МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра математической кибернетики и компьютерных наук

ПРОГНОЗИРОВАНИЕ ЦЕН АКЦИЙ НА ОСНОВЕ ИСТОРИЧЕСКИХ ДАННЫХ И АНАЛИЗА НОВОСТНОЙ ЛЕНТЫ

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 273 группы
направления 01.04.02 — Прикладная математика и информатика
факультета КНиИТ
Аветисяна Севака Юриковича

Научный руководитель
доцент, к. ф.-м. н. _____ С. В. Миронов

Заведующий кафедрой
к. ф.-м. н. _____ С. В. Миронов

Саратов 2019

ВВЕДЕНИЕ

Машинное обучение с каждым днем занимает все большее место в повседневной жизни в виду огромного спектра решаемых задач. Основная идея машинного обучения в том, чтобы компьютер не просто использовал заранее прописанный алгоритм, а сам обучился решению поставленной задачи. В настоящее время биологи, экономисты, финансисты, социологи и другие ученые обрабатывают огромные объёмы информации, которую нужно как-то группировать, разбивать на заранее определённое число классов, то есть решать задачу классификации. Обычно экспериментальные данные плохо структурированы и весьма неточны. Поэтому традиционные математические методы классификации для таких данных работают плохо.

В последние годы все большую популярность приобретают методы Machine Learning — машинного обучения, в рамках которых программа сначала «обучается» на данных, про которые известно, к каким классам они принадлежат, при этом подбираются (настраиваются) определённые коэффициенты, с помощью которых в дальнейшем программа может определять принадлежность к классам новых данных.

Под глубоким обучением будем понимать совокупность методов машинного обучения, основанных на обучении представлениям, а не на специализированных алгоритмах, разработанных для конкретных задач. Многие методы глубокого обучения были известны еще в 1980-ых годах, но были неэффективны ввиду недостаточно вычислительной мощности, что делало невозможным в приемлемые сроки обучать сложные нейронные сети. Сейчас же, с ростом технологического прогресса, в том числе мощности графических процессоров, стало возможным не поддававшихся эффективному решению ранее, например, задача в компьютерном зрении, машинном переводе, распознавании и генерации речи. Важно отметить, что такие системы способны во многих случаях решать поставленную задачу не хуже, а порой даже лучше человека.

Алгоритмы машинного обучения применяются во многих областях человеческой жизнедеятельности, одна из таких областей — прогнозирование стоимости акций. Существует две методики прогнозирования цен акций — фундаментальный и технический анализ.

В случае фундаментального анализа аналитики оценивают информацию, которая больше относится к компании, чьи акции торгуются на бирже,

нежели к самим акциям. Решения о тех или иных действиях на рынке принимаются на основе анализа предыдущей деятельности компании, прогнозах выручки и прибыли и так далее.

При техническом анализе рассматривается поведение цены акций и выявляются его разнообразные паттерны (используется анализ временных рядов). В случае применения методов машинного обучения для обработки торговых данных, чаще используют именно метод технического анализа — цель заключается в том, чтобы понять, может ли алгоритм точно определять паттерны поведения акции во времени. Тем не менее, машинное обучение может использоваться также для оценки и прогнозирования результатов деятельности компании для дальнейшего использования при фундаментальном анализе. В конечном итоге, наиболее эффективным методом автоматизированного предсказания цены акций и генерирования инвестиционных рекомендаций является гибридный подход, сочетающий в себе подходы фундаментального и технического анализа.

Целью данной выпускной квалификационной работы является разработка программы на основе алгоритмов глубокого обучения для решения задачи прогнозирования трендов роста и падения стоимости акций на основе исторических данных и информации из новостной ленты.

Были поставлены следующие задачи:

— рассмотреть существующие алгоритмы машинного обучения и техники, используемые при построении глубоких нейронных сетей;
— рассмотреть существующие методы обучения нейронных сетей;
— рассмотреть существующие библиотеки глубокого обучения для языка java и python;
— подготовить тренировочные данные: исторические данные о стоимости акций за последние 10 лет, заголовки новостных статей;
— реализовать программу для прогнозирования тренда роста и падения акций на основе алгоритмов машинного обучения и анализа тональности текста.

1 Обзор литературы по машинному обучению

Данный раздел посвящен обзору литературы по машинному обучению. В нем рассмотрены фундаментальные исследования по теории статического обучения Владимира Вапника, проблемы недообученность и переобученность. В статье В. Вапника [1], подводятся краткие итоги теории статистического обучения к 1998 году. Статья состоит из четырех частей: сходимость обучающего процесса, скорость сходимости, контролирование обобщающих способностей обучающего алгоритма и конструирование обучающих алгоритмов. В статье Cherkassky, Ma [2] приводится сравнение двух подходов к оцениванию функций на основе конечного набора известных значений. Первый подход — это так называемая аппроксимация функций (FA — Function Approximation). Второй подход — теория статистического обучения (VC — Vapnic—Chervonenkis Theory). Авторы проводят сравнение в теоретическом и практическом плане. В статье тех же авторов [3] сравниваются подходы к оценке выбора модели между классическими методами (AIC, BIC) и методом SRM. Статья является ответом на похожую статью Hastie [4], в которой авторы усомнились в каких бы то ни было преимуществах подхода SLT. В статье Shao [5] рассматривается предложенный Вапником [6] метод экспериментального измерения VC-размерности и предлагается усовершенствованная версия, названная авторами оптимизированным экспериментальным дизайном. Предложенный Вапником метод заключается в том, что генерируется случайная выборка данных на этих данных оценивается способность алгоритма к их разделению. Отличие метода, предложенного в Shao [5], от метода Вапника [6] заключается в том, что используемые для конструирования точки имеют различные весовые коэффициенты, назначаемые алгоритмом. В статье Platt [7] представлен алгоритм SMO (Sequential Minimal Optimization). Данный алгоритм разбивает решение задачи КП на множество более простых подзадач. В статье Scheinberg [8] описывается метод ASM (Active Set Method). Как утверждает автор, данный метод был разработан для задач большой размерности. Как и SMO, ASM разбивает задачу на подзадачи, но, в отличие от SMO, он работает сразу с неким подмножеством векторов. Большая часть алгоритма — это работа с матрицами. Следующая статья Cherkassky, Ma [9] посвящена настройке параметров SVM для задачи регрессии. В такой задаче необходимо подбирать два параметра — (ширина

«трубки») и C (параметр сложности SVM). Если используется гауссово ядро, то необходимо еще подбирать параметр p (стандартное отклонение) для этого ядра. Для оценки эффективности предложенных настроек производятся измерения на искусственных данных. В статье Liu [10] приводится исследование относительно нового алгоритма машинного обучения — ELM. ELM (Extreme Learning Machine) — это алгоритм обучения, который по структуре представляет из себя нейронную сеть с одним скрытым слоем. Обучение же данной сети происходит по-другому. Веса для скрытого слоя назначаются случайным образом, а для открытого — взятием псевдообратной матрицы. Также есть версии ELM, где функцией активации для нейронов скрытого слоя является нестандартная функция.

2 Обзор алгоритмов машинного обучения

В данном разделе рассмотрены следующие алгоритмы машинного обучения:

— Логистическая регрессия;
— Искуственные нейронные сети;
— Многослойный персептрон;
— Деревья решений и случайный лес.

Также рассмотрены подходы, применяемые при обучении нейронных сетей. Наконец, рассмотрены основы анализа тональности текста.

Логистическая регрессия — метод построения линейного классификатора, позволяющий оценивать апостериорные вероятности принадлежности объектов классам.

Модель искусственного нейрона была предложена Уорреном МакКаллоком (Warren McCulloch) и Уолтером Питтсом (Walter Pitts) в 1943 году. В качестве основы для своей модели авторы использовали биологический нейрон. Искусственные нейронные сети бывают двух видов: без циклов (сети прямого распространения сигнала), и с циклами (рекуррентные сети).

Многослойный персептрон — нейронная сеть прямого распространения. Входной сигнал распространяется от слоя к слою, в прямом направлении. Многослойный персептрон является обобщением однослойного персептрона Розенблатта.

Под деревом решений будем понимать алгоритм принятия решений при прогнозировании, широко применяющееся в статистике и анализе данных. Структура дерева представляет собой листья и ветки. На рёбрах дерева решения записаны атрибуты, от которых зависит целевая функция, в листьях записаны значения целевой функции, а в остальных узлах — атрибуты, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение. Подобные деревья решений широко используются в интеллектуальном анализе данных. Цель состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной на основе нескольких переменных на входе. Цель всего процесса построения дерева принятия решений — создать модель, по которой можно было бы классифицировать случаи и решать, какие значения может принимать целевая функция, имея на входе несколько переменных.

Случайный лес — алгоритм машинного обучения, предложенный Лео Брейманом и Адель Катлер, заключающийся в использовании комитета (ансамбля) решающих деревьев. Алгоритм сочетает в себе две основные идеи: метод бэггинга Бреймана, и метод случайных подпространств, предложенный Tin Kam Ho. Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт очень невысокое качество классификации, но за счёт их большого количества результат получается хорошим.

Обучение нейронной сети — это процесс определения весов соединений между нейронами таким образом, чтобы сеть приближала необходимую функцию с заданной точностью. Существует три подхода к обучению нейронных сетей: обучение с учителем (supervised learning), обучение без учителя (unsupervised learning) и обучение с подкреплением (reinforcement learning). При обучении с учителем на вход сети подаются наборы входных сигналов (объектов), для которых заранее известен правильный ответ (обучающее множество). Веса меняются по определенным правилам в зависимости от того, правильный ли выходной сигнал выдала сеть. При обучении без учителя на вход сети подаются объекты, для которых правильный выходной сигнал заранее не известен. Обучение с подкреплением предполагает наличие внешней среды, с которой взаимодействует сеть. Обучение происходит на основании сигналов, полученных от этой среды. Нейронные сети Макаллока— Питтса не обучались. Веса для всех входов нейронов должны были быть заданы заранее. Идея обучения нейронных сетей была придумана Дональдом Хэббом в 1949 году. Согласно Хэббу, связи нейронов, которые активируются вместе, должны усиливаться, а связи нейронов, которые срабатывают отдельно друг от друга, должны ослабевать. Обучение с учителем, когда на основе ответа, выдаваемого сетью и правильного ответа, корректировались веса во входном слое, было предложено Хэббом. А.В. Новиковым была доказана сходимость предложенного метода обучения нейрона на основе правил Хэбба при условии линейной разделимости выборки.

Анализ тональности в тексте является одним из направлений в анализе естественно-языковых текстов. Тональностью называется эмоциональная оценка, которая выражена в тексте. Она может иметь одномерное эмотивное пространство (два класса) или многомерное (несколько классов).

3 Реализация программы для прогнозирования тренда роста и падения стоимости акций

В данной главе рассмотрим реализацию программ для прогнозирования тренда роста и падения стоимости акций. Для начала рассмотрим существующие библиотеки машинного обучения для языков python и Java. Определим методы сборка и подготовки исторических данных о стоимостях акций а так же данных из новостной ленты. Наконец, рассмотрим реализацию программ на основе алгоритмов машинного обучения, а именно случайного леса, многослойного персептрона, логистической регрессии, а так же модели нейронной сети LSTM. Для каждой из полученных моделей создадим график предсказаний тренда роста и падения, проанализируем его и попытаемся выявить, в каком количестве случаев алгоритм выдал верное решение.

Рассмотрим основные библиотеки для языков python и java, используемые в машинном и глубинном обучении. В настоящее время создано большое количество программных систем для обучения глубоких нейронных сетеи. Для языка python среди наиболее популярных из них можно отметить Caffe, Theano, TensorFlow, Torch, Keras, cuDNN, skikit и CNTK. Среди наиболее популярных для языка java можно отметить Weka, MOA, Deeplearning4j и MALLET.

TensorFlow — открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия Применяется как для исследований, так и для разработки собственных продуктов Google. Основное API для работы с библиотекой реализовано для Python, также существуют реализации для C++, Haskell, Java и Go.

На сегодняшний день TensorFlow очень популярен для решения задач глубинного обучения, он используется множеством других библиотек, о которых дальше и пойдет речь.

Scikit-learn — библиотека, которая предоставляет реализацию целого ряда алгоритмов для обучения с учителем (Supervised Learning) и обучения без учителя (Unsupervised Learning) через интерфейс для языка программирования Python.

Одна из основных концепций библиотеки scikit-learn — библиотека с вы-

соким уровнем надежности и поддержки, большое внимание уделяется вопросам удобства использования, качества кода, документации и оптимизации скорости работы библиотеки.

Несмотря на то что весь интерфейс библиотеки представлен на Python, но использование библиотек, написанных на C во внутренней реализации некоторых частей scikit-learn, позволяет значительно повысить скорость работы, например, использование NumPy для работы с массивами и для операций с матрицами.

NLTK — пакет библиотек и программ для символьной и статистической обработки естественного языка, написанных на языке программирования Python. Содержит графические представления и примеры данных. Сопровождается обширной документацией, включая книгу с объяснением основных концепций, стоящих за теми задачами обработки естественного языка, которые можно выполнять с помощью данного пакета.

Deeplearning4j — один из инновационных участников, который вносит значительные инновации в экосистему Java. Идея DeepLearning4j состоит в том, чтобы объединить глубокие нейронные сети и углублённое обучение для бизнес-среды. Deeplearning4j — отличный инструмент, который работает с Hadoop, фреймворком для разработки и выполнения распределённых программ, работающих на кластерах с тысячами узлов. Hadoop обладает огромной вычислительной мощностью и возможностью обработки практически неограниченного числа параллельных задач. Глубокие нейронные сети и глубокое обучение с подкреплением (deep reinforcement learning) способны к распознаванию образов и целевому машинному обучению. Это означает, что Deeplearning4j очень полезен для определения моделей и настроений в речи, звуке и тексте. Кроме того, библиотеку можно использовать для обнаружения аномалий в данных временных рядов, таких как финансовые транзакции.

Все перечисленные системы глубокого обучения нейронных сетей могут использовать для ускорения обучения как многоядерные процессоры, так и ускорители вычислении GPU (включая оптимизированную библиотеку cuDNN). Причем существенным преимуществом является то, что нет необходимости переделывать программу, распараллеливание на CPU и GPU выполняется автоматически. Системы Caffe и Theano дополнительно поддерживают ускорители Intel Xeon Phi, которые также помогают существенно сокра-

тить время обучения глубоких нейронных сетей. Почти все системы, кроме Theano, можно использовать для распределенного обучения нейронных сетей на вычислительных кластерах.

Для получения исторической информации о стоимости акций вопспользуемся сервисом Yahoo Finance. В результате будут получены данные о стоимости акций на начало, конец торгов за указанный промежуток времени с периодичностью в 1 день. Для получения заголовков новостных статей воспользуемся New York Times Archive API. В рамках данной работы используется поле *headline*, в котором находится заголовок статьи, а так же дата публикации. В течение календарного дня публикуется большое количество новостей. В свою очередь информация о стоимости акций на бирже доступна лишь с минимальным интервалом в один календарный день. Встает вопрос, каким образом объединить данные из новостной ленты и данные о стоимости акций для дальнейшего использования в обучении нейронной сети. В данной главе рассмотрены три способа решения данной задачи: сбор всех новостей, подсчет наиболее релевантной новости за календарный день и подсчет среднего для новости за календарный день на основе анализа тональности.

Реализация программы для прогнозирования стоимости акций на основе алгоритмов машинного обучения и анализа тональности текста состоит из следующих шагов:

— прочитать тестовые данные;
— дополнить данные результатом анализа тональности заголовков новостей;
— воспользоваться готовыми реализациями алгоритмов случайного леса, многослойного персептрона и логистической регрессии из пакета sklearn;
— описать модель нейронной сети LSTM;
— обучить модели;
— построить графики с актуальной и предсказанной ценой.

Для проверки качества полученного результата подсчитаем относительный прирост r_t стоимости акции за день t по следующей формуле:

$$r_t = \log_{10} \frac{p_t}{p_{t-1}}$$

где p_t — стоимость акции в день t. Положительное значение r_t сигнализирует о положительном тренде и, соотвественно, росте, отрицательное же — о падении стоимости. Таким образом подсчитаем r_t для актуального и предсказанного моделью значений, будем считать, что предсказание верно, если тренд в обоих случаях совпадает.

Подсчитаем для каждой из моделей процентное соотношение правильных предсказаний к общему числу предсказываемых значений. В случае со случайным лесом получим $\approx 54\%$, для модели логистической регресии $\approx 51\%$, для многослойного персептрона — 56%. Таким образом, ни одна из моделей не дала сколько нибудь хорошего результата, предсказывая тренд правильно в среднем в половине случаев.

Рассмотрим результат работы нейронной сети на трех наборах данных, подготовленных ранее. На одном и том же наборе данных запустим программу дважды, в первом случае будем подавать на вход все данные из набора, во втором — только лишь данные об исторических стоимостях. Таким образом попробуем выяснить наличие или отсутствие влияния новостей на тренды роста или падения акций. Для полученных значений подсчитаем процентное соотношение правильных предсказаний к общему числу предсказываемых значений, так же, как и в прошлой главе.

В первом случае подадим на вход новости, усредненные в каждый из календарных дней. Данная модель была успешна лишь в 74% случаев. Во втором случае подадим на вход данные со всеми имеющимися новостями. С учетом всех новостей и результатов анализа тональности текста модель предсказала тренд верно в 86% случаев, без учета результатов анализа тональности лишь в $\approx 83\%$. Наконец, подадим на вход наиболее релевантные новости за кажендарный день. С учетом наиболее релевантных новостей и результатов анализа тональности текста модель предсказала тренд верно в $\approx 80\%$, без учета результатов анализа тональности — 74%.

Подводя итоги сравнительного анализа, можно сказать, что анализ тональности новостей улучшает работу модели, тренд роста и падения стоимости акций предсказывается правильно в не менее чем 74% случаев. Наиболее удачные результаты показала модель, принимающая на вход наиболее релевантные новости, правильно предсказав тренд в 86% случаев.

ЗАКЛЮЧЕНИЕ

С момента возникновения нейронных сетей произошло много изменений в их архитектуре и методах обучения. На данный момент существует два типа архитектур: сверточные сети и рекурентные сети. В рамках данной выпускной квалификационной работы были рассмотрены существующие методы, используемые при построении и обучении глубоких нейронных сетей. В частности были рассмотрены архитектуры сверточных, рекурентных сетей, а так же сетей с долго-краткосрочной памятью. Также были рассмотрены библиотеки машинного обучения для языка java и python. Были подготовлены тренировочные данные: исторические данные о стоимости акций за последние 10 лет, заголовки новостных статей. Были разработаны программы для прогнозирования цен акций на основе исторических данных с использованием алгоритмов машинного обучения (случайный лес, многослойный персептрон, логистическая регрессия) и глубокого обучения (LSTM) в частности. Очевидно, что существует огромное количество параметров, влияющих на тренды роста и падения, и, безусловно, новости — один из таких. Проведя сравнительный анализ результатов можно сказать, что глубокая нейронная сеть дала более качественные результаты, предсказав тренды роста и падения стоимости акций в большем количестве случаев.

Данная работа была представлена на студенческой научной конференции факультета КНиИТ 2019 года.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Vapnik, V. N.* An Overview of Statistical Learning Theory / V. N. Vapnik. — Wiley-Interscience, 1999.

2 *Cherkassky, V.* Another look at statistical learning theory and regularization / V. Cherkassky, Y. Ma // *Neural Network.* — 2009. — Vol. 22. — Pp. 958–969.

3 *Cherkassky, V.* Comparison of model selection for regression / V. Cherkassky, Y. Ma // *Neural Computation.* — 2003. — Vol. 15. — Pp. 1477 – 1480.

4 *Hastie, T.* The elements of statistical learning: Data mining, inference and prediction. / T. Hastie, R. Tibshirani, J. Friedman. — Springer, 2001.

5 *Shao, X.* Measuring the vc-dimension using optimized experimental design / X. Shao, V. Cherkassky, W. Li // *Neural Computation.* — 2000. — Vol. 12. — Pp. 1969–1986.

6 *Vapnik, V. N.* Measuring the vc-dimension of a learning machine / V. N. Vapnik, E. Levin, Y. L. Cun // *Neural Computation.* — 1994. — Vol. 6. — Pp. 851–876.

7 *Platt, J.* Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines [Электронный ресурс] / J. Platt. — URL: https://www.researchgate.net/publication/2624239_Sequential_Minimal_Optimization_A_Fast_Algorithm_for_Training_Support_Vector_Machines (Дата обращения 02.10.2018). Загл. с экр. Яз. англ.

8 *Scheinberg, K.* An efficient implementation of an active set method for svms / K. Scheinberg // *J. Mach. Learn. Res.* — 2006. — Vol. 7. — Pp. 2237–2257.

9 *Cherkassky, V.* Practical selection of svm parameters and noise estimation for svm regression / V. Cherkassky, Y. Ma // *Neural Network.* — 2004. — Vol. 17. — Pp. 113–126.

10 *Liu, X.* A comparative analysis of support vector machines and extreme learning machines / X. Liu, C. Gao, P. Li // *Neural Network.* — 2012. — Vol. 33. — Pp. 58–66.