Internship weekly report

Mathis Embit

MILES, LAMSADE

May 27, 2024

Outline

Introduction

This week: code.

Custom or Hugging Face

2 choices:

- Make my code clean and continue with it
- Use Hugging Face PEFT library

For now we will use Hugging Face PEFT library.

Hugging Face PEFT

Hugging Face PEFT [Man+22] is a collection of Parameter-Efficient Fine-Tuning (PEFT) methods such as LoRA or Prefix tuning.

Using HF PEFT

PEFT library is organized as follow:

- peft_model.py
- tuners/
  - lora/
  - ...
  - p_tuning/
  - prefix_tuning/
  - prompt_tuning/
- ...

We are intersted in 3 of their tuners: p_tuning, prefix_tuning, prompt_tuning (what they call prompt-based methods).

PeftModel

The PeftModel class defined in **peft_model.py** allows to create new model like this:

model = get_peft_model(model, peft_config)

It also provides child classes: PeftModelForSequenceClassification, PeftModelForCausalLM, PeftModelForSeq2SeqLM, PeftModelForTokenClassification, PeftModelForQuestionAnswering, PeftModelForFeatureExtraction.

tuners

Each prompt-based method tuner is organized as follow:

- **model.py** which contains the encoder used to generate the virtual tokens via a forward method
- **config.py** which contains initialization specifications, tokenizer, encoder config (nb of layer, hidden size, ...), ...

The encoders are:

- an LSTM for p_tuning
- an MLP for prefix_tuning
- a single embedding matrix for prompt_tuning

References I

[Man+22] Sourab Mangrulkar et al. *PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods*. https://github.com/huggingface/peft. 2022.