**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](#_Toc56941402)

[1 Основные понятия и определения 5](#_Toc56941403)

[2 Классификация моделей временных рядов 9](#_Toc56941404)

[2.1 Регрессионные модели 10](#_Toc56941405)

[2.2 Адаптивные методы прогнозирования временных рядов 14](#_Toc56941406)

[2.3 Авторегрессионные модели прогнозирования 19](#_Toc56941407)

[2.4 Нейросетевые модели 24](#_Toc56941408)

[2.5 LSTM 28](#_Toc56941409)

[2.6 Модели на базе цепей Маркова 30](#_Toc56941410)

[2.7 Модели на базе деревьев классификации и регрессии 32](#_Toc56941411)

[3 Практическая часть 34](#_Toc56941412)

[ЗАКЛЮЧЕНИЕ 37](#_Toc56941413)

[СПИСОК ИПСОЛЬЗОВАННЫХ ИСТОЧНИКОВ: 38](#_Toc56941414)

# ВВЕДЕНИЕ

**Актуальность.** Показания приборов говорят нам о таких вещах, как начальная температура, концентрация реагента, процент катализатора и т.д. Некоторые из этих показаний можно отслеживать через равные интервалы времени, например, каждые пять минут, а некоторые можно измерять непрерывно. Полученные данные могут быть зафиксированы в различном виде, чаще всего применяются специальные таблицы, в которых записываются процесс и соответствующее для него значение параметров в конкретный момент времени. Такие данные можно не просто хранить, а можно использовать для анализа и исследования на предмет существования зависимостей.

Для любых задач с изменяющимися количественным переменными особый интерес представляет исследование влияния некоторых переменных на остальные. Даже когда не существует связи между переменными, мы можем стремиться к тому, чтобы отобразить её с помощью математического уравнения. Уравнение может быть физически бессмысленным, но при этом оно может оказаться весьма ценным для предсказания значений ряда переменных при определённых ограничениях.

Искусственные нейронные сети представляют собой современную технологию, которая способна обучаться на основе данных при участии учителя и без его вмешательства. Нейронные сети хорошо применяются не только для задач распознавания зрительных образов, задач классификации, но и для задач прогнозирования. Помимо нейросетевой модели для задачи прогнозирования будут рассмотрены некоторые модели прогнозирования временных рядов, которые наиболее актуальны на сегодняшний день.

**Цели и задачи.** Целью работы является проверка возможности прогнозирования суммарного торгового трафика импорта для некоторых крупных стран нейросетевыми методами. Разработка новой модели и соответствующего ей метода прогнозирования.

Для этого необходимо решить следующие задачи:

1. проанализировать предметную область;
2. осуществить обзор моделей и методов прогнозирования временных рядов, выявить достоинства и недостатки каждого класса моделей;
3. выявить наиболее используемые классы моделей прогнозирования, определить перспективные подходы, которые могут позволить устранить недостатки авторегрессионного класса моделей;
4. разработать собственную модель для прогнозирования на основе нейронных сетей; выполнить программную реализацию алгоритмов;
5. оценить эффективность предложенной модели прогнозирования при решении задачи прогнозирования различных временных рядов.

**Объект и предмет исследования.** Объектом исследования работы являются временные ряды. Предметом исследования является модель прогнозирования временного ряда нейросетевыми методами.

**Метод исследования.** Для решения поставленных задач в работе использованы метода математического моделирования, анализ временных рядов, регрессионный анализ, методы объектно-ориентированного программирования.

**Результаты исследования.** В ходе исследования были получены результаты:

1. выявлены недостатки моделей прогнозирования;
2. разработана модель и метод прогнозирования временных рядов.

**Практическая ценность работы.** Модель, которая была разработана в ходе исследования, может применяться для прогнозирования временных рядов различных предметных областей.

**Структура работы.** Работа включает введение, три главы, заключение, список использованных источников.

В первой главе описаны основные понятия и определения: временные ряды, сезонность, цикличность, тренды.

Во второй главе рассмотрена классификация моделей прогнозирования, описаны статические и структурные модели прогнозирования.

В третьей главе описана программная реализация нейросетевой модели, которая была реализована с применение библиотек tensorflow/keras, языка программирования Python.

# 1 Основные понятия и определения

Пусть дан временной ряд где . – модель временного ряда, где , – горизонт прогнозирования, – вектор параметров модели.

Одним из главных требований к временному ряду является его стационарность, которая состоит в том, что распределение его значений является инвариантным относительно момента времени, для которого оно построено. Для характеристики стационарности используется то, что для двух выборок, которые были построены в разные моменты времени, закон распределения должен оставаться тем же. Если же элемент рассматриваемых временных рядов многомерен, то следует анализировать каждый компонент элемента временного ряда и проверить, что он имеет равномерное распределение на отрезке , где – математическое ожидание, – её дисперсия. [http://www.ysu.am/files/Paper4.pdf]

Основные явления в эконометрических временных рядах:

1. тренды;

2. сезонная компонента;

3. циклическая компонента.

Рассмотрим каждое из явлений более подробно.

Под трендом понимается длительная тенденция изменения показателей временного ряда. Наиболее распространённым способом моделирования тенденций временных рядов является внедрение и построение различных аналитических функций, которые характеризуют зависимость уровней ряда от времени. Используются линейные, степенные или экспоненциальные функции. Например, уравнение прямой линии , парабола 2-го порядка , логарифмическая , степенная , показательная , гиперболическая , логистическая , тригонометрическая . Также возможно использование комбинированных функций.

Тип тренда устанавливают на основе данных временного ряда, путём осреднения показателей динамики ряда, на основе статистической проверки гипотезы о постоянстве параметров графика.

Методы оценки, как правило, разделяют на параметрические и непараметрические. Вторые не применяются для прогнозирования, они необходимы в том случае, когда не удаётся найти подходящую функцию. Параметрические же, в свою очередь, рассматривают временной ряд как гладкую функцию от : , ; после чего различными методами оцениваются параметры функции , например, методом наименьших квадратов. Метод аналитического выравнивания позволяет оптимально подобрать функцию, которая будет наилучшим образом описывать зависимость уровней именно этого временного ряда от времени.Параметры функции определяются на основе метода наименьших квадратов, который можно представить следующей формулой:

Этот метод заключается в минимизации евклидова расстояния между двумя векторами – вектором восстановленных значений зависимой переменной и вектором фактических значений зависимой переменной. Этот метод также называют методом нахождения оптимальных параметров линейной регрессии, таких, что сумма квадратов ошибок (регрессионных остаток) минимальна.

Циклическая компонента отражает циклические изменения уровней временного ряда для периодов свыше 1 года. Также стоит отметить, что циклическая компонента связана с циклами деловой активности, её периодичность составляет от 2 до 10 лет. Эту компоненту сложно идентифицировать, если анализировать данные за непродолжительный, относительно цикла, период времени. В этом случае компоненту невозможно отделить от трендовой.

Под сезонностью понимают периодические колебания, наблюдаемые на временных рядах. В экономике многие явления характеризуются периодически повторяющимися сезонными эффектами. Например, розничные продажи растут с приближением новогодних праздников, а после них спад. Соответственно, временные ряды отражающие эти сезонные эффекты содержат периодические колебания.

Перед выделением сезонных колебаний необходимо вычислить период сезонности. В большинстве случаев период известен из контекста задачи (если рассматривать розничные продажи, то период будет равен году). Однако если период не известен заранее, то его можно найти с помощью автокорреляционной функции. Таким образом, сезонная компонента может отражать квартальные, месячные и недельные циклы.

Функции обнаружения сезонности встроены во многие программы, предназначенные для работы со статистическими данными, такие как Statistica.

Также иногда выделяют нерегулярную компоненту, которая отражает нерегулярные колебания уровней временного ряда, которые невозможно предсказать. Эта компонента является следствием однократных, а не систематических событий, влияющих на уровни ряда.

Существует два способа, с помощью которых компоненты временных рядов могут взаимодействовать:

1. аддитивная модель:

;

1. мультипликативная модель:

,

где – уровень ряда динамики, – трендовая компонента, – циклическая компонента, – сезонная компонента, – нерегулярная компонента. [https://bigenc.ru/economics/text/2087489]

Выбор той или иной модели зависит от характера исходных данных. Например, если каждый год амплитуда циклических и сезонных изменений носит постоянный характер, то чаще всего используют аддитивную модель, при росте амплитуды этих изменений вместе с ростом показателей – мультипликативную. Стоит отметить, что в практике прогнозирования чаще всего используют мультипликативную модель.

Адаптивная модель с мультипликативной сезонностью была предложена П. Р. Уинтерсом. Из-за экспоненциальной схемы модель становится сложнее, зато и точность прогнозов для большинства товаров существенно возрастает.

Говоря о прогнозировании временных рядов, необходимо различить два взаимосвязанных понятия: метод прогнозирования, модель прогнозирования.

Метод прогнозирования – последовательность действий, которые нужно совершить для получения модели прогнозирования временного ряда. Модель прогнозирования – функциональное представление, которое адекватно описывает временной ряд и является основой для получения будущих значений процесса.

Метод прогнозирования содержит последовательность действий, в результате выполнения которой определяется модель прогнозирования конкретного временного ряда. Метод прогнозирования также включает действия по оценке качества прогнозных значений.

Общий итеративный подход к построению модели прогнозирования состоит из следующий шагов.

Шаг 1. На первом шаге на основании предыдущего собственного или стороннего опыта выбирается общий класс моделей для прогнозирования временного ряда на заданный горизонт.

Шаг 2. Определенный общий класс моделей обширен. Для непосредственной подгонки к исходному временному ряду, развиваются грубые методы идентификации подклассов моделей. Такие методы идентификации используют качественные оценки временного ряда.

Шаг 3. После определения подкласса модели, необходимо оценить ее параметры, если модель содержит параметры, или структуру, если модель относится к категории структурных моделей. На данном этапе обычно используется итеративные способы, когда производится оценка участка (или всего) временного ряда при различных значениях изменяемых величин. Как правило, данный шаг является наиболее трудоемким в связи с тем, что часто в расчет принимаются все доступные исторические значения временного ряда.

Шаг 4. Далее производится диагностическая проверка полученной модели прогнозирования. Чаще всего выбирается участок или несколько участков временного ряда, достаточных по длине для проверочного прогнозирования и последующей оценки точности прогноза. Выбранные для диагностики модели прогнозирования участки временного ряда называются контрольными участками (периодами).

Шаг 5. В случае если точность диагностического прогнозирования оказалась приемлемой для задач, в которых используются прогнозные значения, то модель готова к использованию. В случае если точность прогнозирования оказалось недостаточной для последующего использования прогнозных значений, то возможно итеративное повторение всех описанных выше шагов, начиная с первого.

# 2 Классификация моделей временных рядов

Обычно модели временных рядов разделяют на статические и структурные модели, каждый из которых также включает в себя модели.

В статистических моделях зависимость будущего значения от прошлого задается в виде некоторого уравнения. К ним относятся:

1. регрессионные модели (линейная регрессия, нелинейная регрессия);
2. авторегрессионные модели (ARIMAX, GARCH, ARDLM);
3. модель экспоненциального сглаживания;
4. модель по выборке максимального подобия;

В структурных моделях зависимость будущего значения от прошлого задается в виде некоторой структуры и правил перехода по ней. К ним относятся:

1. нейросетевые модели;
2. модели на базе цепей Маркова;
3. модели на базе классификационно-регрессионных деревьев.

Классификация моделей временных рядов представлена на рисунке 1.

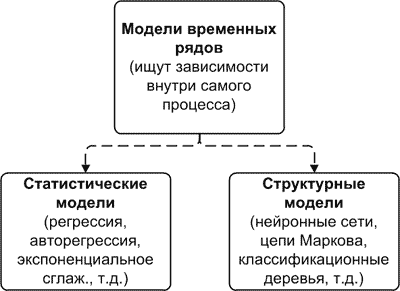


Рисунок 1 – Классификация моделей временных рядов.

Далее рассмотрим более подробно каждый из разделов моделей.

# 2.1 Регрессионные модели

Регрессионный анализ – метод моделирования измеряемых данных и исследования их свойств. Данные состоят из пар значений зависимой переменной (переменной отклика) и независимой переменной (объясняющей переменной). Регрессионная модель есть функция независимой переменной и параметров с добавленной случайной переменной. Параметры модели настраиваются таким образом, что модель наилучшим образом приближает данные. Критерием качества приближения (целевой функцией) обычно является среднеквадратичная ошибка: сумма квадратов разности значений модели и зависимой переменной для всех значений независимой переменной в качестве аргумента. Относительно характера распределения этой величины делаются предположения, называемые гипотезой порождения данных. Для подтверждения или опровержения этой гипотезы выполняются статистические тесты, называемые анализом остатков. При этом предполагается, что независимая переменная не содержит ошибок. Регрессионный анализ используется для прогноза, анализа временных рядов, тестирования гипотез и выявления скрытых взаимосвязей в данных.

Регрессия – зависимость математического ожидания (например, среднего значения) случайной величины от одной или нескольких других случайных величин (свободных переменных), то есть . Регрессионным анализом называется поиск такой функции , которая описывает эту зависимость.

Линейная регрессия – одна из важнейших и широко используемых техник регрессии. Эта самый простой метод регрессии. Одним из его достоинств является лёгкость интерпретации результатов. Как было замечено ранее, в основе модели регрессии лежат исследования зависимостей одной переменной от другой. Иногда две переменные связаны точным уравнением прямой линии. Уравнение прямой может быть полезно во многих ситуациях для обобщения наблюдаемой зависимости одной переменной от другой. Такое уравнение можно получить методом наименьших квадратов, когда существуют экспериментальные данные.

Пусть линия регрессии переменной . Тогда можно записать линейную модель

,

так что для данного соответствующее значение состоит из величины плюс добавка в виде , при учёте которой любой индивидуальный получает возможность не попасть на линию регрессии.

Уравнение линейной модели, которое было записано выше, представляет из себя модель, в которую «мы верим». Начинать необходимо с предположения, что эта модель установлена, но она требует проверки на последующих стадиях. Первоначальное допущение верности установленной модели называют постулированием модели или допущение о её правильности. Величины , – параметры модели.

Когда говорят о том, что модель линейна или нелинейна, то имеют в виду линейность и или нелинейность по параметрам. Величина наивысшей степени предикаторав модели называется порядком модели. Например, модель, которая имеет вид:

представляет из себя регрессионную модель второго порядка (по ) и линейная (по ). Обозначение вида чаще всего используется в полиномиальных моделях. [Draper N., Smith H. Applied regression analysis]

Под многомерной регрессией понимают линейную регрессию в -мерном пространстве, где объекты и признаки являются -мерными векторами. Пусть существует множество объектов , множество ответов , набор вещественнозначных признаков , . Введём матричные обозначения: матрица информации , целевой вектор , вектор параметров и диагональная матрица весов:

*, , ,, .*

Алгоритм можно записать формулой:

.

Функционал квадрата ошибки в матричном виде:

Для точности оценки регрессии как линейной, так и многомерной модели можно воспользоваться методом наименьших квадратов. Основной проблемой многомерной линейной регрессии является вырожденность, или, в более общем случае, мультиколлинеарность матрицы , которую приходится обращать. Подобные проблемы возникают, когда среди признаков есть почти линейно зависимые.  
Мультиколлинеарность матрицы определяется её числом обусловленности:

,

где — собственные значения матрицы . Чем больше число обусловленности, тем ближе матрица к вырожденной и тем неустойчивее обратная к ней матрица. Плохая обусловленность матрицы: . Матрицу принято считать плохо обусловленной, если её число обусловленности превышает . [http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D0%BC%D0%B5%D1%80%D0%BD%D0%B0%D1%8F\_%D0%BB%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F\_%D1%80%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D1%8F]

Нелинейная регрессия – это способ нахождения нелинейной модели взаимосвязи между зависимой переменной и набором независимых переменных. В отличие от традиционной линейной регрессии, которая ограничена оценкой линейных моделей, нелинейная регрессия может оценивать модели с произвольными взаимосвязями между независимыми и зависимыми переменными. Это достигается при помощи итерационных алгоритмов оценки.

Нелинейная модель , .

Функционал среднеквадратичного отклонения:

Если функция – непрерывно дифференцируема, то можно применить метод стохастического градиента. Для получения наилучших результатов с меньшими затратами на время применяют более трудоёмкий с точки зрения вычислений метод. Этот метод Ньютона-Рафсона можно описать следующими этапами:

1. Начальное приближение

.

2. Итерационный процесс

*,*

где – градиент функционала в точке , матрица вторых производных называется гессианом функционала в точке , – величина шага (чаще полагают, что ).

Линеаризация в окрестности текущего :

Метод Ньютона-Гаусса также хорошо помогает в сведении к линейной регрессии, как и метод Ньютона-Рафсона. Однако стоит отметить, что скорость сходимости зависит от выбора , от выбора самой модели, в частности её характеристика по приближению выборки.

Матричные обозначения для метода Ньютона-Гаусса:

– -матрица первых производных;

– вектор значений .

Формула t-й итерации метода Ньютона-Гаусса:

*.*

*–* это решение задачи многомерной линейной регрессии:

*.*

# 2.2 Адаптивные методы прогнозирования временных рядов

Адаптивные методы прогнозирования временных рядов представляют из себя методы, цель которых заключается в построении самокорректирующихся (самонастраивающихся) экономико-математических моделей, которые способны отражать изменяющиеся во времени условия, учитывать информационную ценность различных членов временной последовательности и давать достаточно точные оценки будущих членов данного ряда. Такие модели предназначаются прежде всего для краткосрочного прогнозирования.

Последовательность процесса адаптации в основном выглядит следующим образом. Пусть модель находится в некотором исходном состоянии (т.е. определены текущие значения ее параметров) и по ней делается прогноз. Выжидаем, пока истечет одна единица времени (шаг моделирования), и анализируем, насколько далек результат, полученный по модели, от фактического значения ряда. Ошибка прогнозирования через обратную связь поступает на вход системы и используется моделью в соответствии с ее логикой для перехода из одного состояния в другое с целью большего согласования своего поведения с динамикой ряда. На изменения ряда модель должна отвечать "компенсирующими" изменениями. Затем делается прогноз на следующий момент времени, и весь процесс повторяется.

Предполагаем, что задан временной ряд:  , где  - значение временного ряда в момент времени .  - прогноз значения временного ряда в момент времени , сделанное в момент времени .

Простейшие адаптивные модели:

1. экспоненциальное сглаживание, Модель Брауна;

2. модели линейного роста, которые включают в себя Модель Хольта, модель линейного роста Брауна, модель прогнозирования Дж. Бокса и Г. Дженкинса;

3. сезонные модели: модель Хольта-Уинтерса, модель Тейла-Вейджа.

При экспоненциальном сглаживании, в частности при Модели Брауна, предполагается, что ряд генерируется моделью, которая записана следующей формулой:

*.*

В этой формуле – есть варьирующий по времени средний уровень ряда, выступает в качестве белого шума. Прогноз временного ряда получается по формуле:

,

#### где значение экспоненциальной средней в момент времени . можно найти, используя:

#### где представляет из себя параметр сглаживания.

Главное достоинство такой прогнозной модели состоит в том, что она способна последовательно адаптироваться к новому уровню процесса без значительного реагирования на случайные отклонения.

Недостаток: экспоненциальная средняя дает систематическую ошибку, когда временной ряд имеет тенденцию линейного роста

Прогноз для моделей линейного роста может быть получен по уравнению:

,

где ,  – текущие оценки коэффициентов адаптивного полинома первого порядка.

В разных моделях эти коэффициенты вычисляются различными способами. Рассмотрим три модели и методы вычисления оценок коэффициентов адаптивного полинома первого порядка.

В [модели Хольта](http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%A5%D0%BE%D0%BB%D1%8C%D1%82%D0%B0) используются формулы:

,

,

где  выступают в качестве параметров адаптации, для которых должно выполняться: .

Если рассматривать модель линейного роста Брауна, стоит отметить, что эта модель является частным случаем модели Хольта и для вычисления оценок применяют:

;

,

где  – ошибка прогноза,  - коэффициент дисконтирования, характеризующий обесценивание данных наблюдения за единицу времени.

Для модели прогнозирования Дж.Бокса и Г.Дженкинса справедливы:

.

.

Данная модель не дает преимуществ перед моделью Хольта, так как коэффициент  часто оказывается близким к нулю.

Рассмотрим сезонные модели. А именно Модель Хольта-Уинтерса, которая учитывает мультипликативный тренд и сезонность, и Модель Тейла-Вейджа, которая, в свою очередь, будет брать во внимание аддитивный тренд и сезонность.

Рассматривая Модель Хольта-Уинтерса, нельзя не упоминуть о модель Хольта, ведь именно на основе этой модели Уинтерс создал свою прогностическую модель, которая способна учитывать экспоненциальный тренд и аддитивную сезонность.

В модели Хольта также необходимо решить задачу прогнозирования временного ряда. При существовании линейного тренда на данных временного ряда использование модели Брауна не имеет смысла. Для учёта влияния линейного тренда используют модель Хольта. Однако эта модель не учитывает сезонность.

Для описания математических формул для Модели Хольта-Уинтерса зададим временной ряд где . Для решения задачи прогнозирования временного ряда будет верно:

,

,

,

,

где – период сезонности, , – сезонный профиль, – параметр тренда, – параметр прогноза, который «очищен» от влияния тренда и сезонности.

Параметры , , должны находиться в диапазоне от до . Чаще всего их предлагается находить экспериментальным путём. Уинтерс также отмечал, что один такой набор весов можно использовать для широкого класса продуктов. [*Winters P.R.* Forecasting sales by exponentially weighted moving averages //Management Science.]

Теперь рассмотрим усложнённую модель Хольта, которая учитывает сезонность и аддитивный тренд. Для неё справедливы формулы:

,

,

,

,

где – период сезонности, , – сезонный профиль, – параметр тренда, – параметр прогноза, который «очищен» от влияния тренда и сезонности. Используя метод минимизации квадратичной ошибки, выбор параметров , , происходит экспериментально, диапазон сохраняется, как и для модель Хольта-Уинтерса.

Рассмотрим стохастический процесс Тейла и Вейджа. При изучении экспоненциальной средней они предложили применить двухпараметрический предикатор Хольта для прогнозирования некоторого вероятностного процесса, который ярко характеризуется трендом. Процесс Тейла-Вейджа аналитически записывается так:

*;*

где – значение уровня исследуемого временного ряда в момент времени , – прирост уровня от момента к моменту , , – временный последовательности с нулевым математическим ожиданием, постоянными дисперсиями и отсутствуем ковариации. Временной ряд не является стационарным и не имеет строго определённой автоковариационной функции. Но Нерлов и Вейдж показали, что из вышеупомянутых уравнений следует стационарность вторых разностей процесса .

Схема составления прогноза выглядит следующим образом:

;

*,*

где . Если ошибку прогноза, сделанного в момент времени на шаг вперёд, обозначить через , то уравнения адаптации примут вид:

;

.

Ошибка прогноза является суммой трёх компонент: ошибки оценки уровня процесса в момент , ошибки оценки прироста уровня в момент и комбинации случайных компонент и в момент времени . Очевидно также, что определение оптимальных и определению оптимальных и .

Используя соотношение:

,

Нерлов и Вейдж показали, что проблема прогнозирования эквивалента задаче прогнозирования второй разности и что при ограничениях, наложенных на параметры адаптации, ошибка прогноза является линейной комбинацией текущего и прошлых значений стационарного ряда :

*,*

где – сходящий ряд весов.

В результате минимизации дисперсии ошибки прогноза на один шаг вперёд были получены следующие результаты:

;;; ; ; *.*

[*Лукашин Ю. П.* Адаптивные методы краткосрочного прогнозирования временных рядов]

#### **2.3 Авторегрессионные модели прогнозирования**

#### Это широчайший и один из двух наиболее широко применимых классов моделей:

* ARIMAX (autoregression integrated moving average extended
* GARCH (generalized autoregressive conditional heteroskedasticity)
* ARDLM (autoregression distributed lag model)

Авторегрессионная (AR) модель — модель временных рядов, в которой значения временного ряда в данный момент линейно зависят от предыдущих значений этого же ряда. Авторегрессионный процесс порядка *p* (AR(*p*)-процесс) определяется следующим образом где  {\displaystyle a\_{1},\ldots ,a\_{p}} — параметры модели (коэффициенты авторегрессии), {\displaystyle c} — постоянная (часто для упрощения предполагается равной нулю), а {\displaystyle \varepsilon \_{t}} — белый шум.

Простейшим примером является авторегрессионный процесс первого порядка AR(1)-процесс:

.

Для данного процесса коэффициент авторегрессии совпадает с коэффициентом автокорреляции первого порядка.

Другой простой процесс — процесс Юла — AR(2)-процесс:

.

Под ARIMAX понимают некую математическую модель для анализа временных рядов, которая объединяет в себе интегрированную авторегрессию, скользящее среднее и возможность учёта дополнительных внешних факторов. Также стоит отметить, что модели ARIMAX часто используют при решении задач, в которых требуется построить прогноз на основе имеющихся данных и вычислить соответствующие последующие значения ряда на основе предыдущих. Важным примечанием является тот факт, что для того чтобы получить прогнозируемые данные необходимо предварительное обучение узла.

Обучение узла может производиться вручную или автоматически. В некоторых компонентах используются самообучающиеся алгоритмы. Например, кластеризация, самоорганизующиеся карты, квантование. Такие узлы требуют первоначального обучения. Ручное обучение или переобучение позволяет контролировать параметры переобучения Модели с возможностью просмотра полученных результатов. Автоматический способ, в свою очередь, гораздо быстрее и хорошо подходит при незначительных изменениях в исходных данных.

Особенности настройки с помощью ARIMAX:

1. Настройка входных столбцов – заключается в правильном задании назначений столбцов входного набора данных. Необходимо определить, какой из столбцов является прогнозируемым, какой выступает в качестве входного, какой отвечает за те данные, которые не участвуют в построении модели.

2. Нормализация данных – этап, который обычно пропускают для прогнозируемых данных.

3. При настройке ARIMAX структура определение структуры модели происходит автоматически. Параметры в процессе вычисления подбираются таким образом, чтобы минимизировать значение AIC.

Порядок p авторегрессионной части AR определяет число предыдущих значений ряда, учитываемых при построении модели.

Порядок d разностей ряда, порядок интегрирования, задает порядок при необходимости привести исходный ряд к [стационарному](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D1%80%D0%BD%D0%BE%D1%81%D1%82%D1%8C).

Порядок q части скользящего среднего, части MA, определяет размер скользящего окна для сглаживания исходного ряда. Для все трёх порядков верно, что устанавливается значение целого типа, при чём это значение больше 0.

4. Есть возможности включения расчёта сезонности. Это позволит задать параметры для сезонной составляющей.

Горизонтом прогнозирования можно назвать временной интервал, в пределах которого прогноз выполняется с заданной точностью. Кроме этого, под горизонтом прогнозирования понимают параметр модели прогнозирования на основе метода скользящего окна. Выделяют три уровня горизонта прогнозирования:

* долгосрочный;
* среднесрочный;
* краткосрочный.

Особенности прогнозирования временных рядов по средствам ARIMAX:

1. Горизонт прогноза задает количество значений, которые будут спрогнозированы и добавлены в выходной набор в конце исходного временного ряда.

2. Важно рассчитать ошибку аппроксимации. Будет добавлен в выходной набор столбец со средними отклонениями прогнозируемых значений от фактических.

3. Доверительный интервал прогноза указывается в процентах от 0 до 100, имеет вещественный тип, обычно равен 95.

Процесс авторегрессии порядка , который обозначается AR(p), описывается формулой:

*,*

где – вещественная константа, – коэффициенты, – ошибка прогнозирования. Для определения , используют метод наименьших квадратов или метод максимального правдоподобия.

Также используется другой тип модели и часто применяется совместно с авторегрессией. Эта модель скользящего среднего порядка и описывается формулой:

*,*

где – порядок скользящего среднего, – ошибка прогнозирования.

Целесообразно объединить в одной модели авторегрессию и скользящее среднее. Общая модель обозначается ARMA(p,q). Если в качестве входных данных используется не сами значения временного ряда, а их разность -того порядка, то модель носит название авторегрессии проинтегрированного скользящего среднего. Такую модель называют ARIMA(p,d,q). Развитием модели ARIMA(p,d,q) является модель ARIMAX(p,d,q), которая записывается уравнением:

*,*

где – коэффициенты внешних факторов .

[ <https://help.loginom.ru/userguide/processors/datamining/arimax.html>]

GARCH-модель – модель, которая чаще всего используется для прогнозирования ситуации на финансовых рынках в условиях нестабильности. Для таких ситуаций характерно такое понятие, как дисперсия на различных интервалах наблюдения, которую также называют гетероскедастичностью. Линейную регрессионную модель в таких случаях не применяют.

Говоря о модели GARCH, нельзя не вспомнить о ARCH-модели, в которой используется условная, зависимая от времени дисперсия, выражаемая через квадрат значений показателей прошлых периодов. Для ARCH-модели верно:

*,*

где – коэффициент задержки.

Позднее была предложена GARCH-модель. Она является обобщенной авторегрессионной моделью гетероскедастичности. GARCH-модель предполагает, что на текущую изменчивость дисперсии влияют предыдущие изменения показателей и предыдущие оценки дисперсии. Расчёт дисперсии для GARCH-модели:

,

где – количество предшествующих оценок, которые влияют на текущее значение, – весовые коэффициенты,  отражающие степень влияния предыдущих оценок на текущее значения. Модель обозначают GARCH. [https://wiki.loginom.ru/articles/garch-model.html]

ALRDM-модель – авторегрессионная модель с распределённым лагом. Часто при моделировании процессов на изучаемую переменную влияют и текущие значения процесса, и его лаги, то есть значения временного ряда, предшествующие изучаемому моменту времени. ALRDM-модель описывается уравнением:

*,*

где – коэффициенты, – величина лага. Модель обозначают ALRDM, чаще всего применяют при экономических процессах. [https://www.mbureau.ru/articles/dissertaciya-model-prognozirovaniya-vremennyh-ryadov-glava-1]

# 2.4 Нейросетевые модели

Нейронная сеть – это математическая модель, а также её программное или аппаратное воплощение, которая построена по принципу организации и функционирования биологических нейронных сете, которые, в свою очередь определяются как сети нервных клеток живого организма. [https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F\_%D1%81%D0%B5%D1%82%D1%8C]

Другими словами, под понятием нейронных сетей подразумевают компьютерные системы, которые собраны из сотен, тысяч или миллионов клеток мозга, способные обучаться и действовать по принципу чрезвычайно похожему на то, как работает мозг человека. Обычная искусственная нейронная сеть состоит из сотен, тысяч или миллионов нейронов, названных блоками, которые выстроены в слои, где каждый блок соединён с соседним как в собственном слое, так и в ближайшем. Некоторые из блоков выступают в качестве блоков ввода и призваны получать из внешнего мира информацию. Эти блоки соединяются со скрытыми блоками, которые обрабатывают полученные данные и занимают большую часть искусственного мозга. Также существует блок вывода, который, в свою очередь, занимается извлечением полученной и обработанной информации. Соединение между блоками характеризуется числом, которое называется весом, которое может быть как положительным, так и отрицательным. Чем больше вес связи, тем сильнее один блок влияет на другой. [https://www.youtube.com/watch?v=L1tsgJ9m1zM]

Нейросети часто стали применяться в практических целях: задачи прогнозирования, задачи распознавания образов, задачи управления и другие. Рассмотрим первую задачу и определим основные понятия в задаче прогнозирования. Стоит также отметить, что нейросети не программируются в привычном смысле этого слова, они обучаются.

В общем случае выделяют три фундаментальных класса нейросетевых архитектур:

1) однослойные сети прямого распространения;

2) многослойные сети прямого распространения;

3) рекуррентные сети.

Дадим общее описание для каждой архитектуры.

В многослойной нейронной сети нейроны располагаются по слоям. В самом простом случае в такой сети существует входной слой узлов источника, информация от которого передаётся на выходной слой нейронов. Такая сеть носит название сеть прямого распространения (ацикличная сеть). Так на рисунке 2 показана структура ацикличной сети для случая четырёх узлов в каждом из слоёв. Эта нейронная сеть называется однослойной, при этом слой вычислительных элементов есть единственный слой, существующий в этой сети.

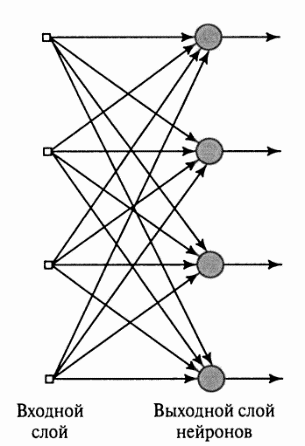


Рисунок 2 – Ацикличная сеть с 1 слоем нейронов.

Следующий класс нейронных сетей прямого распространения характеризуется существованием одного или нескольких скрытых слоёв, узлы которого называют скрытыми нейронами. Добавляя один или несколько скрытых слоёв, можно выделить статистики высокого порядка. Такая сеть позволяет выделять глобальные свойства данных с помощью локальных соединений за счёт наличия дополнительных синаптических связей и повышения уровня взаимодействия нейронов.

Нейронная сеть, в которой все узлы каждого конкретного слоя соединены со всеми узлами смежных слоёв, считается полносвязной. Так на рисунке 3 показана полносвязная нейронная сеть. Если некоторые из синаптических связей отсутствуют, такая связь называется неполносвязной.

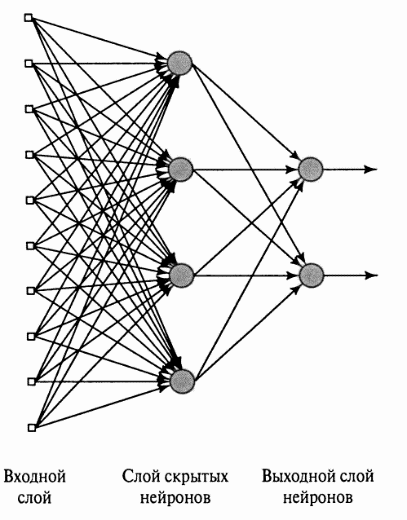


Рисунок 3 – Полносвязная сеть прямого распространения с 1 скрытым и 1 выходным слоем.

Рекуррентная нейронная сеть (RNN) отличается от сети прямого распространения наличием, по крайней мере, одной обратной связи. Так рекуррентная сеть может состоять из единственного слоя нейронов, каждый из которых направляет свой выходной сигнал на входы остальных нейронов слоя.

Такой тип нейронной сети хорошо подходит для временных рядов. RNN обрабатывают временной ряд шаг за шагом, поддерживая внутреннее состояние от временного шага к временному шагу.

Архитектура RNN представлена на рисунке 4. Как можно заметить в приведённой структуре отсутствуют обратные связи нейронов с самими собой. RNN на рисунке 4 также не имеет скрытых нейронов. На рисунке 5 показан другой класс RNN, который включает скрытые нейроны. Обратные связи для данного классе исходят из скрытых и из выходных нейронов.

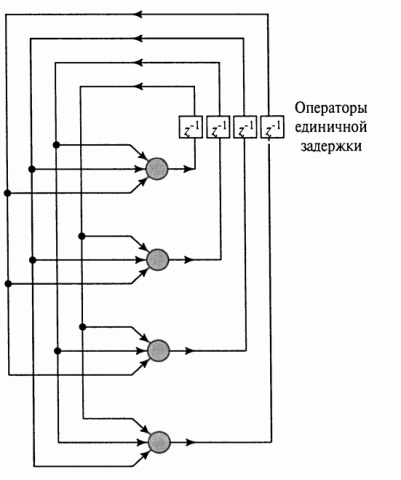


Рисунок 4 – RNN без скрытых нейронов и обратных свзяей нейронов с самими собой.

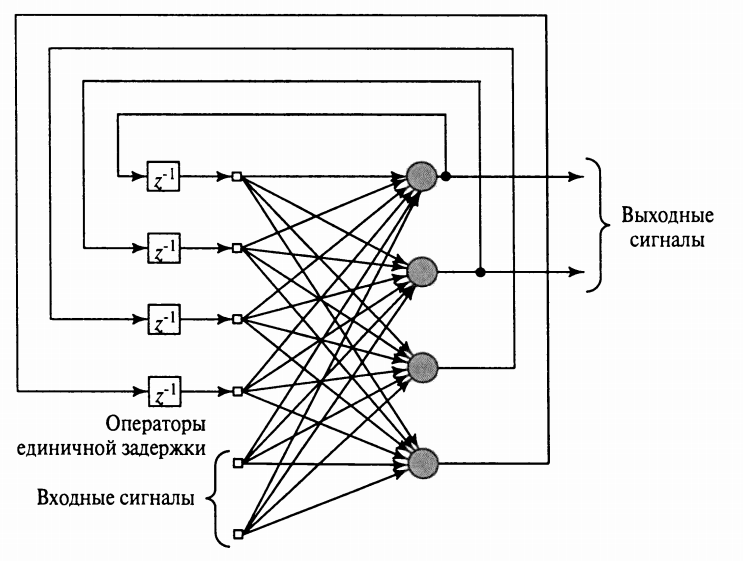


Рисунок 5 – RNN со скрытыми нейронами.

Присутсвие обратных связей в сетях, которые представлены на рисунке 2 и 3, показывает на способность таких сетей к обучению и на их высокую производительность. Обычно обратная связь подразумевает использование элементов еденичной задержки, это приводит к нелинейному динамеческому поведению при существовании в сети нелиненых нейронов. [Нейронные сети. Полный курс Автор: Саймон Хайкин]

Можно сделать вывод о том, что рекуррентные сети отлично помогают в тех случаях, когда необходим эффект накопления. То есть, например, читая какую-то статью, есть возможность понимать смысл каждого слова на основе значений предыдущих слов. С этим не могут справиться простые нейронные сети. А принцип того, что мысли имеют свойство накапливться и влиять на друг друга используется в сетях LSTM.

RNN использует информацию, которую получила ранее, для решения последующих задач. Иногда для решения задач есть необходимость посмотреть только последнюю информацию. В тех случаях, когда разрыв между предыдущей информацией и местом, в котором она будет нужна, невелик. RNN справится с этой задачей. Однако, по мере увеличения этого разрыва RNN теряют связь между информацией. LSTM способен решить проблему «долгосрочных связей» не только в теории, но и на практике.

# 2.5 LSTM

LSTM RNN (Long Short-Term Memory Recurrent Neural Network) – рекуррентная нейронная сеть с долгой краткосрочной памятью. LSTM была изобретена в 1977 году З. Хохрайтером и Ю. Шмидхубером. Как и все другие модификации нейронных сетей, LSTM является универсальной моделью, которая способна решать большой круг различных проблем.

В отличие от традиционных RNN, LSTM специально разработаны для устранения проблемы долгосрочной зависимости. Основная особенность LSTM – запоминание информации в течение длительных периодов времени, отсюда следует, что они практически не требуют обучения. Это достигается за счет того, что повторяющийся модуль модели имеет комбинацию четырех слоев, взаимодействующих друг с другом.

Другими словами, LSTM отлично походит для задач, когда необходимо с течением времени накапливать знания об успешности прошлых предсказаний. LSTM применяется, когда существует сильная зависимость текущих значений временного ряда от предыдущих, но величина лага, имеющий наибольшее влияние на показатель, динамична, то есть коэффициенты, стоящие перед лаговыми переменными постоянного изменяются.

Далее опишем принцип работы сети LSTM.

Важным и ключевым понятием является состояние ячейки, которая напоминает конвейерную ленту. Оно проходит сквозь всю цепочку, подвергается при этом незначительным линейным преобразованиям.

Гейты, выступая в качестве структуры, предназначены для изменения количества информации в состояниях этих ячеек. Гейты состоят из сигмовидного слоя нейронной сети, на выходе которого выдаются «0» и «1», и операции поточечного умножения.

LSTM содержит три таких гейта для контроля состояния ячеек.

Во-первых, слой утраты. Здесь решается вопрос о том, какую информацию необходимо удалить из состояния ячейки, сигмовидный слой принимает решение. Слой получает на вход , и выдает «0» или «1» для каждого номера состоянии ячейки. Формула для «слоя гейта утраты»:

.

Во-вторых, слой сохранения. Здесь решается вопрос о том, какую информацию необходимо сохранить в состоянии ячейки. Процесс разбивается на две части. Сначала сигмовидный слой решает, какие значения требуется обновить. Затем слой tanh создаёт вектор новых значений-кандидатов, которые добавляются в состояние. На следующем шаге происходит объединение двух значений для обновления состояния.

.

*,*

где – вектор новых значений-кандидатов.

В-третьих, формирование нового состояния. Теперь необходимо обновить предыдущее состояние ячейки для получения нового состояния. Для этого:

,

где – старое состояние, второе слагаемое выступает в качестве новых значений кандидатов, которые масштабируются в зависимости от того, как мы решили обновить каждое значение состояния.

Необходимо решить, какой результат ожидаем нами на входе. Результат будет являться отфильтрованным состоянием ячейки:

,

,

где – выходной сигнал сигмовидного гейта, необходим для того, чтобы разместить все значения в интервале . [https://neurohive.io/ru/osnovy-data-science/lstm-nejronnaja-set/] [https://web.archive.org/web/20161123045043/http://deeplearning.cs.cmu.edu:80/pdfs/Hochreiter97\_lstm.pdf]

На рисунке 6 изображены четыре слоя нейронной сети в желтых прямоугольниках, точечные операторы в зеленых кружках, ввод в желтых кружках и состояние ячейки в голубых кружках.

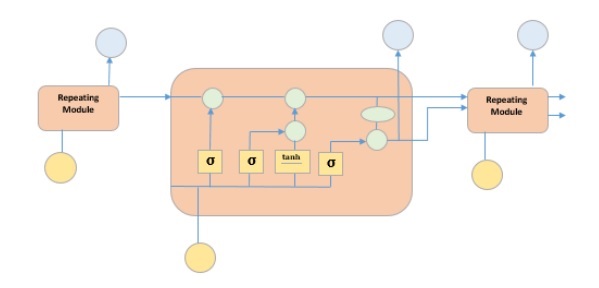


Рисунок 6 – Схема LSTM

Модуль LSTM имеет состояние ячейки и три шлюза, что дает им возможность выборочно изучать, отключать или сохранять информацию от каждого из модулей.

# 2.6 Модели на базе цепей Маркова

Рассмотрим систему, которая состоит из семейства случайных переменных и определяется следующей формулой:

.

Значение , которое характеризует случайную переменную в дискретный момент времени , называет состояние системы в этот момент. Пространство состояний – пространство всех возможных значений, которые могут принимать эти случайные переменные. Если структура стохастического процесса такова, что условное распределение вероятности зависит исключительно от предыдущего значения и не зависит от всех более ранних значений, то такой процесс называется цепью Маркова, или Марковской цепью.

Формально можно записать так:

*.*

Это соотношение называется свойством Маркова. Формально это свойство описывают следующим образом.

Последовательность случайных переменных принимает форму цепи Маркова, если вероятность нахождения системы в состоянии в некоторый момент времени зависит исключительно от вероятности нахождения системы в момент времени в состоянии .

Можно сделать вывод о том, что цепи Маркова имеет смысл рассматривать как порождающие модели, которые состоят из множества попарно связанных друг с другом состояний. Каждый раз, когда система переходит в конкретное состояние, именно с ним ассоциируется выход системы.

Другими словами, модели прогнозирования на основе цепей Маркова предполагают, что будущее состояние процесса зависит только от его текущего состояния. Процессы, которые моделируются цепями Маркова, относятся к процессам с короткой памятью.

Также затронем понятие вероятности перехода, так как в цепи Маркова переход из одного состояния в другое является вероятностным. Однако при этом выход системы является детерминированным. Пусть вероятность перехода:

*,*

где являются условными вероятностями, подчиняющиеся:

*,* для всех . [1]

Предполагается, что вероятности перехода фиксированы и не изменяются во времени. Из этого следует, что последнее условие выполняется при в любой момент времени. Такую цепь называют гомогенной по времени.

Если система имеет конечное множество возможных состояний , вероятности перехода формируют матрицу размерности :

.

Элементы матрицы удовлетворяют условиям [1], последнее из которых гарантирует равенство суммы всех элементов каждой из строк единице. Такая матрица называется стохастической. Любая стохастическая матрица может служить матрицей вероятностей перехода.

# 2.7 Модели на базе деревьев классификации и регрессии

Рассмотрим деревья классификации и регрессии – CART. CART представляет из себя алгоритм бинарного дерева решений, который предназначен для решения задач классификации и регрессии.

Стоит также отметить, что деревья CART всё чаще применяют для прогнозирования временных рядов. Модели CART используют при моделировании процессов, на которые влияют внешние и категориальные факторы. При влиянии внешних непрерывных факторов используют деревья регрессии, при категориальных – деревья классификации. Также имеет место быть смешение этих деревьев при влиянии обоих факторов.

По модели CART прогнозируемое значение временного ряда зависит от предыдущих значений и некоторых независимых переменных. Рассмотрим бинарное классификационно-регрессионное дерево, которое описано в источнике [1].

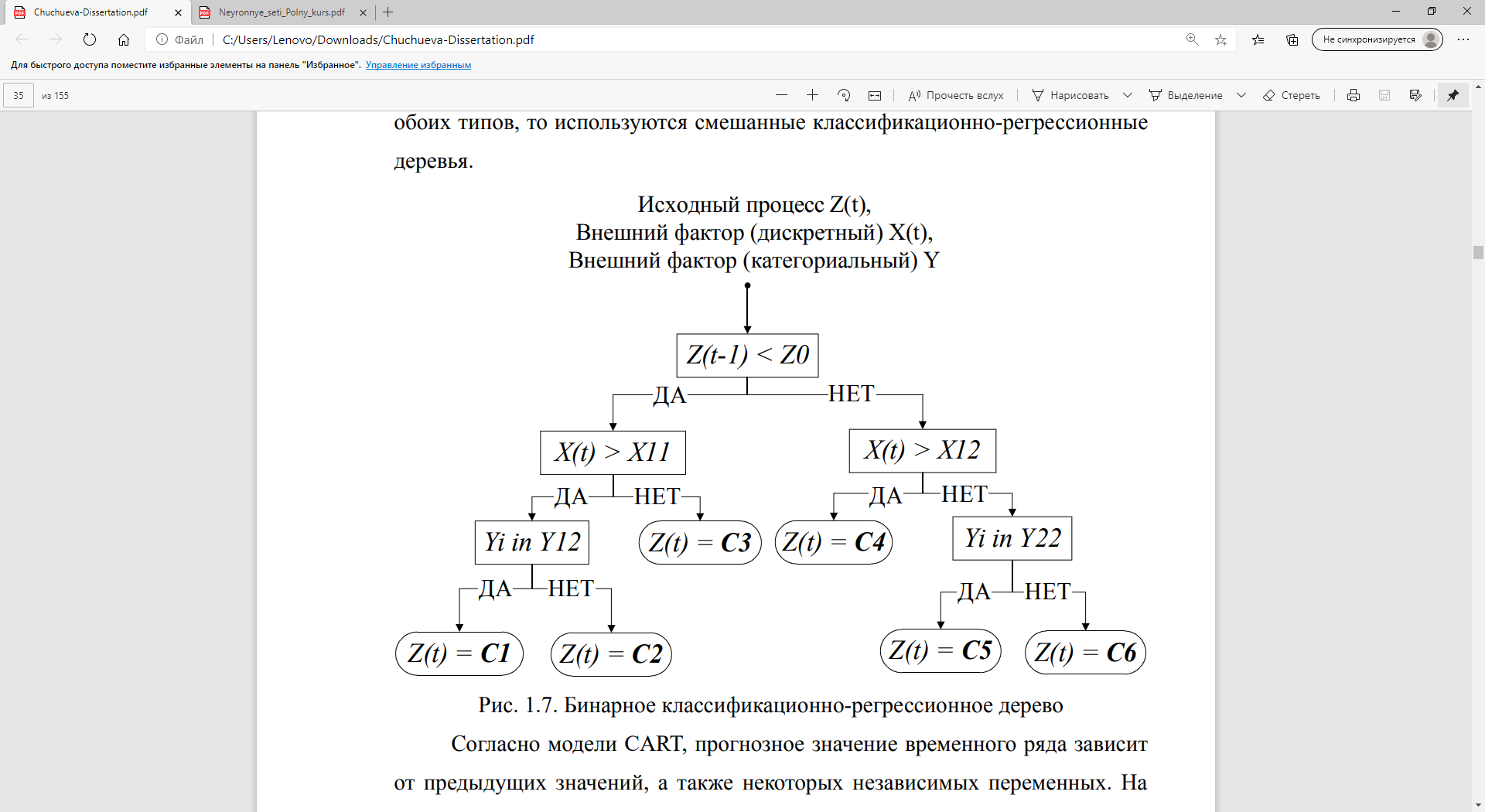


Рисунок 7 – CART

Сначала предыдущее значение процесса сравнивается с константой . При выполнении неравенства проверяется неравенство , при невыполнении проверяется неравенство . Проверки продолжаются до того момента, пока не будет обнаружен лист дерева, в котором происходит определение будущего значения процесса . При поиске этого листа используются непрерывные и категориальные переменные, для которых выполняется проверка присутствия значения в одном из заранее определенных подмножеств. Значения пороговых констант и подмножеств выполняется на этапе обучения дерева.

Так можно сделать вывод о том, что CART – модель, в основе которой лежит зависимость будущей величины процесса посредствам построения структуры дерева.

[Hannes Y.Y., Webb P. Classification and regression trees: A User Manual for IdentifyingIndicators of Vulnerability to Famine and Chronic Food Insecurity]

# 3 Практическая часть

**3.1 Описание программных и математических инструментов**

Реализация алгоритма была осуществлена с помощью использования существующего программного пакета TensorFlow и библиотеки NumPy. Программа была написана на высокоуровневом языке программирования Python.

Стандартная библиотека Python включает набор полезных функций.

Pandas, который был использован для парсинга файла, представляет из себя программную библиотеку на языке Python для обработки и анализа данных.  Работа Pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами.

NumPy является фундаментальным пакетом для научных вычислений на языке программирования Python. Эта библиотека также добавляет поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами.

TensorFlow представляет из себя мощную библиотеку для создания нейронных сетей. TensorFlow предназначен для проектирования, создания, изучение моделей глубокого обучения, и численных вычислений.

Важную роль метод Бокса-Кокса играет в приведении исходной «ненормальной статистики» к «нормальному» виду. Среди множества таких методов преобразований наиболее популярным считается метод Бокса-Кокса.

Для исходной последовательности , , , однопараметрическое преобразование Бокса-Кокса с параметром определяется следующим образом:

*.*

Параметр можно выбирать, максимизирую логарифм правдоподобия. Еще один способ поиска оптимального значения параметра основан на поиске максимальной величины коэффициента корреляции между квантилями функции нормального распределения и отсортированной преобразованной последовательностью. [http://www.machinelearning.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D0%BE%D0%B4\_%D0%91%D0%BE%D0%BA%D1%81%D0%B0-%D0%9A%D0%BE%D0%BA%D1%81%D0%B0]

Существует также вариант, когда предполагается работа не только с положительными, но и с отрицательными значениями. Для нашего случая этот вариант рассматривать не имеет смысла.

Воспользовавшись источником [3], был скачан дистрибутив Anaconda3-2020.11-Windows-x86.exe для работы с нейронными сетями. Стоит отметить, что при установке Anaconda устанавливается и Python.

Были также установлены библиотеки: matplotlib(3.2.1), tensorflow (2.2.0), pandas (1.0.3), sklearn (0.22.1), numpy (1.18.1). Установка библиотек было произведена через conda. Процесс установки можно видеть на рисунке 8.

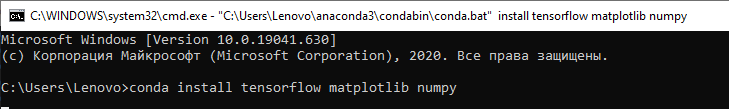


Рисунок 8 – Установка необходимых библиотек

В ходе выполнения практической части столкнулись с проблемой переобучения. Стоит отметить, что при решение многих задач нейросетевыми методами может возникнуть проблема переобучения, называемая оверфитом. Оверфит возникает при решении задач обучения по прецедентам, когда вероятность ошибки обученного алгоритма на объектах тестовой выборки оказывается существенно выше, чем средняя ошибка из обучающей выборки.

За последние несколько лет были предложены различные методы решения этой проблемы, одно из которых мы применили в своей дипломной работе, Dropout. Главная идея этого метода заключается в том, что вместо обучения одной DNN происходит обучение ансамблей DNN с последующим усреднением результатов.

Сети для обучения получаются с помощью исключения из сети нейронов с вероятностью , так, что вероятность того, что нейрон останется в сети составляет .

Для обучения сети был выбран метод обратного распространения ошибки.

Данный метод используется при обновлении весов в нейронной сети. Данный способ обучения нейронной сети помогает отрегулировать каждый вес пропорционально тому, насколько он способствует общей ошибке. При итеративном уменьшении ошибки каждого веса получится ряд весов, которые дают хорошие прогнозы.

Метод обратного распространения ошибки является специфической реализацией градиентного спуска в пространстве весов многослойных сетей прямого распространения. Основная идея этого метода заключается в вычислении частных производных функции сети по всем элементам настраиваемого вектора весов для данного вектора .

**3.2 Описание работы программы**

В практической части дипломной работы была реализована нейронная сеть с архитектурой LSTM-RNN.

Из источника [2] был скачан файл MDBS.csv. В нём находятся данные, характеризующие суммарный торговый трафик, указанный в долларах, для различных стран с 1957 года по 2020 год. Так как в файле MDBS.csv много параметров, большую часть из которых не имеет смысла применять относительно нашей задачи, был произведено извлечение нужных параметров из файла с помощью pandas.

Данные товарного оборота представляют из себя классический пример временного ряда, с выраженной цикличностью и, как правило, растущим трендом. Для корректного построения модели данные были приведены к виду стационарного временного ряда.

Для этого на первом этапе производилось логарифмирование значений временного ряда для устранения экспоненциально растущей амплитуды. Далее каждое значение в временной точке было заменено на разницу между -ой и -ой точкой (нулевая точка теряется), что позволяет избавится от тренда.

Полученные данные разбивались на небольшие вектора с длиной равной

,

где - это длина временного интервала, используемая для предсказания (по умолчанию 36 месяцев), - это длина предсказываемого участка (по умолчанию 3 месяца).

Каждый вектор был разбит на два фрагмента, длиной и соответственно. Левый фрагмент был определен как исходные данные, правый фрагмент как целевые (target). Таким образом, мы получили готовый датасет для обучения предсказательной модели.

Архитектуру модели представлена на рисунке 8.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type)                 Output Shape              Param #

=================================================================

lstm\_11 (LSTM)               (1, 64)                   24320

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_11 (Dropout)         (1, 64)                   0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_22 (Dense)             (1, 64)                   4160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_23 (Dense)             (1, 3)                    195

=================================================================

Total params: 28,675

Trainable params: 28,675

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Рисунок 8 – Модель LSTM-RNN

В качестве основного слоя был добавлен блок из 64 ячеек с архитектурой LSTM. Слой DropOut был добавлен для регуляризации (с долей обнуляемых весов 0.3). Необходимость регуляризационного слоя была очевидна, так как без него наблюдался рост значения функции потерь на валидационной части выборки уже с ранних итераций обучения, что явно говорит о наличии переобучения. Dense слой был использован для более аккуратного предсказания сложной зависимости.

Подробное описание LSTM слоя определено в первой главе.

В результате работы программы были построены следующие графики.

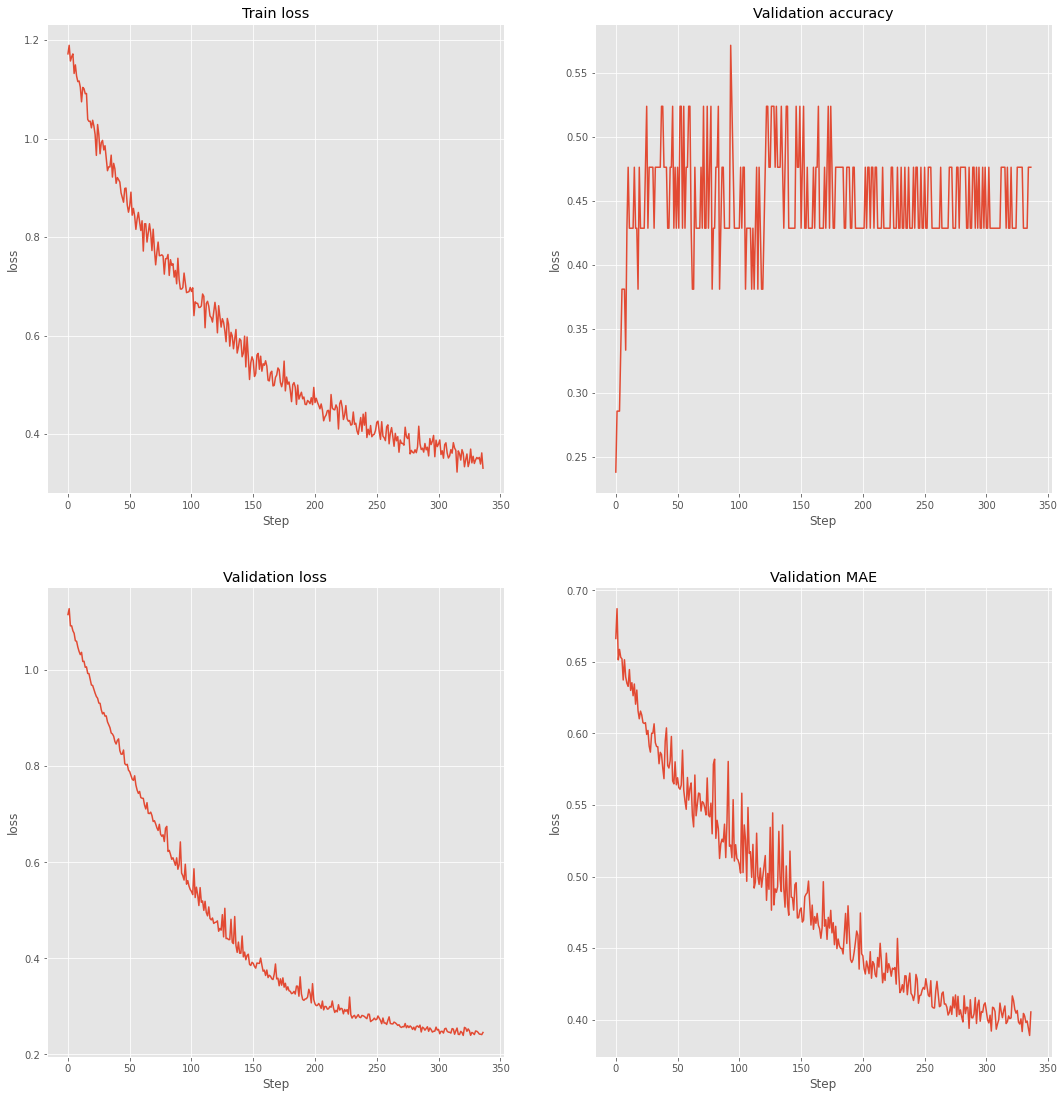


Рисунок 9 –

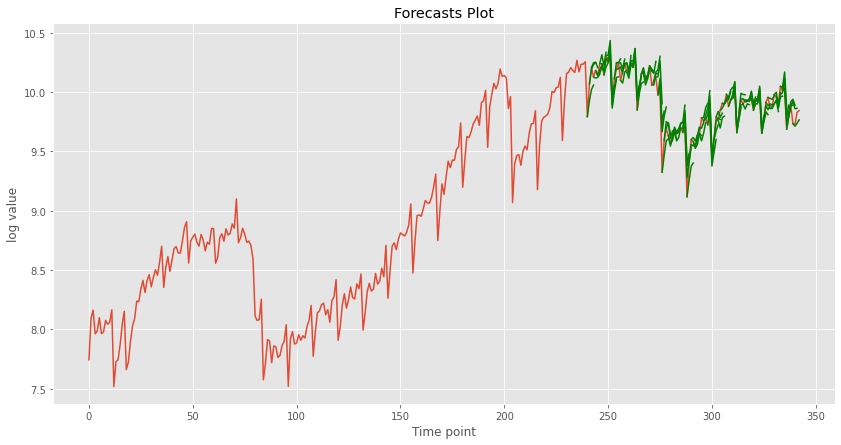


Рисунок 10 –

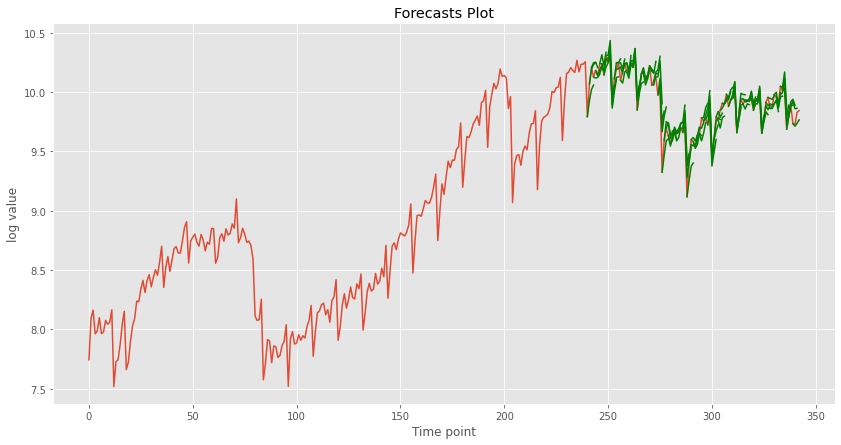
****

Рисунок 11 –

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ: