POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

AI Powered Pick-up

Laurea Magistrale in Music and acoustic Engineering

Author: Emanuele Voltolini

Advisor: Prof. Fabio Antonacci

Co-advisor: Sebastian Gonzalez

Academic year: 2021-2022

1. Introduction

Nowadays, digital techniques in music industry and production are becoming more and more popular alongside the classic analog ones or in some cases directly replacing them. One of the most innovative research area in this sense is represented by the analog audio effect modeling, which exploits DSP (Digital Signal Processing) or modern deep learning techniques to create digital models of analog amplifiers, pedals and other audio effects.

Two of the main non deep learning based methodologies are represented by WDFs (Wave Digital Filters) and block-oriented Wiener-Hammerstein model. The first are a particular kind of digital filters based on physical modeling principles; the second is a parametric model adaptable to many distortion effects. Both methods present difficulty in handling multiple nonlinearities and they are often demanding from a computational point of view.

Trying to overcome the difficulty in modeling multiple nonlinearities, a deep learning based method is introduced by Zhang's [5] work. Although the authors reported clearly audible differences between the resulting model and the target device. An improvement on perceptual results is brought by Wright [4]. The paper shows how good results can be achievable with RNN (Recurrent Neural Network) and WaveNet models, explaining also the possibility in terms of real-time applications. Steinmetz and Reiss [3] carry on Wright's work applying a new model based on TCNs (Tempoal Convolutional Networks) on a more complex audio effects (dynamic range compressor). It is shown how this new architecture is more efficient from a computational effort making the model particularly suitable for real-time implementations.

In this thesis work we apply deep learning to acoustic guitar pickup - microphone black-box sound modeling. Since this is a new field of research, we selected one of the most used deep neural network in black-box sound modeling, a RNN with an LSTM (Long Short Term Memory) unit [1]. In order to test this model, we create a training dataset composed by pairs of microphone and pick-up acoustic guitar recordings. Furthermore, we studied the loss function implemented by Wright [4] to see if it could fit our task. We evaluated the best performing model in terms of ESR (Error to Signal Ratio) both in time and frequency domain. Finally, we presented a comparison between different models based on the ESR values taking into account also the author's perceptual evaluation.

2. Model and Methods

In order to accomplish our task, a deep learning based approach is adopted. First implemented in [4], the chosen neural network model is the RNN (Recurrent Neural Network). This network receives the piezo-electric pickup recording as input and the cardiod microphone ones as target.

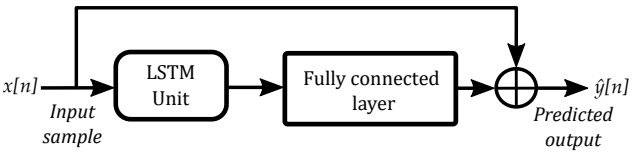


Figure 1: RNN model. The input x_n goes first to the LSTM unit and then into a fully connected layer. The output of the latter is summed with the initial x_n to obtain the predicted output $\hat{y}_n$

Fig 1 shows the entire architecture. The RNN network is composed by an LSTM (Long-Short Term Memory) unit, followed by a Fully Connected layer. At each time step n a single input sample $x[n]$ is fed into the LSTM unit. The output of the latter goes into the Fully Connected layer to produce a single output which is summed with the initial input $x[n]$ to obtain the final predicted output $\hat{y}[n]$. By doing so, the network just learns to predict the difference between input and output samples.
The state of the LSTM unit is made of two vectors: the *hidden state* h and the *cell state* c. For each time step n, $x[n]$, $h[n-1]$ and $c[n-1]$ are used to calculate the LSTM's output $h[n]$ and $c[n]$.
The size of both the hidden and cell states is equal to the LSTM's hyperparameter *hidden size*. Increasing the hidden size generally results in the model being more accurate. However it increases the number of learnable parameters in the network, as well as the processing power required to run it. The PyTorch machine learning library was used to implement the whole RNN model.

2.1. Data acquisition process

The diagram of the data acquisition process is shown in Figure 2. An acoustic guitar is simultaneously recorded from its piezo-electric pickup and a professional microphone placed in front of

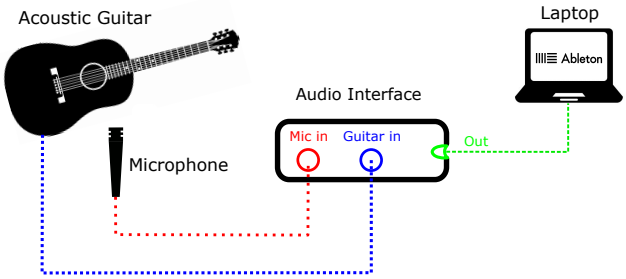
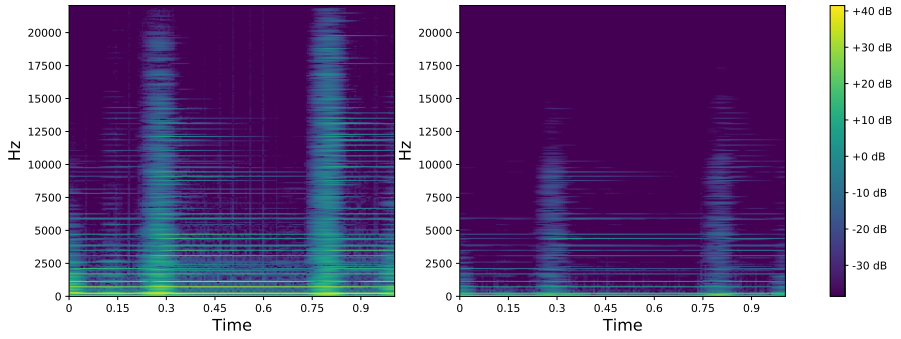


Figure 2: Data acquisition process. The signal is recorded simultaneously from the acoustic guitar pick-up and a microphone using an audio interface. Its output is connected via USB to a laptop. Ableton Live® was used to record and export the audio.

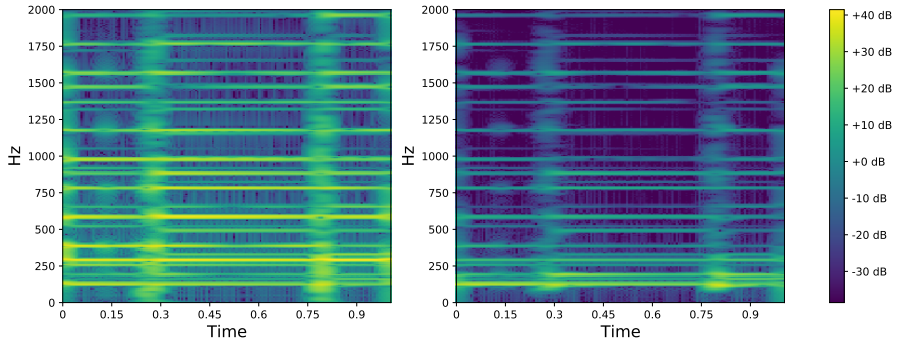
it. In order to do that, an audio interface is used. A jack cable connects the guitar pickup to the first channel, while an XLR cable carries the microphone signal to the second channel. The audio interface is connected to a laptop using an USB cable. To record the multi-track we relied on Ableton Live®. This software allow us to record and edit multiple audio tracks at the same time. Finally, all the recorded tracks (pickup and microphone version) have been exported in mono audio files. The instrument has been recorded in a small room with no particular acoustic treatments.

2.2. Acquisition parameters and data description

The guitar and microphone signals are acquired at 44.1kHz. We obtain two mono audio tracks for each recording. All the audio lengths are between 1 and 2 minutes.
Figure 4 shows the time domain representation of the two obtained signals. The curves exhibit a different trend. The blue one referring to the pick-up acquired signal is richer in high frequencies components than the orange one referring to the SM57. This characteristic is reflected in time domain by the abrupt changes of the the blue curve. We can observe it also in the spectrograms of the two signals in figure 3.
As a matter of fact, we can see that the energy of the microphone one is concentrated more on frequencies lower than 1 kHz. The pick-up spectrograms instead present a lot of energy also in the mid-high frequencies. These differences can



(a)



(b)

Figure 3: Spectrograms of the pick-up and microphone signals. The image refers to 1 second of audio. For both images we have the guitar pick-up signal on the left and microphone signal on the right. (a) refers to the entire audio bandwidth, (b) refers to a low-mids frequency band (0 Hz - 2 kHz).

be clearly heard in the two recordings. The microphone audio is characterized by a darker tone with respect to the piezo pick-up one.

2.3. Pre-processing and training

Before entering in the network loop, the training data are pre-processed. Since the majority of the energy of the target is concentrated around lower frequencies, a low pass filter is applied to the input training signal. We use Butterworth digital low pass filter.

In order to have balance between training and validation data, each audio is split into 2 parts. By doing so we increase the number and variability in the selection process. 70% of these segments are assigned to the training data and 20% to the validation. Once the splitting is defined, the training and validation arrays are obtained concatenating the respective audio segments. Furthermore, as test data we used an audio which is a mixture of guitar playing styles. In order to be processed by the neural network, the dimensions of the three data arrays are modified and they are converted into tensors. The training array is split into overlapping batches of $segment\_length = 7$ second. Furthermore, we use an *overlap* parameter to control the percentage of overlapping between two consecutive segments.

The model is trained using Adam optimizer. The RNN is trained for 1000 epochs. The validation loss is calculated every three epochs. If the validation loss does not improve within 200 epochs, the training stops. The starting learning rate value $LR_i = 0.01$ is decreased dynamically by a multiplicative factor $k = 0.7$ every 3 epochs the validation loss is not improving. Furthermore, to avoid local minima the learning rate is reset to $0.8LR_i$ at epoch 500 and $0.1LR_i$ at epoch 700.

2.4. Loss function

We studied the loss function used in Wright's work [4]. For a signal of length N the loss function ε is the result of the sum of two contributions:

$$\varepsilon = \varepsilon_{ESR} + \varepsilon_{DC} \tag{1}$$

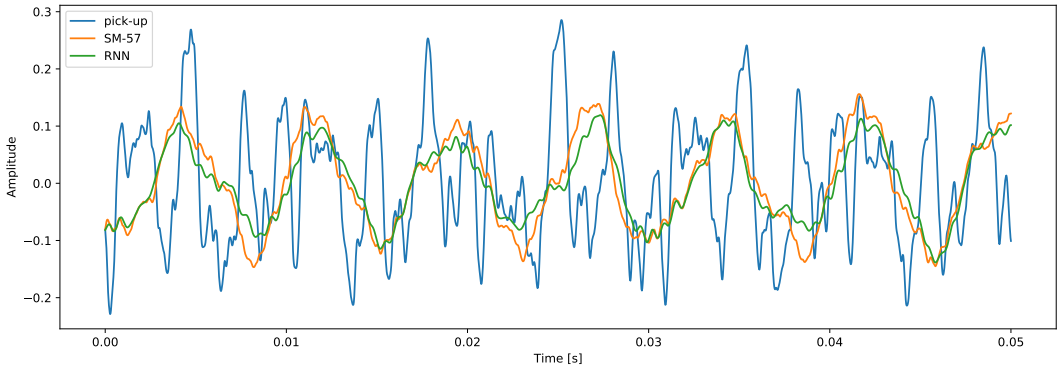


Figure 4: Comparison between recorded signals and the network output in time domain. The image shows 0.05 seconds of audio. The image refers to a guitar strumming audio. The blue one refers to the signal recorded with the pick-up, the orange is obtained using the SM57 microphone and the green one is the RNN output.

The first component is the error to signal ratio (ESR) with respect to the training data, calculated as:

$$\varepsilon_{ESR} = \frac{\sum_{n=0}^{N-1} |y_p[n] - \hat{y_p}[n]|^2}{\sum_{n=0}^{N-1} |y_p[n]|^2} \tag{2}$$

Where $y_p[n]$ and $\hat{y}_p[n]$ are respectively the target signal and the output of the neural network at sample n. For both signals a low-passed A-Weighting filter has been applied. Its purpose is to emphasise the frequencies in the loss function, based on their perceived loudness. The denominator in the ESR normalises the loss with regards to the target signal energy. As a matter of fact it prevents the loss function to be dominated by the segments of signal with higher energy.
The second additional member ε_{DC} of the equation (1) represents the difference in DC offset between the target and neural network output:

$$\varepsilon_{DC} = \frac{|\frac{1}{N}\sum_{n=0}^{N-1}(y[n] - \hat{y}[n])|^2}{\frac{1}{N}\sum_{n=0}^{N-1}|y[n]|^2} \tag{3}$$

The target $y[n]$ and the network's output $\hat{y}[n]$ have not been filtered.
We calculated the two components of the loss function for some of the audio segments and we have seen that the contribution of the DC components is always close to 0, on average in the order of 10^{-4}. Therefore we decide to neglect the DC component and use only the ESR for the loss evaluation.
In order to understand if the network is able to learn using the ε_{ESR} loss function, we built a fake signal which tries to emulate what the network should do with the original signal. The *fake signal* is obtained as a sum of the input signal, at which a smoothing algorithm is applied, with 4 sinusoids at different frequencies. We observed that the fake signal has lower ESR values with respect to the raw input, suggesting that the loss function we chose could perform well on the task we aim to.

3. Results

We analysed the output of the best performing model in terms of ESR (Error to Signal Ratio) score. The analysis of the network's audio output is carried out both in time and frequency domain.
For what concern the time domain, the network processing has a significant problem: since the final audio is obtained by an overlap and add process, discontinuities between two consecutive segments cause a click noise. The reason of this issue regards the time step t as $t = (1 - overlap) * segment\_len$ which has a jump. In order to solve this problem, we applied triangular windowing where each segment is multiplied by a triangular window during the overlap and add process. Each triangular window has a unitary amplitude and it share the same length and overlapping factor of the audio chunks.
Figure 4 shows results in time domain of an audio characterized by strumming chords. As it can be seen the RNN signal follows quite well

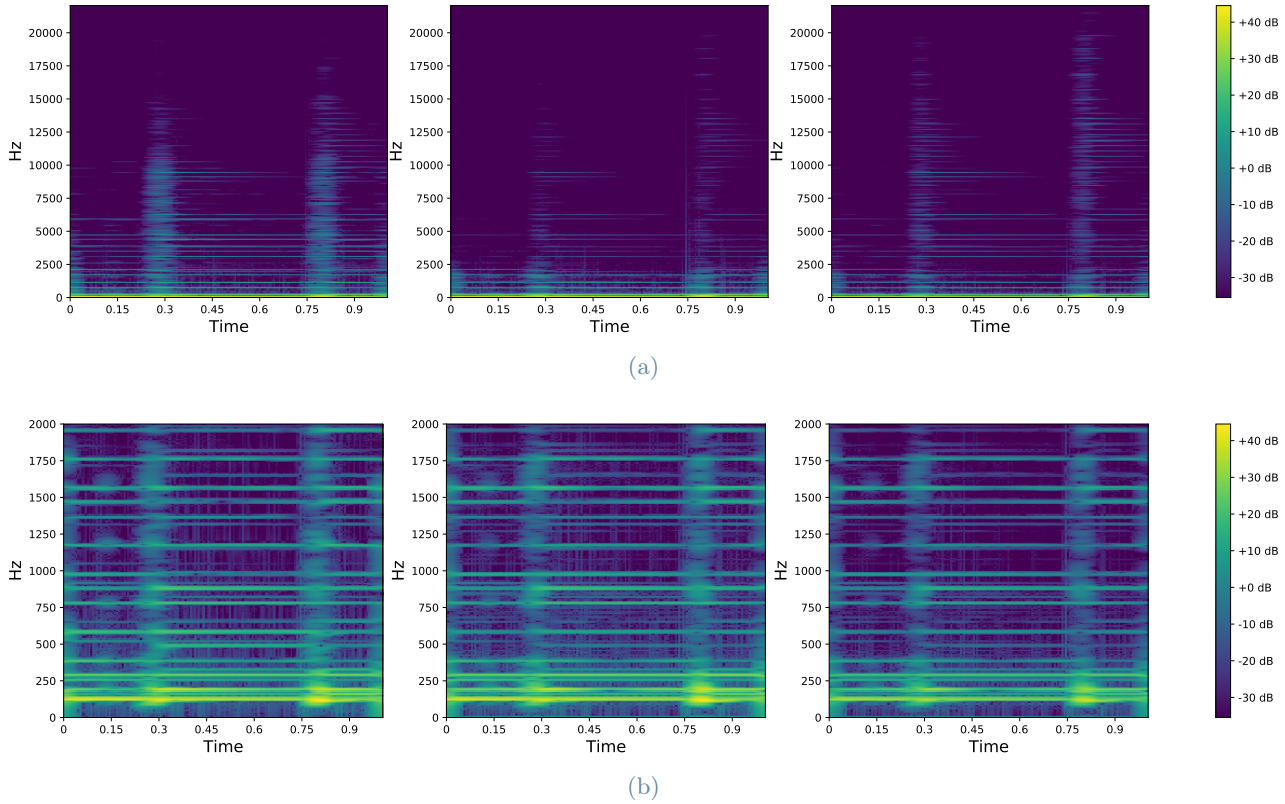


Figure 5: Spectrograms of microphone, "best" network output and model "sum" output signals. The image refers to 1 second of audio. For both images we have the microphone signal on the left, the model "best" output in the center and the model "sum" output on the right. (a) refers to the entire audio bandwidth, (b) refers to a low-mids frequency band (0 Hz - 2 kHz).

the microphone one, however it struggles during abrupt changes (which correspond to high frequency components), even if the overall trend is respected.

Figure 5 shows at its center the spectrograms of the network's output. On one hand, figure 5a demonstrates the fact that those frequencies higher than 3kHz are attenuated in the RNN output with respect to the target, reflecting the results obtained in the time domain. The main reasons could be that the model is not complex enough to capture all the high frequencies characteristics of the target signal, or we do not have enough data for the training. On the other hand, figure 5b shows the network performance in terms of low and mid range frequencies (80Hz - 2kHz). It is appreciable how these components are well represented in the model output.

In order to overcome high frequencies limitations of the model previously described, we propose a solution based on the consideration of two different network's outputs instead of relying just on a single model. The two summed components are respectively the output of the best performing model in terms of ESR loss and the output of a simpler model (hidden size of 16 instead of 96), which slightly performs in a worse manner in terms of loss but it presents a bigger number of higher frequency components. As a result the new gained complete spectrum (Fig. 1 - right) has more energy in the upper range of frequencies with respect to the best performing model one (Fig. 1 - center). This change in the frequency domain is appreciable also from a perceptual point of view; as a matter of fact the addition of the high frequencies makes all the overtones of the acoustic steel strings audible.

We finally performed a perceptual analysis (conducted by the author) among different models' output. We found that the best performing networks in terms of ESR, are not necessarily the best from a perceptual point of view. As we described before, in the spectrogram of the best model the frequencies higher than 3 kHz are attenuated, this turns out in a darker tone of the audio with respect to the target one. On the

other hand simpler models maintain the high frequency components giving a better perceptual impression, even with a lower ESR score.

4. Conclusions

This thesis aimed to black-box modeling acoustic guitar pickup - microphone sound using deep leaning model based on a recurrent neural network (RNN) with a long-short term memory (LSTM) unit. The network has shown its ability of following the trend of the target microphone signal in time domain, given as input the pick-up one. However, the model is not able to properly capture the high frequencies components of the spectrum, which are attenuated for frequencies greater than 3 kHz. Moreover, we proposed a solution based on the combination of two different models which seems to produce appreciable auditory results, comparable with the original microphone recordings.

As we describe in Section 3, a possible problem we can highlight corresponds to the fact that the best model output does not correspond to the best perceptual audio results. Since the perceptual analysis is conducted by the author of this thesis, we suggest as a future development to verify the perceptual analysis with a proper test such as webMUSHRA (Multiple Stimuli with Hidden Reference and Anchor) [2].

We also know there is room for improvements regarding the used data acquisition process, which has been done in a small room with no special acoustic treatment. A possible solution we suggest is to redo the data acquisition process in a controlled environment such as an anechoic chamber. As a consequence all room's spectral components contributions are eliminated.

Moreover, we find that all the proposed models reach a plateau in the training process. Because the complexity of the task, a possible cause could be the lack of data. Therefore we propose as a future development to expand the dataset with a new set of acoustic guitar recordings using the same equipment. In this sense, a further step could be to try modeling different type of microphones, defining different training dataset for each one of them.

Following Wright [4] and Steinmetz's works [3], we suggest to do a real-time implementation of the model to see its computational effort, which could be compared with two or more other neural network models.

To the best of our knowledge, there are no previous researches on this thesis' task. Therefore, our main contribution to the state of the art is given by the demonstration that the acoustic guitar pickup - microphone sound modeling can be done using deep neural networks. To conclude, we believe this work represents a first step in this newer field of research.

5. Acknowledgements

I would like to give my thanks to my supervisor Professor Fabio Antonacci and my co-advisor Sebastian Gonzalez for what I have learnt and for their support and patience.

References

[1] DE Rumelhart, GE Hinton, and RJ Williams. Learning internal representations by error propagation. in the pdp research group (eds.), parallel distributed processing: Explorations in the microstructure of cognition (vol. 1, pp. 318-362), 1986.

[2] Michael Schoeffler, Sarah Bartoschek, Fabian-Robert Stöter, Marlene Roess, Susanne Westphal, Bernd Edler, and Jürgen Herre. webmushra—a comprehensive framework for web-based listening tests. *Journal of Open Research Software*, 6(1), 2018.

[3] Christian J Steinmetz and Joshua D Reiss. Efficient neural networks for real-time analog audio effect modeling. *arXiv preprint arXiv:2102.06200*, 2021.

[4] Alec Wright, Eero-Pekka Damskägg, Lauri Juvela, and Vesa Välimäki. Real-time guitar amplifier emulation with deep learning. *Applied Sciences*, 10(3):766, 2020.

[5] Zhichen Zhang, Edward Olbrych, Joseph Bruchalski, Thomas J McCormick, and David L Livingston. A vacuum-tube guitar amplifier model using long/short-term memory networks. In *SoutheastCon 2018*, pages 1–5. IEEE, 2018.