Using Artificial Neural Networks for Instrument Classification of Audio Signals

Saksham Goel

Abstract— **This project examines the applications of using different types of Deep Learning Architectures like Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) for instrument recognition. The project is divided into two section. First section of the project explores the time dependency of the audio signals hence uses architectures like Simple Vanilla RNN, Gated Recurrent Unit (GRU) [8], Long Short Term Memory (LSTM) [9] [12] and Bidirectional Long Short Term Memory (BLSTM) [5]. Second section on the contrary explores the local dependency of the audio signals hence uses techinuqes like CNN's on top of Multiresolution Recurrence Plots (MRP) [11] and RNN's on top of CNN's [10].**

I. INTRODUCTION

Music is one of the most popular source of entertainment for us humans and boasts one of the biggest entertainment industries. A lot of research is being done currently for novel ways of queriying music and sound signals. This project aims to set up a proof of concept for using different deep learning architectures for the task of instrument classification of audio signals so that it can be used to label sound recordings. This project focuses pn training different deep learning architectures as mentioned in the abstract and performing a comparison of the accuracies. The problem statement is as follows: *Given an audio signal with a predoiminated sound of a given instrument, the model will classify some t second sliding windows of that given audio excerpt into one category from any of the following categorie mentioned in Table I.*

Audio recognition is very widely researched filed in todays time. A lot of research is being conducted in the field of speech recognition to make the existing phone assitants like Siri, Google Home better and much more accurate. Audio data is very special because of its inherent structure which dictates dependency between sound signals at two different signals. To think about it, audio signals are temporal data much like time series data such that they have long range dependency between signals. Because of this property it is hard to use traditional machine learning algorithms because they make assumptions like the features of the data are independent and identically distributed. One of the most popular deep learning architecture used in the field of dependent values and modeling time series like data is Recurrent Neural Networks. They have been proven effective in sequence modelling [7] and also in handling temporal data of varying lengths [7]. RNN stores the information about time series in a hidden unit and use these to actually compute the output values which is why they are excellent at modelling Time-Series data. For this project we are using the following 4 types of RNN's:

- Simple Vanilla RNN (Figure: 1)
- Long Short Term Memory (Figure: 2)
- Gated Recurrent Unit (Figure: 3)
- Bidirectional Long Short Term Memory

Other than modelling the audio signals using architectures focusing on time series data, I will also try to use Convolutional Neural Net-

TABLE I
NUMBER OF TRAINING SAMPLES BY INSTRUMENT CLASS

Instrument Category	Number of Audio Files
Electric Guitar	760
Piano	721
Saxophone	626
Violin	580
Human Singing Voice	778

work for modelling neighboring value dependencies. This project with use the following 3 types of CNN's:

- Simple Vanilla CNN (Figure: 4)
- CNNs on MRP
- CNN + RNN

II. RESOURCES

A. Dataset

The techinuqes we are using for this project are supervised learning algorithms hence we require data with annotations about the class they belong to. For this project we are using the Instrument Recognition in Music Audio Signals (IRMAS) Dataset [1]. The IRMAS dataset currently has divided the dataset into Training and Test sets. Training dataset contains 6705 audio excerpts in 16 bit stereo wav format audio files sampled at 44.1 kHz. All of these files are 3 second excerpts from more than 2000 distinct recordings. The number of audio files for a subset of all classes that I will be training on is given in Table I. For the test set there are 2874 excerpts in 16 bit stereo wav format sampled at 44.1kHz.

B. Implementation

For this project the implementation will all be done in Python. Python is an open source interpreted high-level programming language for general-purpose programming. Currently python supports a lot of powerful libraries like Keras, Tensorflow and Pytorch which provide enough resources to set up different deep learning architectures that have been mentioned before. Along with these libraries there also exist some other libaries like numpy, scipy, sklearn and matplotloib which will help in data loading, preprocessing, analysis and visualizations. All of these libaries have lots of great documentation available online and also a lot of blog articles on websites like Medium and TowardsDataScience containing helpful resources related to the full spectrum. On my initial analysis, there is not much code available online directly corresponding to my problem. I will be coding the chunk of the whole project but most of it will be based off online discussions and blogs where people try to use these architectures for different yet similar problems.

III. EXPERIMENT

For this project I will be setting up different experiments in different notebooks that will walkthrough what type of experiment is being conducted along with the final assessment of that experiment. Most of the experiments would correspond to using different architectures with the same input dataset and similar values of the hyperparameters. For this part of the experiment, I would first compare all the 4 types of RNN's for some binary classification and then compare them on the multiclass comparison containing all the classes that I am considering. Similarly I would also experiment with the two types of CNN models discussed earlier with binary and multiclass classification. These experiments would lead to analysis based on using different architectures for the same dataset over the type of classification being performed. The other type of experiments that I will be performing includes experimenting with the data preprocessing which includes different types of normalization (mean normalization, variance normalization, uniform normalization) on the data and also experimenting with windowing techniques which would deal with the amount of data being loaded. I am

especially considered with windowing because it would highly affect the amount of data on which I will be training and also the values of the audio signal I will be using. I am planning to experiment with different sliding window sizes to get a better understanding of the effect of the initial dataset size on the classification accuracy of the model. Other than these I will also do some experimentation to understand the effect of hyperparameters of the architectures on their performance. These experiments however would be each architecture specific. For hyperparameters I will be experimenting with different activation functions, dropout percentages, number and size of layers.

IV. TIMELINE

This project has a lot of different parts belonging to it. Hence for the timeline I have divided it into many parts, such that each deal with a particular part of the whole project and the pipeline that needs to be created. The different parts of the pipeline is as follows:

1) Dataset exploration, exploring different online available datasets.
2) Data loader, developing scripts for loading data from the sample audio files. This includes loading data in the format usable by RNN and loading data usable by CNN.
3) Data preprocessing, developing scripts to preprocess the data. This includes normalizing, sliding windows and also constructing MRP.
4) Model training, developing Ipython notebooks that will load data from the given training data and fit different models with a particular set of hyperparameters on the training data.
5) Visualization, developing Ipython notebooks to visualize the convergence, loss and accuracy plots.
6) Analyzing results, developing Ipython notebooks that will run inference on test data and calculate different accuracy metrics like precision, recall, overall classification accuracy, F-1 score.
7) Hyperparameter Tuning, doing some experimentation with different hyperparameters for the model architectures and retraining and redoing the analysis.
8) Conclusion, discussing the different results and trying to understand why id we get such results.

V. ANALYSIS

For this project I will be analyzing the classification accuraccy of the models on a seperate hold out test set after they have been trained. For the evaluation purposes I will be using three different metrics which are classification accuracy, precision and recall. For terms of analysis, I will draw some confusion matrices for comparsion of the overall accuraccies, precision and recall while also use the confusionmatrix to create a heatmap for easy visualization. I will also consider the amount of iterations it took for different models to converge and overall complexity of the models. Considering all these facts together I will try to analyze why each model performs the way it performs and try to select a model which seems to maximize the accuraccy while minimizing the complexity and convergence time. Because I will be performing a lot of experiments and each experiments tries to change one particular variable like model architecture or hyperparameters or data preprocessing, it will help me create a lot of different tables where I will discuss the affect of each variable as well which will help in learning about the effect of each variable when thinking about them in domain of Audio Classification.

VI. APPENDIX: A

Images for different architectures that have been referred in this writeup.

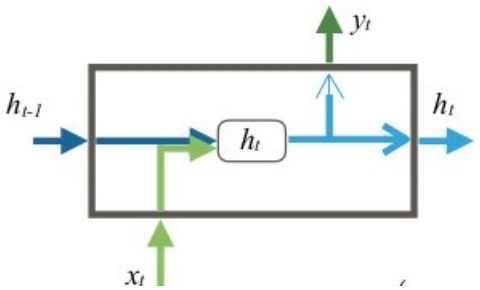

Fig. 1. RNN cell

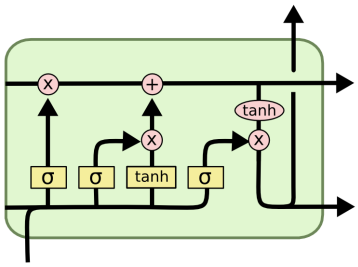

Fig. 2. LSTM cell

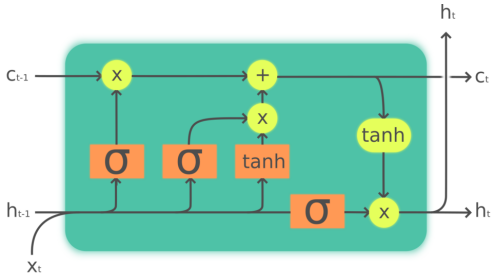

Fig. 3. GRU cell

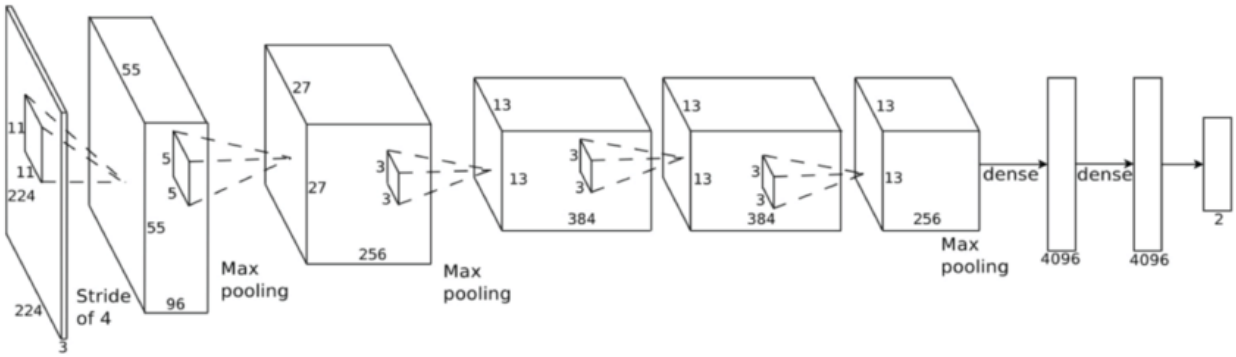

Fig. 4. CNN architecture (Alexnet)

REFERENCES

[1] Bosch, Juan J., et al. "A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals." ISMIR. 2012.

[2] Sak, Haim, Andrew Senior, and Franoise Beaufays. "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." Fifteenth annual conference of the international speech communication association. 2014.

[3] Sak, Haim, Andrew Senior, and Franoise Beaufays. "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition." arXiv preprint arXiv:1402.1128 (2014).

[4] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[5] Graves, Alex, and Navdeep Jaitly. "Towards end-to-end speech recognition with recurrent neural networks." International Conference on Machine Learning. 2014.

[6] Visin, Francesco, et al. "Renet: A recurrent neural network based alternative to convolutional networks." arXiv preprint arXiv:1505.00393 (2015).

[7] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).

[8] Cho, Kyunghyun, et al. "On the properties of neural machine translation: Encoder-decoder approaches." arXiv preprint arXiv:1409.1259 (2014).

[9] Graves, Alex. "Generating sequences with recurrent neural networks." arXiv preprint arXiv:1308.0850 (2013).

[10] Aggarwal, Karan, et al. "A Structured Learning Approach with Neural Conditional Random Fields for Sleep Staging."

[11] Park, Taejin, and Taejin Lee. "Musical instrument sound classification with deep convolutional neural network using feature fusion approach." arXiv preprint arXiv:1512.07370 (2015).

[12] Tang, Chun Pui, et al. "Music Genre classification using a hierarchical Long Short Term Memory (LSTM) model." (2018).