

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

Студент гр. 8383

Мололкин К.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

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.

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

Выполнение работы

Задача бинарной классификации является одним из основных видов задач, для решения которых применяются нейронные сети. Для выполнения поставленной задачи был загружен файл с обучающими данными и написана программа на языке python с использованием библиотек keras для обучения ИНС, pandas для загрузки данных, scikit-learn для обучения модели и matplotlib для рисования графиков. Код программы представлен в приложении А.

Первой была изучена сеть с двумя слоями, 60 нейронов на первом и 1 нейрон на втором. На рис. 1 представлен график потерь, а на рис. 2 график точности.

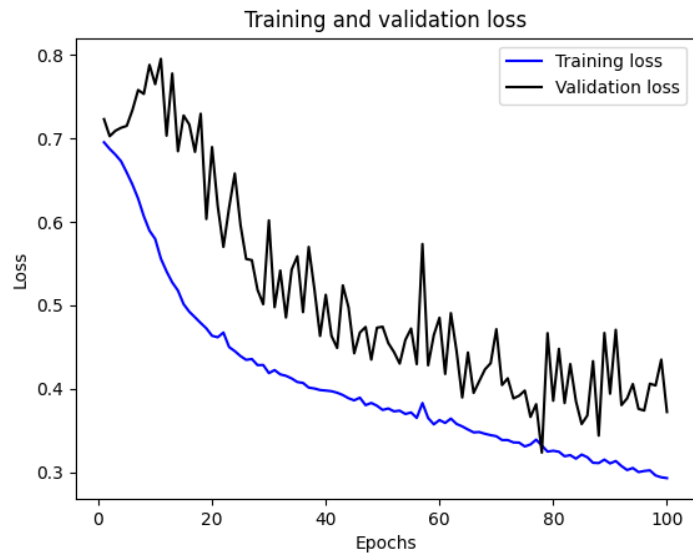


Рисунок 1 – график потерь для первой модификации

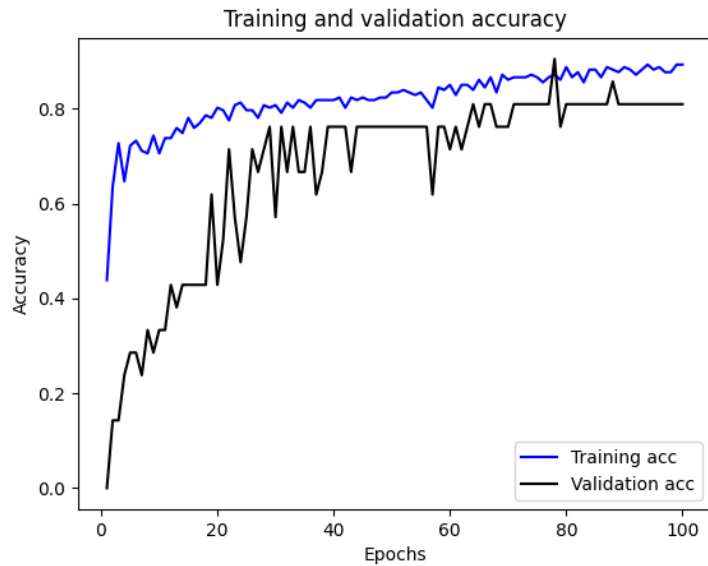


Рисунок 2 – график точности для первой модификации

Вторая модификация состоит из двух слоёв с 30 нейронами на первом и 1 нейроном на втором слое. На рис. 3 представлен график потерь, а на рис. 4 график точности.

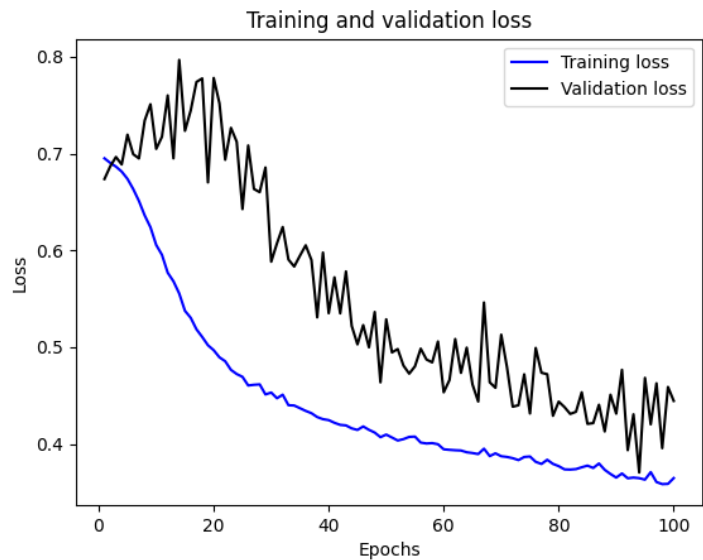


Рисунок 3 – график потерь для второй модификации

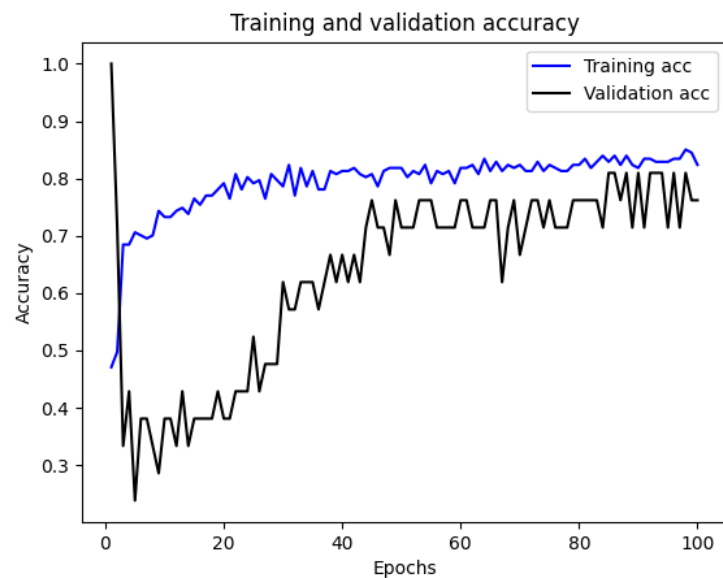


Рисунок 4 – график точности для второй модификации

Третья модификация состоит из трех слоёв с 60 нейронами на первом, 30 нейронами на втором и 1 нейроном на третьем слое. На рис. 5 представлен график потерь, а на рис. 6 график точности.

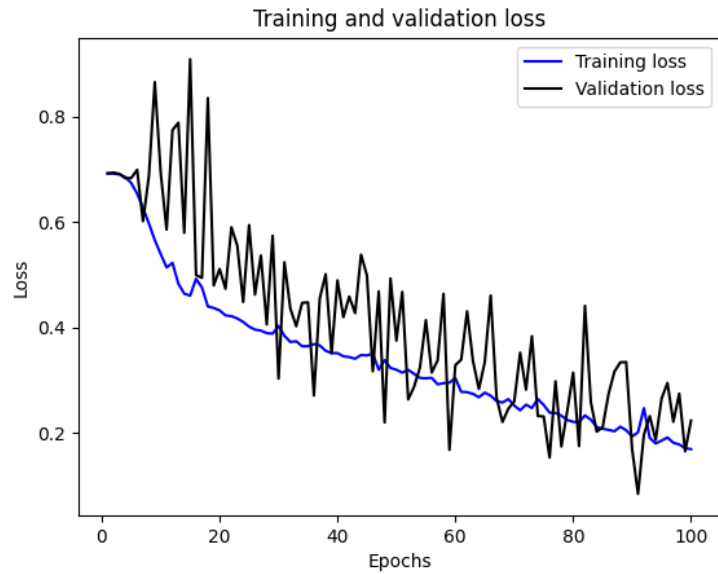


Рисунок 5 – график потерь для третьей модификации

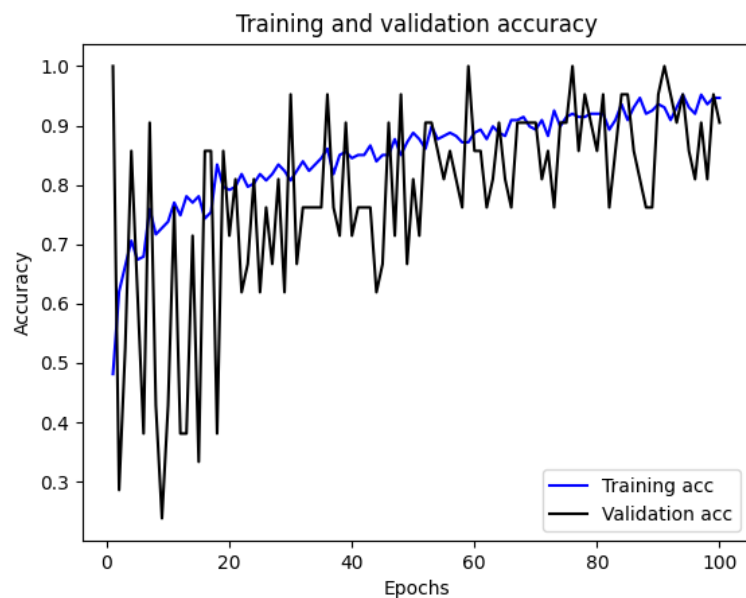


Рисунок 6 – график точности для третьей модификации

Вывод

Во время выполнения лабораторной работы была создана ИНС, которая реализует классификацию между камнями (R) и металлическими цилиндрами (M) на основе данных об отражении сигналов радара от поверхностей. Из полученных результатов можно сделать вывод, что наилучшей является модель с тремя слоями с 60 нейронами на первом, 15 нейронами на втором и 1

нейроном на последнем слое. Потери на валидации равны 0.2, а точность на валидации равна 0.9

Приложение А

Листинг программы

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

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

loss_values = history_dict['loss']
val_loss_values = history_dict['val_loss']
epochs = range(1, len(loss_values) + 1)
plt.plot(epochs, loss_values, 'b', label='Training loss')
plt.plot(epochs, val_loss_values, 'k', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

acc_values = history_dict['accuracy']
val_acc_values = history_dict['val_accuracy']
plt.plot(epochs, acc_values, 'b', label='Training acc')
plt.plot(epochs, val_acc_values, 'k', label='Validation acc')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
```

```
plt.show()
```