

## Задание №2, вариант №1

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

Генерация 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)
```

## Решение

Была написана следующая модель:

```
model.add(Dense(2, activation="relu"))
model.add(Dense(36, activation="relu"))
model.add(Dense(36, activation="relu"))
model.add(Dense(1, activation="sigmoid"))
```

Как мы видим, она состоит из входного слоя из 2 нейронов (функция активации RELU), двух скрытых слоев с 36 нейронами на каждой (функция активации RELU), а также выходным слоем с 1 нейроном и функцией активации сигмоидой.

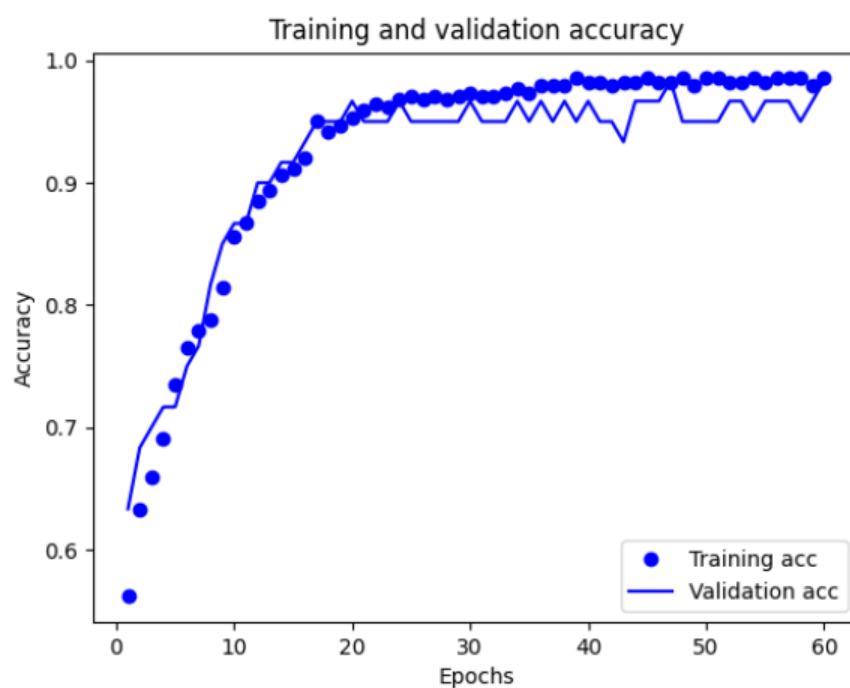
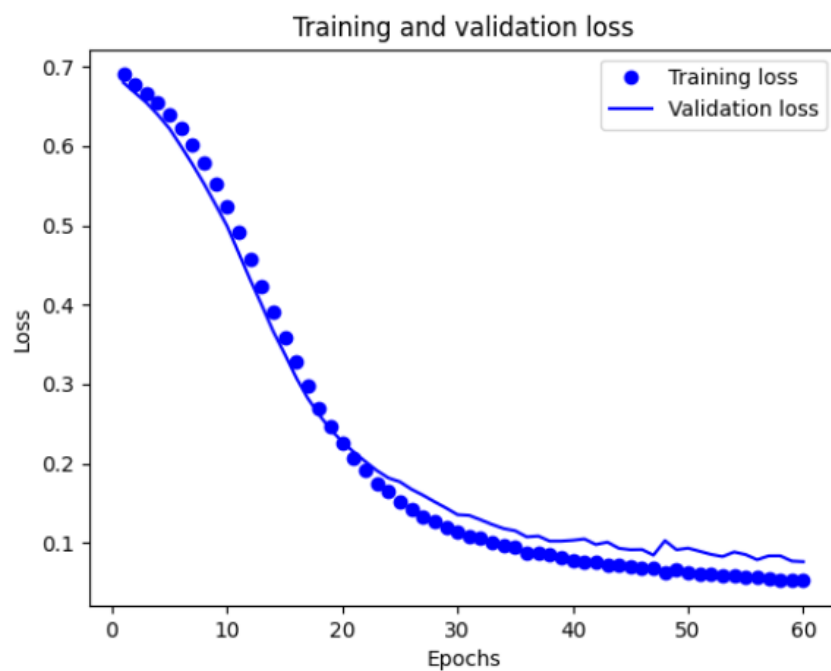
Инициализация параметров обучения была следующая:

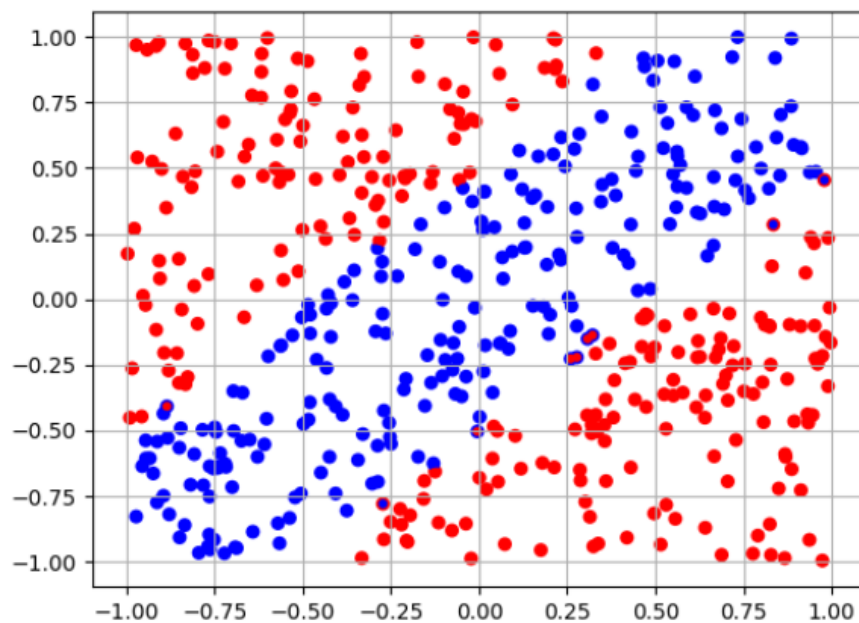
```
model.compile(optimizer='rmsprop', loss='binary_crossentropy',
metrics=['accuracy'])
```

Обучение проводится в течение 60 эпох пакетами по 15 образцов, для валидации отведено 15% данных:

```
model.fit(train_data, train_label, epochs=60, batch_size=15,
validation_split=0.15)
```

Далее приведены графики потерь, точности и результата бинарной классификации:





Для оценки модели после ее обучения на тестовых данных, использовалась функция `evaluate`:

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

Для данной модели точность составила 97%  
[0.07575623691082001, 0.9700000286102295]

Данная модель показала стабильность в обучении и её точность после многочисленных запусков в среднем колебалась между 89-99%.