Fundamental aspects of noise in analog-hardware neural networks

Nadezhda Semenova, Javier Porte, Louis Andreoli, Maxime Jacquot, Laurent Larger, Daniel Brunner

Fundamental aspects of noise in analog-hardware neural networks

N. Semenova,[1,2,a] X. Porte,[1] L. Andreoli,[1] M. Jacquot,[1] L. Larger,[1] and D. Brunner[1]
[1] *Département d'Optique P. M. Duffieux, Institut FEMTO-ST, Université Bourgogne-Franche-Comté CNRS UMR 6174, Besançon, France.*
[2] *Department of Physics, Saratov State University, Astrakhanskaya str. 83, 410012 Saratov, Russia.*

(Dated: 23 July 2019)

We study and analyze the fundamental aspects of noise propagation in recurrent as well as deep, multi-layer networks. The main focus of our study are neural networks in analogue hardware, yet the methodology provides insight for networks in general. The system under study consists of noisy linear nodes, and we investigate the signal-to-noise ratio at the network's outputs which is the upper limit to such a system's computing accuracy. We consider additive and multiplicative noise which can be purely local as well as correlated across populations of neurons. This covers the chief internal-perturbations of hardware networks and noise amplitudes were obtained from a physically implemented recurrent neural network and therefore correspond to a real-world system. Analytic solutions agree exceptionally well with numerical data, enabling clear identification of the most critical components and aspects for noise management. Focusing on linear nodes isolates the impact of network connections and allows us to derive strategies for mitigating noise. Our work is the starting point in addressing this aspect of analogue neural networks, and our results identify notoriously sensitive points while simultaneously highlighting the robustness of such computational systems.

The implementation of neural networks in classical, digital hardware has been identified as a serious performance bottleneck. This strongly boosts efforts to realize neural networks in analogue systems hosting the physical links between neurons. Such systems promise to significantly improve speed and energy efficiency, yet they are fundamentally prone to noise originating from their analogue components. Here, we study for the first time how such noise propagates through recurrent and deep, multi-layer networks, and derive an analytical description for such systems. While noise certainly cannot be fully suppressed, our work shows that analogue neural networks can be surprisingly robust. From an architecture point of view, it turns out that the system's sensitivity to noise is mainly located in the first and final layers. Surprisingly, only noise correlated across populations of neurons proofs to have crucial effects for information propagation through the network; meanwhile purely local, uncorrelated noise can mostly be ignored or mediated. These are indispensable insights for designing high future analogue hardware neural networks.

I. INTRODUCTION

Networks are the underlying principle for countless physical systems and information processing concepts. Particularly in computing, they support a wide range of powerful algorithms such as Hopfield[1] and neural[2] networks as well as in Ising machines[3]. Especially neural networks have recently resulted in a revolution of modern computing[4]. By now, deep neural networks achieve super-human performance in computational tasks previously deemed un-solvable by computers. They are now established as indispensable and as the current disruptive computing technology.

A fundamental aspect of neural networks is the propagation of information between nodes along weighted connections, making parallelism essential in neural network computing concepts. However, our digital computing architectures are serial and spatially separate memory from the location of information transformation. As each of a neural network's connection-weight corresponds to one value of memory, the ratio between floating point operations (FLOP) and memory access (byte) is fundamentally skewed when compared to classical computing. As a consequence, high-performance neural network computing is today performed on special-purpose integrated circuits (SAICs) such a graphic processors (GPU) or Google's tensor processing unit (TPU)[5].

Maximal computing performance can therefore only be achieved if neural networks are fully hardware implemented. In such systems, the overhead due to serial communication is avoided: each neuron corresponds to a simple nonlinear component, each connection to a direct physical link. Motivated by the potential benefits, research activity along these lines has lately exploded. Novel physical components such as lasers[6], memristors[7] and spin-torque oscillators[8] have been shown to serve as excellent analogue neurons. Simultaneously, mostly optical concepts for parallel networks based on holography[9], diffraction[10,11], integrated networks of Mach-Zehnder modulators[12] and wavelength division multiplexing[13] have been demonstrated. Several review articles summarize various trends inside this area[14–17].

It is equally clear that, besides the potentially large benefits, such parallel and analogue hardware platforms face new, fundamental challenges. Among the most ba-

[a] Electronic mail: nadezhda.semenova@femto-st.fr

sic differences to digital circuits, is that the noise of individual components propagates along the network. Consequently, an important concern is that such systems might ultimately succumb to the detrimental impact of noise, rendering computing impossible. Regardless of its relevance, no previous study analyses this aspect in detail, but focus on noise propagation in serial analogue circuits[18] and the interaction between noise and learning[19]. Here, we study fundamental properties of noise and propagation along connections of networks, considering feed forward neural networks (FNNs) as well as recurrent neural networks (RNNs). We consider correlated and uncorrelated, multiplicative and additive noise present in individual neurons and use noise amplitudes extracted from a physical experiment[10]. This covers a large range of noise sources possibly encountered in analogue hardware neural networks. We assume linear neurons in order to exclusively focus on the mixing of noise due to network-connections. Analytical dependencies of noise propagating are derived, providing a clear understanding of the relevant processes. Finally, we provide guidelines for hardware architectures and identify the critical points where such systems are most vulnerable.

II. ANALOGUE HARDWARE NEURAL NETWORKS

Neural networks can be found in multiple configurations which can be categorized by their connectivity. In general, a neuron's internal state x sums input from other neurons according to connectivity weights $\mathbf{W}$. The internal state is then mapped onto its output y'. Figure 1(a) schematically illustrates such a unit, also referred to as a perceptron. For neuron i, the set of governing equations is

$$x_i = \sum_j^I W_{i,j} x_j + b_i, \tag{1}$$
$$y'_i = f(x_i). \tag{2}$$

In this simple description the number of neurons is I, and b_i is a constant bias offset. As previously mentioned, we will restrict this work to a linear mapping according to $f(x) = \alpha x$. Importantly, if the neuron is located in the first layer, then its input x_j is replaced by th system's input signal u.

This set of equations forms the basis of our description. It will be adopted to different architectures, while neurons will include sources of noise. Typically, we normalize connection matrices to their largest eigenvalue, which (i) maintains the signal amplitude for FNNs, or (ii) makes RNNs more comparable to FNNs. In FNNs, connections $\mathbf{W}$ are established in a cascaded manner, connecting the neurons of a layer to the neurons of a previous layers. In RNNs, connections $\mathbf{W}$ of a hidden layer establish connections between neurons from the same layer. This introduces a temporal context in the RNNs state, and such systems feature short term memory and can be employed to process temporal information[20].

In addition, we restrict our treatment to connection matrices with purely positive connections. This was (i) motivated by our hardware-system, and (ii) corresponds to a restriction commonly found in optical networks[9].

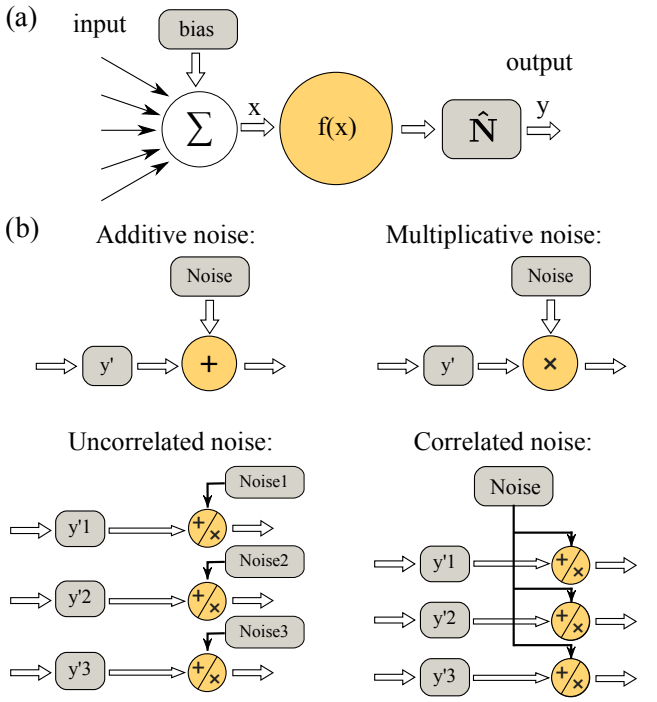

FIG. 1. Schematic diagram showing the signal propagation through one noisy neuron.

Finally we only utilize single neurons in the first (input) and final (output) layer.

A. Different sources of noise

For simplicity we start by considering sources and the impact of noise upon a single neuron. A schematic illustration of the relevant processes are shown in Fig. 1(a). After a neuron's internal state x has been mapped onto its noiseless output y' according to Eq. (2), the signal becomes perturbed by noise operator $\hat{\mathbf{N}}$:

$$y = y' + \hat{\mathbf{N}}(y'). \tag{3}$$

We focus on two noise families, see Fig. 1(b). For additive noise, operator $\hat{\mathbf{N}}$ does not depend on incoming signal y', i.e. has properties and characteristics independent of the neuron's internal state. If, however, noise is multiplicative, then operator $\hat{\mathbf{N}}$ depends on y' and hence on x.

In this work we consider only white Gaussian noise sources according to:

$$\text{additive noise: } \hat{\mathbf{N}}(y') = \sqrt{2D_A}\xi_A$$
$$\text{multiplicative noise: } \hat{\mathbf{N}}(y') = y' \cdot \sqrt{2D_M}\xi_M,$$

where ξ_A and ξ_M are sources of white Gaussian noise with zero expectation value and a variances of 1. The variance of the noise operator is controlled by noise intensities D_A and D_M.

In our description, each ξ value can depend on four indexes: (i) - time $t \in [1, T]$ (T is the length of the input sequence), (ii) - repetition number $k \in [1, K]$ (needed for

SNR calculation), (iii) - layer number $n \in [1, N]$ and (iv) - a neuron's index $i \in [1, I_n]$ in a layer n. At this stage we need to contemplate about the potential impact typical hardware circuitry can impose. Most fundamentally speaking, component (neuron) internal processes will be locally unique and hence result in noise which is different for each neuron at each instant in time. The describing term depends on each index t, k, n, i and hence we refer to this type as *uncorrelated noise*. A schematic circuit noise-model for uncorrelated noise is illustrated in Fig. 1(b). In the present work we denote uncorrelated noise by the letter $'U'$. Uncorrelated additive noise, for example, is governed by $\xi_A^U = \xi_A^U(t, k, n, i)$ with D_A^U as its noise intensity.

Another feature commonly encountered in hardware circuits is that a few elements impact / control the overall circuit's state. A simple example would be a circuit's supply voltage or its temperature. Noise in such central components will perturb large fractions of the circuit in a comparable manner. In our considerations such noise is the same inside one layer and therefore only depends on time t, number of repetition k and the layer's number n. It is therefore correlated across neuron populations, and *correlated noise* is denoted by the letter $'C'$. A schematic circuit noise-model for correlated noise is illustrated in Fig. 1(b). Correlated additive noise, for example, is governed by $\xi_A^C = \xi_A^C(t, k, n)$ with D_A^C as its corresponding noise intensity.

Each noisy neuron can therefore exhibit four different types of noise: correlated additive ξ_A^C and correlated multiplicative ξ_M^C, as well as uncorrelated additive ξ_A^U and uncorrelated multiplicative ξ_M^U. In the presence of both types of noise the output signal emitted by a single neuron at time t is

$$y_{n,i}^t = y_{n,i}'^t \cdot \left(1 + \sqrt{2D_M^U}\xi_{Mn,i}^U\right)\left(1 + \sqrt{2D_M^C}\xi_{Mn}^C\right) + \sqrt{2D_A^U}\xi_{An,i}^U + \sqrt{2D_A^C}\xi_{An}^C, \tag{4}$$

Crucially, the introduced four noise classes were motivated by findings during the careful characterization of our experiment[10]. We found that the different components exhibit and induce different noise characteristics. The camera, for example, mostly contributed uncorrelated multiplicative noise, which we attributed to timing-jitter in the device's clock. The illuminating laser, on the other hand, influences all neuron-states simultaneously and results in correlated multiplicative noise. The same type of noise is induced by the spatial light modulator, and importantly it is of significantly larger intensity and hence dominates over the laser's contribution. The SLM also contributed uncorrelated, additive noise, and so does the system's readout-detector. We have characterized each of these sources individually and in detail. Noise intensities used on the following are based upon the values we obtained.

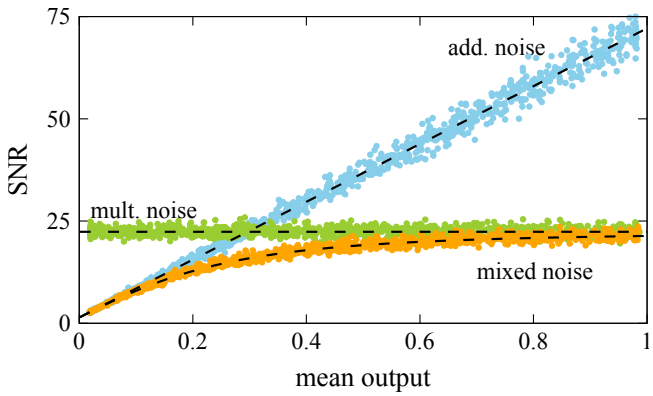

FIG. 2. Signal-to-noise ratio for one neuron in the case of only additive noise (blue points) with intensity $D_A^U = 10^{-4}$, only multiplicative noise (green points) with intensity $D_M^U = 10^{-3}$ and mixed noise (orange points) with both types of noise of the same intensities. Parameters: $\alpha = 1$, $b = 0.02$.

B. Signal to noise ratio

To quantify the corruption of a signal by noise, we employ the signal-to-noise ratio (SNR)

$$\mathrm{SNR}(y^t) = \frac{E(y^t)}{\sigma(y^t)}, \tag{5}$$

where y is the noisy signal, $E(\cdot)$ and $\sigma(\cdot)$ are the signal's mean value and standard deviation, respectively.

There are many other well-known method for noise analysis, for example autocorrelation, cross-correlation and consistency based analysis. However in the present article, we (i) know the temporal characteristics of the noise-sources, and (ii) are more interested in signal amplitude-related effects which enable a first interpretation of results with respect to a system with nonlinear neurons. We therefore break the overall SNR down into its values at different amplitudes $E(y)$. This allows to associate system performance to potential modifications by nonlinearities. It furthermore illustrates the importance of exploiting a hardware system's entire dynamical amplitude range.

We follow that strategy and procedure for characterizing deep FNNs as well as RNNs. The network's input signal at time t is u^t, leading to network output y^t. We repeat the same input sequence u^t for several repetitions $k \in [1, K]$ and obtain the sequence of the output values y_k^t. We average the obtained sequence across all ks and obtain the mean-response $E(y^t) = 1/K \sum_{k=1}^K y_k^t$. The standard deviation of sequence $\sigma(y_k^t)$ is obtained following the same strategy.

We first consider the impact of additive and multiplicative noise ($D_A^U = 10^{-4}, D_M^U = 10^{-3}$) on the SNR of a single neuron. Figure 2 shows the SNR depending on the mean output values $E(y^t)$ for the cases of only additive (blue points), only multiplicative (geen points), as well as a combination of both noise types, hence mixed noise (orange points).

A single-neuron's SNR has features which form the basis for interpreting more complex systems. Due to

the zero expectation value for the input signal and all noise sources, the mean value of the neuron's output is $E(y^t) = f(x) = u^t$. As derived in Appendix A, the variance of the noisy output sequence is $\mathrm{Var}(y^t) = \sigma^2(y^t) \approx 2D_A^U + 2D_M^U f(x)^2$, leading to $\mathrm{SNR}(y^t) = E(y^t)/\sqrt{2D_A^U + 2D_M^U (E(y^t))^2}$. For the case of only multiplicative noise this simplifies to $\mathrm{SNR}(y^t) = 1/\sqrt{2D_M^U}$, which is constant for each output signal. As shown by the green data in Fig. 2, noise output signal level increase both linearly and the SNR is constant. For the case of additive noise only, the SNR becomes $\mathrm{SNR}(y^t) = E(y^t)/\sqrt{2D_A^U}$, and the SNR of additive noise therefore increases linearly with $E(y^t)$ according to slope $1/\sqrt{2D_A^U}$, see blue data in Fig. 2. Finally, the SNR for mixed noise combines both features. For small output values it coincides with the SNR for additive noise, while for larger output values the SNR is limited by the constant SNR of multiplicative noise.

Before we can apply the single neuron nomenclature to networks, we need to introduce a final tool. Each node of an artificial neural network can exhibit some bias b. In particular for the here considered linear system with only uni-polar (positive) connection weights, this bias can accumulate during its propagation through network layers, creating a constant output signal offset. For deep FNNs, this offset increases with the number of layers. According to unfolding in time[21], a RNN can be mapped onto a FNN, linking the strength of a RNN's internal coupling too the depth of a corresponding FNN. Similar offset accumulation is therefore found when increasing a RNN's internal connection strengths. This offset is constant and does therefore not contribute to information processing, yet it would increase $E(y^t)$ and hence induce artefacts in $\mathrm{SNR}(y^t)$. We calculate the system's response without noise on a zero-input signal 0, and by that isolate the contribution of the constant biases. We refer to this value as the *shifting constant* C, which needs to be subtracted from each output value before calculating the SNR. We are able to analytically describe this step for FNNs and RNNs and hence maintain the generality of our analysis.

III. NOISE IN DEEP FEED-FORWARD NETWORKS

We consider the FNN schematically illustrated in Fig. 3. The network consists of four layers, with one neuron in the first and last layer $I_1 = I_4 = 1$, and $I_2 = I_3 = 200$ neurons in the two hidden layers. The neuron in the first layer is connected to the system's input u^t.

The system's evolution is governed by

$$x_{n,i}^t = \sum_{j=1}^{I_{n-1}} W_{i,j}^n y_{n-1,j}^t + b_i, \quad (6)$$

$$y_{n,i}'^t = f\left(x_i^t\right), \quad (7)$$

and the structure of Eq. 7 connects neurons of layer n exclusively to neurons in layer $n-1$. For generality, we focus on global coupling between each layer with equal weights. The impact of inter-layer connectivity is discussed in Section VI. The connections from layer $n-1$ to layer n is determined by matrix $\mathbf{W}^n$ of size $I_{n-1} \times I_n$ and for global coupling $W_{i,j}^n = 1/I_{n-1}$. For sake of clarity we label $\mathbf{W}^2$ and $\mathbf{W}^N$ as input $\mathbf{W}^{\mathrm{in}}$ and output $\mathbf{W}^{\mathrm{out}}$ matrices, respectively.

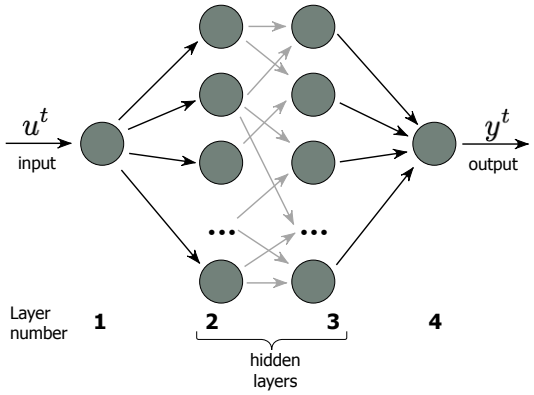

FIG. 3. Schematic representation of a simple feed-forward neural network (FNN) consisting of $N = 4$ layers. The central two layers are referred to as hidden layers, which each hosts $I_2 = I_3 = 200$ neurons. The first and final layers consist of single neurons, $I_1 = I_4 = 1$.

For a FNN with such a topology, the shifting constant in the last layer is

$$C_4 = b + \alpha(b + \alpha(b + (0 + b))) = b(1 + 2\alpha^2) + \alpha b, \quad (8)$$

where $(0 + b)$ is the signal coming from the first layer, and transformations $\alpha(b + \cdots)$ are made in each hidden layer. For the more general case of an unknown number of layers $N \geq 4$, the shifting constant is

$$C_N = b(1 + 2\alpha^{N-2}) + b \sum_{j=1}^{N-3} \alpha^j. \quad (9)$$

A. Correlated and uncorrelated noise

Within our framework, the state of each noisy neuron inside a hidden layer of a FNN exhibits two multiplicative and two additive noise sources, see Eq. (4). Figure 4 shows the numerically obtained SNR for exclusively uncorrelated noise in panel (a), and for exclusively correlated noise in panel (b). Both panels are identically scaled, and the detrimental impact of correlated noise compared to uncorrelated noise is clearly visible. However, corresponding dependencies do not differ qualitatively. Furthermore, there is little quailitative difference between a single node perturbed by noise, Fig. 2, and a FNN's output, Fig. 4. From this comparison, we can conclude that additive, multiplicative and mixed noises have a very comparable effect on a FNN's and on a single noise neuron's output signal. The only notable difference is the contribution of multiplicative noise for small $E(y^t)$ values, which is due to shifting constant C_4.

In general, the overall SNR is reduced by the network. However, the accumulation of noise is very little: a single noisy neuron has less than twice the SNR compared to the collective of over 400 noisy neurons. In particular for

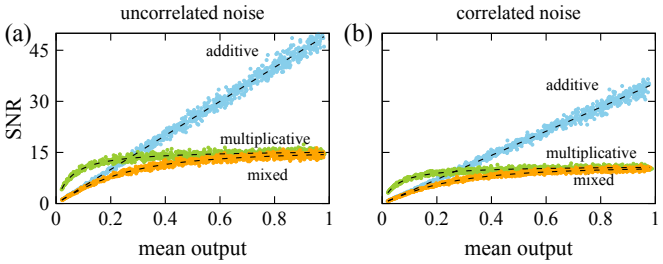

FIG. 4. SNR dependences for uncorrelated noise (a) and correlated noise (b). Blue points correspond to only additive noise of intensity $D_A^U = D_A^C = 10^{-4}$, green points indicate to only multiplicative noise with intensities $D_M^U = D_M^C = 10^{-3}$, orange points match mixed noise with combination of both noise intensities. Parameters: $\alpha = 1$, $b = 0.02$.

the case of uncorrelated noise the reduction in SNR is moderate.

As the connection between the FNN layers is global, each neuron is driven by an averaged-state of the previous layer, a mean-field so to speak. This is where a network inherently aids keeping the influence of noise in check. Uncorrelated noise is different for each neuron of a preceding layer, and the influence of these different, uncorrelated noise-terms is efficiently averaged out by the network's connections. For correlated noise the situation is different. This term is identical for all neurons in preceding layers, averaging cannot curb the propagation of correlated noise. This is an essential insight: in analogue neural networks an optimal network connectivity does not only address best the task it is proposed to solve, but also will depend on the network's very own noise properties.

Similar considerations also highlight the importance of neurons located in the first and last layers. The first layer spreads its noise across the entire system, and hence induces an additional form of correlated noise. The system's output, on the other hand, is the position where the computational results is presented to the outside world, and no averaging by network-connections is suppressing suppressing its uncorrelated noise.

B. Signal to noise ratio

The previous line of argument can be formulated analytically. Here we focus on the main results, the full derivation is given in Appendix A.

The mean value and variance of the the first layer are

$$E(y_1^t) = E(y_1'^t) = u^t + b, \tag{10}$$

$$\mathrm{Var}(y_1^t) = \sigma_{add}^2 + E^2(y_1^t) \cdot \sigma_{mult}^2, \tag{11}$$

with σ_{add}^2 is the combined perturbation due to additive noise $\sigma_{add}^2 = 2D_A^U + 2D_A^C$, and for multiplicative noise $\sigma_{mult}^2 = 2D_M^U + 2D_M^C + 4D_M^C D_M^U$.

Due to the here imposed symmetric topology, we find that

$$E(y_{n,i}^t) = E(y_{n,i}'^t) = E(y_n'^t) = \alpha(E(y_{n-1}'^t) + b) \tag{12}$$

This formula is valid for any hidden layer. The variable $y_n'^t$ does not contain the noise of nth layer but has all the noise from previous layers. The corresponding variance can be approximated according to

$$\mathrm{Var}(y_n'^t) \approx \alpha^2\big[2D_A^C + 2D_M^C E^2(y_{n-1}'^t) + (1 + 2D_M^C)\mathrm{Var}(y_{n-1}'^t)\big], \quad 2 < n < N. \tag{13}$$

Equation (13) is used until $n = 2$. The variance for the 2nd layer is $\mathrm{Var}(y_{2,n}'^t) = \alpha^2 \mathrm{Var}(y_1^t)$. Equations (12) and (13) are recursively defined in function of the network's depth. From these relationships it becomes clear that the variance of hidden layers does not contain uncorrelated noise-contributions originating from the previous hidden layers.

Finally, the Nth layer mean signal and variance, hence of the output signal are

$$E(y_N^t) = E(y_{N-1}^t) + b \tag{14}$$

$$\mathrm{Var}(y_N^t) \approx \sigma_{add}^2 + E^2(y_N^t) \cdot \sigma_{mult}^2 + (1 + \sigma_{mult}^2) \cdot \mathrm{Var}(y_N'^t). \tag{15}$$

Equations (11) and (15) confirm that the first and last layer add both, correlated and uncorrelated noise to the output signal.

Based on these considerations, we arrive at the system's final SNR

$$\mathrm{SNR}(y_N^t) = \frac{E(y_N^t)}{\big(\mathrm{Var}(y_N^t)\big)^{1/2}}, \tag{16}$$

which only requires knowledge of individual component's noise properties. Figure 4 shows analytically derived dependencies as dashed lines. The agreement between the numerical simulation and the analytical description is excellent. The noise of hardware neurons inside a network can be characterized in model systems, while the properties of input as well as the desired output signals are typically known for most tasks. The here derived analytical description is therefore of important predictive value for estimating hardware-linked performance limits in future analogue neural networks.

IV. NOISE IN RECURRENT NETWORKS

Following the previously developed techniques, we now turn to propagation of noise through RNNs. Figure 5(a) shows the general scheme of such a network. In our case, the network contains one input and one output neuron (orange circles), while the single hidden layer hosts $I_2 = 200$ neurons. As for the case of the feed-forward network, the additional bias constant acts for all network nodes. The evolution of neurons in the hidden-layer is given by

$$y_{2,i}'^{t+1} = f\big(\gamma \mathbf{W}^{\mathrm{in}} y_1^{t+1} + \beta \mathbf{W} \mathbf{y}_2^t + b\big), \tag{17}$$

The output signal of our RNN is

$$y_3'^{t+1} = \mathbf{W}^{\mathrm{out}} \mathbf{y}_2^{t+1} + b. \tag{18}$$

Matrices $\mathbf{W}^{\mathrm{in}}$ and $\mathbf{W}^{\mathrm{out}}$ define the connections between the hidden layer to the input and output neurons, respectively. In contrast to FNNs, the system's state does not

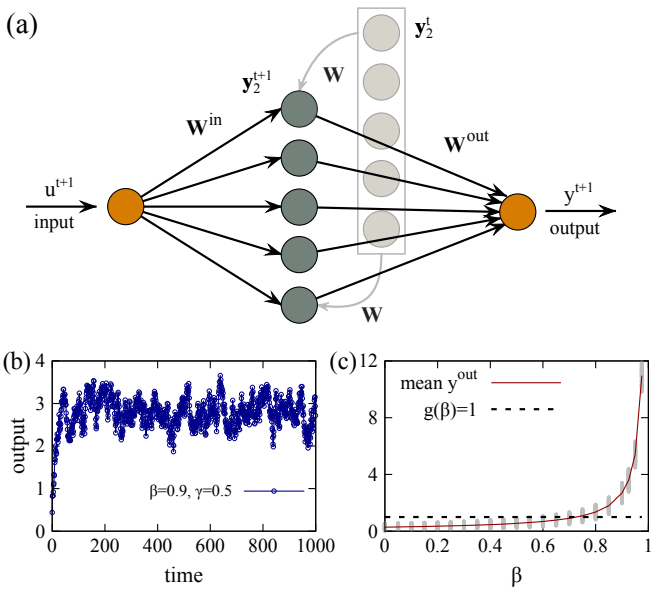

FIG. 5. Schematic representation of simple recurrent neural network (1) consisting of input and output neurons (orange circles) and hidden (recurrent) layer (gray circles) of $I_2 = 200$ nodes. The panel (b) illustrates the output signal for $\gamma = 0.5$, $\beta = 0.9$. The panel (c) depicts shifts of output signal range (gray points) and focusing mean value (red line). The coupling is global, $\alpha = 1$, $b = 0.02$.

only depend on input signal u^{t+1}, but also on the hidden layer's state at previous times according to $\mathbf{W}$ in Eq. 17. There is only one recurrent layer, so the index n is not used for RNN. Parameters β and γ can be interpreted as balancing coefficient between the system's previous state x^t or the current input signal u^{t+1}. If $\beta = 0$ and $\gamma = 1$ the network has a FNN topology with a single hidden layer. For $\gamma = \beta = 0.5$ the input signal and the recurrent signal have equal force.

As in FNNs, it is important to consider the impact of constant signal accumulation in an RNN. However, the interplay between recurrence and input signal leads to a nonlinear modification of an RNN's output signal. As the nodes are linear and we normalized all matrices to their largest eigenvalue, a FNN's output range depends only on the input signal's range. In a RNN, this limit is fundamentally harder to obtain. Figure 5(b) shows a RNN's output signal for $\alpha = 1$, $\beta = 0.9$ and $\gamma = 0.5$, indicating the problem's temporal sensitivity via a short initial transient. After this transient, the network's signal is dispersed around some mean value. However, due to the recurrence output range depends not only on the input signals amplitude range, but also on its temporal characteristic. As before, we drive the system with a random signal uniformly distributed within interval $u^t \in (0, \cdots, 1)$. Figure 5(c) shows the mean output signal as a red line for different strength of the network internal coupling β.

As expected, for β approaching 1 the system destabilizes, and the offset quickly diverges towards infinity (with its pole at $\beta = 1$). In a nonlinear system, this is the point were chaos due to sensitivity to initial conditions would start to arise. However, as we lack nonlinearity, resulting dynamics are not bound through nonlinear stretching and folding. However, we can approximate the amplitude-range of the output if the input signal is known. The mean value of the output signal $E(y_3^t)$ can be found for each time t.

$$\begin{aligned} E(y_3^t) &= E(y_2'^t) + b, \quad \text{where} \\ E(y_2'^t) &= \alpha\Big(\gamma(u^t + b) + \beta E(y_2'^{t-1}) + b\Big). \end{aligned} \tag{19}$$

The full technique is given in Appendix B. Based on this technique, we obtain:

$$\begin{aligned} y_{min} &= min\big(E(y_3^t)\big) \\ y_{max} &= max\big(E(y_3^t)\big). \end{aligned} \tag{20}$$

They will be employed for normalizing output signal y_3^t in order to keep it within the range $y_3^t \in (0, \cdots, 1)$.

A. Correlated and uncorrelated noise

Figure 6(a) and (b) show the SNR for $\beta = \gamma = 0.5$ for exclusively uncorrelated and correlated noise, respectively. Noise intensities are identical to the once of the FNN-section, and the resulting SNR has overall similar properties and scale as the one found for the 4-layer FNN. Again, correlated noise significantly reduces the system's performance when compared to uncorrelated noise, which indicates that the temporal averaging in an RNN suppresses uncorrelated noise in a similar fashion as in deep FNNs.

Increasing β confirms this general observation, see Fig. 6(a) and (d) for exclusively uncorrelated and correlated noise at for $\beta = 0.9$ and $\gamma = 0.5$, respectively. The increased contribution of the recurrent signal has multiple consequences. On one hand β improves the additive SNR, but then on the other hand significantly reduces the multiplicative SNR. Ultimately, the output's SNR is dominated by the latter contribution, and hence increasing β increases the RNN's susceptibility to noisy neurons.

B. Noise in recurrent networks

Here, we face the same challenge as when deriving the RNN's output amplitude range: the analytical SNR description requires some knowledge of the input signal's temporal nature. As before we will focus on the final form of the equations; the complete derivation can be found in Appendix B.

For a noisy RNN, the mean value is (19). The noise-induced variance of its output signal is given by

$$\begin{aligned} \mathrm{Var}(y_3^t) = \sigma_{add}^2 + \sigma_{mult}^2 E^2(y_3^t) + (1 + \sigma_{mult}^2)\times \\ \big[2D_A^C + 2D_M^C E^2(y_2'^t) + (1 + 2D_M^C)\mathrm{Var}(y_2'^t)\big]. \end{aligned} \tag{21}$$

Variables σ_{add}^2 and σ_{mult}^2 are identical to the once previously introduced in Sec. III B. Furthermore, $y_2'^t = y_{2,i}'^t$ contains the averaged signal passed on from the recurrent layer of previous time $(t-1)$ to the output neuron for the case of global coupling. Its variance is

$$\begin{aligned} \mathrm{Var}(y_2'^t) \approx \alpha^2\gamma^2\mathrm{Var}(y_1^t) + \alpha^2\beta^2\big[2D_A^C + \\ 2D_M^C E^2(y_2'^{t-1}) + (1 + 2D_M^C)\mathrm{Var}(y_2'^{t-1})\big]. \end{aligned} \tag{22}$$

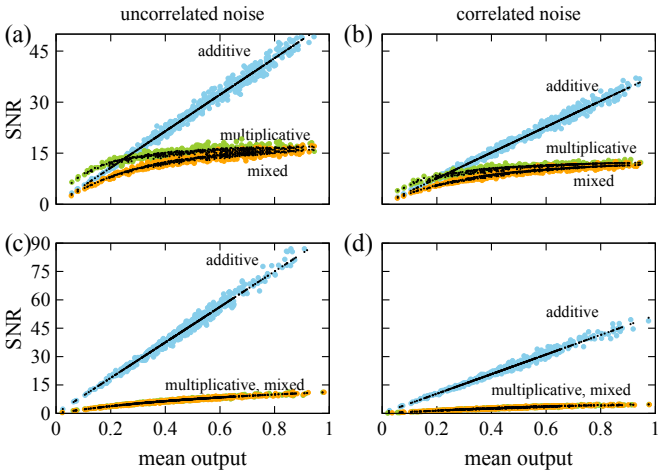

FIG. 6. SNR dependences for uncorrelated noise (left panels) and correlated noise (right panels). Blue points correspond to only additive noise of intensity $D_A^U = D_A^C = 10^{-4}$, green points indicate to only multiplicative noise with intensities $D_M^U = D_M^C = 10^{-3}$, orange points match mixed noise with combination of both noise intensities. Top panels were prepared for $\beta = 0.5$, bottom panels correspond to $\beta = 0.9$. Other parameters are: $\gamma = 0.5$, $I_2 = 200$, $\alpha = 1$.

with mean value

$$E(y_2'^t) = E(y_{2,i}^t) = \alpha\big(\gamma E(y_1^t) + \beta E(y_2'^{t-1}) + b\big) \qquad (23)$$

Again the uncorrelated noise comes only from the input and output neurons. The nodes from the recurrent layer transmit only their correlated noise. It means that, as with FNN, the main noise effect comes from the input and output neuron. Equations (19)–(23) demonstrate good correspondence with the numerical simulation (Fig. 6 black points).

V. NOISE PROPAGATION VERSUS CONNECTIVITY

Previously we considered networks were connectivity in all matrices was 100 % and uniform. This enabled a clean derivation of analytical models, however, it does not correspond to typical topology-statistics of neural networks. In the following sections we will replace the fully connected matrices by matrices consisting of random entries and with a certain fraction of non-zero connections, i.e. connectivity.

A. Connectivity in FNNs

We will again focus on the case of a deep FNN with a single input and output neuron, $I_1 = I_4 = 1$, and hosting with $I_2 = I_3 = 200$ neurons in the hidden layers. Matrices $\mathbf{W}^{in} \in \mathbb{R}^{1\times I_2}$ and $\mathbf{W}^{out} \in \mathbb{R}^{I_3\times 1}$ determine the system's connection to the in and output, respectively. Coupling between hidden layers is given by $\mathbf{W}^3 \in \mathbb{R}^{I_2\times I_3}$. All matrices have a percentage of ρ non-zero entries, which are drawn from a random distribution. Finally, matrices are normalized to their largest eigenvalue. For $\rho = 100$ % we would again obtain global coupling, however now according to random weight distributions.

Figure 7 shows the SNR for uncorrelated and correlated noise in panels (a) and (b). The orange (blue) data have been obtained using $\rho = 1$ % ($\rho = 100$ %) connectivity of $\mathbf{W}^3$, i.e. between the two hidden layers, while $\mathbf{W}^2$ and $\mathbf{W}^4$ were fully connected. The dashed line was obtained based on the analytical SNR description based on Eqs. (15)-(16), crucially dervide under the assumption of symmetric and full connectivity. From the data it is clear that the SNR's dependency on the connectivity between both hidden layers is very weak. Furthermore, for correlated as well as uncorrelated noise the analytical SNR description perfectly agrees with the numerically obtained data. This leads to an interesting conclusion: attenuation of uncorrelated noise is already established by averaging tue to input and readout matrices $\mathbf{W}^2$ and $\mathbf{W}^4$.

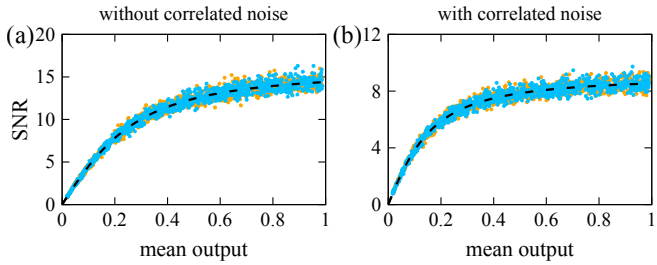

FIG. 7. Signal-to-noise ratio in the last layer of FNN for the case with multiplicative correlated noise of intensity $D_M^C = 10^{-3}$ (b) and without it (a). Other parameters are $D_A^U = 10^{-4}$, $D_M^U = 10^{-3}$. Blue points correspond 100% connection and orange ones to 1% connection between hidden layers. Both dependences are almost the same.

For this reason, we now turn our attention to the SNR's dependency upon the readout connectivity. The corresponding SNR, see Fig. 8 was obtained for $\rho = 1$ % of $\mathbf{W}^3$ and $\rho = 100$ % of $\mathbf{W}^{in}$, while orange (blue) data corresponds to $\rho = 1$ % ($\rho = 100$ %) connectivity for $\mathbf{W}^{out}$, respectively. Only in the case of a final layer connectivity of $\rho = 1$ % we obtain a reduction of the SNR-function for uncorrelated noise, hence a reduced suppression of such. Such a low final layer connectivity is unlikely for most applications.

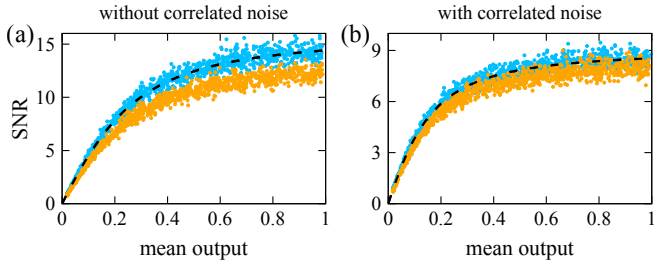

FIG. 8. Signal-to-noise ratio in the last layer of FNN for the case with multiplicative correlated noise with intensity $D_M^C = 10^{-3}$ (b) and without it (a). Other parameters are $D_A^U = 10^{-4}$, $D_M^U = 10^{-3}$. Blue points correspond 100% connection and orange ones to 1% connection in the output matrix $\mathbf{W}^{\text{out}}$. The matrix between hidden layers has 1% connection.

B. Coupling in RNN

Guided by the previous FNN-results we can immediately focus on the connectivity of the RNN's readout layer. Crucially, we have confirmed that, as for the FNN, ρ of the RNN's hidden layer $\mathbf{W}^{out}$ has negligible impact upon the system's SNR. However, there is a fundamental difference when discussing the RNN's connectivity. Besides the percentage of non-zero connections ρ, parameter β greatly influences the system's sensitivity as well as its connection topology, and is therefore included in our considerations.

For each noise-type and final layer connectivity we investigate two β-values. In Fig. 9 we show the SNR for $\beta = 0.5$ and final layer connectivity $\rho = 1$ % ($\rho = 100$ %) as orange (blue) data. Data for $\beta = 0.9$ and final layer connectivity $\rho = 1$ % ($\rho = 100$ %) is shown in gray (green). For uncorrelated noise, panel (a), we find that for both, connectivity as well as β have a strong impact. Again, higher connectivity suppresses propagation of uncorrelated noise, while larger β's increase the system's sensitivity. As expected, we find very little impact of the final layer's connectivity upon the suppression of correlated noise for $\beta = 0.5$. For $\beta = 0.9$ we find that connectivity has no positive effect upon correlated noise suppression; a ρ of the final layer beyond ~ 10 % is sufficient for that purpose.

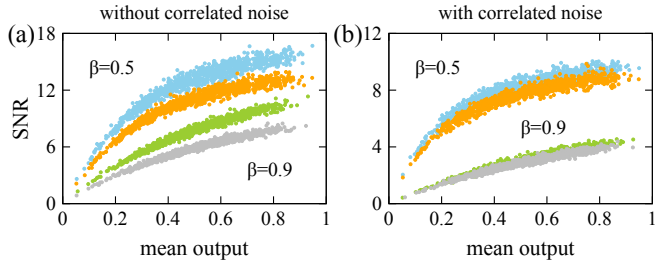

FIG. 9. Signal-to-noise ratio in RNN for the case with multiplicative correlated noise with intensity $D_M^C = 10^{-3}$ (b) and without it (a). Other parameters are $D_A^U = 10^{-4}$, $D_M^U = 10^{-3}$. Blue points ($\beta = 0.5$) and green points ($\beta = 0.9$) correspond 100% connection and orange ($\beta = 0.5$) and gray ($\beta = 0.9$) ones are for 1% connection in the output matrix $\mathbf{W}^{\text{out}}$. The connectivity of the hidden (input) layer is 1 % (100 %).

VI. IMPACT OF DEPTH

The final topological consideration in our work will be the impact of a FNN's and RNN's depth upon the SNR. The feed-forward network in the simplest case has only one hidden layer ($N = 3$), and its SNR for exclusively uncorrelated noise ($D_M^U = 10^{-3}$, $D_A^U = 10^{-4}$) will serve as the reference for deeper FNNs. Figure 11(a) illustrates the SNR's dependency upon the number of layers for five cases: without correlated noise (black data), with additive correlated noise (light red ($D_A^C = 10^{-4}$) and light blue ($D_A^C = 10^{-3}$) data) and with multiplicative correlated noise (dark red ($D_M^C = 10^{-4}$) data and dark blue

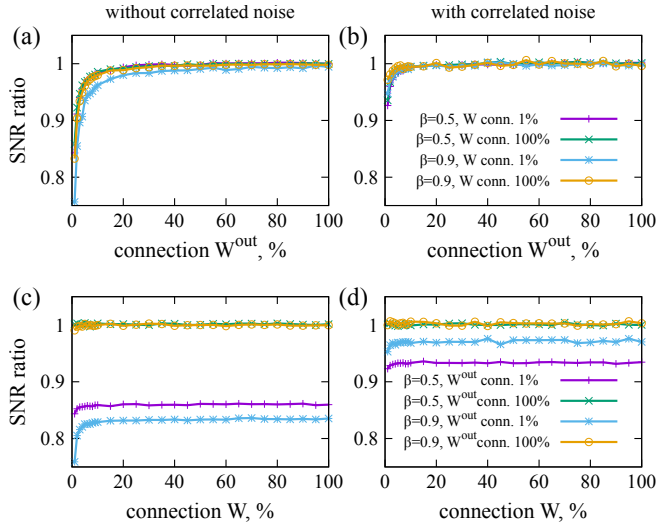

FIG. 10. SNR ratio for RNN. SNR dependences are normalized to SNR for corresponding β and nose intensities but for global coupling in all the matrices.

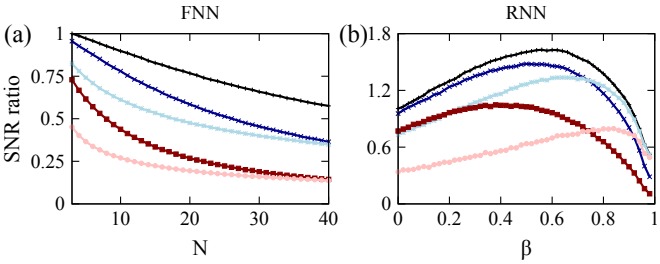

FIG. 11. SNR ratio for different number of layers in FNN (a) and different recurrency impact β in RNN (b). The color scheme is the same for both panels. Black lines correspond to the case without correlated noise. Other lines correspond to different correlated noise intensities: dark blue – $D_M^C = 10^{-4}$, dark red – $D_M^C = 10^{-3}$, light blue – $D_A^C = 10^{-4}$, light red – $D_A^C = 10^{-3}$. Fixed parameters are $D_A^U = 10^{-4}$, $D_M^C = 10^{-4}$, $D_M^U = 10^{-3}$, $\alpha = 1$, $b = 0.02$.

($D_M^C = 10^{-3}$) data). In general, the FNN is more resilient against multiplicative noise, and while the FNN's depth certainly has an impact, it requires approximately 20 layers for the SNR to be reduced to half of the single hidden layer system.

Figure 11(b) illustrates the RNN's dependency upon its internal coupling strength β for the same five configurations of noise. The SNR reference was obtained for the case without correlated noise and $\beta = 0$, black data. As mentioned before, the internal coupling strength can be linked to the depth of a corresponding mapping of the RNN upon an FNN via the unfolding in time technique. For $\beta = 1$ the depth of such a system becomes infinity, and it is hence not surprising that for this case the SNR drops to zero for all noise-configurations. Here we would like to point out a difference we expect for the case of nonlinear RNNs. For $\beta > 1$ the nonlinearity results in characteristic dynamical regimes, and for the case of periodic orbits we expect that the system will still preserve a large degree of its noise-resilience[22]. Apart from the SNR when approaching the critical value of $\beta = 1$ we

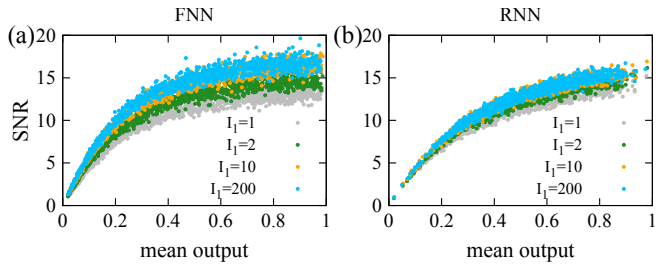

FIG. 12. Signal-to-noise ratio of the output signal in FNN (a) and RNN (b) for the case different numbers of nodes in the input layer and noiseless neurons in the final readout layer. Parameters are $D_A^U = 10^{-4}$, $D_M^C = 10^{-4}$, $D_M^U = 10^{-3}$, $\alpha = 1$, $b = 0.02$, $\beta = 0.5$.

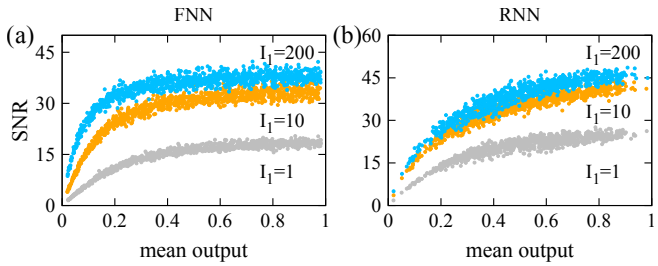

FIG. 13. Signal-to-noise ratio of the output signal in FNN (a) and RNN (b) for the case different numbers of nodes in the input layer and turned off noise in the output layer. Parameters are $D_A^U = 10^{-4}$, $D_M^C = 10^{-4}$, $D_M^U = 10^{-3}$, $\alpha = 1$, $b = 0.02$, $\beta = 0.5$.

find that the system has an ideal operational point for maximum noise mitigation. The particular value of the optimal β depends on the type and amplitude of noise.

VII. STRATEGIES FOR NOISE MITIGATION

In the previous sections we have continuously encountered two essential aspects of a analogue neural networks sensitivity to noise. The first is that uncorrelated noise can efficiently be suppressed through averaging by the numerous network's connections. And second is that the global SNR dependencies of FNN's and RNN's share strong similarities. We will now use these features for devising first strategies for boosting neural network noise mitigation.

The first argument in the previous paragraph leads the way to an intuitive strategy. The fact that the first layer neuron acts as drive for all other neurons morphs this particular element into the ultimate source of globally correlated noise. We therefore attempt to de-correlate this impact and increase the number of input neurons where each of the input neurons receives the same input signal. Figure 12 illustrates the system's SNR for identical noise conditions for different $1 \leq I_1 \leq 200$. Panel (a) shows the impact of this strategy upon a FNN, panel (b) for a RNN ($\gamma = \beta = 0.5$). For the FNN, the impact of increasing I_1 is significant and improve the SNR by ~ 30 % for $I_1 \gtrsim 10$. For the RNN we do not find significant impact.

The relevance of the last layer is due to the fact that fundamentally there its no further averaging taking place after. The most immediate conclusion is therefore to place substantial effort on low-noise last layer neurons in the fabrication and design of an analogue neural networks. In the case of the systems as in[10], this corresponds to the selection of a low-noise detector in the output. Figure 13 shows the great effectiveness of this strategy. In both cases, for the FNN, panel (a), and RNN, panel (b) ($\gamma = \beta = 0.5$), the benefit upon the system's SNR is significant when combined with multiplexing of the first layer input neurons. Based on noise-less readout neurons and duplicating the input-neuron to around 10 copies approximately doubles the system's SNR. Most importantly, we have also analysed the situation for only uncorrelated noise. There, the SNR can be increased close to 10 times following this strategy.

In some types of networks, it is impossible to alter the neuron type. Then the idea of turned off noise can be transformed to the next way. Previously the feed-forward network with four layers of nodes I_1, I_2, I_3, I_4 with $I_4 = 1$ has been considered. Now we remove the last layer and read out all the signal from each node of the 3rd layer. Then we average this set of signals over the number of nodes in the third layer I_3 and get almost the same effect of noise reduction. The same can be made for RNN.

VIII. CONCLUSION

To conclude, we have analysed the interplay between noise and network connections in great detail. We have considered a large variety of networks topologies and parameters are have analysed their output susceptibility to noisy neurons. We find that the in general the networks are quite resilient towards such noise units.

The principle results of our work are confirmed by analytical descriptions. They identify the fundamental laws with govern the propagation of noise across future hardware implementations. Based on the gain insight identify noise which is correlated across populations of neurons together with the noise present in input and output neurons as the main nuisance in such systems. Our work identifies the fundamental importance of stability in global parameters, for example in the stability of a analogue neural network's power supply. Individual neurons, on the other hand, can be of lesser quality as their local noise will largely be averaged out by the network itself.

We directly leverage this insight and propose noise mitigation strategies which have a great effect. Under good conditions this approach can results into an SNR improvement by up to one order of magnitude.

Our work focused on networks of linear components. By that it highlighted the fundamental interaction between noisy neurons and network connections. However, we presented a large fraction of our results as a SNR dependency against the average output amplitude. This allows for a first, heuristic mapping of our findings onto

nonliear system's for which the nonlinearity is known. An important task is now to include such nonlinear noisy neurons and to understand their role inside the noisy orchestra in detail.

ACKNOWLEDGMENTS

The authors acknowledge the support of the Region Bourgogne Franche-Comté. This work was supported by the EUR EIPHI program (Contract No. ANR-17-EURE-0002), the BiPhoProc project (Contract No. ANR-14-OHRI- 0002-02), by the Volkwagen Foundation (NeuroQNet). N.S. is supported by Vernadski scholarship of the French Government. X.P. has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 713694 (MULTIPLY).

Appendix A: Analytical prediction of SNR in FNN

In this section the analytical prediction of SNR in FNN is obtained. The next formulas are valid only for three main conditions: (i) the coupling between layers is global; (ii) the number of nodes in hidden layer is large $I_n \gg 1$; (iii) all noise sources ξ have zero mean value and 1 variance. The statistical characteristics of noise are controlled by corresponding noise intensities.

In the present manuscript we consider the case when the function $f(x) = \alpha x$ affects only in hidden layers. Following the nomenclature of Sec. III the evolution equation for the first layer $y_1'^t$ and the value after noise influence y_1^t are described by

$$\begin{aligned} &y_1'^t = u^t + b \\ &y_1^t = y_1'^t \cdot (1 + \sqrt{2D_M^C}\xi_{M1}^C)(1 + \sqrt{2D_M^U}\xi_{M1}^U) + \\ &\sqrt{2D_A^C}\xi_{A1}^C + \sqrt{2D_A^U}\xi_{A1}^U. \end{aligned} \tag{A1}$$

The expected value for the first layer is $E(y_1^t) = u^t + b$. The corresponding variance can be found using three main rules of random variables algebra: (i) the variance of sum is $Var(\xi + \eta) = Var(\xi) + Var(\eta)$, where ξ and η are random variables which are not correlated to each other. (ii) the variance of their multiplication is $Var(\xi \cdot \eta) = (E^2(\eta) + Var(\eta)) \cdot Var(\xi) + E^2(\xi)Var(\eta)$. (iii) the multiplication by some constant has the variance $Var(c\xi) = c^2 Var(\xi)$. Then the variance in the first layer is:

$$Var(y_1^t) = \sigma_{add}^2 + \sigma_{mult}^2 (u^t + b)^2, \tag{A2}$$

where $\sigma_{add}^2 = 2D_A^C + 2D_A^U$ shows the variance coming from only additive noises and $\sigma_{mult}^2 = 2D_M^C + 2D_M^U + 4D_M^C D_M^U$ is the variance of the multiplier with multiplicative noises after $y_1'^t$.

The general noisy equation for each layer n has a common form

$$\begin{aligned} y_{n,i}^t = y_{n,i}'^t \cdot (1 + \sqrt{2D_M^C}\xi_{M1}^C)(1 + \sqrt{2D_M^U}\xi_{M1}^U) + \\ \sqrt{2D_A^C}\xi_{A1}^C + \sqrt{2D_A^U}\xi_{A1}^U, \quad 1 \le n \le N \end{aligned} \tag{A3}$$

and depends on the state without noise $y_{n,i}'^t$. The coupling is global between hidden layers. Then each ith neuron of the nth layer receives the same signal from the previous one. Therefore the index i can be neglected in the equation for $y_{n,i}'^t$:

$$\begin{aligned} y_n'^t = y_{n,i}'^t = \alpha\big[\tfrac{1}{I_{n-1}} \cdot \textstyle\sum_{j=1}^{I_{n-1}} y_{n-1,j}^t + b\big] = \\ \alpha\big[b + \sqrt{2D_A^C}\xi_{An-1}^C + y_{n-1}'^t(1 + \sqrt{2D_M^C}\xi_{Mn-1}^C) \cdot \\ \big(1 + \tfrac{1}{I_{n-1}} \textstyle\sum_{j=1}^{I_{n-1}} \sqrt{2D_M^U}\xi_{Mn-1,j}^U\big) + \\ \tfrac{1}{I_{n-1}} \textstyle\sum_{j=1}^{I_{n-1}} \sqrt{2D_A^U}\xi_{An-1,j}^U\big]. \end{aligned} \tag{A4}$$

Roughly speaking, the variance of the sum divided by I_n equals to the sum of variances divided by I_n^2. At $I_n \gg 1$ these components are much smaller than the rest, so they can be ignored. They have no impact on expected values and variances.

$$y_n'^t \approx \alpha\big[b + y_{n-1}'^t\big(1 + \sqrt{2D_M^C}\xi_{Mn-1}^C\big) + \sqrt{2D_A^C}\xi_{An-1}^C\big]. \tag{A5}$$

It is clearly seen that the equation (A5) contains only correlated noise. Only this type of noise goes from the hidden layers to the output signal. The expected value for hidden layer is

$$E(y_n'^t) = E(y_{n,i}^t) = \alpha(b + E(y_{n-1}^t)), \quad 1 < n < N \tag{A6}$$

The corresponding variance is

$$Var(y_n'^t) \approx \alpha^2\big[2D_A^C + 2D_M^C E^2(y_{n-1}'^t) + (1 + 2D_M^C)Var(y_{n-1}'^t)\big]. \tag{A7}$$

The equations (A5), (A7) are valid only for the hidden layers $2 < n < N$. The second layer has a small difference because the previous 1st layer has only one node and there is no average over I_1. Both correlated and uncorrelared noise from the first layer propagates through the network.

$$y_2'^t = y_{2,i}'^t = \alpha(y_1^t + b) \tag{A8}$$

$$Var(y_2'^t) = \alpha^2 \cdot Var(y_1^t). \tag{A9}$$

The expected value in the 2nd layer is described by (A6). The state equation of the last Nth layer is

$$y_N'^t = \frac{1}{I_{N-1}} \cdot \sum_{j=1}^{I_{N-1}} y_{N-1,j}^t + b. \tag{A10}$$

And the corresponding variance can be calculated using (A5) but without α. The noisy equation (A3) occurs also for $n = N$. Then the variance for the output signal is

$$\begin{aligned} Var(y_N^t) \approx \sigma_{add}^2 + E^2(y_N^t)\sigma_{mult}^2 + (1 + \sigma_{mult}^2) \cdot Var(y_N'^t) = \\ \sigma_{add}^2 + E^2(y_N^t)\sigma_{mult}^2 + (1 + \sigma_{mult}^2)\big[2D_A^C + \\ 2D_M^C E^2(y_{N-1}'^t) + (1 + 2D_M^C)Var(y_{N-1}'^t)\big]. \end{aligned} \tag{A11}$$

The last term with $[\cdots]$ contains the noisy impact from the previous (N-1)th layer. The variable $Var(y_{N-1}'^t)$ can be replaced by the same term but with noise from the

(N-2)th layer and so on until $Var(y_2'^t)$. The variance for the second layer is calculated using (A9). In the end the final variance of the output signal contains correlated and uncorrelated noise from the first and the last layer and only correlated noise from hidden layers.

The expected value of the output signal is

$$E(y_N^t) = E(y_N'^t) = E(y_{N-1}'^t) + b. \tag{A12}$$

The signal-to-noise ratio can be found as follows:

$$SNR(y_N^t) = \frac{E(y_N^t)}{Var\big(y_{N-1}^t\big)^{1/2}}. \tag{A13}$$

Appendix B: Analytical prediction of SNR in RNN

This section is devoted to the prediction of SNR dependence in the recurrent network. Three conditions are used here as for FNN: (i) global coupling between layers; (ii) the number of nodes in the recurrent layer is large $I_2 \gg 1$; (iii) the input and output layer has only one node.

The structure of RNN is very similar to FNN and the noise has the same affect. So some of the equations will be the same as in Appendix A.

The node in the first layer before the noise influence is described by the equation for $y_1'^t$ and by y_1^t after it.

$$\begin{array}{l} y_1'^t = u^t + b \\ y_1^t = y_1'^t \cdot (1 + \sqrt{2D_M^C}\xi_{M1}^C)(1 + \sqrt{2D_M^U}\xi_{M1}^U) + \\ \sqrt{2D_A^C}\xi_{A1}^C + \sqrt{2D_A^U}\xi_{A1}^U. \end{array} \tag{B1}$$

Then the expected value and variance in the input layer are:

$$\begin{array}{c} E(y_1^t) = E(y_1'^t) = u^t + b \\ Var(y_1^t) = \sigma_{add}^2 + \sigma_{mult}^2 E^2(y_1^t), \end{array} \tag{B2}$$

where $\sigma_{add}^2 = 2D_A^C + 2D_A^U$, $\sigma_{mult}^2 = 2D_M^C + 2D_M^U + 4D_M^C D_M^U$. It is the same as for FNN.

Due to the global coupling the state equation for recurrent layer before the noise influence is the same for each ith node:

$$y_2'^t = y_{2,i}'^t = \alpha\big(\gamma y_1^t + \beta \frac{1}{I_2}\sum_j^{I_2} y_{2,j}^{t-1} + b\big). \tag{B3}$$

The equation after noise influence is:

$$\begin{array}{l} y_{2,i}^t = y_2'^t \cdot (1 + \sqrt{2D_M^C}\xi_{M2}^C)(1 + \sqrt{2D_M^U}\xi_{M2,i}^U) + \\ \sqrt{2D_A^C}\xi_{A2}^C + \sqrt{2D_A^U}\xi_{A2,i}^U. \end{array} \tag{B4}$$

Then the equation for the state without noise can be transformed as:

$$\begin{array}{l} y_2'^t = \alpha(\gamma y_1^t + b) + \alpha\beta\big[\sqrt{2D_A^C}\xi_{A2}^C + \\ \frac{1}{I_2}\sum_{j=1}^{I_2}\sqrt{2D_A^U}\xi_{A2,j}^U + y_2'^{t-1}(1 + \sqrt{2D_M^C}\xi_{M2}^C)\times \\ \big(1 + \frac{1}{I_2}\sum_{j=1}^{I_2}\sqrt{2D_M^U}\xi_{M2,j}^U\big)\big] \approx \alpha(\gamma y_1^t + b) + \\ \alpha\beta\big[\sqrt{2D_A^C}\xi_{A2}^C + y_2'^{t-1}(1 + \sqrt{2D_M^C}\xi_{M2}^C)\big]. \end{array} \tag{B5}$$

It is the recurrent formula. The equation contains all the noise from the input node and only correlated noise from the recurrent layer at time $(t-1)$. The corresponding expected value and variance are:

$$E(y_2'^t) = E(y_{2,i}^t) = \alpha\big(\gamma E(y_1^t) + \beta E(y_2'^{t-1}) + b\big) \tag{B6}$$

$$\begin{array}{l} Var(y_2'^t) \approx \alpha^2\gamma^2 Var(y_1^t) + \alpha^2\beta^2\big[2D_A^C + \\ 2D_M^C E^2(y_2'^{t-1}) + (1 + 2D_M^C)Var(y_2'^{t-1})\big]. \end{array} \tag{B7}$$

It is the recurrent formula working for $1 < t \le T$ until $t = 1$. The initial state for t=1 has

$$\begin{array}{ll} E(y_2'^1) = \alpha\big(\gamma E(y_1^1) + b\big) = \alpha\gamma(u^1 + b) + \alpha b \\ Var(y_2'^1) = \alpha^2\gamma^2 Var(y_1^1) & \text{for } t = 1. \end{array} \tag{B8}$$

Finally the output neuron before noise impact has the state equation:

$$\begin{array}{l} y_3'^t = \frac{1}{I_2}\sum_{j=1}^{I_2} y_{2,j}^t + b \approx b + \sqrt{2D_A^C}\xi_{A2}^C + \\ \qquad\qquad y_2'^t(1 + \sqrt{2D_M^C}\xi_{M2}^C). \end{array} \tag{B9}$$

Its variance is

$$Var(y_3'^t) = 2D_A^C + 2D_M^C E^2(y_2'^t) + (1 + 2D_M^C)Var(y_2'^t). \tag{B10}$$

The output signal y_3^t with noise is described by similar equation to (B1). The expected value and variance of the output signal at time t is:

$$E(y_3^t) = E(y_3'^t) = E(y_2'^t) + b \tag{B11}$$

$$\begin{array}{l} Var(y_3^t) = \sigma_{add}^2 + \sigma_{mult}^2 E^2(y_3^t) + (1 + \sigma_{mult}^2)Var(y_3'^t) \approx \\ \sigma_{add}^2 + \sigma_{mult}^2 E^2(y_3^t) + (1 + \sigma_{mult}^2)\times \\ \big[2D_A^C + 2D_M^C E^2(y_2'^t) + (1 + 2D_M^C)Var(y_2'^t)\big]. \end{array} \tag{B12}$$

The term in $[\cdots]$ brackets shows the noise impact from the recurrent layer in time t. It is described by the recurrent formula (B7) with the noise impact from all the previous time moments.

Renormalization

The range of the output signal in RNN depends not only on the range of the input signal but also on the input sequence itself. The prediction of output limits can be prepared only for known input signal. The mean output signal is (B11) where the mean of recurrent layer $E(y_2'^t)$ can be found using the recurrent formula (B6) with initial state (B8). It doesn't reliant on noise intensities, but strongly depend on the input sequence. To keep the same scale one can calculate the limits for the renormalization is:

$$\begin{array}{ll} y_{min} = min\big(E(y_3^t)\big), & t_0 < t \le T \\ y_{max} = max\big(E(y_3^t)\big), & t_0 < t \le T \end{array} \tag{B13}$$

starting from some transient time t_0. In numerical simulation and analytics the renormalization is used:

$$\big(y_3^t\big)_R = \frac{y_3^t - y_{min}}{y_{max} - y_{min}}. \tag{B14}$$

The variable y_3^t here is multiplied by $1/(y_{max} - y_{min})$. Then the analytical formula for the output variance in the case of normalization:

$$Var\left(y_3^t\right)_R = Var(y_3^t)/(y_{max} - y_{min})^2. \tag{B15}$$

[1] J. Hopfield, Proceedings of the national academy of ... **79**, 2554 (1982).

[2] W. S. McCulloch and W. Pitts, The Bulletin of Mathematical Biophysics (1943), 10.1007/BF02478259.

[3] S. Utsunomiya, K. Takata, and Y. Yamamoto, Optics express **19**, 18091 (2011).

[4] Y. LeCun, Y. Bengio, and G. Hinton, Nature **521**, 436 (2015).

[5] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, R. Boyle, P.-l. Cantin, C. Chao, C. Clark, J. Coriell, M. Daley, M. Dau, J. Dean, B. Gelb, T. V. Ghaemmaghami, R. Gottipati, W. Gulland, R. Hagmann, C. R. Ho, D. Hogberg, J. Hu, R. Hundt, D. Hurt, J. Ibarz, A. Jaffey, A. Jaworski, A. Kaplan, H. Khaitan, A. Koch, N. Kumar, S. Lacy, J. Laudon, J. Law, D. Le, C. Leary, Z. Liu, K. Lucke, A. Lundin, G. MacKean, A. Maggiore, M. Mahony, K. Miller, R. Nagarajan, R. Narayanaswami, R. Ni, K. Nix, T. Norrie, M. Omernick, N. Penukonda, A. Phelps, J. Ross, M. Ross, A. Salek, E. Samadiani, C. Severn, G. Sizikov, M. Snelham, J. Souter, D. Steinberg, A. Swing, M. Tan, G. Thorson, B. Tian, H. Toma, E. Tuttle, V. Vasudevan, R. Walter, W. Wang, E. Wilcox, and D. H. Yoon, arXiv:1704.04760 , 1 (2017).

[6] D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, Nature communications **4**, 1364 (2013).

[7] T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, and E. Eleftheriou, Nature Nanotechnology **11**, 693699 (2016).

[8] J. Torrejon, M. Riou, F. A. Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. D. Stiles, and J. Grollier, Nature **547**, 428 (2017).

[9] D. Psaltis, D. Brady, X.-G. Gu, and S. Lin, Nature **343**, 325 (1990).

[10] J. Bueno, S. Maktoobi, L. Froehly, I. Fischer, M. Jacquot, L. Larger, and D. Brunner, Optica **5**, 756 (2018).

[11] X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, M. Jarrahi, and A. Ozcan, Science **26**, 1 (2018).

[12] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljacic, Nature Photonics **11**, 441446 (2017).

[13] A. N. Tait, T. F. De Lima, E. Zhou, A. X. Wu, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, Scientific Reports **7**, 1 (2017).

[14] G. Van der Sande, D. Brunner, and M. C. Soriano, Nanophotonics **6**, 561 (2017).

[15] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, Neural Networks **115**, 100 (2019).

[16] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, Arxiv (2017).

[17] J. Hasler and B. Marr, Frontiers in neuroscience **7**, 118 (2013).

[18] R. Sarpeshkar, Neural Computation **10**, 1601 (1998).

[19] A. Murray, Electronics Letters **27**, 1546 (1991).

[20] H. Jaeger, "Short term memory in echo state networks," Tech. Rep. (German National Research Institute for Computer Science, 2002).

[21] R. J. Williams, "Complexity of exact gradient computation algorithms for recurrent neural networks," Tech. Rep. (Boston: Northeastern University, College of Computer Science., 1989).

[22] B. A. Marquez, L. Larger, M. Jacquot, Y. K. Chembo, and D. Brunner, Scientific Reports **8**, 3319 (2018).

[23] V. Kimelblat, Fluctuation and Noise Letters **10**, 181 (2011).

[24] J. Bueno, D. Brunner, M. Soriano, and I. Fischer, Optics Express **25**, 2401 (2017).

[25] B. Penkovsky, X. Porte, M. Jacquot, L. Larger, and D. Brunner, Arxiv (2019).

[26] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljacic, Nature Photonics **11**, 441446 (2017).

[27] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, Proceedings of the 32nd International Conference on International Conference on Machine Learning **37**, 1737 (2015).

[28] S. Wakelin and A. C. Walker, Physics Education **29**, 155 (1994).

[29] R. D. Hof, MIT Technology Review (2014).

[30] M. C. Soriano, S. Ortín, D. Brunner, L. Larger, C. R. Mirasso, I. Fischer, and L. Pesquera, Optics express **21**, 12 (2013).

[31] D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, Nature communications **4**, 1364 (2013).

[32] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer, Optics express **20**, 3241 (2012).

[33] S. Scardapane and D. Wang, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **7**, e1200 (2017).

[34] B. K. Jenkins, A. A. Sawchuk, T. C. Strand, R. Forchheimer, and B. H. Soffer, Applied Optics **23**, 3455 (1984).

[35] J. Hopfield and D. Tank, Biological cybernetics **52**, 141 (1985).