CS244N Assignment 2

liuhanzuo

January 16, 2025

1 Problem 1

a

Note that

$$y_w = \begin{cases} 1 & \text{if } w = o \\ 0 & \text{otherwise} \end{cases}$$

Thus

$$-\sum_{w \in Vocab} y_w \log \hat{y}_w = -\log \hat{y}_o$$

b

$$J(v_c, o, U) = -\log P(O = o|C = c) = \log \sum_w \exp(u_w^T v_c) - u_o^T v_c$$

$$\frac{\partial}{\partial v_c} J(v_c, o, U) = \frac{\partial}{\partial v_c}(\log \sum_w \exp(u_w^T v_c) - u_o^T v_c) = \frac{\sum_w \exp(u_w^T v_c) u_w}{\sum_w \exp(u_w^T v_c)} - u_o$$

When the gradient is zero, we have

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

Explanation: the probability for a prediction u_w is $\hat{y_w} = \frac{\exp u_w^T v_c}{\sum_w \exp u_w^T v_c}$. Thus we hope that when the gradient is zero, the expectation of the word is

$$\sum_w \hat{y_w} u_w = u_o$$

$$\hat{y}^T \cdot U = y^T \cdot U$$

c

If we have $u_x = \alpha u_y$, then the normalization term

$$\frac{u_x}{\|u_x\|_2} = \frac{u_y}{\|u_y\|_2}$$

When calculating the cosine similarity of two vectors of u_x and u_y, we want to calculate

$$\langle u_x, u_y \rangle = \frac{u_x}{\|u_x\|_2} \cdot \frac{u_y}{\|u_y\|_2}$$

The terms in normalized terms indicates the probability distribution for prediction of words.

d

$$\frac{\partial}{\partial u_o} J(v_c, o, U) = \frac{v_c \exp(u_o^T v_c)}{\sum_w \exp(u_w^T v_c)} - v_c = v_c(\hat{y_o} - 1)$$

$$\frac{\partial}{\partial u_w} (v_c, o, U)(w \neq o) = \frac{v_c \exp(u_w^T v_c)}{\sum_w \exp(u_w^T v_c)} = v_c \hat{y_w}$$

c

$$\frac{\partial}{\partial U} J(v_c, o, U) = \sum_w \frac{\partial}{\partial u_w} J(v_c, o, U) \cdot \delta_w^T$$

Where δ_w is the one-hot vector for word w.

2 Problem 2

a

m is the momentum term, which takes the tendency of the gradient change, maintain most part of the previous term and add a small part of the current gradient. It can be seen as a weighted average of the previous gradients. If a stratically change occurs on current gradient, the momentum will not vary that much(because the gradient term is multiplied by $1 - \beta_1$). Thus the momentum could make the learning process more fluent and stable.
Since Adam divide the term by $\sqrt{v}$, the larger update parameters would be the one with smaller variance. This could make the learning process more stable and avoid the overshooting problem.

b

$$\gamma = \frac{1}{1 - p_{\text{drop}}}$$

This equation can be easily gained from the expectation:

$$\mathbb{E}[h_{\text{drop}}]_i = \begin{cases} h_i & w.p.\ 1 - p_{\text{drop}} \\ 0 & w.p.\ p_{\text{drop}} \end{cases}$$

To balance the expectation, we need multiple each term by $\frac{1}{1-p_{\text{drop}}}$.
To avoid the model only use part of the nuerons and ignore the others. With constant dropour rate, the model will be trained with all the nuerons. Final evaluation should use all nuerons we train for a better performance(the h in the equation, without multiplying scalar γ)

3 Problem 3

a

$[ROOT, presented, my]$,$[findings, at, the, NLP, conference]$,NOTHING,$SHIFT$.
$[ROOT, presented, my, findings]$,$[at, the, NLP]$,NOTHING,$SHIFT$.
$[ROOT, presented, findings]$,$[at, the, NLP]$,findings→I,$LEFT-ARC$.
$[ROOT, presented]$,$[at, the, NLP]$,presented→findings,$RIGHT-ARC$.
$[ROOT, presented, at]$,$[the, NLP.conference]$,NOTHING,$SHIFT$.
$[ROOT, presented, at, the]$,$[NLP, conference]$,NOTHING,$SHIFT$.
$[ROOT, presented, at, the, NLP]$,$[conference]$,NOTHING,$SHIFT$.
$[ROOT, presented, at, the, NLP, conference]$,$\emptyset$,NOTHING,$SHIFT$.
$[ROOT, presented, at, the, conference]$,$\emptyset$,conference→NLP,$LEFT-ARC$.
$[ROOT, presented, at, conference]$,$\emptyset$,conference→the,$LEFT-ARC$.
$[ROOT, presented, conference]$,$\emptyset$,conference→at,$LEFT-ARC$.
$[ROOT, presented]$,$\emptyset$,presented→conference,$RIGHT-ARC$.
$[ROOT]$,$\emptyset$,ROOT→presented,$RIGHT-ARC$.

b

Every word will be parsed in (SHIFT) for one step, and parsed out(left-arc or right-arc) for one step. Thus the total step will be $2n$.

c,d

```
CODE IMPLEMENTATION
```

e

i.

$$h_i = \text{ReLU}\left(\sum_k x_k W_{k,i} + b_1^i\right)$$

$$\frac{\partial h_i}{\partial x_j} = W_{j,i} \cdot \mathbb{I}\left(\sum_k x_k W_{k,i} + b_1^i > 0\right)$$

ii.

$$\hat{y_c} = \text{softmax}(l_c)$$

$$\frac{\partial CE(y,\hat{y})}{\partial l_i} = \sum_j \frac{\partial CE(y,\hat{y})}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial l_i} = \frac{\partial CE(y,\hat{y})}{\partial \hat{y}_c} \cdot \frac{\partial \hat{y}_c}{\partial l_i}$$

$$= -\frac{y_c}{\hat{y}_c} \cdot \begin{cases} \frac{e^{l_c}}{\sum_j e^{l_j}} - \left(\frac{e^{l_c}}{\sum_j e^{l_j}}\right)^2 & i = c \\ -\frac{e^{l_c} e^{l_i}}{\left(\sum_j e^{l_j}\right)^2} & i \neq c \end{cases}$$

$$= y_c \cdot \begin{cases} \hat{y}_c - 1 & i = c \\ \hat{y}_c & i \neq c \end{cases}$$

iii. The Final train loss is 0.0664, and the UAS is 89.20.

f

`USELESS, SKIPPED`

g

Firstly, the POS tag helps to determine the specific meaning of the word. Enhance the accuracy of the parser.
Secondly, It helps the model to generalize on unseen words.