**What is Model Training in AI?**

**Wednesday, April 23, 2025 6:39 PM**

→ **Process involved in the model Building:**

* Data Ingestion
* Data Analysis
* Data Preprocessing
* Model Building
* Model Evaluation

→ **Model building in Classical Machine Learning:**

✔ **Supervised Modelling**

* Linear Regression
* Logistic Regression
* Support Vector Machine
* Decision Tree
* Random Forest
* Light Gradient Boosting Machine (LGBM)
* Xtreme Gradient Boosting Machine (XGBM)
* Naïve Bayes

✔ **Unsupervised Modelling**

* Clustering: K-Means, Hierarchical Clustering, DB Scan Clustering

→ **Model Building in Deep learning**

✔ ANN: Artificial Neural Network (ANN) for Regression and Classification  
✔ CNN: Convolution Neural Network (CNN) for Grid Like Data Ex. Images

We used CNN-based architectures to solve tasks such as:

* **Image Classification**
* **Object Detection**
* **Object Segmentation**
* **Object Tracking**
* **Optical Character Recognition (OCR)**

**Image Classification**: LeNet-5, AlexNet, VGG-16/19, ResNet (e.g., ResNet-50), DenseNet, EfficientNet  
**Object Detection**: R-CNN, Fast R-CNN, Faster R-CNN, YOLO (v1–v8), SSD, DETR  
**Object Segmentation**: FCN, U-Net, SegNet, Mask R-CNN, DeepLab (v1–v3+)  
**Object Tracking**: SORT, Deep SORT, SiamMask, SiamRPN, Tracktor++  
**Optical Character Recognition (OCR)**: Tesseract OCR, CRNN, EAST, CRAFT, PaddleOCR

✔ **RNN: Recurrent Neural Network RNN**  
✔ Variant of RNN: LSTM & GRU for Sequence Like Data Ex. Text  
✔ We use this architecture for Sequence to Sequence learning

Using **RNN, LSTM, and GRU** architectures, we built advanced models such as Encoder-Decoder and Encoder-Decoder with Attention, which were applied to various NLP tasks including:

* Text Classification
* Text Summarization
* Question Answering
* Text Generation (e.g., Next Word Prediction)
* Text Translation

This process is known as **Language Modeling.**  
Keep the following points in mind:  
a. It involves training the model from scratch on a corpus (dataset).  
b. It is also referred to as task-specific training, as the model is optimized for a particular language task.

One of the key milestones was the idea of **Universal Language Modeling using Encoder-Decoder architectures.** This is where the foundation of fine-tuning in NLP was laid out.

The typical process looked like this:  
a. **Pretraining Phase** (Unsupervised)  
 - The model was first pretrained on large-scale text using tasks like machine translation.  
b. **Fine-tuning Phase** (Supervised)  
 - Then, the same model was fine-tuned on specific downstream tasks, such as text classification, using labeled data.  
c. **Model Architecture**  
 - The architecture used for this training was the Encoder-Decoder with Attention mechanism.

However, there were some limitations with this early approach:  
a. The pretraining task was limited to machine translation, which restricted generalization to other tasks.  
b. The model architecture was based on LSTM, which struggled with long-range dependencies and was computationally inefficient.

**Then Came the Game-Changer: The Transformer**

Introduced in the paper *“Attention is All You Need”*, the Transformer architecture revolutionized NLP.

* Unlike LSTMs, Transformers do not rely on recurrence.
* Instead, they use a novel concept called **self-attention**, which allows the model to attend to all positions in the input sequence simultaneously — enabling much faster and more effective learning.

**Stage – Components**

* **Input Layer**: Token Embedding, Positional Encoding
* **Encoder Layer**: Multi-Head Self-Attention, Add & Norm, FFN
* **Decoder Layer**: Masked Self-Attention, Encoder-Decoder Attention, Add & Norm, FFN
* **Output Layer**: Linear, Softmax

After the successful release of the Transformer architecture, which proved to be a highly powerful model, several **variants** started emerging — such as **BERT, GPT, T5, and XLM.**

These models marked the beginning of what we now call the **initial wave of Large Language Models (LLMs).**  
While they weren’t as massive as today’s models (like GPT-4), they were considered a **major milestone** in the evolution of large-scale language modeling.

Unlike earlier models that relied heavily on machine translation, these new models introduced more sophisticated pretraining objectives, such as:

* **Masked Language Modeling (MLM)** → used in BERT
* **Causal Language Modeling (CLM)** → used in GPT

This shift in strategy helped these models learn **deep contextual representations** of language, making them highly effective across a wide range of downstream NLP tasks.

**Understanding the Evolution of LLMs: A Quick Story**

**a. BERT**

* Pretraining: Used Masked Language Modeling (MLM) — where random words in a sentence are masked and the model learns to predict them.
* Fine-tuning: Then fine-tuned for downstream tasks like text classification, NER, etc.
* Encoder-based model.

**b. GPT**

* Pretraining: Used Auto-Regressive Language Modeling (CLM) — predicting the next word in a sequence.
* Further Training: Later versions were trained on conversation-style datasets to enable dialogue understanding.
* Decoder-only model.

**Stage 1: Pretraining Phase (Foundation Model Building)**

This stage focuses on **teaching the model language itself** using huge amounts of raw text.

* **Data Collection**  
   ▪ Massive-scale datasets (web text, books, code, forums, etc.)
* **Data Cleaning**  
   ▪ Remove low-quality, duplicated, or irrelevant content
* **Data Analysis**  
   ▪ Check for distribution, biases, toxicity, imbalance, etc.
* **Data Preprocessing**  
   ▪ Tokenization, normalization, filtering, formatting for training
* **Unsupervised Pretraining**  
   ▪ Using architectures like GPT (decoder-only)  
   ▪ The model learns to generate coherent and meaningful text
* **Evaluation**  
   ▪ Perplexity, loss, token accuracy on held-out validation data

**What is Pretraining?**

* Unsupervised learning phase
* Typically uses Auto-Regressive Language Modeling (e.g., predict the next token)
* Builds a foundation model that understands grammar, facts, and reasoning
* This model is referred to as a Pretrained Language Model (PLM)

**Stage 2: Supervised Fine-Tuning (SFT)**

Now the base model is refined to follow human instructions for specific tasks.

* **Supervised Finetuning** using labeled datasets  
   ▪ Tasks include classification, summarization, QA, reasoning, dialogue  
   ▪ Often uses techniques like PEFT (Parameter Efficient Fine-Tuning) — e.g., LoRA, QLoRA

**Model Evaluation**

* Performance is evaluated on benchmarks like HELM, MMLU, SuperGLUE, etc.

**Stage 3: Alignment (Controlled & Human-like Responses)**

To ensure safe, helpful, and human-aligned outputs:

* **RLHF (Reinforcement Learning from Human Feedback)**  
   ▪ Use PPO (Proximal Policy Optimization) to align responses with human preferences
* **Modern Alternative: DPO (Direct Preference Optimization)**  
   ▪ Simpler and more stable than PPO, now widely used in 2024+

**Model Evaluation**

* Human evals, reward models, safety & helpfulness scores

**Welcome to the LLM Era!**

Following this pipeline, we’ve seen the rise of powerful open and closed-source LLMs:

* LLaMA Series (Meta)
* Mistral Series (Mistral AI)
* Gemini Series (Google DeepMind)
* DeepSeek Series
* Claude Series (Anthropic)
* GPT Series (OpenAI)

**Stage – Description**

1. **Pretraining**: On large corpus using MLM (BERT) or CLM (GPT) with unsupervised objectives
2. **Fine-tuning**: On specific tasks (e.g., classification, QA, summarization) with labeled data
3. **Instruction Tuning**: Train model to follow human instructions effectively
4. **RLHF (Optional)**: Refine responses using reinforcement learning with human feedback
5. **Continuous Training**: New data added over time (optional, depending on use case)