

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»

**Факультет программной инженерии и компьютерной техники**

## **ЛАБОРАТОРНАЯ РАБОТА №4**

по дисциплине

“Системы искусственного интеллекта”

Вариант №1

**Студент:**

Воробьев Кирилл  
Олегович

Группа Р33302

**Преподаватель:**

Королёва Ю.А.



Санкт-Петербург, 2022

## Цель работы

Исследование алгоритмов решения задач методом поиска

## Задание

Var	Part1 func	Part2 data	Hyperparameters
1	Absolute(Sin(x)) X: 6,3..6.3 Y: 0..1.2	CIFAR10	Layers count, neurons count per layer

There are represented such hyperparameters as

- Layer count
- Neurons count per layer (actually it's not hyperparameter but structure parameter)
- Learn rate
- Regularization L1 and L2
- Output layer activation type
- Layer activation type
- Loss function type
- Epoch count

- 1) By changing these hyperparameters try to reach max accuracy value(at least 0.95) for Part2 model with fixed epoch count 20
- 2) Change 1st hyperparameter's value from min to max with minimal step depends on your variant
- 3) Show impact on result using graphs
- 4) Describe impact of each hyperparameter on accuracy.
- 5) Set hyperparameter value back to one which produced max accuracy
- 6) Repeat 2-5 steps for second hyperparameter

Make a report including:

- Each hyperparameter description and its impact on accuracy.
- Hyperparameters' values which were used to reach accuracy value 0.95
- Graphs for these hyperparameters' values

## Выполнение работы

### Часть 1

```
# Your variant function
def main_func(x):
    return np.abs(np.sin(x))
def main_func_noisy(x):
    return main_func(x) + np.cos(4*x + 1) / 5

def result_func(xy):
    return main_func(xy[0]) > xy[1]

def result_func_noisy(xy):
    return main_func_noisy(xy[0]) > xy[1]
```

Параметры, при которых удалось достичь наибольшей достоверности:

Скользящее:

Loss function: 

Binary crossentropy

Batch size:

Learn rate: 

-0.1

0

0.001

0.0001

0.0005

0.05

0.1

0.5

1

5

Regularization L1: 

-0.1

0

0.0001

0.0005

0.001

0.005

0.01

0.05

0.1

Regularization L2: 

-0.1

0

0.0001

0.0005

0.001

0.005

0.01

0.05

0.1

Output layer activation type: 

sigmoid

Epoch count:

Neurons count in layer 1:

Neurons count in layer 2:

Neurons count in layer 1:

Neurons count in layer 1:

Layer 1 activation type: 

relu

Layer 2 activation type: 

relu

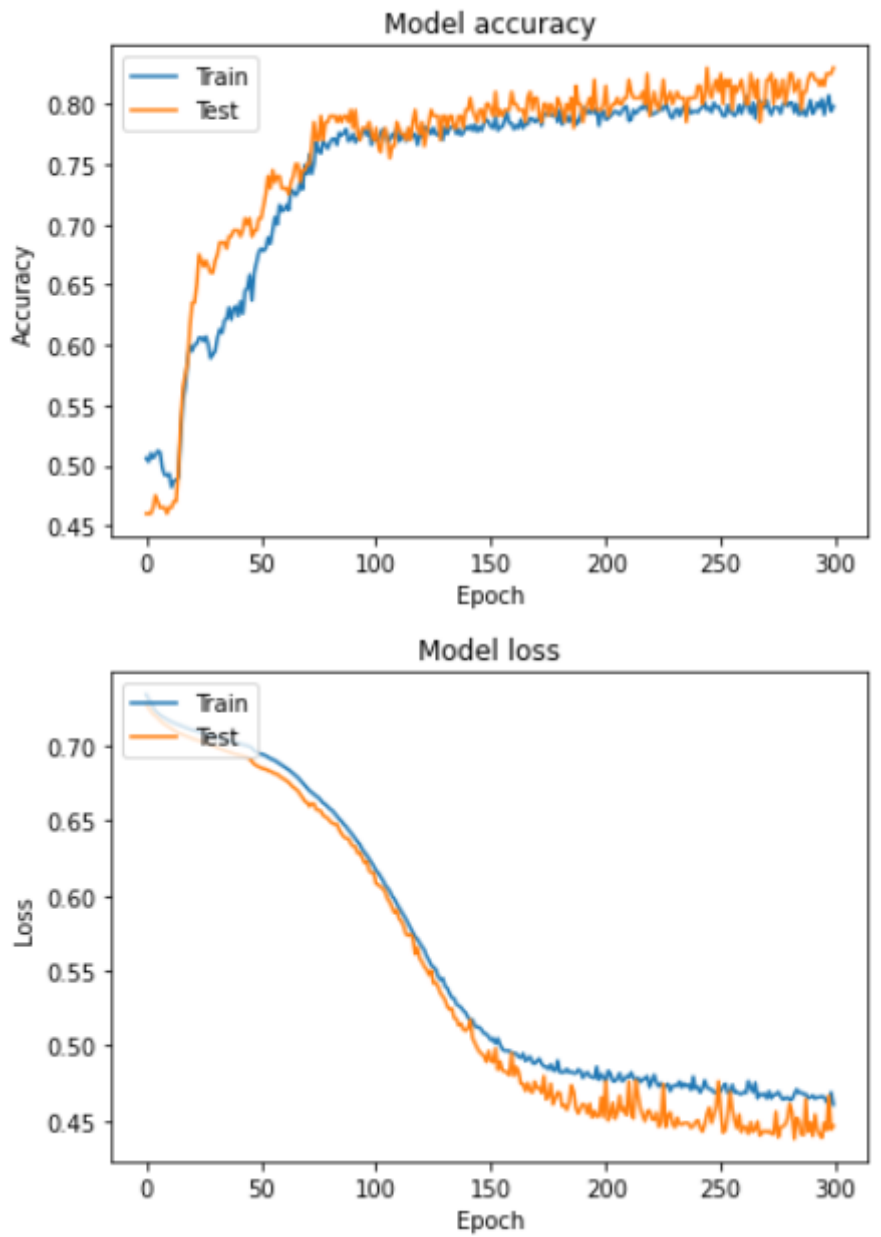
Layer 3 activation type: 

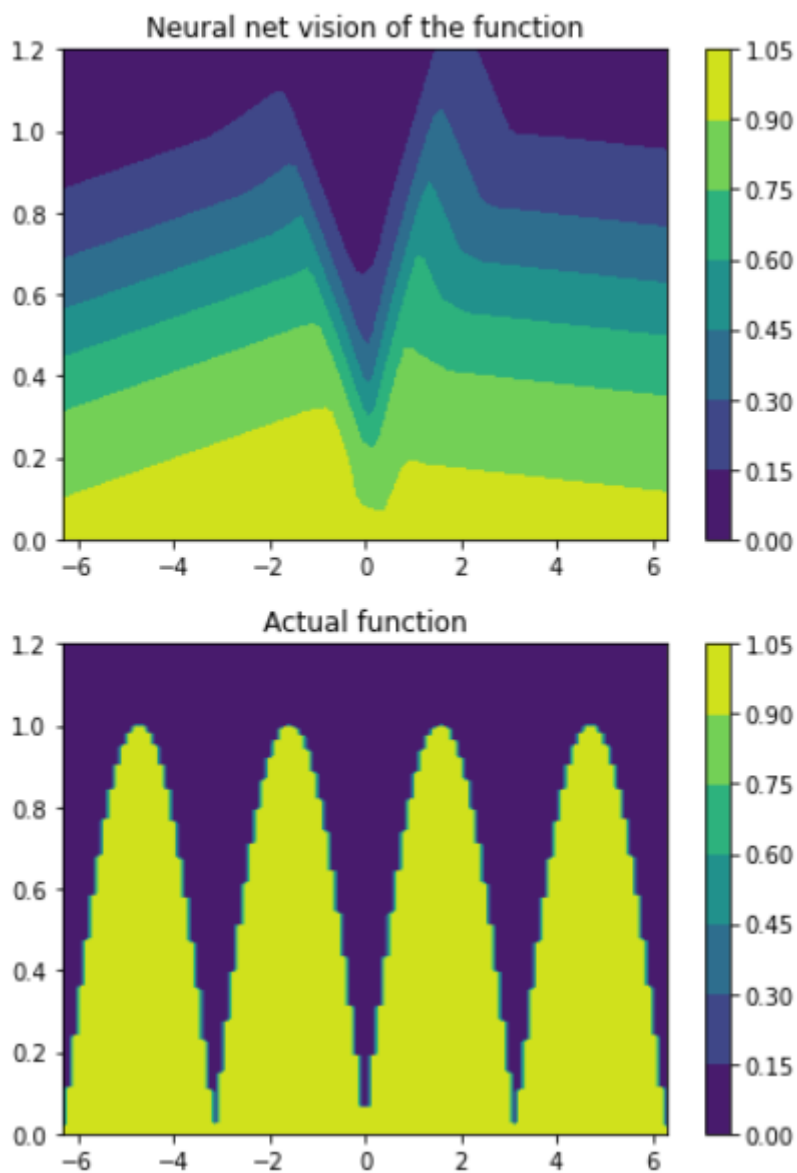
relu

Layer 4 activation type: 

relu

Accuracy: 0.8299999833106995





## Часть 2

Layers:  2

Loss function:

Batch size:

Learn rate:

Regularization L1:

Regularization L2:

Output layer activation type:

Epoch count:  20

Neurons per layer:  20

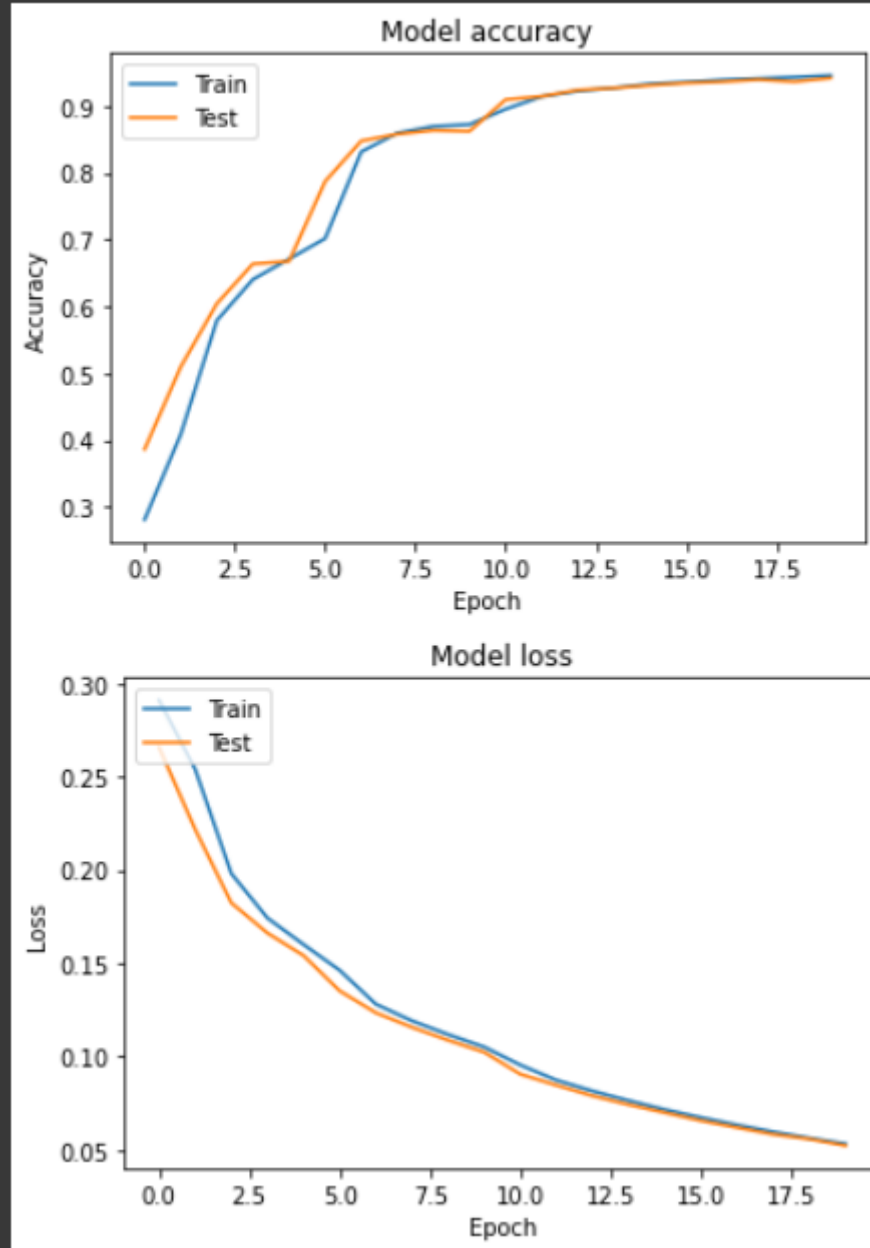
Layer 1 activation type:

Layer 2 activation type:

Layer 3 activation type:

Layer 4 activation type:

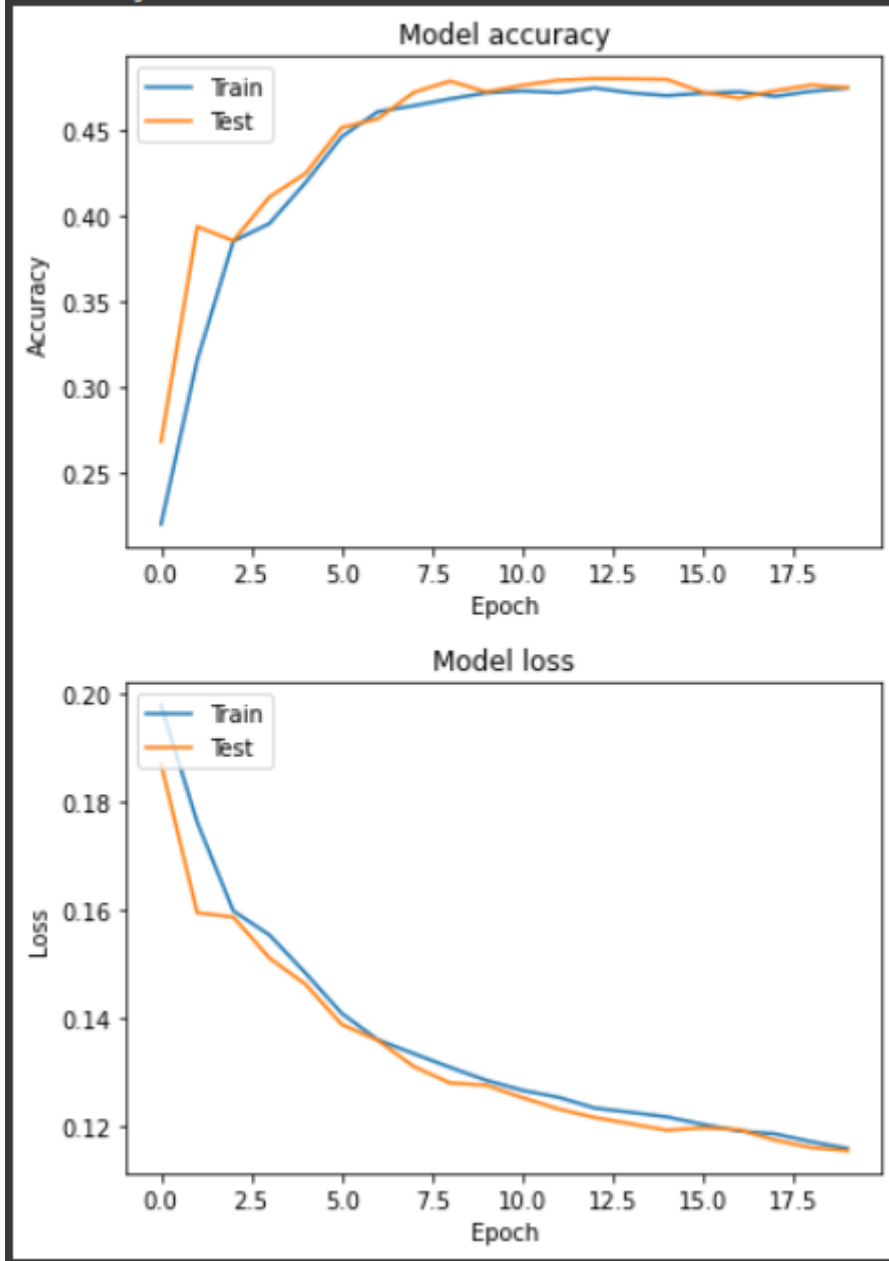
Accuracy: 0.9415000081062317



Зависимость от Layer count:

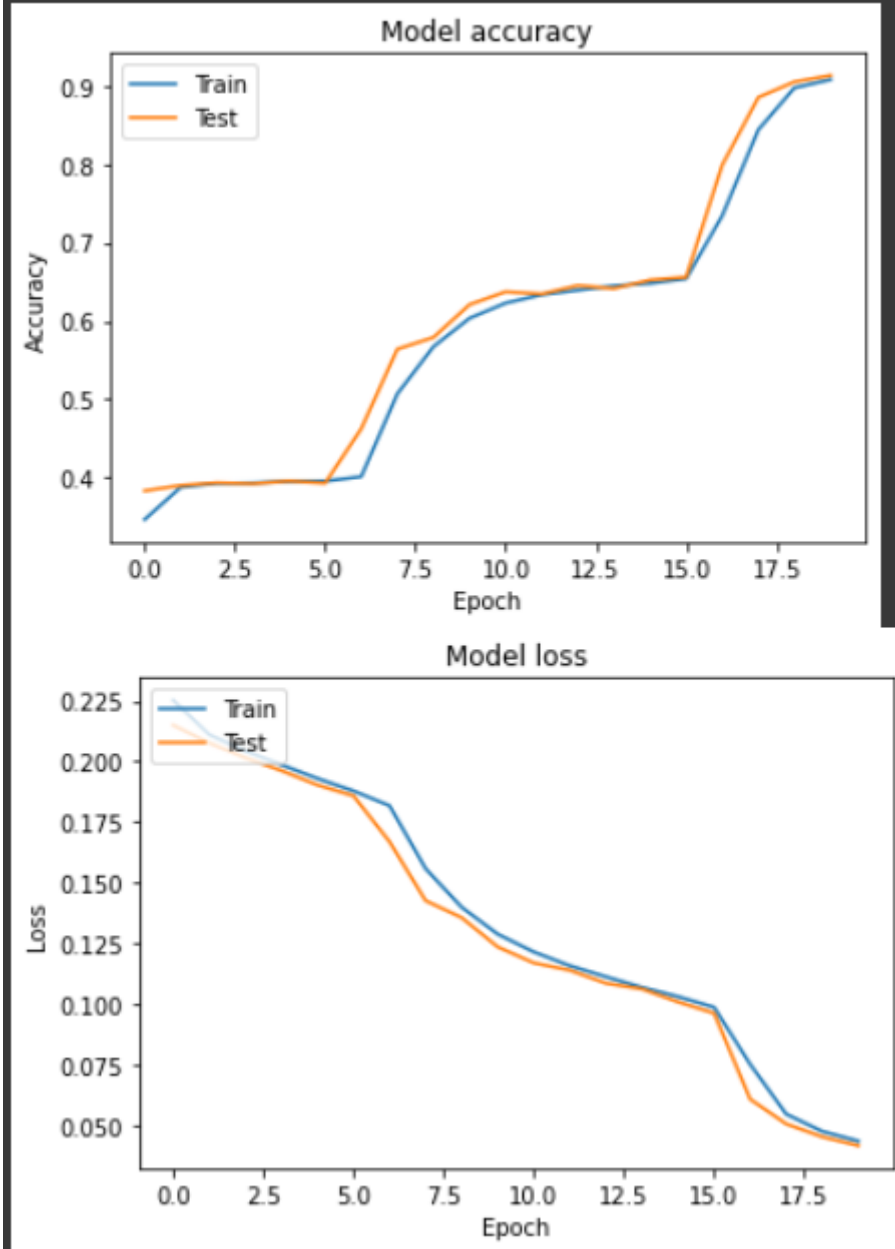
Layer count = 0:

Accuracy: 0.474700003862381



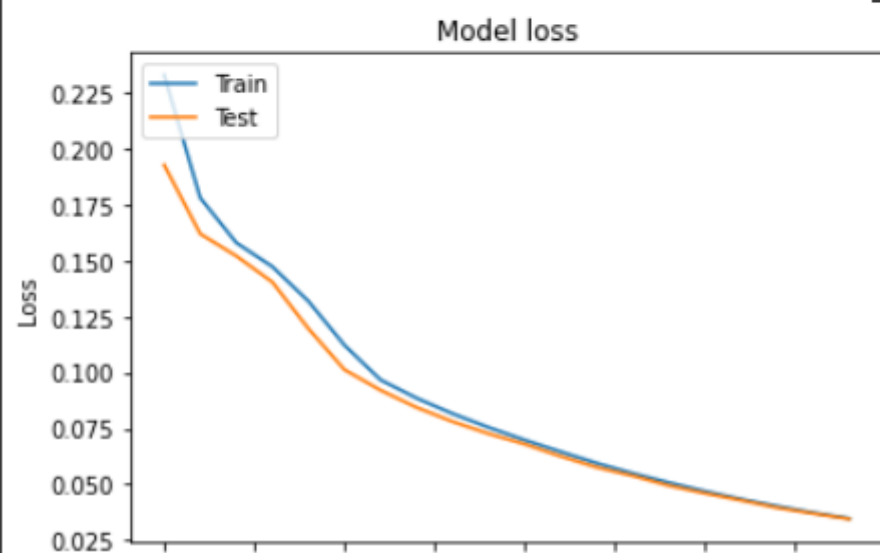
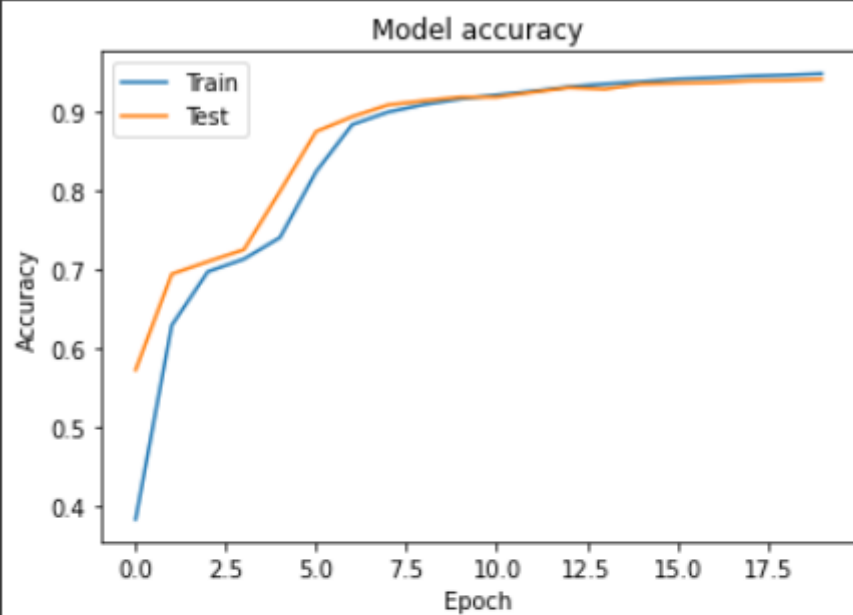
Layer count = 1:

Accuracy: 0.9132999777793884



Layer count = 2:

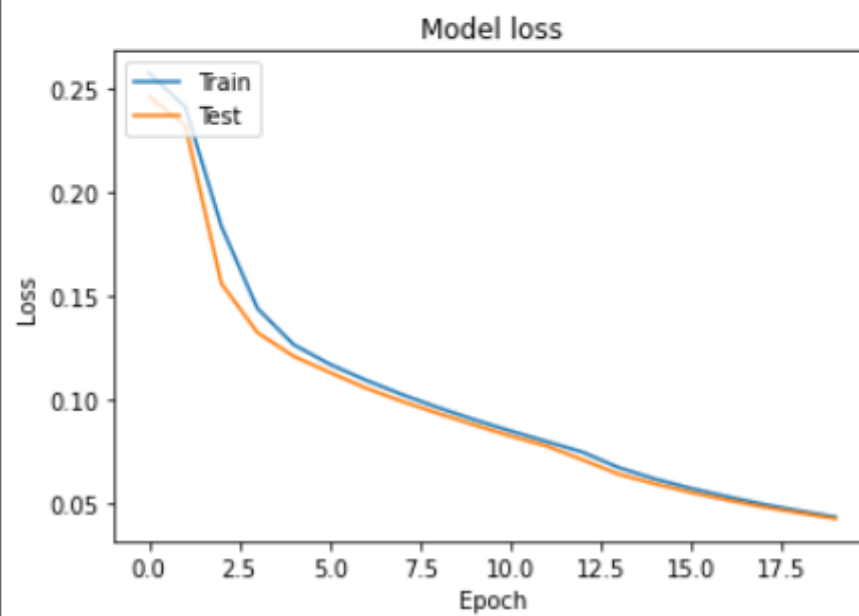
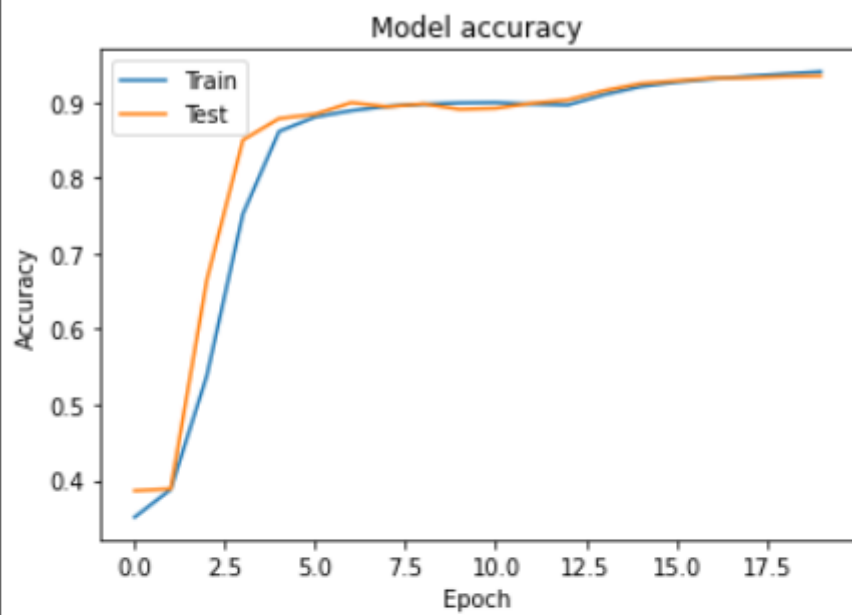
Accuracy: 0.9398999810218811



Layer count = 3:

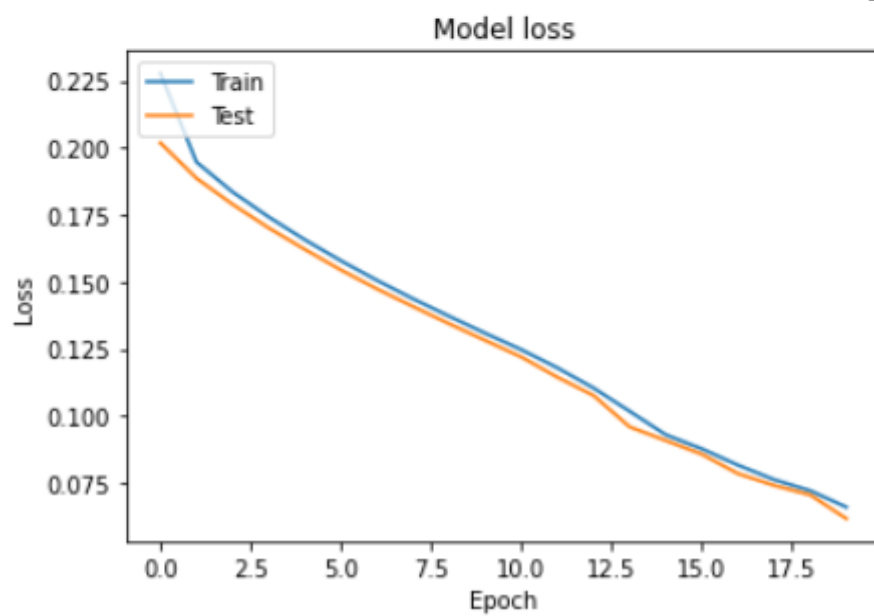
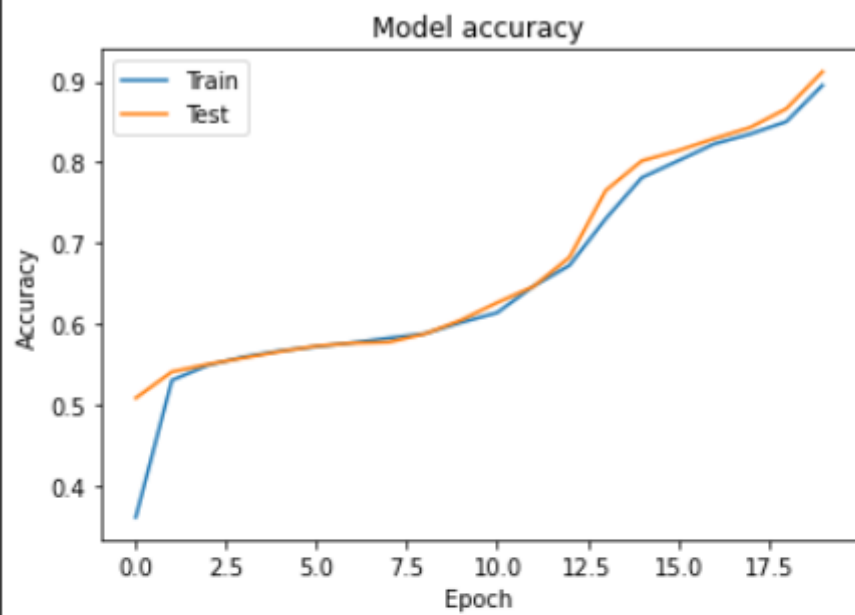


Accuracy: 0.9347000122070312



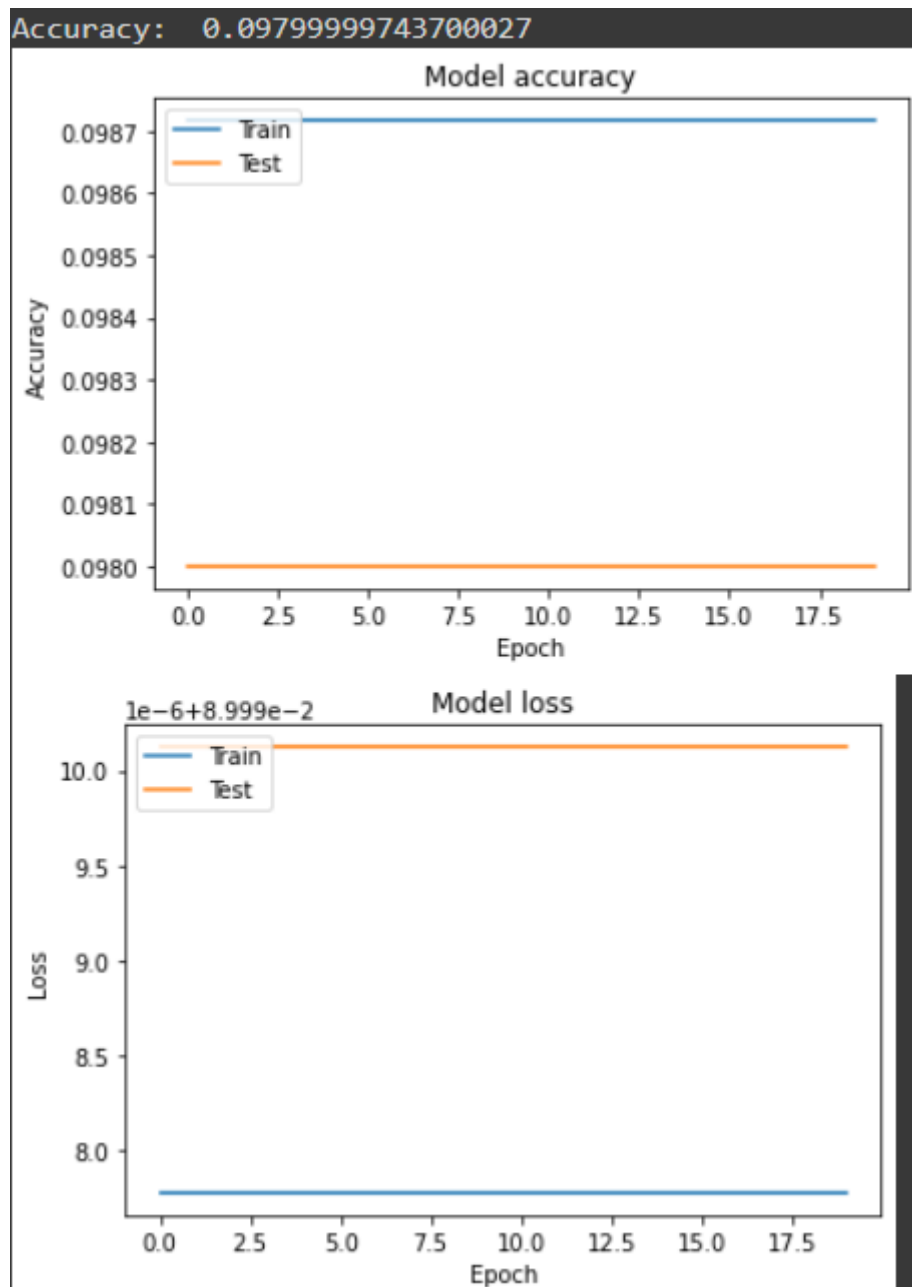
Layer count = 4:

Accuracy: 0.9122999906539917



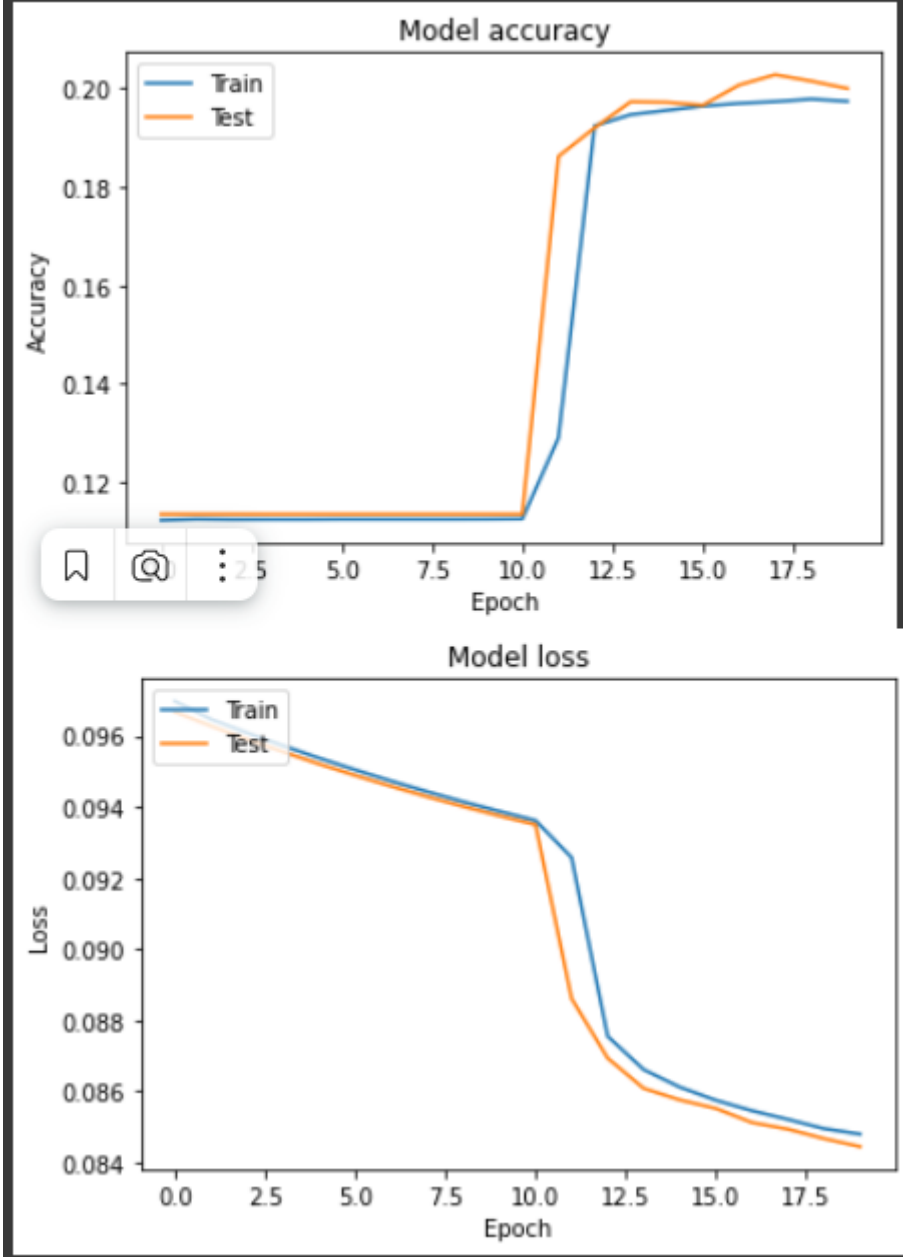
Зависимость от neurons count per:

neurons count per = 0:



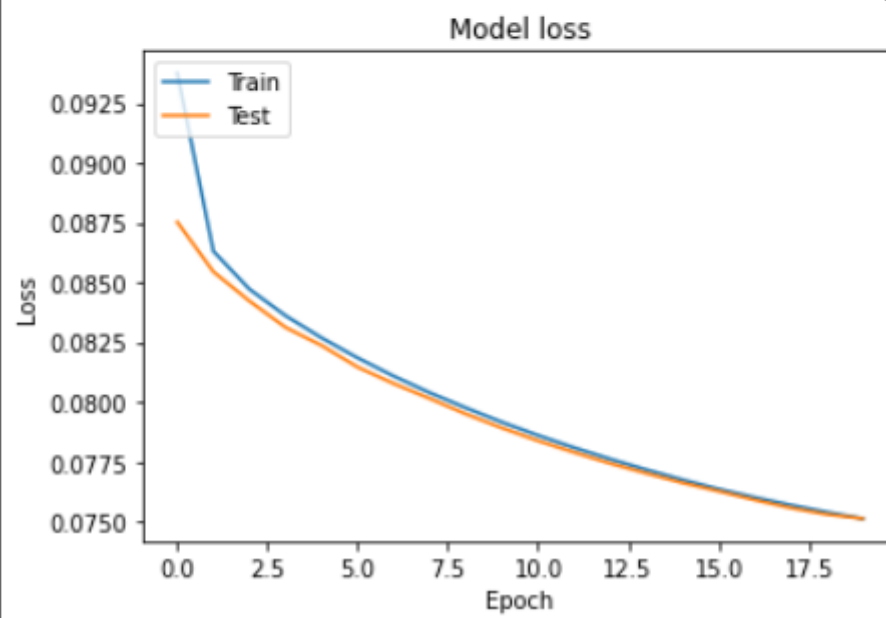
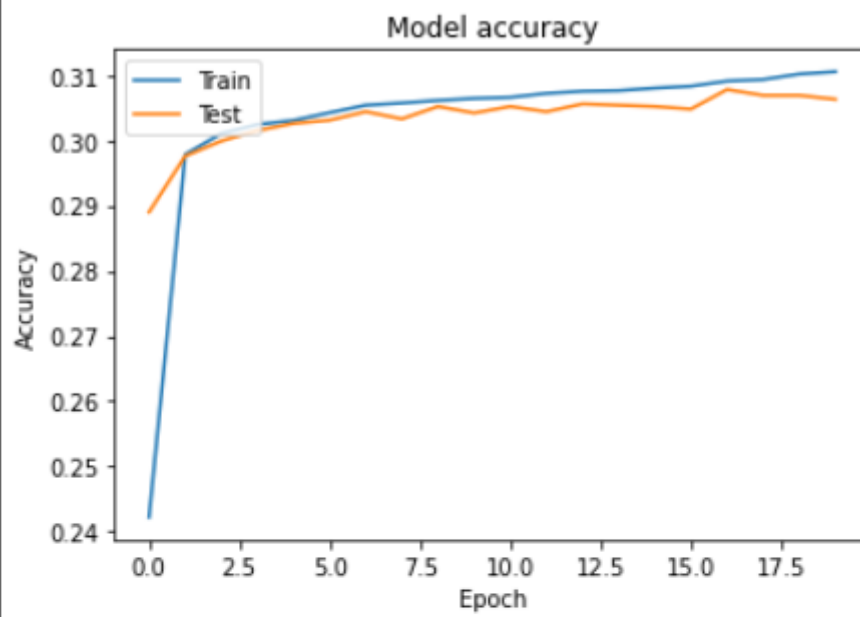
neurons count per = 1:

Accuracy: 0.20000000298023224



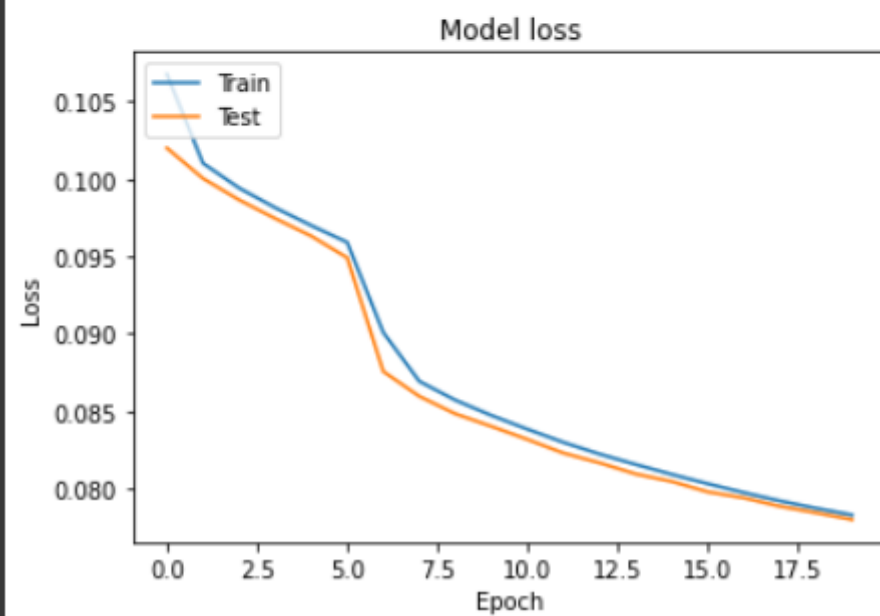
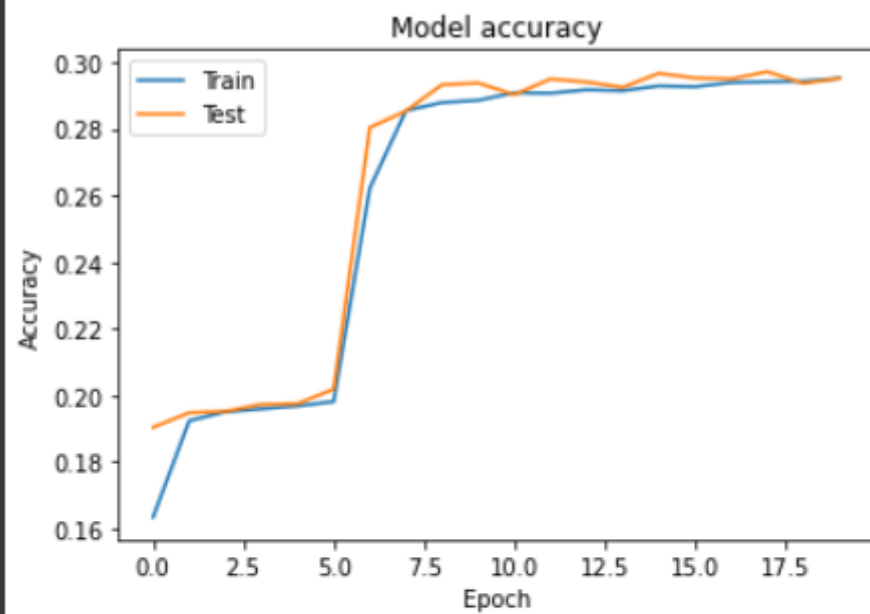
neurons count per = 2:

Accuracy: 0.30630001425743103

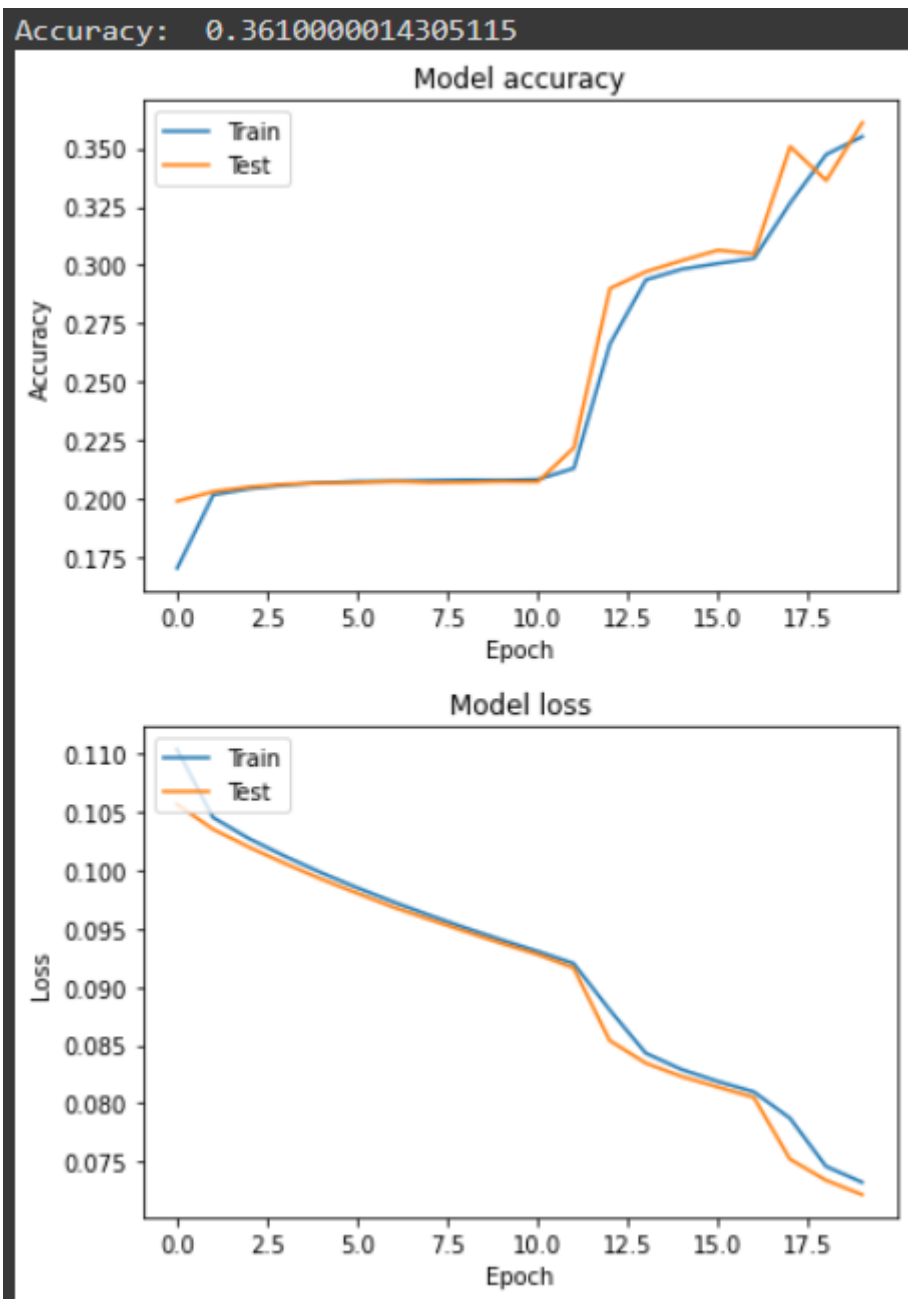


neurons count per = 3:

Accuracy: 0.295199990272522

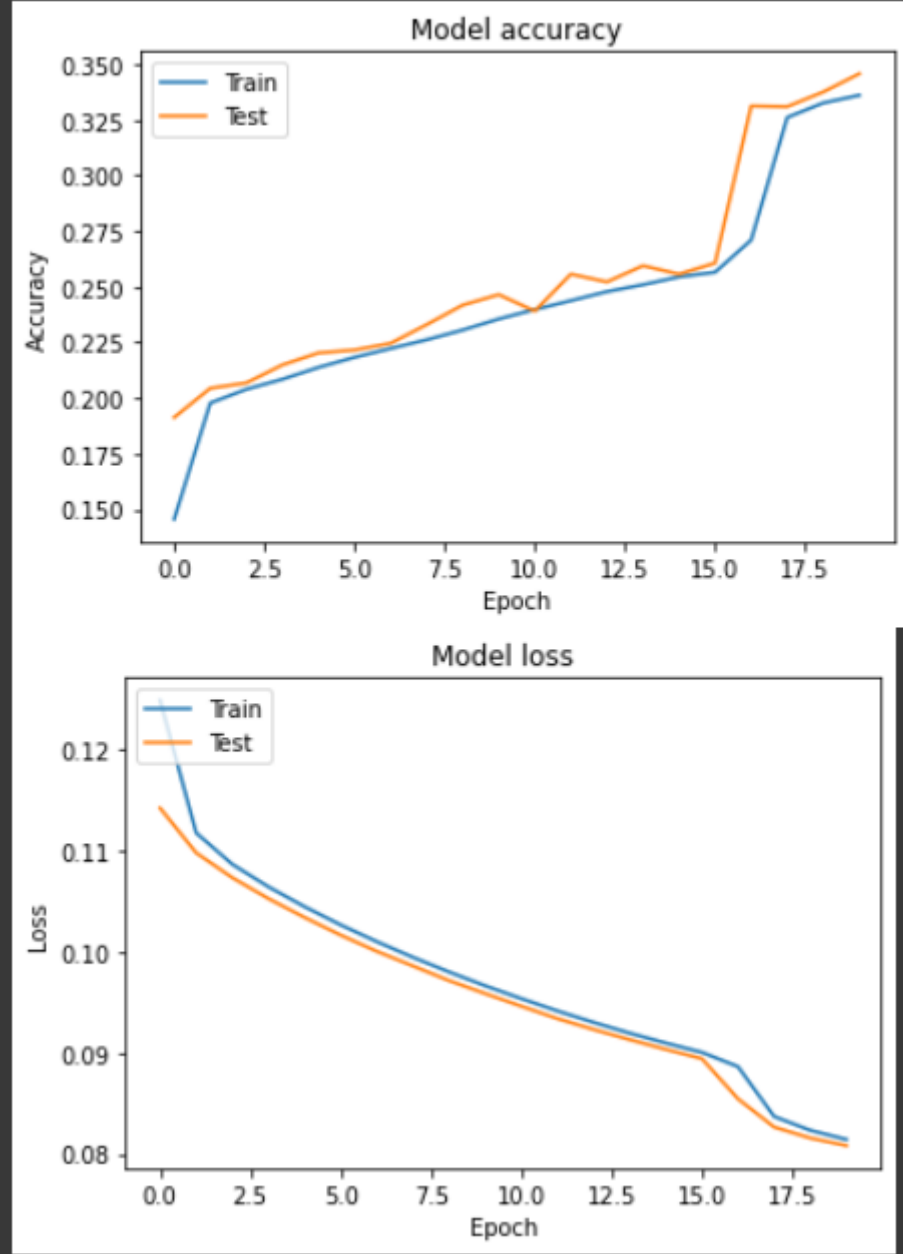


neurons count per = 4:



neurons count per = 5:

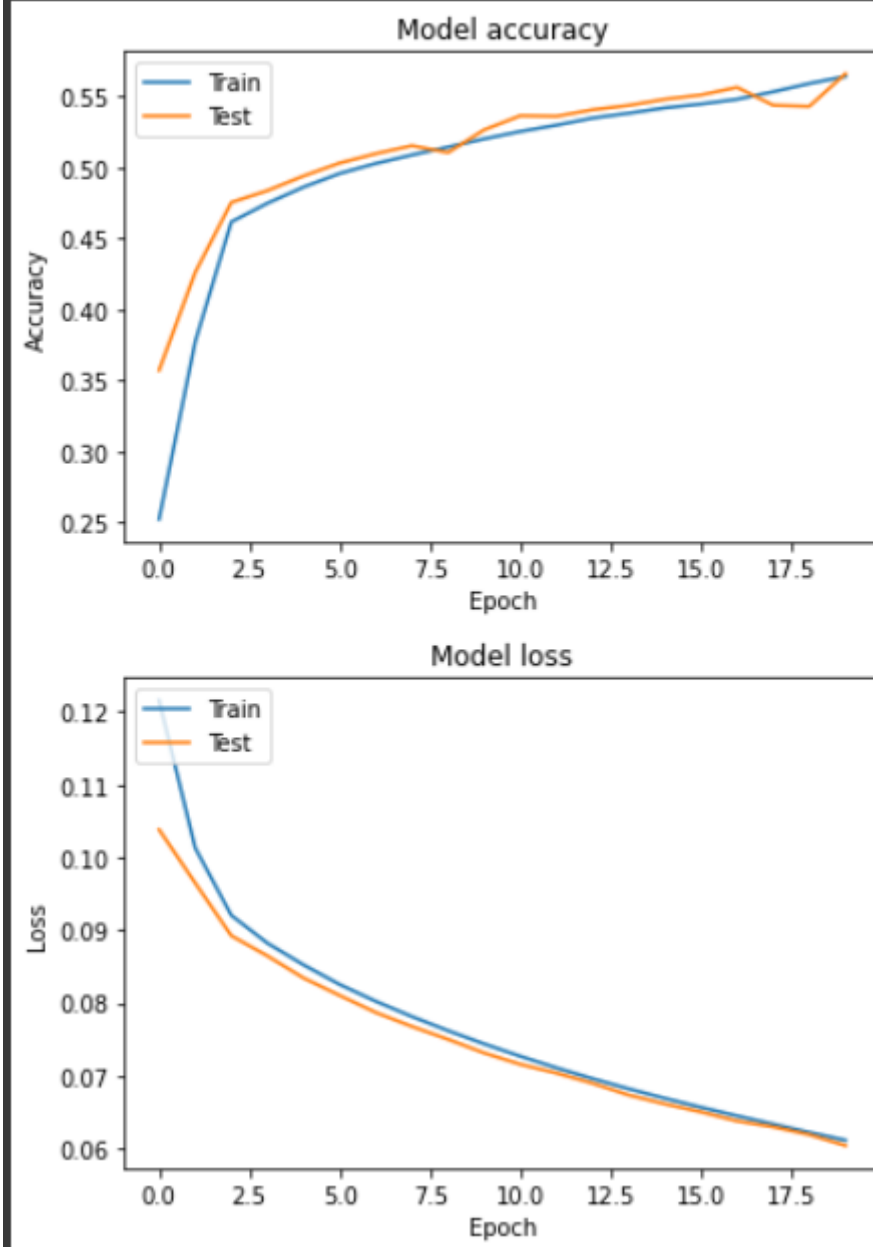
Accuracy: 0.3458000123500824



neurons count per = 6:

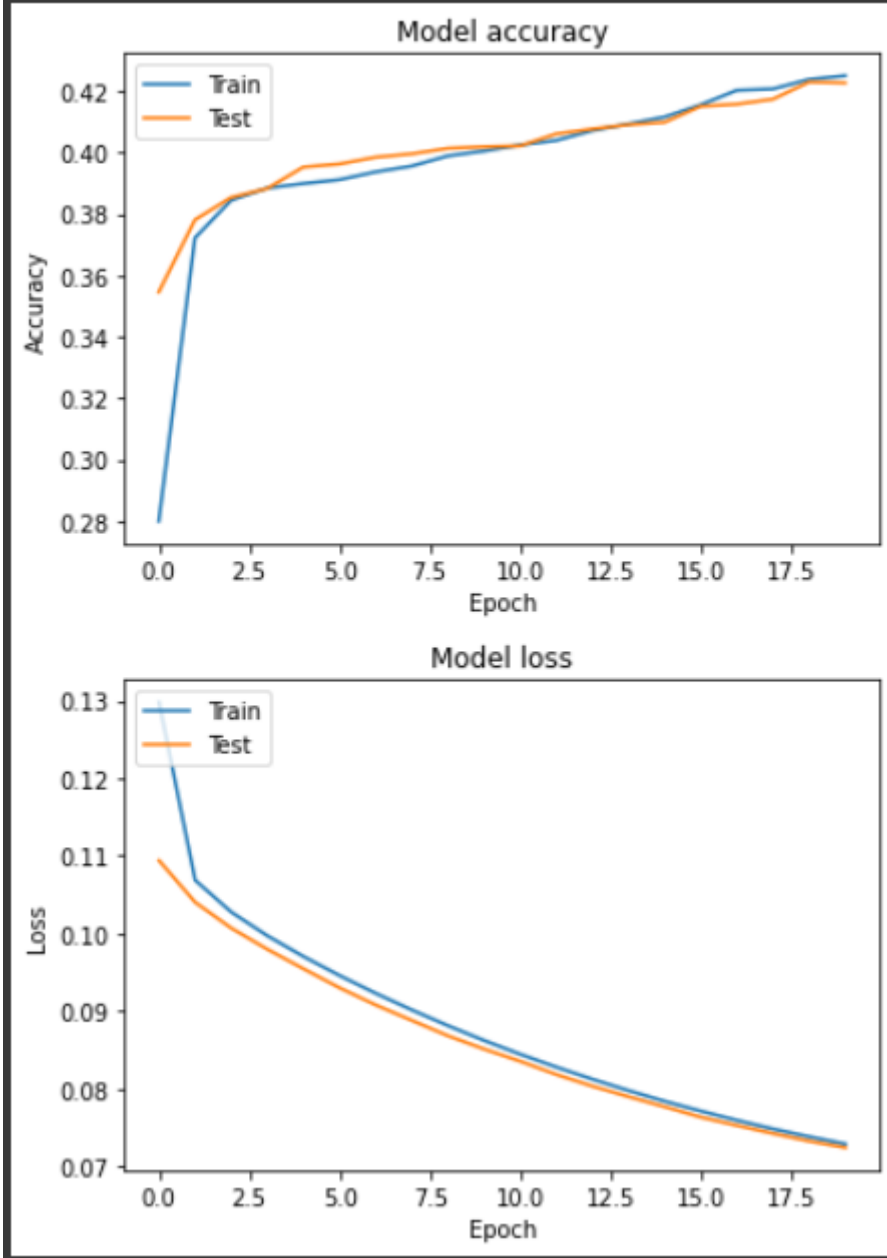


Accuracy: 0.5658000111579895



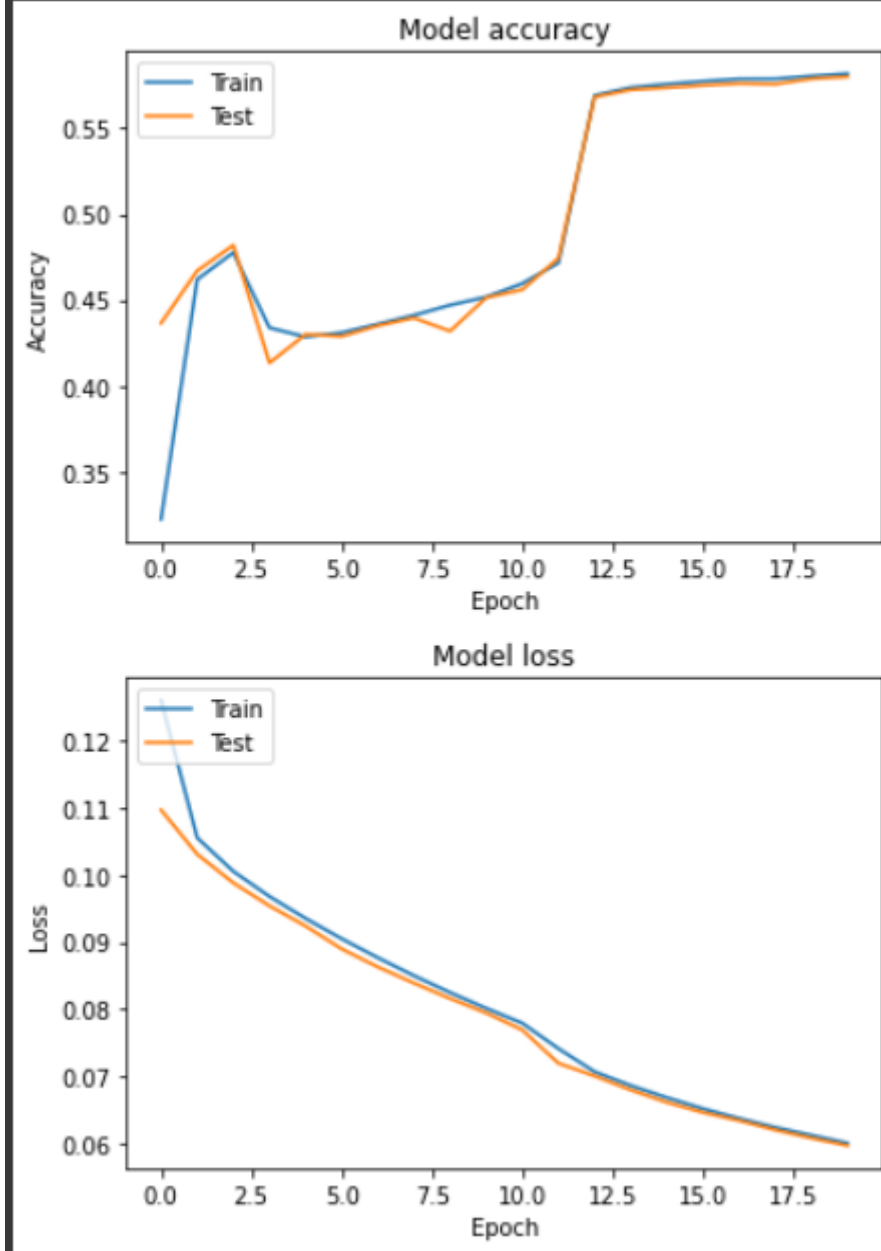
neurons count per = 7:

Accuracy: 0.4226999878883362

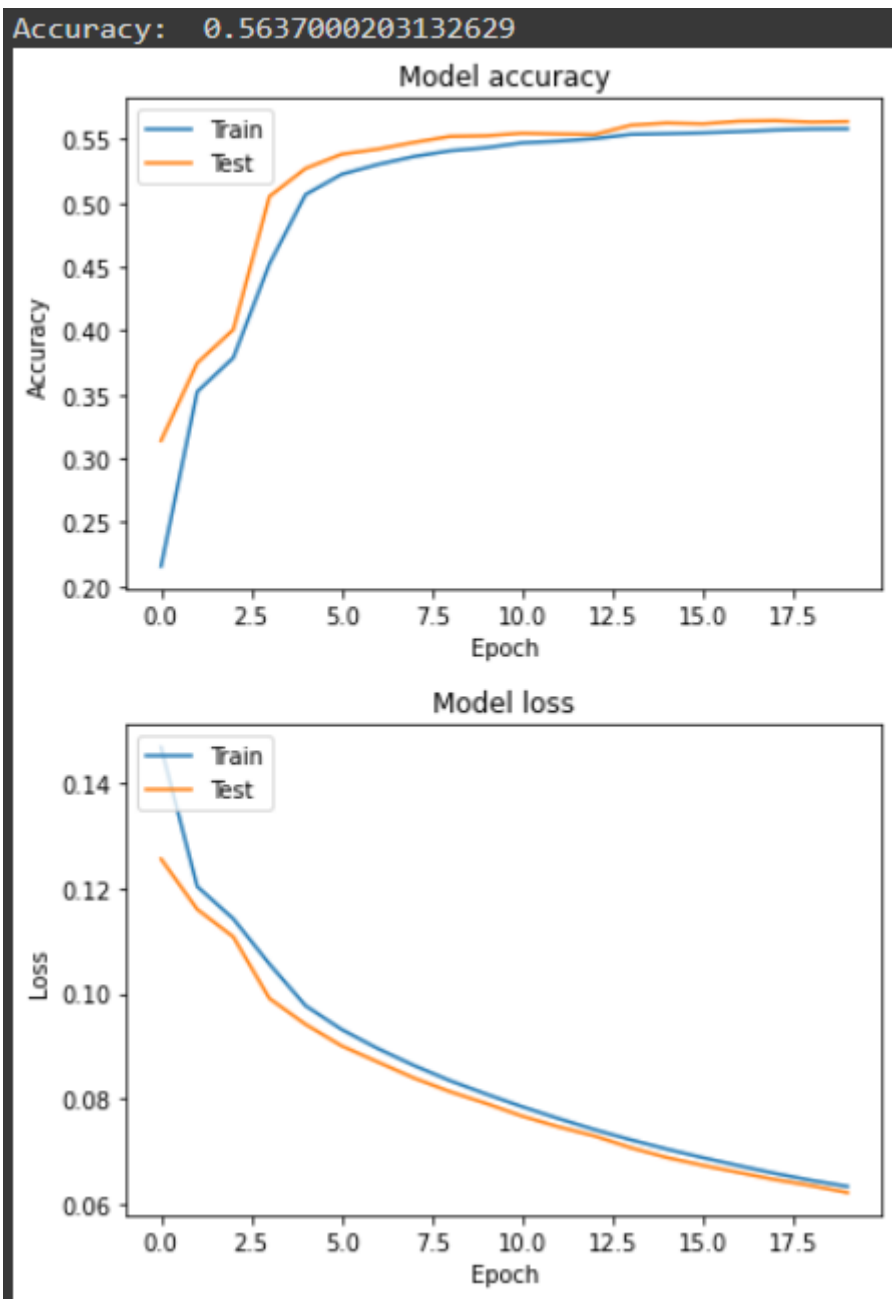


neurons count per = 8:

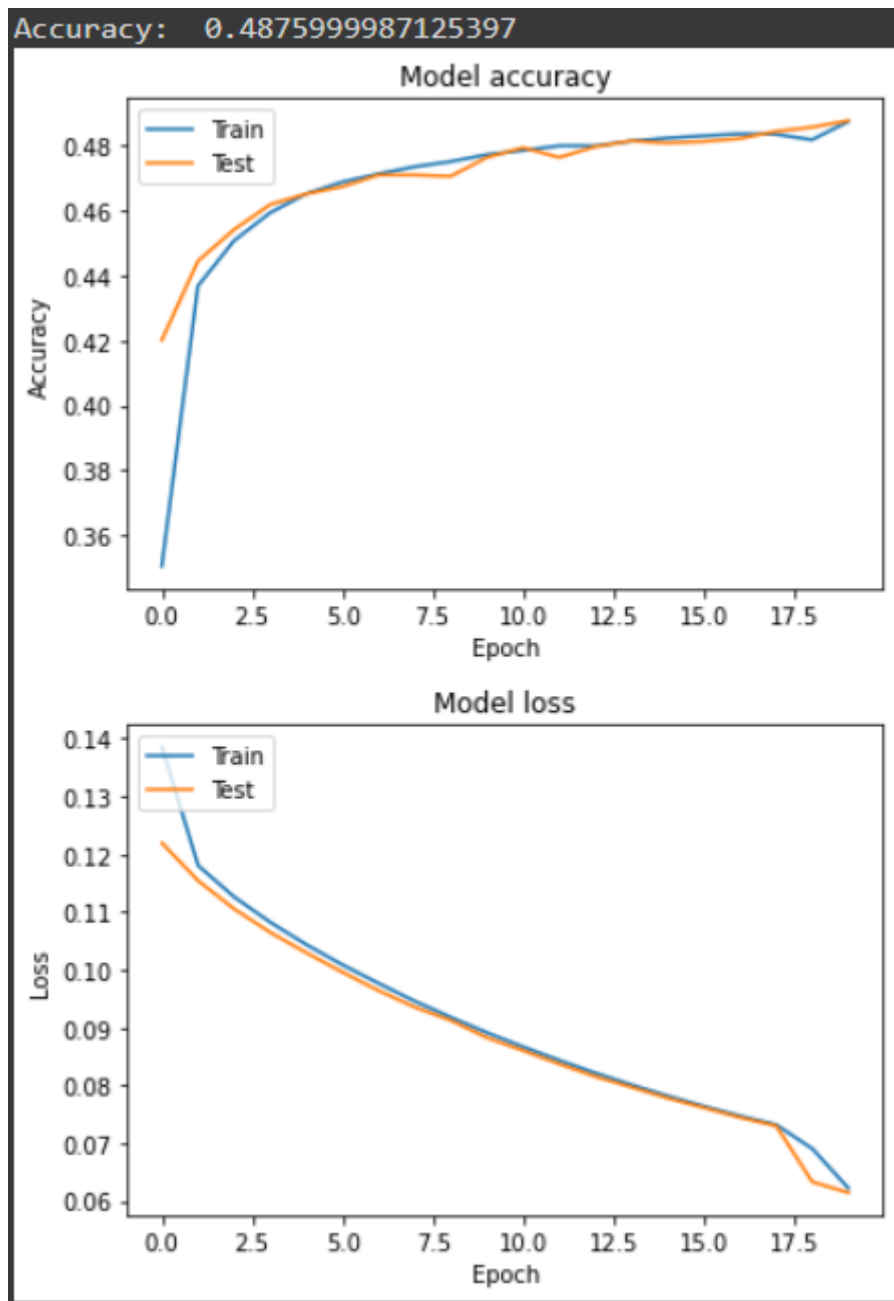
Accuracy: 0.5800999999046326



neurons count per = 9:

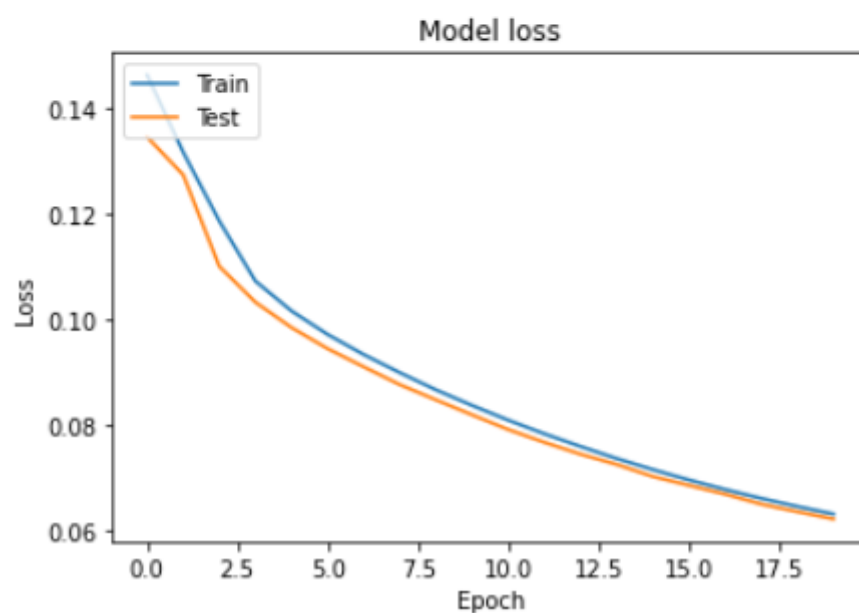
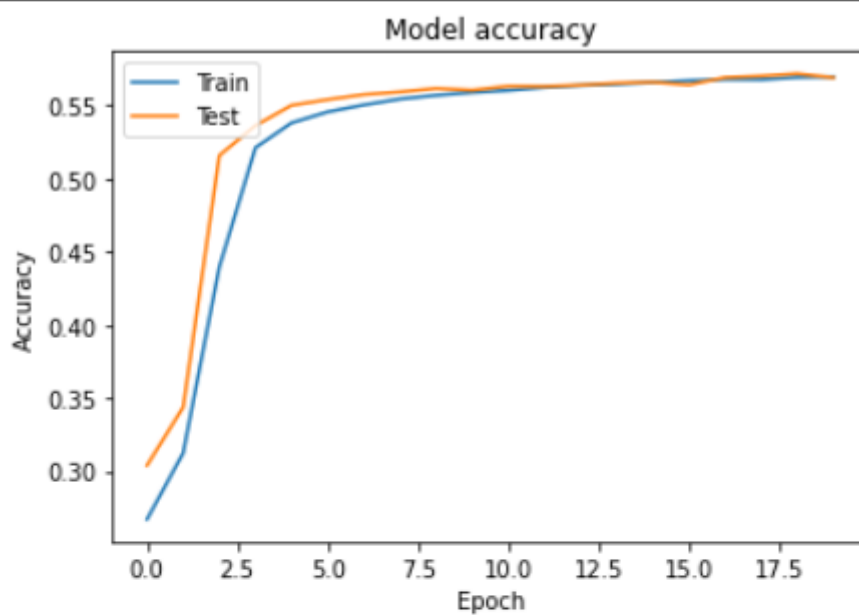


neurons count per = 10:



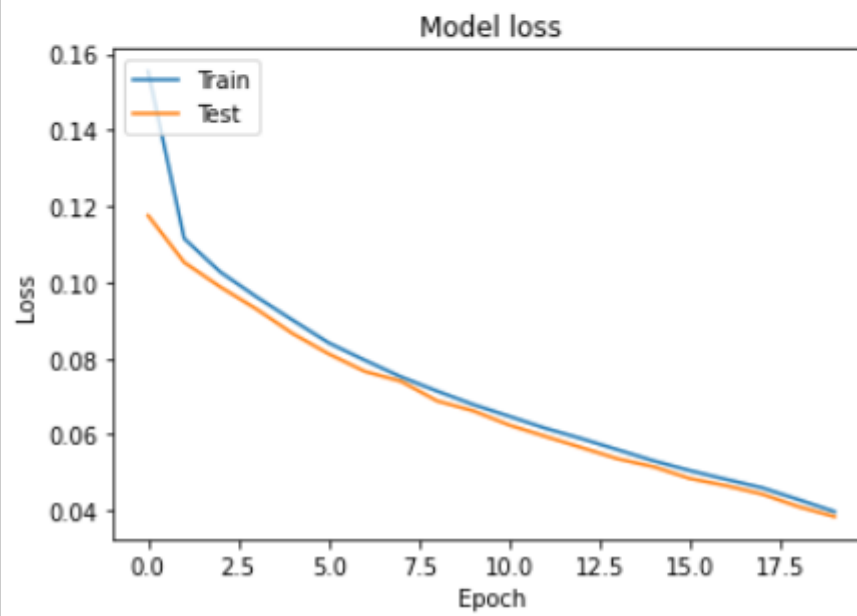
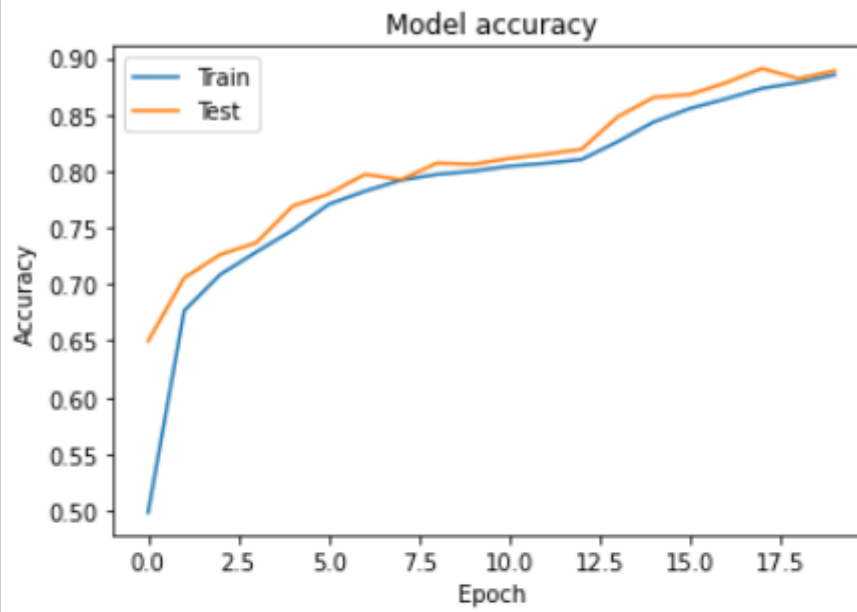
neurons count per = 11:

Accuracy: 0.569100022315979



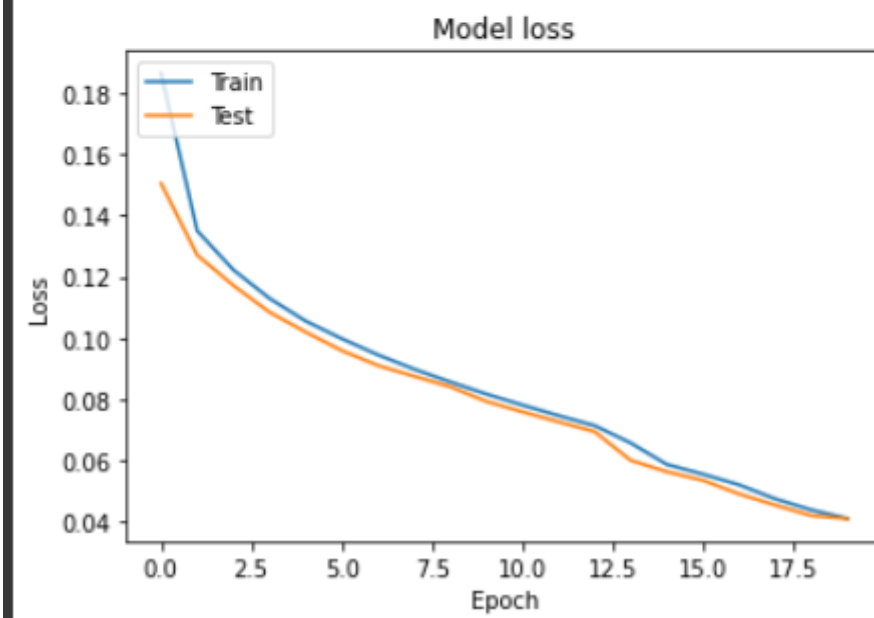
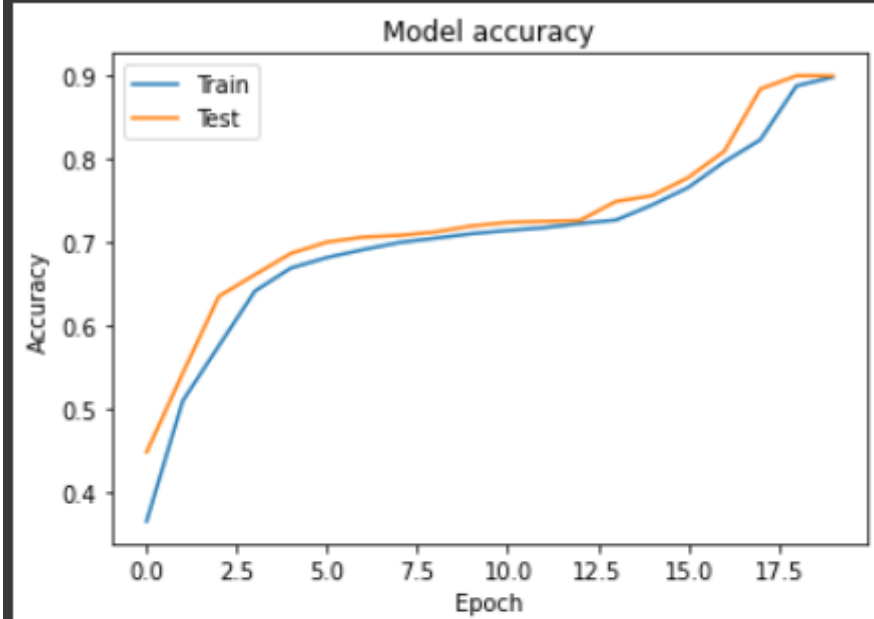
neurons count per = 12:

Accuracy: 0.888700008392334



neurons count per = 13:

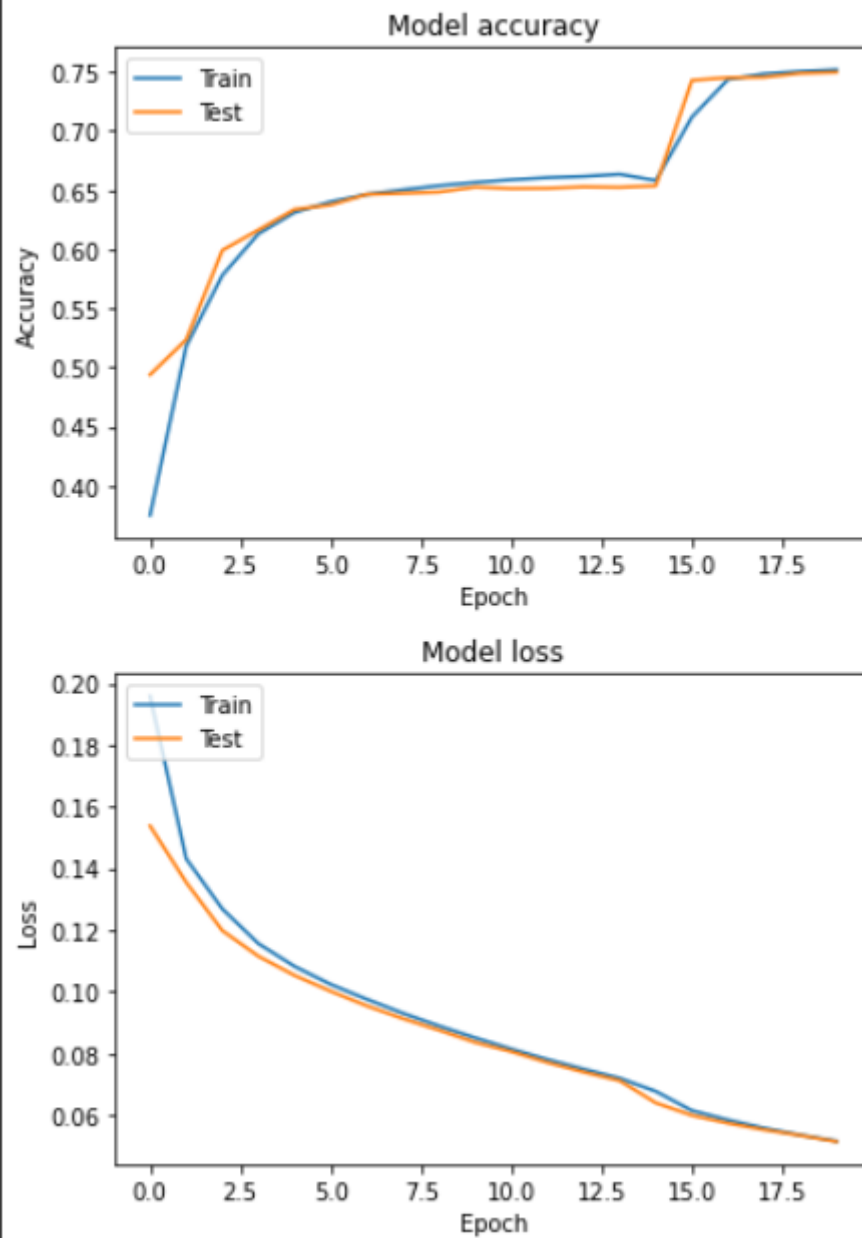
Accuracy: 0.8986999988555908



neurons count per = 14:

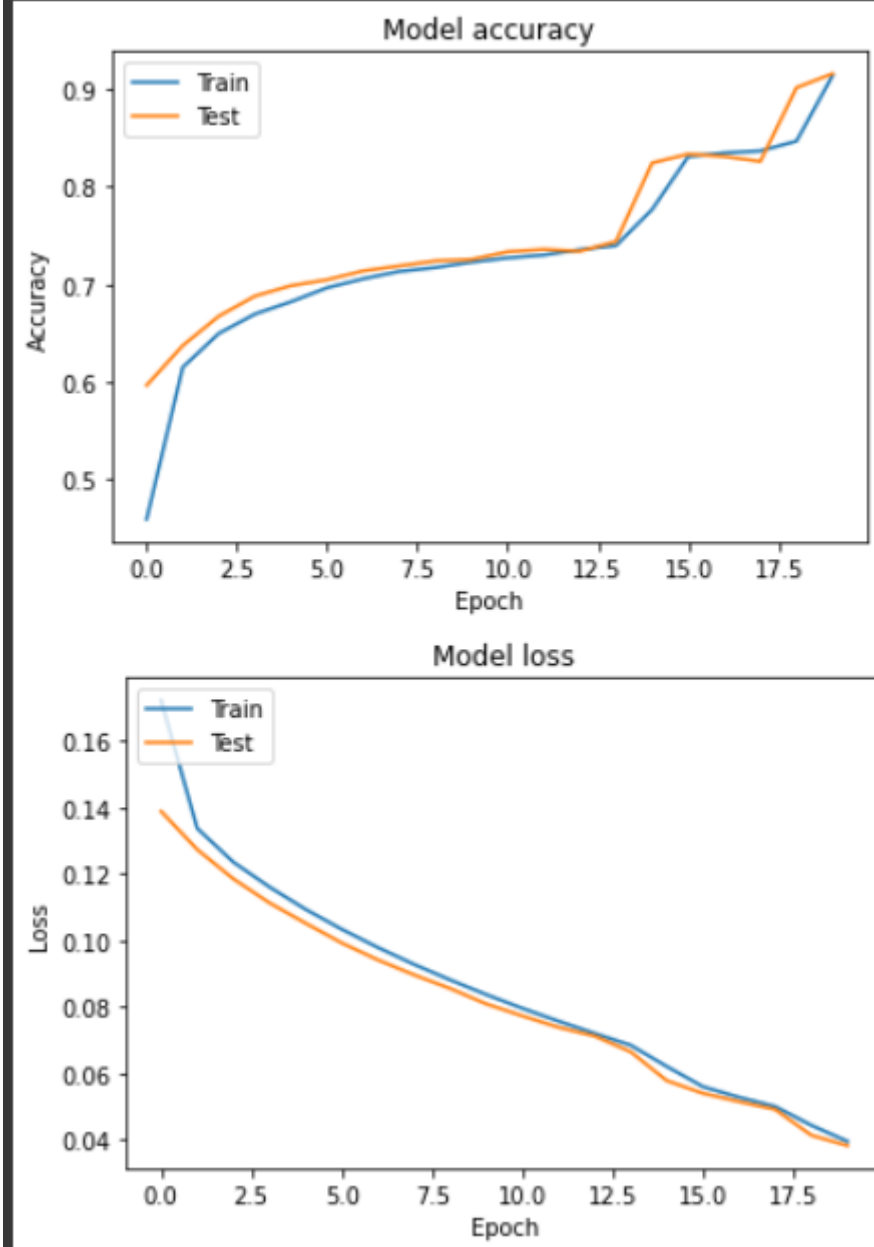


Accuracy: 0.7498000264167786



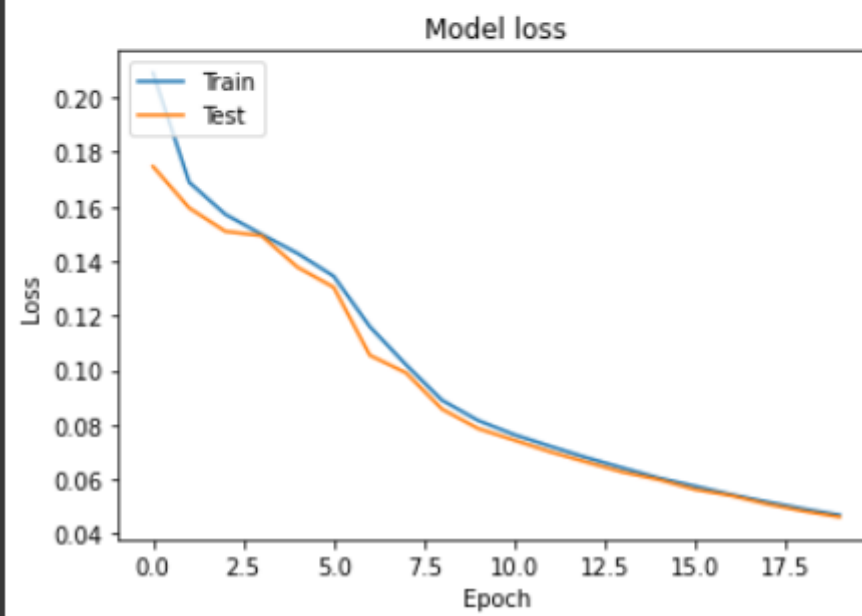
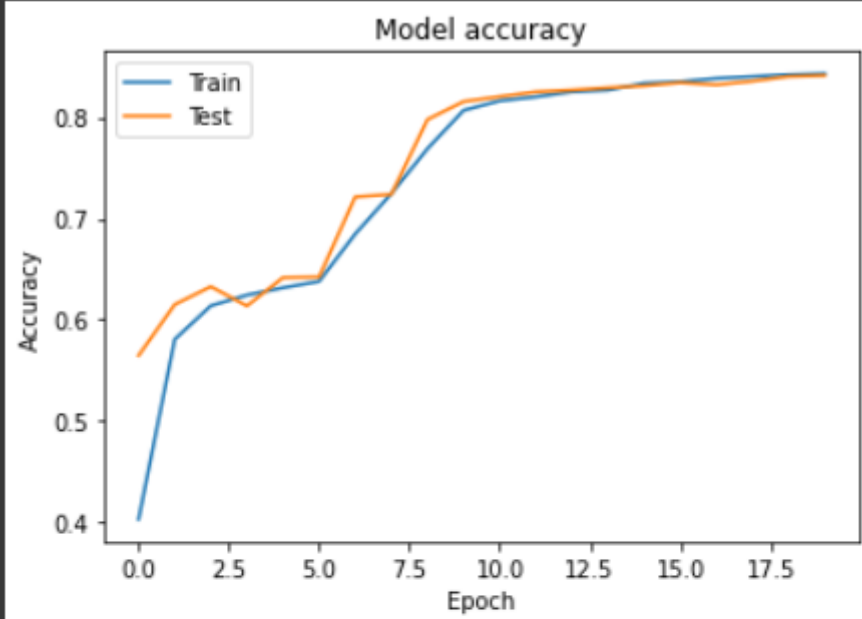
neurons count per = 15:

Accuracy: 0.9161999821662903



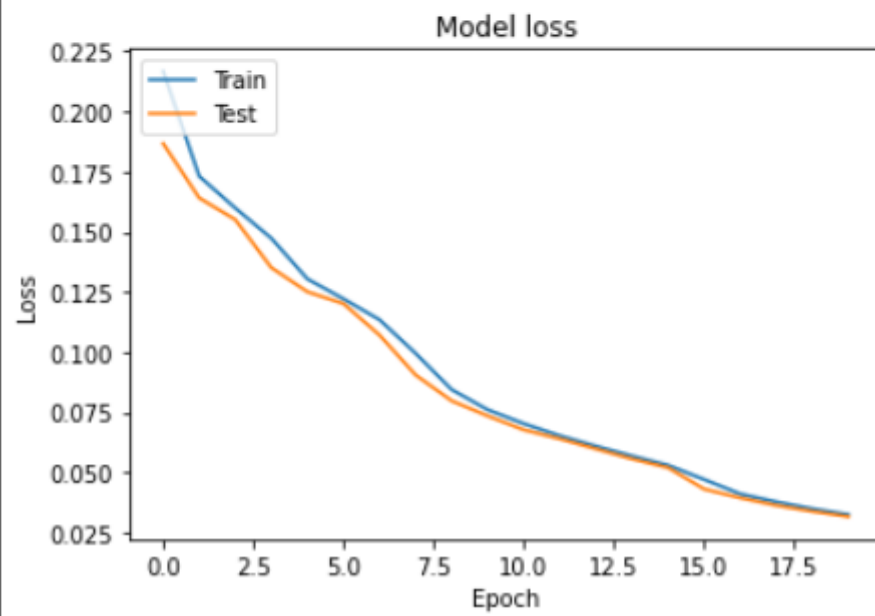
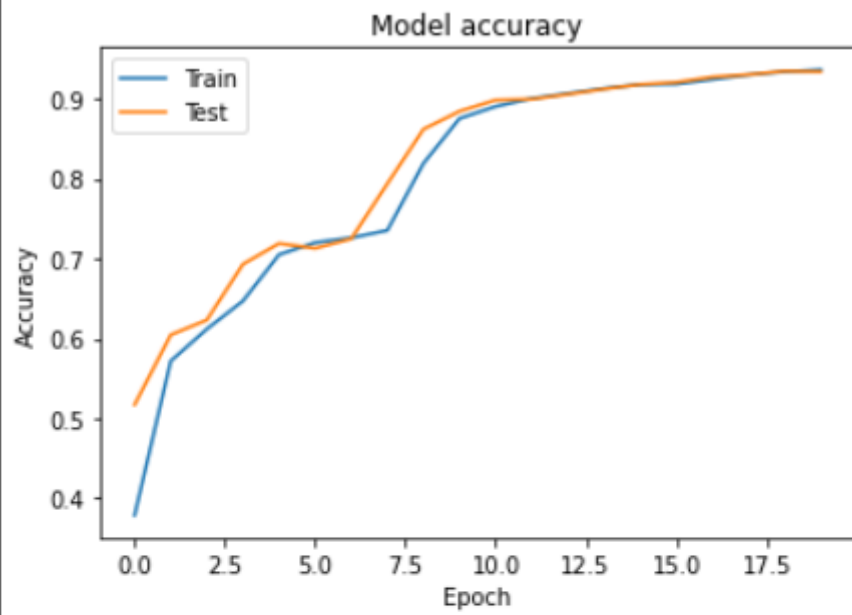
neurons count per = 16:

Accuracy: 0.8411999940872192



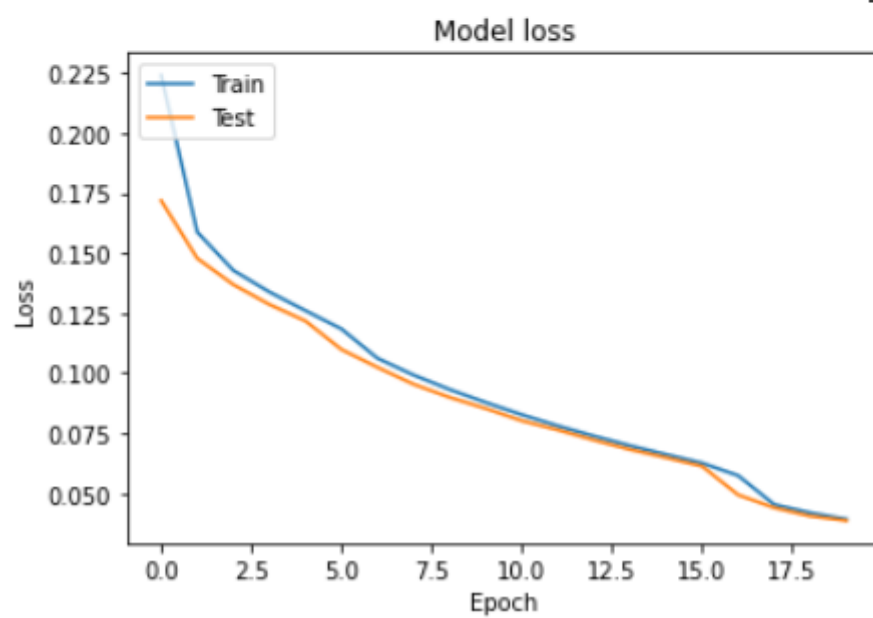
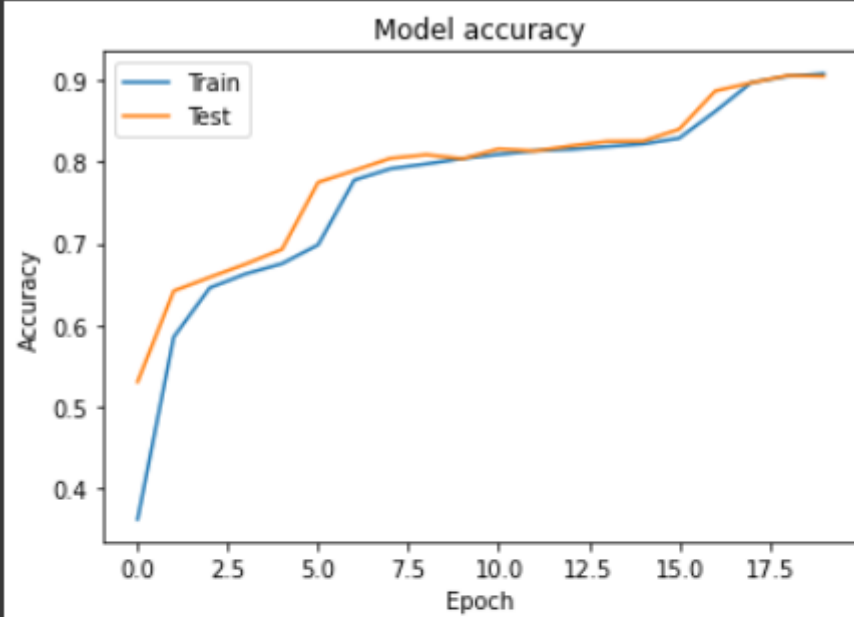
neurons count per = 17:

Accuracy: 0.933899998664856



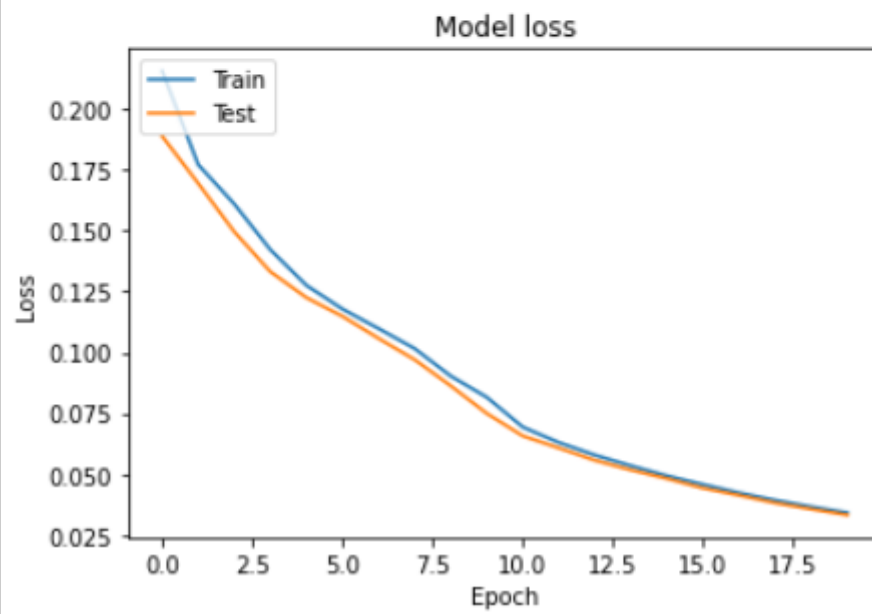
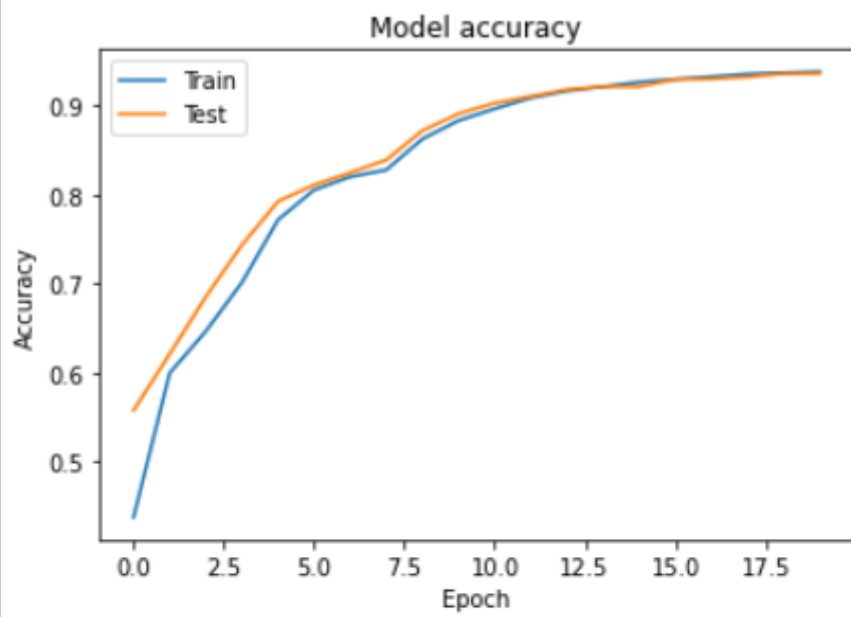
neurons count per = 18:

Accuracy: 0.9050999879837036



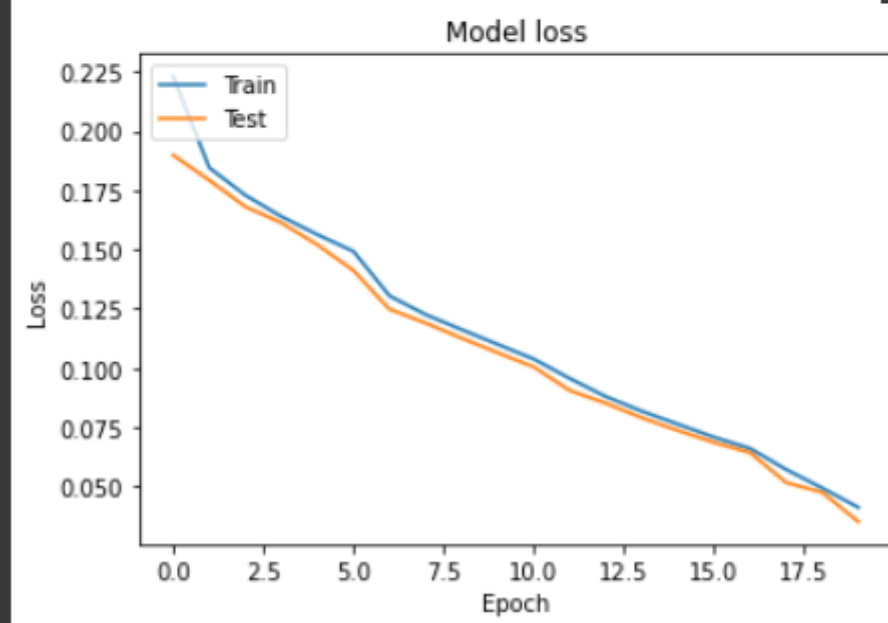
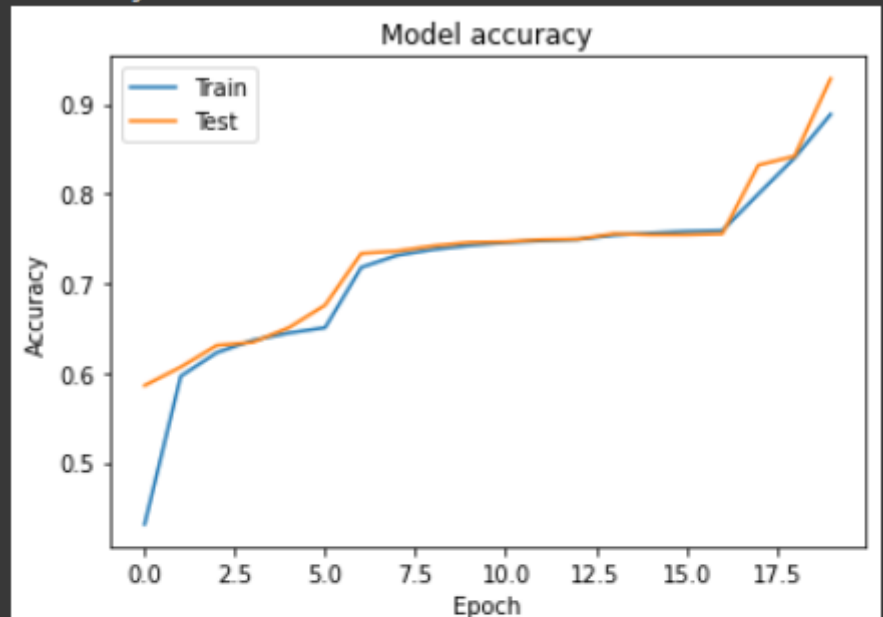
neurons count per = 19:

Accuracy: 0.9358999729156494



neurons count per = 20:

Accuracy: 0.9283999800682068



## Вывод

В ходе данной лабораторной работы я познакомился с обычными нейронными сетями, посмотрел, как влияют различные параметры и функции (Learn rate, Regularization L1 и L2, Output layer activation type, Layer activation type, Loss function type, Epoch count и в особенности Layer count и Neurons count per layer) на результаты достоверности.