

# Лабораторная работа №5

## Задание

Возьмите созданную нейронную сеть, датасет fashion-mnist и попытайтесь улучшить точность обучения. Варианты для улучшения:

1. Используйте разное количество нейронов на входном слое: 400, 600, 800, 1200.
2. Добавьте в нейронную сеть скрытый слой с разным количеством нейронов: 200, 300, 400, 600, 800.
3. Добавьте несколько скрытых слоев в сеть с разным количеством нейронов в каждом слое.
4. Используйте разное количество эпох: 10, 15, 20, 25, 30.
5. Используйте разные размеры мини-выборки (batch\_size): 10, 50, 100, 200, 500.

Опишите влияние (или его отсутствие) на точность работы вашей нейронной сети изменяемых параметров.

Сохраните два варианта сети, при котором точность нейронной сети минимальна и максимальна, выведите точность и сделайте вывод о переобучении. Необходимо менять не менее трех параметров (то есть использовать не менее трех вариантов из списка выше).

## Выполнение

Сделав импорт необходимых библиотек для работы с данными и TensorFlow мы можем приступить к созданию модели:

```
1 model = Sequential()
2
3 model.add(Dense(400, input_dim=784, activation="relu")) # 400 neurons
4 model.add(Dense(10, activation="softmax"))
5 model.compile(loss="categorical_crossentropy", optimizer="SGD", metrics=
  ["accuracy"])
6 model.fit(X_train, y_train, batch_size=10, epochs=10, validation_split=0.2,
  verbose=1) #batch_size=10 epochs=10
```

Здесь мы указываем форму модели с 400 нейронами, с размером выборки 10 и с эпохой 10.

В последствии мы будем менять эти значения, чтобы сделать оптимальную нейронную сеть и вывести зависимость точности модели при разных входных значениях.

Проверяем качество обучения:

```
1 scores = model.evaluate(X_test, y_test, verbose=1)
```

```
1 10000/10000 [=====] - 1s 80us/sample - loss: 0.3566 - acc:
  0.8714
```

Сохраняем модель, чтобы не пришлось обучать его заново:

```
1 model.save("400n-10batch-10epoch.h5")
```

Таким же образом, выполняем для разных конфигурационных значений:

```
1 model = Sequential()
2 model.add(Dense(600, input_dim=784, activation="relu")) # 600 neurons
3 model.add(Dense(10, activation="softmax"))
4 model.compile(loss="categorical_crossentropy", optimizer="SGD", metrics=
  ["accuracy"])
5 model.fit(X_train, y_train, batch_size=50, epochs=15, validation_split=0.2,
  verbose=0) #batch=50;epochs=15
6 scores = model.evaluate(X_test, y_test, verbose=1)
7 model.save("600n-50batch-15epoch.h5")
```

```
1 model = Sequential()
2 model.add(Dense(800, input_dim=784, activation="relu")) # 800 neurons
3 model.add(Dense(10, activation="softmax"))
4 model.compile(loss="categorical_crossentropy", optimizer="SGD", metrics=
  ["accuracy"])
5 model.fit(X_train, y_train, batch_size=100, epochs=20, validation_split=0.2,
  verbose=0) #batch=100;epochs=20
6 scores = model.evaluate(X_test, y_test, verbose=1)
7 model.save("800n-100batch-20epochs.h5")
```

```
1 model = Sequential()
2 model.add(Dense(1200, input_dim=784, activation="relu")) # 1200 neurons
3 model.add(Dense(10, activation="softmax"))
4 model.compile(loss="categorical_crossentropy", optimizer="SGD", metrics=
  ["accuracy"])
5 model.fit(X_train, y_train, batch_size=200, epochs=100, validation_split=0.2,
  verbose=0) #batch=200;epochs=25
6 scores = model.evaluate(X_test, y_test, verbose=1)
7 model.save("1200n-200batch-25epoch.h5")
```

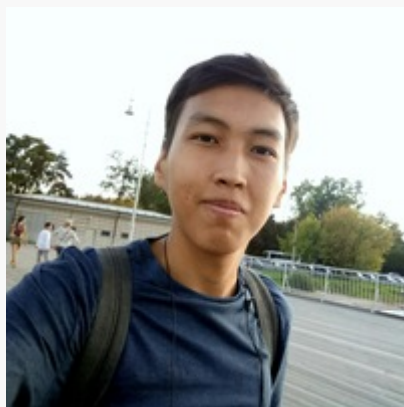
```
1 model = Sequential()
2 model.add(Dense(1200, input_dim=784, activation="relu")) # 1200 neurons
3 model.add(Dense(10, activation="softmax"))
4 model.compile(loss="categorical_crossentropy", optimizer="SGD", metrics=
  ["accuracy"])
5 model.fit(X_train, y_train, batch_size=200, epochs=100, validation_split=0.2,
  verbose=0) #batch=500;epochs=30
6 scores = model.evaluate(X_test, y_test, verbose=1)
7 model.save("1200n-500batch-30epoch.h5")
```

Как итог, мы получили такую таблицу:

Neurons	batch_size	epochs	acc
400	10	10	0.8714
600	50	15	0.8552
800	100	20	0.8488
1200	200	25	0.8692
1200	500	30	0.8673

На ней мы можем наблюдать переобучение, а следовательно регресс точности модели. Поэтому при выборке конфигурационных переменных, мы не должны допускать переобучения.

## Authors



[Arthur Kupriyanov](#)



[Artyom Kolokolov](#)

Группа: P3212