INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

Transcription of Percussion Instruments

by

Pranav Sankhe

Supervisor: Prof. Preeti Rao

July 2019

0.1 Abstract

The tabla is a percussion instrument of the Indian subcontinent, consisting of a pair of drums, used in traditional, classical, popular and folk music. The audio of tabla is composed of many modalities. We wish to develop an end to end transcription system for tabla audio and to reconstruct the audio from the obtained transcription. We also intend to study the presence of (acoustic-prosodic) correspondences between the recitation and playing of tabla compositions like intensity and F0 variations, at the bol/stroke level (sort of like word-level in speech) and identify those that are well-correlated. In particular for music education, it would be useful to provide a rating and to identify the nature of the error in the learning process.

Recently data driven signal processing approaches have been gaining momentum to solve problems like the one elaborated above. The use of machine learning to solve complex non linear problems have been dominating the speech and music research which is evident in the recently published papers. In particular, we have explored the use of NMF (Non-Negative Matrix Factorization) to solve the problem of drum transcription for drum-only-recordings. While this approach works significantly well, having seen the developments in DNN models for Piano transcription, we are motivated to employ a similar architecture for tabla transcription.

There are few challenges in implementing a neural network architecture for tabla transcription which we will elaborate over in later sections. We intend to find a way around these challenges and come up with a suitable and robust transcription and evaluation system.

Contents

Chapter 1

Challenges

Here we enumerate over the challenges we face in developing a dual objective transcription system for percussion instruments.

- Interference of multiple instruments
- Playing Techniques
- Recording Conditions and Post Production
- Insufficient real world datasets

1.1 Interference of multiple instruments

The superposition of various instruments makes the recognition of a specific instrument difficult due to the overlaps in both spectral and temporal domain.
Percussive Instruments:
A basic drum kit includes drums of different sizes and well- distinguishable timbral characteristics. In a more advanced setup for studio recordings, similar drums with subtle variations in timbre often appear, resulting in sounds that are harder to differentiate. In the case of tabla, we have two indvidual tablas which the player uses to compose music. Seperating and identifying the audio from these two tablas is a research problem in its own right. This problem of interference is more severe when these sounds occur simultaneously.
Melodic Instruments:
The wide range of sounds produced from a drum kit or a tabla can potentially coincide

with sound components of many melodic instruments. Pre-processing steps for suppression of melodic instruments have been proposed but haven't been able to achieve substantial improvement.

1.2 Playing Techniques

Playing techniques is an important aspect of expressive musical performance. For drum instruments, these techniques include basic rudiments as well as timbral variations.

Approaches to model playing techniques:

- A study on the automatic identification of timbral variations of the snare drum sounds induced by different excitations has been done where a classification task is formulated to differentiate sounds from different striking locations (center, halfway, edge, etc.) with different excitations (strike, rim shot, and brush)
- The discrepancy between more expressive gestures on a larger dataset with combinations of different drums, stick heights, stroke intensities, strike positions, and articulations has been explored
- Different playing techniques for cymbal sounds have been investigated too. The Differentiation is based on either the position where the cymbal is struck (bell, body, edge), how a hi-hat is played (closed, open, chick), or other special effects such as choking a cymbal with the playing hand.

All of these studies showed promising results in classifying the isolated sounds, however, when the classifier is applied to the real-world recordings, the performance dropped drastically

1.3 Recording Conditions and Post Production

In practice, it is likely that we have to deal with convolutive, time-variant, and non-linear mixtures instead of linear superpositions of single drum sounds. The acoustic conditions of the recording room and the microphone setup lead to reverberation effects that might be substantial The recording engineer will likely apply equalization and filtering to the microphone signal Mostly, the resulting signal alterations can be modeled as convolution with one or more impulse responses. Non-linear effects such as dynamic compression and distortion might be applied to the drum recordings.

Consequences:

- Any methods involving machine-learning might deteriorate if the training data does not match the target data.
- Any methods involving decomposition based on a linear mixture model might be affected when the observed drum mixtures violate these basic assumptions.

A possible strategy to counter these challenges might be data augmentation. In our case the amount of training data could be greatly enhanced by applying diverse combinations of audio processing algorithms including reverberation, distortion and dynamics processing.

1.4 Insufficient real world datasets

Size: The most common issue of all the existing drum transcription datasets is the insufficient amount of data. Since these datasets are created under very different conditions, they can- not be easily integrated into one large entity. Given that tabla is an Indian classical instrument, the dataset of tabla recordings is limited in size.

Complexity: The existing datasets have the tendency of over-simplifying the ADT problem. Only the drum sequences with basic patterns are presented in the dataset.

Diversity: Most of these datasets do not cover a wide range of music genre and playing style. The limitation in terms of diversity can hinder the system's capability of analyzing a wider range of music pieces. Particularly, the lack of any singing voice in the corpora ENST-Drums and IDMT-SMT-Drums indicates their insufficiency. Tests on tracks containing singing voice revealed that this poses a big problem, especially for RNN-based ADT methods.

Homogeneity: Since each dataset is most likely to be generated under fixed conditions the audio files within the same dataset tend to have high similarities.

Chapter 2

Design patterns a transcription system

Basic design patterns of a transcription system are enumerated as follows:

- **Feature Representation**
- **Event Segmentation**
- **Activation Function**
- **Feature Transformation**
- **Event Classification**
- **Language Model**

Feature Representation

Audio signals can be represented into feature representations that are better suited for certain processing tasks. A natural choice is Short Time Fourier Transform(STFT). These representations are beneficial for untangling and emphasizing the important information hidden in the audio signal. These includes Band-pass filters with predefined center frequencies and bandwidths, Harmonic-Percussive Source Separation, etc.

Event Segmentation

Event Segmentation involves detecting the temporal location of musical events in a continuous audio stream. We compute suitable novelty functions and employ various Peak picking strategies to extract local extrema. Recently, learned features using Machine Learing techniques have shown to yield superior performance as compared to handcrafted ones for event segmentation as often the handcrafted features are approximates.

Activation Function

Map feature representations into activation functions(AF) which indicated the activity level of drum instruments. Techniques which fall under the roof of activation function are NMF, Probabilistic Latent Component Analysis, Deep Neural Networks, etc. The defining factor of this approach is the concept AF, which generates the activity of a specific instrument over time. With the activation functions for every drum instrument, the even segmentation step can be as simple as finding local maxima of those activation functions by means of suitable peak-picking algorithms. Two families of algorithms for deriving activation functions:

- **Matrix Factorization Algorithms**
- **Deep Neural Networks**

Matrix Factorization Algorithms

This approach uses magnitude spectrograms and applies matrix factorization algorithms in order to decompose the spectrogram into basis functions and their corresponding activation functions. Early systems used methods such as Independent Subspace Analysis, Prior Subspace Analysis and Non-Negative Independent Component Analysis. The basic assumption of these algorithms is that the target signal is a superposition of multiple, statistically independent sources. This assumption is problematic since the activations of the different drum instruments are usually rhythmically related. When the signal contains both drums and melodic instruments, this assumption may be more severely violated. Recently, more and more systems opted for NMF, which has less strict statistical assumptions about the sources. In NMF, the only constraint is the non-negativity of the sources, which is naturally given in magnitude spectograms. NMF-based ADT systems include basic NMF as well as related concepts such as Non-negative Vector Decomposition, Non-Negative Matrix Deconvolution, Semi-Adaptive NMF, Partially-Fixed NMF, and Probabilistic Latent Component Analysis (PLCA). Most of these factorization-based methods require a set of predefined basis functions as prior knowledge; when this predefined set does not match well with the components in the target signal, the resulting performance may decrease significantly.

Deep Neural Networks

DNNs are a machine learning architecture that allow to learn non-linear mappings of arbitrary inputs to target outputs based on training data. They are usually constructed as a cascade of layers consisting of learnable, linear weights and simple non-linear functions. The learning of the weight parameters is performed by variants of gradient descent. RNNs can in principle also perform sequence modeling, similar to the more classic methods such as HMM. However, the lack of large amounts of training data and the applied training methods, prohibit RNN to perform well. Some of the factorization-based approaches can also be used to reconstruct the magnitude spectrogram of drum sources and serve as source separators. This type of approach takes care of simultaneous events without the need of introducing combined classes during training. However, when the multiple sources overlap in the spectral domain, cross-talk between activation functions will appear and degrade the performance. Furthermore, the use of magnitude spectrograms neglects the phase, which could potentially strip away critical information.

Feature Transformation

This design pattern provides a transformation of the feature representation to a more compact form. This goal can be achieved by different techniques such as feature selection, Principal Component Analysis (PCA), or Linear Discriminant Analysis (LDA).

Event Classification

This processing step aims at associating the instrument type (e.g., KD, SD, or HH) with the corresponding musical event. In the majority of papers, this is achieved through machine learning methods that can learn to discriminate the target drum instruments based on training examples. Inexpensive alternatives include clustering and cross-correlation.

Language Model

This pattern takes the sequential relationship between musical events into account. Usually this is achieved using a probabilistic model capable of learning the musical grammar and inferring the structure of musical events. LMs are based on classical methods such as Hidden Markov Models (HMM) or more recent methods such as RNNs.

Chapter 3

NMF for Drum Transcription

Let $X \epsilon R(K \times T)$ be the magnitude spectrogram where K = frequency bins and T is the time length. We intend to decompose the mixture spectrogram X into spectral basis functions $B(:, r)$ and corresponding time-varying gains $G(r, :)$. Intuitively, speaking, the templates comprise the spectral content of the mixture's constituent components, while the activations describe when and with which intensity they occur.NMF typically starts with a suitable initialization of matrices B and G. Subsequently, these matrices are iteratively updated to approximate X with respect to a cost function L. The detection of candidate onset events is typically approached by picking the peaks in the activation function G(r,:)

Dataset

The dataset used was IDMT-SMT Drums. This dataset consists of 608 WAV files (44.1 kHz, Mono, 16bit). The approximate duration is 2:10 hours. Drums used are Kick Drum, Snare Drum and Hi-Hat.. There are 104 polyphonic drum set recordings (drum loops). All the 104 polyphonic drum set recordings are annotated in time and the drum being played. The recordings are from three different sources:

- Real-world, acoustic drum sets (RealDrum)
- Drum sample libraries (WaveDrum)
- Drum synthesizers (TechnoDrum)

Method

Obtaining the templates

- Get the list of audio and the corresponding annotation files
- Concatenate all the audio and the corresponding annotations in time
- Considering the ground truth activations as the initialization of the activation matrix in NMF, we estimate the basis vectors i.e the templates for individual drums for each audio file
- The templates are saved in the file-system as a numpy matrix

Predicting Activations

- Load the saved templates
- Compute the spectrogram of the test audio file
- Considering the generated templates as the initialization of the basis matrix in NMF, we estimate the activations vectors i.e the activations for individual drums
- Apply peak picking on the computed activations in order to select the prominent bursts

Evaluation and Conclusions

We created the templates from the drum recordings produced by drum synthesizers and we evaluated our system on the real world drum recordings i.e. drum recordings played by an actual drum-set

- We used F-measure as our evaluation metric and we use the mir_eval library to compute the metrics
- The F-measure we get is 0.659 and the reported F-measure in the literature using NMF on the IDMT-SMT dataset is around 0.7
- In the current implementation the number of basis vectors assigned is one per drum. We can explore the possibility of assigning more than one basis vectors to a drum.

- Throughout the evaluation, we can observe that the Snare Drum(SD) in particular have many spurious activations of significant magnitude. Tampering with the computed template of SD can reduce the spurious activations like smoothening.

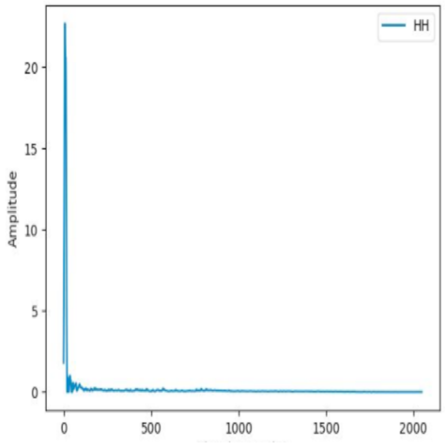

FIGURE 3.1: HH template

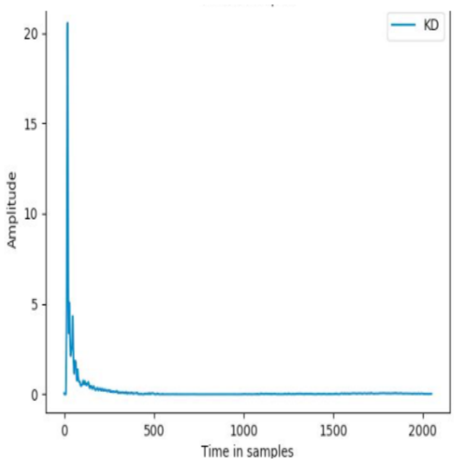

FIGURE 3.2: KD template

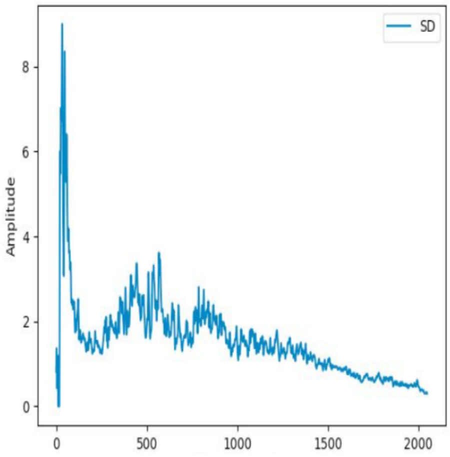

FIGURE 3.3: SD template

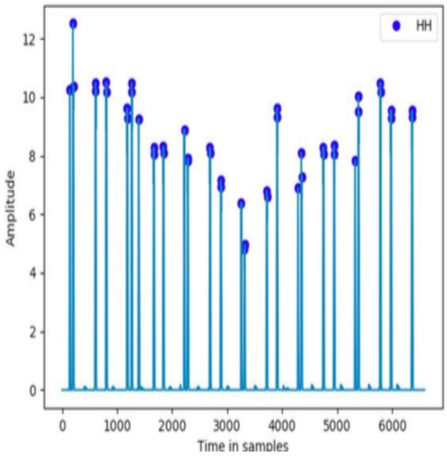

Figure 3.4: HH activation

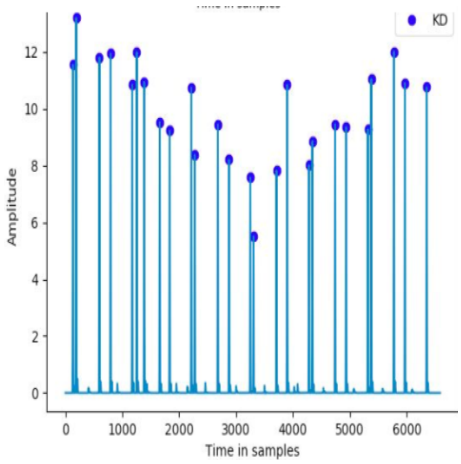

Figure 3.5: KD activation

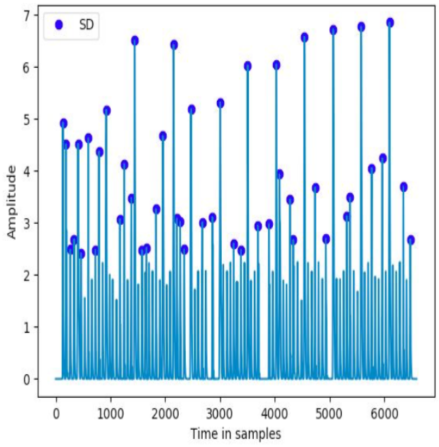

Figure 3.6: SD activation

Chapter 4

State of Art Drum Transcription Model

Richard Vogl Et. al in their paper, Drum Transcription via Joint Beat And Drum Modeling Using Convolution Reccurent Neural Networks have presented the state of art drum transcription system. They use metadata like meters and tempo as they think it helps to transcribe better. The authors claim that the system has the means to utilize information on the rhythmical structure of a song. They evaluate three different architecture designs in particular viz. recurrent, convolutional, and recurrent-convolutional neural networks and they show that learning beats jointly with drums can be beneficial for the task of drum detection. CRNNs should result in a model, in which the convolutional layers focus on acoustic modeling of the events, while the recurrent layers learn temporal structures of the features.

About the Metadata

- Additional information required by a musician to perform a piece
- This information comprises meter, over- all tempo, indicators for bar boundaries, indications for local changes in tempo, dynamics, and playing style of the piece.
- To obtain some of this information, beat and down-beat detection methods can be utilized.
- Beats provide tempo information
- downbeats add bar boundaries
- combination of both provides indication for the meter within the bars

Feature Extraction

- They use log magnitude spectrogram as an input
- Window size = 46 milliseconds (2048 samples)
- The frequency bins mel scaled
- The positive first order differential over time of this spectrogram is calculated and concatenated.

Neural Network Models

Convolutional Neural Network

- While plain CNNs do not feature any memory, the spectral context allows the CNN to access surround- ing information during training and classification.
- How- ever, a context of 25 frames (250ms) is not enough to find repetitive structures in the rhythm patterns.
- Therefore, the CNN can only rely on acoustic, i.e., spectral features of the signal.
- with advanced training methods like batch normalization, as well as the advantage that CNNs can easily learn pitch invariant kernels, CNNs are well- equipped to learn a task adequate acoustic model.

Convolutional Bidirectional RNN

- Convolutional recurrent neural networks (CRNN) repre- sent a combination of CNNs and RNNs.
- In this work, the convolutional layers directly process the input features
- The recurrent layers are placed after the convolutional layers and are supposed to serve as a means for the network to learn structural patterns

Drum Detection with Oracle Beat Features

- In addition to the input features, the annotated beats, represented as the target functions for beats and downbeats, are included as input features.
- Using the results of these experiments, it can be investigated if the prior knowledge of beat and downbeat positions is beneficial for drum detection.

Joint Drum and Beat Detection

- As input for training, again, only the spectrogram features are used.
- Targets for training of the NNs comprise, in this case, drum and beat activation functions.
- Beats and drums are closely related, because usually drum pat- tern are repetitive on a bar-level (separated by downbeats) and drum onsets often correlate with beats.

Commonalities between NN architectures considered

- All NNs are trained using the same input features
- The RNN architectures are implemented as bidirectional RNNs (BRNN)
- the output layers consist of three or five sigmoid units, representing three drum instruments under observation (drum only) or three drum instruments plus beat and downbeat (drum and beats)
- the NNs are all trained using the RMSprop optimization algorithm

Joint Detection

- In this work, neural networks for joint beat and drum detection are trained in a multi-task learning fashion.
- It is possible to extract drums and beats separately using existing work and combine the results afterwards
- They show that it is beneficial to train for both tasks together, allowing a joint model to leverage commonalities of the two problems

Discussion

- CNN with a large enough spectral context (25 frames in this work) can detect drum events better than RNNs.
- The results for CNNs and CRNNs show that convolutional feature processing is beneficial for drum detection.

- The findings considering drum detection results for multi-task learning are also promising
- The low results of beat and downbeat tracking are a limiting factor
- As a next step, to better leverage multi-task learning effects, beat detection results must be improved
- Trained RNNs seem to learn only an acoustic, but not a structural model for drum transcription. (Due to the less number of time-steps used)
- The difference between the magenta model and the model considered in this paper is that there's no concatenation of the predicted beats and downbeats.

Chapter 5

Implemented Model

The model is based on the premise that the frames containing an onset are more important than those who don't in terms of perception and hence transcription. This idea is manifested by training a dedicated note onset detector and using the raw output of that detector as additional input for the framewise note activation detector. The input data representation used is mel-scaled spectrograms with log amplitude of the input raw audio with 229 logarithmically-spaced frequency bins, a hop length of 512, an FFT window of 2048, and a sample rate of 16kHz.

Model Description

The onset detector is composed of the acoustic model, followed by a bidirectional LSTM with 128 units in both the forward and backward directions, followed by a fully connected sigmoid layer with 88 outputs for representing the probability of an onset for each of the 88 piano keys.The frame activation detector is composed of a separate acoustic model, followed by a fully connected sigmoid layer with 88 outputs. Its output is concatenated together with the output of the onset detector and followed by a bidirectional LSTM with 128 units in both the forward and backward directions. Finally, the output of that LSTM is followed by a fully connected sigmoid layer with 88 outputs.

`Image source: https://arxiv.org/pdf/1710.11153.pdf`

Dataset

Datasets of Tabla recordings are very limited in both size and variability. Hence we created our own dataset by taking help from the UPF's Tabla Solo Dataset. We also

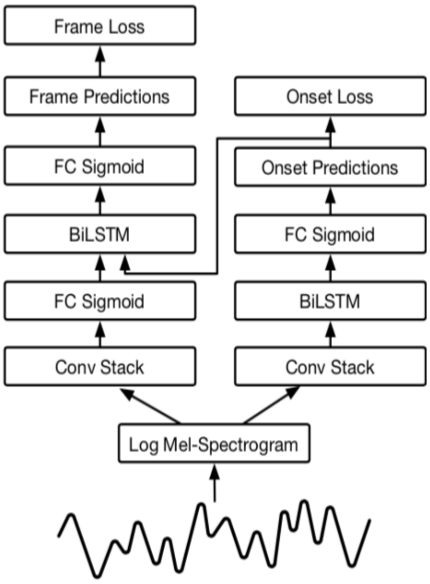

FIGURE 5.1: Model Architecture

used the isolated tabla sounds recorded in our lab itself.
Strategies to generate tabla recordings:

- Reading the annotations of the Tabla Solo Dataset and substiuting the recorded isolated strokes
- Generate your own compositions with varying tempo value[70, 80, 100]
- Interchanged beat sequences among the the available scores to generate new compositions(1415)

Training RNNs over long sequences can require large amounts of memory. To expedite training, we split the training audio into smaller files. We split the audio files in 20 second splits which allowed us to achieve a batch size of 8 during training.

Model Specifics

- The input data representation used is same as that of the magenta's implementation.
- The fully connected sigmoid layers had 19(number of bols).
- Our loss function is the sum of two cross-entropy losses: one from the onset side and one from the note side.

- The learning rate was set to 0.0006 and was decayed with a rate of 0.98 every 10000 iterations.

Results

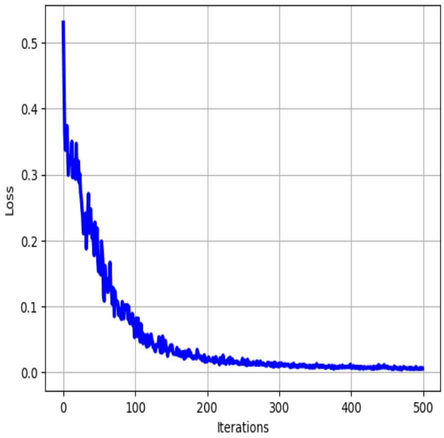

Figure 5.2: Loss

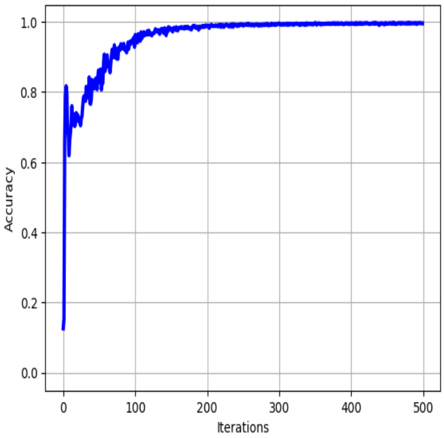

Figure 5.3: Accuracy

Discussion

We shall compare the results of the model which has been implemeted with changes in the architecture and with the state of art system for drums.

- Both the models indicate to the fact that multitask learning is certainly beneficial. We trained the current model without the onset detector and observed that upto 200 iterations, the f-measure was 15% less $(0.82 - 0.97)$.

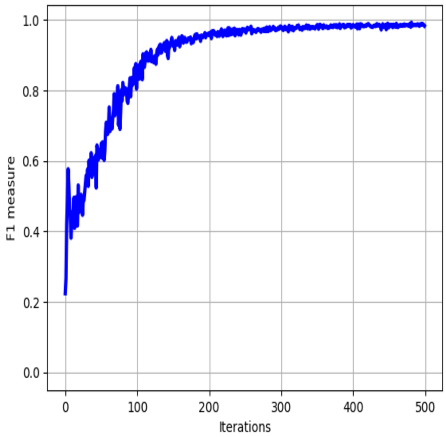

Figure 5.4: F-Measure

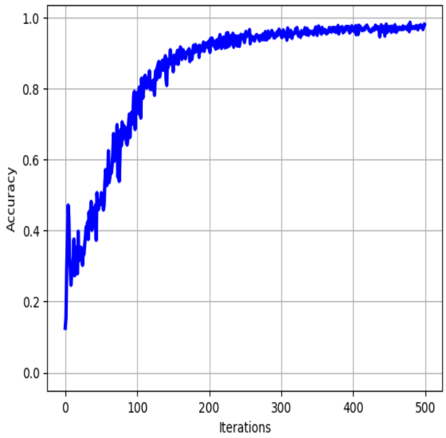

Figure 5.5: precision

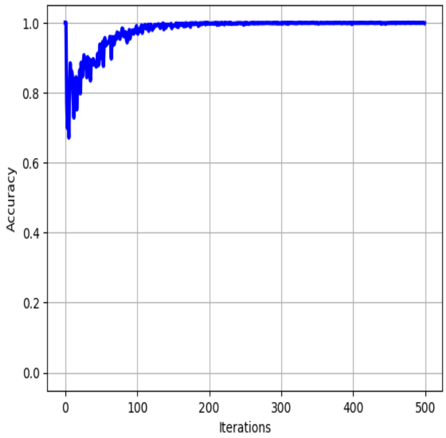

Figure 5.6: recall

- Since in the drum transcription model, the beats and the downbeats are computed and used as labels, it renders the results prone to errors. As against here we are joinly detecting the onsets and the bols which do not come from any extra

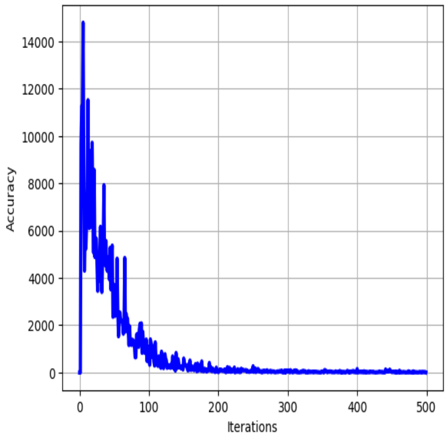

Figure 5.7: False Negatives

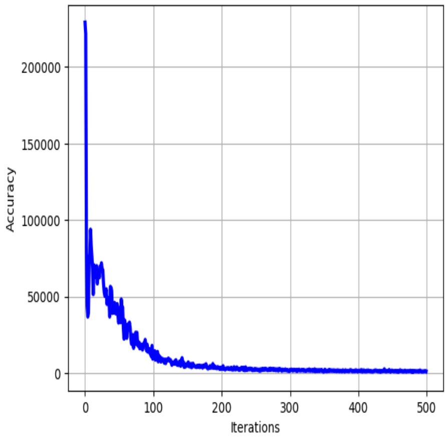

Figure 5.8: False Positives

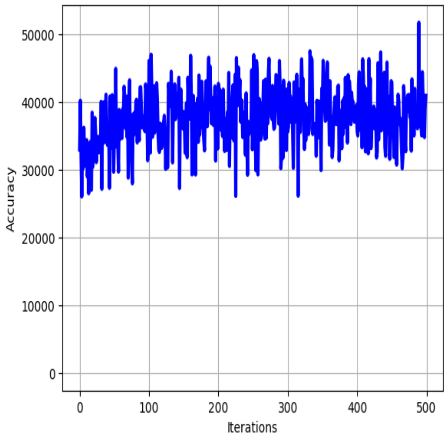

Figure 5.9: True Positives

processing but from the dataeset.

- When you train the model with forward only LSTMs instead of the bi-directional ones, there's only a small accuracy drop of around 8% which is encouraging because forward only LSTMs can be used

- The results show that near perfect transcription has been achieved.
- The reason for such high accuracy is that the dataset over which the model has been trained is less complex.
- Also relatively, transcribing percussion instruments is easy because of the less variability in sound
- We can possibily achieve better results by combining an acoustic model with a language model
- Another direction is to go beyond traditional spectrogram representations of audio signals
- The inference is being carried out after splitting the audio files because the placeholders for data have been hardcoded with the variable shape. This can be undone because TensorFlow allows unspecified variable size
- Converting the transcriptions to drum recordings to analyse the perception quality of the transcriptions.