CS 224n Assignment #3: Dependency Parsing

Intae Jun

1 Machine Learning & Neural Networks (8 points)

(a) i. $\boldsymbol{m}$ accumulates the gradients over time, giving more weight on gradients accumulated over multiple steps (the degree of weighting depends on the setting of β_1). This accumulation helps to sommoth out the updates, reducing the variance in the direction of parameter changes and leading to more stable convergence.

ii. $\boldsymbol{v}$ is the moving average of the squared gradients. It captures the magnitude of the gradients over time, giving more weight on accumulated magnitude (if β_2 is set as > 0.5). Dividing by $\sqrt{\boldsymbol{v}_{t+1}}$ serves to normalize. If a parameter has had consistently large gradients, i.e., the element of $\boldsymbol{v}_{t+1}$ is quite large, then its reciprocal will be smaller than others, which leads to smaller update of that element. Conversely, for parameters that have had consistently small gradients, the division by $\sqrt{\boldsymbol{v}_{t+1}}$ will lead to larger updates. This normalization would help the optimizer to stabilize updates by scaling and prevent it from slower convergence of some parameters.

(b) i. For ith element, $\mathbb{E}_{p_{\text{drop}}}[\boldsymbol{h}_{\text{drop}}]_i = \gamma(0 \cdot p_{\text{drop}} + h_i \cdot (1 - p_{\text{drop}})) \equiv h_i$. Hence γ must be $1/(1 - p_{\text{drop}})$.

ii. During training, dropout helps prevent overfitting by ensuring the model doesn't overly depend on specific neurons, promoting better generalization. However, during evaluation, dropout is not applied so that all neurons are active, ensuring consistent and reliable predictions. This allows the model to fully leverage the learned features and produce stable outputs without the randomness introduced by dropout.

2 Neural Transition-Based Dependency Parsing (44 points)

(i) The sequence of transitions needed for parsing *"Today I parsed a sentence"*. is as follows:

Stack	Buffer	New dependency	Transition
[Root]	[Today, I, parsed, a, sentence]		Initial Configuration
[Root, Today]	[I, parsed, a, sentence]		SHIFT
[Root, Today, I]	[parsed, a, sentence]		SHIFT
[Root, Today, I, parsed]	[a, sentence]		SHIFT
[Root, Today, parsed]	[a, sentence]	parsed→I	LEFT-ARC
[Root, parsed]	[a, sentence]	parsed→Today	LEFT-ARC
[Root, parsed, a]	[sentence]		SHIFT
[Root, parsed, a, sentence]	[]		SHIFT
[Root, parsed, sentence]	[]	sentence←a	LEFT-ARC
[Root, parsed]	[]	parsed→	RIGHT-ARC
[Root]	[]	Root→parsed	RIGHT-ARC

(ii) $2n$ times of transitions are needed: n times for shifting each word from a butter to a stack, n times for removing each word except "Root" from a stack based on dependency.