CS 391L	Machine Learning WB	Spring 2023

Homework 4

Lecturer: Inderjit Dhillon — Date Due: Apr 17, 2023

Keywords: *Language Modelling, Transformer, PyTorch*

1 Problem Statement

In this homework assignment, you will be training a transformer-based language model on transcripts of class lectures. The main objective of this assignment is to implement the forward pass of the transformer and tweak different hyperparameters of the model to analyze their impact on the training and validation loss.

1.1 Dataset

The data required for this assignment is provided in Canvas → Files → HW4_dataset. The dataset consists of transcripts of our class lectures. The transcripts need to be preprocessed and tokenized to generate input and target sequences (code for this is already given in starter files). The training task is to predict the next token in the sequence given the previous tokens. The dataset is split into training, and validation sets.

1.2 Code

Most of the code has been provided to you, and your task is to implement the forward pass of the transformer. The forward pass takes two inputs, x and y. If y is none, then you need to return just the logits. Otherwise, you need to compute binary cross-entropy loss between the predicted outputs and targets. Here are more detailed steps that can be taken in the forward pass of the transformer:

1. Embed the input sequence x using an embedding layer. The embedding layer maps each token in the input sequence to a vector in a high-dimensional space. The resulting tensor has shape `[batch_size, seq_len, embedding_dim]`.
2. Add position embeddings to the input sequence. Position embeddings are learned vectors that encode the position of each token in the sequence. The position embeddings are added element-wise to the output of the embedding layer to create a new tensor of shape `[batch_size, seq_len, embedding_dim]`.
3. Pass the input sequence through the transformer layers. The transformer consists of several layers, each of which applies a series of operations to the input sequence. Each layer takes as input the output of the previous layer and applies a multi-head attention mechanism followed by a feedforward neural network. The output of the last layer is a tensor of shape `[batch_size, seq_len, embedding_dim]`.
4. Pass the output of the transformer layers through the language model head. The language model head is a linear layer that maps the output of the transformer to a tensor of shape `[batch_size, seq_len, vocab_dim]`, where `vocab_dim` is the size of the vocabulary. This tensor contains the logits for each token in the output sequence.
5. If y is None, return the logits. The logits represent the unnormalized probabilities for each token in the output sequence. The argmax of the logits for each position in the sequence gives the predicted token for that position.

6. If y is not None, compute the cross-entropy loss between the predicted outputs and the target sequence y.

You will also need to evaluate your model on validation data and generate sample model outputs. The code for this has already been provided to you.

1.3 Hyperparameters

In addition to implementing the forward pass of the transformer, you will also need to tweak different hyperparameters of the model, such as the number of heads, the number of layers, the embedding dimension, and the block size. You will then need to report the training and validation loss with different hyperparameters. You can try 3 different values for each hyperparameter (one below the default value and one above) independently and report the results in a table.

1.4 Deliverables

The deliverables for this homework assignment are as follows:

- Code for the forward pass of the transformer.
- Evaluation of the model on validation data and visualization of a generated sample of outputs.
- Report on the impact of different hyperparameters on the training and validation loss.