MatSoc, IITK

Architecting Intelligence

Assignment 3

Mentors: Basudev Mohapatra
Khushal Wadhwa
Mansi Ghodke

Due Date: January 12th, 2026

Total Marks: 40

Questions (10 Marks Each)

Q1. **Why Encoders Got Fired?**

Transformer architectures can be broadly categorized into Encoder-Only, Encoder–Decoder, and Decoder-Only models, each originally designed for different classes of language tasks. Encoder–Decoder Transformers were first introduced for Sequence-to-Sequence problems such as Machine Translation, while Encoder-Only Models focus on representation learning through bidirectional context. With the rise of Large Language Models trained on **massive corpora**, architectural choices have become closely tied to training efficiency, inference cost, and scalability to long sequences. As models grow in parameter count and context length, design decisions related to Attention Mechanisms, information flow, and computational reuse become increasingly important, motivating comparisons between different Transformer variants.

Conceptually, why is a **Decoder-only Transformer more scalable** for LLMs than Encoder–Decoder architectures?

Q2. **Weaponized Autocomplete**

Language Modeling is commonly formulated as a **probabilistic task** over sequences of tokens, where models learn statistical patterns from large-scale text data. One widely used objective in this setting is **Next-Token Prediction**, which trains a model to estimate the probability distribution of possible continuations given a preceding context. This objective has been adopted across many modern Language Models, regardless of their downstream applications. At the same time, **Natural Language Processing** encompasses a wide range of tasks—such as Translation, Summarization, and Reasoning—that appear, at first glance, to require specialized supervision or task-specific objectives. Understanding the relationship between these diverse tasks and a single training objective is central to the theory behind Large Language Models.

Why is **Next-Token Prediction** sufficient to learn complex language abilities such as reasoning, translation, and summarization?

Q3. **Honey, I Shrunk the Matrix**

This problem explores how a high-dimensional matrix can be factorized into two smaller, low-rank matrices using **Singular Value Decomposition (SVD)** without losing information. This concept is the mathematical foundation of **Low-Rank Adaptation (LoRA)**, demonstrating how large neural network weights can be efficiently approximated or updated using significantly fewer trainable parameters.

Make a copy of the following notebook and start exploring!!
Honey, I Shrunk the Matrix

Q4. **Train Less, Brag More: LoRA Edition**

Modern Deep Learning models often contain millions of parameters, making full fine-tuning computationally expensive and memory intensive. **Low-Rank Adaptation (LoRA) is a parameter-efficient fine-tuning technique** that updates only a small number of additional trainable parameters while keeping the original model weights frozen. **In this assignment, you will first train a baseline neural network for handwritten digit classification using the MNIST dataset**. You will then analyze its weaknesses and implement LoRA to improve performance on difficult classes. By completing the provided code blocks, you will gain hands-on experience with efficient fine-tuning and model adaptation in PyTorch.

Make a copy of the following notebook and start!
Train Less, Brag More