The Derivative of Softmax Activation

I found an article that presents a derivation of the derivative of softmax. But this article presents another one I learned from my colleague Ying Cao and is much more concise.

Update 2017/12/10: I just read this awesome blog post, which presents extactly the same derivation as in this article.

The Softmax Activation

The softmax function, $g(z_1, \ldots, z_K)$, as explained in the previous article, has multivariate inputs, $z_1, \ldots, z_K$, and multivariate outputs, $y_1 = \frac{z_1}{\sum_k z_k}, \ldots, y_K = \frac{z_K}{\sum_k z_k}$.

The Derivative of Softmax

In a previous article, we also explained that the partial derivative, $\frac{\partial g}{z_k}$, is essential to the backpropagation algorithm. In this section, let us derive $\frac{\partial g}{z_k}$.

Because softmax has both multivariate input and output, and each of them is K-dimensional, there are $K \times K$ derivatives:

$$\frac{\partial y_i}{\partial z_j},\ 1 \leq i \leq K,\ 1 \leq j \leq K$$

For those elements where $i = j$, we have

$$\frac{\partial y_i}{\partial z_i} = \frac{\partial \frac{e^{z_i}}{\sum_k e^{z_k}}}{\partial z_i} = \frac{e^{z_i} \sum_k e^{z_k} - e^{z_i} e^{z_i}}{\left(\sum_k e^{z_k}\right)^2} = \frac{e^{z_i}}{\left(\sum_k e^{z_k}\right)^2} \frac{\sum_k e^{z_k} - e^{z_i}}{\left(\sum_k e^{z_k}\right)^2} = y_i(1 - y_i)$$

For cases that $i \neq j$, we have

$$\frac{\partial y_i}{\partial z_j} = \frac{\partial \frac{e^{z_i}}{\sum_k e^{z_k}}}{\partial z_j} = \frac{0 \sum_k e^{z_k} - e^{z_i} e^{z_j}}{\left(\sum_k e^{z_k}\right)^2} = -y_i y_j$$

The Cost

When we train a neural network, we need a cost L. Please be aware the output of the cost is a scalar value, not multivariate.

For those whose output layer is softmax, the cost should take two vectors inputs: the softmax output, $y = \{y_1, \dots, y_K\}$, and the truth (label), $t = \{t_1, \dots, t_K\}$. For example, the mean-square-error

$$L(y,t) = \frac{1}{2}\sum_{k=1}^{K}(y_k - t_k)^2$$

According to the multivariate chain rule:

$$\frac{\partial L}{\partial z_k} = \sum_{j=1}^{K}\frac{\partial L}{\partial y_j}\frac{\partial y_j}{\partial z_k}$$

where $\frac{\partial y_j}{\partial z_k}$ is what we examined in the previous section, and for mean-square-error,

$$\frac{\partial L}{\partial y_j} = 2(y_j - t_j)$$

And for cross-entropy cost

$$L(y,t) = \sum_k t_k \log(y_j)$$

we have

$$\frac{\partial L}{\partial y_j} = \frac{t_j}{y_j}$$

Backpropagation

Given the cost, we have

$$\frac{\partial L}{\partial z_k} = \sum_{j=1}^{K}\frac{\partial L}{\partial y_j}\frac{\partial y_j}{\partial z_k} = \sum_{j=1}^{K}\frac{\partial L}{\partial y_j}(-y_j y_k) + \frac{\partial L}{\partial y_k}y_k y_k + \frac{\partial L}{\partial y_k}y_k(1-y_k)$$

Please be aware that the second the the third terms to the right hand side replaces a term in the summation to be the correct one. By merging them, we get

$$\frac{\partial L}{\partial z_k} = y_k \left(\frac{\partial L}{\partial y_k} - \sum_{j=1}^{K} \frac{\partial L}{\partial y_j} y_j \right)$$