

**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 цветов отраженных сигналов радара**

Студентка гр. 7382

\_\_\_\_\_

Дерябина П.С.

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**

- 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 к выполнению задания**

- Изучить влияние кол-ва нейронов на слое на результат обучения модели.
- Изучить влияние кол-ва слоев на результат обучения модели
- Построить графики ошибки и точности в ходе обучения
- Провести сравнение полученных сетей, объяснить результат

## **Основные теоретические положения**

Задача бинарной классификации является одним из основных видов задач, для решения которых применяются нейронные сети.

Основным строительным блоком нейронных сетей является слой (или уровень), модуль обработки данных, который можно рассматривать как фильтр для данных. Он принимает некоторые данные и выводит их в более полезной форме. В частности, слои извлекают представления из подаваемых в них данных, которые, как мы надеемся, будут иметь больше смысла для решаемой задачи. Фактически методика глубокого обучения заключается в объединении простых слоев, реализующих некоторую форму поэтапной очистки данных.

Модель глубокого обучения можно сравнить с ситом, состоящим из последовательности фильтров все более тонкой очистки данных — слоев.

Чтобы подготовить сеть к обучению, нужно настроить три параметра для этапа компиляции:

1. функцию потерь, которая определяет, как сеть должна оценивать качество своей работы на обучающих данных и, соответственно, как корректировать ее в правильном направлении, в данной работе применяется функция для бинарной классификации – `binary_crossentropy`
2. оптимизатор — механизм, с помощью которого сеть будет обновлять себя, опираясь на наблюдаемые данные и функцию потерь;
3. метрики для мониторинга на этапах обучения и тестирования — здесь нас будет интересовать только точность (доля правильно классифицированных изображений).

### Ход работы

Код программы, реализующий структуру ИНС, приведен в приложении А.

Посмотрим на графики функции потерь и точности модели в зависимости от эпохи для начальных данных на рис. 1 и рис. 2 соответственно. Параметры: 60 нейронов на первом слое.

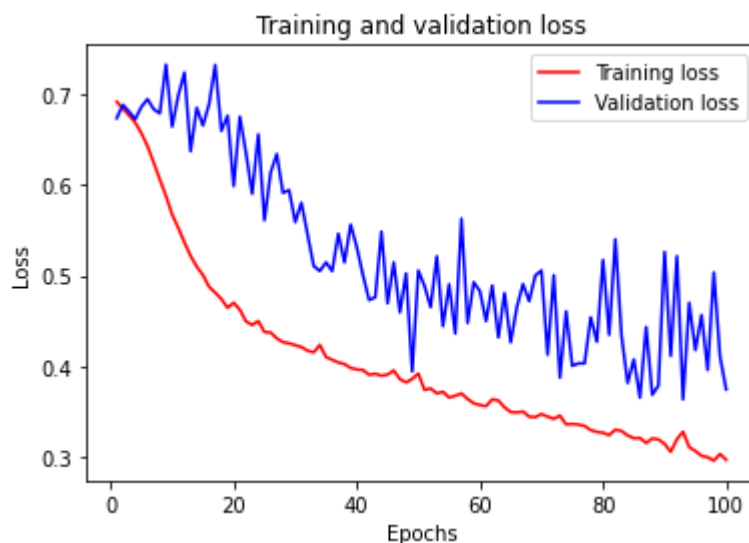


Рисунок 1 – Начальные параметры, график функции потерь



Рисунок 2 – Начальные параметры, график точности модели

Из рис. 1-2 видим, что модели не хватает точности и меньшего значения функции потерь.

Уменьшим количество нейронов первого слоя с 60 до 30 и посмотрим, что получилось на рис. 3 и рис. 4.

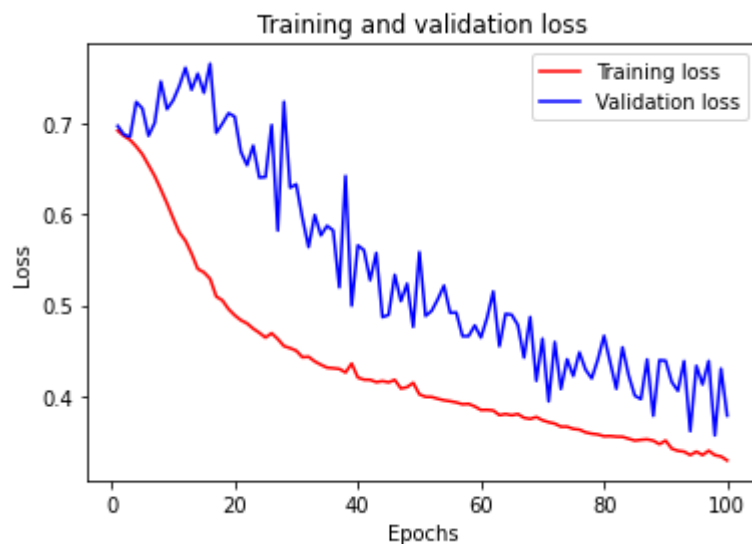


Рисунок 3 – Убрали половину нейронов, график функции потерь

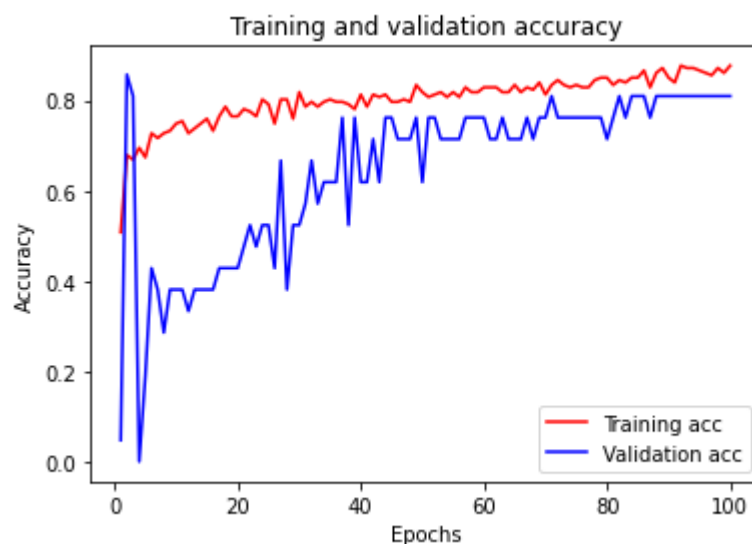


Рисунок 4 – Убрали половину нейронов, график точности модели

Видим, что сильной разницы между двумя моделями нет: итоговые значения метрик мало отличаются, то есть часть нейронов в первом слое была избыточна. Добавим (см. рис. 4-5) промежуточный полносвязный слой с 15 нейронами, это позволяет рассматривать не только сами начальные признаки, но и их комбинации, поэтому может произойти улучшение модели.

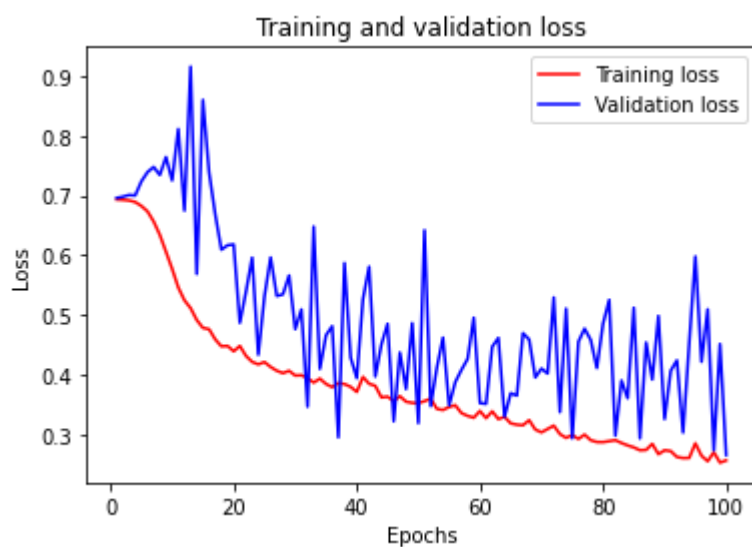


Рисунок 5 – Добавили слой, график функции потерь

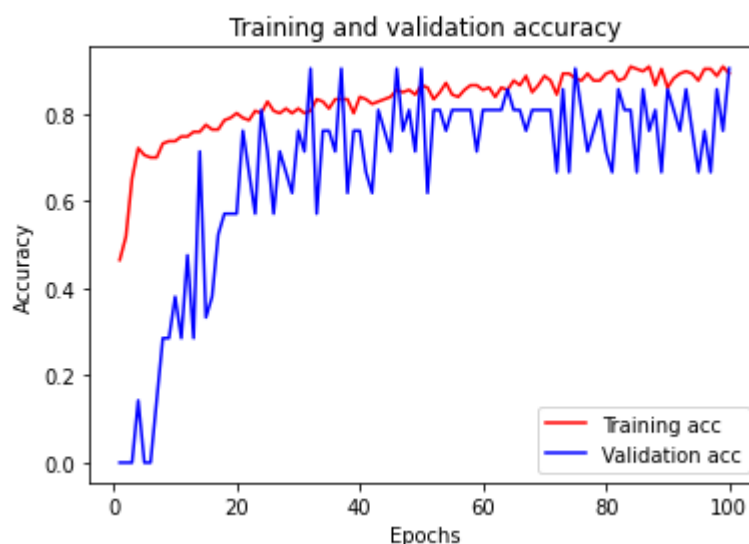


Рисунок 6 – Добавили слой, график точности

Действительно, точность модели повысилась, а функция потерь понизилась.

## Вывод

Изучив основы работы искусственных нейронных сетей, узнали, что изменение количества нейронов во входном слое может напрямую влиять на количество признаков, с которыми будет работать нейронная сеть, нашли избыточность в исходной модели, а также выяснили, что дополнительные слои могут позволить получить более высокую точность благодаря учету в том числе и комбинаций сигналов.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД

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

#1
dataframe = pandas.read_csv("sonar.all-data", header=None)
dataset = dataframe.values
X = dataset[:,0:60].astype(float)
Y = dataset[:,60]

#2
encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)

#3
model = Sequential()
model.add(Dense(30, input_dim=60, kernel_initializer='normal', activation='relu'))
model.add(Dense(15, kernel_initializer='normal', activation='relu'))
model.add(Dense(1, kernel_initializer='normal', activation='sigmoid'))

#4
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

#5
h = model.fit(X, encoded_Y, epochs=100, verbose=0, batch_size=10, validation_split=0.1)
history = h.history

# Получение ошибки и точности в процессе обучения
loss = history['loss']
val_loss = history['val_loss']
acc = history['accuracy']
val_acc = history['val_accuracy']
epochs = range(1, len(loss) + 1)

# Построение графика ошибки
plt.plot(epochs, loss, 'r', 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, 'r', 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()
```