Stanford CS224n HW A3

David Chen

August 2020

Written Questions

1. Machine Learning & Neural Networks

(a) i. If most of the SGD updates are heading in one direction, and one noisy update is in the opposite direction, vanilla SGD will have some zigzags. However, in momentum, a noisy update in the wrong direction will only act to slow down the movement in the overall correct direction, reducing the variance and oscillation from each individual update. This low variance means that SGD+Momentum more frequently takes steps in the correct direction, and additionally the momentum can help prevent SGD from getting stuck in saddle points.

ii. By dividing the update by the square root of the gradient magnitude rolling average, Adam effectively assigns lower learning rates to weights with high gradients (or frequent updates), and higher learning rates to weights with low gradients (or infrequent updates). Since learning rates are adapted to the parameters, the updates become more robust. Larger updates for small gradients can also help with escaping from places with low gradients.

(b) i. γ must be equal to p_{drop}, because we want the expected output of each layer to be the same. We don't want to set some layers to 0, and then pass a significantly smaller output to the next layer, since this would mess up at test time when we turn off dropout.

ii. Dropout during training helps prevent overfitting, since it prevents the network from memorizing training examples, and rather forces different parts of the network to learn the same things. During evaluation, however, we want to make use of everything the model has learned, so we use all the learned neurons.

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) A sentence containing n words will be parsed in 2n steps, since each word will be shifted onto the stack from the buffer and then mapped to a dependency.

(e) **Default Parameters:**

```
batch_size=1024, n_epochs=10, lr=0.0005)

Epoch 10 out of 10
Average Train Loss: 0.01810444533337085
Evaluating on dev set
- dev UAS: 87.89
================================================================================
TESTING
================================================================================
Restoring the best model weights found on the dev set
Final evaluation on test set
- test UAS: 88.78
Done!
```

Custom Parameters (slight hyperparameter tuning):

```
batch_size=512, n_epochs=10, lr=0.0005

Epoch 10 out of 10
Average Train Loss: 0.014971269847214367
Evaluating on dev set
- dev UAS: 87.83
================================================================================
TESTING
================================================================================
Restoring the best model weights found on the dev set
Final evaluation on test set
- test UAS: 89.09
Done!
```

(f) i. Verb Phrase Attachment Error:
Incorrect dependency: wedding → fearing
Correct dependency: heading → fearing

ii. Coordination Attachment Error:
Incorrect dependency: makes → rescue
Correct dependency: rush → rescue

iii. Prepositional Phrase Attachment Error:
Incorrect dependency: named → midland
Correct dependency: guy → midland

iv. Modifier Attachment Error:
Incorrect dependency: elements → most
Correct dependency: crucial → most