

Research Report for Assignment 1

1. Hardware & System Environment

Hardware: Google Colab cloud instance (NVIDIA GPU, CUDA-enabled)

Operating System: Linux

Local machine: MacBook (macOS, browser-based access)

2. Training Environment Setup

All the codes were functioned under Python 3.12. To set up training environment used for data import, fine-tuning and model evaluation, I imported following libraries:

```
!pip install -U transformers accelerate peft datasets
!pip install finnhub-python
!pip install -U bitsandbytes>=0.46.1
import os
import re
import csv
import math
import time
import json
import random
import finnhub
import datasets
import pandas as pd
import yfinance as yf
from datetime import datetime
from collections import defaultdict
from datasets import Dataset
from openai import OpenAI
import torch
from datasets import Dataset
from transformers import (
    AutoModelForCausalLM,
    AutoTokenizer,
    TrainingArguments,
    Trainer,
    DataCollatorForLanguageModeling,
    BitsAndBytesConfig # Added BitsAndBytesConfig import
)
from peft import LoraConfig, get_peft_model
```

During the process of code functioning, each time training model I would encounter the problem that there existed out-of-memory crashes. To solve this, firstly, I changed the imported model Qwen2.5-7B-Instruct into Qwen2.5-1.5B-Instruct. But the crash happened as well. Next, I modified the max sequence length from 4096 to 1024, but the platform still did not have enough memory. So last, I switched from full training to inference-only evaluation. And finally there turned out a result.

3. Brief Analysis of the Dow30 Dataset

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
warnings.warn(
Train size: 1230
Test size: 300
```

	prompt	answer	period	label	symbol
0	[INST]<<SYS>>\nYou are a seasoned stock market...	[Positive Developments]:\n1. Market Outperform...	2023-05-14 to 2023-05-21	up by 3-4%	AXP
1	[INST]<<SYS>>\nYou are a seasoned stock market...	[Positive Developments]:\n1. American Express'...	2023-05-21 to 2023-05-28	up by 2-3%	AXP
2	[INST]<<SYS>>\nYou are a seasoned stock market...	[Positive Developments]:\n1. Increased dividen...	2023-05-28 to 2023-06-04	up by more than 5%	AXP
3	[INST]<<SYS>>\nYou are a seasoned stock market...	[Positive Developments]:\n1. The stock outperf...	2023-06-04 to 2023-06-11	up by 1-2%	AXP
4	[INST]<<SYS>>\nYou are a seasoned stock market...	[Positive Developments]:\n1. American Express ...	2023-06-11 to 2023-06-18	up by 0-1%	AXP

The dataset source is FinGPT Dow Jones 30 dataset. And time span is from May 2023 to May 2024. The train size is 1230 while the test size is 300. The structure of dataset is made up with prompt, answer, period, label and symbol.

4. The process of fine-tuning models

I implemented the LoRA fine-tuning pipeline following the PEFT framework. The training process was tested on Qwen models, while attempts on larger models such as Llama-3.1-8B and DeepSeek-R1-Distill-Llama-8B were limited by hardware constraints and access permissions.

LoRA process is successfully finished as below:

```
from peft import LoraConfig, get_peft_model, TaskType

peft_config = LoraConfig(
    task_type=TaskType.CAUSAL_LM,
    inference_mode=False,
    r=8,
    lora_alpha=16,
    lora_dropout=0.1,
```

```
target_modules=[  
    "q_proj", "k_proj", "v_proj",  
    "o_proj",  
    "gate_proj", "up_proj", "down_proj",  
],  
bias="none",  
)
```

```
model = get_peft_model(model, peft_config)  
model.print_trainable_parameters()
```

```
trainable params: 9,232,384 || all params: 1,552,946,688 || trainable%: 0.5945
```

Instead of updating all parameters of the pretrained LLM, LoRA injects low-rank trainable adapters into selected linear layers. During fine-tuning, only these adapters are optimized, while the original model weights remain frozen. The proportion of parameters trained is 0.5945%.

However, when training models, the crash happened due to training instability on limited hardware. So I adopted an inference-based comparison approach to evaluate model behavior.

5. Evaluation Metrics Results

(1) Inference Time Cost

Qwen-1.5B inference is fast and stable, and larger models could not be reliably loaded in constrained environments.

(2) Accuracy of Execution

The accuracy of execution can be decided by whether outputs follow prompt instructions, whether answers mention relevant financial entities and whether reasoning steps are coherent. Based on an inference output shown below, the model gave a relatively comprehensive answer to the prompt and was able to generate responses that referenced macroeconomic factors such as inflation and interest rates. Overall, the reasoning process was logically coherent, but remained relatively high-level due to the limited extent of fine-tuning and resource constraints.

You are a financial analyst.

Market information:

Dow Jones Index fell by 1.2% today.
Major declines were observed in financial and technology stocks.
Investors showed concerns about interest rate uncertainty and inflation.

Question:

What is today's market sentiment and possible short-term outlook?

Today's market sentiment seems to be pessimistic, with investors expressing concern over the uncertain nature of interest rates and potential in

Possible short-term outlook:

- There might be further decline in stock prices due to continued uncertainty around monetary policy and economic growth expectations.
- Market participants could seek out defensive assets like bonds or cash equivalents to protect against further losses.
- Some traders may take advantage of the

(3) Depth of Analysis

Depth of analysis was assessed qualitatively through inference-based evaluation.

Although full instruction fine-tuning could not be completed due to hardware constraints, the instruction-tuned base model (Qwen2.5-Instruct) demonstrated the ability to reference macroeconomic trends such as inflation and interest rate changes. The responses incorporated relevant financial terminology and maintained a coherent analytical structure.

However, the depth of analysis remained relatively high-level, and no quantitative reasoning was observed, reflecting the limitations of inference-only evaluation and partial fine-tuning attempts.

(4) Clarity & Professionalism

From the perspective of clarity, the answer listed possible points one by one and firstly state conclusion followed by explanation. So the extent of clarity is enough. From the perspective of professionalism, the model demonstrates basic financial literacy by incorporating commonly used financial terminology in relevant contexts such as 'defensive assets', 'cash equivalents' and so on.