26CS801 **Design of Large Language Models** **3-0-2 4**

Learning Objectives

This course is intended to provide a comprehensive understanding of the transformer-based design of modern Large Language Models. In this course, we will cover mathematical foundations and the design and implementation of LLMs as well as training and fine-tuning of the models. An integral part of the course will be the implementation of an LLM from scratch.

Syllabus

1. Review of Neural Networks and their training

- Vector, matrix and tensor operations
- Review of key component — layers and activation functions
- Simple networks — logistic regression and multi-class classification
- Regularization techniques — L1/L2 regularization and dropout
- Gradient descent and back propagation
- Training concepts — learning rate, momentum
- Normalization — batch and layer normalizations
- Attention mechanisms and transformers
- High-level overview of LLMs
- Readings: Handout

2. Representing text: Tokenization, Embeddings and Encodings

- Word2Vec
- Tokenization of text
- Encoding schemes— BPE (byte-pair encoding)
- Positional Encoding
- **Readings**
  - Mikolov et al Efficient Estimation of Word Representations in Vector Space
  - Raschka Chapter 2

Assignment: Implementing a basic text processing pipeline

3. Transformers

- The rationale for an attention mechanism
- Dot product attention
- Multiple heads
- Self-attention and causal attention

- Assignment: Implementing a basic self-attention mechanism from scratch

4. Implementing a GPT model from scratch to generate text

- The transformer block
- Residual connections
- Layer normalization: theory and purpose
- The GPT-2 architecture
- Alternative architectures — encoder-only, decoder-only and encoder-decoder models
- Lab: Building a basic transformer block
- Readings:
  - Raschka chapter 4
  - Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

5. Pretraining

- Causal Language Modeling
- Loss functions
- Training protocol and monitoring
- Decoding strategies
- Lab: Load GPT-2 weights and continue pretraining on a corpus
- Reading:
  - Raschka Chapter 5

6. Finetuning

- Instruction fine-tuning and classification fine-tuning
- Self-instruct & the Alpaca method
- Training Efficiency o Parameter-efficient Finetuning
- LORA
- Quantization
- Readings:
  - Raschka Chapters 6 & 7

7. Alignment

- The importance of aligning AI behaviour with human values and intentions
- Intro to reinforcement learning from human feedback (RLHF)
- Direct Preference Optimization (DPO)
- Lab: Implement DPO
- Readings:
  - Christiano et al. (2017) “Deep Reinforcement Learning from Human Preferences”

- Rafailov et al Direct Preference Optimization: Your Language Model is Secretly a Reward Model

Course outcomes

CO1 - Understand the principles underlying LLM architectures

CO2 - Develop an understanding of methods for training LLMs

CO3 - Gain an in-depth understanding of LLM implementations through building one from scratch

CO4 - Evaluate and compare different LLM architectures and their performance characteristics across various applications and use cases

CO5 - Apply best practices for prompt engineering, fine-tuning, and responsible deployment of LLMs in real-world applications

Evaluation pattern

Components	Continuous Assessment (Assignments/Presentation)	Internal Examination	Written Examination at the end of the semester
Weightage %	20	30	50

Course Material

- Raschka, Sebastian Build a Large Language Model (From Scratch), 2024, Manning Press. Papers
- Christiano et al. (2017) “Deep Reinforcement Learning from Human Preferences”
- Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
- Gholami et al. A Survey of Quantization Methods for Efficient Neural Network Inference
- Lu et al. Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment
- Mikolov et al Efficient Estimation of Word Representations in Vector Space
- Rafailov et al Direct Preference Optimization: Your Language Model is Secretly a Reward Model
- Wang et al. Self-Instruct: Aligning Language Models with Self-Generated Instructions

Total 60 hours – 4 credit course

Evaluation pattern:

	Weightage	Component
Internal	70	Attendance, participation, weekly assignments
External	30	End-semester exam paper
Total	100	