

Part 1

a)

- i) Momentum uses a moving average over the past gradients. So, if the gradient experiences a lot of change in one step, it wouldn't result in the weight update a lot. Momentum was designed to speed up learning in directions of low curvature, without becoming unstable in directions of high curvature.
- ii) The more the term m/v , the larger the update will be. So if m gets higher or v gets lower, the updates will be larger. m and v are moving averages of gradient and squared gradient. Near the minimum this term is near 0. Choosing right amount for β_1 and β_2 is important in updates.

It uses moving average of squared gradients to normalize the gradient. This normalization balances the step size (momentum), decreasing the step for large gradients to avoid exploding and increasing the step for small gradients to avoid vanishing.

b)

i)

$$\begin{aligned}
 E_{p_{\text{drop}}} [h_{\text{drop}}]_i &= p_{\text{drop}} \cdot h_{\text{drop}} + (1 - p_{\text{drop}}) \cdot h_{\text{drop}} \\
 &= p_{\text{drop}} \cdot (\delta \cdot 1 \odot h) + 0 = h_i \\
 \Rightarrow p_{\text{drop}} \cdot \delta \cdot h &= h_i \Rightarrow \delta = \frac{1}{p_{\text{drop}}}
 \end{aligned}$$

- ii) During training we use dropout to improve the model prediction by cutting off some of the neurons. But during evaluation we need to use all the neurons and their weights for a better prediction.

Part 2

a)

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)

We will need 'n' shifts to move all items from buffer to stack. Then there will be 'n-1' arcs between the words and one arc from ROOT. So, the sentence will be parsed in $n + n - 1 + 1 = 2n$ steps.

e)

As it is seen, the best UAS on dev set is 88.51 and on test set is 88.89.

```

Anaconda Prompt (anaconda)
1445850it [00:00, 32165563.10it/s]
- dev UAS: 87.17
New best dev UAS! Saving model.

Epoch 5 out of 10
100%|████████████████████████████████████████████████████████████████████████████████| 1848/1848 [01:58<00:00, 15.59it/s]
Average Train Loss: 0.08584392689434546
Evaluating on dev set
1445850it [00:00, 28852777.30it/s]
- dev UAS: 87.86
New best dev UAS! Saving model.

Epoch 6 out of 10
100%|████████████████████████████████████████████████████████████████████████████████| 1848/1848 [01:59<00:00, 15.42it/s]
Average Train Loss: 0.08036224598274667
Evaluating on dev set
1445850it [00:00, 26301603.59it/s]
- dev UAS: 88.17
New best dev UAS! Saving model.

Epoch 7 out of 10
100%|████████████████████████████████████████████████████████████████████████████████| 1848/1848 [01:56<00:00, 15.87it/s]
Average Train Loss: 0.07623679440637882
Evaluating on dev set
1445850it [00:00, 29281590.11it/s]
- dev UAS: 88.30
New best dev UAS! Saving model.

Epoch 8 out of 10
100%|████████████████████████████████████████████████████████████████████████████████| 1848/1848 [01:55<00:00, 15.98it/s]
Average Train Loss: 0.07275127800729819
Evaluating on dev set
1445850it [00:00, 23873452.64it/s]
- dev UAS: 88.39
New best dev UAS! Saving model.

Epoch 9 out of 10
100%|████████████████████████████████████████████████████████████████████████████████| 1848/1848 [01:58<00:00, 15.65it/s]
Average Train Loss: 0.0695593068356025
Evaluating on dev set
1445850it [00:00, 24180926.03it/s]
- dev UAS: 88.40
New best dev UAS! Saving model.

Epoch 10 out of 10
100%|████████████████████████████████████████████████████████████████████████████████| 1848/1848 [01:59<00:00, 15.52it/s]
Average Train Loss: 0.0662910147082238
Evaluating on dev set
1445850it [00:00, 29077169.34it/s]
- dev UAS: 88.51
New best dev UAS! Saving model.

=====
TESTING
=====
Restoring the best model weights found on the dev set
Final evaluation on test set
2919736it [00:00, 32256966.03it/s]
- test UAS: 88.89
Done!

```

f)

i) Error type: verb phrase attachment error

Incorrect dependency: wedding → fearing

Correct dependency: heading → fearing

ii) Error type: coordination attachment error

Incorrect dependency: makes → rescue

Correct dependency: rush → rescue

iii) Error type: prepositional phrase attachment error

Incorrect dependency: named → Midland

Correct dependency: guy → Midland

iv) Error type: modifier attachment error

Incorrect dependency: elements → most

Correct dependency: crucial → most