



Classifying Music Subgenres Using Neural Networks

Presented by: Ryan Cafarelli, Sam Schrader,
Rohith Kumar Sura, and Daniel Tucek



Introduction

Problem: Automatic classification of EDM subgenres

- Useful for recommendation systems and suggesting tags for user content

Generalization of our networks:

- Input- Audio files that are preprocessed into other features (e.g. Spectrograms and MFCCs)
- Output- The predicted label for the music subgenre



Related Work

- A Genre classification using Support Vector Machine (SVM) and their comparison in terms of accuracy with other methods.
- A classifier for Jazz subgenre music classification using an LSTM layer as the core.
- A 5-layer Independently RNN with scattering coefficient for preprocessing the data to complete the music genre classification.
- Music genre classification with different activation functions in a neural network.
- Examined the performance of convolutional neural networks (CNN) and recurrent neural networks (RNN).

Technique	No. of Genres	Accuracy
SVM	5	92%
LSTM	3 Jazz subgenres	80.30%
IndRNN	7	96%
RNN	10	33%
CNN	10	59%



Dataset and Features

- Top 100 songs in each of 23 genres on Beatport as of November 29, 2016 (Set 1 from Caparrini)
 - Pulled 2-minute audio samples from Beatport using a web scraper
 - 2,258 out of 2,300 tracks (~98.2%)
- Different preprocessing for each model
 - Spectrograms for CNN
 - Time series of MFCCs for LSTM

Genre	Track Count
TechHouse PsyTrance BigRoom HardDance Techno Minimal Trance	100
FutureHouse ElectroHouse Dubstep House	99
IndieDanceNuDisco ReggaeDub GlitchHop HardcoreHardTechno DrumAndBass DeepHouse ProgressiveHouse	98
FunkRAndB	97
Breaks ElectronicaDowntempo	96
Dance	94
HipHop	93

Methods

CNN

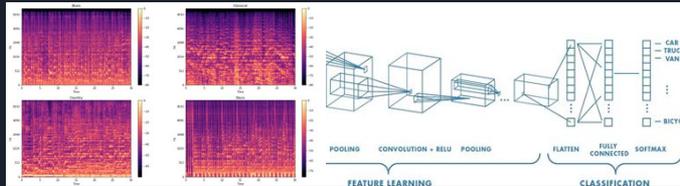
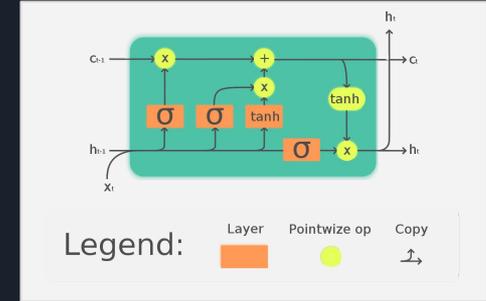


Image classification of audio spectrograms

- Segmentation of image into 5 second intervals
- Multi-layer approach:
 - Convolution (filters: 8, 16, 32, 64, 128)
 - Normalization
 - Activation (relu)
 - Max Pooling
- Fully connected classification:
 - Flatten
 - Dropout (30%)
 - Dense (softmax)

Packages (Librosa, Keras, Adam)

RNN-LSTM



- Input: MFCCs that were calculated by dividing the audio time series into equal length segments
- Layers:
 - Two LSTM (tanh activation and sigmoid recurrent activation)
 - Dense (relu)
 - Dropout (30%)
 - Dense (softmax)

Packages (Librosa, Keras, Adam)

Preliminary Results

CNN

Training: 21%

Testing: 19%

Parameters/Hyperparameter:

- Batch size: 32
- Learning Rate: 0.0005
- Epochs: 20
- Image size: 720x720

Experiments to hopefully improve:

- More training data
- Fewer nodes in model

RNN-LSTM

Training: 62%

Validation: 43%

Testing: 45%

Hyperparameters:

- Batch size: 32
- Learning Rate: 0.001
- Epochs: 30

