Assignment 2: Building a Small-Scale Foundation Model from Scratch

Student: Xiaoying Bao; **Course:** Infrastructure & DevOps with LLMs; **Date:** February 8, 2026

1. Model Architecture and Parameters

Implemented a transformer-based Mini-GPT for causal language modeling with next-token prediction.

Core Components

Multi-Head Self-Attention: 4 heads with scaled dot-product attention and causal masking to prevent attending to future tokens.

Feed-Forward Network: Hidden dimension 4× embedding size with GELU activation.

Architecture: Pre-layer normalization with residual connections, dropout (p=0.1) for regularization.

Model Specifications

Component	Exp 1&2	Exp 3
Parameters	13.3M	27.3M
Embedding dim	128	256
Layers	2	2
Attention heads	4	4
Vocabulary	50,257 (GPT-2)	50,257
Max sequence	128 tokens	128 tokens

2. Dataset Details

Data Source: Preprocessed corpus from Assignment 1

- Total samples: 2,473,387 tokenized blocks
- Total tokens: 316+ million
- Domains: Wikipedia, OpenWebText, C4

DataLoader:

- Batch size: 32 (Exp 1&2), 16 (Exp 3)
- Sequence length: 128 tokens
- Total batches: 77,294 (Exp 1&2), 154,587 (Exp 3)

3. Training Setup and Experiments

Configuration

Optimizer: AdamW (weight_decay=0.01, gradient clipping max_norm=1.0)

Loss: Cross-entropy with proper next-token shifting

Hardware: Google Colab Pro Tesla T4 GPU

Hyperparameter Experiments

Experiment 1 - Baseline

- Config: lr=1e-3, embed=128, batch=32
- Training: 2 epochs, 4.2 hours

Experiment 2 - Lower Learning Rate

- Config: lr=5e-4, embed=128, batch=32
- Training: 2 epochs, 4.3 hours

Experiment 3 - Wider Model

- Config: lr=5e-4, embed=256, batch=16
- Training: 2 epochs, 6.9 hours

Results

Experiment	Final Loss	Final Perplexity
Exp 1 (lr=1e-3)	5.0772	160.32
Exp 2 (lr=5e-4)	5.0670	158.69
Exp 3 (embed=256)	**4.8460**	**127.23** ← Best

Key Findings:

1. **Learning rate impact:** Lower LR (5e-4) showed marginal improvement over baseline (1e-3)
2. **Model capacity:** Doubling embedding dimension yielded 20% perplexity reduction, indicating capacity as primary bottleneck
3. **Training stability:** All experiments converged smoothly without divergence

4. Observations and Challenges

Challenge 1: Loss Calculation Bug

Issue: Initial implementation produced near-zero loss after first epoch.

Cause: Incorrect sequence alignment for next-token prediction.

Solution: Implemented proper shifting: `shift_logits = logits[:, :-1, :]` and `shift_targets = targets[:, 1:]`

Impact: Corrected implementation produced realistic loss values (4-5 range), validating training correctness.

Challenge 2: GPU Acceleration

Issue: CPU training required 13+ hours per epoch.

Solution: Migrated to Colab T4 GPU, achieving 8× speedup (2 hours per epoch).

Learning: GPU essential for transformer training; resource selection critically impacts feasibility.

Challenge 3: Model Size vs Speed Trade-off

Observation: Experiment 3 (27M params) required 60% more time but achieved 20% better perplexity.

Analysis: Performance gains justified additional compute cost for this dataset scale.

Limitations

No Validation Set: All data used for training; cannot assess true generalization. Production systems require train/val/test split with early stopping.

Limited Epochs: Trained 2 epochs due to time constraints; full convergence may require 5-10 epochs.

Future Improvements: Implement validation monitoring, learning rate scheduling, and comprehensive ablation studies.

5. Conclusion

Successfully implemented Mini-GPT transformer achieving perplexity of 127.23 on 2.4M training samples. Hyperparameter experiments demonstrated that model capacity (embedding dimension) impacts performance more significantly than learning rate for this dataset scale. The project validates understanding of transformer architecture, provides hands-on experience with GPU-accelerated training, and establishes foundation for production ML system development.

Key achievements: Correct transformer implementation, systematic experimentation, efficient training pipeline utilizing cloud GPU resources.