MUS 470 Optional Lab Assignment

Audio Transformer

Fall 2024

Preamble:

This project assignment is framed as a group project which is open-ended. It could serve as a preliminary set of programming exercises and then evolve into a year-long project. It is centered around Machine Learning but also has significant audio work in the input layer, so could also serve as a joint interdisciplinary project between students in both CSML and CSDA programs.

The project is described as a series of tutorials and steps which culminate in a modification of a neural network using tools developed by the instructor which are implemented in Python. The neural network is a Transformer which can be built with PyTorch and Python. The dataset is the same one that we used for our TensorFlow Audio Speech Recognition assignment: the Speech Commands Dataset. The tutorials begin with a full implementation of a Transformer for NLP and point the way toward converting this to an audio Transformer network.

This project could make use of AI tools to implement various parts. For example, in order to modify the input layer of the vision transformer to accept audio spectrograms, one could consult chatGPT or other coding AI assistants for suggestions.

The last step, the most open-ended part, is the replacement of audio tokenization using cycles instead of spectrograms. It would be good to have a proof of concept for this by doing cycle interpolation for the Speech Commands data set. This is started already but needs to be filled in. The steps are to first break up audio into chunks, then compute f_0 in each chunk using cycles and zero-crossings to enhance crude first guess (say using STFT), then use these cycles with interpolation to reconstruct audio files. The interpolation phase needs to use changing f_0, which requires changing cycle length (or wavelength), and also a choice of spline interpolation, with linear as the default. The resulting audio files can be tested for audio quality as well as basic recognition by simply substituting these interpolated file for the original set, then doing retraining and inference.

Overview:

The main goal of this lab assignment is to implement a transformer neural network model for speech recognition based on the Audio Spectrogram Transformer (AST), and then to extend this model by modifying the input layer. This project can be broken into the following stages:

1. Use the tutorial by Umar Jamil "Coding a Transformer from scratch in PyTorch" to build a langage translation generative transformer model with Python and PyTorch which translates between English and Italian.
2. Use the tutorial by Ross Wightman "Vision Transformer (ViT) in PyTorch" to build a Transformer that classifies images.
3. Modify this ViT model to do speech recognition by following the description in the article called "AST: Audio Spectrogram Transformer".
4. Use the dataset called Speech Commands to train the AST model to perform the same audio recognition functions as the tensorflow model that directs movement of the turtle with recorded voice commands.
5. Modify the input layer in the AST model to accept tokenized audio based on cycles.

The Transformer neural network architecture has become state of the art in ML (Machine Learning) and AI (Artificial Intelligence) in the past seven years. It was first introduced in a now famous paper by Google researchers, called "Attention is all you need", which appeared online in the Arxiv in 2017 (see links below). The Transformer was

first used in NLP (Natural Language Processing) and quickly dominated tasks such as language translation. GPT-1 (Generative Pretrained Transformer) was developed at OpenAI in 2018, and GPT-3 was released as beta to the public in 2020. Its successor GPT-3.5 became ChatGPT, which took the world by storm in fall of 2022.

Due to its success with NLP, the Transformer was quickly adapted to other tasks, such as image recognition. Since image recognition had been dominated by convolutional models, the first efforts combined the attention mechanism in Transformers with convolutional layers. The first model to use only attention layers applied to image recognition, which could be called a pure transformer, was called the Vision Transformer or ViT, introduced in 2021. Around the same time ViT was modified to do speech recognition which was called AST (Audio Spectrogram Transformer) since images based on spectrograms were used to represent the audio samples.

Implementation:

To learn about neural network models, as with most scientific and engineering tasks, it is good to work through an entire development pipeline, or as much as possible, so that one can then tweak the pipeline for other experiments or adaptations. This means that one has to use a relatively small model, or what is sometimes called a 'toy model' which can reasonably fit onto a laptop. For neural networks, this means that we need to fit the training data onto a laptop and be able to train the model on this data in a reasonable amount of time. We saw how this was doable for the tensorflow speech recognition model using the (abbreviated) Speech Commands Dataset.

The most prevalent Library for neural network development environment is now PyTorch, which we will use in this project. There is a Transformer implementation in PyTorch, as well as several modules which form the core of the Transformer, such as Attention. To see how these can be built from other libraries in PyTorch, it is useful to see the NLP tutorial (by Umar Jamil) which also does training and inference. This means that one can get a full introduction to Transformers and have one working locally on a laptop which does language translation. The resulting model is a generative one, with language excerpts as input and output. It is an important example of the main context for Transformers and it is a complete implementation, including code, training, and inference.

The next step is the Vision Transformer, which performs image recognition. This requires modifying the input layer in order to accept pieces of images instead of words. Since the input words in the NLP transformer are converted to vectors (called word embedding) of size 512 (ie. containing 512 floats) the input image should be broken into pieces which should each hold around that many floats. So from a data input viewpoint it is straightforward to convert an image into a sequence of rectangular chunks each of the same size. This process is discussed in the paper and tutorial for ViT. One thing to note which is missing from the ViT tutorial is training and inference of the model.

The AST model is based quite closely on ViT, with the basic idea being to replace images used in ViT with mel spectrograms. This follows the same idea that we saw with the CNN (convolutional neural network) which also starts with input consisting of audio converted to spectrograms.

Some important links:

- Tutorial 1: (Umar Jamil) Language Translation Transformer: language transformer video
  https://www.youtube.com/watch?v=ISNdQcPhsts&t=106s
- Code for Tutorial 1: language transformer code
  https://github.com/hkproj/pytorch-transformer/
- Vision Transformer ViT article: vision transformer article
  https://arxiv.org/pdf/2010.11929
- Paper Walkthrough: Vision Transformer (ViT) vision transformer walkthrough
  https://towardsdatascience.com/paper-walkthrough-vision-transformer-vit-c5dcf76f1a7a
- Tutorial 2: (Ross Wightman) Vision Transformer implementation using PyTorch:
  vision transformer implementation
  https://www.youtube.com/watch?v=ovB0ddFtzzA
- Code for Tutorial 2: vision transformer code
  https://github.com/jankrepl/mildlyoverfitted/tree/master/github_adventures/vision_transformer
- Tutorial 3: PyTorch Vision Transformer PyTorch Paper Replicating
  https://www.learnpytorch.io/08_pytorch_paper_replicating/
- AST (Audio Spectrogram Transformer) article: audio transformer paper
  https://arxiv.org/pdf/2104.01778
- AST Code Base on github: AST code base
  https://github.com/YuanGongND/ast
- PyTorch Tutorial Website : Learn PyTorch
  https://www.learnpytorch.io/00_pytorch_fundamentals/
- Blog about Launching the Speech Commands Dataset: speech commands blog
  https://ai.googleblog.com/2017/08/launching-speech-commands-dataset.html
- Book on Speech and Language Processing by Jurafsky (see chapter 16): speech processing chapter
  https://web.stanford.edu/~jurafsky/slp3/
- Audio Processing (good for recording with python to wav file) Tutorial: audio processing in python
  https://www.youtube.com/watch?v=n2FKsPt83_A&t=960s
- Mel Scale and Mel Spectrogram Explanation: mel spectrogram
  https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53