Derivative of Softmax with Cross-Entropy Loss

Let's first define softmax, a useful function that offers a smooth (and differentiable) version of the max function with a nice probabilistic interpretation. We denote the softmax function for a logit x_j of K logits (outputs) as $\sigma(x_j)$.

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}$$

Notice that the softmax function properly normalizes the k logits, so we can interpret each $\sigma(x_j)$ as a probability and thee largest logit x_j will have the greatest mass in the distribution.

$$\sum_{j=1}^{K} \sigma(x_j) = \frac{1}{\sum_{k=1}^{K} e^{x_k}} \sum_{j=1}^{K} e^{x_j} = 1$$

For two distributions P and Q, we define cross-entropy over discrete events X as

$$CE = \sum_{x \in X} P(x) \log \frac{1}{Q(x)} = -\sum_{x \in X} P(x) \log Q(x)$$

Cross-entropy comes from information theory, where it is defined as the expected information information quantified as $\log \frac{1}{q}$ of some subjective distribution Q over an objective distribution P. It quantifies how much information we (our model Q) receives when we observe true outcomes from Q – it tells us how far our model is from the true distribution P. We minimize Cross-Entropy when $P = Q$. The value of cross-entropy in this case is known simply as the entropy.

$$H = \sum_{x \in X} P(x) \log \frac{1}{P(x)} = -\sum_{x \in X} P(x) \log P(x)$$

This is the irreducible information we receive when we observe the outcome of a random process. Consider a coin toss. Even if we know the Bernoulli parameter p (the probability of heads) ahead of time, we will never have absolute certainty about the outcome until we actually observe the toss. The greater p is, the more certain we are about the outcome and the less information we expect to receive upon observation.

We can normalize our network output using softmax and then use Cross-Entropy as an objective function. Our softmax outputs represent Q, our subjective distribution. We will denote each softmax output as $\hat{y}_j$ and represent the true distribution P with output labels $y_j = 1$ when the label is for each output. We let $y_j = 1$ when the label is j and $y_j = 0$ otherwise. The result is a degenerate distribution that will aim to estimate P when averaged over the training set. Let's formalize this objective function and take the derivative.

$$
\begin{aligned}
L(\hat{y}, y) &= CE(\hat{y}, y) \\
&= -\sum_{i=1}^{K} y_i \log \hat{y}_i \\
&= -\sum_{i=1}^{K} y_i \log \sigma(x_i) \\
&= -\sum_{i=1}^{K} y_i \log \frac{e^{x_i}}{\sum_{k=1}^{K} e^{x_k}} \\
\frac{\partial L(\hat{y}, y)}{\partial x_j} &= \frac{\partial}{\partial x_j} \left(-\sum_{i=1}^{K} y_i \log \frac{e^{x_i}}{\sum_{k=1}^{K} e^{x_k}} \right) \\
&= \frac{\partial}{\partial x_j} \left(-\sum_{i=1}^{K} y_i (\log e^{x_i} - \log \sum_{k=1}^{K} e^{x_k}) \right) \\
&= \frac{\partial}{\partial x_j} \left(\sum_{i=1}^{K} -y_i \log e^{x_i} + y_i \log \sum_{k=1}^{K} e^{x_k} \right) \\
&= \frac{\partial}{\partial x_j} \left(\sum_{i=1}^{K} -y_i x_i + \sum_{i=1}^{K} y_i \log \sum_{k=1}^{K} e^{x_k} \right)
\end{aligned}
$$

The next step is a little bit more subtle, but recall there is only a single true label for each example and therefore only a single y_i is equal to 1; all others are 0. Therefore we can imagine expanding the sum over i in the second term and only one term of this sum will be 1 and all the others will be zero.

$$
\frac{\partial L(\hat{y}, y)}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\sum_{i=1}^{K} -y_i x_i + \log \sum_{k=1}^{K} e^{x_k} \right)
$$

Now we take the partial derivative and remember that the derivative of a sum is the sum of the derivative of its terms and that any term without x_j can be discarded.

$$
\begin{aligned}
\frac{\partial L(\hat{y}, y)}{\partial x_j} &= \frac{\partial}{\partial x_j}(-y_j x_j) + \frac{\partial}{\partial x_j} \log \sum_{k=1}^{K} e^{x_k} \\
&= -y_j + \frac{1}{\sum_{k=1}^{K} e^{x_k}} \cdot \left(\frac{\partial}{\partial x_j} \sum_{k=1}^{K} e^{x_k} \right) \\
&= -y_j + \frac{1}{\sum_{k=1}^{K} e^{x_k}} \cdot (e^{x_j}) \\
&= -y_j + \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}
\end{aligned}
$$

That second term looks very familiar, huh?

$$
\begin{aligned}
\frac{\partial L(\hat{y}, y)}{\partial x_j} &= -y_j + \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \\
&= -y_j + \sigma(x_j) \\
&= \sigma(x_j) - y_j \\
&= \hat{y}_j - y_j
\end{aligned}
$$

After all that work we end up with a very simple and beautiful expression for the derivative of softmax with cross-entropy divergence with respect to its input. What this is telling us is that when $y_j = 1$, the gradient is negative and thus the opposite direction of the gradient is positive: it is telling us to increase the probability mass of that specific output through the softmax.