Fine-tuning Language Model For Text Classification

Overview

Define Task

Text Classification (Multi-Class)

Hardware Setup

Operating System: Ubuntu 20.04 LTS
Graphic Card: Geforce RTX 1050 Ti
Graphic Driver: Nvidia 470 + CUDA 11.4
Google Colab: Tesla-T4
ML Framework: Pytorch

Model Selection

Encoder vs Decoder
BERTs VS LLMs

Parameter-Efficient Fine-Tuning (PEFT)

Low-Rank Adaptation (LoRA, QLoRA)

RIsk Classification

Supervised fine-tuned Encoder Model,
Few-Short Learned LLM + SFT Encoder Model

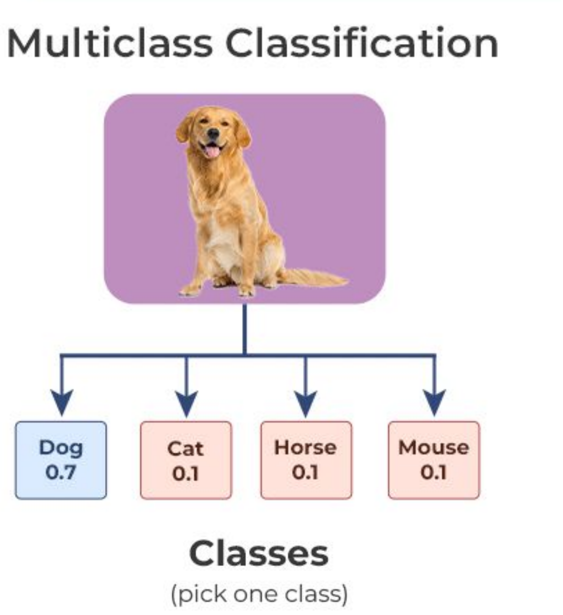
Model Consideration

Encoder based models: (BERT, DistilBERT, ...)

- Strong embedding capability as encoder based model
- SOTA performance on classic ML tasks including text-classificati
- Lightweight by today's standard

What about decoder based models such as LLMs? (Mistral-7B, Phi-2)

- Fine-tuning can be very computationally heavy.
- Few-short learning is usually enough for the task.
- Very good for generation task (translation, conversation), but for tasks that are very embedding focused, encoder type models usually performs better.



Encoder VS Decoder

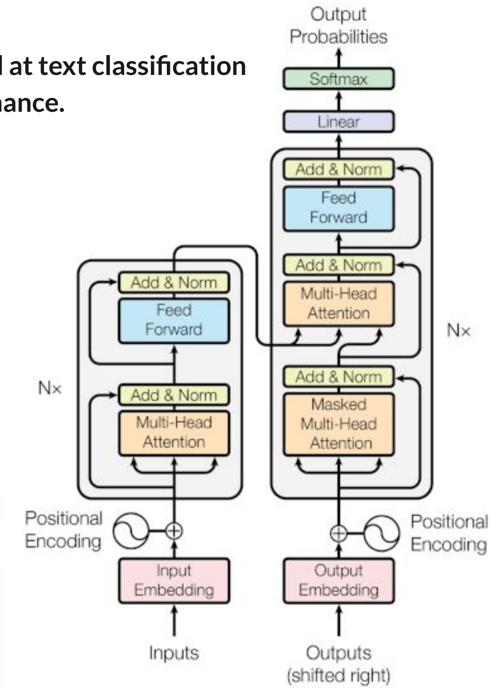
DistilBERT is selected for the following consideration:

- BERT based encoder models are already quite good at text classification
- 40% smaller than BERT while training 97% performance.
- LLMs are heavy to fine-tune

BERT

Encoder

Good at Analyzing Text but do not generate text *(BERT)*



GPT

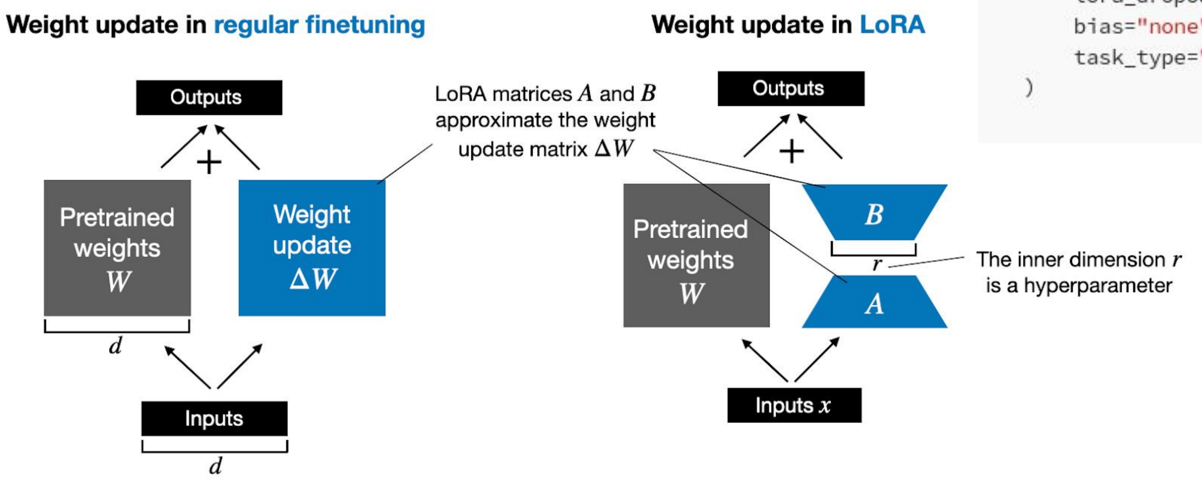
Decoder

Good at Generating Text with Prompts *(GPT series : GPT2,3,3.5,4, Turbo models, Mistral, Llama, Falcon, Vicuna etc)*

Low-Rank Adaptation (LoRA & QLoRA)

- Finetune large models with low compute
- Adapt large models in a low-data regime
- Further optimisation of QLoRA with the introduction of 4-bit quantization
- Easy to implement with the Hugging Face PEFT package.

```
from peft import LoraConfig
config = LoraConfig(
    r=8,
    lora_alpha=16,
    target_modules=["q", "v"],
    lora_dropout=0.01,
    bias="none"
    task_type="SEQ_2_SEQ_LM",
)
```



Application for Risk Classification

Supervised fine-tuned (SFT) BERT based encoder models:

- Binary Classification: Single risk type classification. Easiest to prepare data but require one model per risk type.
- Multi-Class Classification: Risk severity classification. (quantised output)
- Multi-Label Classification: Multiple risk types detection. Performance won't be as good as binary classifiers and class imbalance and be a challenge.
- Tabular Classification: Feed extra attributes related to the content as tabular data for training.

Few-Short Learned LLM (with SFT Encoder)

- Few-short prompt a LLM with a small set of representative (boundary) data.
- Use LLM directly for classification , or as a data labeller for labelling and augmentation to train a smaller model.

Thank you.