Jacob Haywood

801182240

Assignment 4

Github: https://github.com/jhaywoo6/Deep_Learning

The given assignment was to design, test, and evaluate a transformer model with RNN based models from assignments 3 and 4. The same inputs and combinations for the RNN based models were replicated with the transformer-based models, with the addition of varying the number of layers in the transformer architecture as well as the number of heads. The results of these tests will be presented here, with comments comparing them with Assignments 3 and 4's RNN based models. The results of Assignments 3 and 4 can be found through the github link.

Problem 1:

Model	Sequence Length	Loss	Validation Accuracy	Training time	Parameters	Size (MB)	Predicted next character
1	10	3.729479	0.172131	28.33655	2653742	10.12322	
2	20	2.163639	0.238683	51.89002	2653742	10.12322	t
3	30	2.188981	0.264463	58.39602	2653742	10.12322	

The first test with varying sequence length for the provided text preformed quite poorly compared to similar RNN models. While sequence length 10 is similar in performance, varying it did not lead to significant increases in accuracy. It also tended to pick blank spaces as shown in the predicted next char column. An assumption could be made that the input data provided is far too short to make an accurate model, but problem 2 may show that either this is true to a much greater extent than expected, or test error is to blame.

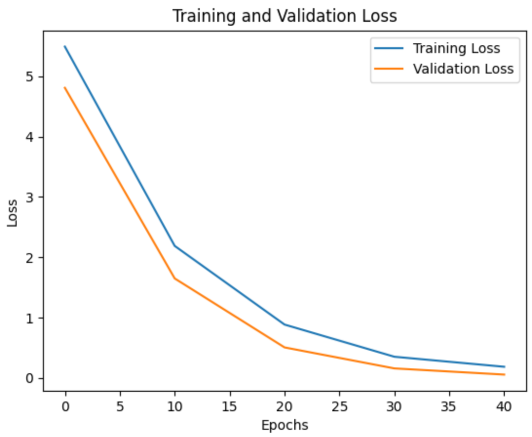
Problem 2:

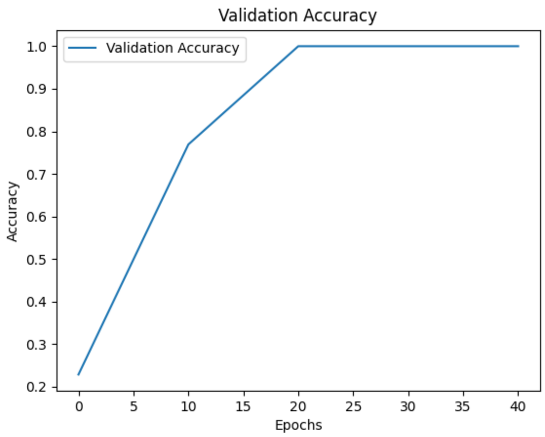
Model	Sequence Length	Training Loss	Validation Loss	Validation Accuracy	Training time	Inference Time	Fully Connected Layers	Parameters	Size (MB)	nhead
1	20	2.12487	2.057504	0.379759	1357.529	0.497324	1	609729	2.325932	2
2	20	2.05376	1.967327	0.400517	1188.245	0.420698	1	609729	2.325932	4
3	30	2.131892	2.048533	0.378452	1695.434	0.417211	1	609729	2.325932	2
4	30	2.11451	2.034534	0.381146	1805.346	0.42571	1	609729	2.325932	4
5	50	2.200628	2.13079	0.353521	2828.286	0.417831	1	609729	2.325932	2
6	50	2.184811	2.115306	0.358087	2754.728	0.42383	1	609729	2.325932	4
7	20	1.926262	1.787281	0.449118	2186.3	0.627297	2	1202753	4.588139	2
8	20	1.915149	1.765871	0.456101	2144.569	0.636348	2	1202753	4.588139	4
9	30	2.002744	1.877084	0.423768	3150.027	0.631343	2	1202753	4.588139	2
10	30	1.992892	1.856472	0.429727	3082.931	0.631346	2	1202753	4.588139	4
11	50	2.073612	1.964604	0.397463	5094.428	0.648853	2	1202753	4.588139	2
12	50	2.072621	1.960696	0.399455	4740.833	0.703892	2	1202753	4.588139	4
13	20	1.725886	1.478221	0.538644	3533.677	1.19516	4	2388801	9.112553	2
14	20	1.723658	1.478901	0.539892	3526.234	1.17515	4	2388801	9.112553	4
15	30	1.808453	1.578608	0.509094	5213.681	1.163154	4	2388801	9.112553	2
16	30	1.811941	1.57688	0.5101	5446.858	1.169123	4	2388801	9.112553	4
17	50	1.906155	1.714578	0.467248	9665.996	1.089084	4	2388801	9.112553	2
18	50	1.908351	1.700656	0.47261	9193.191	1.03007	4	2388801	9.112553	4

Problem 2 proved to be problematic to test given the available hardware. The test was performed on an Nvidia 4070 card, and yet due to the large size of the shakespear dataset, training all requested versions of the model took several days, and the resulting predicted test strings were very poor. The model would simply repeat the same character repeatedly. Multiple hidden sizes could not be tested due to the long amount of training time required that would push back the presentation of this report by a week or more. Strangely, the final accuracy remained just under half for most models. In this case, the RNN styled models were significantly better as they could provide coherent phrases that match the original text. It is admitted that the implementation of the transformer was likely flawed, perhaps lacking the ability to remember prior data as an attention based model should, or the criteria could have been flawed, so in a future test these elements should be carefully rewritten prior to testing all requested combinations.

Problem 3 & 4:

All variations of the training resulted in loss and accuracy graphs that appear like this for each translation direction, nhead amount, and layers.





The loss and accuracy criteria were flawed as resulting sample text was very poor in quality. The only variation of values that produced text for English to French was two layers with 2 nhead:

Sample Translations:

Input: They laugh at the joke

Target: Ils rient de la blague

Predicted: du shopping

Input: She catches the bus

Target: Elle attrape le bus

Predicted: les

French to English was much more favorable, with much fewer instances of blank predictions. Results were still poor, especially compared to RNN based models. There wasn’t a significant perceivable difference in types of predictions across all variables. Much context was lost and only select snips had correctly guessed a handful of correct words.

Sample Translations:

Input: Ils jouent aux jeux vidéo

Target: They play video games

Predicted: play soccer every weekend

Input: Le chien aboie bruyamment

Target: The dog barks loudly

Predicted: cat meows loudly

Input: Elle enseigne l'anglais à l'école

Target: She teaches English at school

Predicted: music

Input: Ils mettent la table

Target: They set the table

Predicted: off the table

Input: Ils sont étudiants

Target: They are students

Predicted: dinner for

Sample Translations:

Input: Elle danse avec grâce

Target: She dances gracefully

Predicted: dances gracefully

Input: Ils lisent des livres à la bibliothèque

Target: They read books at the library

Predicted: goodbye at university books at university

Input: Il chante magnifiquement

Target: He sings beautifully

Predicted: sings beautifully

Input: Ils visitent souvent des musées

Target: They visit museums often

Predicted: goodbye

Input: Il peint des paysages

Target: He paints landscapes

Predicted: for his family

Overall conclusions:

Longer training times, larger data requirements, and specific set up make transformers significantly more difficult to properly implement compared to similar RNN models. While a well built and trained transformer may be very effective at dealing with very large datasets, the sets tested here are not large or complex enough to justify choosing a transformer based model when similar RNN based models are more accurate with far less training time given the same time. However, should the same tests be attempted again, adjusting the transformer model and loss criteria may lead to better results than that presented here.