

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

ОТЧЕТ
по практической работе №2
по дисциплине «Искусственные нейронные сети»
Тема: Создание простой нейронной сети с использованием библиотеки
Keras
Вариант 3

Студентка гр. 8383

Максимова А.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы.

Реализовать модель ИНС, способную провести бинарную классификацию по сгенерированным данным.

Данные.

```
def genData(size=500):  
    data = np.random.rand(size, 2)*2 - 1  
    label = np.zeros([size, 1])  
    for i, p in enumerate(data):  
        if (p[0]+0.2)**2 + (0.6*p[1])**2 >= 0.25:  
            label[i] = 0.  
        else:  
            label[i] = 1.  
    div = round(size*0.8)  
    train_data = data[:div, :]  
    test_data = data[div:, :]  
    train_label = label[:div, :]  
    test_label = label[div:, :]  
    return (train_data, train_label), (test_data, test_label)
```

В данной функции генерируется массив размера (size, 2), состоящий из случайных чисел в интервале от 0 до 1 (далее значение удваивается и вычитается 1). После элементы каждой строки подставляются в уравнение и в зависимости от полученного значения распределяются по классам. В конце данные разделяются на обучающие и тестовые подмножества.

Модель ИНС.

```
#создание последовательной модели  
model = Sequential()  
  
#добавление слоев  
model.add(Dense(16, activation='relu', input_shape=(2,)))  
model.add(Dense(16, activation='relu'))  
model.add(Dense(1, activation='sigmoid'))  
  
#настройка обучения модели  
model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['accuracy'])  
  
#обучение  
hist = model.fit(train_data, train_label, epochs=90, batch_size=15, validation_split=0.2) |  
#hist - словарь с данными обо всем происходившем во время обучения
```

Для данной задачи было использовано 2 полносвязных слоя с 16 нейронами - функция активации: relu; 1 выходной слой с 1 нейроном - функция активации: sigmoid.

Настройка обучения модели: в качестве метода оптимизации используется RMSProp, функция потерь - бинарная кросс-энтропия (в основном используется при бинарной классификации), метрика - точность.

На этапе обучения были подобраны значения параметров epochs, batch_size и validation_split, при которых НС показывает лучшие результаты (минимальная ошибка и максимальная точность).

Результаты тестирования.

size = 500

График ошибок:

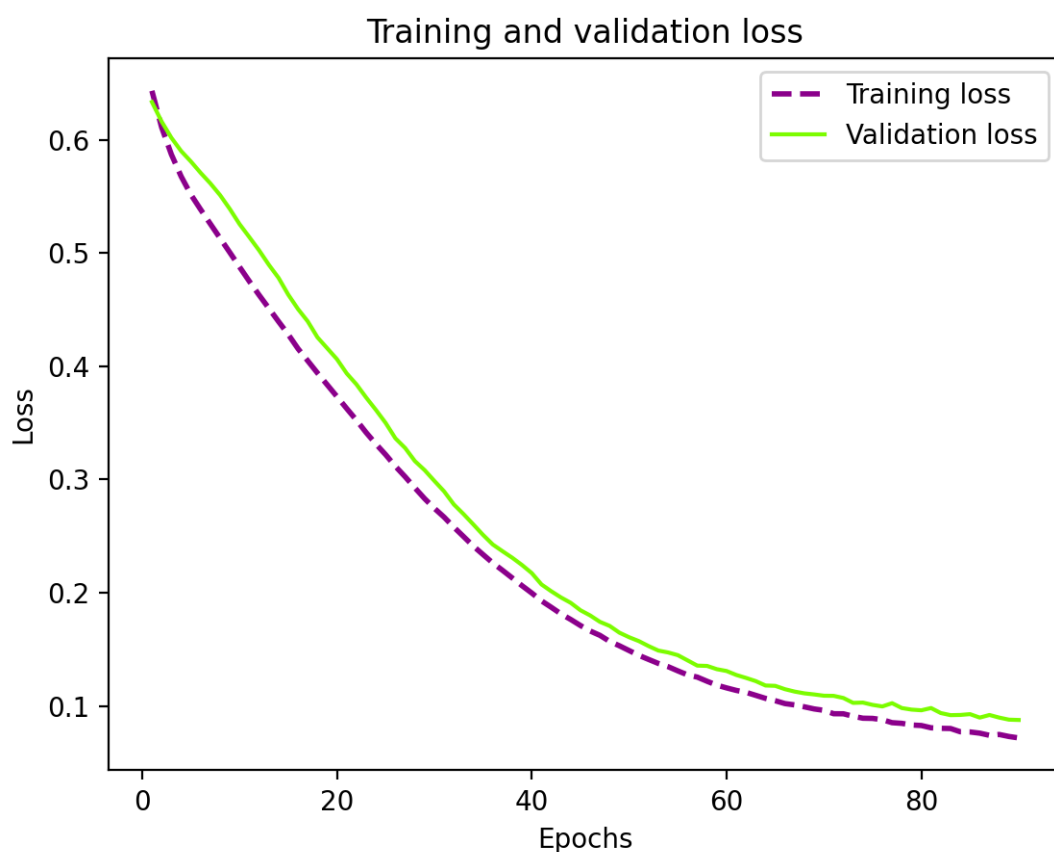
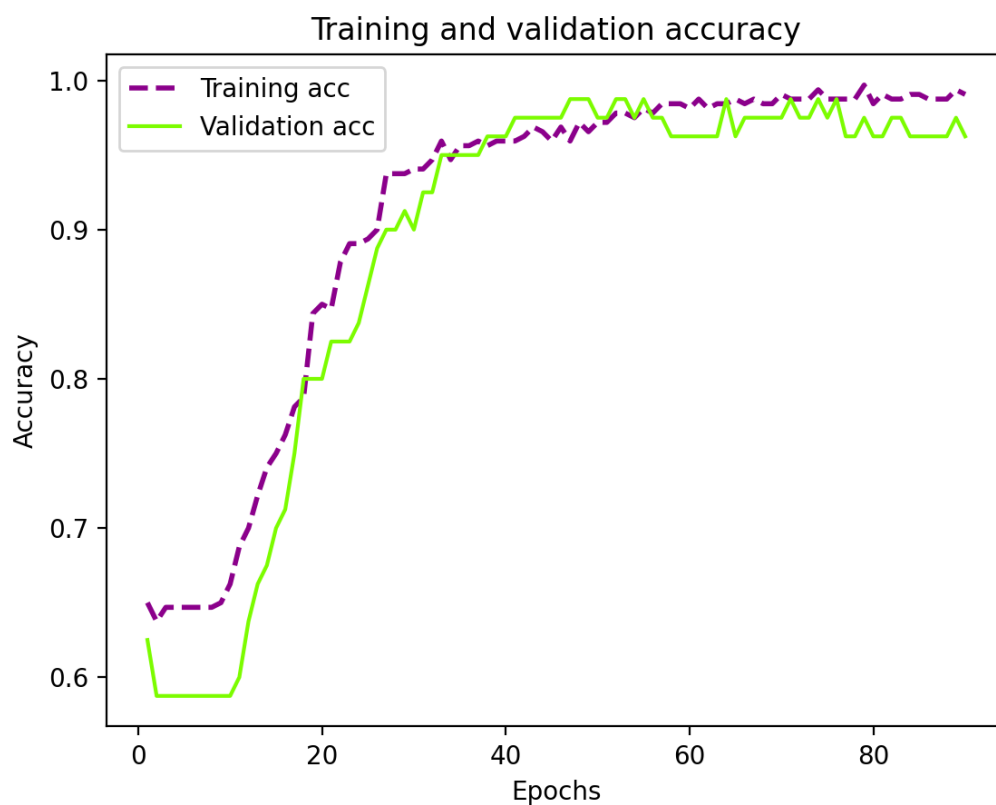


График точности:



Оценка модели на контрольных данных:

```
[0.09522011131048203, 0.9700000286102295]
```

График бинарной классификации:

