What are large language models (LLMs)?

Large language models (LLMs) are a category of foundation models trained on immense amounts of data making them capable of understanding and generating natural language and other types of content to perform a wide range of tasks.

LLMs have become a household name thanks to the role they have played in bringing generative AI to the forefront of the public interest, as well as the point on which organizations are focusing to adopt artificial intelligence across numerous business functions and use cases.

LLMs are a class of foundation models, which are trained on enormous amounts of data to provide the foundational capabilities needed to drive multiple use cases and applications, as well as resolve a multitude of tasks. This is in stark contrast to the idea of building and training domain specific models for each of these use cases individually, which is prohibitive under many criteria (most importantly cost and infrastructure), stifles synergies and can even lead to inferior performance.

LLMs represent a significant breakthrough in NLP and artificial intelligence, and are easily accessible to the public through interfaces like Open AI’s Chat GPT-3 and GPT-4, which have garnered the support of Microsoft. Other examples include Meta’s Llama models and Google’s bidirectional encoder representations from transformers (BERT/RoBERTa) and PaLM models. IBM has also recently launched its Granite model series on watsonx.ai, which has become the generative AI backbone for other IBM products like watsonx Assistant and watsonx Orchestrate.

In a nutshell, LLMs are designed to understand and generate text like a human, in addition to other forms of content, based on the vast amount of data used to train them. They have the ability to infer from context, generate coherent and contextually relevant responses, translate to languages other than English, summarize text, answer questions (general conversation and FAQs) and even assist in creative writing or code generation tasks.

They are able to do this thanks to billions of parameters that enable them to capture intricate patterns in language and perform a wide array of language-related tasks. LLMs are revolutionizing applications in various fields, from chatbots and virtual assistants to content generation, research assistance and language translation.

As they continue to evolve and improve, LLMs are poised to reshape the way we interact with technology and access information, making them a pivotal part of the modern digital landscape.

Transformers and Attention Mechanism: The Backbone of LLMs

Introduction

In our journey through the universe of Large Language Models, we’ve traversed the basic neural network terrains and delved deep into deep learning. Now, it’s time to explore the architecture and mechanisms that form the backbone of most modern LLMs: Transformers and Attention Mechanisms.

Setting the Stage: From RNNs and LSTMs to Transformers

Before the inception of the Transformer architecture, Recurrent Neural Networks (RNNs) and their evolved counterparts, Long Short-Term Memory units (LSTMs), were the go-to structures for sequence-to-sequence tasks in NLP. They processed sequences step-by-step, holding potential memory from previous steps. However, they had limitations, especially when dealing with long sequences, often leading to the vanishing gradient problem and difficulties in handling long-range dependencies.

Enter Transformers, which brought parallel processing to sequences and effectively addressed the issues of long-range dependencies through self-attention mechanisms.

The Transformer Architecture

Transformers, introduced in the paper “Attention is All You Need” by Vaswani et al., have become the foundation for most state-of-the-art NLP models.

Key Components:

1. Self-Attention Mechanism: Instead of processing sequences step-by-step, it allows the model to focus on different parts of the input simultaneously.

2. Positional Encoding: Since Transformers don’t process data sequentially, they need a way to consider the position of words in a sequence. Positional encoding is added to give the model information about the position of words.

3. Feed-Forward Neural Networks: Each attention output is passed through a feed-forward neural network (the same one for each position).

4. Stacked Layers: Both the encoder and decoder of the transformer contain multiple identical layers stacked on top of each other.

Attention Mechanism: What Does it Pay Attention to?

The essence of the Attention Mechanism is its ability to focus on specific parts of the input data, akin to how humans pay attention to specific parts of a sentence while comprehending or responding.

The attention mechanism computes a weighted sum of input values (or values from the previous layer), where the weights are decided based on the query, key, and value representations of the data.