Assignment 4 Writeup

DO NOT TAG

Name: Jing Yu
GT Email: jyu497@gatech.edu

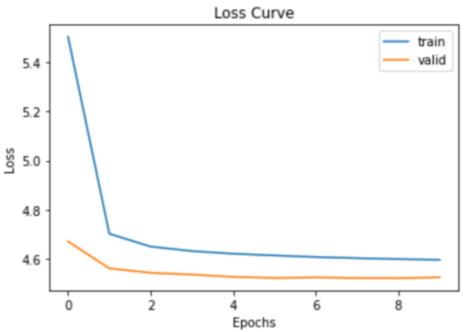
Seq2Seq Results

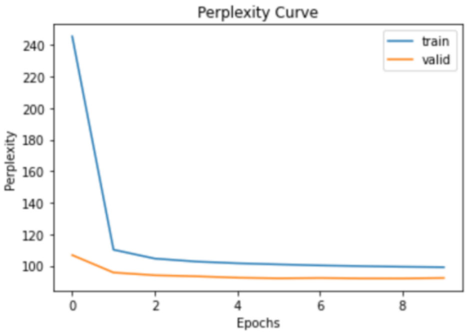
Put your results from training before and after hyperparameter tuning here.

Results for default configuration using RNN		Results for default Configuration Using LSTM	
Training Loss	4.5970	Training Loss	4.3465
Training Perplexity	99.1836	Training Perplexity	77.2043
Validation Loss	4.5216	Validation Loss	4.2301
Validation Perplexity	91.9815	Validation Perplexity	68.7232
Result for your Best Model using RNN after hyperparameter tuning		Resut for your Best Model using LSTM after hyperparameter tuning	
Training Loss	4.5970	Training Loss	3.6090
Training Perplexity	99.1836	Training Perplexity	36.9287
Validation Loss	4.5216	Validation Loss	3.7294
Validation Perplexity	91.9815	Validation Perplexity	41.6544
Your best model configuration for RNN after hyperparameter tuning		Your best model configuration for LSTM after hyperparameter tuning	
emb_size=32, hidden_size=64, dropout=0.2, learning_rate=1e-3 EPOCHS=10		emb_size=32, hidden_size=128, dropout=0.2, learning_rate=1e-3 EPOCHS=40	

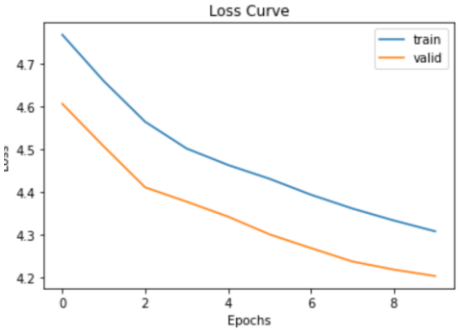
Seq2Seq Curves

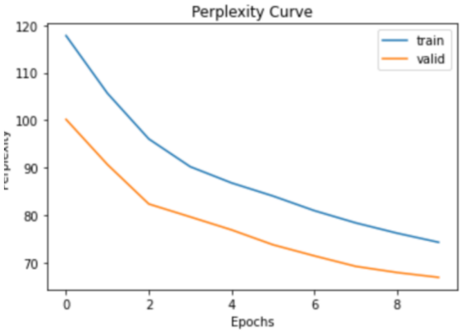
Default RNN



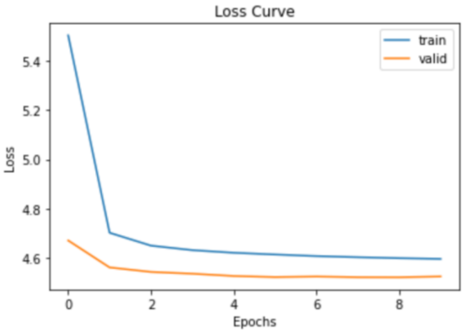


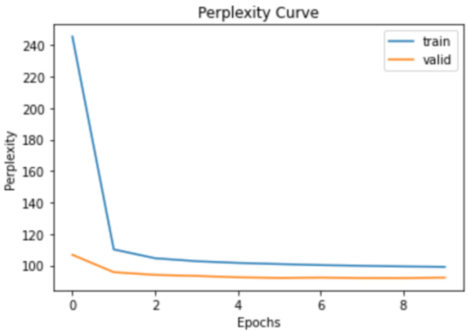
Default LSTM



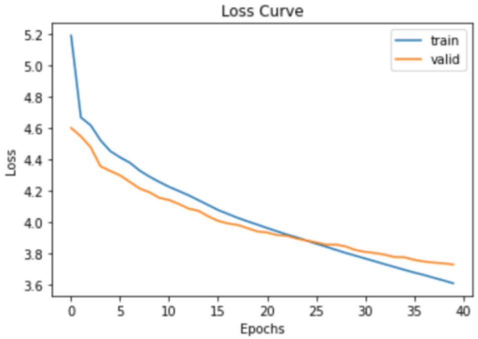


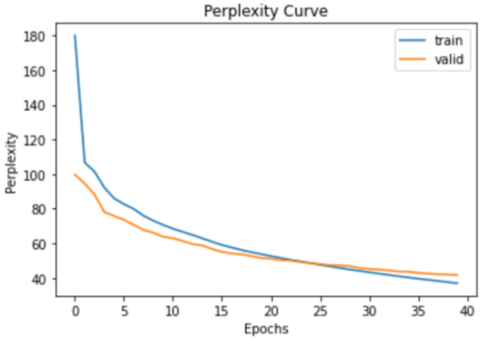
Best RNN





Best LSTM





Seq2Seq Explanation

Explain what you did here and why you did it to improve your model performance. Compare and explain the differences when using LSTM vs RNN. You can use another slide if needed.

- emb_size: defines the length of the vectors used to represent words. In generally, a larger dimensionality will result in more representation power. But will converge slower.
- hidden_size: larger hidden_size => more complex model, more representation power. But will converge slower.
- dropout: prevents overfitting by making each node in hidden state unavailable for an observation with a given probability. Larger dropout => more regularization.
- leaning_rate: the step size to take in learning parameters. Larger learning_rate => faster converge.
- EPOCHS: the iteration size to learn parameters. Larger epoch => more converge.

- The LSTM was developed to address the vanishing gradient problem of the Simple RNN that limited the training of deep RNNs.
- LSTM includes a 'memory cell' that can maintain information in memory for long periods of time.
- LSTM consistently outperformes Simple RNN, but more computational and memory expensive.

Transformer Results

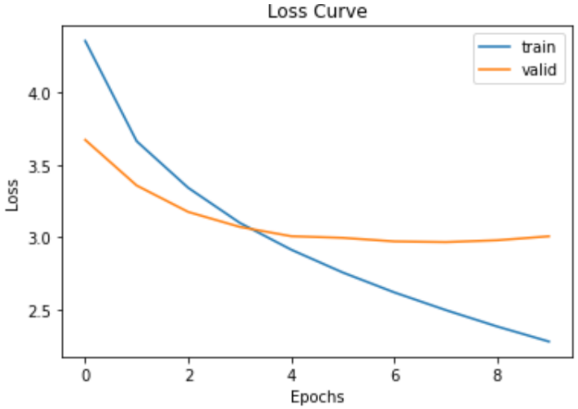
Put your results from training before and after hyperparameter tuning here.

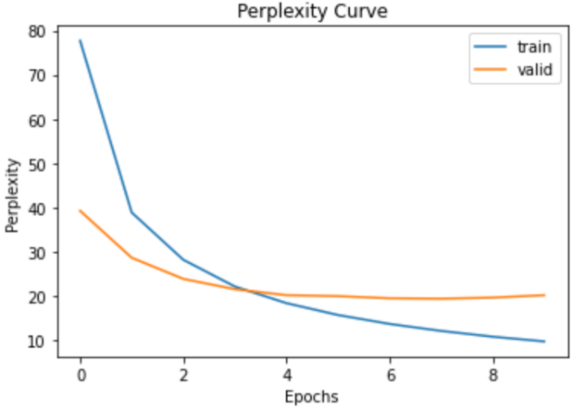
Results for default configuration			
Training Loss	2.2831	Validation Loss	3.0074
Training Perplexity	9.8067	Validation Perplexity	20.2349
Result for your Best Model			
Training Loss	1.8265	Validation Loss	3.2875
Training Perplexity	6.2121	Validation Perplexity	26.7766
Your best model configuration after hyperparameter tuning			
learning_rate=5e-4, EPOCHS=20, batch_size=64			

Table 2

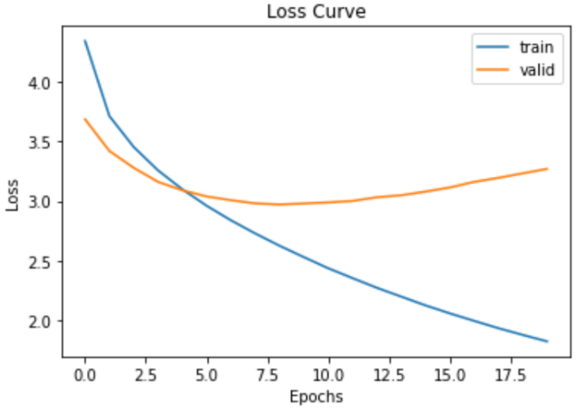
Transformer Curves

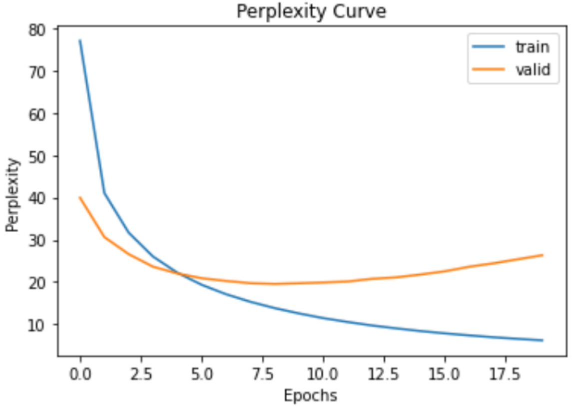
Default Transfer





Best Transfer





Transformer Explanation

Explain what you did here and why you did it to improve your model performance. You can use another slide if needed.

- leaning_rate: the step size to take in learning parameters. Larger learning_rate => faster converge.
- EPOCHS: the iteration size to learn parameters. Larger epochs => more converge.
- batch_size: too large of a batch size will lead to poor generalization. Since Transformer tend to overfit more, I decreased the batch_size.

Transformer Translation Results

Put translation results for your best model (1st 9 sentences) here

Input sentence	Back translation
'<sos>', 'a', 'young', 'boy', 'jumps', 'into', 'water', '.', '<eos>'	'<sos>', 'a', 'little', 'boy', 'is', 'into', 'into', 'water', '.', '<eos>'
'<sos>', 'a', 'native', 'woman', 'is', 'working', 'on', 'a', 'craft', 'project', '.', '<eos>'	'<sos>', 'a', 'is', 'works', 'working', 'a', 'a', 'wall', '.', '<eos>'
'<sos>', 'an', 'asian', 'woman', 'sitting', 'outside', 'an', 'outdoor', 'market', 'stall', '.', '<eos>'	'<sos>', 'an', 'asian', 'asian', 'sitting', 'front', 'in', 'market', '.', '<eos>'
'<sos>', 'woman', 'standing', 'on', 'a', 'brick', 'wall', 'and', 'taking', 'a', 'picture', '<eos>'	'<sos>', 'woman', 'standing', 'on', 'a', 'stone', 'wall', 'photographs', 'photographs', 'photographs', 'picture', 'a', '.', '.', '<eos>'
'<sos>', 'there', 'are', 'construction', 'workers', 'working', 'hard', 'on', 'a', 'project', '.', '<eos>'	'<sos>', 'construction', 'workers', 'workers', 'on', 'on', 'a', 'metal', '.', '.', '<eos>'
'<sos>', 'a', 'man', 'in', 'a', 'cluttered', 'office', 'is', 'using', 'the', 'telephone', '<eos>'	'<sos>', 'a', 'man', 'is', 'a', 'a', 'a', 'office', 'a', 'office', '.', '.', '<eos>'
'<sos>', 'two', 'chinese', 'people', 'are', 'standing', 'by', 'a', 'chalkboard', '.', '<eos>'	'<sos>', 'two', 'cars', 'stand', 'on', 'a', 'doorway', '.', '.', '<eos>'
'<sos>', 'children', 'are', 'playing', 'a', 'sport', 'on', 'a', 'field', '.', '<eos>'	'<sos>', 'children', 'are', 'playing', 'on', 'a', 'a', 'field', 'field', '<eos>'
'<sos>', 'a', 'man', 'is', 'working', 'at', 'a', 'construction', 'site', '.', '<eos>'	'<sos>', 'a', 'man', 'working', 'working', 'a', 'construction', 'construction', 'construction', 'site', '<eos>'

LSTM Translation Results

Put translation results for your best model (1st 9 sentences) here

Input sentence	Back translation
'<sos>', 'a', 'young', 'boy', 'jumps', 'into', 'water', '.', '<eos>'	'<sos>', 'a', 'little', 'boy', 'is', 'through', 'the', 'the', '.', '<eos>'
'<sos>', 'a', 'native', 'woman', 'is', 'working', 'on', 'a', 'craft', 'project', '.', '<eos>'	'<sos>', 'a', 'woman', 'is', 'at', 'a', 'a', 'a', '.', '<eos>'
'<sos>', 'an', 'asian', 'woman', 'sitting', 'outside', 'an', 'outdoor', 'market', 'stall', '.', '<eos>'	'<sos>', 'an', 'asian', 'woman', 'is', 'at', 'a', 'a', 'a', '.', '.', '<eos>'
'<sos>', 'woman', 'standing', 'on', 'a', 'brick', 'wall', 'and', 'taking', 'a', 'picture', '<eos>'	'<sos>', 'woman', 'is', 'a', 'a', 'a', 'a', '.', '.', '<eos>'
'<sos>', 'there', 'are', 'construction', 'workers', 'working', 'hard', 'on', 'a', 'project', '.', '<eos>'	'<sos>', 'construction', 'workers', 'are', 'on', 'a', 'a', 'a', '.', '.', '<eos>'
'<sos>', 'a', 'man', 'in', 'a', 'cluttered', 'office', 'is', 'using', 'the', 'telephone', '<eos>'	'<sos>', 'a', 'man', 'is', 'a', 'a', 'a', 'a', '.', '.', '<eos>'
'<sos>', 'two', 'chinese', 'people', 'are', 'standing', 'by', 'a', 'chalkboard', '.', '<eos>'	'<sos>', 'two', 'construction', 'are', 'are', 'on', 'a', 'a', '.', '<eos>'
'<sos>', 'children', 'are', 'playing', 'a', 'sport', 'on', 'a', 'field', '.', '<eos>'	'<sos>', 'children', 'are', 'a', 'a', 'a', 'a', '.', '.', '<eos>'
'<sos>', 'a', 'man', 'is', 'working', 'at', 'a', 'construction', 'site', '.', '<eos>'	'<sos>', 'a', 'man', 'is', 'on', 'a', 'a', '.', '.', '<eos>'

Compare LSTM to Transformer

Compare your LSTM results to your Transformer Results both quantitatively and qualitatively and explain the differences.

- Attention is an improvement to the model that allows the decoder to “pay attention” to different words in the input sequence as it outputs each word in the output sequence.
- Having attention, Transformer results in dramatically better performance than LSTM in loss and translation.
- While Transformer also seems to overfit more, thus shows problems with generalization.
- Transformer can allow both data and model parallel training, thus is much more efficient than LSTM.

Theory question

Beam search

Use the notation in “When to Finish? Optimal Beam Search for Neural Text Generation (modulo beam size) ” [Huang 18’]:

If $B_{i,1} \leq best_{\leq i}$ then $B_{i,j} \leq B_{i,1} \leq best_{\leq i}$ for all items $B_{i,j}$ in beam B_i. Future descendants grown from these items will only be no better, since probability ≤ 1, so all items in current and future steps are no better than $best_{\leq i}$.