Creating SVA for Formal Verification from Natural Language Specification

iSpec.ai

Our Approach to creating SVA from English

- Objective: The goal of developing a service like iSpec.ai is to **provide customers with a method for generating SystemVerilog Assertions by articulating their desired tasks in plain English.** This falls under the category of Machine Translation tasks. To accomplish this objective, we've leveraged state-of-the-art LLMs (Large Language Models) known for their exceptional ability to map dependencies between input and output sequences.

- In the industry, there are **generally two approaches to harnessing the capabilities of LLMs**, which are becoming increasingly popular: **Prompt Engineering and Fine-Tuning a LLM for Downstream tasks**. At Agnisys, we have employed a combination of both techniques to create iSpec.ai as a service.

- Until recently, Fine-Tuning a LLM presented significant challenges, especially on consumer hardware. However, with the emergence of **Parameter-Efficient Fine-Tuning techniques like QLORA, doors have opened to efficiently utilize LLMs for specific downstream tasks.**

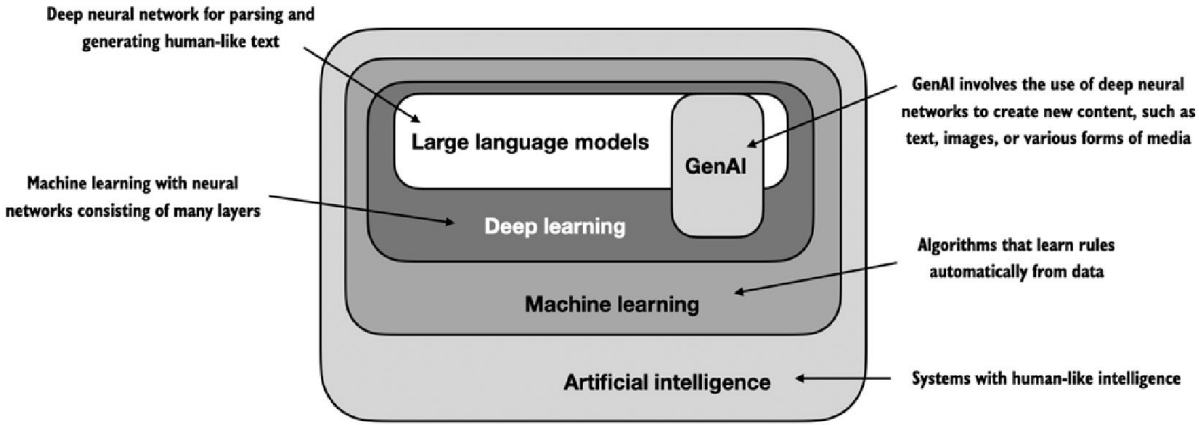
- We will delve into the details of our approach in the following slides.

What Is An LLM?

- An LLM, a large language model, is a **neural network designed to understand, generate, and respond to human-like text.** These models are deep neural networks trained on massive amounts of text data, sometimes encompassing large portions of the entire publicly available text on the internet.

- LLMs **utilize an architecture called the transformer, which allows them to pay selective attention to different parts of the input** when making predictions, making them especially adept at handling the nuances and complexities of human language.

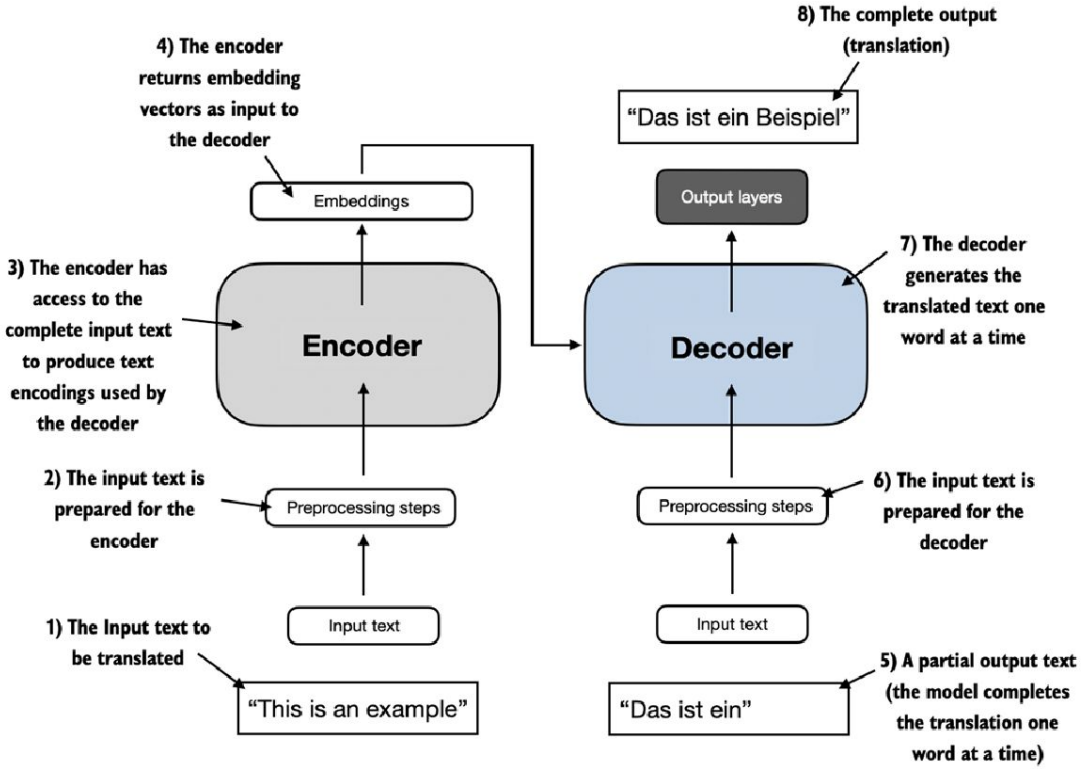
Where Does LLMs fit in the entire AI Space ?



[Image Source: https://www.manning.com/books/build-a-large-language-model-from-scratch]

Transformer Architecture

- The initial architecture proposed in the **'Attention is All You Need'** paper represented a significant milestone in the development of Transformer models. However, in subsequent advancements, state-of-the-art **Large Language Models (LLMs) predominantly utilize only the Decoder component, with slight variations from the original architecture.**
- A **key component of transformers and LLMs is the self-attention mechanism** which allows the model to weigh the importance of different words or tokens in a sequence relative to each other. This mechanism **enables the model to capture long range dependencies and contextual relationships within the input data**, enhancing its ability to generate coherent and contextually relevant output.

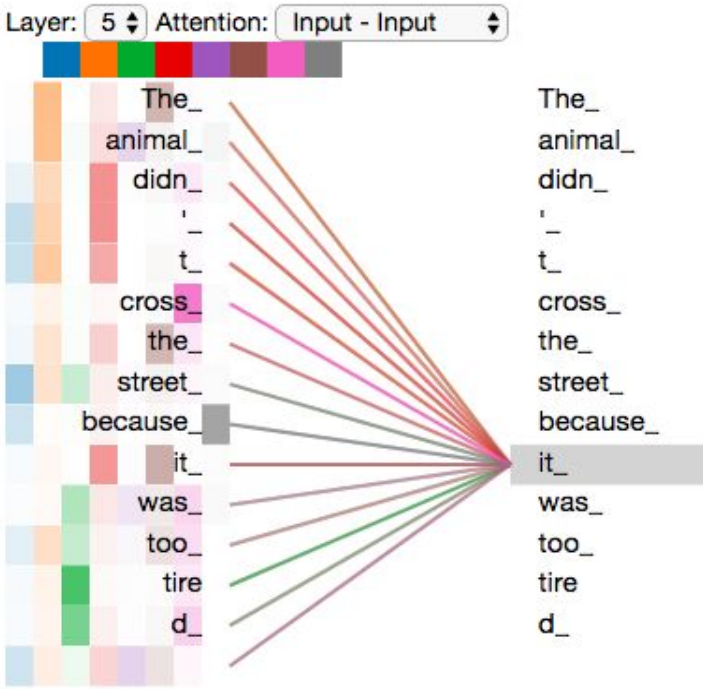


[Image Source: https://www.manning.com/books/build-a-large-language-model-from-scratch]

Unveiling Self-Attention: A Concise Overview

- Self-attention is a neural network **mechanism where each element in a sequence evaluates its importance relative to other elements in the same sequence.** It allows models to capture intricate dependencies within the data, enhancing tasks like natural language processing and time series analysis.

- LLMs based on a **decoder-only architecture utilize a variation of self-attention known as Causal Self-Attention.** This variant differs from traditional self-attention in that it only considers tokens that have occurred previously and lacks awareness of future tokens.

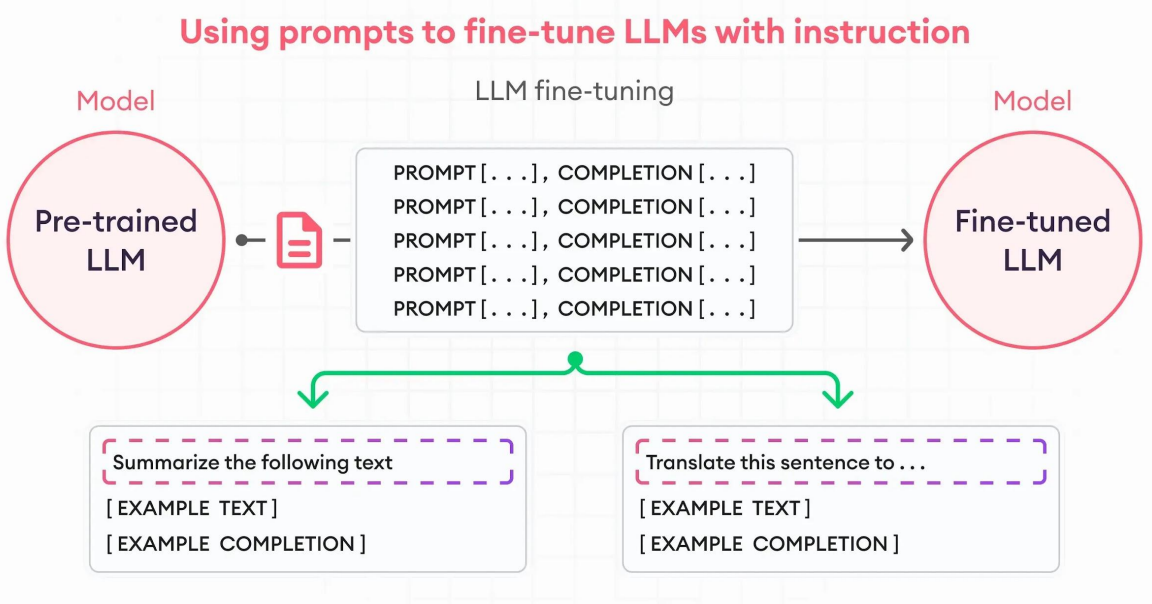


[Image Source : http://jalammar.github.io/illustrated-transformer/]

What is LLM Fine Tuning ?

- Large language model (LLM) fine-tuning is the **process of taking pre-trained models and further training them on smaller, specific datasets** to refine their capabilities and improve performance in a particular task or domain.

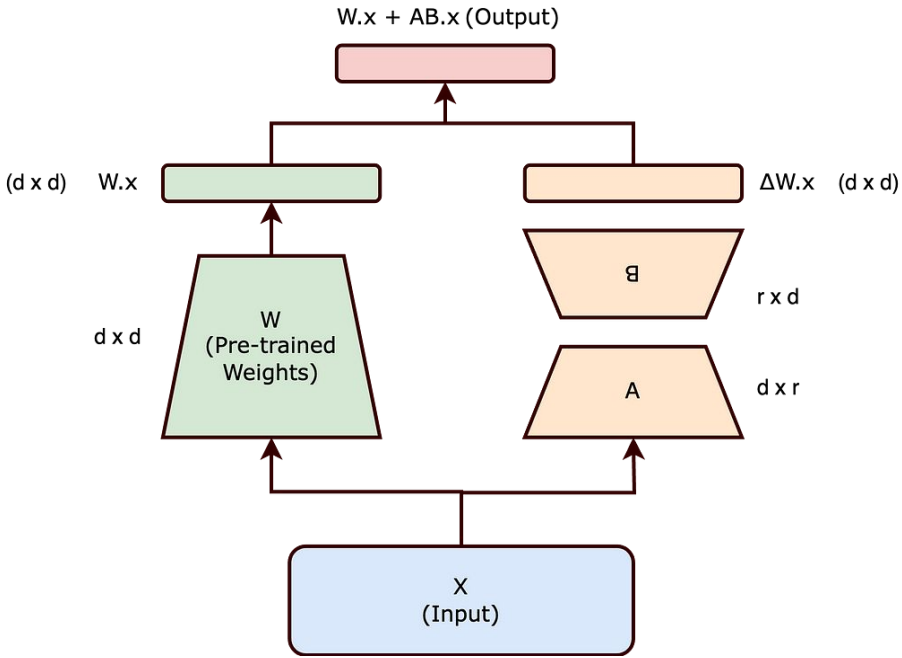
- The strategy we employed to enhance the model's performance on our specific task is **instruction fine-tuning**. This approach involves training the machine learning model using examples that illustrate how the model should respond to a given query.



[Image Source : https://www.superannotate.com/blog/llm-fine-tuning]

Parameter Efficient Fine Tuning : QLORA

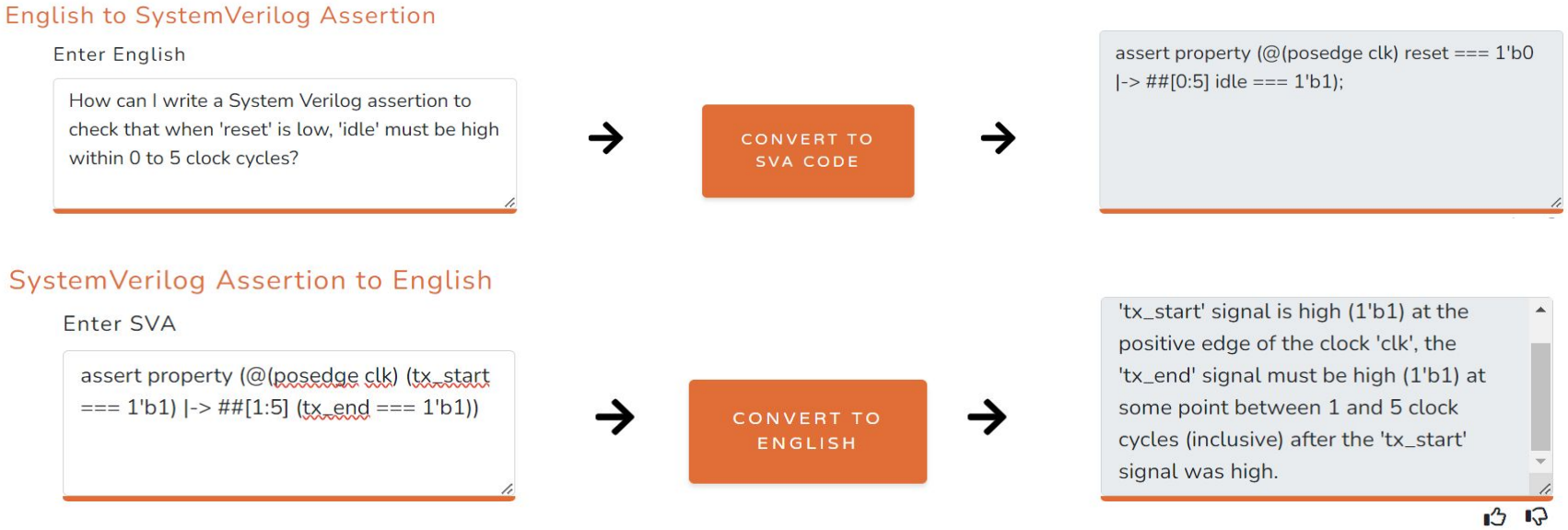
- Before the emergence of PEFT techniques, the predominant method for Instruction Fine-Tuning was Full Fine-Tuning, where all of the model's weights are updated. This process results in a new version of the model with updated weights. It's crucial to note that, similar to pre-training, **Full Fine-Tuning demands significant memory and compute resources to store and process all the gradients, optimizers, and other components** during training. Consequently, **training the model on consumer hardware becomes extremely challenging and often unfeasible.**
- Recently, researchers have proposed some intriguing and innovative techniques that make it feasible to train these models on consumer hardware. One such technique that has garnered significant attention is QLORA, which combines Quantization and Low-Rank Adaptation.
- **Quantization** is a technique to reduce the computational and memory costs of running inference by **representing the weights and activations with low-precision data types like 8-bit integer (int8) instead of the usual 32-bit floating point (float32).**
- While **LORA** is a technique that **achieves a drastic reduction in the number of trainable weights by selecting matrices (A) and (B) with a lower rank (r), thereby significantly reducing the number of trainable parameters**, the pretrained model weights are kept frozen while allowing the adapter matrices to train on the fine-tuning dataset.



[Image Source : https://towardsdatascience.com/understanding-lora-low-rank-adaptation-for-finetuning-large-models-936bce1a07c6]

iSpec.ai Demo: Transforming SystemVerilog Assertion Creation

- iSpec.ai provides comprehensive functionality, encompassing both English-to-SVA and SVA-to-English generation capabilities, catering to a wide range of user needs and preferences.



AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

THANK YOU