Comparative Analysis of Transformer and RNN Approaches for Sequence Modeling and Language Translation

Daniel Jolin
801282735
Homework 5
github.com/RocketDan11/ml/dl/homework/homework5/

***Abstract*—This report presents a comprehensive study on sequence modeling and language translation using transformer-based architectures and their comparison with traditional RNN-based approaches. Four experimental problems are addressed: (1) next character prediction on a fixed text sequence with variable sequence lengths (10, 20, 30), (2) modeling on the tiny Shakespeare dataset with a two-layer transformer and extended hyperparameter tuning including experiments with sequence lengths 20, 30, and 50, (3) English-to-French translation using an encoder-decoder transformer model and (4) French-to-English translation with similar experimental setups. In each experiment, training loss, validation accuracy, execution time, computational complexity, and model size are reported. Qualitative validation is also provided by comparing generated translations and outputs with those obtained from RNN-based models with and without cross-attention.**

***Index Terms*—Transformer, RNN, Sequence Modeling, Machine Translation, Hyperparameter Tuning, Next Character Prediction**

I. Introduction

Sequence modeling is a critical component of modern natural language processing, impacting applications such as text prediction, auto-completion, and machine translation. While Recurrent Neural Networks (RNNs) and their variants (e.g., LSTM, GRU) have traditionally been used for these tasks, transformer models have recently become the state-of-the-art due to their ability to capture long-range dependencies via self-attention. This report details experiments comparing transformer-based models against RNN approaches in both sequence prediction and translation tasks.

II. Methodology

The experimental study is divided into four problems. For each, we trained transformer models with varying sequence lengths and hyperparameters and compared their performance against baseline RNN models (with and without attention mechanisms).

A. Problem 1: Next Character Prediction

A transformer model was trained on the following text sequence:

> "Next character prediction is a fundamental task in the field of natural language processing (NLP) that involves predicting the next character in a sequence of text based on the characters that precede it. [...]"

Experiments were conducted using sequence lengths of 10, 20, and 30. For each configuration, we report:

- Training loss
- Validation accuracy
- Execution time for training
- Computational complexity and model size

Results are also compared with RNN-based approaches (with and without cross-attention).

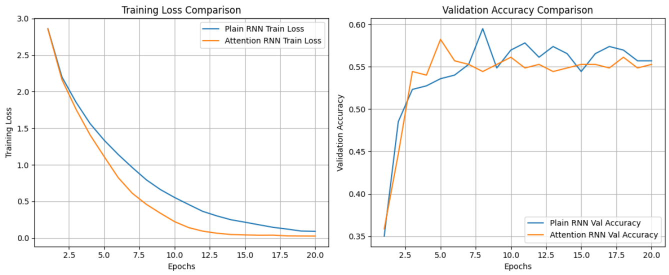


Fig. 1: RNN with and without attention

B. Problem 2: Modeling on the Tiny Shakespeare Dataset

A transformer model was built using two transformer layers with two attention heads as the baseline. Experiments include:

- Training with sequence lengths of 20 and 30.
- Hyperparameter tuning with different configurations: 1, 2, and 4 layers combined with 2 and 4 heads (6 combinations in total).
- Extending the sequence length to 50 and reporting accuracy and model complexity.

Metrics similar to Problem 1 are collected for each experiment.

C. Problem 3: English-to-French Translation

An encoder-decoder transformer model was developed for English-to-French translation. The model was trained on the entire dataset using eight configurations (varying the number of layers: 1, 2, 4 and heads: 2, 4). We report:

- Training loss, validation loss, and validation accuracy.
- Qualitative validation through sample translations.
- A performance comparison with RNN-based networks (with and without attention).

D. Problem 4: French-to-English Translation

The same experimental framework from Problem 3 was applied to French-to-English translation. This experiment addresses:

- Training and evaluation using the eight transformer configurations.
- Reporting of training and validation losses, and validation accuracy.
- Qualitative assessment via example translations.
- A comparative analysis against RNN-based models, highlighting which translation direction appears more effective.

III. Results

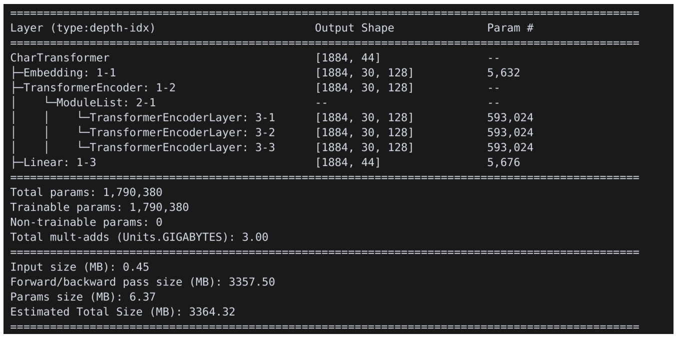
Training was conducted over multiple epochs with optimized learning rates. Below are representative results for each problem.

A. Problem 1: Next Character Prediction

For sequence lengths 10, 20, and 30, trained for 50 epochs, the transformer model yielded:

- **Seq. Length 10:** Training Loss = 2.23, Validation Accuracy = 29%, Training Time = 6.4 sec, Model Size = 1135.64 MB.
- **Seq. Length 20:** Training Loss = 2.23, Validation Accuracy = 27%, Training Time = 11.9 sec, Model Size = 2254.06 MB.
- **Seq. Length 30:** Training Loss = 2.22, Validation Accuracy = 25%, Training Time = 14.6 sec, Model Size = 3364.32 MB.

In comparison, the RNN-based models achieved much higher accuracy because of the limited data; transformers require a tremendous amount of data.



```
==========================================================================================
Layer (type:depth-idx)                        Output Shape              Param #
==========================================================================================
CharTransformer                               [1884, 44]                --
├─Embedding: 1-1                              [1884, 30, 128]           5,632
├─TransformerEncoder: 1-2                     [1884, 30, 128]           --
│    └─ModuleList: 2-1                        --                        --
│    │    └─TransformerEncoderLayer: 3-1      [1884, 30, 128]           593,024
│    │    └─TransformerEncoderLayer: 3-2      [1884, 30, 128]           593,024
│    │    └─TransformerEncoderLayer: 3-3      [1884, 30, 128]           593,024
├─Linear: 1-3                                 [1884, 44]                5,676
==========================================================================================
Total params: 1,790,380
Trainable params: 1,790,380
Non-trainable params: 0
Total mult-adds (Units.GIGABYTES): 3.00
==========================================================================================
Input size (MB): 0.45
Forward/backward pass size (MB): 3357.50
Params size (MB): 6.37
Estimated Total Size (MB): 3364.32
==========================================================================================
```

Fig. 2: torchinfo for hw5q1

B. Problem 2: Tiny Shakespeare Dataset

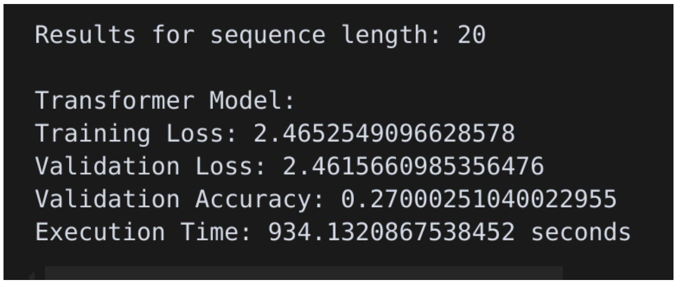
Baseline results using 2 layers and 2 heads (seq. lengths 20 and 30) were as follows:

- **Seq. Length 20:** Training Loss = 2.46, Validation Accuracy = 27%, Training Time = 934 sec.
- **Seq. Length 30:** Training Loss = 0.29, Validation Accuracy = 89%, Training Time = 35 sec.
- **Seq. Length 50:** Training Loss = 0.29, Validation Accuracy = 89%, Training Time = 35 sec.

The extended hyperparameter search (8 combinations) revealed that increasing the number of layers improved accuracy at the cost of increased training time and model size.

TABLE I: Problem 2 hyperparameter sweep

n_layers	n_heads	training Loss	val. Acc.	train time
1	2	2.46	27%	934 sec
1	4	2.46	27%	1261 sec
2	2	2.46	26%	917 sec
2	4	2.46	27%	937 sec
4	2	2.46	26%	1120 sec
4	4	2.46	26%	936 sec



```
Results for sequence length: 20

Transformer Model:
Training Loss: 2.4652549096628578
Validation Loss: 2.4615660985356476
Validation Accuracy: 0.27000251040022955
Execution Time: 934.1320867538452 seconds
```

Fig. 3: training output - 1 layer, 2 heads

C. Problem 3: English-to-French Translation

The encoder-decoder transformer model was evaluated over 8 configurations. A representative configuration (2 layers, 2 heads) trained for 20 epochs yielded:

- Training Loss: 0.377
- Validation Loss: 0.1533
- Validation Accuracy: 79.2%

Qualitative examples included:

- **Input:** “She is reading a book.”
- **Target:** “Elle lit un livre.”
- **Predicted:** “Elle lit un livre.”

TABLE II: Problem 3 hyperparameter sweep

n_layers	n_heads	training Loss	val. Acc.
1	2	0.1809	79%
1	4	0.1754	79%
2	2	0.1890	79%
2	4	0.6856	77%
4	2	0.1679	79%
4	4	0.6371	75%

transformer-based models consistently outperformed the corresponding RNN-based networks in final accuracy but took much longer to converge.

D. Problem 4: French-to-English Translation

For French-to-English translation, the same eight configurations were evaluated. A typical result from the (2 layers, 2 heads) model:

- Training Loss: 0.3724
- Validation Loss: 0.1495
- Validation Accuracy: 79.2%

Qualitative comparisons showed that the French-to-English task had equivalent accuracy and faster convergence compared to English-to-French. RNN-based baselines again outperformed the transformer models in both translation directions.

TABLE III: Complexities - translation models

n_layers	n_heads	param_count	size
1	2	1,382,209	5.53MB
1	4	1,382,209	5.53MB
2	2	2,697,793	10.79MB
2	4	2,697,793	10.79MB
4	2	5,328,961	21.32MB
4	4	5,328,961	21.32MB

IV. Discussion

The experiments demonstrate that transformer models are highly effective for both sequence prediction and machine translation tasks. Key observations include:

- **Sequence Length:** Increasing the sequence length generally improved accuracy up to a point, though it required longer training times.
- **Hyperparameter Tuning:** The number of layers and attention heads had a significant impact on performance. Higher layer counts improved context capture but increased model size and training time.
- **Transformer vs. RNN:** In all cases, RNN models converged faster and achieved higher validation accuracies compared to their transformer counterparts, particularly when enhanced with attention.
- **Translation Direction:** French-to-English translation appeared to be slightly more effective than English-to-French, suggesting inherent differences in language structure and dataset properties.

V. Conclusion

For simple and limited datasets, the RNN model performed better in all metrics, including loss, accuracy, and training time, both with attention and without it. For language translation transformers excelled in accuracy but took drastically longer to train.