CS224n HW1

Xinyu Tan

July 10, 2017

1 Softmax

(a)

Take a look at any element i,

$$\text{softmax}(\boldsymbol{x} + c)_i = \frac{e^{\boldsymbol{x}_i + c}}{\sum_j e^{\boldsymbol{x}_j + c}} = \frac{e^c \cdot e^{\boldsymbol{x}_i}}{e^c \cdot \sum_j e^{\boldsymbol{x}_j}} = \text{softmax}(\boldsymbol{x})_i$$

Therefore, we have $\text{softmax}(\boldsymbol{x} + c) = \text{softmax}(\boldsymbol{x})$

(b)

Note the first case illustrate the broadcasting principle (dimension match) in numpy:

```python
if len(x.shape) > 1:
    #matrix
    x = x - np.max(x, axis=1, keepdims=True)
    x = np.exp(x) / np.sum(np.exp(x), axis=1, keepdims=True)
else:
    # vector
    x = x - x.max() # normalize
    x = np.exp(x)/np.sum(np.exp(x))
```

2 Neural Network Basics

(a)

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1 - e^{-x})^2} = \sigma(x)(1 - \sigma(x))$$

(b)

First, we have

$$\hat{y}_i = \frac{e^{\theta_i}}{\sum_j e^{\theta_j}}$$

For one-hot encoding, only k-th element in $\boldsymbol{y}$ is *one*, so we have

$$CE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\log \hat{y}_k = -\log \frac{e^{\theta_k}}{\sum_j e^{\theta_j}}$$

$$= -\theta_k + \log \sum_j e^{\theta_j}$$

Therefore,

$$\frac{\partial CE(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \theta_k} = -1 + \frac{e^{\theta_k}}{\sum_j e^{\theta_j}}$$

$$\frac{\partial CE(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \theta_i} = \frac{e^{\theta_i}}{\sum_j e^{\theta_j}}, \forall i \neq k$$

Put them altogether,

$$\frac{\partial CE(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{\theta}} = -\boldsymbol{y} + \text{softmax}(\boldsymbol{\theta}) = \hat{\boldsymbol{y}} - \boldsymbol{y}$$

(c)

We have

$$\hat{\boldsymbol{y}} = \text{softmax}(\boldsymbol{h}\boldsymbol{W}_2 + \boldsymbol{b}_2)$$

$$= \text{softmax}\left(\text{sigmoid}(\boldsymbol{x}\boldsymbol{W}_1 + \boldsymbol{b}_1) + \boldsymbol{b}_2\right)$$

Denote $\boldsymbol{a}_2 = \boldsymbol{h}\boldsymbol{W}_2 + \boldsymbol{b}_2$ and $\boldsymbol{a}_1 = \boldsymbol{x}\boldsymbol{W}_1 + \boldsymbol{b}_1$, then

$$\frac{\partial CE(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{x}} = \frac{\partial CE(\boldsymbol{y}, \hat{\boldsymbol{y}})}{\partial \boldsymbol{a}_2} \frac{\partial \boldsymbol{a}_2}{\partial \boldsymbol{x}}$$

$$= (\hat{\boldsymbol{y}} - \boldsymbol{y}) \frac{\partial \boldsymbol{a}_2}{\partial \boldsymbol{h}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}$$

$$= (\hat{\boldsymbol{y}} - \boldsymbol{y}) \boldsymbol{W}_2^T \sigma'(\boldsymbol{a}_1) \boldsymbol{W}_1^T$$

(d)

There are in total $D_x H + H + H D_y + D y$ parameters.

3   word2vec

(a)

For a word $\boldsymbol{o}$, the loss function is

$$L = -y_o \log \hat{y}_o = -\log \frac{\exp(\boldsymbol{u}_o^T \boldsymbol{v}_c)}{\sum_{w=1}^{W} \exp(\boldsymbol{u}_w^T \boldsymbol{v}_c)}$$

$$= -\boldsymbol{u}_o^T \boldsymbol{v}_c + \log \sum_{w=1}^{W} \exp(\boldsymbol{u}_w^T \boldsymbol{v}_c)$$

Therefore, derivative with respect to $\boldsymbol{v}_c$ is:

$$\frac{\partial L}{\partial \boldsymbol{v}_c} = -\boldsymbol{u}_o + \frac{1}{\sum_{l=1}^{W} \exp(\boldsymbol{u}_l^T \boldsymbol{v}_c)} \sum_{w=1}^{W} \exp(\boldsymbol{u}_w^T \boldsymbol{v}_c) \boldsymbol{u}_w$$

$$= -\boldsymbol{u}_o + \sum_{w=1}^{W} p(w|c) \boldsymbol{u}_w$$

where

$$p(w|c) = \frac{\exp(\boldsymbol{u}_w^T \boldsymbol{v}_c)}{\sum_{l=1}^{W} \exp(\boldsymbol{u}_l^T \boldsymbol{v}_c)}$$

.

(b)

Similarly,

$$\frac{\partial L}{\partial \boldsymbol{u}_w} = \begin{cases} -(1 - \hat{y}_o)\boldsymbol{v}_c(& w = o \\ \hat{y}_w \boldsymbol{v}_c & w \neq o \end{cases}$$

(c)

Given

$$J_{\text{neg-sample}}(\boldsymbol{o}, \boldsymbol{v}_c, \boldsymbol{U}) = -\log \sigma(\boldsymbol{u}_o^T \boldsymbol{v}_c) - \sum_{k=1}^{K} \log \sigma(-\boldsymbol{u}_o^T \boldsymbol{v}_c)$$

we have

$$\frac{\partial J}{\partial \boldsymbol{v}_c} = -\sigma(-\boldsymbol{u}_o^T \boldsymbol{v}_c)\boldsymbol{u}_o + \sum_{k=1}^{K} \sigma(\boldsymbol{u}_k^T \boldsymbol{v}_c)\boldsymbol{u}_k$$

and

$$\frac{\partial J}{\partial \boldsymbol{u}_w} = \begin{cases} -\sigma(-\boldsymbol{u}_w^T \boldsymbol{v}_c)\boldsymbol{v}_c & w = o \\ \sigma(\boldsymbol{u}_w^T \boldsymbol{v}_c)\boldsymbol{v}_c & w \neq o \end{cases}$$

(d)

Let's denote $\boldsymbol{U}$ the matrix that aggregates all the output word vectors.

1. Skip-gram:
$$\frac{\partial J_{\text{skip-gram}}}{\partial \boldsymbol{U}} = \sum_{j \neq 0} \frac{\partial F(\boldsymbol{u}_{c+j}, \boldsymbol{v}_c)}{\partial \boldsymbol{U}}$$

$$\frac{\partial J_{\text{skip-gram}}}{\partial \boldsymbol{v}_c} = \sum_{j \neq 0} \frac{\partial F(\boldsymbol{u}_{c+j}, \boldsymbol{v}_c)}{\partial \boldsymbol{v}_c}$$

2. CBOW:
$$\frac{\partial J_{\text{CBOW}}}{\partial \boldsymbol{U}} = \frac{\partial F(\boldsymbol{u}_c, \hat{\boldsymbol{v}})}{\partial \boldsymbol{U}}$$

$$\frac{\partial J_{\text{CBOW}}}{\partial \boldsymbol{v}_j} = \frac{\partial F(\boldsymbol{u}_c, \hat{\boldsymbol{v}})}{\partial \hat{\boldsymbol{v}}}$$