Neural Modelling of Audio Effects

Alec Wright
Neural Modelling of Audio Effects

Alec Wright

A doctoral thesis completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of School of Electrical Engineering, at a public examination held at the lecture hall Jeti of the school on 15 December 2023 at 12:00.

Aalto University
School of Electrical Engineering
Department of Information and Communications Engineering
Audio Signal Processing
Supervising professor
Prof. Vesa Välimäki, Aalto University, Finland

Thesis advisor
Prof. Vesa Välimäki, Aalto University, Finland

Preliminary examiners
Prof. Berrak Sisman, Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, USA
Dr. Marco Martinez, Sony AI, Japan

Opponent
Prof. Josh Reiss, Centre for Digital Music, Queen Mary University of London, UK

Aalto University publication series
DOCTORAL THESES 217/2023

© 2023 Alec Wright

ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Unigrafia Oy
Helsinki 2023

Finland
Abstract

Alec Wright

**Name of the doctoral thesis**
Neural Modelling of Audio Effects

**Publisher**
School of Electrical Engineering

**Unit**
Department of Information and Communications Engineering

**Series**
Aalto University publication series DOCTORAL THESES 217/2023

**Field of research**
Acoustics and Audio Signal Processing

**Manuscript submitted** 5 June 2023  
**Date of the defence** 15 December 2023

**Permission for public defence granted (date)** 17 November 2023  
**Language** English

**Abstract**

Neural networks and other machine learning based approaches to audio effects processing have become increasingly popular in recent years. This thesis focuses on the design and training of neural network architectures for the emulation of specific analog audio devices from data. The digital emulation of analog audio devices is commonly known as virtual analog, and popular effects processing devices for virtual analog modelling include guitar amplifiers, distortion pedals, time-varying effects, and compressors.

Whilst analytical methods based on circuit analysis are capable of producing realistic, efficient and accurate models of devices, these approaches are limited by the fact that creating a model of a specific device is time-consuming and requires expert knowledge. In contrast, neural network based methods allow for greater automation in the modelling process, and can be applied relatively easily to a range of devices as long as sufficient data is available. This thesis proposes a number of neural network based methods for audio effects modelling, and shows that they achieve excellent perceptual emulation quality. The proposed models include convolutional, recurrent and differentiable digital signal processing based architectures. There is a focus on models with low computational cost and low latency, such that they are suitable for real-time processing as part of a music production workflow. Methods for modelling Low-Frequency Oscillator (LFO) modulated time-varying effects, compressors, guitar amplifiers and distortions pedals are proposed.

In addition to the neural network architectures themselves, this thesis also provides practical details and methods for training the models. This includes the proposal and validation of a novel perceptually motivated pre-emphasis filter, used to model non-linear audio effects processing. Additionally a pruning method is applied and shown to achieve significant reduction in model size and inference cost for guitar amplifier and distortion effects modelling.

Finally, this thesis presents a novel method for the task of modelling non-linear audio effects processing when paired training data is unavailable. This allows for complex non-linear effects processing to be emulated from recordings, whilst requiring no knowledge of the specific devices used to create the recording.

**Keywords**
Audio Effects Processing, Deep Learning, Neural Networks, Nonlinear Systems, Machine Learning

**ISBN (printed)** 978-952-64-1575-8  

**ISSN (printed)** 1799-4934  
**ISSN (pdf)** 1799-4942

**Location of publisher** Helsinki  
**Location of printing** Helsinki  
**Year** 2023

**Pages** 135  
Abstract

Neural networks and other machine learning based approaches to audio effects processing have become increasingly popular in recent years. This thesis focuses on the design and training of neural network architectures for the emulation of specific analog audio devices from data. The digital emulation of analog audio devices is commonly known as virtual analog, and popular effects processing devices for virtual analog modelling include guitar amplifiers, distortion pedals, time-varying effects, and compressors.

Whilst analytical methods based on circuit analysis are capable of producing realistic, efficient and accurate models of devices, these approaches are limited by the fact that creating a model of a specific device is time-consuming and requires expert knowledge. In contrast, neural network based methods allow for greater automation in the modelling process, and can be applied relatively easily to a range of devices as long as sufficient data is available.

This thesis proposes a number of neural network based methods for audio effects modelling, and shows that they achieve excellent perceptual emulation quality. The proposed models include convolutional, recurrent and differentiable digital signal processing based architectures. There is a focus on models with low computational cost and low latency, such that they are suitable for real-time processing as part of a music production workflow. Methods for modelling Low-Frequency Oscillator (LFO) modulated time-varying effects, compressors, guitar amplifiers and distortion pedals are proposed.

In addition to the neural network architectures themselves, this thesis also provides practical details and methods for training the models. This includes the proposal and validation of a novel perceptually motivated pre-emphasis filter, used to model non-linear audio effects processing. Additionally a pruning method is applied and shown to achieve significant reduction in model size and inference cost for guitar amplifier and distortion effects modelling.

Finally, this thesis presents a novel method for the task of modelling non-linear audio effects processing when paired training data is unavailable. This allows for complex non-linear effects processing to be emulated from recordings, whilst requiring no knowledge of the specific devices used to create the recording.
Preface

I started my doctoral studies at the Aalto Acoustics Lab in 2019. There are many many people who made my years living and working in Finland an unforgettable and wonderful experience, and I am grateful to all of them.

Firstly, I want to thank my supervisor Professor Vesa Välimäki, for taking a chance on me, and supporting me and my research. I also want to thank all my co-authors, who made invaluable contributions to producing this thesis. I am also extremely grateful to have worked at the Aalto Acoustics Lab, which is made great by all the passionate and supportive researchers who work there, so thanks to all of you. I would also like to thank the pre-examiners, Dr. Marco Martínez and Prof. Berrak Sisman, for reviewing the thesis and providing valuable feedback, and Prof. Josh Reiss for agreeing to act as my opponent.

Outside of work, many people contributed to making living in Finland a truly wonderful experience and I would like to thank them as well. I was fortunate to have many excellent housemates in Finland, so if we lived together at Avaruuskatu or Ruusulantaku, thank you for making me feel at home, and for all the kitchen chats. I also want to thank all those that invited or accompanied me on trips to mökki during my time in Finland.

I’m lucky to have many friends across the UK and abroad, so thanks to all of you for your friendship, company, and conversation over the years. I am also grateful to my family who supported (and visited me!) throughout my time in Finland. Finally, I’m eternally grateful to Catie for loving and believing in me.

Edinburgh, November 19, 2023,

Alec Wright
Contents

Preface 3
Contents 5
List of Publications 7
Author's Contribution 9
List of Figures 11
Abbreviations 13

1. Introduction 15

2. Neural Network Architectures for Audio Effects Processing 19
   2.1 Convolutional Neural Networks .............................. 19
      2.1.1 One-Dimensional Convolutional Layers .............. 20
      2.1.2 Convolutional Architectures for Audio Effects Modelling ........................................ 22
   2.2 Recurrent Neural Networks .................................. 23
      2.2.1 Recurrent Neural Network Architectures for Audio Effects Modelling ............................. 26
   2.3 Differentiable Digital Signal Processing Models ........... 27

3. Model Training 29
   3.1 Loss Functions for Supervised Training ...................... 29
      3.1.1 Parameter Loss Functions ........................... 29
      3.1.2 Audio Domain Loss Functions ....................... 30
   3.2 Neural Network Pruning .................................... 33
   3.3 Audio Effect Emulation with Unpaired Data .................. 34

4. Summary of Main Results 35

5. Conclusion 39
List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.


Author’s Contribution

Publication I: “Real-time black-box modelling with recurrent neural networks”

The author planned the study in collaboration with the co-authors. The author implemented the model architecture and training and carried out all the model training reported in the study. The author collected additional training data that was used for the study with the assistance of the second author. The author wrote the majority of paper, with the assistance of the co-authors. The implementation and results described in Section 5 was carried out by the second author.

Publication II: “Real-time guitar amplifier emulation with deep learning”

The author planned the study in collaboration with the co-authors. The author implemented the model architectures and training and carried out all the model training reported in the study. The author wrote the majority of paper, with the exception of Section 2, which was written by the co-authors. The author planned, designed, and conducted the listening test.

Publication III: “Neural modeling of phaser and flanging effects”

The author planned the study in collaboration with the co-author. The author collected all the data used in the study and implemented the described methods, models and training. The author wrote the paper with help from the co-author.
Author's Contribution

Publication IV: “Grey-box modelling of dynamic range compression”

The author planned the study in collaboration with the co-author. The author implemented the described models and training. The author wrote the paper with the help of the co-author.

Publication V: “Perceptual loss function for neural modeling of audio systems”

The author planned the study in collaboration with the co-author. The author implemented the methods described and carried out all the model training reported in the study. The author wrote the majority of paper, with the help of the co-author. The author planned, designed, and conducted the listening test.

Publication VI: “Pruning deep neural network models of guitar distortion effects”

The author planned the study in collaboration with the co-authors. The author performed analysis of the resulting models and produced figure 7. The author planned, designed, and conducted the listening test which validated the efficacy of the proposed method.

Publication VII: “Adversarial guitar amplifier modelling with unpaired data”

The author planned the study in collaboration with the co-authors. The author implemented the models and carried out all training reported in the study. The author planned, designed, and conducted the listening tests. The author wrote the paper with help from the co-authors.
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The distinctive sound of vacuum-tube (left) [1] distortion in amplifiers, and flanging achieved using tape machines (right) [2], are both examples of analog audio effects which have been emulated using solid-state circuits.</td>
<td>15</td>
</tr>
<tr>
<td>2.1</td>
<td>A non-causal one-dimensional convolutional layer of kernel size 3 being applied to an input of length 5, visualised (left) without channels, with each colour representing one of the shared weight groups, and (right) with 3 input and 3 output channels.</td>
<td>20</td>
</tr>
<tr>
<td>2.2</td>
<td>A one-dimensional convolutional layer of kernel size 3 being applied to an input of length 5, visualised (left) without zero-padding, (middle) with zero-padding on both sides, resulting in a layer that preserves input size but is non-causal, and (right) with zero padding applied to the left side only, resulting in a causal layer that preserves the input size.</td>
<td>21</td>
</tr>
<tr>
<td>2.3</td>
<td>Comparison between two stacks of one-dimensional convolutional layers, with exponentially increasing dilation (left) and no dilation (right), showing that whilst both layers have the same number of parameters, the use of dilation greatly increases the receptive field.</td>
<td>21</td>
</tr>
<tr>
<td>2.4</td>
<td>A single-input single-output one-dimensional convolutional layer and the equivalent FIR filter.</td>
<td>22</td>
</tr>
<tr>
<td>2.5</td>
<td>Recurrent neural network model with explicit recurrence (left) and in unfolded form (right). $x$, $s$ and $h$ represents the RNN input, state and output, respectively, and $\theta$ represents the learnable parameters of the network.</td>
<td>23</td>
</tr>
</tbody>
</table>
2.6 Alternative RNN model with explicit recurrence (left) and in unfolded form (right). $x$ and $y$ represents the RNN input and output, respectively, and $\theta$ represents the learnable parameters of the network. In this case, the output of the RNN is fed directly to the next layer. 

3.1 Pre-emphasis filter proposed in Publication V, based on the A-Weighting curve with additional low-pass filtering applied.
Abbreviations

**BPTT**  Back-Propagation Through Time  
**DSP**  Digital Signal Processing  
**DDSP**  Differentiable Digital Signal Processing  
**GRU**  Gated Recurrent Unit  
**LFO**  Low-Frequency Oscillator  
**LTI**  Linear Time Invariant  
**LSTM**  Long Short-Term Memory  
**LTH**  Lottery Ticket Hypothesis  
**MLP**  Multilayer Perceptron  
**TBPTT**  Truncated Back-Propagation Through Time  
**RNN**  Recurrent Neural Network  
**VA**  Virtual Analog
1. Introduction

Audio effects modelling is a field that seeks to recreate the sound of popular audio effects. Generally the purpose is to replace one audio effects system with an alternative that is cheaper, more reliable, lighter or easier to use.

Early examples include vacuum tube, or thermionic valve (see Fig. 1.1), guitar amplifiers. The invention of transistors quickly led to transistorised guitar amplifiers [3], which have often been criticised as sounding inferior to vacuum-tube guitar amplifiers [4, 5, 6, 7, 8]. This in turn led to a number of attempts to emulate the sound of vacuum-tube amplifiers using solid-state circuits [9, 10, 11, 12].

Another example is for the flanging effect that has been popular since the 1960s. This was originally achieved by using two reel-to-reel tape machines [13], making it expensive and impractical. Attempts to emulate the tape flanging effect include using cascaded all-pass filters (known as phasing, now an audio effect in its own right [14]), or later, by implementing an analog delay line using a bucket-brigade device [15].

Figure 1.1. The distinctive sound of vacuum-tube (left) [1] distortion in amplifiers, and flanging achieved using tape machines (right) [2], are both examples of analog audio effects which have been emulated using solid-state circuits.

Since the 1990s, with the rapidly increasing availability of computational power, there has been extensive academic and commercial interest in creating digital algorithms to emulate analog musical devices. This allows
for audio effects processing algorithms to be downloaded as software, without requiring any hardware beyond a computer and an audio interface. Emulating the characteristics of analog musical equipment digitally is an active research field often referred to as Virtual Analog (VA) [16]. Digital models of analog audio effects include the aforementioned vacuum tube amplifiers, for both guitar [17, 18, 19, 20, 21] and general audio amplifiers [22, 23], as well as flanging [24], phasers [25, 26, 27], and bucket brigade devices [28, 29]. Other popular analog audio effects for digital modelling include compressors [30] and limiters [31], distortion effects [32, 33, 34], synthesiser filters [35, 36, 37], synthesiser oscillators [38, 39, 40, 41], tape delay [42] and reverb [43, 44]. Much of the research also focuses on more general concepts and frameworks that are useful for modelling physical systems, such as work on dispersion [45], finite difference simulation [46] and circuit modelling [47, 48, 49, 50, 51].

There are many challenges when it comes to VA modelling. Many audio effects processing devices that are popular modelling targets are non-linear time-invariant (LTI) systems, which can make them challenging to emulate accurately. For example, guitar amplifiers, which are non-linear systems with memory, or phaser and flanging effects, which are time-varying. In addition to the complex behaviour of the systems themselves, there are a number of artefacts and issues that can arise when discretising continuous systems, such as frequency warping [52], aliasing [53], or stability [32].

A further challenge for VA models comes from the restricted computational budget and real-time low-latency run-time requirements imposed by the use of the resulting algorithm within a digital audio workstation (DAW). Musicians are generally very sensitive to latency, especially when playing an instrument in an interactive setting. Additionally, it is not unusual for DAW users to run hundreds of audio plugins simultaneously in a single project. This means that the computational cost of each plugin must be very low. This restriction in the computational cost of the algorithm is made more challenging by the relatively high sample rates that are used in music production settings, which are generally at least 44.1 kHz and often higher [54].

Finally, due to the complex and non-linear nature of human hearing, evaluating VA models is no simple task. When designing audio effects models, there are often trade-offs that must be made between model accuracy and computational cost. Furthermore, determining what level of accuracy is sufficient, in terms of objective metrics, generally requires tedious and time-consuming perceptual evaluation via listening tests. This also presents challenges as it can be difficult to know which aspects of an audio effects design will be perceptually relevant.

Approaches to creating digital emulations generally exist on a spectrum from white-box to black-box [23, 55]. White-box techniques are a model driven approach, in which a model describing the target system’s behaviour...
is constructed. These include circuit modelling techniques such as wave-
digital filters [56] and physical modelling schemes such as finite difference
methods [46]. Black-box techniques represent a more data driven approach,
in which data is used to fit a model to emulate the target systems input-
output behaviour. Examples of this are the swept sine method for room
impulse response measurement [57] and Volterra series methods [58].
Modelling approaches that incorporate a partial theoretical structure as
well as data are known as grey-box models [21, 27].

This thesis focuses on the development, design and training of neural
networks for modelling audio effects. Neural networks are powerful non-
linear function approximators, which transform their inputs over a number
of successive layers, with each generally consisting of multiplications and
summations with learnable parameters followed by non-linear activation
functions. Neural networks are a black-box method, and recent years have
seen deep neural networks applied to achieve state of the art results at a
wide variety of tasks.

Neural networks are data-driven models, which are optimised to achieve
a desired task using data. The advantage of this approach is that a single
model architecture can be applied to model a variety of devices, provided
that data is available. This drastically reduces the amount of engineering
effort required to model a given device [59]. A major downside associated
with neural networks are that they often are very large, with modern
networks often containing billions of parameters. For the real-time and
low processing-power constraints found in music production, this is clearly
problematic. Whilst audio effects processing models are generally much
smaller than those used for more mainstream tasks such as language mod-
elling or audio synthesis, careful consideration and design is still required
to produce networks which are suitable for use in a music production envi-
ronment. Recent years have seen neural network models applied to a range
of audio related tasks, with examples including instrument synthesis [60],
audio effects modelling [61, 62], automatic mixing [63, 64] and perceptual
audio comparison [65].

This thesis presents a variety of research results relating to audio ef-
fects modelling using neural networks. The main focus is on producing
lightweight models with low inference cost that sound perceptually very
similar to the target audio effect. The thesis consists of this introductory
document, as well as seven peer-reviewed publications. Publications I-IV
are concerned with modelling popular analog audio effects, with a focus on
run-time efficiency and lightweight design. These effects are guitar am-
plifiers and distortion pedals (Publications I and II), phaser and flanging
effects (Publication III), and dynamic range compressors (Publication IV).
Publications V and VI present methods that can be applied during model
training, either to improve the perceptual performance (Publication V) or
to reduce inference cost (Publication VI).
Finally, Publication VII proposes a novel scheme for modelling audio effects accurately using unpaired data. This final publication is significant because it allows for the blind emulation of audio effects, in the case where the specific effects processing devices to be emulated are unknown, something which is not possible using existing analytical modelling approaches.

The introductory portion of this thesis is structured as follows. Chapter 2 provides an overview of the different audio effects processing architectures that were used during this thesis, relating them to traditional Digital Signal Processing (DSP) structures, and providing practical details for applying them to audio effects modelling. Chapter 3 presents details on training methods for neural modelling of audio effects. Chapter 4 summarises the main results of each publication, and finally, Chapter 5 presents the conclusions of the thesis.
2. Neural Network Architectures for Audio Effects Processing

There has recently been an explosion of research activity applying neural network and deep learning techniques to audio effects processing and modelling tasks. This research activity includes this thesis and the publications it contains, as well as other recent theses [66, 61] and publications from industry [67, 68, 69, 59, 70]. Neural networks traditionally refer to specific architectures based on a combination of linear layers and memoryless non-linear activation functions.

Neural network architectures bear many similarities to traditional digital signal processing architectures, which is something that will be explored further in this section. In light of this, the term ‘neural’ is sometimes also applied to describe a modelling approach based on traditional digital signal processing structures, and trained using Stochastic Gradient Descent (SGD) [68, 71]. One key aspect common to neural approaches is that the models consist of a chain of differentiable signal processing blocks.

This section describes three broad categories of neural network layers commonly used for audio effects modelling tasks, Convolutional layers, Recurrent layers, and Differentiable Digital Signal Processing (DDSP) layers\(^1\). This section also provides examples of their use. This is included in Publications I-VII, as well as in other published research.

2.1 Convolutional Neural Networks

Convolutional neural network layers apply filter kernels over inputs and are widely used for image and audio tasks. Both one-dimensional and two-dimensional convolutional layers are commonly used for audio tasks. One-dimensional convolutional layers can be applied to a raw audio waveform, in which case the convolution is applied across the time axis of the input. Two-dimensional convolutional layers are well suited to image

\(^1\)Multilayer Perceptrons (MLPs) are less suited to audio effects processing, and to sequence modelling tasks generally, as they have a fixed input/output size and don’t explicitly utilise temporal/spatial information
Neural Network Architectures for Audio Effects Processing

Figure 2.1. A non-causal one-dimensional convolutional layer of kernel size 3 being applied to an input of length 5, visualised (left) without channels, with each colour representing one of the shared weight groups, and (right) with 3 input and 3 output channels.

processing, so are often applied to image-like representations of audio, such as spectrograms. In this case, the convolution is applied across the temporal frames and frequency bins of the spectrogram. One-dimensional convolutional layers can also be applied to spectrograms, by treating the frequency bins as input channels of the convolutional layer, an architecture that was used in the discriminator model from Publication VII. Application of a convolutional layer can be thought of as applying a fully connected layer at each spatial or temporal location of the input signal. This allows convolutional layers to process audio signals of arbitrary length.

2.1.1 One-Dimensional Convolutional Layers

In a one-dimensional convolutional layer, filters kernels are applied to the layer input, as shown in Fig. 2.1. Whilst the name implies convolution, neural network libraries usually calculate the cross-correlation between filter kernels and input channels. However, as the convolution can be calculated simply by flipping the filter kernel relative to the input and then finding the cross-correlation, this distinction is unimportant in this context. Convolutional layers typically have multiple input and output channels. In this case, a separate filter kernel is learned for each possible pair of input and output channel, as shown in Fig. 2.1 (right).

Zero-Padding and Causality

It can be seen in Fig. 2.2 that the length of the convolutional layer output is shorter than the input. This is a feature of convolutional layers which is desirable for many applications, however for audio effects processing it is important that the model output is the same length as the input. Zero-padding can be used to ensure that the output size matches the input size. Zero padding is often applied equally to both sides of the input, however, as shown in Fig. 2.2 (middle), this results in a non-causal system, as the output sample requires future samples to calculate. Whilst this is unimportant for some applications, for real-time audio effects processing this introduces additionally latency. By applying zero padding to the left side of the input only, as shown in Fig. 2.2 (right), the system remains causal and the input size is preserved. A convolutional network for sequence
Neural Network Architectures for Audio Effects Processing

**Figure 2.2.** A one-dimensional convolutional layer of kernel size 3 being applied to an input of length 5, visualised (left) without zero-padding, (middle) with zero-padding on both sides, resulting in a layer that preserves input size but is non-causal, and (right) with zero padding applied to the left side only, resulting in a causal layer that preserves the input size.

**Figure 2.3.** Comparison between two stacks of one-dimensional convolutional layers, with exponentially increasing dilation (left) and no dilation (right), showing that whilst both layers have the same number of parameters, the use of dilation greatly increases the receptive field.

processing that is causal and preserves input length is often referred to as a Temporal Convolutional Network (TCN).

**Receptive Field and Dilation**
The receptive field of a convolutional layer or CNN refers to the region of inputs that affect a given output. Each output of a convolutional layer is influenced by a limited number of inputs, depending on the layer’s kernel size. A CNN model has a fixed receptive field, outside of which it has no memory. When modelling audio effects this is significant as any proposed CNN model should have a receptive field similar to the memory of the target effect. Dilation is achieved by adding zeros between filter kernel weights. This is a popular method for increasing the model receptive field without adding additional learnable parameters. By stacking convolutional layers whilst exponentially increasing dilation, a large receptive field can be achieved with relatively few layers, as shown in Fig. 2.3. This idea was popularised for audio by the WaveNet model [72].

**Signal Processing Comparisons**
Convolutional layers are multiple-input multiple-output (MIMO) FIR filters. A traditional single-input single-output (SISO) FIR filter can be implemented as a special case of the one-dimensional convolutional layer, as shown in Fig. 2.4, by excluding the bias term and setting the number of input and output channels to 1. This is a convenient and efficient way of implementing FIR filters using a deep learning library. This approach was used, for example, to implement the pre-emphasis filters in Publication V.
Figure 2.4. A single-input single-output one-dimensional convolutional layer and the equivalent FIR filter.

In neural networks, linear layers such as convolutions are generally followed by non-linear activation functions. Previous work has also noted the similarity of this structure to Wiener-Hammerstein models [73]. Kuznetsov et al. [71] used FIR filters implemented via one-dimensional convolutional layers as a baseline model, combining this with a memoryless multilayer perceptron (MLP) to implement a Wiener-Hammerstein model.

One key difference between FIR filters and one-dimensional convolutional layers is the inclusion of the bias term in the output of the layer. The bias term can, however, also be considered as a parameter of the activation function that follows the convolutional layer.

2.1.2 Convolutional Architectures for Audio Effects Modelling

A number of CNNs for audio effects modelling have been proposed. A feedforward variant of the WaveNet architecture, based on a speech denoising model [74], has been used for modelling distortion effects such as guitar amplifiers [75] and distortion pedals [73]. This architecture is based on multiple stacks of 1D convolutional layers, with the dilation of the layers increasing exponentially with depth. This architecture was used as a baseline model in Publications I and II, in addition to being used as the generator model in Publication VII.

Models based on stacks of dilated convolutional layers have been used for a variety of effects processing tasks. This includes virtual analog modelling of specific devices, such as in [62], where rapidly increasing dilation factors were used to further increase receptive field, as well as in [76], where time-varying feature modulation using an RNN was used to increase model memory. Other uses include as a transformation network in an automatic multitrack mixing system [64] and as a neural proxy model within an audio effects style transfer framework [77].

Another class of audio effects modelling architectures uses one-dimensional convolutional layers to map the input audio to a latent space, where a transformation is applied before it is used to re-synthesise the target audio [61]. The transformation applied to the latent space can be a feedfoward
neural network, as was originally proposed to model a range of non-linear audio effects [78], or an RNN, which was proposed for modelling time-varying effects. This type of model and its variants have been successfully applied to model a wide variety of audio effects [79, 61].

2.2 Recurrent Neural Networks

Recurrent neural networks (RNNs) are another class of neural network that are popular for sequential processing tasks. The key difference between CNN based models and RNN based models are that RNNs have a state which is updated as each input is processed, as shown in Fig. 2.5. As an input sequence is processed the recurrence, shown in the left hand side of Fig. 2.5, is unfolded into a computational graph, shown on the right hand side of Fig. 2.5. This unfolding results in a computational graph with length depending on the length of the processed sequence.

A consequence of this is that training and inference must be carried out recursively, whereas with CNNs the model is feedforward, allowing processing to be fully parallelised. For audio effects modelling, this generally results in slower training times for RNN based models, in comparison to feedforward models. Conversely, however, the inclusion of recursion in the model results in fewer model parameters being required and faster inference time, as was shown in Publications I and II. The inclusion of a recurrence mechanism means that the network state can in theory retain information over arbitrarily long sequences, in contrast to CNNs which have a maximum memory length, or receptive field, determined by the specific architecture design choices.

Later work proposed an improved implementation of convolutional layers for real-time audio processing which reduced the computational cost considerably [80], however, models based on RNNs were still shown to be more efficient, especially for smaller model sizes. In practice, the real-time performance of these models is influenced strongly by the implementation as well as the processor architecture.
Gated Recurrent Neural Networks

When training RNN models, gradients are propagated through the unfolded computational graph, in a process called back-propagation through time (BPTT). One well known issue that can arise during BPTT are exploding or vanishing gradients [81]. A popular solution to the problem is to use gated RNN architectures, which allow gradients to flow through time without exploding or vanishing [82]. Two widely used examples of such architectures are Long Short-Term Memory (LSTM) [83] and Gated Recurrent Units (GRUs) [84], both of which were used for guitar amplifier modelling in Publications I and II.

Output and State Recurrence

The recurrence depicted in Fig. 2.5 produces an output, $y$, at each time-step, and has recurrent connections between subsequent hidden states, and as such can be referred to as having state recurrence. This type of recurrent network was used in Publications I-IV. An alternative type of recurrent network architecture is depicted in Fig. 2.6. In this case, the only recurrence is connecting the output, or prediction, of the network to the hidden layer, and as such can be described as having output recurrence. The output recurrence formulation of RNN is less powerful than that which is shown in Fig. 2.5, as it is only able to transmit data indirectly via its output $y$ [82].

To accurately model a system using an RNN based on output recurrence, the outputs, and thus the training data, must capture all the information about the past that is required for accurate prediction. Whilst in many circumstances this is quite impractical or impossible, in the case of circuit modelling this information is readily available as the target device can have its circuit states measured directly. This was the approach proposed by Parker et al. [67], in which an RNN based on output recurrence was trained using data captured directly from the target circuit using electrical probes.

---

**Figure 2.6.** Alternative RNN model with explicit recurrence (left) and in unfolded form (right). $x$ and $y$ represent the RNN input and output, respectively, and $\theta$ represents the learnable parameters of the network. In this case, the output of the RNN is fed directly to the next layer.
**Teacher Forcing**

The output recurrence scheme allows for teacher forcing to be implemented. This is where, as the recurrence is unfolded, instead of using the network's predicted previous output, the ground truth is used instead. For the state recurrence scheme this is generally not possible, as the states are a latent representation learned during training and as such no ground truth values exist. One advantage of teacher forcing is that it allows training without BPTT, as the previous output for each time-step is contained in the training data.

The major drawback to teacher forcing is that at inference time, where the ground truth information is generally not available, the network outputs that are fed back into the network will potentially be quite different to the inputs seen by the network. This can cause a performance gap between training and inference. This problem is often referred to as exposure bias, and was studied in the context of VA modelling of circuits by Peussa et al. [85, 69].

**State-Initialisation and Truncated Backpropagation Through Time**

As can be seen in Figures 2.5 and 2.6, when starting to process an input sequence \( x \), an initial value for the recurrent input to the model is required. For output recurrent models, this information is readily available from the data during training time. For hidden recurrent models, the initial hidden state is usually either learned [86] or set to a vector of zeros [87, 88]. Proper hidden state initialisation has been shown to be very important for modelling audio systems such as loudspeakers [89].

When training models of audio effects this presents a problem. Because the training data generally consists of a very long sequence of input and target audio samples, this is usually divided into short segments, with each segment being treated as a single training example. This results in segments of target signal which contain information from previous inputs, which are not provided to the model during training. The ideal state initialisation will depend heavily on the previous input samples, which makes choosing an appropriate state initialisation challenging. One solution is to allow the RNN to process some samples of input, such that its state incorporates information from recent inputs, and use this as the initial hidden state.

This introduces some additional cost to training, as each time the state is initialised part of the input sequence must be processed without carrying out any parameter updates. In order to reduce the number of times state initialisation is carried out, training can be carried out using Truncated Backpropagation Through Time (TBPTT). The idea behind TBPTT is to divide a longer sequence into many shorter sub-sequences. If these are naively treated as individual training examples, however, then the model...
will not be able to utilise temporal dependencies longer than the sub-sequence length. In TBPTT, the full sequence is still processed in order, however, BPTT is carried out over each sub-sequence. Crucially, the hidden states are retained after TBPTT is carried out, and used as the initial states when processing the next sub-sequence. This allows for the hidden state information to propagate over the full sequence length, whilst still carrying out many parameter updates.

2.2.1 Recurrent Neural Network Architectures for Audio Effects Modelling

To the author’s knowledge, the earliest published use of an RNN for audio effects modelling was from Covert and Livingston in 2013 [90], which proposed using an RNN based implementation of a non-linear AutoRegressive eXogenous network to model a vacuum-tube guitar amplifier. The proposed model used output recurrence (see Sec. 2.2), and was trained using teacher forcing. The authors reported good performance on the training data, but poor performance on the test set, likely due to exposure bias.

In 2018, Schmitz proposed a single layer LSTM model [91] for guitar amplifier emulation, as well as a number of recurrent and hybrid architectures [92, 66]. Whilst these models produced perceptually accurate results, the modelling process described involves splitting the input audio into overlapping segments, and individually processing each overlapping segment to produce a single output sample. This limits the length of time-dependencies that the model can learn to exploit, and additionally results in a significant amount of additional computational cost when compared to processing the whole input as one continuous time sequence. Whilst the described method can be parallelised on a GPU to achieve real-time processing speeds, performance on a CPU was not evaluated.

In 2019, the model proposed in Publication I uses a single LSTM or GRU layer, follower by a linear layer, as well as proposing a training scheme based on TBPTT. The model was validated by modelling a guitar amplifier and a distortion pedal. Whilst the architecture is similar that proposed by Schmitz [91], the model proposed in Publication I outputs a predicted output for each time-step of input. This greatly increases inference speed, allowing for real-time processing on a CPU, which was also demonstrated and measured. Publication II provided further validation of the model, comparing it with a previously proposed convolutional model [73] and conducting formal listening tests which demonstrated that in some cases the sound produced by the models is indistinguishable from the reference device. This architecture has also been applied to modelling time-varying effects in Publication III and compressors in Publication IV.

Other recurrent neural networks proposed for audio effects modelling are the State-Trajectory Network proposed by Parker et al. [67], where
additional state information was measured from the target circuit and used as the target data. This results in an output recurrent neural network. Peussa et al. [69] proposed using a GRU model with circuit state information from the target circuit as an additional objective. They proposed using a learnable affine mapping between the GRU states and the circuit states, resulting in a state recurrent neural network.

Whilst it seems intuitive that models trained using more circuit state information would perform better, the results presented by Peussa et al. [69] don’t show this conclusively, instead indicating that which method performs best is influenced by factors such as model size, training hyper-parameters and choice of target device.

### 2.3 Differentiable Digital Signal Processing Models

The final category of neural architectures for audio effects modelling is based on utilising traditional DSP structures. This idea is often referred to as Differentiable Digital Signal Processing (DDSP) and was largely popularised by Engel et al. in 2020 [60]. The use of DDSP components is intended to incorporate a strong inductive bias into the models, ideally producing models with high perceptual emulation quality, more efficient inference and interpretable behaviour.

This idea has been exploited in the audio effects modelling literature too. In the previous sections, it can already be seen that convolutional layers and recurrent networks bear a striking similarity to FIR and non-linear state-space models, respectively. In addition to this, a number of models have made use of learnable IIR filters and memoryless non-linear mappings.

Specifically models that incorporate biquadratic filters have been a popular choice. In 2020, Kuznetsov et al. [71] noted the similarity between RNNs and IIR filters, and proposed directly optimising the parameters of various IIR filters. Also in 2020, Nercessian [93] proposed a machine learning approach to carrying out parametric equalizer matching, based on differentiable biquadratic filters.

In 2021 Nercessian et al. [68] proposed an extension to this approach, by learning IIR filters directly and during training applying them in the frequency domain via their frequency response. This greatly improves training speed, in comparison to applying the filters directly as a recursion. In Publication IV, a learnable IIR filter in the form of a one-pole filter was used as part of a model for a Dynamic Range Compressor (DRC). This further extended the training-time frequency domain filtering method, using an input buffer to allow training to be carried out over short time windows as is desired for TBPTT, whilst still learning a one-pole filter with a long time-constant.
3. Model Training

The previous section described different forms of neural network architectures for audio effects modelling. This section describes different optimisation methods that can be used for the creation and refinement of audio effects processing algorithms.

3.1 Loss Functions for Supervised Training

The majority of literature on neural audio effects modelling uses supervised training methods. This follows logically as often it is the behaviour of a physical device that is being modelled, allowing for the collection of perfectly aligned input-output audio data pairs. There are a great number of possible loss functions for supervised audio effects modelling. This section describes the most frequently used loss functions for audio effects modelling.

3.1.1 Parameter Loss Functions

Parameter loss functions directly compare predicted parameters of a model to target parameters. This is distinct to loss functions that compare audio directly. This can applied when attempting to find a model inverse, that is, a function that takes the output of a pre-defined model (i.e. a synthesiser, parametric EQ or other audio effect) and attempts to predict the parameters used to create that output. This type of system was proposed, for example, by Gabrielli et al. in 2017 [94], where the authors trained a neural network to analyse audio generated by a physical model of a pipe organ, with the objective of predicting the parameters used to synthesise it. Other uses include the task of predicting the parameters of an audio effect that has been applied to some audio [95].

One issue that arises when using parameter loss functions is that the parameter space doesn’t always correspond well perceptually with the audio output. For example, some parameters might have little to no
impact on the resulting sound, especially in certain areas of the parameter space. This makes the model inversion task ill-posed as for parts of the parameter space, there is no unique audio output, and therefore no unique solution. This also means that the perceptual relevance of parameter prediction errors is hard to determine, as some parameters will have a greater perceptual impact than others, as noted in [94].

Parameter loss functions have also been used to optimise parametric equalisers (EQs) [96, 97, 93]. In [96, 97] the authors use a neural network to predict EQ band gains, with the aim of matching a target frequency response at the center frequencies of the EQ bands. Nercessian [93] proposed an EQ matching system that uses a parameter loss in addition to calculating the loss directly on the overall frequency response of the EQ, reporting that including both the frequency response and parameter losses improved performance in comparison to just using the parameter loss. This represents a special case because the system being optimised, a parametric EQ, is a linear time-invariant (LTI) system, allowing it to be fully described by its frequency response. For audio effects that include non-linear or time varying components, the behaviour of the system cannot be described completely using a frequency response. In this case, model optimisation generally relies on minimising the difference between audio examples that have been processed by the target system and the model. This requires the use of audio domain loss functions, which are described in the following section.

### 3.1.2 Audio Domain Loss Functions

The behaviour of non-linear dynamic systems, such as guitar amplifiers, can't be described using linear system theory. This means that modelling these devices either requires white-box methods, that involve direct simulation of the inner workings of the device, or black-box methods, where measurements of the device are used to optimise the parameters of a model. The latter requires the use of audio domain loss functions, which compare outputs of the target device with outputs of the model being optimised.

#### Time Domain Losses

Time-domain loss functions directly compare raw audio waveforms on a sample-by-sample basis. This requires that the signals being compared are well aligned in the time-domain. Most time-domain loss functions simply take the absolute error, L1, or the squared error, L2, over the audio signals being compared.

Another variant is the log-cosh loss function proposed by Chen et al. in 2018 [98] and applied to audio effects modelling by Hawley et al. [99] in 2019. The log-cosh loss function is similar to the absolute error but includes additional rounding near zero. This is thought to improve training.
Pre-Emphasis Filtering

It is well known that human perception of loudness is dependent on frequency [100, 101]. Particularly, for a constant sound pressure level, low frequencies are perceived more quietly in comparison to frequencies in the most sensitive area around 3000 Hz. One pre-processing technique that is commonly used for speech processing is pre-emphasis filtering [102]. This was initially applied to neural network modelling of guitar amplifiers by Dämskägg et al. [75], where a first-order high-pass filter was applied to both model output and target data before computing the loss. This is intended to adjust how much different frequencies contribute to the overall loss, effectively weighting the loss toward higher frequencies. This was extended and validated in Publication V, where we proposed a new pre-emphasis filter, shown in Fig. 3.1, based on a low-passed A-weighting curve, which is a frequency weighting curve intended to compensate for the relative loudness that different frequencies are perceived at. Our results validated the use of pre-emphasis filters for modelling distortion effects, and showed that the use of our proposed A-weighting pre-emphasis filter can produce a statistically significant improvement in perceptual performance whilst having no impact of inference cost.

Normalisation

Loss functions can also include a normalisation term, such as the Error-to-Signal Ratio (ESR) used by Dämskägg et al. [75] or the normalised MSE loss used by Parker et al. [67]. In this case, the error is divided by the energy of the target signal. Without the normalisation term, the loss becomes very small in areas where the target signal level is low, resulting in less accurate modelling performance for low input levels. Normalisation is also used in the Spectral Convergence frequency domain loss function.
Model Training

[103], where the loss term is normalized by the Euclidean length of the STFT magnitudes of the target signal.

Time-Alignment

One drawback of time-domain loss functions is that waveform similarity doesn’t always correspond well with perceptual similarity: it is possible that similar waveforms sound quite different, and that similar sounding waveforms are very different when compared sample-wise. Simple examples of this are when comparing a waveform with an inverted or slightly delayed version of itself. These two waveforms will sound identical, however, when compared sample-by-sample will appear very different.

This means that a successful model will have to learn to replicate any delay, dispersive or not, between the input and target audio found in the dataset. A non-dispersive delay (i.e time lag applied to the whole signal) is quite trivial to model using a convolutional architecture with sufficiently long receptive field. Recurrent networks are known to have difficulties remembering information exactly from past inputs [104]. So whilst it is possible for RNN models to learn to delay inputs, this increases the difficulty of the modelling task, and if possible, should be avoided to reduce the required model capacity and training time.

For models incorporating or exclusively using traditional DSP blocks, the specific architecture places constraints on the delay that can be learned by the model. In Nercessian et al. [68] the authors model a distortion effect using many blocks consisting of IIR filters and memoryless non-linearities. To allow the model to delay and also invert the input signal, they propose using a learnable delay line at the input to the model.

Another solution to this problem was proposed by Martínez Ramírez et al. [105], which used a modified delay-invariant time-domain loss function, where time delay between the loss function inputs is found and compensated for via auto correlation. Additionally, the loss is calculated for the time-aligned signals with and without phase inversion being applied to the model output, with the smaller value being used as the loss. This means that the model won’t be penalised in the loss function for failing to invert or apply a delay to the input signal.

Finally, a loss function based on the short-term energy of the output and target signals also relaxes the requirement for strict time-alignment between signals. This has been used previously for training speech synthe- sisers [106, 107]. The loss function compares energy in the time domain over short signal segments, at multiple resolutions. This was applied to digital modelling of analog compressors in Publication IV, to allow training of a DDSP-based model which lacked the capability to apply phase delay to the input signal.
**Frequency-Domain Loss Functions**

The use of a frequency domain loss function also reduces the time-alignment required between target and model outputs. These loss functions compare time-frequency representations of audio, typically a spectrogram or mel-spectrogram. The phase information is generally discarded before the loss is calculated. This reduces the loss functions sensitivity to differences in phase. Frequency domain loss functions have been used in combination with a time domain loss function for audio effects processing tasks such as equaliser matching [108] and reverb [109] or compressor modelling [99]. The use of a single time-frequency representation results in a trade-off between time and frequency resolution. To avoid having to pick a single set of analysis parameters for the time-frequency representation, it is common to use multiple time-frequency representations simultaneously in the loss function [110, 111]. This is generally referred to as a multi-resolution loss function, and this has also been used in training of audio effects processing models, for example for automated mixing task [63, 64], and in conjunction with a time domain loss, modelling analog compressors [62].

**3.2 Neural Network Pruning**

Neural network pruning refers to the task of reducing a neural network’s size through the removal of parameters - typically the weights and biases of the network. A great deal of work has been published generally on the topic, however, in the field of audio effects processing there exists very little published research. The general idea underpinning neural network pruning is that these models are often vastly overparameterised, and, often 90% or even more of their parameters can be removed whilst maintaining model performance, or even in some cases improving it. A variety of techniques for pruning exist [112]. One such method that has been successfully applied for audio models is based on the ‘Lottery Ticket Hypothesis’ (LTH), which postulates that dense feedforward networks contain subnetworks, or ‘winning tickets’, which can perform similarly to the original densely connected network when trained in isolation. This has been successfully applied to various generative audio models [113].

Audio effects processing models tend to be many orders of magnitude smaller than the kinds of neural networks used for image processing or audio synthesis, however model compression is still desirable for audio effects processing as the models are usually deployed in real-time scenarios which require minimal latency. Furthermore, modern music production workflows often involve the simultaneous use of many audio effects and synthesis algorithms. Finally, the models are invariably intended to be operated at at least 44.1 kHz sample rate, which means that even for a small model, the number of operations per second required during inference
Model Training

can quickly become quite large. In Publication VI we applied LTH based pruning methods to RNN based models of guitar amplifiers and distortion effects, removing the majority of the model parameters, reducing model inference cost, all whilst having minimal impact on model performance.

3.3 Audio Effect Emulation with Unpaired Data

A relatively recent application of machine learning for audio effects processing is based around the task of emulating the audio effects processing applied to a given reference signal, when no corresponding unprocessed version of the signal is available.

Most neural network based audio effect emulation models rely on a dataset of paired audio examples, consisting of unprocessed input signal and the target signal which is a processed version of the input. This allows for the network to directly learn the transformation applied to the input signal by optimising the network parameters such that its output is sufficiently similar to the target.

For some tasks, it is desirable to emulate the effects processing applied to a signal without requiring access to the original input. This can be described as blind estimation of audio effects applied to a signal, where it is not required to know what effects were actually applied to the signal. The objective being a system that can apply some audio effects processing to an input signal, such that it has a similar style to a reference. This might be done because the unprocessed input signal is unavailable, or because the desired signal processing algorithm should account for differences in musical content between the input to the network and the provided reference.

This problem formulation has been applied to achieve style transfer of audio effects by Steinmetz et al. [77], where a self-supervised scheme was used to train a neural network to control audio effects processors. Another method was proposed by Koo et al. [70] to apply style transfer of mixing style, utilising source separation to obtain separated stems to use as references. Both of these methods use pre-defined audio effects processing blocks to train the model in a self-supervised manner, using either differentiable implementations of the audio effects themselves, or neural proxy models trained to emulate them. In Publication VII we propose an approach to a related task, accurately emulating the non-linear audio effects processing applied to a guitar. This can also be viewed as a form of audio effects processing style transfer, where a model is optimised to create an accurate black-box emulation of a single effects processing chain.
4. Summary of Main Results

Publication I - “Real-Time Black-Box Modelling With Recurrent Neural Networks”

Sound examples available at:

Publication I presents a detailed description and validation of two recurrent neural network models, using either a GRU or LSTM unit, as applied to the task of modelling a guitar amplifier and a distortion effects pedal. Whilst it is not the first study to suggest applying recurrent neural networks to this task [114, 92, 66], the research demonstrates that a simple single-layer RNN is sufficient for very accurate modelling, provided that the training is handled properly.

The paper provides a description of the training procedure, including the loss functions used to train the model and the TBPTT process, which is essential for training the model successfully. A real-time C++ implementation is also presented, along with real-time performance metrics. The proposed model is compared to an existing baseline convolutional neural network based model [75] in terms of loss achieved on the test set, as well as real-time performance. The proposed model is shown to be comparable or slightly better than the baseline in terms of the test loss, whilst achieving a significant reduction in computational cost at runtime.

Publication II - “Real-Time Guitar Amplifier Emulation with Deep Learning”

Sound examples available at:
http://research.spa.aalto.fi/publications/papers/applsci-deep/

Publication II is an extended version of a previously published conference paper [73], with the main contribution being further comparison and validation of convolutional and recurrent neural network models of guitar
amplifiers. The paper presents experiments carried out to determine the most effective architectural choices, balancing model accuracy, measured using a time-domain loss function, and computational cost at inference time. Further experiments were carried out to validate and compare the two model architectures, with objective and subjective evaluations of the models being presented. The subjective evaluation consisted of a Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) listening test [115], which showed that in all cases the neural network models produce emulations of Excellent quality in terms of the MUSHRA grading scale, and that in some cases the neural network models are indistinguishable from the target device.

**Publication III - “Neural Modeling of Phaser and Flanging Effects”**

**Sound examples available at:**
http://research.spa.aalto.fi/publications/papers/jaes-tvfx/

Publication III is an extended version of a previously published conference paper by the same authors [116], which introduces and validates a method for modelling two popular time-varying effects, a phaser and a flanger.

The paper provides background information on how phasing and flanging effects work, as well introducing a novel neural network based modelling procedure. The proposed modelling procedure requires access to the Low-Frequency Oscillator (LFO) signal used to modulate the effect during training, and as such methods for accurately measuring the LFO signal from an analog effects unit are proposed. Validation of the LFO measurement scheme was carried out using simulated data. The overall approach was then validated by applying it to emulate two analog effects pedals.

**Publication IV - “Grey-Box Modelling of Dynamic Range Compression”**

**Sound examples available at:**
http://research.spa.aalto.fi/publications/papers/dafx22-DDRC/

Publication IV presents a machine learning based approach to modelling an analog Dynamic Range Compressor (DRC), with a focus on creating a lightweight and efficient model with interpretable user controls. The proposed method uses both DDSP layers, static compression curves and one-pole filters, as well as an RNN, optimising all model parameters jointly via stochastic gradient descent to achieve the modelling task. The method is validated by fitting it to a digital compressor algorithm. A model of a target analog compressor unit is created using the method, and this is then compared to a black-box baseline model. The results show the proposed
model achieves similar accuracy in terms of the test loss, with a much smaller computational cost and more interpretable user facing controls.

**Publication V - “Perceptual Loss Function for Neural Modeling of Audio Systems”**

Publication V explores the use of pre-emphasis filtering during the training of neural network models of audio systems. A number of pre-emphasis filters were evaluated, with a listening test being carried out to validate their perceptual effect on the trained models. The results indicate that the use of pre-emphasis filters during training can improve perceptual similarity of the resulting models to the target devices, whilst having no impact on processing speed at inference time.

**Publication VI - “Pruning Deep Neural Network Models of Guitar Distortion Effects”**

Publication VI applies pruning methods from the deep learning literature to the recurrent neural network models proposed in Publication I, as applied to the task of guitar amplifier and distortion effects modelling. The goal of the pruning process is to reduce the computational cost of running the model at inference time. The results show that for all tested devices, the majority of the model parameters can be pruned without reducing model accuracy in terms of test loss, provided retraining is carried out after pruning. It was found that the choice of target device has a considerable impact on the amount of pruning that can be applied, with some devices allowing for more than 99% of the parameters to be pruned whilst improving the test loss achieved, whilst others allowing only 95% of parameters to be pruned before model performance deteriorated. Whilst this difference sounds relatively small, due to the details of the C++ implementation this has a significant effect on the inference time speedup gained by the pruning process. Furthermore, the results of a listening test are presented, showing that the pruning either has no effect on the perceptual similarity between model and target device outputs, or actually improves it slightly.

**Publication VII - “Adversarial Guitar Amplifier Modelling with Unpaired Data”**

**Sound examples available at:**
https://ljuvela.github.io/adversarial-amp-modeling-demo/

Publication VII presents a novel method for modelling guitar amplifiers when paired data is unavailable, making supervised training as carried out
in Publications I-VI impossible. The use case of the method is intended to be situations where only a target guitar tone is available, and the original input guitar signal is not, for example, when the material available is only a processed recording of a guitar performance. The proposed framework is based on an adversarial approach. Experiments were carried out to validate the approach, including listening tests which demonstrate the perceptual similarity between the trained models and the target effects processing. This publication is, to our knowledge, the first published work demonstrating perceptually accurate emulation of a non-linear audio effect, without the requirement of specific knowledge of the target device (i.e. a circuit schematic) or access to the device for collection of paired training data.
In this thesis, we have discussed the application of neural network models and training techniques to the creation of audio effects processing algorithms. This includes different architecture types, based on convolutional, recurrent and DDSP layers, or combinations of these, as well as different methods that can be applied during training.

In Publications I-IV these architectures were applied to create models of various popular analog audio effects. Publications I and II presented real-time capable models of guitar amplifiers and distortion effects. Publication III proposed a novel training data collection scheme which allows for the modelling of the popular time-varying phaser and flanging effects, and Publication IV proposed a grey-box model for dynamic range compression which utilised DDSP layers to greatly reduce model inference cost.

Publication V proposed a new perceptually motivated pre-emphasis filter to be used when training audio models. The study showed that use of the proposed filter improves perceptual performance, while having no impact of inference cost. Publication VI applied pruning techniques to distortion effects models, showing that most parameters can be removed from existing RNN models, and in some cases this can even improve model performance, whilst also reducing model inference cost. Finally Publication VII proposed a scheme for modelling a non-linear audio effects processing chain blindly using unpaired data.

This thesis joins the rapidly growing body of research that uses neural networks and other machine learning techniques to derive new models for audio effects processing. These techniques are of academic and commercial interest, as evidenced by the growing body of research published by academic institutions [66, 61, 77, 76] and industry [67, 117, 68, 59, 70].

There are still limitations associated with the methods discussed in this thesis. When creating fully parameterised models of analog devices, data is required that captures the device behaviour over many possible parameter permutations. Capturing this data is tedious and impractical if done manually. Recent work has proposed solving this issue by either automating the data collection process [59], or by utilising white-box models to reduce
the amount of data required to capture the device behaviour [117, 118]. Furthermore, there is evidence that the presented models don’t properly reproduce the perceptual characteristics when modelling devices with long temporal dependencies, such as compressors [62]. Recent work has proposed architectures that address this shortcoming [119, 76, 120], however formal listening tests are required to perceptually evaluate the results.

There are other neural network architectures and techniques that can be applied to audio effects modelling. Diffusion probabilistic models have recently been applied to emulate audio effects with stochastic elements, such as gramophones [121] or tape machines [122].

Future research on this topic will naturally involve further development of VA modelling using neural network models. This includes further improving the perceptual accuracy of models whilst reducing training time and inference cost, and developing more methods to reduce the amount of training data required to model devices with large parameter spaces. Developing a better understanding of the different objective metrics used for evaluating audio effects processing models would also be of great benefit. This could be used for developing new loss functions for neural network training, or to improve understanding of existing ones, as current practices vary significantly and there is little work comparing perceptual performance achieved by the use of different loss functions.

Beyond the world of VA modelling, there are many exciting applications for machine learning and audio effects processing. These include developing new audio effects which are not possible using analog equipment, for example instrument-to-instrument transfer, as well as intelligent systems designed to emulate the behaviour of audio engineers and musicians. Research into these areas will help digital audio algorithms achieve new goals, greatly expanding the toolset available to musicians and allowing them to go beyond what is possible with analog systems.
References


References


References


