

Задание

Создание модели ИНС, которая способна провести бинарную классификацию по сгенерированным данным

Выполнение работы

Функция генерации данных – 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)
```

Для решения поставленной задачи была создана модель ИНС с тремя слоями: два скрытых слоя с 35 нейронами, использующих функцию активации relu и выходной слой с одним нейроном и функцией активации sigmoid.

```
model = Sequential()
model.add(Dense(35, activation="relu"))
model.add(Dense(35, activation="relu"))
model.add(Dense(1, activation="sigmoid"))
```

В качестве функции потерь выбрана бинарная кросс-энтропия, метрика – точность.

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

Обучение модели проводится в течение 40 эпох пакетами по 10 образцов, 10% данных отводится на валидацию данных.

```
H = model.fit(train_data, train_label, epochs=40, batch_size=10,
validation_split=0.1)
```

На рисунках 1, 2 представлены графики потерь и точности данных – тренировочных и валидационных.

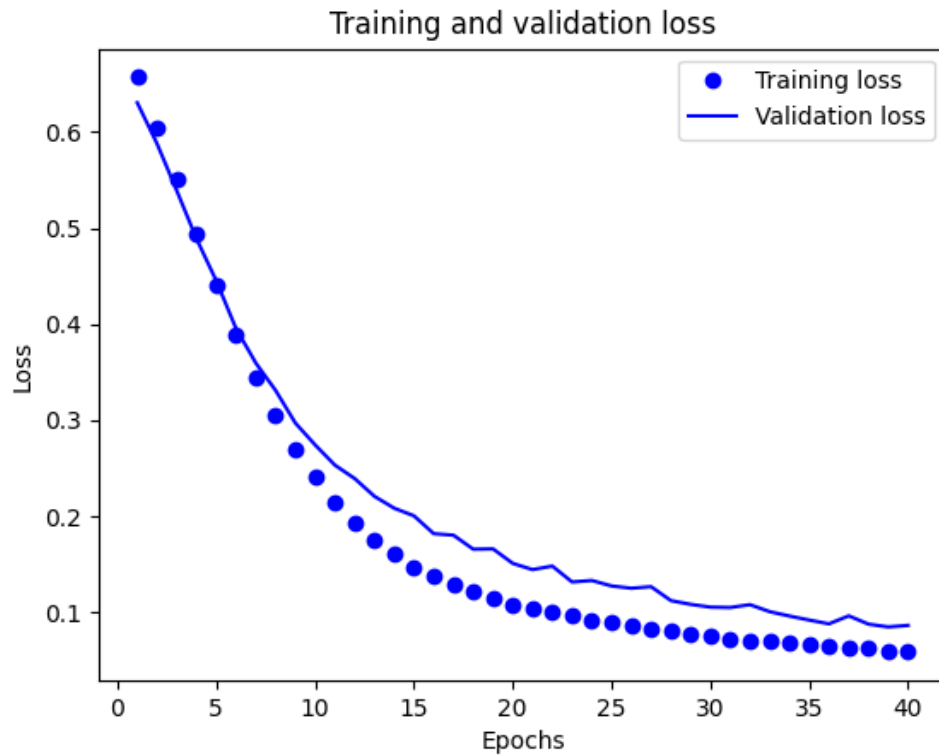


Рисунок 1 – Потери

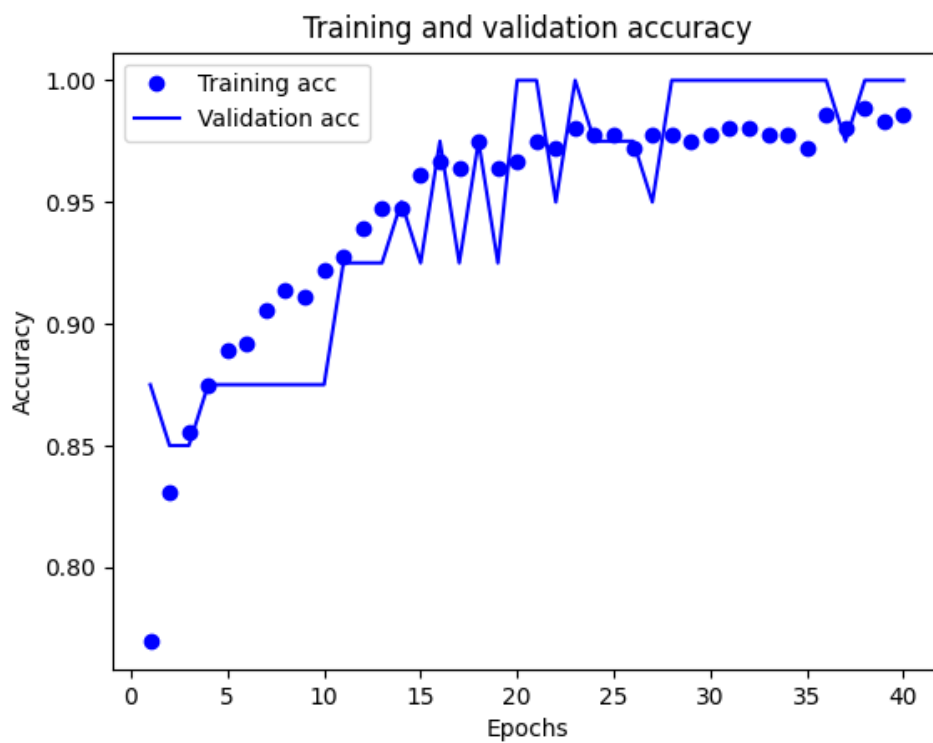


Рисунок 2 – Точность

После обучения модели происходит оценка по тестовым данным:

4/4 [=====] - 0s 1000us/step - loss:
0.0508 - accuracy: 1.0000
[0.05079331248998642, 1.0]

При повторных запусках программы точность не опускалась ниже 97%, при этом точность 99% - встречалась чаще всего.

На рисунке 3 приведено визуальное бинарное распределение точек.

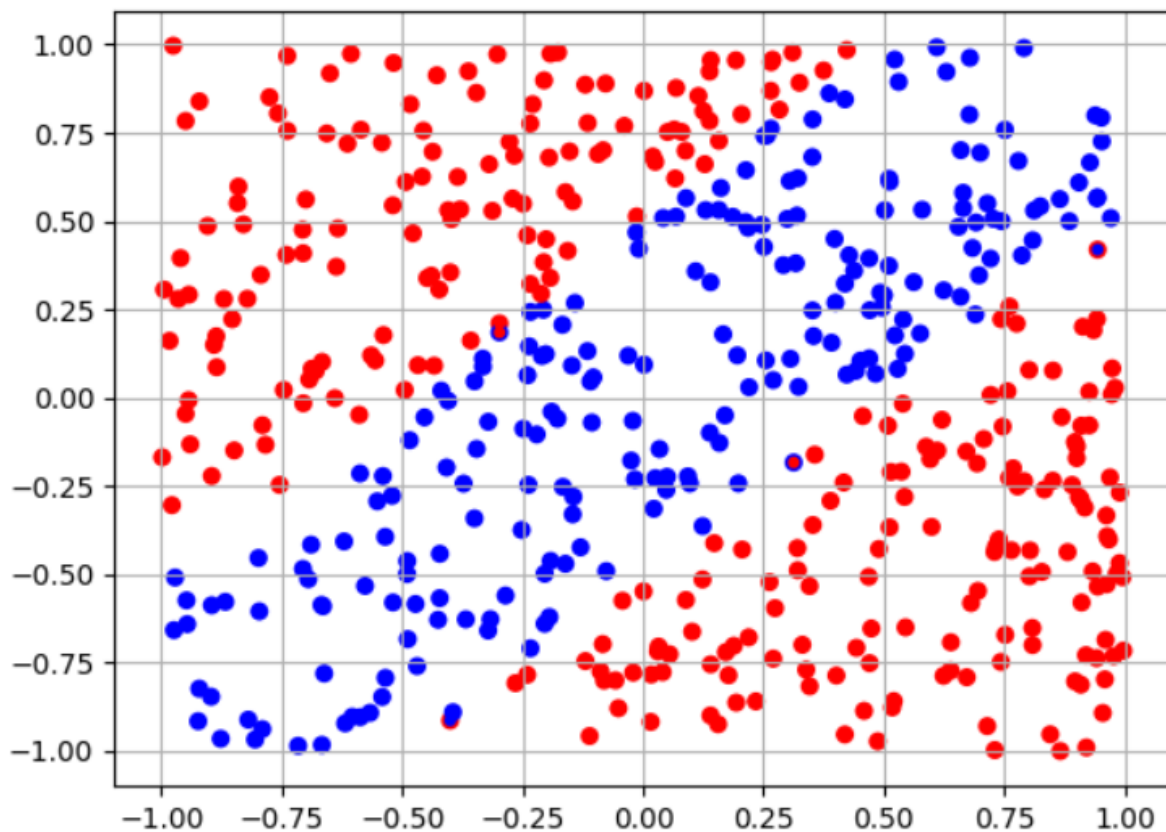


Рисунок 3 – Бинарное распределение точек