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.

```
!pip install --upgrade sagemaker datasets
!pip install awscli --quiet
!aws configure
!aws configure get region
```

```
Requirement already satisfied: opentelemetry-semantic-conventions==0.48b0 in /usr/local/lib/python3.10/dist-packages (from o
Requirement already satisfied: wrapt<2,>=1.10 in /usr/local/lib/python3.10/dist-packages (from deprecated>=1.2.6->openteleme
Requirement already satisfied: smmap<6,>=3.0.1 in /usr/local/lib/python3.10/dist-packages (from gitdb<5,>=4.0.1->gitpython<4
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth~=2.0->data
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth~=2.0->databricks-s
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->
AWS Access Key ID [****************B5MN]:
AWS Secret Access Key [****************yta9]:
Default region name [us-east-1]:
Default output format [None]:
us-east-1
```

Select the model to fine-tune

```
model_id, model_version = "meta-textgeneration-llama-2-7b", "2.*"
```

In the cell below, choose the training dataset text for the domain you've chosen and update the code in the cell below:

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
!aws configure

role_arn = "arn:aws:iam::317957367373:role/SageMakerExecutionRole"

model_id = "meta-textgeneration-llama-2-7b"

estimator = JumpStartEstimator(
    model_id=model_id,
    environment={"accept_eula": "true"},
    instance_type="ml.g5.8xlarge",
    role=role_arn
)

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

training_data_s3_path = "s3://sagemaker-us-east-1-317957367373/financialDataset.txt"

estimator.fit({"training": training_data_s3_path})
```

```
Preparing metadata (setup.py): started
Preparing metadata (setup.py): finished with status 'done'
Processing ./lib/huggingface-hub/huggingface_hub-0.24.2-py3-none-any.whl (from -r requirements.txt (line 8))
Processing ./lib/inflate64/inflate64-0.3.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (from -r requirements.
```

```
Processing ./lib/scipy/scipy-1.11.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (from -r requirements.txt (li
Processing ./lib/shtab/shtab-1.7.1-py3-none-any.whl (from -r requirements.txt (line 34))
Processing ./lib/termcolor/termcolor-2.3.0-py3-none-any.whl (from -r requirements.txt (line 35))
Processing ./lib/texttable/texttable-1.6.7-py2.py3-none-any.whl (from -r requirements.txt (line 36))
Processing ./lib/tokenize-rt/tokenize_rt-5.1.0-py2.py3-none-any.whl (from -r requirements.txt (line 37))
Processing ./lib/tokenizers/tokenizers-0.19.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (from -r requiremen
Processing ./lib/torch/torch-2.2.0-cp310-cp310-manylinux1_x86_64.whl (from -r requirements.txt (line 39))
Processing ./lib/transformers/transformers-4.43.1-py3-none-any.whl (from -r requirements.txt (line 40))
Processing ./lib/triton/triton-2.2.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (from -r requirements.txt (l
Processing ./lib/trl/trl-0.8.1-py3-none-any.whl (from -r requirements.txt (line 42))
Processing ./lib/typing-extensions/typing_extensions-4.8.0-py3-none-any.whl (from -r requirements.txt (line 43))
Processing ./lib/tyro/tyro-0.7.3-py3-none-any.whl (from -r requirements.txt (line 44))
Processing ./lib/sagemaker_jumpstart_script_utilities/sagemaker_jumpstart_script_utilities-1.1.9-py2.py3-none-any.whl (from
Processing ./lib/sagemaker_jumpstart_huggingface_script_utilities/sagemaker_jumpstart_huggingface_script_utilities-1.2.7-py2
Requirement already satisfied: numpy<2.0.0,>=1.17 in /opt/conda/lib/python3.10/site-packages (from accelerate==0.33.0->-r re
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.10/site-packages (from accelerate==0.33.0->-r requi
Requirement already satisfied: psutil in /opt/conda/lib/python3.10/site-packages (from accelerate==0.33.0->-r requirements.t
Requirement already satisfied: pyyaml in /opt/conda/lib/python3.10/site-packages (from accelerate==0.33.0->-r requirements.t
Requirement already satisfied: click>=8.0.0 in /opt/conda/lib/python3.10/site-packages (from black==23.7.0->-r requirements.
Requirement already satisfied: platformdirs>=2 in /opt/conda/lib/python3.10/site-packages (from black==23.7.0->-r requiremen
Requirement already satisfied: tomli>=1.1.0 in /opt/conda/lib/python3.10/site-packages (from black==23.7.0->-r requirements.
Requirement already satisfied: pyarrow>=8.0.0 in /opt/conda/lib/python3.10/site-packages (from datasets==2.14.1->-r requirem
Requirement already satisfied: dill<0.3.8,>=0.3.0 in /opt/conda/lib/python3.10/site-packages (from datasets==2.14.1->-r requ
Requirement already satisfied: pandas in /opt/conda/lib/python3.10/site-packages (from datasets==2.14.1->-r requirements.txt
Requirement already satisfied: requests>=2.19.0 in /opt/conda/lib/python3.10/site-packages (from datasets==2.14.1->-r requir
Requirement already satisfied: tqdm>=4.62.1 in /opt/conda/lib/python3.10/site-packages (from datasets==2.14.1->-r requiremen
Requirement already satisfied: xxhash in /opt/conda/lib/python3.10/site-packages (from datasets==2.14.1->-r requirements.txt
Requirement already satisfied: multiprocess in /opt/conda/lib/python3.10/site-packages (from datasets==2.14.1->-r requiremen
Requirement already satisfied: fsspec>=2021.11.1 in /opt/conda/lib/python3.10/site-packages (from fsspec[http]>=2021.11.1->d
Requirement already satisfied: aiohttp in /opt/conda/lib/python3.10/site-packages (from datasets==2.14.1->-r requirements.tx
Requirement already satisfied: six in /opt/conda/lib/python3.10/site-packages (from fire==0.5.0->-r requirements.txt (line 7
```

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.

```
#from sagemaker.serializers import JSONSerializer
#from sagemaker.deserializers import JSONDeserializer

finetuned_predictor = estimator.deploy(

#predictor = estimator.deploy(
     initial_instance_count=1,
     instance_type="ml.g5.4xlarge",
#    serializer=JSONSerializer(),
#    deserializer=JSONDeserializer(),
)
```

```
----------!
```

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

```
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"

- "Traditional approaches to data management such as"
- "A second important aspect of ubiquitous computing environments is"
- "because ubiquitous computing is intended to"
- "outline the key aspects of ubiquitous computing from a data management perspective."

```
payload = {
    "inputs": "the relative volume for the long out of the money options, indicates",
    "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 relative volume for the long out of the money options, indicates
> [{'generated_text': ' that the market maker is trying to keep the price in check.\nThe market maker will try to make sure

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

```
payload = {
    "inputs": "The results are encouraging for aggressive investors",
    "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 results are encouraging for aggressive investors
> [{'generated_text': ", but there are a number of risks to be aware of.\nThe S&P 500 (SNPINDEX:^GSPC) has been on a tear la

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

```
payload = {
    "inputs": "The results for the short in the money options",
    "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 results for the short in the money options
> [{'generated_text': ' are also shown in the chart below.\nThe results for the short in the money options are also shown in

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

```
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': ' that the proposed hybrid system can be a good alternative to the traditional grid-connected solar sy

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

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.

Use the output from this notebook to fill out the "model fine-tuning" section of the project documentation report

After you've filled out the report, run the cells below to delete the model deployment

```
IF YOU FAIL TO RUN THE CELLS BELOW YOU WILL RUN OUT OF BUDGET TO COMPLETE THE PROJECT
```

```
finetuned_predictor.delete_model()
finetuned_predictor.delete_endpoint()
```