Artificial Intelligence (Machine Learning & Deep Learning) [Course]

Week 10 – Hugging Face -Fine-Tuning LLMs (PEFT, QLoRA)

[See examples / code in GitHub code repository]

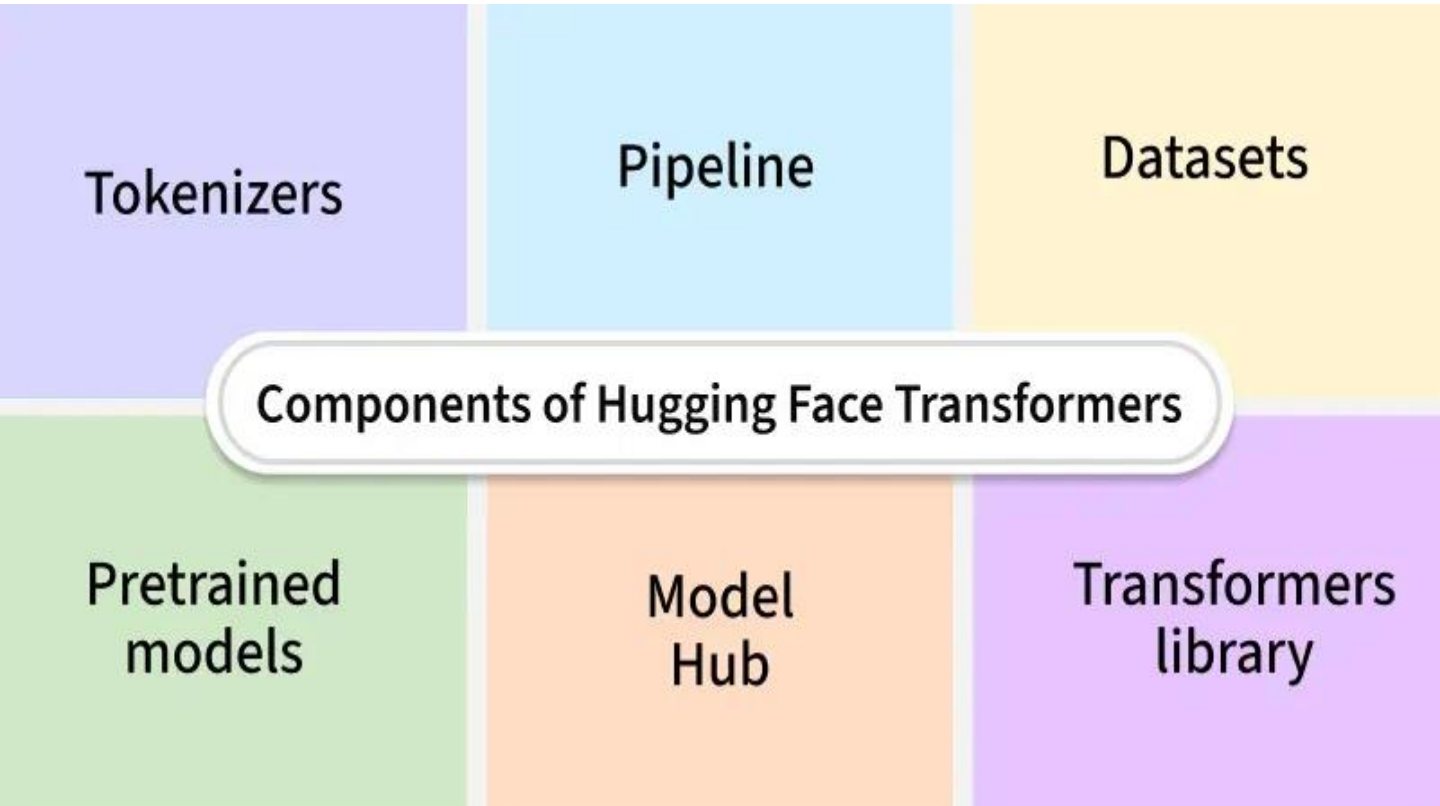
It is not about Theory, it is 20% Theory and 80% Practical – Technical/Development/Programming [Mostly Python based]

Hugging Face & Transformers

Hugging Face Transformers is an open source library that provides easy access to thousands of machine learning models for natural language processing, computer vision and audio tasks. Built on top of frameworks like PyTorch and TensorFlow it offers a unified API to load, train and deploy models such as BERT, GPT and T5. Its versatility and large model hub make it a go-to tool for both beginners and researchers to build AI applications with minimal effort.

Core Components

Lets see core components of Hugging Face Transformers:



Reference:

https://www.geeksforgeeks.org/artificial-intelligence/Introduction-to-hugging-face-transformers/

Hugging Face & Core Elements

Tokenizers: This is responsible for efficiently converting raw text into tokens that transformer models can understand. It ensures text is appropriately tokenized, padded and truncated to match the model's input requirements.

Pipeline: Pipeline abstraction provides a simple interface for running pre trained models on a variety of tasks. It allows users to easily interact with models without writing custom code making it accessible for beginners or for rapid prototyping.

Datasets: This provides access to a wide range of datasets for training and evaluating models. It simplifies the data pipeline, supporting large scale datasets and making it easy to load, filter and preprocess data for use with transformer models.

Transformers Library: It supports PyTorch, TensorFlow and JAX enabling users to train, fine tune and use pre trained models across different frameworks. It removes much of the complexity, allowing users to focus on model development and experimentation.

Model Hub: This is a central repository that hosts thousands of pre trained models from Hugging Face and the community. Users can easily download models, fine tune them and share them with others.

Pre trained Models: Hugging Face provides a vast collection of pre trained models for NLP tasks including text classification, translation, question answering, text generation and more. These models are built on transformer architectures like BERT, GPT-2, T5, RoBERTa, DistilBERT and others.

Reference:

Llama (Large Language Model Meta AI)

Llama (**Large Language Model Meta AI**) is a family of large language models (LLMs) released by Meta AI starting in February 2023. The latest version is Llama 4, released in April 2025.

Llama models come in different sizes, ranging from 1 billion to 2 trillion parameters. Initially only a foundation model, starting with Llama 2, Meta AI released instruction fine-tuned versions alongside foundation models.

Reference:

https://www.llama.com/

https://www.meta.ai/

https://www.llama.com/docs/how-to-guides/prompting/

https://en.wikipedia.org/wiki/Llama_(language_model)

Reference:

https://www.geeksforgeeks.org/artificial-intelligence/Introduction-to-hugging-face-transformers/

Google Colaboratory (Colab) and Visual Studio Code (VS Code)

Google Colaboratory (Colab) and Visual Studio Code (VS Code) are distinct tools used for coding and development, each with its own strengths and weaknesses.

Google Colab:

Cloud-based Jupyter environment:
Colab operates entirely in your web browser, providing a free, cloud-based Jupyter Notebook environment.

Free GPU/TPU access:
It offers free access to powerful hardware like GPUs and TPUs, making it ideal for machine learning and deep learning tasks without requiring local setup.

Easy sharing and collaboration:
Notebooks can be easily shared and collaborated on, similar to Google Docs.

No installation required:
Users can immediately start coding without any local software installation.

Limitations:
Free tier has limitations on session duration, storage space (linked to Google Drive), and resource availability.

Visual Studio Code (VS Code):

Lightweight, powerful local editor:
VS Code is a highly customizable and extensible source code editor installed locally on your machine.

Extensive ecosystem:
It boasts a vast marketplace of extensions for various languages, frameworks, and tools, enhancing functionality significantly.

Robust debugging and version control:
Offers powerful debugging tools and seamless integration with version control systems like Git.

Offline capability:
Can be used offline and provides full control over your local development environment.

Requires local setup:
Users need to install VS Code and relevant extensions, and manage their local environment and dependencies.

Google Colaboratory (Colab) and Visual Studio Code (VS Code)

Key Differences and Use Cases:

Environment:
Colab is cloud-based, while VS Code is a local desktop application.
Hardware access:
Colab provides free access to GPUs/TPUs, while VS Code relies on your local machine's hardware.
Setup:
Colab requires no setup, while VS Code requires installation and configuration.
Collaboration:
Colab excels in real-time notebook collaboration, while VS Code's collaboration features are typically through shared repositories.
Flexibility and Customization:
VS Code offers far greater customization and a broader range of extensions than Colab.

Choosing between them:

Choose Colab for:
Quick prototyping, experimentation with machine learning models, projects requiring free GPU/TPU access, and easy sharing of notebooks.
Choose VS Code for:
Developing production-ready software, large-scale projects, projects requiring specific local configurations, and when full control over the development environment is needed.
It is also possible to integrate Colab notebooks within VS Code to leverage the strengths of both platforms.

References:

https://datasciencenotebook.org/compare/colab/vscode
https://deepnote.com/compare/colab-vs-vscode

Google Colaboratory (Colab) - Hugging Face meta-llama/Llama-2-7b-chat-hf

Llama 2
Llama 2 is a collection of pretrained and fine-tuned generative text models ranging in scale from 7 billion to 70 billion parameters. This is the repository for the 7B fine-tuned model, optimized for dialogue use cases and converted for the Hugging Face Transformers format. Links to other models can be found in the index at the bottom.

Meta developed and publicly released the Llama 2 family of large language models (LLMs), a collection of pretrained and fine-tuned generative text models ranging in scale from 7 billion to 70 billion parameters. Our fine-tuned LLMs, called Llama-2-Chat, are optimized for dialogue use cases. Llama-2-Chat models outperform open-source chat models on most benchmarks we tested, and in our human evaluations for helpfulness and safety, are on par with some popular closed-source models like ChatGPT and PaLM.
Model Developers Meta
Variations Llama 2 comes in a range of parameter sizes — 7B, 13B, and 70B — as well as pretrained and fine-tuned variations.
Input Models input text only.
Output Models generate text only.

Llama 2 – hosted on Hugging Face URL:
https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

Google Colaboratory (Colab) - Hugging Face meta-llama/ Llama-2-7b-chat-hf -References:
https://colab.research.google.com/#fileId=https%3A//huggingface.co/meta-llama/Llama-2-7b-chat-hf.ipynb

Fine-Tuning LLMs (PEFT, QLoRA)

Fine-tuning large language models (LLMs) is used for adapting LLM's to specific tasks, improving their accuracy and making them more efficient. However full fine-tuning of LLMs can be computationally expensive and memory-intensive. QLoRA (Quantized Low-Rank Adapters) is a technique used to significantly reduces the computational cost while maintaining model quality.

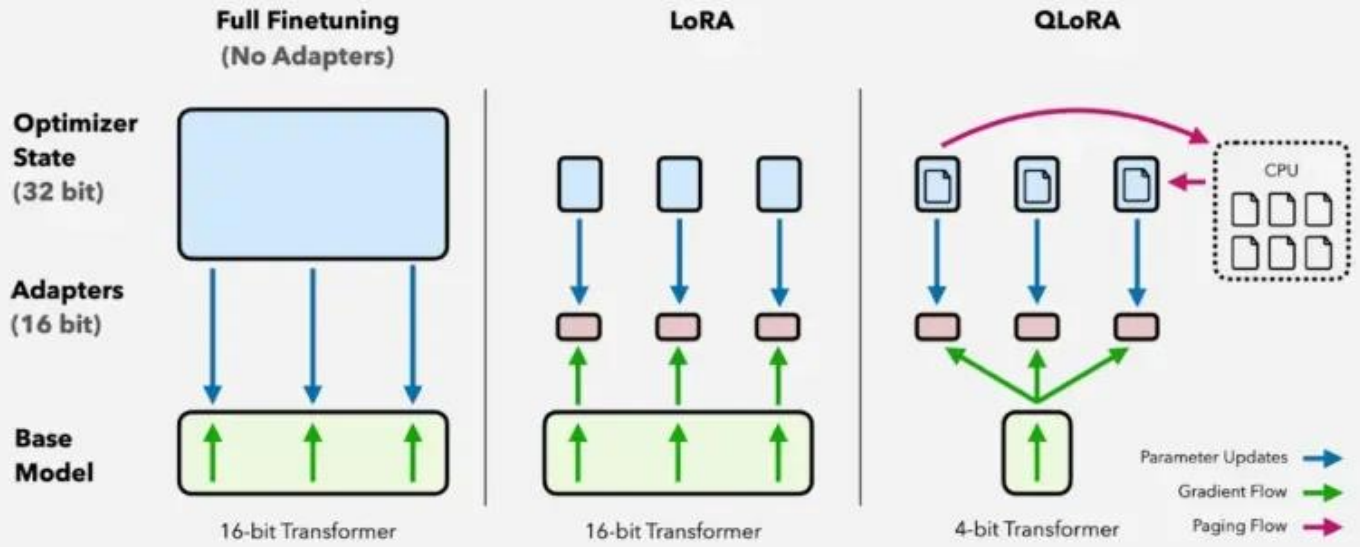
What is QLoRA?

QLoRA is a advanced fine-tuning method that quantizes LLMs to reduce memory usage and applies Low-Rank Adaptation (LoRA) to train a subset of model parameters. This allows:

Lower GPU memory requirements : Fine-tuning large models on consumer GPUs.

Faster training : Using fewer parameters speeds up the process.

Preserved model quality : Achieves similar performance to full fine-tuning.



Reference:

https://www.geeksforgeeks.org/nlp/fine-tuning-large-language-models-llms-using-qlora/

Reference Code: https://github.com/ShahzadSarwar10/FULLSTACK-WITH-AI-BOOTCAMP-B1-MonToFri-2.5Month-Explorer/blob/main/Week10/Case-10-2-Fine-TuningLargeLanguageModels-LLMs-Using-QLoRA.py

Thank you - for listening and participating

- ❑**Questions / Queries**
- ❑**Suggestions/Recommendation**
- ❑**Ideas…..?**

Shahzad Sarwar
Cognitive Convergence
https://cognitiveconvergence.com
shahzad@cognitiveconvergence.com
voice: +1 4242530744 (USA) +92-3004762901 (Pak)