Step 3: Model Fine-tuning

In this notebook, you'll fine-tune the Meta Llama 2 7B large language model, deploy the fine-tuned model, and test it's text generation and domain knowledge capabilities.

Fine-tuning refers to the process of taking a pre-trained language model and retraining it for a different but related task using specific data. This approach is also known as transfer learning, which involves transferring the knowledge learned from one task to another. Large language models (LLMs) like Llama 2 7B are trained on massive amounts of unlabeled data and can be fine-tuned on domain domain datasets, making the model perform better on that specific domain.

Input: A train and an optional validation directory. Each directory contains a CSV/JSON/TXT file. For CSV/JSON files, the train or validation data is used from the column called 'text' or the first column if no column called 'text' is found. The number of files under train and validation should equal to one.

- **You'll choose your dataset below based on the domain you've chosen**

Output: A trained model that can be deployed for inference.
After you've fine-tuned the model, you'll evaluate it with the same input you used in project step 2: model evaluation.

Set up

Install and import the necessary packages. Restart the kernel after executing the cell below.

---

[1]:
```
!pip install --upgrade sagemaker datasets
```

```
Requirement already satisfied: sagemaker in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (2.232.1)
Requirement already satisfied: datasets in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (3.0.1)
Requirement already satisfied: attrs<24,>=23.1.0 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (23.2.0)
Requirement already satisfied: boto3<2.0,>=1.34.142 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (1.35.16)
Requirement already satisfied: cloudpickle==2.2.1 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (2.2.1)
Requirement already satisfied: docker in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (7.1.0)
Requirement already satisfied: google-pasta in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (0.2.0)
Requirement already satisfied: importlib-metadata<7.0,>=1.4.0 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (6.11.0)
Requirement already satisfied: jsonschema in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (4.23.0)
Requirement already satisfied: numpy<2.0,>=1.9.0 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (1.22.4)
Requirement already satisfied: packaging>=20.0 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (21.3)
Requirement already satisfied: pandas in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (2.2.2)
Requirement already satisfied: pathos in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (0.3.2)
Requirement already satisfied: platformdirs in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (4.2.2)
Requirement already satisfied: protobuf<5.0,>=3.12 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (4.25.4)
Requirement already satisfied: psutil in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (6.0.0)
Requirement already satisfied: pyyaml~=6.0 in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (6.0.1)
Requirement already satisfied: requests in /home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages (from sagemaker) (2.32.3)
```

To create a finance domain expert model:

- `"training": f"s3://genaiwithawsproject2024/training-datasets/finance"`

To create a medical domain expert model:

- `"training": f"s3://genaiwithawsproject2024/training-datasets/medical"`

To create an IT domain expert model:

- `"training": f"s3://genaiwithawsproject2024/training-datasets/it"`

[*]:
```
from sagemaker.jumpstart.estimator import JumpStartEstimator
import boto3

estimator = JumpStartEstimator(model_id=model_id,  environment={"accept_eula": "true"},instance_type = "ml.g5.2xlarge")

estimator.set_hyperparameters(instruction_tuned="False", epoch="5")

#Fill in the code below with the dataset you want to use from above
estimator.fit({"training": f"s3://genaiwithawsproject2024/training-datasets/finance"})
```

```
/home/ec2-user/anaconda3/envs/python3/lib/python3.10/site-packages/sagemaker/base_serializers.py:28: UserWarning: A NumPy version >=1.23.5 and <2.3.0 i
s required for this version of SciPy (detected version 1.22.4)
  import scipy.sparse
sagemaker.config INFO - Not applying SDK defaults from location: /etc/xdg/sagemaker/config.yaml
sagemaker.config INFO - Not applying SDK defaults from location: /home/ec2-user/.config/sagemaker/config.yaml
Using model 'meta-textgeneration-llama-2-7b' with wildcard version identifier '*'. You can pin to version '4.8.0' for more stable results. Note that mo
dels may have different input/output signatures after a major version upgrade.
INFO:sagemaker:Creating training-job with name: meta-textgeneration-llama-2-7b-2024-09-27-12-11-49-897
2024-09-27 12:11:51 Starting - Starting the training job
2024-09-27 12:11:51 Pending - Training job waiting for capacity...
2024-09-27 12:12:16 Pending - Preparing the instances for training...
2024-09-27 12:12:56 Downloading - Downloading input data.....................
```

```
2024-09-27 12:23:45,439 sagemaker-training-toolkit INFO     Reporting training SUCCESS

2024-09-27 12:23:56 Uploading - Uploading generated training model
2024-09-27 12:24:39 Completed - Training job completed
Training seconds: 703
Billable seconds: 703
```

Deploy the fine-tuned model

Next, we deploy the domain fine-tuned model. We will compare the performance of the fine-tuned and pre-trained model.

[4]:
```
'''
# Do not use estimator.deploy() without mentioning the instance_type.
# It's because when you call estimator.deploy() without explicitly setting the instance_type for the endpoint,
# SageMaker selects a default instance type for hosting, which, in this case, is ml.g5.12xlarge.
# However, Udacity doesn't allow instance type more than "ml.*.2xlarge".
'''
finetuned_predictor = estimator.deploy(instance_type="ml.g5.2xlarge", initial_instance_count=1)
```

```
INFO:sagemaker:Creating model with name: meta-textgeneration-llama-2-7b-2024-09-27-12-25-15-869
INFO:sagemaker:Creating endpoint-config with name meta-textgeneration-llama-2-7b-2024-09-27-12-25-15-861
INFO:sagemaker:Creating endpoint with name meta-textgeneration-llama-2-7b-2024-09-27-12-25-15-861
------------!
```

Evaluate the pre-trained and fine-tuned model

Next, we use the same input from the model evaluation step to evaluate the performance of the fine-tuned model and compare it with the base pre-trained model.

Create a function to print the response from the model

[5]:
```
def print_response(payload, response):
```

Create a function to print the response from the model

[5]:
```
def print_response(payload, response):
    print(payload["inputs"])
    print(f"> {response}")
    print("\n==================================\n")
```

Now we can run the same prompts on the fine-tuned model to evaluate it's domain knowledge.

Replace "inputs" in the next cell with the input to send the model based on the domain you've chosen.

For financial domain:

"inputs": "Replace with sentence below from text"

- "The investment tests performed indicate"
- "the relative volume for the long out of the money options, indicates"
- "The results for the short in the money options"
- "The results are encouraging for aggressive investors"

For medical domain:

"inputs": "Replace with sentence below from text"

- "Myeloid neoplasms and acute leukemias derive from"
- "Genomic characterization is essential for"
- "Certain germline disorders may be associated with"
- "In contrast to targeted approaches, genome-wide sequencing"

For IT domain:

"inputs": "Replace with sentence below from text"

[7]:
```
payload = {
    "inputs": """
    The investment tests performed indicate
    """,
    "parameters": {
        "max_new_tokens": 64,
        "top_p": 0.9,
        "temperature": 0.6,
        "return_full_text": False,
    },
}
try:
    response = finetuned_predictor.predict(payload, custom_attributes="accept_eula=true")
    print_response(payload, response)
except Exception as e:
    print(e)
```

```
    The investment tests performed indicate

> [{'generated_text': '\n    that the performance of the model is comparable to that of the\n    conventional neural network.\n\n    The performance of the model can be improved by including more\n    features in the model, such as the historical prices of the\n    stocks and the volume of the stock
s.\n\n    '}]

==================================
```

Do the outputs from the fine-tuned model provide domain-specific insightful and relevant content? You can continue experimenting with the inputs of the model to test it's domain knowledge.