Transcription of Percussion Instruments

Pranav Sankhe

Supervisor : Preeti Rao
Indian Institute of Technology, Bombay

May 30, 2019

TASK DESCRIPTION

- **Drums**
  - Detection and classification of drum events
  - Estimating the onset and the offset timings of the classified drum events
  - Transcription of drum notes in the presence of polyphonic music
- **Tabla**
  - Detection and classification of tabla events
  - Estimating the onset and the offset timings of the classified tabla events
  - Transcription of tabla notes in the presence of polyphonic music

Particularities & Challenges

- Transient-like sound components exhibiting broadband spectra
- Locally periodic
- Sounds are superimposed
- High variability within a single tabla
- Special playing techniques introduce variety in the spectrum
- Interference of multiple instruments
- Insufficient real world datasets

Design patterns a transcription system

- Feature Representation
  - Natural Choice: Time and Frequency Representation :- **Spectrograms**
  - Includes pre-processing steps intended to emphasize the target drum signal
    - Band-pass filters with predefined center frequencies and bandwidths
    - Harmonic-Percussive Source Separation
- Activation Function
  - Map feature representations into activation functions which indicated the activity level of drum instruments
  - **Technique:** NMF, Probabilistic Latent Component Analysis, Deep Neural Networks
- Event Segmentation: Detect the temporal location of musical events in a continuous audio stream.
- Event Classification: Aims at associating the instrument type with the corresponding musical instrument

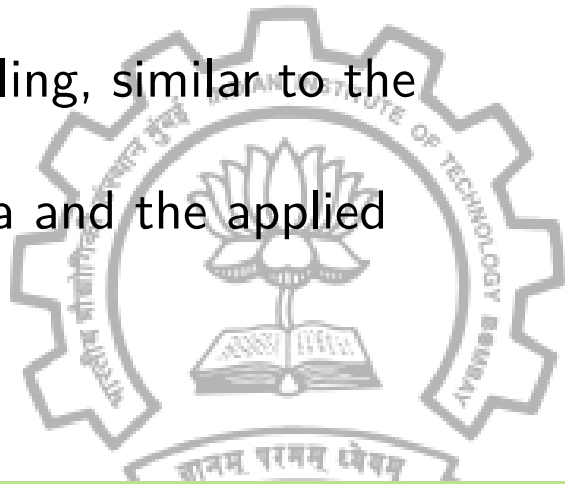
Feature Representations

- Natural Choice: Time and Frequency Representation :- **Spectrograms**
- Includes pre-processing steps intended to emphasize the target drum signal
  - Band-pass filters with predefined center frequencies and bandwidths
  - Harmonic-Percussive Source Separation

Activation Based Approach

Activation funcstions generates the activity of a specific instrument over time.

- **Matrix Factorization Algorithms**
  - Matrix factorization algorithms aim at decomposing the spectrogram into basis functions and their corresponding activation functions
  - Assumes that the target signal is a superposition of multiple, statistically independent sources. Problematic since the activations of different drums are usually rhythmically related.
- **Deep Neural Networks**
  - 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



NMF-based Approach

- Let $X \epsilon R(K \times T)$ be the magnitude spectrogram
- 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 used for NMF based approach

The dataset consists of 608 WAV files (44.1 kHz, Mono, 16bit). The approximate duration is 2:10 hours. Drums involved: Kick Drum, Snare Drum and Hi-Hat

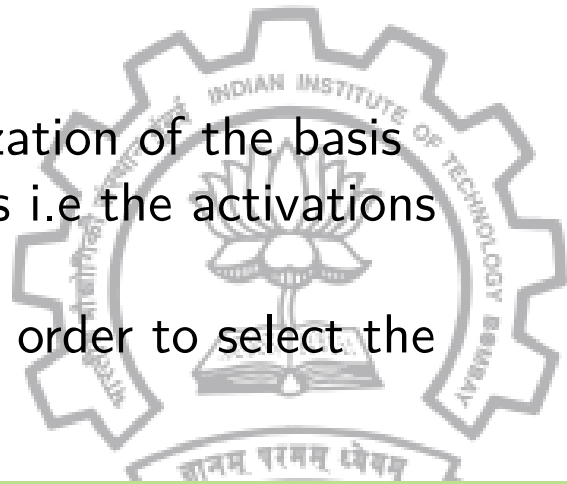
- There are 104 polyphonic drum set recordings (drum loops)
- All the 104 polyphonic drum set recordings are annotated in time and the drum being played.

The recordings are from three different sources:

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

Method

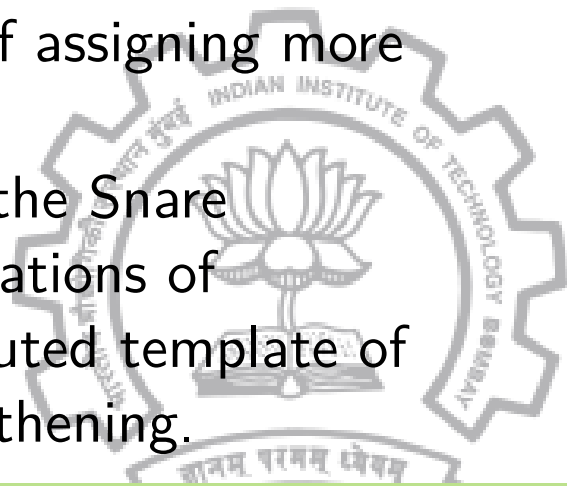
- **Obtaining the templates**
  - Get the list of audio and the corresponding XML files
  - Concatenate all the audio and the corresponding annotations in time
  - Considering the ground truth activations as the initialization of the activation matrix in NMF, we estimate the basis vectors i.e the templates 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.

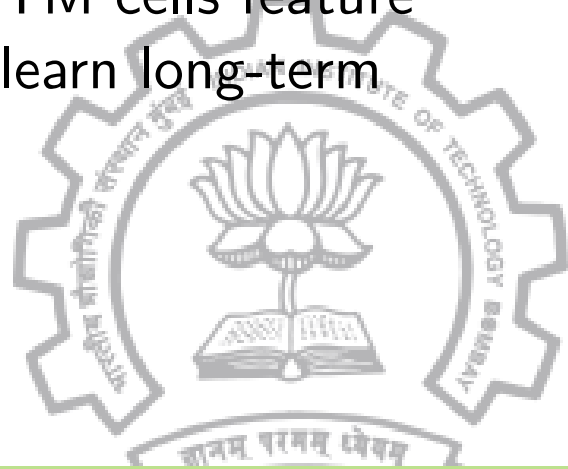


DNN-based Approach

- In contrast to the NMF-based systems, the mixture spectrogram X is processed as a time-series in a frame-wise fashion, i.e., we insert each individual spectral frame xt sequentially into a trained RNN.
- Basic RNN Model:- RNNs represent an extension of DNNs featuring additional recurrent connections within each layer which provide the single layers with the previous time step's outputs as additional inputs

DNN-based Approach

- Bidirectional RNNs:- BRNN layers consist of two RNN sub-layers, one with recurrent connections in forward direction and the other in backward direction
- RNNs with Label Time Shifts:- Comparable results with Bidirectional RNNs can be achieved with RNNs using a label time-shift which allows an RNN to access information before and after the true start of drum sound events.
- LSTMs:- In addition to recurrent connections, LSTM cells feature an internal memory, which allows the network to learn long-term dependencies.
- GRUs:- Less parameters than LSTMs



State of Art Drum Transcription Model

- The authors evaluated and compared the performance of CNNs, RNNs and CRNNs
- CRNNs 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.
- They show that learning beats jointly with drums can be beneficial for the task of drum detection.

State of Art Drum Transcription Model

- 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.
- 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

Magenta's Piano Transcription 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 Architecture

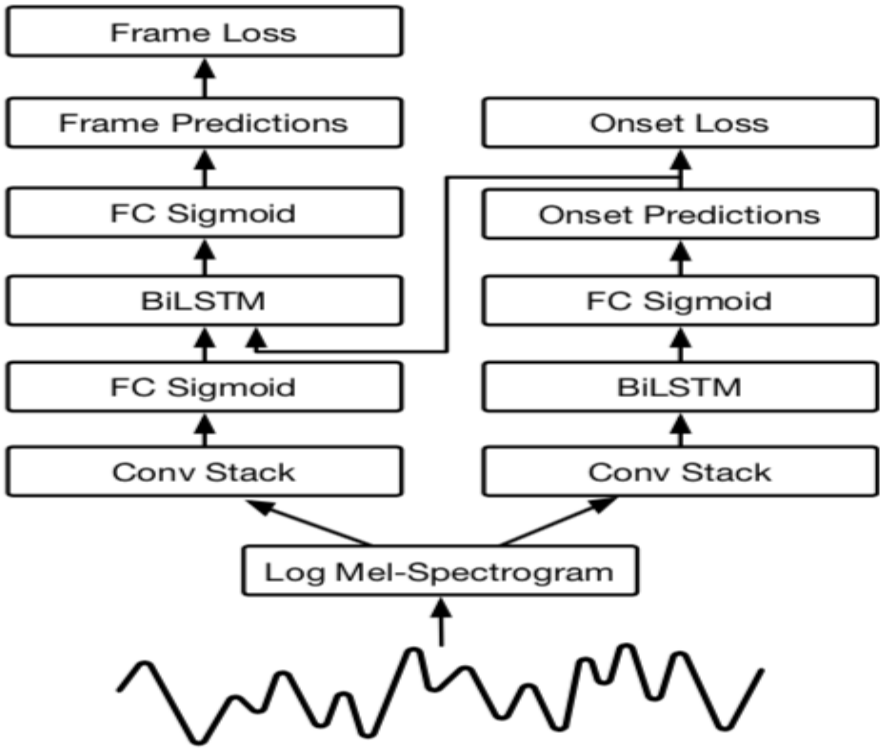
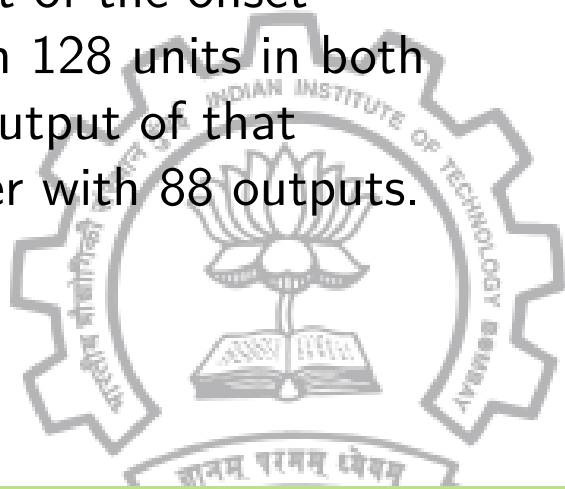


Figure: Model Architecture

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

Magenta's Piano Transcription Model

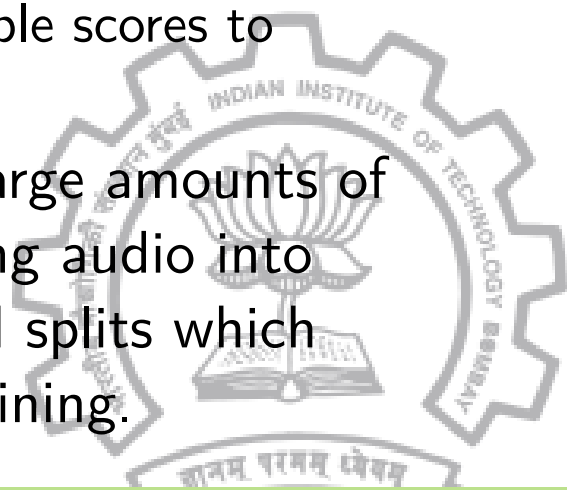
- **Model Description**
  - The onset detector is composed of the acoustic model, followed by a bidirectional LSTM with 128 units in both the forward and backward directions, followed by a fully connected sigmoid layer with 88 outputs for representing the probability of an onset for each of the 88 piano 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.



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 used the isolated tabla sounds recorded in our lab itself.

- Two 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) units instead of 88.
- 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.
- Inference is performed over the split audio files

Results

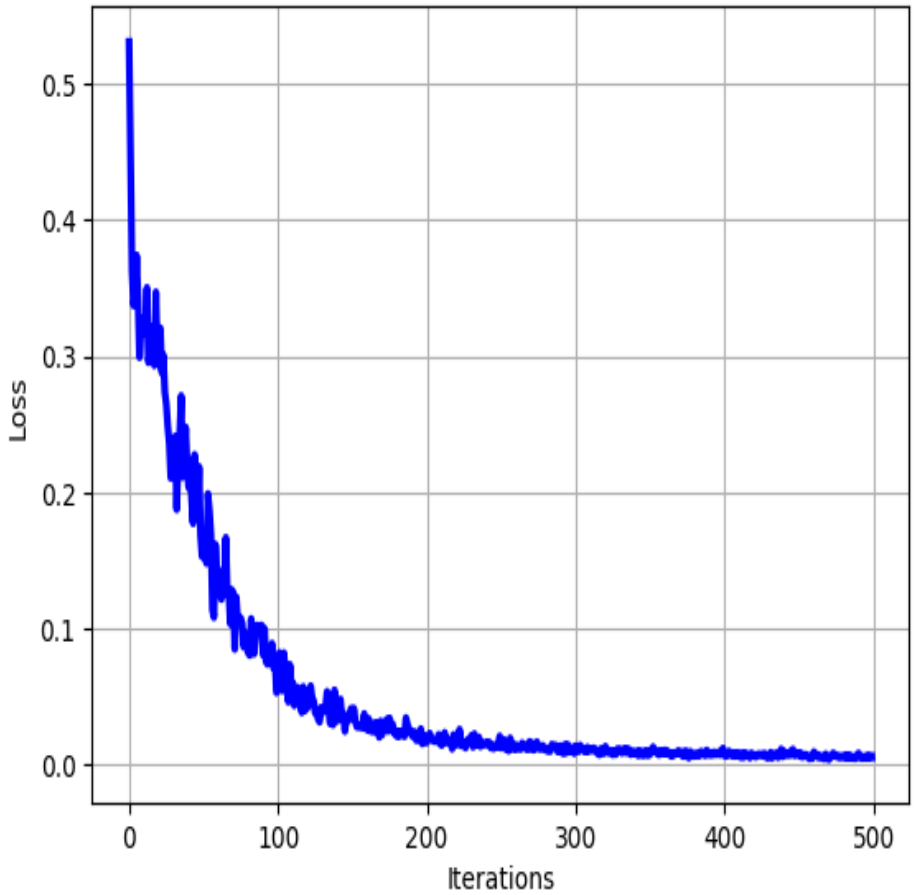


Figure: Combined Loss

Results

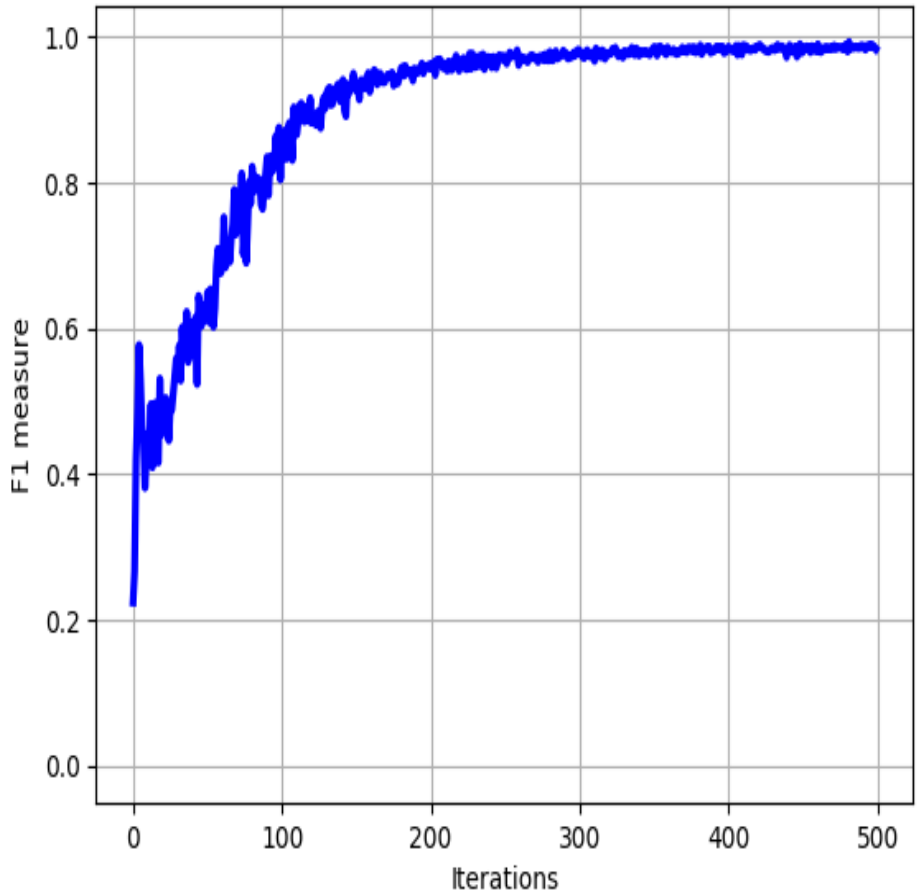


Figure: F-Measure

Results

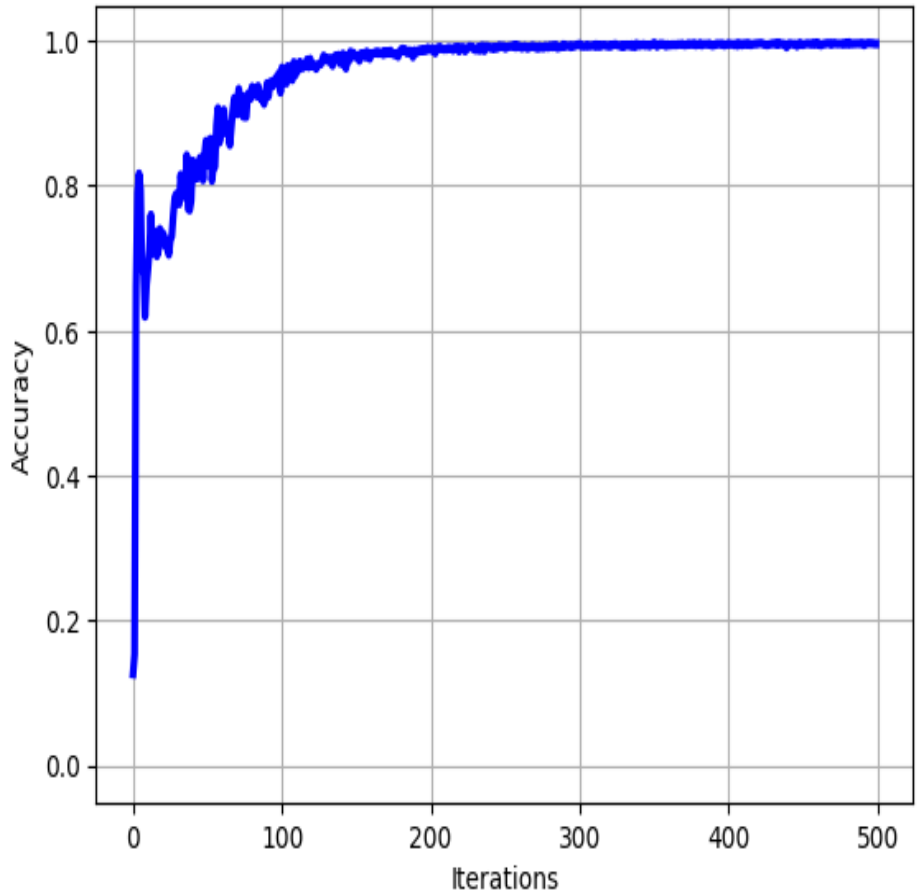


Figure: Accuracy

Discussion

- 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
- 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.