Machine Learning Assignment2

Alokam Gnaneswara Sai

April 2024

Question 1

Part (a)

Uploaded the files

Part (b)

Uploaded the files

Part (c)

Training Strategy

The GPT2 model was fine-tuned using a standard training loop implemented in the `trainutils.py` file. The GPT2 model was fine-tuned for the classification task on the COLA dataset, which is a binary classification problem. .The model was trained with the following hyperparameters:

- Learning Rate: `1e-3`
- Number of Epochs: `10`
- Batch Size: `128`
- Optimizer: Adam
- Loss Function: Cross-Entropy Loss

Model Details

I have utilized both the GPT2 base and medium variants, and the outcomes are as follows:

- GPT2 Variant Used: Medium
- Total Number of Parameters: `356.40M`
- Number of Trainable Parameters: `1.68M`
- Reduction in Parameters: `99.53%`

I have also experimented with the GPT2 base variant, with the following results:

- GPT2 Variant Used: Base
- Total Number of Parameters: `125.03M`
- Number of Trainable Parameters: `0.63M`
- Reduction in Parameters: `99.50%`

Optimal Hyperparameters

The optimal hyperparameters chosen for fine-tuning were:

- Learning Rate: `1e-3`
- Number of Epochs: `10`
- Batch Size: `128`
- LoRA Rank: `4`

Results

The maximum accuracy achieved on the COLA validation dataset was `82.73%` using gpt2 medium variant .

Plots

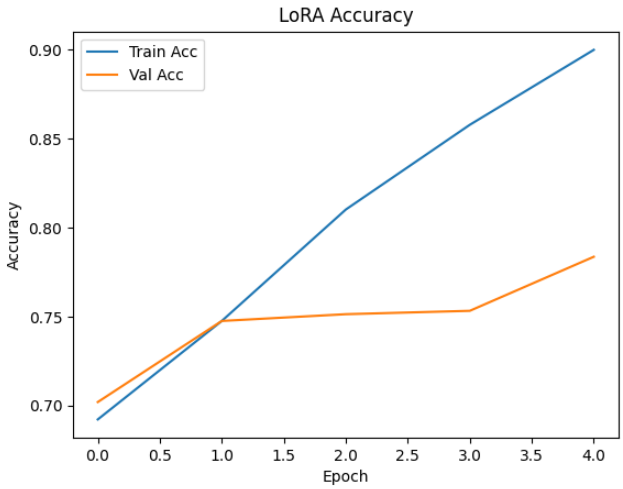

Figure 1: LoRA accuracy vs epoch plot for base variant

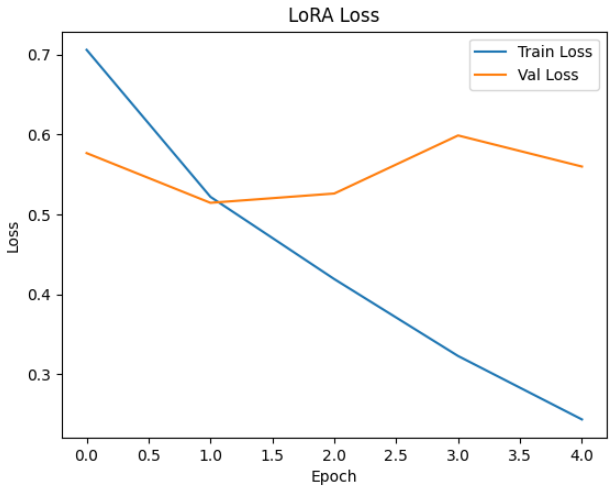

Figure 2: LoRA loss vs epoch plot for base variant

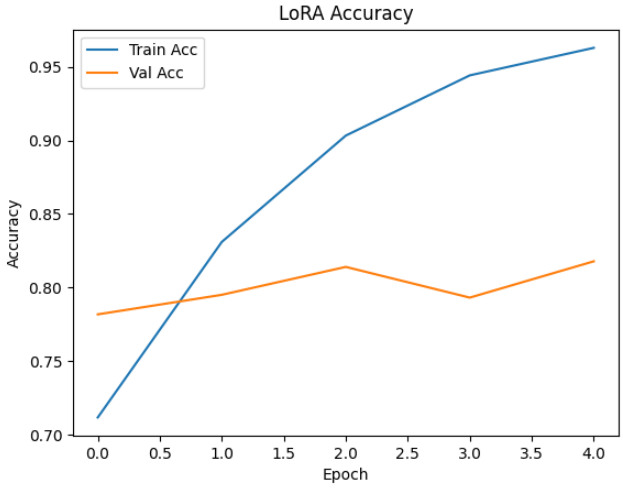

Figure 3: LoRA accuracy vs epoch plot for gpt2-medium variant

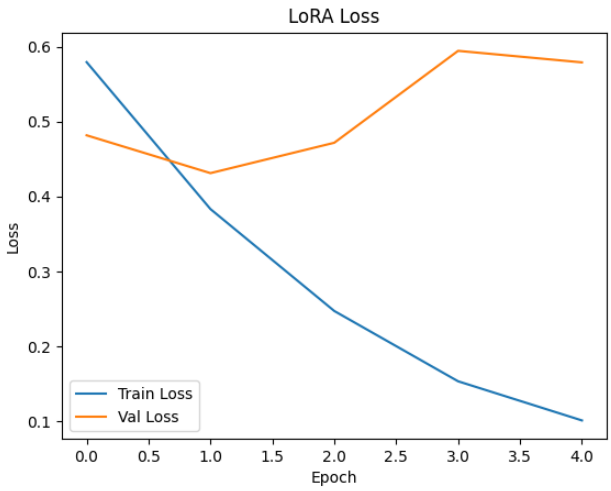

Figure 4: LoRA loss vs epoch plot for gpt2 medium variant

Question 2

Part (a)

Uploaded the files

Part (b)

Knowledge Distillation from GPT to DistilRNN

To distill knowledge from the fine-tuned GPT model (teacher model) to the DistilRNN model (student model) for the COLA classification dataset, the distillation loss function used is a combination of soft target loss and true label loss.

In this approach, the teacher model's output probabilities are softened using a temperature parameter $T = 2$, and the student model's output probabilities are also scaled with the same temperature. Then, the Kullback-Leibler divergence between the soft target probabilities and the student's predicted probabilities is calculated, which represents the soft target loss. Additionally, the true label loss, calculated using the standard cross-entropy loss, is also considered. y is true labels

$$cross\_entropy\_loss = CrossEntropyLoss(student\_output, y)$$

The final loss is a combination of these two losses.
In this approach, the distillation loss function is computed as follows:

$$distill\_loss = \frac{1}{N \times T^2} \times KLDivLoss\left(log\_softmax(student\_output), softmax(teacher\_output)\right) \tag{1}$$

Where:

- N is the batch size,
- T is the temperature parameter set to 2,
- $log\_softmax$ represents the logarithm of the softmax function,
- $softmax$ denotes the softmax function,
- $KLDivLoss$ represents the Kullback-Leibler divergence loss.

Additionally, the true label loss, calculated using the standard cross-entropy loss, is also considered. The final loss is a combination of these two losses, weighted by a factor $\alpha = 0.8$:

$$loss = \alpha \times distill\_loss + (1 - \alpha) \times cross\_entropy\_loss \tag{2}$$

DistilRNN Architecture

The architecture of the DistilRNN model is as follows:

- The model begins with an embedding layer, which maps input tokens to dense vectors of size 768.
- Next, the embedded sequences are passed through a two-layer RNN (Recurrent Neural Network) with hidden size 768. The RNN processes the input sequences in a batch-first manner.
- A ReLU activation function is applied to the output of the RNN to introduce non-linearity.
- Finally, a linear layer projects the output of the RNN to a 2-dimensional space, corresponding to the number of output classes (binary classification).

Optimal Training Hyperparameters

The optimal training hyperparameters used for training the DistilRNN model are as follows:

- Batch size: 128
- Learning rate: 1×10^{-3}
- Number of epochs: 5

Results

The maximum accuracy achieved on the COLA validation dataset by the DistilRNN model is reported along with the training plots, which illustrate the model's performance over the course of training.

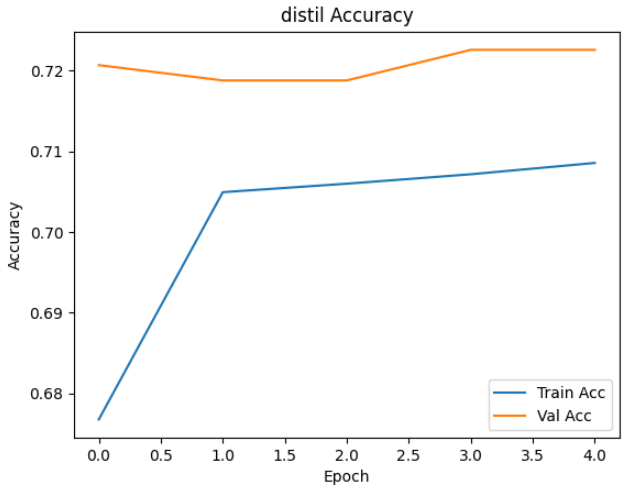

Figure 5: distil accuracy vs epoch plot

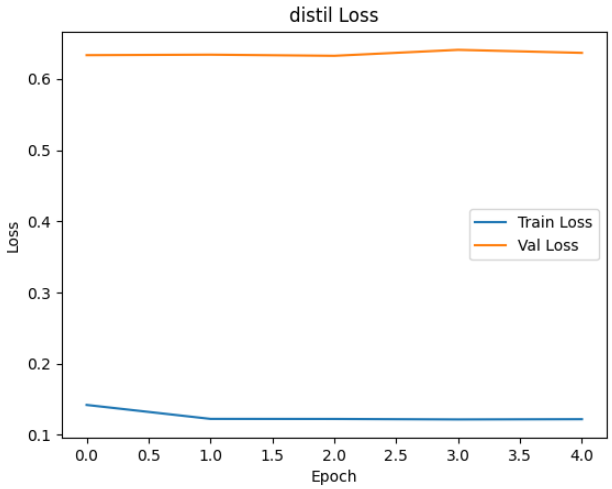

Figure 6: distil loss vs epoch plot

Part (c)

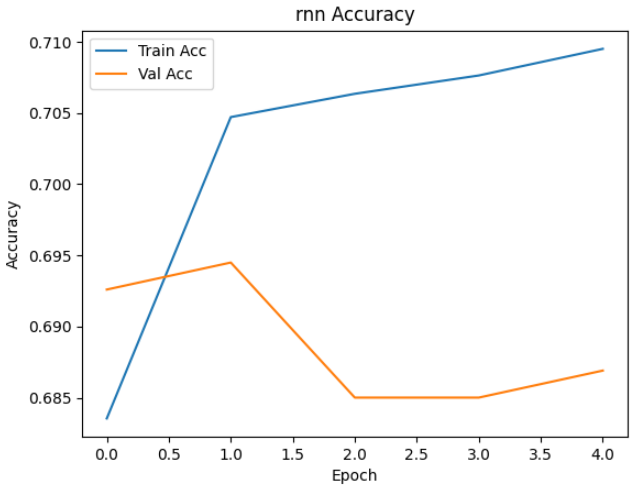

Figure 7: RNN accuracy vs epoch plot

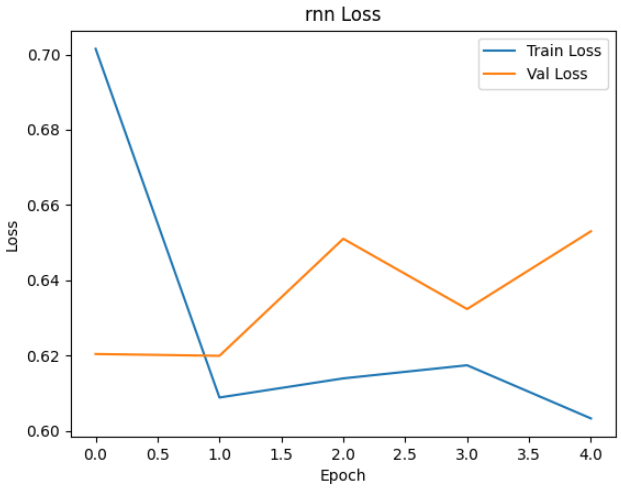

Figure 8: RNN loss vs epoch plot

To compare the performance of the DistilRNN model trained with knowledge distillation (KD) to the performance of the student RNN model trained without KD on the COLA validation set, observed a significant improvement in validation accuracy.

- The student RNN model achieved a validation accuracy of 68%.
- In contrast, the DistilRNN model trained with knowledge distillation achieved a validation accuracy of 71%.

This represents an increase of 3% in validation accuracy when using knowledge distillation compared to training the model without it.