LeslieDeras
801320720
Homework #5
https://github.com/lesliederas/5106.git

Problem 1: Without

Model	Seq Len	Final Loss	Val Acc	Train Time	Params
Transformer (2L, 2H)	**10**	**1.7248**	0.1000	0.70 sec	1,206,301
LSTM (2L)	10	2.4305	**0.0750**	**0.60 sec**	932,765
Transformer (2L, 2H)	**20**	**1.9243**	**0.2368**	0.92 sec	1,206,301
LSTM (2L)	20	2.5566	**0.2368**	**0.72 sec**	932,765
Transformer (2L, 2H)	**30**	**1.7958**	**0.2222**	**1.03 sec**	1,206,301
LSTM (2L)	30	2.5343	0.1111	1.46 sec	932,765

Training Loss: The Transformer consistently achieved **lower training loss** across all sequence lengths, indicating better learning and optimization capacity. For example, at sequence length 30, the Transformer reached a loss of **1.7958**, compared to the LSTM's **2.5343**.

Validation Accuracy: The Transformer outperformed or matched the LSTM in validation accuracy at sequence lengths 20 and 30, reaching a peak of **0.2368**. However, both models performed worse at sequence length 10, with accuracy dropping to **0.1000** for the Transformer and **0.0750** for the LSTM.

Training Time: The LSTM trained slightly faster at shorter sequence lengths (10 and 20), but the Transformer was **more efficient at longer sequences**, such as length 30, where it trained in **1.03 seconds** versus the LSTM's **1.46 seconds**.

Model Size: The LSTM had a smaller model footprint (**932,765 parameters**) compared to the Transformer (**1,206,301 parameters**), making it potentially more suitable for deployment in resource-constrained environments.

The **Transformer model demonstrated superior performance** in terms of accuracy and training loss, especially for **longer input sequences**, despite having a larger size. The LSTM

showed efficiency in training time and memory, but its learning capacity and generalization were limited compared to the Transformer.

With Attention

Seq Length	Model	Epoch 1 Loss	Epoch 1 Val Acc	Epoch 10 Loss	Epoch 10 Val Acc	Training Time (sec)	Model Size (parameters)
10	**Transformer (2 layers, 2 heads)**	**3.2060**	**0.2250**	**1.7473**	**0.2250**	**0.75**	**1,206,301**
	LSTM with Attention (2 layers)	**3.3420**	**0.2500**	**2.7378**	**0.1750**	**0.70**	**933,022**
20	**Transformer (2 layers, 2 heads)**	**3.2324**	**0.0526**	**1.8209**	**0.1316**	**0.90**	**1,206,301**
	LSTM with Attention (2 layers)	**3.3501**	**0.1053**	**2.8217**	**0.1053**	**0.79**	**933,022**
30	**Transformer (2 layers, 2 heads)**	**3.2249**	**0.1111**	**1.7194**	**0.0833**	**1.00**	**1,206,301**
	LSTM with Attention (2 layers)	**3.3345**	**0.0278**	**2.7135**	**0.1944**	**1.64**	**933,022**

1. **Performance:**
    - The Transformer (2 layers, 2 heads) generally outperforms the LSTM with Attention (2 layers) in both training loss and validation accuracy across all sequence lengths.
    - The Transformer model achieves lower losses and higher validation accuracies, especially at larger sequence lengths (20 and 30).
    - The LSTM with Attention model shows a slower improvement and more fluctuation in performance, with validation accuracy remaining relatively low throughout the epochs.
2. **Training Time:**

- The Transformer model takes slightly more time to train as sequence length increases (from 0.75 sec for length 10 to 1.00 sec for length 30).
- The LSTM with Attention model, while still efficient, shows variability in training time, with the highest time of 1.64 sec for sequence length 30.

3. **Model Size:**
   - The Transformer model is significantly larger, with a model size of 1,206,301 parameters.
   - The LSTM with Attention model is smaller, with 933,022 parameters.

Transformer models tend to perform better in terms of accuracy and loss but come at the cost of larger model size and increased training time.
LSTM with Attention models are more lightweight in terms of parameters and tend to require less training time but struggle to achieve similar performance to the Transformer models, particularly with longer sequences.

Problem 2

Part 1.

For Sequence Length 20:

Model	Train Loss	Test Loss	Accuracy	Training Time	Model Size (parameters)
Transformer	1.9982	1.7386	0.4754	932.76 sec	1,206,301
RNN (LSTM)	1.5795	1.4110	0.5663	653.07 sec	933,022

For Sequence Length 30:

Model	Train Loss	Test Loss	Accuracy	Training Time	Model Size (parameters)

Transformer	2.0048	1.7454	0.4759	1221.25 sec	1,206,301
RNN (LSTM)	1.5696	1.4032	0.5663	490.21 sec	933,022

Transformers provide powerful performance but come with a trade-off in terms of higher computational costs and slower training times.
LSTMs (RNN-based models) perform similarly in accuracy but are much more computationally efficient in terms of training time and model size.

Part 2

Layers	Heads	Seq Length	Train Loss	Test Loss	Accuracy	Execution Time (sec)	Model Size (params)
1	2	20	1.9914	1.7715	0.4680	553.77	2,933,313
1	2	30	2.0257	1.7795	0.4669	863.79	2,935,873
2	2	20	2.0048	1.7504	0.4709	1,255.21	5,827,137
2	2	30	1.9925	1.7468	0.4767	1,485.66	5,829,697
4	2	20	3.3225	3.3168	0.1526	1,779.35	11,614,785
4	2	30	3.3226	3.3192	0.1515	2,148.48	11,617,345
1	4	20	1.9594	1.7151	0.4840	503.77	2,933,313

1	4	30	1.9605	1.7183	0.4832	845.06	2,935,873
2	4	20	1.9932	1.7296	0.4727	1,344.60	5,827,137
2	4	30	2.0035	1.7494	0.4742	1,609.23	5,829,697
4	4	20	3.3225	3.3179	0.1531	1,653.29	11,614,785
4	4	30	3.3228	3.3177	0.1524	2,167.91	11,617,345

Increasing the number of layers and heads consistently leads to higher computational complexity, both in terms of execution time and model size.
The 4-layer architectures with 2 or 4 heads lead to much higher model sizes and execution times but show significantly worse performance in terms of accuracy and loss (especially with longer sequences).

TrainingTime:
As the number of layers and heads increases, the execution time also increases. For example, with 4 layers and 4 heads, the training time for sequence length 30 is over 2,100 seconds, while with 1 layer and 2 heads, it's only 503.77 seconds for sequence length 20.

Accuracy and Loss:
For smaller architectures (1 layer, 2 heads or 1 layer, 4 heads), the models achieve better accuracy and lower losses compared to the deeper models (4 layers).
The best accuracy is achieved by the 1 layer, 4 heads model with 0.4840 for sequence length 20 and 0.4832 for sequence length 30.

Scalability:
Larger architectures are more computationally demanding but do not necessarily lead to better accuracy or performance, particularly for this problem and dataset. For example, 4 layers with 4 heads have significantly higher training times and worse performance.

1-layer models (especially with **4 heads**) provide the best trade-off between performance and computational complexity.
Increasing the number of layers and heads leads to higher computational costs without substantial improvements in accuracy, especially for sequence lengths beyond 20.

Part 3

Layers	Heads	Seq Length	Train Loss	Test Loss	Accuracy	Execution Time (sec)	Model Size (params)
1	2	50	2.0004	1.7293	0.4789	953.50	2,940,993
2	2	50	2.0495	1.7989	0.4575	1,827.85	5,834,817
4	2	50	3.3234	3.3147	0.1519	5,403.94	11,622,465
1	4	50	1.9798	1.7499	0.4751	954.57	2,940,993
2	4	50	2.0020	1.7084	0.4874	1,846.68	5,834,817
4	4	50	3.3234	3.3163	0.1535	4,659.46	11,622,465

Accuracy: Sequence length 50 typically improves accuracy for lower layer and head configurations, but higher layers and heads have poor accuracy, as shown above with very low accuracy values (around 0.15).

Execution Time: As expected, the execution time increases with both sequence length and the number of layers/heads. The most complex configurations take significantly longer.

Model Size: The model size scales up with the number of layers and heads, confirming that increasing complexity results in larger models.

Question 3.

Model Configuration	Train Loss	Validation Loss	Validation Accuracy
1 Layer, 2 Heads	0.1854	5.5629	0.3176
1 Layer, 4 Heads	0.0342	5.2265	0.3108
2 Layers, 2 Heads	0.4509	5.3616	0.2230
2 Layers, 4 Heads	0.4433	5.4547	0.2500
4 Layers, 4 Heads	4.6159	6.4736	0.1554
4 Layers, 4 Heads	4.6763	6.6078	0.1554

GRUwith attention

Validation loss	Validation Accuracy	Epochs	Learning rate	Hidden size	Training loss
0.4059873691995353	79%	30	.008	256	30 epochs & print loss every 2 epochs

Accuracy: The models with 1 layer and 2 heads perform better than those with more layers or heads. The accuracy improves slightly for models with fewer layers and heads.
Validation Loss: Generally, as the model increases (more layers or heads), the validation loss tends to increase as well, which may indicate overfitting.

GRU Without

Validation loss	Validation Accuracy	Epochs	Learning rate	Hidden size	Training loss
0.41216158807478614	77%	30	.009	256	30 epochs & print loss every 2

					epochs

Model Configuration	Train Loss (Final Epoch)	Val Loss (Final Epoch)	Val Accuracy (Final Epoch)
1 Layer, 2 Heads	**0.1854**	**5.5629**	**0.3176**
1 Layer, 4 Heads	**0.0342**	**5.2265**	**0.3108**
2 Layers, 2 Heads	**0.4509**	**5.3616**	**0.2230**
2 Layers, 4 Heads	**0.4433**	**5.4547**	**0.2500**
4 Layers, 2 Heads	**4.6159**	**6.4736**	**0.1554**
4 Layers, 4 Heads	**4.6763**	**6.6078**	**0.1554**

- **Final Evaluation Loss: 0.4122**
- **Final Evaluation Accuracy: 0.7699**

Best Performing Model: The Transformer with 1 layer and 4 heads achieved the best results in terms of training loss (0.0342) but with slightly lower validation accuracy (0.3108) .
Worst Performing Model: The Transformer with 4 layers and 4 heads showed minimal improvement and had the lowest validation accuracy (0.1554), suggesting overfitting or difficulty in learning the translation task.
Your Transformer model shows a better balance between train loss and validation accuracy when compared with an RNN-based model with and without attention.

Problem 4

Layers	Heads	Val Accuracy
1	**2**	**0.0000**
1	**4**	**0.3333**
2	**2**	**0.3333**

2	4	0.0000
4	2	0.3333
4	4	0.3333

Without Attention

Epoch	Training Loss
0	3.9163
14	2.1659
28	0.5033

Validation Loss: 0.3432

Validation Accuracy: 97.17%

With attention

Epoch	Training Loss
0	4.1185
14	2.1074
28	0.2866

Validation Loss: 0.1745

Validation Accuracy: 97.35%

Comparison

Model	Val Accuracy	Val Loss
Transformer (best)	**33.33%**	**~2.6–4.5**
RNN (no attention)	**97.17%**	**0.3432**
RNN (with attention)	**97.35%**	**0.1745**

English → French is slightly harder due to:
Richer grammar/gender conjugation in French.
Word order differences.
RNNs (especially with attention) still perform well.

RNNs with attention outperform Transformers in your experiments because:

- They handle small data better.
- Their attention is explicit and targeted.
- They generalize well with fewer parameters.
- They're more stable during training in low-resource settings.

If you scale up to a bigger dataset, Transformers will likely shine — but for now, RNNs are the best for language translation