



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ

Робототехника и комплексная автоматизация (РК)

КАФЕДРА

Системы автоматизированного проектирования (РК6)

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

«Применение метода дообучения свёрточных нейронных сетей (CNN) для решения задач технического анализа при управлении активами на фондовом рынке»

Студент РК6-86Б

(Подпись, дата)

Онюшев А.А.

И.О. Фамилия

Руководитель

(Подпись, дата)

Витюков Ф.А.

И.О. Фамилия

2024

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой РК6
А.П. Карпенко

«_____» _____ 2024 г.

З А Д А Н И Е

на выполнение научно-исследовательской работы

по теме: применение метода дообучения свёрточных нейронных сетей (CNN) для решения задач
технического анализа при управлении активами на фондовом рынке

Студент группы РК6-86Б

Онюшев Артем Андреевич
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.) учебная
Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения НИР: 25% к 5 нед., 50% к 11 нед., 75% к 14 нед., 100% к 16 нед.

Техническое задание: Написание модуля дополнительного обучения нейронной сети на основе
данных полученных за новый торговый день. Исследование эффективности данного метода. Подбор
наилучших параметров.

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на 30 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.):

Дата выдачи задания «7» февраля 2024 г.

Руководитель НИР

(Подпись, дата)

Витюков Ф.А.
И.О. Фамилия

Студент

(Подпись, дата)

Онюшев А.А.
И.О. Фамилия

СОДЕРЖАНИЕ

ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ	4
ВВЕДЕНИЕ	5
1. Разработка модуля дообучения	6
1.1. Реализация класса создания датасетов.....	8
1.2. Реализация метода дообучения.....	10
2. Проведение исследования	11
2.1. Исследования	12
2.2. Эффективность метода дообучения	15
2.3. Закономерности	16
2.4. Оптимальные настройки.....	17

ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

НС – Нейронная сеть.

CNN – (Convolutional Neural Networks) Свёрточная НС.

ДС – Дата сет.

Функция-потерь – Функция, показывающая расстояние от предугаданного НС ответа до истинного ответа.

Функция-активации – Функция, отвечающая за изменение весов синапсов.

Тикер – Краткое название в биржевой информации котируемых инструментов (акций, облигаций, индексов).

ЕМА – (Exponential Moving Average) Экспоненциальная скользящая средняя. Один из показателей, помогающих при техническом анализе.

Stop-Loss – Биржевая заявка трейдера, в которой условием исполнения указано достижение цены, которая хуже, чем текущая рыночная.

Take-Profit – Биржевая заявка трейдера, которая выставляется заранее, чтобы в случае роста рынка зафиксировать прибыль по бумаге.

MSE – (Mean Squared Error) Метрика, используемая для оценки эффективности работы регрессионной модели.

MAE – (Mean Absolute Error) Метрика, используемая для оценки эффективности работы регрессионной модели.

ВВЕДЕНИЕ

В современном мире доступ к знаниям обширен и большинство людей обучаются чему-то новому каждый день. Новые знания помогают находить новые пути решения проблемы, помогают принимать решения основываясь на похожих ситуациях в прошлом и просто делают человека многогранным. Нейронные сети берут свое начало в биологии и анатомии человека и имеют внутри себя многие схожие системы и архитектуры. Работы, написанные мною до этого, использовали только простой (конечный) метод прямого обучения нейронной сети, в данной же работе, основываясь на примере реального человека, который получает новые знания каждый день, будет проведено исследование с использованием так называемого цикла дообучений НС. Каждый новый день у брокера появляется новая сущность ДС с информацией о выставленных им take-profit и stop-loss. Этой новой информации и будем дообучать НС.

Целью данной научно-исследовательской работы является разработка и проведение исследования модуля дообучения для CNN. Также в данной работе будут найдены оптимальные настройки для такой НС.

1. Разработка модуля дообучения

Добавление в проект дообучения, предположительно, поможет увеличить точность предугаданных НС значений. Чтобы реализовать такой модуль, нужно разобраться, как будет работать наше дообучение. В данной работе я буду использовать такой вариант:

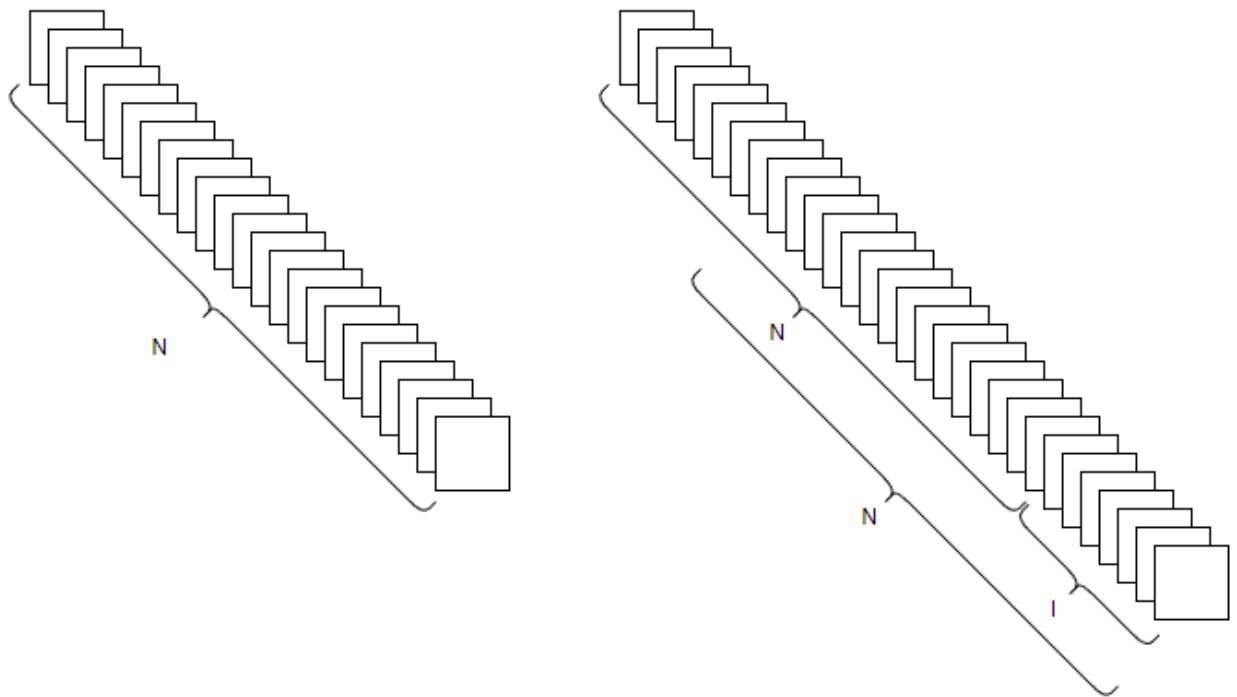


Рис. 1 Вариант дообучения.

Пусть у нас имеется ДС состоящий из N сущностей и начинающийся с 0-ой позиции. Обучим нашу НС на этом ДС используя обычное (конечное) обучение. После первого дня использования НС мы получим от брокера ещё одну сущность в наш ДС (того $N+1$). Не будем терять возможность получения нейронной сетью новой информации и дообучим её используя ДС, состоящий также из N сущностей, но уже начинающийся с 1 позиции (напомню, что у нас, на данный момент, всего $N+1$ сущность всего). На i -ый день у нас уже будет $N+I$ сущностей и мы опять дообучим НС ДС из N сущностей но с i -ой позиции и так далее.

Такой метод дообучения позволит нам обучить сперва НС всем данным, что у нас были до начала работы брокера, а далее дообучать актуальными данными на основе уже поведения брокера.

1.1. Реализация класса создания датасетов

Как можно заметить, для такого обучения, нам часто придется создавать ДС, начинающиеся с разных позиций, поэтому удобно будет вынести эту функциональность в отдельный класс.

Листинг 1 – Класс создания датасетов.

```
1 class Dataloader():
2     def __init__(self):
3         options_path = 'config.yml'
4         with open(options_path, 'r') as options_stream:
5             options = yaml.safe_load(options_stream)
6             self.dataset_options = options.get('dataset')
7             data_path = self.dataset_options.get('data_file_name')
8             EMA_N = self.dataset_options.get('EMA')
9             self.data = load_data(data_path, EMA_N).data
10
11     def get_dataloader(self, start: int | None = None, stop: int
12 | None = None, additional: bool = False, test: bool = False) ->
13 torch.utils.data.DataLoader:
14     """
15     Метод создает даталоадер с нужными настройками в
16 зависимости от этапа (обучение/дообучение). \n
17 С нужной начальной позиции и до нужной конечной позиции.
18
19 :param start: С какой позиции
20 :param stop: По какую позицию
21 :param additional: Дообучение (Да/Нет)
22 :param test: Валидация (Да/Нет)
23 :return: loader: Возвращает даталоадер
24 """
25
26     loader_options = self.dataset_options.get('train_loader')
27     if additional is True:
28         loader_options =
29 self.dataset_options.get('additional_loader')
30     if test is True:
31         loader_options =
32 self.dataset_options.get('test_loader')
33
34     if start is None:
35         start = loader_options.get('start')
36     if stop is None:
37         stop = loader_options.get('stop')
38
39     pdl = PreDataLoader(self.data,
40 candle_count=self.dataset_options.get('candle_count'),
41                        start=start, stop=stop,
42 normalization_pred=self.dataset_options.get('normalization'),
43                        vers=self.dataset_options.get('vers'),
44 lbl=get_label(self.dataset_options.get('label'), False if
45 additional or test else True))
```



```
35         ds = DataSet(pdl.batches)
36
37         loader = torch.utils.data.DataLoader(
            ds, shuffle=loader_options.get('shuffle'),
            batch_size=loader_options.get('batch_size'),
            num_workers=loader_options.get('num_workers'),
            drop_last=loader_options.get('drop_last')
        )
38         return loader
```

В ходе выполнения нашей программы, будет создаваться много ДС, поэтому в данной реализации при инициализации класса (она происходит единоразово при вызове конструктора класса) в параметр `self.data` класса будут записаны все имеющиеся сущности, а уже позже, при вызове функции `get_dataloader()`, из них будут выбраны сущности с нужной позиции и в нужном количестве.

1.2. Реализация метода дообучения

Так как теперь в работе участвует две разных версии обучения – простая (конечная) и дообучения, удобно будет сразу же вынести эти методы в функции.

Функции большого размера, их можно посмотреть в файле по ссылке (https://github.com/RelaxFM/CNN_VKR/blob/master/Func/learning_version.py).

Функция простого (конечного) метода обучения названа в файле, как “feedforward”. Функция дообучения же названа, как “additional_learning”.

Теперь вся необходимая функциональность у нас готова и можно переходить к экспериментальной части.

2. Проведение исследования

Возложим на исследование несколько целей:

- 1) Провести большое количество исследований;
- 2) Доказать эффективность использования метода дообучения;
- 3) Выявить какие-то закономерности;
- 4) Выявить оптимальные настройки.

Так как, при проведении предыдущего исследования на CNN с простым (конечным) методом обучения, функции-ошибки MAE и MSE показали схожие результаты по эффективности, то в данном исследовании будут приведены примеры только с одной из метрик. Будем использовать MAE – она немного лучше подходит для данного вида задач.

2.1. Исследования

Мы уже выбрали нужную нам реализацию метода дообучения и теперь нам остается только решить с какими настройками количества эпох проводить тесты. Так как в предыдущих работах для простого (конечного) метода обучения были выбраны 10000, 6400, 3200, 1600, 800, 400, 200, 100, 50, то для простоты сравнения будем использовать эти количества для тестов при использовании простого метода. Для дообучения же будем использовать другие величины эпох, а именно: 1000, 512, 256, 128, 64, 32, 16, так как дообучение это, возможно, бесконечное количество этапов простого обучения.

Все полученные исследования можно найти по ссылке (<https://github.com/Relax-FM/Diploma/blob/main/Diploma/excel/DIPLOMA.xlsx>) на странице с названием «ВКР».

Вот часть из них:

MAE	feedforward	CNN	Time	Relation	Average_I	Average_I	Standard_Error, %	Max_error	
		10000	303.76	266.883	0.5352	0.5137	0.1952	36.47	3012.5
		6400	193.88	285.79	0.5352	0.5673	0.1774	33.1403	3593.116
		3200	100.60	275.59	0.5352	0.4757	0.2286	42.7062	3933.08

Рис. 2 Часть эксперимента с простым (конечным) обучением.

MAE	adittional	CNN	Time	Relation	Average_I	Average_I	Standard_ Error, %	Max_error	
		1000	1351.0	206.03	0.5352	0.5003	0.1458	27.236	3192.7
		512	721.89	233.22	0.5352	0.5379	0.1344	25.122	3020.67
		256	368.28	253.41	0.5352	0.5371	0.151	28.122	3717.36
		128	185.44	233.56	0.5352	0.5373	0.154	28.768	3584.97
		64	97.554	253.27	0.5352	0.5434	0.1441	26.919	3575.38
		32	49.222	246.968	0.5352	0.5265	0.1503	28.090	3455.447
		16	28.547	251.83	0.5352	0.5386	0.1512	28.260	3209.02
1000	1375.9	219.95	0.5352	0.4766	0.1603	29.96	3581.503		
512	740.98	218.544	0.5352	0.5295	0.1256	23.47	3464.65		
256	383.83	216.15	0.5352	0.5278	0.1387	25.92	3506.72		
128	190.14	218.57	0.5352	0.5298	0.1411	26.362	3528.90		
64	98.267	229.67	0.5352	0.5266	0.1445	26.998	3598.98		
32	53.72	232.57	0.5352	0.5258	0.1520	28.398	3453.747		
16	29.25	253.47	0.5352	0.5431	0.1493	27.893	3962.455		
1000	1397.9	240.374	0.5352	0.4753	0.1998	37.33	3993.47		
512	741.91	258.23	0.5352	0.5372	0.1802	33.663	4116.83		
256	374.90	243.70	0.5352	0.5145	0.1894	35.385	4267.04		
128	197.00	241.28	0.5352	0.5189	0.1967	36.74	3953.2		
64	95.99	241.89	0.5352	0.5055	0.2005	37.45	3951.2		
32	52.64	242.19	0.5352	0.4969	0.2019	37.725	4194.01		
16	28.091	249.14	0.5352	0.5006	0.207	38.683	4231.87		

Рис. 3 Часть эксперимента с дообучением для простого обучения с рис.2.

2.2. Эффективность метода дообучения

Как можно увидеть по метрикам или графикам, результаты, полученные после дообучения выглядят более точными, нежели результаты, полученные после простого обучения. Это говорит о том, что использование метода дообучения приводит к увеличению аналитических способностей НС и повышению точности.

Вот некоторые графики для сравнения до дообучения и после:

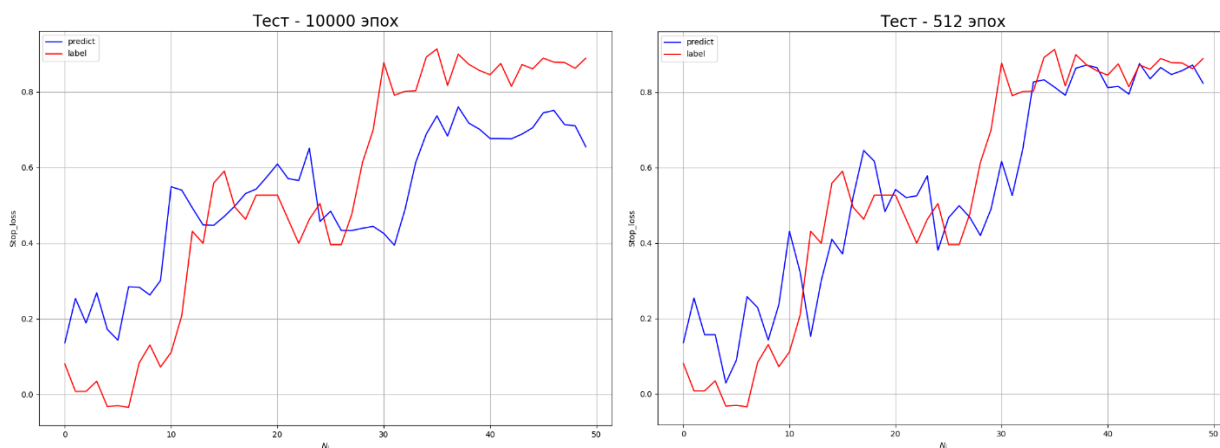


Рис. 4 Слева изображен график до дообучения, справа – после.

(Сверху подписано количество эпох, использованных на этапе простого обучения и дообучения, соответственно)

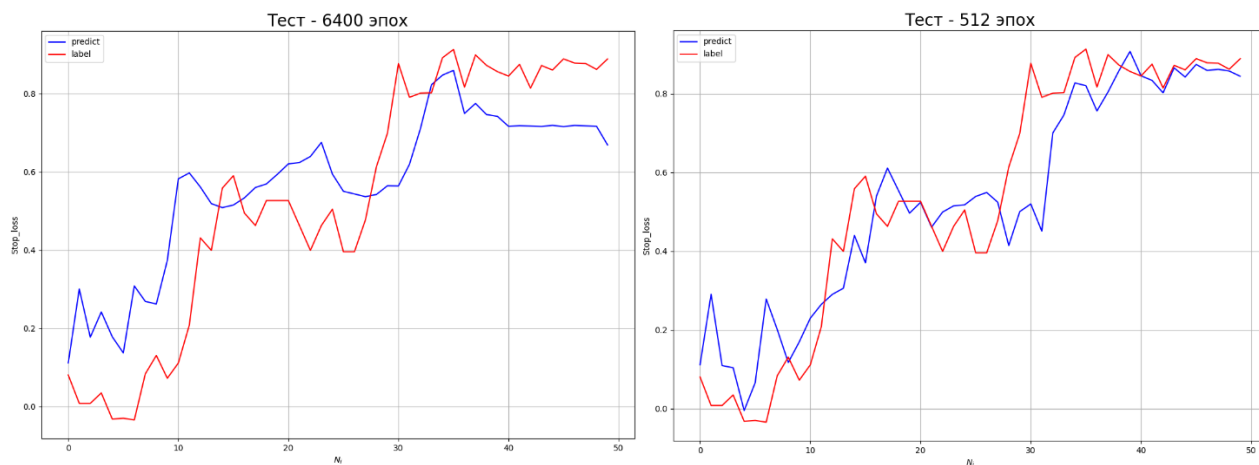


Рис. 5 Слева изображен график до дообучения, справа – после.

По графикам можно также заметить, что после дообучения, график предсказаний (выделен синим цветом) расположен более близко к графику истинных значений. Следовательно, является более точным.

2.3. Закономерности

При проведении исследования, удалось выяснить, что при малом количестве эпох на этапе обучения в левой части графика появляются большие осцилляции, связанные, вероятно, с недостаточным обучением НС. Начиная с 1600 эпох для этапа простого обучения осцилляции начинают проходить и появляется более плавный график.

На этапе дообучения достаточно хорошие результаты показывают метрики и графики для 512 эпох. Можно выбрать это число эпох, как оптимальное.

В целом, этап дообучения дает хорошую аппроксимацию истинного графика. Даже при условии, что на этапе простого обучения используют малое количество эпох, этап дообучения сглаживает график предсказаний, тем самым приближая его к истинному графику.

Также удалось выяснить, что этап дообучения позволяет повысить точность примерно на 25%, но при этом придется пожертвовать временем. Это время в масштабах целого дня не сильно большое и им можно пренебречь для такого рода задач.

2.4. Оптимальные настройки

Сложно подобрать оптимальные настройки под задачи похожего типа, но для рассматриваемого случая оптимальными настройками можно назвать:

1. Для этапа простого обучения – 6400 эпох.
2. Для этапа дообучения – 512 эпох.