Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизации обработки информации (АОИ)

Отчёт к Лабораторной работе №4

по дисциплине «Нейронные сети и их применение»

Выполнил:

Студент группы 422–M1

\_\_\_\_Белоус Г.В.

Принял:

К.т.н., Доцент кафедры АОИ

\_\_\_\_Аксёнов С.В.

Оглавление

[1 Введение 3](#_Toc131164226)

[1.1 Цель лабораторной работы 3](#_Toc131164227)

[1.2 Задание на лабораторную работу 3](#_Toc131164228)

[2 Ход выполнения работы 5](#_Toc131164229)

[2.1 Обучение четырех регрессионных модели, построенных на наборе данных с помощью четырёх алгоритмов 5](#_Toc131164230)

[2.2 Исследовать работы алгоритмов с разными значениями гиперпараметров 8](#_Toc131164231)

[2.3 Реализация приложения осуществляющее загрузку модели 9](#_Toc131164232)

[3 Вывод 10](#_Toc131164233)

# Введение

## 1.1 Цель лабораторной работы

Получение навыков создания полносвязного бинарного и многоклассового классификаторов, а также нейросетевого регрессора с помощью библиотеки построения нейронных сетей.

## **1.2 Задание на лабораторную работу**

Построить три нейросетевые модели:

a) Бинарный Классификатор

b) Многоклассовый классификатор

c) Регрессор

По выборкам:

a) Оценка вероятности, того, что клиент откроет банковский депозит в результате маркетинговой акции: https://archive.ics.uci.edu/ml/datasets/Bank+Marketing Класс: атрибут 21 - y - has the client subscribed a term deposit? (binary: 'yes', 'no').

b) Оценка здоровья внутриутробного развития плода: https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification Метка класса: fetal\_health.

c) Оценка уровня потребления ресурсов домохозяйством. Набор данных DS\_2019\_public.csv. Описание атрибутов – файл recs2009\_public\_codebook.xlsx. Предсказываемый признак – один из атрибутов (использовать только один из): TOTALBTU, TOTALBTUSPH, TOTALBTUCOL, TOTALBTUWTH, TOTALBTURFG, TOTALBTUOTH.

Выполнить загрузку и предварительную обработку данных из наборов. Разделить каждую выборку на обучающую, тестовую и валидационную. Произвести обучение набора нейросетевых архитектур, отличающихся разным набором параметров: число слоёв, количество нейронов в слоях, функции активации в слоях, процедур оптимизации:

- Подобрать архитектуры нейронных сетей, которые с одной стороны позволяют получить модели с лучшими метриками качества работы, с другой стороны не являются избыточными и не переобученными.

- Вычислить следующие метрики работы:

Для бинарного классификатора: Recall, Precision, Weighted Accuracy, AUC для всех исследованных моделей.

Для многоклассового классификатора: Recall, Precision, Weighted Accuracy, AUC для всех классов всех исследованных моделей. Вывести ROC-кривые для каждого класса в лучшем классификаторе.

Для регрессора: MSE, MAE, R2 для всех полученных моделей.

Сделать выводы по результатам построения моделей.

Содержание отчета.

1. Описание наборов данных.

2. Признаки, которые были использованы для анализа.

2. Параметры архитектур и обучения нейронных сетей, использованные для обучения.

3. Графики обучения для архитектур нейронных сетей с лучшими характеристиками эффективности

5. ROC-кривые классов для лучших классификаторов.

6. Оценки моделей на тестовых выборках в виде таблиц/ диаграмм, отображающих метрики качества.

7. Программный код.

# Ход выполнения работы

Датасеты, данные из который будут обрабатываться в данной работе, представляет из себя: a) набор данных из 45211 записей с 17 признаками как численных(int), так и категориальных строковых(string); b) набор данных из 2126 записей с 22 численными признаками(int); c) набор данных из 10875 записей с 117 численными признаками(int, float).

Перед созданием моделей наборы данных были проверен на отсутствие пропущенных записей, в наборе данных c) были удалены четыре столбца [30, 56, 96, 114], в которых были данные смешанного типа. Данные во всех датасетах были нормализированы в a) необходимо было нормализовать строковые значения, а в b) целевой столбец был переделан под категориальный с помощью метода to\_categorical().

На обработанных данных был использован метод train\_test\_split, разделяющий всю выборку на матрицу признаков для тренировки модели и вектор целевого значения quality (train), матрицу признаков и вектор предсказываемых значений для проверки моделей (test), а так же вариационную выборку необходимую для обучения многослойной нейронной сети (val).

## Создание модели бинарного классификатора

Бинарный классификатор представляет из себя нейронную сеть из трех слоев:

* Входной слой с шестнадцатью нейронами, каждый из которых принимает по столбцу из датасета a);
* Скрытый слой состоящий из восьми нейронов и функцией активации ReLU (rectified linear unit), количество в восемь нейронов было выбрано исходя из рекомендации брать среднее значение между количеством входных и выходных нейронов;
* Выходной слой состоит из одного нейрона и функцией активации sigmoid, которая приводит выход модели к диапазону от 0 до 1, что позволяет интерпретировать ее выход как вероятность принадлежности к положительному классу.

Оптимизатор был выбран adam, так как данный оптимизатор показывает более гладкий график, в отличии от SGD и RMSprop.

Для обучения модели были определены значения эпох (150) и размер выборки (10).

В итоге модель была обучена за 87 эпох, график функции ошибки (binary\_crossentropy) от количества эпох обучения показан на рисунке 2.1:

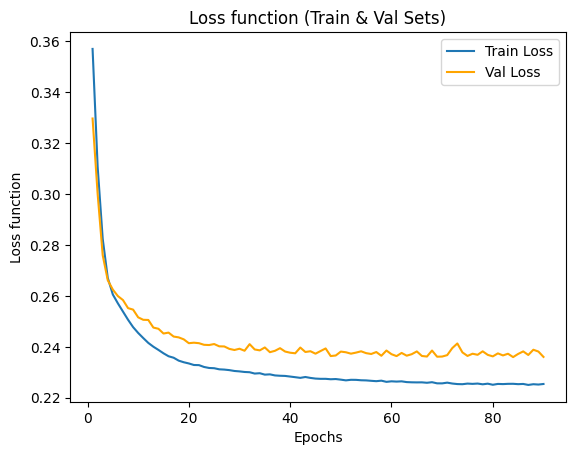


Рисунок 2.1 – График величины функции ошибки от количества эпох при обучении модели

На рисунке 2.2 показан график точности модели от количества эпох обучения:

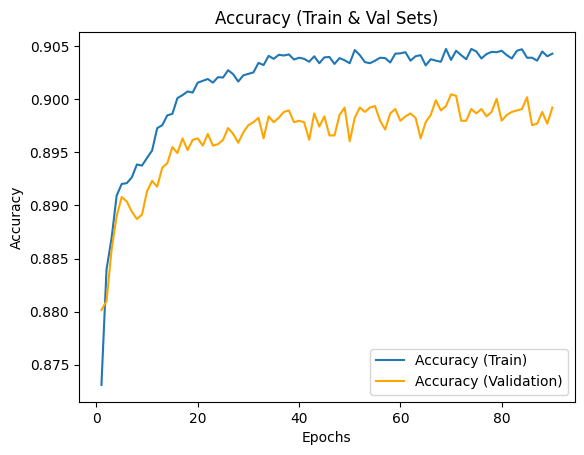


Рисунок 2.2 – График точности модели от количества эпох при обучении модели

## Создание модели многоклассового классификатора

Многоклассовый классификатор представляет из себя нейронную сеть из трех слоев:

* Входной слой с двадцати одним нейронами, каждый из которых принимает по признаку из датасета a);
* Скрытый слой состоящий из восьми нейронов и функцией активации ReLU (rectified linear unit);
* Выходной слой состоит из трех нейронов и функцией активации softmax, каждый нейрон показывает вероятность принадлежности к одному из трёх классов целевого признака.

Оптимизатор был выбран adam, так как данный оптимизатор показывает более гладкий график, в отличии от SGD и RMSprop.

Для обучения модели были определены значения эпох (50) и размер выборки (10).

В итоге модель была обучена за 50 эпох, график функции ошибки (categorical\_crossentropy) от количества эпох обучения показан на рисунке 2.3:

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 2.3 – График величины функции ошибки от количества эпох при обучении модели

На рисунке 2.4 показан график точности модели от количества эпох обучения:

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 2.4 – График точности модели от количества эпох при обучении модели

На рисунке 2.5 продемонстрированы ROC-кривые для каждого класса модели:

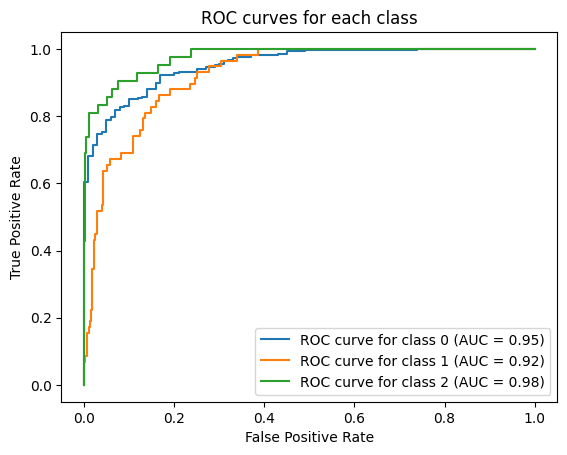


Рисунок 2.5 – ROC-кривые для каждого класса

Наибольшую площадь под кривой демонстрирует класс 2.

## Создание модели регрессора

В качестве целевого атрибута был выбран столбец TOTALBTUCOL.

Регрессор представляет из себя нейронную сеть из трех слоев:

* Входной слой со сто двенадцатью нейронами, каждый из которых принимает по определенному столбцу из датасета a);
* Скрытый слой состоящий из двухсот двадцати четырех нейронов и функцией активации ReLU (rectified linear unit);
* Выходной слой состоит из одного нейрона, так как в результате предсказания регрессионной модели необходимо получить значение TOTALBTUCOL.

Оптимизатор был выбран adam, так как данный оптимизатор показывает более гладкий график, в отличии от SGD и RMSprop, как функция потери была взята средне квадратическая ошибка, а в качестве метрики обучения средне абсолютная.

Для обучения модели были определены значения эпох (350) и размер выборки (40).

В итоге модель была обучена за триста пятьдесят эпох, график функции ошибки (MSE) от количества эпох обучения показан на рисунке 2.6:

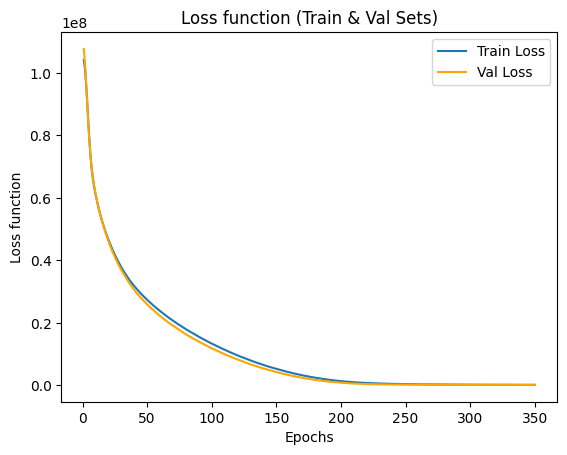


Рисунок 2.6 – График величины функции ошибки от количества эпох при обучении модели

На рисунке 2.6 показан график средней абсолютной ошибки модели от количества эпох обучения:

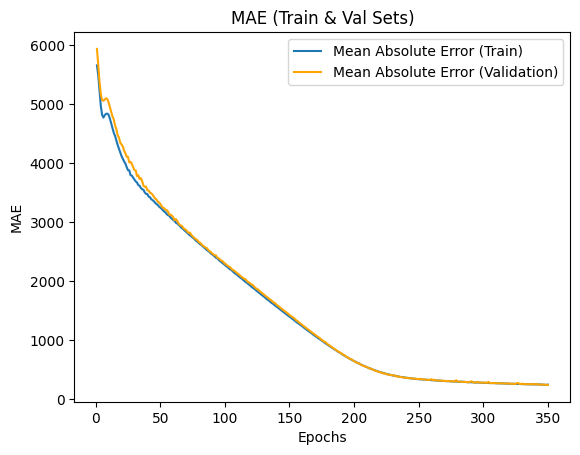


Рисунок 2.6 – График средней абсолютной ошибки модели от количества эпох при обучении модели

## Итоговые метрика каждой модели

В таблице 2.1 собраны метрики всех моделей полученные с помощью библиотеки sklearn.metrics.

Таблица 2.1 – Метрики качества моделей

|  |  |  |  |
| --- | --- | --- | --- |
| Метрика\Модель | Бинарный классификатор | Многоклассовый классификатор | Регрессор |
| Recall | 0: 0.97  1: 0.38 | 0: 0.94  1: 0.66  2: 0.79 | - |
| Precision | 0: 0.92  1: 0.59 | 0: 0.93  1: 0.63  2: 0.85 | - |
| Weighted Accuracy | 0.89 | 0.88 | - |
| AUC | 0.89 | 0: 0.95  1: 0.92  2: 0.98 | - |
| MSE | - | - | 236155.54 |
| MAE | - | - | 247.31 |
| R2 | - | - | 0.9965 |

# Вывод

В ходе выполнения лабораторной работы были построены модели полносвязного бинарного и многоклассового классификаторов, а также нейросетевого регрессора с помощью библиотеки построения нейронных сетей.

Стоит отметить, что добиться высокой точности классификации в случае с набором данных a) не получилось, несмотря на попытку изменять количество нейронов, размера выборки и количества эпох, это может быть связано с малым процентом от общего количества второго класса, что и показали метрики в таблице 2.1, то же самое можно сказать и для модели многоклассового классификатора. Как видно из картинок 2.2 и 2.4, точность модели во время обучения не может удержаться на одном уровне и не поднимается выше 0.9, хотя набор данных a) имеет больший размер в сравнении с датасетом b). Так на графике функции потерь 2.1 у бинарного классификатора виден быстрый расход значения для train и val выборки, что обозначает переобучение модели.

При этом регрессор показал лучший результат в сравнении с классификаторами, если судить по таблице 2.1, предсказания данной модели показывают маленькое отклонение от реальных значений. Их рисунка 2.6 можно увидеть некий скачок в значении MAE на 20 эпохах, так же видно, что после 230 эпох абсолютная ошибка уменьшается не так стремительно, как до этого.

Листинг программы находится в соответствующей папке Lab4 по ссылке на репозиторий: https://github.com/peremichka256/NSiIP