Using Deep Learning Architectures to Classify Audio Excerpts

Saksham Goel
goelx029@umn.edu

December 14, 2018

1 Problem description

Our problem is the classification/identification of an instrument based on a sound snippet in the form of a stereo sound. For solving this problem we will be using the Instrument Recognition in Music Audio Signals (IRMAS) Dataset [2]. The problem statement in a more formal sense is as follows: Given an audio excerpt which contains some musical audio with a predoiminated sound of a given instrument, need to classify 3 second windows of that given audio excerpt into one category from any of the following category: "Cello", "Clarinet", "Flute", "Acoustic Guitar", "Electric Guitar", "Organ", "Piano", "Saxophone", "Trumpet", "Violin", and "Human Singing Voice". For this problem, we hope to solve the problem using different machine learning algorithms. The focus for this project would be to use different Deep Learning Architectures used in Image Classification, Sequence Modelling and Speech Recognition to classify the audio files. The primary architectures that are being considered include Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN).

2 Dataset

As mentioned earlier we will be using the IRMAS Dataset for finding a solution to our problem. The IRMAS dataset currently has divided the dataset into Training and Test sets. Training Dataset contains 6705 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 each given class is given in Table 1.:

For the Test Set there are 2874 excerpts in 16 bit stereo wav format sampled at 44.1kHz. Considering that these files are not necessarily 3 seconds long, while all of the training files are just 3 seconds long, we are using a Sliding Window approach to actually predict the class for each test file. The sliding window would be 3 second window slided over 1 second interval. Because we are using deep learning architectures to construct our classification model, we have to predefine the size of the input vector/matrix and considering that the model will be trained over the Training Set containing 3 second audio files, it will only allow inputs of that length, until we use a zero padding framework which will not be a included in this project. Because of this inherent dependcy that will only allow input lengths to be of 3 seconds, we will use the already mentioned sliding window technique to do inference.

Table 1: Number of Training Samples by Instrument Class

Audio File Class	Number of Audio Files
Cello	388
Clarinet	505
Flute	451
Acoustic Guitar	637
Electric Guitar	760
Organ	682
Piano	721
Saxophone	626
Trumpet	577
Violin	580
Human Singing Voice	778

3 Related work

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. Considering the recent popularity of Deep Learning Architectures in the field of Computer Vision and also their rise in popularity because of their promising results, and their inherent claim that they are based on Human Brain (Neurons), I think it is exciting to use these architectures for classification of the audio sinals.

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 [4] and also in handling temporal data of varying lengths [4]. RNNs are composed of small unit called RNN Cells, the main idea behind Recurrent Neural Networks is to store a hidden state (a feature vector) that represents the current state of the RNN cell. This hidden state is computed using the input values and the hidden state value from the previous cell. Hence, whenever we initialize a RNN it is important to initialize the hiddent state for the first cell (h_0) and the matrices used for the computation. This kind of architecture resembles a dependency of the current hidden state value on the previous hidden state values and the current input value. In mathematical terms we can see that the equation looks somthing like:

$$h_t = f(x_t, h_{t-1})$$

Here h_t represents the hidden state of the current cell or t^{th} cell, x_t represents the t^{th} input and h_{t-1} represents the $t-1^{th}$ hidden cell state. This kind of relationship modelling is what makes RNN's very apt for modeling these intricate dependency between input data. Apart from traditional Vanilla RNNs, there are two main types of Recurrent Neural Networks that have been studied at extent. One of them is called Gated Recursive Neural Network which features a block named Gated

Recurrent Unit (GRU) which is extensivley used in fields of machine translation [3]. The other architecture is called Long Short-Term Memory Recurrent Neural Network which features a block named LSTM unit [5]. The advantage of using LSTM architecture is that LSTM is very good at solving the problem of vanishing gradients when the data has very long temporal dependencies. With the inclusion of different gates (inherent computations to calculate the hidden state) like memory and forget gates which affect the computation of the hidden state, LSTM essentially can use the values to include the memory from previous timesteps and influence the output accordingly, hence effectively capturing long distance temporal dependencies. On the other hand GRU is very much similar to LSTM using different gates, but has a different architecture for the GRU unit. GRU combines some of the gates in LSTM cell (like forget and input gate) to essentially reduce the number of learnable parameters. Due to this, both of them differ in the way they let the output affect the memory inclusion, and also how the output value is calculated [4], while trying to solve the same problem of long term temporal dependencies. The advantage of GRU over LSTM is that it used smaller set of learnable parameters and hence is easier to train and expected to reach a state of convergence much faster. On the other hand advantages of LSTM is that they can learn more abstract long term dependencies because of their intricate structure. Considering that the audio data has a high number of features (a lot of sound signals in the 3 second window \1.2 Million sound signals per 3 second window) it will be good to use GRU so that we can essentially train the model easily and reach convegence early, however LSTM may proove to be more accurate.

Other than these simple RNN architectures, RNN has been used in Acoustic Modelling by Google. The architectures used by Google for accoustic modelling [8] for various research tasks are RNN architectures like Long Short-Term Memory (LSTM) RNN, Deep Long Short-Term Memory (DLSTM) RNN, Long Short-Term Memory Projected (LSTMP) RNN and Deep Long Short-Term Memory Projected (DLSTMP) RNN to combat various problems like Vanishing Gradients, Long Range Dependency between temporal sequences and also to combat the huge number of learnable parameters. The DLSTM essentially repeats the number of LSTM units so that there can different learnable parameter matrices. This helps to learn different reoresentations from the same audio data and hence could be useful and worth a try. The latter two types of RNN mentioned are specifically designed to preserve as much information while essentially decreasing the number of parameters by adding a linear projection layer [8] after each LSTM unit. Because of this reduction in number of parameters, it promises less computation and faster convergence while having similar performance.

Other than that the architecture some researchers used for speech recognition [6] is a Bidirectional Long Short-Term Memory (BLSTM) RNN. A BLSTM, differs from the previous known architectures in the form that it uses both the previous time stamps hidden state and the future timestamps hidden state along with the input value to calculate the current hidden state. In essence as compared to earlier models which only depend on the previous timestamps (just information from past) while BLSTM takes into account the information from both the past and future values. In this manner BLSTM are able to better understand the context and eliminate ambiguity. We are trying to train a BLSTM just to check whether it performs better then other models, and if it does then why might it be doing that, so that we can better understand the music audio data in light of classification of instruments.

Other than RNN the other deep learning architecture that I am considering is Convolutional Neural Networks (CNN). CNN's have primarily been used extensively in the field of Computer Vision for tasks like Image Classification [7] and Object Detection. Their popularity in the field

of computer vision arises from the fact that they are very good at extracting/learning different high dimensional features from the input images. Convolutional Neural Networks use as the name suggests, a method called Convolution. Convolution refers to using feature maps/filters that are convolved (slided) on top of the input feature matrices to result into output feature matrices. This kind of convolution property helps them learn various high dimensional features which is the strength of the CNN. This property makes CNN's very different from other traditional Machine Learning Algorithms, where the features have to be already specified. Considering the CNN's learn these high dimensional feature maps from the given input we aim to use CNNs to extract these features and then build a neural network model on top of these features to classify the audio dataset. Although primarily CNN's have not been used much in the field like speech recognition and sequence modelling, CNN's can still proove to be very powerful tools for feature extraction. We aim to use these extracted features from CNN from the given input image and try to either classify based on that. Also, another way is to use this extracted features as the initial hidden state for a Vanilla RNN model and try to use signal values from the audio with this hidden state initialization in the RNN model to further classify the given audio signal to an instrument class. The motivation for this method comes from [1], where a RNN was used on top of the output of a CNN to model for both local and temporal dependcy of the input sleep staging data.

Another architecture we are going to consider is ReNet which is a Recurrent Neural Network Based Alternative to Convolutional Neural Network [9]. We are considering this architecture because of its novel approach that combines the feature learning of Convolutional Neural Network with a temporal sequencing technique of Recurrent Neural Networks. We think that this architecture would be able to extract various features from the audio dataset while also preserve the temporal information hence effectively combining the strengths of two and may provide better classification results.

To conclude, there has already been a lot of work using different Deep Learning Architectures in different problem domains, and thus we aim to try to apply these techniques and compare the result and try to gain a better understanding about which architecture seems to be the best fit and understand why that is the case.

References

[1] K. Aggarwal, S. Khadanga, S. R. Joty, L. Kazaglis, and J. Srivastava. A structured learning approach with neural conditional random fields for sleep staging.

[2] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera. A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In *ISMIR*, pages 559–564, 2012.

[3] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[5] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[6] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, pages 1764–1772, 2014.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] H. Sak, A. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014.

[9] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio. Renet: A recurrent neural network based alternative to convolutional networks. *arXiv preprint arXiv:1505.00393*, 2015.