

Аверина Ольга, гр. 8383, Вариант 1

## Задание

Необходимо дополнить следующий фрагмент кода моделью ИНС, которая способна провести бинарную классификацию по сгенерированным данным.

### Вариант 1:

```
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

Была создана модель с тремя слоями: два скрытых слоя с 20 и 15 нейронами и функцией активации relu и один выходной слой с 1 нейроном и сигмоидной функцией активации:

```
model = Sequential() model.add(Dense(20,  
activation='relu'))  
model.add(Dense(15, activation='relu'))  
model.add(Dense(1, activation='sigmoid'))
```

Для настройки обучения модели была использована функция compile:

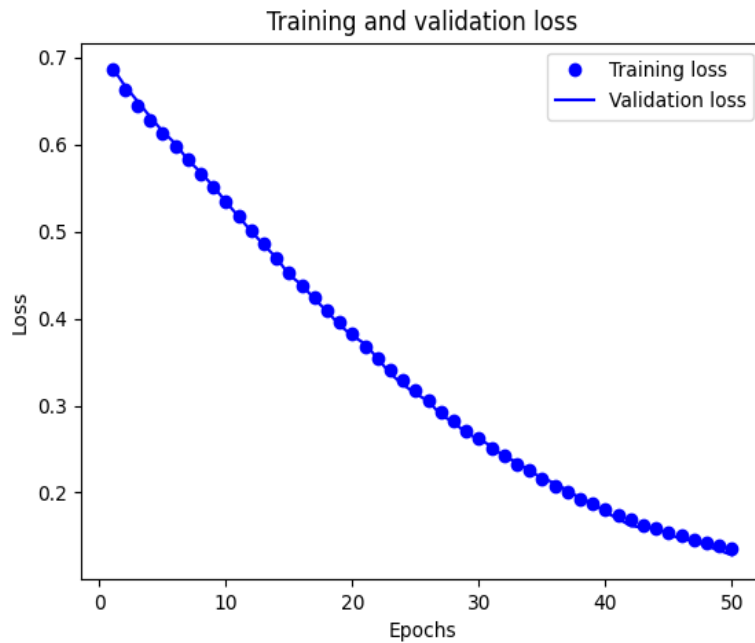
```
model.compile(optimizer='rmsprop', loss='binary_crossentropy',  
metrics=['accuracy'])
```

Обучение модели происходило за 50 эпох батчами по 32 образа:

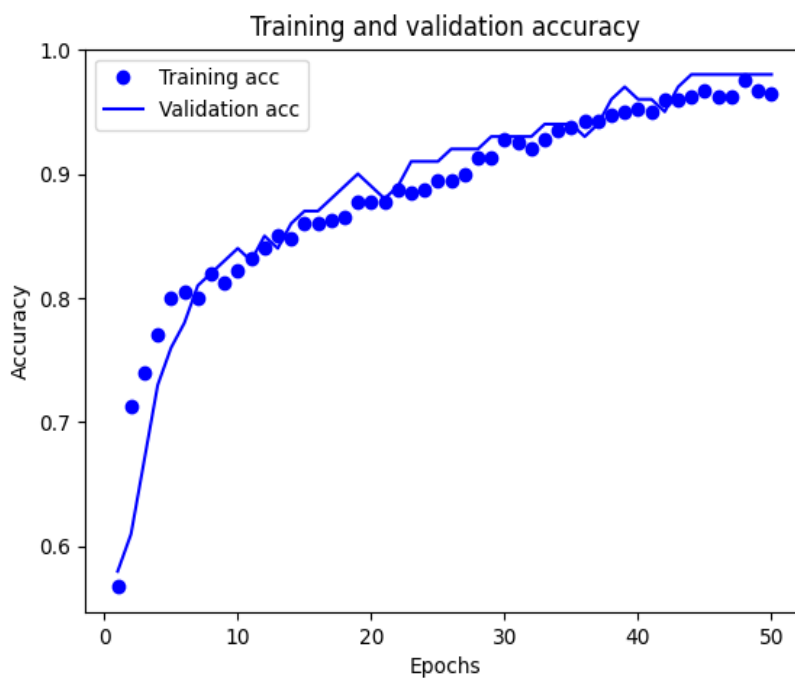
```
H = model.fit(train_data, train_label,
```

```
batch_size=32,  
epochs=50,  
validation_data=(test_data, test_label), verbose=0)
```

## График ошибок



## График точности



Для оценки модели после обучения на тестовых данных была использована функция `evaluate`:

```
results = model.evaluate(test_data, test_label)
```

Результаты:

4/4 [=====] - 0s 520us/step - loss: 0.1284 - accuracy: 0.9800

[0.12837395071983337, 0.9800000190734863]

При 10ти запусках программы максимальное значение точности было 98%, минимальное — 90%.

График бинарной классификации

