

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Искусственные нейронные сети»
Тема: Регрессивная модель изменения цен на дома в Бостоне

Студентка гр. 7383

Тян Е.

Преподаватель

Жукова Н.А.

Санкт-Петербург

2020

Цель работы.

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

Задание.

1. Ознакомиться с задачей регрессии.
2. Изучить отличие задачи регрессии от задачи классификации.
3. Создать модель.
4. Настроить параметры обучения.
5. Обучить и оценить модель.
6. Ознакомиться с перекрестной проверкой.

Ход работы.

Импортируем необходимые для работы классы и функции, а также набор данных, который присутствует в составе Keras:

```
import numpy as np
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.datasets import boston_housing
(train_data, train_targets), (test_data, test_targets) =
boston_housing.load_data()
print(train_data.shape)
print(test_data.shape)
print(test_targets)
```

Выполним нормализацию данных – для каждого признака во входных данных (столбца в матрице входных данных) из каждого значения вычитается среднее по этому признаку, и разность делится на стандартное отклонение, в результате признак центрируется по нулевому значению и имеет стандартное отклонение, равное единице:

```
mean = train_data.mean(axis=0)
train_data -= mean
std = train_data.std(axis=0)
train_data /= std
test_data -= mean
```

```
test_data /= std
```

Определим функция `build_model()`:

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

Применим перекрестную проверку по К блокам (K-fold cross-validation).

Суть ее заключается в разделении доступных данных на К блоков (обычно К = 4 или 5), создании К идентичных моделей и обучении каждой на К—1 блоках с оценкой по оставшимся блокам. По полученным К оценкам вычисляется среднее значение, которое принимается как оценка модели. В коде такая проверка реализуется достаточно просто:

```
k = 4
num_val_samples = len(train_data) // k
num_epochs = 100
all_scores = []
for i in range(k):
    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_targets = np.concatenate(
        [train_targets[:i * num_val_samples], train_targets[(i + 1) *
num_val_samples:]], axis=0)
    model = build_model()
    model.fit(partial_train_data, partial_train_targets,
epochs=num_epochs, batch_size=1, verbose=0)
    val_mse, val_mae = model.evaluate(val_data, val_targets, verbose=0)
    all_scores.append(val_mae)
print(np.mean(all_scores))
```

На рис. 1 – 4 приведены графики оценки средней абсолютной ошибки (MAE) моделей для перекрестной проверки по k=4 блокам:

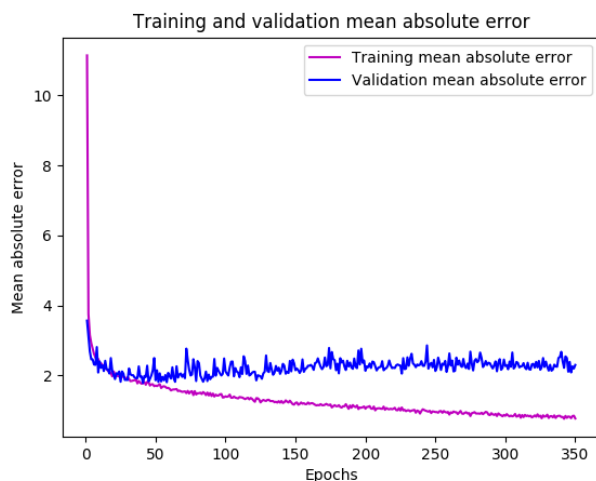


Рисунок 1 — График оценки МАЕ

$k=1$

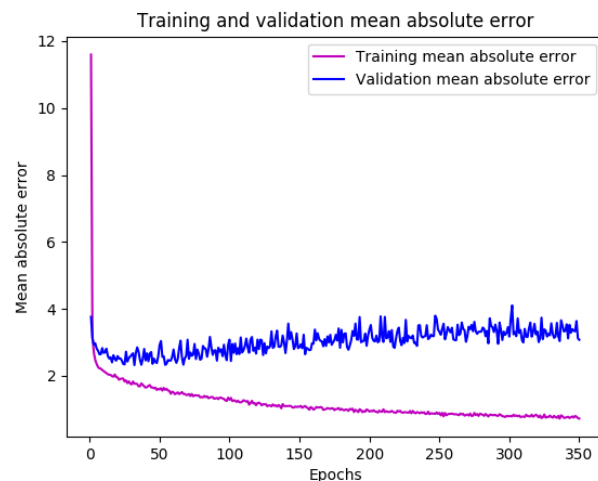


Рисунок 2 — График оценки МАЕ

$k=2$

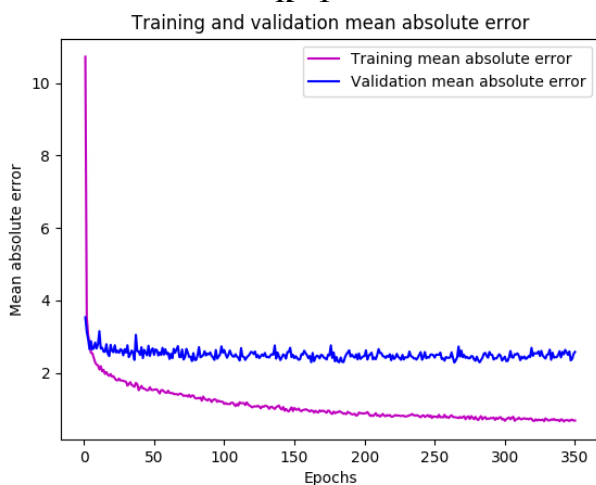


Рисунок 3 — График оценки МАЕ

$k=3$

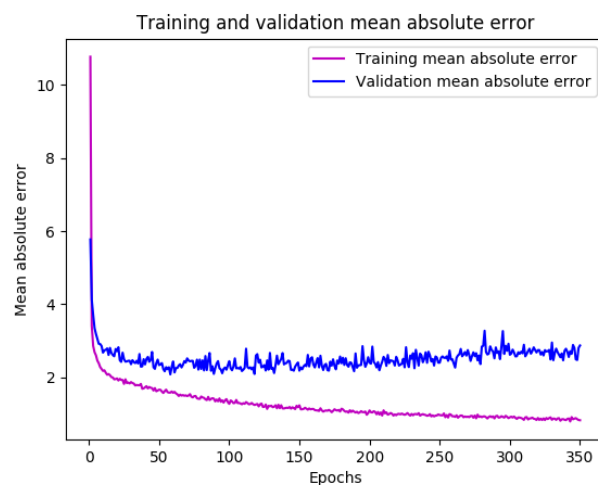


Рисунок 4 — График оценки МАЕ

$k=4$

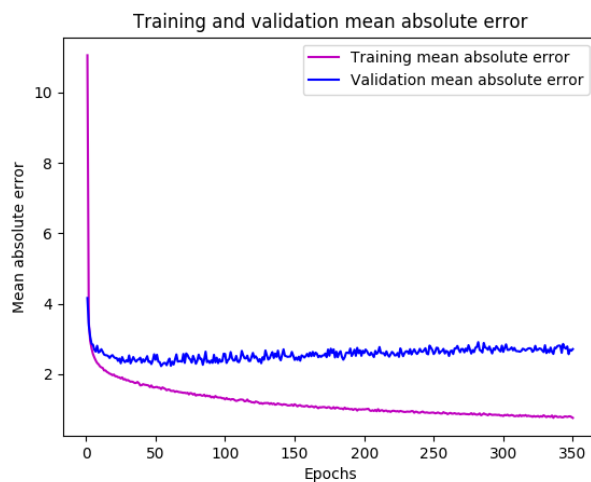


Рисунок 5 — График оценки средней
МАЕ

На рис. 5 представлена средняя оценка МАЕ для всех моделей. Можно заметить, что после приблизительно количества эпох 40 график МАЕ для

проверочных данных начинает стремиться вверх, что может являться последствием переобучения.

Применим перекрестную проверку по К блокам при различных К:

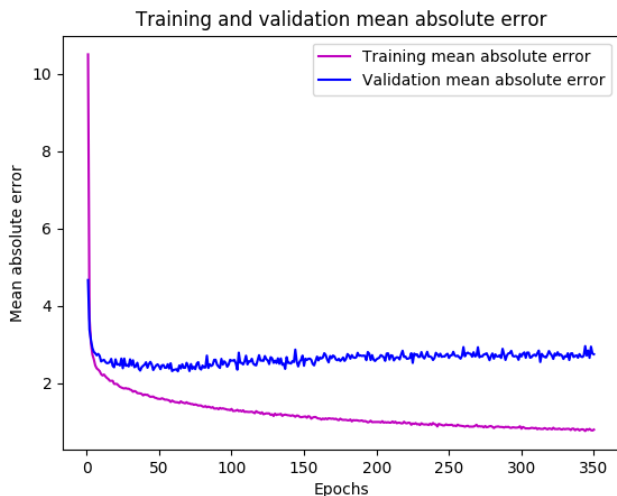


Рисунок 6 — График оценки средней MAE при $k=2$

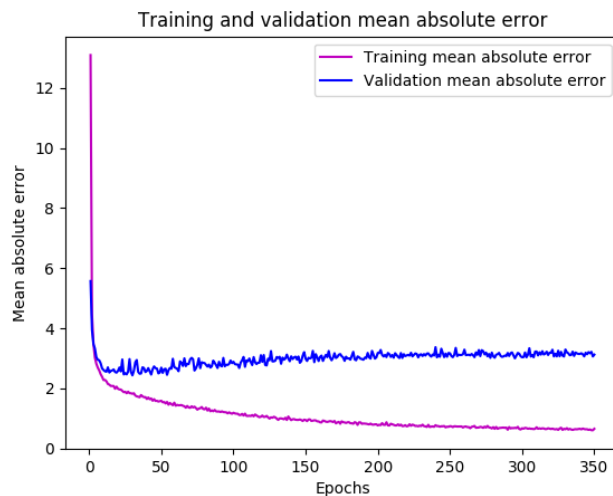


Рисунок 7 — График оценки средней MAE при $k=4$

На рис. 6 представлен график средней оценки MAE при $k=2$. На рис. 7 представлен график оценки MAE при $k=4$. Можно заметить, что при $k=2$ переобучение наступает в точке 20, при $k=4$ – в точке 40. Что свидетельствует о том, что перекрестная проверка по К блокам является один из средств снижения возможности переобучения.

Выводы.

В ходе выполнения лабораторной работы были выявлены различия задач классификации и регрессии: задача классификации – используется для нечисловых данных, сможет ли человек выплатить кредит, в результате получает множество значений, определяющих принадлежность к тому или иному классу, задача регрессии – в основном используется для числовых данных, когда нужно предсказать, вероятных заработок человека с данной профессией, в результате – единственное значение, выражение количества. Была выявлена точка переобучения: 40 эпох. Была применена перекрестная проверка по К блокам при различных К, что объясняет: при увеличении К переобучение должно стремиться к минимуму.