

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

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

Студентка гр. 8383

Сырцова Е.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель

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 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

Была построена нейронная сеть, код которой представлен в приложении А.

После запуска программы с предложенным кодом были получены графики ошибок и точности, а значение точности работы ИНС равно 91,4%. Графики представлены на рис. 1-2.

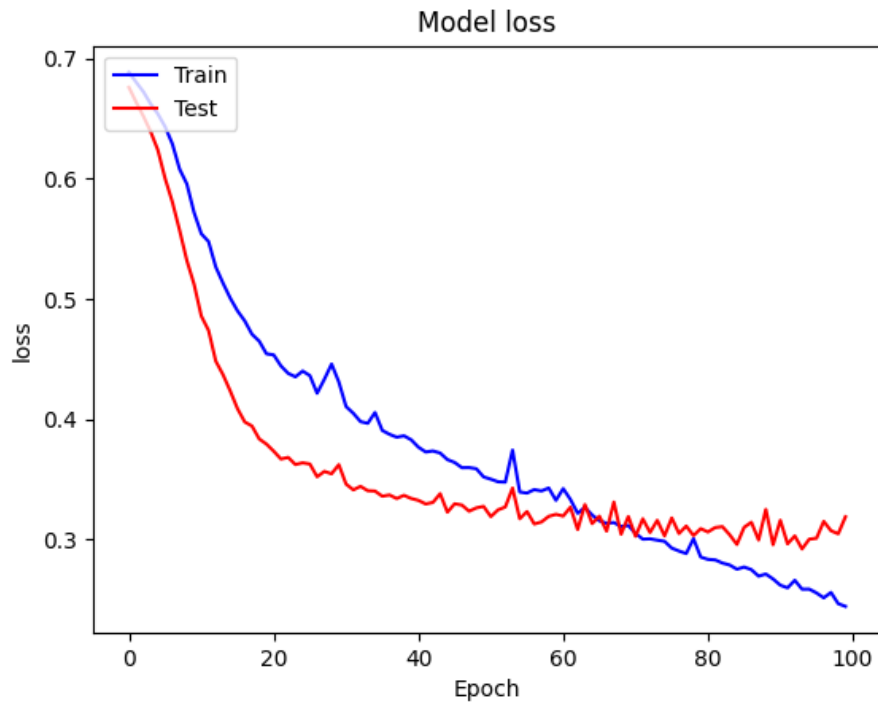


Рисунок 1 – График ошибок исходной сети

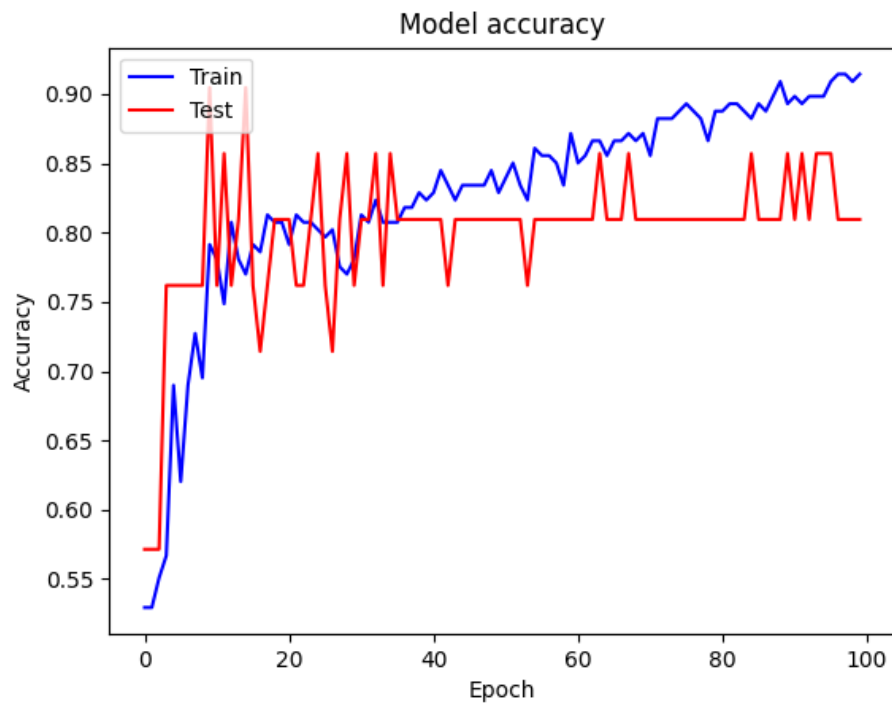


Рисунок 2 – График точности исходной сети

В представленном наборе данных присутствует некоторая избыточность, т.к. с разных углов описывается один и тот же сигнал. Вероятно, что некоторые углы отражения сигнала имеют большую значимость, чем другие.

Теперь уменьшим размер входного слоя в два раза и сравним с результатами первоначальной архитектуры. Точность вычислений сети равна 86,6%, это говорит о том, что изменение количества нейронов во входном слое напрямую влияет на количество признаков, с которыми будет работать нейронная сеть. Графики представлены на рис. 3-4.

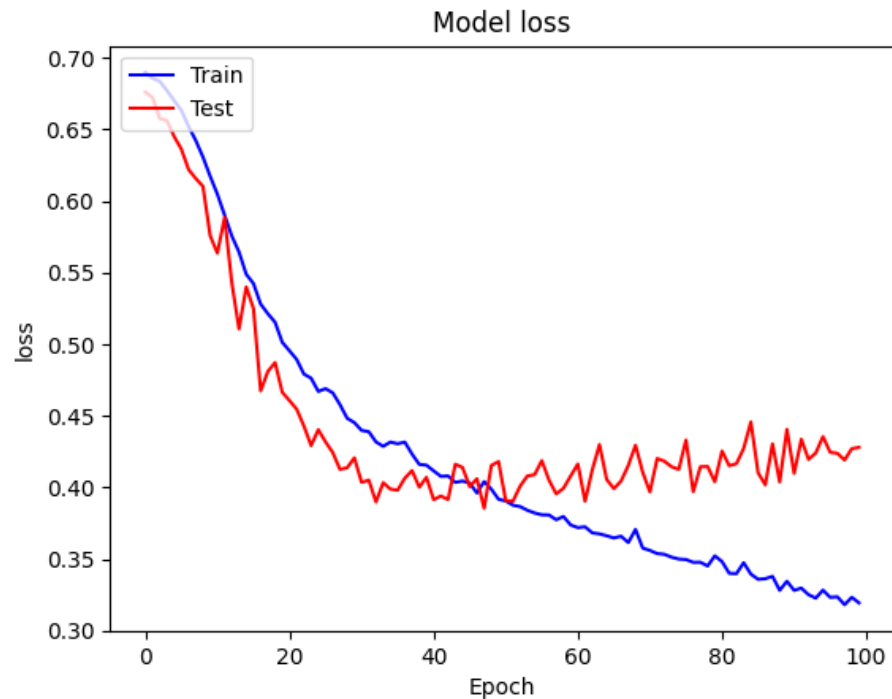


Рисунок 3 – График ошибок

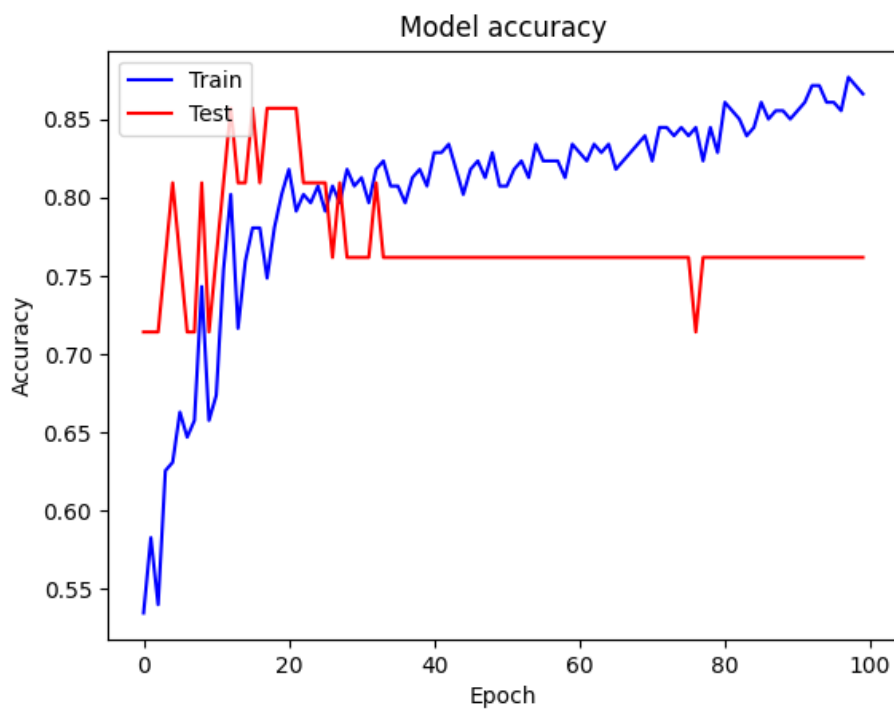


Рисунок 4 – График точности

Теперь добавим еще один скрытый слой с 15 нейронами. Точность стала равна 90,3%. Графики изображены на рис. 5-6.

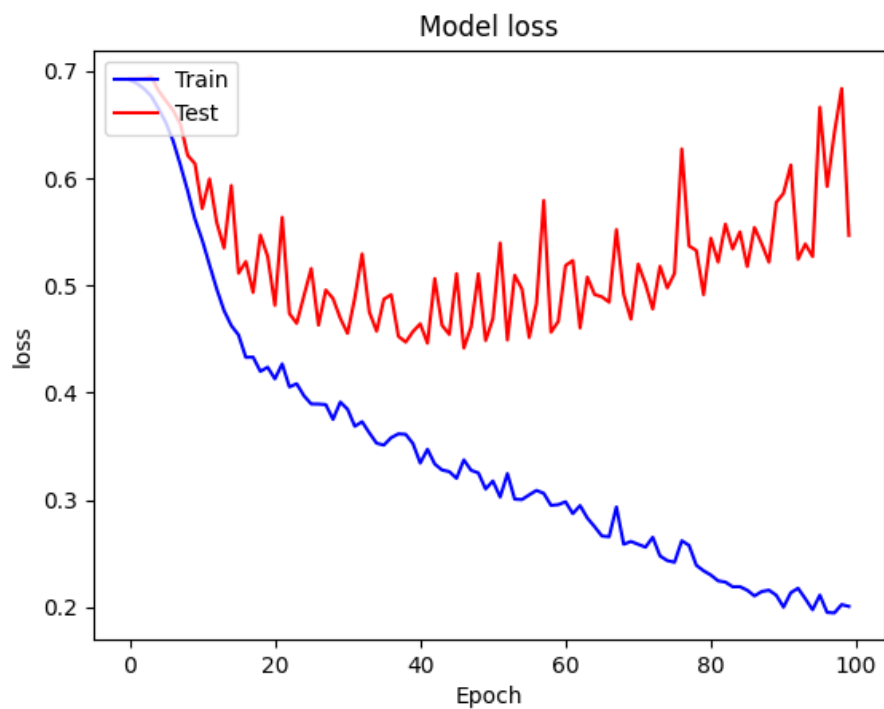


Рисунок 5 – График ошибок

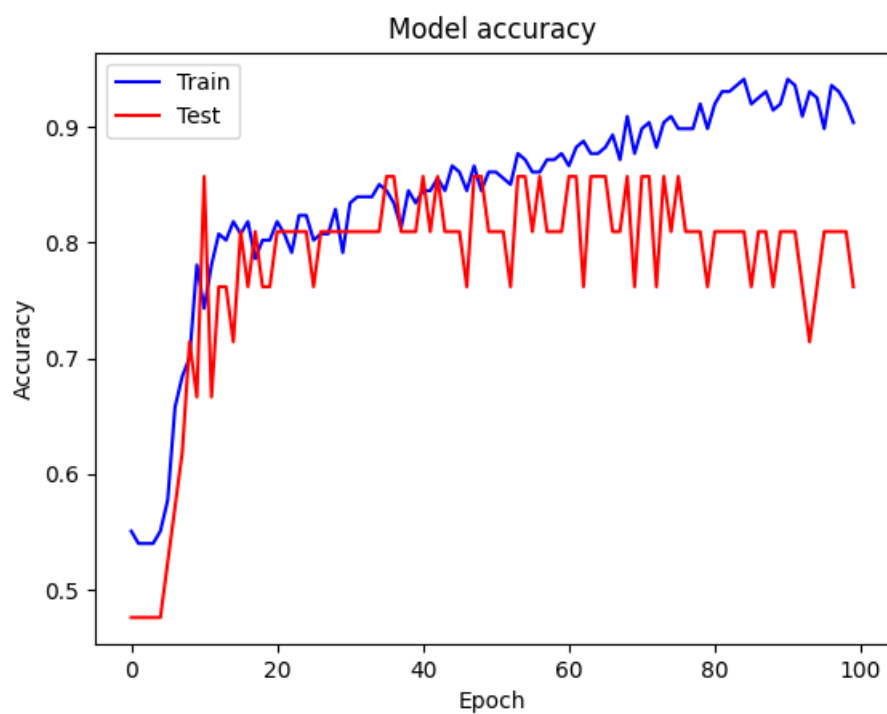


Рисунок 6 – График точности

Вывод

В процессе выполнения данной работы были получены навыки в бинарной классификации в библиотеке keras. Анализируя полученные данные, 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 ошибки на тестовых данных в сравнении с обучаемыми. Также можно наблюдать падение точности модели с уменьшением размера входного слоя в ней.

ПРИЛОЖЕНИЕ А

```
import pandas
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential
from sklearn.preprocessing import LabelEncoder
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(15, 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)
plt.figure(1)
plt.plot(history.history['acc'], 'b', markersize=2)
plt.plot(history.history['val_acc'], 'r', markersize=2)
plt.title('Model accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(['Train', 'Test'], loc='upper left')
plt.figure(2)
plt.plot(history.history['loss'], 'b', markersize=2)
plt.plot(history.history['val_loss'], 'r', markersize=2)
plt.title('Model loss')
plt.xlabel('Epoch')
plt.ylabel('loss')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```