CS224N Winter 2016 Homework [1]

SUNet ID: [06074217]
Name: [Jiajun Sun]

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

Problem 1

(a) The proof is shown as follow:

$$softmax(x+c)_i = \frac{e^{x_i+c}}{\sum_j e^{x_j+c}} = \frac{e^{x_i}e^c}{\sum_j e^{x_j}e^c} = \frac{e^{x_i}}{\sum_j e^{x_j}} = softmax(x)$$

Problem 2

(a) The gradient of sigmoid function

$$\frac{\partial \sigma(x)}{\partial x} = \frac{1}{(1+e^{-x})^2} e^{-x} = \sigma(x)^2 (\frac{1}{\sigma(x)} - 1) = \sigma(x) - \sigma(x)^2$$

(b) The gradient of cross entropy loss function

$$\frac{\partial CE(y, \hat{y})}{\partial \theta} = -\sum_i y_i \frac{softmax(\theta)'}{\hat{y}_i} = -\sum_i y_i \frac{softmax(\theta)'_i}{softmax(\theta)_i}$$

Consider only the k-th element of y is one:

$$\frac{\partial CE(y, \hat{y})}{\partial \theta} = -\frac{softmax(\theta)'_k}{\hat{y}_k}$$

Where,

$$\begin{aligned} softmax(\theta)'_k &= \frac{\partial softmax(\theta)_k}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} \frac{e^{\theta_k}}{\sum_j e^{\theta_j}} \\ &= \frac{\frac{\partial e^{\theta_k}}{\partial \theta_i}}{\sum_j e^{\theta_j}} - e^{\theta_i} \frac{e^{\theta_k}}{(\sum_j e^{\theta_j})^2} \\ &= \frac{\mathbf{y} e^{\theta_k}}{\sum_j e^{\theta_j}} - \frac{e^{\theta} e^{\theta_k}}{(\sum_j e^{\theta_j})^2} \\ &= \hat{y}_k (\mathbf{y} - \hat{\mathbf{y}}) \end{aligned}$$

Therefore,

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

(c) Use back propogation,

$$\frac{\partial J}{\partial \mathbf{x}} = \frac{\partial J}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{x}}$$

first calculate gradient of first two:

$$\frac{\partial J}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}} = \frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial \mathbf{h}} = (\hat{\mathbf{y}} - \mathbf{y}) \mathbf{W_2}^T$$

Then calculate the gradient of the hidden layer:

$$\frac{\partial \sigma(\mathbf{x}\mathbf{W_1} + \mathbf{b_1})}{\partial \mathbf{x}} = \sigma(\mathbf{x}\mathbf{W_1} + \mathbf{b_1}))(1 - \sigma(\mathbf{x}\mathbf{W_1} + \mathbf{b_1})))\mathbf{W_1}^T$$

By combining them together we got:

$$\frac{\partial J}{\partial \mathbf{x}} = [(\hat{\mathbf{y}} - \mathbf{y}) \mathbf{W_2}^T][\sigma(\mathbf{x}\mathbf{W_1} + \mathbf{b_1}))(1 - \sigma(\mathbf{x}\mathbf{W_1} + \mathbf{b_1}))]\mathbf{W_1}^T$$

(d) There are parameters stored in $\mathbf{W_1}$, $\mathbf{W_2}$, $\mathbf{b_1}$, $\mathbf{b_2}$.

$\mathbf{W_1}$ has dimension $D_x \times H$
$\mathbf{W_2}$ has dimension $D_y \times H$
$\mathbf{b_1}$ has dimension $1 \times H$
$\mathbf{b_2}$ has dimension $1 \times D_y$

Therefore, there are $H(D_y + D_x) + D_y + H$ parameters in this neural network.

Problem 3

(a) First denote:

$$Z = u^T \nu_c$$

Use chain rule to derive the gradient:

$$\frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial \nu_c} = \sum_i \sum_j \frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial \hat{\mathbf{y}}_i} \frac{\partial \hat{\mathbf{y}}_i}{\partial Z_j} \frac{\partial Z_j}{\partial \nu_c}$$

The result should has the same dimension as ν_c which is W. Assume word o is the expected word, therefore, only when index $i = o$ the gradient is not zero:

$$\frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial \nu_c} = \sum_j \frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial \hat{\mathbf{y}}_o} \frac{\partial \hat{\mathbf{y}}_o}{\partial Z_j} \frac{\partial Z_j}{\partial \nu_c}$$

From problem 2 we know:

$$\frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial \hat{\mathbf{y}}_o} \frac{\partial \hat{\mathbf{y}}_o}{\partial Z_j} = (\hat{y}_j - y_j) = (\hat{y}_j - \mathbf{1}[j = o])$$

And it is easy to find:

$$\frac{\partial Z_j}{\partial \nu_c} = \mathbf{u}_j$$

Therefore, combine them together:

$$\frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial \nu_c} = \sum_j (\hat{y}_j - \mathbf{1}[j = o])\mathbf{u}_j$$

(b) Use chian rule to derive the gradient:

$$\frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial u_w} = \sum_i \sum_j \frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial \hat{\mathbf{y}}_i} \frac{\partial \hat{\mathbf{y}}_i}{\partial Z_j} \frac{\partial Z_j}{\partial u_w}$$

It is easy to find that index $i = o$ and $j = w$ the gradient is not zero:

$$\frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial u_w} = \frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial \hat{\mathbf{y}}_i} \frac{\partial \hat{\mathbf{y}}_i}{\partial Z_w} \frac{\partial Z_w}{\partial u_w}$$

$$\frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial \hat{\mathbf{y}}_o} \frac{\partial \hat{\mathbf{y}}_o}{\partial Z_w} = (\hat{\mathbf{y}}_\mathbf{w} - \mathbf{y}_\mathbf{w}) = (\hat{\mathbf{y}}_\mathbf{w} - \mathbf{1}[w = o])$$

$$\frac{\partial Z_w}{\partial u_w} = \nu_\mathbf{c}$$

Therefore, combine them together:

$$\frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial u_w} = (\hat{\mathbf{y}}_\mathbf{w} - \mathbf{1}[w = o])\nu_\mathbf{c}$$

(c) Gradient for ν_c:

$$\begin{aligned}
\frac{\partial J}{\partial \nu_c} &= -\frac{\frac{\partial \sigma(u_o^T \nu_c)}{\partial \nu_c}}{\sigma(u_o^T \nu_c)} - \sum_k \frac{\frac{\partial \sigma(-u_k^T \nu_c)}{\partial \nu_c}}{\sigma(-u_k^T \nu_c)} \\
&= -\frac{\sigma(u_o^T \nu_c)(1 - \sigma(u_o^T \nu_c))u_o}{\sigma(u_o^T \nu_c)} - \sum_k \frac{\sigma(-u_k^T \nu_c)(1 - \sigma(-u_k^T \nu_c))(-u_k)}{\sigma(-u_k^T \nu_c)} \\
&= (\sigma(u_o^T \nu_c) - 1)u_o + \sum_{k, k \neq o} (1 - \sigma(-u_k^T \nu_c))u_k
\end{aligned}$$

Gradient for u_w:

$$\begin{aligned}\frac{\partial J}{\partial u_w} &= -\frac{\frac{\partial \sigma(u_o^T \nu_c)}{\partial u_w}}{\sigma(u_o^T \nu_c)} - \sum_k \frac{\frac{\partial \sigma(-u_k^T \nu_c)}{\partial u_w}}{\sigma(-u_k^T \nu_c)} \\ &= -\mathbf{1}[o=w]\frac{\sigma(u_o^T \nu_c)(1-\sigma(u_o^T \nu_c))u_o}{\sigma(u_o^T \nu_c)} - \mathbf{1}[o \neq w]\frac{\sigma(-u_w^T \nu_c)(1-\sigma(-u_w^T \nu_c))(-\nu_c)}{\sigma(-u_w^T \nu_c)} \\ &= \mathbf{1}[o=w](\sigma(u_o^T \nu_c)-1)\nu_c + \mathbf{1}[o \neq w](1-\sigma(-u_w^T \nu_c))\nu_c\end{aligned}$$

when $o = w$

$$\frac{\partial J}{\partial u_0} = (\sigma(u_o^T \nu_c) - 1)\nu_c \tag{1}$$

when $o \neq w$

$$\frac{\partial J}{\partial u_w} = (1 - \sigma(-u_w^T \nu_c))\nu_c \tag{2}$$

(d) In skip-gram, we use the center word to predict context words, then we take gradient W.R.T ν_c:

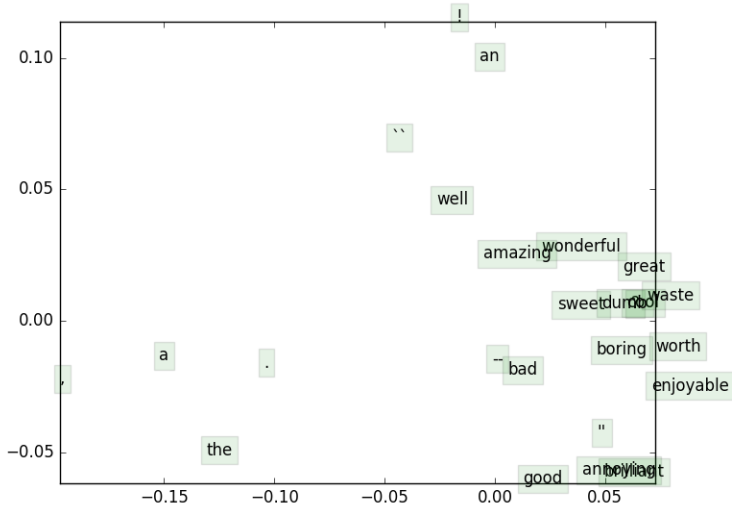
$$\frac{\partial J_{skip-gram}(word_{c-m\ldots c+m})}{\partial \nu_c} = \sum_{-m \leq j \leq m, j \neq 0} \frac{\partial F(w_{c+j}, \nu_c)}{\partial \nu_c}$$

$$\frac{\partial J_{skip-gram}(word_{c-m\ldots c+m})}{\partial \nu_k} = 0, k \neq c$$

In CBOW, the predicted vector contain all input vectors of that corpus, therefore take gradient W.R.T to ν_{c+j}:

$$\frac{\partial J_{CBOW}(word_{c-m\ldots c+m})}{\partial \nu_{c+j}} = \frac{\partial F(w_c, \hat{\nu})}{\partial \hat{\nu}}$$

$$\frac{\partial J_{CBOW}(word_{c-m\ldots c+m})}{\partial \nu_{c+k}} = 0, k \notin [-m, m]$$



(e) In the above plot we can see: most adjective word convey sentiment meanings are clustered together. This plot shows the different roles a word play in a sentence. If visulise more dimensions, we can expect the seperation of different adjective words.

Problem 4

(a)

(b) By adding regularization is to prevent overfitting. Regularization by adding penaties to the norm of estimates can limit the variance of the model with a compromise on the training error.

(c)
```
def chooseBestModel(results):
    """Choose the best model based on parameter tuning on the dev set

    Arguments:
    results -- A list of python dictionaries of the following format:
        {
            "reg": regularization,
            "clf": classifier,
            "train": trainAccuracy,
            "dev": devAccuracy,
            "test": testAccuracy
        }

    Returns:
    Your chosen result dictionary.
    """
    bestResult = None

    ### YOUR CODE HERE
    dev_accuracy_list = [model['dev'] for model in results]
    idx = np.argmax(dev_accuracy_list)
    bestResult = results[idx]
    ### END YOUR CODE

    return bestResult
```

(d) For pretrained vector, when regularization value is 12.1, the train, dev and test are 38.659, 37.148 and 37.557 respectively.
For our own vector, when regularization value is $2.61E-06$, the train, dev and test are 31.016, 32.698 and 30.271 respectively.

The best results are listed in the table below:

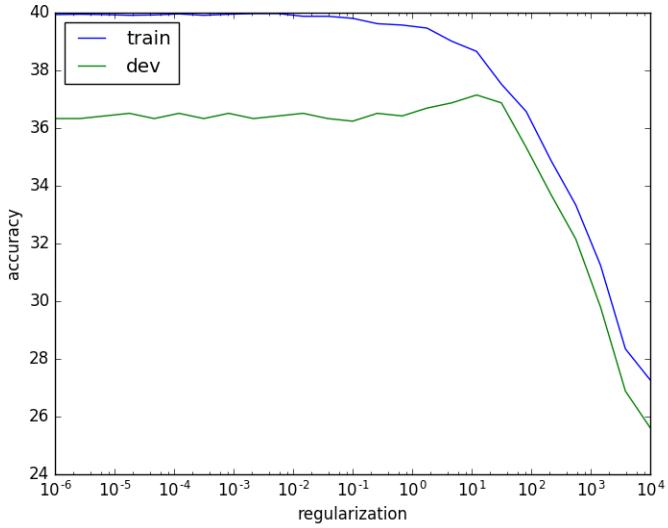
	Pretrained	your-vectors
train	39.958	31.180
test	37.148	32.698
dev	37.557	30.271

There are three reasons that Pretrained vectors has higher accuracy:

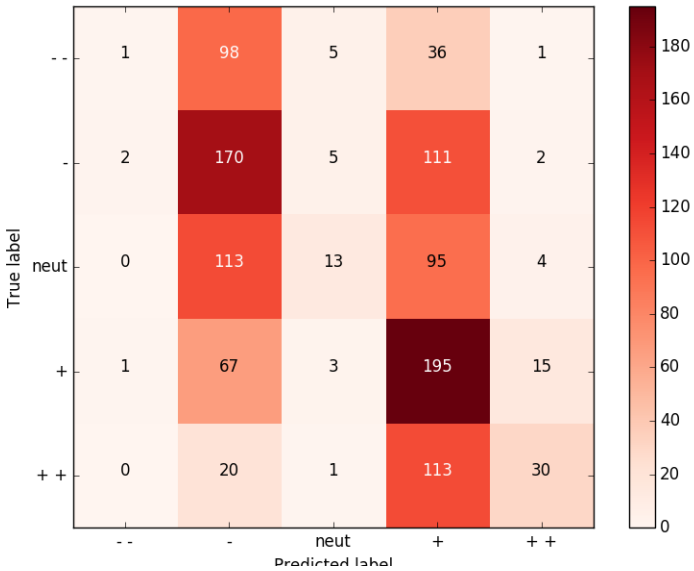
(1) GloVe combine u and v, therefore the vector it generates might be more representative than word2vec methods.
(2) Pretrained and the vectors we trained use different datasets. Wikipedia may have larger vacabulary sets than SST datasets. As a result, word vectors trained from Wikipedia will contain more meanings and more comprehensive.
(3) When training our own word vectors we use a one layer model, this model could be simpler than

how GloVe vector was trained. By implementing a deeper layer model might increase the accuracy of our word vectors.

(e) Both train and dev accuracy decrease with the increase of regularization parameters. With higher regularization value, the regularization parameter C for logistic regreesion become smaller. As described in sklearn, smaller C means stronger regularization strength. This will lead model underfitting.



(f) Prediction on positive meaning and negative meaning are relative balanced but slightly towards the positive side. Our model does not have enough variance to predict the true label. Most of the prediction falls on eight moderate positive or moderate negative.



(g) **Example 1: 3 1 we know the plot 's a little crazy , but it held my interest from start to finish.**

The word "crazy" has convey too much negative meaning, however the sentiment analysis does not consider the "but" which flip the meaning of the sentence.

Example 2: 3 1 not far beneath the surface , this reconfigured tale asks disturbing questions about those things we expect from military epics.
This sentence's meaning is not obvious. By adding up all vector can not figure out the meaning

Example 3: 4 1 manages to transcend the sex , drugs and show-tunes plot into something far richer.
Here transcend should have more weights than other words.