

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

ОТЧЕТ
по лабораторной работе №5
"Распознавание объектов на фотографиях"
по дисциплине «Искусственные нейронные сети»

Студентка гр. 8383

Преподаватель

Ишанина Л.Н.

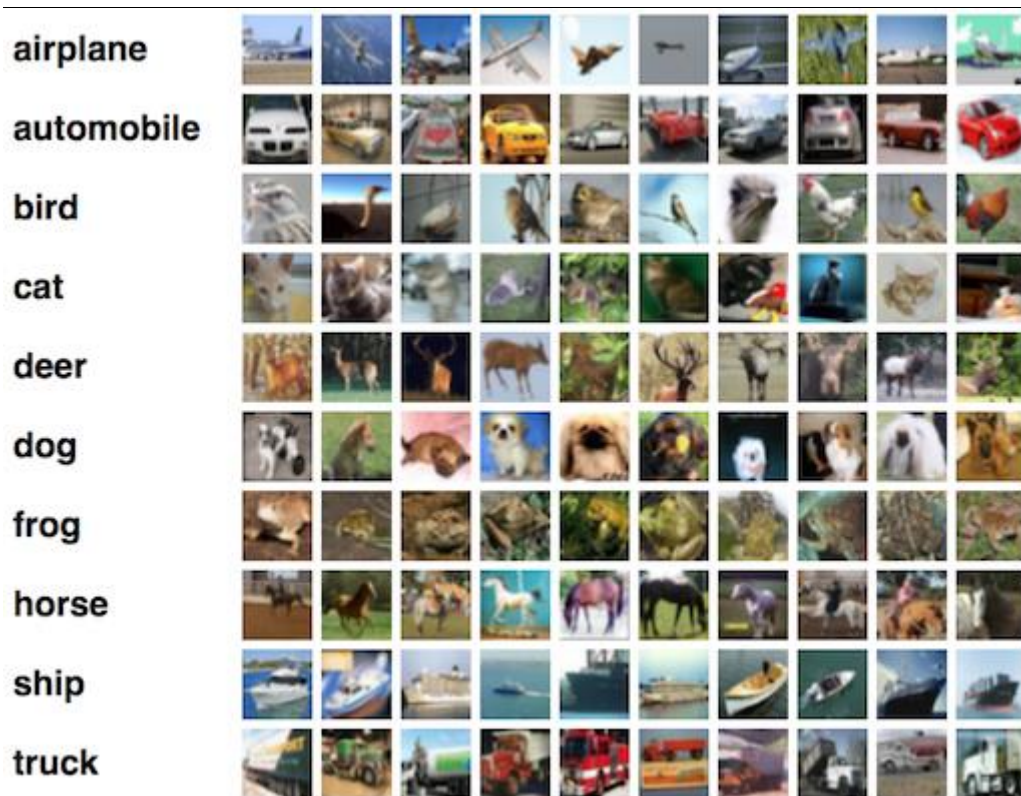
Жангиров Т.Р.

Санкт-Петербург

2021

Цель.

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).



Задание.

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

Построить и обучить сверточную нейронную сеть

Исследовать работу сеть без слоя Dropout

Исследовать работу сети при разных размерах ядра свертки

Выполнение работы.

Были подключены необходимые библиотеки:

```
from keras.datasets import cifar10
from keras.models import Model
from keras.layers import Input, Convolution2D, MaxPooling2D, Dense, Dropout, Flatten
import numpy as np
```

Далее созданы переменные, которые представлены в листинге ниже:

```
batch_size = 64
num_epochs = 15
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
```

`batch_size` — количество обучающих образцов, обрабатываемых одновременно за одну итерацию алгоритма градиентного спуска;

`num_epochs` — количество итераций обучающего алгоритма по всему обучающему множеству;

`kernel_size` — размер ядра в сверточных слоях;

`pool_size` — размер подвыборки в слоях подвыборки;

`conv_depth` — количество ядер в сверточных слоях;

`drop_prob` (dropout probability) — мы будем применять dropout после каждого слоя подвыборки, а также после полносвязного слоя;

`hidden_size` — количество нейронов в полносвязном слое MLP

Затем была создана модель, представленная в листинге ниже:

```
inp = Input(shape=input_shape)
conv_1 = Convolution2D(conv_depth_1, (kernel_size, kernel_size),
                      padding='same', activation='relu')(inp)
conv_2 = Convolution2D(conv_depth_1, (kernel_size, kernel_size),
                      padding='same', activation='relu')(conv_1)
pool_1 = MaxPooling2D(pool_size=pool_size)(conv_2)
if dropout:
    drop_1 = Dropout(drop_prob_1)(pool_1)
```

```

else:
    drop_1 = pool_1
conv_3 = Convolution2D(conv_depth_2, (kernel_size, kernel_size),
                        padding='same', activation='relu')(drop_1)
conv_4 = Convolution2D(conv_depth_2, (kernel_size, kernel_size),
                        padding='same', activation='relu')(conv_3)
pool_2 = MaxPooling2D(pool_size=pool_size)(conv_4)
if dropout:
    drop_2 = Dropout(drop_prob_1)(pool_2)
else:
    drop_2 = pool_2
flat = Flatten()(drop_2)
hidden = Dense(hidden_size, activation='relu')(flat)
if dropout:
    drop_4 = Dropout(drop_prob_2)(hidden)
else:
    drop_4 = hidden
out = Dense(num_classes, activation='softmax')(drop_4)
model = Model(inp, out)

```

Результаты работы

1) Были выбраны следующие параметры:

batch_size = 64;

num_epochs = 15;

kernel_size = 3;

Используются слои Dropout.

Графики точности и потерь представлены на рис. 1-2.

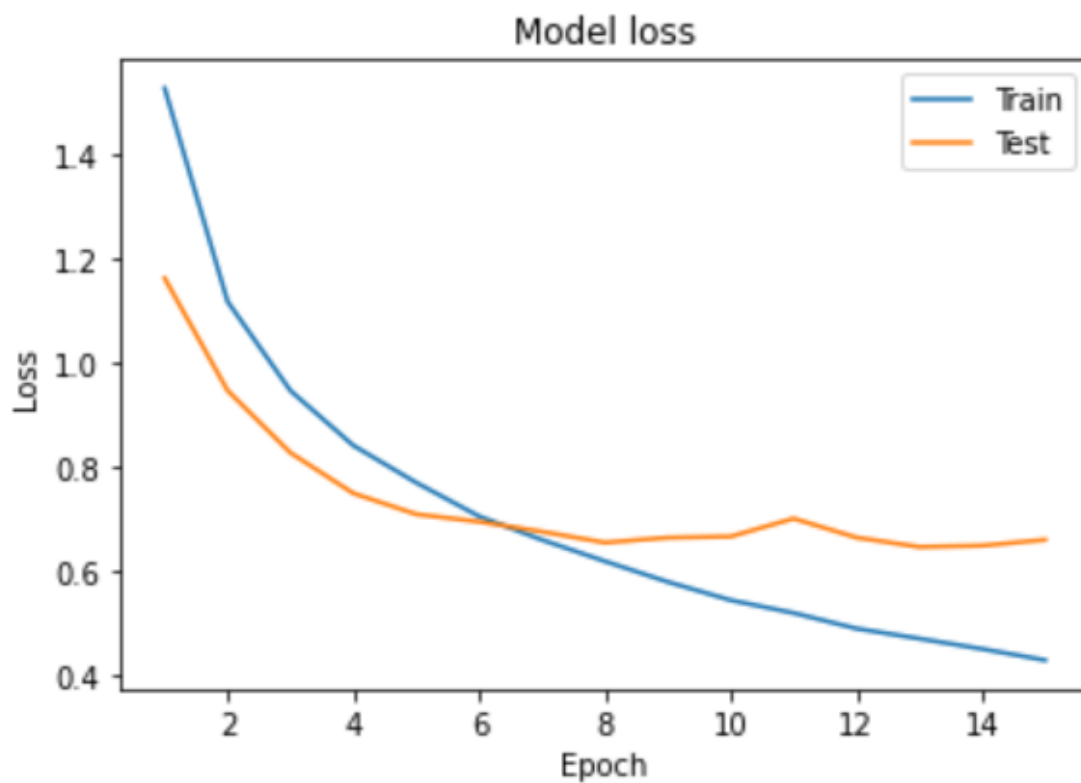


Рисунок 1 – графики потерь

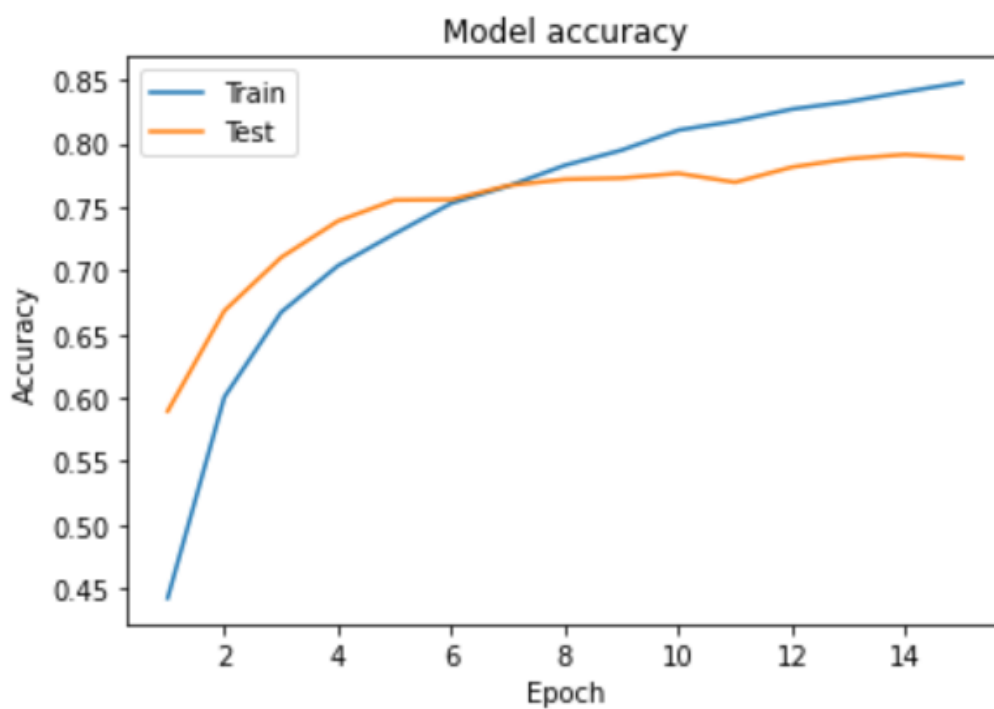


Рисунок 2 – графики точности

Исходя из графиков видно, что явного переобучения не наблюдается.

Значения потери и точности на последней эпохе:

val_loss: 0.6399 - val_accuracy: 0.7835

2) Далее были выбраны следующие параметры:

`batch_size = 64;`

`num_epochs = 15;`

`kernel_size = 2;`

Используются слои Dropout.

Графики точности и потерь представлены на рис. 3-4.

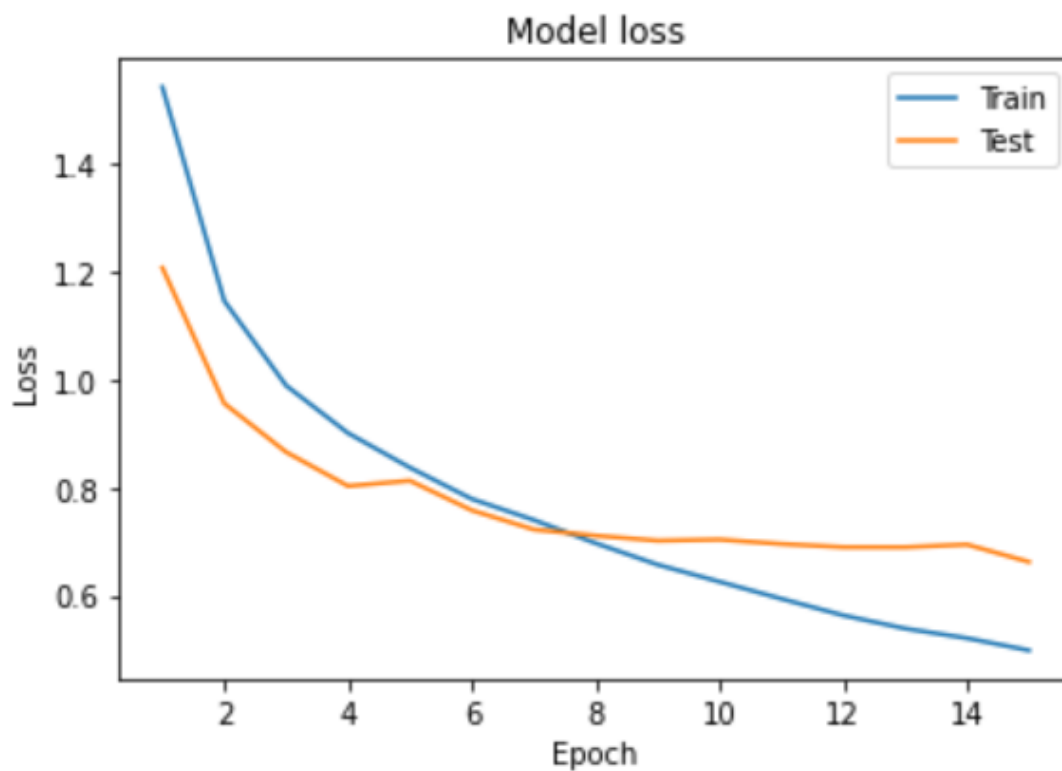


Рисунок 3 – графики потерь

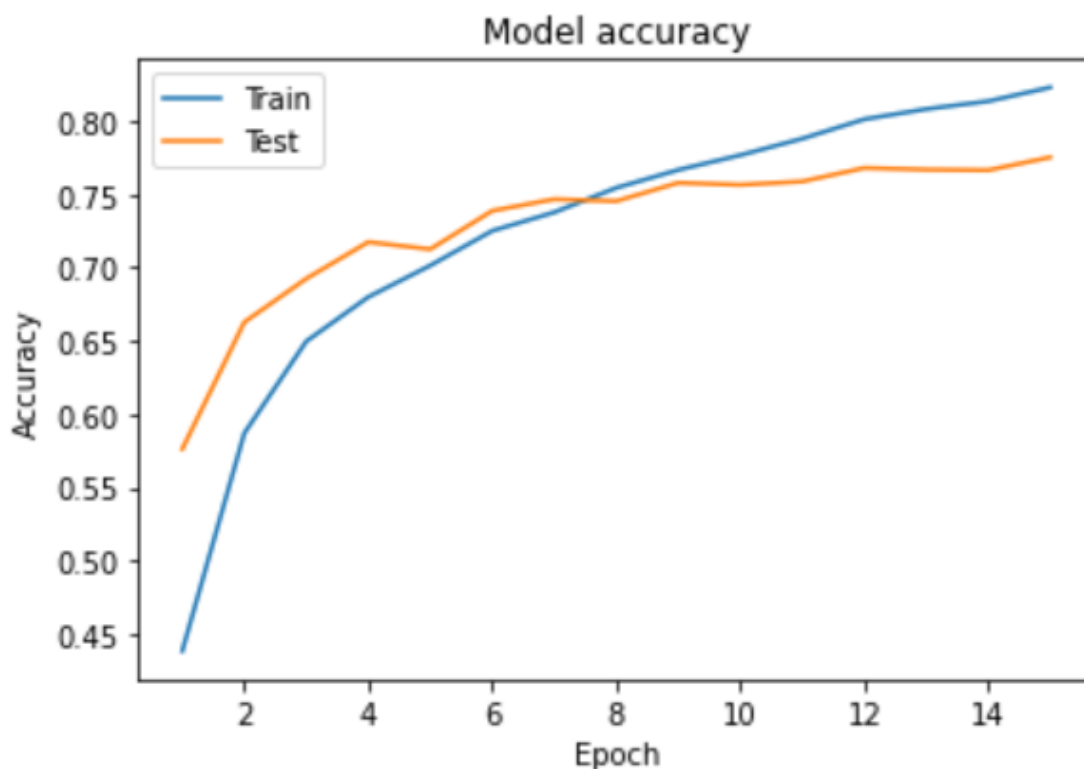


Рисунок 4 – графики точности

Исходя из графиков видно, что явного переобучения не наблюдается.

Значения потери и точности на последней эпохе:

`val_loss: 0.6622 - val_accuracy: 0.7751`

По сравнению с предыдущим запуском значение точности уменьшилось, а значение потери возросло. Следовательно, при `kernel_size = 2` модель обучается хуже.

3) Далее были выбраны следующие параметры:

`batch_size = 64;`

`num_epochs = 15;`

`kernel_size = 4;`

Используются слои Dropout.

Графики точности и потерь представлены на рис. 5-6.

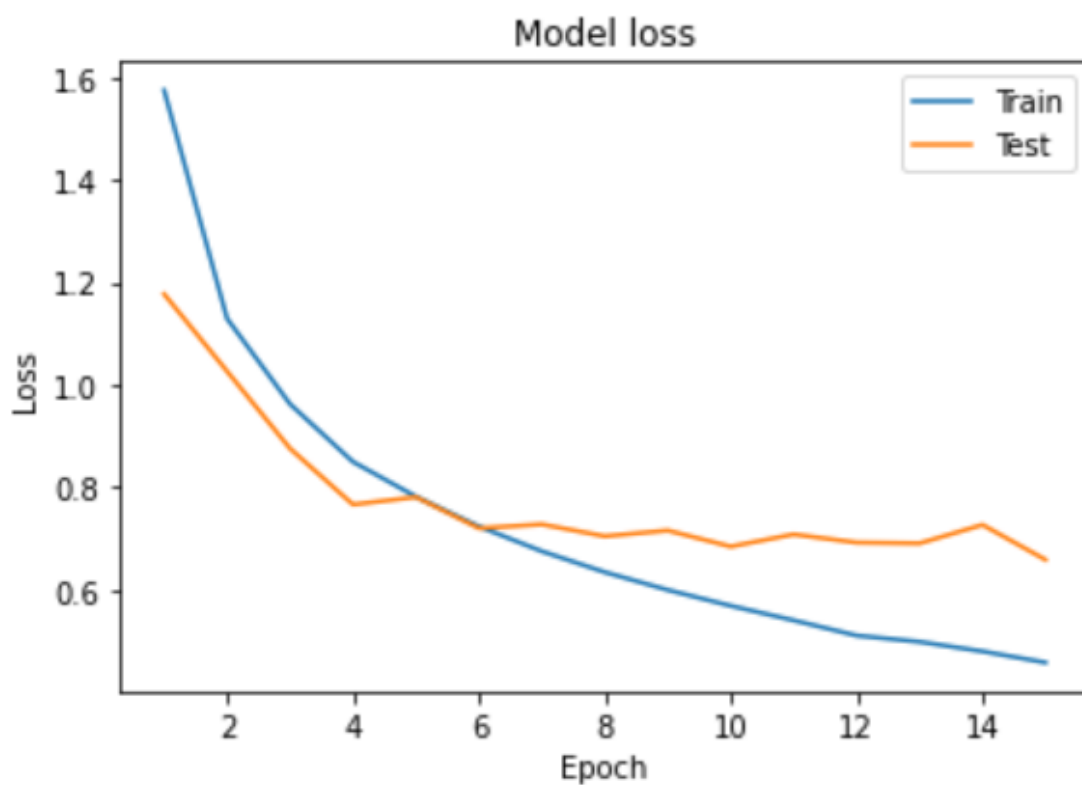


Рисунок 5 – графики потерь

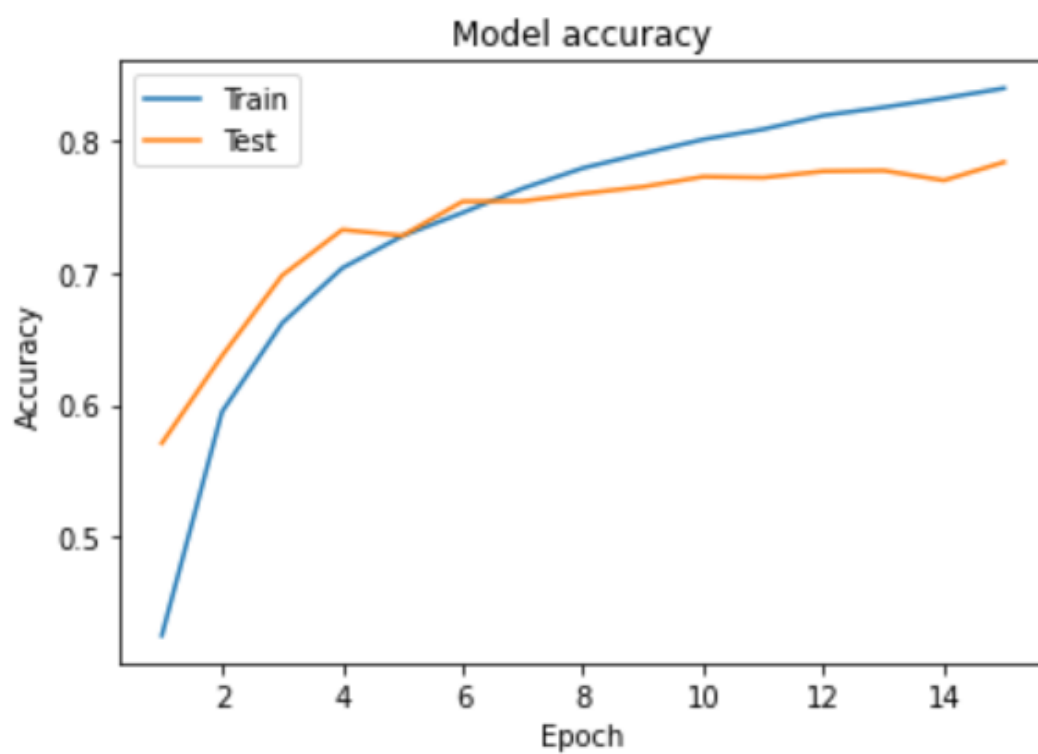


Рисунок 6 – графики точности

Исходя из графиков видно, что явного переобучения не наблюдается.
Значения потери и точности на последней эпохе:

val_loss: 0.6581 - val_accuracy: 0.7831

По сравнению с предыдущим запуском (при `kernel_size = 2`) значение точности увеличилось, а значение потери уменьшилось. Но если сравнить текущий запуск с первым запуском (`kernel_size = 3`), то значение точности практически одинаково, а значение потери увеличилось. Следовательно, лучше всего обучается модель при `kernel_size = 3`.

4) Далее были выбраны следующие параметры:

`batch_size = 64;`

`num_epochs = 15;`

`kernel_size = 3;`

Не используются слои Dropout.

Графики точности и потерь представлены на рис. 7-8.

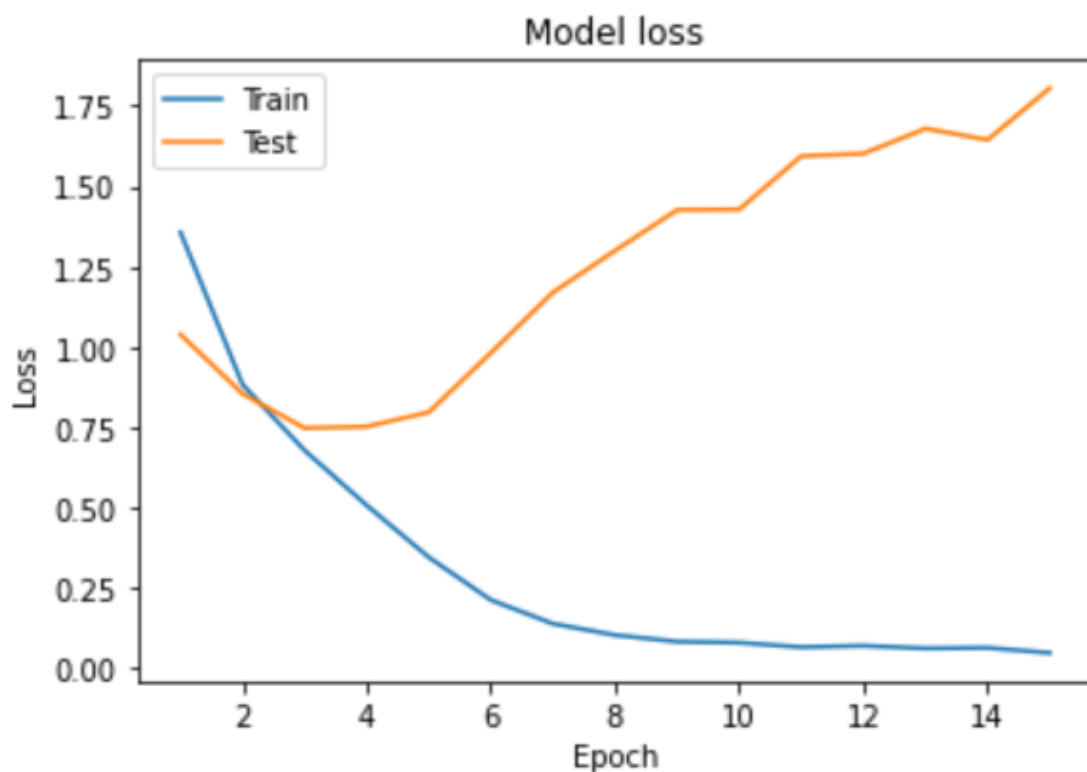


Рисунок 7 – графики потери

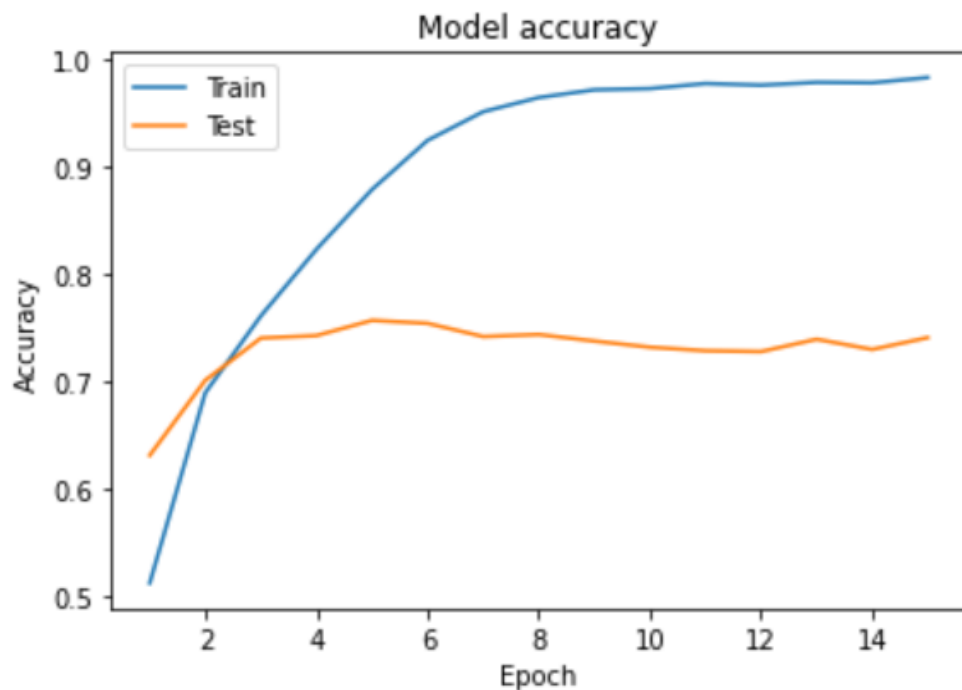


Рисунок 8 – графики точности

Значения потери и точности на последней эпохе:

val_loss: 1.8052 - val_accuracy: 0.7415

Исходя из графиков видно, что происходит переобучение и в результате дается низкая точность и высокие потери. Таким образом, слой Dropout позволяет избежать переобучения и повысить точность на данных для проверки.

Выводы.

В ходе выполнения лабораторной работы было проведено ознакомление со сверточными нейронными сетями. Было изучено построение модели в Keras в функциональном виде, а также работа слоя Dropout.