Multilayer Perceptron

Legend :

$a_0^{l_0}$ Input n°0 of layer n°0

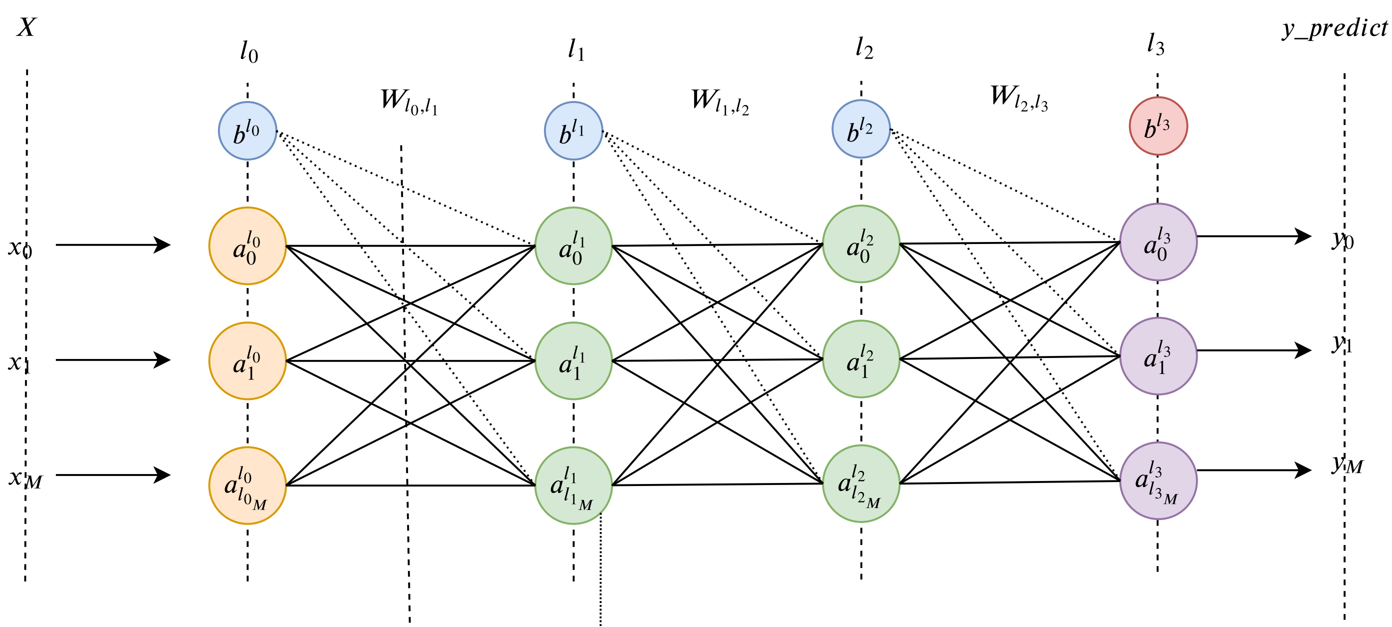
b^{l_0} Bias of layer n°0

W_{l_0,l_1} Matrix of weights beetwen layer n°0 and layer n°1

x_0 First feature of the input vector

y_0 Probability that the input vector is of class n°0

M is used to mean length



Matrice de taille

$l_{0_M} \times l_{1_M}$

$$W_{l0_M,l1_M} = \begin{pmatrix} w_{a_0^{l_0},a_0^{l_1}} & w_{a_0^{l_0},a_1^{l_1}} & \cdots & w_{a_0^{l_0},a_{l1_M}^{l_1}} \\ w_{a_1^{l_0},a_0^{l_1}} & w_{a_1^{l_0},a_1^{l_1}} & \cdots & w_{a_1^{l_0},a_{l1_M}^{l_1}} \\ \vdots & \vdots & \ddots & \vdots \\ w_{a_{l0_M}^{l_0},a_0^{l_1}} & w_{a_{l0_M}^{l_0},a_1^{l_1}} & \cdots & w_{a_{l0_M}^{l_0},a_{l1_M}^{l_1}} \end{pmatrix}$$

Neuron n°i value:

$$a_i^{l_1} = \phi(b_{l_0} + \sum_{j=0}^{M} w_j^{l_0,l_1} a_j^{l_0})$$

Description of backward propagation learning

$$y_{l_0,l_1} = \phi(X \cdot W_{l_0,l_1} + bias_{l_0,l_1})$$

$$y_{l_1,l_2} = \phi(y_{l_0,l_1} \cdot W_{l_1,l_2} + bias_{l_1,l_2})$$

$$y\_predict = \phi(y_{l_1,l_2} \cdot W_{l_2,l_3} + bias_{l_2,l_3})$$

$$X \rightarrow W_{l_0,l_1} \rightarrow \phi(r) \rightarrow W_{l_1,l_2} \rightarrow \phi(r) \rightarrow W_{l_2,l_3} \rightarrow \phi(r) \rightarrow y\_predict$$

$$\delta_{3,4} = \frac{\partial C(y, y\_predict)}{\partial y}$$

$$\epsilon_{2,3} = \delta_{3,4} \cdot W_{l_2,l_3}$$

$$\delta_{2,3} = \epsilon_{2,3} \cdot \frac{\partial \phi}{\partial y}(y_{l_1,l_2})$$

$$\epsilon_{1,2} = \delta_{2,3} \cdot W_{l_1,l_2}$$

$$\delta_{1,2} = \epsilon_{1,2} \cdot \frac{\partial \phi}{\partial y}(y_{l_0,l_1})$$

$$\epsilon_{0,1} = \delta_{1,2} \cdot W_{l_0,l_1}$$

$$\delta_{0,1} = \epsilon_{0,1} \cdot \frac{\partial \phi}{\partial y}(X)$$

$$W_{l_0,l_1} += \delta_{0,1} \cdot X \cdot \mu$$

$$b_{l_0,l_1} += \delta_{0,1} \cdot \mu$$

$$W_{l_1,l_2} += \delta_{1,2} \cdot y_{l_0,l_1} \cdot \mu$$

$$b_{l_1,l_2} += \delta_{1,2} \cdot \mu$$

$$W_{l_2,l_3} += \delta_{2,3} \cdot y_{l_1,l_2} \cdot \mu$$

$$b_{l_2,l_3} += \delta_{2,3} \cdot \mu$$

FORWARD PROPAGATION

BACKWARD PROPAGATION

$y\_predict$	Output vector
X	Input vector
$\phi(\vec{r})$	Activation function
μ	Learning rate
$C(y, y_p redict)$	Error function

$$\phi(r) = \begin{pmatrix} \phi(r_0) \\ \phi(r_1) \\ \cdots \\ \phi(r_M) \end{pmatrix}$$

MLP project architecture