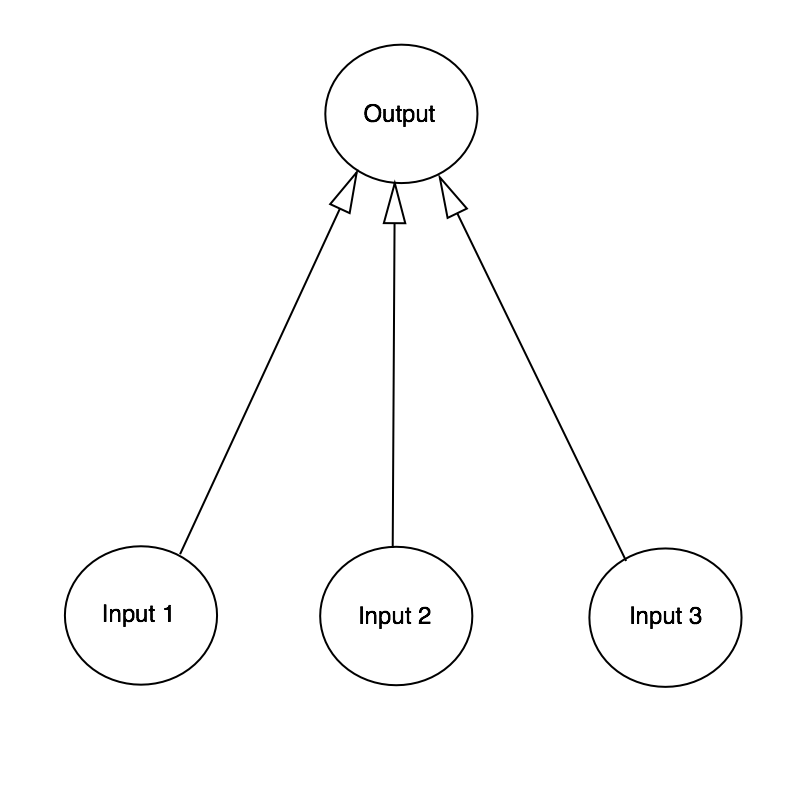
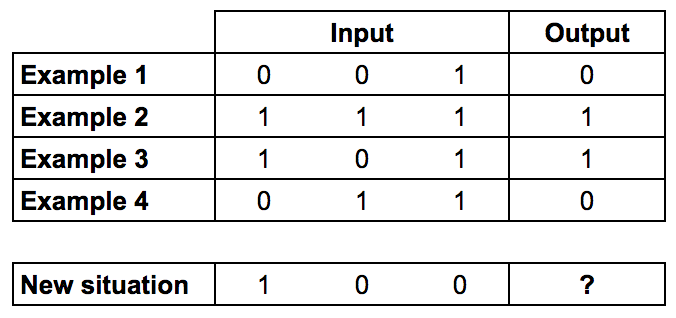
**How to build a simple neural network in 9 lines of Python code?**

* from numpy import exp, array, random, dot
* training\_set\_inputs = array([[0, 0, 1], [1, 1, 1], [1, 0, 1], [0, 1, 1]])
* training\_set\_outputs = array([[0, 1, 1, 0]]).T
* random.seed(1)
* synaptic\_weights = 2 \* random.random((3, 1)) – 1
* for iteration in xrange(10000):
* output = 1 / (1 + exp(-(dot(training\_set\_inputs, synaptic\_weights))))
* synaptic\_weights += dot(training\_set\_inputs.T, (training\_set\_outputs - output) \* output \* (1 - output))
* print 1 / (1 + exp(-(do t(array([1, 0, 0]), synaptic\_weights))))





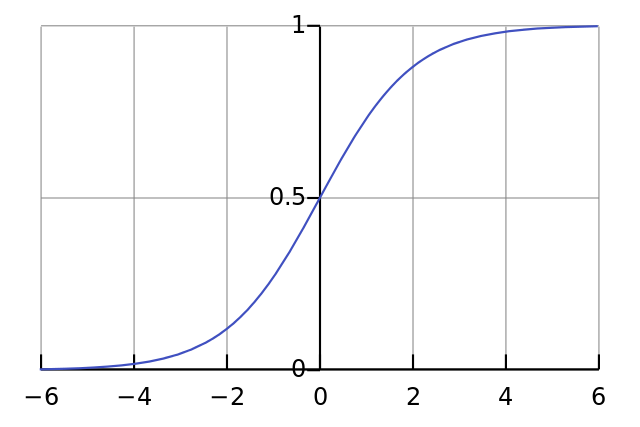
**Formula for calculating the neuron’s output**

https://cdn-images-1.medium.com/max/1600/1*RV7-CFkmmByfcXKkPcbAYQ.png

**Next we normalise this, so the result is between 0 and 1. For this, we use a mathematically convenient function, called the Sigmoid function:**

https://cdn-images-1.medium.com/max/1600/1*5il5GLo0gamypklQQ_z0AA.png

**If plotted on a graph, the Sigmoid function draws an S shaped curve.**



So by substituting the first equation into the second, the final formula for the output of the neuron is:

https://cdn-images-1.medium.com/max/1600/1*7YdyG6fc6f6zMmx3l0ZGsQ.png

**Formula for adjusting the weights**https://cdn-images-1.medium.com/max/1600/1*SQBjpbBcCT3lTQlPEdr1eg.png

The gradient of the Sigmoid curve, can be found by taking the derivative:

https://cdn-images-1.medium.com/max/1600/1*HdHm9u3_wjwBPmwuLg3D3g.png

So by substituting the second equation into the first equation, the final formula for adjusting the weights is:

https://cdn-images-1.medium.com/max/1600/1*Jow4WVWNOp6rtiJ7vNQ0gQ.png

You should get a result that looks like:

|  |
| --- |
| Random starting synaptic weights: |
|  | [[-0.16595599] |
|  | [ 0.44064899] |
|  | [-0.99977125]] |
|  |  |
|  | New synaptic weights after training: |
|  | [[ 9.67299303] |
|  | [-0.2078435 ] |
|  | [-4.62963669]] |
|  |  |
|  | Considering new situation [1, 0, 0] -> ?: |
|  | [ 0.99993704] |