A3 CS224N

mhmd.sl.elhady

August 2019

1 Machine Learning and Neural Networks

1.1 A

i. local calculation of gradient descent can be noisy, m stores information about previous gradient steps so that next gradient step will be dependent on both local calculation of gradients and past gradients. This smoothing reduce variance so convergence is faster.
ii.

- Adam use the adaptive learning rate trick so each batch can be updated according to its gradient values.
- Weights that receive high gradient values will reveive most updates
- Weights (infrequent) will receive the lowest gradient values.

1.2 B

i. To be solved ii. At test time we want the network to see all of the input so that the output activations can represnt their equivalent on training time. However; on training time we want to prevent network too see some activations too frequent.

2 Neural Transition-Based Dependency Parsing

2.1 A

Stack	Buffer	New Dependency	Config
[ROOT]	[I,Parsed,this,sentence,correctly]		init
[ROOT,I]	[parsed,this, sentence, correctly]		SHIFT
[ROOT,I,parsed]	[this,sentence,correctly]		SHIFT
[ROOT,parsed]	[this,sentence,correctly]	I <—- parsed	LEFT-ARC
[ROOT,parsed,this]	[sentence,correctly]		SHIFT
[ROOT,parsed,this,sentence]	[correctly]	this <—- sentence	SHIFT
[ROOT,parsed,sentence]	[correctly]		LEFT-ARC
[ROOT,parsed]	[correctly]	parsed —->sentence	RIGHT-ARC
[ROOT,parsed,correctly]	[]		SHIFT
[ROOT,parsed]	[]	parsed —>correctly	RIGHT-ARC
[ROOT]	[]	ROOT —->parsed	RIGHT-ARC

2.2 B

Since its an acyclic graph (no word can be dependent of itself), we have total of N arc operations , in addition to N shift operations where N is the total number of tokens in buffer at initial configuration.

This gives us 2N in total