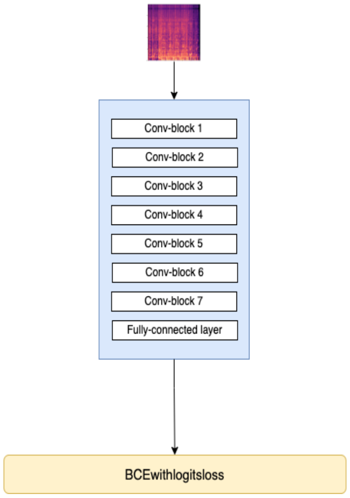
Introduction:

Music tagging is problem of identifying the tags associated with audio file. This is multi-label classification problem where a single audio file can be associated to multiple labels. For this assignment, we have implemented one algorithm(Baseline) and proposing second algorithm which can be used for zero-shot learning.

Dataset:

Dataset consist around 663 audio file. 560 files are used for training and 100 files are used for validation and 3 files are discarded. To speed up training process, all MP3 files are converted to wav and later save into numpy array. First, input audio file is normalized and then we extracted the spectrogram of random 30 second chunk of the audio file with default parameters. The audio files which are smaller than 30 seconds are padded with zeros. Feature extraction is done during training so to get different audio chunk of 30 seconds for different iteration. During preprocessing, data_mapping.pkl has prepared which contain mapping of audio and tags. Train_ids.pkl and val_ids.pkl contain file names used for training and validation.



Fig(1)

Method 1: Baseline

Network:

The figure 1 represents the overall architecture of baseline which consists of 7 conv blocks where each block consists of 2-D Conv layer with filter size 3X3, followed by batchnorm, relu and max pooling. Every layer has a different number of output filters. 128, 256, 256,256,512,512, 256 are the number of filters used in each conv block. After conv block, one fully connected layer of dimension of 67.

Loss : To solve the problem of multilabel classification we have used Binary cross entropy loss. To make training more stable we have utilized BCEwithlogit loss, the loss combines sigmoid and BCE loss.

Training: The model trained on 560 audio files and 100 files is used as validation data while training. Initialize the network with random weights and train it with SGD optimiser minimizing the total loss with learning rate of 0.001. The model is trained for 100 epochs.

Experiments:

Experimented with sigmoid layer and BCE loss separately. The Training was highly unstable and model got overfit. BCEwithlogitsloss has stablize the training and make validation loss stable.

Evaluation:

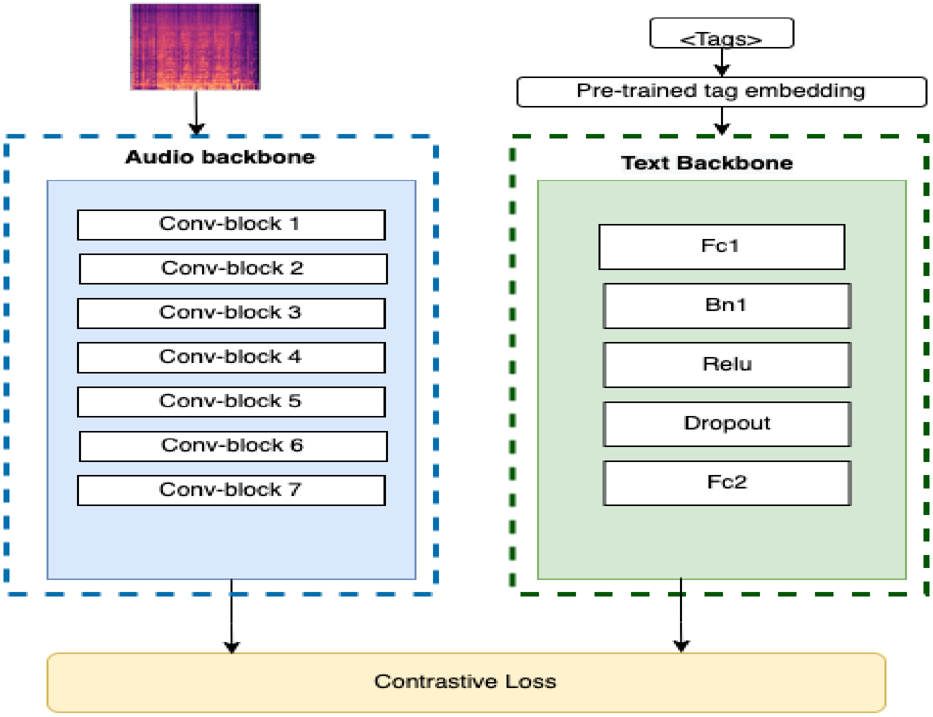
Trained models are evaluated with mean average precision (MAP) over the labels and precision at 10 (P@10).

Further Work:

- Training precision is around 0.63 so there is definitely way to improve the architecture and play around parameters like learning rate and optimizer to make improve training .
- Tag level accuracy/precision can also be calculated and training further can be improved by giving weightage to poorly predicted tags.
- Transfer learning using pretrained weights of tag-based model
- Experiment around selecting different parameters of spectrogram(hop length, framesize)
- Few possible experiment with number of conv block.

Method 2: Multimodal Learning Tag Based Retrieval

Above method work for specific and limited number of tags. In the real world, music tag query can have a huge number of tags so the music information retrieval system needs to be more flexible and not restricted to a limited number of classes. Metric learning is one of the ways to expand it, it uses distance based metrics to measure the similarity. Fig(2) represents the proposed network architecture.



Fig(2)

Input:

Audio Input: Spectrogram is given as input to the audio backbone (similar to baseline).
Text Input: Tags corresponding to audio input are in the form of text. The text tags are converted into embedding using the pre-trained word embedding model which gives a n-dimension vector.

Network:

Audio backbone: Similar to baseline architecture without final fully connected layer.
Text backbone: Text backbone consists of two fully convolutional layers with batchnorm, relu and dropout. The output layer has an embedding of n-dimensional vectors.

Losses: To train the network we can use any contrastive loss by using pair of audio embedding and text embedding. The sampling of negative audio-text pairs will be challenging.

Evaluation:

1. Tags are first converted into embedding using a pretrained model similar to the training. Later converted tags embedding passed to the text backbone of the network to extract the embedding n-dimensional vector. Similarly, using the trained audio backbone, audio embedding is generated of n-dimensional vectors. Any distance metric(cosine similarity) can be use to calculate distance between embeddings.
2. Zero-shot based Evaluation: The idea behind using the similarity basis learning for tag classification tasks is to use trained models on unseen labels.