

Intro to Deep Learning

- 1) The neuron model
- 2) Gradient descent
- 3) Architectures (mostly for vision)
- 4) Model evaluation

Intro to Deep Learning

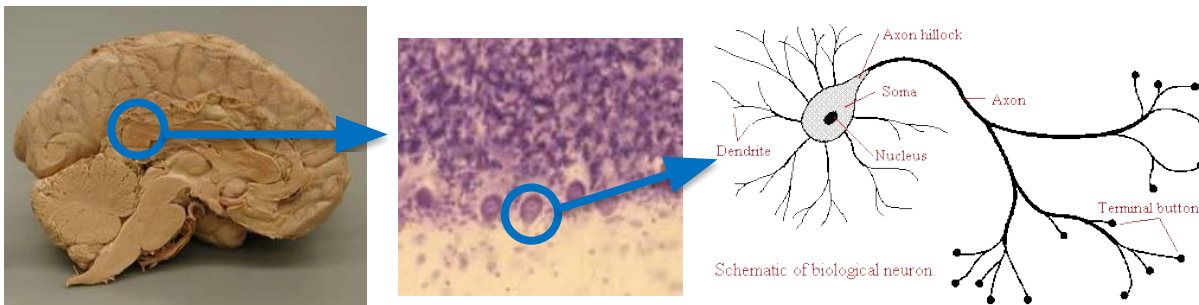
- 1) The neuron model**
- 2) Gradient descent
- 3) Architectures (mostly for vision)
- 4) Model evaluation

The starting point: the neuron model.

Originally inspired by the brain

Neural nets are interconnected networks of simple processing units, a.k.a neurons

It remains just a parallel, the artificial neuron is just an approximation!

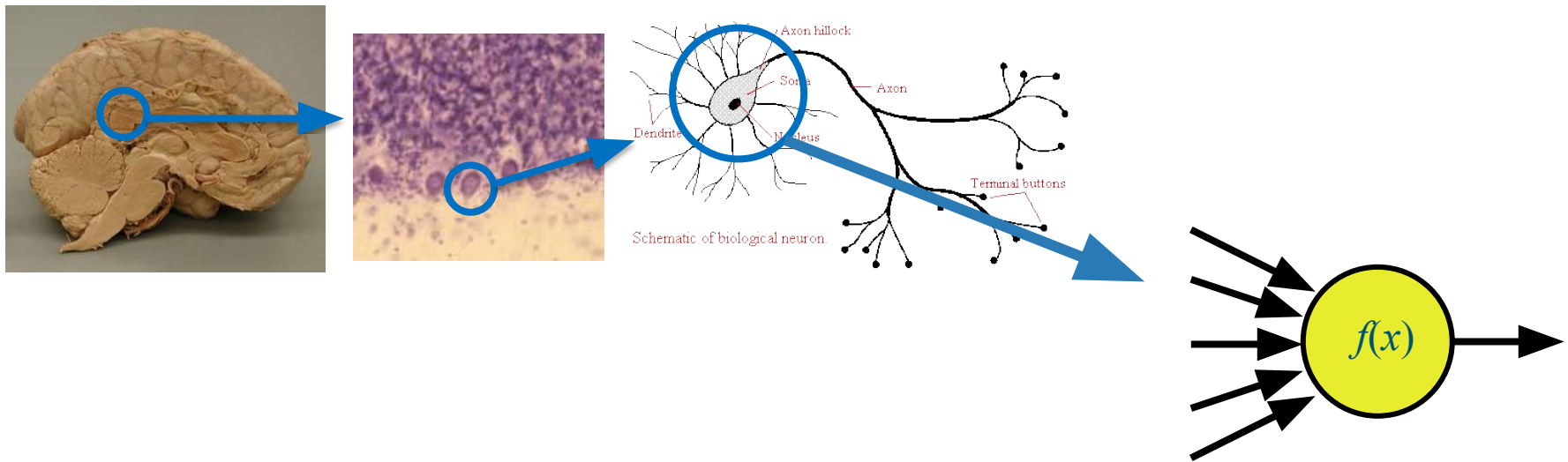


The starting point: the neuron model.

Originally inspired by the brain

Neural nets are interconnected networks of simple processing units, a.k.a neurons

It remains just a parallel, the artificial neuron is just an approximation!

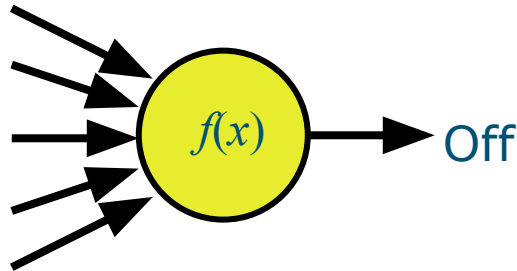


The starting point: the neuron model.

Originally inspired by the brain

Neural nets are interconnected networks of simple processing units, a.k.a neurons

It remains just a parallel, the artificial neuron is just an approximation!

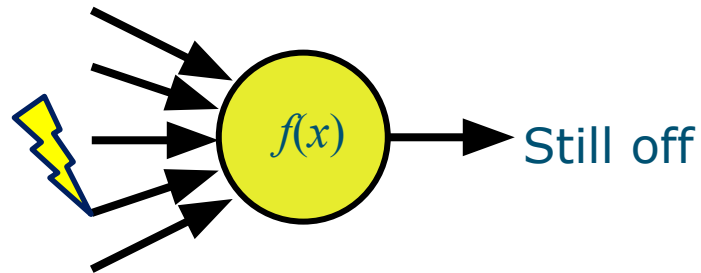


The starting point: the neuron model.

Originally inspired by the brain

Neural nets are interconnected networks of simple processing units, a.k.a neurons

It remains just a parallel, the artificial neuron is just an approximation!

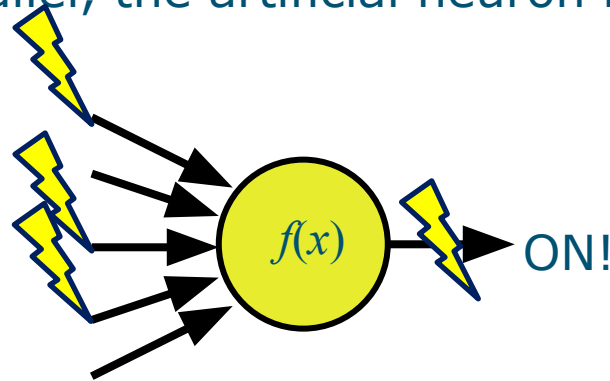


The starting point: the neuron model.

Originally inspired by the brain

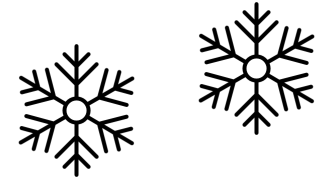
Neural nets are interconnected networks of simple processing units, a.k.a neurons

It remains just a parallel, the artificial neuron is just an approximation!



History of great moments

- 1943 : McCulloch and Pitts: model of neuron
- 1958 : Rosenblatt: **perceptrons**
- 1968 : Minsky and Papert point out limitations: perceptrons are linear
- 1982 : Hopfield network (associative memory), Kohonen's self-organising map (clustering), Fukushima's Neocognitron (vision)
- 1986 : Rumelhart, Hinton and Williams: training of **multilayer perceptrons**
- 1989 : LeCun introduces convolutional networks
- 2012 : Hinton (re)introduces these: **deep learning**
- 2022 : OpenAI launches ChatGPT

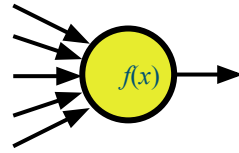


1st AI Winter
(models too simple)



2nd AI Winter
(not enough data nor compute)



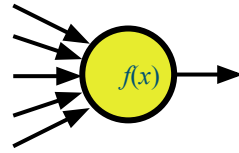


The starting point: the neuron model.

The neuron was first proposed by Warren McCulloch and Walter Pitts in 1943. It was modelled with electrical circuits.

The output y was modelled as a **weighted linear combination** of inputs x_j (*j are the variables describing each one of your samples*) plus a bias (β_0)

$$y = \sum_j \beta_j x_j + \beta_0$$



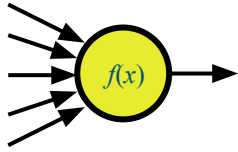
The starting point: the neuron model.

The neuron was first proposed by Warren McCulloch and Walter Pitts in 1943. It was modelled with electrical circuits.

The output y was modelled as a **weighted linear combination** of inputs x plus a bias (β_0)

Moreover, a transfer function, or **activation function** was used to make a decision :

$$y = f \left(\sum_j \beta_j x_j + \beta_0 \right)$$



The starting point: the neuron model.

$$y = f\left(\sum_j \beta_j x_j + \beta_0\right)$$

Let's call v the result of the parenthesis. Examples of activation functions:

$$f(v) = v$$



Linear: Outputs the linear combination itself

$$f(v) = \frac{1}{1 + \exp(-v)}$$



Sigmoid: Outputs numbers between 0 and 1

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

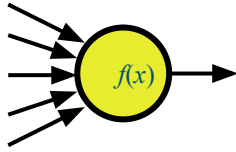


Step: Classifies into two groups

$$f(v) = \begin{cases} v & : v \geq 0 \\ 0 & : v < 0 \end{cases}$$

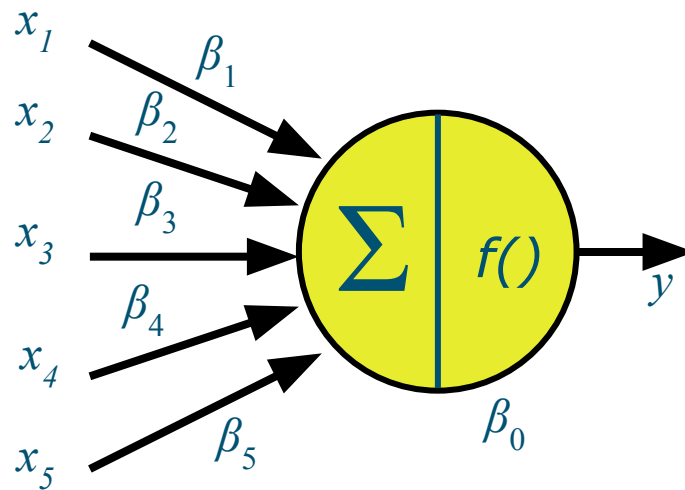


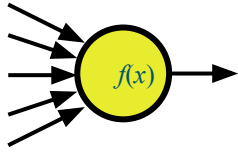
ReLU: Outputs positive numbers



The starting point: the neuron model.

$$y = f\left(\sum_j \beta_j x_j + \beta_0\right)$$





The starting point: the neuron model.

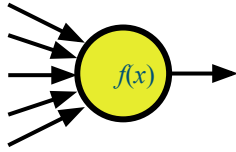
$$y = f(\boldsymbol{\beta}^\top \mathbf{x} + \beta_0)$$

β_1	β_2	β_3	β_4	β_5
-----------	-----------	-----------	-----------	-----------

x_1
x_2
x_3
x_4
x_5

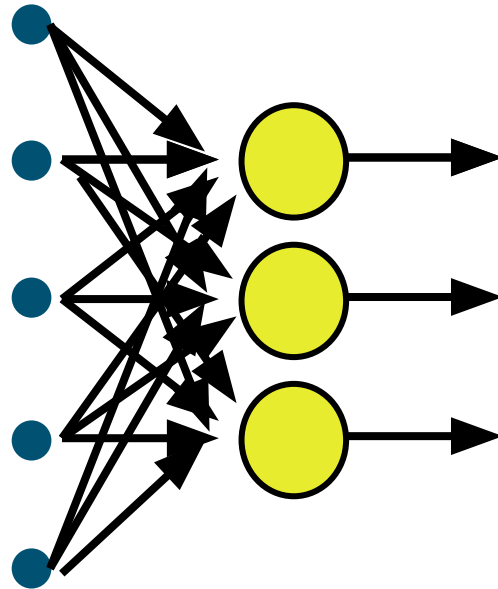
$$+ \beta_0 = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_0 =$$

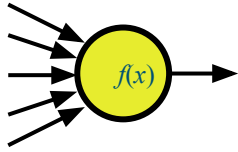
y



The starting point: the neuron model.

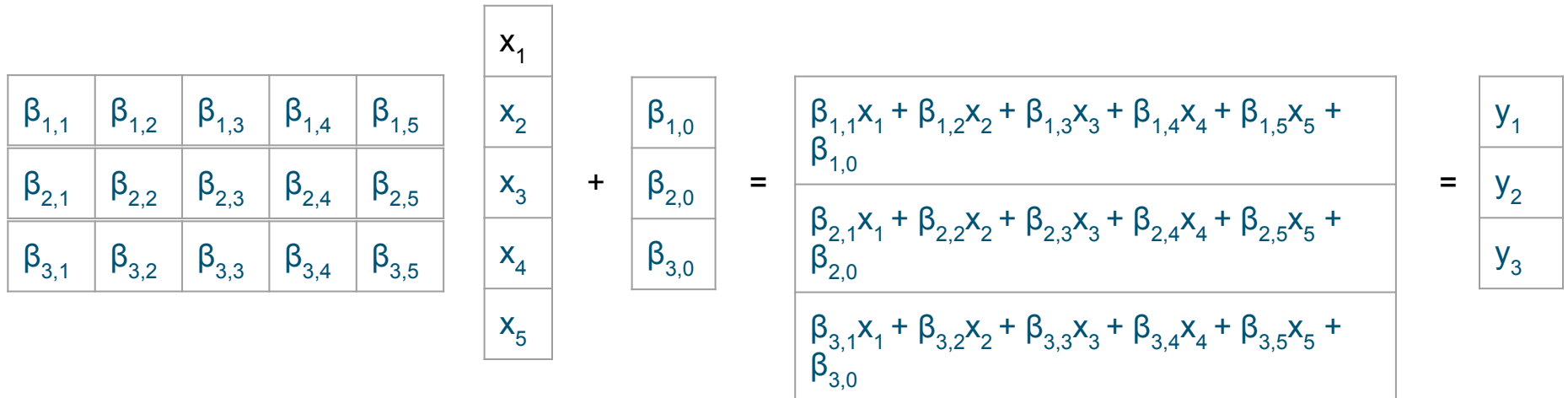
$$y = f(\boldsymbol{\beta}^T \mathbf{x} + \beta_0)$$

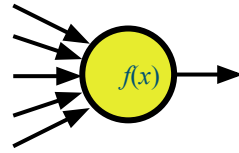




The starting point: the neuron model.

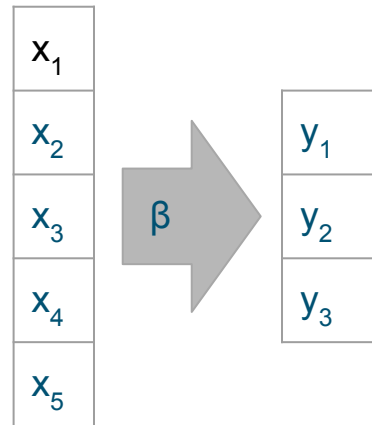
$$y = f(\boldsymbol{\beta}^T \mathbf{x} + \beta_0)$$

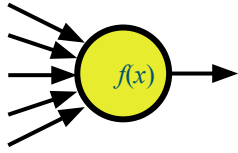




The starting point: the neuron model.

$$y = f(\boldsymbol{\beta}^T \mathbf{x} + \beta_0)$$





The starting point: the neuron model.

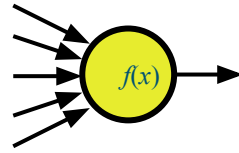
With neural networks, it all becomes matrix multiplications.

You know what else requires many matrix multiplications?

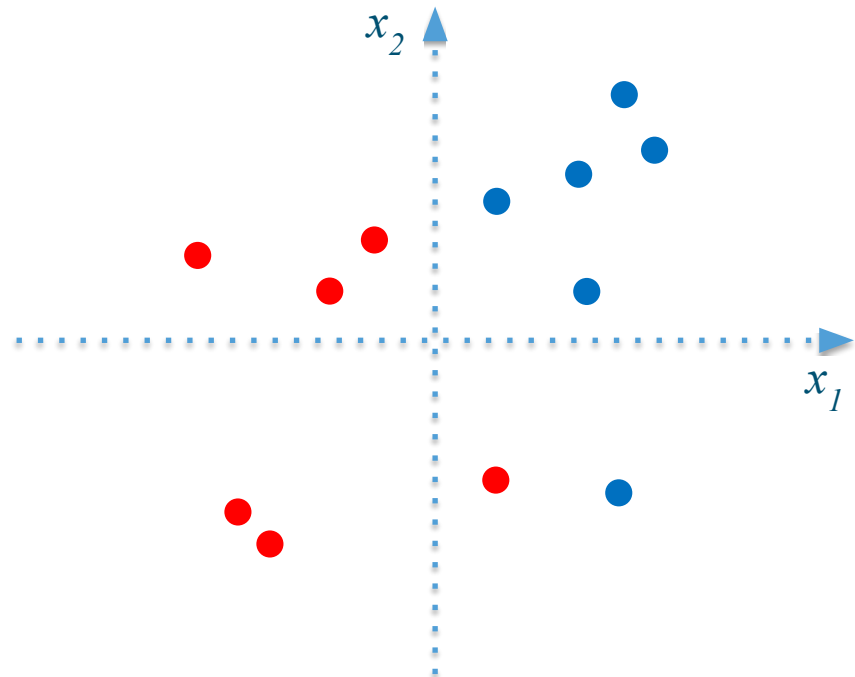
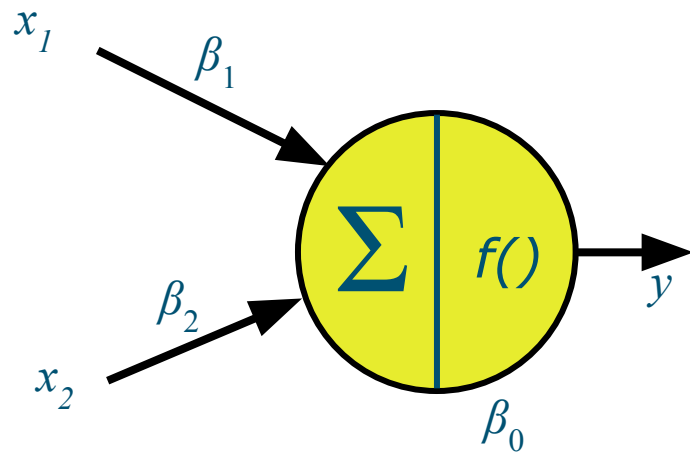
Video games!



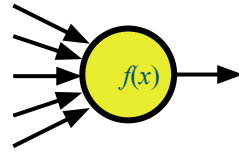
Learning the weights of a single neuron



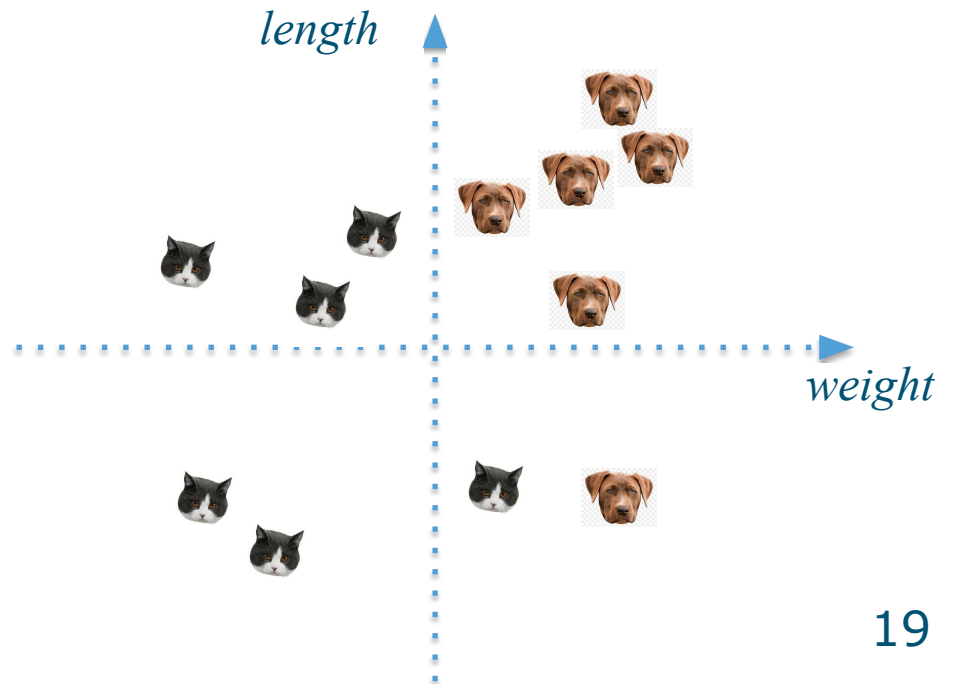
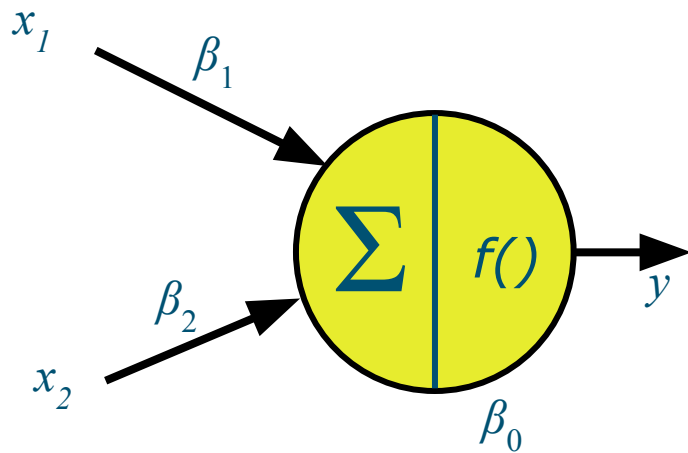
$$y = f(\beta_1 x_1 + \beta_2 x_2 + \beta_0)$$



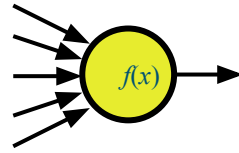
Learning the weights of a single neuron



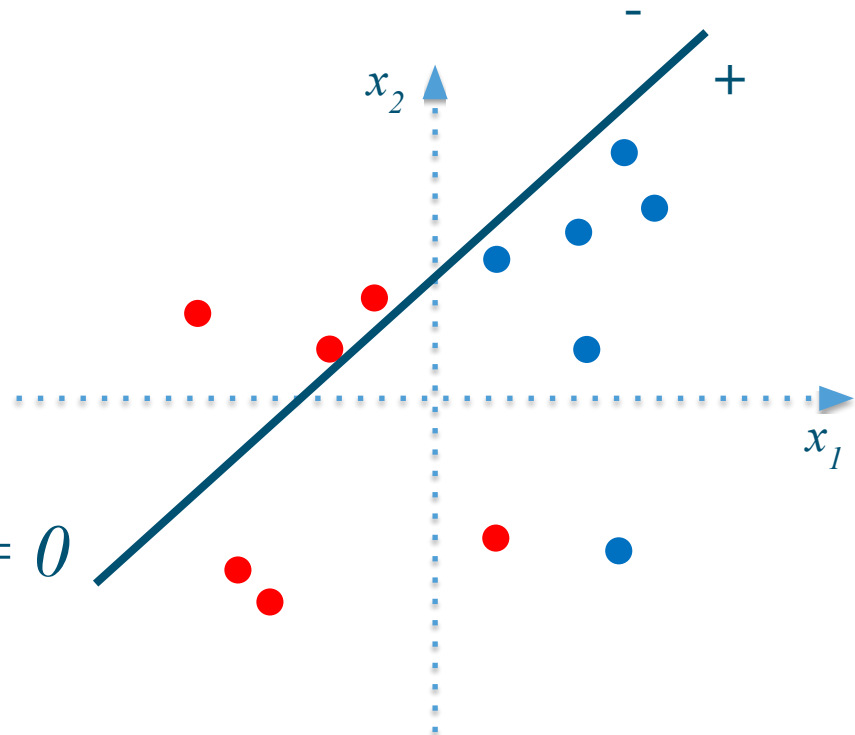
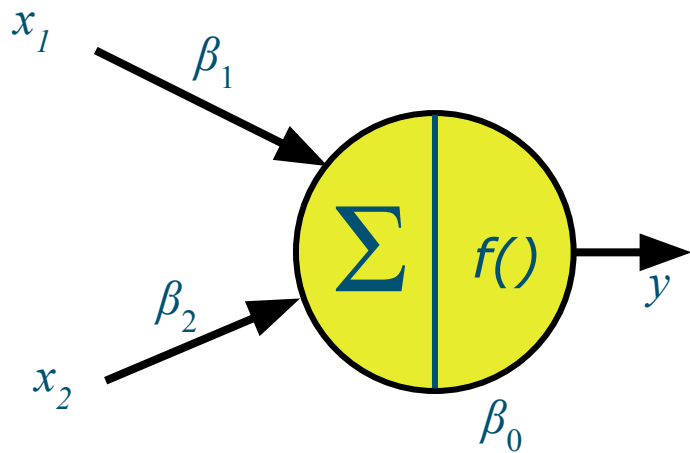
$$y = f(\beta_1 x_1 + \beta_2 x_2 + \beta_0)$$



Learning the weights of a single neuron



$$y = f(\beta_1 x_1 + \beta_2 x_2 + \beta_0)$$



$$\beta_1 x_1 + \beta_2 x_2 + \beta_0 = 0$$