Project 5: LLM

Yelin Zhang

Preprocessing

- Question and passage into prompt template

Question: ...\n ### Context: ...\n ### Answer: ...

- Labels into strings (True -> "true", False -> "false")
- Absolutely nothing else (Lower case, stemming, word remval, etc).

- LLama tokenizer: TikToken

Input/output format

- Max context length: 8k

- Input: Prompt from preprocessing (### Question: …\n ### Context: …\n ### Answer: …)
- Output: 1 token which represents the answer ("true", "false")

- Train and Validation data has answer included in prompt for supervised fine tuning
- Test does not have the answer in prompt

Network architecture

- LLama 3.2
- Embedding: In 128256, Embedding dim 2048
- 16 Llama Transformer layers
- Head: Linear, In 2048, Out 12825

- Optimizer: AdamW
- Loss: Cross entropy loss
- LR: 1e-4, Cosine scheduling

Experiments

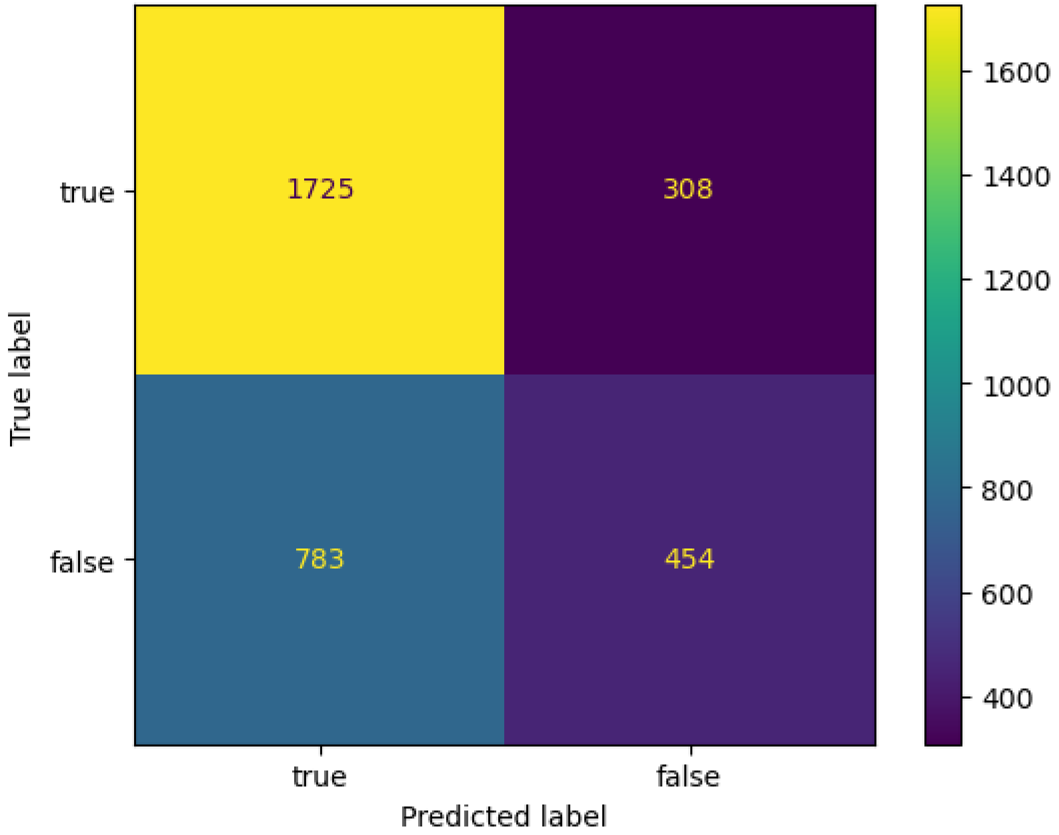
- Lora parameters
- R: 1,16,128,256
    - Lora attention dimension: the higher the more parameters can be changed by Lora
- Alpha: 0,1,16,128,256
    - Lora scaling: Scales the Lora weights, how strongly the weights are affected
- Max experiments: 4*5=20

Lots of conflicting and unclear info online therefore, try and see what happens

Results

	Train	Valid	Test
Loss	15.1237	1.87837	-
Accuracy	0.6829	0.665	0.6663

Best parameter: Alpha 256, R 256



Comparison test metrics

	Majority class	Word embeddings	RNN	Transformer	Pre trained Transformer	LLM
Accuracy	0.6218	0.6165	0.6397	0.6287	0.6218	0.6663

🎉 Last project best accuracy 🎉

Interpretation

- Lora Alpha seems a lot more important than R
- The impact it alpha had on the loss curve is noticeable, meanwhile R seems negligible
- Only 1 epoch is enough to fine tune for simple classification
- 1B model should be good enough, but SFT might be messing with the weights too much