CS224N: NATURAL LANGUAGE PROCESSING WITH DEEP LEARNING ASSIGNMENT #2

ANTHONY HO

1. (a) Please see the coding portion of the assignment.

   (b) Please see the coding portion of the assignment.

   (c) The purpose of the placeholder variables is to allocate storage for data/labels before building the computation graph. The feed dictionaries allows us to inject data/labels into the placeholders in a computation graph. Please see the coding portion of the assignment for implementation.

   (d) Please see the coding portion of the assignment.

   (e) When the model's `train_op` is called, (1) it creates a gradient descent optimizer; (2) it calls `add_loss_op` to compute the cross entropy loss based on the data, labels, and current values of the variables `W` and `b`; (3) it computes the gradients w.r.t the loss via automatic differentiation; (4) and at the end it updates the values of the variables `W` and `b` in the direction of the gradient and in proportion to the learning rate as defined in `Config`.
   Please see the coding portion of the assignment for implementation.

2. (a) The sequence of transitions are:

stack	buffer	new dependency	transition
[ROOT]	[I, parsed, this, sentence, correctly]		Initial Configuration
[ROOT, I]	[parsed, this, sentence, correctly]		`SHIFT`
[ROOT, I, parsed]	[this, sentence, correctly]		`SHIFT`
[ROOT, parsed]	[this, sentence, correctly]	parsed→I	`LEFT-ARC`
[ROOT, parsed, this]	[sentence, correctly]		`SHIFT`
[ROOT, parsed, this, sentence]	[correctly]		`SHIFT`
[ROOT, parsed, sentence]	[correctly]	sentence→this	`LEFT-ARC`
[ROOT, parsed]	[correctly]	parsed→sentence	`RIGHT-ARC`
[ROOT, parsed, correctly]	[]		`SHIFT`
[ROOT, parsed]	[]	parsed→correctly	`RIGHT-ARC`
[ROOT]	[]	ROOT→parsed	`RIGHT-ARC`

   (b) A sentence containing n words will be parsed in $2n$ steps, since each word must be first shifted from the buffer into the stack and then removed from the stack as a dependent of another item.

   (c) Please see the coding portion of the assignment.

   (d) Please see the coding portion of the assignment.

   (e) Please see the coding portion of the assignment.

   (f) For the following equation to be true:

$$\mathbb{E}_{p_{drop}}[\boldsymbol{h}_{drop}]_i = h_i$$

   γ must fulfill the following criteria:

$$\begin{aligned} \mathbb{E}_{p_{drop}}[\boldsymbol{h}_{drop}]_i &= h_i \\ \implies \gamma(1 - p_{drop})h_i &= h_i \\ \implies \gamma &= \frac{1}{1 - p_{drop}} \end{aligned}$$

(g) (i) By using $\boldsymbol{m}$ and a β_1 of 0.9, the new $\boldsymbol{\theta}$ would only be updated slightly towards the new direction and would be largely the same as the previous $\boldsymbol{\theta}$. It helps the updates in $\boldsymbol{\theta}$ to maintain a relatively steady trajectory and prevents the updates from "diffusing around" too much, and thus helps speeding up reaching the local optimum.

(ii) Since Adam divides the updates by $\sqrt{\boldsymbol{v}}$, the model parameters that have smaller magnitudes will get larger updates. This might help with combating the "saturated neurons" problem by giving a "boost" to the updates of the parameters that are "saturated" to get out of the "plateaus".

(h) Please see the coding portion of the assignment for implementation. The best UAS achieved on the dev set is 88.66 and the UAS achieved on the test is 89.17.

3. (a) (i) Let's denote k as the index for the target word. Since $\boldsymbol{y}^{(t)}$ is a one-hot vector:

$$\mathrm{PP}^{(t)}\left(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)}\right) = \frac{1}{\sum_{j=1}^{|V|} y_i^{(t)} \cdot \hat{y}_i^{(t)}} = \frac{1}{\hat{y}_k^{(t)}} \tag{1}$$

and:

$$CE(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)}) = -\sum_{j=1}^{|V|} y_i^{(t)} \log \hat{y}_i^{(t)} = -\log \hat{y}_k^{(t)} \tag{2}$$

Therefore, combining equation (1) and (2):

$$CE(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)}) = -\log \hat{y}_k^{(t)} = -\log \frac{1}{\mathrm{PP}^{(t)}(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)})} = \log \mathrm{PP}^{(t)}(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)}) \tag{3}$$

(ii) We can rewrite the log of geometric mean perplexity using equation (3):

$$\begin{aligned}
\log\left(\prod_{t=1}^{T} \mathrm{PP}^{(t)}(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)})\right)^{1/T} &= \frac{1}{T}\log\left(\prod_{t=1}^{T} \mathrm{PP}^{(t)}(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)})\right) \\
&= \frac{1}{T}\sum_{t=1}^{T} \log \mathrm{PP}^{(t)}(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)}) \\
&= \frac{1}{T}\sum_{t=1}^{T} CE(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)})
\end{aligned}$$

Since $\left(\prod_{t=1}^{T} \mathrm{PP}^{(t)}(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)})\right)^{1/T}$ is a positive function, minimizing $\log\left(\prod_{t=1}^{T} \mathrm{PP}^{(t)}(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)})\right)^{1/T}$ is equivalent to minimizing $\left(\prod_{t=1}^{T} \mathrm{PP}^{(t)}(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)})\right)^{1/T}$ itself. Therefore, minimizing the geometric mean perplexity $\left(\prod_{t=1}^{T} \mathrm{PP}^{(t)}(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)})\right)^{1/T}$ is equivalent to minimizing the arithmetic mean cross-entropy loss $\frac{1}{T}\sum_{t=1}^{T} CE(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)})$.

(iii) If $\bar{P}(\boldsymbol{x}_{\text{pred}}^{(t+1)} = \boldsymbol{x}^{(t+1)} | \boldsymbol{x}^{(t)}, \ldots, \boldsymbol{x}^{(1)}) = 1/|V|$, it means $\mathrm{PP}^{(t)}(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)}) = 1/(1/|V|) = |V|$. When $|V| = 10000$, the corresponding cross-entropy loss is $\log(10000) = 9.21$.

(b) Let's denote:

$$\begin{aligned}
\boldsymbol{z}^{(t)} &= \boldsymbol{W}_h \boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e \boldsymbol{e}^{(t)} + \boldsymbol{b}_1 \in \mathbb{R}^{D_h \times 1} \\
\boldsymbol{\theta}^{(t)} &= \boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2 \in \mathbb{R}^{|V| \times 1}
\end{aligned}$$

We can define and compute the values of the following error terms:

$$\begin{aligned}
\boldsymbol{\sigma}_1^{(t)} &= \frac{\partial J^{(t)}}{\partial \boldsymbol{\theta}^{(t)}} = \frac{\partial CE(\boldsymbol{y}^{(t)}, \hat{\boldsymbol{y}}^{(t)})}{\partial \boldsymbol{\theta}^{(t)}} = \hat{\boldsymbol{y}}^{(t)} - \boldsymbol{y}^{(t)} \in \mathbb{R}^{|V| \times 1} \\
\boldsymbol{\sigma}_2^{(t)} &= \frac{\partial J^{(t)}}{\partial \boldsymbol{z}^{(t)}} = \boldsymbol{\sigma}_1^{(t)} \frac{\partial \boldsymbol{\theta}^{(t)}}{\partial \boldsymbol{h}^{(t)}} \frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{z}^{(t)}} = \boldsymbol{U}^{\top}\left(\hat{\boldsymbol{y}}^{(t)} - \boldsymbol{y}^{(t)}\right) \circ \boldsymbol{h}^{(t)} \circ (1 - \boldsymbol{h}^{(t)}) \in \mathbb{R}^{D_h \times 1}
\end{aligned}$$

Therefore,

$$\frac{\partial J^{(t)}}{\partial \boldsymbol{U}} = \frac{\partial J^{(t)}}{\partial \boldsymbol{\theta}^{(t)}} \frac{\partial \boldsymbol{\theta}^{(t)}}{\partial \boldsymbol{U}} = \boldsymbol{\sigma}_1^{(t)} \left(\boldsymbol{h}^{(t)}\right)^\top \in \mathbb{R}^{|V| \times D_h}$$

$$\frac{\partial J^{(t)}}{\partial \boldsymbol{e}^{(t)}} = \frac{\partial J^{(t)}}{\partial \boldsymbol{z}^{(t)}} \frac{\partial \boldsymbol{z}^{(t)}}{\partial \boldsymbol{e}^{(t)}} = \boldsymbol{W}_e^\top \boldsymbol{\sigma}_2^{(t)} \in \mathbb{R}^{d \times 1}$$

$$\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{W}_e}\right|_{(t)} = \frac{\partial J^{(t)}}{\partial \boldsymbol{z}^{(t)}} \left.\frac{\partial \boldsymbol{z}^{(t)}}{\partial \boldsymbol{W}_e}\right|_{(t)} = \boldsymbol{\sigma}_2^{(t)} \left(\boldsymbol{e}^{(t)}\right)^\top \in \mathbb{R}^{D_h \times d}$$

$$\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{W}_h}\right|_{(t)} = \frac{\partial J^{(t)}}{\partial \boldsymbol{z}^{(t)}} \left.\frac{\partial \boldsymbol{z}^{(t)}}{\partial \boldsymbol{W}_h}\right|_{(t)} = \boldsymbol{\sigma}_2^{(t)} \left(\boldsymbol{h}^{(t-1)}\right)^\top \in \mathbb{R}^{D_h \times D_h}$$

$$\frac{\partial J^{(t)}}{\partial \boldsymbol{h}^{(t-1)}} = \frac{\partial J^{(t)}}{\partial \boldsymbol{z}^{(t)}} \frac{\partial \boldsymbol{z}^{(t)}}{\partial \boldsymbol{h}^{(t-1)}} = \boldsymbol{W}_h^\top \boldsymbol{\sigma}_2^{(t)} \in \mathbb{R}^{D_h \times 1}$$

(c) The “unrolled”network for 3 timesteps is shown in figure 1.

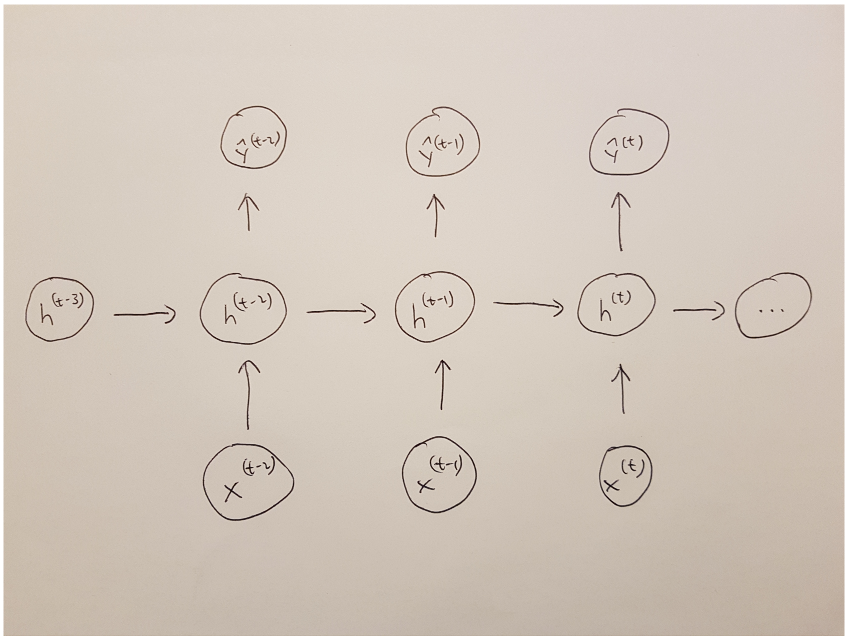

FIGURE 1. The “unrolled”network for 3 timesteps.

Let's compute the gradient of $J^{(t)}$ w.r.t. $\boldsymbol{z}^{(t-1)}$:

$$\frac{\partial J^{(t)}}{\partial \boldsymbol{z}^{(t-1)}} = \frac{\partial J^{(t)}}{\partial \boldsymbol{h}^{(t-1)}} \frac{\partial \boldsymbol{h}^{(t-1)}}{\partial \boldsymbol{z}^{(t-1)}} = \boldsymbol{\gamma}^{(t-1)} \circ \boldsymbol{h}^{(t-1)} \circ (1 - \boldsymbol{h}^{(t-1)}) \in \mathbb{R}^{D_h \times 1}$$

Therefore,

$$\frac{\partial J^{(t)}}{\partial \boldsymbol{e}^{(t-1)}} = \frac{\partial J^{(t)}}{\partial \boldsymbol{z}^{(t-1)}} \frac{\partial \boldsymbol{z}^{(t-1)}}{\partial \boldsymbol{e}^{(t-1)}} = \boldsymbol{W}_e^\top \left(\boldsymbol{\gamma}^{(t-1)} \circ \boldsymbol{h}^{(t-1)} \circ (1 - \boldsymbol{h}^{(t-1)})\right) \in \mathbb{R}^{d \times 1}$$

$$\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{W}_e}\right|_{(t-1)} = \frac{\partial J^{(t)}}{\partial \boldsymbol{z}^{(t-1)}} \left.\frac{\partial \boldsymbol{z}^{(t-1)}}{\partial \boldsymbol{W}_e}\right|_{(t-1)} = \left(\boldsymbol{\gamma}^{(t-1)} \circ \boldsymbol{h}^{(t-1)} \circ (1 - \boldsymbol{h}^{(t-1)})\right) \left(\boldsymbol{e}^{(t-1)}\right)^\top \in \mathbb{R}^{D_h \times d}$$

$$\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{W}_h}\right|_{(t-1)} = \frac{\partial J^{(t)}}{\partial \boldsymbol{z}^{(t-1)}} \left.\frac{\partial \boldsymbol{z}^{(t-1)}}{\partial \boldsymbol{W}_h}\right|_{(t-1)} = \left(\boldsymbol{\gamma}^{(t-1)} \circ \boldsymbol{h}^{(t-1)} \circ (1 - \boldsymbol{h}^{(t-1)})\right) \left(\boldsymbol{h}^{(t-2)}\right)^\top \in \mathbb{R}^{D_h \times D_h}$$

(d) To perform backpropagation for a single timestep, we will have to compute $\frac{\partial J^{(t)}}{\partial \boldsymbol{U}}$, $\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{W}_e}\right|_{(t)}$, $\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{W}_h}\right|_{(t)}$, $\frac{\partial J^{(t)}}{\partial \boldsymbol{b}_1}$, and $\frac{\partial J^{(t)}}{\partial \boldsymbol{b}_2}$. Since $\frac{\partial J^{(t)}}{\partial \boldsymbol{b}_1} = \boldsymbol{\sigma}_2^{(t)}$ and $\frac{\partial J^{(t)}}{\partial \boldsymbol{b}_2} = \boldsymbol{\sigma}_1^{(t)}$, their computations are implicit to the computations of $\frac{\partial J^{(t)}}{\partial \boldsymbol{U}}$, $\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{W}_e}\right|_{(t)}$, and $\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{W}_h}\right|_{(t)}$ and thus won't contribute to the overall time complexity. The rest of the terms that involve matrix multiplications are as follows:

Term	Matrix multiplication	Time complexity
$\boldsymbol{\sigma}_2^{(t)}$	$\boldsymbol{U}^\top \left(\hat{\boldsymbol{y}}^{(t)} - \boldsymbol{y}^{(t)}\right)$	$\mathcal{O}(D_h\|V\|)$
$\frac{\partial J^{(t)}}{\partial \boldsymbol{U}}$	$\boldsymbol{\sigma}_1^{(t)} \left(\boldsymbol{h}^{(t)}\right)^\top$	$\mathcal{O}(D_h\|V\|)$
$\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{W}_e}\right\|_{(t)}$	$\boldsymbol{\sigma}_2^{(t)} \left(\boldsymbol{e}^{(t)}\right)^\top$	$\mathcal{O}(D_h d)$
$\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{W}_h}\right\|_{(t)}$	$\boldsymbol{\sigma}_2^{(t)} \left(\boldsymbol{h}^{(t-1)}\right)^\top$	$\mathcal{O}(D_h^2)$

Therefore, given $\boldsymbol{h}^{(t-1)}$, the time complexity of performing backpropagation for a single timestep is:

$$\mathcal{O}(D_h|V| + D_h d + D_h^2)$$

(e) Since we can store and reuse $\boldsymbol{\gamma}^{(t-i)}$ at each timestep, we will just have to repeat the single timestep in part (d) by T times. Therefore, the time complexity of computing the gradient of the loss w.r.t. the model parameters across the entire sequence is:

$$\mathcal{O}(T(D_h|V| + D_h d + D_h^2))$$

(f)