gradient of a softmax input into a cross entropy loss function

(3 points) Derive the gradient with regard to the inputs of a softmax function when cross entropy loss is used for evaluation, i.e., find the gradients with respect to the softmax input vector $\boldsymbol{\theta}$, when the prediction is made by $\hat{\boldsymbol{y}} = \text{softmax}(\boldsymbol{\theta})$. Remember the cross entropy function is

$$CE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_i y_i \log(\hat{y}_i) \tag{3}$$

where $\boldsymbol{y}$ is the one-hot label vector, and $\hat{\boldsymbol{y}}$ is the predicted probability vector for all classes. (*Hint: you might want to consider the fact many elements of $\boldsymbol{y}$ are zeros, and assume that only the k-th dimension of $\boldsymbol{y}$ is one.*)

Softmax Gradient

We know softmax (normalized exponential):

$$\sigma(\theta)_j = e^{\theta_j} / \sum_k e^{\theta_k}, \text{ for } k = 1, ..., n$$

where n is the number of dimension of the input vector z.

Let's consider the gradient with respect to theta_i, where first i = j,

$$\frac{d\sigma(\theta_j)}{d\theta_j} = (\sum_k e^{\theta_k} e^{\theta_j} - e^{\theta_j} d/d\theta_j [\sum_k e^{\theta_k}]) / (\sum_k e^{\theta_k} * \sum_k e^{\theta_k})$$

we know

$$d/d\theta_j [\sum_k e^{\theta_k}] = d/d\theta_j [e^{\theta_j} + \sum_{k \neq j} e^{\theta_k}] = e^{\theta_j}$$

so,

$$\frac{d\sigma(\theta_j)}{d\theta_j} = (\sum_k e^{\theta_k} e^{\theta_j} - e^{\theta_j} e^{\theta_j}) / (\sum_k e^{\theta_k} * \sum_k e^{\theta_k}) = \frac{e^{\theta_j}}{\sum_k e^{\theta_k}} * \frac{\sum_k e^{\theta_k} - e^{\theta_j}}{\sum_k e^{\theta_k}} = \sigma(\theta_j)(1 - \sigma(\theta_j))$$

In the case where i ≠ j,

$$\frac{d\sigma(\theta_j)}{d\theta_i} = \frac{0 - e^{\theta_j} e^{\theta_i}}{\sum_k e^{\theta_k} \sum_k e^{\theta_k}} = -\sigma(\theta_j)\sigma(\theta_i).$$

In summary for an input vector x,

$$\frac{d}{dx_i} softmax(x_j) = \begin{cases} softmax(x_j)(1 - softmax(x_j)) & : i = j \\ -\ softmax(x_j) softmax(x_i) & : i \neq j \end{cases}$$

Gradient of Cross Entropy (with softmax input)

With respect to theta,

$$CE(\theta) = -\sum_i y_i \ln(softmax(\theta))$$

The gradient when k, the index of the nonzero element in y_i, equals j is

$$\frac{d}{d\theta_{j=k}} CE(\theta) = \frac{d}{d\theta_k}[-y_k \ln(softmax(\theta_k)) - \sum_{i \neq k} y_i \ln(softmax(\theta_i))]$$

implying

$$\frac{d}{d\theta_{j=k}} CE(\theta) = -y_k 1/softmax(\theta_k) * softmax(\theta_k)(1 - softmax(\theta_k)) - 0$$

$$= softmax(\theta_k) - 1$$

When k ≠ j,

$$\frac{d}{d\theta_{j \neq k}} CE(\theta) = \frac{d}{d\theta_j}[-y_j \ln(softmax(\theta_j)) - \sum_{i \neq j} y_i \ln(softmax(\theta_i))]$$

$$= 0 - \sum_{i \neq j} y_i 1/softmax(\theta_i) * (-softmax(\theta_j) softmax(\theta_i))$$

because softmax(theta_i) still is a function of theta_k too (since it's in the denominator), giving

$$= \sum_{i \neq j} y_i softmax(\theta_j) = softmax(\theta_j)$$

Resources and Mistakes

- don't confuse softmax and sigmoid!
- softmax(x_i) is still a function of x_j even when j ≠ i, since all entries are in the denominator!

A nice walkthrough: https://eli.thegreenplace.net/2016/the-softmax-function-and-its-derivative/

+ Type '/' for commands