CS 391L **Machine Learning WB** **Spring 2025**

Homework 4

Lecturer: Inderjit Dhillon Date Due: Apr 18, 2025

Keywords: *Transformer, Language Modeling, PyTorch*

Language Modeling with Transformers

In this assignment, you will implement a Transformer-based language model and train it on transcripts from class lectures. You will construct all major components of the model from scratch using PyTorch, including attention layers, transformer blocks, and text generation routines. You will also experiment with architectural choices such as dropout and residual connections to evaluate their effect on model performance.

(a) **Dataset Preparation (10 points)**
Implement a custom `Dataset` class to generate input-target token pairs for training. Each sample should consist of a sequence of `block_size` input tokens and the corresponding target tokens shifted by one position. Complete the `__getitem__` method to return (x, y) pairs.

(b) **Attention Mechanism (10 points)**
Implement the forward pass of the multi-head attention mechanism. You are provided with a `MultiHeadAttentionOp` class that performs causal attention. In the `Attention` class, implement:

- Linear projections for q, k, and v
- The attention operation using the provided module
- The output projection layer

(c) **Transformer Layer (20 points)**
Implement a single transformer encoder layer. Your implementation should include:

- Multi-head self-attention with residual connections and layer normalization
- A feed-forward network (FFN) with GELU activation
- A second residual connection and layer normalization

(d) **Full Transformer Model (20 points)**
Implement the full transformer model. Your model should:

- Embed token and positional indices
- Apply a stack of transformer layers
- Project the final hidden representations to vocabulary logits
- Return cross-entropy loss when target y is provided; otherwise, return logits

(e) **Autoregressive Token Generation (20 points)**
Implement a generation function that uses the trained model to produce text. Begin with a `[BOS]` token and iteratively sample the most likely next token until `max_new_tokens` is reached. At each step, use the last `BLOCK_SIZE` tokens as context.

(f) Effect of Dropout (10 points)
Train your model with different dropout rates (between $[0, 1]$). Report and compare training and validation losses. Comment on the regularization effect of dropout in this context.

(g) Effect of Residual Connections (10 points)
Repeat training without using residual connections. Report the model's performance and explain how residuals affect training dynamics and gradient flow.

(h) [Optional] Model Improvements
Experiment with architectural or training improvements (e.g., learning rate schedules, more layers, activation changes). Report the best validation loss you can achieve. We will report the best validation losses submissions get at the end of the assignment.