SPECTRAL CONVOLUTIONAL NEURAL NETWORK FOR MUSIC CLASSIFICATION

Thomas Lidy
TU Wien,
Institute of Software Technology
and Interactive Systems
lidy@ifs.tuwien.ac.at

ABSTRACT

Our system submitted to the MIREX 2015 music/speech classification and detection as well as the audio classification tasks for genre and mood recognition comprises a Convolutional Neural Network approach adapted to psycho-acoustically motivated spectral input to the network.

1. INTRODUCTION

Convolutional Neural Networks have shown tremendous success in image retrieval in the past few years. Also in Music Information Retrieval, their applicability has been shown to benefit some tasks, such as Onset detection. We employ a first version of a Convolutional Neural Network to the music/speech classification and detection as well as the audio classification tasks for genre and mood recognition.

2. APPROACH

Even though in image retrieval one of the claimed benefits is that “raw” data can be used as the input to a neural network to automatically learn features from it to classify and detect objects on images, it has been argued that raw untransformed data is usually not beneficial as an input to neural networks that are trained on audio data. Typically the first step is a spectral analysis, followed by some psycho-acoustically adaptation of the spectral bands.

2.1 Preprocessing

In our case, we use spectrogram analysis using the periodogram function with a window length of 1024 frames (on input data with 44 kHz). Then the resulting frequency bands are grouped to 40 Mel-frequency bands using a Mel filter. Next, a log-transform is applied on the frequency data.

2.2 Sampling

Instead of using an entire audio file (every analyzed spectrogram frame) as input, we sample only short spectrogram segments at a few random positions. Concretely, in this submission, we sample from 15 random positions per audio input file and extract 80 consecutive spectrogram frames with 40 Mel bands each from these positions. This leads to 15 40x80 input matrices per audio file as an input to the convolutional neural network.

2.3 Neural Network Architecture

We use a Convolutional Neural Network with 1 Convolutional Layer, 1 Fully Connected Hidden Layer and 1 Softmax Layer for class output.

The Convolutional Layer processes 320 input units, which are interpreted as a (40x80) image. On this image, a Filter stage is applied, with 15 filters each of size (12x8). Then a Max pooling stage takes place with size (2x1), i.e. two vertical frequency bands are joined to reduce the vertical resolution (frequency bands). The output of the Convolutional Layer is fed into a fully connected hidden layer with 200 units. Its output is processed by a Softmax Layer with one output unit for each class to be predicted. The class prediction is decided by the unit that has maximum softmax value.

2.4 Training and Validation

The network is trained over 150 epochs with a learn rate of 0.05. The input training set is in fact split in 80:20 fashion where only 80% of input data are actually used for training and 20% of the data are used to validate the accuracy of the model. The model is adapted in each epoch using Gradient Descent and a mini-batch-size of 40 instances. If a better model is found before 150 epochs, this one will be used, but training stage will always run through 150 epochs.

MIREX is running a three-fold cross-validation, but above approach will be used in each of the cross-validation runs (i.e. training input is 80% of 66% of the full data set).

3. IMPLEMENTATION

The system is implemented in Python 2.7 using numpy, scipy and other libraries for scientific computation, and the Theano library for neural network computation (which supports both CPU and GPU computation).
Input/output formats and command line calls follow mostly the MIREX recommendations.