

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

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Искусственные нейронные сети»**  
**Тема: «Бинарная классификация отраженных сигналов радара»**

Студентка гр. 7381

Преподаватель

Судакова П.С.

Жукова Н.А.

Санкт-Петербург

2020

### **Цель работы.**

Реализовать классификацию между камнями (R) и металлическими цилиндрами (M) на основе данных об отражении сигналов радара от поверхностей.

60 входных значений показывают силу отражаемого сигнала под определенным углом. Входные данные нормализованы и находятся в промежутке от 0 до 1.

### **Порядок выполнения работы.**

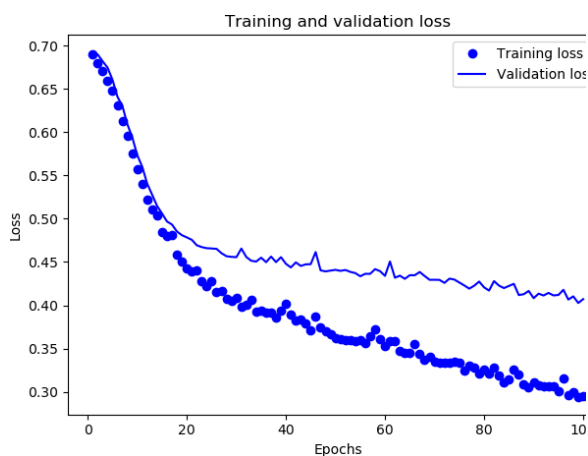
- Ознакомиться с задачей бинарной классификации;
- Загрузить данные;
- Создать модель ИНС в tf.Keras;
- Настроить параметры обучения;
- Обучить и оценить модель;
- Изменить модель и провести сравнение;
- Объяснить результаты.

### **Требования.**

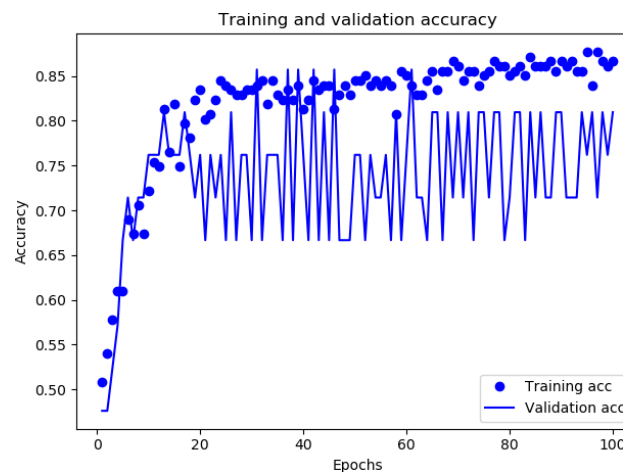
1. Изучить влияние кол-ва нейронов на слое на результат обучения модели;
2. Изучить влияние кол-ва слоев на результат обучения модели;
3. Построить графики ошибки и точности в ходе обучения;
4. Провести сравнение полученных сетей, объяснить результат.

### **Ход работы.**

Результаты обучения нейронной сети при изначальных параметрах представлены на рис. 1.



А

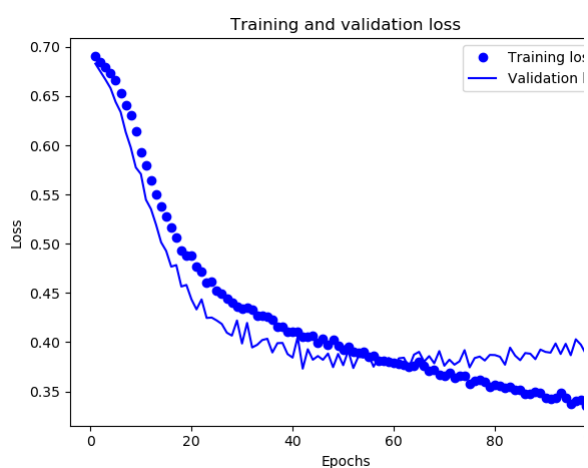


Б

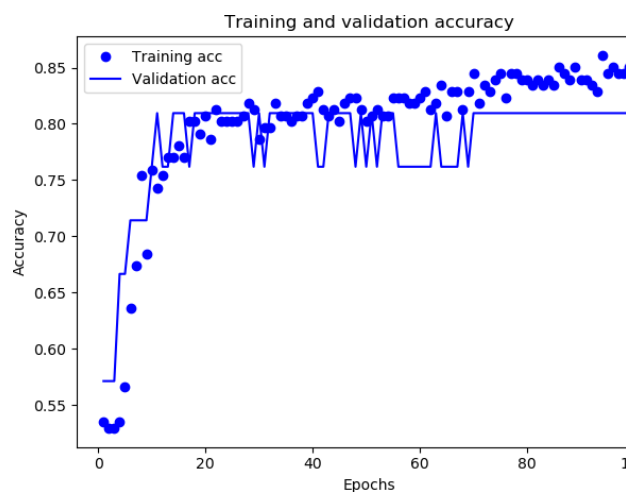
Рисунок 1 – Потери (а) и точность (б) модели с входным слоем из 60 нейронов.

Теперь уменьшим количество нейронов на входном слое в 2 раза.

Результаты представлены на рис. 2.



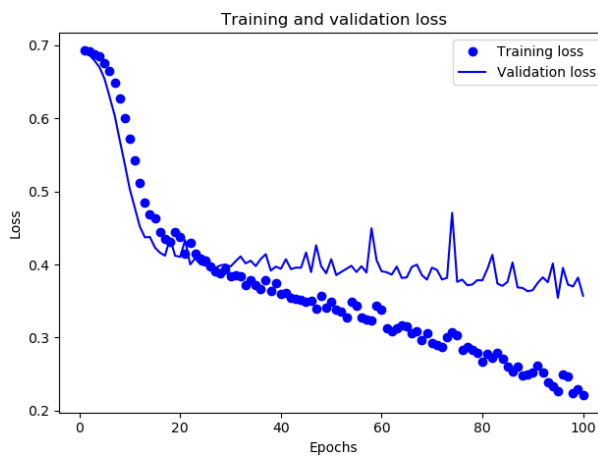
А



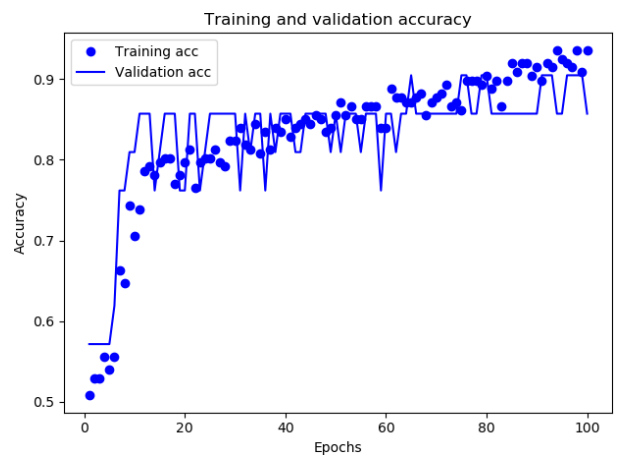
Б

Рисунок 2 – Потери (а) и точность (б) модели с входным слоем на 30 нейронов.

Добавим в сеть второй скрытый слой на 15 нейронов.



а



б

Рисунок 3 – Потери (а) и точность (б) модели с входным слоем на 60 нейронов и скрытым слоем на 15 нейронов.

### Выводы.

В ходе выполнения лабораторной работы была решена задача бинарной классификации.

## ИСХОДНЫЙ КОД

```
import pandas
from tensorflow.keras.layers
import Dense from
tensorflow.keras.models import
Sequential
from tensorflow.keras.utils import
to_categorical from
sklearn.preprocessing import
LabelEncoder

import matplotlib.pyplot as plt

dataframe = pandas.read_csv("sonar.csv",
header=None) dataframe =
dataframe.sample(frac=1)
dataset = dataframe.values

print(dataset)

X =
dataset[:,0:60].astype(fl
oat) Y = dataset[:,60]

encoder =
LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)

model = Sequential()
model.add(Dense(60, input_dim=60, kernel_initializer='normal',
activation='relu')) model.add(Dense(15,
kernel_initializer='normal', activation='relu'))
model.add(Dense(1, kernel_initializer='normal',
activation='sigmoid'))

model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

H = model.fit(X, encoded_Y, epochs=100, batch_size=10,
validation_split=0.1)

# Получение ошибки и точности в
процессе обучения loss =
H.history['loss']
val_loss =
H.history['val_loss']
acc = H.history['acc']
val_acc =
H.history['val_acc']
epochs = range(1,
```

```
len(loss) + 1)
```

```
# Построение графика ошибки
plt.plot(epochs, loss, 'bo',
label='Training loss') plt.plot(epochs,
val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
# Построение графика
точности plt.clf()
plt.plot(epochs, acc, 'bo',
label='Training acc') plt.plot(epochs,
val_acc, 'b', label='Validation acc')
plt.title('Training and validation
accuracy') plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend(
(
```