LLMs vs. Traditional ML Models

LLMs differ from traditional ML models in both how they are trained and what they can do.

Training Differences

There are two main ways that the training of LLMs differs from the training of traditional ML models: **architecture** and **data**.

Architecture

As mentioned previously, LLMs are typically trained using the Transformer architecture, which is a type of neural network architecture that utilizes a computationally complex technique called the attention mechanism.

Traditional ML models, on the other hand, might use neural networks, but they also might use simpler, less computationally complex algorithms such as linear regression or decision trees. The main limitation of these architectures is their inability to handle long sequences of input data.

Data

LLMs are trained on unstructured text data that does not require manual annotation. This means that they can utilize very large amounts of data from sources such as books, websites, and PDFs.

In contrast, traditional ML models are typically trained on much smaller, labeled datasets. These datasets require more manual annotation and curation efforts, but require less computational cost for model training.

Differences in Capabilities

Traditional ML models are trained to perform specific tasks. They can achieve very good performance, with lower computational cost, but can only be used to handle the specific circumstances they were trained to handle.

LLMs are sometimes referred to as *foundation models* because they are so adaptable to different situations. They can be used to perform various tasks without any further training, because their training data is so vast and covers so many different contexts. If they need to be adapted for a particularly niche task, they can also be **fine-tuned** with a dataset similar to the datasets used to train traditional ML models.

Comparison Table

Attribute	Traditional ML Models	LLMs
Architecture	Varies; can be neural networks or simpler algorithms	Typically based on Transformer architecture
Training Data	Structured	Unstructured
Capabilities	Targeted	Broad
Computational Complexity	Lower	Higher