Assignment 2: Mini GPT-Style Language Model Training Report

1. Dataset and Preprocessing

- Source: OpenWebText via Hugging Face dataset "dylanebert/openwebtext"

- Raw downloaded text size: slightly above 1 GB

- Number of raw documents loaded: 250,000

- Cleaning: remove empty / very short docs, normalize whitespace, drop obviously corrupted/boilerplate content

- Deduplication: remove duplicate or near-duplicate documents

- Documents kept after cleaning and deduplication: 250,000

- Tokenizer: GPT-2 tokenizer (AutoTokenizer.from_pretrained("gpt2"))

- Vocabulary size: 50,257

- pad_token: set to EOS token, pad_token_id = 50,256

- Sequence chunking: concatenate cleaned text and split into fixed-length blocks

- Block size: 128 tokens

- Total number of token sequences: 100,000

- Saved preprocessed dataset file: train_tokens_128.pt

2. Model Architecture (MiniGPT)

- Model type: small GPT-style Transformer for next-token prediction

- Token embedding layer: dimension d_model = 128

- Positional embedding layer: learned positions up to length 128

- Transformer encoder stack:

- Number of layers: 2

- Number of attention heads: 4 per layer

- Feed-forward dimension: 4 * d_model = 512

- Activation: GELU

- Dropout: 0.1

- Output layer: linear projection to vocabulary size (50,257)

- Approximate total trainable parameters: 13,328,977

3. Training Setup

- Training data subset: first 20,000 sequences from train_tokens_128.pt

- Sequence length: 128 tokens

- Task: next-token prediction (causal language modeling)

- Input sequence: tokens positions 0 .. L-2

- Label sequence: tokens positions 1 .. L-1

- Loss function: cross-entropy loss with ignore_index = pad_token_id

- Optimizer: Adam

- Learning rate: 1e-3 (baseline experiment)

- Batch size: 32

- Number of epochs: 2

- Hardware: CPU only (no GPU)

- Logging:

- Print loss every 100 training steps

- Compute average loss per epoch

- Compute perplexity per epoch as exp(average_loss)

- Checkpoint: save model state_dict to mini_gpt_full.pt at the end of training

4. Training Results (Baseline, lr = 1e-3)

- Dataset statistics:

- Total sequences in file: 100,000

- Sequences used for training: 20,000

- Model statistics:

- Using device: CPU

- Model parameters: 13,328,977

- Example step-level losses (Epoch 1):

- Step 100: loss = 7.2599

- Step 200: loss = 7.0102

- Step 300: loss = 4.8721

- Step 400: loss = 3.0098

- Step 500: loss = 1.9653

- Step 600: loss = 1.1757

- Epoch-level metrics:

- Epoch 1: avg_loss = 4.6811, perplexity = 107.89

- Epoch 2: avg_loss = 0.4957, perplexity = 1.64

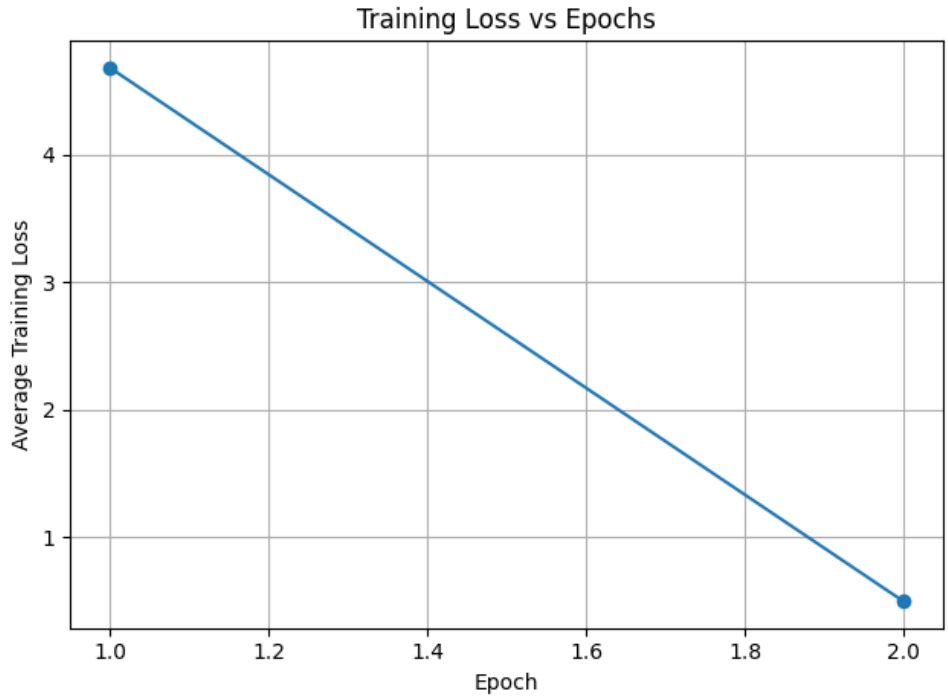
- Interpretation:

- Loss and perplexity drop quickly from epoch 1 to epoch 2

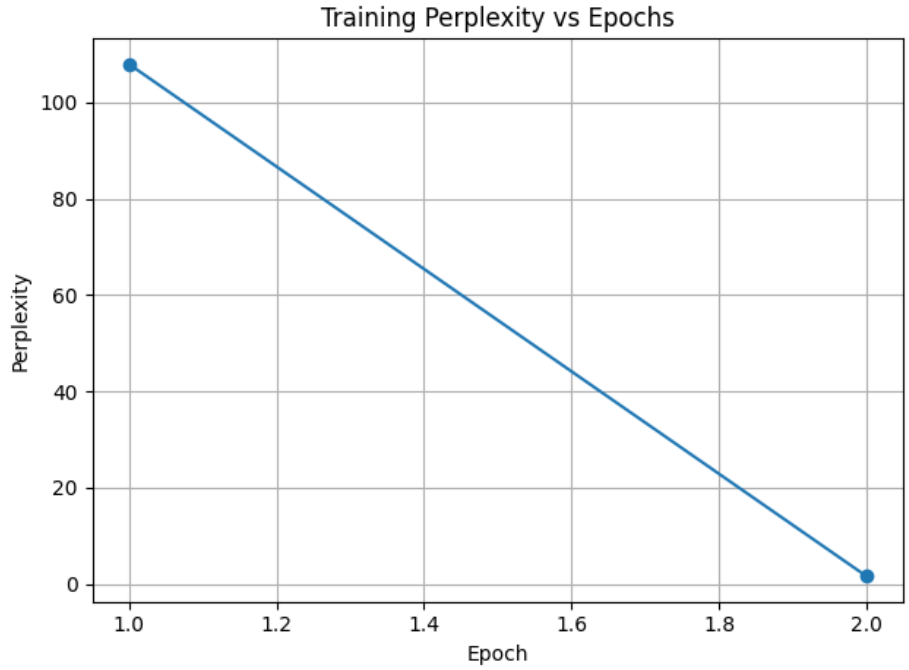
- Perplexity near 1.6 indicates strong fitting of the training subset

5. Visualizations

- Figure 1: Training loss vs. epochs



- Figure 2: Training perplexity vs. epochs



6. Observations and Challenges

- Model behavior:

- Fast convergence on the small training subset

- Very low final perplexity suggests possible overfitting to 20,000 sequences

- Data pipeline:

- Preprocessing from Assignment 1 (cleaning, deduplication, tokenization, sequence chunking) works end-to-end

- No major issues with pad_token_id, attention masks, or tensor shapes

- Practical constraints:

- Disk space limitations during dataset downloads ("no space left on device" warnings)

- CPU-only training, requiring a small model and a subset of the dataset

- Occasional library warnings (e.g., Jupyter progress bar backend), but no critical impact

7. Summary

- Implemented a small GPT-style Transformer (MiniGPT) for next-token prediction

- Reused and validated the Assignment 1 preprocessed dataset (train_tokens_128.pt)

- Trained on 20,000 sequences with lr = 1e-3, batch size 32, 2 epochs, CPU-only

- Logged and visualized training loss and perplexity

- Observed strong convergence and potential overfitting on the training subset

- Established a complete pipeline: raw web text -> preprocessed tokens -> language model training -> checkpoint + curves -> analysis