Nanophotonics 2020; 9(13): 4139–4147

Research article

Louis Andreoli, Xavier Porte*, Stéphane Chrétien, Maxime Jacquot, Laurent Larger and Daniel Brunner

Boolean learning under noise-perturbations in hardware neural networks

https://doi.org/10.1515/nanoph-2020-0171
Received March 6, 2020; accepted May 25, 2020; published online June 24, 2020

Abstract: A high efficiency hardware integration of neural networks benefits from realizing nonlinearity, network connectivity and learning fully in a physical substrate. Multiple systems have recently implemented some or all of these operations, yet the focus was placed on addressing technological challenges. Fundamental questions regarding learning in hardware neural networks remain largely unexplored. Noise in particular is unavoidable in such architectures, and here we experimentally and theoretically investigate its interaction with a learning algorithm using an opto-electronic recurrent neural network. We find that noise strongly modifies the system's path during convergence, and surprisingly fully decorrelates the final readout weight matrices. This highlights the importance of understanding architecture, noise and learning algorithm as interacting players, and therefore identifies the need for mathematical tools for noisy, analogue system optimization.

Keywords: Boolean learning; neural networks; noise.

***Corresponding author: Xavier Porte**, FEMTO-ST/Optics Dept., UMR CNRS 6174, Univ. Bourgogne Franche-Comté, 15B avenue des Montboucons, 25030, Besançon Cedex, France, E-mail: javier.porte@femto-st.fr. https://orcid.org/0000-0002-9869-7170
Louis Andreoli, Maxime Jacquot, Laurent Larger and Daniel Brunner: FEMTO-ST/Optics Dept., UMR CNRS 6174, Univ. Bourgogne Franche-Comté, 15B avenue des Montboucons, 25030, Besançon Cedex, France
Stéphane Chrétien: FEMTO-ST/Optics Dept., UMR CNRS 6174, Univ. Bourgogne Franche-Comté, 15B avenue des Montboucons, 25030, Besançon Cedex, France; Laboratoire ERIC, UFR ASSP, Universite Lyon 2, 5 avenue Mendes France, 69676 Bron Cedex, France National Physical Laboratory, Teddington, UK The Alan Turing Institute, London, UK

1 Introduction

In recent years, neural networks (NNs) take centre-stage in advancing computation [1]. Optimized by training, such *learning* machines provide key advantages for solving abstract computational problems and already outperform humans in numerous tasks previously deemed impossible for classically (algorithmically) programmed computers [1–3].

However, NNs are still mostly emulated by traditional Turing/von Neumann computers. The absence of computing hardware supporting fully parallel NNs reduces energy efficiency and overall speed, and new paradigms addressing these problems are desirable. The lack of a parallel network substrate is a fundamental roadblock and is an active area of research since decades, with current analogue hardware either implementing the full network [4–6] or the neurons [7–11]. An implementation of nonlinear neurons, fully-parallel information transduction and learning on a substrate level promises a revolution, and photonic NNs [12, 13] remain a highly promising avenue [4, 5].

Noise is an inseparable companion of analogue hardware [14], yet the fundamental aspects of optimizing a noisy NN [15–17] have so far hardly been explored – neither in experiments [18, 19] nor in theory [14]. Here, we investigate the interactions between noise, learning rules and the topology of an error landscape for the first time. We experimentally implement a NN with 961 electro-optical neurons via a spatial-light modulator (SLM) [17], use diffraction [4, 5, 10, 20] to physically realize the network's internal connections and a digital micro-mirror device (DMD) for programmable Boolean readout weights [17]. The particular NN task consists in one-step-ahead prediction of the chaotic Mackey–Glass times series. Learning exclusively modifies the readout connections [21] via an evolutionary Boolean algorithm based on the error gradient only, and optimization is using either fully random (Markovian) or structured (greedy) exploration.

The statistics of the experimentally obtained learning trajectories prove that noise and exploration strategy strongly interact. Noise induces a kind of random forcing upon the descent algorithm, which strongly modifies the system's path during its convergence towards a local minimum. We find that noise decorrelates the final weight configurations: starting from identical weight configurations and exploring the error landscape's dimensions in identical sequences always leads to clearly differentiated local minima. Quite astonishingly, all minima are spaced at an almost constant distance from each other, which for the generally non-trivial error landscape topologies is unusual at the least. Noise therefore appears to arrange minimizers in periodic positions, much like competitive Brownian walkers with non-local interactions [22]. These fundamental effects highlight the importance of considering hardware architecture, noise and learning algorithm as intimately linked.

2 Neural network hardware

A recurrent NN inspired by reservoir computing, illustrated in Figure 1a, was our experimentally realized NN test bench. Figure 1b schematically depicts the experiment. An optical plane wave E^0 illuminates the SLM's pixels, and the reflected field is filtered by a polarizing beam splitter (PBS). The SLM combined with the PBS creates a cos(·) non-linearity and the SLM's pixels physically encode the NNs state. A quarter wave plate located between the PBS and the mirror directs the signal towards a camera, and a double pass through the diffractive optical element (DOE) establishes the recurrent connections $\boldsymbol{W}^{\mathrm{DOE}}$ [10, 17, 20]. Camera state $\hat{x}_i^{\mathrm{cam}}(n)$ at integer time n

$$\hat{x}_i^{cam}(n) = \alpha \left| \sum_j^N W_{i,j}^{\mathrm{DOE}} E_j(n) \right|^2$$

is combined with external input information $u\,(n+1)$ and sent to the SLM, creating the network's state according to

$$\hat{x}_i(n+1) = \alpha |E_i^0|^2 \cos^2\left[\beta \cdot x_i^{cam}(n) + \gamma W_i^{\mathrm{inj}} u(n+1) + \theta_i\right]. \tag{1}$$

Here, $N = 961$ is the recurrent layer's number of nodes, $\beta = 0.8$ the feedback gain, $\gamma = 0.4$ the input injection gain and α a normalization parameter. The optical electric field and the nonlinearity's bias offset for node i are E_i^0 and θ_i, respectively. Input information $u(n+1)$ is injected into the system according to random connections $\boldsymbol{W}^{\mathrm{inj}}$.

The polarization reflected by the PBS is imaged onto the DMD, whose mirrors are programmed to fixed angles of ±12° from normal incidence. A photodiode only detects optical signals reflected of mirrors with −12°, thus implementing a Boolean readout weight matrix $W_{i=1\ldots N}^{\mathrm{DMD}}(k)$. The RC's output is

$$\begin{aligned} y^{\mathrm{out}}(k, n+1) &\propto \left| \sum_i^N W_i^{\mathrm{DMD}}(k)\left(E_i^0 - E_i(n+1)\right)\right|^2 \\ &\propto \left| \sum_i^N W_i^{\mathrm{DMD}}(k) \tilde{x}_i(n+1)\right|^2. \end{aligned} \tag{2}$$

Here, k is the learning epoch and $\tilde{x}_i$ is the optical field of node i arriving at the detector. As in reservoir computing, we restrict learning to the optimization of the readout weights. Finally, the absence of negative weights is partially mitigated by distributing the offset phases $\theta_i|_{i=1\cdots N}$ randomly between $\theta_0+\delta\theta_i$ and $\theta_0+\Delta\theta+\delta\theta_i$, where $\delta\theta_i$ is a random Gaussian distribution [17]. Internal $\boldsymbol{W}^{\mathrm{DOE}}$ and readout $\boldsymbol{W}^{\mathrm{DMD}}(k)$ connections are, however, realized in passive and fully parallel photonic hardware.

As the network is constructed of physical neurons it harbours noise, which can either be additive or multiplicative, as well as correlated or uncorrelated [14]. The main sources of noise in our experiment are the SLM and the camera, in relation to which the illumination laser and output detector can be considered as noiseless, and so are the internal coupling and readout matrices implemented by the DOE and DMD, respectively. All relevant noise sources are therefore reservoir-internal, and our following discussion is by no means limited to systems where the readout layer is implement physically. More details about the theoretical treatment and propagation of noise in NNs

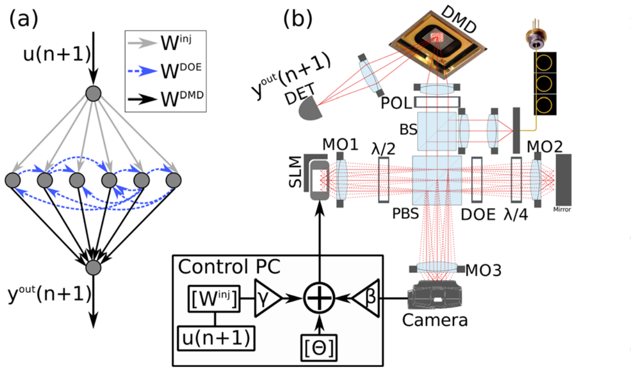

Figure 1: (a) Schematic illustration of a recurrent neural network. (b) Photonic implementation of a spatio-temporal neural network with 961 nodes. An optical plane wave illuminates the spatial light modulator (SLM), the neural network states are encoded by the SLM pixels. These are imaged on the camera, passing through a polarizing beam splitter (PBS) and the diffractive optical element (DOE) creating the coupling between network states. The information detected by the camera is used to drive the SLM. The network's output weights are realized via a digital micro-mirrors device (DMD) which creates Boolean readout weight matrix $\boldsymbol{W}^{\mathrm{DMD}}$.

as well as the individual noise sources and their respective amplitudes and statistics can be found in [14].

3 Boolean evolutionary learning

Most current learning techniques require complete knowledge of the internal network's state [21], all connection weights and potentially all gradients [1]. In a hardware network this demands probing (and most probably externally storing) the value of each node and connection, which necessitates auxiliary circuitry of a complexity potentially exceeding the actual neural network. This jeopardizes precisely the benefits one targets when mapping a neural network onto hardware. We therefore employ learning that only tracks the computation error's evolution, and hence imposes no constraint on the type of neurons, and more broadly, on hidden layers as a whole. Such an implementation's complexity does not depend on, and hence does not limit the NN's size, which is crucial considering the importance of scalability for computing.

Here, we optimize the DMD's configuration simply by measuring the impact of output mirrors' modifications onto computing error $\epsilon(k)$. The objective is to modify $\boldsymbol{W}^{\mathrm{DMD}}(k)$ during $k = 1, 2, \dots K$ learning epochs such that output $y^{\mathrm{out}}(n+1)$ best approximates target $\tau(n+1)$. Our Boolean learning algorithm can be divided into three conceptual sections:

3.1 Mutation

$$\mathbf{W}^{\mathrm{select}}(k) = \mathrm{rand}(N) \cdot \mathbf{W}^{\mathrm{bias}}(k), \tag{3}$$

$$l(k) = \max\left(\mathbf{W}^{\mathrm{select}}(k)\right), \tag{4}$$

$$W^{\mathrm{DMD}}_{l(k)}(k+1) = \neg\left(W^{\mathrm{DMD}}_{l(k)}(k)\right), \tag{5}$$

$$\mathbf{W}^{\mathrm{bias}}(k+1) = 1/N + \mathbf{W}^{\mathrm{bias}}(k), W^{\mathrm{bias}}_{l(k)} = 0. \tag{6}$$

We create a vector with N random elements, independently and identically distributed between 0 and 1 (rand (N)). $\boldsymbol{W}^{\mathrm{bias}}$ offers the possibility to modifying the otherwise stochastic $\mathbf{W}^{\mathrm{select}}(k) \epsilon \mathbb{R}^N$ in Eq. (3), whose largest entry's position, $l(k)$, determines the Boolean readout weight $W^{\mathrm{DMD}}_{l(k)}(k)$ to be mutated via a logical inversion operator $(\neg(\cdot))$, see Eqs. (4) and (5).

A fully stochastic Markovian descent is obtained with $\boldsymbol{W}^{\mathrm{bias}} = 1$ and excluding Eq. (6). However, here we also investigate exploration which makes mutating a particular connection in near succession unlikely. There, $\boldsymbol{W}^{\mathrm{bias}}$ is randomly initialized at $k = 1$, and at each epoch Eq. (6) increases the bias of all connections by $1/N$, while the currently modified connection's bias is set to zero. On average, the probability of again probing a particular weight reaches unity only after N learning epochs have passed, and we therefore refer to this biased exploration as *greedy* learning.

According to these instructions, our algorithm only probes, hence potentially mutates one mirror at a time. We have considered updating more than one mirror at each k, yet simplified numerical simulations indicate that convergence was significantly faster for updating only one weight at a time.

3.2 Error and reward signals

$$\epsilon(k) = \frac{1}{T}\sum_{n=1}^{T}\left(\tau(n+1) - \tilde{y}^{\mathrm{out}}(k, n+1)\right)^2, \tag{7}$$

$$r(k) = \begin{cases} 1 & \text{if } \Delta\epsilon(k) < 0 \\ 0 & \text{if } \Delta\epsilon(k) \geq 0 \end{cases} \tag{8}$$

$$\epsilon^{\min}(k) = (1 - r(k))\epsilon(k-1) + r(k)\epsilon(k), \tag{9}$$

$$k^{\min} = (1 - r(k))k^{\min} + r(k)k. \tag{10}$$

Mean square error $\epsilon(k)$ is obtained from a sequence of T data points according to Eq. (7), and comparison to the previous error assigns a reward $r(k) = 1$ only if a modification $\Delta\epsilon(k) = \epsilon(k) - \epsilon(k-1)$ was beneficial, see Eq. (8). In that case, the minimum error $\epsilon^{\min}_k$ and the best learning epoch $k^{\min}$ are updated following Eqs. (9) and (10). From the system's output $y^{\mathrm{out}}(k, n+1)$ we subtract the mean and normalize by its standard deviation, creating output $\tilde{y}^{\mathrm{out}}(k, n+1)$ which is used in Eq. (7).

3.3 Descent action

$$W^{\mathrm{DMD}}_{l(k),k} = r(k)W^{\mathrm{DMD}}_{l(k)}(k) + (1 - r(k))W^{\mathrm{DMD}}_{l(k)}(k-1). \tag{11}$$

Based on reward $r(k)$, the DMD's current configuration either accepts or rejects the previous modification, Eq. (11). For a noise-less system, reward $r(k)$ is therefore simply based on the gradient found at position $l(k)$. We will refer to this hypothetical gradient of a noise-less system as the *systematic* gradient.

4 Results

While such Boolean learning has been applied to a wide range of computational problems, recurrent neural networks have a particular relevance for dynamical signal processing, and we therefore explore one-step-ahead prediction of the chaotic Mackey–Glass sequence with a Lyapunov exponent of ~$3{\cdot}10^{-3}$. This particular input is a commonly employed benchmark test and our results are therefore directly comparable to other works such as Mackey-Glass prediction based on a semiconductor laser delay reservoir with weights optimized and applied in an offline procedure [23], as well as the seminal work on RC [21] – where however the time step was twice as large.

The chaotic sequence acting as input information $u(n{+}1)$ has zero mean and is normalized to its standard deviation, making error $\epsilon(k)$ the normalized mean square error. Its first two hundred points are used as training signal $u(n{+}1)$, of which we however removed the first 30 time steps due to their transient nature. The result is a training signal with $T = 200 - 30 = 170$ data points for which the target is $\tau(n+1) = u(n+2)$. Finally, the testing error is determined with an independent set of 9000 data-points unused in the training sequence. Based on mutation of the readout weights, our concept explores an error landscape with height $\epsilon(k)$ and position $\boldsymbol{W}^{\text{DMD}}(k)$, and reward $r(k)$ drives the configuration from $\boldsymbol{W}^{\text{DMD}}(1)$ to a local minima at $\boldsymbol{W}^{\text{DMD}}(k^{\text{min}})$. There our system will remain trapped due to an exploration step size of 1. We will refer to one complete learning process for $k{:}1{\rightarrow}k^{\text{min}}$ as a *minimizer*.

Understanding why generalization is possible for a training set size (T = 170) not orders of magnitude larger than the number of to be optimized weights (N = 961) is an interesting question. Recent results on deep neural networks, triggered by the insightful analysis from Ref. [24], show that overparametrisation may not preclude generalization. See Ref. [25] for an account to this phenomenon using random matrix theory, starting from simple linear models and generalizing to kernel estimation. In our setting we, however, might additionally postulate that we work below the overparametrization barrier due to the Boolean entries of $\boldsymbol{W}^{\text{DMD}}(k)$, which brings substantial rigidity into play. The price one pays is making the problem harder from a computational optimization viewpoint [26].

Typically, the main metric for evaluating learning are speed of convergence k^{min} and final inference error ϵ^{min}. However, in analogue neural hardware, reproducibility as well as robustness to noise and parameter drifts also play an essential role. We start by collecting statistical information and measure 20 (14) curves for the greedy (Markovian) exploration. All measurements started at an identical position $\boldsymbol{W}^{\text{DMD}}(1)$ and we therefore focus on the algorithm's exploration of the error-landscape. Results are shown in Figure 2, with individual learning curves as grey lines and their average as red crosses. Panel (a) shows data for the greedy, panel (b) for Markovian exploration.

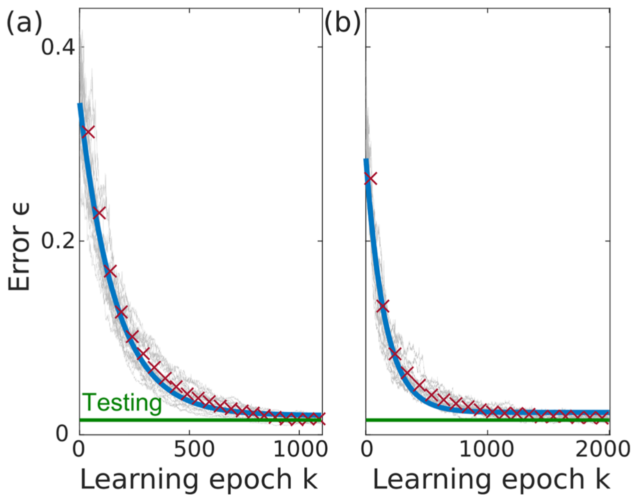

Figure 2: Learning performance, with individual (averaged) trajectories shown by grey data (red crosses). The green line is the testing error, and the blue is an exponential fit. Panel (a) and (b) were obtained with greedy and Markovian exploration, respectively.

4.1 Average and local features of convergence and minima

On average, the error landscape topology excellently follows an exponential decay for both exploration strategies, see fit (blue line) to the average error (red crosses) in Figure 2. Comparing individual trajectories, however, reveals strong inter-trial differences significantly exceeding the noise level. This diversity corresponds to the error landscape's topological richness probed by the different random descents, and trajectories range from rather smooth descents to paths including steep drops. No correlation between the starting $\epsilon(1)$ and best performance ϵ^{min} was found, and for the many different minimizers our system never got stuck in a local minima with bad performance. The different impact of a greedy or Markovian exploration sequence can be seen when comparing Figure 2a,b, respectively. Greedy exploration arrives at a minimum error of $\epsilon^{\text{min}} = (14.2 \pm 2.9) \cdot 10^{-3}$ after k^{min} = 973.6 ± 63.7 learning epochs, while Markovian exploration arrives at a slightly lower error $\epsilon^{\text{min}} = (13.4 \pm 1.9) \cdot 10^{-3}$ at the expense of a prolonged learning effort k^{min} = 1856.5 ± 175.1. Crucially, the system's testing error (green line in Figure 2) excellently matches its training error, hence ruling out over fitting. Noteworthy, convergence for both cases scales linear with network size

N [27], yet greediness approximately halves k^{min} compared to Markovian decent.

Nevertheless, despite the small deviations of ϵ^{min}, we find that optimal DMD configurations of individual learning trials have negligible correlation between each other. All minimizers therefore arrive at different local minima, and we encounter a surprising regularity in their geometric arrangement inside the high dimensional error landscape. The separation between two Boolean readout configurations $\boldsymbol{W}^{\text{DMD,a}}$ (k^{a}) and $\boldsymbol{W}^{\text{DMD,b}}$ (k^{b}) is determined by Hamming distance $H(k^{\text{a}}, k^{\text{b}}) = \sum_i \left| W_i^{\text{DMD,a}}(k^{\text{a}}) - W_i^{\text{DMD,b}}(k^{\text{b}}) \right|$. For the 20 minimizers we obtain 20 (20–1)/2 = 190 distances between their respective minima, and their statistical distribution obtained for greedy exploration is shown in Figure 3. The red line is a Gaussian fit centred at $H = 419$ and with a half width at $1/e$ of 14. Data shows a very specific and unusual error landscape topology: local minima appear not to be irregularly distributed, nor located in a particular region. Instead, the negligible correlations between the minimas' locations, and the systematic and narrow distribution of inter-minima distances shown in Figure 3 reveals their almost uniform distribution across the error landscape. Again, we find that Markovian exploration results in an identical behaviour.

5 Noise sensitivity

To further investigate this phenomena, we reduce the number of uncertainties during learning. We measure three minimizer paths starting at the same $\mathbf{W}_1^{\text{DMD}}$. One of the three minimizers acts as a master and defines mutation sequence $l(k), k \in [1 \ldots K]$, which the other two minimizers follow as slaves. Crucially, all three compute their own rewards $r(k)$ and hence independently evaluate mutating the same weight. Keeping the potentially systematic error of a slow experimental parameter drift in mind, the three systems are evaluated at each learning epoch k before advancing to $k + 1$. A single minimizer takes ~20 h, and sequential evaluation would amplify susceptibility to slow experimental parameter drifts which take place on the scale of hours in our experiment.

Results are shown in Figure 4a. The blue, green and red lines correspond to the different errors $\epsilon(k)$, plotted on a semi-logarithmic scale for the master and two slaves, respectively. The different data have a high degree of

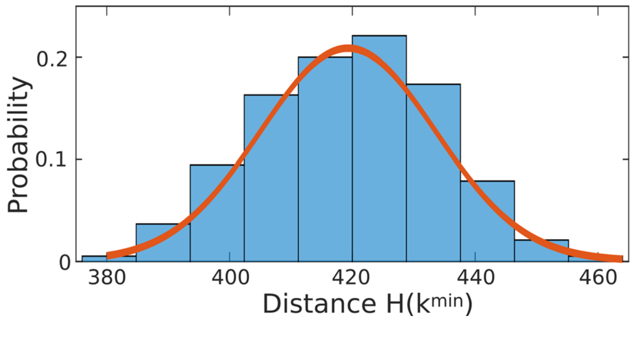

Figure 3: Probability distribution of the Hamming distances at $k = N$, obtained from the different greedy learning curves displayed in Figure 2a. Red curve is a Gaussian fit which is centred at 419 and has a standard deviation of 29.

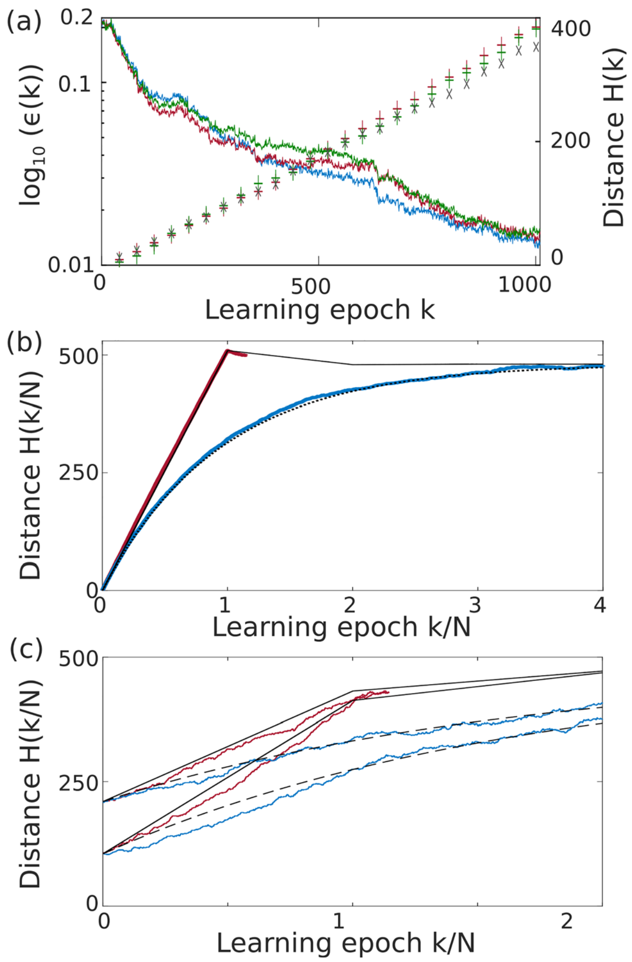

Figure 4: (a) Three minimizers starting from identical position are measured in parallel. The red and green (slaves) optimization paths test the dimension $l(k)$ determined by the blue minimizer (master), reward of each mutation are evaluated and applied individually. The lines show the individual errors on a logarithmic scale, crosses depict the Hamming distance's evolution between the three minimizers. (b) Hamming distance evolution with learning epoch k normalized by the network's Size N. Random mutation (blue line) leads to a smooth saturation function behaviour, while biased (greedy) mutation (red dashed line) results in linear intervals of length N. (c) The same characteristics are found for two minimizers starting already separated by a distance (1)≠0, and the behaviour is therefore generally true for Boolean learning in noisy hardware.

similarity with an average correlation of 99.4%, yet locally one can identify some significant differences. We computed the temporal evolution of the Hamming distances $H(k)$ of the two slaves to their master (red and green crosses) and between the two slaves (grey crosses); all three grow linearly at essentially the same rate. Without noise, each minimizer's reward $r(k)$ would be identical and they would consequently all follow the same trajectories $\boldsymbol{W}^{\mathrm{DMD}}(k)$ and arrive at the same minima $\boldsymbol{W}^{\mathrm{DMD}}(k^{\mathrm{min}})$.

To understand this behaviour, we therefore have to consider the impact of noise and learning upon the system's error $\epsilon(k)$. The response of error $\epsilon(k)$ to a modification in the system's output $\Delta y^{\mathrm{out}}(k)$ is

$$\Delta\epsilon(k) = \dot{\epsilon}(k)\cdot\Delta y^{\mathrm{out}}(k), \tag{12}$$

and $\Delta y^{\mathrm{out}}(k)$ is the mean modification of output $y^{\mathrm{out}}(k,n+1)$ within a certain window, which during training contains T sample points. Some general considerations regarding our system are in order. The amplitudes of all network nodes $\tilde{x}_i$ are Gaussian-distributed due to the SLM's illumination by a collimated Gaussian beam. Randomly changing one readout weight therefore results in $\Delta y^{\mathrm{out}}(k)$ according to a normalized Gaussian distribution with a width of $\Delta y^{\mathrm{out,learn}}(k) = \sigma^l(k)$. We have carefully characterized the noise of all elements in our opto-electronic NN, and their accumulated impact upon $y^{\mathrm{out}}(k)$ is excellently approximated by Gaussian white noise with a width of $\Delta y^{\mathrm{out,noise}}(k) = \sigma^n(k)$ [14]. Readout weights $\boldsymbol{W}^{\mathrm{DMD}}$ remain approximately evenly distributed between zeros and ones for all k. Learning does only modify readout connections and therefore neither modifies $\tilde{x}_i$ nor the system's noise, making both independent of learning. We can therefore assume that modifications to y^{out} induced by learning and noise remain constant for all k, hence $\sigma^l(k) = \sigma^l$ and $\sigma^n(k) = \sigma^n$.

The fact that according to Eq. (12) noise (σ^n) and learning (σ^l) both modify the system's error according to the same relationship is of general importance. Convergence during learning is characterized by $\epsilon(k)$ and $\dot{\epsilon}(k)$, and both depend on the particular computational tasks. Yet, the ratio between the error's susceptibility towards σ^n and σ^l remains constant as both scale with the same constant $\dot{\epsilon}(k)$. This in turn imposes the same constant relative susceptibility of reward $r(k)$ towards σ^n and σ^l, meaning that neither $\dot{\epsilon}(k)$ nor $\epsilon(k)$ modify the interplay between noise and learning upon the system's weights. The following discussion, results and observation are fully general and independent of task and learning algorithm – as long as these do not change amplitudes σ^n and σ^l. Noise and weight modifications are therefore independent players, whose action upon learning is somehow competitive.

The objective of modifying a readout weight is to probe the error landscape's systematic gradient. However, this action is contaminated by noise which can potentially exceed the systematic gradient in the opposite direction. The consequence is a change in the sign of $\Delta\epsilon(k)$, in which case reward $r(k)$ is inverted. How likely such a modification takes place depends on the relative amplitudes of σ^n and σ^l, and C is the constant probability of such a modification occurring. The analytical derivation of C is possible, yet beyond the scope of this manuscript.

Probability C is the driving force behind the growing separation between two identical minimizers, and two situations are relevant. The first situation occurs when $r(k)$ for one minimizer is inverted by noise while the other preserves its systematic value, which has a probability of $C(1-C)+(1-C)C = 2C(1-C)$. The other situation is if both minimizers have an identical reward $r(k)$, which can either be the consequence of both retaining their systematic result, or for both being inverted by noise, with a combined probability of $C(1-C)^2+C^2 = 1-2C(1-C)$. The first situation leads to $H(k+1) \neq H(k)$, the second to $H(k+1) = H(k)$, and the Hamming distance's rate equation is

$$\begin{aligned}\Delta H(k+1) = \rho^{\mathrm{id}}(k)2C(1-C)\\ -\rho^{\mathrm{op}}(k)2C(1-C).\end{aligned} \tag{13}$$

Here, $\rho^{\mathrm{id}}(k)$ and $\rho^{\mathrm{op}}(k)$ are the probability of finding both minimizers' weights $l(k)$ to be identical or opposite, respectively. Using $\rho^{\mathrm{id}}(k) = 1-\rho^{\mathrm{op}}(k)$, we arrive at

$$H(k+1) = H(k) + \tilde{C}\left(1-2\rho^{\mathrm{op}}(k)\right), \tag{14}$$

$$\Delta H(k+1) = \tilde{C}\left(1-2\rho^{\mathrm{op}}(k)\right), \tag{15}$$

where $\tilde{C} = 2C(1-C)$.

The Hamming distance's evolution is therefore governed by noise quantified through constant $\tilde{C}$, and by how the learning algorithm picks weight $W^{\mathrm{DMD}}_{l(k)}(k)$ from a population with a certain $\rho^{\mathrm{op}}(k)$, and neither error nor gradient play a role.

For fully random mutation, the probability of a weight to be selected is identical at every k, and hence the Hamming distance at the previous epoch k determines the probability of two weights being opposite in their configuration: $\rho^{\mathrm{op}}(k) = H(k)/N$. For greedy learning, the bias term in Eq. (3) causes mutations hardly ever to repeat the same weight during an interval specified by $k = k' + aN, k' \in [1,N]$, with non-negative integer a. The probability of both minimizers to be configured opposite for all k' and a specific a is therefore their Hamming distance at the end of the previous interval: $\rho^{\text{op}}(k) = H(aN)/N$, which results in constant slopes $\Delta H(a)$ for each k'.

Figure 4b shows the evolution of Hamming distance $H(k/N)$, and since all minimizers start at the same position $\boldsymbol{W}^{\text{DMD}}(1)$ we always have $H(1) = 0$. Greedy mutations in the experiment (analytics) are the red line (black dashed line), while random mutations in the experiment (analytics) are the blue line (black solid line). For both scenarios, greedy and random descent, the experimental data is the average obtained from 20 minimizers. We then changed the starting conditions and realized two parallel minimizers which started with a separation $H(1) > 0$, see Figure 4c. In general, the evolution according to Eq. (15) perfectly reproduces results of the highly different experimental learning scenarios. In particularly for the averaged data, where we always arrive at $H(k)|_{k\to\infty} = N/2$, regardless of the algorithm.

Different minimizers therefore always arrive at final readout configurations which share no common feature. This suggests a closer look into the role and relevance of individual weights: how many induce a systematic contribution to convergence at all, and if their gradients depend on the sequence of previous mutations. We optimized readout weights via two minimizers starting at different random positions $\boldsymbol{W}^{\text{DMD,a}}(1)$ and $\boldsymbol{W}^{\text{DMD,b}}(1)$, which arrived at two distinct local minima $M_a = \mathbf{W}^{\text{DMD, a}}(k^{\text{min, a}})$ and $M_b = \mathbf{W}^{\text{DMD, b}}(k^{\text{min, b}})$. Once there, we determined the list of m weights where M_b differs from M_a. The list is randomly arranged in sequence $\mathbf{l} \in [l(1), l(2), \ldots, l(m-1), l(m)]$ according to which we invert the corresponding weights $W^{\text{DMD, a}}_{l(k)}(k)$ and $W^{\text{DMD, b}}_{l(k)}(k)$ for $k \in [1, m]$. Importantly, this mutation is always kept and no optimization based on reward $r(k)$ is taking place. Starting from M_a (M_b), this results in a random path $P_a : M_a \to M_b$ ($P_b : M_b \to M_a$). As the weights addressed in sequence $\mathbf{l}(k)$ are the ones in an opposite configuration for M_a and M_b, P_a and P_b connect both minima along inverted trajectories, see Figure 5a. We probe error $\epsilon(k)$ along P_a and P_b and determine error gradients $\dot{\epsilon}^a(k) = \epsilon^a(k) - \epsilon^a(k^{\text{min, a}})$ and $\dot{\epsilon}^b(k) = \epsilon^b(k) - \epsilon^b(k^{\text{min, b}})$. Results are shown in Figure 5b in the $(\dot{\epsilon}_a, \dot{\epsilon}_b)$-plane, and for this experiment we obtained $m = 430$ different dimensions between M_a and M_b. Only ≈11% of the 430 gradients consistently remained below our system's noise floor $\dot{\epsilon}\sigma^n$, indicated by the grey circle.

Weights insensitive (sensitive) to preceding optimizations correspond to linearly independent (linearly dependent) NN dimensions. Linearly independent NN dimensions must always induce the same gradient, regardless of the preceding optimization path, and they therefore have to be located on the red diagonal line in

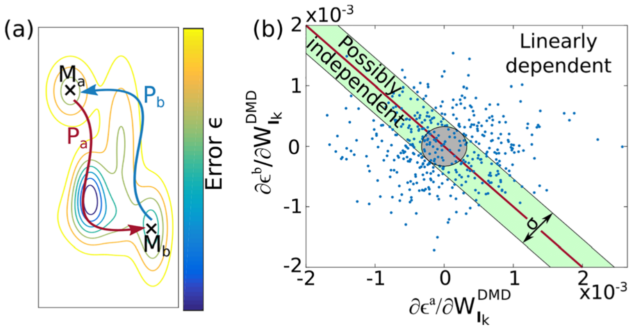

Figure 5: (a) Two inverted paths are probed between two local minima. (b) Error gradient for all readout weights encountered along path one and two as x and y axis, respectively. The red diagonal corresponds to linear independent weights, and the uncertainty induced by noise is indicated by the green area. Within, weights are potentially linear independent. Outside the green area weights are linearly dependent, and for data inside the grey circle of diameter σ no classification is possible.

Figure 5b. The Figure's green area indicates the linearly independent criteria when considering the impact of noise σ^n, and we find ≈30% of 430 dimensions fall into this category. However, this is only a necessary criterion; a sufficient criteria requires allocating dimensions inside this area for all potential configurations of the remaining m–1 dimensions, which for the 2^{429} possibilities is prohibitive to prove experimentally. The NN dimensions whose weight configuration depends on the previously optimized weights lie outside the grey and green areas. This is a sufficient criteria for linear dependent NN dimension, and we find ≈59% of 430 to be contained inside this set. There appears to be no concentration of potentially linearly independent dimensions towards the red diagonal line, and hence we conclude that these also mostly belong to the set of linear dependent dimensions.

6 Discussion

Our experimental findings and analytical descriptions are the first of their kind and stimulate a fundamental discussion. Equation (12) is of interesting consequence for noisy hardware NNs comprising linear readout weights. It links the susceptibility of $\Delta\epsilon(k)$ to NN noise to the system's location inside the error landscape $\epsilon(k)$. Experimentally we obtained noise induced variations $\dot{\epsilon}(150)\sigma^n = 4.4 \cdot 10^{-3}$ and $\dot{\epsilon}(961)\sigma^n = 0.6 \cdot 10^{-3}$. Our learning curves excellently agree with exponential convergence $\epsilon(k) = \epsilon^{(0)} e^{-\alpha k}$, for which $\dot{\epsilon}(k) \propto \epsilon(k)$. Error and gradient therefore evolve in a linearly proportional manner, and $\epsilon(150)/\epsilon(961) = 9.8$ in close agreement with the noise sensitivity's evolution

$\dot{\epsilon}(150)\sigma^n/\dot{\epsilon}(961)\sigma^n \approx 7.3$ confirms this fundamental relationship.

We would like to also propose alternative noise-mitigation approaches which are a derivative of our findings. Simply suppressing noise on a hardware level is potentially expensive, and topological requirements can limit mitigation based on connectivity statistics [14]. One might therefore curb the impact of noise by modified learning strategies. Noise will first of all limit the absolute performance, but also cause this performance to fluctuate, which is an effect one could address for example by amending an optimization's cost function by the gradients encountered in the proximity of a neighborhood. According to Eq. (12) the local gradients $\dot{\epsilon}$, or $\dot{\epsilon}|_{\mathbf{W}^{\text{DMD}}(k)}$ probed during learning also determine the system's performance fluctuation. Making them part of the optimization and not only the absolute error would therefore stabilize the system's performance. The influence of the noise on the performance of the algorithm can be mitigated using aggregated information along the trajectory in the spirit of recent work about variance reduction in stochastic gradient schemes and its extensions to non-convex zeroth-order optimization; see e.g. [28].

Equation (15) shows that for $C > 0$ the Hamming distance between readout weights of two systems will always tend towards complete decorrelation as $H(k)|_{k\to\infty} = N/2$. Even for 100% identical networks one will therefore never obtain similar readout configurations [29]. This finding can most likely be extended to the non-Boolean case and to the weights between layers of analogue deep NNs. The field of learning implemented in physical and hence noisy substrates is only in its infancy [5, 16, 17], and confirmation of our findings in other hardware systems would prove the generality of our result. Finally, the human brain is a very noisy network indeed [30], which suggests that our findings may also have interesting implications for the field of theoretical neuroscience.

We have shown that the large majority of our NN's dimensions are most likely linear dependent. What this means in pratical terms is that each modification of a weight has to be interpreted in the context of all previous modifications. Each configuration $\boldsymbol{W}^{\text{DMD}}(k)$ therefore encodes the history of modifications to the reward due to noise during the previous learning epoch.

One direct consequence for applications is that one cannot simply transfer or swap weight configurations between optimized analogue neural networks, even for potentially available identical twin networks. The reason is that optimized configurations are not only the consequence of error landscape, system properties and noise, but also of the precise history of noise during an exploration path. Even almost perfectly reproducible hardware networks will therefore always have to be individually trained for optimal performance; simply uploading a configuration will potentially not work. A 'school' in which each neural network learns individually might therefore be required. Finally, our findings open a new field where such twin-minimizers could be considered for probing and interrogating unknown hardware neural networks. The average divergences shown in Figure 4b agree exceptionally well with our model, and based on such data one can therefore make accurate inferences about the noise properties of a hardware NN and about its error landscape exploration strategy.

7 Conclusions

In our work we have investigated the intricate interactions between different learning concepts and the noise inherently present in analogue neural networks. We experimentally showed that trajectories of individual minimizers (i.e. learning trajectories) strongly diverge, and were able to analytically link this divergence to a constant ration between output error and noise susceptibility. Our analytical description only assumes a linear multiplication between a NN's state and its readout weights, and hence should be generally applicable to this wide class of analogue hardware NNs.

Acknowledgement: The authors acknowledge the support of the Region Bourgogne Franche-Comté. This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreements No. 860360 (POST DIGITAL) and No. 713694 (MULTIPLY). This work was also supported by the EUR EIPHI program (Contract No. ANR-17-EURE- 0002), the BiPhoProc project (Contract No. ANR-14-OHRI- 0002-02), and by the Volkwagen Foundation (NeuroQNet).

Author contribution: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.

Research funding: This research was funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreements No. 860830 (POST DIGITAL) and No. 713694 (MULTIPLY). This work was also supported by the EUR EIPHI program (Contract No. ANR-17-EURE- 0002), the BiPhoProc project (Contract No. ANR-14-OHRI- 0002-02), and by the Volkwagen Foundation (NeuroQNet).

Conflict of interest statement: The authors declare no conflicts of interest regarding this article.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[2] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, *OpenFace: A General- Purpose Face Recognition Library with Mobile Applications*, Pittsburgh, Carnegie Mellon University, Tech. Rep., 2016.

[3] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, IEEE, 2013, pp. 6645–6649.

[4] X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, M. Jarrahi, and A. Ozcan, "All-optical machine learning using diffractive deep neural networks," *Science*, vol. 26, pp. 1–20, 2018.

[5] Y. Shen, N. C. Harris, S. Skirlo, et al., "Deep learning with coherent nanophotonic circuits," *Nat. Photonics*, vol. 11, pp. 441–446, 2017.

[6] A. N. Tait, S. Member, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, "Broadcast and weight : an integrated network for scalable photonic spike processing," *J. Lightwave Technol.*, vol. 32, no. 21, pp. 3427–3439, 2014.

[7] L. Appeltant, M. C. Soriano, G. V. D. Sande, et al., "Information processing using a single dynamical node as complex system," *Nat. Commun.*, vol. 2, p. 468, 2011.

[8] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," *Opt. Express*, vol. 20, pp. 22783–22795, 2012.

[9] L. Larger, M. C. Soriano, D. Brunner, et al., "Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing," *Opt. Express*, vol. 20, pp. 3241–3249, 2012.

[10] D. Brunner, and I. Fischer, "Reconfigurable semiconductor laser networks based on diffractive coupling," *Opt. Lett.*, vol. 40, pp. 3854–3857, 2015.

[11] J. Torrejon, M. Riou, F. A. Araujo, et al., "Neuromorphic computing with nanoscale spintronic oscillators," *Nature*, vol. 547, pp. 428–431, 2017.

[12] N. H. Farhat, D. Psaltis, A. Prata, and E. Paek, "Optical implementation of the Hopfield model," *Appl. Opt.*, vol. 24, no. 10, pp. 1469–1475, 1985.

[13] D. Psaltis, and N. Farhat, "Optical information processing based on an associative-memory model of neural nets with thresholding and feedback," *Opt. Lett.*, vol. 10, pp. 98–100, 1985.

[14] N. Semenova, X. Porte, L. Andreoli, M. Jacquot, L. Larger, and D. Brunner, "Fundamental aspects of noise in analog-hardware neural networks," *Chaos*, vol. 29, no. 10, p. 103128, 2019.

[15] M. Hermans, P. Antonik, M. Haelterman, and S. Massar, "Embodiment of learning in electro-optical signal processors," *Phys. Rev. Lett.*, vol. 117, 2016, Art no. 128301.

[16] P. Antonik, M. Haelterman, and S. Massar, "Brain-inspired photonic signal processor for generating periodic patterns and emulating chaotic systems," *Phys. Rev. Appl.*, vol. 7, 5 2017, Art no. 054014.

[17] J. Bueno, S. Maktoobi, L. Froehly, et al., "Reinforcement learning in a large scale photonic recurrent neural network," *Optica*, vol. 5, pp. 756–760, 2018.

[18] R. Alata, J. Pauwels, M. Haelterman, and S. Massar, "Phase noise robustness of a coherent spatially parallel optical reservoir," *IEEE J. Select.Top. Quant. Electron.*, vol. 26, no. 1, pp. 1–10, 2020.

[19] M. C. Soriano, S. Ortín, D. Brunner, et al., "Optoelectronic reservoir computing: tackling noise-induced performance degradation," *Opt. Express*, vol. 21, pp. 12–20, 2013.

[20] S. Maktoobi, L. Froehly, L. Andreoli, et al., "Diffractive coupling for photonic networks: how big can we go?," *IEEE J. Select. Top. Quant. Electron.*, vol. 26, no. 1, pp. 1–8, 2020.

[21] H. Jaeger, and H. Haas, "Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, pp. 78–80, 2004.

[22] E. Heinsalu, E. Hernández-García, and C. López, "Competitive brownian and lévy walkers," *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.*, vol. 85, no. 4, pp. 1–10, 2012.

[23] J. Bueno, D. Brunner, M. Soriano, and I. Fischer, "Conditions for reservoir computing performance using semiconductor lasers with delayed optical feedback," *Opt. Express*, vol. 25, no. 3, pp. 2401–2412, 2017.

[24] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine learning and the bias-variance trade-off," *Proc. Natl. Acad. Sci.*, vol. 116, no. 32, pp. 15849–15854, 2018.

[25] T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani, "Surprises in high-dimensional ridgeless least squares interpolation," arXiv preprint arXiv:1903.08560, 2019.

[26] F. Hadaeghi and H. Jaeger, "Computing optimal discrete readout weights in reservoir computing is NP-hard," *Neurocomputing*, vol. 338, pp. 233–236, 2019.

[27] X. Porte, L. Andreoli, M. Jacquot, L. Larger, and D. Brunner, "Reservoir-size dependent learning in analogue neural networks," in *Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions*, I. V. Tetko, V. Kůrková, P. Karpov, and F. Theis, Eds., Cham, Springer International Publishing, 2019, pp. 184–192.

[28] S. Liu, B. Kailkhura, P. -Y. Chen, P. Ting, S. Chang, and L. Amini, "Zeroth-order stochastic variance reduction for nonconvex optimization," in *Advances in Neural Information Processing Systems, 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, Montréal, Canada, 2018, pp. 3727–3737.

[29] M. Freiberger, A. Katumba, P. Bienstman, and J. Dambre, "Training passive photonic reservoirs with integrated optical readout," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 1943–1953, 2019.

[30] A. Suarez-Perez, G. Gabriel, B. Rebollo, et al., "Quantification of signal-to-noise ratio in cerebral cortex recordings using flexible MEAs with co-localized platinum black, carbon nanotubes, and gold electrodes," *Front. Neurosci.*, vol. 12, pp. 1–12, 2018.