

Если заменить сигмоидную функцию активации на функцию активации ReLU:

```
def relu(x):  
    return np.maximum(0, x)  
  
def relu_deriv(x):  
    return np.where(x > 0, 1, 0)
```

```
# прямое распространение (feed forward)  
layer0 = X_train  
layer1 = relu(np.dot(layer0, w0))  
layer2 = relu(np.dot(layer1, w1))  
  
# обратное распространение (back propagation) с использованием градиентного  
спуска  
layer2_error = y_train - layer2  
layer2_delta = layer2_error * relu_deriv(layer2)  
  
layer1_error = layer2_delta.dot(w1.T)  
layer1_delta = layer1_error * relu_deriv(layer1)
```

тогда точность нейронной сети 97.03%. При сигмоидной функции – точность нейронной сети 98.59%.

Если заменим функцию активации на гиперболический тангенс:

```
def tanh(x):  
    return np.tanh(x)  
  
def tanh_deriv(x):  
    return 1 - np.tanh(x)**2  
  
# прямое распространение (feed forward)  
layer0 = X_train  
layer1 = tanh(np.dot(layer0, w0))  
layer2 = tanh(np.dot(layer1, w1))  
  
# обратное распространение (back propagation) с использованием градиентного  
спуска  
layer2_error = y_train - layer2  
layer2_delta = layer2_error * tanh_deriv(layer2)  
  
layer1_error = layer2_delta.dot(w1.T)  
layer1_delta = layer1_error * tanh_deriv(layer1)
```

тогда точность нейронной сети упадёт до 97.3%, выполнялось 16 мин. **Вернём сигмоидную функцию активации.**

Также я внес следующие изменения в нейросеть для повышения её точности:

скорость обучения (learning rate)

```
n = 0.0007
```

Шаг 3. Обучение нейронной сети

присваивание случайных весов

```
w0 = 2*np.random.random((4, 7)) - 1 # для входного слоя - 4 входа, 3 выхода
```

```
w1 = 2*np.random.random((7, 3)) - 1 # для внутреннего слоя - 5 входов, 3 выхода
```

(было 5 вместо 7)

Кол-во эпох

процесс обучения

```
for i in range(700_000): #100,000
```

Заменял веса

```
w0 = 2*np.random.random((4, 8)) - 1 # для входного слоя - 4 входа, 3 выхода
```

```
w1 = 2*np.random.random((8, 3)) - 1 # для внутреннего слоя - 5 входов, 3 выхода
```

Количество эпох сделал

```
for i in range(1_000_000): #100,000
```

Точность нейронной сети 97.34%

Количество эпох сделал

```
for i in range(8_000_000)
```

 – точность нейронной сети составила 97.76%

Вернул исходный код (как в лекции). Точность нейронной сети 98.07%

```
w0 = 2*np.random.random((4, 10)) - 1 # для входного слоя - 4 входа, 10 выходов
```

```
w1 = 2*np.random.random((10, 3)) - 1 # для внутреннего слоя - 10 входов, 3 выхода
```

скорость обучения (learning rate)

```
n = 0.01
```

количество эпох

```
for i in range(500_000): #100,000
```

Выполняется с 18.09 до 18.20. Точность нейронной сети 98.49%

Сделал

количество эпох

```
for i in range(10_000_000): #100,000
```

Выполняется с 18.23 довольно долго, поэтому лучше сдать ДЗ.

