

## Постановка задачи.

Создать модель сети для выполнения бинарной классификации.

## Выполнение работы.

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)
```

Выбрана модель с двумя полносвязными слоями по 32 нейрона:

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

Установлен оптимизатор RMSProp, функция потерь – бинарная кросс-энтропия, метрика – точность

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

Проведено обучение в 50 эпох, с батчем размером 10, 12.5% данных отведены под валидацию:

```
history = model.fit(train_data, train_label, epochs=50, batch_size=10,
validation_split=0.125)
```

Результат запуска на тестовых данных:

4/4 [=====] - 0s 990us/step - loss: 0.0392 -  
accuracy: 0.9900  
[0.03915812820196152, 0.9900000095367432]

При запусках программы точность держится на уровне 98-100%

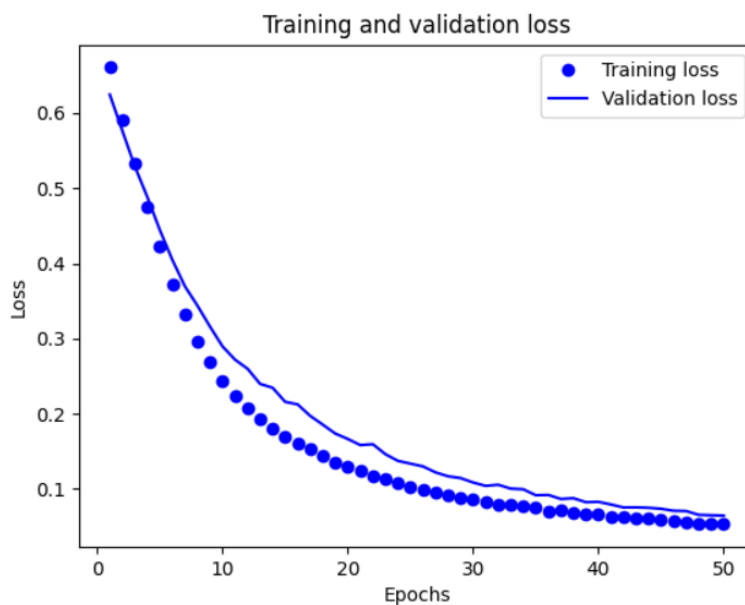


Рисунок 1. Потери

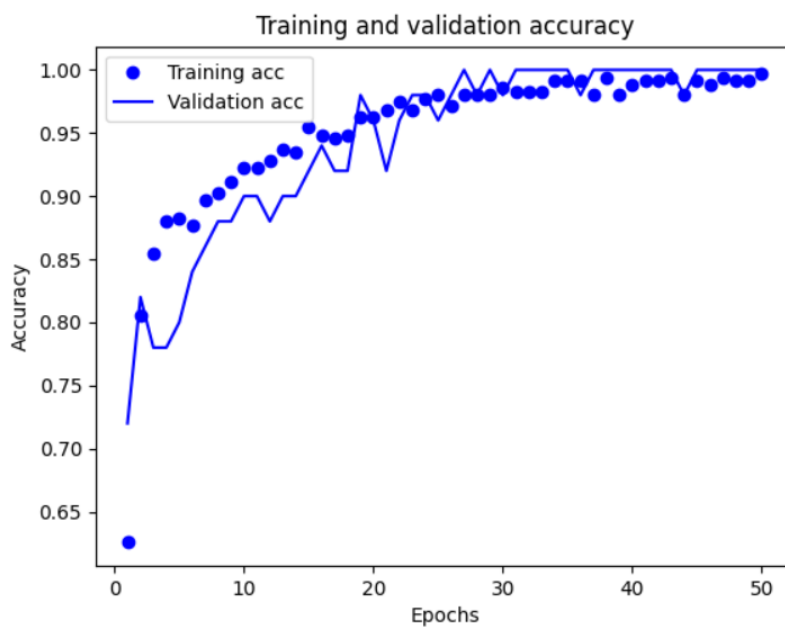


Рисунок 2. Точность

Визуальное распределение точек, на котором видно три ошибки в пограничных областях:

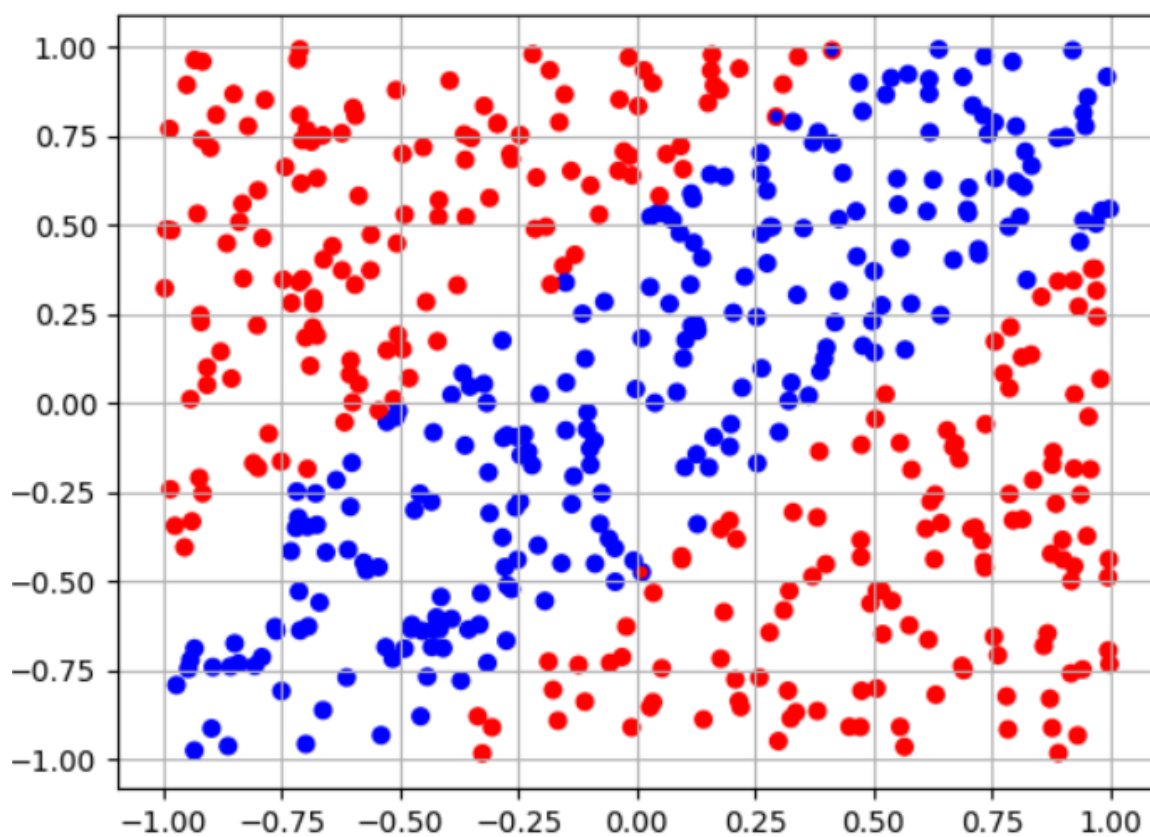


Рисунок 3. Визуальное распределение