Language Modeling with GPT-2 using Data Distillation Methods

Nav Sanya Anand
University of Southern California
anandnav@usc.edu

Abstract

This paper explains the steps taken to do language modeling using a GPT2 for WikiText-2. I will explain what my plan is for each of these steps and the difficulties that came with it.

- Train the model for LM and report the baseline perplexity.
- Perform Data set Distillation on wiki-text-2 and extract 120 data points
- Train a new model from scratch on the Data set Distillation data points

Introduction to the model and training

This work focuses on language modeling using a GPT-2 model and distillation techniques on the WikiText-2 dataset. We trained a new model using distillation methods, specifically SimCLR, to extract 120 data points for training. The new model was then evaluated based on perplexity, and the results were compared with the baseline model.

Language modeling plays a crucial role in various natural language processing tasks. The objective of this research is to improve language modeling performance by utilizing distillation techniques. We leverage the WikiText-2 dataset and GPT-2 model for this purpose.

Plan for each step

These explanations provide a detailed overview of my plan for each step involved in training a GPT2 model for language modeling, performing Dataset Distillation using SimCLR/DeepCluster, and training a new model on the distilled data.

0.1 Train the model for LM and report the baseline perplexity.

Load the GPT2 model and tokenizer using the GPT2LMHeadModel and GPT2Tokenizer classes from the Transformers library. The model object is a pre-trained GPT2 language model, and a tokenizer object is a tool for converting text into a format that the model can understand.

Set the maximum sequence length (max_seq_length) to 128, which determines the length of input sequences for training. The default value for max_seq_length is 512, but it can be set to a smaller value if desired. Any input sequence longer than this length will be truncated.

Load the WikiText-2 dataset using the TextDataset class from the Transformers library, specifying the tokenizer and the file path of the dataset. The dataset is automatically tokenized using the provided tokenizer.

Create data loaders for the training and validation sets using the DataLoader class from PyTorch. Data loaders enable us to efficiently load the dataset in batches for training. We specify the batch size

and a collate function that prepares the data for language modeling. The train_loader object loads data in batches of 16 for training, and the val_loader object loads data in batches of 16 for evaluation.

Set up the training loop by defining the optimizer, number of epochs, and moving the model to the appropriate device (GPU if available). The optimizer determines the optimization algorithm used during training, and the number of epochs determines the number of times the model will iterate over the entire dataset. The training loop is responsible for iterating over the training data, performing the forward pass, calculating the loss, and updating the model's parameters. The optimizer object is used to update the model's parameters, and the loss object is a measure of how well the model predicts the next word in a sequence.

Evaluate the model on the validation set by calculating the perplexity. Perplexity is a commonly used evaluation metric for language modeling tasks. It measures how well the model predicts the next word in a sequence. A lower perplexity indicates better performance.

Report the baseline perplexity, which is the perplexity achieved by the model before any further modifications or enhancements. This provides a benchmark for comparing the performance of subsequent steps.

0.2 Perform Dataset Distillation using SimCLR/DeepCluster on WikiText-2 and extract 120 data points

Dataset Distillation is a technique to extract a smaller set of representative data points from a larger dataset using unsupervised representation learning methods like SimCLR or DeepCluster.

In this step, you would typically implement and apply one of these techniques to the WikiText-2 dataset to obtain a distilled dataset.

SimCLR involves training an encoder network on augmented views of the data to learn meaningful representations. DeepCluster also uses clustering methods to identify representative examples.

The distilled data points are obtained by selecting a subset of the augmented data or clustering the representations to identify representative examples.

The specifics of implementing SimCLR or DeepCluster are beyond the scope of this explanation, as they require additional details and potentially custom code. *Note: Was unable to figure out while writing the code*

Once the distillation process is complete, you would extract 120 representative data points from the distilled dataset. The exact method for selecting these points depends on the technique used and the specific requirements of your task.

0.3 Train a new model from scratch on the distilled data points.

Split the distilled data points obtained in previous step into training and validation sets. This ensures that we have separate data for training the new model and evaluating its performance.

Create new data loaders for the training and validation sets using the same batch size and collate function as in Step 1. These data loaders allow us to load the distilled data in batches for training and evaluation.

Instantiate a new GPT2 model using the GPT2LMHeadModel class from the Transformers library. This creates a new model with the same architecture as the original GPT2 model.

Resize the token embeddings of the new model to match the tokenizer's vocabulary size. This ensures that the model's embeddings are compatible with the tokens used in the distilled data.

Move the new model to the appropriate device (GPU if available) and set it to training mode. This ensures that the model utilizes the available hardware resources and enables gradient calculation during training.

Set up the training loop similar to Step 1, using the distilled data loaders and the new model. Iterate over the distilled training data, compute the loss, and update the model's parameters using backpropagation and gradient descent.

Evaluate the new model on the distilled validation set by calculating the perplexity. This measures how well the new model performs on the distilled data, indicating its ability to predict the next word in the sequences.

Report the performance of the new model in terms of perplexity. A lower perplexity indicates better performance in predicting the next word in the distilled dataset.

Problems faced and what is provided

In the example for Step 2, I have used the ImageFolder dataset class from torchvision to load the WikiText-2 dataset. I defined a transformation transform that applies the tokenize_text function to tokenize the text data. I then create a data loader to iterate over the dataset and tokenize the text data using the GPT2 tokenizer. The tokenized text data is stored in the encoded_text_data list.

Next, I apply K-means clustering to the encoded_text_data to extract 120 representative data points. I use the KMeans class from scikit-learn with n_clusters set to 120. The resulting cluster centers represent the distilled data points. Finally, I convert the cluster centers back to text using the GPT2 tokenizer and print the representative data.

Please note that this is a simplified example, and I need to adapt it to my specific implementation and requirements. Additionally, the SimCLR technique usually involves training a separate encoder network on augmented views of the data, which is not included in this example.