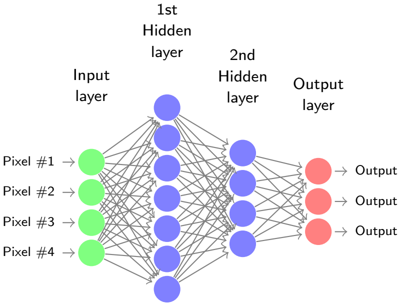
Backprojection

Andrew Swinn

November 28, 2023

Contents

Feedforward



The input matrix has one column / image.

$$input\ s = \begin{pmatrix} pixel_{1\ 1} & pixel_{1\ 2} & \dots & pixel_{1\ 1000} \\ pixel_{2\ 1} & pixel_{2\ 2} & \dots & pixel_{2\ 1000} \\ \dots & \dots & \dots & \dots \\ pixel_{784\ 1} & pixel_{784\ 2} & \dots & pixel_{64\ 1000} \end{pmatrix}$$

The input to neurons in each subsequent layer is:

$$\eta_j = \sum_{i=1}^{784} s_i w_{ij}$$

The neuron output is the governed by the activation function (sigmoidal)

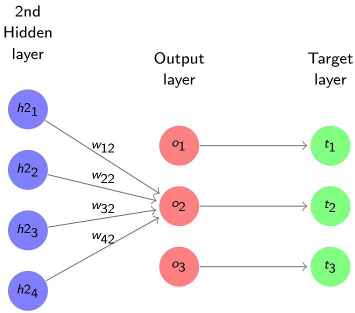
$$h_j = \sigma(\eta_j) \qquad \sigma(z) = \frac{1}{1 + e^{-z}}$$

Finally the output is calculated using the Softmax function.

$$S(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{10} e^{z_j}} \quad for\ i = 1, 2, \dots, 10 \quad z_j = \sum_{i=1}^{64} h_j w_{ij}$$

The output neurons represent a probability distribution. We choose the classification with the highest probability.

Backpropogation - Loss and the last layer



The target vector is a one-hot vector, and represents the correct probability distribution of the classification.

$$target\ t_k = \begin{pmatrix} 0 & 1 & \dots \\ 0 & 0 & \dots \\ 1 & 0 & \dots \end{pmatrix}$$

The cross entropy measures increases with distance between 2 probability distributions.

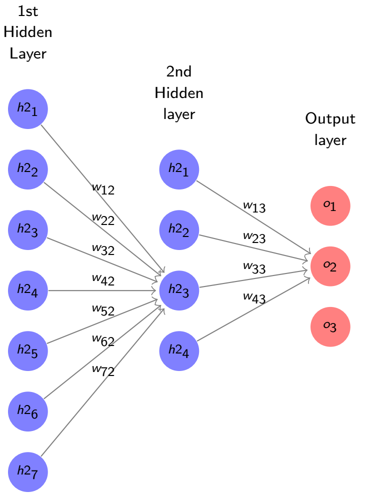
$$Cross\ Entropy\ Error\ CE_j = y_j \log o_j$$

To find the gradient of the error δ_k we differentiate. In this case the gradient is the difference between the target and output nodes.

$$\delta_k = \frac{\partial}{\partial o_k} CE = -(t_k - o_k)$$

Credit/Blame Assignment: weights are changed according to $\Delta w_{ij} = -\delta_j h_i$ - changing weights in proportion to the sending neuron's value AND the error in the resulting neuron.
If o_2 was too high, and $H2_1$ was tiny, make a small change to the weight. Equally,if o_2 was too high, and $H2_1$ was large, make a large change to the weight,

Backpropogation - Loss and the other layers



The error δ_k from the output layer is backpropagated down to subsequent layers by iterating the equation:

$$\delta_j = \left(\sum_k \delta_k w_{jk}\right) h_j (1 - h_j)$$

The first term backpropagates the error using the network weights:

$$\sum_k \delta_k w_{jk} = \begin{pmatrix} w_{11} & w_{21} & \dots \\ w_{31} & w_{12} & \dots \\ w_{22} & w_{32} & \dots \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \dots \end{pmatrix}$$

The second term is the gradient of each sending neuron. In the case or the sigmoidal.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\frac{\partial \sigma}{\partial z} = z(1 - z)$$

$$\Delta w_{ij} = -\epsilon \delta_j h_i$$