BACKPROPAGATION ALGORITHM FOR NEURAL NETWORK COMPUTATION

Eino Uikkanen, www.einouikkanen.fi/en/, eino.uikkanen@iki.fi, v. 28.3.2026

This algorithm applies to a feedforward neural network of arbitrary size, where the activation function is Sigmoid ($\boldsymbol{\sigma}$) and the cost function (below $\boldsymbol{C}$) is Mean Squared Error (MSE). If the neural network differs from these specifications, the algorithm must be modified accordingly. A comprehensive description of backpropagation can be found in the article Neural network calculation algorithms.

Notations and Concepts Used in the Algorithm

- $\boldsymbol{z_n^l}$: The input value of neuron *n* in layer *l*.
- $\boldsymbol{a_n^l}$: The output value of neuron *n* in layer *l*, i.e., the activation of the neuron. The network input values are also treated as activations in the algorithm, even though they are not passed through the activation function.
- $\boldsymbol{\hat{a}_n}$: The target value for neuron *n*.
- $\boldsymbol{w_{mn}^{l\,(i)}}$: The weight coefficient of the synapse (a_m^{l-1},z_n^l) in the layer *l* at the iteration *i*.
- $\boldsymbol{b_m^{l\,(i)}}$: The bias term m of the layer *l* at the iteration *i*.
- $\boldsymbol{\delta_n^l}$ (delta): The partial derivative of the cost function C with respect to the input value z_n^l of neuron *n* in layer *l*, that is $\delta_n^l = \frac{\partial C_n}{\partial z_n^l}$.

Main Steps of the Computation

The computation begins by assigning initial values to the parameters. Then the following steps are repeated until the cost function reaches an acceptably small value or another termination condition is met:

1. Calculate the neural network with the current parameter values (forward feed):

   $z_n^l = \sum_{m=1}^{k}(a_m^{l-1} * w_{mn}^l) + b_m^l,\ a_n^l = \sigma(z_n^l)$
2. Compute the partial derivatives of the cost function with respect to the parameters using their current values (see below "Algorithm for Computing Partial Derivatives").
3. Change the parameter values in the opposite direction of the partial derivatives, scaled by the learning rate η: $w_{mn}^{l\,(i+1)} = w_{mn}^{l\,(i)} - \eta * \frac{\partial C}{\partial w_{mn}^{l\,(i)}},\ b_m^{l\,(i+1)} = b_m^{l\,(i)} - \eta * \frac{\partial C}{\partial b_m^{l\,(i)}}$

Algorithm for Computing the Partial Derivatives

1. Compute the δ-values for the neurons in the output layer.

   $$\delta_n^l = a_n^l * (1 - a_n^l) * (a_n^l - \hat{a}_n)$$
2. Compute the partial derivatives of the corresponding z_n^l-terms with respect to the preceding parameters w_{mn}^l and b_m^l , and multiply the derivatives by the δ-values:

   $$z_n^l = \sum_{m=1}^{k}(\boldsymbol{a_m^{l-1}} * w_{mn}^l) + \mathbf{1} * b_m^l \Rightarrow \frac{\partial z_n^l}{\partial w_{mn}^l} = \boldsymbol{a_m^{l-1}}, \qquad \frac{\partial z_n^l}{\partial b_m^l} = \mathbf{1}$$

   hence

   $$\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{w_{mn}^l}} = \boldsymbol{a_m^{l-1}} * \boldsymbol{\delta_n^l}, \qquad \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{b_m^l}} = \mathbf{1} * \boldsymbol{\delta_n^l}$$
3. If there are more layers,

   compute the δ-values for the preceding layer

   $$\delta_m^{l-1} = a_m^{l-1} * (1 - a_m^{l-1}) * \sum_{n=1}^{k}(w_{mn}^l * \delta_n^l)$$

   and return to step 2.