ML-Assignment2

Pallekonda Naveen Kumar(SR No:22915)

April 2024

1 Data

The Data used is the CoLA Dataset. The public version provided here contains 9594 sentences belonging to training and development sets.

Data Format : Each line in the .tsv files consists of 4 tab-separated columns.
Column 1: the code representing the source of the sentence.
Column 2: the acceptability judgment label (0=unacceptable, 1=acceptable).
Column 3: the acceptability judgment as originally notated by the author.

2 Problem-0

All the libraries are running successfully. python file executed successfully loading the gpt2 model having the 125.03M parameters. and generating the subsequence text for the given prompt.

3 Problem-1 : Low Rank Adaptation

3.1 1a

LoRA is an improved finetuning method where instead of finetuning all the weights that constitute the weight matrix (W) of the pre-trained large language model, two smaller matrices (L and R) that approximate the update to the matrix are fine-tuned.

$$W_0 + \Delta W = W_0 + LR$$

where W_0 $(d \times k)$, L $(d \times r)$, and R $(r \times k)$ and $r \ll d, k$. These matrices constitute the LoRA adapter. Here 'r' is a hyperparameter.I had used the default given rank of 4 because when I tried to use the different rank, Accuracies are decreasing so I used the default rank.

During training, W_0 is frozen and doesn't receive gradient updates, while L and R contain trainable parameters. Both W_0 and $\Delta W = LR$ are multiplied

with the same input, and their respective output vectors are summed coordinate-wise.

Initializations for the matrices can be done in many ways. I tried Kaiming uniform initialization initially I am getting accuracies of around 74. of I used th 0.1e Xavier uniform initialization(which is state of the art techiniques).

I used the dropout of percentage 0.1 that randomly leaves out some weight changes each time to deter overfitting. Code Program files are uploaded

3.2 1b

For Freezing the weights I used the weights.requires_grad = False. Neccesary injections for the LoRA Linear are written. filled all the TODOs.

Code program files are uploaded.

3.3 1c

Training Strategy GPT2 model was finedtuned for doing the classification task of two classes. All the hyperparameters are used the default except the number of epochs

I got the maximum accuracy with the gpt2-medium model of 82.92% validation accuracy.

GPT2-variants used are gpt2(base), gpt2-medium models

3.3.1 GPT variant : GPT2(base)

Optimal Hyperparameters

- Learning rate: 1e-3
- batch_size : 128
- epochs: 10(Initially I observed an increase in validation accuracy and a decrease so I used stopping condition. maximum accuracy reached around 3 or 4 epochs)
- LoRA rank: 4

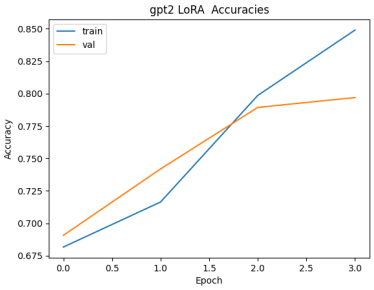
Model Details :

- Optimizer: Adam
- Loss Function: cross entropy loss
- Number of parameters: 125.03M
- Number of trainable parameters: 0.68M
- Reduction: 99.46%

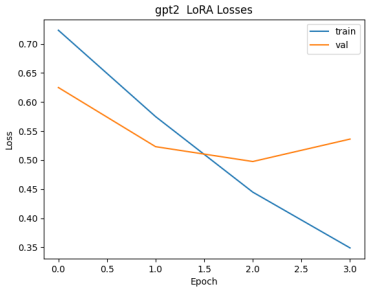
Maximum Accuracy reached when GPT2 base model is used is 79.70%.

As I observed after a constanlty increase in accuracy, after a particular point it started to decrease in validation accuracy so I used the stopping condition. when the accuracy is getting above 79 percent I am stopping (this is observation seen decreasing after reaching 79 percent).

Plots:



(a) GPT Base Model Accuracies vs epochs



(b) GPT Base Model Losses vs epochs

Figure 1: GPT2 Base model

3.3.2 GPT variant: GPT2-Medium

Maximum Accuracy reached when GPT2 medium model is used is 82.92% textbfOptimal Hyperparameters

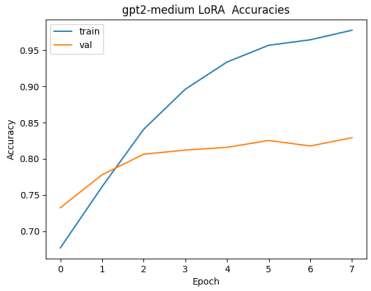
- Learning rate: 1e-3
- batch_size : 128
- epochs : 10(Initially I observed an increase in validation accuracy and a decrease so I used stopping condition. maximum accuracy reached around 7 or 8 epochs)
- LoRA rank : 4

Model Details :

- Optimizer: Adam
- Loss Function: cross entropy loss
- Number of parameters: 356.40M
- Number of trainable parameters: 1.80M
- Reduction: 99.50%

As I observed after a constanlty increase in accuracy, after a particular point it started to decrease in accuracy so I used the stopping condition. when the accuracy is getting above 82 .5percent I am stopping (this is observation seen decreasing after reaching 82.5 percent).

Plots:



(a) GPT2 Medium Model Accuracies vs epochs

(b) GPT2 Medium Model Losses vs epochs

Figure 2: GPT2 Medium model

NOTE: As I am using the GPT2 base model for Knowledge Distillation teacher model but I got maximum vadilation accuracy on CoLA validation set for the gpt2-medium model.

4 Problem-2 : Knowledge Distillation

GPT2 model is trained for classification in the above problem. Now need to distil the knowledge contained in the fine-tuned GPT2 model to a much smaller mode, this is known as Knowledge Distillation.

I used GPT2 Base model for the knowledge Distillation.

4.1 2a

Code program files are uploaded.

4.2 2b

The challenge of deploying large deep neural network models is especially with limited memory and computational capacity. To tackle this challenge, a model compression method was first proposed to transfer the knowledge from a large model into training a smaller model without any significant loss in performance.

In knowledge distillation, a small “student” model learns to mimic a large “teacher” model and leverage the knowledge of the teacher to obtain similar or higher accuracy.

So to distil the knowledge from the fine-tuned GPT2(teacher model) to the DistilRNN model(student model).We have introduced the new parameter called Temperature, that controls the softness of the softmax function and the loss coefficients.

T: Temperature controls the smoothness of the output distributions Larger T leads to smoother distributions, thus smaller probabilities get a larger boost.T value is 2.

loss function caluculation:

- Soften the student logits by applying softmax first and log() second. Teacher_soft_targets(softmax on teacher logits), student_soft_prob(log softmax on student logits).
- Calculate the soft target loss. Scaled by T**2 as suggested by the authors of the paper "Distilling the Knowledge in a Neural Network"

$$\sum (\text{Teacher\_soft\_targets} \cdot (\log(\text{Teacher\_soft\_targets}) - \text{student\_soft\_prob})) \cdot T^2 \tag{1}$$

Distil RNN Architecture:

In model.py, in the DistilRNN class created a RNN architecture with the embeddings layer,RNN layer and Linear layer.

- In the embedding layer the input tokens is mapped to the 768-dimensional embedding vector.(taken from the originally used gpt2 model).
- The embeddings are passed through the RNN layer with 768 input features, 768 hidden units, 1 layer, and batch-first set to True. This means it processes sequences in batch form and returns the output of the final timestep for each sequence
- Linear layer performs the final classification based on the output of the RNN. It takes the output from the last timestep of the RNN (which represents the entire sequence) and maps it to a 2-dimensional space (assuming a binary classification task).
- Teacher Model: GPT2 base model
- Student Model: DistilRNN

Optimal Parameters:

- learning rate : 1e-3
- batch size ; 128
- epochs : 5

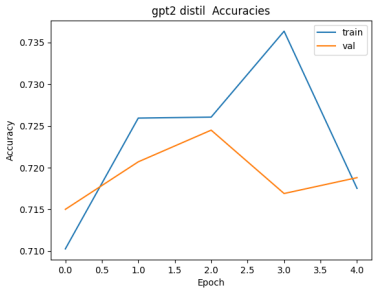
Model Details

- Number of parameters: 125.03M
- Number of trainable parameters: 0.68M
- Reduction: 99.46

Maximum Accuracy reached on CoLA validation dataset by Distil RNN model is 72.45%

Plots:

(I got good accuracies and losses with 3 epochs only but for the comparision with the student model in 2c so I ran the distil model for 5 epochs)



(a) Distill RNN accuracies vs Epochs

(b) Distill RNN losses vs Epochs

Figure 3: GPT2 distil RNN(with KD) model

4.3 2c

As you can see with Knowledge distillation(KD) accuracy obtained is up to 72.45% and without Knowledge distillation accuracy is 69.45%.

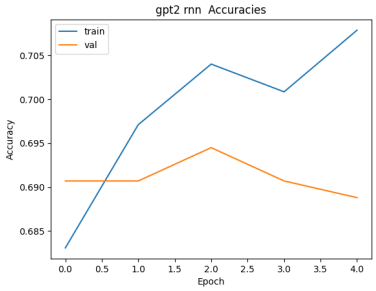
As the student model with KD has increase in acccuracy 3% than the student model without KD. Student model with knowledge transfer from teacher will have more sense of prior knowledge of the dataset than the student model directly on CoLA dataset without any knowledge.

Optimal Parameters:

- learning rate: 1e-3
- batch size: 128
- epochs: 5(as I see increase in accuracy upto 5 epochs)

Plots:

All the plots in the report will be in the folder inside Assignment2 folder as plots.



(a) RNN losses vs Epochs

(b) RNN losses vs Epochs

Figure 4: GPT2 RNN(without KD student) model