**Supportive problem**

Suppose that we are going to work on AND Gate problem. The gate returns if and only if both inputs are true.

|  |  |  |
| --- | --- | --- |
| X1 | X2 | Y |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

We are going to set weights randomly. Let’s say that w1 = 0.9 and w2 = 0.9

### Round 1

We will apply 1st instance to the perceptron. x1 = 0 and x2 = 0.

Sum unit will be 0 as calculated below

Σ = x1 \* w1 + x2 \* w2 = 0 \* 0.9 + 0 \* 0.9 = 0

Activation unit checks sum unit is greater than a threshold. If this rule is satisfied, then it is fired and the unit will return 1, otherwise it will return 0. BTW, modern neural networks architectures do not use this kind of a [step function](https://sefiks.com/2017/05/15/step-function-as-a-neural-network-activation-function/) as activation.

Activation threshold would be 0.5.

Sum unit was 0 for the 1st instance. So, activation unit would return 0 because it is less than 0.5. Similarly, its output should be 0 as well. We will not update weights because there is no error in this case.

Let’s focus on the 2nd instance. x1 = 0 and x2 = 1.

Sum unit: Σ = x1 \* w1 + x2 \* w2 = 0 \* 0.9 + 1 \* 0.9 = 0.9

##### What about errors?

Activation unit will return 1 because sum unit is greater than 0.5. However, output of this instance should be 0. This instance is not predicted correctly. That’s why, we will update weights based on the error.

ε = actual – prediction = 0 – 1 = -1

We will add error times learning rate value to the weights. Learning rate would be 0.5. BTW, we mostly set learning rate value between 0 and 1.

w1 = w1 + α \* ε = 0.9 + 0.5 \* (-1) = 0.9 – 0.5 = 0.4

w2 = w2 + α \* ε = 0.9 + 0.5 \* (-1) = 0.9 – 0.5 = 0.4

Focus on the 3rd instance. x1 = 1 and x2 = 0.

Sum unit: Σ = x1 \* w1 + x2 \* w2 = 1 \* 0.4 + 0 \* 0.4 = 0.4

Activation unit will return 0 this time because output of the sum unit is 0.5 and it is less than 0.5. We will not update weights.

Mention the 4rd instance. x1 = 1 and x2 = 1.

Sum unit: Σ = x1 \* w1 + x2 \* w2 = 1 \* 0.4 + 1 \* 0.4 = 0.8

Activation unit will return 1 because output of the sum unit is 0.8 and it is greater than the threshold value 0.5. Its actual value should 1 as well. This means that 4th instance is predicted correctly. We will not update anything.

### Round 2

In previous round, we’ve used previous weight values for the 1st instance and it was classified correctly. Let’s apply feed forward for the new weight values.

Remember the 1st instance. x1 = 0 and x2 = 0.

Sum unit: Σ = x1 \* w1 + x2 \* w2 = 0 \* 0.4 + 0 \* 0.4 = 0.4

Activation unit will return 0 because sum unit is 0.4 and it is less than the threshold value 0.5. The output of the 1st instance should be 0 as well. This means that the instance is classified correctly. We will not update weights.

Feed forward for the 2nd instance. x1 = 0 and x2 = 1.

Activation unit will return 0 because sum unit is less than the threshold 0.5. Its output should be 0 as well. This means that it is classified correctly and we will not update weights.

We’ve applied feed forward calculation for 3rd and 4th instances already for the current weight values in the previous round. They were classified correctly.