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

ОТЧЕТ

по лабораторной работе №5

по дисциплине «Искусственные нейронные сети»

Тема: Распознавание объектов на фотографиях

Студент гр. 8382 Мирончик П.Д.

Преподаватель Жангиров Т.Р.

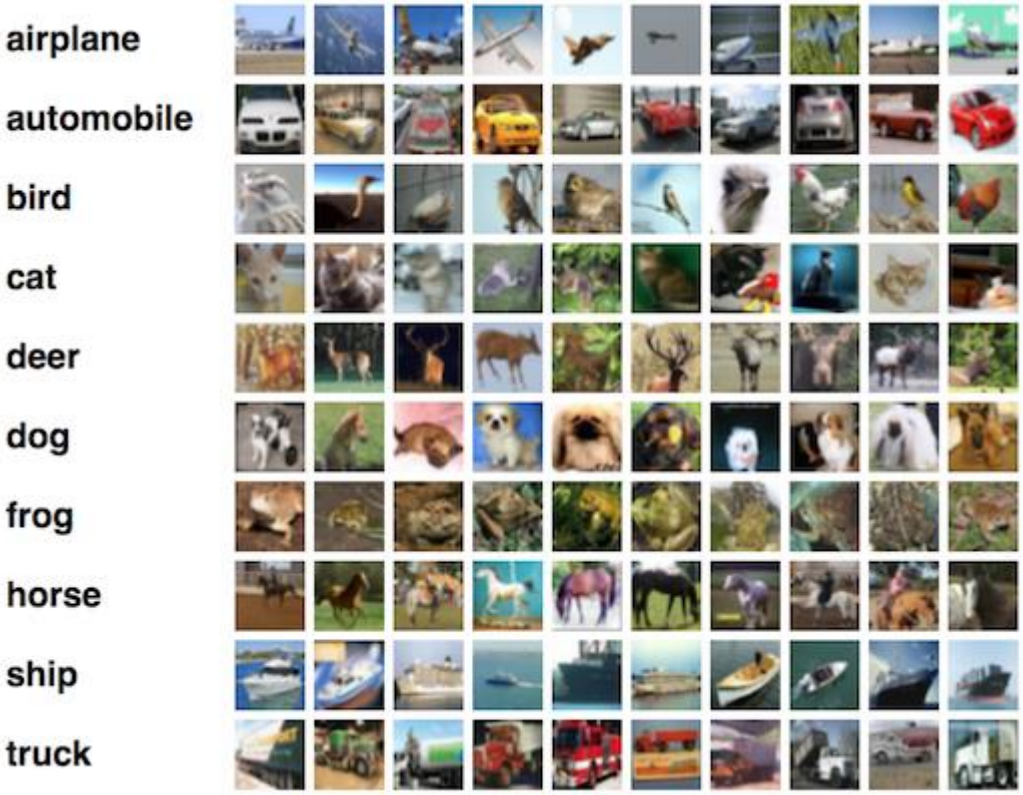
Санкт-Петербург

2021

ЗАДАНИЕ

Распознавание объектов на фотографиях (Object Recognition in Photographs).

CIFAR-10 (классификация небольших изображений по десяти классам: самолет, автомобиль, птица, кошка, олень, собака, лягушка, лошадь, корабль и грузовик).



ЗАДАЧИ

- Ознакомиться со сверточными нейронными сетями
- Изучить построение модели в Keras в функциональном виде
- Изучить работу слоя разреживания (Dropout)

ТРЕБОВАНИЯ

1. Построить и обучить сверточную нейронную сеть
2. Исследовать работу сеть без слоя Dropout
3. Исследовать работу сети при разных размерах ядра свертки

ХОД РАБОТЫ

В работе используется сверточная нейронная сеть, которая позволяет находить на изображении шаблоны независимо от их расположения и изучать пространственные иерархии шаблонов.

Модель представляет из себя четыре слоя свертки (выделение наиболее полезных признаков), два слоя субдискретизации (сокращение параметров путем уменьшения изображения) и два полносвязных слоя (непосредственно классификация).

Параметры модели:

```
num_train, depth, height, width = X_train.shape
batch_size = 32
num_epochs = 200
kernel_size = 3
pool_size = 2
conv_depth_1 = 32
conv_depth_2 = 64
drop_prob_1 = 0.25
drop_prob_2 = 0.5
hidden_size = 512
```

`depth`, `height`, `width` - размеры входного слоя;

`kernel_size` - размер ядря свертки для сверточных слоев (в нашем случае ядро 3x3);

`pool_size` - размер окна субдискретизации;

`conv_depth_1`, `conv_depth_2` - число признаков, которые вычисляют слои свертки (32 для первых двух слоев и 64 для 3 и 4 слоя);

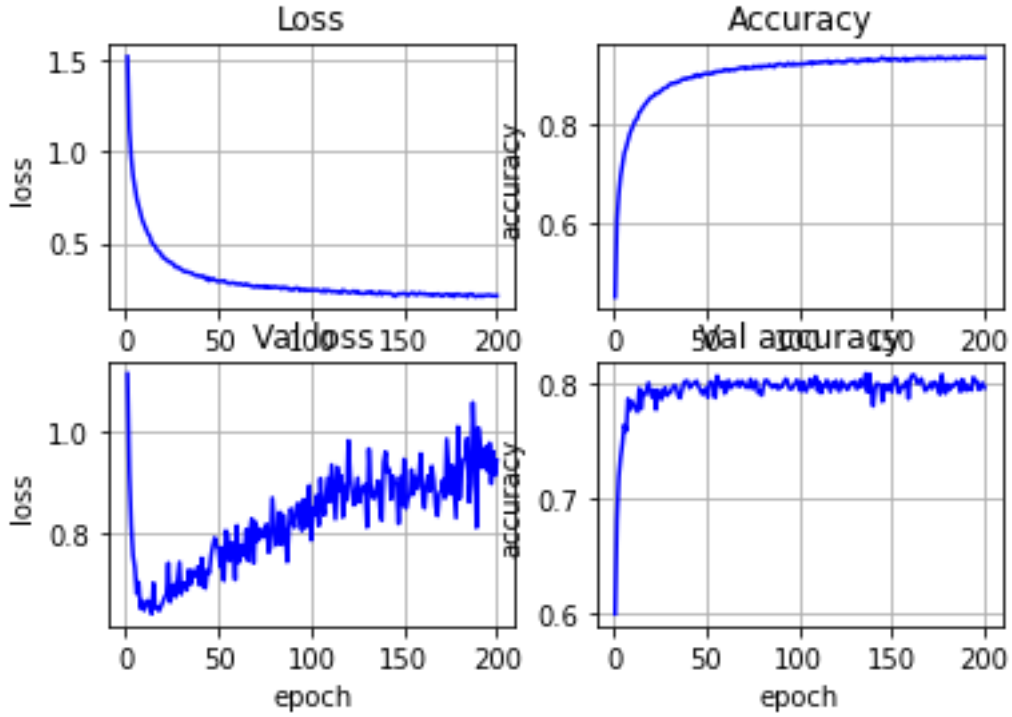
`drop_prob_1`, `drop_prob_2` - вероятности выброса нейронов из слоев.

```
inp = Input(shape=(depth, height, width))

conv_1 = Convolution2D(conv_depth_1, kernel_size,
padding='same', activation='relu')(inp)
conv_2 = Convolution2D(conv_depth_1, kernel_size,
padding='same', activation='relu')(conv_1)
pool_1 = MaxPooling2D(pool_size=(pool_size, pool_size))(conv_2)
drop_1 = Dropout(drop_prob_1)(pool_1)

conv_3 = Convolution2D(conv_depth_2, kernel_size,
padding='same', activation='relu')(drop_1)
conv_4 = Convolution2D(conv_depth_2, kernel_size,
padding='same', activation='relu')(conv_3)
```

```
pool_2 = MaxPooling2D(pool_size=(pool_size, pool_size))(conv_4)
drop_2 = Dropout(drop_prob_1)(pool_2)

flat = Flatten()(drop_2)
hidden = Dense(hidden_size, activation='relu')(flat)
drop_3 = Dropout(drop_prob_2)(hidden)
out = Dense(num_classes, activation='softmax')(drop_3)
```

Обучим построенную выше модель на 200 эпохах:

```
Epoch 200/200
1407/1407 [==============================] - 9s 6ms/step - loss:
0.2119 - accuracy: 0.9322 - val_loss: 0.9439 - val_accuracy:
0.7966
313/313 [==============================] - 1s 3ms/step - loss:
457.8422 - accuracy: 0.4912
```

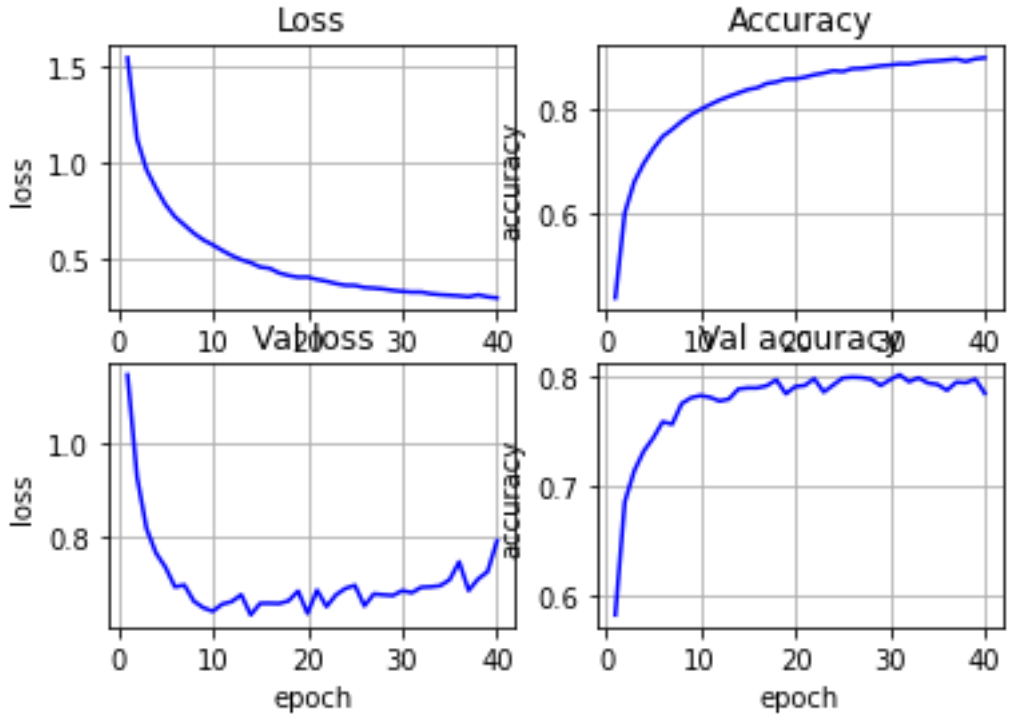


Точность модели на валидационном множестве достаточно неплохая: 79.7%, однако на валидационном множестве она составляет лишь 49.1%. Из графиков видно, что точность на валидационных данных перестает повышаться уже к 40 эпохе, в то время как потери начинают расти - это говорит о переобучении модели.

Попробуем обучить модель в течение 40 эпох:

```
Epoch 40/40
1407/1407 [==============================] - 9s 6ms/step - loss:
0.2940 - accuracy: 0.8989 - val_loss: 0.7936 - val_accuracy:
0.7848
```

```
313/313 [==============================] - 1s 3ms/step - loss:
239.1865 - accuracy: 0.5781
```



Видно, что точность на тестовых данных выросла почти на 10%, при том, что показатели валидационных данных также улучшились.

Попробуем убрать Dropout слои из модели и сравнить результаты.

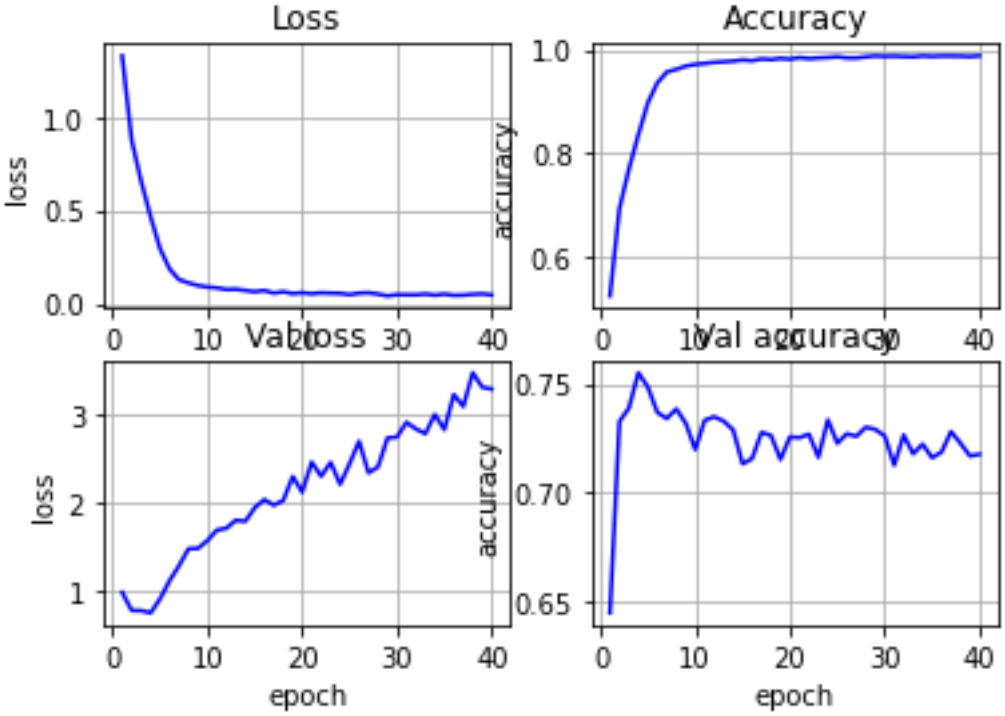
Используемая модель:

```
inp = Input(shape=(depth, height, width))

conv_1 = Convolution2D(conv_depth_1, kernel_size,
padding='same', activation='relu')(inp)
conv_2 = Convolution2D(conv_depth_1, kernel_size,
padding='same', activation='relu')(conv_1)
pool_1 = MaxPooling2D(pool_size=(pool_size, pool_size))(conv_2)

conv_3 = Convolution2D(conv_depth_2, kernel_size,
padding='same', activation='relu')(pool_1)
conv_4 = Convolution2D(conv_depth_2, kernel_size,
padding='same', activation='relu')(conv_3)
pool_2 = MaxPooling2D(pool_size=(pool_size, pool_size))(conv_4)

flat = Flatten()(pool_2)
hidden = Dense(hidden_size, activation='relu')(flat)
out = Dense(num_classes, activation='softmax')(hidden)
```

Результаты:

```
Epoch 40/40
```

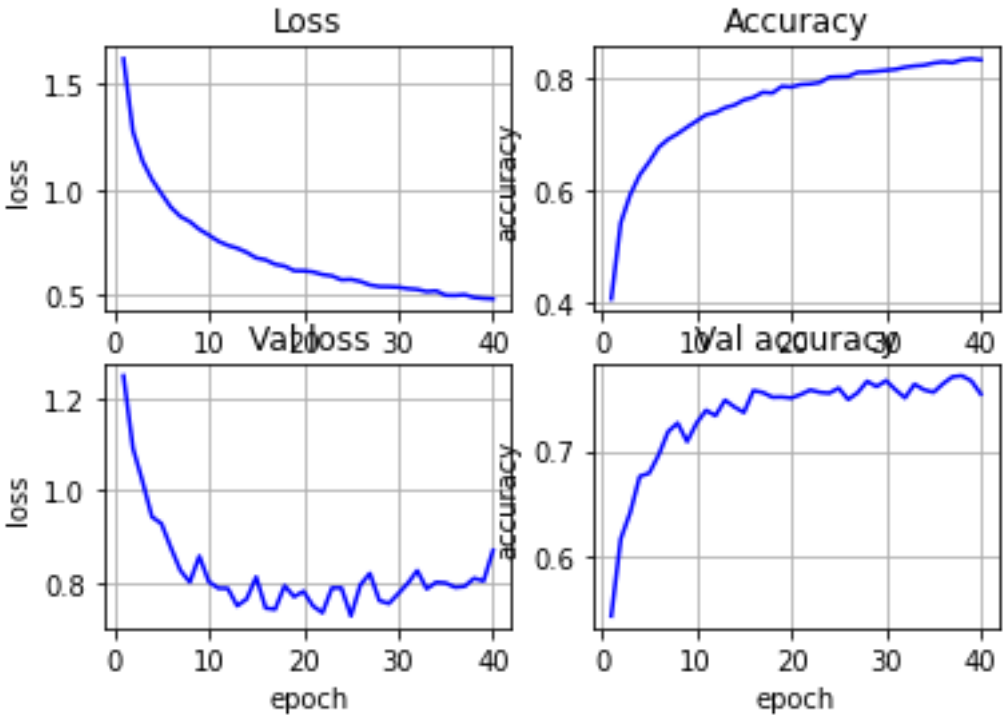
```
1407/1407 [==============================] - 8s 6ms/step - loss:
0.0419 - accuracy: 0.9896 - val_loss: 3.3005 - val_accuracy:
0.7176
313/313 [==============================] - 1s 3ms/step - loss:
1279.4436 - accuracy: 0.5559
```



Модель начала переобучаться уже к 5 эпохе, и, несмотря на примерно схожую точность на тестовых данных, потери модели без dropout слоев значительно выше, поэтому выгоднее использовать модель с Dropdown слоями.

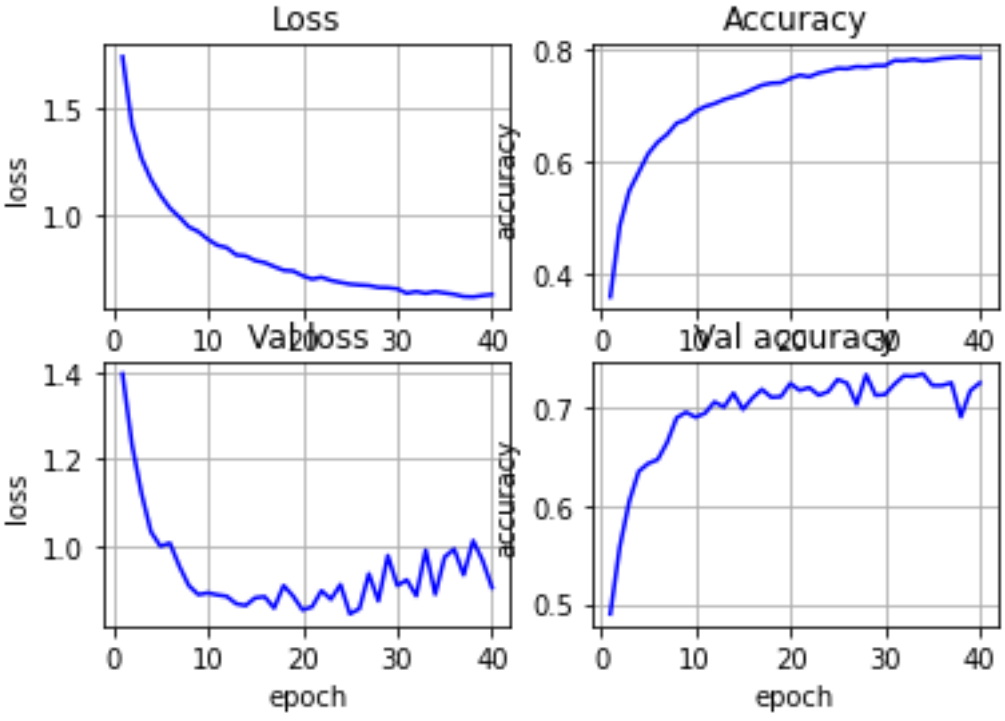
Исследуем работу сети при разных размерах ядра свертки. Установим размер `kernel_size` в 5 (теперь ядро свертки будет иметь размер 5x5 элемента). Рассмотрим результаты обучения и тестирования модели:

```
Epoch 40/40
1407/1407 [==============================] - 13s 9ms/step -
loss: 0.4745 - accuracy: 0.8352 - val_loss: 0.8690 -
val_accuracy: 0.7536
313/313 [==============================] - 1s 4ms/step - loss:
240.7982 - accuracy: 0.5467
```



Установим `kernel_size` в 7:

```
Epoch 40/40
1407/1407 [==============================] - 17s 12ms/step -
loss: 0.6261 - accuracy: 0.7876 - val_loss: 0.9063 -
val_accuracy: 0.7256
313/313 [==============================] - 2s 5ms/step - loss:
497.4786 - accuracy: 0.4699
```



По результатам исследования ИНС с разными значениями kernel_size заметно, что при увеличении размера ядра точность падает, а потери, наоборот, возрастают, как на тестовых.

ВЫВОДЫ

В ходе выполнения лабораторной работы была создана нейронная сеть со сверточной архитектурой, осуществляющая классификацию изображений. Рассмотрены принципы работы нейронных сетей, влияние слоя Dropout размера ядра свертки на результаты работы нейронной сети.