CPSC-8430 Deep Learning Homework 3 Report

Rayid Ali

March 2024

1 Introduction

The objective of this homework was to develop a question answering system using the DistilBERT model and train it on the Spoken-SQuAD dataset. The code implementation follows a structured approach, utilizing various libraries such as PyTorch, Transformers, and TQDM for efficient training and evaluation.

2 Data Loading and Preprocessing

The first step in the implementation is to load and preprocess the Spoken-SQuAD dataset. The `load_squad_data` function reads the JSON files containing the dataset and extracts the relevant information, such as contexts, questions, and answers. The function returns lists of contexts, questions, and answers for both the training and validation sets.

The `preprocess_answers` function is responsible for preprocessing the answers by adding the end index of each answer to the corresponding answer dictionary. It handles cases where the answer text is off by a character or two and adjusts the start and end indices accordingly.

3 Tokenization and Encoding

To prepare the data for input to the DistilBERT model, the code utilizes the `DistilBertTokenizerFast` from the Transformers library. The `tokenize_and_encode` function tokenizes the contexts and questions using the tokenizer and returns the encoded representations.

The `add_token_positions` function is crucial for aligning the answer positions with the token indices. It appends the start and end token positions of the answers to the encoding dictionary. If the start or end position is not found, it is set to the maximum model length or adjusted by shifting the position until a valid token index is found.

4 Dataset and DataLoader

To efficiently feed the data to the model during training and evaluation, the code defines a custom `SquadDataset` class that inherits from `torch.utils.data.Dataset`. This dataset class takes the encoded data and provides methods for accessing individual samples and their corresponding tensors.

The `DataLoader` from PyTorch is used to create data loaders for both the training and validation datasets. The data loaders handle batching and shuffling of the data during training.

5 Model Training

The `train_model` function is responsible for training the DistilBERT model on the Spoken-SQuAD dataset. It iterates over the specified number of epochs and performs the following steps for each batch:

1. Zero the gradients of the optimizer.

2. Move the input tensors to the device (GPU if available, CPU otherwise).
3. Forward pass the input through the model to obtain the outputs, including the loss.
4. Backward pass to compute the gradients.
5. Update the model parameters using the optimizer.

The training progress is displayed using the TQDM library, which provides a progress bar and metrics such as the current epoch and loss.

6 Model Evaluation

The `evaluate_model` function is used to evaluate the trained model on the validation dataset. It performs the following steps:

1. Set the model to evaluation mode.
2. Iterate over the validation data loader and perform forward passes to obtain the predicted start and end logits.
3. Convert the predicted start and end token indices to the corresponding answer text using the tokenizer.
4. Store the predicted answers and the ground truth answers for evaluation.

The `evaluate_predictions` function computes the F1 score between the predicted answers and the ground truth answers. It uses the `compute_f1` function, which normalizes the answers, calculates the precision and recall, and returns the F1 score.

7 Results and Discussion

The trained model is evaluated on the validation dataset, and the evaluation scores, including the F1 score, are printed. The F1 score is a commonly used metric for assessing the performance of question answering systems. It is calculated as follows:

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{1}$$

where precision is the ratio of correctly predicted tokens to the total predicted tokens, and recall is the ratio of correctly predicted tokens to the total ground truth tokens.

The evaluation scores provide insights into the model's performance and its ability to accurately answer questions based on the given context. Further analysis can be conducted by examining specific examples where the model performs well or struggles, and by considering factors such as the complexity of the questions, the length of the contexts, and the quality of the dataset.

The model achieved an F1 score of 61.63

8 Conclusion

In conclusion, this homework assignment involved the implementation of a question answering system using the DistilBERT model and the Spoken-SQuAD dataset. The code follows a structured approach, handling data loading, preprocessing, tokenization, and encoding. The model is trained using PyTorch and the Transformers library, and its performance is evaluated using the F1 score metric.