

CS224N: NATURAL LANGUAGE PROCESSING WITH DEEP LEARNING

ASSIGNMENT #2

ANTHONY HO

1. (a) Please see the coding portion of the assignment.
- (b) Please see the coding portion of the assignment.
- (c) The purpose of the placeholder variables is to allocate storage for data/labels before building the computation graph. The feed dictionaries allows us to inject data/labels into the placeholders in a computation graph.
Please see the coding portion of the assignment for implementation.
- (d) Please see the coding portion of the assignment.
- (e) When the model's `train_op` is called, (1) it creates a gradient descent optimizer; (2) it calls `add_loss_op` to compute the cross entropy loss based on the data, labels, and current values of the variables `W` and `b`; (3) it computes the gradients w.r.t the loss via automatic differentiation; (4) and at the end it updates the values of the variables `W` and `b` in the direction of the gradient and in proportion to the learning rate as defined in `Config`.
Please see the coding portion of the assignment for implementation.

2. (a) The sequence of transitions are:

stack	buffer	new dependency	transition
[ROOT]	[I, parsed, this, sentence, correctly]		Initial Configuration
[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) A sentence containing n words will be parsed in $2n$ steps, since each word must be first shifted from the buffer into the stack and then removed from the stack as a dependent of another item.
- (c) Please see the coding portion of the assignment.
- (d) Please see the coding portion of the assignment.
- (e)
- (f)
- (g)
- (h)
- (i)

3. (a)
(b)
(c)
(d)
(e)
(f)