Cross entropy

Squared error

In the 80's neural net model the error between an output O and the corresponding desired output y was measured by the squared error:

$$\text{one output:}\quad E(O) = (y - O)^2 \qquad \text{multiple outputs:}\quad E(O_1, \ldots, O_k) = \sum_{j=1}^{k} (y_j - O_j)^2$$

If O is computed by a sigmoid: $O = S(h)$. The error derivative is:

$$\frac{dE(h)}{dh} = 2(y - S(h))S(h)(1 - S(h))$$

Observe that the derivative is small when $S(h)$ is approximately 1 or approximately 0. In particular, if the error is large, in the sense that $S(h) \approx 1$ and $y \approx 0$, the small derivative is a problem.

Cross entropy

A modern alternative is to view the desired output values $y_1, \ldots, y_k$ as representatives of a probability distribution. Similarly, the computed output values $O_1, \ldots, O_k$ are viewed as a distribution. The error is a measure of how far the second distribution is from the first distribution. Cross entropy can be used to measure this distance.

Suppose $p = (p_1, \ldots, p_k)$ and $q = (q_1, \ldots, q_k)$ are two discrete distributions given by two probability vectors. Their cross entropy is defined to be:

$$H(p, q) = -\sum_j p_j \log(q_j) = \sum_j p_j \log(1/q_j)$$

Observe that $H(p, p)$ is the entropy of p, and that $H(p, q) \neq H(q, p)$. The cross entropy is used as follows. The probability distribution p is assumed to be fixed, and q is a candidate for p. Then:

$$q_1 \text{ is better than } q_2 \text{ if:}\quad H(p, q_1) < H(p, q2)$$

Single output

If there is one output variable y, and $0 \le y \le 1$, it can be considered as representing the distribution given by the probability vector $(y, 1 - y)$. Assuming that the output O also satisfied $0 \le O \le 1$, its cross entropy with y is given by:

$$H = y \log(1/O) + (1 - y) \log(1/(1 - O))$$

For the sigmoid case where $O = S(h)$, and using ln for log we have:

$$H(h) = y \ln(1/S(h)) + (1 - y) \ln(1/(1 - S(h)))$$

$$\begin{aligned}\frac{dH(h)}{dh} &= -yS(h)(1 - S(h))/S(h) + (1 - y)S(h)(1 - S(h))/(1 - S(h)) = -y(1 - S(h)) + (1 - y)S(h)\\ &= S(h) - y\end{aligned}$$

Observe that this derivative is always big when the approximation is bad.

Multiple outputs

This can be used when $0 \leq y_j \leq 1$, and $0 \leq O_j \leq 1$. For example when the output is computed by sigmoids. No assumption of probability distribution.

$$H = \frac{1}{k} \sum_{j=1}^{k} y_j \log(1/O_j) + (1 - y_j) \log(1/(1 - O_j))$$

For probability distributions:

$$H = \sum_j y_j \log(1/O_j)$$

One hot / Softmax

Here $y_t = 1$, and $y_j = 0$ for all $j \neq t$. Assume $0 \leq O_j \leq 1$.

$$H = \log(1/O_t)$$

Important to remember:

Cross-entropy tends to allow errors to change weights even when nodes saturate (their derivatives are close to 0.)