Multilayer Perceptron

Legend :

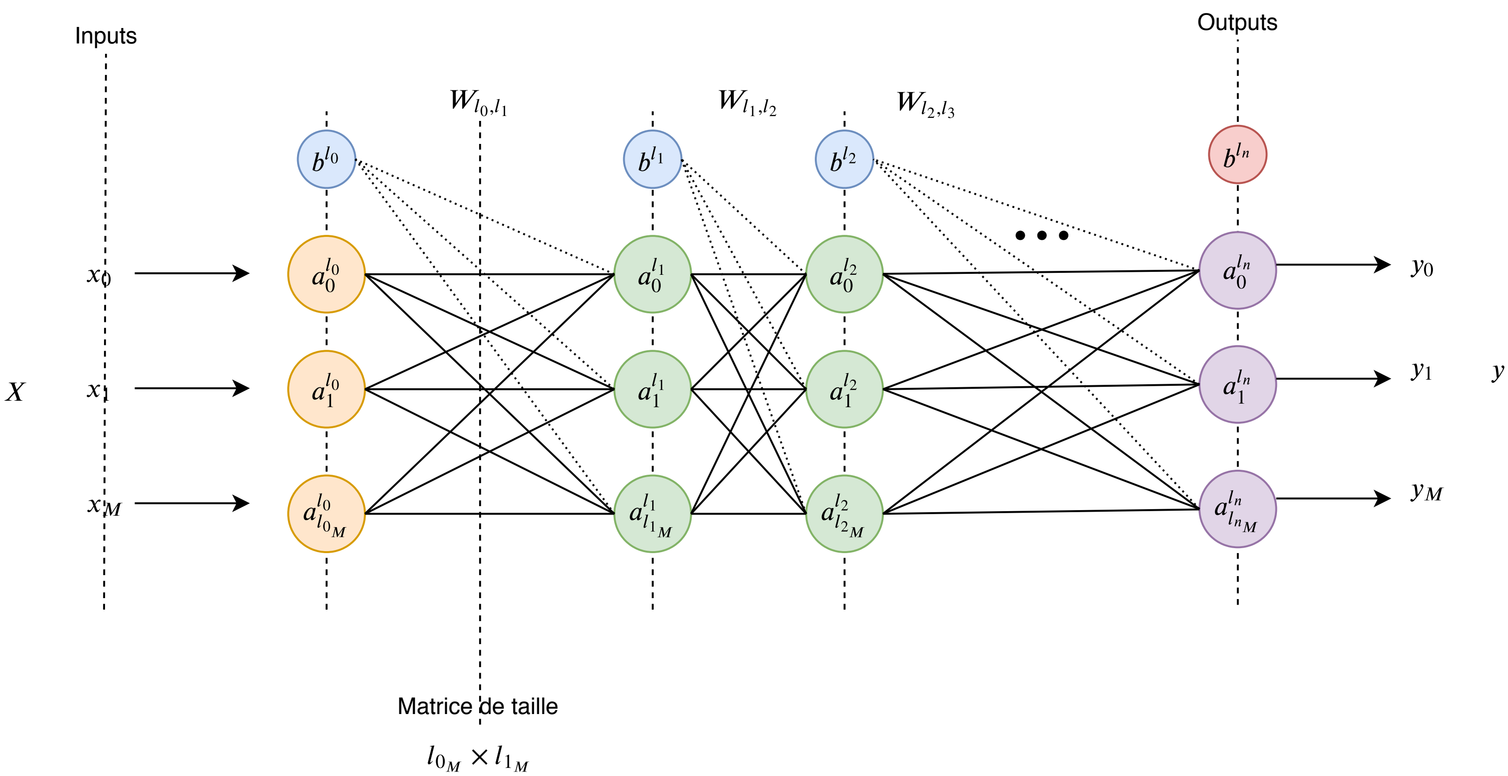
$a_0^{l_0}$ Input n°0 of layer n°0

b^{l_0} Bias layer n°0

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

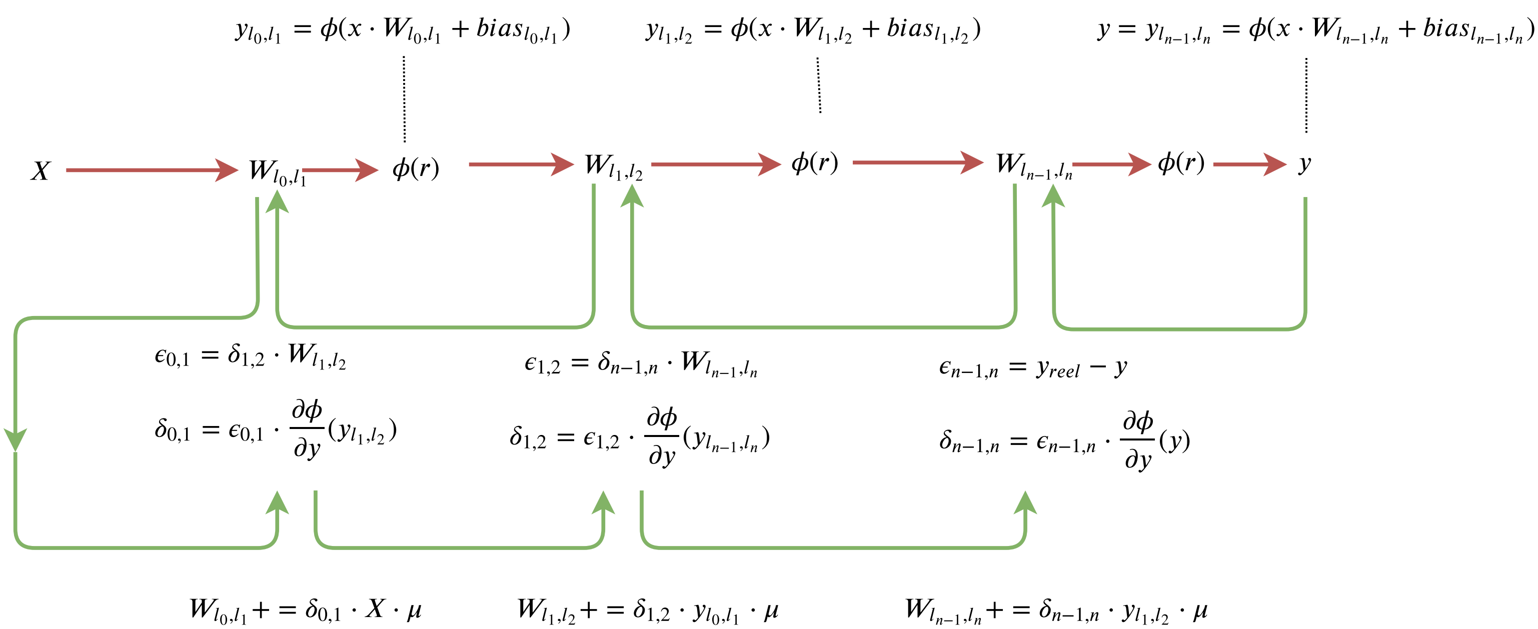
x_0 First feature of the input vector

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



$$W_{l0,l1} = \begin{pmatrix} w_{a_0^{l_0},a_0^{l_1}} & w_{a_0^{l_0},a_1^{l_1}} & \cdots & w_{a_0^{l_0},a_{l1_M}^{l_1}} \\ w_{a_1^{l_0},a_0^{l_1}} & w_{a_1^{l_0},a_1^{l_1}} & \cdots & w_{a_1^{l_0},a_{l1_M}^{l_1}} \\ \vdots & \vdots & \ddots & \vdots \\ w_{a_{l0_M}^{l_0},a_0^{l_1}} & w_{a_{l0_M}^{l_0},a_1^{l_1}} & \cdots & w_{a_{l0_M}^{l_0},a_{l1_M}^{l_1}} \end{pmatrix}$$

Description of backward propagation learning



FORWARD PROPAGATION

BACKWARD PROPAGATION

y Output vectors containing same number of elements as classes

X Input vector

$\phi(\vec{r})$ Activation function

μ Learning rate

Layer's output :

$$\phi(\sum_{j=0}^{M} w_j x_{i,j})$$

Description of forward propagation prediction

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

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

$$y = y_{l_{n-1},l_n} = \phi(x \cdot W_{l_{n-1},l_n} + bias_{l_{n-1},l_n})$$

$$X_{test} \rightarrow W_{l_0,l_1} \rightarrow \phi(r) \rightarrow W_{l_1,l_2} \rightarrow \phi(r) \rightarrow W_{l_{n-1},l_n} \rightarrow \phi(r) \rightarrow y_{prediction}$$