ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Московский институт электроники и математики

Хабибуллин Данил Радикович

Решение задачи регрессии с помощью нейронных сетей. Предсказание результата футбольного матча по результату первого тайма

Отчет по проекту дисциплины «Профориентационный семинар "Введение в специальность"» студента образовательной программы бакалавриата «Прикладная математика»
по направлению подготовки 01.03.04 ПРИКЛАДНАЯ МАТЕМАТИКА

Студент
Д.Р. Хабибуллин

Руководитель проекта
д.ф.-м.н., профессор
В.Ю. Попов

Москва 2025г.

Содержание

Введение

Мир наполнен различными объектами, которые могут каким-либо образом зависеть друг от друга. Зачастую эти самые завимости человеку отследить достаточно сложно, поэтому в наш век достаточно популярен регрессионый анализ, который подразумевает под собой поиск этой зависимости между разными переменными. В особенности популярен регресионный анализ с помощью методов машинного обучения, который позволяет выполнять прогнозирование значения переменной, зависящей от других независимых переменных.

В течение долгого времени было сложно понять, с каким же счетом закончится тот или иной футбольный матч, однако, получившие большую популярность, нейросети могут оказать помощь в этом самом предсказании.

Цель проекта - разработать нейронную сеть, позволяющую спрогнозировать результат футбольного матча по результатам первой половины игры.

Для выполнения цели проекта были поставлены следующие задачи:

1) Познакомиться с работой нейросетей

2) Выполнить обработку данных результатов матчей Испанской Ла Лиги

3) Создать нейросеть по предсказанию результатов футбольного матча по результатам первого тайма и обучить ее

4) Посмотреть на работу нейросети на тестовых данных, после чего оценить ее

1 Теоретическая часть

В данной главе рассматриваются теоретичские вопросы. А именно, что такое нейросеть и как она работает, каким образом она создается и обучается, как оценить ее работу, а также будут упомянуты некоторые аспекты игры в футбол. Все это поможет больше внедриться в тему исследования

1.1 Нейронные сети в регрессионном анализе

Регрессионный анализ - вид анализа, помогающий в поиске факторов, показывающих зависимость целевой переменной от независимой/независимых. Цель задачи регрессии — предсказать значение числовой переменной при данных независимых переменных, которые могут быть либо числовыми, либо категориальными.

Нейронная сеть или нейросеть — математическая модель, работающая по принципам нервной системы живых организмов. Именно они помогают решать самые сложные задачи регрессии с использованием машинного обучения.

Наиболее популярной и достаточно простой нейросетью для решения задач регрессии является многослойный персептрон. Он состоит из:

1. Входного слоя: значения, передаваемые пользователем
2. Скрытых слоев: слои, следующие за входным, между ними вычисляются веса зависимости, а также происходит активация нейронов

3. Выходного слоя: один или несколько персептронов, которые дают окончательный выход сети.

4. Весовых коэффициентов модели: веса связанных нейронов, которые изменяются в процессе обучения модели для достижения оптимального результата.

5. Функции активации: преобразует входной сигнал от одного слоя к другому

Стоит более подробно сказать о функции активации: она определяет, каким образом нейрон будет реагировать на входные данные и передавать данные, благодаря ней добавляется нелинейность в модель, что позволяет нейронной сети обучаться сложным функциям.

Для построения нейронной сети будет использована функция активации, преобразующая выходное значение нейрона в значение из диапазона $[0, +\infty)$. Формула 1 линейного выпрямителя (ReLu) :

$$f(x) = \begin{cases} x, x > 0 \\ 0, x \leq 0 \end{cases} \tag{1}$$

1.2 Ошибка нейронной сети

В машинном обучении функция потерь измеряет, насколько выходные данные модели отклоняются от желаемых целевых значений. Она используется для оценки производительности модели и направления тренировки для улучшения результатов

сети. Для задач регрессии в качестве функции потерь чаще всего используют формулу 2 MSE .

$$MSE = \frac{1}{n}\sum_{n=1}^{n}(y_r - y_p)^2 \tag{2}$$

Обратное распространение ошибки - это алгоритм, который обучает нейронные сети путем расчета градиентов функции потерь с помощью градиентного спуска. Он используется для оптимизации весов и смещений сети следующим образом. Вычисляются выходные значения сети для входных данных. Определяется разница между прогнозируемыми и фактическими значениями для оценки потерь. Градиенты функции потерь рассчитываются, двигаясь назад через сеть, послойно, к начальным весам и смещениям. Веса и смещения сети настраиваются путем вычитания градиентов из их текущих значений, что сводит к минимуму функцию потерь и повышает точность сети.

Оптимизатор нейросети - это алгоритм, который используется для обновления весов и смещений нейронной сети во время процесса обучения. Целью оптимизатора является минимизация функции потерь сети, которая измеряет несоответствие между выходными значениями сети и ожидаемыми значениями. Для создаваемой нейросети отлично подходит Adam - оптимизатор, который использует как импульс, так и адаптивную скорость обучения. Это позволяет настраивать скорость обучения каждой переменной индвиудально.

Регуляризация - это набор методов, используемых в машинном обучении для предотвращения переобучения моделей. Переобучение происходит, когда модель слишком хорошо подстраивается под обучающие данные и теряет способность обобщать на новые данные, на которых нейронная сеть не тренировалась. Дря разработки нейросети понадобятся L2-регуляризация (Ridge) и Dropout.

L2-регуляризация - добавляет к функции потерь штраф, пропорциональный сумме квадратов весов модели. Это приводит к более гладким решениям, где веса меньше по величине.

Dropout - это метод регуляризации, используемый в нейронных сетях, который заключается в случайном обнулении активаций нейронов во время обучения.

1.3 Метрики нейронной сети

Средняя абсолютная ошибка (MAE) - это метрика регрессии, которая измеряет среднее абсолютное различие между предсказанными и действительными значениями. Она определяется формулой 3.

$$MSE = \frac{1}{n}\sum_{n=1}^{n}(y_r - y_p) \tag{3}$$

Также метрикой регрессии может служить MSE, которая выражается формулой 2, только в случае оценки работспособности нейросети теперь проверяются тестовые данные.

1.4 О футболе

Читателю, который не интересовался спортом, будут непонятны некоторые футбольные термины и понятия, поэтому введем их.

Футбол — самый популярный командный вид спорта, целью которого является забитие мяча в ворота соперника любыми частями тела кроме рук большее количество раз, чем это сделала команада соперника. Футбольная встреча (матч) состоит из двух частей (таймов), каждая из которых длится по 45 минут. Между таймами есть перерыв.

Матчи могут играться на разных стадионах. Если команада играет на своем стадионе, то говорят, что она играет дома, в противном случае - на выезде или же в гостях.

Ла Лига (La Liga) или же Испанский футбольный чемпионат — испанская футбольная лига, которая является высшей в системе футбольных лиг Испании. Каждый год в ней принимает участие 20 футбольных команад, 3 из которых (занявшие последние места) опускаются в дивизион ниже - Сегунду, а занявшие с первого по третьего места команды Сегунды в следующем сезоне принимают участие в Ла лиге. Чемпионат проходит следующим образом: каждая команда играет с каждой дважды: один раз у себя дома, второй - на выезде.

2 Практическая часть

В данной части будет описан процесс решения поставленной задачи. Для этого был взят датасет с результатами матчей Ла лиги с ресурса Kaggle, а именно отсюда, благодаря переходу во вкладку Datasets и последующему поиску "La liga".

Анализ данных и обучение нейросетей удобнее всего делать на языке Python в среде Google Colaboratory (Colab), где мы и будем работать в дальнейшем.

2.1 Обработка данных

Загружаем скачанный датасет в Google Disk, а потом подключаем его к нашему блокноту в Google Colaboratory.

Для начала необходимо импортировать следующие библиотеки: pandas, numpy и matplotlib.pyplot, позволяющие выполнять анализ данных, машинное обучение, отрисовку графиков и прочее.

Также необходимо импортировать LabelEncoder из sklearn.preprocessing для преобразования категориальных переменных в числовые, а Sequential,Dropout, Normalization, Adam, layers, regularizers из tensorflow.keras для разработки нейросети.

После подключения всех необходимых библиотек и методов необходимо прочитать датасет с помощью pd.read_csv(). Благодаря методу .head() понятно, каким образом выглядят первые 5 строк нашей базы данных.

	Season	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR
0	1995-96	02-09-1995	La Coruna	Valencia	3	0	H	2.0	0.0	H
1	1995-96	02-09-1995	Sp Gijon	Albacete	3	0	H	3.0	0.0	H
2	1995-96	03-09-1995	Ath Bilbao	Santander	4	0	H	2.0	0.0	H
3	1995-96	03-09-1995	Ath Madrid	Sociedad	4	1	H	1.0	1.0	D
4	1995-96	03-09-1995	Celta	Compostela	0	1	A	0.0	0.0	D

Рис. 1: Первые 5 строк исходного датасета

1. Season - сезон проведения чемпионата

2. Date - дата матча

3. HomeTeam - команда, которая принимает матч (на ее стадионе проходит матч между командами)

4. AwayTeam - команда, которая играет в гостях (на выезде)

5. FTHG - количество голов, забитое в матче командой, играющей дома

6. FTAG - количество голов, забитое в матче командой, играющей в гостях

7. FTR - победитель матча

8. HTHG - количество голов, забитое в первом тайме командой, играющей дома

9. HTAG - количество голов, забитое в первом тайме командой, играющей в гостях

10. HTR - победитель первого тайма

Видно, что в базе данных результаты футбольных матчей Ла Лиги представлены с сезона 1995/1996. Очевидно, что такие игры никак не помогут предсказать результат матча, так как с того времени поменялось много чего: изменились правила игры, прошло несколько поколений футболистов и много чего другого. Никак нельзя оценить готовность команды к матчу по ее играм 30-летней давности, поэтому принято решение работать с играми последних 5 лет, используя метод .drop(). Первые 5 строк измененного датасета стали выглядеть следуюшим образом.

	Season	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR
9284	2019-20	16-08-2019	Ath Bilbao	Barcelona	1	0	H	0.0	0.0	D
9285	2019-20	17-08-2019	Celta	Real Madrid	1	3	A	0.0	1.0	A
9286	2019-20	17-08-2019	Valencia	Sociedad	1	1	D	0.0	0.0	D
9287	2019-20	17-08-2019	Mallorca	Eibar	2	1	H	1.0	0.0	H
9288	2019-20	17-08-2019	Leganes	Osasuna	0	1	A	0.0	0.0	D

Рис. 2: Первые 5 строк измененного датасета

Благодаря длине уникальных значений команд, а также отсортированному количеству матчей понятно, что всего в лиге принимало участие с сезона 2019/2020 27 команд, 7 из которых провели меньше 100 игр в чемпионате. Исходя из этого, было принято решение исключить матчи с этими командами, оставив лишь матчи с 20 командами, которые достаточно принимали участие в Ла лиге.

Является правильным решение убрать столебц FTR (говорящий о победителе первого тайма), так как больше информации нам даст количество забитых голов.

Метод .info помогает разобраться с типом данных столбцов датасета

Столбец	Season	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG
Тип данных	object	object	object	object	int64	int64	object	float64	float64

Таблица 1: Тип данных столбцов датасета.

Нейронная сеть работает с числовыми данными, поэтому все категориальные переменные необходимо первести в числовые с помощью LabelEncoder().

	Season	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG
9284	0	246	1	3	1	0	2	0.0	0.0
9285	0	259	6	14	1	3	0	0.0	1.0
9286	0	259	17	16	1	1	1	0.0	0.0
9289	0	259	19	10	4	4	1	1.0	1.0
9290	0	275	0	11	1	0	2	0.0	0.0
...	...	...	...	...	...	...	...	...	...
10874	4	495	9	19	0	0	1	0.0	0.0
10877	4	495	16	1	3	0	2	1.0	0.0
10879	4	15	0	13	0	2	0	0.0	1.0
10880	4	15	2	5	3	2	2	1.0	2.0
10881	4	15	4	17	3	0	2	1.0	0.0

Рис. 3: Итоговый вид датасета

2.2 Создание нейросети

Перед созданием нейросети необходимо разделить данные на тренировочную и тестовую выборки. Тестовая выборка получается применением метода .sample() к датасету, указанная тестовая выборка равна 85% от общего количества данных. Тестовая выборка получается применением метода .drop(), который исключает тренировочную выборку из датасета. Будет предсказан победитель футбольного матча, поэтому необходимо сделать FTR целевой переменной.

Sequential - это модель, в которой слои располагаются последовательно один за другим, она позволяет сделать многослойный персептрон. Во входном слое располагается нормализация с 300 нейронами. Было решено сделать три скрытых слоя нейронов, в которых распределение нейронов следующее: 1200, 2500, 150. В выходном слое присутствует один нейрон, предсказывающий победителя футбольного матча. Функция активации ReLu позволяет связать входной слой - 1 скрытый слой - 2 скрытый слой - 3 скрытый слой, а выходной слой из одного нейрона связан с последним слоем с помощью линейной активации. Функцией ошибок выбираем MSE, оптмизатор - Adam, а метрику MAE, как и оговаривалось в теоретической части.В скрытые и входной слои добавим L2-регуляризацию , а также применим к ним Dropout для того, чтобы модель не была переобучена.

2.3 Обучение нейросети

Благодаря методу .fit() решено начать обучение модели на выбранных тренировочных данных на 200 эпохах обучения. Для недопущения переобучения выбрана валидационная выборка, составляющая 10% от тестовой выборки.

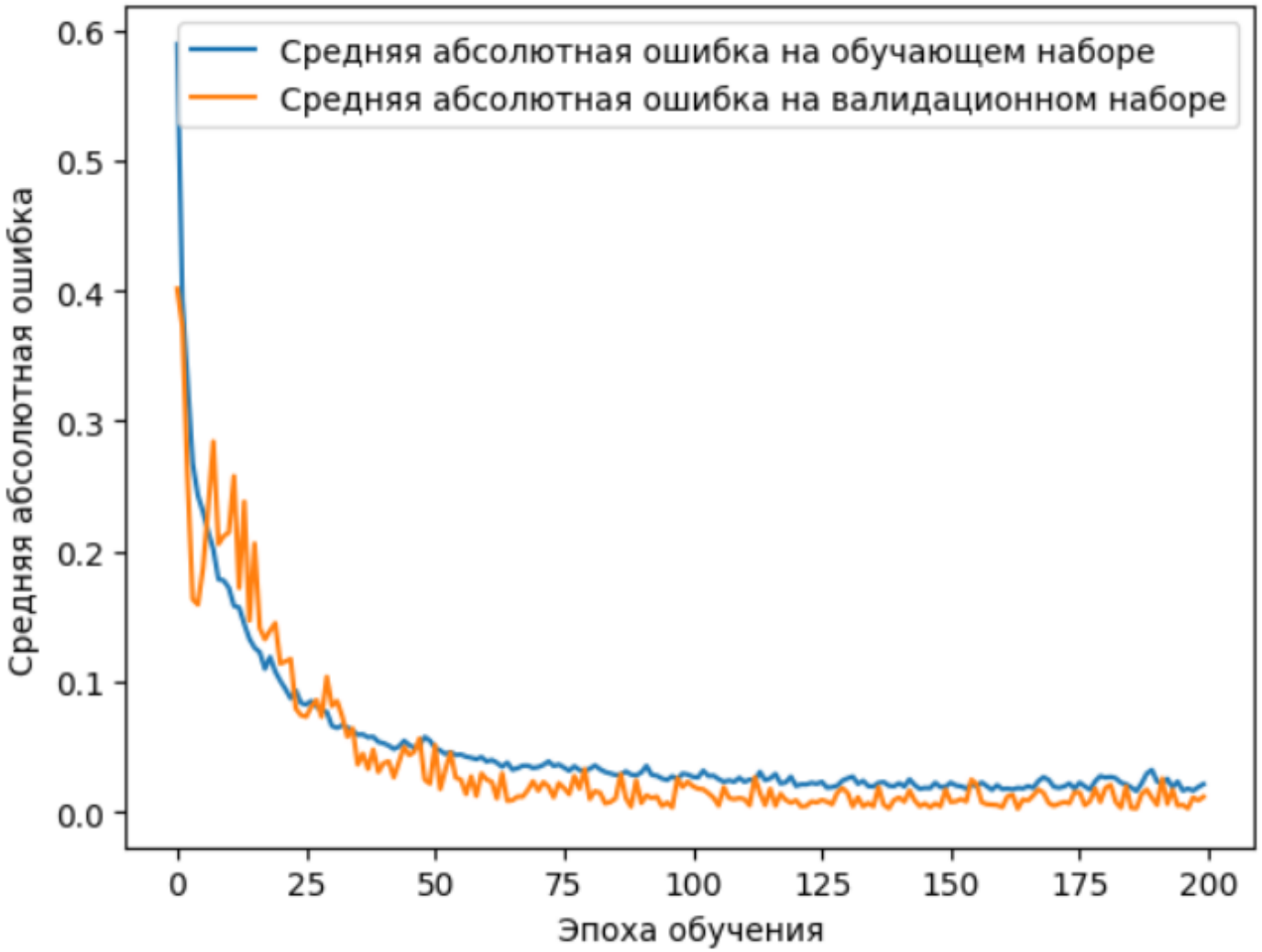

Рис. 4: Результат обучения нейронной сети

Видно, что ошибка после 75 эпохи сильно не изменяется, поэтому увеличивать количество эпох не имеет смысла. Ошибка в - 0.0094 на последней эпохе обучения

достаточно низкая, так как отклонение от следующих значений 0 - победа команды, играющей дома, 1 - ничейный результат, 2 - победа команды, играющей в гостях

2.4 Результат работы нейросети на тестовой выборке

MAE = 0.008799134753644466, что говорит о достаточно высокой точности нашей нейросети. Следовательно, все задачи нашего проекта выполнены.

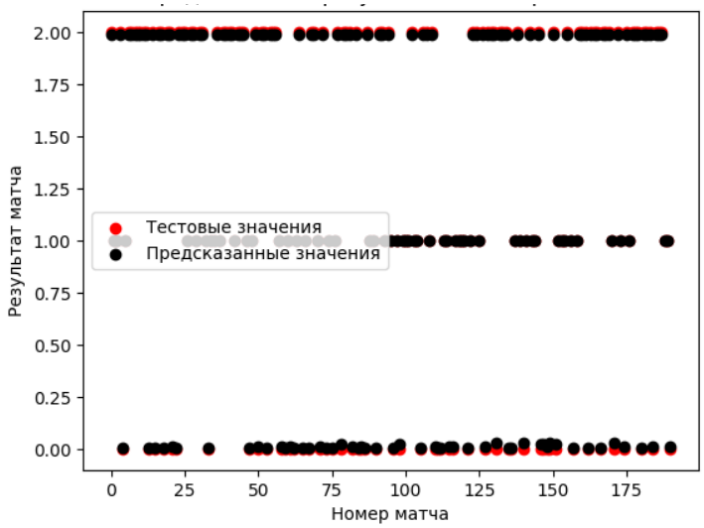

Рис. 5: Предсказанный результат матча и реальный

2.5 Вывод

По результатам работы нейросети понятно, что удалось выполнить цель проекта - разработка нейросети, предсказывающая результат футбольного матча по результатам первого тайма оказалась успешной.

В процессе выполнения проекта удалось узнать о работе нейросетей, структуре многоуровненого персептрона, функции активации ReLu, функции потерь MSE, методе обратного распространения ошибки, оптимизаторе Adam, методах борьбы с переобучением (Dropout + Регуляризация), метриках оценки работы нейросетей.

Данный проект позволил погрузиться машинное обучение и больше узнать о задачах регрессии, что, очевидно, позволяет улучшить знания об анализе данных

3 Литература

1. Документация Keras -https://keras.io/api/layers/core$_l ayers$/

2. Документация Pandas -https://pandas.pydata.org/docs/

3. Статья Mixture Density Networks на Хабр -https://habr.com/ru/articles/433804/

4. Статья Нейронная сеть на Wikipedia - https://ru.wikipedia.org/wiki/Нейронная