Assignment 2 writeup

Thomas Lu

1 Problem 1

(c) Placeholders are nodes in a TF computation graph whose values are populated at runtime. These nodes are often used to populate training data. Feed dictionaries, meanwhile, are what actually do the populating at runtime; they specify at runtime a mapping of placeholder nodes to actual values.

(e) When `train_op` is called, we compute during forward propagation (for a batch) predictions $\hat{y} = \text{softmax}(xW + b)$ and the loss $J = CE(y, \hat{y})$, where CE denotes cross-entropy loss. Then we compute during backpropagation the partial gradients $\partial J/\partial W$ and $\partial J/\partial b$ and add a negative multiple of these gradients to our variables W and b.

2 Problem 2

(a)

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 written sentence of n words will require $2n$ steps: each word requires one step to move it from the buffer to the stack, and one to move it from the stack to the dependency tree.

(f) We have

$$h_i = \mathbb{E}_{drop}[h_{drop}]_i = \gamma(1 - p_{drop})h_i,$$

so $\gamma = 1/(1 - p_{drop})$.

(g)

- (i) Momentum basically slows the rate at which we adjust our updates: instead of immediately updating using the current gradient, we instead continue mostly going in the direction of our previous momentum and only assign a partial influence to the current gradient on our next update. This can help us avoid large "bad" updates when we see a particularly anomalous batch of training data or when we hit a particularly steep gradient wall.

- (ii) Adam amplifies the movement of parameters with small gradient contributions and reduces that of parameters with large gradient contributions. This can help in cases where we might have large almost-flat regions and small steep walls in our gradient function.

(h) The best dev UAS was 88.71, and the tes UAS was 89.19.

3 Problem 3

(a)

(i) Suppose that $y^{(t)} = w \in V$. We have

$$\begin{aligned} CE^{(t)}\left(y^{(t)}, \hat{y}^{(t)}\right) &= -\sum_{j=1}^{|V|} y_j^{(t)} \log \hat{y}_j^{(t)} = -\log \hat{y}_w^{(t)} \\ PP^{(t)}\left(y^{(t)}, \hat{y}^{(t)}\right) &= \frac{1}{\sum_{j=1}^{|V|} y_j^{(t)} \hat{y}_j^{(t)}} = \frac{1}{\hat{y}_w^{(t)}} \\ &= e^{CE^{(t)}\left(y^{(t)}, \hat{y}^{(t)}\right)} \end{aligned}$$

(ii) We have

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

,

from which it is clear that the geometric mean of perplexity and the arithmetic mean of cross-entropy are equivalent objectives.

(iii) The perplexity would be $1/\hat{y}_w^{(t)} = 1/(1/|V|) = |V|$, where w denotes the correct word. For $V = 10000$, the cross-entropy is

$$\log PP\left(y^{(t)}, \hat{y}^{(t)}\right) = \log 10000 \approx 9.2103.$$

(b)

Let $\theta^{(t)} = Uh^{(t)} + b_2$, so that $\hat{y}^{(t)} = \text{softmax}(\theta^{(t)})$. Suppose that $y^{(t)}$ is one-hot at w_t. We then have

$$\begin{aligned} \frac{\partial J^{(t)}}{\partial \theta^{(t)}} &= \frac{\partial}{\partial \theta_j^{(t)}} - \log \frac{e^{\theta_{w_t}^{(t)}}}{\sum_{j=1}^{|V|} e_j^{\theta^{(t)}}} \\ &= -\left(\frac{\partial}{\partial \theta^{(t)}} \log e^{\theta_{w_t}^{(t)}} - \frac{\partial}{\partial \theta^{(t)}} \log \left(\sum_{j=1}^{|V|} e^{\theta_j^{(t)}}\right)\right) \\ &= -\left(\frac{\partial}{\partial \theta^{(t)}} \theta_{w_t}^{(t)} - \frac{\frac{\partial}{\partial \theta^{(t)}} \sum_{j=1}^{|V|} e^{\theta_j^{(t)}}}{\sum_{j=1}^{|V|} e^{\theta_j^{(t)}}}\right) \\ &= \text{softmax}(\theta^{(t)}) - y^{(t)}. \end{aligned}$$

Let

$$\delta_1^{(t)} = \frac{\partial J^{(t)}}{\partial \theta^{(t)}} = \text{softmax}(\theta^{(t)}) - y^{(t)}.$$

Then

$$\frac{\partial J^{(t)}}{\partial U} = \frac{\partial J^{(t)}}{\partial \theta^{(t)}} \frac{\partial \theta^{(t)}}{\partial U} = \delta_1^{(t)} \frac{\partial \theta^{(t)}}{\partial U}.$$

We have

$$\frac{\partial \theta^{(t)}}{\partial U_{ij}} = h_j^{(t)} 1_i,$$

where 1_i denotes the one-hot vector (of equal dimension as $\theta^{(t)}$) with 1 at position i. This implies

$$\begin{aligned}
\frac{\partial J^{(t)}}{\partial U_{ij}} &= \left(\frac{\partial J^{(t)}}{\partial \theta^{(t)}}\right)^T \frac{\partial \theta^{(t)}}{\partial U_{ij}} \\
&= h_j^{(t)} \left(\text{softmax}(\theta^{(t)}) - y^{(t)}\right)_i \\
\Rightarrow \frac{\partial J^{(t)}}{\partial U} &= \left(\text{softmax}(\theta^{(t)}) - y^{(t)}\right) \left(h^{(t)}\right)^T \\
&= \delta_1^{(t)} \left(h^{(t)}\right)^T.
\end{aligned}$$

Now let $z^{(t)} = W_h h^{(t-1)} + W_e e^{(t)} + b_1$, so that $h^{(t)} = \sigma(z^{(t)})$. It is easy to see that

$$\begin{aligned}
\frac{\partial J^{(t)}}{\partial e^{(t)}} &= \left(\frac{\partial z^{(t)}}{\partial e^{(t)}}\right)^T \frac{\partial J^{(t)}}{\partial z^{(t)}} = W_e^T \delta_2^{(t)} \\
\left.\frac{\partial J^{(t)}}{\partial W_e}\right|_{(t)} &= \left(\left.\frac{\partial z^{(t)}}{\partial W_e}\right|_{(t)}\right)^T \frac{\partial J^{(t)}}{\partial z^{(t)}} = \delta_2^{(t)} (e^{(t)})^T \\
\left.\frac{\partial J^{(t)}}{\partial W_h}\right|_{(t)} &= \left(\left.\frac{\partial z^{(t)}}{\partial W_h}\right|_{(t)}\right)^T \frac{\partial J^{(t)}}{\partial z^{(t)}} = \delta_2^{(t)} (h^{(t-1)})^T \\
\frac{\partial J^{(t)}}{\partial h^{(t-1)}} &= \left(\frac{\partial z^{(t)}}{\partial h^{(t-1)}}\right)^T \frac{\partial J^{(t)}}{\partial z^{(t)}} = W_h^T \delta_2^{(t)}
\end{aligned}$$

where

$$\delta_2^{(t)} = \frac{\partial J^{(t)}}{\partial z^{(t)}}$$

and

$$\frac{\partial v}{\partial A}$$

for a vector v and matrix A denotes a vector v' of matrices $v_1', v_2', \dots, v_n'$ with

$$v_k' = \frac{\partial v_k}{\partial A}.$$

It suffices to compute $\delta_2^{(t)}$. We have

$$\begin{aligned}
\delta_2^{(t)} = \frac{\partial J^{(t)}}{\partial z^{(t)}} &= \left(\frac{\partial \theta^{(t)}}{\partial h^{(t)}} \frac{\partial h^{(t)}}{\partial z^{(t)}}\right)^T \frac{\partial J^{(t)}}{\partial \theta^{(t)}} \\
&= \left(U \text{diag}\left(\sigma(z^{(t)} \circ (1 - \sigma(z^{(t)})))\right)\right)^T \delta_1^{(t)} \\
&= h^{(t)} \circ \left(1 - h^{(t)}\right) \circ \left(U^T \delta_1^{(t)}\right)
\end{aligned}$$

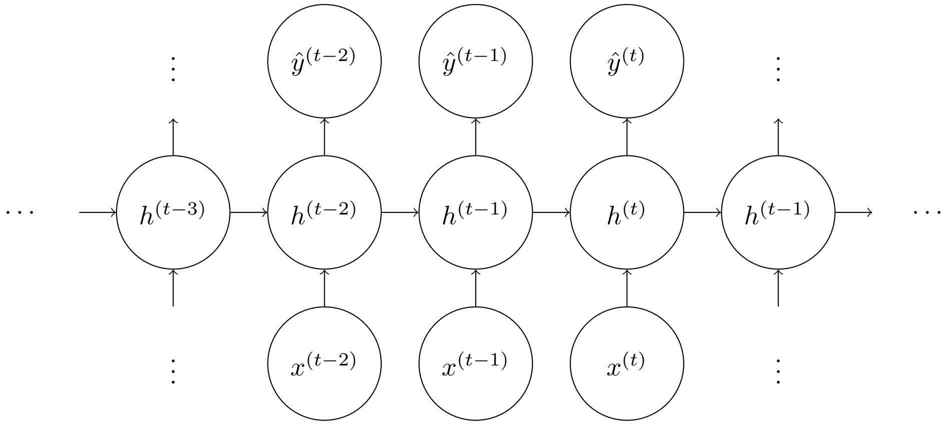

Figure 1: The RNN unrolled to 3 steps.

(c) Figure 1 depicts the unrolled RNN.

We have

$$\begin{aligned}
\frac{\partial J^{(t)}}{\partial e^{(t-1)}} &= \left(\frac{\partial h^{(t-1)}}{\partial e^{(t-1)}}\right)^T \frac{\partial J^{(t)}}{\partial h^{(t-1)}} \\
&= \left(\frac{\partial h^{(t-1)}}{\partial z^{(t-1)}}\frac{\partial z^{(t-1)}}{\partial e^{(t-1)}}\right)^T \gamma^{(t-1)} \\
&= W_e^T \left(h^{(t-1)} \circ (1 - h^{(t-1)}) \circ \gamma^{(t-1)}\right),
\end{aligned}$$

where $\gamma^{(t-1)} = \partial J^{(t)} / \partial h^{(t-1)}$. Similarly,

$$\begin{aligned}
\left.\frac{\partial J^{(t)}}{\partial W_e}\right|_{(t-1)} &= \left(\left.\frac{\partial z^{(t-1)}}{\partial W_e}\right|_{(t-1)}\right)^T \frac{\partial J^{(t)}}{\partial z^{(t-1)}} \\
&= \left(\left.\frac{\partial z^{(t-1)}}{\partial W_e}\right|_{(t-1)}\right)^T \left(h^{(t-1)} \circ (1 - h^{(t-1)}) \circ \gamma^{(t-1)}\right) \\
&= \left(h^{(t-1)} \circ (1 - h^{(t-1)}) \circ \gamma^{(t-1)}\right) (e^{(t-1)})^T \\
\left.\frac{\partial J^{(t)}}{\partial W_h}\right|_{(t-1)} &= \left(\left.\frac{\partial z^{(t-1)}}{\partial W_h}\right|_{(t-1)}\right)^T \frac{\partial J^{(t)}}{\partial z^{(t-1)}} \\
&= \left(\left.\frac{\partial z^{(t-1)}}{\partial W_h}\right|_{(t-1)}\right)^T \left(h^{(t-1)} \circ (1 - h^{(t-1)}) \circ \gamma^{(t-1)}\right) \\
&= \left(h^{(t-1)} \circ (1 - h^{(t-1)}) \circ \gamma^{(t-1)}\right) (h^{(t-2)})^T.
\end{aligned}$$

(d)

- Computing $\partial J^{(t)} / \partial U$ requires constructing a $|V| \times D_h$ matrix, so will require $|V| D_h$ operations.
- Computing δ_2, which is needed in the computations of $\partial J^{(t)} / \partial e^{(t)}$, $\partial J^{(t)} / \partial W_e|_{(t)}$, $\partial J^{(t)} / \partial W_h|_{(t)}$, and

$\partial J^{(t)}/\partial h^{(t-1)}$, requires multiplying a $D_h \times |V|$ matrix by a vector of length $|V|$, so it requires $|V|D_h$ operations. δ_2 is a vector of length D_h.

- Computing $\partial J^{(t)}/\partial e^{(t)}$ requires multiplying a $d \times D_h$ matrix by a vector of length D_h, so it takes dD_h operations.
- Computing $\partial J^{(t)}/\partial W_e|_{(t)}$ requires constructing a $D_h \times d$ matrix, so requires dD_h operations.
- Computing $\partial J^{(t)}/\partial W_h|_{(t)}$ requires constructing a $D_h \times D_h$ matrix, so requires D_h^2 operations.
- Computing $\partial J^{(t)}/\partial h^{(t-1)}$ requires multiplying a $D_h \times D_h$ matrix by a vector of length D_h, so requires D_h^2 operations.

Thus computing the gradients for one time step requires $O(|V|D_h + dD_h + D_h^2)$ operations.

(e)

If we compute the gradients for the time steps in reverse chronological order, we only have to pass error signals from one time step to the previous one once for each time step. Then computing the gradients for all time steps for a sequence of words of length T requires $O(T(|V|D_h + dD_h + D_h^2))$ operations.

(f)

The largest term in the big-O estimate is likely to be $|V|D_h$, which comes from the output layer of the RNN.