_l, l=1,2,\dots,L$ – сигналы, выходящие из первого слоя нейронов;
- $m=1,2,\dots$ - индексы в скобках сверху означают номер слоя нейрона;
- x_1, x_2, \dots, x_N – входные сигналы, которые формируют входной слой;
- g_1, g_2, \dots, g_L - первый скрытый слой;
- y_1, y_2, \dots, y_M – выходные сигналы, которые формируют слой на выходе.

Формулы для расчета выходных сигналов нейронов:

$$g_l = f(\sum_{j=0}^N w_{lj}^{(1)} \cdot x_j), l = 1, 2, \dots, L$$

$$y_i = f(\sum_{l=0}^L w_{il}^{(2)} \cdot g_l) = f(\sum_{l=0}^L w_{il}^{(2)} \cdot f(\sum_{j=0}^N w_{lj}^{(1)} \cdot x_j)), i = 1, 2, \dots, M \quad (2.11)$$

Целевая функция для единичной обучающей выборки $\langle X, D \rangle$ выглядит следующим образом:

$$E(W) = \frac{1}{2} \sum_{i=1}^M (y_i - d_i)^2 \quad (2.12)$$

Целевая функция для множества обучающих пар $\langle X_k, D_k \rangle$, $k = 1, 2, \dots, p$ имеет вид суммы по всем парам:

$$E(W) = \frac{1}{2} \sum_{k=1}^p \sum_{i=1}^M (y_i - d_i)^2$$

Цель обучения многослойного персептрона состоит в том, что при входном векторе X_k необходимо подобрать такие значения весов сети $W_{ij}(1)$ и $W_{ij}(2)$, чтобы выходной вектор Y_l и вектор ожидаемых значений D_k максимально совпадали.

Одним из методов обучения такого типа сетей также является алгоритм обратного распространения ошибки. Предполагается, что вектор будет иметь два прохода. Вектор начинает движение с входного слоя, далее от слоя к слою, при этом самой сетью генерируется набор выходных сигналов.

Во время обратного прохода синаптические веса настраиваются таким образом, что из желаемого выхода сети вычитается фактический, после чего поступает сигнал ошибки. Далее сигнал распространяется по сети в противоположном направлении.

Для уточнения коэффициентов необходима формула метода градиента:

$$w^{k+1} = w^k - \eta \cdot \text{grad}E(w^k) \quad (2.13)$$

В данной формуле основную сложность представляет расчет компонентов градиента целевой функции. Формула:

$$E(W) = \frac{1}{2} \sum_{i=1}^M (f(\sum_{l=0}^L w_{il}^{(2)} \cdot g_l) - d_i)^2 = \frac{1}{2} \sum_{i=1}^M (f(\sum_{l=0}^L w_{il}^{(2)} \cdot f(\sum_{j=0}^N w_{lj}^{(1)} x_j)) - d_i)^2 \quad (2.14)$$

Производные целевой функции по весам нейронов выходного слоя:

$$\frac{\partial E(W)}{\partial w_{il}^{(2)}} = (y_i - d_i) \cdot \frac{\partial f(u_i^{(2)})}{\partial u_i^{(2)}} \cdot g_l, \text{ где } u_i^{(2)} = \sum_{l=0}^L w_{il}^{(2)} \cdot g_l \quad (2.15)$$

Компонент вектора градиента имеет вид:

$$\frac{\partial E(W)}{\partial w_{il}^{(2)}} = \delta_i^{(2)} \cdot g_l \quad (2.16)$$

Компоненты для весов предпоследнего слоя:

$$\frac{\partial E(W)}{\partial w_{ij}^{(1)}} = \delta_i^{(1)} \cdot x_j \quad (2.17)$$

Анализ данных формул позволяет сформулировать правило расчета вектора градиента для весов в каждом слое:

$$\frac{\partial E(W)}{\partial w_{ij}^{(k)}} = \delta_i^{(k)} \cdot x_j \quad (2.18)$$

Модель многослойной сети сигмоидального типа и метод ее обучения, несмотря на свою простоту и удобство имеет трудности, которые либо замедляют процесс обучения, либо вовсе не позволяют сети обучиться. Например, значения весов могут вырасти до очень больших величин. Такая ситуация приводит к тому, что большая часть нейронов на выходе даст огромные значения, а производная в этих значениях будет очень мала.

Процесс обучения может быть остановлен, так как ошибка, посылаемая обратно будет пропорциональна этой производной. Так же необходимо выбрать размер шага, который не будет конечным. Данный алгоритм имеет доказательство сходимости. Сходимость будет либо слишком медленная, если размер шага будет очень мал, либо будет претерпевать постоянную неустойчивость.

2.4 Рекуррентные нейронные сети

В сетях прямого распространения используется единственный выход, который полностью определяет активации всей нейронов в оставшихся слоях. Такую сеть невозможно обучить прогнозировать события, например, в прогнозировании биржевых значений, т.к. неясно как бы могла быть использована информация о предыдущих значениях ряда. Рекуррентные сети могут решить эту проблему. Имею внутри себя циклы рекуррентные сети позволяют информации сохраняться: поведение скрытых нейронов будет определяться не только активацией в других скрытых слоях, но и полученными ранее активациями самих нейронов.

RNN может быть представлена в качестве множества копий одной сети. В данной сети каждая копия передает сообщения другой копии. Имея подобную цепочку структур, как последовательности и списки, RNN является естественной архитектурой нейронной сети, используемой для таких данных.

RNN способна обуславливать модель по всем предыдущим значениям, обработанным из ряда.

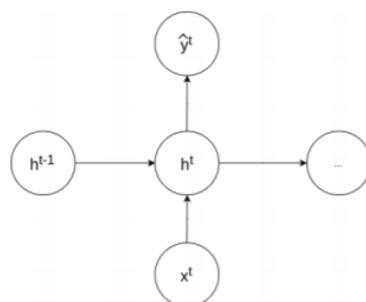


Рис. 2.6 - Нейрон рекуррентной сети.

Каждый слой содержит нейроны, каждый из которых выполняет операцию линейной матрицы на своих входах, после чего следует нелинейная операция. На каждом временном шаге выходные данные предыдущего шага вместе со следующим вектором значение X_t , представляет собой входные данные для скрытого слоя для создания прогнозирования y^t и признаков H_t :

$$\begin{aligned}
 h_t &= \sigma(W^{(hh)}h_{(t-1)} + W^{(hx)}x_t) \\
 \hat{y}_t &= \text{softmax}(W^{(S)}h_t)
 \end{aligned}
 \tag{2.12}$$

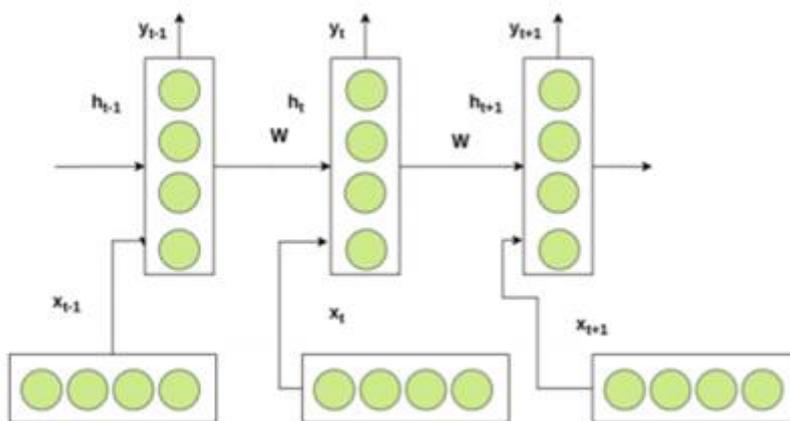


Рис. 2.7 - Рекуррентная нейронная сеть, три временных шага.

- $X_1, \dots, X_t, X_{t+1}, \dots, X_T$ – векторы признаков;
- формула h_t позволяет вычислить входные признаки скрытого слоя на каждом временном шаге t ;
- Y_t : выходное распределение вероятности на каждом временном шаге t ;

В качестве функции потерь используется функция ошибки перекрестной энтропии.

Объем памяти, необходимый для запуска слоя RNN, пропорционален количеству признаков. То есть k признаков, будет храниться в памяти как k векторов. Размер матрицы не масштабируется в зависимости от количества значений признаков. Для рекуррентной сети, состоящей из 1000 рекуррентных слоев, размер матрицы будет всегда 1000 на 10000.

Очень важно различать понятия рекуррентной и рекурсивной нейронной сети. Рекурсивная сеть – это обобщение рекуррентной. В рекуррентной сети все веса общие и размерность остается одинаковой по всей длине последовательности. Все W_{xh} веса будут общими и таким же

будет вес W_{hh} из-за того, что все происходит в единственном нейроне, развертываемом во времени.

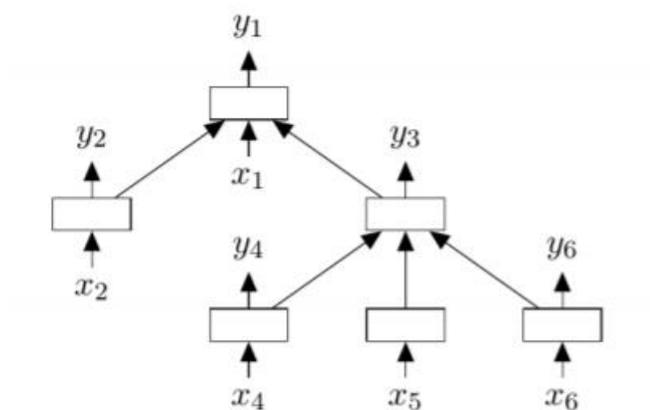


Рис. 2.8 - Схема рекурсивной нейронной сети.

ГЛАВА 3. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ ЗНАЧЕНИЙ ИНДЕКСА S&P500 С ПОМОЩЬЮ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

3.1. Разработка и этапы проектирования приложения для прогнозирования значений S&P500

Для прогнозирования значения индекса S&P500 разработано web-приложение на языке базе JavaScript технологий с использованием библиотеки TensorFlow.js. Все дополнительные модули для работы установлены с помощью пакетного менеджера NPM, с помощью команды `npm install <имя_модуля>`.

Web-приложение состоит из четырех основных компонентов – серверная часть, клиентская часть, база данных, API Worldtradingdata.

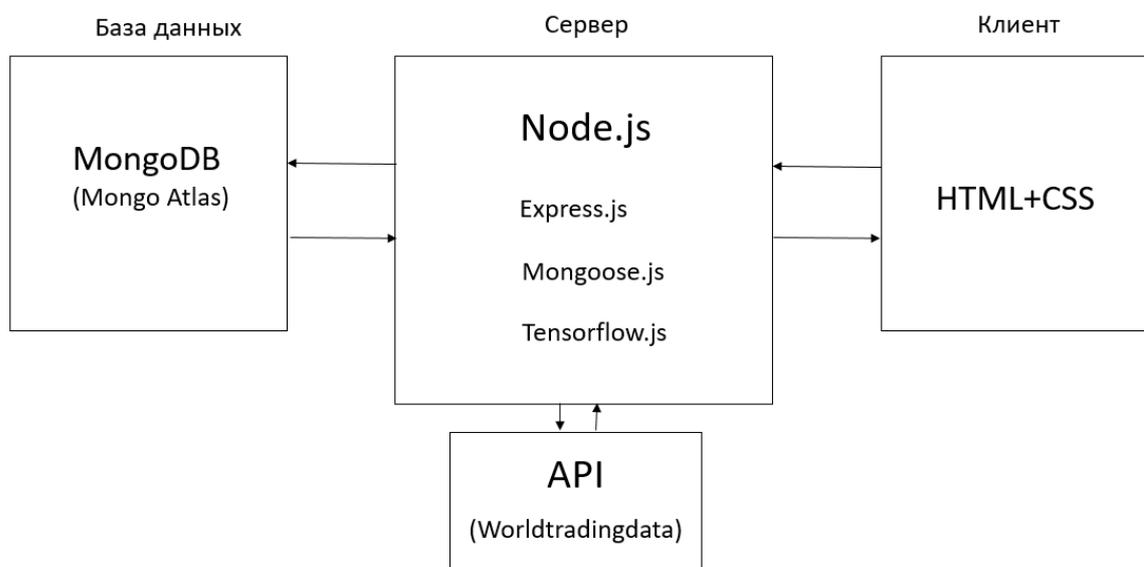


Рис. 3.1 – Основные компоненты приложения.

Разработка серверной части

Серверная часть реализована с помощью среды выполнения Node.js и содержит основные модули Node, отвечающие за обработку данных и взаимодействие компонентов приложения:

- HTTP-сервер;
- Модуль для работы с базой данных;
- Модуль для работы с API Worldtradingdata;

HTTP-сервер разработан с помощью модуля Express.js. Сервер обрабатывает GET запросы и осуществляет операции:

- выгрузку данных из API Worldtradingdata;
- сохранения данных в базу данных;
- выгрузка данных из базы данных;
- обучение модели;
- построение прогнозной модели;

Работа с базой данных MongoDB реализована с помощью модуля Mongoose.js. Для установления соединения с ресурсом, который занимается хостингом базы данных, в методе `mongoose.connect()` необходимо прописать параметры соединения (URI). Схема базы данных реализуется с помощью инициализации нового объекта `new Schema()` и представляет собой объект из массива чисел с плавающей точкой и строки, содержащий название компании или «SP500».

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;
const finSchema = new Schema({
  company : {
    name : String,
    price : [Number]
  }
})
mongoose.model('fin', finSchema);
```

Рис. 3.2 – Задание схемы и инициализация модели.

Модуль для работы с API реализован самостоятельно, при помощи стороннего модуля отправки запросов Request.js. При регистрации и приобретении подписки API Worldtradingdata, пользователю выдается api-token, которой используется в запросах к серверам Worldtradingdata. Запрос

к API осуществляется с помощью задания функции request(URL, cb()). Полный код взаимодействия с API и данными приведен в приложении 2.

Даже при наличии подписки доступ к минутным значениям цен ограничен. Самый поздний торговый день, доступный для загрузки минутных значений цен – 31 Августа 2017 года. Возможный временной период загрузки 5 месяцев. Таким образом с помощью API Worldtradingdata выгружено 41266 минут по 500 компаниям и индексу S&P500 с 03 Апреля 2017 года по 31 Августа 2017 года. Данные представляют собой вещественные числа с двумя знаками после запятой.

Разработка клиентской части

Клиентская часть представляет собой веб-страницу, реализованную с помощью языка разметки HTML и оформленную с применением стилей CSS. В процессе работы приложения реализовано обновление статусов показывающих процесс загрузки данных, готовность данных для обучения, процесса обучения. Обработка статусов реализована с помощью методов document – getElementById() и изменения значения переменной document.textContent.

После обучения и расчета прогнозных значений в приложении визуализируются прогнозные и действительные значения индекса S&P500 с помощью графиков. Визуализация разработана с помощью библиотеки tfjs-vis.

Разработка базы данных

База данных реализована с помощью NoSQL базы данных MongoDB. В терминологии MongoDB – база данных состоит из коллекций (таблиц, в терминологии РСУБД), а коллекции состоят из документов (строк, в терминологии РСУБД). Каждый документ представляет собой объект, хранящий пары ключ-значение.



Рис. 3.3 – Схема базы данных приложения.

Для хостинга базы данных выбран интернет-ресурс MongoDB Atlas (mongodb.com/cloud/atlas). Данный ресурс позволяет в академических целях использовать реализацию MongoDB и предоставляет объем хранения до 4Гб бесплатно.

Настройка MongoDB Atlas осуществляется в графическом интерфейсе в несколько этапов:

- Создание аккаунта;
- Создание проекта и кластера, содержащего базу данных;
- Настройка разрешений для подключения (IP, порт и пр.);

Характеристика базы данных приложения после загрузки данных в MongoDB Atlas:

- Количество кластеров: 1;
- Количество коллекций: 1;
- Имя коллекции: fins;
- Размер коллекции: 290.05Мб;
- Всего документов в коллекции: 502;
- Размер индексов: 36Кб;

```
  _id: ObjectId("5cf7bff91ede2b101cae0c67")
  company: Object
    price: Array
      name: "SP500"
    __v: 0
```

Рис. 3.4 – Пример документа в MongoDB.

3.2 Предварительная обработка данных и программная реализация моделей нейронных сетей

Данные содержат $n = 41266$ минут, которые составляют диапазон времени с Апреля по Август 2017 года. По каждой минуте есть соответствующие значения цен каждой из 500 акций, входящих в состав индекса S&P500, а также значения цен самого индекса.

Этапы обработки данных:

- Очистка данных;
- Создание выборки данных;
- Преобразование данных;

Этап очистки данных включает восполнение пропущенных значений, обнаружение и удаление искаженных данных. Обнаружение и удаление искаженных данных осуществляется ресурсом Worldtradingdata. Восполнение пропущенных значений реализовано по принципу «последнее наблюдение переносится вперед».

Этап создания выборки данных заключается в разделении данных на тренировочный и тестовый наборы. Тренировочный и тестовый набор данных содержит 80% и 20% от общего набора данных соответственно. Данные смещены на одну минуту вперед. Операция смещения данных необходима, т.к. мы обучаем выборку прогнозировать значения цен в следующую минуту по текущим значениям цен. Таким образом каждая строка в наборе данных содержит значение S&P500 в момент времени $t+1$ и

соответствующие данному значению цены 500 компаний, входящих в индекс, в момент времени t .

Этап преобразования данных заключается в нормализации данных и масштабировании их значений в диапазоне $[0, 1]$. Масштабирование данных выполненное в Python с помощью Sklearn функции `MinMaxScaler()`. Передача переменной из скрипта Python на сервер Node осуществляется с помощью модуля `Child_Process.js` и метода `spawn()`.

```
var spawn = require('child_process').spawn,  
ls       = spawn('python', ['pyserial.py']);
```

Рис. 3.5 – Передача переменной из Python на сервер Node.

Входные данные для обучения представляют собой 2-мерный вектор размерностью 33012×500 , вектор значений для обучения – 1-мерный вектор, содержащий 33012 значений.

Входные данные для прогнозирования представляют собой 2-мерный вектор размерностью 8254×500 , выходные данные – 1-мерный вектор, содержащий 8254 значения.

С помощью библиотеки `TensorFlow.js` реализована модель многослойного персептрона (MLP) и модель рекуррентной нейронной сети (RNN).

Для реализации моделей нейронных сетей `TensorFlow.js` использует методы `tf.keras`. `Keras` – это высокоуровневый API для создания моделей машинного обучения в `TensorFlow`. Модель представляет собой граф, состоящий из нескольких слоев, идущих друг за другом. Для реализации модели необходимо:

- Инициализировать модель с помощью функции `tf.keras.Sequential()`;
- Произвести конфигурацию слоев;
- Произвести конфигурацию функции стоимости, оптимизатора, инициализатора;

- Произвести обучение модели с помощью метода compile;

Модель MLP состоит из четырех скрытых слоев. Каждый слой содержит 1024, 512, 256 и 128 нейронов соответственно. Функцией активации является функция выпрямитель (ReLU).

```
export function buildMLPModel(inputShape) {
  const model = tf.sequential();
  model.add(tf.layers.flatten({inputShape}));
  model.add(
    | tf.layers.dense({units: 1024, activation: 'relu'}));
  model.add(
    | tf.layers.dense({units: 512, activation: 'relu'}));
  model.add(
    | tf.layers.dense({units: 256, activation: 'relu'}));
  model.add(
    | tf.layers.dense({units: 128, activation: 'relu'}));
  model.add(tf.layers.dense({units: 128}));
  return model;
}
```

Рис. 3.6 – Функция, реализующая MLP модель.

Модель RNN состоит из четырех слоев – входной слой, reshape-слой, RNN-слой и выходной слой. В соответствии с структурной входных данных в качестве первого слоя используется двумерный входной слой. Вторым слоем является слой рекуррентной нейронной сети. Чтобы повысить качество модели используется RNN с применением LSTM-сеток.

```
export function buildSimpleRNNModel(inputShape) {
  const input_layer_neurons = 100;
  model.add(
    | tf.layers.dense({units: input_layer_neurons, inputShape: [input_layer_shape]}));
  model.add(
    | tf.layers.reshape({targetShape: rnn_input_shape}));
  model.add(
    | tf.layers.rnn({cell: lstm_cells, inputShape: rnn_input_shape, returnSequences: false}));
  model.add(
    | tf.layers.dense({units: output_layer_neurons, inputShape: [output_layer_shape]}));
  return model;
}
```

Рис. 3.7 – Функция, реализующая RNN модель.

Определим функцию стоимости для моделей, которая используется для генерации оценки отклонения между прогнозами сети и реальными результатами наблюдений в ходе обучения. Для решение проблем с

регрессией используем функцию средней квадратичной ошибки (MSE). Функция вычисляет среднее квадратичное отклонение между действительными и прогнозными значениями:

```
const mse = tf.reduce_mean(tf.squared_difference(out, Y))
```

Рис. 3.8 – Реализация MSE.

Оптимизатор осуществляет вычисления, требующиеся для адаптации весов и переменных отклонений нейросети в ходе обучения. Эти вычисления ведут к подсчетам так называемых градиентов, которые обозначают направление необходимого изменения отклонений и весов. В приложении реализован оптимизатор адаптивной оценки моментов.

```
const opt = tf.train.AdamOptimizer().minimize(mse);
```

Рис. 3.9 – Реализация оптимизатора.

Перед началом обучения также необходимо реализовать инициализаторы, которые задают начальные веса. В приложении реализована стандартная стратегия реализации с помощью метода `tf.variance_scaling_initializer()`.

```
const opt = tf.train.AdamOptimizer().minimize(mse);
const sigma = 1
const weight_initializer = tf.variance_scaling_initializer(mode="fan_avg",
distribution="uniform",
scale=sigma)
const bias_initializer = tf.zeros_initializer()
```

Рис. 3.10 – Реализация инициализатора.

3.3 Обучение моделей и прогнозирование значений индекса

После реализации всех основных компонентов приложения произведем обучение сети и проанализируем полученные результаты.

Для обучения сети воспользуемся методом мини-партий. В процессе этого метода обучения из набора данных извлекаются случайные выборки

данных $n = \text{batch_size}$ и загружаются в сеть. Таким образом набор данных для обучения представлен как набор состоящий из $n/\text{batch_size}$ частей.

В процессе обучения одна партия проходит через всю сеть, затем модель сравнивает полученные прогнозные значения с действительными в текущей партии данных. Затем выполняется процесс оптимизации и параметры сети, веса и отклонения обновляются. Процесс повторяется до того момента пока все партии не будут обработаны сетью. Полный обработки цикл называется эпохой.

Обучение сети останавливается при достижении максимального количества эпох. Количество эпох для приложение равно 50, размер партий – 256.

Для обучения модели используется функция `fit()`:

```
model.fit(X_train, Y_train, epochs=50)
```

Рис. 3.10 – Обучение модели.

После прохождения всех эпох тестовые данные прогнозируются с помощью функции `tf.Session().run()`, а затем визуализируются с помощью встроенной в TensorFlow.js библиотеки `tf-vis` (см. рис. 3.11 – 3.12).

В модели многослойного персептрона точность прогнозирования составила 55,2%. Значения среднеквадратичного отклонения и точности составляют: 0,412 и 0,552 соответственно.

При применении модели рекуррентных сетей точность прогнозирования индекса возросла до 63,7%. Значения среднеквадратичного отклонения и точности составляют: 0,319 и 0,637 соответственно.

Данные результаты являются приемлемыми для задачи прогнозирования значений. В данном исследовании агрегированный индекс прогнозируется по значениям активов, входящих в данный индекс. Данный подход дал приемлемый результаты для прогнозирования краткосрочной торговли индексом.

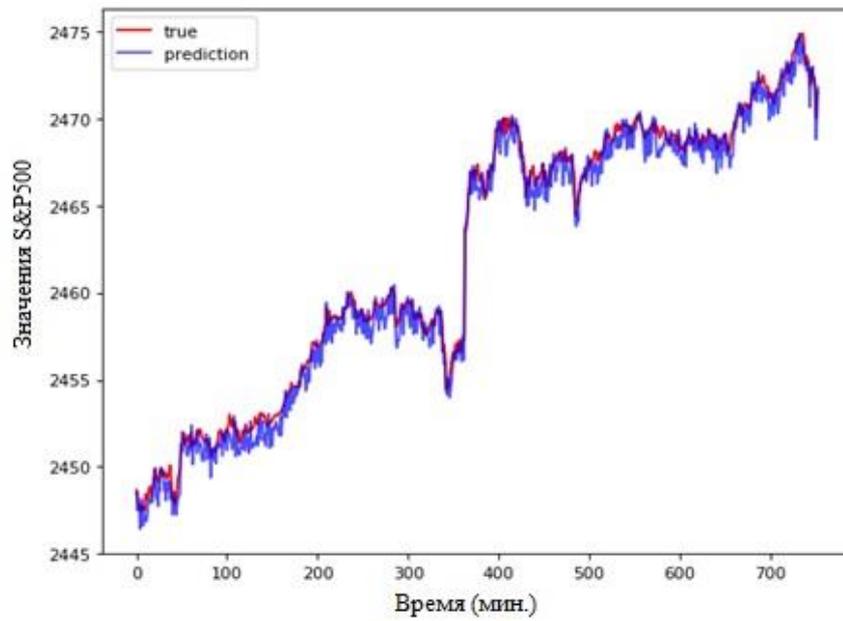


Рис. 3.11 - Действительные и прогнозные значения модели MLP.

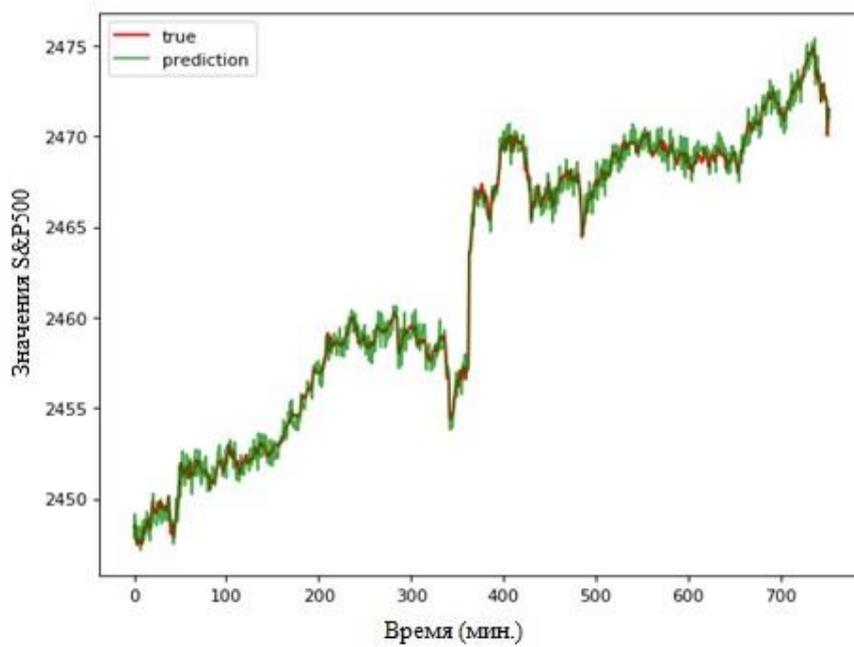


Рис. 3.12 - Действительные и прогнозные значения модели RNN.

ЗАКЛЮЧЕНИЕ

В данной работе было проведено исследование прогнозирования будущих значений индекса S&P500 на основе значений цен 500 акций, входящих в состав индекса.

Исследовательские работы, посвященные задаче прогнозирования цен на фондовых рынках, выделяют большое количество разнообразных подходов и моделей для ее решения.

Все существующие модели можно разделить на две категории – качественные или количественные.

Качественные модели – это модели, где фактор ценообразования определяется самим экспертом, проводящим анализ. При данном подходе факторы ценообразования, а также прогностический результат во многом определяются опытом конкретного эксперта. К качественным моделям относятся экспертные оценки, экспертные модели нечеткой логики, консенсус-прогнозы и пр.

Количественные модели – это класс моделей, которые делятся на два больших класса – линейные и нелинейные. Линейные модели – это класс моделей, традиционно применявшийся для прогнозирования и анализа фондового рынка вплоть до 1960-х годов. Один из самых известных методов этого класса – авторегрессионный метод ARIMA. Среди нелинейных моделей большую группу занимают относительно недавно появившиеся модели, называемые моделями Искусственного Интеллекта. К таким типам моделей относят модели нейронных сетей, генетические алгоритмы, модели нечеткой логики, а также различные модели на стыке направлений искусственного интеллекта – это модели объединяющие принципы нечеткой логики с нейронными сетями.

Согласно исследованию finviz.com наибольшая доля внутрисуточных сделок, заключаемых на рынке NYSE Arca, совершается автоматическими роботизированными системами. Такие системы работают в соответствии с

определенными моделями и представляют интерес с исследовательской точки зрения. Прогнозирование американского индекса S&P500 является важной задачей, т.к. от этого индекса происходят множество самых торгуемых финансовых инструментов. В индекс S&P500 входят 500 американских компаний с наибольшей долей свободно обращаемых акций, рассчитанных по методу средневзвешенного арифметического.

В работе исследовалась задача прогнозирования будущих значений индекса S&P500, на основе значений цен акции 500 компаний, входящих в этот индекс. В исследовании применены модели нейронных сетей. Выбор данных моделей обусловлен приемлемы результатами прогнозирования в исследованиях, посвященных данной задаче.

После проведения экспериментов, модель рекуррентных сетей показала более приемлемые результаты. Это связано с особенностью работы модели – проход, осуществляемы в обе стороны. Значения точности для данной модели составляет 0,637.

Модели нейронных сетей можно успешно использовать для прогнозирования цены индекса S&P500, т.к. индекс является агрегированным показателем, зависящим от значений цен акций, входящих в данный индекс. Но колебания значения индекса определяются не только этим. Биржи являются сложными финансовыми структурами, подверженными колебаниями, определяемые множеством факторов – не только экономических, но и политических. Но даже несмотря на эти факторы, методы машинного обучения дают приемлемый результат при их прогнозировании.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Баффет У. Эссе об инвестициях, корпоративных финансах и управлении компаниями / Уоррен Баффет; сост., авт. предисл. Лоренс Каннигем; пер. с англ. — М.: Альпина Бизнес Букс, 2013.
2. Брейли Р., Майерс С. Принципы корпоративных финансов. — М.: Олимп-бизнес, 2014.
3. Елисеева И.И., Юзбашев М.М. Общая теория статистики: Учебник/Под ред. чл.-корр. РАН И.И. Елисеевой. — 4-е изд., перераб. и доп. — М.: Финансы и статистика, 2014.
4. Доусон М. Програмуємо на Python. — СПб.: Питер, 2014. — 416 с.
5. Хайкин С. Нейронные сети. Полный курс. М.: Издательский дом Вильямс, 2006. 1104 с.
6. Каллан Р. Основные концепции нейронных сетей. М.: Издательский дом Вильямс, 2003.- 288 с
7. Нейронные сети. Statistica Neural Networks: Пер. с англ.. М.: Горячая линия - Телеком, 2000. - 182 с.
8. В.А. Козынченко, А.И. Прус, Нейронная сеть в задачах аппроксимации, IEEE Publications , - Санкт-Петербург, 2014.
9. Лутц М. Изучаем Python, 4-е издание. — Пер. с англ. — СПб.: Символ-Плюс, 2011.
10. Лутц М. Программирование на Python, том I, 4-е издание. — Пер. с англ. — СПб.: Символ-Плюс, 2011.
11. Лутц М. Программирование на Python, том II, 4-е издание. — Пер. с англ. — СПб.: Символ-Плюс, 2011.
12. Прохоренок Н.А. Python 3 и PyQt. Разработка приложений. — СПб.: БХВ-Петербург, 2012.
13. Коупленд Т., Колер Т., Мурин Дж. Стоимость компаний оценка и управление: Пер. с англ. — М.: Олимп-Бизнес, 2012.

14. Калан Р. Основные концепции нейронных сетей.: Пер. с англ. – М.: Издательский дом «Вильямс» - 2001. – с.: ил.
15. Магнус Я.Р. Эконометрика. Начальный курс. Москва, М: Дело, 2005
16. Твид Л. Психология финансов. . — М.: "ИК "Аналитика", 2005 – 376 с.
17. Осовский С. Нейронные сети для обработки информации. М.: Финансы и статистика, 2002. 344 с
18. Оболенский А.Ю., Оболенский И.А. Лекции по аналитической геометрии: учебно-методическое пособие. Москва-Ижевск: Институт компьютерных исследований, 2004. - 216 с.
19. Рудаков А.С. Подходы к решению задачи прогнозирования временных рядов с помощью нейронных сетей // Анализ данных и интеллектуальные системы. №.4. 2008.
20. Рутковская Д., Пилиньский, М., Нейронные сети, генетические алгоритмы и нечеткие системы / М., 2004. 452 с.
21. Семейкин В. Д., Скупченко А.В. Моделирование искусственных нейронных сетей в среде MATLAB // Сист. телекоммуникаций. 2008.
22. Соيفер В.А. Методы компьютерной обработки изображений / М.: Физматлит. 2001. 784 с.
23. Солдатова О.П., Семенов В. В. Применение нейронных сетей для решения задач прогнозирования / СГАУ им. ак. С.П. Королева. 2010.
24. Тархов Д. А. Нейронные сети. Модели и алгоритмы / Изд-во: Радиотехника. 2007. 256 с.
25. Татаренков Д.А. Анализ методов обнаружения лиц на изображении//Технические науки. Молодой ученый. 2015. №4(84).
26. Уоссермен Ф. Нейрокомпьютерная техника: теория и практика / пер. с англ. Ю.А.Зуев, В.А. Точенов.1992.
27. Хайкин С. Нейронные сети. Полный курс / М.: Вильямс, 2006.

28. Халафян А.А. Статистический анализ данных. 3-е изд. учеб./ Бином – Пресс. 2007. 512 с.
29. Царегородцев В. Г. Вычислительные технологии// Вестник КазНУ. ч 3. 2008. с. 308-315.
30. Шамис А.Л. Поведение, восприятие, мышление: проблемы создания искусственного интеллекта / Изд-во: Наука. 2005. 224 с.
31. Шахнов В.А., Власов А.И., Кузнецов А.С. Нейрокомпьютеры: архитектура и схемотехника / М.: Изд-во Машиностроение. 2000. 64 с. 43. Шитиков В.К., Розенберг Г.С., Зинченко Т.Д. М
32. Леван Д.Н., Феоктистов Н.А. Особенности использования многослойного персептрона/ Науковедени. вып. 2. 2014. - 249-256 с.
33. X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. Aistats 9, 2010.
34. I. Sutskever et al. On the importance of initialization and momentum in deep learning. ICML'13 Proceedings of the 30th International Conference on International Conference on Machine Learning Volume 28, 2013. - 1139 – 1147 с.
35. R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. ICML '08 Proceedings of the 25th international conference on Machine learning, 2008. - 160-167 с.
36. B. Pang and L. Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval archive, 2008.
37. J. Schmidhuber. Learning complex, extended sequences using the principle of history compression. Neural Computation 4(2), 1992. - 234-242 с.
38. Fodor, I. K. A survey of dimension reduction techniques. Center for Applied Scienti_c, 2002.
39. Friedman, J., Hastie, T., & Tibshirani, R. The elements of statistical learning (Vol. 1). Springer series in statistics Springer, Berlin, 2001.

40. Grudnitski, G., & Osburn, L. Forecasting s&p and gold futures prices: An application of neural networks. *Journal of Futures Markets*, 1993.
41. Hamid, S. A., & Iqbal, Z. Using neural networks for forecasting volatility of s&p 500 index futures prices. *Journal of Business Research*, 2004.
42. IT Galaxy [Электронный ресурс]// URL: <https://ru.intel.com> (дата обращения 15.02.2016).
43. Аналитические технологии для прогнозирования и анализа данных [On-line]: <http://www.neuroproject.ru/neuro.php>
44. База и генератор образовательных ресурсов [On-line]: <http://bigor.bmstu.ru/?cnt/?doc=NN/base.cou>
45. Введение в теорию нейронных сетей [On-line]: <http://www.orc.ru>
46. Области применения искусственных нейронных сетей [On-line]: <http://www.neuropro.ru/>
47. Портал искусственного интеллекта [On-line]: <http://www.aiportal.ru>
48. US Bureau of Economic Analysis (BEA) [On-line]: <https://www.bea.gov/>
49. Federal Reserve Economic Data (FRED) [On-line]: <https://fred.stlouisfed.org/>
50. US Bureau of Labor Statistics (BLS) [On-line]: <https://www.bls.gov/>

ПРИЛОЖЕНИЕ 1.

СРАВНЕНИЕ БИБЛИОТЕК ДЛЯ МАШИННОГО ОБУЧЕНИЯ

| Наименование | Разработчик | Лицензия | Открытое | Платформа | Язык | Интерфейс | Предварительное обучение | RNN | CNN | Машинная Большихана/ Глубокая сеть доверия | Параллельные вычисления |
|-----------------------------|--|----------------|----------|---|-------------|---|-------------------------------|-----|-----|--|-------------------------|
| Apache Singa | Apache Incubator | Apache 2.0 | Да | Linux, Mac OS X, Windows | C++ | Python, C++, Java | Да | Да | Да | Да | Да |
| Caffe | Berkeley Vision and Learning Center | Лицензия BSD | Да | Linux, Mac OS X, Windows | C++ | Python, MATLAB | Да | Да | Да | Нет | Неизвестно |
| Deeplearning4j | Группа Skyrmind, Adam Gibson | Apache 2.0 | Да | Linux, Mac OS X, Windows, Android | Java | Java, Scala, Clojure, Python(Keras) | Да | Да | Да | Да | Да |
| Dlib ^[en] | Дэвис Кинг | Лицензия Boost | Да | Кроссплатформенность | C++ | Python, C++ | Да | Нет | Да | Да | Да |
| Keras | Франсуа Шолле | Лицензия MIT | Да | Linux, Mac OS X, Windows | Python | Python | Да | Да | Да | Да | Да |
| Microsoft Cognitive Toolkit | Microsoft Research | Лицензия MIT | Да | Windows, Linux (OSX в планах через Docker) | C++ | Python, C++, командная строка, BrainScript (NET в планах) | Да | Да | Да | Нет | Да |
| Neural Designer | Aethelics | Проприетарное | Нет | Linux, Mac OS X, Windows | C++ | Графический интерфейс пользователя | Неизвестно | Нет | Нет | Нет | Неизвестно |
| OpenNN | Aethelics | GNU LGPL | Да | Кроссплатформенность | C++ | C++ | Неизвестно | Нет | Нет | Нет | Неизвестно |
| TensorFlow | Команда Google Brain | Apache 2.0 | Да | Linux, Mac OS X, Windows | C++, Python | Python, C/C++, Java, Go | Да | Да | Да | Да | Да |
| Theano | Монреальский университет | Лицензия BSD | Да | Кроссплатформенность | Python | Python | Через зоопарк моделей Lasagne | Да | Да | Да | Да |
| Torch | Ронан Коллобер, Корай Кавучюглу, Клемент Фарабет | Лицензия BSD | Да | Linux, Mac OS X, Windows, Android, iOS | C, Lua | Lua, LuaJIT, C, библиотека утилит для C++/OpenCL | Да | Да | Да | Да | Да |
| Mathematica | Wolfram Research | Проприетарное | Нет | Windows, Mac OS X, Linux, Облачные вычисления | C++ | командная строка, Java, C++ | Да | Да | Да | Да | Да |

ПРИЛОЖЕНИЕ 2.

ФРАГМЕНТЫ КОДА РАБОТЫ С ДАННЫМИ

Загрузка данных:

```
const URL =
'https://www.worldtradingdata.com/api/v1/history_multi_single_day?symbol=AAPL,MSFT&date=2017-05-04-2017-28-08&'+
'api_token=0UfBvUmJZFeJd0qo17DEHOAJQbLbvHIFxw2ZoGIJb3vkQuoQcw7CP0hаВz7L';
getData : function(res, req){
  request(URL, function (err, res, body) {
    if (err) throw err;
    const data_t = [];
    var objData = JSON.parse(data);
    for (key in objData.intraDay){
      data_t.push(objData.intraDay[key].close)}}})
```

Реализация модели базы данных:

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;
const finSchema = new Schema({
  company : {
    name : String,
    price : [Number]})
mongoose.model('fin', finSchema);
```

Сохранение данных в MongoDB Atlas:

```
save_data : function(app){
  app.get('/save_data', function(req, res){
    for (var i = 0; i < 501; i++){
      new Fin({company: {name: company_name[i],
        price:data[i]}}).save(), res.send('saved')}});
```