**Summer Training Report  
Week 1**

**Exploring LLMs, Transformers, SQL-to-LLM Conversion, and Parameter-Efficient Fine-Tuning**

**Submitted by**

Name: Mahmoud Mustafa Oushi

University ID: 4312239

University: University of Prince Muqrin

Department: College of Computer Science and Cyber Security

Advisor: Dr.Ahmed Elhayek

**Training Organization**

Hulool Al Sahabah for IT & Communications Co

AI Department

**Report Topics – Week 1**

* Introduction to Large Language Models (LLMs)
* Transformer Architecture
* Data Functions and Acquisition
* Converting SQL Data to LLM-Compatible Formats
* Parameter-Efficient Fine-Tuning (PEFT)
* Fine-Tuning Strategies for LLMs

**Abstract**

The task for this week was to get familiar with the concept of Large Language Models (LLM), and get a closer look into the functionality and the goal behind them. Get some knowledge about the Transformers architecture, as well as some other related fine tuning methods such as Parameter-efficient fine tuning (PEFT) and the techniques related to it such as LoRA and QLoRA. After that playing with the idea of handling data in the SQL domain and finding ways to convert them into LLMs suitable formats. To be able to pull that off with much personal and professional gain as possible, online courses offered by Deeplearning.AI and aws, as well as research papers with the likes of “Attention is all you need” were the main sources of information to meet the given tasks.

**1. Introduction**

The current task while writing this report is to get accustomed with the concept of LLMs, but the term LLM is not as direct as one may think. Think more of a foundation model that is trained then prompted to many tasks and fields as a standalone mean such as, NLP (text summarization, NER, Q&A, text translation and classification), can also be used in generative tasks such as text generation (stories and chats), Code (GitHub copilot), and even paired with other models like using prompts for imagery and visualization (Dall-E).

So, what I chose to do for this task is to look at the generative aspects of LLMs for tasks like fine tuning the models to classify the sentiments as well as summary generation using means like zero, one, and few shot inferences, as well as re-parameterization methods like LoRA (Low Rank Adaptation) mainly on the FLAN-T5 model.

**2. Transformer Architectures**

The Transformer Architectures is a type of neural network that to put it in simple terms has the advantage of state and process all inputs at once, through what is known as self-attention which allows them to process sequences in parallel unlike standard RRNs which does it in sequential order, as well as other methods which will be explained in a bit. LLMs utilize this architecture by scaling it to train models with billions of parameters enabling them to handle a large range of language related tasks such as text classification, generation, translation and summarization to name a few.

**2.1. Self & Multi head-Attention, Positional Encoding and Feed Forward**

Now let us take a look at what makes the Transformer as useful and powerful as we came to know it.

Self attention is what allows each token to be weighted against other tokens as a way to know importance in relative to context. By computing attention scores between Query which is like what’s the word looking for, and Key which is what does it offer, and the value which is the actual content. Using the Query and the Key, compute the dot product, then convert to probabilities using softmax. Multiply each score weight with its Value vectors giving us the score of each word. self-attention allows the model to capture long-range dependencies and context, regardless of word position. This is a fundamental reason why Transformers outperform previous models like RNNs in handling complex language tasks. And usually present in the architecture of the Transformer is what’s known as multi head attention which is what’s been described until now but running in parallel with multiple weight matrices which allows the model to be able to process different types of contextual relationships.

Another component is Positional Encoding which is like the models mean of knowing the order of words. Each word position gets mapped to it’s own unique vector and added to the word embedding.

Also another thing present in the architecture is the Feed Forward network after the self attention to add none linearity.

**3. PEFT (LoRA & QLoRA)**

To utilize a model with billions of parameters you will need one powerhouse of a machine, but what if you are a consumer level developer who is trying to dip his toes and play around? One way to do so is to use Parameter Efficient Fine Tuning or PEFT for short. The idea is to freeze the model parameters and utilize what suits your machine and needs, sparing the time and effort as well as the computational power.

The way that is done is by introducing Two low rank matrices that when multiplied, will result in a matrix with the same dimensions as the one of frozen parameters. Then adding the two together introduces a new smaller learn able component to fine tune without changing most of the weights.

Now the concept of QLoRA is pretty much the same except that it introduces the addition of 4bit formatting, unlike LoRA which goes with FP16/BF16, what that does is that it basically does what is known as quantization which is reducing the precision of the model parameters to 4bits which helps to dramatically reduce the size of taken storage. Afterwards normal LoRA is applied on top with the rest of the procedure being fairly the same.

**4. SQL to LLM Formatting**

**4.1. Pandas & SQL-Lite3**

The first and what I found to be the easiest way to convert SQL formatted data to natural language settings required for LLMs is through utilizing the SQL lite and pandas ability to extract the queries and saving then in dictionaries then it is a matter of editing the sentiments according to needs. For example if we have a table with the cols [Name, Age Salary] and however many rows, using the **{.apply (lambda row: …., axis = 1}**, where the dots will be a sort of in between filling decided by you. This will extract the row info from the table, Blend them with the filling in between each and finally gives us a natural setting sentences or text blocks ready to be saved in a JSON format and then used by the LLM.

**4.2. LangChain**

**To be researched still**

**5. Course by DeepLearning.AI & aws**

The most helpful source of information during this stage was the “Generative AI and Large Language Models” course provided by Deeplearning.AI and aws through the coursera platform. The course is self based and divided into three main modules with each focusing on a specific topic. Starting with module 1 which was an introduction to the concept of LLMs and Gen-AI, then a deep dive into the Transformer Architecture and its functionality and relation to the presented topic. Then a look at prompt engineering, as well as the concepts of zero, one, and few shots. The Lab of this module consisted of using the FLAN-T5 model and adjusting it to provide summarization of a given multi samples of texts and then comparing it to human made summary.

Module 2 was all about PEFT techniques and specifically LoRA and prompt tuning. As well as measures like BLEU and ROUGE score for quality evaluation. And a cool concept called chinchilla scaling law which was developed by the research team at Google DeepMind in 2022. What it suggests is that models should be smaller but trained on more data, a number that reappeared in the course more than once was 20x the weights of the model.  
  
Module 3 is mostly just about the ethics of usage and RLHF, which is by the time of writing the reports is yet to finished for the acquiring of the certification. Should be easy enough though, and hopefully fun as well.

**References**

A. Ng, "Generative AI with Large Language Models," DeepLearning.AI, [Online]. Available: <https://www.deeplearning.ai/courses/generative-ai-with-llms/>. [Accessed: Jul. 2, 2025].

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," arXiv preprint arXiv:1706.03762, Jun. 2017. [Online]. Available: <https://arxiv.org/abs/1706.03762>

J. Hoffmann, S. Borgeaud, A. Mensch, et al., "Training compute-optimal large language models," Advances in Neural Information Processing Systems, vol. 36, 2022. [Online]. Available: <https://arxiv.org/abs/2203.15556>