word2vec gradients

Tambet Matiisen

October 6, 2015

1 Softmax loss and gradients

Let's denote

$$x_i = \boldsymbol{w_i}^T \hat{\boldsymbol{r}}$$

x_i is a scalar and can be considered as (unnormalized) "similarity" of vectors $\boldsymbol{w_i}$ and $\hat{\boldsymbol{r}}$. Vectors $\boldsymbol{w_i}$ and $\hat{\boldsymbol{r}}$ are both Dx1-dimensional. Full matrix $\boldsymbol{w}$ is VxD dimensional.

$$\hat{y}_i = Pr(word_i|\hat{\boldsymbol{r}}, \boldsymbol{w}) = \frac{e^{x_i}}{\sum_{j=1}^{V} e^{x_j}}$$

$\hat{y}_i$ turns similarity into probability - probability that predicted word $\hat{\boldsymbol{r}}$ is the word $\boldsymbol{w_i}$. V is the total number of words in vocabulary.

$$J = CE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{i=1}^{V} y_i \log(\hat{y}_i)$$

J is the cross-entropy loss, where $\boldsymbol{y}$ is the vector of target values and $\hat{\boldsymbol{y}}$ is the vector of predictions. In $\boldsymbol{y}$ only y_k is 1 and all others are 0s. The loss could have been written also as $J = -\log(\hat{y}_k)$, but gradients are much easier to write with above notation.

1.1 Gradients with respect to $\hat{\boldsymbol{r}}$

$$\frac{\partial J}{\partial \hat{\boldsymbol{r}}} = \sum_{i=1}^{V} \frac{\partial J}{\partial x_i} \frac{\partial x_i}{\partial \hat{\boldsymbol{r}}}$$

The sum is needed, because all x_i-s depend on $\hat{\boldsymbol{r}}$ and $\hat{y}_i$-s (which depend on x_i-s) are summed together in J.

We know cross-entropy loss derivative with respect to softmax input from task 2b:

$$\frac{\partial J}{\partial x_i} = \hat{y}_i - y_i$$

Also

$$\frac{\partial x_i}{\partial \hat{\boldsymbol{r}}} = \boldsymbol{w_i}$$

Altogether

$$\frac{\partial J}{\partial \hat{\boldsymbol{r}}} = \sum_{i=1}^{V} (\hat{y}_i - y_i) \boldsymbol{w_i}$$

Instead of $\boldsymbol{w_i}$ we can write it in terms of full matrix $\boldsymbol{w}$.

$$\frac{\partial J}{\partial \hat{\boldsymbol{r}}} = \boldsymbol{w}^T (\hat{\boldsymbol{y}} - \boldsymbol{y})$$

1.2 Gradients with respect to $\boldsymbol{w_i}$

$$\frac{\partial J}{\partial \boldsymbol{w_i}} = \frac{\partial J}{\partial x_i} \frac{\partial x_i}{\partial \boldsymbol{w_i}}$$

There is no sum this time, because only one x_i depends on w_i. Because

$$\frac{\partial x_i}{\partial \boldsymbol{w_i}} = \hat{\boldsymbol{r}}$$

the result is

$$\frac{\partial J}{\partial \boldsymbol{w_i}} = (\hat{y}_i - y_i) \hat{\boldsymbol{r}}$$

In terms of full matrix $\boldsymbol{w}$ that is

$$\frac{\partial J}{\partial \boldsymbol{w}} = (\hat{\boldsymbol{y}} - \boldsymbol{y}) \hat{\boldsymbol{r}}^T$$

2 Negative sampling loss and gradient

Let's start with notation again. x_i will stay the same.

$$x_i = \boldsymbol{w_i}^T \hat{\boldsymbol{r}}$$

For probability let's use p_i this time, because now we are using *sigmoid* function:

$$p_i = \sigma(x_i)$$

For negative sampling let's use different notation:

$$z_k = -\boldsymbol{w_k}^T \hat{\boldsymbol{r}}$$

$$q_k = \sigma(z_k)$$

Negative sampling loss becomes then:

$$J = J(\hat{\boldsymbol{r}}, \boldsymbol{w_i}, \boldsymbol{w_{1..K}}) = -\log p_i - \sum_{k=1}^{K} \log q_k$$

2.1 Gradient with respect to $\hat{\boldsymbol{r}}$

$$\frac{\partial J}{\partial \hat{\boldsymbol{r}}} = \frac{\partial J}{\partial p_i}\frac{\partial p_i}{\partial x_i}\frac{\partial x_i}{\partial \hat{\boldsymbol{r}}} + \sum_{k=1}^{K}\frac{\partial J}{\partial q_k}\frac{\partial q_k}{\partial z_k}\frac{\partial z_k}{\partial \hat{\boldsymbol{r}}}$$

Here i is a parameter to loss function J (the target word $\boldsymbol{w_i}$), so we can leave it free. Then derivatives are:

$$\frac{\partial J}{\partial p_i} = -\frac{1}{p_i} \qquad \frac{\partial J}{\partial q_k} = -\frac{1}{q_k}$$

$$\frac{\partial p_i}{\partial x_i} = p_i(1-p_i) \qquad \frac{\partial q_k}{\partial z_k} = q_k(1-q_k)$$

$$\frac{\partial x_i}{\partial \hat{\boldsymbol{r}}} = \boldsymbol{w_i} \qquad \frac{\partial z_k}{\partial \hat{\boldsymbol{r}}} = -\boldsymbol{w_k}$$

Altogether

$$\frac{\partial J}{\partial \hat{\boldsymbol{r}}} = -(1-p_i)\boldsymbol{w_i} + \sum_{k=1}^{K}(1-q_k)\boldsymbol{w_k}$$

2.2 Gradient with respect to $\boldsymbol{w_j}$

$$\frac{\partial J}{\partial \boldsymbol{w_j}} = \frac{\partial J}{\partial p_i}\frac{\partial p_i}{\partial x_i}\frac{\partial x_i}{\partial \boldsymbol{w_j}} + \sum_{k=1}^{K}\frac{\partial J}{\partial q_k}\frac{\partial q_k}{\partial z_k}\frac{\partial z_k}{\partial \boldsymbol{w_j}}$$

Notice that we have to use new index j, because the same word may (or may not) appear in positive part as w_i and in negative part as w_k.

Only new derivatives we need to take here are

$$\frac{\partial x_i}{\partial \boldsymbol{w_j}} = \delta_{ij}\hat{\boldsymbol{r}} \qquad \frac{\partial z_k}{\partial \boldsymbol{w_j}} = -\delta_{kj}\hat{\boldsymbol{r}}$$

Where δ_{xy} is Kronecker delta, which is 1 if $x = y$ and 0 otherwise. Altogether

$$\frac{\partial J}{\partial \boldsymbol{w_j}} = -(1-p_i)\delta_{ij}\hat{\boldsymbol{r}} + \sum_{k=1}^{K}(1-q_k)\delta_{kj}\hat{\boldsymbol{r}}$$

We can simplify it further, if we write it in terms of $\boldsymbol{w_i}$ and $\boldsymbol{w_k}$:

$$\frac{\partial J}{\partial \boldsymbol{w_i}} = -(1-p_i)\hat{\boldsymbol{r}} \qquad \frac{\partial J}{\partial \boldsymbol{w_k}} = (1-q_k)\hat{\boldsymbol{r}}$$

So we have to use the left term to update target vector weights and right term to update all negative word weights. Pay attention, that the target word may also occur in negative sample - we want the model learn, that the same word shouldn't occur next to itself. In this case both updates must be applied to target word vector.