Damskägg, Eero-Pekka; Juvela, Lauri; Välimäki, Vesa

Real-Time Modeling of Audio Distortion Circuits with Deep Learning

Published in:
Proceedings of the 16th Sound & Music Computing Conference SMC 2019

Published: 01/01/2019

Document Version
Publisher's PDF, also known as Version of record

Real-Time Modeling of Audio Distortion Circuits with Deep Learning

Eero-Pekka Damskägg, Lauri Juvela, and Vesa Välimäki
Acoustics Lab, Department of Signal Processing and Acoustics
Aalto University, Espoo, Finland
eero-pekka.damskagg@aalto.fi

ABSTRACT

This paper studies deep neural networks for modeling of audio distortion circuits. The selected approach is black-box modeling, which estimates model parameters based on the measured input and output signals of the device. Three common audio distortion pedals having a different circuit configuration and their own distinctive sonic character have been chosen for this study: the Ibanez Tube Screamer, the Boss DS-1, and the Electro-Harmonix Big Muff Pi. A feedforward deep neural network, which is a variant of the WaveNet architecture, is proposed for modeling these devices. The size of the receptive field of the neural network is selected based on the measured impulse-response length of the circuits. A real-time implementation of the deep neural network is presented, and it is shown that the trained models can be run in real time on a modern desktop computer. Furthermore, it is shown that three minutes of audio is a sufficient amount of data for training the models. The deep neural network studied in this work is useful for real-time virtual analog modeling of nonlinear audio circuits.

1. INTRODUCTION

Guitar distortion effects are traditionally based on analog audio circuitry. These circuits contain nonlinear components, such as diodes, transistors or triodes to produce the desired distortion effect. As most of music production today is carried out using digital audio workstations (DAWs), there is an increasing demand for faithful digital emulations of analog audio effects. The field of virtual analog (VA) modeling is concerned with creating these digital emulations, which allow musicians to record and produce music without investing in expensive analog equipment.

A common approach for VA modeling of distortion effects is "white-box" modeling [1–4]. White-box modeling is based on analysis and discrete-time simulation of the analog circuitry. If the circuit and the characteristics of its nonlinear components are known, white-box modeling can be very accurate. However, circuit simulation can get computationally demanding when there are many reactive components and nonlinear elements in the circuit, and the involved design process can be labor intensive.

An alternative approach for VA modeling is "black-box" modeling. Black-box modeling is based on measuring the circuit's response to some input signals, and creating a model which replicates the observed input-output mapping. Black-box models for VA modeling include block-oriented models, which are based on assumptions about the design of the modeled circuit [5–9]. As an example, a Wiener model [5, 8] emulates the circuit as a linear filter followed by a static nonlinearity. Other black-box modeling methods include Volterra series models [10, 11], dynamical convolution [12] and kernel regression [13].

In our previous work, a deep neural network for black-box modeling of nonlinear audio circuits was presented, and applied to the modeling of a vacuum tube amplifier [14]. The model is based on the WaveNet convolutional neural network [15]. The proposed neural network model is made up of a series of convolutional layers, which consist of a filter followed by a nonlinear activation function. As the filtering and nonlinear processing are applied in several stages, the neural network should be suitable for modeling of a broad range of nonlinear audio circuits.

This work follows the previous work with an emphasis on the real-time performance of the model. Three guitar distortion pedals are modeled in this work: the Ibanez Tube Screamer, the Boss DS-1, and the Electro-Harmonix Big Muff Pi. A hyperparameter search is conducted to find a suitable trade-off between modeling accuracy and computational load. Experiments are carried out to find the minimum amount of data required for successful training. Finally, a low-latency implementation of the proposed deep neural network, which can be run in real time on a consumer-grade computer, is presented.

The rest of this paper is structured as follows. Section 2 provides backround on the modeled distortion effects. Section 3 details the proposed deep neural network for black-box modeling. In Section 4, the developed real-time implementation of the model is presented. In Section 5, the hyperparameter search and its results are detailed, and the effect of the amount of training data on the modeling accuracy is examined. Section 6 presents the modeling results. Finally, Section 7 concludes the paper.

2. MODELED DEVICES

Three guitar distortion effects are considered in this study: the Ibanez Tube Screamer, the Boss DS-1, and the Electro-Harmonix Big Muff Pi. Detailed circuit analyzes of all

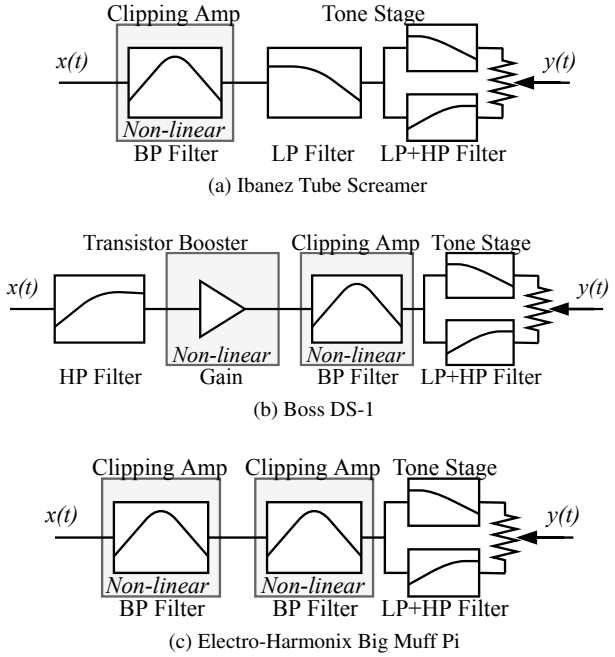

Figure 1: Block diagrams of the distortion effects.

three pedals can be found online [16].

2.1 Ibanez Tube Screamer

The Ibanez Tube Screamer is one of the most well known guitar overdrive pedals. There have been several reissues of the pedal since the release of the original TS808 in the late 1970s [17]. For this study, the TS7 version, which was introduced in the early 2000s, was used. Digital models for the Tube Screamer have been previously proposed by Yeh *et al*. [1, 18], Werner *et al*. [4], and Eichas *et al*. [8].

The simplified structure of the Tube Screamer pedal is shown in Figure 1a. The nonlinear behavior of the pedal occurs in the clipping amp. It is an op-amp-based bandpass filter with diodes in the feedback path of the op amp. After the clipping amp, there is the tone stage, which consists of a passive lowpass filter followed by an active filter, which can act as a low-pass or a high-pass filter depending on the position of the tone potentiometer.

2.2 Boss DS-1

The Boss DS-1 is a famous distortion pedal released in the late 1970s [16]. Its nonlinear characteristics resemble those of a hard clipper. Before this work, digital models for the DS-1 have been proposed by Yeh *et al*. [2, 18].

The DS-1 has two nonlinear stages, as shown in Figure 1b. The transistor booster stage performs high-pass filtering and amplification of the input signal. Nonlinearities are introduced to the signal when the boosted peak-to-peak voltage of the signal exceeds the 9V supply voltage.

The actual distortion effect is produced by the clipping amp. The clipping amp is an op-amp-based bandpass filter with two diodes shunting the output signal to ground. This placing of the diodes introduces a hard-clipping effect. This is in contrast to the soft-clipping effect produced by placing the diodes in the feedback path, as in the Tube Screamer [16]. The tone stage has passive low-pass and high-pass filters whose outputs are mixed based on the setting of the tone knob. Setting the tone knob to the middle position results in a bandstop response, with a center frequency at approximately 500 Hz [16].

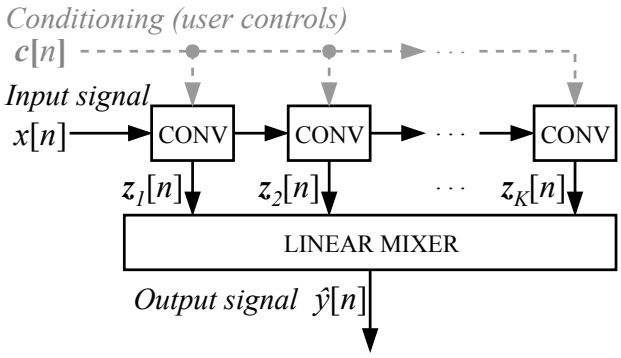

Figure 2: Proposed deep neural network model.

2.3 Electro-Harmonix Big Muff Pi

The Big Muff Pi is a distortion/fuzz effect known for its distinctive long-sustain sound [16, 19]. Electro-Harmonix began mass-producing the pedal in the early 1970s. Since then, various models have been released with different exteriors and with slight circuit modifications [19]. Digital models for the Big Muff have been proposed [8, 20]

A simplified block diagram of the Big Muff is shown in Figure 1c. The circuit has two identical clipping amps in series. However, in some versions, the two clipping amp circuits have different component values, such as different collector resistors [19]. The clipping amp is a transistor-based bandpass filter. As with the Tube Screamer, the clipping in the Big Muff is produced by two diodes placed in the feedback path. The combined effect of the two cascaded soft-clipping amps is hard clipping. The tone stage is similar to the one in the Boss DS-1.

3. DEEP NEURAL NETWORK MODEL

The proposed model for black-box modeling is based on the WaveNet neural network [15]. The original WaveNet is a convolutional autoregressive model, where the previous output sample is fed back to the model for making the next prediction. In our previous work, a feedforward variant of the WaveNet architecture was presented and applied to modeling of a vacuum tube amplifier [14].

The proposed model is shown in Figure 2. The neural network consists of a series of convolutional layers. The raw input waveform is given as input to the first convolutional layer. The convolutional layers apply linear filtering and a nonlinear activation function to the signal.

Optionally, the output of the network can be conditioned on user controls. In the previous work [14], the gain setting of the vacuum tube amplifier was fed to the model along with the input signal, allowing the model to represent different playing configurations of the amplifier. In the experiments of this work, the conditioning is left out,

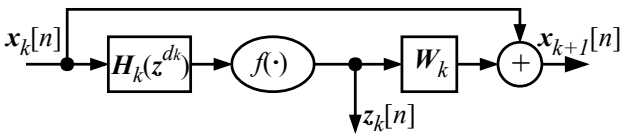

Figure 3: Block diagram of a single convolutional layer.

since we are measuring physical devices, and automatic knob adjustment and data collection is left for future work.

In the previous work, the outputs of the convolutional layers were fed to a three layer "post-processing module" with 1×1 convolutions and nonlinear activation functions. In convolutional neural network terminology, a 1×1 convolution refers to a matrix multiplication applied at each time step in the signal. In this work, the post-processing module is replaced by a linear mixer, i.e., a single linear 1×1 convolution layer. According to our experiments, the network performs similarly or better with the linear output layer, while reducing the complexity and the computational load of the network.

3.1 Convolutional Layer

The convolutional layer used in the model is shown in Figure 3. The input signal is first processed by the dilated causal FIR filter $H_k(z^{d_k})$, where k is the layer index and d_k is the integer-valued "dilation factor" of the filter. Since the convolutional layers generally have multiple channels, the filtering is performed as a multiple-input and multiple-output (MIMO) convolution with a kernel $\boldsymbol{H}_k$. This means that a filter is learned for each pair of input and output channels. The individual filters in the kernel have impulse responses

$$h[n] = \sum_{m=0}^{M-1} w_m \delta[n - md_k], \tag{1}$$

where $\delta[n]$ is the Kronecker delta function, and w_m are the non-zero coefficients of the filters learned by the network.

Next, a nonlinear activation function $f(\cdot)$ is applied to the biased convolution output, producing the layer output

$$\boldsymbol{z}_k[n] = f[(\boldsymbol{H}_k * \boldsymbol{x}_k)[n] + \boldsymbol{b}_k], \tag{2}$$

where $*$ denotes the convolution operator, and $\boldsymbol{b}_k$ is the learned bias term.

The layers include a residual connection, which means that the input to the next layer is

$$\boldsymbol{x}_{k+1}[n] = \boldsymbol{W}_k \boldsymbol{z}_k[n] + \boldsymbol{x}_k[n], \tag{3}$$

where the 1×1 convolution kernel $\boldsymbol{W}_k$ controls the mixing between the layer input $\boldsymbol{x}_k$ and the layer output $\boldsymbol{z}_k$ before the next layer.

Each convolutional layer is a Wiener model: a linear filter followed by a static nonlinearity. Conventional black-box approaches are often based on a Wiener [5, 8], a Hammerstein [6] or a Wiener-Hammerstein [7, 9] model assumption. As a cascade of Wiener models, the proposed neural network makes fewer assumptions about the design of the modeled device, and is expected to be applicable to the modeling of a broad range of nonlinear systems. Furthermore, the deep learning approach optimizes the system response jointly, and not block-by-block, so not only the model but also the optimization makes fewer assumptions about the behavior of the device under study.

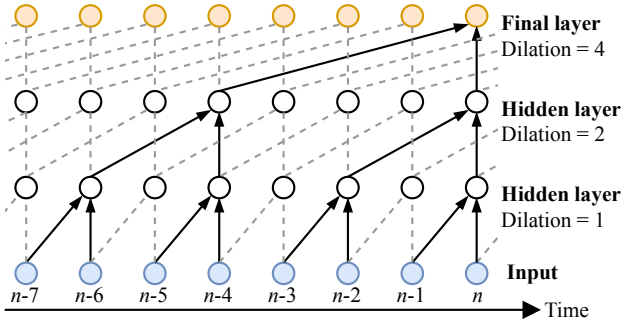

Figure 4: Visualization of three convolutional layers in series and the resulting receptive field of $N = 8$. The figure has been adapted from [15].

3.2 Receptive Field

The proposed neural network is modeling the device under study in a feedforward fashion. The predicted output sample at a time instant n depends only on the N latest input samples, where N is called the receptive field of the model, or the order of the feedforward model. The receptive field depends on the number of convolutional layers, and the lengths of the filters in the layers. This is illustrated in Figure 4. The example network has 3 convolutional layers with dilation factors $d_k = \{1, 2, 4\}$, and $M = 2$ non-zero coefficients for each filter. It can be seen that in this case, the current output sample depends on eight latest input samples. That is, the network has a receptive field of $N = 8$. Generally, the receptive field is given by

$$N = (M - 1) \sum_{k=1}^{K} d_k + 1, \tag{4}$$

where K is the number of convolutional layers. By increasing the dilation by a factor of two in each layer, the model order can be increased to thousands of samples with relatively few layers, allowing feedforward modeling of systems with long impulse responses.

To estimate the required receptive field for modeling of the distortion effects, their linear impulse responses were estimated using the swept-sine technique [6, 21]. A low-level sine sweep was used in order to minimize the effect of circuit nonlinearities in the measurement. The estimated lengths of the impulse responses were approximately 35 ms for the Big Muff, and approximately 45 ms for the Tube Screamer and the DS-1. With a 44.1-kHz sample rate, these correspond to required receptive fields of approximately 1500 to 2000 samples, respectively.

3.3 Loss Function

The neural network was trained by minimizing the error-to-signal ratio (ESR) with respect to the training data. Dur-

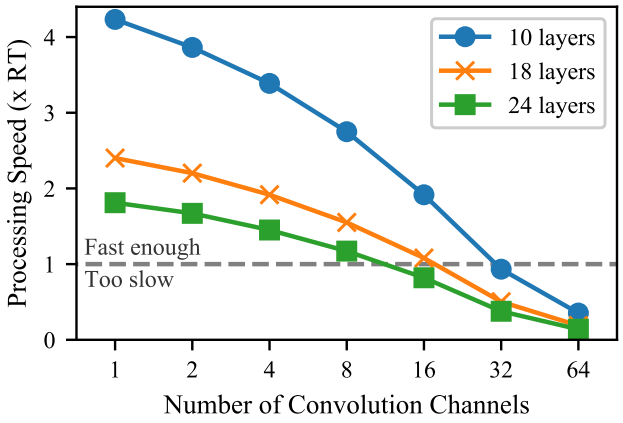

Figure 5: The processing speeds of models with different numbers of layers and convolution channels. The models use the gated activation. The cases above the horizontal dashed line can run in real time.

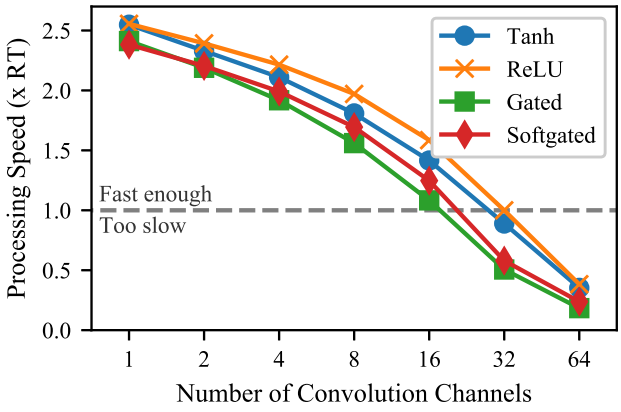

Figure 6: The processing speeds of the 18-layer model with different activation functions and different numbers of convolution channels. The cases above the horizontal dashed line can run in real time.

ing training and validation, a "pre-emphasis" filter was applied to the output and target signals before computing the ESR. For the ith training example, the pre-emphasized ESR is given by

$$\mathcal{E}^{\{i\}} = \frac{\sum_{n=-\infty}^{\infty} |y_p^{\{i\}}[n] - \hat{y}_p^{\{i\}}[n]|^2}{\sum_{n=-\infty}^{\infty} |y_p^{\{i\}}[n]|^2}, \tag{5}$$

where $y_p^{\{i\}}$ is the pre-emphasized target signal, and $\hat{y}_p^{\{i\}}$ is the pre-emphasized neural network output. The ESR can be considered as an energy-normalized sum-of-squares error. Without the energy normalization, the segments in the training data with most energy would dominate the loss.

The pre-emphasis filter was chosen as the first-order high-pass filter with transfer function

$$H(z) = 1 - 0.95z^{-1}, \tag{6}$$

which is very commonly used in speech processing [22]. The purpose of the filtering is to emphasize middle and high frequencies in the loss function. According to our experiments, the neural network struggles at modeling the high-frequency content introduced by the distortion effects without the pre-emphasis filtering.

4. REAL-TIME IMPLEMENTATION

The proposed black-box models were implemented in C++, because the goal was to run the optimized model in real time. The real-time application was built using the open source JUCE framework. JUCE allows building cross-platform audio applications as well as VST, AU, and AAX plugins from a single source code. The Eigen library was used for matrix and vector operations. The source code is available at https://github.com/damskaggep/WaveNetVA.

The C++ implementation of the deep neural network does not currently support model training. Instead, the models are trained using the Tensorflow library. The model hyperparameters and the values of the learned convolution kernels and biases are stored to a JSON file. The trained models can then be loaded to the C++ application.

The real-time performance of the C++ code was estimated for several model configurations. The models were tested using an Apple iMac with an 2.8 GHz Intel Core i5 processor, using a short processing buffer of 64 samples and a sample rate of 44.1 kHz. During the test, all other applications were shut down and the computer was disconnected from the internet. This was done to minimize the effect of other processes in the test.

Figure 5 shows the processing speed of the model with different numbers of layers and different numbers of convolution channels. The models use the gated activation. The processing speed is expressed as a factor of the requirement for real-time application. Clearly, the computational load increases as the number of layers and channels is increased. The largest model running in real time uses 18 layers and 16 channels in the convolutional layers. With 24 layers, a model with 8 convolutional channels can be run in real time.

Figure 6 shows the processing speed of the 18-layer model using different activation functions. The activation functions are detailed in Section 5.2.2. The rectified linear unit (ReLU) is the computationally cheapest and the gated activation is the most computationally expensive activation.

5. EXPERIMENTS

For the experiments described in the following, the neural networks were trained using the Adam optimizer [23]. The validation error was computed after each epoch. Early stopping was used with a patience of 20 epochs. The training data was split into 100 ms training examples, and a mini-batch size of 40 was used. A sample rate of 44.1 kHz was used in the experiments.

5.1 Dataset Generation

Training data was generated by processing audio through the modeled distortion effects. The devices were measured using an audio interface connected to a computer via USB. One output of the audio interface was connected to the input of the measured device. The output of the device was

recorded by connecting it to one of the inputs of the audio interface. The output of the audio interface was also directly connected to another input of the audio interface, in order to estimate the effect of the audio interface in the measurement, as suggested in [9]. The recorded direct signal from the audio interface and the recorded output from the device under study make up the input/target pairs used in the training of the network.

The input sounds processed through the device were obtained from the guitar and bass guitar datasets [1] [2] described in [24, 25], respectively. A random subset with 5 minutes of audio was picked from the datasets, with 2.5 minutes of guitar and 2.5 minutes of bass sounds. The data generated using these sounds was used for training. An additional minute of audio was randomly selected for validation. For testing of the networks, the test set signals from the previous work were reused [14].

All three modeled devices have a knob to control the intensity of the distortion effect and a "Tone" knob to control the filter in the tone stage. For the measurements, all knobs were set to the 12 o'clock, or middle, position. Filtering occurs in the tone stages of all pedals even when the Tone knob is set to the middle position [16]. That is, the middle position of the knob does not indicate an allpass setting for the filters in the tone stages.

5.2 Model Selection

The performance of the proposed neural network depends mostly on the number of channels used in the convolutional layers, the activation function, and the dilation pattern. The choice of these hyperparameters also affects the computational load of the model, as shown in Section 4. Therefore, a hyperparameter search was conducted to find a suitable trade-off between model performance and computational load.

5.2.1 Dilation Pattern

Three different dilation patterns were considered:

$$d_k = \{1, 2, 4, \ldots, 512\},$$
$$d_k = \{1, 2, 4, \ldots, 256, 1, \ldots, 256\}, \text{and}$$
$$d_k = \{1, 2, 4, \ldots, 128, 1, \ldots, 128, 1, \ldots, 128\}.$$

These dilation patterns correspond to models with 10, 18, and 24 convolutional layers, respectively. The number of non-zero coefficients in the filters was set to $M = 3$, which means that, according to Eq. (4), the 10, 18, and 24-layer networks have the receptive fields of $N = 2047, 2045$, and 1530 samples, respectively. At the 44.1-kHz sample rate, these receptive fields correspond to approximately 46, 46, and 35 ms, respectively.

5.2.2 Activation Functions

For the convolutional layers, the performance of the following activation functions are compared: the hyperbolic tangent:

$$\boldsymbol{z} = \tanh(\boldsymbol{H} * \boldsymbol{x}), \tag{7}$$

the rectified linear unit (ReLU):

$$\boldsymbol{z} = \max(0, \boldsymbol{H} * \boldsymbol{x}), \tag{8}$$

and the gated activation, which was used in the original WaveNet [15]:

$$\boldsymbol{z} = \tanh(\boldsymbol{H}_f * \boldsymbol{x}) \odot \sigma(\boldsymbol{H}_g * \boldsymbol{x}), \tag{9}$$

where $\odot$ is the element-wise multiplication operation, $\sigma(\cdot)$ is the logistic sigmoid function, $\boldsymbol{H}_f$ and $\boldsymbol{H}_g$ are the filter and gate convolution kernels, respectively. Finally, the softsign-gated activation, as used in [26], was evaluated:

$$\boldsymbol{z} = g(\boldsymbol{H}_f * \boldsymbol{x}) \odot g(\boldsymbol{H}_g * \boldsymbol{x}), \tag{10}$$

where the hyperbolic tangent and the logistic sigmoid of the standard gated activation are both replaced by the softsign function:

$$g(x) = \frac{x}{1 + |x|}. \tag{11}$$

The softsign nonlinearity can be computationally cheaper than the hyperbolic tangent and the logistic sigmoid functions, as shown in Section 4, while having a similar shape.

With the gated activations, the convolutional layer used in the model can no longer be considered a Wiener model. Instead, it can be described as two parallel Wiener models, whose outputs are multiplied together to produce the layer output.

5.2.3 Results

In the following, the results of the hyperparameter search are presented. As there is an interest on the real-time performance of the models, the validation loss is shown as a function of the processing speed on the developed real-time C++ implementation of the model.

The effect of the choice of dilation pattern on the validation loss is shown in Figure 7. The validation loss is reported as an average loss over all the modeled devices. All models shown in Figure 7 use the gated activation, as given by Eq. (9). The number of convolution channels was varied with values 2, 4, 8, 16, and 32. It can be seen that the 10-layer model performs favorably with respect to the processing speed, while still obtaining a relatively low ESR. The 10-layer model with 16 channels has an average ESR of 4.2%, and runs 1.9 times faster than real time. The 18-layer model with 16 convolution channels had the lowest ESR of the models which run faster than real time. The model has an average ESR of 3.1%, and runs 1.1 times faster than real time.

Overall, the 24-layer model performs more poorly than the 18-layer model. It is possible that this is because the receptive field of the 24-layer model is slightly shorter than the estimated impulse response lengths of the Ibanez Tube Screamer and the Boss DS-1.

The effect of the choice of activation function is shown in Figure 8. The models shown in Figure 8 use 18 layers and the number of convolution channels was again varied

[1] https://www.idmt.fraunhofer.de/en/business_units/m2d/smt/guitar.html

[2] https://www.idmt.fraunhofer.de/en/business_units/m2d/smt/bass_lines.html

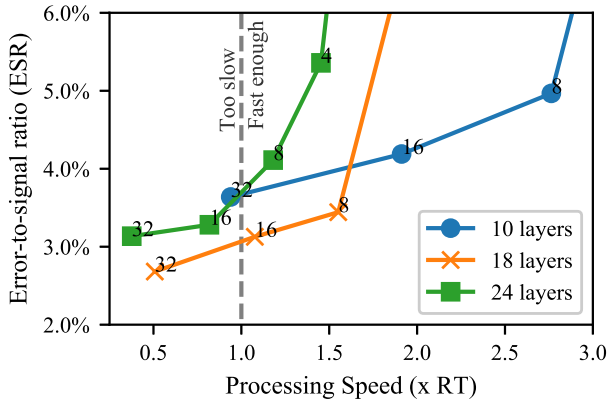

Figure 7: The validation error-to-signal ratio (ESR) as a function of the processing speed, using different numbers of layers and convolution channels. All models shown use the gated activation. The number of convolution channels used is indicated next to each model.

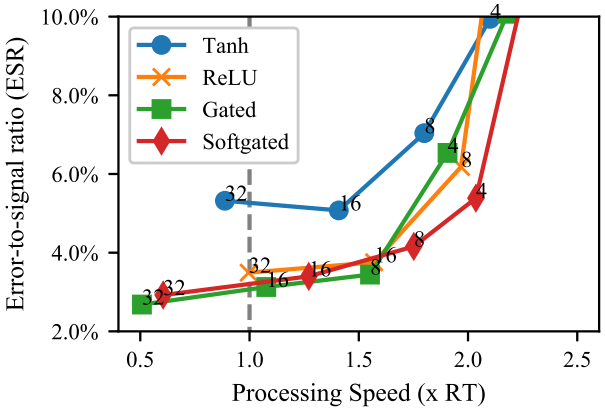

Figure 8: The validation error-to-signal ratio (ESR) as a function of processing speed with different activation functions and different numbers of channels in the convolutional layers. The number of convolution channels used is indicated next to each model.

with values 2, 4, 8, 16, and 32. It can be seen that the hyperbolic tangent activation performs worst out of all activations. The other activation functions perform similarly with each other. This suggest that the ReLU, the softsign-based gated activation and the standard gated activation are all viable options for a real-time application.

Based on the hyperparameter search, three models were chosen for the final evaluation. The hyperparameters of the selected models are shown in Table 1. Only models which run faster than real time were selected. WaveNet1 is the fastest of the selected models, and it has the worst ESR on the validation data. WaveNet3 is the slowest model, and it has the best ESR on the validation data. WaveNet2 is an intermediate model.

5.3 Training Data Length

An interesting question regarding neural networks for virtual analog modeling is the amount of data required for

Table 1: Hyperparameters of selected neural networks.

Model	WaveNet1	WaveNet2	WaveNet3
Activation	Gated	Gated	Gated
Layers	10	18	18
Channels	16	8	16

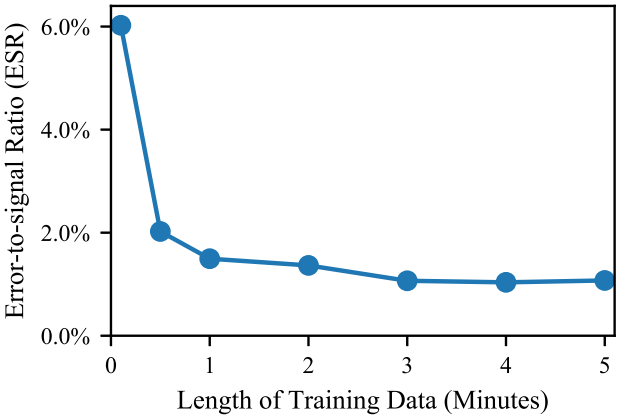

Figure 9: The validation energy-to-signal ratio (ESR) with different amounts of training data.

training a model. To assess the effect of the amount of training data, models were trained with different amounts of training data, and the effect on the validation loss was examined. The results are shown in Figure 9 for WaveNet3, the largest of the selected models. The results are averaged across the three modeled devices.

The validation loss decreases as the amount of training data is increased from 10 seconds to 3 minutes. Increasing the amount of training data past 3 minutes appears to have no significant effect on the validation loss.

6. RESULTS

Table 2 shows the ESRs of the selected models for the Tube Screamer, the DS-1 and the Big Muff. The reported ESR values were computed without pre-emphasis using the unseen test data set.

All selected models achieve a very small ESR on the Tube Screamer, suggesting that the proposed approach leads to a very accurate digital model. With the DS-1 and the Big Muff, the achieved ESR values are higher than with the Tube Screamer. In the tested configurations, the DS-1 and especially the Big Muff are highly nonlinear, due to their cascaded nonlinear stages. We believe that this explains the higher errors when compared to the Tube Screamer, which only has a single soft clipping stage. Overall, the WaveNet3 model has the lowest ESR and WaveNet1 has

Table 2: Error-to-signal ratio for selected test cases.

Model	TS7	DS-1	Big Muff
WaveNet1	0.069%	2.9%	9.9%
WaveNet2	0.050%	3.2%	7.1%
WaveNet3	**0.041%**	**2.1%**	**6.9%**

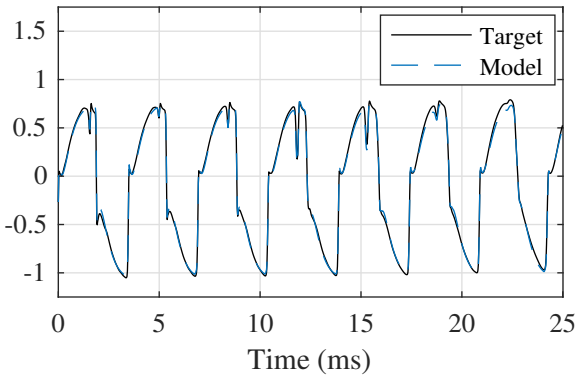

Figure 10: Waveforms of a guitar sound processed through the Boss DS-1, and through the WaveNet1 model.

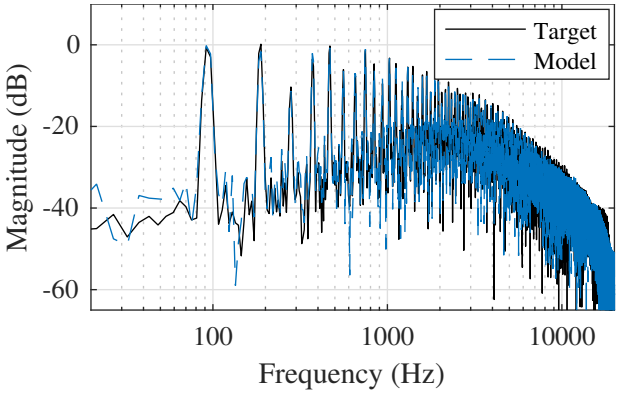

Figure 11: Spectra of a bass guitar sound processed through the Big Muff, and through the WaveNet3 model.

the highest ESR of the tested configurations. However, the differences between the models are relatively small.

The processing speeds of the models are shown in Table 3. The least accurate model (WaveNet1) runs fastest at 1.9 times faster than real time, whereas the most accurate model (WaveNet3) runs 1.1 times faster than real time.

Figure 10 shows the output waveform of the Boss DS-1 distortion effect to an electric guitar input signal from the test data set, and the corresponding output of the WaveNet1 model. The plot shows a good match between the target signal and the model prediction. Figure 11 shows the spectrum of a bass guitar signal processed through the Big Muff distortion effect, and the spectrum of the same signal processed through the WaveNet3 model. The spectrum of the model output matches the target spectrum well.

In order to estimate the aliasing introduced by the models, Figure 12 shows the spectrum of a 1245 Hz sinusoid fed through the WaveNet2 model of the Big Muff pedal. It appears that even though the models were trained with non-aliased data, the learned models suffer from aliasing. However, while the aliasing is evident with a high-frequency sinusoidal input, no clear aliasing could be heard in the guitar and bass sounds processed through the models.

Several audio samples from all models are available online at the accompanying web page [27].

Table 3: Processing speeds of the selected models reported as real-time (RT) factors. The fastest result is highlighted.

Model	WaveNet1	Wavenet2	WaveNet3
Speed (× RT)	**1.9**	1.6	1.1

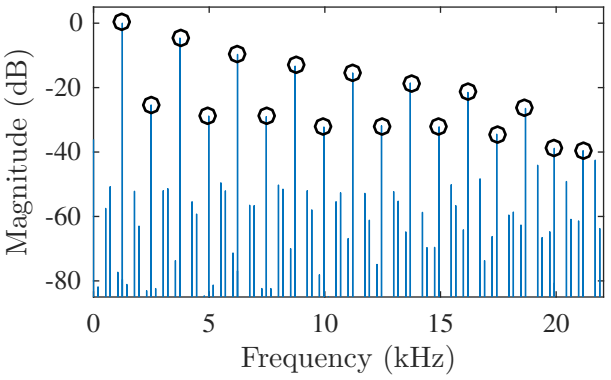

Figure 12: Spectrum of a 1245 Hz sinusoid fed through the WaveNet2 model of the Big Muff pedal. The black circles indicate the non-aliased components.

7. CONCLUSIONS

This work considered the use of deep neural networks for modeling of audio distortion effects. Three well-known guitar distortion pedals were modeled using a feedforward variant of the WaveNet neural network. Different model configurations were examined to find a suitable compromise between modeling accuracy and computational load. A real-time and low-latency implementation of the proposed deep neural network was developed. The results suggest that the proposed deep learning approach can be used to train accurate digital models of analog distortion effects, which can be run in real-time on a consumer-grade desktop computer. Future work will further study the aliasing behavior of neural network models.

Acknowledgments

This work has been partially supported by the NordForsk Nordic University Hub "Nordic Sound and Music Computing Network – NordicSMC", project number 86892. We acknowledge the computational resources provided by the Aalto Science-IT project.

8. REFERENCES

[1] D. T. Yeh, J. Abel, and J. O. Smith, "Simulation of the diode limiter in guitar distortion circuits by numerical solution of ordinary differential equations," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Bordeaux, France, Sept. 2007, pp. 197–204.

[2] D. T. Yeh, J. S. Abel, A. Vladimirescu, and J. O. Smith, "Numerical methods for simulation of guitar distortion circuits," *Computer Music J.*, vol. 32, no. 2, pp. 23–42, 2008.

[3] R. C. D. Paiva, S. D'Angelo, J. Pakarinen, and V. Välimäki, "Emulation of operational amplifiers and diodes in audio distortion circuits," *IEEE Trans. Circ. Syst. II: Express Briefs*, vol. 59, no. 10, pp. 688–692, Oct. 2012.

[4] K. J. Werner, V. Nangia, A. Bernardini, J. O. Smith III, and A. Sarti, "An improved and generalized diode clipper model for wave digital filters," in *Proc. Audio Eng. Soc. 139th Conv.*, New York, NY, Oct. 2015.

[5] J. Schattschneider and U. Zölzer, "Discrete-time models for non-linear audio systems," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Trondheim, Norway, Dec. 1999, pp. 45–48.

[6] A. Novak, L. Simon, F. Kadlec, and P. Lotton, "Nonlinear system identification using exponential swept-sine signal," *IEEE Trans. Instr. Meas.*, vol. 59, no. 8, pp. 2220–2229, Aug. 2010.

[7] C. Kemper, "Musical instrument with acoustic transducer," Aug. 2014, US Patent 8796530B2.

[8] F. Eichas and U. Zölzer, "Black-box modeling of distortion circuits with block-oriented models," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Brno, Czech Republic, Sept. 2016, pp. 39–45.

[9] ——, "Gray-box modeling of guitar amplifiers," *J. Audio Eng. Soc.*, vol. 66, no. 12, pp. 1006–1015, Dec. 2018.

[10] T. Hélie, "Volterra series and state transformation for real-time simulations of audio circuits including saturations: Application to the Moog ladder filter," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 4, pp. 747–759, May 2010.

[11] S. Orcioni *et al.*, "Identification of Volterra models of tube audio devices using multiple-variance method," *J. Audio Eng. Soc.*, vol. 66, no. 10, pp. 823–838, Oct. 2018.

[12] M. J. Kemp, "Analysis and simulation of non-linear audio processes using finite impulse responses derived at multiple impulse amplitudes," in *Proc. Audio Eng. Soc. 106th Conv.*, Munich, Germany, May 1999.

[13] D. J. Gillespie and D. P. Ellis, "Modeling nonlinear circuits with linearized dynamical models via kernel regression," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, New Paltz, NY, USA, Oct. 2013, pp. 93–96.

[14] E.-P. Damskägg, L. Juvela, E. Thuillier, and V. Välimäki, "Deep learning for tube amplifier emulation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Brighton, UK, May 2019.

[15] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *ArXiv pre-print*, 2016, arXiv:1609.03499 [cs.SD].

[16] ElectroSmash, "ElectroSmash – Electronics for audio circuits," Available online at: https://www.electrosmash.com, accessed: 2019-01-29.

[17] Analog Man, "Ibanez Tube Screamer history," Available online at: http://www.analogman.com/tshist.htm, accessed: 2019-01-29.

[18] D. T. Yeh, J. S. Abel, and J. O. Smith, "Simplified, physically-informed models of distortion and overdrive guitar effects pedals," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Bordeaux, France, Sept. 2007, pp. 10–14.

[19] The Big Muff Pi Page, "Evolution of the Big Muff Pi circuit," Available online at: http://www.bigmuffpage.com/Big_Muff_Pi_versions_schematics_part1.html, accessed: 2019-03-26.

[20] K. J. Werner, V. Nangia, J. O. Smith III, and J. S. Abel, "Resolving wave digital filters with multiple/multiport nonlinearities," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Trondheim, Norway, Sept. 2015, pp. 387–394.

[21] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Proc. Audio Eng. Soc. 108th Conv.*, Paris, France, Feb. 2000.

[22] J. D. Markel and A. H. Gray, *Linear Prediction of Speech*. Berlin Heidelberg: Springer-Verlag, 1976.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, San Diego, CA, May 2015.

[24] J. Abeßer, P. Kramer, C. Dittmar, G. Schuller, and I. Fraunhofer, "Parametric audio coding of bass guitar recordings using a tuned physical modeling algorithm," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Maynooth, Ireland, Sept. 2013, pp. 154–161.

[25] C. Kehling, J. Abeßer, C. Dittmar, and G. Schuller, "Automatic tablature transcription of electric guitar recordings by estimation of score- and instrument-related parameters," in *Proc. Int. Conf. Digital Audio Effects (DAFx)*, Erlangen, Germany, Sept. 2014, pp. 219–226.

[26] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proc. 35th Int. Conf. Machine Learning*, Stockholm, Sweden, Jul. 2018.

[27] E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-time modeling of audio distortion circuits with deep learning," accompanying web page, available online at: http://research.spa.aalto.fi/publications/papers/smc19-black-box/.