Slimmable NAM: Neural Amp Models with adjustable runtime computational cost

Steven Atkinson
Atkinson Advanced Modeling, LLC
steve@atkinsonadvancedmodeling.com

Abstract

This work demonstrates "slimmable Neural Amp Models", whose size and computational cost can be changed without additional training and with negligible computational overhead, enabling musicians to easily trade off between the accuracy and compute of the models they are using. The method's performance is quantified against commonly-used baselines, and a real-time demonstration of the model in an audio effect plug-in is developed.

1 Introduction

In recent years, Neural Amp Modeler (NAM)[1] has seen widespread adoption for data-driven virtual analog modeling of musical equipment. With this adoption, it has become the norm that musicians do not train the models they use. Therefore, they are often unable to tailor the modeling process to suit the computational limitations of their use case. For users who find that models are too CPU-intensive to be used in practical settings, model distillation [1] is a common workaround. However, without access to typical ML compute (e.g. a GPU), this is cumbersome and/or time-consuming, and it disrupts creative workflows. By contrast, finite impulse responses (IRs)—a ubiquitous data-driven model for musicians—can be easily made more computationally-lightweight by truncation to an arbitrary length, allowing musicians to trade off accuracy for compute on the fly. One might reasonably desire a similar option in the realm of neural modeling.

This work demonstrates introduces "slimmable NAMs"—neural networks whose size (and, therefore, computational cost) can be reduced after training with negligible computational cost. We focus on NAM's stacked WaveNet architecture due to its strong performance, though the method can be adapted to NAM's other open-source architectures in a similar way. A slider control is introduced to the graphical user interface (GUI) of an audio effect plugin that controls the size of the model in real-time, allowing for its user to easily explore the trade-off between compute and accuracy.

2 Method

A WaveNet [2] is a multi-layer convolutional neural network whose layers operate on a time series of c-dimensional vectors. To slim the network from a width of c to c', the weights $\mathbf{W} \in \mathbb{R}^{c \times c \times k}$ and biases $\boldsymbol{b} \in \mathbb{R}^c$ of each convolutional layer are truncated, yielding $\mathbf{W}' \in \mathbb{R}^{c' \times c' \times k}$ and $\boldsymbol{b}' \in \mathbb{R}^{c'}$, respectively. For the affine projections connecting the inputs of dimension d_x and outputs of dimension d_y to the network ($d_x = d_y = 1$ for mono audio data), only the rows of the input projection and columns of the output projection are truncated so that the input and output dimensionality is preserved.

NAM trains models through supervised learning using a "dry/wet" pair of audio files. During training, in each mini-batch, the network is slimmed to a randomly-chosen $1 \leq c' \leq c$, and the slimmed

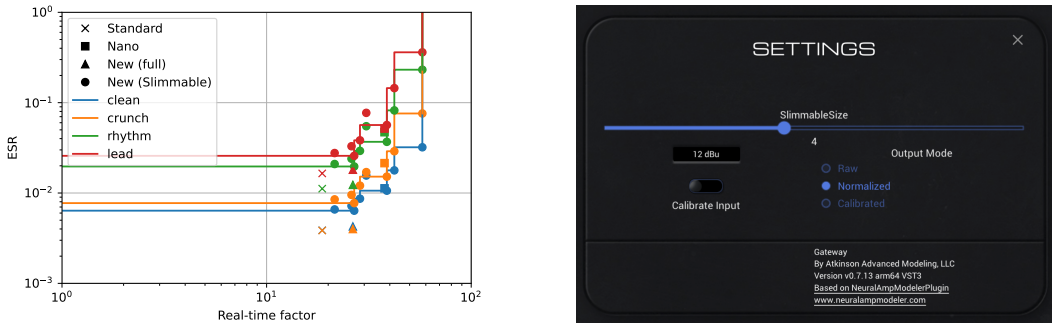
[1] https://neuralampmodeler.com

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI for Music.

network's predictions are supervised against the targets. The resulting neural network can be used with any number of channels when making predictions. Open-source code is available for training and real-time prediction [3, 4], and installers are available for a plugin that integrates the latter [5].

3 Results

Models were trained on recordings of guitar amplifiers spanning a range of tones: a "clean" tone recorded from a Fender Deluxe Reverb, a "crunch" tone recorded from a Morgan MVP23, and hi-gain "rhythm" and "lead" tones recorded from an Omega Ampworks Obsidian. Figure 1a shows the trade-off between compute and accuracy for the slimmable NAM alongside other commonly-used models. For audio examples as well as links to model files, see the video demonstration.[2] Instead of using NAM's "standard" architecture, which[3] stacks two WaveNet models with different widths in series, this demonstration uses a new architecture with a single WaveNet module for simplicity. Fig. 1b shows the settings page of the plugin, adapted from NeuralAmpModelerPlugin[4]. Moving the slider adjusts c', allowing the user to audition models of different sizes in real time.



(a) (b)

Figure 1: a: Real-time factor (higher is better) and accuracy (error-signal ratio [6]; lower is better) for different models of the four tones considered. The Pareto front is drawn for the slimmable NAM. "New (full)" refers to the a model using the slimmable architecture trained only for use at full size as normal. b: Plugin settings page, showing the ability to slim the loaded network on the fly.

4 Conclusions

This work demonstrates Slimmable NAMs and demonstrates their performance and practical use as musical tools.

Decreasing the computational cost of a neural network without additional training is not new; examples include slimmable neural networks [7] which limit the width of the network as in our work; as well as dynamic gating and conditional computation [8]; multi-exit and early-exit networks [9]; and supernets [10]. In audio, Slimmable neural networks have already been applied in speech applications [11]. In virtual analog modeling, pruning has been explored for reducing the computational cost of neural models [12], though this requires additional training. Slimmable models have value in NAM due to their ability to simultaneously remove the compute barrier to real-time prediction while simultaneously placing no additional restrictions or complications to the model-fitting process, making them an attractive solution for both model-makers and the musicians who play them. The slimmable WaveNet architecture introduced by this work is under investigation as the next-generation default model architecture for NAM, with the hope that it will increase the accessibility of the project to musicians.

[2] `https://youtu.be/93WAQsFu694`

[3] As of `https://github.com/sdatkinson/neural-amp-modeler/releases/tag/v0.12.0`

[4] `https://github.com/sdatkinson/NeuralAmpModelerPlugin`

References

[1] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *International Conference on Learning Representations*, 2018.

[2] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *Proc. SSW 2016*, pages 125–125, 2016.

[3] Steven Atkinson. SlimmableNamTrain, November 2025. `https://github.com/Atkinson-Advanced-Modeling/SlimmableNamTrain/releases/tag/v0.1.0`.

[4] Steven Atkinson. SlimmableNamDsp, November 2025. `https://github.com/Atkinson-Advanced-Modeling/SlimmableNamDsp/releases/tag/v0.1.0`.

[5] Steven Atkinson. Gateway, November 2025. `https://drive.google.com/drive/folders/1S0px4-D8jmRq1lS6wfxMloE8Z3Ap0l_d?usp=drive_link`.

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

[7] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. In *International Conference on Learning Representations*, 2019.

[8] Ali Ehteshami Bejnordi and Ralf Krestel. Dynamic channel and layer gating in convolutional neural networks. In *German Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 33–45. Springer, 2020.

[9] Aaqib Saeed. Binary early-exit network for adaptive inference on low-resource devices. *arXiv preprint arXiv:2206.09029*, 2022.

[10] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once for all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020.

[11] Mohamed Elminshawi, Srikanth Raj Chetupalli, and Emanuël AP Habets. Dynamic slimmable networks for efficient speech separation. *arXiv preprint arXiv:2507.06179*, 2025.

[12] David Südholt, Alec Wright, Cumhur Erkut, and Vesa Välimäki. Pruning deep neural network models of guitar distortion effects. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:256–264, 2022.