Assignment 2

Yash Patel
SR Number: 22759

April 27, 2024

Dataset: COLA dataset

Each data point consists of a sentence containing tokens. These tokens can be tokenized using pre-trained tokenizers provided by Hugging Face. The tokenization process, as well as the conversion of tokens to their corresponding token IDs, has already been completed.

Question 1: Low Rank Adaptation

Implementation Details: LoRALinear Class

The `LoRALinear` class implements a linear layer using LoRA (Low-Rank Adaptation) decomposition. It provides the following functionality:

- **Initialization:** It takes input and output dimensions, a rank for the decomposition, and an α parameter. It initializes two parameters, X and Y, using random values scaled by the rank and zeros, respectively.
- **Forward Method:** It applies the LoRA decomposition to the input x using the parameters X and Y, and then scales the result by the α parameter.

This layer reduces the computational complexity of a standard linear layer by approximating the weight matrix with a lower-rank matrix, which can be more efficient in certain scenarios.

Training Strategy

- **Model Selection:** GPT-like model with LoRA layers.
- **Loss Function:** Cross-entropy loss for classification tasks, knowledge distillation loss for distillation.
- **Optimizer:** Adam optimizer with a learning rate of 2×10^{-5}.
- **Training Loop:** 3 epochs, printing and saving metrics every epoch.

Model Details

- **Total Parameters:** 125.03M
- **Trainable Parameters:** 0.63M
- **Reduction in Parameters:** 99.5%

Hyperparameters

- **Learning Rate:** 2×10^{-5}.
- **Alpha (KD):** 0.5.
- **Temperature (KD):** 2.

Results

- **Maximum Accuracy on COLA Validation Dataset:** 69.25%.
- **Training Plots:** Plots showing training and validation loss, and accuracy.

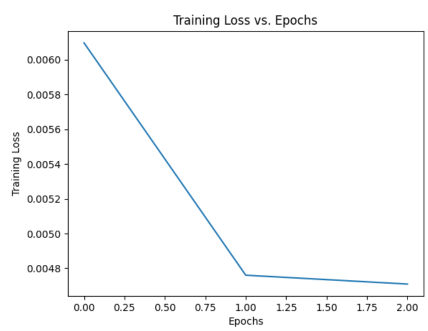

Figure 1: Training Loss VS Epochs

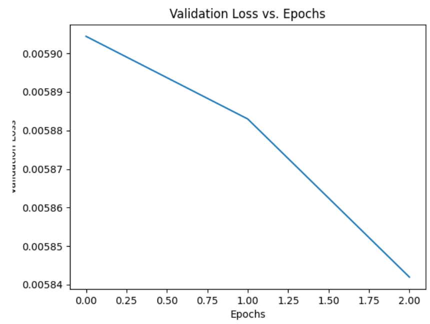

Figure 2: Validation Loss VS Epochs

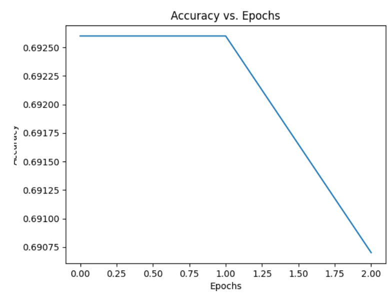

Figure 3: Accuracy VS Epochs

Question 2: Knowledge Distillation

Implementation Details

The 'DistilRNN' class implements a simple RNN-based model for sequence classification. Following are details about the implementation of it. Please refer to the code file to understand the variable naming used below.

- **Input Embedding**: The model starts with an embedding layer ('self.em') that converts input token indices into dense vectors.
- **Recurrent Neural Network (RNN)**: The RNN layer ('self.rnn') processes the embedded input sequences, maintaining a hidden state that captures information about the sequence history.
- **Output Layer**: The output of the RNN at the final time step is passed through a linear layer ('self.fc') to map it to the output space.
- **Forward Pass**: During the forward pass, input sequences are converted into dense embeddings using the embedding layer. These embeddings are

then processed by the RNN, which updates its hidden state at each time step. The final hidden state is used for classification.

The ‘DistilRNN‘ model uses an RNN architecture to learn representations of input sequences and make predictions for binary classification tasks.

Training Strategy

The DistilRNN model was trained using the knowledge distillation loss in addition to the standard cross-entropy loss. The model was trained for a total of 3 epochs using the Adam optimizer with a learning rate of 2×10^{-5}. The batch size was set to B.

Optimal Hyperparameters

The optimal hyperparameters for the DistilRNN model were found to be $\alpha = 0.5$ and $T = 2$, where α is the distillation weight and T is the temperature parameter.

Performance Metrics

The maximum accuracy achieved by the DistilRNN model on the COLA validation dataset was 69.07%.

Training Plots

The following plots show the training loss and accuracy of the DistilRNN model over the training epochs.

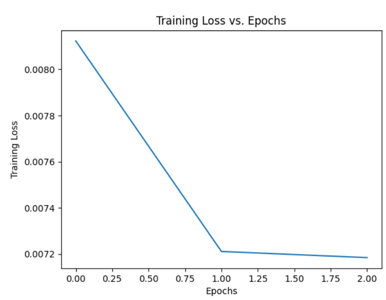

Figure 4: Training Loss VS Epochs

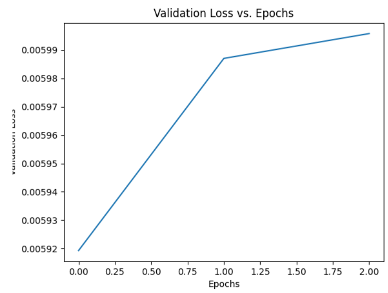

Figure 5: Validation Loss VS Epochs

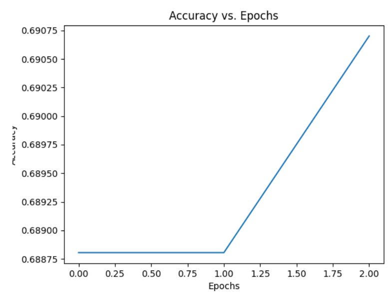

Figure 6: Accuracy VS Epochs

RNN Model

The RNN model is a basic recurrent neural network used for text classification.

Performance Metrics

The maximum accuracy achieved by the RNN model on the COLA validation dataset was 69.25%.

Training Plots

The following plots show the training loss and accuracy of the RNN model over the training epochs.

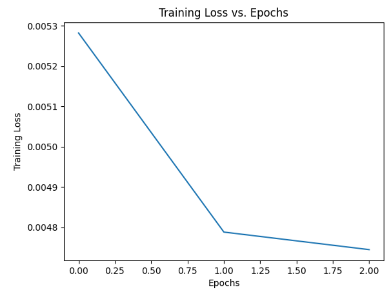

Figure 7: Training Loss VS Epochs

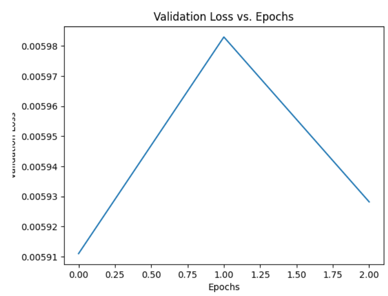

Figure 8: Validation Loss VS Epochs

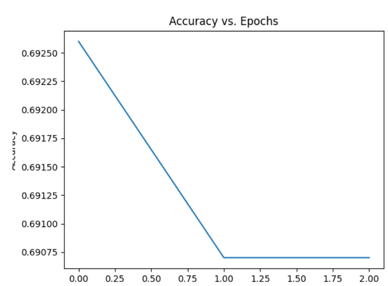

Figure 9: Accuracy VS Epochs