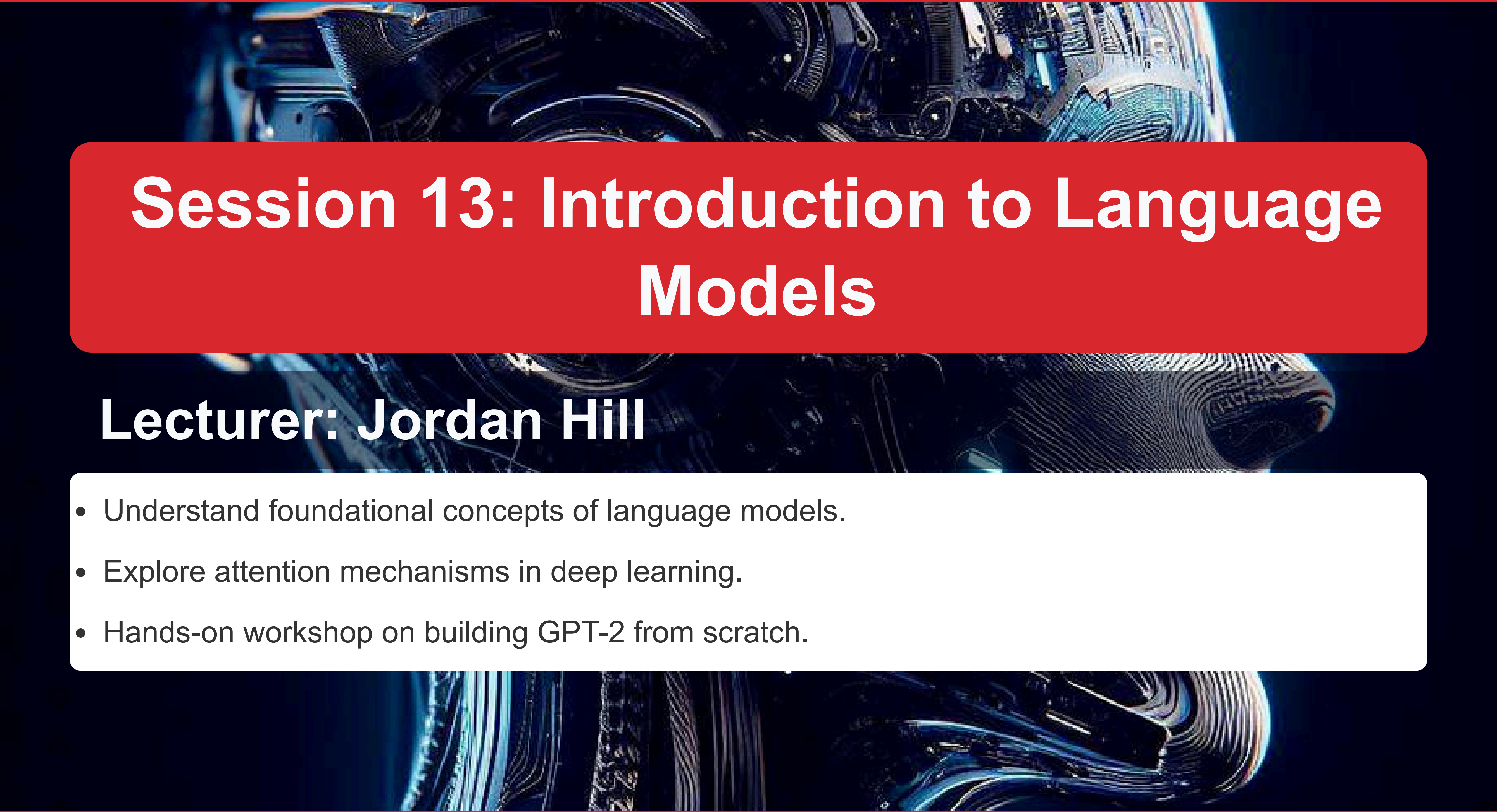


Session 13: Introduction to Language Models

Lecturer: Jordan Hill

- Understand foundational concepts of language models.
- Explore attention mechanisms in deep learning.
- Hands-on workshop on building GPT-2 from scratch.

Learning Objectives

- Grasp the fundamentals of modern language models.
- Introduce the core ideas of the attention mechanism step by step.
- Gain practical experience with GPT architectures by building GPT-2.
- Understand LoRA and its role in fine-tuning large language models.
- Discuss challenges associated with model sizes and explore solutions.

Introduction to Language Models

- **Language Models (LMs)**: Models that learn the probability of sequences of words.
- **Popular Models**: LLaMA, GPT-2, GPT-3.
- **Applications**: Text generation, translation, sentiment analysis.

Language models enable machines to understand and generate human language by predicting the next word in a sequence.

The Need for Attention

- Traditional models struggle with long sequences and context.
- **Challenge**: Maintaining relevant information over long distances in text.
- **Solution**: Introduce mechanisms that allow models to focus on important parts of the input.

Introducing the Attention Mechanism

Intuition Behind Attention

- Mimics cognitive attention in humans.
- Allows models to dynamically highlight relevant information.
- Improves handling of dependencies in sequences.

Core Concepts of Attention

Step 1: Understanding Key Components

- **Query (Q)**: Represents the current word we're focusing on.
- **Key (K)**: Represents all words in the input sequence.
- **Value (V)**: Holds the information of each word.

Step 2: Calculating Attention Scores

- **Attention Score**: Measures the similarity between the Query and each Key.
- Computed using dot products and scaling factors.
- **Softmax Function**: Converts scores into probabilities.

Step 3: Generating the Weighted Sum

- **Weighted Sum**: Combines the Values, weighted by the attention scores.
- Emphasizes important words while diminishing less relevant ones.
- Resulting vector captures context relevant to the Query.

Types of Attention Mechanisms

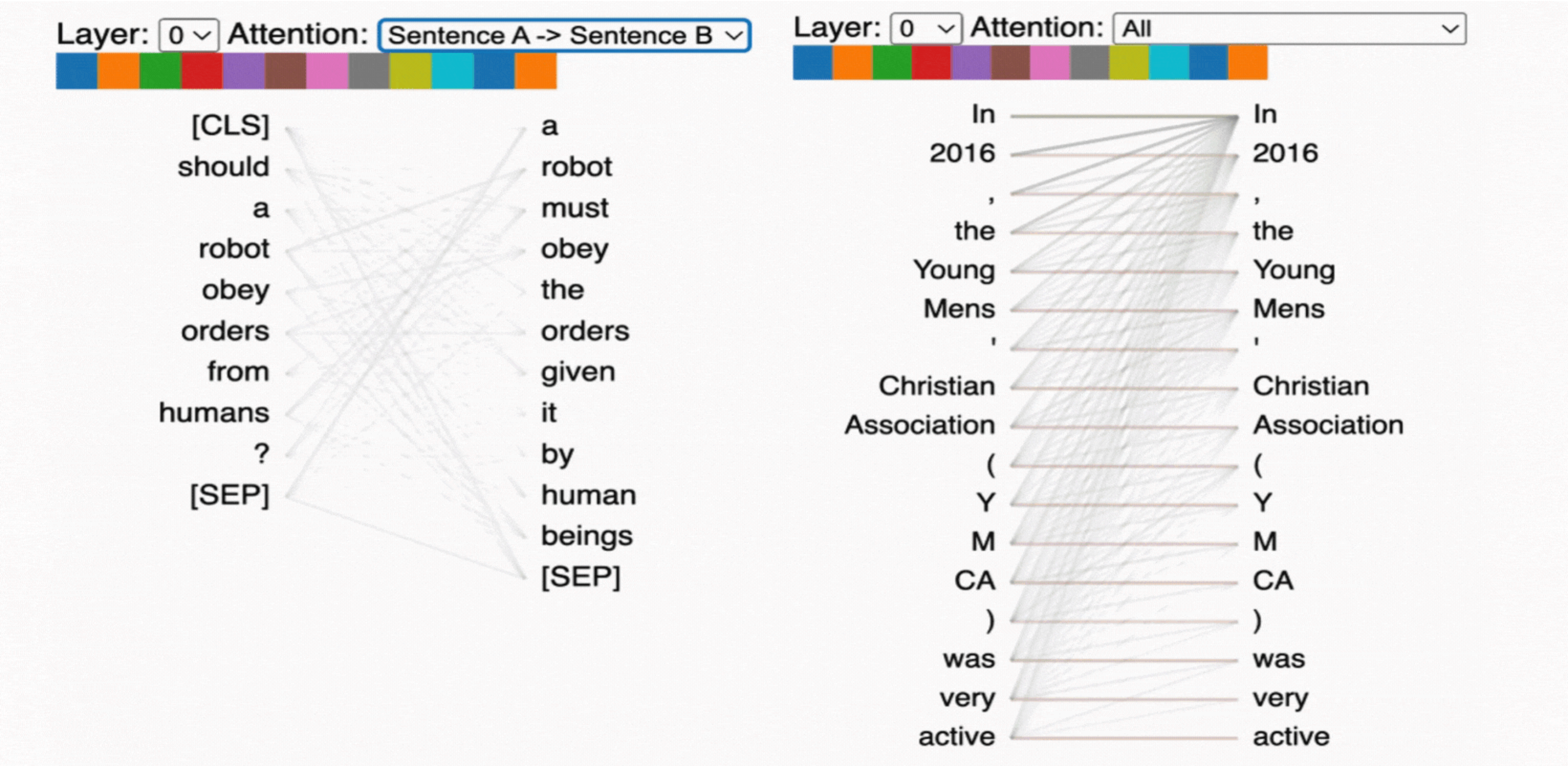
Self-Attention

- **Definition**: The model attends to different positions within the same sequence.
- **Purpose**: Captures dependencies between all words in the input.

Cross-Attention

- **Definition**: The model attends to a different sequence (e.g., during translation).
- **Purpose**: Aligns and relates information from two distinct sequences.

Visualization of the Attention Mechanism



Layer: 0 Attention: Sentence A -> Sentence B

Layer: 0 Attention: All

```
<iframe width="100%" height="580" src="https://www.youtube.com/embed/eMlx5fFNoYc" title="Attention Mechanism" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" referrerpolicy="strict-origin-when-cross-origin" allowfullscreen></iframe>
```

Benefits of the Attention Mechanism

- **Improved Context Understanding**: Models can focus on relevant parts of the input.
- **Handling Long Sequences**: Effectively manages dependencies over long distances.
- **Parallelization**: Allows for more efficient training compared to recurrent models.

Applications of Attention

- **Natural Language Processing**: Machine translation, text summarization.
- **Computer Vision**: Image captioning, visual question answering.
- **Speech Recognition**: Focuses on relevant segments of audio input.

Break

That was a lot, lets take a 30 min break

Model Sizes

- **Model Scale** :
  - Small, Medium, Large, and Extra-Large models.

Quantization

Quantization represents data with fewer bits, making it a useful technique for reducing memory-usage and accelerating inference

The basic idea behind quantization is quite easy: going from **high-precision representation** (usually the regular 32-bit floating-point) for weights and activations to a **lower precision** data type. The most common lower precision data types are:

> The two most common quantization cases are float32 -> float16 and float32 -> int8.

Math People: see here for an in-depth explanation

Activity: Behind the Pipeline

Behind the Pipeline: Huggingface LLM course: Chapter 2

Kaggle Notebook: Working with the model and tokenizer

Working with pytorch tensors: Tensors — Pytorch

Kaggle Notebook: Pytorch Tensor Tutorial

Extension Activities

1. **Introduction to GPT-2 Architecture**
    - Understand how transformers utilize attention mechanisms.
    - Explore the structure of GPT-2 and its components.
2. **Code-Along Session**
    - Follow along with the coding demonstration.
    - Build the model step by step.
3. **Experimentation**
    - Generate text samples.
    - Do some fine-tuning

Key Concepts of GPT-2

- **Transformer Architecture**: Relies heavily on self-attention mechanisms.
- **Layer Stacking**: Multiple layers allow for capturing complex patterns.
- **Fine-Tuning**: Customize the model for specific tasks or styles.

By constructing GPT-2, you'll gain hands-on experience with attention mechanisms in practice.

```
<iframe width="100%" height="580" src="https://www.youtube.com/embed/kCc8FmEb1nY"
title="Building GPT-2 from Scratch" frameborder="0" allow="accelerometer; autoplay;
clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
referrerpolicy="strict-origin-when-cross-origin" allowfullscreen></iframe>
```

Summary and Q&A

Today we covered:

- Fundamentals of language models.
- Core ideas of the attention mechanism.
- Practical implementation of GPT-2.

Reflection Questions:

- How does the attention mechanism improve language model performance?
- What are the potential challenges when implementing attention-based models?

Contact: Jordan.Hill@nmtafe.wa.edu.au

Looking Ahead

- **Next Topic**: Latest Developments
- **Preparation**: Explore how transformers are applied in computer vision.

Continue expanding your knowledge by seeing how attention mechanisms revolutionize different domains.