Post-Lab Write-Up: Transformer-Based LLM

Artur Zhdan

February 16, 2025

1 Model Description

We implemented a Transformer-based Large Language Model (LLM) inspired by nanoGPT. The model consists of:

- **Embedding Layer:** Converts tokens into dense vectors.
- **Positional Encoding:** Adds sequential information to embeddings.
- **Transformer Encoder:** Comprises 6 layers with 8 attention heads each, allowing the model to attend to different parts of the sequence.
- **Feed-Forward Layer & Dropout:** Introduces non-linearity and regularizes the model with a dropout rate of 0.1.

These parameters were selected to strike a balance between model performance and the computational limits of a single NVIDIA GeForce RTX 3050 Ti GPU.

2 Evaluation

The model was trained on the Tiny Shakespeare dataset. Training and validation loss curves showed a steady decrease, indicating effective learning. Additionally, sample outputs exhibit text generated in a style reminiscent of Shakespeare, confirming that the model captures key linguistic patterns from the dataset.

3 Reflection

This lab provided hands-on experience with Transformer architectures and model training in PyTorch. I learned the importance of parameter tuning and balancing model complexity with available computational resources. The exercise deepened my understanding of both theoretical concepts and practical challenges in deep learning.

4 Jupyter Notebook

The complete implementation—including data preprocessing, model definition, and training routines—is available in the accompanying Jupyter Notebook file.