LLM Homework Response Analysis Report

EECS 182 Deep Neural Networks - Homework 11

This report analyzes an LLM's responses to homework questions, comparing them against the official answer key. Each question includes the LLM's reasoning, final answer, and an assessment of correctness.

Problem 1: LoRA (Low-Rank Adaptation)

Part (a): Addressing Underfitting in LoRA

LLM's Response: The LLM identified that underfitting indicates insufficient model capacity. The key insight was that the rank k of the low-rank matrices A and B controls expressiveness, since the product AB has rank at most k.

LLM's Final Answer:

- Primary solution: Increase the rank k of the LoRA adaptation
- Additional suggestions: Apply LoRA to more layers, increase learning rate

Answer Key Response: The answer key lists three strategies: (1) Increase rank k to allow more expressive power, (2) Change initialization using SVD of the original weight matrix or gradient information, (3) Adjust learning rate or other hyperparameters (referencing LoRA+ paper).

✓ **CORRECT** - The LLM captured the primary solution (increasing rank) and mentioned learning rate adjustment.

Part (b): Zero Initialization Problem

LLM's Response: The LLM analyzed gradient flow with A = 0 and B = 0, showing that both gradients will be zero, preventing any parameter updates. The model will never adapt from W_0.

✓ **CORRECT** - The LLM's mathematical analysis is rigorous and matches the answer key's reasoning.

Part (c): Xavier Initialization Problem

LLM's Response: The LLM explained that Xavier initialization produces non-zero values in both A and B, meaning AB ≠ 0 at initialization. This causes the effective weights W_0 + AB to immediately differ from pretrained weights, destroying the pretrained representations.

✓ **CORRECT** - The LLM correctly identified the problem and the standard solution.

Part (d): Memory Requirements

LLM's Response: LoRA parameters: mk + kl = k(m + l). LoRA memory: 4k(m + l) bytes. Full-rank memory: 4ml bytes. LoRA is much more efficient when k << min(m, l).

✓ **CORRECT** - Identical formulas to the answer key.

Problem 2: Transformer Interpretability

Part (a)(i): Zero-Layer Model Logits

LLM's Final Answer: L = W_U * W_E * T

✓ **CORRECT** - Identical to answer key.

Part (a)(ii): Algorithm Interpretation

LLM's Response: The model implements a bigram/unigram prediction model where each position's prediction depends only on the token at that position itself.

✓ **CORRECT** - Same interpretation as answer key (simple position-wise word lookup).

Part (b)(i): Multi-Head Attention Equivalence

LLM's Response: Showed that partitioning W_O^H into blocks column-wise gives the correct equivalence through block matrix multiplication.

✓ **CORRECT** - Identical construction and proof.

Part (b)(ii): Advantages and Disadvantages

View	Advantage	Disadvantage
Concatenation	Computationally efficient (single large matmul)	Obscures independent head contributions
Additive	Interpretable; shows heads contribute independently	Less efficient with multiple matmuls

✓ **CORRECT** - Same advantages/disadvantages identified.

Part (c)(i): QK Circuit Derivation

LLM's Final Answer: S_ij = X_i^T * W_QK * X_j where W_QK = W_Q^T * W_K

✓ **CORRECT** - Identical derivation and result.

Part (c)(ii): QK Matrix Interpretations

LLM's Answers (all correct):

1. W_QK = I: Looks for similar tokens (dot product similarity)
2. W_QK = diag(1,0,0): Only cares about first dimension; dimensions 2-3 are ignored
3. W_QK = diag(1,1,-1): Rewards similarity in dims 1-2, dissimilarity in dim 3
4. W_QK = -I: Looks for dissimilar/opposite tokens

✓ **CORRECT** - All interpretations match exactly.

Parts (d)(i)-(d)(iii): Attention-Weighted Averages

✓ **CORRECT** - All three sub-parts correct: attention-weighted average derivation, final residual stream expression, and column space proof.

Parts (e)(i)-(e)(v): SVD Analysis of OV Circuit

✓ **CORRECT** - All five sub-parts correct: rank bound, SVD decomposition, read/write subspaces, value projection connection, and synthesis.

Problem 5: Fermi Estimation

Part	Key Result	Status
(b) Chinchilla Scaling	N_{opt} ≈ 0.60 C^{0.452} D_{opt} ≈ 0.27 C^{0.548}	✓
(c) 100T Parameters	C = 10³⁰ FLOPs D = 1.7x10¹⁵ tokens	✓
(d) Dataset Size	~6 billion books ~350x Library of Congress	✓
(e) Memory (GPT-6)	200 TB ~2000 H200 GPUs	✓
(f) Memory Cost	SSD: ~$10,000/year DRAM: ~$800,000/year	✓
(g) Latency/Bandwidth	GPT-6: ~41.7 sec/token GPT-3: ~73 ms/token	✓
(g) Activation Memory	~9 MB/token (LLM) vs 2.4 MB/token (Key)	■
(h) Training Cost	GPT-6: ~$210 billion ~195 days with 30M GPUs	✓
(j) Inference Cost	$0.42 vs $2.78 discrepancy	■
(k) Landauer Limit	3.5 million x theoretical min	✓
(l) Environmental	~$62k carbon cost (~1%)	✓

Detailed Error Analysis

Error 1: Problem 5(g) - Activation Memory Calculation

- LLM's Answer: ~9 MB per token, ~39,000 tokens for batch
- Correct Answer: ~2.4 MB per token, ~150,000 tokens for batch

What Went Wrong: The LLM correctly identified the formula (96 layers x 96 heads x 128 dimension), but appears to have included additional activation storage (perhaps Q, K, V, and output) without clearly stating this,

arriving at roughly 4x the answer key's value. The correct calculation is: 96 x 96 x 128 = 1,179,648 floating point values; at 16-bit (2 bytes): 1,179,648 x 2 = 2,359,296 bytes (approximately) 2.4 MB.

Error 2: Problem 5(j) - Inference Cost Calculation

- LLM's Answer: $0.42 for 1M tokens
- Correct Answer: ~$2.78 for 1M tokens

What Went Wrong: The discrepancy stems from different interpretations of H200 throughput. The LLM used the theoretical peak of 1.98 PF/s, while the answer key uses 1.08×10^{18} FLOPs per hour (which equals 0.3 PF/s), suggesting a ~30% utilization assumption. Both approaches are valid given different assumptions, but the answer key's approach accounts for realistic utilization.

Problem 6: Soft-Prompting Language Models

Part	Question	LLM Answer	Status
(a)	Loss computation tokens	Positions 50-71 (reasoning, answer, newline)	✓
(b)	Trainable parameters	5E parameters (5 soft prompt vectors)	✓
(c)(i)	Precompute representations?	TRUE - causal masking allows precomputation	✓
(c)(ii)	Soft vs hard prompt performance?	TRUE - soft prompt strictly contains hard prompt	✓
(c)(iii)	Full finetuning better?	FALSE - overfitting risk with more capacity	✓
(c)(iv)	Catastrophic forgetting?	FALSE - base weights remain frozen	✓
(d)	MAML adaptation	Learn initialization P_0 for soft prompt vectors	✓

Part (d) Details: The LLM correctly explained that MAML would learn a meta-learned initialization P_0 for soft prompt vectors. At meta-training time, sample tasks, adapt P_0 with gradient steps on support data, evaluate on query data, and update P_0 through the inner loop. At test time, initialize from P_0* and take gradient steps on the new task.

Overall Performance Summary

Metric	Count	Percentage
Total Questions Analyzed	34	100%
Fully Correct	32	94.1%
Partially Incorrect	2	5.9%

Key Findings

- **Strong Theoretical Performance:** The LLM demonstrated excellent performance on theoretical questions involving mathematical derivations and conceptual understanding (Problems 1, 2, and 6), achieving 100% correctness.
- **Numerical Estimation Challenges:** The two errors occurred in numerical estimation questions (Problem 5) where assumptions about real-world parameters led to different values than the answer key.
- **Error Types:** Both errors involved interpretation of system specifications: (1) activation memory accounting (whether to include all intermediate activations), and (2) GPU utilization assumptions (theoretical peak vs. realistic throughput).
- **Mathematical Rigor:** The LLM consistently provided rigorous mathematical derivations with proper notation and step-by-step reasoning, matching the answer key's approach.
- **Conceptual Understanding:** Strong grasp of core concepts including LoRA adaptation, transformer interpretability, attention mechanisms, and meta-learning was demonstrated throughout.

Conclusion

The LLM performed exceptionally well on this advanced deep learning homework, demonstrating strong theoretical knowledge and mathematical reasoning capabilities. The 94.1% accuracy rate across 34 questions, with errors limited to numerical estimation under ambiguous assumptions, indicates robust understanding of transformer architectures, parameter-efficient fine-tuning, and computational scaling laws. The LLM's ability to provide detailed mathematical derivations and conceptual explanations that match expert-level answer keys suggests it could be a valuable tool for learning and verification in advanced technical coursework.