CS224D Assignment1

Nathan Wan

April 2016

1 Softmax

(a)

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

2 Neural Network Basics

(a)

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

By chain rule:

$$
\begin{aligned}
\frac{d}{dx}\sigma(x) &= \frac{-1}{(1+e^{-x})^2}\frac{d}{dx}(1+e^{-x}) \\
&= \frac{-1}{(1+e^{-x})^2}(-e^{-x}) \\
&= \frac{1}{1+e^{-x}}(\frac{1+e^{-x}-1}{1+e^{-x}}) \\
&= \frac{1}{1+e^{-x}}(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}) \\
&= \sigma(x)(1-\sigma(x))
\end{aligned}
$$

(b) NOTE: Please note that variables $\hat{y}$, y, x, θ, h and a without subscripts are vectors.

Since

$$\hat{y} = softmax(\theta)$$

then

$$\hat{y}_i = softmax(\theta)_i = \frac{e^{\theta_i}}{\sum_j e^{\theta_j}}$$

Using division derivative rule:

$$\frac{\partial}{\partial \theta_{w=i}} softmax(\theta)_i = \frac{(\sum_j e^{\theta_j})e^{\theta_i} - (e^{\theta_i})^2}{(\sum_j e^{\theta_j})^2} = softmax(\theta)_i - softmax(\theta)_i^2$$

and

$$\frac{\partial}{\partial \theta_{w \neq i}} softmax(\theta)_i = \frac{(\sum_j e^{\theta_j})0 - e^{\theta_i}(e^{\theta_w})}{(\sum_j e^{\theta_j})^2} = -softmax(\theta)_i softmax(\theta)_w$$

thus

$$\frac{\partial}{\partial \theta} CE(y, \hat{y}) = \frac{\partial}{\partial \theta} - \sum_i y_i log(\hat{y}_i) = -\sum_i y_i \frac{\partial}{\partial \theta} log(\hat{y}_i) = -y_k \frac{1}{\hat{y}_k} \frac{\partial}{\partial \theta} \hat{y}_k$$

since $y_i = 0 : \forall i \neq k$. For $\theta_{w=k}$

$$-\frac{1}{\hat{y}_w} \frac{\partial}{\partial \theta} \hat{y}_w = -\frac{1}{\hat{y}_w}(1 - \hat{y}_w)\hat{y}_w = \hat{y}_w - 1$$

and for $\theta_{w \neq k}$

$$-\frac{1}{\hat{y}_w} \frac{\partial}{\partial \theta} \hat{y}_w = -\frac{1}{\hat{y}_w}(-\hat{y}_k \hat{y}_w) = \hat{y}_w$$

Since $\mathbf{y}$ is a one-hot vector,

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

(c) Let $\theta = hW_2 + b_2$ so that

$$\hat{y} = softmax(hW_2 + b_2) = softmax(\theta)$$

and let $a = xW_1 + b_1$ so that

$$h = \sigma(xW_1 + b_1) = \sigma(a)$$

Also consider for any linear layer $y = xW + b$ in a network, the partial derivative must be

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial x} W^T$$

First, with the input layer

$$\frac{\partial J}{\partial x} = \frac{\partial J}{\partial a}\frac{\partial a}{\partial x} = \frac{\partial J}{\partial a} W_1^T$$

by chain rule and since the partial is applied element-wise

$$\frac{\partial J}{\partial a} = \frac{\partial h}{\partial a}\frac{\partial J}{\partial h} = \left(h(h-1) \circ \frac{\partial J}{\partial h}\right)$$

but is equivalent to a square matrix with those values along the diagonal. Since h is a linear transformation of θ

$$\frac{\partial J}{\partial h} = \frac{\partial J}{\partial \theta}\frac{\partial \theta}{\partial h} = \frac{\partial J}{\partial \theta} W_2^T$$

And as seen from the previous section, $\frac{\partial J}{\partial \theta} = \frac{\partial}{\partial \theta} CE(y, \hat{y}) = \hat{y} - y$. When put together,

$$\frac{\partial J}{\partial x} = \left(h(h-1) \circ (\hat{y} - y) W_2^T\right) W_1^T$$

Also note that dimensions check out:

$$\mathbb{R}^{1 \times D_x} = \left(\mathbb{R}^{1 \times H} \circ (\mathbb{R}^{1 \times D_y}) \mathbb{R}^{D_y \times H}\right) \mathbb{R}^{H \times D_x}$$

(d) Between the input and hidden layer, there is a linear transformation of D_x input units and a bias term for h hidden layer units,

$$(D_x + 1) \times H$$

and similarly between the hidden and output layers,

$$(H + 1) \times D_y$$

. Total parameters is the sum of those two quantities.

$$(D_x + 1)H + (H + 1)D_y$$

3 word2vec

(a) It is given

$$\hat{y}_o = \frac{exp(u_o^T v_c)}{\sum_w exp(u_w^T v_c)}$$

and that $u_i : i \in |U|$ and v_o are column vectors.

$$\begin{aligned}
\frac{\partial}{\partial v_c} J_{sm} &= \frac{\partial}{\partial v_c} - log(\hat{y}_o) \\
&= -\frac{\partial}{\partial v_c} u_o^T v_c + \frac{\sum_w exp(u_w^T v_c) \frac{\partial}{\partial v_c} u_w^T v_c}{\sum_w exp(u_w^T v_c)} \\
&= -u_o + \sum_w \hat{y}_w u_w
\end{aligned}$$

(b)

$$\begin{aligned}
\frac{\partial}{\partial u_i} J_{sm} &= \frac{\partial}{\partial u_i} - log(\hat{y}_o) \\
&= -\frac{\partial}{\partial u_i} u_o^T v_c + \frac{\sum_w exp(u_w^T v_c) \frac{\partial}{\partial u_i} u_w^T v_c}{\sum_w exp(u_w^T v_c)} \\
&= -\frac{\partial}{\partial u_i} u_o^T v_c + \sum_w \hat{y} \frac{\partial}{\partial u_i} u_w^T v_c
\end{aligned}$$

Let's distinguish between $i = o$ and $i \neq o$.

$$\frac{\partial}{\partial u_{i=o}} J_{sm} = -v_c + \hat{y}_i v_c = (\hat{y}_i - 1) v_c$$

$$\frac{\partial}{\partial u_{i \neq o}} J_{sm} = \hat{y}_i v_c$$

and so

$$\frac{\partial}{\partial U} J_{sm} = v_c * (\hat{y} - y)$$

where $U \in \mathbb{R}^{d \times V}$, $v_c \in \mathbb{R}^{d \times 1}$ and $(\hat{y} - y) \in \mathbb{R}^{1 \times V}$.

(c) Given

$$J_{ns} = -log(\sigma(u_o^T v_c)) - \sum_k log(\sigma(-u_k^T v_c))$$

and that

$$\frac{\partial}{\partial x} log(\sigma(x)) = \frac{1}{\sigma(x)} (1 - \sigma(x)) \sigma(x) = 1 - \sigma(x)$$

and that

$$\sigma(-x) = 1 - \sigma(x)$$

we see that

$$\begin{aligned}\frac{\partial}{\partial v_c}J_{ns} &= -(1-\sigma(u_o^T v_c))u_o - \sum_k (1-\sigma(-u_k^T v_c))(-u_k)\\ &= -(1-\sigma(u_o^T v_c))u_o + \sum_k \sigma(u_k^T v_c)u_k\end{aligned}$$

and

$$\begin{aligned}\frac{\partial}{\partial u_{i=o}}J_{ns} &= -(1-\sigma(u_o^T v_c))v_c\\ \frac{\partial}{\partial u_{i\in K}}J_{ns} &= -(1-\sigma(-u_k^T v_c))(-v_c)\\ &= \sigma(u_k^T v_c)v_c\end{aligned}$$

This cost function is more efficient to compute because there is no summation over the entire vocabulary of $|U|$, instead, we only look at K samples so the computation speed up is on the order of $\frac{O(|U|)}{O(K)}$.

(d) For Skipgram:

$$\frac{\partial}{\partial v_c}J_{sg} = \sum_{-m\le j\le m, j\ne 0} \frac{\partial}{\partial v_c}F(u_{c+j}, v_c)$$

and

$$\frac{\partial}{\partial U}J_{sg} = \sum_{-m\le j\le m, j\ne 0} \frac{\partial}{\partial U}F(u_{c+j}, v_c)$$

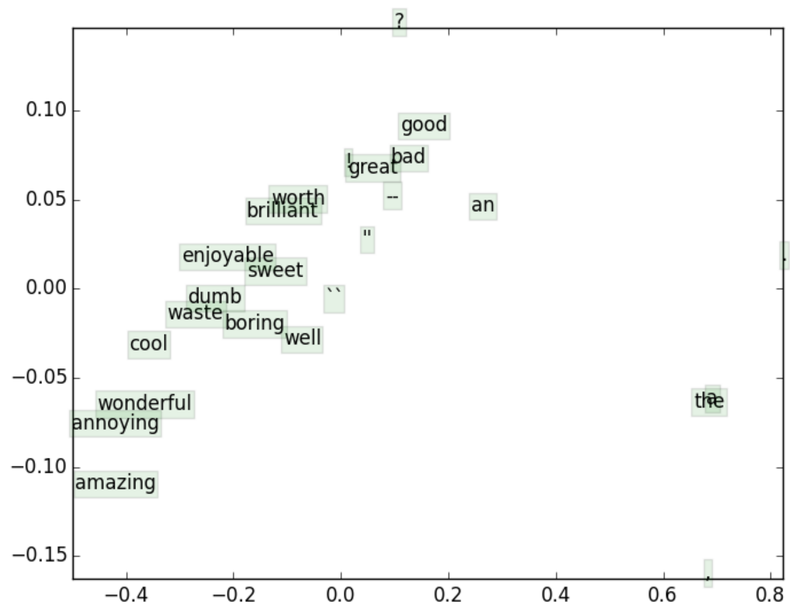
For CBOW, note that since the center word is an output vector:

$$\frac{\partial}{\partial U}J_{CBOW} = \frac{\partial}{\partial U}F(u_c, \hat{v})$$

and for $j \in \{-m \le j \le m, j \ne 0\}$,

$$\frac{\partial}{\partial v_j}F(u_c, \hat{v}) = \frac{\partial F}{\partial \hat{v}}\frac{\partial \hat{v}}{\partial v_j} = \frac{\partial}{\partial \hat{v}}F(u_c, \hat{v})$$

(g) Some stop words, like ”a” and ”the” are clearly separate from the other words; ”an” is also a stop word, but probably there wasn't enough examples for it to be distinguished. Similarly, only some more frequent punctuation like ”?” are distinguishable. The adjectives are grouped in a very distinct shape, but it's difficult to glean something significant from the words within the shape.



4 Sentiment Analysis

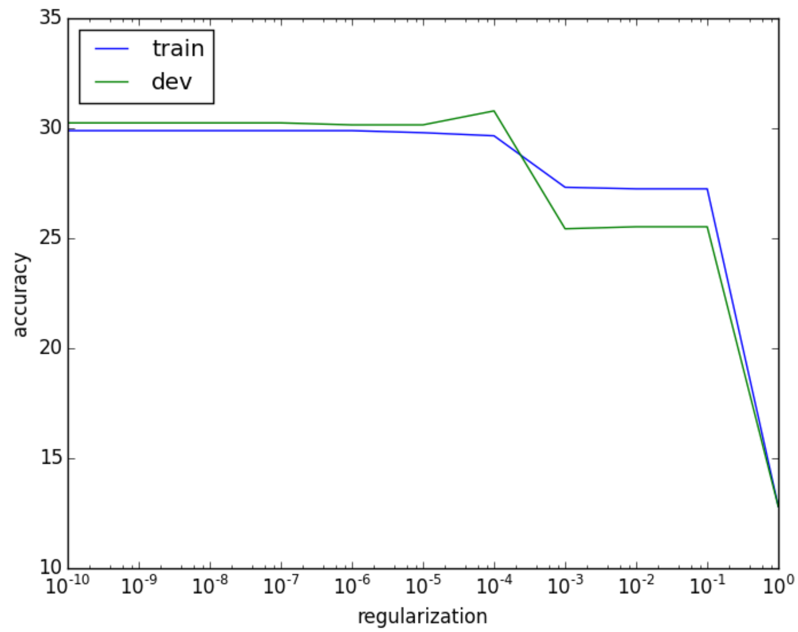
(b) Regularization keep the model parameters from overfitting training data.

(c) I selected the regularization value that maximized dev accuracy. I first swept values from 10^{-10} to 10^{0} before narrowing in on the 10^{-4} region. Despite a step of $log(.2)$, the best value was still 10^{-4}:

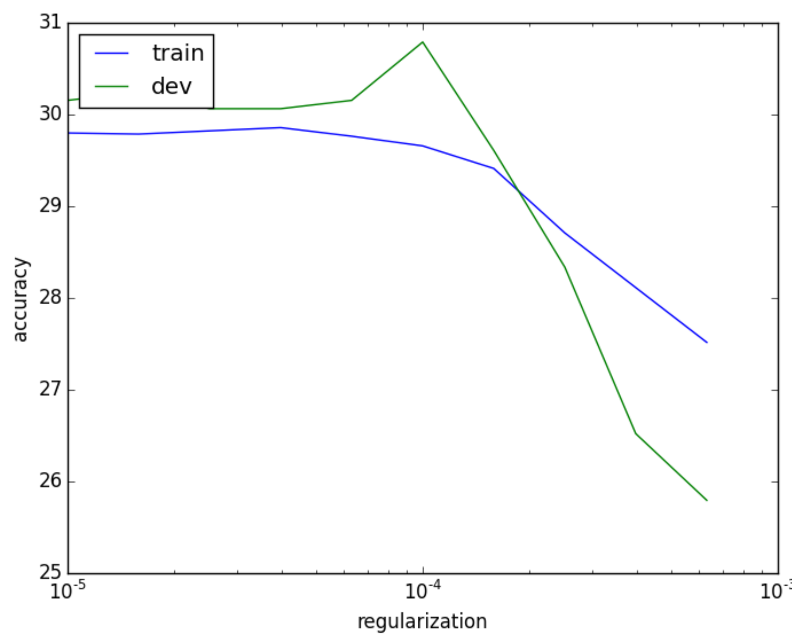
Train	29.658240
Dev	30.790191
Test	27.285068

Oddly, by accident I tested 10^{-6} and I actually got a test accuracy of 28.1, which suggests we should have a stronger bias towards lower regularization values.

(d) As we increase the regularization factor, we see a small increase in accuracy until about 10^{-4}. After which, regularization will keep the model from correctly learning features in the data.



35
30
25
20
15
10
train
dev
accuracy
10^{-10}
10^{-9}
10^{-8}
10^{-7}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
regularization



31
30
29
28
27
26
25
train
dev
accuracy
10^{-5}
10^{-4}
10^{-3}
regularization