CS224N-2019 Assignment 3 Written Part

Machine Learning & Neural Networks

(a) Adam Optimizer

i. m consists of two parts. We could see the first part $\beta_1 m$ as a push to maintain its origin trend while the other $(1-\beta_1)\delta$ as a push to a variance. β_1 is often set to 0.9, which implies that m tends to keep its direction preventing from overshooting. Hence, The low variance could help J reduce the vibration on the vertical direction while downward to the convergence point.

ii. Weights that receive high gradients will have their effective learning rate reduced. And Weights that receive small / infrequent updates will have effective learning rate increased.

(b) Dropout

i. It's obvious that we have $\mathbb{E}[X]_i = \mathbb{E}[X_i]$. Hence,

$$\begin{aligned}
\mathbb{E}_{p_{drop}}[h_{drop}]_i &= \mathbb{E}_{p_{drop}}[\gamma d \cdot h]_i \\
&= \mathbb{E}_{p_{drop}}[\gamma d_i \cdot h_i] \\
&= \gamma \mathbb{E}_{p_{drop}}[d_i \cdot h_i] \\
&= \gamma[p_{drop} \times 0 + (1-p_{drop}) \times 1]h_i \\
&= \gamma(1-p_{drop})h_i
\end{aligned}$$

According to the topic, we have

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

ii. Dropout is design for preventing overfitting and improving robustness of model. However, applying dropout in evaluation actually adds occasionality in it and decrease model's reliability in turn.

Neural Transition-Based Dependency Parsing

(a) Transition process is as following:

Stack	Buffer	New Dependency	Transition
[ROOT]	[I, parsed, this, sentence, correctly]		Initial Conguration
[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

Stack	Buffer	New Dependency	Transition
[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) $2n$. For each word in sentence, it takes one step to shift word from buffer to stack and another one to pop from stack. Hence the total step is double the number of words.

(f) Dependency parse correction is as following:

	Error Type	Incorrect Dependency	Correct Dependency
i.	Verb Phrase Attachment Error	wedding→fearing	heading→fearing
ii.	Coordination Attachment Error	rescue→and	rush→and
iii.	Prepositional Phrase Attachment Error:	named→Midland	guy→Midland
iv.	Modier Attachment Error	elements→most	crucial→most