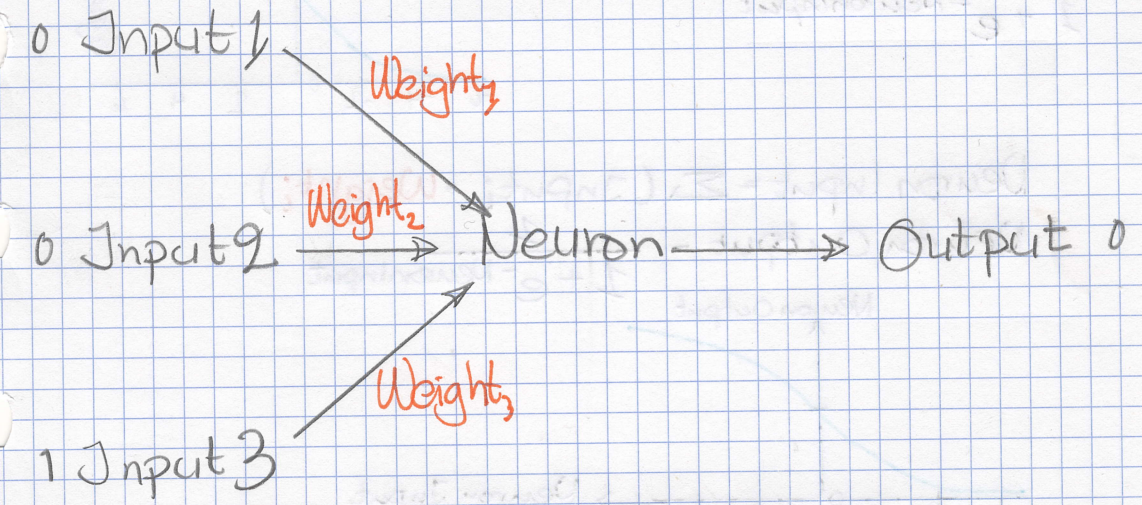
Build neural networks

Supervised learning = training data with correct answers labelled

The Challenge

	Input			Output
Example 1	0	0	1	0
Example 2	1	1	1	1
Example 3	1	0	1	1
Example 4	0	1	1	0
New Solution	1	0	0	?

Designing our architecture



Weights

Incoming signal to the neuron =

$Input_1 \cdot Weight_1 + Input_2 \cdot Weight_2 + Input_3 \cdot Weight_3$

$$\sum (Input_i \cdot Weight_i)$$

Activation Function

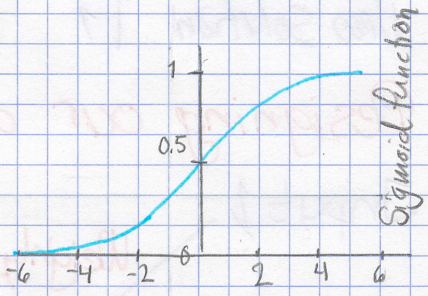
Neuron Output =
ActivationFunction (IncomingSignal)

We will use the Sigmoid function as our Activation Function.

$$\text{Sigmoid function} = \frac{1}{1+e^{-x}}$$

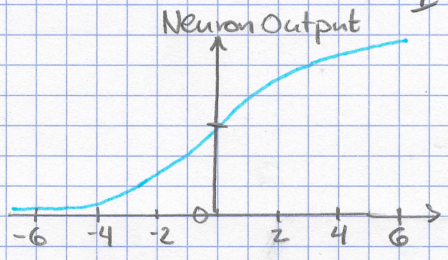
Euler's number
$e = 2.71828...$

$$\text{Neuron Output} = \frac{1}{1+e^{-\text{NeuronInput}}}$$



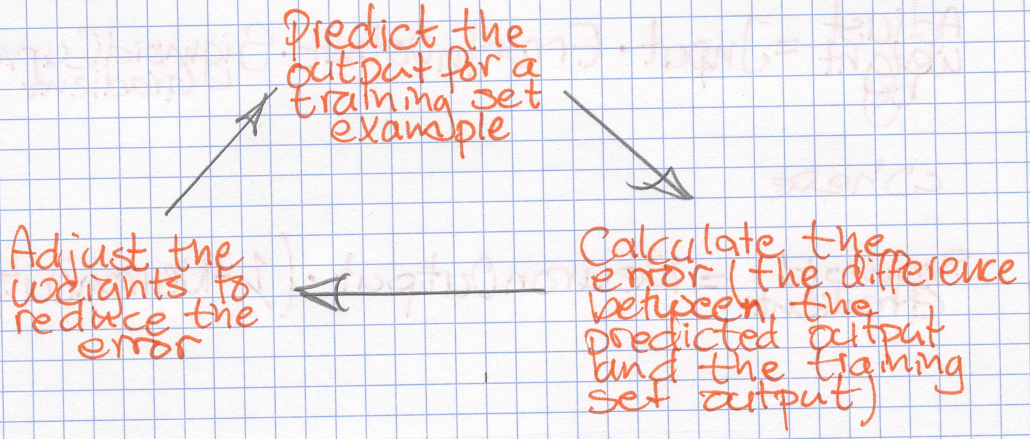
$$\text{Neuron Input} = \sum_i (\text{Input}_i \cdot \text{Weight}_i)$$

$$\text{Neuron Output} = \frac{1}{1+e^{-\text{NeuronInput}}}$$



Training process

First we assign random numbers to our weights.



Error Cost Function

How can we measure the total error of the neural network?

$$\text{Error Cost function} = \sum \frac{1}{2}(\text{CorrectOutput} - \text{PredictedOutput})^2$$

Adjusting the weights

Gradient descent

Gradient = slope

Gradient descent finds the minimum by taking steps proportional to the negative gradient.

What is the gradient of the error cost function?

$$\text{Error Cost function Gradient} = -\text{Input} \cdot \text{ErrorInOutput} \cdot \text{SigmoidCurve}_{\text{Gradient}}$$

$$\text{Adjust weight by} = \text{Input} \cdot \text{ErrorInOutput} \cdot \text{SigmoidCurve}_{\text{Gradient}}$$

$$\text{Sigmoid Gradient} = \text{NeuronOutput} \cdot (1 - \text{NeuronOutput})$$

$$\text{Adjust weight by} = \text{Input} \cdot \text{Error} \cdot \text{InOutput} \cdot \text{SigmoidCurve Gradient}$$

where

$$\text{Sigmoid Gradient} = \text{NeuronOutput} \cdot (1 - \text{NeuronOutput})$$