федеральное государственное автономное образовательное учреждение высшего образования

Санкт-Петербургский национальный исследовательский УНИВЕРСИТЕТ информационных технологий, механики и оптики

**Факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 3**

**ПО ДИСЦИПЛИНЕ «РАЗРАБОТКА иНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ»**

Выполнил:

Чудаков М.И.

Группа: P3419

Преподаватель:

Жукова Н.А.

Санкт-Петербург

2019/2020

**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.

**Ход работы**

Набор данных

**import** numpy **as** np  
**from** tensorflow.keras.datasets **import** boston\_housing  
**from** tensorflow.keras.layers **import** Dense  
**from** tensorflow.keras.models **import** Sequential  
  
  
**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

Сеть заканчивается одномерным слоем, не имеющим функции активации (это линейный слой). Это типичная конфигурация для скалярной регрессии (целью которой является предсказание одного значения на непрерывной числовой прямой). Применение функции активации могло бы ограничить диапазон выходных значений: например, если в последнем слое применить функцию активации sigmoid, сеть обучилась бы предсказывать только значения из диапазона между 0 и 1.

В данном случае, с линейным последним слоем, сеть способна предсказывать значения из любого диапазона.

(train\_data, train\_targets), (test\_data, test\_targets) = boston\_housing.load\_data()  
  
print(train\_data.shape)  
print(test\_data.shape)  
print(test\_targets)  
  
mean = train\_data.mean(axis=0)  
train\_data -= mean  
std = train\_data.std(axis=0)  
train\_data /= std  
  
test\_data -= mean  
test\_data /= std  
  
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))

|  |  |
| --- | --- |
| **Номер эксперемента** | **Отклонение $** |
| 1 | 2623 |
| 2 | 2511 |
| 3 | 2576 |
| 4 | 2612 |
| 5 | 2568 |

**Вывод**

Итого среднее отклонение колеблется в порядке 2500-2600 долларов США, стоимость самой недвижимости составляет 10 000 – 50 000, т.е. отклонение составляет 5,1% - 25,5%.