

Лабораторная работа 1

Прогнозирование цены на жилье с помощью глубоких нейронных сетей

Задание.

Набор данных по жилью в Бостоне входит в состав Keras, получить доступ к нему можно с помощью следующей команды: `keras.datasets.boston_housing`.

- 1) Разберите фрагмент кода 1, дайте пояснения к происходящему в нем. Модифицируйте данный фрагмент, определив только один слой с одним нейроном с линейной функцией активации. Приведите сравнение с исходным вариантом.
- 2) Модифицируйте фрагмент кода 1 с учетом следующих требований:
 - a. Добавьте в модель регуляризацию L2 ($\lambda = 0.1$)
 - b. Добавьте в модель дропаут (коэффициент 0.2)
 - c. К предыдущему варианту добавьте еще один слой и увеличьте количество нейронов до 128 в первых двух слоях.
 - d. Увеличьте коэффициент отсева до 0.3

По каждому варианту добавьте вывод и сравнение, указывая топологию, значение ошибки обучения и ошибки теста. Подведите итог.

Фрагмент кода 1 – DNN с двумя скрытыми слоями, используемый для прогнозирования цен на жилье

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import numpy as np
import logging
tf.get_logger().setLevel(logging.ERROR)

EPOCHS = 500
BATCH_SIZE = 16

#Чтение и стандартизация данных
bostonhousing = keras.datasets.boston_housing
(raw_x_train, y_train), (raw_x_test,
    y_test) = bostonhousing.load_data()
x_mean = np.mean(raw_x_train, axis=0)
x_stddev = np.std(raw_x_train, axis=0)
x_train = (raw_x_train - x_mean) / x_stddev
x_test = (raw_x_test - x_mean) / x_stddev
```

```
# Создание и обучение модели
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=[13]))
model.add(Dense(64, activation='relu')) # мы занимаемся DL
model.add(Dense(1, activation='linear'))
model.compile(loss='mean_squared_error', optimizer='adam'
metrics =['mean_absolute_error'])
model.summary()
    history = model.fit(x_train, y_train, validation_data=(
        x_test, y_test), epochs=EPOCHS, batch_size=BATCH_SIZE
        verbose=2, shuffle=True)
# Печать первых 4 прогнозов
predictions = model.predict(x_test)
for i in range(0, 4):
    print('Prediction: ', predictions[i], ', true value: ',
y_test[i])
```