

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

ЛАБОРАТОРНАЯ РАБОТА №2
по дисциплине «Искусственные нейронные сети»
Тема: «Бинарная классификация отраженных сигналов радара»

Студент гр. 7381

Павлов А.П.

Преподаватель

Жукова Н. А.

Санкт-Петербург

2020

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.

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:
 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

Выполнение работы.

- 1) Была создана и обучена модель искусственной нейронной сети в соответствии с условиями (весь код представлен в приложении А).
- 2) При исследовании разных архитектур и обучение при различных параметрах обучения ИНС необходимо было:
 - уменьшить размер входного слоя в два раза
 - добавить скрытый слой в архитектуру сети с 15 нейронами

В итоге получаем 4 модели ИНС, которые нужно сравнить. Ниже на рис. 1-8 представлены графики точности и ошибок моделей в ходе обучения. Графики аналогичны графикам из предыдущей лабораторной работы.

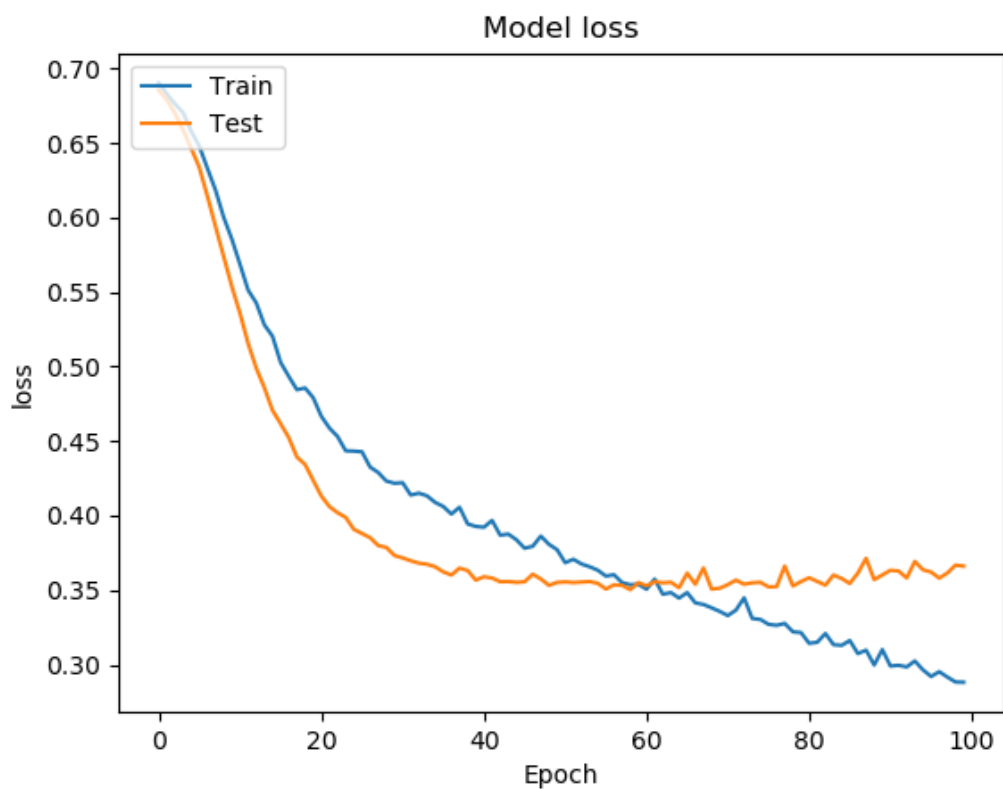


Рисунок 1 – график ошибок изначальной модели ИНС

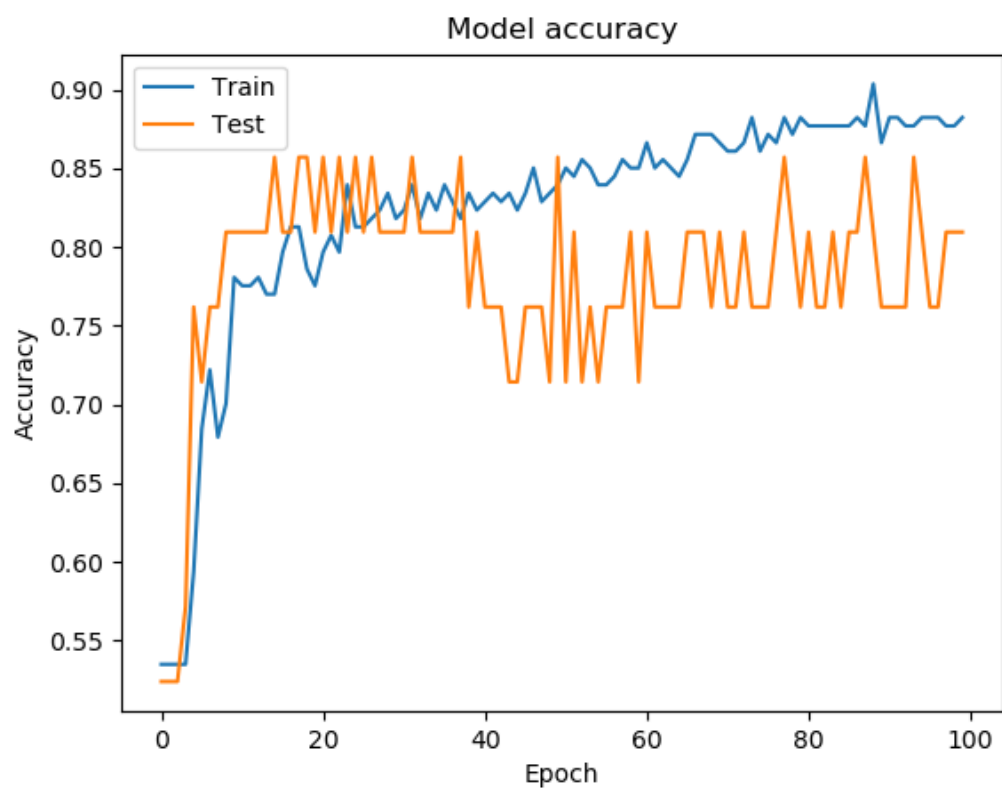


Рисунок 2 – график точности изначальной модели ИНС

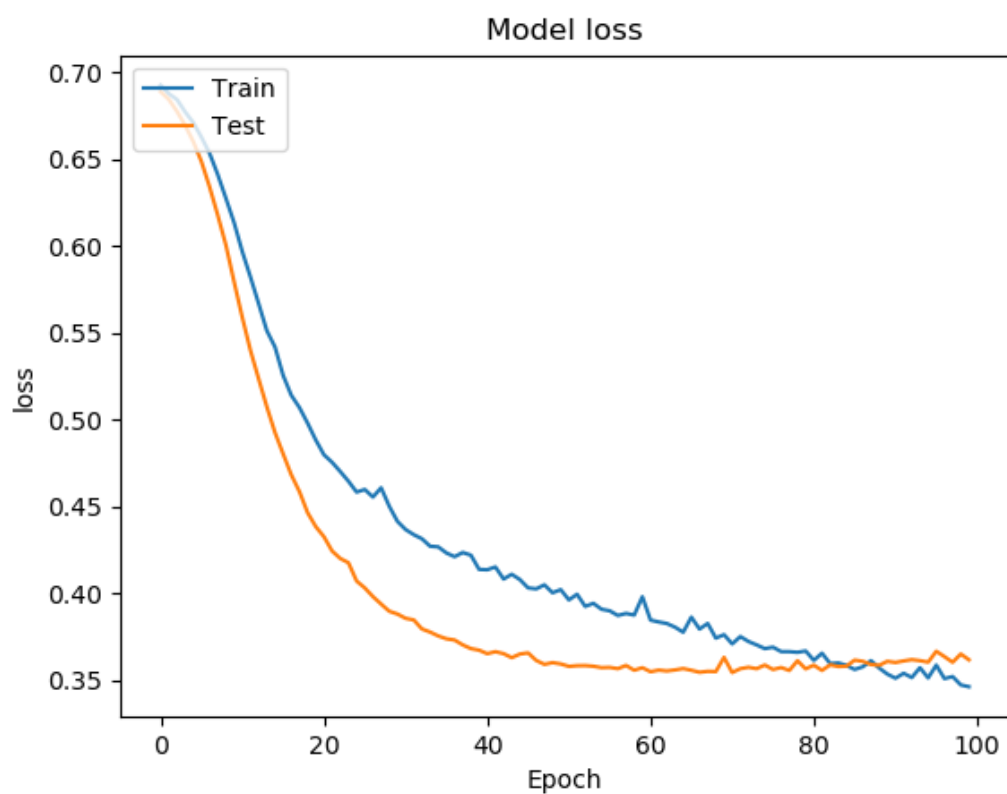


Рисунок 3 – график ошибок модели ИНС с уменьшенным количеством нейронов в 2 раза во входном слое

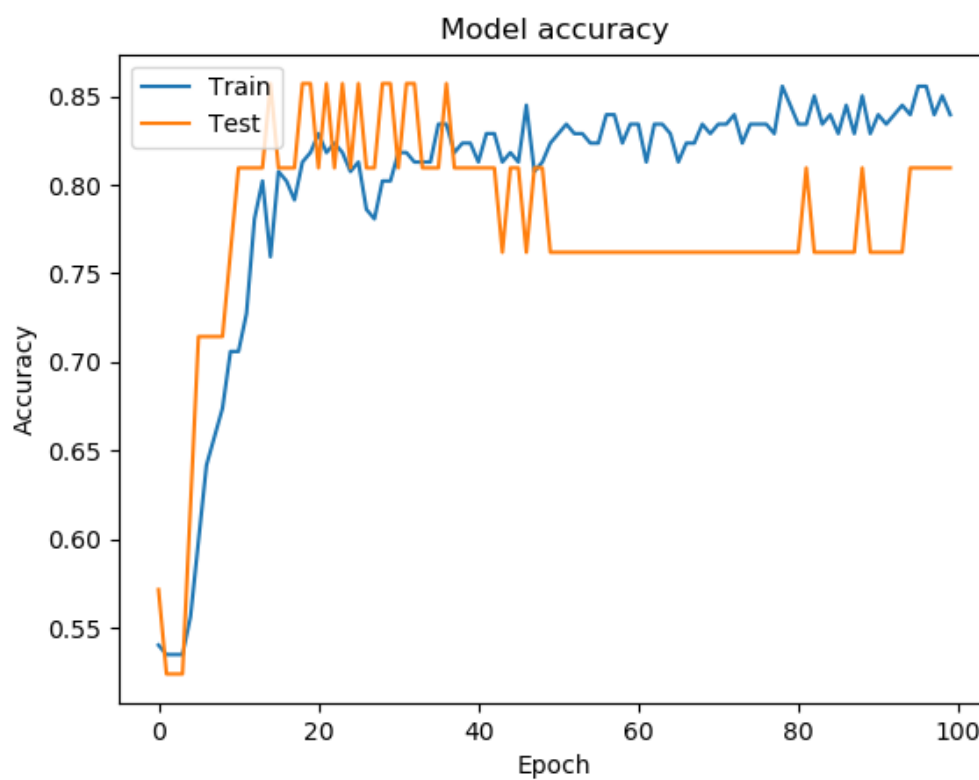


Рисунок 4 – график точности модели ИНС с уменьшенным количеством нейронов в 2 раза во входном слое

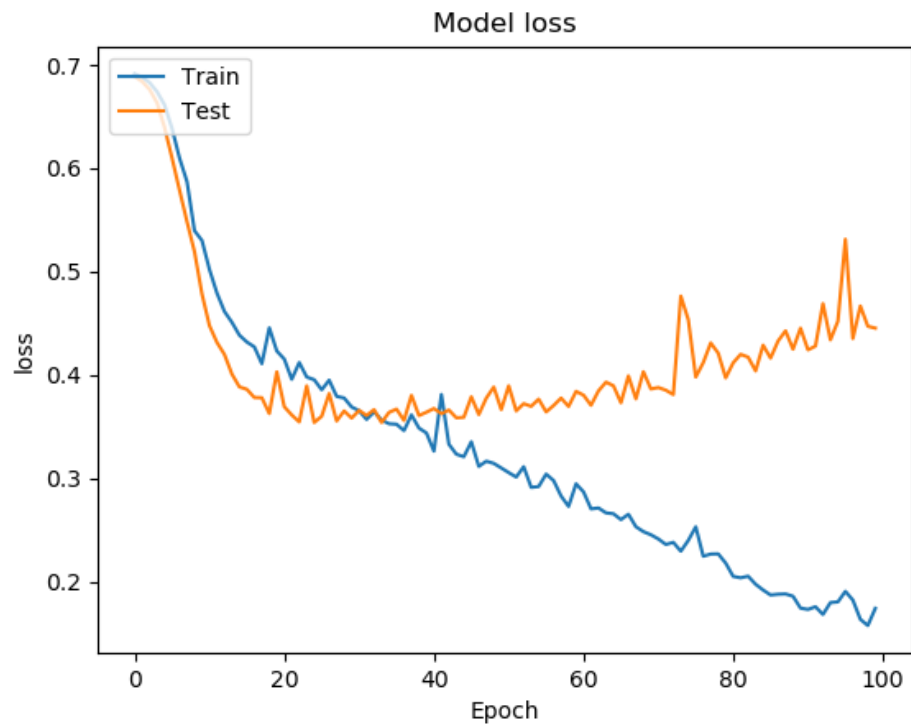


Рисунок 5 – график ошибок модели ИНС с скрытым слоем

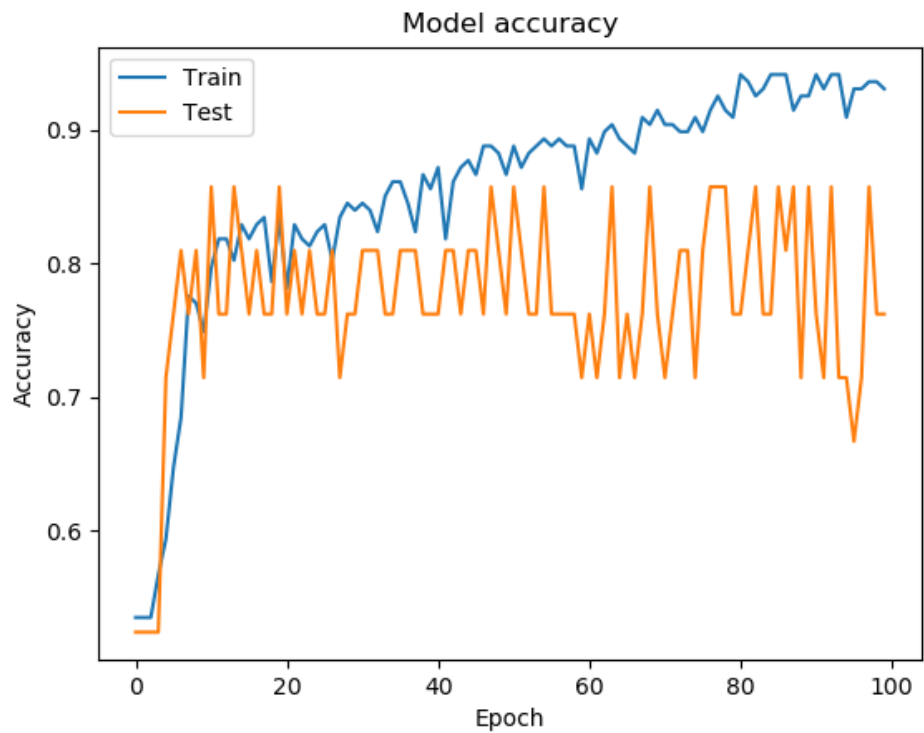


Рисунок 6 – график точности модели ИНС с скрытым слоем

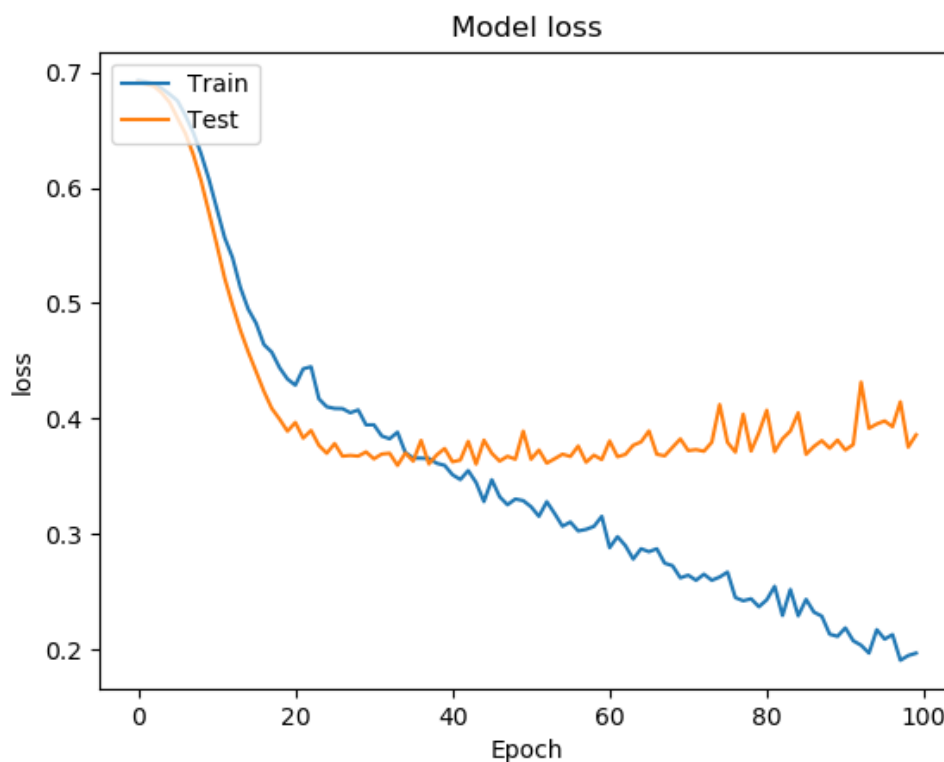


Рисунок 7 – график ошибок модели ИНС с уменьшенным количеством нейронов в входном слое и скрытым слое

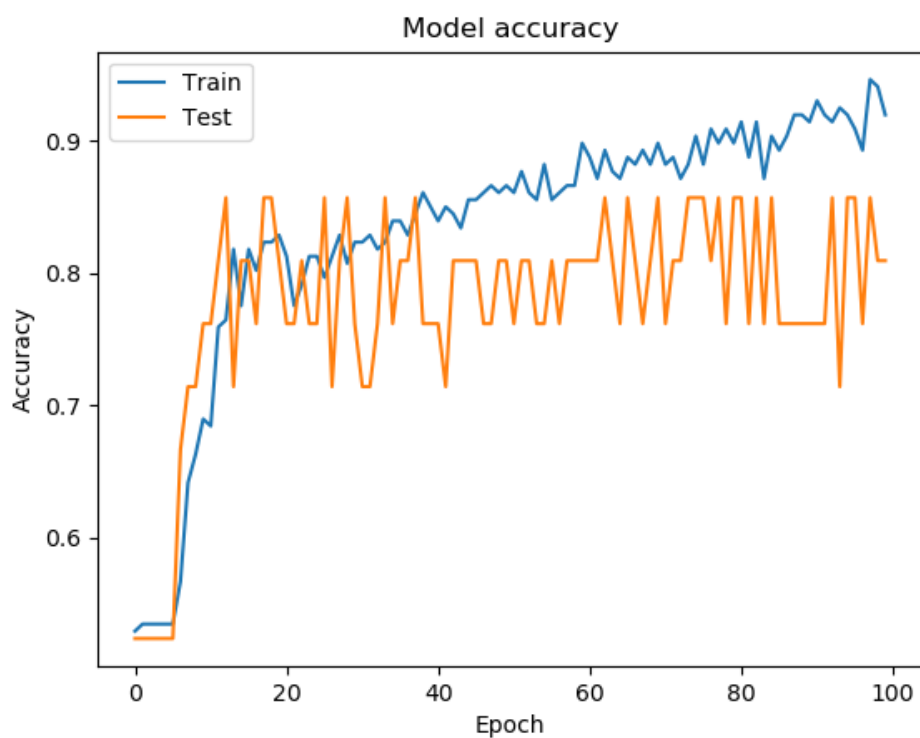


Рисунок 8 – график точности модели ИНС с уменьшенным количеством нейронов в входном слое и скрытым слое

Анализируя полученные графики, можно сделать вывод о том, что с введением скрытого слоя, появляется эффект переобучения сети, в связи с избыточной сложностью модели для конкретной задачи. Это отражается в меньшей точности и большей ошибке на тестовых данных в сравнении с обучаемыми. Также можно наблюдать небольшой прирост точности модели с уменьшением размера входного слоя в ней.

Вывод.

В ходе выполнения данной работы были получены навыки в бинарной классификации в библиотеке keras, была изучена зависимость количества слов и нейронов в них от результирующей точности и потери обучаемой модели ИНС.

Приложения

Приложение А

```
import pandas
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt
import numpy as np

if __name__ == '__main__':
    dataframe = pandas.read_csv('sonar.csv', header=None)
    dataset = dataframe.values
    rand=list(range(len(dataset)))
    random.seed(6)
    random.shuffle(rand)
    dataset=dataset[rand]
    X = dataset[:, 0:60].astype(float)
    Y = dataset[:, 60]

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

    model = Sequential()
    model.add(Dense(30, kernel_initializer='normal',
activation='relu', input_dim=60))
    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'])

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

    plt.plot(history.history['acc'])
    plt.plot(history.history['val_acc'])
    plt.title('Model accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.show()

    plt.plot(history.history['loss'])
    plt.plot(history.history['val_loss'])
    plt.title('Model loss')
    plt.ylabel('loss')
    plt.xlabel('Epoch')
```



```
plt.legend(['Train', 'Test'], loc='upper left')  
plt.show()
```