Assignment 1

fro11o

CS224d: Deep Learning for Natural Language Processing

August 16, 2017

1 Softmax

(a)

$$\begin{aligned}\text{softmax}(\boldsymbol{x}+c)_i &= \frac{e^{x_i+c}}{\sum_j e^{x_i+c}} \\ &= \frac{e^{x_i}\cdot e^{c}}{\sum_j e^{x_j}\cdot e^{c}} \\ &= \frac{e^{x_i}}{\sum_j e^{x_j}} \\ &= \text{softmax}(\boldsymbol{x})_i.\end{aligned}$$

2 Neural Network Basics

(a)

$$\begin{aligned}\sigma(x)' &= \frac{\partial}{\partial x}\sigma(x) \\ &= \frac{e^{-x}}{(1+e^{-x})^2} \\ &= \frac{1}{1+e^{-x}}\cdot\frac{e^{-x}}{1+e^{-x}} \\ &= \sigma(x)(1-\sigma(x)).\end{aligned}$$

(b)
Assume the k-th dimension of $\boldsymbol{y}$ is one.

$$
\begin{aligned}
\frac{\partial}{\partial \theta_j} CE(\boldsymbol{y}, \hat{\boldsymbol{y}}) &= \frac{\partial}{\partial \theta_j} - \log \hat{\boldsymbol{y}}_k \\
&= -\frac{1}{\hat{\boldsymbol{y}}_k} \cdot \frac{e^{\theta_k}(\sum_i e^{\theta_i}) \cdot [\![j = k]\!] - e^{\theta_k} \cdot e^j}{(\sum_i e^{\theta_i})^2} \\
&= -\frac{(\sum_i e^{\theta_i})}{e^{\theta_k}} \cdot \frac{e^{\theta_k}(\sum_i e^{\theta_i}) \cdot [\![j = k]\!] - e^{\theta_k} \cdot e^{\theta_j}}{(\sum_i e^{\theta_i})^2} \\
&= \frac{e^{\theta_j}}{(\sum_i e^{\theta_i})} - [\![j = k]\!] \\
&= (\hat{\boldsymbol{y}} - \boldsymbol{y})_j
\end{aligned}
$$

After vectorizing the result, we have

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

(c)
Let

$$
\begin{aligned}
\boldsymbol{z}^{(3)} &= \boldsymbol{h}\boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)} \\
\boldsymbol{z}^{(2)} &= \boldsymbol{x}\boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)}
\end{aligned}
$$

We have

$$
\frac{\partial}{\partial \boldsymbol{z}^{(3)}} J_{CE} = \hat{\boldsymbol{y}} - \boldsymbol{y}
$$

$$
\begin{aligned}
\frac{\partial}{\partial \boldsymbol{b}^{(2)}} J_{CE} &= \sum_i \frac{\partial J_{CE}}{\partial z_i^{(3)}} \frac{\partial z_i^{(3)}}{\partial \boldsymbol{b}^{(2)}} \\
&= \frac{\partial J_{CE}}{\partial \boldsymbol{z}^{(3)}} \\
&= \hat{\boldsymbol{y}} - \boldsymbol{y}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial}{\partial W_{xy}^{(2)}} J_{CE} &= \sum_i \frac{\partial J_{CE}}{\partial z_i^{(3)}} \frac{\partial z_i^{(3)}}{\partial W_{xy}^{(2)}} \\
&= \frac{\partial J_{CE}}{\partial z_y^{(3)}} \cdot h_x \\
\frac{\partial}{\partial \boldsymbol{W}^{(2)}} J_{CE} &= \boldsymbol{h}^T(\hat{\boldsymbol{y}} - \boldsymbol{y})
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial}{\partial a_x^{(2)}} J_{CE} &= \sum_i \frac{\partial J_{CE}}{\partial z_i^{(3)}} \frac{\partial z_i^{(3)}}{\partial a_i^{(2)}} \\
&= (\hat{\boldsymbol{y}} - \boldsymbol{y})(\boldsymbol{W}_x^{(2)})^T \\
\frac{\partial}{\partial \boldsymbol{a}^{(2)}} J_{CE} &= (\hat{\boldsymbol{y}} - \boldsymbol{y})(\boldsymbol{W}^{(2)})^T
\end{aligned}
$$

$$\frac{\partial}{\partial z_x^{(2)}} J_{CE} = \sum_i \frac{\partial J_{CE}}{\partial z_i^{(3)}} \frac{\partial z_i^{(3)}}{\partial z_x^{(2)}}$$
$$= \frac{\partial J_{CE}}{\partial a_x^{(2)}} \cdot \sigma'(z_x^{(2)})$$
$$\frac{\partial}{\partial \boldsymbol{z}^{(2)}} J_{CE} = \frac{J_{CE}}{\partial \boldsymbol{a}^{(2)}} \circ \sigma'(\boldsymbol{z}^{(2)})$$

$$\frac{\partial}{\partial \boldsymbol{b}^{(1)}} J_{CE} = \frac{\partial}{\partial \boldsymbol{z}^{(2)}} J_{CE}$$
$$\frac{\partial}{\partial \boldsymbol{W}^{(1)}} J_{CE} = \boldsymbol{x}^T \frac{\partial J_{CE}}{\partial \boldsymbol{z}^{(2)}}$$

$$\frac{\partial}{\partial \boldsymbol{x}} J_{CE} = \frac{\partial}{\partial \boldsymbol{a}^{(1)}} J_{CE}$$
$$= \frac{\partial J_{CE}}{\partial \boldsymbol{z}^{(2)}} (\boldsymbol{W}^{(1)})^T$$

(d)

$$D_x \cdot H + H \cdot D_y + H + D_y$$

3 word2vec

(a)
Let $\boldsymbol{y}, \hat{\boldsymbol{y}} \in \mathbb{R}^{W \times 1}$, and $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{W \times d}$.

$$\frac{\partial J_{CE}}{\partial \boldsymbol{v}_c} = (\hat{\boldsymbol{y}} - \boldsymbol{y})\boldsymbol{U}$$

(b)

$$\frac{\partial J_{CE}}{\partial (\boldsymbol{u}_w)_x} = \sum_i \frac{\partial J_{CE}}{\partial \theta_i} \frac{\partial \theta_i}{\partial (\boldsymbol{u}_w)_k}$$
$$= (\hat{\boldsymbol{y}} - \boldsymbol{y})_w \cdot (\boldsymbol{v}_c)_x$$
$$\frac{\partial J_{CE}}{\partial \boldsymbol{u}_w} = (\hat{\boldsymbol{y}} - \boldsymbol{y})_w \cdot \boldsymbol{v}_c$$
$$\frac{\partial J_{CE}}{\partial \boldsymbol{U}} = (\hat{\boldsymbol{y}} - \boldsymbol{y})^T (\boldsymbol{v}_c)$$

(c)

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

Let $\boldsymbol{u}_w$ be the word vector of word $\boldsymbol{w}$.

$$\frac{\partial J_{NS}}{\partial \boldsymbol{u}_w} = \begin{cases} -[1-\sigma(\boldsymbol{u}_w^T \boldsymbol{v}_c)]\boldsymbol{v}_c & \text{if } \boldsymbol{w} = \boldsymbol{o} \\ [1-\sigma(-\boldsymbol{u}_w^T \boldsymbol{v}_c)]\boldsymbol{v}_c & \text{if } \boldsymbol{w} \neq \boldsymbol{o}, \boldsymbol{w} \in \{1, 2, ..., K\} \\ 0 & \text{otherwise} \end{cases}$$

In J_{CE} we need to calculate gradient of all W word vectors, but the number decreases to $K+1$ when we apply negative sampling. (speed-up ration $\approx \frac{K}{W}$)
(d)

$$\frac{\partial J_{SG}}{\partial \boldsymbol{v}_c} = \sum_{-m \leq j \leq m, j \neq 0} \frac{\partial F(\boldsymbol{w}_{c+j}, \boldsymbol{v}_c)}{\partial \boldsymbol{v}_c}$$

$$\frac{\partial J_{SG}}{\partial \boldsymbol{U}} = \sum_{-m \leq j \leq m, j \neq 0} \frac{\partial F(\boldsymbol{w}_{c+j}, \boldsymbol{v}_c)}{\partial \boldsymbol{U}}$$

$$\begin{aligned} \frac{\partial J_{CBOW}}{\partial \boldsymbol{v}_{c+j}} &= \frac{\partial J_{CBOW}}{\partial \hat{\boldsymbol{v}}} \frac{\partial \hat{\boldsymbol{v}}}{\partial \boldsymbol{v}_{c+J}} \\ &= \frac{\partial F(\boldsymbol{w}_c, \hat{\boldsymbol{v}})}{\partial \hat{\boldsymbol{v}}} \end{aligned}$$

$$\frac{\partial J_{CBOW}}{\partial \boldsymbol{u}_w} = \begin{cases} \frac{\partial F(\boldsymbol{w}_c, \hat{\boldsymbol{v}})}{\partial \boldsymbol{u}_w} & \text{if } \boldsymbol{w} = \boldsymbol{w}_c \\ 0 & \text{otherwise} \end{cases}$$

(g)

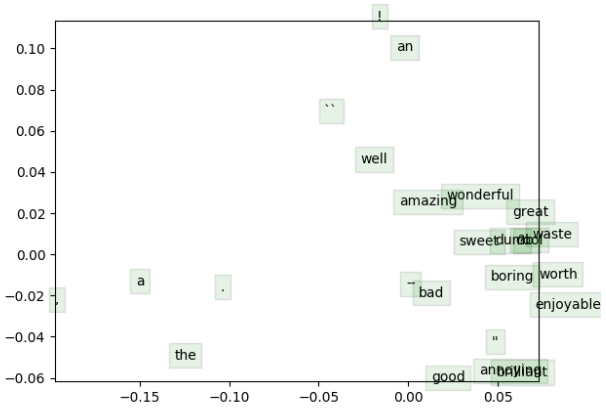

Figure 1: q3_word_vectors.png

4 Sentiment Analysis

(b)
Imposing such a constraint can be interpreted as the prior Bayesian belief that the optimal weights are close to zero.
(c)

```
bestResult = None
for res in results:
    if bestResult == None or \
        bestResult["dev"] < res["dev"]:
        bestResult = res
return res
```

(d)

option	train	dev	test
yourvectors	31.133	32.607	30.362
pretrained	38.642	36.876	37.692

Table 1: Comparison of two different word vector source

The training data used in pretrained wordvectors is larger than ours. It has larger vocabulary size. Also the pretrained model is GloVe which is more powerful than our simple skipgram model.
(e)

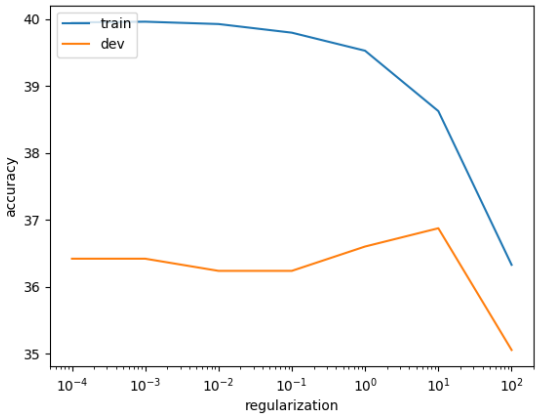

Figure 2: q4_reg_v_acc.png

When λ is small, we have the best training accuracy. That means we let the model overfit our training data. By increasing λ we may find better developing accuracy, and we believe that it is more generalized.
(f)
It seems like our model will predict + and − the most. Not the neutral one.

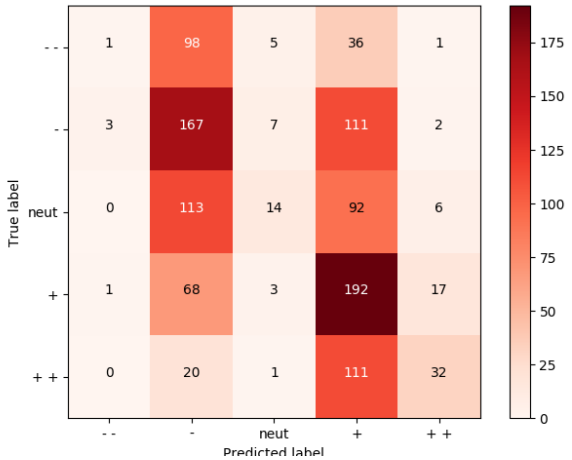

Figure 3: q4_dev_conf.png

(g)

(True: 3, Predicted: 1) and if you 're not nearly moved to tears by a couple of senses, you 've got ice water in your veins .

I guess the model doesn't know the if statement and it just gives a low score caused by some negative word such as tear.

(True: 0, Predicted: 2) i had to look away - this was god awful .

The usage of word "god" is difficult to our model.

(True: 1, Predicted: 3) the best that can be said about the work here of scottish director ritchie ... is that he obviously does n't have his heart in it .

Um... maybe our model doesn't know the meaning of sentence "the best that...".