

Noise mitigation strategies in physical feedforward neural networks

N. Semenova^{1,2} and D. Brunner¹

¹*Département d'Optique P. M. Duffieux, Institut FEMTO-ST, Université Bourgogne-Franche-Comté CNRS UMR 6174, Besançon, France*

²*Institute of Physics, Saratov State University, 83 Astrakhanskaya str., 410012 Saratov, Russia*

(*Electronic mail: daniel.brunner@femto-st.fr)

(*Electronic mail: semenovani@sgu.ru)

(Dated: 19 May 2022)

Physical neural networks are promising candidates for next generation artificial intelligence hardware. In such architectures, neurons and connections are physically realized and do not leverage digital concepts with their practically infinite signal-to-noise ratio to encode, transduce and transform information. They therefore are prone to noise with a variety of statistical and architectural properties, and effective strategies leveraging network-inherent assets to mitigate noise in an hardware-efficient manner are important in the pursuit of next generation neural network hardware. Based on analytical derivations, we here introduce and analyse a variety of different noise-mitigation approaches. We analytically show that intra-layer connections in which the connection matrix's squared mean exceeds the mean of its square fully suppresses uncorrelated noise. We go beyond and develop two synergistic strategies for noise that is uncorrelated and correlated across populations of neurons. First, we introduce the concept of *ghost neurons*, where each group of neurons perturbed by correlated noise has a negative connection to a single neuron, yet without receiving any input information. Secondly, we show that pooling of neuron populations is an efficient approach to suppress uncorrelated noise. As such, we developed a general noise mitigation strategy leveraging the statistical properties of the different noise terms most relevant in analogue hardware. Finally, we demonstrate the effectiveness of this combined approach for trained neural network classifying the MNIST handwritten digits, for which we achieve a 4-fold improvement of the output signal-to-noise ratio and increase the classification accuracy almost to the level of the noise-free network.

I. INTRODUCTION

During the past years, neural networks (NNs) have provided solutions to previously unsolvable computing problems¹. Among others, these tasks include image recognition and classification^{2,3}, improvement of sound recordings, speech recognition⁴ and prediction of climatic phenomena⁵. The basic principle of NNs is signal propagation between nonlinear neurons along connections according to some connection coefficients or connection weights. Among the most pressing objectives today is to implement NN topologies in hardware that drastically reduces the energy consumption compared to current NN hardware, and research activity along these lines has lately exploded. Special purpose NN chips, i.e. the newest generation of tensor and graphic processing units, allow low (2-6 bit) resolution computing⁶.

Combined with the need for removing the von Neumann bottleneck, the interest into low precision digital NN computing actually suggest analogue implementations of NN, i.e. in-memory computing leveraging computing with physical neural networks^{7,8}, as promising substrates. At current digital resolutions for NN computing, analogue implementations substantially profit from the favorable energy usage per unit of information given by fundamental thermodynamics⁹. Physical NNs target encoding a NN's topology in a tunable analogue circuit, for example in electronic¹⁰⁻¹² and photonic systems¹³. Physical NNs leveraging lasers¹⁴⁻¹⁸, and spin-torque oscillators¹⁹ as neurons have been demonstrated. A physical NN's connections have been realized using holography²⁰, diffraction^{21,22}, integrated networks of Mach-Zender modulators²³, wavelength division multiplexing²⁴,

and 3D printed optical interconnects²⁵⁻²⁷. Such, analog NN hardware is fundamentally prone to noise, and previous works provide strategies for reducing an analogue physical neuron's noise specific for the particular hardware²⁸⁻³². Previously, we derived analytical descriptions of noise propagation and potential accumulation in deep NNs^{33,34}. The analytic equations describing the signal to noise ratio (SNR) at the output of a physical NN identified the most relevant sources of noise as well as strategies for effective noise suppression. Here, we introduce and discuss several approaches of noise mitigation that are tailored to mitigate the most relevant generic types of noise. Importantly, individual strategies can be combined into a general noise mitigation framework that is adjustable to the particularities of a specific NN hardware architecture.

First, we discuss which sections of NNs are most affected by particular noise types, which is followed by analytically describing how one can leverage statistical properties of a NNs connectivity matrices to reduce noise simply by means of a noise-optimized topology. Next, we go beyond pure statistics-based strategies and introduce *ghost neurons*. A ghost neuron is a single neuron per layer that does not receive any input, and whose output is subtracted from each neuron in this layer in order to remove correlated additive noise. Furthermore, we discuss the impact of pooling neuron populations within layers, i.e. combining several neurons receiving the same input into one 'macro' neuron. Averaging the outputs of its individual elements, the macro neuron has reduced sensitivity to both types of uncorrelated noise. Finally, we apply the suggested noise mitigation techniques to reduce noise in NN trained to recognize MNIST digits database, where we achieve an excellent 4-fold suppression of noise at the final

output layer of the 3 layer NN.

II. SYSTEM UNDER STUDY

Our work focuses on deep feed-forward neural networks (FNNs). These are networks consisting of a linear input and output layer, plus potentially several hidden layers, and information propagates strictly uni-directional from a preceding to a following layer. A schematic illustration of such a FNN is shown in Fig. 1(a). The input layer comprising I_1 linear neurons receives input according to vector $\vec{u}(t)$, while the output layer with I_3 linear or nonlinear neurons provides output vector $\vec{y}^{\text{out}}(t)$. Here, we generally consider one hidden layer with $I_2 = 100$ neurons with $f(\cdot)$ as their nonlinear activation function. The connection topology between layers n and $(n+1)$ is captured by connection matrix \mathbf{W}^n that is of dimension $I_n \times I_{n+1}$. Then the signals coming to neurons belonging to layer $(n+1)$ are \vec{a}_{n+1} , and after activation function they transform to the noise-less signals \vec{x}_{n+1} :

$$\vec{a}_{n+1} = \mathbf{W}^n \cdot \vec{y}_n, \quad \vec{x}_{n+1} = f(\vec{a}_{n+1}), \quad (1)$$

where \vec{y}_n is the noisy signal from layer n . If noise is turned off then $\vec{y}_n = \vec{x}_n$

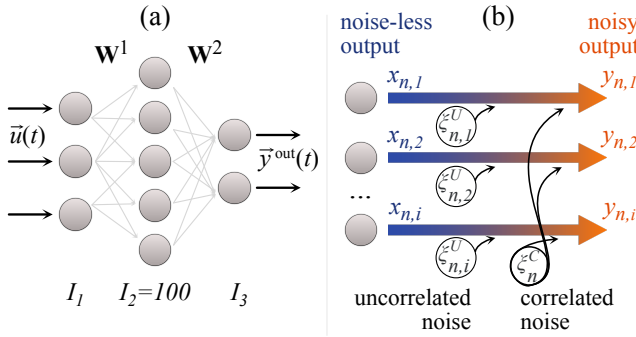


FIG. 1. (a) Schematic representation of a feed-forward neural network, and (b) how uncorrelated and correlated noise is introduced into neurons.

Thus, we come to the main aspect of this article: the mitigation of noise and avoiding its accumulating as information propagates to the physical NN's output $\vec{y}^{\text{out}}(t)$. Previously, we analytically captured the general impact of noise on FNNs with linear³³ and nonlinear neurons that were trained with error back propagation³⁴. Here, we substantially extend our analysis and derive noise reduction strategies. Here, noise is introduced identical as in^{33,34}, and we include additive and multiplicative noise, which are the most common types of noise found in analogue hardware. The signals of noisy neurons i in layer n are

$$\begin{aligned} y_{n,i} &= x_{n,i} + \sqrt{2D_A} \cdot \xi_{n,i}^A(t) \quad \text{additive noise,} \\ y_{n,i} &= x_{n,i} \cdot (1 + \sqrt{2D_M} \cdot \xi_{n,i}^M(t)) \quad \text{multiplicative noise,} \end{aligned} \quad (2)$$

where indices A and M indicate the noise type. ξ is the white Gaussian noise source with zero mean and unity variance, whose variance is controlled by noise intensity D as

$\text{Var}[\sqrt{2D} \cdot \xi_{n,i}(t)] = 2D$. We will denote $E[\cdot]$ as the expected value and $\text{Var}[\cdot]$ as variance of a random variable. The expected value of neuron's noisy output coincides with its noise free value $E[y_{n,i}] = E[x_{n,i}]$. The variance of signal with additive or multiplicative noise is $\text{Var}[y_{n,i}] = 2D_A + \text{Var}[x_{n,i}]$ and $\text{Var}[y_{n,i}] = 2D_M \cdot (E^2[y_{n,i}] + \text{Var}[x_{n,i}])$, respectively. Without noise-contamination in previous layers, both variances become $2D_A$ or $2D_M \cdot E^2[y_{n,i}]$ ³⁴.

Furthermore, noise can be correlated or uncorrelated across numbers of neurons, such as all neurons in one layer. We use indices 'C' and 'U' to label these two features, see schematic illustration in Fig. 1(b). Combining all four noise types leads to the general description for the output of the i th neuron in layer n :

$$y_{n,i}(t) = \sqrt{2D_A^C} \xi_{n,i}^{C,A}(t) + \sqrt{2D_A^U} \xi_{n,i}^{U,A}(t) + x_{n,i}(t) \cdot \left(1 + \sqrt{2D_M^C} \xi_{n,i}^{C,M}(t)\right) \left(1 + \sqrt{2D_M^U} \xi_{n,i}^{U,M}(t)\right). \quad (3)$$

To characterize the noise level in numerical simulation, we use SNR, calculated as a ratio between expected value of the output signal and corresponding standard deviation or square root of variance³⁵: $\text{SNR}[\vec{y}^{\text{out}}] = E[\vec{y}^{\text{out}}] / \sqrt{\text{Var}[\vec{y}^{\text{out}}]}$. In order to numerically determine the SNR, we repeat the same input signal $K = 300$ times to calculate mean and standard deviation for each entry in the noise-less input sequence.

III. PRINCIPLES OF NETWORK TOPOLOGY AND NOISE ACCUMULATION

A. Linear vs. nonlinear FNNs

Nonlinearity can have a significant impact on noise propagation. In³³, we showed that the FNN similar to Fig. 1(a) but with only linear neurons results in SNR curves as in Fig. 2(a) for additive (blue), multiplicative (orange) and mixed (green) uncorrelated noise. For FNNs with nonlinear neurons³⁴, the SNR relationship intimately depends on particularities of the nonlinear activation functions, see Fig. 2(b) using the same color scheme. For both cases, the properties of mixed noise (additive & multiplicative) is the superposition of both individual dependencies. The main overall result was that correlated noise accumulates stronger than uncorrelated noise. If, for example, connections are global and highly uniform, uncorrelated noise is essentially suppressed through averaging across the many connections.

B. Input and output layers

Highly relevant for a physical NN noise are its in and output layers^{33,34}. In particular for a single input neuron, i.e. scalar input information, all noise present at the input drives responses in the following layers, and can therefore not be suppressed through averaging. Similarly, noise-suppression through averaging along many network connections is impossible at the FNN's output, and noise in readout neurons is

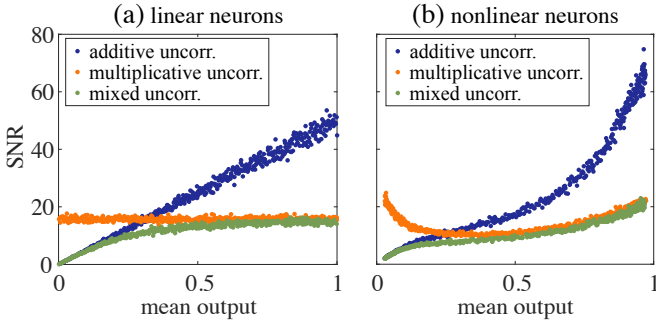


FIG. 2. SNR in two FNNs which are schematically shown in Fig. 1(a) with one linear neurons in the input and output layers $I_1 = I_3 = 1$. Neurons in hidden layer $I_2 = 100$ are linear in panel (a) and nonlinear with sigmoid activation function $f(x) = 1/(1 + \exp(-7(x - 0.5)))$. Figures are prepared for additive (blue dots), multiplicative (orange) and mixed (green) noise with intensities $D_A^U = 10^{-4}$, $D_M^U = 10^{-3}$

another major influence³³. Placing relatively more resources to reduce hardware noise in the input and output layer is therefore an important guide of physical NN hardware design. However, such 'special' in and output neurons might not always be feasible or economic, or the attainable performance might be not sufficient for particular settings. We therefore propose several techniques that allow to further reduce noise accumulation without changing the properties of neurons themselves.

C. Impact of intra-hidden layer connection topology

In³⁴, we considered trained FNNs and developed the analytical treatment enabling the accurate prediction of noise. Importantly, our analytics show that accumulation of different noise types is greatly influenced by the connection matrices' statistics. Details of the analytical derivation can be found in Appendix .

Noise propagation and accumulation is greatly influenced by the squared mean

$$\mu^2(\mathbf{W}^n) = \left(\frac{1}{I_n I_{n+1}} \sum_{i,j} W_{i,j}^n \right)^2 \quad (4)$$

and the mean of the square

$$\eta(\mathbf{W}^n) = \frac{1}{I_n I_{n+1}} \sum_{i,j} (W_{i,j}^n)^2 \quad (5)$$

of connection matrix \mathbf{W}^n . A hidden layer's noise-induced variance is determined by, both, noise in the current as well as by noise coming from previous layers. The impact of correlated noise in the current layer scales according to

$$I_n^2 \cdot \mu^2(\mathbf{W}^n), \quad (6)$$

while the impact of uncorrelated noise *and* the noise from the previous layer scales according to

$$I_n \cdot \eta(\mathbf{W}^n), \quad (7)$$

see Ref.³⁴ and Appendix. There, by changing the statistics of \mathbf{W}^n , we can therefore greatly influence the accumulation of noise.

Figure 3 shows the numerical results leveraging our findings. Here, we focus on the relevant aspects by only considering a FNN schematically illustrated in Fig. 3. The layer consists of $I = 100$ nonlinear neurons, and at each time iteration they receive the same input signal $u(t)$ randomly drawn from the interval $[0;1]$. All neurons exhibit the same noisy additive and multiplicative noise that is in parts correlated as well as uncorrelated, parameters are given in the caption of Fig. 3. This noisy layer is connected to a single linear and noiseless output neuron according to connection matrix \mathbf{W} .

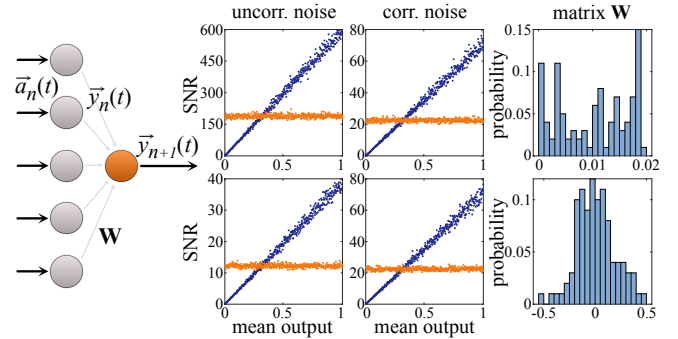


FIG. 3. SNR for different noise intensities and connection matrices \mathbf{W} . Blue dots show the SNR curves with only additive noise, while orange dots are prepared for only multiplicative noise. The top panels correspond to the matrix with $I\mu^2(\mathbf{W}) > \eta(\mathbf{W})$, namely $I\mu^2(\mathbf{W}) = 0.0103$, $\eta(\mathbf{W}) = 1.44 \cdot 10^{-3}$. The bottom panel correspond to the opposite case when $I\mu^2(\mathbf{W}) < \eta(\mathbf{W})$, namely $I\mu^2(\mathbf{W}) = 0.0101$, $\eta(\mathbf{W}) = 0.0340$. Noise intensities are $D_A^U = D_A^C = 10^{-4}$, $D_M^U = D_M^C = 10^{-3}$, $I = 100$.

Figure 3 shows SNR curves for additive (blue) and multiplicative (orange) noise sources for two statistically different connection matrices. For a matrix for which $I\mu^2(\mathbf{W}) > \eta(\mathbf{W})$ the accumulation of uncorrelated noise and noise from previous layers is effectively removed, see top panels in Fig. 3. On the other hand, a matrix with $I\mu^2(\mathbf{W}) < \eta(\mathbf{W})$ increases uncorrelated noise (bottom panels in Fig. 3), and the corresponding SNRs become lower. These relations between matrices do not influence correlated noise's contribution, and for comparable levels of correlated and uncorrelated noise, one will see mainly the impact of correlated noise for $I\mu^2(\mathbf{W}) > \eta(\mathbf{W})$ and the one of uncorrelated noise if $I\mu^2(\mathbf{W}) < \eta(\mathbf{W})$. An important conclusion is that if uncorrelated noise dominates, one can simply leverage learning (optimization) algorithms that force the system towards a topology with $I\mu^2(\mathbf{W}) > \eta(\mathbf{W})$. A common mechanism for inducing correlating noise is a noisy power supply in a general sense. In electronics, this could be the circuit stabilising V_{dd} , while in optics this could be a pump or illumination source of photonic neurons. Since a general system will only have very few of such components, it appears feasible that these should receive an increased attention during the design stage.

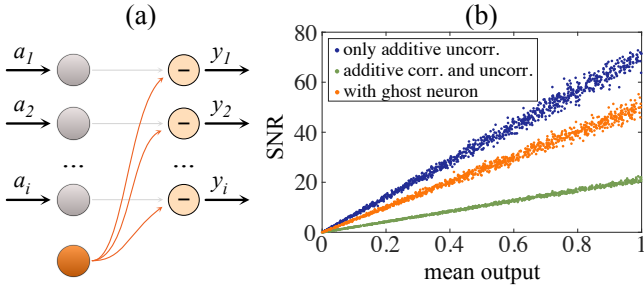


FIG. 4. Schematic representation, how the ghost neuron can be added to the network with direct coupling (panel (a)). Panel (b) shows SNR for the case without ghost neuron with only additive uncorrelated noise (blue points), with both types of additive noise (green) and for the case with one ghost neuron (orange color). Noise intensities are $D_A^U = 10^{-4}$, $D_A^C = 10^{-3}$

IV. GHOST NEURONS FOR ADDITIVE CORRELATED NOISE MITIGATION

Let us consider a FNN layer illustrated in Fig. 4(a) comprising of $I = 100$ nonlinear and noisy neurons. Each neuron i receives input signal a_i emulating a neuron's input from the previous layer. Then the output of neuron i including correlated and uncorrelated additive noise is

$$y_i = f(a_i) + \sqrt{2D_A^C} \xi^{C,A} + \sqrt{2D_A^U} \xi_i^{U,A}, \quad (8)$$

$$\text{Var}[y_i] = 2D_A^C + 2D_A^U.$$

We now suppress additive noise and include an extra neuron with identical noise properties. Importantly, this *ghost neuron* receives no input, but simply mimics the noise within the layer. The ghost neuron's output is then simply subtracted from each neuron's output, before this value y_i propagates to the next later, which results in

$$y_i = \left(f(a_i) + \sqrt{2D_A^U} \xi_i^{U,A} - \sqrt{2D_A^U} \xi_g^{U,A} \right), \quad (9)$$

$$\text{Var}[y_i] = 4D_A^U.$$

As can be seen from Eq. (9), a ghost neuron fully suppresses correlated additive noise, yet the impact of uncorrelated additive noise is doubled. We confirm this in numerical simulation shown in Fig. 4(b). However, as we showed before, uncorrelated noise can be suppressed leveraging coupling statistics, in particular $I\mu^2(\mathbf{W}) > \eta(\mathbf{W})$. Rather than simply subtracting the ghost neuron's values as in Fig. 4(a), we now assign a weight to the ghost neuron's connection W_g , Fig. 5(a). The output transforms into

$$y = \sum_{j=1}^{I_n} W_{j1}^n \left(f(u) + \sqrt{2D_A^U} \xi_{n,j}^{U,A} + \sqrt{2D_A^C} \xi_n^{C,A} \right) + W_g \left(\sqrt{2D_A^U} \xi_g^{U,A} + \sqrt{2D_A^C} \xi_n^{C,A} \right), \quad (10)$$

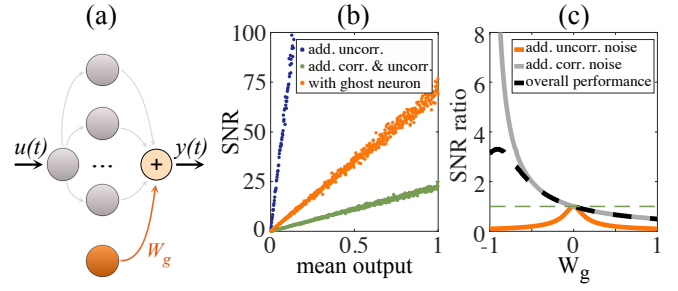


FIG. 5. Noise mitigation with ghost neuron for the uniform coupling schematically shown in the panel (a) and the SNR obtained in numerical simulation for $W_g = -1$ (panel (b)). SNR is prepared for the case without ghost neuron with only additive uncorrelated noise (blue points), with both types of additive noise (green) and for the case with one ghost neuron (orange color). Panel (c) shows noise mitigation with ghost neuron for the uniform coupling shown depending on the weight of the ghost neuron W_g . Noise intensities are the same as in Fig. 4.

and the corresponding variance is

$$\text{Var}[y] = \sum_{j=1}^{I_n} (W_{j1}^n)^2 \cdot 2D_A^U + W_g^2 \cdot 2D_A^U + \left(\sum_{j=1}^{I_n} W_{j1}^n + W_g \right)^2 \cdot 2D_A^C \approx 2D_A^U \cdot \left(W_g^2 + I_n \eta(\mathbf{W}^n) \right) + 2D_A^C \cdot \left(W_g + I_n \mu(\mathbf{W}^n) \right)^2. \quad (11)$$

For the special case of a uniform connection matrix $W_{j1}^n = 1/I_n$, the variance transforms to

$$\text{Var}[y] = 2D_A^U \cdot \left(\frac{1}{I_n} + W_g^2 \right) + 2D_A^C \cdot \left(1 + W_g \right)^2. \quad (12)$$

However, according to Eqs. (11,12), W_g impacts correlated and uncorrelated noise differently. The multiplier of uncorrelated noise $\left(W_g^2 + I_n \eta(\mathbf{W}^n) \right)$ shows that a ghost neuron increases the corresponding variance. The multiplier of correlated noise $\left(W_g + I_n \mu(\mathbf{W}^n) \right)^2$ indicates that if $W_g = -I_n \mu(\mathbf{W}^n)$ or $W_g = -1$ for uniform connectivity, then correlated noise is fully suppressed. Figure 5(b) numerically shows the case $W_g = -1$, which completely suppresses correlated additive noise, but at the same time increases uncorrelated noise. As a consequence, one needs to optimize W_g in function of the different noise amplitudes. Figure 5(c) shows the averaged ratio between SNRs obtained with and without ghost neuron depending on its weight W_g . Three types of noise are considered: additive uncorrelated noise (orange), additive correlated noise (gray) and both noise types (black). The best overall performance can be achieved when $W_g = -1$.

V. POOLING. UNCORRELATED NOISE REDUCTION

In this section we discuss a common strategy to reduce uncorrelated noise without constraining connections \mathbf{W} . This

method consists of combining several neurons into a distinct subgroups called pools. Each unit inside a pool of m neurons receives the same input, see In Fig. 6(a). The combined and hence averaged output signal of a pool is transmitted to the next layer. Each k th neuron of the i th group receiving the input signal a_i , has its own output value $y_{i,k}$ including noise and each group produces the averaged output $y_i^{\text{pool}} = \frac{1}{m} \sum_{k=1}^m y_{i,k}$. We used $m = 3$ in Fig. 6(a).

For uncorrelated additive and multiplicative noise, the variance of the corresponding output without pooling is³³

$$\text{Var}[y_j] = 2D_A^U + 2D_M^U \cdot E^2[y_j]. \quad (13)$$

Using a pool with m neurons then results in

$$\begin{aligned} \text{Var}[y_i^{\text{pool}}] &= \text{Var}\left[\frac{1}{m} \sum_{k=1}^m y_{i,k}\right] = \frac{1}{m^2} \cdot \text{Var}\left[\sum_{k=1}^m y_{i,k}\right] = \\ &= \frac{1}{m^2} \cdot \sum_{k=1}^m \left(2D_A^U + 2D_M^U \cdot E^2[y_{i,k}]\right) = \\ &= \frac{1}{m} \cdot \left(2D_A^U + 2D_M^U \cdot E^2[y_i]\right), \end{aligned} \quad (14)$$

as the variance of the i th neuron pool output. Comparing Eqs. (13, 14), one can see that average pooling reduces the variance of uncorrelated additive and multiplicative noise m times, while the SNR improves by \sqrt{m} . Figure 6 shows the SNR for additive and multiplicative noise separately (panels (b) and (c), respectively) and for the mixed uncorrelated noise (d).

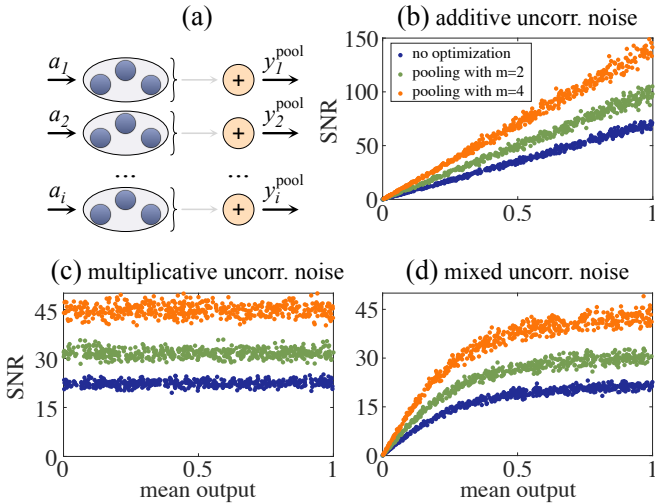


FIG. 6. Scheme of noise mitigation with average pooling (a) and improvement of SNR due to pooling technique (b–d). Noise intensities are $D_M^U = 10^{-3}$, $D_A^U = 10^{-4}$.

VI. COMBINING BOTH TECHNIQUES

Ghost neurons therefore remove correlated additive noise, while uncorrelated noise can be addressed using average pooling. Crucially, both concepts can be combined, and Fig.

7(a) illustrates the corresponding architecture, while panel (b) shows the SNR using average pooling in the case of, both, additive correlated and uncorrelated noise. Comparing Fig. 6(a) and Fig. 7(b), one can see the deteriorating effect of pooling when correlated noise is present. However, adding a ghost neuron substantially improves the situation, see Fig. 7(c).

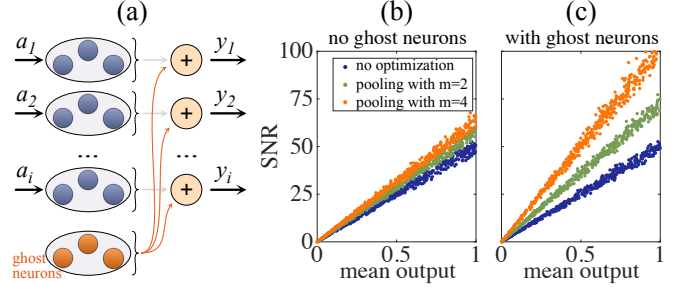


FIG. 7. Scheme of noise mitigation with combined pooling ghost neuron technique (a) and improvement of SNR due to only pooling (b) and combined (c) techniques for additive correlated and uncorrelated noise. Noise intensities are $D_A^U = D_A^C = 10^{-4}$.

VII. APPLICATION TO TRAINED NETWORK

In this section we apply the described above techniques to trained FNN. The noise-free network is trained to recognize MNIST handwritten digits from³⁶ using the open-source python software library Keras³⁷, using a network consisting of three layers whose connections were optimized with standard error back propagation. The first layer receives the input image's 28×28 pixels. The hidden layer has 100 nonlinear neurons with sigmoid activation function $f(x) = \frac{1}{1+e^{-x}}$, and the ten possible digits results in 10 nonlinear neurons with the same activation function in the hidden layer. The network's classification result is given by the output neuron with the largest value. With our proof-of-concept NN, we obtain a classification accuracy of 97.54% for the test data without noise.

Figure 8(a), green shows the SNR in the output layer for 500 randomly drawn digits without any noise mitigation strategy for $D_A^U = 10^{-4}$ and $D_A^C = 10^{-3}$. Figure 8(a) shows the ghost neuron's impact when applied only in the final (blue data) as well as in all layers (orange data) with $W_g = -1$. Again, we can see that mitigation of noise in the final layer is the most relevant. Secondly, we test pooling in a trained network with uncorrelated additive and multiplicative noise with noise intensities $D_A^U = 10^{-4}$ and $D_M^U = 10^{-3}$. The SNR without ($m = 1$, green data) and with average pooling ($m = 2$ for blue data and $m = 4$ for orange data) is shown in Fig. 8(b). However, we found almost no difference between pooling in all layers or only in the final one, which is because of the strong suppression of uncorrelated noise by a densely connected network consequence of training, for which $I\mu^2(\mathbf{W}) > \eta(\mathbf{W})$. We numerically confirmed that the SNR in our trained network is improved by a factor \sqrt{m} for $m = 2$ and $m = 4$.

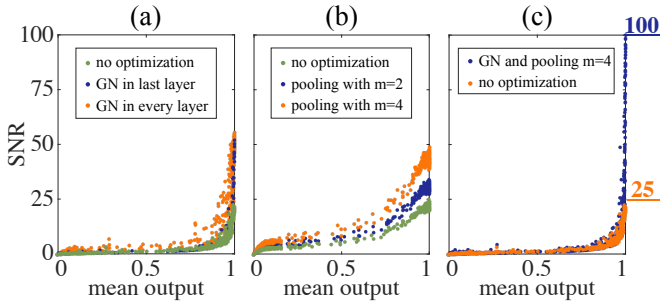


FIG. 8. Noise reduction in FNN trained for MNIST digits recognition. Green dependencies in panels (a,b) are prepared without any noise reduction, and demonstrate the SNR of the output FNN signal. Panel (a) shows the noise reduction in trained network with additive uncorrelated $D_A^U = 10^{-4}$ and correlated $D_A^C = 10^{-3}$ noise using ghost neuron. Panel (b) demonstrates uncorrelated noise reduction with intensities $D_A^U = 10^{-4}$ and $D_M^U = 10^{-3}$ using pooling method with $m = 2$ and $m = 4$. Panel (c) shows the result combining both techniques for additive noise $D_A^U = 10^{-4}$, $D_A^C = 10^{-3}$.

Finally, Fig. 8(c) shows the SNR for combination of both techniques of ghost neuron in the last layer and pooling with $m = 4$ for FNN with additive noise $D_A^U = 10^{-4}$, $D_A^C = 10^{-3}$. Panel (c) demonstrates SNR with combined optimization (blue) and without it (orange), providing maximum SNR values 100 and 25, respectively. Thus, combining technique leads to a 4-fold SNR improvement and consequently a 16-fold variance reduction.

All previous conclusions regarding the improvement of the noisy FNN were made with respect to SNR. However, the accuracy is more important characteristics for classification and recognition tasks. For the noise-free FNN it is 97.54%, while it drops to 92.97% for noisy FNN with additive noise $D_A^U = 10^{-4}$, $D_A^C = 10^{-3}$. Using the combined technique from the previous paragraph, the accuracy can be improved slightly to 93.1%. Meanwhile, the best performance can be achieved when using adaptive ghost neuron weights depending on matrices statistics: $W_{gi}^n = -\sum_{j=1}^{I_n} W_{ij}^n$. If these ghost neurons are added to every layer optimized with pooling, then the range of SNR values remains the same as in Fig. 8(c), but the accuracy becomes 97.49%, which much closer to the noise-free FNN.

VIII. CONCLUSIONS

We have proposed several noise reduction strategies specifically leveraging our previous analytical insights obtained in^{33,34}, mitigating uncorrelated noise and additive correlated noise. First, we show how the the particular statistics of connection matrices allow the mitigation of particular noise types. Such strategies can be used to amend optimization (learning) algorithms. We go beyond and introduce two complementary techniques of the case when statistics of intra-layer connections cannot be modified. Correlated additive noise can be removed using ghost neurons, while average pooling works

well for, both, uncorrelated additive and multiplicative noise without impacting correlated noise. Furthermore, we show how both techniques can be combined to form a comprehensive topology to suppress noise on a physical NN's hardware level. All above techniques were successfully applied to a NN for MNIST handwritten digit recognition, where they showed a reduction in the noise level in agreement to our analytical descriptions and almost complete noise suppression in terms of network accuracy.

ACKNOWLEDGMENTS

N. Semenova is supported by Russian Science Foundation (Project No. 21-72-00002).

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Appendix: Importance of connection matrices statistics

In order to illustrate the accumulation of noise, let us consider the vector of signals coming from noisy layer n to $(n + 1)$:

$$\vec{a}_{n+1} = \mathbf{W}^n \cdot \vec{y}_n, \quad \text{or} \quad a_{n+1,i} = \sum_{j=1}^{I_n} W_{ij}^n \cdot y_{n,j}. \quad (\text{A.1})$$

According to nomenclature of the main part of article, this value further transforms to $\vec{x}_{n+1} = f(\vec{a}_{n+1})$ after activation function and finally to \vec{y}_{n+1} after the noise impact.

Substituting the noise to $y_{n,j}$, Eq. (A.1) transforms to

$$a_{n+1,i} = \sum_{j=1}^{I_n} W_{ij}^n \cdot \left(\sqrt{2D_A^C} \xi_{n,j}^{C,A} + \sqrt{2D_A^U} \xi_{n,j}^{U,A} \right) + \sum_{j=1}^{I_n} W_{ij}^n x_{n,j} \cdot \left(1 + \sqrt{2D_M^C} \xi_{n,j}^{C,M} \right) \left(1 + \sqrt{2D_M^U} \xi_{n,j}^{U,M} \right). \quad (\text{A.2})$$

All terms and multipliers of correlated noise do not depend on index j and they can be therefore moved out of sums:

$$a_{n+1,i} = \sqrt{2D_A^C} \xi_{n,j}^{C,A} \cdot \sum_{j=1}^{I_n} W_{ij}^n + \sqrt{2D_A^U} \cdot \sum_{j=1}^{I_n} W_{ij}^n \xi_{n,j}^{U,A} + \left(1 + \sqrt{2D_M^C} \xi_{n,j}^{C,M} \right) \cdot \sum_{j=1}^{I_n} W_{ij}^n x_{n,j} \left(1 + \sqrt{2D_M^U} \xi_{n,j}^{U,M} \right). \quad (\text{A.3})$$

The variance of this noisy signal will be determined based on the basic arithmetic principles of calculating the variance of random variables³⁸ such as:

$$\begin{aligned} \text{Var}[c \cdot \xi] &= c^2 \cdot \text{Var}[\xi]; & \text{Var}[\xi + c] &= \text{Var}[\xi]; \\ \text{Var}[\xi \pm \zeta] &= \text{Var}[\xi] + \text{Var}[\zeta]; \\ \text{Var}[\xi \cdot \zeta] &= (\text{E}^2[\xi] + \text{Var}[\xi]) \text{Var}[\zeta] + \text{E}[\zeta] \text{Var}[\xi], \end{aligned}$$

where ξ and ζ are some uncorrelated random variables, c is some constant or noise-free variable. Then the variance of Eq. (A.3) is

$$\begin{aligned} \text{Var}[a_{n+1,i}] &= 2D_A^C \left(\sum_{j=1}^{I_n} W_{ij}^n \right)^2 + 2D_A^U \sum_{j=1}^{I_n} (W_{ij}^n)^2 + \\ &(1 + 2D_M^C) \cdot \text{Var} \left[\sum_{j=1}^{I_n} W_{ij}^n x_{n,j} (1 + \sqrt{2D_M^U} \xi_n^{U,M}) \right] + \\ &2D_M^C \cdot \text{E}^2 \left[\sum_{j=1}^{I_n} W_{ij}^n x_{n,j} (1 + \sqrt{2D_M^U} \xi_n^{U,M}) \right] = \\ &2D_A^C \left(\sum_{j=1}^{I_n} W_{ij}^n \right)^2 + 2D_M^C \cdot \left(\sum_{j=1}^{I_n} W_{ij}^n \text{E}[x_{n,j}] \right)^2 + \\ &2D_A^U \sum_{j=1}^{I_n} (W_{ij}^n)^2 + 2D_M^U (1 + 2D_M^C) \sum_{j=1}^{I_n} (W_{ij}^n)^2 \text{E}^2[x_{n,j}] \\ &+ (1 + 2D_M^C) (1 + 2D_M^U) \cdot \sum_{j=1}^{I_n} (W_{ij}^n)^2 \text{Var}[x_{n,j}]. \end{aligned}$$

For simplification, we assume that $\sum_{j=1}^{I_n} (W_{ij}^n)^2 \approx I_n \cdot \eta(\mathbf{W}^n)$

and $\left(\sum_{j=1}^{I_n} W_{ij}^n \right)^2 \approx I_n^2 \cdot \mu^2(\mathbf{W}^n)$, where $\eta(\cdot)$ is the mean of the square and $\mu(\cdot)$ is the mean (see Eqs. (4,5), main text). Then

$$\begin{aligned} \text{Var}[a_{n+1,i}] &\approx 2D_A^C \cdot I_n^2 \mu^2(\mathbf{W}^n) + 2D_A^U \cdot I_n \eta(\mathbf{W}^n) + \\ &2D_M^C \mu^2(\text{E}[\vec{x}_n]) \cdot I_n^2 \mu^2(\mathbf{W}^n) + \\ &2D_M^U (1 + 2D_M^C) \eta(\text{E}[\vec{x}_n]) \cdot I_n \eta(\mathbf{W}^n) + \\ &(1 + 2D_M^C) (1 + 2D_M^U) \eta(\mathbf{W}^n) \cdot \text{Var}[\vec{x}_n]. \end{aligned} \quad (\text{A.4})$$

We will not go into detail about the last term of Eq. (A.4) as it is not the subject of this article, and it has been described and analyzed in Ref.³⁴. It is clearly seen, that all rest terms with $I_n^2 \mu^2(\mathbf{W}^n)$ are related to correlated noise as:

$$I_n^2 \mu^2(\mathbf{W}^n) \cdot \left\{ 2D_A^C + 2D_M^C \cdot \mu^2(\text{E}[\vec{x}_n]) \right\}, \quad (\text{A.5})$$

while terms with $I_n \eta(\mathbf{W}^n)$ are

$$I_n \eta(\mathbf{W}^n) \cdot \left\{ 2D_A^U + 2D_M^U (1 + 2D_M^C) \cdot \eta(\text{E}[\vec{x}_n]) \right\}. \quad (\text{A.6})$$

Comparing Eqs. (A.5) and (A.6) one can see that if $I_n \mu^2(\mathbf{W}^n) > \eta(\mathbf{W}^n)$, then the impact of uncorrelated noise is less than the correlated noise when noise intensities are the same $D_A^U = D_A^C$, $D_M^U = D_M^C$.

¹Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature* **521**, 436–444 (2015).

²A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Commun. ACM* **60**, 84–90 (2017).

³D. Maturana and S. Scherer, “Voxnet: A 3d convolutional neural network for real-time object recognition,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015) pp. 922–928.

⁴A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013) pp. 6645–6649.

⁵S. Kar and J. M. F. Moura, “Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise,” *IEEE Transactions on Signal Processing* **57**, 355–369 (2009).

⁶S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, “Deep Learning with Limited Numerical Precision,” *Proceedings of the 32nd International Conference on International Conference on Machine Learning* **37**, 1737–1746 (2015).

⁷L. G. Wright, T. Onodera, M. M. Stein, T. Wang, D. T. Schachter, Z. Hu, and P. L. McMahon, “Deep physical neural networks trained with back-propagation,” *Nature* **601**, 549–555 (2022).

⁸D. Marković, A. Mizrahi, D. Querlioz, and J. Grollier, “Physics for neuro-morphic computing,” *Nature Reviews Physics* **2**, 499–510 (2020).

⁹K. Boahen, “A neuromorph’s Prospectus,” *Computing in Science & Engineering* **19**, 14–28 (2017).

¹⁰Z. Wang, S. Joshi, S. Savel’Ev, W. Song, R. Midya, Y. Li, M. Rao, P. Yan, S. Asapu, Y. Zhuo, H. Jiang, P. Lin, C. Li, J. H. Yoon, N. K. Upadhyay, J. Zhang, M. Hu, J. P. Strachan, M. Barnell, Q. Wu, H. Wu, R. S. Williams, Q. Xia, and J. J. Yang, “Fully memristive neural networks for pattern classification with unsupervised learning,” *Nature Electronics* **1**, 137–145 (2018).

¹¹P. Lin, C. Li, Z. Wang, Y. Li, H. Jiang, W. Song, M. Rao, Y. Zhuo, N. K. Upadhyay, M. Barnell, Q. Wu, J. J. Yang, and Q. Xia, “Three-dimensional memristor circuits as complex neural networks,” *Nature Electronics* **3**, 225–232 (2020).

¹²Q. Xia and J. J. Yang, “Memristive crossbar arrays for brain-inspired computing,” *Nature Materials* **18**, 309–323 (2019).

¹³J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. Le Gallo, X. Fu, A. Lukashchuk, A. S. Raja, J. Liu, C. D. Wright, A. Sebastian, T. J. Kippenberg, W. H. P. Pernice, and H. Bhaskaran, “Parallel convolutional processing using an integrated photonic tensor core,” *Nature* **589**, 52–58 (2021).

¹⁴D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, “Parallel photonic information processing at gigabyte per second data rates using transient states,” *Nature communications* **4**, 1364 (2013).

¹⁵R. M. Nguimdo, P. Antonik, N. Marsal, and D. Rontani, “Impact of optical coherence on the performance of large-scale spatiotemporal photonic reservoir computing systems,” *Opt. Express* **28**, 27989–28005 (2020).

¹⁶C. Huang, V. J. Sorger, M. Miscuglio, M. Al-Qadasi, A. Mukherjee, L. Lampe, M. Nichols, A. N. Tait, T. F. de Lima, B. A. Marquez, J. Wang, L. Chrostowski, M. P. Fok, D. Brunner, S. Fan, S. Shekhar, P. R. Prucnal, and B. J. Shastri, “Prospects and applications of photonic neural networks,” *Advances in Physics: X* **7**, 1981155 (2022), <https://doi.org/10.1080/23746149.2021.1981155>.

¹⁷T. Wang, S.-Y. Ma, L. G. Wright, T. Onodera, B. C. Richard, and P. L. McMahon, “An optical neural network using less than 1 photon per multiplication,” *Nature Communications* **13**, 123 (2022).

¹⁸S. S. Panda and R. S. Hegde, “Fault tolerance and noise immunity in freespace diffractive optical neural networks,” *Engineering Research Express* **4**, 011301 (2022).

¹⁹“Neuromorphic computing with nanoscale spintronic oscillators,” *Nature* **547**, 428–431 (2017).

²⁰D. Psaltis, D. Brady, X.-G. Gu, and S. Lin, “Holography in artificial neural networks,” *Nature* **343**, 325–330 (1990).

²¹J. Bueno, S. Maktoobi, L. Froehly, I. Fischer, M. Jacquot, L. Larger, and D. Brunner, “Reinforcement Learning in a large scale photonic Recurrent Neural Network,” *Optica* **5**, 756 – 760 (2018).

²²X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, M. Jarrahi, and A. Ozcan, “All-Optical Machine Learning Using Diffractive Deep Neural Networks,” *Science* **26**, 1–20 (2018).

²³Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljacic, “Deep Learning with Coherent Nanophotonic Circuits,” *Nature Photonics* **11**, 441–446 (2017).

²⁴A. N. Tait, T. F. De Lima, E. Zhou, A. X. Wu, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, “Neuromorphic photonic networks using silicon photonic weight banks,” *Scientific Reports* **7**, 1–10 (2017).

²⁵J. Moughames, X. Porte, M. Thiel, G. Ulliac, L. Larger, M. Jacquot, M. Kadic, and D. Brunner, “Three-dimensional waveguide interconnects for scalable integration of photonic neural networks,” *Optica* **7**, 640–646 (2020).

²⁶Dinc, Niyazi Ulas, Psaltis, Demetri, and Brunner, Daniel, “Optical neural networks: The 3d connection,” *Photoniques*, 34–38 (2020).

²⁷J. Moughames, X. Porte, L. Larger, M. Jacquot, M. Kadic, and D. Brunner, “3d printed multimode-splitters for photonic interconnects,” *Opt. Mater.*

- Express **10**, 2952–2961 (2020).
- ²⁸B. Dolenko and H. Card, “Neural learning in analogue hardware: effects of component variation from fabrication and from noise,” *Electronics letters* **29**, 693–694 (1993).
- ²⁹J. Misra and I. Saha, “Artificial neural networks in hardware: A survey of two decades of progress,” *Neurocomputing* **74**, 239–255 (2010), *artificial Brains*.
- ³⁰A. A. Dibazar, A. Bangalore, Hyungook Park, S. George, W. Yamada, and T. W. Berger, “Hardware implementation of dynamic synapse neural networks for acoustic sound recognition,” in *The 2006 IEEE International Joint Conference on Neural Network Proceedings* (2006) pp. 2015–2022.
- ³¹M. C. Soriano, S. Ortín, L. Keuninckx, L. Appeltant, J. Danckaert, L. Pesquera, and G. van der Sande, “Delay-based reservoir computing: noise effects in a combined analog and digital implementation,” *IEEE transactions on neural networks and learning systems* **26**, 388–393 (2015).
- ³²R. Frye, E. Rietman, and C. Wong, “Back-propagation learning and non-idealities in analog neural network hardware,” *IEEE Transactions on Neural Networks* **2**, 110–117 (1991).
- ³³N. Semenova, X. Porte, L. Andreoli, M. Jacquot, L. Larger, and D. Brunner, “Fundamental aspects of noise in analog-hardware neural networks,” *Chaos: An Interdisciplinary Journal of Nonlinear Science* **29**, 103128 (2019), <https://doi.org/10.1063/1.5120824>.
- ³⁴N. Semenova, L. Larger, and D. Brunner, “Understanding and mitigating noise in trained deep neural networks,” *Neural Networks* **146**, 151–160 (2022).
- ³⁵B. Everitt, *The Cambridge Dictionary of Statistics* (Cambridge University Press, Cambridge, UK New York, 1998).
- ³⁶Y. LeCun, <http://yann.lecun.com/exdb/mnist/index.html> (2021).
- ³⁷F. Chollet *et al.*, “Keras,” GitHub (2015), <https://github.com/fchollet/keras>.
- ³⁸D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers – 3rd ed.* (John Wiley Sons, 2002).