1. Machine Learning & Neural Networks

a) Adam optimization

i) By maintaining a sort of exponential smoothing, or rolling average, of the loss function's gradients, we effectively control a "memory bank" that tracks previous update steps and fuses them together to some degree, β1, with new gradient information. That allows smoother transitions.

ii) Because v is a rolling average of the magnitudes of gradients, parameters of the model with corresponding small gradients, i.e. small v entries, will get larger updates. We're cutting out a smaller amount from such parameters during an update because we divide them by their small gradient magnitude/norm.

b) Dropout

i) -

ii) The goal of dropout is to reduce overfitting, so the model performs better on more general datasets. Now, during evaluation we're concerned with how well the model handles unseen data. When we dropout units, we add noise to predictions and dampen accuracy. Thus, if we were to apply dropout during evaluation time, we would not be able to fairly assess the generalization power of the network.

2. Neural Transition-Based Dependency Parsing

a)

Stack	Buffer	New Dependency	Transition
(ROOT)	[I, parsed, this, sentence, correctly]		Initial Config
(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) Worst case: linear time O(N)