DEEP LEARNING

MAJOR PROJECT REPORT

Vicky Kumar Muarya (B21BB033), **Jarpala Ashok** (B21BB038), **Biyaram Saste** (B21BB037), Sourabh Singh (B21BB031)

NOTE:- Please find the code through the Colab repository and kaggle notebook links below:

1. https://colab.research.google.com/drive/1T7Sjfy8haiJuWPf8DvMXLNb2ipYSqFHl
2. https://www.kaggle.com/code/vickykumarmaurya/dl-project-final-project

PROBLEM STATEMENT

In the contemporary landscape of music streaming services, the conventional methods of music discovery based solely on artist and song titles are proving insufficient. The inherent variability and subjectivity of music make accurate genre classification a challenging task. Our project seeks to develop a robust deep learning model for the automated classification of music genres from audio clips. The objective is to enhance more accurate genre labels.

MOTIVATION AND SOLUTION STRATEGY

In an era where music streaming reigns supreme, conventional discovery methods fall short of capturing the nuanced essence of each track. By harnessing the power of deep learning, our project aims to revolutionize music classification. Through precise genre identification from audio clips, we pave the way for more personalized and enriching music recommendations. By addressing the inherent variability and subjectivity of music, we unlock new dimensions of discovery for listeners worldwide.

After exploring the literature that is based on or closely related to our problem statement and with the pre-knowledge we have with the deep learning course, we on discussion, concluded a strategy of using Convolutional Neural networks (CNN), Recurrent Neural networks(RNN),

Long-short term memory (LSTM), Autoencoders and Multi layered perceptron(MLP). For a better precise classification that is with a good accuracy, we have aligned a skeleton of the model where we planned of implementing two individual combinations : Autoencoder+CNN and RNN+LSTM that works on Mal_spectrograms (image data) and sequential data independently and further based on probabilistic conditions, implementing a multilayer perceptron for the avoiding false positive classifications which would significantly improve the accuracy.

DATASET

GTZAN Dataset - Music Genre Classification:

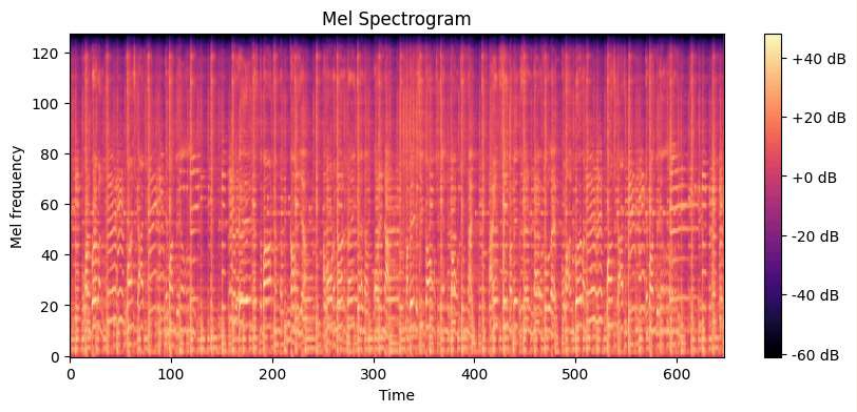
genres original - A collection of 10 genres with 100 audio files each, all having a length of 30 seconds

images original - A visual representation for each audio file. One way to classify data is through neural networks. Because NNs usually take in some sort of image representation, the audio files were converted to Mel Spectrograms to make this possible.

2 CSV files - Containing features of the audio files. One file has for each song (30 seconds long) a mean and variance computed over multiple features that can be extracted from an audio file. The other file has the same structure, but the songs were split before into 3 seconds audio files.

DATA PREPROCESSING AND VISUALIZATION

The dataset was loaded with the help of PyTorch's "Dataset" and "Dataloader" classes with a batch size of 64. The choice of an optimum batch size can be attributed to the meager computation resources provided by Kaggle. This data was then normalized using the "transforms" module of the framework. The train data was split into train and validation and test datasets where we ensured that the data of these sets does not overlap. Some visualization was performed to gain insight into the data.



INNOVATIONS

Our approach for music genre classification:

Based on our literature survey, we found few classical and few hybrid approaches that are dealing with the classification of music genres better to best. Some of those classical approaches are using CNNs and RNNs directly and fine tuning those model architectures for better performance. One such approach that captured our attention was using RNNs+LSTMs which have resulted in acquiring a good performance. Although, our key innovation we believed in is as follows:

1. Using Autoencoder + Convolutional Neural networks:

We have used an autoencoder to extract features from the Mel spectrograms (we obtained after preprocessing the data). Utilizing the latent space representations from the autoencoder as input to a CNN for classification is what we thought would be much more effective. CNNs are well-suited for image-like data such as spectrograms.

Combining autoencoders and Convolutional Neural Networks (CNNs) for music genre classification is like using two powerful tools together. Autoencoders with latent space representation help by simplifying the music data into a smaller, but still useful form called the latent space. This space keeps important details about the music. Then, CNNs come in and use this simplified information from that latent space output to figure out the genre of the music. CNNs are good at spotting patterns and connections in data. By working together, autoencoders and CNNs make it easier to tell different music genres apart. This combination helps to make the classification process more accurate and efficient when dealing with the GTZAN dataset.

2. Using Recurrent Neural Networks + Long short term Memory

Combining Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) networks is a way we believe good to classify music genres. RNNs are good at understanding the order of things, which is important for music because it's a sequence of sounds. LSTMs, a special type of RNN, are even better at this because they remember things for longer. By putting RNNs and LSTMs together, we get the best of both worlds. RNNs help understand the sequence of music, while LSTMs make sure we don't forget important parts. This combination lets us learn about the structure of music and accurately figure out its genre. So, using RNNs with LSTMs is a smart choice for classifying music genres effectively, especially with datasets like GTZAN.

3. Multi layer perceptron

We are proceeding by training both these models independent of each other on image dataset and sequential (audio) dataset. Our key idea is that, upon training these two models individually, we have created a dictionary by concatenating the data from both the models which we used for MLP. This dictionary is to store that merged data in an order as for a particular audio file we will have 3 components as its image representation, sequential representation and its label. Now, upon giving this dictionary data to an MLP, as our dataset has 10 classes, it applies a probability distribution over 10 classes as a process of one-hot encoding, it will assign 0 and 1 or true or false values and assigns all such a data point with its corresponding label to the genre that data point belongs to. This way, the false positives or significantly avoided and enhances the classification accuracy over the merged data of above mentioned two models (autoencode+CNN and RNN-LSTM). This fusion of information may lead to a more robust and accurate classification system, as it combines the strengths of multiple approaches.

Moreover, apart from the classical approaches, we believed that our idea will deal with the music genre classification task more efficiently. Though we do not guarantee that it is the case, we enthusiastically focussed on this idea and in implementing it.

MODEL ARCHITECTURES

CNN

```
Undercomplete(
  (encoder): Sequential(
    (0): Conv2d(4, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (1): ReLU()
    (2): Conv2d(16, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (3): ReLU()
    (4): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (5): ReLU()
    (6): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (7): ReLU()
  )
  (decoder): Sequential(
    (0): ConvTranspose2d(128, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), output_padding=(1, 1))
    (1): ReLU()
    (2): ConvTranspose2d(64, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), output_padding=(1, 1))
    (3): ReLU()
    (4): ConvTranspose2d(32, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), output_padding=(1, 1))
    (5): ReLU()
    (6): ConvTranspose2d(16, 4, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), output_padding=(1, 1))
    (7): ReLU()
  )
)
```

LSTM

```
LSTMModel(
  (lstm1): LSTM(1292, 64, batch_first=True)
  (lstm2): LSTM(64, 64, batch_first=True)
  (fc): Linear(in_features=64, out_features=64, bias=True)
  (dropout): Dropout(p=0.3, inplace=False)
  (output_layer): Linear(in_features=64, out_features=10, bias=True)
)
```

TRAINING & RESULTS

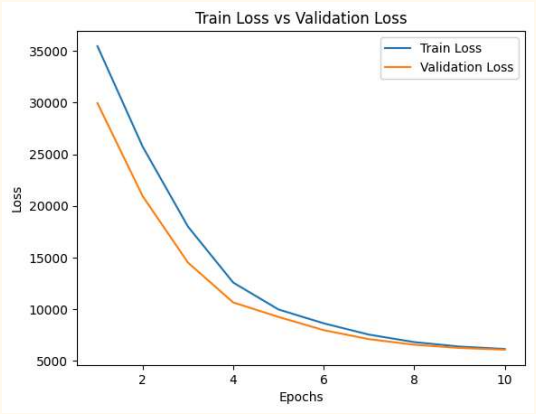
Num. of epochs for CNN = 10

Num. of epochs for Auotoencoder = 10

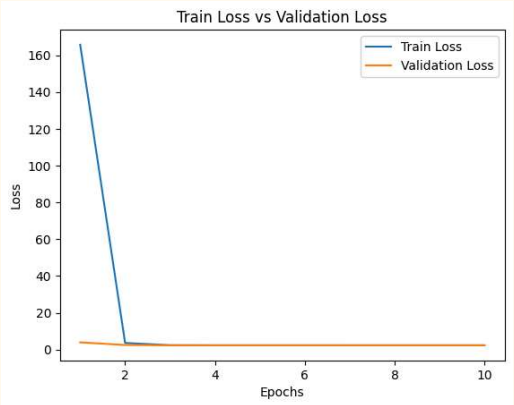
Num. of epochs for LSTM = 60

Num. of epochs for MLP = 10

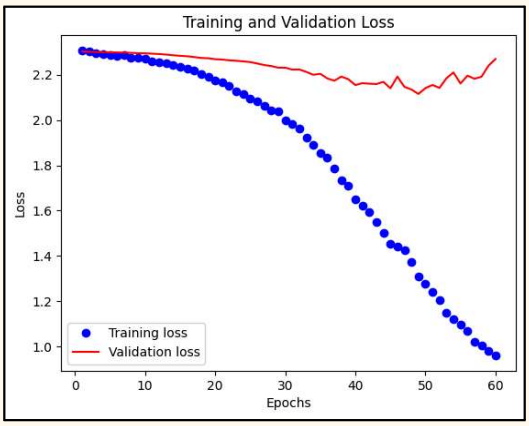
Batch size = 64 (for all)



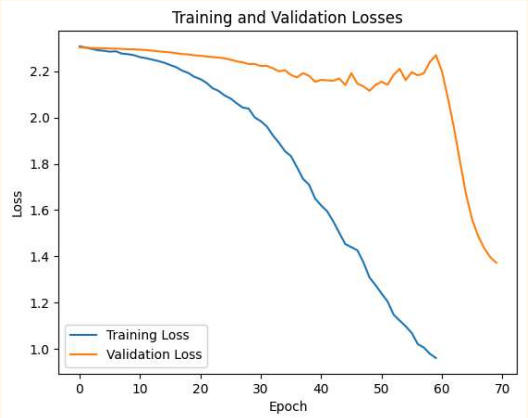
For autoencoder



for CNN



For LSTM



For MLP

Model Type	Test loss	Test accuracy
Autoencoder+CNN	2.3082618713378906	7.5%
(RNN+LSTM) **LSTM**	2.0855790367126463	0.244
MLP	2.2260	24.50%

ANALYSIS OF THE SOLUTION AND WEAKNESSES

As explained above, the model and architecture we have planned have a good capability and potential in dealing with such a music genre classification based on those audio clips where we have original (audio) sequential data and image data. As we have implemented the combinations like **Autoencoder+CNN and RNN+LSTM** and further an **MLP.** The latent space feature capture ability of autoencoders in combination of CNNs on one hand and sequential patterns and long term memory on the other when dealt with such a 10 class probability distribution applied Multi layered perceptron, should be one of the best for such c=genre classification problems.

Although we believed that we would be acquiring the best accuray, the results we have obtained are not very convincing. This is all due to the fact that we have not been that good in dealing with such a large data and fine tuning the model appropriately. Though our idea is good, we have somehow very less succeeded in solving the problem. Our weaknesses we have figured out are, we have not performed an effective preprocessing and normalized the data. Rather than going deeper into refining the model, it was more like we have given a trial. Our major weaknesses are, we have not experimented with hyperparameters. We have not gone much deeper in regularization steps to prevent any overfitting of the data. We have trained for a very small number of epochs. As the model was complex, after some developments, the model's performance went out of our knowledge and we were inxcapable of fine tuning it. If all these things were addressed properly, the model would have perfomed very best and we would have acquired relatively best accuracy.

CONCLUSIONS

The above results show that the models we have used will be so much potential for music genre classification upon fine tuning them and taking into account all the needed steps for denoising and normalizing the data. Although the results are not very much convincing in terms of the losses and accuracy, it is all due to the only reason we are not that best at fine tuning and experimenting with the hyper parameters. One sure thing we can say is the architecture we have constructed and the steps we followed while in training and evaluating will work very efficiently upon further developing them at the best possible level.

CONTRIBUTIONS

Vicky Kumar Maurya: Explored the dataset. Understood different aspects of the audio file (amplitude time graph, pitch vs time graph , analyzing specific frequency for each genre in audio data)and has worked on preprocessing the data that could be input to the autoencoder. Explored on designing new audio files using graph data structure and then using for classification, explored over faster RCNN model to capture region specific features, explored on the combination of LSTM and CNN. Worked on constructing Autoencoder+CNN model and. Worked on refining LSTM model and also worked on implementing Multi Layer perceptron.

Jarpala Ashok: Explored the dataset and explored the literature on the variety of architectures that are already available on this dataset and captured key strategies. Worked on designing the skeleton models for the task.Over the observations, brought out the concluded model architecture of using Autoencoder+CNN and RNN+LSTM and merging their data to implement a MLP to leverage the classification. Worked on constructing baseline architecture. Assisted other team members in refining the individual combinations of the models and in final integration of those individual combinations. Worked between the rest 3 of the team members, assisted them in understanding the task and needs and brought up all the applicable strategies to overcome them. Also, Made all the reports.

Biyaram Saste: Explored the literature on the GTZAN dataset. Right from the beginning, focussed on understanding the task deeper and all its needs. Suggested few ideas and architectures for solving the problem. Assisted Vikcy Kumar in preprocessing the data. Further, once the final model architecture was confirmed, he worked on acquiring the needed references and useful pytorch code snippets. Later, till the end, he worked on constructing RNN+LSTM model and faced the challenges of refining the model for a better performance. Sorted out the

issues in doing so and brought out the high weighted LSTM architecture from the combination of RNN+LSTM which was relatively more useful in overall classification.

Sourabh Singh: Helped in understanding the dynamics of GTZAN dataset in the context of genre classification, explored the problem statement with all different perspectives and suggested the strategies to conclude a suitable approach. After the final architecture was concluded, was engaged in working on RNN+LSTM model from a baseline to refined model. Focussing more on improving the performance and accuracy of RNN+LSTM model, have experimented with all the possible modifications of hyperparameters and fine tuning the model, was able to obtain best accuracy for this component of the entire work.

REFERENCES

1. https://arxiv.org/abs/2309.04861
2. https://arxiv.org/abs/2401.04737
3. https://arxiv.org/abs/2307.10773
4. https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification/code
5. https://medium.com/@premtibadiya/music-genre-classification-using-rnn-lstm-1c212ba21e0
6. https://www.kaggle.com/code/vickykumarmaurya/dl-project-final-project

THANK YOU