Group Project / Assignment 4: Instruction finetuning a Llama-3.2 model

Assignment due 21 April 11:59pm

Welcome to the fourth and final assignment for 50.055 Machine Learning Operations. The third and fourth assignment together form the course group project. You will continue the work on a chatbot which can answer questions about SUTD to prospective students.

This assignment is a group assignment.

- Read the instructions in this notebook carefully
- Add your solution code and answers in the appropriate places. The questions are marked as **QUESTION:**, the places where you need to add your code and text answers are marked as **ADD YOUR SOLUTION HERE**. The assignment is more open-ended than previous assignments, i.e. you have more freedom how to solve the problem and how to structure your code.
- The completed notebook, including your added code and generated output will be your submission for the assignment.
- The notebook should execute without errors from start to finish when you select "Restart Kernel and Run All Cells..". Please test this before submission.
- Use the SUTD Education Cluster to solve and test the assignment. If you work on another environment, minimally test your work on the SUTD Education Cluster.

Rubric for assessment

Your submission will be graded using the following criteria.

1. Code executes: your code should execute without errors. The SUTD Education cluster should be used to ensure the same execution environment.
2. Correctness: the code should produce the correct result or the text answer should state the factual correct answer.
3. Style: your code should be written in a way that is clean and efficient. Your text answers should be relevant, concise and easy to understand.
4. Partial marks will be awarded for partially correct solutions.
5. Creativity and innovation: in this assignment you have more freedom to design your solution, compared to the first assignments. You can show of your creativity and innovative mindset.
6. There is a maximum of 310 points for this assignment.

ChatGPT policy

If you use AI tools, such as ChatGPT, to solve the assignment questions, you need to be

transparent about its use and mark AI-generated content as such. In particular, you should include the following in addition to your final answer:

- A copy or screenshot of the prompt you used
- The name of the AI model
- The AI generated output
- An explanation why the answer is correct or what you had to change to arrive at the correct answer

Assignment Notes: Please make sure to save the notebook as you go along. Submission Instructions are located at the bottom of the notebook.

Finetuning LLMs

The goal of the assignment is to build a more advanced chatbot that can talk to prospective students and answer questions about SUTD.

We will finetune a smaller 1B LLM on question-answer pairs which we synthetically generate. Then we will compare the finetuned and non-finetuned LLMs with and without RAG to see if we were able to improve the SUTD chatbot answer quality.

We'll be leveraging `langchain`, `llama 3.2` and `Google AI STudio with Gemini 2.0`.

Check out the docs:

- LangChain
- Llama 3.2
- Google AI Studio

Note: Google AI Studio provides a lot of free tokens but has certain rate limits. Write your code in a way that it can handle these limits.

Install dependencies

Use pip to install all required dependencies of this assignment in the cell below. Make sure to test this on the SUTD cluster as different environments have different software pre-installed.

In [ ]:
```
# QUESTION: Install and import all required packages
# The rest of your code should execute without any import or dependency errors.

# **--- ADD YOUR SOLUTION HERE (10 points) ---**
```

Generate training data

The first step of the assignment is generating synthetic question-answer pairs which can be used for finetuning an LLM model. Use the Google AI studio with the Gemini models to create -high-quality QA training data.

In [ ]:
```
# QUESTION: Use langchain and the Google AI Studio APIs and a model from the Gemini
# to create a text-generation chain that can produce and parse JSON output.
# Test it by having the LLM generate a JSON array of 3 fruits

#--- ADD YOUR SOLUTION HERE (20 points)---
```

Generate topics

When generating data, it is often helpful to guide the generation process through some hierachical structure. Before we create question-answer pairs, let's generate some topics which the questions should be about.

In [ ]:
```
# QUESTION: Create a function 'generate_topics' which generates topics which prospe
#
# Generate a list of 20 topics

#--- ADD YOUR SOLUTION HERE (20 points)---
```

In [ ]:
```
# test topic generation
print(generate_topics(3))
```

In [ ]:
```
# Generate a list of 20 topics
# We save a copy to disk and reload it from there if the file exists
```

Generate questions

Now generate a set of questions about each topic

In [ ]:
```
# QUESTION: Create a function 'generate_questions' which generates quetions about a
# Generate a list of 10 questions per topics. In total you should have 200 question
#

#--- ADD YOUR SOLUTION HERE (20 points)---
```

In [ ]:
```
# test it
print(generate_questions("Academic Reputation and Program Quality", 3))
```

In [ ]:
```
# # QUESTION: Now let's put it together and generate 10 questions for each topic. S

#--- ADD YOUR SOLUTION HERE (20 points)---
```

Generate Answers

Now create answers for the questions.

You can use the Google AI Studio Gemini model (assuming that they are good enough to generate good answers), your RAG system from assignment 3 or any other method you choose to generate answers for your question dataset.

Note: it is normal that some LLM calls fail, even with retry, so maybe you end up with less than 200 QA pairs but it should be at least 160 QA pairs.

In [ ]:
```
# QUESTION: Generate answers to al your questions using Gemini, your SUTD RAG syste
# Split your dataset in to 80% training and 20% test dataset.
# Store all questions and answer pairs in a huggingface dataset `sutd_qa_dataset` a

#--- ADD YOUR SOLUTION HERE (40 points)---
```

In [ ]:
```
# test the chain
question = "When was SUTD founded?"

# Now run the answer generation chain
response = generate_answer(question)
print("\nModel Response:")
print(response["answer"])
```

In [ ]:
```
# now run the chain for all questions to collect context and generate answers
```

Finetune Llama 3.2 1B model

Now use your SUTD QA dataset training data set to finetune a smaller Llama 3.2 1B LLM using parameter-efficient finetuning (PEFT). We recommend the unsloth library but you are free to choose other frameworks. You can decide the parameters for the finetuning. Push your finetuned model to Huggingface.

Then we will compare the finetuned and non-finetuned LLMs with and without RAG to see if we were able to improve the SUTD chatbot answer quality.

In [ ]:
```
# QUESTION: Finetune a Llama 3.2 1B model on the training split of your SUTD QA dat
# You need to prepare your dataset accordingly and set the hyperparameters for the
# Push your finetuned model to the Hugginface model hub {YOUR_HF_NAME}/llama-3.2-1B

#--- ADD YOUR SOLUTION HERE (50 points)---
```

In [ ]:
```
# QUESTION: Load a non-finetuned Llama 3.2 1B model and your finetuned SUTD QA Llam
# Ask it a simple test question (e.g. "What is special about SUTD?") to check that

#--- ADD YOUR SOLUTION HERE (10 points)---
```

In [ ]:
```
# try out the llms

query = "What is special about SUTD?"

print("Question:", query)
response_base = llm_base.invoke(query,  pipeline_kwargs={"max_new_tokens": 512})
print("Answer base:", response_base)

print("---------")
response_finetune = llm_finetune.invoke(query, pipeline_kwargs={"max_new_tokens": 5
print("Answer finetune:", response_finetune)
```

Integrate and evaluate

Now integrate both the non-finetuned Llama 3.2 1B model and your finetuned model into your SUTD chatbot RAG system. Generate responses to the 20 questions you have collected in assignment 3 using these 4 appraoches

1. non-finetuned Llama 3.2 1B model without RAG
2. finetuned Llama 3.2 1B SUTD QA model without RAG
3. non-finetuned Llama 3.2 1B model with RAG
4. finetuned Llama 3.2 1B SUTD QA model with RAG

Compare the responses and decide what system produces the most accurate and high quality responses

In [ ]:
```
# QUESTION: Re-create the RAG chatbot system you have created in assignment 3 but w

#--- ADD YOUR SOLUTION HERE (40 points)---
```

Bonus points: LLM-as-judge evaluation

Implement an LLM-as-judge pipeline to assess the quality of the different system (finetuned vs. non-fintuned, RAG vs no RAG)

In [ ]:
```
# QUESTION: Implement an LLM-as-judge pipeline to assess the quality of the differe

#--- ADD YOUR SOLUTION HERE (40 points)---
```

Bonus points: chatbot UI

Implement a web UI frontend for your chatbot that you can demo in class.

In [ ]:
```
# QUESTION: Implement a web UI frontend for your chatbot that you can demo in class
```

```
#--- ADD YOUR SOLUTION HERE (40 points)---
```

End

This concludes assignment 4.

Please submit this notebook with your answers and the generated output cells as a **Jupyter notebook file** via github.

Every group member should do the following submission steps:

1. Create a private github repository **sutd_5055mlop** under your github user.
2. Add your instructors as collaborator: ddahlmeier and lucainiaoge
3. Save your submission as assignment_04_GROUP_NAME.ipynb where GROUP_NAME is the name of the group you have registered.
4. Push the submission files to your repo
5. Submit the link to the repo via eDimensions

Assignment due 21 April 2025 11:59pm