

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра математического обеспечения и применения ЭВМ**

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

Студент гр. 8383

\_\_\_\_\_

Костарев К.В.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2021

## Постановка задачи.

Создать модель сети с использованием библиотеки Keras для решения задачи бинарной классификации.

### Вариант 1.

#### Функция генерации данных:

```
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] + .5 >= p[1]) and (p[0] - 0.5 <= p[1]):
            label[i] = 1.
        else:
            label[i] = 0.
    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)
```

## Выполнение работы.

В данном случае была выбрана модель с двумя полносвязными слоями с функцией активации Relu по 32 и 16 нейронов соответственно. Третий слой содержит единственный нейрон для классификации с функцией активации сигмоида.

```
model = models.Sequential()
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))
```

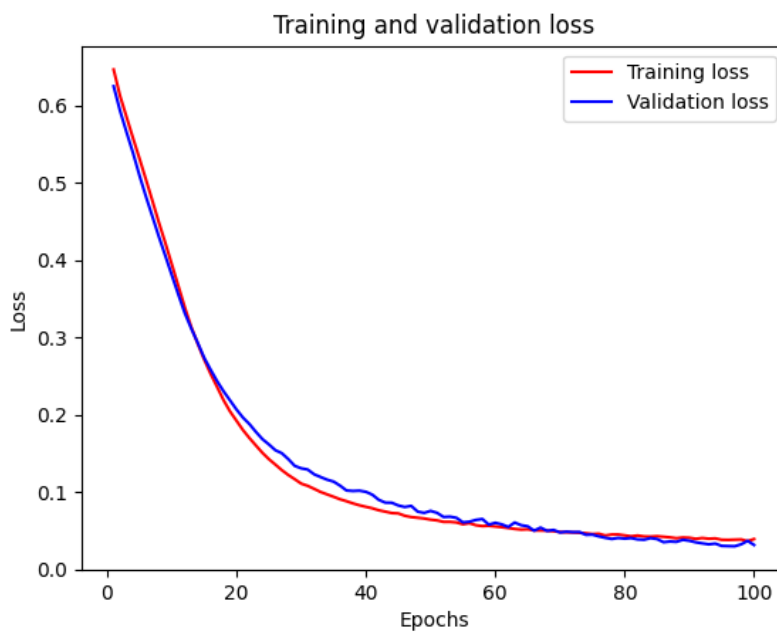
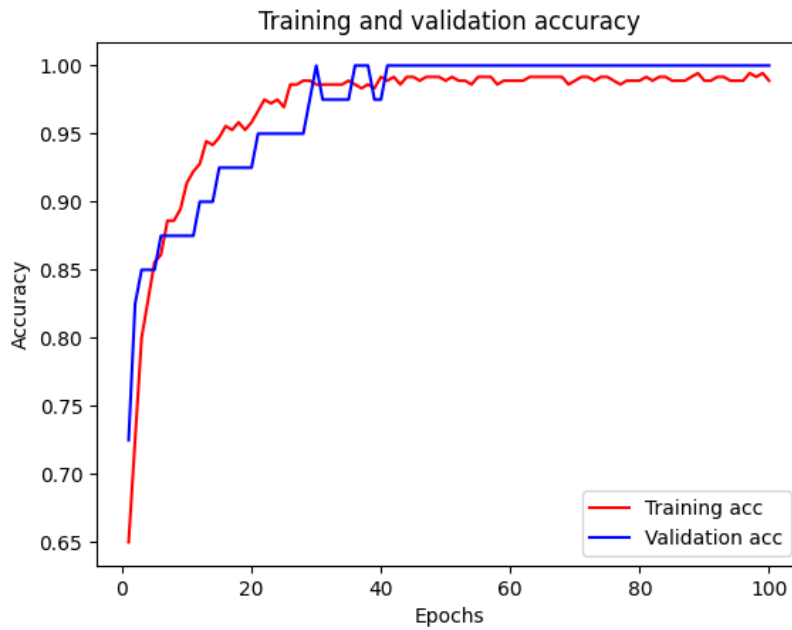
В качестве функции оптимизации выбрана RMSProp, функции потерь – бинарная кроссэнтропия, метрика – точность.

```
model.compile(optimizer='rmsprop', loss='binary_crossentropy',
metrics=['accuracy'])
```

Обучение проводится в течение 100 эпох, с размером выборки 20 и 10% обучающих данных.

```
H = model.fit(train_data, train_label, epochs=100, batch_size=20,
validation_split=0.1)
```

В результате такой настройки в течение второй половины эпох точность не опускается ниже 100%.



На визуальном распределении видно, что есть 5 ошибок на пограничных областях.

