**Slide1:**

While we are getting started, I would love to know what brings you to this workshop.

**Those were some interesting answers.**

**Schedule:**

0:05 - 0:35 Learning about LLMs and their working.

0:35 - 0:50 Fine-tuning LLMs for your tasks

0:50 - 1:00: Ethical AI

**Slide 2:**

But before going into more details I would like to begin by acknowledging that I .

**Slide 3:**

Active participation makes the session so much fun and gives me and your peers much more energy. Your voices and perspectives enlivens the session. We encourage you to engage with each other and us.

The participants window lists everyone in the session and click the icons at the bottom to communicate with the us.

You can also use the Chat windows to comment or ask questions at any time. It is also a good place to share problems with your audio connection.

**Slide 4:**

So, to touch various viewpoints of machine learning regression, we have the following learning objectives for this workshop:

**Slide 5:**

For hands-on exercises, we will use [Python](https://www.python.org/) on [Jupyter Notebooks](https://jupyter.org/). You don’t need to have Python installed. Please make sure that you have a [UBC Syzygy](https://ubc.syzygy.ca/) or a [Google Colaboratory](https://colab.research.google.com/) account. (You will need a CWL login to access Syzygy.) hands-on exercises, programming tools and libraries, such as [Python] and [scikit-learn] prior familiarity with Python programming is recommended, we do not study the codes in detail

**Slide 6:**  
Before diving into LLMs, let’s quickly go over some basics that make LLMs, LLMs.

* Neural Networks: A type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain.
  + RNN: a type of neural network designed to work with sequence data. Explore LSTMs and GRUs, two RNN variants that are capable of learning long-term dependencies.
* Natural Language Proceesing: A branch of AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.
  + Text Preprocessing: Tokenization (splitting text into words or sentences), stemming (reducing words to their root form), lemmatization (similar to stemming but considers the context), stop word removal.
  + Feature Extraction Techniques: Converting text data into a format that can be understood by machine learning algorithms. Key methods include Bag-of-words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and n-grams.
  + Word Embeddings: Word embeddings are a type of word representation that allows words with similar meanings to have similar representations. Key methods include Word2Vec, GloVe, and FastText.

**Slide 7:**

* LLMs: Read out definition

**Slide 8:**

Let’s pretend. A lot of you might have already used chatGPT. So it’s not really something different.

**Slide 9:**

Previous one was an easier task, but how about this one?

Even for such tasks, that needs a lot of time for us to do. LLMs can do it in few seconds.

May not be the best job as a human. But quicker. A person using it still need domain knowledge to understand the code and fix few things.

**Slide 10:**  
Some of the other applications of LLMs are:

**Slide 11:**Now lets’s understand the architecture of LLMs.

* **Encoder-decoder transformer**
  + **Encoder**: Think of an encoder like someone who reads a story and takes notes on important details. In the context of a Language Model, the encoder takes a sentence or a piece of text as input and processes it to understand the key information. It's like breaking down a sentence into smaller parts and figuring out the main ideas.
  + **Decoder**: Now, imagine a decoder as someone who uses those notes to tell a story in a different language. In the Language Model, the decoder takes the encoded information and generates a new piece of text. It's like translating the summarized notes into a different language or context.
  + **Summary**: Together, they make sure the story is understood and shared in a new way. So, in summary, an encoder processes information, and a decoder uses that information to generate something new, forming a crucial part of how Language Models work.
* Tokenization
  + Tokenization is the process of breaking down text into smaller units called tokens.
  + Tokens can be words, subwords, or even characters, depending on the chosen tokenization strategy.
  + This step is crucial for the model to understand and process the input text efficiently.
* Attention
  + Think of a classroom where a teacher is explaining a lesson. The teacher pays attention to different students at different times based on their needs or questions. Similarly, the attention mechanism in a transformer allows the model to focus on different parts of a sentence or text, giving importance to specific words based on the context.
  + Components of Attention:
    - Query: Think of it as the question the student asks.
    - Key: Think of it as the student who has the answer.
    - Value: Think of it as the actual answer the student provides.
* Text generation
  + LLMs utilize the decoder part of the transformer architecture for text generation, sampling or selecting tokens sequentially.

**Slide 12:**

That was about the LLM model’s internal architecture and working. Let’s now look at how they are trained and how they work in production, etc.

A LLM application as a whole consists of more elements than just the LLM model:

* There is the agent you interact with
* There are prompts that you give

**Slide 13:**

BERT:

BERT, developed by Google, is a cutting-edge natural language processing model that understands context by considering both preceding and following words, significantly enhancing the representation of word meanings in a bidirectional manner.

GPT (Generative Pre-trained Transformer) (by OpenAI):

* GPT, pioneered by OpenAI, is a state-of-the-art language model that utilizes a transformer architecture, capable of generating coherent and contextually relevant text. It demonstrates powerful language understanding and generation through pre-training on diverse datasets.

LLaMA (by Meta):

* LLaMA, developed by Meta, is an acronym for Language Model for Many Applications. Although specific details may vary, it is an initiative focused on creating versatile language models that can be applied across a broad range of tasks and applications within the Meta ecosystem.

**Slide 14:**

* **Pre-training:** Initially, a language model is trained on a massive amount of diverse and general language data. This phase helps the model learn the intricacies of language, grammar, and contextual understanding. Models like BERT, GPT, and LLaMA go through this extensive pre-training to capture a broad understanding of language.
* **Fine-tuning:** After pre-training, the model can be fine-tuned for specific tasks or domains. This involves training the model on a smaller, task-specific dataset. The goal is to adapt the pre-trained model to the nuances and characteristics of the target task, improving its performance on that particular application.

Fine-tuning allows leveraging the general knowledge acquired during pre-training while tailoring the model to perform exceptionally well on targeted tasks, making LLMs versatile tools for a wide array of natural language processing applications.

**LoRA: Low Rank Adaptation of LLM**

Helps in efficiently fine tuning LLM with lesser computation requirements.

Reduces the number of parameters that require fine-tuning.

For content for the Ethics slides use <https://ubc-library-rc.github.io/llm/content/introduction.html>