Project: Movie Review Classification with Transformers

Hyeokgi Kim

January 5, 2026

1 Introduction

In this project, you will implement sequence classification models for sentiment analysis on the **IMDB Movie Review Dataset**. You will compare a baseline Recurrent Neural Network (RNN) against a Vanilla Transformer [1] and a custom-tweaked Transformer.

2 Tasks

The project consists of three parts. Use the IMDB dataset (binary classification, 25k train/25k test).

2.1 Task 1: Baseline Model (LSTM/GRU)

Implement a baseline model using a Recurrent architecture.

- **Allowed:** You may use standard high-level layers such as `nn.LSTM` or `nn.GRU` for this task.
- Record the accuracy and loss curves for this baseline to serve as a benchmark.

2.2 Task 2: Vanilla Transformer

Implement the **Encoder** part of the Transformer manually.

- **Do not** use high-level classes like `nn.TransformerEncoder` or `MultiheadAttention`. Implement the attention mechanism from scratch.
- Required components: Positional Embeddings, Multi-Head Self-Attention, Feed-Forward Networks, Residuals, and LayerNorm.

- Aggregation: Use Global Average Pooling or a [CLS] token to condense the sequence for the final linear classifier.

2.3 Task 3: Custom Tweaks

Implement at least **two** tweaks to improve the Transformer's performance (e.g., Pre-Norm architecture, Warmup schedulers, Gelu/SwiGLU activations, or pretrained embeddings).

3 Implementation Details

3.1 Tokenization Strategy

To ensure a fair comparison between models, use a consistent **word-level tokenizer**.

1. **Vocabulary:** Build a vocabulary of the top N frequent words (e.g., $N = 20,000$).
2. **Special Tokens:** Explicitly handle `<PAD>` (index 0) and `<UNK>` (index 1).
3. **Truncation:** Truncate or pad reviews to a fixed length (e.g., 256 tokens) to manage the $O(L^2)$ memory cost of attention.

3.2 Padding Masks

You must generate padding masks so the attention mechanism ignores `<PAD>` tokens.

```
# Example: Creating a mask for padding tokens
# mask shape: (batch_size, seq_len)
# True indicates the position should be ignored (is padding)
src_key_padding_mask = (src == pad_token_idx)
```

3.3 Training

Use Binary Cross Entropy Loss (`BCEWithLogitsLoss`). Train for sufficient epochs to observe convergence.

4 Report & Submission

Submit `code.zip` (with readme) and `report.pdf`. Do not submit datasets or checkpoints. The report must include:

1. **Model Overview:** Brief description of your implementations.
2. **Tweaks:** Explanation of the custom tweaks used in Task 3.
3. **Results:** A table comparing Test Accuracy for Baseline, Vanilla, and Tweaked models.
4. **Visualization:** Loss curves and (optional) Attention Weight heatmaps.

You don't have to spend too much time writing the report; focus on clarity and conciseness.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *NeurIPS*, 2017.