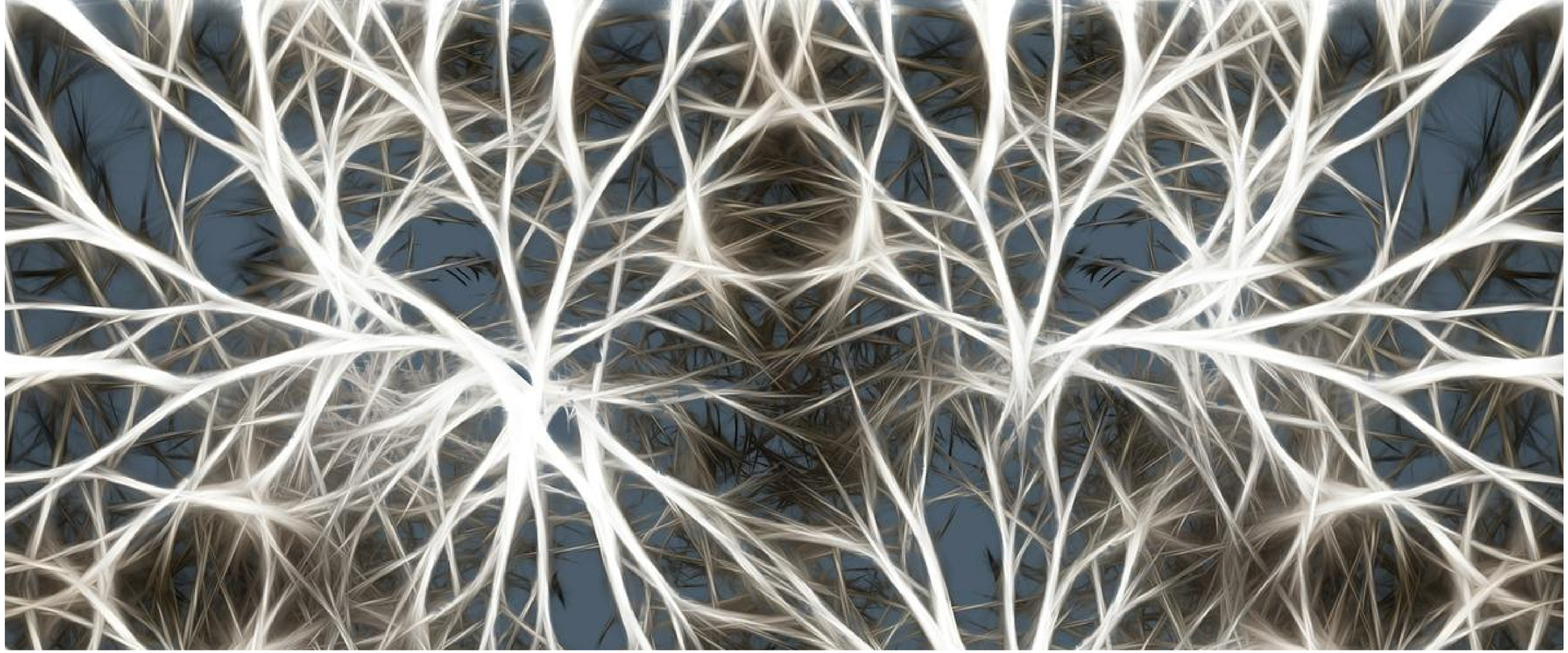


Training of a Neural Networks

Viacheslav Kovalevskyi

0.25

0.25

0

0.5

-0.4

0.9

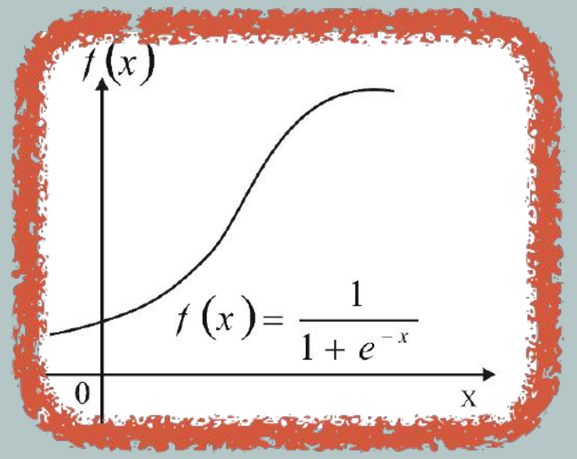
$$f(x) = \begin{cases} 1 & x \geq 0.5 \\ 0 & x < 0.5 \end{cases}$$

-1

1

1

0



0.25

0.25

0

0.5

-0.4

0.9

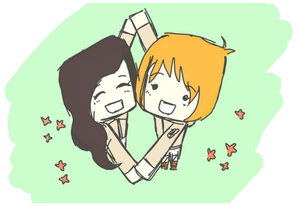
-1

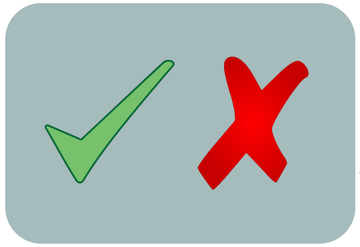
1

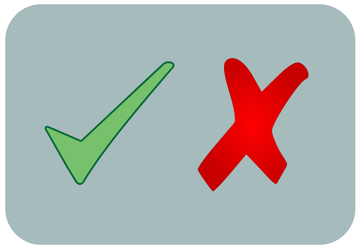
1

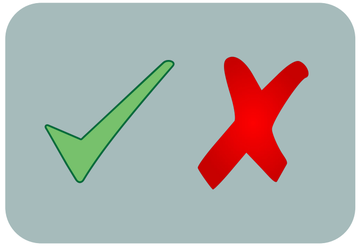
0


VODKA









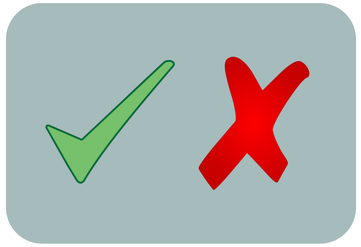
0.79

0.44

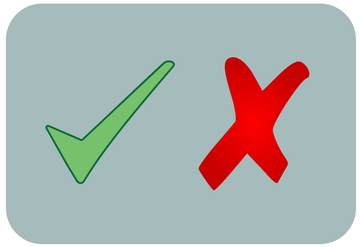
0.85

0.43

0.29

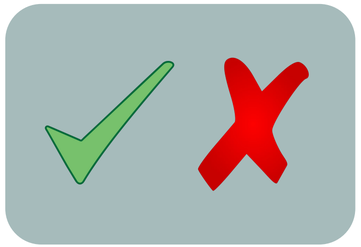


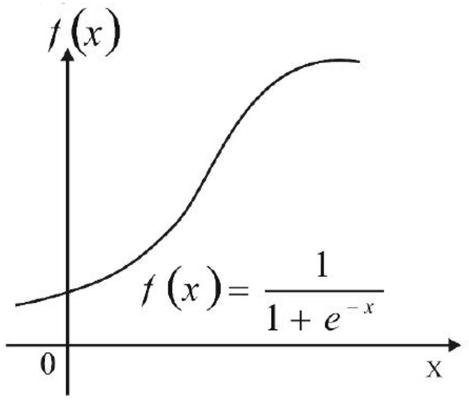
0.43



0.5

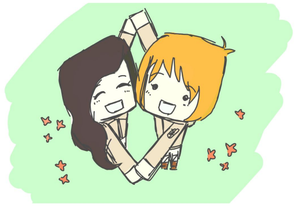
0.52





1

0



1*0.79 + 1*0.44 + 0*0.43 = 1.23

sigmoid(1.23) = 0.77

0.79

0.44

0.43

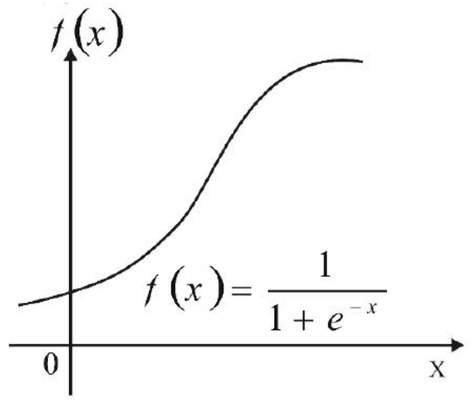
0.85

0.43

0.29

0.77

0.78



0.5

0.52

0.69

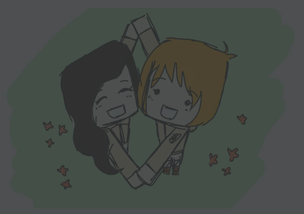
expected

0.77*0.5 + 0.78*0.52 = 0.79

sigmoid(0.79) = 0.69

1*0.85 + 1*0.43 + 0*0.29 = 1.28

sigmoid(1.28) = 0.78



0.79

0.44

0.43

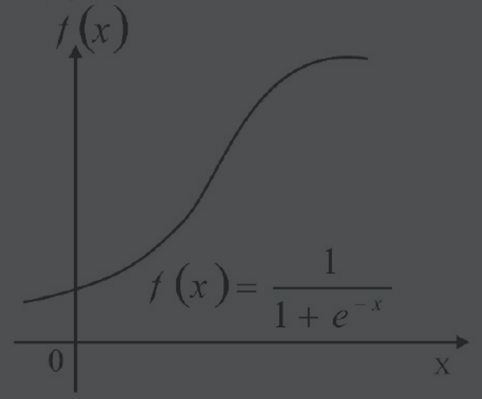
0.85

0.43

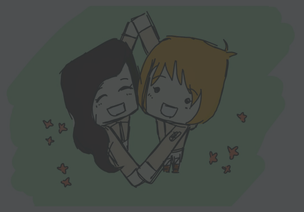
0.29

0.5

0.52



expected :



0.79

0.44

0.43

0.85

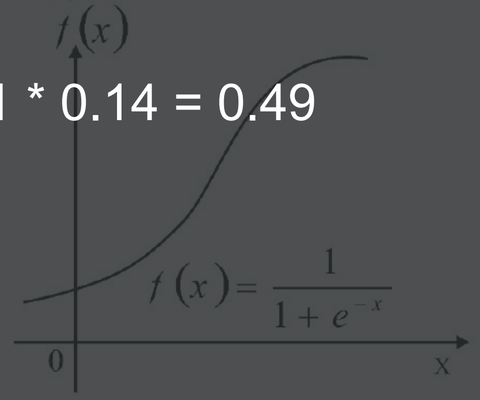
0.43

0.29

0.77

0.78

weight_1=weight_1 - output1 * weights_delta * learning_rate
learning_rate=0.1
output1 = 0.77
weight_1=0.5 - 0.77 * 0.1 * 0.14 = 0.49



0.49

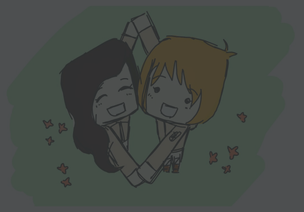
0.52

0.69*

**sigmoid(0.79)*

expected
:

error = actual - expected
error = 0.69 - 0 = 0.69
weights_delta= error * sigmoid(x)dx
sigmoid(x)dx=sigmoid(x)(1 - sigmoid(x))
sigmoid(x)dx=0.69 * (1 - 0.69)= 0.21
weights_delta=0.69 * 0.21 = 0.14



0.79

0.44

0.43

0.85

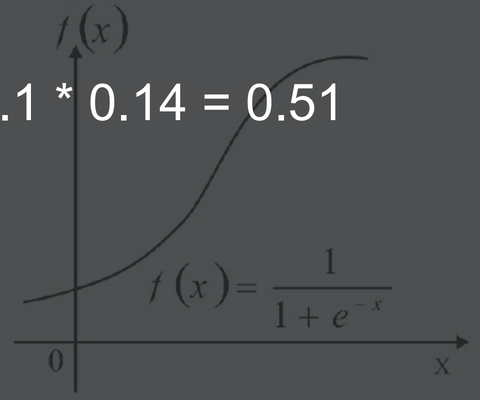
0.43

0.29

0.77

0.78

weight_2=weight_2 - output2 * weights_delta * learning_rate
learning_rate=0.1
output2 = 0.78
weight_2=0.52 - 0.78 * 0.1 * 0.14 = 0.51



0.49

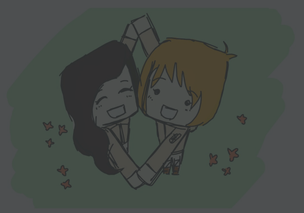
0.52

0.69*

**sigmoid(0.79)*

expected
:

error = actual - expected
error = 0.69 - 0 = 0.69
weights_delta= error * sigmoid(x)dx
sigmoid(x)dx=sigmoid(x)(1 - sigmoid(x))
sigmoid(x)dx=0.69 * (1 - 0.69)= 0.21
weights_delta=0.69 * 0.21 = 0.14



0.79

0.44

0.43

0.85

0.43

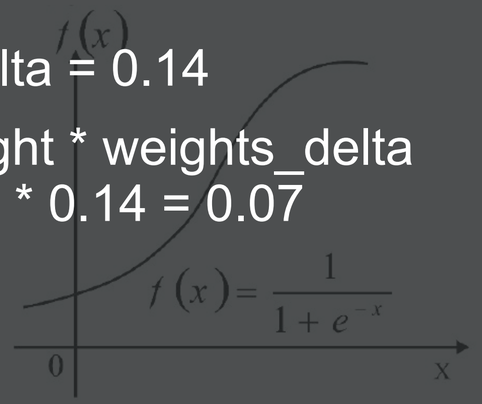
0.29

0.77

0.78

weights_delta = 0.14

error = weight * weights_delta
= 0.49 * 0.14 = 0.07



0.49

0.51

0.69*

**sigmoid(0.79)*

expected
:

THANK 

 @b0noi