UNIVERSITÀ PARTHENOPE
DiST
Dipartimento di Scienze e Tecnologie

Machine Learning (part II)

Error Back-Propagation

Angelo Ciaramella

Introduction

- Goal
  - Evaluating the gradient of the error function for a feed-forward NN
- Learning
  - error backpropagation

Simple linear model

- Outputs

$$y_k = \sum_i w_{ki} x_i$$

- Error

$$E(\mathbf{w}) = \sum_{n=1}^{N} E_n(\mathbf{w})$$

- For a particular input n, the sum of squares error

$$E_n = \frac{1}{2} \sum_k (y_{nk} - t_{nk})^2$$

Simple linear model

- Gradient

$$\frac{\partial E_n}{\partial w_{ji}} = (y_{nj} - t_{nj})x_{ni}$$

- Interpretation
  - Local computation involving the product of an error signal associated with the output and the associated input

Multy-Layer Perceptron

- Each neuron

$$a_j = \sum_i w_{ji} z_i$$

- Non-linear atctivation function

$$z_j = h(a_j)$$

- Forward propagation
  - For each pattern in the training set we calculate all the activations and we obtain the output

Multy-Layer Perceptron

- Derivative of the error

$$E(\mathbf{w}) = \sum_{n=1}^{N} E_n(\mathbf{w})$$

- Partial derivative and chain rule

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} \qquad \delta_j \equiv \frac{\partial E_n}{\partial a_j}$$

- We can write

$$a_j = \sum_i w_{ji} z_i \qquad \frac{\partial a_j}{\partial w_{ji}} = z_i$$

Multy-Layer Perceptron

- and

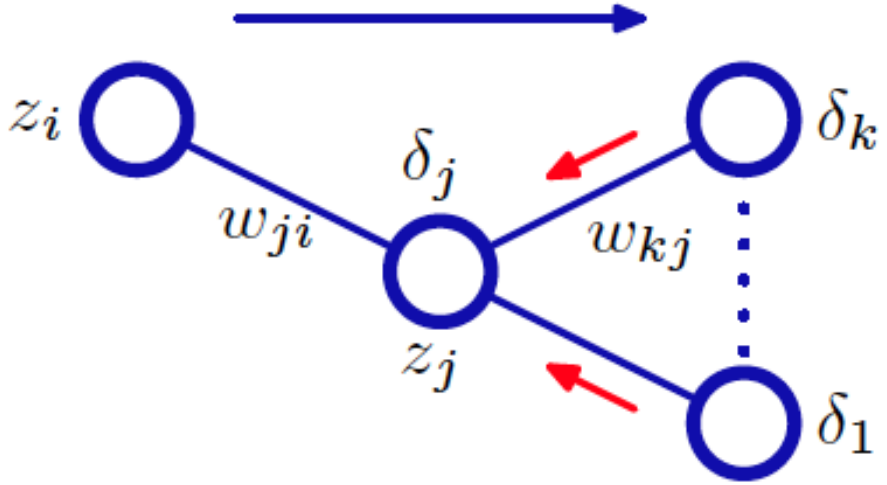
$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i$$

- For the output unit

$$\delta_k = y_k - t_k$$

Multi-Layer Perceptron

- for hidden units



$$\delta_j \equiv \frac{\partial E_n}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial a_j}$$

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k$$

Simple example: two-layer NN

- Activation of hidden units

$$h(a) \equiv \tanh(a)$$

- Derivative

$$h'(a) = 1 - h(a)^2$$

- Error

$$E_n = \frac{1}{2} \sum_{k=1}^{K} (y_k - t_k)^2$$

Simple example: two-layer NN

- Forward propagation

$$
\begin{aligned}
a_j &= \sum_{i=0}^{D} w_{ji}^{(1)} x_i \\
z_j &= \tanh(a_j) \\
y_k &= \sum_{j=0}^{M} w_{kj}^{(2)} z_j
\end{aligned}
$$

Simple example: two-layer NN

- Derivatives

$$\delta_k = y_k - t_k$$

- Hidden units

$$\delta_j = (1 - z_j^2) \sum_{k=1}^{K} w_{kj} \delta_k$$

- First and second layer

$$\frac{\partial E_n}{\partial w_{ji}^{(1)}} = \delta_j x_i, \qquad \frac{\partial E_n}{\partial w_{kj}^{(2)}} = \delta_k z_j$$

Exercise

- Logistic sigmoid activation function

$$h(a) \equiv \frac{1}{1 + e^{-a}}$$

$$h'(a) \equiv h(a)(1 - h(a))$$