

# Backpropagation algorithm: Summary

MALIS

November 11, 2019

Cheat sheet to the backpropagation algorithm

## 1 Feed-forward step

For an input vector  $\mathbf{x}_n$  do a forward step to compute the activations and outputs for all layers in the network:

$$a_j^{(l)} = \sum_i w_{ji}^{(l)} z_i^{(l-1)} + b_j^{(l)} \quad (1)$$

$$z_j^{(l)} = h(a_j^{(l)}) \quad (2)$$

Please refer to Table 1 for the definition of each symbol.

## 2 Backward step

1. Calculate the error functions  $\delta$  starting from the output units:

$$\delta_k^{(L)} = 2(z_k^{(L)} - y_k) \cdot h'(a_k^{(L)}) \quad (3)$$

2. Calculate the remaining error functions by working backwards using the back-propagation algorithm

$$\delta_j^{(l)} = h'(a_j^{(l)}) \sum_k w_{kj}^{(l+1)} \delta_k^{(l+1)} \quad (4)$$

3. Estimate the required derivatives  $\nabla_{\mathbf{w}} L$

$$\frac{\partial L}{\partial w_{kj}^{(l)}} = \delta_k^{(l)} z_j^{l-1} \quad (5)$$

Note that the bias term for a layer  $l$ , the input is  $z = 1$  so,  $\frac{\partial L}{\partial b_k^{(l)}} = \delta_k^{(l)}$ .

Symbol	Description
$w_{kj}^{(l)}$	The weight associated to connection from node $j$ in layer $l - 1$ to node $k$ in layer $l$ .
$b_j^{(l)}$	The bias of node $j$ in layer $l$ . Equivalently can be denoted $w_{j0}^l$ .
$a_j^{(l)}$	Weighted input emitted from node $j$ in layer $l$ . Denoted the activation.
$z_j^{(l)}$	Output of node $j$ in layer $l$ after applying a non-linear activation function $h(\cdot)$ to $a_j^{(l)}$ .
$E$	The loss/cost function.
$\delta_j^{(l)}$	Error function of node $j$ in layer $l$ . $\delta_j^{(l)} = \frac{\partial E}{\partial a_j^{(l)}}$
$h(\cdot)$	Non-linear activation function.
$L$	Number of layers
$y_k$	Target of the $k^{th}$ network output.

Table 1: List of symbols in the back propagation algorithm

4. Change the weights based on estimated gradients by  $-\alpha \cdot \nabla_{\mathbf{w}} L$ , where  $\alpha$  is the learning rate.
5. Go back to forward step and repeat until a number of iterations or a desired minimum.