Project Deliverable 2

Jacky Zhang

March 2022

1 Problem Statement

The initial project proposed was to implement a birdsong classifier using a convolutional neural network.

2 Data Preprocessing

The dataset that I am working with is a subset of the Cornell Birdcall Identification dataset, which itself is a subset of the Xeno-canto dataset. I originally intended to train on the entire dataset, however, the training time would be very long. Furthermore, since I do not have a lot of experience with neural networks, I decided to start on a smaller scale. I trained on the birds with codes from A to B, which contains 4525 samples. For data preprocessing, I loaded the audio file using torchaudio, converted it to mono, cropped a random 5 second portion of it, and transformed it into a mel spectrogram.

3 Machine Learning Model

The model that I initially proposed was a convolutional neural network, which I implemented using PyTorch. Currently, I am finetuning a pretrained EfficientNet-B7 model and using a stochastic gradient descent optimizer. The EfficientNet-B7 model has 813 layers, mostly made up of 54 MBConv Blocks, otherwise known as Inverted Residual Blocks. I set aside 20% of the dataset as the validation set. The model is underfitting as the accuracy on both the training and validation set is relatively low and was increasing each epoch before the notebook crashed. I saved the weights that had the best validation accuracy to the Kaggle output directory, but they keep getting lost when the notebook restarts. So I have to figure out how to persist them.

4 Preliminary Results

The model was able to obtain around a 60% accuracy on the training and validation sets, however, convergence was not reached as the Kaggle notebook kept restarting or shutting down during the long training process. As a proof of concept, I fed a significantly smaller dataset of cat and dog sounds through the pipeline and managed to obtain over 90% on the test set.

5 Next Steps

I think that the general approach is fine, but I am facing some difficulty training on such a large dataset. I plan to further restrict the dataset to a smaller number of birds and train only on audio clips with ratings of 4 or 5. This should hopefully speed up the training time. I might also switch back to MobileNet or EfficientNet-B0, which have fewer parameters and train faster. I could also play around with the optimizer as currently I am using stochastic gradient descent, but apparently Adams might converge quicker. In the future, I could potentially rent out a VM on GCP to train with. Changing some hyperparameters might also help, for example, I could shorten the length of each sample from 5 seconds to 3 seconds to speed up training.