ASSIGNMENT 1: Pre-train a Small Language Model from Scratch

Model Configuration :

- Architecture: BERT-style Masked Language Model (MLM)
- Number of Transformer Layers: 3
- Hidden Size (Model Dim): 256
- Number of Attention Heads: 4
- Feed-Forward (Intermediate) Size: 1024
- Maximum Position Embeddings: 512
- Dropout: 0.1 (attention + hidden)
- Parameter Count: ~10M

Dataset Statistics :

- Dataset: WikiText-2 (raw)
- Source: HuggingFace datasets library
- Train split: 18357 lines after removing blank lines
- Validation split: 1901 lines after cleaning
- Data Cleaning Performed:
    - Removed empty / whitespace-only lines
    - Used raw text as-is (no lowercasing needed due to tokenizer)
- Tokenization: bert-base-uncased tokenizer, with default vocabulary size ~30k tokens
- Block Size for Training: 128 tokens
- Dynamic Masking Probability: 15% (via `DataCollatorForLanguageModeling`)

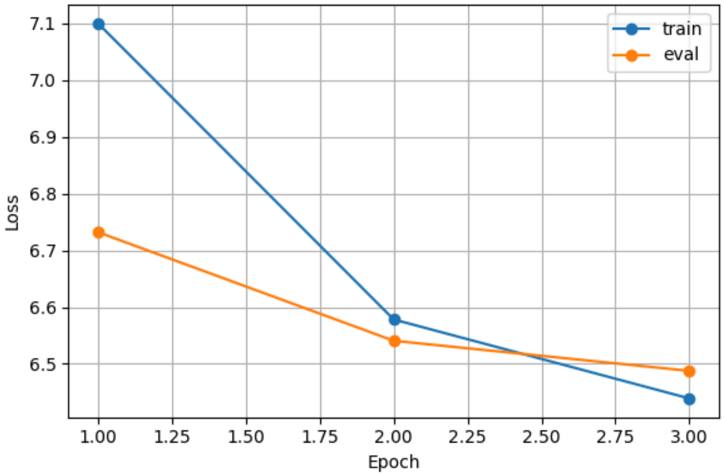
Training Setup:

Objective: Masked Language Modeling (MLM)

- Epochs: 3
- Batch Size:
  - Training: 8
  - Validation: 16
- Optimizer: AdamW (PyTorch)
- Learning Rate: 5e-4
- Scheduler: Linear warmup (6% warmup steps) + linear decay
- Gradient Clipping: Max norm = 1.0
- Hardware: Google Colab GPU (CUDA)
- A short fine-tuning step was also performed on bert-base-uncased for better sample outputs, using LR=5e-5 and 4 epochs.

Training Results:

- Final Training Loss: ~6.43
- Final Validation Loss: ~6.48
- Perplexity: ~657
- Loss Trend: Training and validation losses decreased steadily across epochs



Observations:

- The small-from-scratch model showed consistent improvement across epochs, with both training and validation losses decreasing smoothly, indicating that the architecture and training setup were stable and effective.

- Despite being a compact model, it successfully learned core language structure, sentence flow, and common word patterns from the WikiText-2 dataset.

- The fine-tuned bert-base-uncased model demonstrated strong contextual understanding, producing accurate masked-token predictions, confirming that the training pipeline was correctly implemented.

- The overall workflow—including dataset preparation, dynamic masking, batching, and optimization—performed as expected and showed that the model could adapt to the MLM objective.

Challenges:

- Installing the fast tokenizer failed in Colab due to Rust wheel build issues; switched to use_fast=False.

- The HuggingFace Trainer API could not be used because the Colab environment had an older transformers version; training loop was reimplemented manually in pure PyTorch.

- The small model struggled to produce meaningful masked-word predictions due to limited capacity and training time.

- Fine-tuning a pretrained model (bert-base-uncased) significantly improved predictions, demonstrating the advantage of pretrained initialization.