Databricks Foundation Model Fine-Tuning Process

Dataset Preparation

- **Input**: Your training dataset (e.g., text data, CSV files).
- **Process**:
  - Preprocess the dataset using **Databricks notebooks**.
  - Use libraries like **Hugging Face's** `datasets` and **PySpark** for large-scale preprocessing and tokenization.
- **Output**: Preprocessed dataset stored in **S3** or Databricks DBFS for easy access.

Model Setup in Databricks

- **Input**: Pre-trained **foundation model** (e.g., **Phi-2** from Hugging Face).
- **Process**:
  - Set up a **GPU-enabled cluster** (using instance types like `g4dn.xlarge` or `p3.2xlarge`).
  - Install necessary libraries such as **transformers**, **datasets**, **accelerate**, and **boto3** (for S3 integration).
  - Load the pre-trained foundation model (e.g., **Phi-2**) and tokenizer from Hugging Face Hub.
- **Output**: The model is ready for fine-tuning on the dataset.

Fine-Tuning in Databricks

- **Input**: Pre-trained model, tokenized dataset.
- **Process**:
  - Fine-tune the foundation model using Hugging Face's **Trainer** API in Databricks notebooks.
  - Leverage **GPU resources** to speed up training.
  - Store training configurations such as batch size, number of epochs, and learning rate in the Databricks notebook.
  - Monitor training performance and adjust parameters as needed.
- **Output**: Fine-tuned model.

Model Storage in S3

- **Input**: Fine-tuned model (weights, tokenizer).
- **Process**:
  - Save the fine-tuned model back to **AWS S3** for easy access and deployment.
  - Use the **boto3** library to interact with AWS S3 and upload model artifacts.
- **Output**: Fine-tuned model saved in **S3**.

Deployment to AWS SageMaker (Optional)

- **Input**: Fine-tuned model in **S3.**
- **Process**:
    - Use **AWS SageMaker** for model deployment**.**
    - Deploy the model as a **real-time endpoint** for inference, enabling users or systems to interact with the fine-tuned model.
- **Output**: **SageMaker Endpoint** serving the fine-tuned model.

Monitoring and Optimization

- **Input**: Logs from **Databricks** and **SageMaker** (using AWS CloudWatch)**.**
- **Process**:
    - Monitor the performance of both the training process and the deployed endpoint**.**
    - Adjust training parameters, instance types, or resources as required to optimize performance and reduce cost**.**
- **Output**: Optimized training and inference workflows.

—----------

Finetuning in AWS —HF→S3—>Aws sagemaker

End-to-End Flow

1. **Raw Dataset** → Preprocess & Upload → **S3**
2. **S3 Dataset** → Fine-Tune with Hugging Face → **Fine-Tuned Model**
3. **Fine-Tuned Model** → Upload to **S3**
4. **S3 Model** → Deploy to **SageMaker Endpoint**
5. **Endpoint** → Serve Predictions
6. **CloudWatch Monitoring** → Optimize and Improve