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.8382 _____ Терехов А.Е.

Преподаватель _____ Жангриров 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 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.

Основные теоретические положения

Импортируем необходимые для работы классы и функции. Кроме Keras понадобится Pandas для загрузки данных и scikit-learn для подготовки дан-

ных и оценки модели.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential
from collections import namedtuple
```

Листинг 1: Подключение модулей

Набор данных загружается напрямую с помощью pandas. Затем необходимо разделить атрибуты (столбцы) на 60 входных параметров (X) и 1 выходной (Y).

```
dataframe = pandas.read_csv("sonar.csv", header=None)
dataset = dataframe.values
X = dataset[:,0:60].astype(float)
Y = dataset[:,60]
```

Листинг 2: Обработка данных

Выходные параметры представлены строками (“R” и “M”), которые необходимо перевести в целочисленные значения 0 и 1 соответственно. Для этого применяется LabelEncoder из scikit-learn.

```
encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
```

Листинг 3: Обработка данных

Теперь можно задать базовую архитектуру сети.

```
model = Sequential()
model.add(Dense(60, input_dim=60, init='normal', activation='relu'))
model.add(Dense(1, init='normal', activation='sigmoid'))
```

Листинг 4: Обработка данных

Чтобы подготовить сеть к обучению, нужно настроить еще 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; Для задач бинарной классификации применяется функция 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).

```
model.compile(optimizer='adam',loss='binary crossentropy', metrics=['
    accuracy'])
```

Листинг 5: Обработка данных

Теперь можно начинать обучение сети, для чего в случае использования библиотеки Keras достаточно вызвать метод fit сети — он пытается адаптировать (fit) модель под обучающие данные.

```
model.fit(X, encoded_Y, epochs=100, batch_size=10, validation_split=0.1)
```

Листинг 6: Обработка данных

В процессе обучения отображаются четыре величины: потери сети на обучающих данных и точность сети на обучающих данных, а также потери и точность на данных, не участвовавших в обучении.

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. Изменение количества нейронов во входном слое напрямую влияет на ко-

личество признаков, с которыми будет работать нейронная сеть.

- Необходимо уменьшить размер входного слоя в два раза и сравнить с результатами первоначальной архитектуры.

Нейронная сеть с несколькими слоями позволяет находить закономерности не только во входных данных, но и в их комбинации. Также, дополнительные слои позволяют ввести нелинейность в сеть, что позволяет получать более высокую точность.

- Необходимо добавить промежуточный (скрытый) слой Dense в архитектуру сети с 15 нейронами и проанализировать результаты.

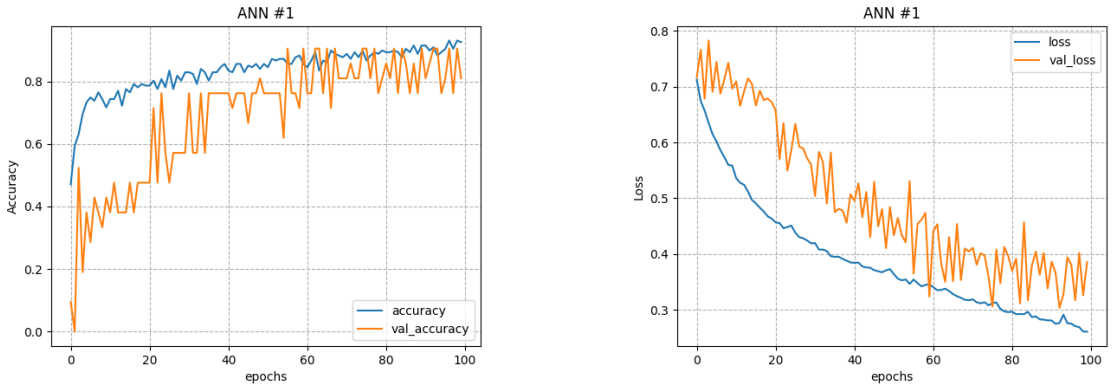
Ход работы

В ходе лабораторной работы были созданы и обучены 4 сети. Все сети имеют выходной слой с одним нейроном и функцией активации sigmoid. Компиляция выполняется с оптимизатором adam, функцией потерь перекрестная бинарная классификация и метрикой accuracy. Обучение происходит в течение 100 эпох, размером серии 10 и разбиением на тренировочные-валидационные данные в пропорциях 90 к 10. Различающиеся параметры представлены в таблице 1.

Таблица 1: Слои сетей

№ сети	Ф-ция акт-ции	Кол-во нейронов
1	relu	60
2	relu	30
3	relu	30
	relu	15
4	relu	60
	relu	15

При обучении первой сети были получены графики, представленные на рисунке 1.

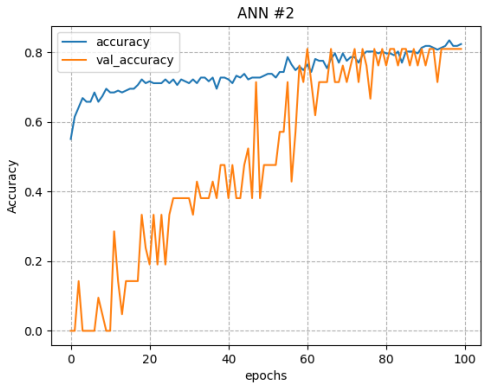


(a) Точность (b) Потери

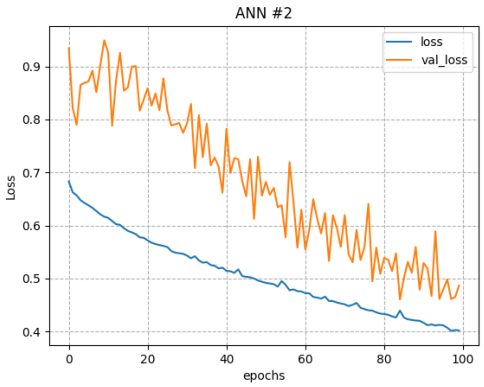
Рис. 1: Графики для ИНС #1

По графикам можно заметить, что сеть смогла достичь точности примерно 85%.

Затем было уменьшено число исследуемых признаков до 30. При обучении второй сети были получены графики, представленные на рисунке 2.



(a) Точность

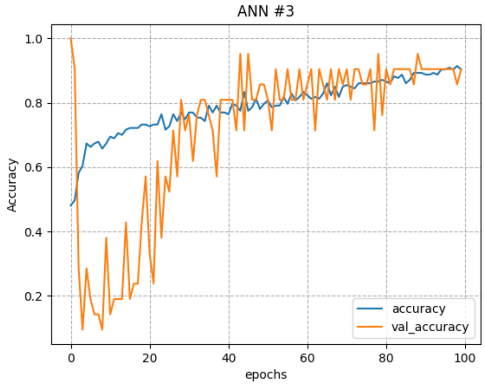


(b) Потери

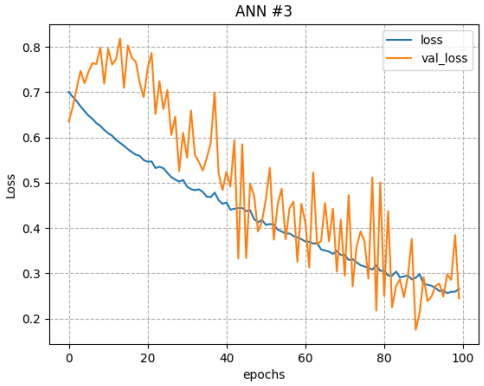
Рис. 2: Графики для ИНС #2

Судя по полученным графикам, можно сделать вывод, что данные избыточны, так как при уменьшении количества исследуемых признаков в два раза, точность практически не изменилась.

При обучении третьей сети были получены графики, представленные на рисунке 3.



(a) Точность



(b) Потери

Рис. 3: Графики для ИНС #3

Так как при добавлении еще одного слоя удалось увеличить показатель точности до примерно 90%, можно сделать вывод, что были обнаружены закономерности не только во входных данных, но и также в их комбинации.

Вывод

В ходе лабораторной работы были обучены три модели. Первая модель имела всего два слоя: входной в 60 нейронов и выходной в 1 сигмоидальный нейрон. Данная модель за 100 эпох достигла точности в примерно 80%. Вторая модель отличалась от первой количеством нейронов во входном слое. В результате обучения были получены результаты аналогичные результатам первой модели, что сказало об избыточности данных. В третью модель был добавлен скрытый слой в 15 нейронов. Метрика данной модели оказалась лучшей среди всех рассмотренных архитектур.