Análise Quantitativa do Trade-off entre Especialização e Generalização em LLMs via Fine-Tuning

Gabriel da Silva Freitas
Instituto de Computação - ICOMP
Universidade Federal do Amazonas - UFAM
Manaus, Brasil
gabriel.freitas@icomp.ufam.edu.br

***Index Terms*—component, formatting, style, styling, insert**

I. Metodologia

Foram criados 3 arquivos para a execução deste trabalho. Os arquivos são:

1) preprocess.py
2) train.py
3) evaluate.py

1) preprocess.py: O código preprocess.py carrega o dataset spider diretamente da função load_dataset(). Em seguida ele seleciona alguns exemplos da base de dados apra inserir no prompt-few-shot. A proxima etapa é dividir os dados em treinamento e validação. Isso é feito no código apresentado na figura 1.

```
val_data = spider["validation"].map(process_val_example)

# 4. Salvar dados processados
train_data.save_to_disk(f"{save_path}/train")
val_data.save_to_disk(f"{save_path}/validation")
```

Fig. 1. Divisão e salvamento da base de dados

Nesse código, também definida uma função para gerar o prompt-few-shot para o modelo **google/gemma2bit**. Essa função atende à forma de padronização de fine-tunning de usuário-resposta para o gemma-2b. A figura **??** exibe a codificação da função.

```
def create_prompt(question, few_shot_examples, include_answer=True):
    """
    Cria prompt no formato do modelo google/gemma-2b-it.
    """
    system_msg = "You are a SQL expert. Convert the following natural language questions to SQL queries."
    # (começa com uma instrução de sistema (opcional), mas ajuda a alinhar)
    prompt = (
        "<start_of_turn>user\n"
        f"{system_msg}\n"
        "<end_of_turn>\n"
    )
    # Adiciona exemplos few-shot
    for ex in few_shot_examples:
        prompt += (
            "<start_of_turn>user\n"
            f"Question: {ex['question']}\n"
            "<end_of_turn>\n"
            "<start_of_turn>model\n"
            f"SQL: {ex['query']}\n"
            "<end_of_turn>\n"
        )
    # Adiciona a pergunta atual
    prompt += (
        "<start_of_turn>user\n"
        f"Question: {question}\n"
        "<end_of_turn>\n"
    )
    if include_answer:
        prompt += "<start_of_turn>model\nSQL:"  # Vou, 3 hours ago • ao ...
    return prompt
```

Fig. 2. Enter Caption

A. Dados

O tratamento dos dados que irão ser utilizados no modelo segue as etapas apresentadas no seguinte passo a passo:

- **Pré-processamento:** Os dados foram previamente processados e utilizando a biblioteca datasets do Hugging Face. Foram divididos em dois conjuntos: treinamento (processed_data/train) e validação (processed_data/validation).
- **Carregamento dos dados:** A função load_from_disk() recupera ambos os conjuntos. Cada amostra dos dados possui um campo de texto denominado ”text”, que é usado diretamente pelo modelo durante o treinamento.
- **Tokenização:** O tokenizador é configurado com as seguintes propriedades
  - O Padding é realizado no lado direito das sequências (padding_side=”right”).
  - O Token de padding é definido como o token de finalização (eos_token), garantindo que não haja ambiguidade entre fim de sequência e padding.
  - O Comprimento máximo das sequências é 1024 tokens (max_seq_length=1024).
- **Distribuição dos Dados:** O parâmetro group_by_length=True garante que os dados sejam agrupados por tamanho de sequência, otimizando o uso de memória e acelerando o treinamento.

B. Configuração do LoRA

A técnica LoRA foi empregada para adaptar o modelo de maneira eficiente, reduzindo o número de parâmetros treináveis e o custo computacional. A adaptação ocorre nas projeções de atenção chave (q_proj) e valor (v_proj), que são os módulos mais sensíveis e impactantes no comportamento do modelo.

C. Configurações do modelo

O modelo escolhido foi **o google/gemma2bit**. Esse modelo foi escolhido devido a falta de uma GPU que tenha consiga surportar os modelos **mistralai/Mistral7BInstructv0.2** e

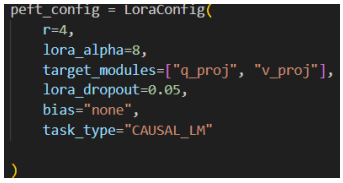

Fig. 3. Modelo do peft para configuração do LoRA

metallama/Llama38BInstruct. Após várias tentativas de usar esses modelos de 7B ou 8B de parâmetros, foi decidido que o trabalho não iria a lugar nenhum e também não havia mais tempo para ficar esperando por esses modelos. Então decidi usar esse e tentar pelo menos chegar em algum lugar. Mesmo assim, utilizei quantização de 4-bit para otimização e eficiência computacional.

D. Configurações do treinamento

O treinamento foi conduzido utilizando o SFTTrainer da biblioteca trl com os parâmetros apresentados na figura 4. O treinamento é monitorado via TensorBoard, com checkpoints regulares e uso de EarlyStopping para evitar overfitting. Seeds foram fixados tanto para CPU (torch.manual_seed(42)) quanto para GPU (torch.cuda.manual_seed_all(42)), assegurando resultados consistentes entre execuções.

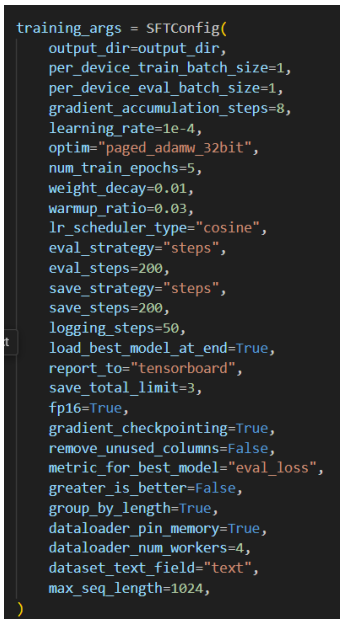

Fig. 4. Argumentos para o treinamento

II. Resultados

III. Discussão

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, “Title of paper if known,” unpublished.
[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.