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

# ОТЧЕТ

**по лабораторной работе №2**

**по дисциплине «Искусственные нейронные сети»**

**Тема: «Бинарная классификация отраженных сигналов радара»**

Студентка гр. 8382 _____ Звегинцева Е.Н.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург

2021

**Цели.**

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.

**Задачи.**

- 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.**

- Изучить влияние количества нейронов на слое на результат обучения модели
- Изучить влияние количества слоев на результат обучения модели
- Построить графики ошибки и точности в ходе обучения
- Провести сравнение полученных сетей, объяснить результат

**Ход работы.**

1) Ознакомилась с задачей бинарной классификации, загрузила данные. Создала и обучила модель ИНС в tf.Keras в соответствии с условиями (код представлен в приложении А).

2) При исследовании различных архитектур и обучении ИНС при разных параметрах необходимо было:

- уменьшить размер входного слоя в два раза и сравнить с результатами первоначальной архитектуры
- добавить промежуточный (скрытый) слой Dense в архитектуру сети с 15 нейронами и проанализировать результаты

Результаты обучения ИНС при изначальных параметрах представлены на рис. 1.

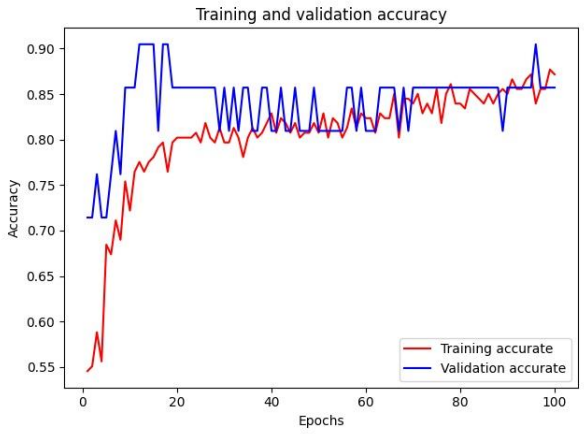
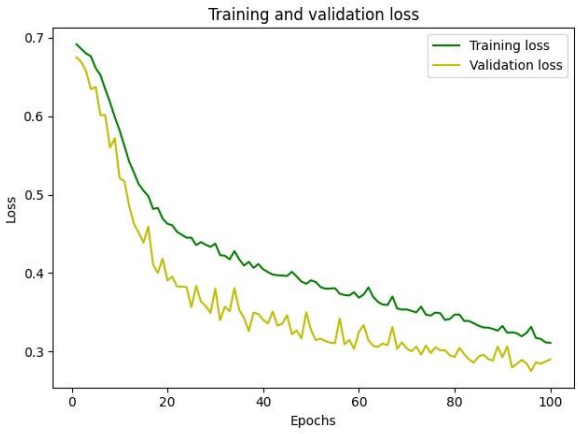
| Графики точности | Графики потерь |
|---|---|
|  |  |



Рисунок 1 – Точность и потери модели со входным слоем из 60 нейронов

Теперь уменьшим количество нейронов на входном слое в 2 раза. Результаты представлены на рис. 2.

Высокая точность была достигнута быстрее в 1ом случае. Итоговая точность в обеих сетях составляет 85%, потери с уменьшением нейронов увеличились, во втором случае точность в 85% была достигнута позднее.

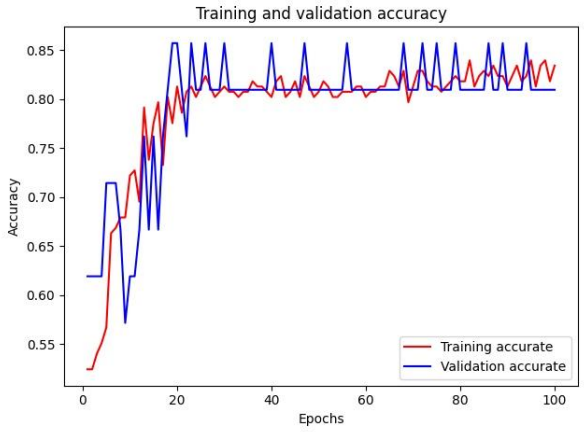
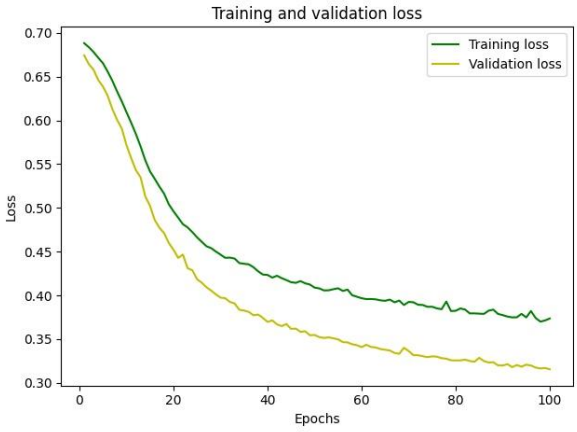
| Графики точности | Графики потерь |
| --- | --- |
|  |  |



Рисунок 2 – Точность и потери модели со входным слоем из 30 нейронов

Теперь добавим в сеть скрытый слой на 15 нейронов. Результаты представлены на рис. 3 и рис.4. Точность свыше 90% была быстрее достигнута у модели со слоями на 60 и 15 нейронов. Потери незначительно увеличились у модели со слоями на 30 и 15 нейронов.

Если сравнивать модели с одним и двумя слоями, где первый слой на 60 нейронов, то при добавлении 2го слоя была быстрее достигнута и стабильней высокая точность 90%, потери же увеличились с появлением 2го слоя на 5-10%.

Если сравнивать модели с одним и двумя слоями, где первый слой на 30 нейронов, то при добавлении 2го слоя была быстрее достигнута и стабильней высокая точность свыше 90%, потери же увеличились с появлением 2го слоя на 5%. По графику можно сделать вывод, что может произойти переобучение модели.

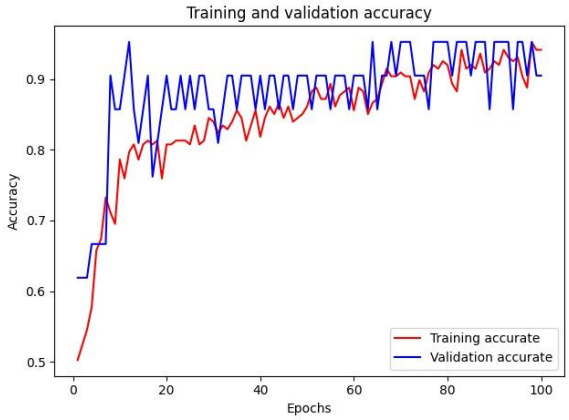
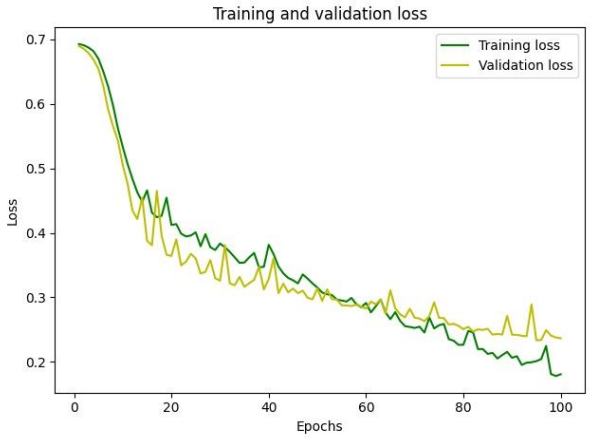
| Графики точности | Графики потерь |
| --- | --- |
|  |  |



Рисунок 3 – Точность и потери модели с 2 слоями на 60 и 15 нейронов

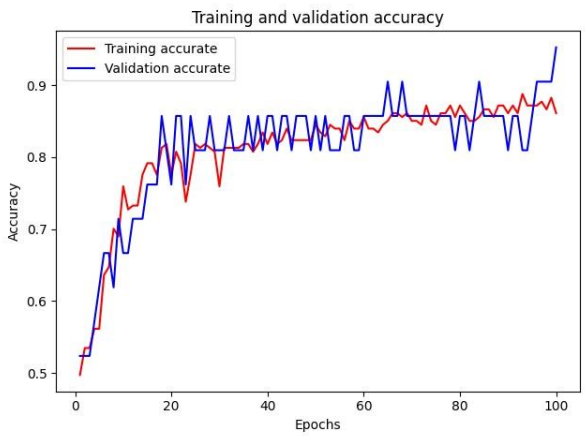
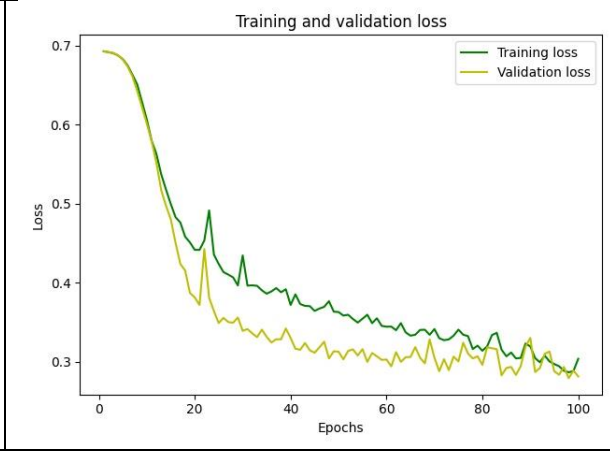
| Графики точности | Графики потерь |
| --- | --- |
|  |  |



Рисунок 4 – Точность и потери модели с 2 слоями на 30 и 15 нейронов

3) Изучим влияние количества слоев на результат обучения модели

Таблица 1 – Графики точности для разного количества скрытых слоев

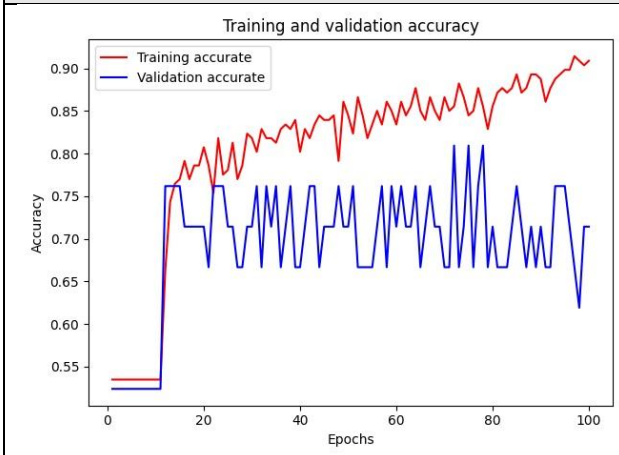
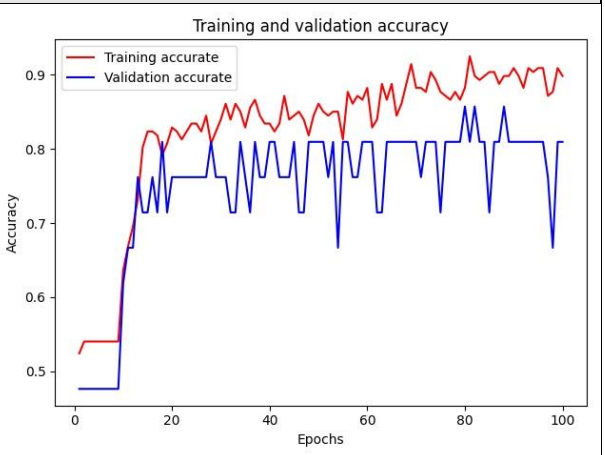
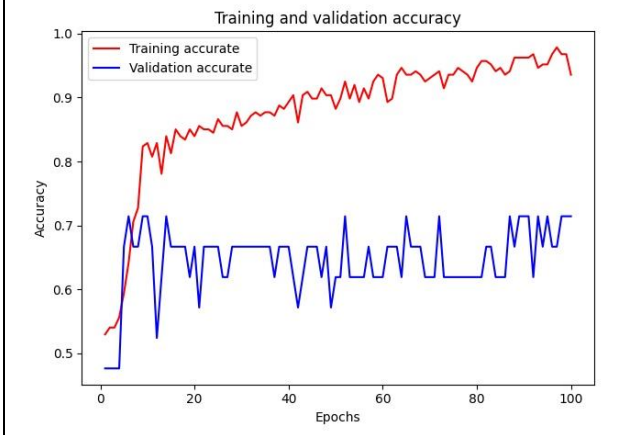
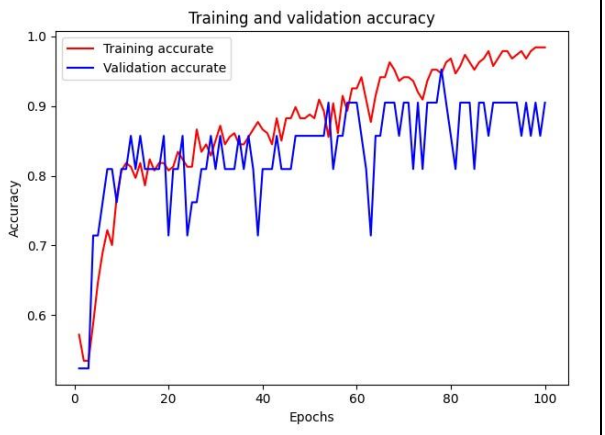
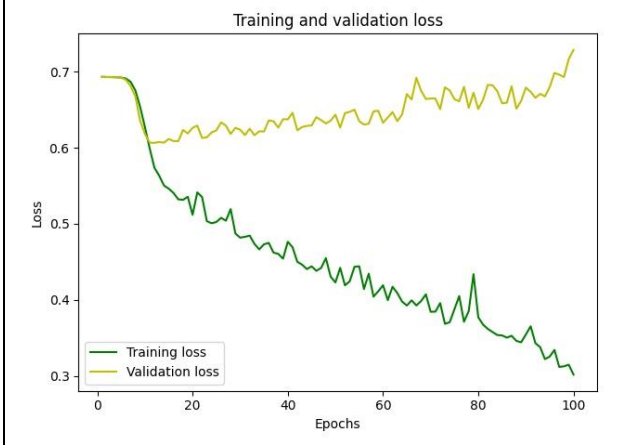
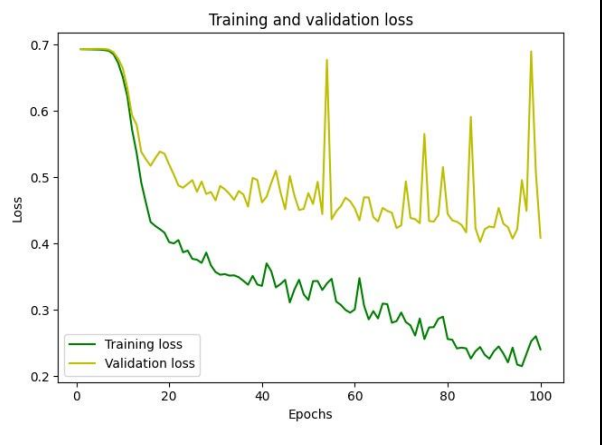
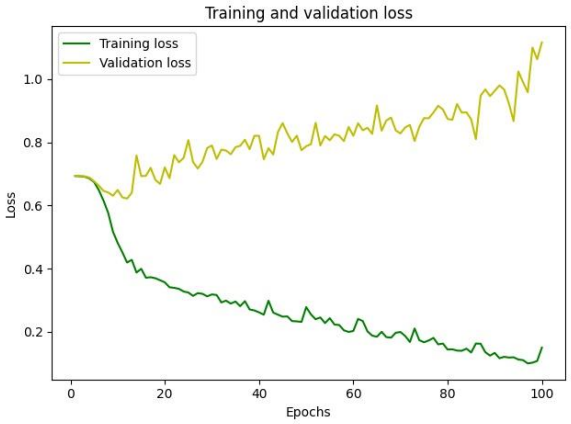
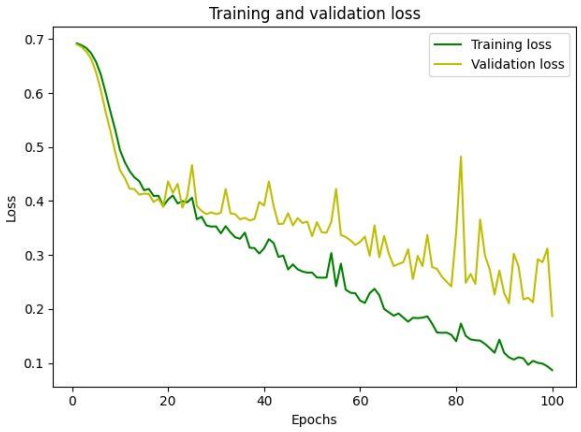
| 4 | 3 |
|---|---|
|  |  |
| 2 | 1 |
|  |  |



Таблица 2 – Графики потерь для разного количества скрытых слоев

| 4 | 3 |
|---|---|
|  |  |

| 2 | 1 |
| --- | --- |
|  |  |





Анализируя полученные графики, можно сделать вывод о том, что с введением скрытых слоев, появляется эффект переобучения сети, в связи с избыточной сложностью модели для конкретной задачи. Это отражается в меньшей точности и большей ошибки на тестовых данных в сравнении с обучаемыми.

4) Изучить влияние количества нейронов на слое на результат обучения модели

Так как выгодней оказалась модель с одним скрытым слоем, попробуем поменять количество нейронов на нем

Таблица 3 – Графики точности для разного количества нейронов

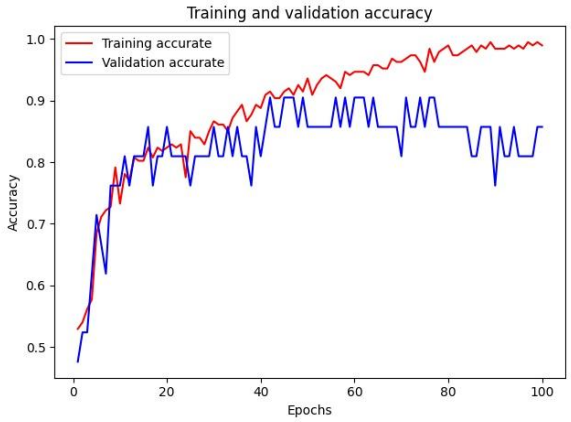
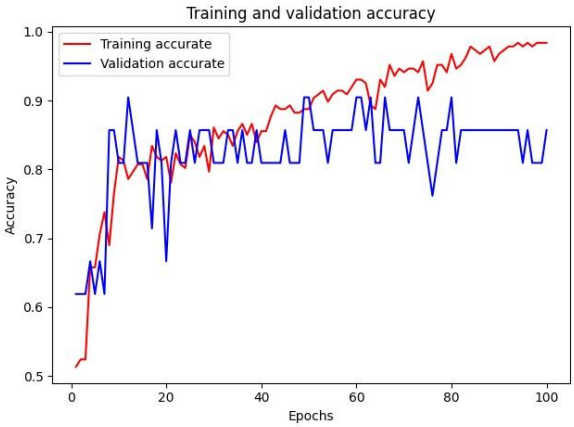
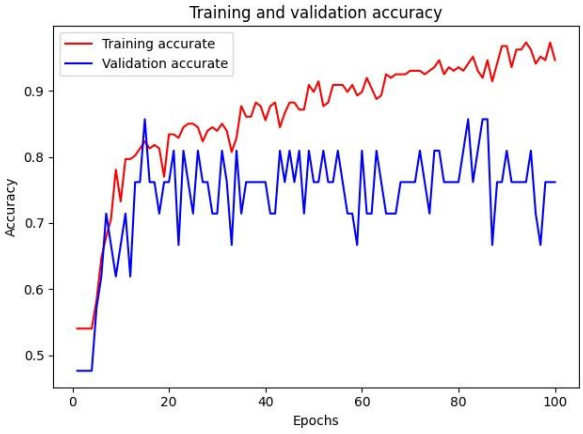
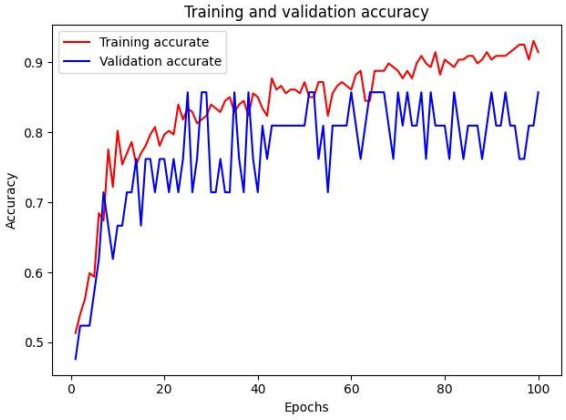
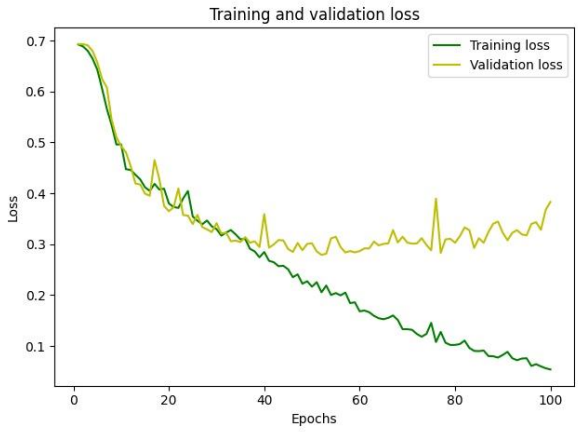
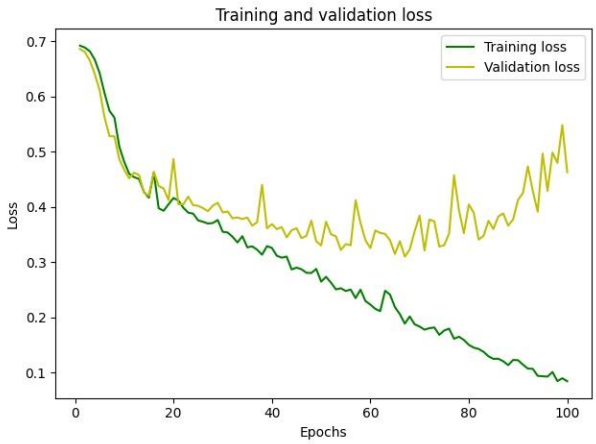
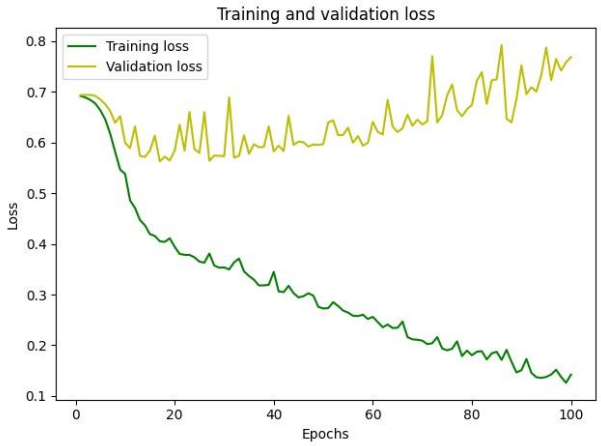
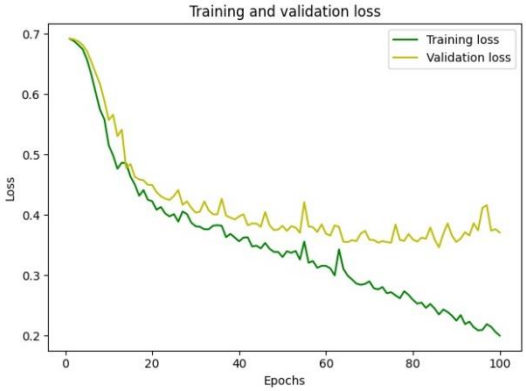
| 40 | 30 |
|---|---|
|  |  |
| **20** | **15** |
|  |  |

Таблица 4 – Графики потерь для разного количества нейронов

| 40 | 30 |
|---|---|
|  |  |
| **20** | **15** |
|  |  |



Анализируя полученные графики, можно сделать вывод о том, что появляется эффект переобучения сети, он уменьшается с уменьшением количества нейронов, это происходит в связи с избыточной сложностью модели для конкретной задачи. Это отражается в меньшей точности и большей ошибки на тестовых данных в сравнении с обучаемыми.

Рассмотрим еще разное количество нейронов на входном слое. Скрытые слои добавлять не будем.

Таблица 5 – Графики точности для разного количества нейронов

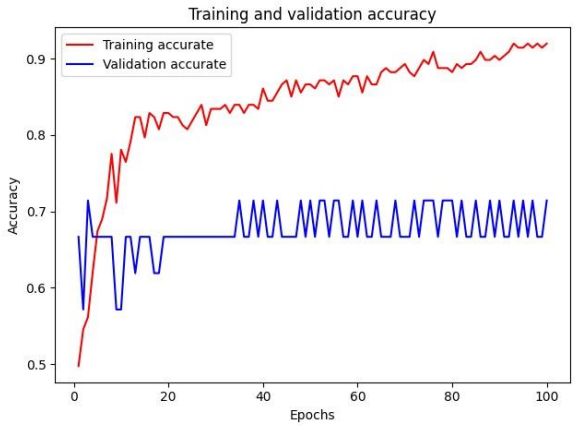
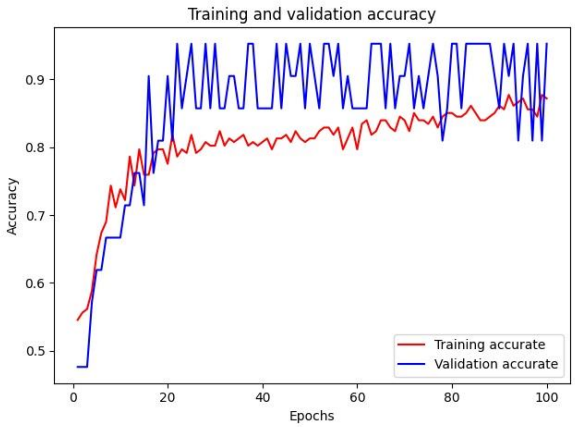
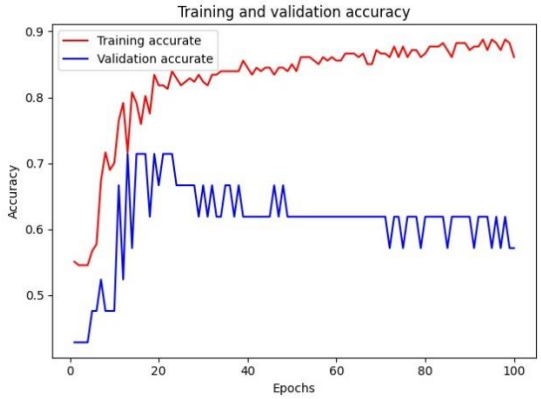
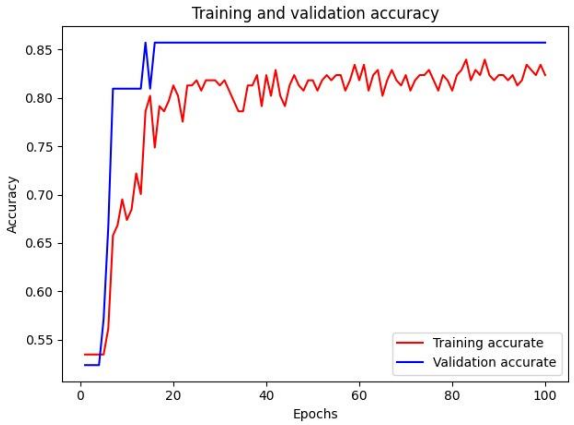
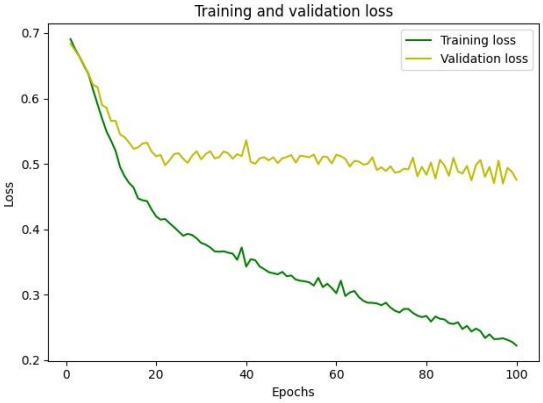
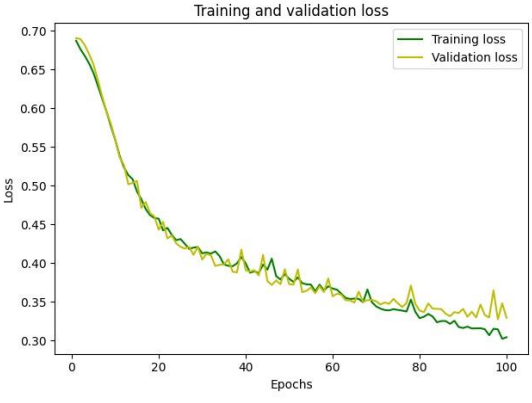
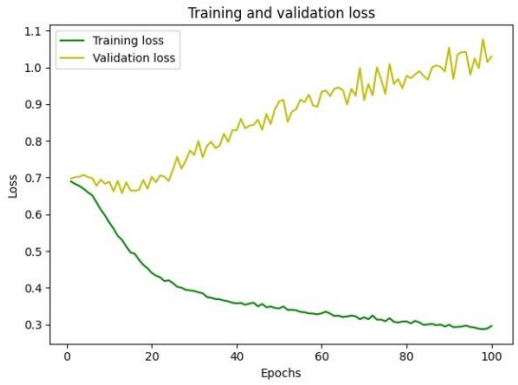
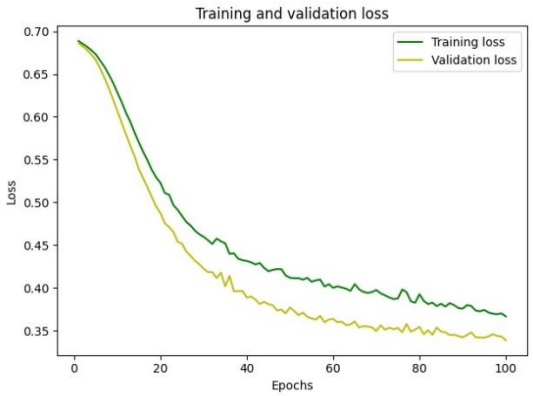
| 80 | 60 |
| --- | --- |
|  |  |
| 40 | 20 |
|  |  |



Таблица 6 – Графики потерь для разного количества нейронов

| 80 | 60 |
| --- | --- |
|  |  |

| 40 | 20 |
| --- | --- |
|  |  |

Анализируя полученные графики, можно сделать вывод о том, что при количестве нейронов больше 60 появляются очень большие потери и происходит переобучение. Самая высокая точность и наименьшие потери достигаются при 60 нейронах.

**Выводы.**

В ходе лабораторной работы произошло ознакомление с задачей бинарной классификации, создание модели ИНС в tf.Keras и настройки параметров обучения. Обучая модель и анализируя полученные результаты был сделан вывод, что модель лучше обучается одном скрытом слое, так как потери минимальные.

# ПРИЛОЖЕНИЕ А

## Исходный код программы

```
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
from tensorflow.keras.models import load_model
import matplotlib.pyplot as plt
import numpy as np

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

np.random.shuffle(dataset)

X = dataset[:,0:60].astype(float)
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(1, kernel_initializer='normal', activation='sigmoid'))

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

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

history_dict = history.history

#График ошибки
loss_values = history_dict['loss']
val_loss_values = history_dict['val_loss']
epochs = range(1, len(loss_values) +1)
plt.plot(epochs, loss_values, 'g', label='Training loss')
plt.plot(epochs, val_loss_values, 'y', label='Validation loss')
plt.title('Training and validation loss')
```

```
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

#График точности
plt.clf()
acc_values = history_dict['accuracy']
val_acc_values = history_dict['val_accuracy']
plt.plot(epochs, acc_values, 'r', label='Training accurate')
plt.plot(epochs, val_acc_values, 'b', label='Validation accurate')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```