7 Logistic neurons

Contents

Abstract

To extending the learning rule for linear neurons to nonlinear multi-layer neurons, we need to extend we need to first extend the learning rule to a *single nonlinear* neuron (we use logistic neurons, though any nonlinear neuron can be used).

(C.f this to chapter 4 of my logistic regression notes from the Machine Learning course)

7.1 Learning the weights of a logistic output neuron

7.1.1 The derivatives of a logistic neuron

Recall the *logit*, $z = b + \sum_i x_i w_i$ and the sigmoid neuron $y = \frac{1}{1+e^{-z}}$ (a special case of the logistic function). Therefore

$$\frac{\partial z}{\partial w_i} = x_i, \ \frac{\partial z}{\partial x_i} = w_i, \ \frac{dy}{dz} = y\,(1-y)$$

Hence the learning rule for a logistic neuron is (using the chain rule for $\frac{\partial y}{\partial w_i} = \frac{\partial z}{\partial w_i}\frac{\partial y}{\partial z} = x_i y\,(1-y)$)

$$\begin{aligned} \frac{\partial E}{\partial w_i} &= \sum_n \frac{\partial y_n}{\partial w_i}\frac{\partial E}{\partial y_n} \\ &= -\sum_n (x_n)_i y_n\,(1-y_n)(t_n - y_n) \end{aligned}$$

Note the delta-rule is the same as the green terms; the extra term defines the slope of the logistic.

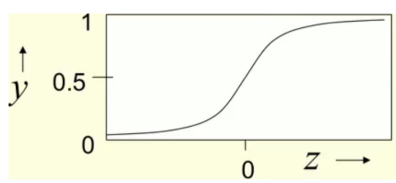

Figure 7.1: Sigmoid neurons are non-linear, yet give a real-valued output that is a smooth and bounded function of their total input, and they also have nice derivatives (i.e. they are a linear function on $\frac{\partial E}{\partial y_n}$, noting $\frac{\partial y_n}{\partial w_i}$ is fixed) - this makes learning easy (this linearity makes it easy to choose small changes in the weights to achieve any desired small change in the output).