CS224n HW2

Xinyu Tan

August 27, 2017

1 Neural Transition-based Dependency Parsing

(a)

stack	buffer	new dependency	transition
[root]	[I, parsed, this, sentence, correctly]		Initial Configuratio
[root, I]	[parsed, this, sentence, correctly]		SHIFT
[root, I, parsed]	[this, sentence, correctly]		SHIFT
[root, parsed]	[this, sentence, correctly]	parsed $\rightarrow$ I	LEFT-ARC
[root, parsed, this]	[sentence, correctly]		SHIFT
[root, parsed, this, sentence]	[correctly]		SHIFT
[root, parsed, sentence]	[correctly]	sentence $\rightarrow$ this	LEFT-ARC
[root, parsed]	[correctly]	parsed $\rightarrow$ sentence	RIGHT-ARC
[root, parsed, correctly]	[]		SHIFT
[root, parsed]	[]	parsed $\rightarrow$ correctly	RIGHT-ARC
[root]	[]	root $\rightarrow$ parsed	RIGHT-ARC

(b)

The sentence will be parsed in $2n$ times. Each word will be pushed into stack once, and each word only depends on one other word. Therefore, the process is in $O(n)$ time complexity.

(f)

We need to satisfy: $\mathbb{E}_{p_{\text{drop}}}[\boldsymbol{h}_{\text{drop}}]_i = \gamma(1 - p_{\text{drop}})\boldsymbol{h}_i = \boldsymbol{h}_i$, then we have:

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

(g)

(i)

(ii)

2 Recurrant neural networks: Language Modeling

(a)

Perplexity:

$$PP^{(t)}\left(y^{(t)}, \hat{y}^{(t)}\right) = \frac{1}{y_k^{(t)}\hat{y}_k^{(t)}} = \frac{1}{\hat{y}_k^{(t)}}$$

Cross-entropy loss:

$$J^{(t)}(\theta) = -y_k^{(t)} \log \hat{y}_k^{(t)} = -\log \hat{y}_k^{(t)}$$

Then, it is easy to derive that

$$PP^{(t)}\left(y^{(t)}, \hat{y}^{(t)}\right) = e^{J^{(t)}(\theta)}$$

Therefore, minimizing perplexity equals to minimizing the cross-entropy.

For a vocabulary of $|V| = 10000$ words, if the model is completely random, then the perplexity will be 10000, and then the cross entropy will be $\log 10000 = 9.21$.

(b)

The derivatives:

$$\frac{\partial J^{(t)}}{\partial \boldsymbol{b}_2} = \frac{\partial J^{(t)}}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{b}_2} = \hat{\boldsymbol{y}}^{(t)} - \boldsymbol{y}^{(t)}$$

Since $\boldsymbol{\theta} = \boldsymbol{h}^{(t)}U + \boldsymbol{b}_2$, every θ_i depends on every h_j, i.e.,

$$\frac{\partial J}{\partial h_j^{(t)}} = \sum_{i=1}^{|V|} \frac{\partial J}{\partial \theta_i} \frac{\partial \theta_i}{\partial h_j^{(t)}} = \sum_{i=1}^{|V|} (\hat{y}_i - y_i) U_{ji}$$

Hence, we have:

$$\boldsymbol{\delta^{(t)}} = \frac{\partial J^{(t)}}{\partial \boldsymbol{h}^{(t)}} = \frac{\partial J^{(t)}}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{h}^{(t)}} = (\hat{\boldsymbol{y}} - \boldsymbol{y}) \boldsymbol{U}^T$$

Often times, when it's a lot of matrix multiplications, it's very hard to know when to use matrix multiplication, when to use $\odot$ (element multiplication). I find it helpful to write down the element-wise formula, then it's clearer which affects which.

Next, let's calculate $\frac{\partial J^{(t)}}{\partial \boldsymbol{H}}$. We have $\boldsymbol{h}^{(t)} = \sigma(\boldsymbol{h}^{(t-1)}\boldsymbol{H} + \boldsymbol{e}^{(t)}\boldsymbol{I} + \boldsymbol{b}_1)$, element wise:

$$h_j^{(t)} = \sigma\left(\sum_{k=1}^{D_n} h_k^{(t-1)} H_{kj} + \sum_{k=1}^{d} e_k^{(t)} I_{kj} + b_{1j}\right)$$

Conceptually,

$$\frac{\partial J^{(t)}}{\partial H_{kj}} = \frac{\partial J^{(t)}}{\partial h_j^{(t)}} \frac{\partial h_j^{(t)}}{\partial H_{kj}}$$

Therefore,

$$\frac{\partial J^{(t)}}{\partial \boldsymbol{H}} = \boldsymbol{h}^{(t-1)^T} \cdot \left(\boldsymbol{\delta^{(t)}} \odot \boldsymbol{h}^{(t)} \odot \left(1 - \boldsymbol{h}^{(t)}\right)\right)$$

Similarly, we have

$$\frac{\partial J^{(t)}}{\partial \boldsymbol{I}} = \boldsymbol{e}^{(t)^T} \cdot \left(\boldsymbol{\delta^{(t)}} \odot \boldsymbol{h}^{(t)} \odot \left(1 - \boldsymbol{h}^{(t)}\right)\right)$$

$$\frac{\partial J^{(t)}}{\partial \boldsymbol{L}_{x^{(t)}}} = \frac{\partial J^{(t)}}{\partial \boldsymbol{e}^{(t)}}^T = \boldsymbol{\delta^{(t)}} \odot \boldsymbol{h}^{(t)} \odot \left(1 - \boldsymbol{h}^{(t)}\right) \cdot \boldsymbol{I}^{T^T} = \boldsymbol{I} \cdot \left(\boldsymbol{\delta^{(t)}} \odot \boldsymbol{h}^{(t)} \odot \left(1 - \boldsymbol{h}^{(t)}\right)\right)^T$$

Additionally,

$$\boldsymbol{\delta^{(t-1)}} = \frac{\partial J^{(t)}}{\partial \boldsymbol{h}^{(t-1)}} = \sum_{k=1}^{D_h} \frac{\partial J^{(t)}}{\partial h_k^{(t)}} \frac{\partial h_k^{(t)}}{\partial \boldsymbol{h}^{(t-1)}} = \left(\boldsymbol{\delta}^{(t)} \odot \boldsymbol{h}^{(t)} \odot \left(1 - \boldsymbol{h}^{(t)}\right)\right) \boldsymbol{H}^T$$

(c)

$$\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{H}}\right|_{(t-1)} = \frac{\partial J^{(t)}}{\partial \boldsymbol{h}^{(t-1)}} \frac{\partial \boldsymbol{h}^{(t-1)}}{\partial \boldsymbol{H}} = \boldsymbol{h}^{(t-2)^T} \cdot \left(\boldsymbol{\delta^{(t-1)}} \odot \boldsymbol{h}^{(t-1)} \odot \left(1 - \boldsymbol{h}^{(t-1)}\right)\right)$$

$$\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{I}}\right|_{(t-1)} = \frac{\partial J^{(t)}}{\partial \boldsymbol{h}^{(t-1)}} \frac{\partial \boldsymbol{h}^{(t-1)}}{\partial \boldsymbol{I}} = \boldsymbol{e}^{(t-1)^T} \cdot \left(\boldsymbol{\delta^{(t-1)}} \odot \boldsymbol{h}^{(t-1)} \odot \left(1 - \boldsymbol{h}^{(t-1)}\right)\right)$$

$$\left.\frac{\partial J^{(t)}}{\partial \boldsymbol{L}_{x^{t-1}}}\right|_{(t-1)} = \left.\frac{\partial J^{(t)}}{\partial \boldsymbol{e}^{(t-1)}}\right|_{(t-1)} = \frac{\partial J^{(t)}}{\partial \boldsymbol{h}^{(t-1)}} \frac{\partial \boldsymbol{h}^{(t-1)}}{\partial \boldsymbol{e}^{(t-1)}} = \left(\boldsymbol{\delta^{(t-1)}} \odot \boldsymbol{h}^{(t-1)} \odot \left(1 - \boldsymbol{h}^{(t-1)}\right)\right) \cdot \boldsymbol{I}^T$$

(d)

Given $\boldsymbol{h}^{(t-1)}$, forward pass requires to compute: $\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{h}^{(t-1)}\boldsymbol{H} + \boldsymbol{e}^{(t)}\boldsymbol{I} + b_1\right)$ and $\hat{y}^t = \text{softmax}\left(\boldsymbol{h}^{(t)}\boldsymbol{U} + b_2\right)$. We know that for a $m \times n$ and $n \times l$ matrix multiplication, the complexity is $O(mnl)$. Therefore, the forward pass complexity is:

$$O(D_h^2 + dD_h + D_h + D_h|V| + |V|) \approx O(D_h^2 + dD_h + D_h|V|)$$

Similarly, the backward pass complexity (from $\boldsymbol{\delta}^{(t)}$) is approximately:

$$O(D_h^2 + dD_h + D_h|V|)$$

Due to that $|V| >> D_h$ or d, therefore, the major time consuming step is softmax ($O(D_h|V|)$)