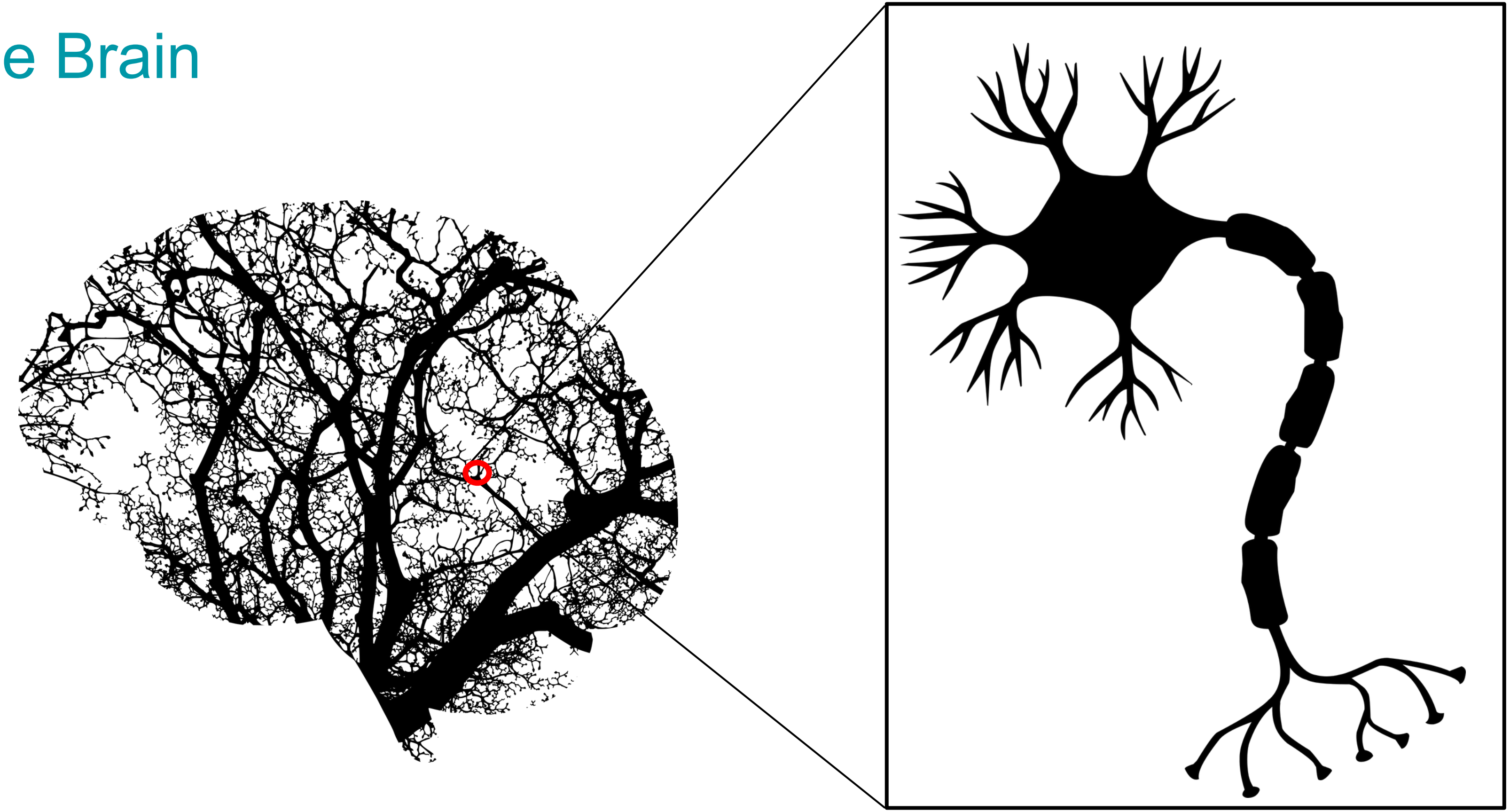
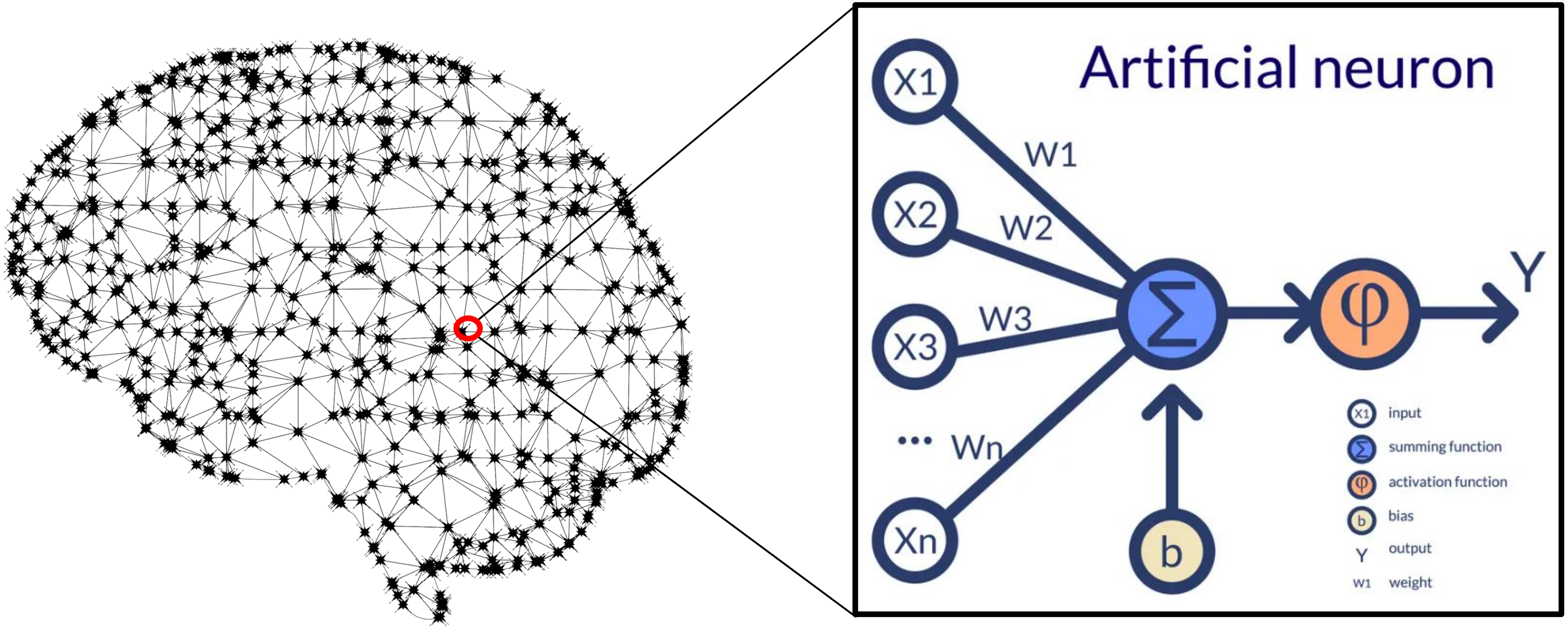
Machine Learning

Lecture 3

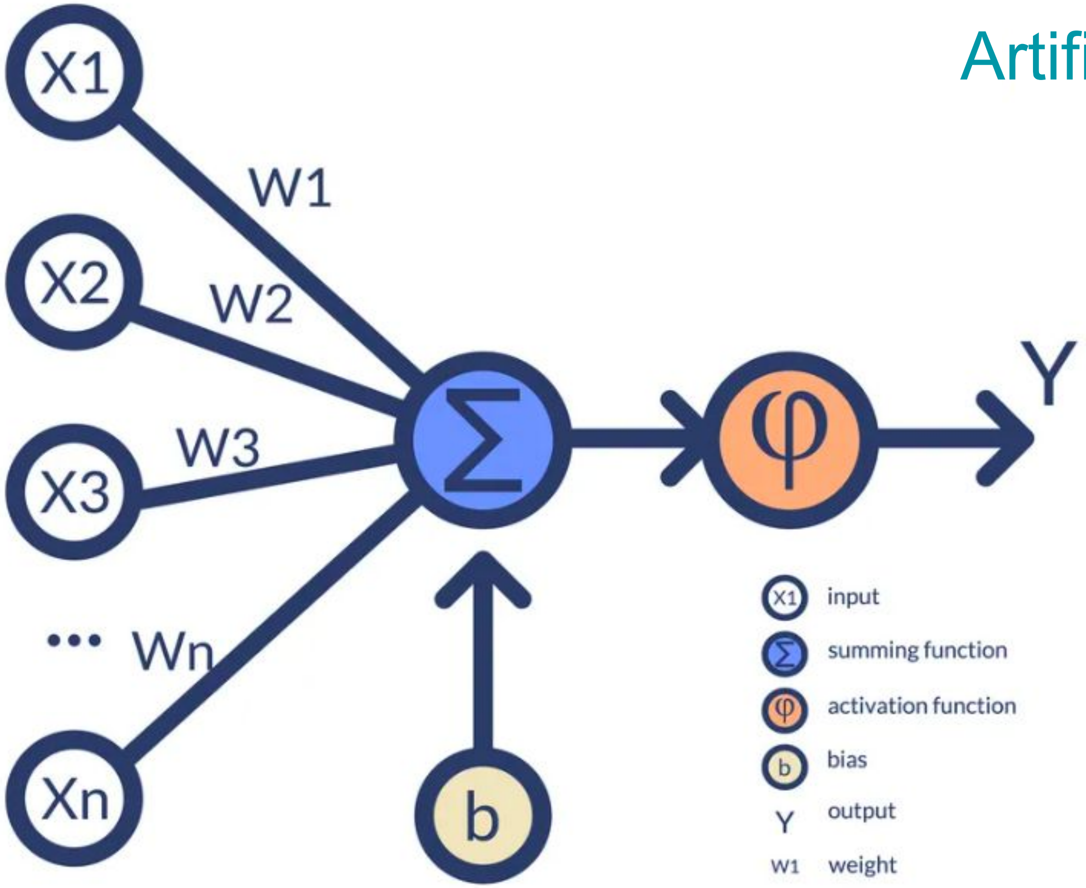
The Brain



Artificial Neural Network

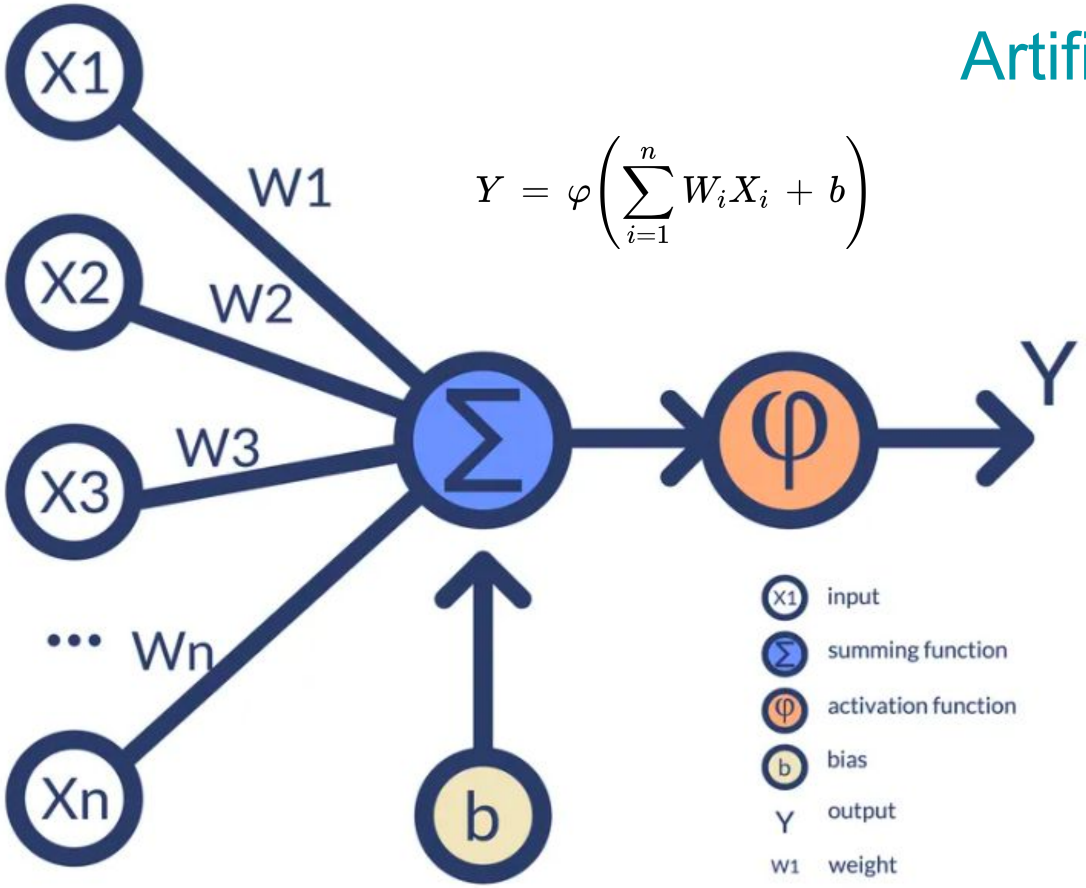


Artificial Neuron



Artificial Neuron

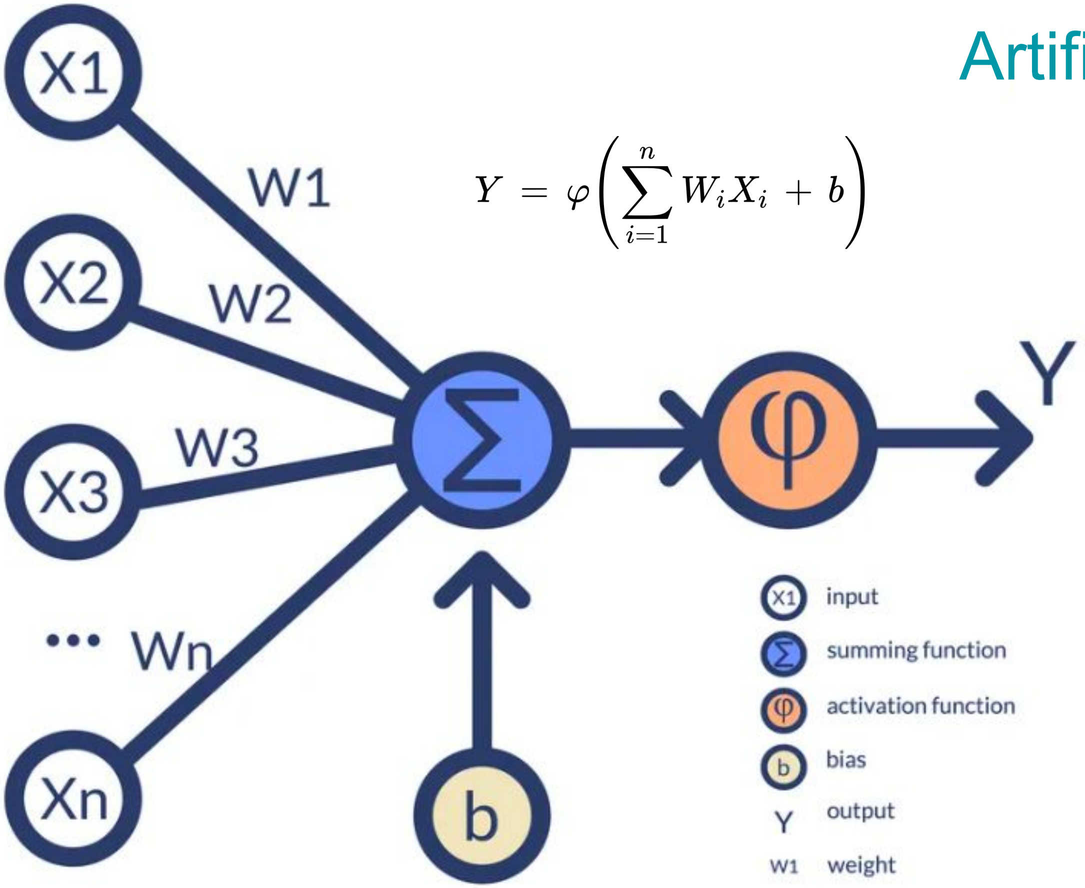
$$Y = \varphi\left(\sum_{i=1}^{n} W_i X_i + b\right)$$



Artificial Neuron

Perceptron

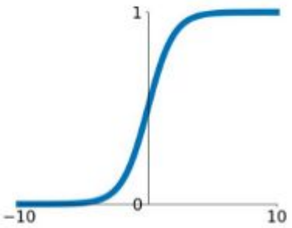
$$Y = \varphi\left(\sum_{i=1}^{n} W_i X_i + b\right)$$



Activation Functions (φ)

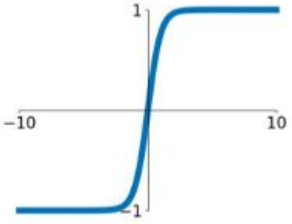
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



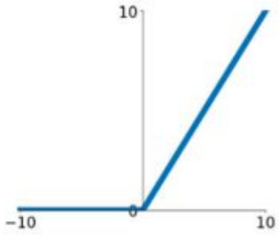
tanh

$\tanh(x)$



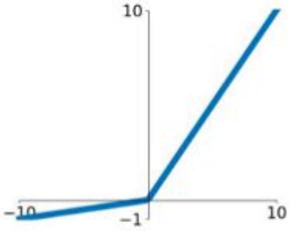
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

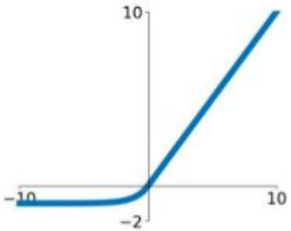


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

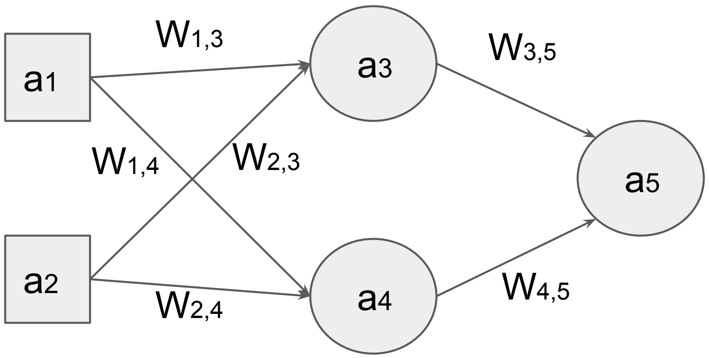
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Feed Forward Neural Network

Single Layer Perceptrons

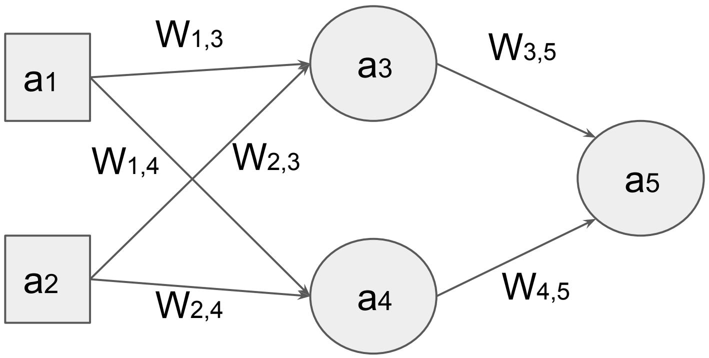


$$
\begin{aligned}
a_5 &= \varphi(W_{3,5}.\,a_3 + W_{4,5}.\,a_4) \\
&= \varphi(W_{3,5}\,.\,\varphi(W_{1,3}.\,a_1 + W_{2,3}.\,a_2) + W_{4,5}.\,\varphi(W_{1,4}.\,a_1 + W_{2,4}.\,a_2))
\end{aligned}
$$

- ➔ Feed-forward network = a parameterized family of nonlinear functions
- ➔ Adjusting weights changes the function: do learning this way!

Feed Forward Neural Network

Single Layer Perceptrons



$$a_5 = \varphi(W_{3,5}.\, a_3 + W_{4,5}.\, a_4) = \varphi(Net_5)$$
$$= \varphi(W_{3,5}\,.\, \varphi(W_{1,3}.\, a_1 + W_{2,3}.\, a_2) + W_{4,5}.\, \varphi(W_{1,4}.\, a_1 + W_{2,4}.\, a_2))$$

- ➔ Feed-forward network = a parameterized family of nonlinear functions
- ➔ Adjusting weights changes the function: HOW ?

Backpropagation

Multilayer Perceptron with a single hidden layer:

$$L(x, y, \theta) = \frac{1}{2}(y - a_5)^2$$

we want to know how much change in $W_{3,5}$ affects the total Loss, AKA the gradient of $W_{3,5}$, i.e $\rightarrow$ $\frac{\mathrm{d}L}{\mathrm{d}W_{3,5}}$

Using chain rule : $$\frac{\mathrm{d}L}{\mathrm{d}W_{3,5}} = \frac{\mathrm{d}L}{\mathrm{d}a_5} \cdot \frac{\mathrm{d}a_5}{\mathrm{d}Net_5} \cdot \frac{\mathrm{d}Net_5}{\mathrm{d}W_{3,5}}$$

Now,

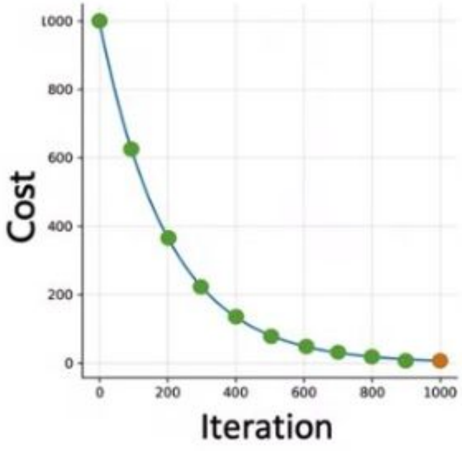
$$\Rightarrow \frac{\mathrm{d}L}{\mathrm{d}a_5} = \frac{\mathrm{d}\left(\frac{1}{2}(y - a_5)^2\right)}{\mathrm{d}a_5} = -(y - a_5)$$

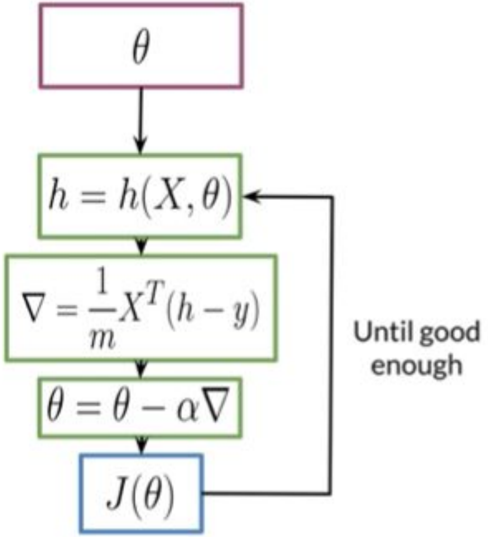
$$\Rightarrow \frac{\mathrm{d}a_5}{\mathrm{d}Net_5} = \frac{\mathrm{d}\left(1 + e^{-Net_5}\right)^{-1}}{\mathrm{d}Net_5} = \frac{e^{-Net_5}}{\left(1 + e^{-Net_5}\right)^2} = a_5(1 - a_5)$$

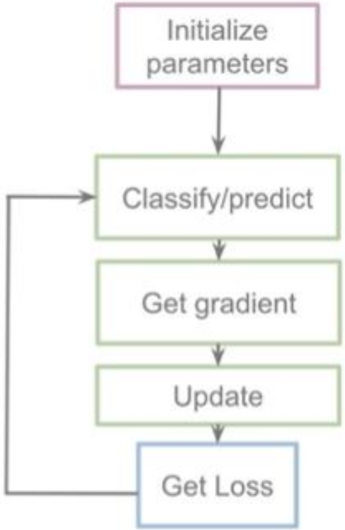
$$\Rightarrow \frac{\mathrm{d}Net_5}{\mathrm{d}W_{3,5}} = \frac{\mathrm{d}(W_{3,5}a_3 + W_{4,5}a_4)}{\mathrm{d}W_{3,5}} = a_3$$

Backpropagation

$$\Delta W_{3,5} = -(y - a_5)\,.\,a_5(1 - a_5)\,.\,a_3$$







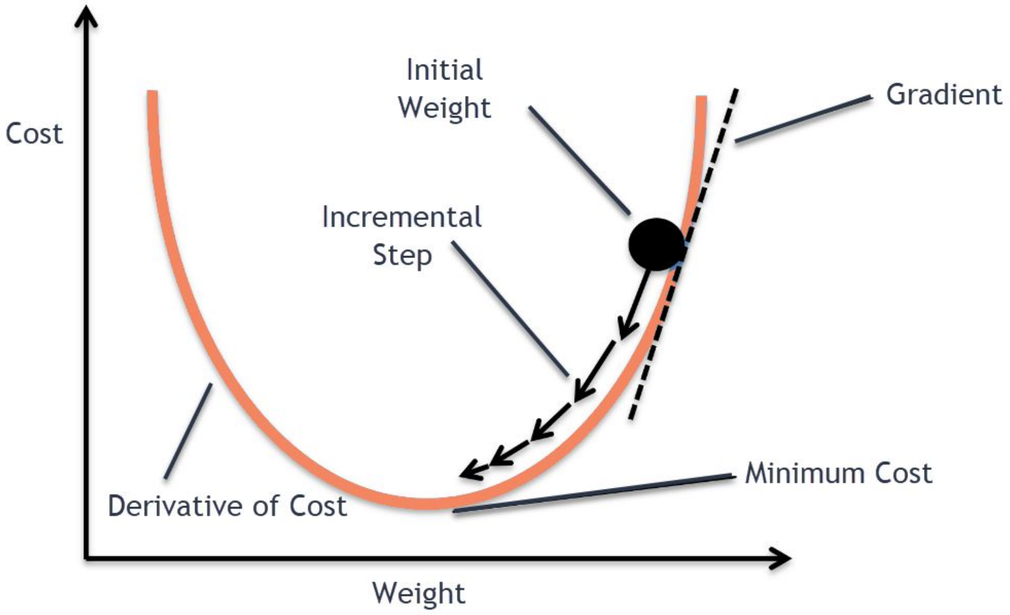
Gradient Descent

$$\min_{\theta} J(\theta)$$

$$\theta \leftarrow \theta - \alpha \frac{\mathrm{d}}{\mathrm{d}\theta} J(\theta)$$

$$\frac{\mathrm{d}}{\mathrm{d}\theta} J(\theta) = (h(x, \theta) - y)\, x$$

$$\theta \leftarrow \theta - \alpha(h(x, \theta) - y)x$$



α → learning rate

Multi-layer Neural Network

Perceptrons