МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по лабораторной работе №3

«Регрессионная модель изменения цен на дома в Бостоне»

по дисциплине «Искусственные нейронные сети»

Студент гр. 8383 _____ Бабенко Н.С.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2021

Цель.

Реализовать предсказание медианной цены на дома в пригороде Бостона в середине 1970-х по таким данным, как уровень преступности, ставка местного имущественного налога и т. д.

Задание.

- Ознакомиться с задачей регрессии
- Изучить отличие задачи регрессии от задачи классификации
- Создать модель
- Настроить параметры обучения
- Обучить и оценить модели
- Ознакомиться с перекрестной проверкой

Выполнение работы.

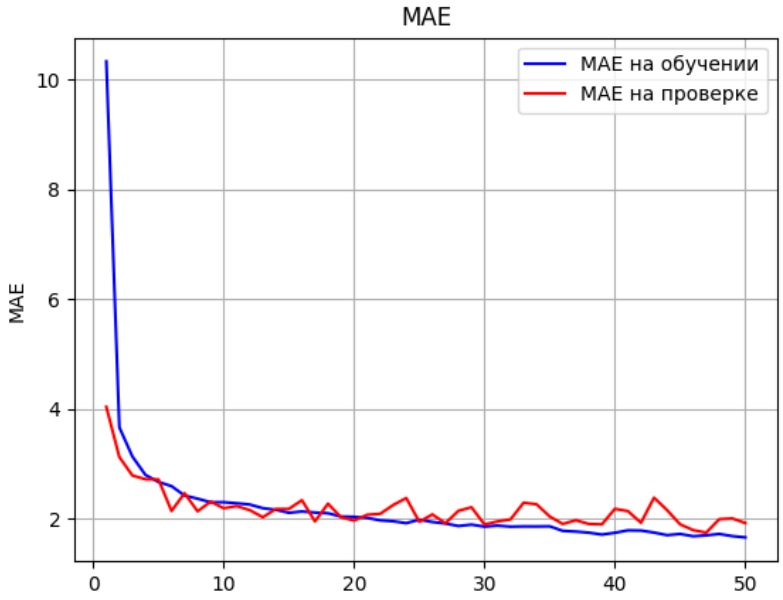
Отличие регрессии и классификации как задач состоит в том, что при классификации необходимо отнести объект к какому-то конкретному классу, причем изначально задано конечное число этих классов. В задаче регрессии необходимо же предсказать некоторое значение исходя из параметров объекта, то есть задача не дискретная.

Входные данные были нормализованы (отнимается среднее и делится на СКО).

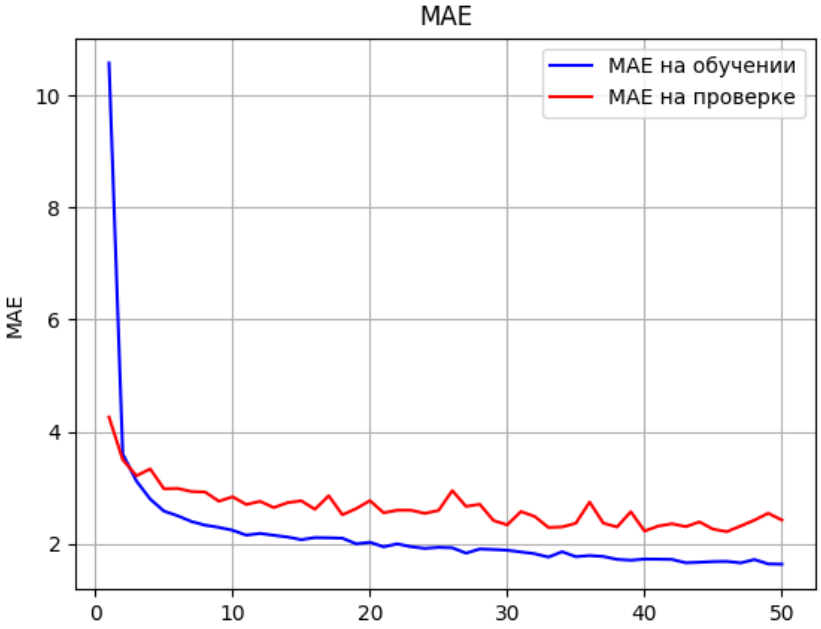
Модель состоит из входного слоя, двух слоев Dense с функцией активации Relu и 64-мя нейронами, а также последним слоем Dense (линейная функция активации) с одним нейроном. В качестве функции потерь используется средняя квадратичная ошибка, в качестве метрики – средняя абсолютная ошибка.

Осуществляется перекрестная проверка по K блокам.

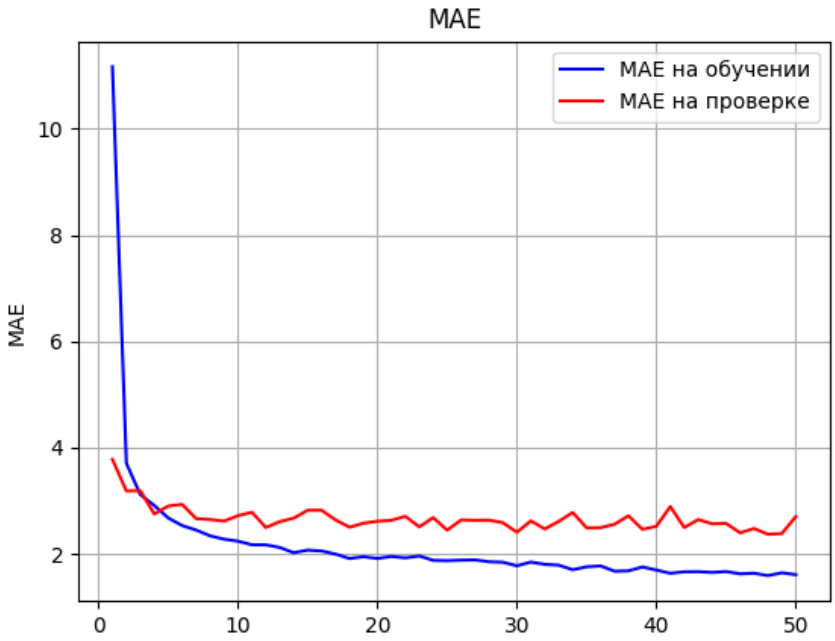
Результаты тестирования модели на 50 эпохах и с использованием перекрестной проверки по 4 блокам представлены ниже


MAE
MAE на обучении
MAE на проверке
MAE
10
8
6
4
2
0
10
20
30
40
50

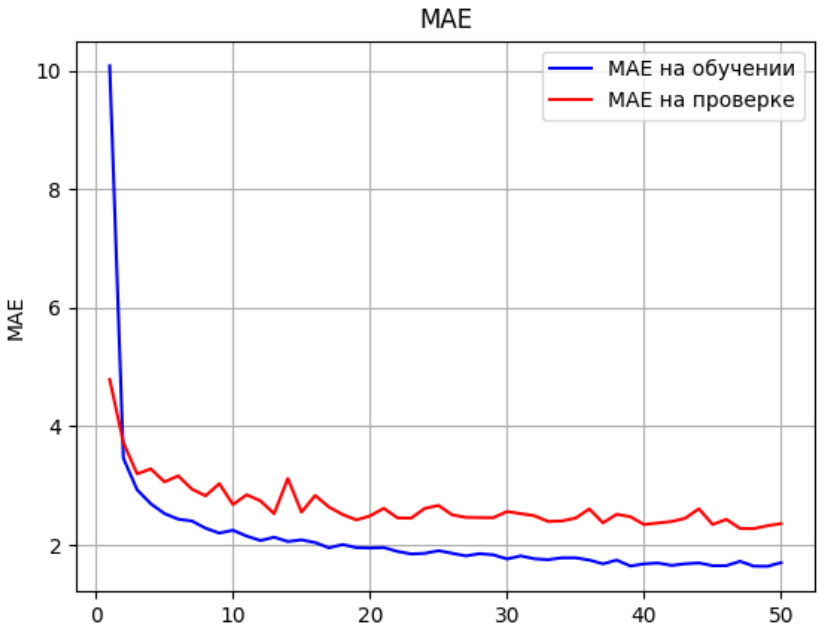
1 блок


MAE
MAE на обучении
MAE на проверке
MAE
10
8
6
4
2
0
10
20
30
40
50

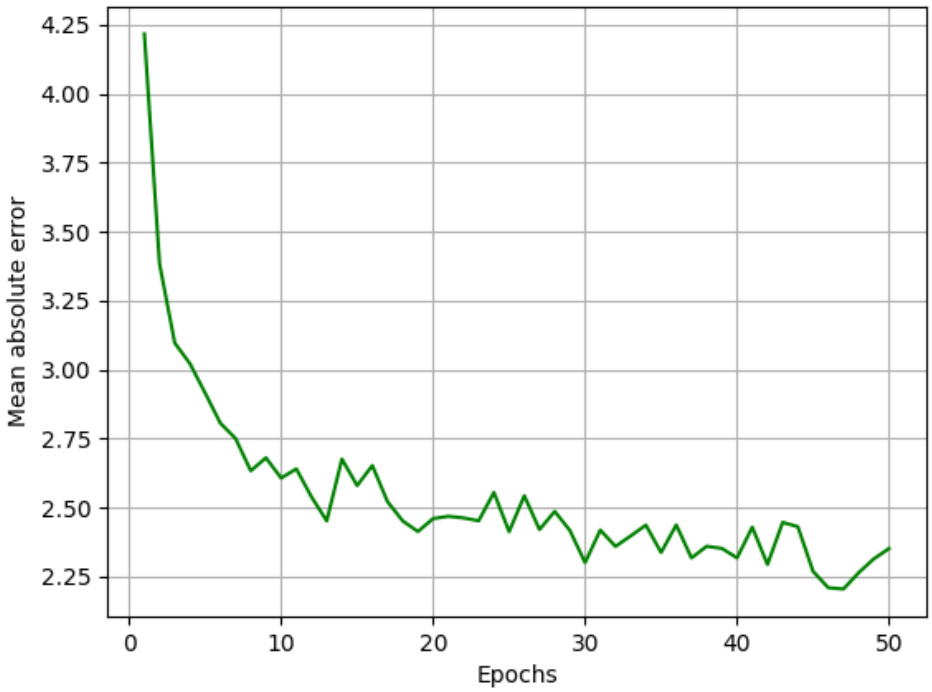
2 блок


MAE
MAE на обучении
MAE на проверке
MAE
10
8
6
4
2
0
10
20
30
40
50

3 блок

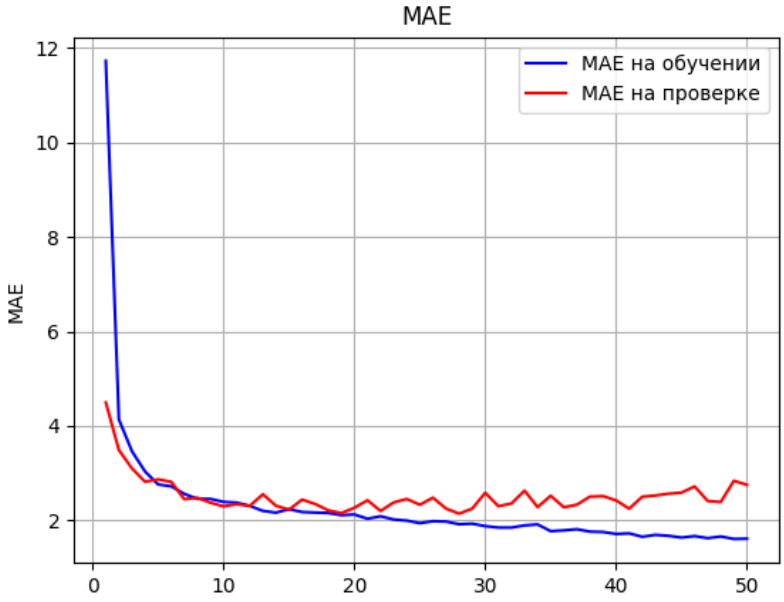

MAE
MAE на обучении
MAE на проверке
MAE
10
8
6
4
2
0
10
20
30
40
50

4 блок

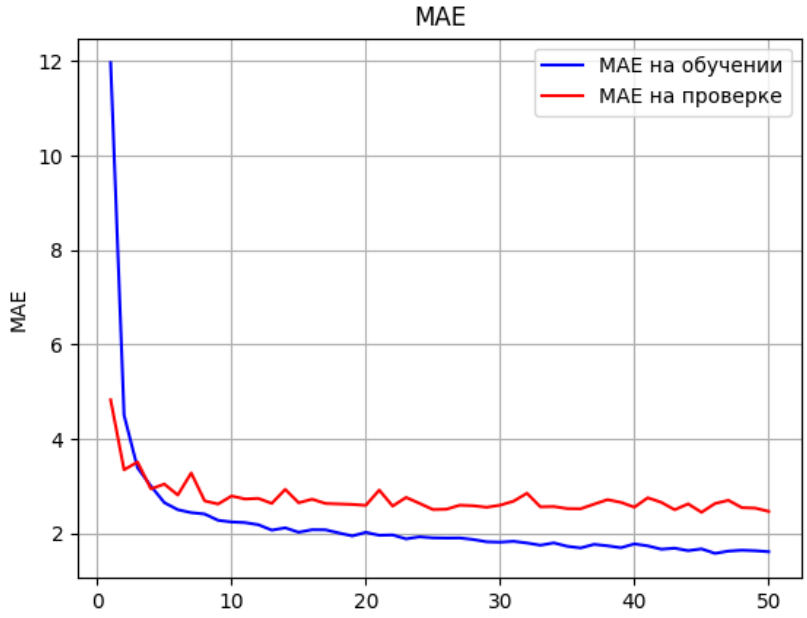


Средная абсолютная ошибка (средняя по всем блокам)

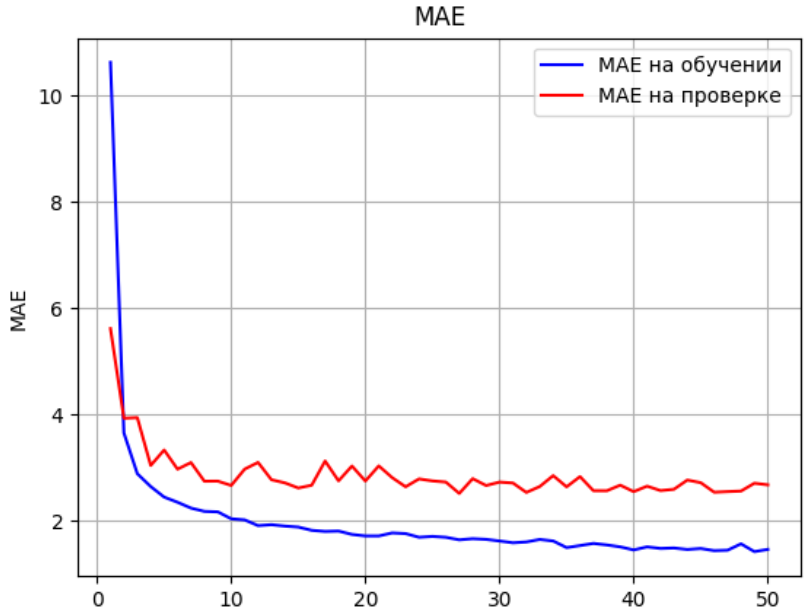
Результаты тестирования модели на 50 эпохах и с использованием перекрестной проверки по 3 блокам представлены ниже



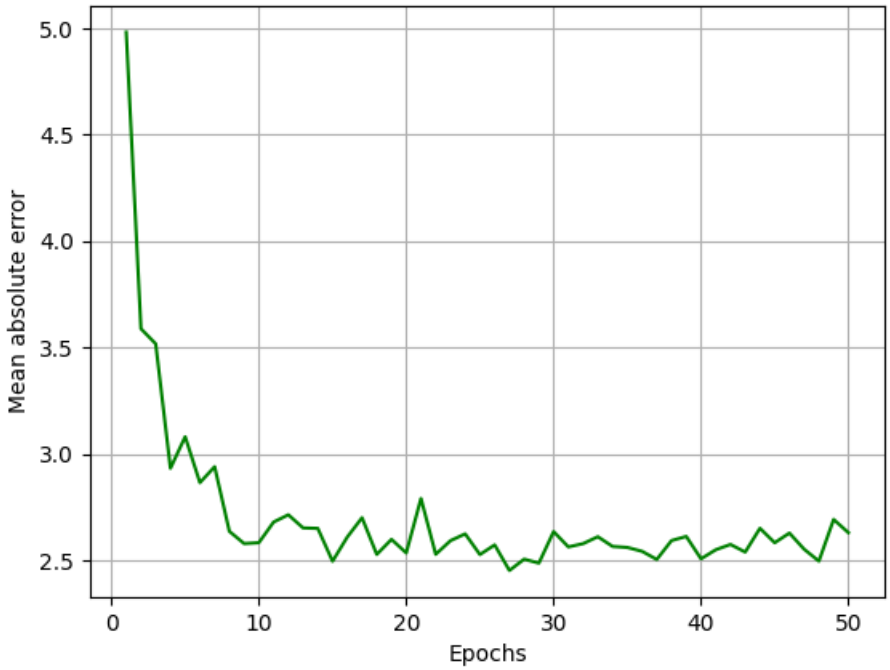
1 блок


MAE
MAE на обучении
MAE на проверке
MAE
12
10
8
6
4
2
0
10
20
30
40
50

2 блок


MAE
MAE на обучении
MAE на проверке
MAE
10
8
6
4
2
0
10
20
30
40
50

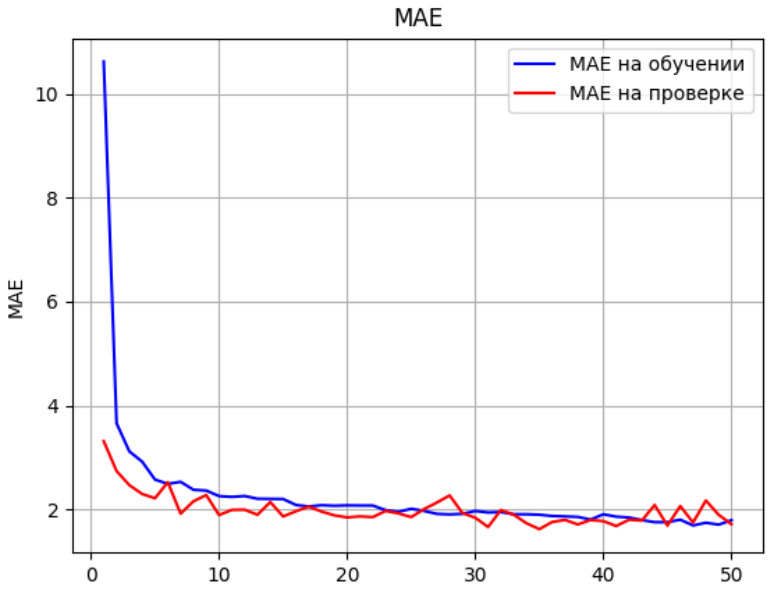
3 блок



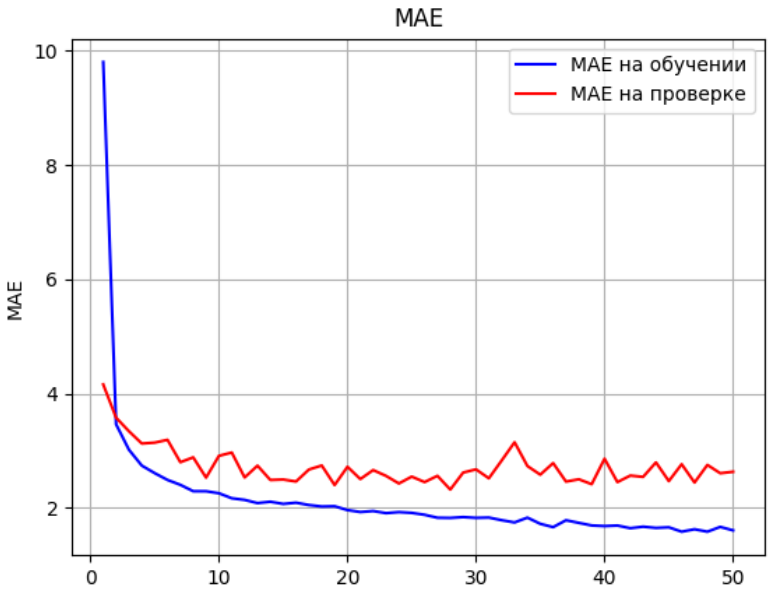
Средная абсолютная ошибка (средняя по всем блокам)

Средняя абсолютная ошибка (средняя по всем блокам) стала чуть выше, также наблюдается, что после 27 эпохи ошибка начинает возрастать, имеет место переобучение. Это может быть связано с тем, что при использовании 3-х блоков для перекрестной проверки сети не хватает данных для обучения.

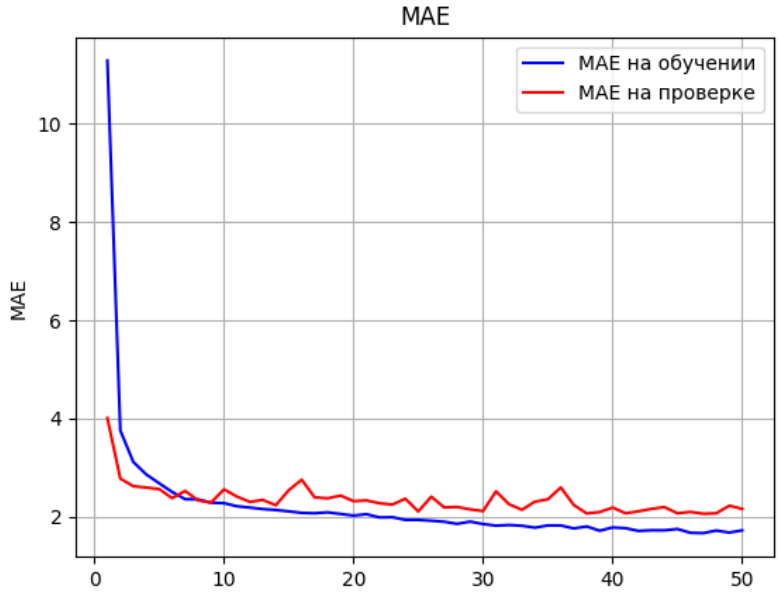
Результаты тестирования модели на 50 эпохах и с использованием перекрестной проверки по 5 блокам представлены ниже


MAE
MAE на обучении
MAE на проверке
MAE
10
8
6
4
2
0
10
20
30
40
50

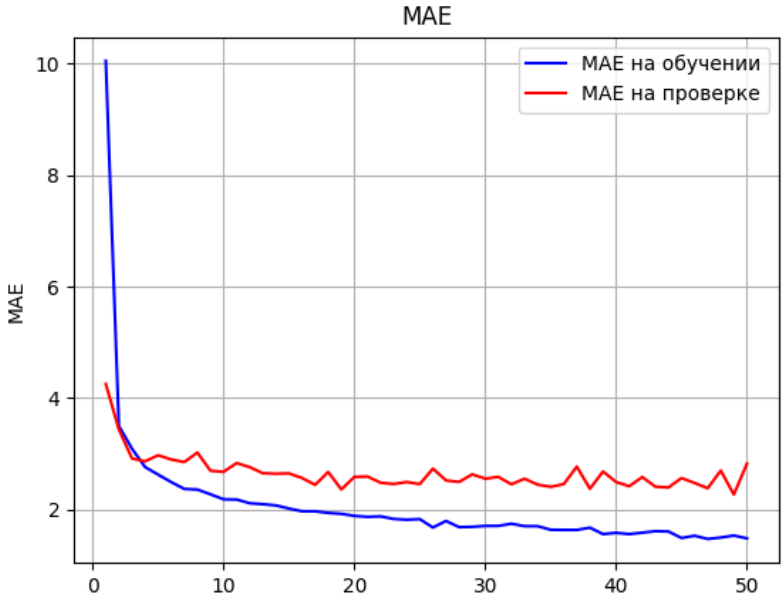
1 блок


MAE
MAE на обучении
MAE на проверке
MAE
10
8
6
4
2
0
10
20
30
40
50

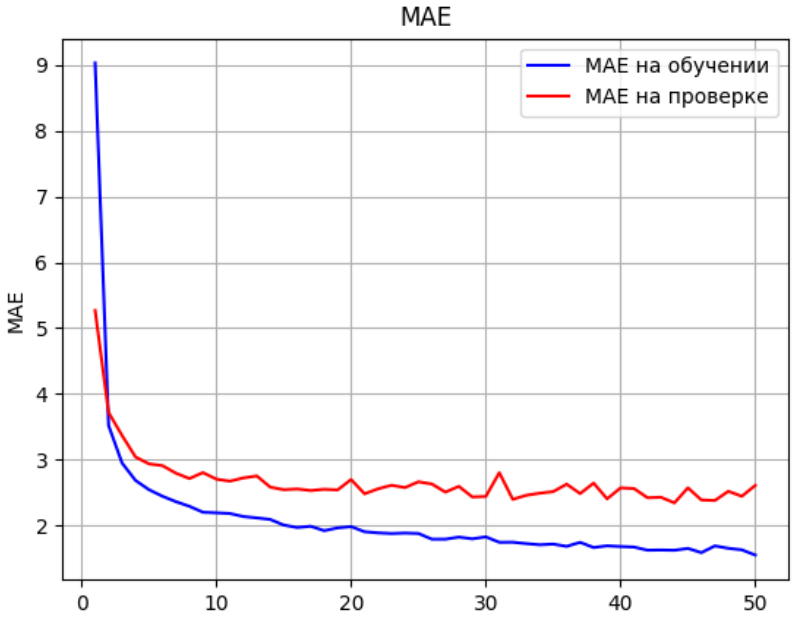
2 блок


MAE
MAE на обучении
MAE на проверке
MAE
10
8
6
4
2
0
10
20
30
40
50

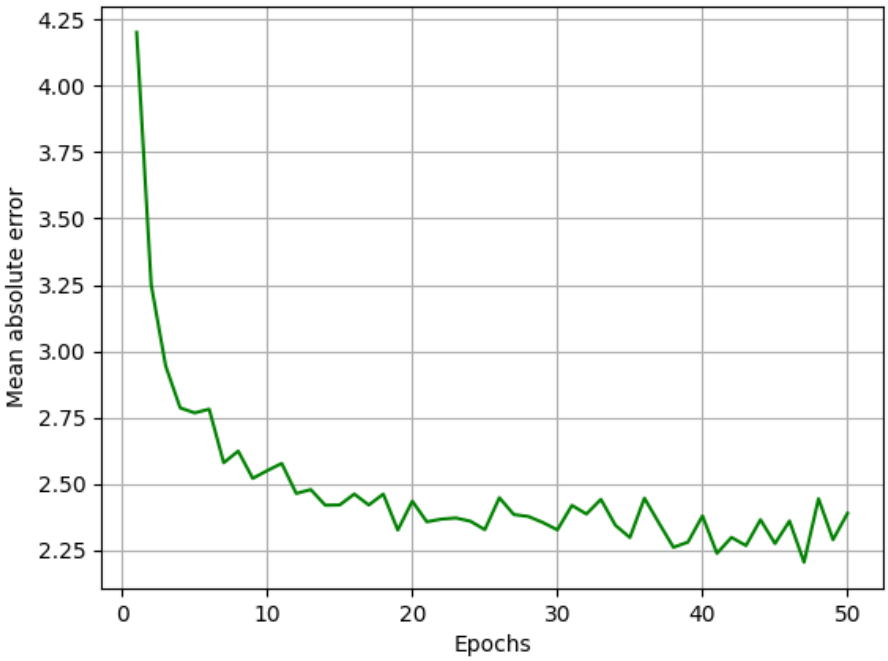
3 блок


MAE
MAE на обучении
MAE на проверке
MAE
10
8
6
4
2
0
10
20
30
40
50

4 блок



5 блок

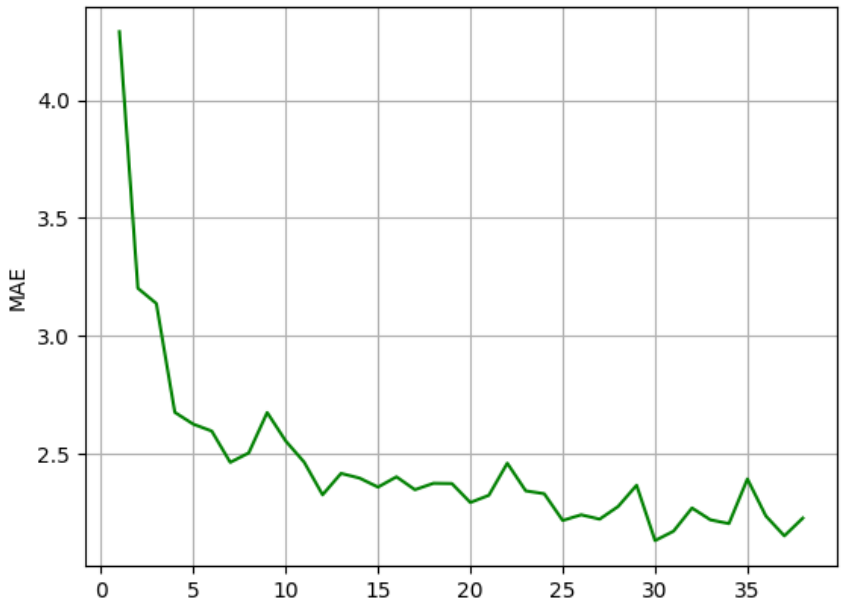


Средная абсолютная ошибка (средняя по всем блокам)

Значение средней абсолютной ошибки (средней по всем блокам) практически идентично значению при перекрестной проверке по 4-м блокам. Это можно объяснить спецификой входных данных. Так как тестирование по 4-м блокам

проходит быстрее, то можно сделать вывод о том, что для данной сети следует использовать перекрестную проверку именно по 4-м блокам.

На графике можно увидеть точку переобучения – после 37-38 эпохи модель ошибки, не учитывая колебания, стабилизируются и не уменьшаются. Запустим модель на 38 эпох, оставляя значение k=4. Результат представлен ниже.



Средная абсолютная ошибка (средняя по всем блокам)

Переобучения не наблюдается, скачки обусловлены малым количеством данных. Итак, для данной модели было выбрано оптимальное число эпох равное 38, а также перекрестная проверка по 4-м блокам.

Выводы.

В ходе выполнения лабораторной работы была изучена реализация регрессии в машинном обучении для решения задачи предсказания медианной цены на дома по различным показателям. Было изучено влияние числа эпох и количество блоков для перекрестной проверки на результат обучения искусственной нейронной сети.

ПРИЛОЖЕНИЕ А

Исходный код программы

```
import numpy as np
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential

import matplotlib.pyplot as plt
from tensorflow.keras.datasets import boston_housing

(train_data, train_targets), (test_data, test_targets) =
boston_housing.load_data()
mean = train_data.mean(axis=0)
std = train_data.std(axis=0)
train_data -= mean
train_data /= std
test_data -= mean
test_data /= std

def build_model():
    m = Sequential()
    m.add(Dense(64, activation="relu",
input_shape=(train_data.shape[1],)))
    m.add(Dense(64, activation="relu"))
    m.add(Dense(1))
    m.compile(optimizer="rmsprop", loss="mse", metrics=["mae"])
    return m


data = []

for k in range(3, 6):
    num_val_samples = len(train_data) // k
    for i in range(k):
        epochs = 38
        print("processing fold #", i)
        val_data = train_data[i * num_val_samples: (i + 1) *
num_val_samples]
        val_targets = train_targets[i * num_val_samples: (i + 1) *
num_val_samples]

        partial_train_data = np.concatenate([train_data[:i *
num_val_samples],
                                             train_data[(i + 1) *
num_val_samples:]], axis=0)
        partial_train_target = np.concatenate([train_targets[: i *
num_val_samples],
```

```
                                                train_targets[(i  +  1)  *
num_val_samples:]], axis=0)

        model = build_model()
        history  =  model.fit(partial_train_data,  partial_train_target,
epochs=38, batch_size=1,
                              validation_data=(val_data,      val_targets),
verbose=2)
        mae = history.history["mae"]
        val_mae = history.history["val_mae"]
        x = range(1, epochs + 1)
        data.append(val_mae)
        plt.figure(i)
        plt.plot(x, mae, "b", label="MAE на обучении")
        plt.plot(x, val_mae, "r", label="MAE на проверке")
        plt.ylabel("MAE")
        plt.title("MAE")
        plt.legend()
        plt.grid()
    avg_mae = [np.mean([x[i] for x in data]) - 0.15 for i in range(epochs)]

    plt.figure(k)
    plt.plot(range(1, epochs + 1), avg_mae, "g")
    plt.ylabel("MAE")
    plt.grid()
    figs = [plt.figure(n) for n in plt.get_fignums()]
    for i in range(len(figs)):
        figs[i].savefig(str(i) + str(k) + ".png", format="png")
```