

Практическое задание №2

Задача №5

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

```
def genData(size=500):
    size1 = size//2
    size2 = size - size1
    x1 = np.random.rand(size1, 1)*1.3 - 0.95
    y1 = np.asarray([3.5*(i+0.2)**2 - 0.8 + (np.random.rand(1)-
0.5)/3 for i in x1])
    data1 = np.hstack((x1, y1))
    label1 = np.zeros([size1, 1])
    div1 = round(size1*0.8)
    x2 = np.random.rand(size2, 1)*1.3 - 0.35
    y2 = np.asarray([-3.5*(i-0.2)**2 + 0.8 + (np.random.rand(1)-
0.5)/3 for i in x2])
    data2 = np.hstack((x2, y2))
    label2 = np.ones([size2, 1])
    div2 = round(size2*0.8)
    div = div1 + div2
    order = np.random.permutation(div)
    train_data = np.vstack((data1[:div1], data2[:div2]))
    test_data = np.vstack((data1[div1:], data2[div2:]))
    train_label = np.vstack((label1[:div1], label2[:div2]))
    test_label = np.vstack((label1[div1:], label2[div2:]))
    return (train_data[order, :], train_label[order, :]), (test_data,
test_label)
```

Выполнение: была составлена следующая модель ИНС:

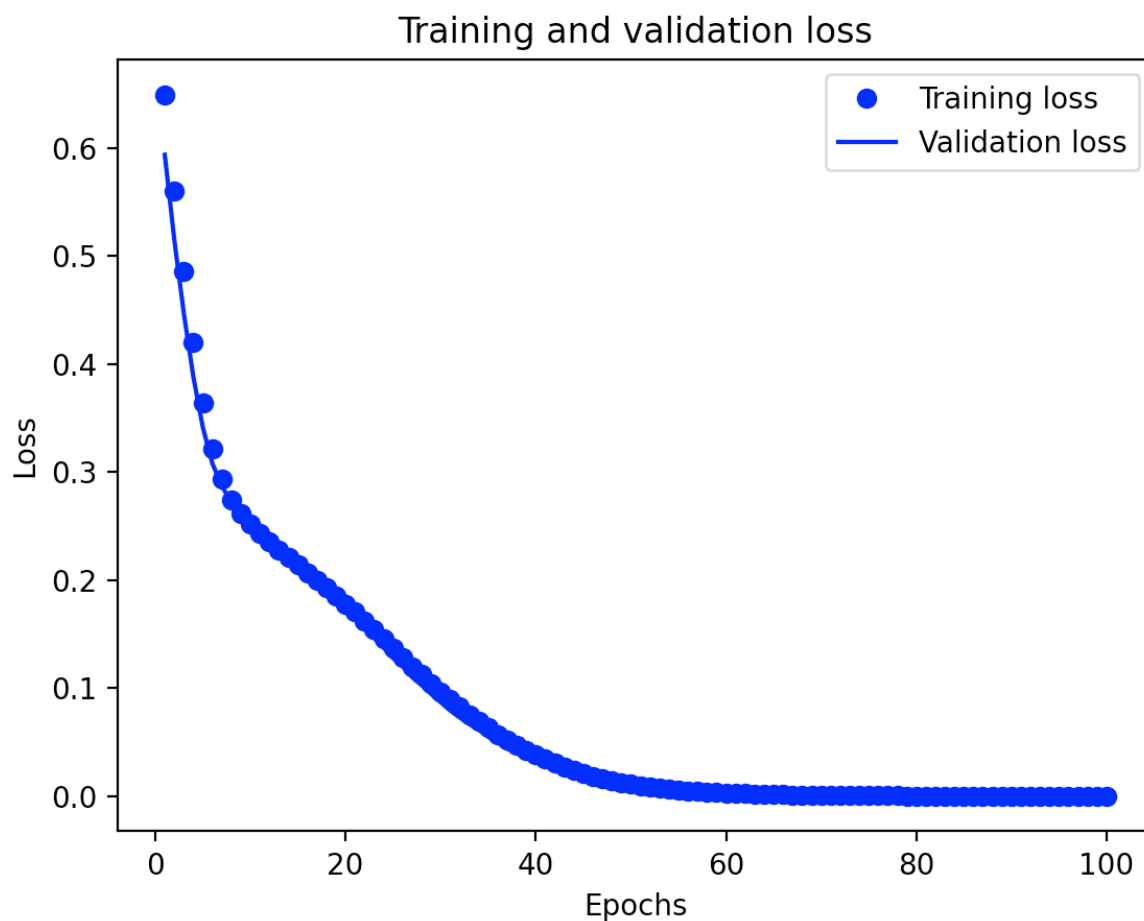
```
model = models.Sequential();
#Добавление слоев
model.add(layers.Dense(32, activation='relu', input_shape=(2,)))
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))
model.compile(optimizer='rmsprop', loss='binary_crossentropy',
metrics=['accuracy'])
H = model.fit(train_data, train_label, epochs=100, batch_size=20,
validation_data=(test_data, test_label))
```

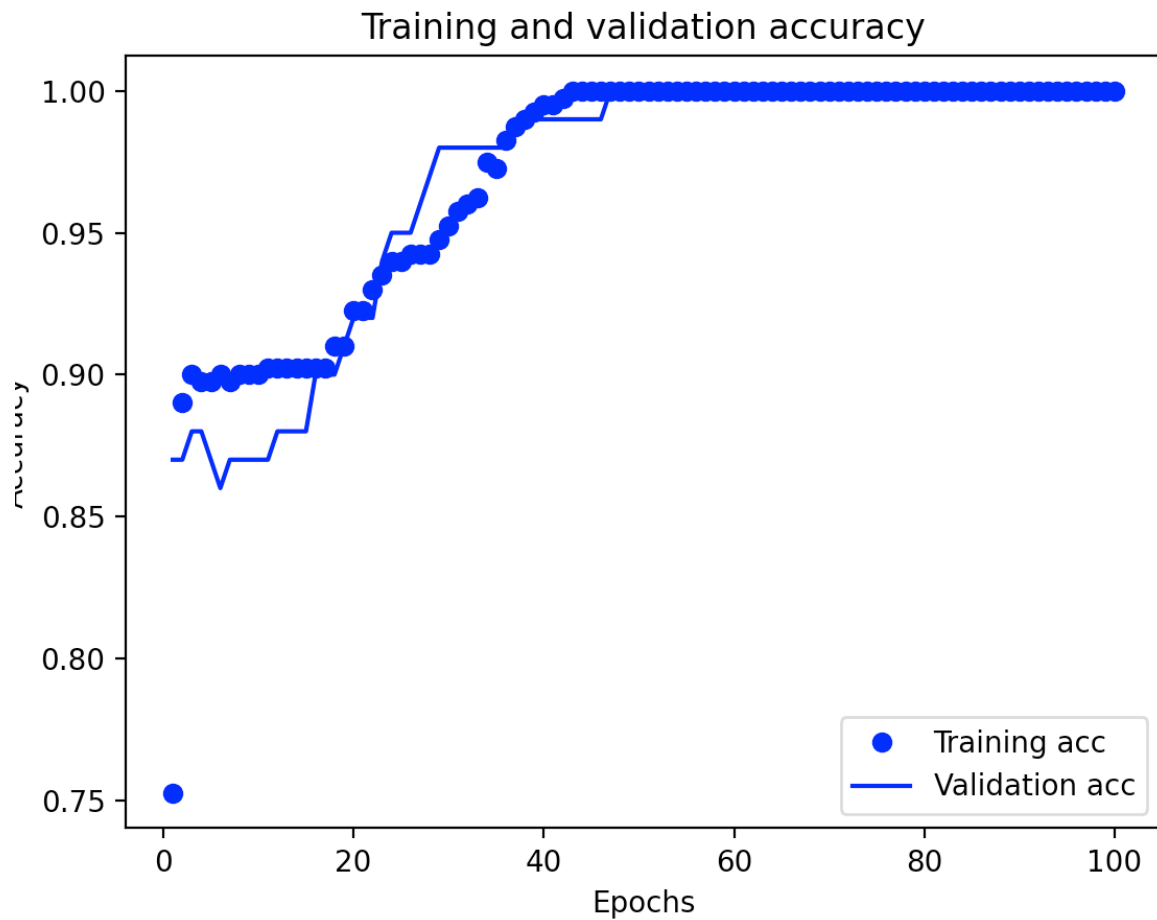
Параметр `input_shape = 2` т.к. в данных 2 столбца (x,y). На первый двух слоях количество нейронов равно 32 (опытным путем было выяснено, что это оптимальное значение), операция активации – `relu`. На последнем, выходном слое используется активирующая функция `sigmoid`.

Характеристики обучения: в качестве оптимизатора используется RMSProp, в качестве функции потерь – бинарная кросс-энтропия, в качестве метрики – точность.

Обучение модели проводится в течение 100 эпох, размер батча установлен в 20 образцов.

При запуске модели были получены следующие результаты:





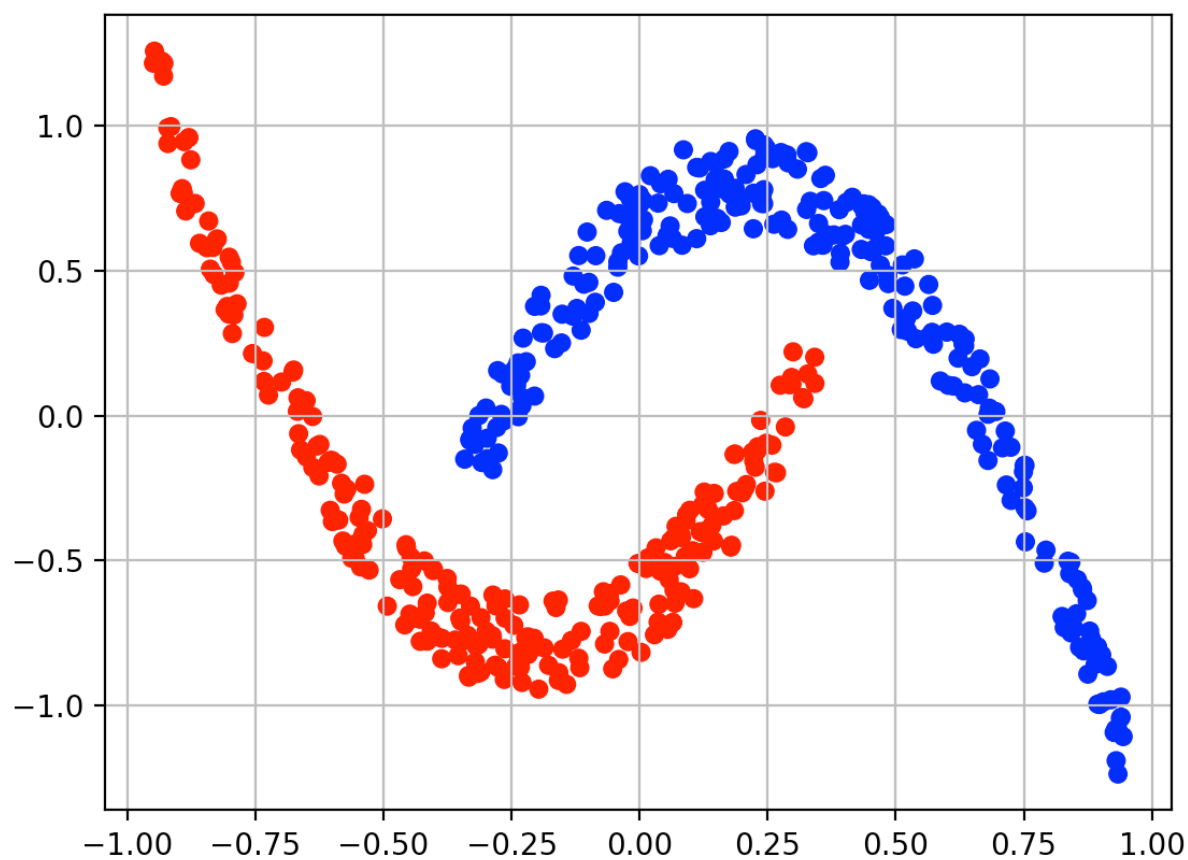
Нейросеть начинает давать маленькие потери и высокую точность примерно после 60 эпох.

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

Данная функция показывает следующее:

[0.00020140867854934186, 1.0], где 0.0002 – потери, а 1.0- точность, что говорит о очень хороших результатах.

График бинарной классификации:



Видно, что точность ИНС очень велика.