TECHNICAL UNIVERSITY IN KOSICE

FEI
FAKULTA
ELEKTROTECHNIKY
A INFORMATIKY

Stochastic modeling and data analysis

2020

Authors: Kukhar Denys,
Lutak Kristian

Scope knowledge needed

Nowadays, deep learning is more and more used for Music Genre Classification: particularly Convolutional Neural Networks (CNN) taking as entry a spectrogram considered as an image on which are sought different types of structure.

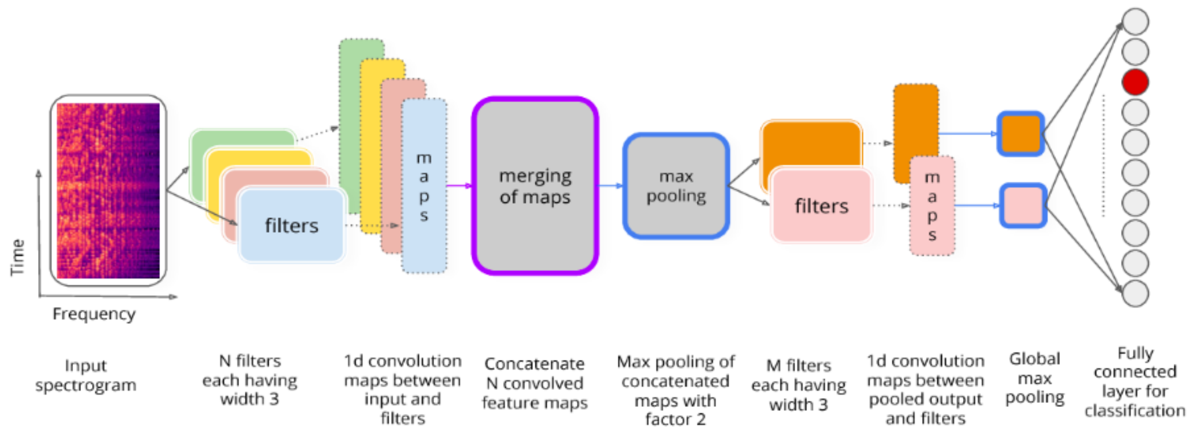
Convolutional Neural Networks (CNN) are very similar to ordinary Neural Networks: they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they still have a loss function (e.g. SVM/Softmax) on the last (fully-connected) layer and all the tips/tricks we developed for learning regular Neural Networks still apply.

ConvNet architectures make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. These then make the forward function more efficient to implement and vastly reduce the number of parameters in the network.

They are capable of detecting primary features, which are then combined by subsequent layers of the CNN architecture, resulting in the detection of higher-order complex and relevant novel features.

The dataset consists of 1554 audio tracks each 2-3 seconds long. It contains 2 categories (healthy, pathological), each represented by 777 samples. The samples are all in .wav format. Each sample represents one of three (i, u, a) vowel sounds of the same lengths.

Network design


Time
Frequency
filters
m a p s
merging of maps
max pooling
filters
m a p s
Input spectrogram
N filters each having width 3
1d convolution maps between input and filters
Concatenate N convolved feature maps
Max pooling of concatenated maps with factor 2
M filters each having width 3
1d convolution maps between pooled output and filters
Global max pooling
Fully connected layer for classification

Experiments

In scope of this project we have tried many strategies and applied many adjustments, both major and minor. We found out that analyzing sound performs better when pathologic is being detected based on vowel samples.

Compile/train the network using Stochastic Gradient Descent(SGD). Gradient Descent works fine when we have a convex curve. But if we don’t have a convex curve, Gradient Descent fails. Hence, in Stochastic Gradient Descent, few samples are selected randomly instead of the whole data set for each iteration.

Firstly, we compared performance when using Adam as optimazer that led us towards better results. That’s the reason we chose to stick with this. Secondly, we have run tests indicating of how much epochs should be optimal. After 50 epochs accuracy seems not to change much.

One thing was mentioning is that with such analyzing sound problem we basically fully rely on quality of the given data. Even if you try to examine the class of sample by ear you will definitely find yourself confused many times.

With less samples we have managed to achieve 70% accuracy in predicting. When more data were loaded to training though the accuracy began to slightly descend. Having 777 samples for each category we have got 64% accuracy.

Suggested implementation and program flow

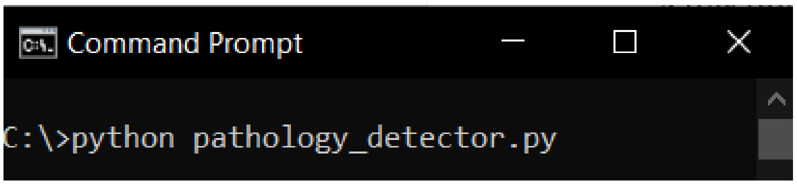
In order to create specgram of the sound we iteratively use specgram function from matplotlib library for conversion WAV files to PNG format. All audio files primarily placed in its directory in wav_data will have its specgram generated in corresponding folder img_data.

Next, we split 80% data for training and 20% for testing against, we perform mutations on the data such as zooming and random transformations, rescale all pixel values from 0-255, so after all our pixel values are in range (0, 1).

After this we create CNN model according to the network design diagram before compiling. Once the model is trained, we write prediction results made with our model to CSV file.

Example of usage

If you want to test the project with your own picture specgrams, replace test dataset in code with the one yours and comment image generating section out. To launch the program in default mode (no changes, generating of pictures), run python script with python.



```
C:\>python pathology_detector.py
```