

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

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

Студент гр. 7381

Трушников А.П.

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.

Порядок выполнения работы.

- 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.

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.

Посмотрим, как поведет себя сеть при количестве слоев и нейронов в них по умолчанию. Результаты обучения сети показаны на рис.1-2.

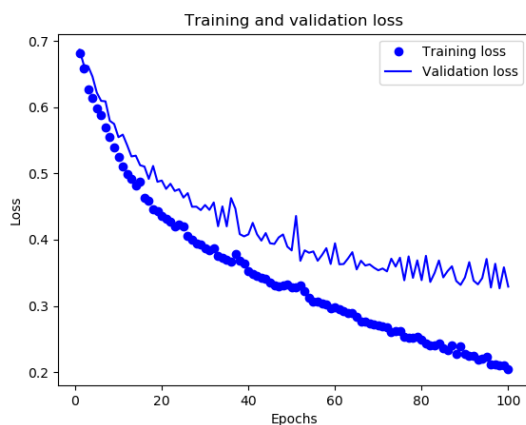


Рисунок 1 – График ошибок при конфигурации сети по умолчанию

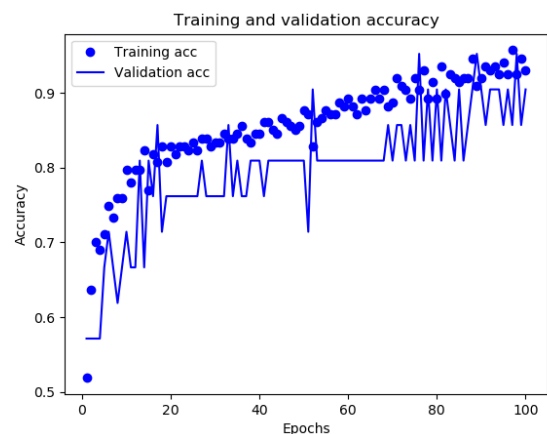


Рисунок 2 - Точность модели при конфигурации сети по умолчанию

Как видно из графиков переобучение сети происходит практически самого начала обучения сети.

Посмотрим, как поведет себя сеть при наличии от одного до двух скрытых слоев по 60 нейронов в каждом. Графики приведены на рис. 3 – 4 (а) 1 скрытый слой, (б) 2 скрытых слоя.

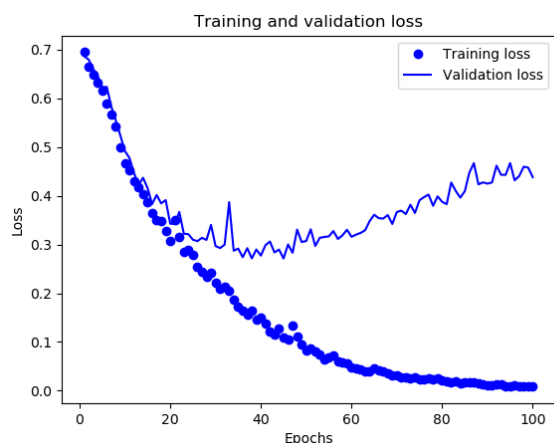


Рисунок 3(а) – график потерь.

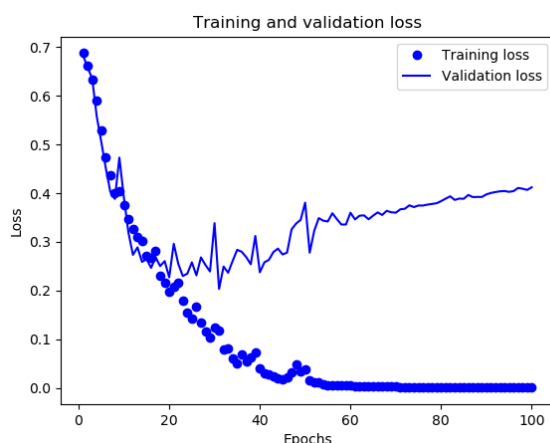


Рисунок 3(б) – график потерь.

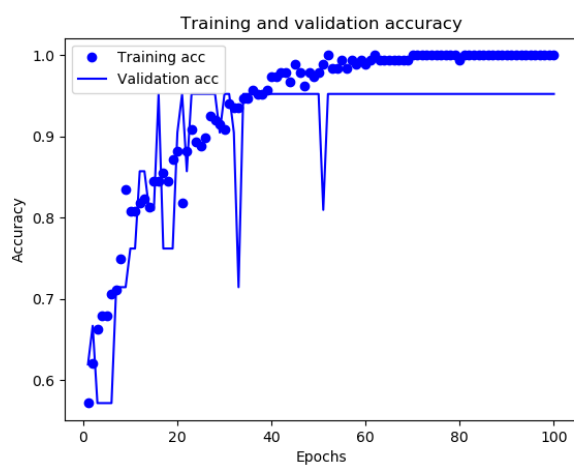


Рисунок 4(а) – точность.

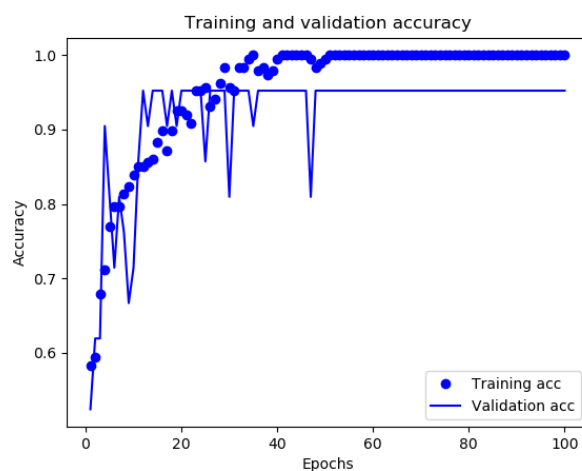


Рисунок 4(б) – точность.

В обоих случаях наблюдается переобучение после ~20 эпохи.

Уменьшим количество нейронов в два раза – по 30 нейронов в скрытом слое. Графики приведены на рис. 5 – 6 (а) 1 скрытый слой, (б) 2 скрытых слоя.

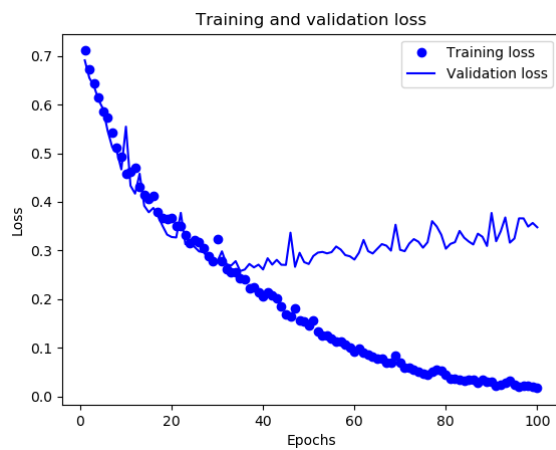


Рисунок 5(а) – график потерь.

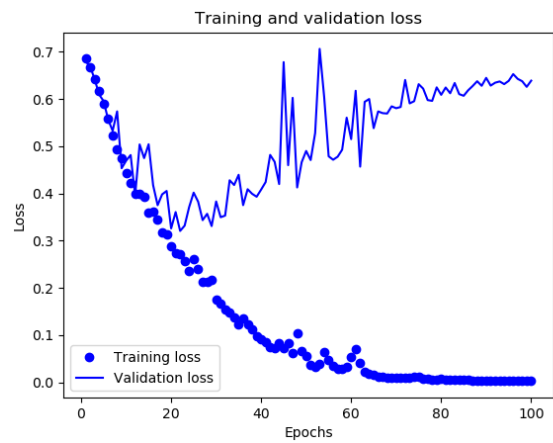


Рисунок 5(б) – график потерь.

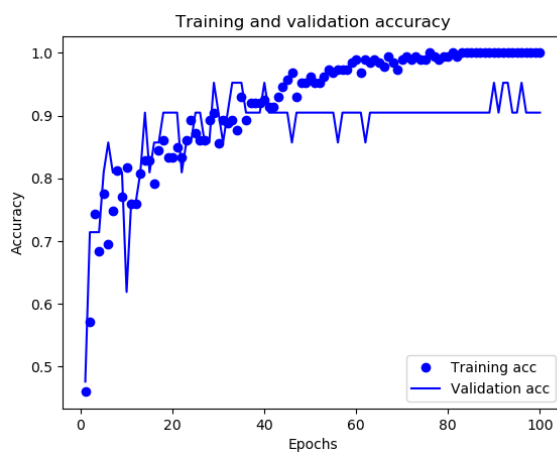


Рисунок 6(а) – точность

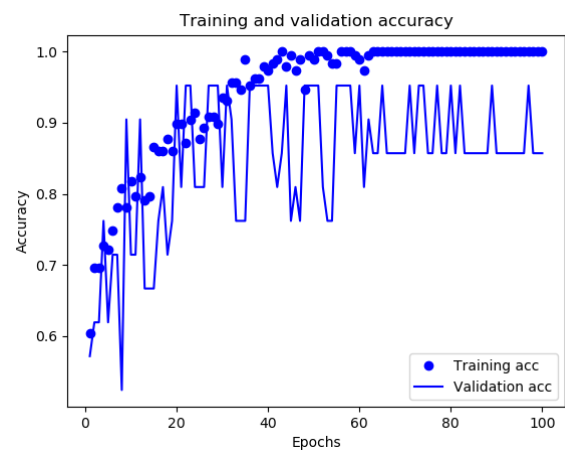


Рисунок 6(б) – точность

При 1 скрытом слое в 30 нейронов переобучение наблюдается после 30 эпохи. В остальных случаях переобучение происходит гораздо раньше.

Теперь смоделируем модель с 30 нейронами во входном слое. Результаты представлены на рис.7-8

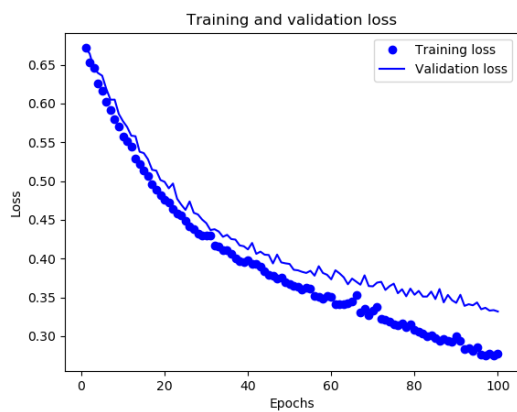


Рисунок 7 – график потерь

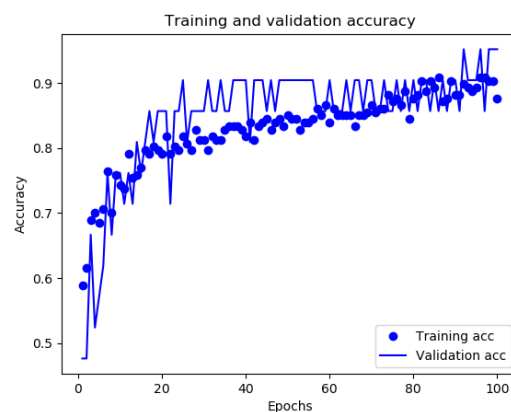


Рисунок 8 – точность

Переобучение происходит примерно после 20 эпохи.

Добавим в модель 1 скрытый слой с 30 нейронами. Результаты представлены на рис.8 – 9.

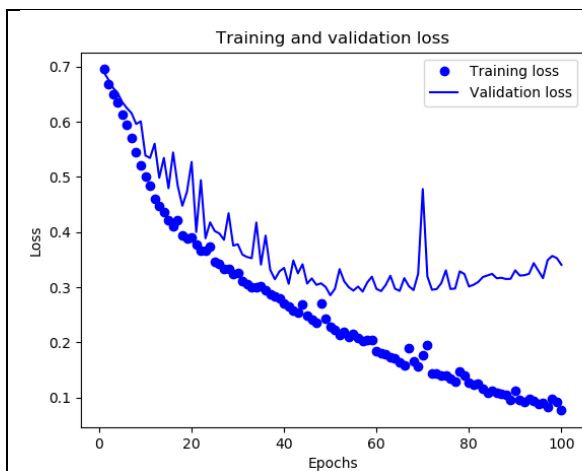


Рисунок 8 – график потерь

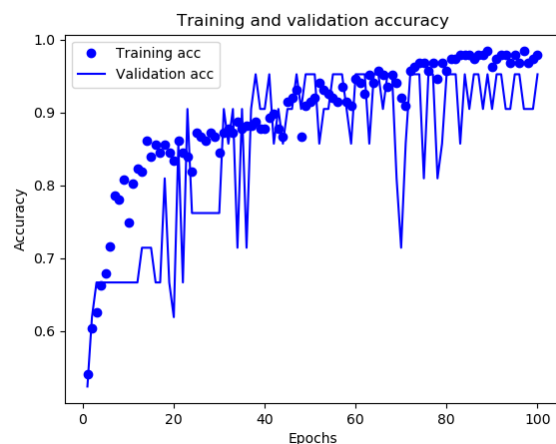


Рисунок 9 - точность

Выводы.

В ходе выполнения лабораторной работы была изучена задача бинарной классификации, было изучено влияние количества слоёв и количества нейронов на слое на обучение модели. В итоге улучшений пришли к

приемлемой модели из 2х нейронных слоёв по 60 и 30 нейронов, при количестве эпох 30.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД

```
import pandas
import random
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
rand = list(range(len(dataset)))
random.seed(123)
random.shuffle(rand)
dataset = dataset[rand]

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, activation='relu'))
model.add(Dense(30, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
history = model.fit(X, encoded_Y, epochs=50, batch_size=10,
validation_split=0.1)

# Получение ошибки и точности в процессе обучения

loss = history.history['loss']
val_loss = history.history['val_loss']
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
epochs = range(1, len(loss) + 1)

# Построение графика ошибки
```

```
plt.plot(epochs, loss, 'g', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

Построение графика точности

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