Seminar Audio Processing and Indexing - Final Project

A Deep Learning Approach to Instrument Detection and Chord Estimation via Frequency-based Feature Extraction

Francesco Bortolussi
s1928120

January 23, 2018

1 Introduction

The automatic recognition of musical instruments' timbres and pitches has been a focus of study for a number of years; recognizing which instruments are playing in a given piece of music can be helpful in a lot of different ways: when building recommender systems based on musical genres, or to help a user search for specific pieces of music by expanding music's metadata. By the same token, detecting the exact notes played in a piece of music is of primary importance in the music industry: the demand for transcriptions is very high on a daily basis, especially among the more inexperienced musicians. "Transcription" can be everything from the simple guessing of the chords of a song, to the note for note re-creation on of an orchestral score. In both these applications, there is not yet a definitive approach that would solve all problems, and expert human intervention has to still be present most of the times.

The goal of this project was to explore some of the potential solution to both of these problems; in particular, the scope of this project was to study a deep learning approach to provide solutions to both of the problem while using a general purpose Neural Network.

2 Problem Description

In this project, two problems were studied:

- Instrument Detection: the problem was to recognize and classify different instruments based on short audio files. The goal was to identify single instruments that were playing alone. The classification was conducted with regards to these 4 instruments:
  - Piano.
  - Guitar.
  - String ensemble pizzicato.
  - Hammond Organ.
- Chords Detection: the problem was to recognize the pitch of the individual notes of a chord played by a piano; this meant that the input would be an audio file of a piano chord, and the output would be a textual representation of the individual pitches and the relative octaves (e.g. "C4"). The problem was set up to detect chords in the span of 2 octaves (24 consecutive notes), without loss of generality; there is no intrinsic difference between different octaves, so this limitation would not hinder the effectiveness of the solution.

While the first problem was already been solved with different techniques for individual instrument, algorithms trying to detect multiple instrument still do not yield satisfactory results; the simple approach presented in this paper can be potentially expandable into featuring more instruments at the same time, as it would be compatible with the core ideas of the algorithm. However, this was out of the scope of the project.

The main contribution of this paper was about the detection of chords; the main focus was on analyzing only the piano. In order to have a good understanding of the core idea, it was necessary to simplify the initial approach to a specific case; moreover, it was necessary to obtain a first set of satisfactory results before expanding the project onto a more ambitious ground. An original idea presented in this project had to do with the fact that a chord would be correctly detected if each one of the notes played was detected. In other words, it did not matter if the chord was recognized by its theoretical label. There were a number of problems that were linked to the theoretical labeling of chords:

- Inversions of a chord have a completely different harmonic quality and theoretical function, so it would be debatable to consider them as the same chord.
- Enharmonic creates confusion on the exact name of a chord, and is usually context dependent (sometimes it is instrument dependent). An example would be: "C# major" is equal to "Db major".
- Some of the more complicated chords (e.g. jazz chords) have different way of being named, and it is usually a matter of context (e.g. "C/F" or "FMaj7sus2" can be equivalent).

These theoretical considerations were considered pointless for the scope of this project, as they were tied to problems more related to music theory and harmony, than to the problem of automatic pitch detection. Some other papers explore the possibility of bypassing the individual nature of the notes forming a chord, but they rather focus on the labeling of chords in terms of "major", "minor", "augmented", "seventh", etc.; some of the more recent attempts combine state of the art segmentation algorithms with deep learning techniques to classify chords by their theoretical label [2] [4].

The main assumption made for this project was that the information contained in the frequency spectrum was enough to detect every single note of a chord. One reasoning was that experienced musicians with a developed ear (be it through the possession of "perfect pitch" or not) are able to hear the individual notes that form a chord [1]. This ability is independent of the instrument, and it is not tied to other factors, like how loud the sound source is, or how distorted it is, etc.. If these other factors are not necessary for the recognition of chords, then the spectrum analysis would be sufficient for this task.

Additionally, the fact that skillful people could solve this task meant that it was possible to solve it at all in the first place, with a potential degree of accuracy that would at least push towards 100% (i.e. as good as the human counterpart). This was necessary because of a fundamental property of sound waves: whenever two sounds are summed together, and are played simultaneously, the frequencies are effectively summed. This means that the spectrum would be a representation of the sum of all the individual notes' frequencies. Moreover, all the notes played by an instrument present a fundamental frequency, that describes what note has been played, and the harmonics that make up the "timbre" of the sound. These issues point to the fact that it would be very complex to tackle this task without a machine learning approach, since there would be no direct way to read the frequency spectrum.

As it was already stated, the frequency spectrum contains the information about the pitch of the sound and the timbre. The timbre is entirely dependent on the type and the number of "harmonics". On the other hand, the pitch is dependent on the frequency of the fundamental frequency. By looking at the problems laid out in this paper, it is easy to see that both of them could be solved by using a very similar approach, and that the algorithms could be potentially be combined to have a detection of both notes and instruments.

2.1 Algorithms

The main approach of this project was a deep learning based algorithm. For the first problem, the framework was to build a neural network that would classify 4 different instruments.

For the second problem, the neural network had to detect the notes of any given chord. In order to summarize the chord detection problem into a classification problem, there were 2 possible ways to build a neural network framework:

1. There would be only one neural network with 88 output nodes: every node would correspond to a note on the piano. The downside of this approach was that there would have been multi-

ple outputs for any input; in this case the correct output would be picked as the k nodes with the highest confidence value, which could be imprecise or with a complex implementation.

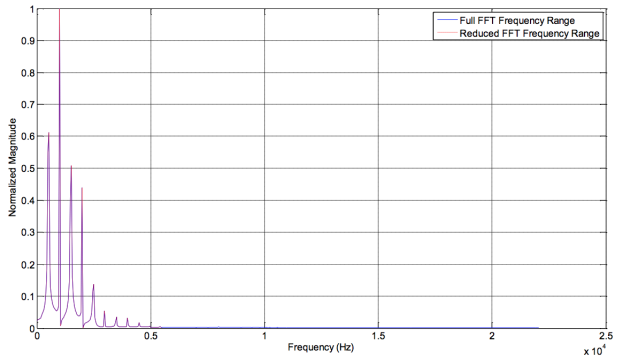
2. There would be one trained neural network per note of the piano (i.e. 88 different neural network). All the neural networks would have the same exact structure, but the weights would be unique to a given note the network has to detect (so there would be only one network effectively, with 88 different sets of weights). As output, the network would have 1 if a given chord contained the individual note, 0 otherwise. This meant that the chord detection would be the output of a series of 88 binary classifiers.

It was chosen to use the second approach, as it could definitely provide more stable results, and it had a less complex implementation. All the neural networks would be trained beforehand, so that the testing would be much faster. Real-time chord feedback was not in the scope of this project, and it would not be necessary in most applications (as the transcription is usually done on an offline source).

The pipeline consisted in two main steps, and the process was identical for both problems:

- Feature extraction: the audio files was analyzed in the following way: for each audio file, the FFT was applied to extract the frequency spectrum; every audio file was about 1 second long. Only the positive frequencies were considered, and only up to 6000Hz. In fact, the pitch of the highest note of the piano is around 4186.01Hz (the overhead is to consider also some of the potential harmonics). From the 6000 remaining points, the output was down-sampled to 3000 points. These points would then be saved in a csv file, as a 3000 dimensional vector.

Figure 1: Example of frequency spectrum.



- Neural Network classification: for both experiments, the data would be fed into a neural network with 3000 input nodes. It was decided to use a fully connected Neural Network, and later in this chapter the motivation behind this decision will be explained. The structure of the network was as following: the optimization was done through Adaptive Moment Estimation ("Adam"); the loss function that was minimized was the binary cross-entropy; the network was comprised of the following layers:
  - Input layer: 3000 nodes
  - 10 Hidden layers with the following nodes: 3500, 2500, 1500, 1000, 800, 750, 600, 500, 400, 250 nodes
  - Output layer: 1 node for the chord detection problem, 4 nodes for the instrument detection problem.

  Every hidden layer had a rectified linear unit (relu) as activation function; the input layer had a sigmoid function as activation function for the chord detection part (because the problem is a binary classification problem), while it had a softmax activation function for the instrument detection part.

2.2 Choice of Neural Network type

It was decided to use a fully connected neural network with multiple hidden layers as a consequence of a process of elimination reasoning. Although other types of neural networks are more complex,

most of the other well known structure did not fit the problem requirements (for the exception of the Convolutional neural network):

- Convolutional Neural Network: this type of network could have worked for the chord detection problem, as the input could have been a graphical representation of the spectrum. Since the CNN mainly trains on images, it would have made sense to use this network if the simpler fully connected solution failed to provide satisfactory results. For more complex implementations, or when considering more instruments, a natural evolution of this project could potentially use a CNN.
- Recurrent Neural Network / LSTM: pitch recognition is independent of prior history, and RNNs usually rely on previous samples to predict future ones. This network was then ruled out, even though some research [...]

3 Dataset

In order to accumulate a large amount of training data, instead of searching for multiple recordings to extract audio from, it was decided to synthesize the sounds. This way, the samples would be generated, allowing for as much training/test data as possible. For both the experiments, random MIDI data was initially generated (both samples of single notes and chords). Then with the help of a Digital Audio Workstation (DAW), the MIDI samples were exported as WAV files.

3.1 Instrument Detection

As previously stated, the 4 instruments considered were:

- Piano.
- Guitar.
- String ensemble pizzicato.
- Hammond Organ.

For this particular experiment, the piano and the string pizzicato sounds were generated using a commercial sample library. On the other hand, the guitar and the organ sounds were synthesized. The difference between the two types of sound generation did not yield any change in the result, as both tools allowed for high quality audio samples. The audio snippets were all very short (about 1 second long); this was to ensure that it would be possible to detect an instrument with any given time window (i.e. as short as a second). The sounds were both single notes and chords.

For each instruments, the number of training samples were about 1000, while the number of test samples were about 120. This meant that the training set was comprised of about 4000 samples, while the test set was comprised of about 500 samples. Every sample was unique in terms of notes that were being played and in terms of velocity; velocity is a MIDI parameter with interval $v \in [0, 127]$, which refers to the "loudness", or the "dynamic" at which the note(s) was being played.

3.2 Chord Detection

Every chord or note necessary for the training/testing of the network was generated with a high quality piano commercial sample library. All the audio samples were about 1 second long.

For each network, the data was generated in the following manner:

- A positive example was defined as "any chord that had the target note in it". This meant that any sample with only the target note, or any chord (up to three notes) with the target note in it, would be labeled as a positive example. About 1100 different positive examples were generated. The samples would differ in which note were being played and in velocity.
- A negative example was defined as "any chord that did not have the target note in it". Any other chord (or note) was a good negative candidate. About 1100 different negative examples were generated, similarly to the positive examples.

Among the 2200 samples for each note, 2000 were used as training data, 200 were used as test data. The 2200 samples were generated for each note, which meant that in total there would potentially be $N = 2200 * 88 = 193600$ samples; for this project, only 2 octaves were considered: from C3 to B4. This meant that there were a total of $N' = 2200 * 24 = 52800$ samples.

4 Experiments

For both problems, the neural networks were first trained on the training set and then tested on the test set. For the instrument detection implementation, it was decided to show the results on a confusion matrix; on the other hand, for the chord estimation algorithm it was decided to report the values for the overall accuracy (in percentage).

4.1 Instrument Detection

The neural network was first trained on the 4000 samples of the training data for 50 epochs. The batch size was equal to 25. The results were as follows:

Table 1: Confusion Matrix

		Actual class			
		Piano	**Guitar**	**Pizz.**	**Organ**
Predicted Class	**Piano**	122	0	0	0
	Guitar	0	122	0	0
	Pizz.	0	0	122	0
	Organ	0	0	0	122

Every sample in the test set was correctly predicted, which meant that the network was working accordingly for the 4 instruments that it was trained on.

It might be possible that with more than one instrument overlapping with each other the network would not work as well. It is also important to point out that it was not tested on a continuous audio file, but instead it was tested on small audio clips. Perhaps if an audio file were to be artificially split into smaller files, the network would not work as well with those, as they would have a different "attack" (i.e. when the note starts) or a different length.

As already stated, a way to expand on this implementation would be to add examples with more than one instrument playing at once. An obvious question would be if would it be possible to train a neural network to recognize combination of sounds which it was not trained with. In other words, it is unclear whether or not an "unseen" pair of instruments playing together (but known individually) would be correctly classifiable. If the answer is negative, then the training would require every possible combination of instruments playing simultaneously, which would increase the necessary training data considerably.

Another possible development would be to recognize instruments when affected by effects, such as "reverb", delay-based effects (e.g. "chorus"), etc.; a neural network can be trained to detect instruments also when augmented by certain types of effects. Although, it would be hard to draw the line on where a certain instrument remains as such, when a heavy usage of some effects would be used (e.g. how much reverb should be used until a piano does not sound like a piano anymore?).

Lastly, another interesting analysis would be to see if "abnormal attack timings" affect the classification. Since the timbre of an instrument does not entirely depend on the attack, it would be interesting to see if the neural network would be able to recognize attack timings that do not naturally belong to an instrument (e.g. a piano with a long attack timing instead of a short one). Some findings show that the attack might be important to the instrument classification problem (commonly referred as "envelope", when the combination of "attack", "sustain", "decay", and "release" is considered) [3].

4.2 Chord Detection

The neural network was trained on each one of the 24 notes considered for 50 epochs. The batch size was set to 25. It was observed that the converge to 100% would happen between the first 5 epochs; convergence was then achieved in under 5 minutes, on a machine without GPU.

For each one of the notes, after the training phase, the accuracy achieved was:

- **Accuracy on the test set** $\Rightarrow$ 100% for each one of the notes.

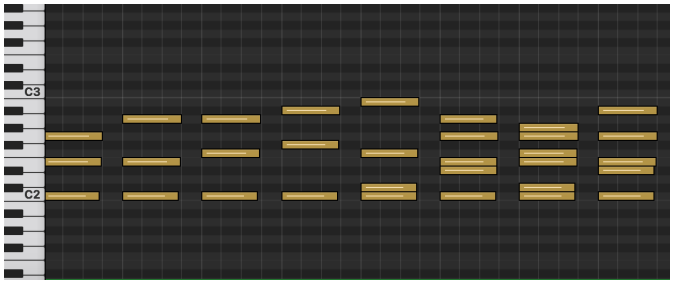
It was clear that the experiment was a success: each target note was recognized between chords of 2-3 notes.

In a follow-up experiment, chords comprised of 4 to 5 notes were also tested successfully (100% accuracy reached), when the network was trained only with chords up to 3 notes. Surprisingly, this meant that the network did not need to train on 4-notes chords to recognize individual target notes inside 4-notes chords. Even if this does not apply (hypothetically) to 20+ notes chords, it is a testament to how it would not probably be needed to train the network on combinations of 20 notes simultaneously. It would be important to take this case into account when expanding on this project because harmonies comprised of 20-30 notes can be achieved in a short amount of time when aided by the sustain pedal; in fact, it is possible to play multiple chords in rapid succession with the sustain pedal activated, so that the sound of all the chords would overlap.

4.2.1 Real World Scenario

Another experiment was conducted to test the effectiveness of the neural network on a real world scenario: it was created a succession of chords using 2 different commercial audio libraries, which featured 2 different pianos compared to the one it was trained with. The digital pianos were both considered to have "high quality" sound (compared to the original piano), which made the comparison fair. A short MIDI file of a pseudo-random chord progression was fed into the two libraries. To make the results more visible, only notes of one octave were played, without loss of generality. Here is the midi representation of the chords that were played:

Figure 2: MIDI of the chord progression.



The WAV files were then generated from the MIDI performance and fed as an input to the series of neural networks. The WAV files had to be segmented first, using an Hann window. In the following tables, there are the predictions for every chord (from left to right) in terms of present notes and absent notes; "0" means that the note is absent, "1" means that the note is present. Every column represents a chord. Here are the results:

Piano library 1:

Table 2: Piano library #1

B3	0	0	0	0	1	0	0	0
A#3	0	0	0	1	0	0	0	0
A3	0	1	1	0	0	1	0	0
G#3	0	0	0	0	0	0	1	0
G3	1	0	0	0	0	1	1	1
F#3	0	0	0	1	0	0	0	0
F3	0	0	1	0	1	0	1	0
E3	1	1	0	0	0	1	1	1
D#3	0	0	0	0	0	1	0	0
D3	0	0	0	0	0	0	0	0
C#3	0	0	0	0	1	0	0	0
C3	1	1	1	1	1	1	0	0

- **Precision**: 100%
- **Recall**: 84.38%

Piano library 2:

Table 3: Piano library #2

B3	0	0	0	0	1	0	0	0
A#3	0	0	0	1	0	0	0	0
A3	0	1	1	0	0	1	0	0
G#3	0	0	0	0	0	0	0	0
G3	1	1	1	0	0	1	1	1
F#3	0	0	0	1	0	0	0	0
F3	1	1	1	0	1	0	1	0
E3	1	1	0	0	0	0	0	0
D#3	0	0	0	0	0	1	0	0
D3	0	0	0	0	0	0	0	0
C#3	0	0	0	0	0	0	0	0
C3	0	0	0	0	0	0	0	0

- **Precision**: 78.95%
- **Recall**: 50%

Sound-wise, the first piano library is very similar to the original piano library use to train the neural networks. On the other hand, the second piano library differs quite a bit in sound, which explains the worse results overall. These outcomes bring some interesting arguments forward:

- It might be necessary to expand the training data to different piano libraries, since the slight difference between piano sounds has a negative impact on results; this is a direct consequence of the creation of sample libraries, where the virtual instrument is a collection of recordings of different pianos.
- To further strengthen the robustness of the algorithm, it might be necessary to also include slight detuning or other artifacts (such as reverb/early reflections); a piano recording is not always "dry" (without any reverb) or perfectly tuned to an $A = 440Hz$.
- Lastly, the results would improve a lot if there were "chopped up" audio segments in the training data; since the segmentation is not always clean, and since some notes often carry over from one chord to another, having sounds without "attack" (i.e. that began before the audio clip) in the training data would make the network even more robust.

5 Conclusions

In this project, there was an attempt to solve two problems in the automatic classification of sounds: instrument detection and chord detection. Even though the objectives of these two problems were seemingly different, the underlining structure of the solutions shared a lot of similarities. The framework was entirely based on frequency-based features extraction and neural networks. The approach of identifying chords through individual notes was a deviation from the standard "chord labeling" algorithms, which can potentially be developed to support more complex datasets. The two implementations can also be combined to feature pitch and chord detection of different instruments, as the algorithm can work for instruments outside the piano, with little modifications.

The objectives set for the scope of this assignment were met, and the results were overall satisfying. Some improvements can be made on both the problems that were formulated in this project, and the general direction was laid out in the relevant chapters.

Future developments could feature a more robust neural network for the detection of chords for different pianos (different sounds and effects), multi-instrument detection, chord estimation from automatically segmented audio files, and finally a user interface that would be able to communicate the results in a user-friendly manner.

References

[1] R. Beato. Perfect pitch: The world's greatest ear!! URL https://www.youtube.com/watch?v=t3Cb1qwCUvI.

[2] J. Deng and Y.-K. Kwok. Automatic chord estimation on seventhsbass chord vocabulary using deep neural network. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016. doi: 10.1109/icassp.2016.7471677.

[3] B. Toghiani-Rizi and M. Windmark. Musical instrument recognition using their distinctive characteristics in artificial neural networks. *CoRR*, abs/1705.04971, 2017. URL http://arxiv.org/abs/1705.04971.

[4] X. Zhou and A. Lerch. Chord detection using deep learning. In M. Müller and F. Wiering, editors, *ISMIR*, pages 52–58, 2015. ISBN 978-84-606-8853-2. URL http://dblp.uni-trier.de/db/conf/ismir/ismir2015.html#ZhouL15.