Вариант 5

(a **xor** b) **and** (b **xor** c)

Выполнение работы.

Была реализована функция f1, в которой все операции реализованы как поэлементные операции над тензорами

```
def f1 (weigts, data):

    layerPrevious = data

    for index in range(weigts.shape[0] // 2):    # цикл проходится по каждому
из слоев

        layerCurr = np.zeros(weigts[index*2].shape[1])  # создается пустой
слой

        for i in range(weigts[index*2].shape[1]):    # цикл проходится по
индексам нейронов нынешнего слоя

            for j in range(weigts[index*2].shape[0]):    # цикл проходится по
индексам нейронов предыдущего слоя
                layerCurr[i] += layerPrevious[j] * weigts[index*2][j][i]    #
сумма весов и значений предыдущего нейрона
            layerCurr[i] += weigts[index*2+1][i]    # прибаление нейрона
смещения

            if (index+1 != weigts.shape[0] // 2): # для послднего используем
функцию активации sigmoid для остальных relu
                layerCurr[i] = max(layerCurr[i], 0) # relu
            else:
                layerCurr[i] = 1.0/(1.0 + np.exp(-layerCurr[i]))  # sigmoid

        layerPrevious = layerCurr

    return layerPrevious
```

Была реализована функция f2, , в которой все операции реализованы с использованием операций над тензорами из NumPy

```
def f2(weigts, data):

    layerPrevious = data

    for i in range(weigts.shape[0] // 2):    # цикл проходится по каждому из
слоев

        layerCurr = np.dot(layerPrevious, weigts[i*2]) + weigts[i*2 + 1]    #
сумма весов, значений предыдущего нейрона
                                                                            #
и нейронов смещения
        if (i+1 != weigts.shape[0] // 2):    # для послднего используем
функцию активации sigmoid для остальных relu
            layerCurr = np.maximum(layerCurr, 0)    # relu
        else:
            layerCurr = 1.0/(1.0 + np.exp(-layerCurr))    # sigmoid

        layerPrevious = layerCurr
```

```
    return layerPrevious
```

Была реализована модель нейронной сети. Описание модели:

- Входной слой имеет 3 нейрона
- Скрытые слои имеют по 32 нейрона, функция активации relu
- Выходной слой имеет 1 нейрон, функция активации sigmoid
- На скрытых и выходном слое включен параметр нейрона смещения

Датасет:

```
train_data = np.asarray([   [0, 0, 0],
                            [0, 0, 1],
                            [0, 1, 0],
                            [0, 1, 1],
                            [1, 0, 0],
                            [1, 0, 1],
                            [1, 1, 0],
                            [1, 1, 1]])

# (a xor b) and (b xor c)
train_label = np.asarray([0,0,1,0,0,1,0,0])
```

Прогоним датасет через необученную сеть и функции. Результаты запуска:

[0 0 0] = 0
model = 0.5
f1 = 0.5
f2 = 0.5
[0 0 1] = 0
model = 0.517202
f1 = 0.517202
f2 = 0.517202
[0 1 0] = 1
model = 0.538017
f1 = 0.538017
f2 = 0.538017
[0 1 1] = 0
model = 0.568462
f1 = 0.568462
f2 = 0.568462
[1 0 0] = 0
model = 0.430928
f1 = 0.430928
f2 = 0.430928
[1 0 1] = 1
model = 0.43863
f1 = 0.43863
f2 = 0.43863
[1 1 0] = 0
model = 0.476918
f1 = 0.476918

```
        f2    = 0.476918
[1 1 1]  = 0
        model =  0.50816
        f1    =  0.50816
        f2    =  0.50816
```

Прогоним датасет через обученную сеть и функции. Результаты запуска:

```
[0 0 0]  = 0
        model =  0.137098
        f1    =  0.137098
        f2    =  0.137098
[0 0 1]  = 0
        model =  0.043783
        f1    =  0.043783
        f2    =  0.043783
[0 1 0]  = 1
        model =  0.825718
        f1    =  0.825718
        f2    =  0.825718
[0 1 1]  = 0
        model =  0.050668
        f1    =  0.050668
        f2    =  0.050668
[1 0 0]  = 0
        model =  0.050854
        f1    =  0.050854
        f2    =  0.050854
[1 0 1]  = 1
        model =  0.561933
        f1    =  0.561933
        f2    =  0.561933
[1 1 0]  = 0
        model =  0.020324
        f1    =  0.020323
        f2    =  0.020323
[1 1 1]  = 0
        model =  0.118249
        f1    =  0.118249
        f2    =  0.118249
```

Видно, что результаты модели и функция не отличаются, что говорит о том, что функции правильно симулируют работу нейронной сети.