Ahem... detection in deep learning
Francesco Gadaleta
1. Choose content
2. Prepare script
3. Talk
4. Post-production
5. Publish
Ahem deep learning... ahem

The first artificial brain that provides support to humans who deal with sound... and podcast episodes at worldofpiggy.com
Sound Feature Engineering

the easy part

- Compute **MFCC** Mel-frequency **cepstral coefficients**
- Compute **chromagram** from a waveform or power spectrogram
- Compute **spectral contrast**, using method defined in [IEEE2002]
- Computes **tonal centroid** features, as described in [AMCMM 2006]
Feature Engineering

- Compute feature
- Normalize
- Keep if useful

Repeat for all features
Expert knowledge required
Repeat for all sound types
Time & Frequency
Representing the same signal in different domains
Time Space
how humans look at sound

Fig. Sampling and quantization of a signal (red) for 4-bit LPCM

Amplitude

Time
I’m gonna represent “every” function as a sum of sine and cosine waves

Joseph Fourier

\[
f(x) = \sum_{n=0}^{\infty} A_n \sin (nx) + B_n \cos (nx)
\]

\[
f(x) = \sum_{n=0}^{\infty} A_n \sin (nx) + B_n \cos (nx)
\Rightarrow f(x) \cos (3x) = \sum_{n=0}^{\infty} A_n \sin (nx) \cos (3x) + B_n \cos (nx) \cos (3x)
\Rightarrow \int_0^{2\pi} f(x) \cos (3x) \, dx = \int_0^{2\pi} \left( \sum_{n=0}^{\infty} A_n \sin (nx) \cos (3x) + B_n \cos (nx) \cos (3x) \right) \, dx
\]

\[
= \sum_{n=0}^{\infty} A_n \int_0^{2\pi} \sin (nx) \cos (3x) \, dx + B_n \int_0^{2\pi} \cos (nx) \cos (3x) \, dx
\]

\[
= B_3 \int_0^{2\pi} \cos (3x) \cos (3x) \, dx
\]

\[
= \pi B_3
\]

\[
A_n = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin (nx) \, dx
\]

\[
B_n = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos (nx) \, dx
\]
Frequency Space
how that smartass of Fourier looked at signals
From Time to Frequency and back

\[ F(k, l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) e^{-i2\pi \left( \frac{ki}{N} + \frac{lj}{N} \right)} \]

\[ f(a, b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k, l) e^{i2\pi \left( \frac{ka}{N} + \frac{lb}{N} \right)} \]
Can we “see” sound?

LSD
When data is not enough, make data
### C³ Data Preparation

**Collect**

- Enough podcast episodes
- Enough *ahem* sounds at [worldofpiggy.com/podcast](http://worldofpiggy.com/podcast)

**Clean**

- Select and process sound to calculate spectral signal
- This is a linear operation in $O(n \log(n))$ (FFT over $n$ samples)

**Convert**

- Convert each sound frame to spectral image (png, jpeg, tif...)
Ahem detector explained

6000 samples
3000 overlap
\(~ 1/6 \text{ sec.} @ 44.1 \text{ Hz}\)

Time space

Frequency space

463x191 px
32 bit RGB
10 seconds

441000 PCM samples

150 Spectral images
12
minutes
10000
Spectral Images
Convolutional Neural Networks

How good is an artificial brain to noise detection?
Convolutions first
Average pixels in the neighborhood of central pixel

```
0 0 0 0 0
0 1 1 1 0
0 1 1 1 0
0 1 1 1 0
0 1 1 1 0
0 0 0 0 0
0 0 0 0 0
```
Take the difference of adjacent pixels
GPU powered convolutions

~1 hour

20 hours

Performance benchmark provided by NVIDIA
https://devblogs.nvidia.com/parallelforall/cudnn-v2-higher-performance-deep-learning-gpus
Deep Learning for Sound/Image

Similar topology of GoogLeNet

Repeat 2 times

- Convolution + nonlinearity
- Max pooling
- Convolution + pooling layers
- Fully connected layers

- 32 filters 3x3
- 2x2 max pool
- Dropout = 25%
- Dropout = 50%
6000 samples
3000 overlap
~ 1/6 sec. @ 44.1 Hz

Time space

Frequency space

[start, start+6000]
Data scientists are... *ahem* really cool and deep learning *ahem* is... *uhhhhh*... fantastic!

Data scientists are [...] really cool and deep learning is [...] fantastic!
demo
It works! with just “a few” false positives

More training = Less errors

WAIT

WHAT?
Performance

5 epochs
38 hours
GPU GTX 780

81%
accurate
on test set
Let's review some concepts

Lack of Data
Deep learning needs a lot of data to work well. Data transformation can help

Think different™
Transform the same problem into something deep learning can solve

Knowledge transfer
*Train* somewhere
*Tune* something
*Predict* somewhere else
NLP, Speech, Image classification
ahem... thanks!
References


Deep learning
http://www.nature.com/nature/journal/v521/n7553/full/nature14539.html

[CONV 2013] ImageNet Classification with Deep Convolutional Neural Networks
Convolution explained

Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

Source pixel

Convolution kernel (emboss)

New pixel value (destination pixel)

\[
\begin{align*}
&\begin{pmatrix}
4 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
\end{pmatrix} \\
&+ (-4 \times 2) \\
&= -8
\end{align*}
\]