SPECULATIVE MACHINE LEARNING IN SOUND SYNTHESIS

Luc DÖBEREINER (doebereiner@iem.at) and **David PIRRÒ** (pirro@iem.at)

Institute of Electronic Music and Acoustics, **University of Music and Performative Arts Graz**, Austria

ABSTRACT

This paper explores the intersection of technical, conceptual, and artistic aspects of an ongoing engagement with machine learning (ML), specifically with generative ML in computer music, emphasizing the dynamics of the learning process over its outcomes. Unlike most approaches that prioritize generating "realistic" instrumental or vocal sounds from large datasets, this work integrates the sound synthesis model directly into the learning system, allowing neural networks to produce sound in real-time by continuously adapting and predicting. This method treats the learning process's behavior as a generative sound synthesis process, making it perceptible. The research delves into how various parameters, such as learning rate and ridge regularization, influence the generation of sonorous behavior, and speculates on the possibilities of a radical reformulation of standard learning functions with a particular interest in complex, chaotic, and relational properties of adaptive computational processes. The paper outlines the motivation behind this exploration, introduces the "Speculative Sound Synthesis" project as its context, presents two case studies to illustrate the approach, describes several experimental artistic applications, and concludes with reflections on the findings and future research directions.

1. INTRODUCTION

Over the past decade, AI and in particular machine learning techniques and algorithms have gained a central role in computer music and electronic music composition. Much of the attention in machine learning in sound synthesis has been focused on clearly defined processing tasks such as source separation [1], timbre transfer [2], singing voice synthesis [3], or parameter generation for synthesis using known methods, such as frequency modulation [4, 5]. While some approaches have utilized latent space representations [6, 7] and recurrent neural network dynamics in conjunction with granular synthesis and echo state networks [8] to explore novel approaches, the focus of most research in this area has largely been on the simulative generation of "realistic" vocal or instrumental sounds.

As Jonathan Impett writes, "An AI is essentially a memory machine" [9, p. 225]. One way to understand deep learning systems is as a form of memory of certain past givens. It then becomes an artistic question of how to inscribe, recollect, overwrite and navigate within such a compressed memory. Current synthesis models leveraging deep learning mostly utilize extensive datasets for training and, crucially, they separate the training phase from the inference or prediction phase. The memory is built and later activated. The training phase is therefore regarded as a preparatory step that is carried out with a view to optimal generalization for another task. However, the training phase can also be seen as an adaptive computational process, which itself harbors musically and sonically interesting dynamics and potentials. The two approaches described in this paper are technically and conceptually very different, but both take the adaptive dynamic of machine learning as their starting point. The approach described first speculatively challenges common autoregressive technologies of time-series prediction, while the second approach embraces the emergent behavior or recurrent networks of interacting and independently learning neurons. In both approaches, the learning process is not used for the sake of some external optimization goal, but is itself part of the synthesis process. In other words, we are interested in the specific forms of continuous memory inscription and reactivation as a form of computational performance.

1.1 Speculative Sound Synthesis

The systems and approaches described in this paper were developed as part of the research project Speculative Sound Synthesis, [1] which aims to reconsider the relationship between technology in computer music and artistic practice. It seeks to uncover the aesthetic possibilities of sound synthesis that may go unnoticed if technology is viewed solely as a functional tool for control and execution. In contrast, the approaches described in this text aim to experimentally and exploratively open up machine learning technologies, speculate on their paradigms, and expose their computational materiality for aesthetic experience, musical performance, interaction, and composition.

The models described in this text are *speculative sound models*, which is a central methodological concept of this project. They materially manifest speculations with technological constructs that challenge instrumentality, control, and representation. This is why hyperparameters, such as network structure, learning rate, ridge parameters, and activation functions, become central artistic material instead of being merely subserviant functional components for the realization of external and predefined tasks.

[1] https://speculative.iem.at

1.2 Speculative Propositions

With these aesthetic, technological and methodological considerations in mind, we formulate a set of basic speculative propositions that inform the two approaches described in the remainder of this paper:

1. The adaptive training stage is an essential element of the audible dynamics (online learning) and does not exclusively occur before sound synthesis. Training, learning, and inference all happen during performance. The aim is to create a machine learning model that is perceivable and tangible in both sonic and performative terms. It is a memory in touch with its input and output, learning while it is performing.

2. The sound synthesis model is an integral part of the machine learning system, with the neural network directly producing time-domain signals in "real-time". The synthesis method is thus not external to the model, it is not controlled by the model but what is audible is itself and output of the model.

3. In line with the previous presuppositions, the input data is intentionally kept small. The usage of big data is entangled with structures of power and economic exploitation, with interpolating reproduction of memorized data points. In contrast we regard the small data approach as a form of artistic reappropriation of technology and as a method to expose its learning dynamics rather than to generalize as well as possible with regard to input data. Small datasets can be influenced by the biases and tendencies of musicians and composers, which can be part of the compositional process, and the agency of training datasets can thus become more tangible.

2. AUTOREGNN: AUTOREGRESSIVE FEED-FORWARD NEURAL SYNTHESIS

AutoRegNN [2] is an autoregressive feed-forward neural network that predicts the value of the next sample based on previously generated samples. In principle this method is similar to the seminal WaveNet [10] architecture, although it works with much smaller amounts of input data, does not use convolutional layers and is intended for online training while generating sound. It is implemented as a unit generator (UGen) for *SuperCollider*. It consists of an input layer of variable size, a hidden layer also of variable size, and an output layer, which is always of size one. At each data point, the network utilizes a sequence of previous data points and predicts a single output sample. Autoregression involves using the network's own past predictions as inputs to predict subsequent time steps, thus creating a feedback loop. For each audio sample AutoRegNN performs two separate forward passes, one using external input data, i.e. an input signal whose dynamics are to be learned, and as second pass using its own past output data, the output of which is used as the direct audible output of the system.

[2] The code can be downloaded here: `https://git.iem.at/lucdoebereiner/autoregnn`

The output of the first pass, using the external input, is used to compute the loss and thus to adjust the weights and biases using backpropagation. In addition to the squared error the loss function also contains a ridge regression regularization term proportional to the square of the magnitude of the coefficients. This form of regularization penalizes large weights and thus draws them closer to zero. This stabilizes the learning phase and serves to prevent exploding gradients, which is especially important if the learning rate is modulated and may intentionally be used to generate chaotic behavior. The loss function is:

$$L = (y - \hat{y})^2 + \lambda \sum_{j=1}^{M} w_j^2$$

Where:

- L is the loss function, combining squared error with an L2 regularization term.
- y is the actual value for the single sample (external input).
- $\hat{y}$ is the predicted value by the neural network for the single sample.
- λ is the regularization parameter, controlling the strength of the weight penalty.
- M represents the total number of weights in the neural network.
- w_j denotes the j-th weight in the neural network.

The weights are adjusted after the production of each single sample in order to minimize the loss using backpropagation. At initialization the size of the input layer and the size of the hidden layer (number of perceptrons) can be set. There are two activation functions used throughout the network: "tanh" and "sine" (for details, see subsection 2.1.2). The size of the input layer determines the number of previous samples considered. This also conditions the temporal relationships and frequencies that the network learns to model. In place of the convolutional layers in WaveNet, we have tried various methods of downsampling and filtering the input in order to allow for temporal relations and lower frequencies to be learned, while keeping the input data small. The present iteration employs a cascade of one-pole low pass filters that process the input before it is fed into the input layer. The coefficients for the filters are calculated using the formula $e^{-1/(2^k)}$, where k ranges from 1 to n (n represents the input size). Each succeeding filter is fed into the following one which has proven to increase sensitivity across a wide range of frequencies while reducing the input size.

The model depends strongly on its random initialization, especially if λ, the regularization paramter, is small. A significant parameter to consider is the learning rate. High learning rates can result in chaotic dynamics and rapidly changing timbral output, whereas very small learning rates can generate very gradual changes in pitch and timbral development. The learning rate is thus not treated as hyperparameter to be optimized with regard to the model's ability

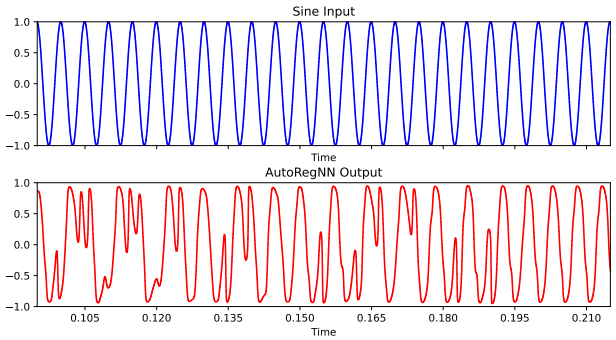

Figure 1. Learning a 200 Hz sine wave signal.

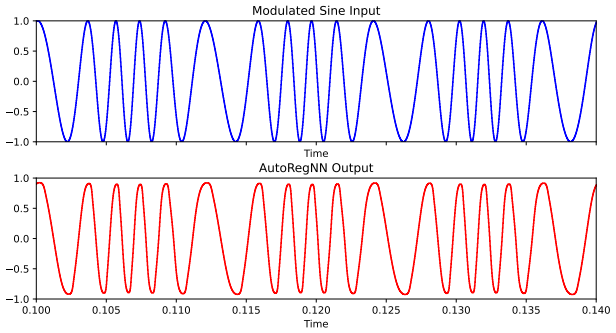

Figure 2. Learning a frequency modulated sine wave.

to generalize, but much rather as a parameter that dyanmically influences the musical and timbral behavior of the synthesis process. Hence, the learning process merges with the actual sound synthesis, and its artifacts and materiality, including chaotic and periodic behavior, serve as both artistic and musical material.

2.1 Examples and Artistic Experiments

This section presents several examples and artistic experiments that illustrate the capabilities of AutoRegNN. The code and audio and video recordings for each of these are available online. [3]

2.1.1 Examples

Fig. 1 shows 120 ms of the input and output of a sine wave with a frequency of 200 Hz fed input an AutoRegNN with and input size of 12, a hidden layer of size 8, a learning rate of 0.1 and a ridge parameter of 0.0001. It demonstrates the initial adaptation as the output is initial somewhat unstable but progressively takes on the input frequency. Lower learning rates lead to longer phases of initial adaptation.

Fig. 2 shows 40 ms of a frequency modulated sine wave. The main difference with regards to the example show in fig. 2 is that the learning rate is much higher (0.6) and the ridge parameter is also much higher (0.01). This leads to a very quick and stable adaptation.

Fig. 3 shows 20 ms of using a 180 Hz sawtooth wave as input. In this example, the learning rate is also modulated using a 180 Hz sawtooth wave with values between 0.01

[3] `https://speculative.iem.at/docs/speculations/Experiments_thoughts/autoregnn/`

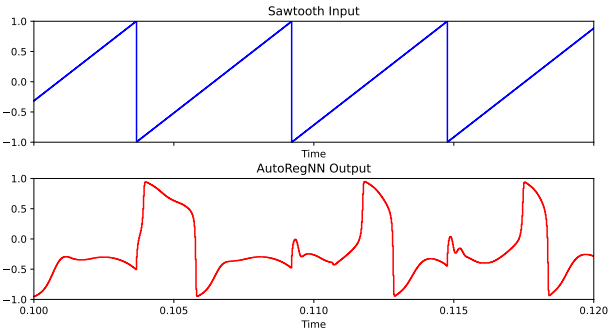

Figure 3. Learning a 180 Hz sawtooth wave with modulated learning rate.

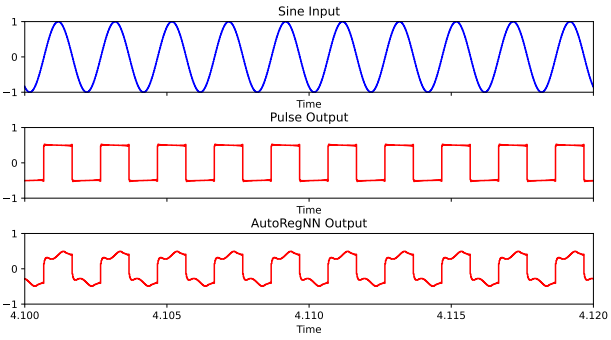

Figure 4. Interpolating between two sets of learned network parameters.

and 1. The ridge parameter is relatively small at 0.00001 in order to allow for a higher degree of unstable and chaotic behavior. Modulating the learning rate can create interesting rich timbres that are still coupled to the input, while exhibiting a certain independence.

There are a number of slightly different variants of AutoRegNN, one of which takes two audio inputs and learns two sets of network parameters (weights and biases), while using a linear interpolation between both sets of parameters for the generation of the output. Fig. 4 shows 20 ms of such an interpolation between sets trained on a sine wave and a pulse wave, both with a frequency of 500 Hz. In this example, the output is generated with the mean values of the trained weights and biases. The learning rate is 0.5 and the ridge parameter is 0.001.

2.1.2 Artistic Experiments

The artistic experiments sketched in this sections serve to document discoveries made during the conceptual and technological deveopment of AutoRegNN. However, the are to be understood as speculative starting points. They point towards possible experimental artistic scenarios and warrant further exploration, expecially with regard to their unique performative affordances and musical potential.

There are several variants of the AutoRegNN SuperCollider ugen that read and write their model parameters form and into external buffers. This allows an interaction with the weights and biases during training and prediction. Any external process that manipulates buffers can be used to

interfere, couple or process the learned paramemeters. It thus further exposes the inner workings of the neural network and can be used to turn it itself into an instrument for sound performance. In one artistic experiment, we have created a multi-slider graphical user interface that exposes all weights and biases as indiviual faders. The direct manipulation of the internal parameters allows for speculative exploration of the network. One can find thresholds at which the system jumps from one type of behavior to another. In conjunction with the control of the learning rate and the ridge regression, one can thus explore the possibilities of the neural network by changing the individual weights. One can find neighboring states while negotiating between manual manipulation and the counterforce of learning (controlled by the learning rate). This creates a kind of speculative co-agency between machine learning and manual exploration.

The activation function variation is a productive feature for sound and behavior. AutoRegNN currently allows interpolation between two activation functions: tanh and sine. Tanh produces more predictable and stable results, while sine is used similarly to a sine map and exhibits chaotic dynamics, but remains derivable and compatible with gradient descent and backpropagation for optimization. It is possible to interpolate between the two activation functions independently for each neuron of the network. This allows for the production of adaptive sounds and sound dynamics that are both chaotic and highly complex in terms of timbre.

Another promising application that produces complex, chaotic, emergent sound textures with intriguing gesturality can be produced by constructing feedback networks of multiple AutoRegNN units. For example, we constructed circular networks of four AutoRegNN units, one of which is initialized with and external signal (noise, e.g.) at the beginning. The initial signal fades out and the generators continue to excite and imitate each other. With different ways of mixing, filtering and delaying the feedback signal between the four generators, a wide variety of distinctive sonic and musical dynamics are created.

We also extended the idea of feedback networks of AutoRegNN generators and conducted experiments with models that incorporate acoustic feedback, where microphone signals provide the input training signal. We also conducted experiments with different interconnected models that interact through microphones in space and by attaching transducers and contact microphones to percussion instruments, especially drums and cymbals. Very small learning rates have been shown to produce interesting gradual changes in pitch and timbre. The ability to mimic their input depends on the type of physical instrument to which each generator is connected via playback through a transducer. For example, when certain resonant frequencies are reached, the gradually evolving timbre can suddenly fall into unstable transients and destabilize the whole network of generators. Moreover, the physical setup enables interactions between materials where the algorithmic, the object and a musician's agency are intertwined, such as through the damping, striking, and wiping of a drum.

Moreover, AutoRegNN has recently been employed by the composer Farzaneh Nouri in an octophonic interactive sound installation titled. *Embedded/Embodied*[4] In conjunction with several layers of AI sound recognition, generation, and communication, the work uses multiple instances of AutoRegNN as a multi-agent system fed by microphone inputs located in different parts of the installation's environment. The agents capture and process sounds from their environment and are used to generate emergent responses that shape and reveal their understanding. Nouri describes the work as an acoustemological exploration.

3. FORGET LEARNING

Most machine learning systems based on neural networks share a few similar paradigms that vary depending on the specific implementation. The experimental system we describe below is the result of a speculative approach that questions some of these paradigms and seeks alternative approaches to the use of neural networks. In its implementation, we followed both scientific and aesthetic criteria.

It is crucial to understand that this case study does not aim to produce a novel system that successfully performs a set of predefined tasks. The system we describe here does not successfully learn patterns or produce categorizations of some input signals. The neural network we develop here is studied in terms of a sound generator that perceptually reveals a specific temporality of the learning process. The dynamics arise from the evolution of the whole system, seen as a collection of individual neurons, each following the same set of rules.

The starting point for this speculative system has been the exploration of the connection between neural networks, as they are interpreted and implemented in machine learning applications, and the computational modeling of their physiological counterparts, biological neural networks, whose organization and structure serve as inspiration for the former. Specifically, from the similarities and, especially, the differences in the understanding of the concepts of learning and adaptation, ideas and inspirations are derived that are at the center of this *speculative sound model*. These inspirations are further put to the test and refined in the practical implementation.

Learning, in the context of neural dynamics is understood as a process of *adaptation* [11, 12] to stimuli that each neuron in a network undergoes. Neurons learn by adapting their activity pattern in response to the input stimuli they receive: in particular, neurons adapt by *reducing* their response to repeated stimuli. The well-known auditory masking effects can be seen as an example of this adaptation mechanism: contrary to some understandings of neural network learning, this means that repeated stimuli are masked or "filtered" by the neurons. Thus, when a neuron (or a network) recognizes a learned stimulus, it does not as one might expect, indicate this with a peak of activity or a strong response. That is, "any adaptive system converges to a state in which any kind of stimulation ceases" [13].

[4] https://2024.sonicacts.com/programme/embedded-embodied

Thus, from the perspective of neural dynamics, for one neuron learns or adapts by finding a configuration of its internal behavior that is less sensitive to the recurrent stimuli and thus more sensitive, i.e. producing a stronger response to input signals that deviate from known patterns.

This mechanism is further developed in recent advances in neuroscience, particularly in the context of the theory of predictive coding. This theory sees the brain as essentially a predictive machine, constantly generating "mental models" of the environment: these models are used to infer or predict input sensory signals, and are then compared with the actual signals received to construct a new model. It has been proposed to apply this perspective not only to the brain as a whole, but also to individual neurons. Thus, *learning* for a single neuron means adapting the weights it applies to its inputs in order to be able to *predict* future inputs. Predicting in this context means minimizing the "surprise" defined as the difference between the actual and the predicted or adapted inputs [14]. This learning rule theory seems to be supported by neural metabolic mechanisms and in particular by the so-called "lazy neuron principle". This principle states that the activity and adaptation mechanisms adopted by neurons are aimed at maximizing their impact on the whole network, while at the same time minimizing their power consumption [15]. That is, each neuron seems to be interested in eliciting maximum activity in the surrounding network, since this means maximizing blood flow and thus nutrient and energy supply, while at the same time minimizing its own energy expenditure in producing electrical signals. In short, neurons seem to want to have a positive energy balance. In mathematical terms (generalizing from [14]):

$$E_i(\mathbf{w}_i) = -f(\sum_{k=1} w_{i,k} x_k) + g(\sum_{j=1} x_j) \qquad (1)$$

Here E_i is the energy balance dependent on $\mathbf{w}_i$ the vector of all weights of neuron i, f is the activation function of that neuron, $w_{i,k}$ is the weight for input signal x_k to the neuron and g is function of the output signals of all neurons in the vicinity.

Another critical aspect of these studies is that learning or adaptation is a process that emerges from the behavior of individual neurons. It is a behavior that affects both how each neuron responds to its environment (the surrounding neurons in the network), but also how that neuron interacts with its environment, i.e., how it "injects" its own activity into it. Learning is thus a global dynamic of a network, resulting from the behavior of each of its constituent units.

This auditory case study explores the dynamics of a network of neurons that behave according to the abovementioned "lazy neuron principle". In this network learning or adaptation to external input signals is the result of the interaction of all neurons in the network and is not controlled and governed by an external function of error minimization.

3.1 Implementation

In this case study, we speculatively simulate the temporal behavior of a recurrent neural network that is constantly adapting to its own dynamics or to some input signal, as in the previous understanding. The neurons are modeled according to the "lazy neuron principle", and learning is understood here as a global process of the whole network that emerges from the behavior of the neurons and is thus not governed by an external optimization process to achieve a predetermined goal. The simulated network is recurrent and consists of a small number of neurons that exchange their output signals. Only one neuron receives the external signal input and the signal produced by one other neuron in the network is taken as the output of the system and audified (see figure 5). We pose that neurons in this network have a defined spatial relationship, and thus, as in biological networks, given the finite conductance of physiological neurons, output signals are delayed differently before being input to other neurons in the network (see figure 6). That is, including the delay factors d_k for the inputs, the above equation 1 becomes:

$$E_i(\mathbf{w}_i) = -f(\sum_{k=1} w_{i,k} x_k(t - d_k)) + g(\sum_{j=1} x_j(t)) \quad (2)$$

In this case study, neural networks are implemented as dynamical systems composed of many interacting elements. We are not interested in "successful" learning, but rather in the temporal dynamics of the learning system. By simulating the network and directly translating its output into sound, we aim to make its behavior audible. Furthermore, since we are not trying to find the best parameters

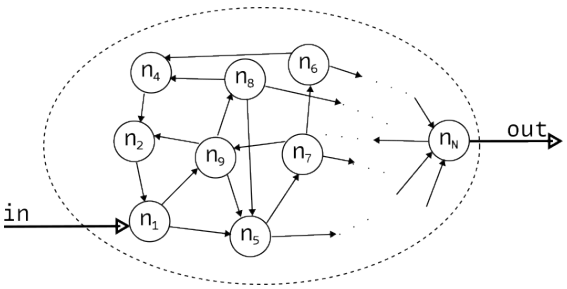

Figure 5. Sketch of the implemented network. Connections in the network are recurrent, the input signal affects one neuron only, while the output signal is read only one other neuron in the network

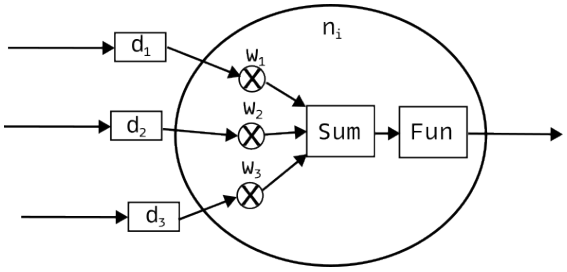

Figure 6. Sketch of the neurons structure in the network. Input signals from other neurons are delayed by a fixed time d_i, multiplied by the respective weight w_i and summed and passed as argument to the activation function.

for a given task, we keep most of the system's parameters flexible, allowing them to be modulated even while the system is running. Thus, the space of behaviors formulating this specific understanding of learning and adaptation is exposed to exploration or composition and also performance as a sound synthesis instrument. Learning rates, delays, and even the shape of activation functions are preserved as real-time parameters.

The departure of this implementation is thus the maximization of the energy balance function E_i of each neuron i (see equation 1) with respect to that neuron's weights $w_{i,j}$. We formulate this process of gradient ascent in differential form as:

$$w'_{i,j}(t) = +\lambda \frac{\partial E_i(\mathbf{w}_i)}{\partial w_{i,j}} \tag{3}$$

Here, $w'_{i,j}(t)$ is the time derivative of the weight $w_{i,j}$ of neuron i, and λ is the so-called *learning rate*. The sign is positive in this case, as the system tries to maximize its energy balance. In typical backpropagation processes, the exact mathematical form of this derivative is determined prior to the start of the process. In this case study, however, since the choice of the particular form of the activation function is left as a parameter of the system, and since the dependence of the input signals on the weights of each neuron is not known beforehand, an exact mathematical form of the function to be maximized by ascent cannot be derived *a priori*. Thus, we have to find another way to numerically compute the gradient ascent.

As we focus on the temporal dynamics of the system, we compute the gradient of the energy balance in equation 3 using the temporal delays of the weights. The partial derivative in equation 3 can be approximated numerically with

$$\frac{\partial E(\mathbf{w}_i)}{\partial w_{i,j}} = \frac{E(\mathbf{w}_i | w_{i,j}(t-1)) - E(\mathbf{w}_i)}{w_{i,j}(t-1) - w_{i,j}(t)} \tag{4}$$

Where $\mathbf{w}_i | w_{i,j}(t-1)$ is the vector of all weights for neuron i, where only it j-th has taken one step (audio sample) before the current one. It can be seen that the above system works very well in finding the local minimum for the function E as long as the difference $w_{i,j}(t-1) - w_{i,j}(t)$ is not 0. This means that, if the weight $w_{i,j}$ has converged and therefore the above difference is zero, the weifth will not adapt anymore and move to a new minimum when the function or the input signals change.

To solve this problem, we introduce an additional slow dynamics into the optimizing system, which induces very slow (with respect to the dynamics of the input signals) and small (with respect to the values of the weights) oscillations around the local minimum. That is, we made the system third-order and introduced a small limit cycle:

$$\begin{aligned} w'_i(n) =& \lambda v_i + s(r - \sqrt{(w_i - u_i)^2 + v_i^2}) w_i \\ v'_i(n) =& -\lambda w_i + \frac{\partial E(\mathbf{w}_i)}{\partial w_{i,j}} \\ & + s(r - \sqrt{(w_i - u_i)^2 + v_i^2}) v_i - a v_i \\ u'_i(n) =& w_i - u_i \end{aligned} \tag{5}$$

Where r is the radius of the limit cycle (typically $r \ll w_{i,j}$), s is the strength of the limit cycle attractor, and the u_i component of the dynamical system acts as a leaky integrator of the w_i dynamics. The resulting behavior of the weights w_i is to converge to the local minimum of the function f with rapidly decreasing oscillations and then remain in a kind of *hovering* behavior, slowly oscillating around the converged point. In a way, this hovering behavior allows the neuron to continuously "sense" variations in its inputs and its environment and whenever there is a change in either be able to react and adjust its weights accordingly, finding a new optimal configuration.

This system was programmed and simulated using the *henri* [5] programming language developed by David Pirrò. *henri* is an experimental programming language, developed using the LLVM compiler infrastructure, [6] and originates mainly from the programmer's artistic practice. *henri* is a programming language designed specifically for the formulation and execution of dynamical systems (in the form of systems of ordinary differential equations) using a variable-order symplectic integrator ([16,17]). Dynamical systems programmed in *henri* are simulated (integrated) in audio rate and thus can be used directly for sound synthesis.

3.2 Examples and Artistic Experiments

The networks developed in this case study have been used to synthesize sound in live-electronics performance settings. Each simulated network runs in real-time, and the temporal dynamics of the continuously changing outputs of their neurons are directly used to synthesize sound. Implemented with *henri*, the temporal behavior of the network is directly made audible, while it is still possible to influence the network parameters in real-time during the simulation. This allows the system to be used as a generative sound synthesis engine. The parameters exposed to external interaction, rather than eliciting immediate responses, can influence the system's temporal behavior at different scales. Choosing and exploring different learning rates for the neuron's own energy consumption, the influx of surrounding neuron energies, the attraction of the limit cycle part, the "friction" value, and the shape of the activation functions are part of the performance.

The networks realized so far consist of a small number of neurons, typically 16 to 32, with a small number of connections each (e.g., 3 to 5). The delays used are either randomly chosen or modified while the simulation is running, with durations close to 50 milliseconds. Some sound

[5] https://git.iem.at/davidpirro/henri
[6] https://llvm.org/

examples are collected on the public web page accompanying this paper. [7] In addition, the *henri* code (a *C* version is planned) used to generate these examples is also made available. [8]

Engaging with these networks requires navigating a complex landscape of behaviors, many of which are challenging to control or "tame." It involves dedicating time to explore subtle changes in parameters and allowing the system to settle into a particular mode. During interaction, these networks exhibit a form of "resistance," a distinct agency of the system that is readily perceptible, making the interaction even more sensible.

With some parameter sets, networks can enter complex self-oscillatory behavior, with different spectral patterns and temporal articulations. These depend strongly on the number of simulated neurons, the chosen activation and energy functions (see equation 1), and the selected delay values. At the same time, the network is open to being influenced by or adapting to external input signals. External sound sources, such as microphone signals or sounds produced by other performers, can be used as input to neurons in the system, and the adaptation behavior to these signals can be modified. Exploring the threshold between a "solipsistic" mode of self-oscillation and a more reactive, input-dependent mode has proven to be one of the most rewarding areas of behavior to explore during performance. The ability to both produce individual sounds and be influenced by external signals is aesthetically and musically inspiring.

As a counterpart to the exploratory attitude of the performative setting, one of the next artistic case studies involves realizing a sound installation for a public space using multiple boards with simple STM32 microcontrollers, [9] a microphone, and a small loudspeaker. Each unit will embody a simplified instance of a neuron in a network, connected through their microphones.

4. CONCLUSIONS

In this paper, we have presented innovative approaches to sound synthesis that leverage the adaptive dynamics of machine learning, not as a means to an end but as an integral part of the creative process. Through the methodological approach of the Speculative Sound Synthesis project, we have explored the aesthetic and performative potentials of speculative sound models, challenging traditional notions of instrumentality and control in computer music. We have described two experiment machine learning systems developed in ways that that diverge from conventional paradigms of neural network applications. The AutoRegNN model demonstrates the richness of incorporating the learning process into sound generation, allowing for real-time, dynamic interactions between the algorithm and its output. The second model described follows the "lazy neuron principle," emphasizing the global dynamics of learning and adaptation as emergent from individual neuron interactions, rather than being driven by external optimization or error minimization.

Our artistic experiments highlight the vast possibilities for future research and artistic exploration, underscoring the significance of small data approaches and the reappropriation of technology for creative expression. These findings open new pathways for understanding and utilizing machine learning in sound synthesis, emphasizing continuous interaction, adaptation, and the exploration of computational materiality as musical and artistic material.

[7] `https://speculative.iem.at/docs/speculations/Experiments_thoughts/unlearn/`

[8] `https://git.iem.at/davidpirro/unlearn`

[9] `https://www.st.com`

Acknowledgments

This project is funded by the Austrian Science Fund (FWF) within the Programme for Arts-based Research (PEEK) – PEEK AR 713-G.

5. REFERENCES

[1] K. Schulze-Forster, G. Richard, L. Kelley, C. S. J. Doire, and R. Badeau, "Unsupervised music source separation using differentiable parametric source models," 2022. [Online]. Available: https://arxiv.org/abs/2201.09592

[2] F. Ganis, E. F. Knudesn, S. V. K. Lyster, R. Otterbein, D. Südholt, and C. Erkut, "Real-time timbre transfer and sound synthesis using ddsp," 2021. [Online]. Available: https://arxiv.org/abs/2103.07220

[3] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Sinsy: A deep neural network-based singing voice synthesis system," 2021. [Online]. Available: https://arxiv.org/abs/2108.02776

[4] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," 2020. [Online]. Available: https://arxiv.org/abs/2001.04643

[5] M. J. Yee-King, L. Fedden, and M. d'Inverno, "Automatic programming of vst sound synthesizers using deep networks and other techniques," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, pp. 150–159, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:4326127

[6] A. Caillon and P. Esling, "Rave: A variational autoencoder for fast and high-quality neural audio synthesis," 2021. [Online]. Available: https://arxiv.org/abs/2111.05011

[7] K. Tatar, D. Bisig, and P. Pasquier, "Introducing latent timbre synthesis," 2020. [Online]. Available: https://arxiv.org/abs/2006.00408

[8] C. Kiefer, "Sample-level sound synthesis with recurrent neural networks and conceptors," *PeerJ Computer Science*, vol. 5, p. e205, Jul. 2019. [Online]. Available: http://dx.doi.org/10.7717/peerj-cs.205

[9] J. Impett, "Intelligent company: Co-creative ai as anamnesis," in *Choreomata*, R. A. Trillo and M. Poliks, Eds. Chapman and Hall/CRC, 2023, pp. 219–239.

[10] A. v. d. 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," 2016. [Online]. Available: https://arxiv.org/abs/1609.03499

[11] J. Benda, "Neural adaptation," *Current Biology*, vol. 31, no. 3, pp. R110–R116, 2021.

[12] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[13] J. A. Martín H, J. de Lope, and D. Maravall, "Adaptation, anticipation and rationality in natural and artificial systems: computational paradigms mimicking nature," *Natural Computing*, vol. 8, pp. 757–775, 2009.

[14] A. Luczak, B. L. McNaughton, and Y. Kubo, "Neurons learn by predicting future activity," *Nature machine intelligence*, vol. 4, no. 1, pp. 62–72, 2022.

[15] A. Luczak and Y. Kubo, "Predictive neuronal adaptation as a basis for consciousness," *Frontiers in Systems Neuroscience*, vol. 15, p. 767461, 2022.

[16] E. Hairer, M. Hochbruck, A. Iserles, and C. Lubich, "Geometric numerical integration," *Oberwolfach Reports*, vol. 3, no. 1, pp. 805–882, 2006.

[17] G. Strang, "On the construction and comparison of difference schemes," *SIAM journal on numerical analysis*, vol. 5, no. 3, pp. 506–517, 1968.