

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

Студент гр. 7382

Бахеров Д.В.

Преподаватель

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.

1. Ознакомиться с задачей бинарной классификации
2. Загрузить данные
3. Создать модель ИНС в tf.Keras
4. Настроить параметры обучения
5. Обучить и оценить модель
6. Изменить модель и провести сравнение. Объяснить результаты

Требования к выполнению задания.

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.

На первом слое имеем 60 нейронов, что равно количеству элементов, которые подаются на вход НС. График ошибки и точности модели изображен на рис. 1 и 2 соответственно.

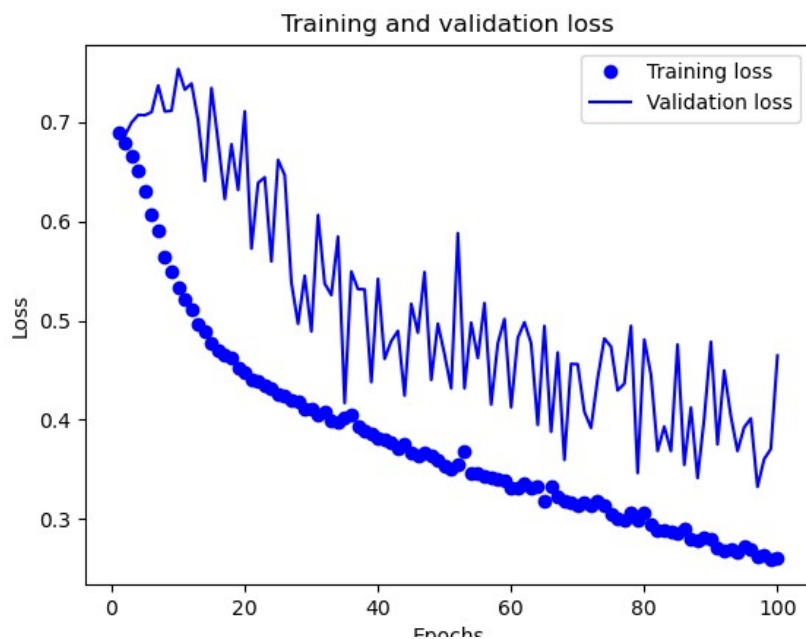


Рисунок 1 – График ошибки модели при 60 нейронах

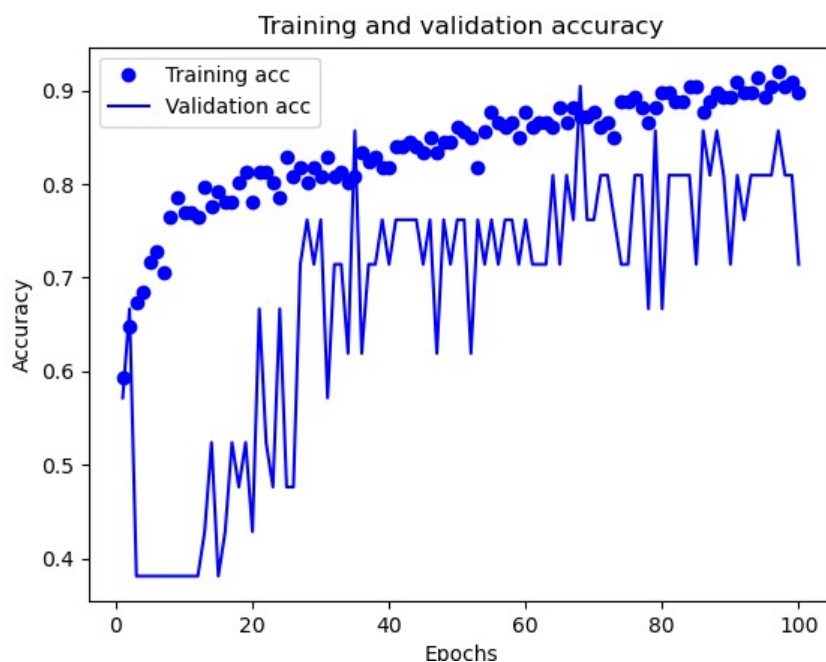


Рисунок 2 – График точности при 60 нейронах

Уменьшим размер входного слоя в два раза. Из графиков на рис. 3 и рис. 4 видно, что результат очень схож с начальным. Выходит, что в 1 случае излишнее количество нейронов в первом слое.

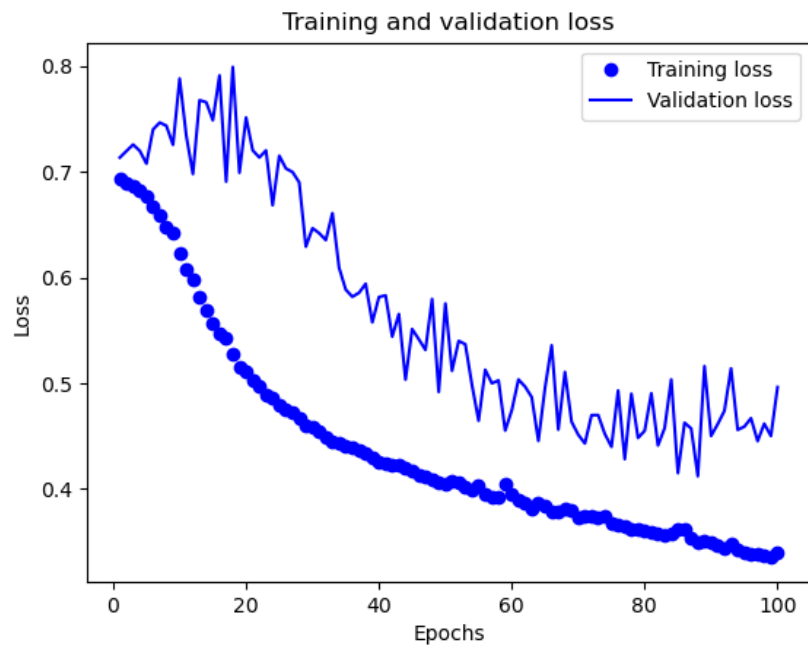


Рисунок 3 – График ошибки модели при 30 нейронах

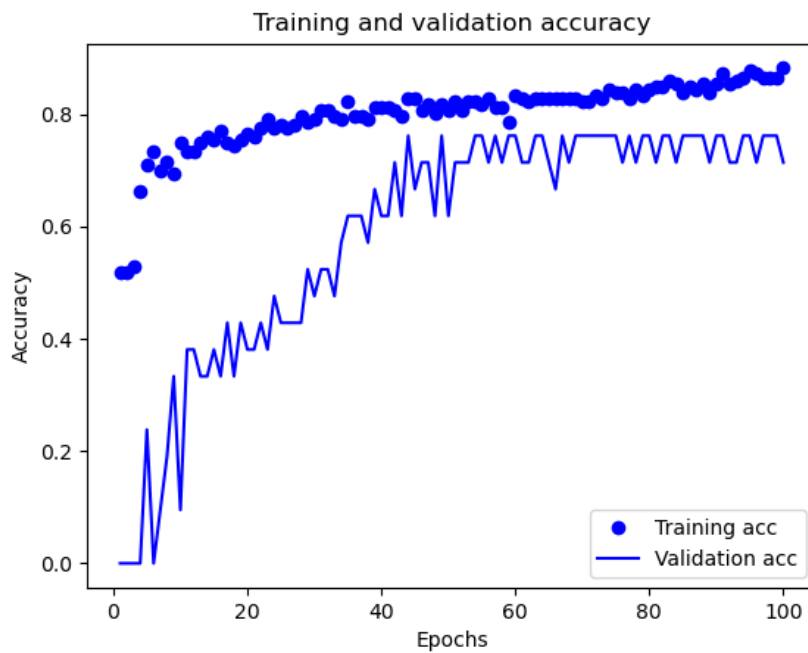


Рисунок 4 – График точности при 30 нейронах

Добавили промежуточный (скрытый) слой Dense в архитектуру сети с 15 нейронами. Из графиков на рис. 5 и рис. 6 видно, что точность

увеличилась и ошибка стала меньше. Т.е. добавляя второй слой мы начали рассматривать еще и комбинации изначальных признаков (сигналов).

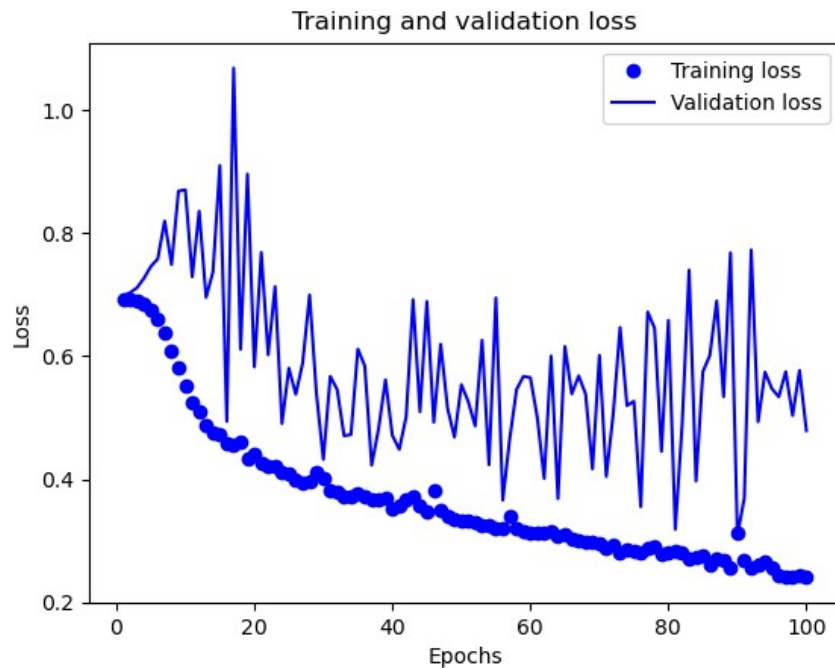


Рисунок 5 – График ошибки модели при 30 нейронах на 1 слое и 15 нейронах на 2 слое

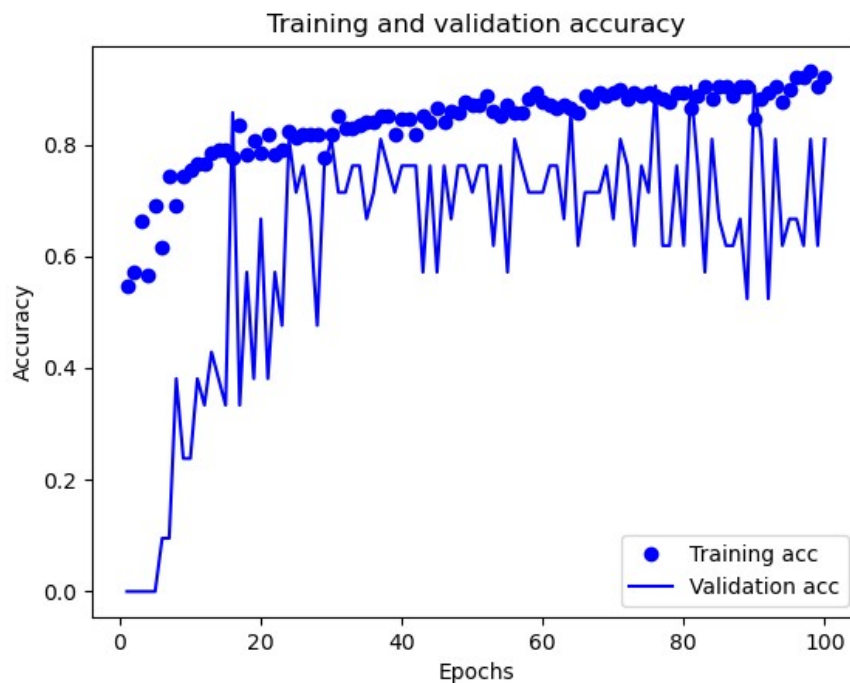


Рисунок 6 – График точности при 30 нейронах на 1 слое и 15 нейронах на 2 слое

Выводы.

В ходе работы было выявлено, что изменение количества нейронов во входном слое напрямую влияет на количество признаков, с которыми будет работать нейронная сеть. Так, сеть с 30 и 60 нейронами в 1 слое показала одинаковые результаты, следовательно не обрабатывая на каждой итерации все элементы результат остается неизменным. Добавляя второй слой мы начали рассматривать еще и комбинации изначальных признаков, что и дало выигрыш по сравнению со 2 состоянием.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
import pandas
from keras.layers import Dense
from keras.models import Sequential
from keras.utils import to_categorical
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(30, input_dim=60, init='normal', activation='relu'))
model.add(Dense(15, init='normal', activation='relu'))
model.add(Dense(1, init='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']
acc_values = history_dict['accuracy']
val_acc_values = history_dict['val_accuracy']

epochs = range(1, len(loss_values)+1)

plt.plot(epochs, loss_values, 'bo', label='Training loss')
plt.plot(epochs, val_loss_values, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

plt.clf()
plt.plot(epochs, acc_values, 'bo', label='Training acc')
plt.plot(epochs, val_acc_values, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

