Data-based analysis of Laplacian Eigenmaps for manifold reduction in supervised Liquid State classifiers

Paolo Arena, Luca Patanè, Angelo Giuseppe Spinosa

Dipartimento di Ingegneria Elettrica Elettronica e Informatica, Università degli Studi di Catania, Viale Andrea Doria, 6, 95125 Catania CT (Italy)

Abstract

The manuscript introduces a data-driven technique founded on Laplacian eigenmaps for manifold reduction in bio-inspired Liquid State classifiers. Starting from a preliminary introduction about the algorithm and the need of using manifold reduction methods for data representation, a statistical analysis of hyperparameters involved in the Laplacian Eigenmaps technique is presented and the effects of quantisation on trained weights is discussed with a view to efficiently implement multiple parallel mappings in the digital domain.

Keywords: Reservoir Computing, Classification, Laplacian, Manifold reduction

1. Introduction

Classification is a fundamental skill in modern decision making and control systems. In living beings this capability is vital: it allows them to correctly process and categorise a large amount of sensory information and elicit a clear decision among different alternative choices. In Neurobiology, the role of even single, individual neurons was outlined to elicit a specific behaviour in front of suitable sensory stimuli. A classical example refers to the discovery of *Command neurons*, i.e. neurons which have critical role in the generation of a behaviour. Command neurons were initially discovered in invertebrates many decades ago, but are continuously being identified in higher animals [19]. How can a single neuron elicit a complex decision, i.e. how can the nervous system entrust one neuron the responsibility to classify among

different behavioural choices, being them the result of complex sensory processing phases? An answer could come from invertebrate neuroscience: in insects specific neuropiles, the Mushroom Bodies (MBs), are responsible for feature learning, whereas single extrinsic (i.e. output) neurons have the role to match MBs dynamics with other stimuli to elicit precise and specific behavioural responses. In [25] the authors argue, after their experiments, that those single neurons endowed with such decision making role are not selective by themselves, but act as general sensors of the underlying neural activity of the MBs, from which they take information. In other terms, they act as carriers of information selectivity and discrimination already performed in the MBs [3, 5]. One paradigm related to this concept is reservoir computing (RC) [17], which includes two main neural structures: Echo State and Liquid State Networks. In RC input signals stimulate a reservoir lattice that is made up of recurrently connected neurons; these generate a highly nonlinear, highly dimensional complex dynamics in a huge dimensional space [10], which should be able, in principle, to separate and make linearly separable all the input features embedded into the external stimuli. A simple linear readout map should then be enough for a correct classification. Under this perspective, even if the majority of applications of RC networks deal with signal and system dynamics prediction, typical classification tasks can be efficiently faced, and recent literature has been focalising on this [22], [32]. Typically, independently of the specific type of application, in RC networks the readout map is massively connected to the reservoir layer and this often reflects into a quite large dimension of the readout weight matrix. Again, in relation to the concept of *Command neuron*, and in comparison with the RC paradigm, it arises that if the reservoir layer works well, the readout map dimension could be hugely less than the dimension of the reservoir lattice. Translating into an engineering perspective, an efficient method is needed for dimensionality reduction, which could take into account both the reliability of the classification task and the dimension pruning. Moreover, it is likely that many readout maps draw information from the same lattice to provide concurrently different associations, exploiting the same spatial temporal, complex dynamics for many behavioural purposes. This concept finds its biological counterpart in the *Neural Reuse* paradigm: many neurons in the nervous system work concurrently in many tasks [1],[2]. So, a really efficient RC structure, drawing information from the same input space, can serve different outputs through different readout maps. For such an interesting purpose, the problem arises of the explosion of parameters that should

be used and trained if the different readout maps should be all massively connected to the same reservoir lattice. Hopefully, most of the mappings should not need the complete dimensional space as represented by the reservoir lattice. So, from an engineering perspective, a massive connection could be useless in these cases. Moreover, to fully exploit the space representation of the reservoir lattice, the massive implementation of all the readout maps could be computationally huge, preventing hardware solutions. A significant advantage could come from the memorisation of the readout weights in a compact, quantised, digital form which should, at the same time, preserve the quality of the mapping. All these problems are faced with in this manuscript. From the computational perspective, data representation plays a key role in classification. Most of real-life datasets consider patterns whose length, (i.e. the number of features), is significantly high. In several structures belonging to RC, the reservoir lattice is structurally separated from the readout map [17]; moreover, networks are usually trained in a supervised way and only the output weights are changed according to a suitable optimality criterion. Another important issue in machine learning tasks is the need of having a solution that should be as more numerically stable as possible, possibly obtained via a reduced dimensionality of the parameter space. This is why dimensionality reduction strategies like Principal Component Analysis (PCA) or Independent Component Analysis (ICA) are used in many engineering problems, including classification [28]. However, linear techniques cannot deal with complex nonlinear data and thus new methods have been proposed, like Laplacian Eigenmaps. These belong to the nonlinear dimensionality reduction techniques group, which includes also other strategies like diffusion maps or Hessian LLE (Local Linear Embedding). To some extent, Laplacian eigenmaps technique shares some aspects with LLE, a method consisting in finding an algebraic application from a high dimensional to a low dimensional space such that local distances between points are preserved. A comparison between this technique and Hessian LLE is reported in [12] and a more complete comparative review of dimensionality reduction algorithms is reported in [31]. All of these techniques are processing methods as well as regularisation algorithms. In [18], manifold reduction and regularisation approaches have been compared with different datasets: it has been shown that direct classification in high-dimensional spaces subjected to regularisation algorithms is usually slower and weaker than classification with manifold reduction. Assuming this result, Laplacian Eigenmaps could be very fast, but hyperparameters on which their operating principle is based

on are non-trivially determined. Here we provide a way to compute them directly starting from known data; from these hypotheses, statistical studies can be performed in order to validate the goodness of hyperparameters on classification. In this paper a statistical strategy for efficiently exploiting the Laplacian Eigenmaps approach is proposed and applied to a bio-inspired reservoir network. Moreover, the robustness of this network undergoing the manifold reduction against noise is investigated and an approach for trained weights quantisation, accomplished after validating the scalar product invariance hypothesis, is proposed. The paper is organised as follows: in Section 2 the strategy is introduced and the neural network **utilised for our simulations is** presented; in Section 3 the approach for the definition of the parameters is introduced, together with its assessment in terms of residuals, statistical and quantisation noise; **Section 4 reports the results obtained by carrying out several simulations; in Section 5 some comments and further considerations are reported, whereas** Section 6 concludes the paper.

2. Laplacian eigenmaps for classification

Before introducing the details of the proposed approach, the peculiarities of the algorithm and the network structure are introduced.

2.1. Problem definition

Typically, reservoir-based networks are trained by adopting the Least Mean Square (LMS) criterion as reported in [21]. This consists in solving a linear system of equations

$$\mathbf{T} = \mathbf{Z}\mathbf{W} \tag{1}$$

for the unknown $\mathbf{W}$. Let $\mathbf{t}_i$ and $\mathbf{w}_i$ be the i-th column of $\mathbf{T}$ and $\mathbf{W}$, respectively; if $\mathbf{w}_i = \mathbf{Z}^+\mathbf{t}_i$, where $\mathbf{Z}^+ \equiv \left(\mathbf{Z}^{\mathrm{T}}\mathbf{Z}\right)^{-1}\mathbf{Z}^{\mathrm{T}}$ is the Moore-Penrose matrix of $\mathbf{Z}$ [13], then $\|\mathbf{Z}\mathbf{w}_i - \mathbf{t}_i\| \geq \|\mathbf{Z}^+\mathbf{w}_i - \mathbf{t}_i\|\ \forall \mathbf{w}_i$, thus the Moore-Penrose matrix gets the optimal set of values with respect the well-known LMS problem (which takes into account a functional like $J(\omega) = \sum e(\omega)^2$ that ought to be minimised). Observe that Moore-Penrose matrix is defined also for complex matrices, but in our simulations we have dealt with real matrices only. Even if the unique requirement to calculate the Moore-Penrose matrix is that $\left(\mathbf{Z}^{\mathrm{T}}\mathbf{Z}\right)$ is either positive definite or negative definite, in practice this

matrix may be ill-posed and its inversion may be computationally expensive. In [26] solutions consisting in applying some regularisation techniques, like the well-known Tikhonov one for which $\mathbf{w}_i = \left(\mathbf{Z}^{\mathrm{T}}\mathbf{Z} + h^2\mathbf{I}\right)^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{t}_i$, have been introduced and discussed, but they have not been meant as manifold reduction methods. The problem to be addressed is to reduce significantly the dimensionality of the data acquired during the learning phase in a RC network. Since the reservoir layer contains neurons which are characterised by nonlinearities, a usually formulated hypothesis to apply the manifold reduction is linearity within the output layer, i.e. from the signals provided by the neurons of the liquid lattice, through the readout weights, to the output layer. Considering a classification task whose complementary aim is the exploitation of the space-time characteristics of RC, each input pattern p is characterised by an ensemble of constant values (representing the input features) that persistently excite the input layer for a a given time interval t_p. Moreover, in our implementation, only the readout weights have undergone a training procedure, and they have been changed *ad hoc*. The problem of finding the best set of weights is exactly stated by Equation 1, where $\mathbf{T} \in \mathbb{R}^{t_p n_p, m}$, $\mathbf{Z} \in \mathbb{R}^{t_p n_p, q}$ and $\mathbf{W} \in \mathbb{R}^{q,m}$ are orderly the matrix of all the m output signals, the "reservoir" matrix (containing the output signals from the q neurons in the lattice) and the matrix of all the trainable weights; in particular, t_p, n_p, q and m are the time (measured in samples) associated to each learning pattern, the number of learning patterns, the number of neurons constituting the processing layer and the number of output signals, respectively. Because of its structure, the problem can be solved by computing the Moore-Penrose matrix of $\mathbf{Z}$, leading to the final solution:

$$\mathbf{W}_{\mathrm{opt}} = \left(\mathbf{Z}^{\mathrm{T}}\mathbf{Z}\right)^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{T} \equiv \mathbf{Z}^{+}\mathbf{T} \tag{2}$$

From a computational perspective, this problem is not so trivial, because the matrix multiplication $\mathbf{Z}^{\mathrm{T}}\mathbf{Z}$ is $O\left(t_p n_p q^2\right)$ and its inversion is $O\left(q^3\right)$ by adopting the Gauss-Jordan elimination. Generally, reservoir matrices are noise-affected structures with multiple entries and their manipulation is not easy; additionally, a problem that could emerge is the so-called "curse of dimensionality" [8], which is the problem of an "overdimensioned" data representation, for which the need of finding a dimensionally reduced subspace that can represent them referring to a suitable optimality criterion emerges naturally. Moreover, in many applications the reservoir matrix is particularly sensible to noise because of its high condition number; dimensionality reduc-

tion methods aim at developing a meaningful representation of the original data in such a way the "transformed" reservoir matrix is better numerically posed. The algorithm we have adopted is functionally similar to that one reported in [15], but some modifications have been introduced to face with the different class of problems we have had to deal with. In particular, as reported in the next paragraphs, a semi-empirical approach, based on some statistical indices applied to our data has been adopted and residuals computation has been modified; nevertheless, the model selection criterion has not changed, keeping it founded on the Hannan-Quinn (HQ) index [16]:

$$\mathrm{HQ}(\beta) = \ln\left(\sigma_\beta^2\right) + 2\beta\left(m + q - n_{\mathrm{zero}}\right) N^{-1} \ln\left(\ln\left(N\right)\right) \tag{3}$$

In our simulations, σ_β^2 is the cumulative squared sum of residuals evaluated between signals in the original frame and in the transformed (reduced) β-dimensional frame (whose expression will be provided in the next paragraphs), N is the number of observations (i.e., $N = t_p n_p$) and n_{zero} is the number of null eigenvalues of the Laplacian matrix of the representative graph built by applying the algorithm (as described below). Observe that $n_{\mathrm{zero}} \geq 1$: whenever $n_{\mathrm{zero}} > 1$ the algorithm should take into account the presence of n_{zero} connected components, instead of a single one, within the graph.

The HQ index takes into account two contributions: errors between the signals (i.e. the capability of reproducing the input pattern efficiently enough with respect to the full dimension representation) and dimensions (degrees of freedom). In fact, if the error becomes very small it could happen that the dimension term is prohibitively big and vice-versa: minimum HQ is thus achieved when the best trade-off is reached.

2.2. Manifold reduction through Laplacian matrix

The algorithm we have applied is fully explained in [7]; here, a brief overview is reported in order to highlight the practical implications for further considerations. The first step consists in creating an undirected graph whose nodes come from the reservoir layer and from which an optimal and well-posed reservoir matrix can be calculated. This new matrix works as snapshot that crams time similarities among dynamic evolutions of neurons over all the patterns constituting the input dataset. Here, similarity can be modelled in multiple ways; in our case, we have simplified the algorithm by using a parameter k which has been used as follows: if $\mathbf{z}_i$ and $\mathbf{z}_j$ are two generic columns of $\mathbf{Z}$, then these neurons are temporally close (i.e., similar) if:

$$\text{dist}(i,j) = \|\mathbf{z}_i - \mathbf{z}_j\|_2 \leq k \tag{4}$$

where k has been assumed to be the neighbourhood size (for the sake of simplicity, we will indicate with $\mathbf{D}$ the symmetric matrix containing all the Euclidean distances, whose (i,j) element equals $\text{dist}(i,j)$). When Inequality 4 holds, a new edge between these two neurons/nodes is added to the graph $G = (V, E)$, where $|V| = q$ and $|E| = 0$ initially. The weight of the edge from the i-th node to the j-th node, namely e_{ij}, is computed as follows:

$$e_{ij} = e^{-\frac{\text{dist}(i,j)^2}{\sigma}} \tag{5}$$

Hyperparameters k and σ are the only quantities that determine the structure of the graph and therefore its characteristics. Once the matrix $\mathbf{W}_{\mathcal{G}}$ which contains all the weights e_{ij} for each edge is known, the degree matrix $\mathbf{D}_{\mathcal{G}}$ is easily computed because it is diagonal and its (i,i) element is given by $\sum_l e_{li}$ (row-wise summation). Finally, the Laplacian matrix is $\mathbf{L}_{\mathcal{G}} = \mathbf{D}_{\mathcal{G}} - \mathbf{W}_{\mathcal{G}}$. Now, the crux of the matter is to find a basis for a d-dimensional embedding map, with $d \ll q$, that represents the original manifold arising from $\mathbf{Z}$, namely $\widehat{\mathbf{Z}}$ (in the following, $\widehat{\cdot}$ will be referred to a generic variable represented with respect the embedding map basis). Formally, the problem consists in finding a transformation matrix $\mathbf{F} = [\mathbf{f}_1, ..., \mathbf{f}_d]$ where $\{\mathbf{f}_i, i = 1, ..., d\} \subseteq \{\mathbf{v} \in \mathbb{R}^{q,1} : \mathbf{L}_{\mathcal{G}}\mathbf{v} = \lambda \mathbf{D}_{\mathcal{G}}\mathbf{v}\}$ and the unknown d is chosen in such a way the Hannan-Quinn index in Equation 3 is minimum. Observe that columns of $\mathbf{F}$ are "ordered" such that $\mathbf{f}_i$ is the generalised eigenvector associated to the eigenvalue $\lambda_i \leq \lambda_{i+1}$, $\forall i$. Since the Laplacian matrix is positive-semidefinite, its minimum eigenvalue is $\lambda_0 = 0$ and it has to be discarded for the overall computation. Once the transformation matrix is known, a new reservoir matrix can be calculated as $\widehat{\mathbf{Z}} = \mathbf{ZF}$, whose pseudoinverse is simpler to obtain. The optimal weights with respect the embedding map frame are:

$$\widehat{\mathbf{W}}_{\text{opt}} = \widehat{\mathbf{Z}}^{+}\mathbf{T} \tag{6}$$

In the following Section the proposed algorithm for properly selecting hyperparameters k and σ following a data driven approach is presented in relation to both a specific network architecture and benchmark classification tasks.

2.3. Network description

The proposed network (a simplified version of the system described in [6]) is a simple three-layer architecture with one main computational core, acting as intermediate layer, where both the Cellular Nonlinear Networks (CNNs) [23] and Liquid State Networks (LSNs) paradigms are applied. More detailly, neurons within the liquid lattice are locally connected according to the CNN rule whereas the overall neural activity is "stored" in proper readout structures which provide the network output. Moreover, only the weights from the liquid lattice to the output layer undergo a batch training. Massive connections are established from the liquid layer to the output one, while random connections are created from the input layer to the reservoir layer and among neurons belonging to the latter. The input layer is composed of n_i Class I excitable Izhikevich neurons, arranged as a 1D array and whose dimension matches the number features for each pattern of the dataset used for classification, while the reservoir layer is a n_r-by-n_c grid of Class I excitable neurons with a predefined internal connectivity scheme. A single neuron of this type is characterised by the following dynamical equations:

$$\begin{cases} \dot{v} = 0.04v^2 + 5v + 140 - u + I \\ \dot{u} = a(bv - u) \\ \text{if } v \leq 30 \text{ mV, then } v \leftarrow c, u \leftarrow u + d \end{cases} \tag{7}$$

In this work, connections among these neurons have been established according to two different properties: position and type. If $p_{\text{exc}} \in [0, 1]$ is the percentage of excitatory neurons and $p_{\text{inh}} = 1 - p_{\text{exc}}$ is the percentage of inhibitory neurons, then $\mathbb{V}_{\text{exc}}$ ($\mathbb{V}_{\text{inh}}$) is the set of all the $p_{\text{exc}} n_r n_c$ excitatory ($p_{\text{inh}} n_r n_c$ inhibitory) neuron indices (an index can range from 1 to $n_r n_c$). If i-th neuron position is (r_i, c_i) and j-th neuron position is (r_j, c_j), then their mutual Euclidean distance is simply indicated as $\text{dist}(i, j) = \|(r_i - r_j, c_i - c_j)\|_2$. Finally, the probability of having a link between these neurons is given by:

$$P(i, j) = K(i, j)C(i, j) \tag{8}$$

where functions K and C are governed by the following equations:

$$K(i, j) = \begin{cases} 0 & \text{dist}(i, j) > 2 \\ 0.5 & 1 < \text{dist}(i, j) \leq 2 \\ 1 & 0 \leq \text{dist}(i, j) \leq 1 \end{cases}$$

$$C(i,j) = \begin{cases} 0.2 & i,j \in \mathbb{V}_{\text{inh}} \\ 0.4 & i \in \mathbb{V}_{\text{exc}}, j \in \mathbb{V}_{\text{inh}} \\ 0.6 & i \in \mathbb{V}_{\text{inh}}, j \in \mathbb{V}_{\text{exc}} \\ 0.8 & i,j \in \mathbb{V}_{\text{exc}} \end{cases}$$

Moreover, our networks adopt toroidal boundary conditions, implying that for each neuron its nearest neighbours include also those nodes placed on opposite edges. Several studies have been performed about this kind of network topology, like [11], which is frequently used on top-performing supercomputers in most of the cases. For instance, communication networks are designed such that several parameters, like throughput, are maximised; it has been shown that torus topology is characterised by lower packet transmission time over the mesh topology, no matter if contention holds or not [9]. Moreover, a network with toroidal connections (i.e., whose graph is properly said toroidal) provide multiple potential paths and routing is more direct, no matter what kind of manifold embeds it (like a plane as ours) [27].

To reconcile biological facts, the input neurons are connected to the lattice ones with a probability of 25% and fixed weights; moreover, the majority of liquid neurons are supposed to be excitatory with $p_{\text{exc}} = 0.75$ [4], while $n_r = n_c = 8$ during next simulations.

In order to be applied, Equation 8 requires that internal connections are randomly set; however, this choice is not particularly indicated whenever locality is meant as an operating hypothesis, because it could happen that randomness is not "sufficient". Contrarily, by adopting even simple rules to establish if two neurons can be linked together limits the possibilities of having a lot of connections, leading to sparse connectivity schemes. This explains why the CNN paradigm has been used here to reproduce this kind of spatial organisation: Biology provides some examples that describe how locality influences neural ensembles. A famous example is reported in [20], where Kenyon Cell axons in locusts are spatially arranged in a local, honeycomb-like fashion. Figure 1 shows a scheme of the adopted network. The liquid lattice is represented through its connectivity graph, where each spot represents the presence of a link between two neurons. Bright (dark) spots represent excitatory (inhibitory) weights. A special care will be paid in the following to the readout map configuration, which classically involves a full connectivity between the liquid lattice and the output neurons.

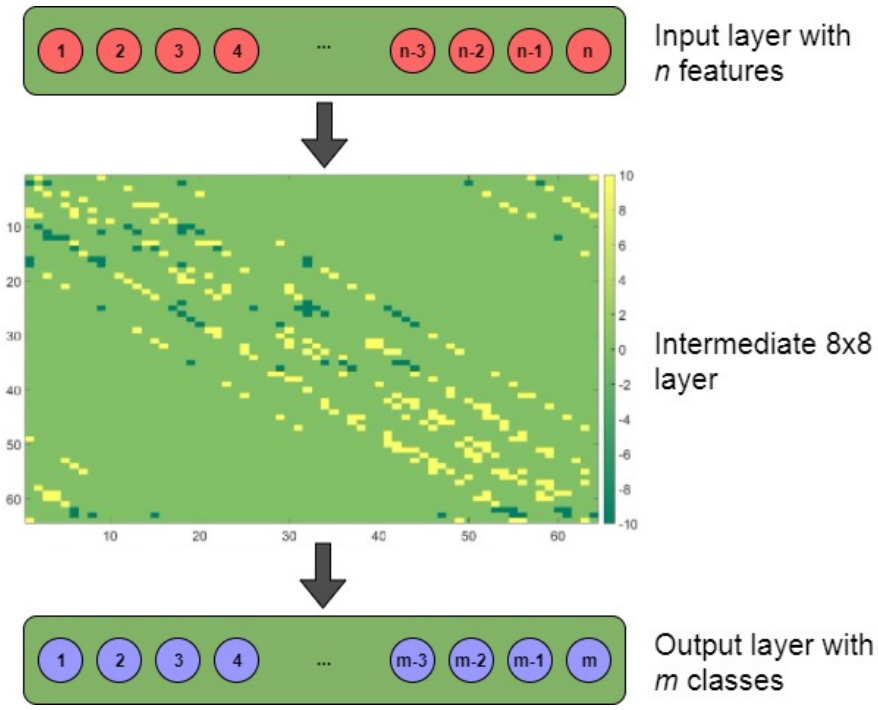

Figure 1: Functional organisation of the network. Input signals are randomly connected to the spiking liquid lattice, whose topology is defined by Eq. 8, with toroidal boundary conditions.

2.4. Targets encoding

A preliminary procedure to cope with datasets for multivariate, supervised classification is targets encoding, through which a pattern is expressed in terms of time-varying function. In other words, this step aims at developing the internal structure of T. Each pattern must produce a significant input current that ought to last enough to elicit a meaningful internal activity within the network. We have previously expressed the duration of each pattern with t_p, thus we can introduce two different time-varying functions, namely two exponential functions $1 - e^{-t/\tau}$ with two time constants only, the faster of which encodes the right class whilst the slower one is employed for all the other classes.

3. Algorithm tuning and details

3.1. A statistical approach for hyperparameters definition

The Laplacian-based algorithm assumes that two neurons (or nodes) whose temporal distance is relatively small are sufficiently similar and therefore they can be linked together and added to a representative graph whose weights are usually modulated as exponential functions as in Equation 5. An example of how these distances are distributed is shown in Figure 2. Nevertheless, the original algorithm does not provide any tip to choose the parameters like the neighbourhood size or the kernel width; in order to automatise the overall procedure, a statistical solution that reflects the temporal structure of the reservoir matrix has been adopted. Let $\mathrm{Perc}\left(\mathrm{vec}(\mathbf{D}), n\right)$ be the n-th percentile of the vectorised matrix $\mathbf{D}$, then we set

$$k = \mathrm{Perc}\left(\mathrm{vec}(\mathbf{D}), n\right) \tag{9}$$

$$\sigma = \mathrm{Perc}^2\left(\mathrm{vec}(\mathbf{D}), n\right) \tag{10}$$

where n can be tuned as preferred. Generally, a good choice of this parameter depends on how the distances are distributed, because when n becomes too small (big) then the probability of having a connection between two neurons decreases (increases) and therefore multiple isolated nodes appear (many nodes would be considered similar even if they are not). Also the HQ index changes with respect to the selected percentile. Figure 3 reports some trends with respect to two percentiles. What emerges from this figure is that when the percentile is high, the minimum HQ is reached when the reduced dimension is exactly equal to $q - n_{\mathrm{zero}}$ and this is not what we are looking for, because this does not provide any significant manifold reduction. However, a deeper analysis has shown that even smaller percentiles can guarantee good performance by exploiting a reduced dimensionality. In fact, HQ can reach a minimum for a much reduced dimension if the percentile is eased off: for instance, the aforementioned figure shows how the HQ has behaved when the percentile has been set to 35. Since our procedure has been devoted to classification, we have tested the behaviour of the network for several values of percentiles supposing a noisy environment: classification performance has been essentially invariant to percentiles above 20 (lower values have led to poorer performance). Thus, in terms of classification efficiency, percentiles 35 and 90 have been almost equivalent. Figure 4 depicts two examples of

weights distribution for both the 35-th and 90-th percentile. Because of Equations 9 and 10, higher neighbourhood sizes determine additional edges whose strength is further increased due to higher kernel widths. Moreover, Table 1 reports the condition numbers of the weight matrix without and with manifold reduction obtained in 5 different simulations. The straightforward application of such a reduction technique to the case study introduced above has its clearest effect in an improved numerical stability of the $\widehat{\mathbf{Z}}^{+}$ matrix through a sensible decrease in its condition number.

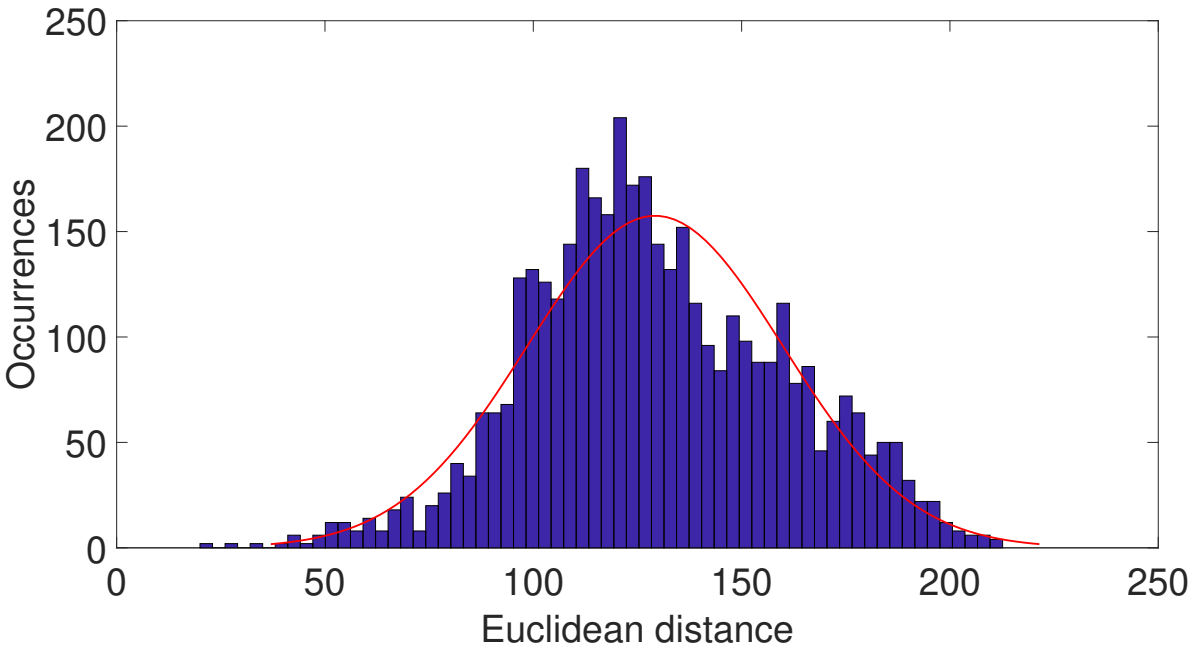

Figure 2: Distribution of Euclidean distances. This plot has been obtained for one simulation among all the others, but these are qualitatively similar and show a tendency to be normally distributed.

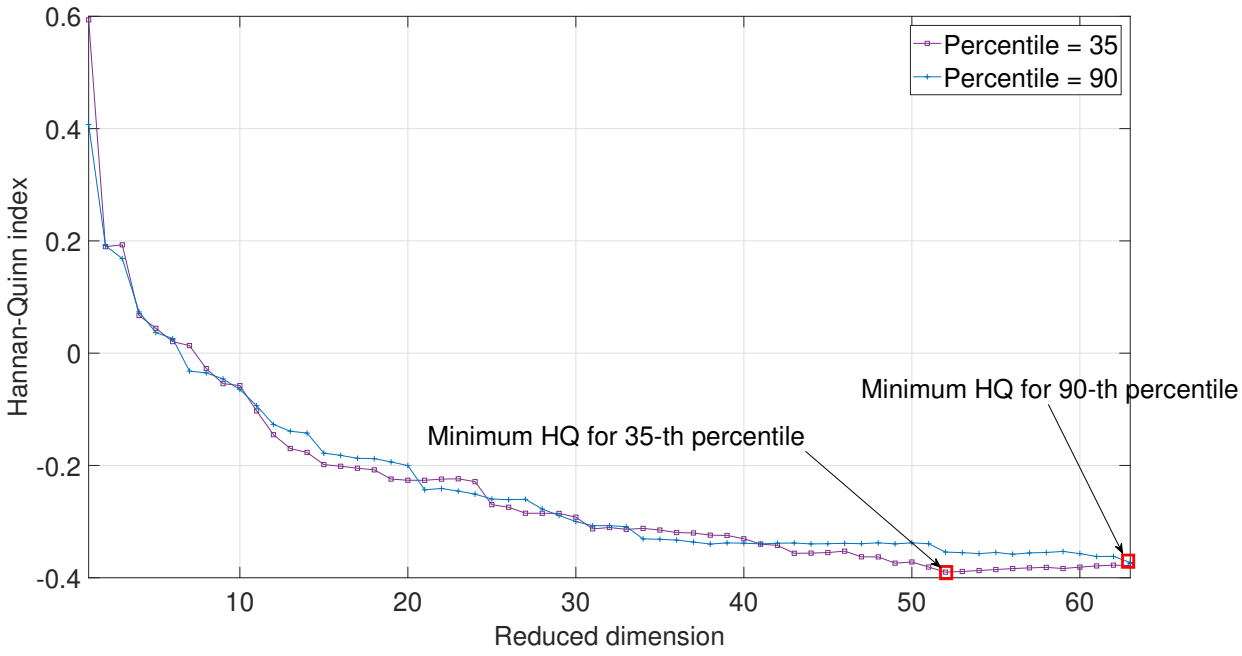

Figure 3: **Example of relationship between the Hannan-Quinn index trend and percentiles. Each percentile determines the effect of the total reduction over the reservoir matrix, thus the higher the percentile is, the lower the reduction is and then the more similar the original and the transformed data are. In our case study, our networks have been able to show acceptable outcomes in spite of the much lower percentile employed to reduce the reservoir matrix.**

	Condition numbers	
Simulation	*Reduction Off*	*Reduction On (35-th perc.)*
1	1.106e4	2367
2	1.37e4	3511
3	1.265e4	3254
4	1.162e4	3068
5	1.147e4	3139

Table 1: **Example that shows how manifold reduction can reduce condition numbers of ill-posed matrices.**

3.2. Residuals computation for a time-based winner-takes-all classification

The cases under analysis are typical supervised classification problems, where the output classes are known *a priori* for all the patterns of the dataset.

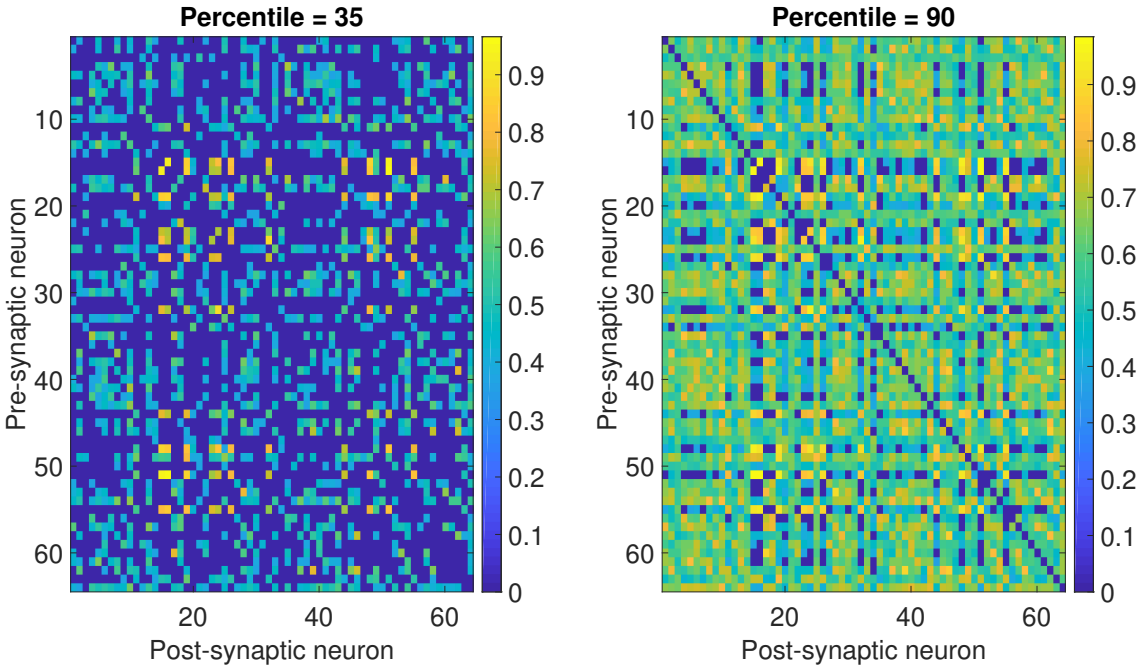

Figure 4: Examples of weights distribution for both the 35-th and 90-th percentile. **It is quite clear that as the percentile increases, the probability of having new connections increases too. Eventually, this may lead to a very interlocked graph.**

When both the learning and the test phases are completed, deciding if a pattern belongs to a class can be done in multiple ways; here, a temporal averaging approach has been adopted, as briefly discussed here.

Looking at the previous notation, let $\langle \mathbf{t}_i \rangle_p$ be the time mean of the i-th output signal over the p-th pattern, then $\forall p,\ \exists w \in \{1, 2, ..., m\} : \langle \mathbf{t}_w \rangle_p = \max_j \langle \mathbf{t}_j \rangle_p$; in this way, w-th index is referred to the winning class. A similar strategy has been adopted in [22], where authors have introduced two aggregation operators (both in time and space). Our approach preserves the temporal aggregation operation, but instead of computing readout time-varying weights as outcomes of the training phase, our weights are simply averaged over each pattern. Moreover, averaging is not weighted; this allowed us to avoid the computation of a reduced basis of orthonormal functions from which time-varying weights are expressed, leading to a structurally simpler approach. Additionally, input patterns **have been** presented as stepwise

constant time-series obtained by keeping the i-th feature constant for a predefined amount of time: this produces a significant neuronal dynamics. As introduced before, the residuals should be specialised for our Winner-Takes-All-like approach in order to improve the classification goodness. A way to fit it the residuals in this sense is:

$$\epsilon(p) = \frac{1}{(m-1)\langle\widehat{t_w}\rangle_p - \sum_{j\neq w}\langle\widehat{t_j}\rangle_p} \tag{11}$$

In this way, if $\epsilon \ll 1$ then $\langle\widehat{t_w}\rangle_p \gg \frac{\sum_{j\neq w}\langle\widehat{t_j}\rangle_p + 1}{m-1}$ and therefore the difference among the winning target and all the others is maximised. Eventually:

$$\sigma_\beta^2 \equiv \sigma_\beta^2(p) = \frac{\sum_p \|\epsilon(p)\|_2^2}{p} \tag{12}$$

In Figure 5, ϵ term is plotted for each pattern and as a function of the reduced dimension. This shows the benefit of the strategy in terms of discrimination capability.

3.3. Analysis of uniformly and normally distributed noise on classification performance

An interesting result we have obtained regards the effects of noise over the classifier, for several percentiles. Here, we have supposed to add either an uniformly or normally distributed noise to the set of output weights; formally, if $\mathbf{w_i}$ is the i-th weight vector connecting the liquid layer to the i-th output neuron, whose generic entry is w_{ij}, then w_{ij} is updated as

$$w_{ij} \leftarrow (1-\phi)w_{ij} + 2\phi w_{ij} r \tag{13}$$

where $\phi \in [0,1]$ is the noise intensity and $r \in [0,1]$ is a random variable. In our setup, $\phi = \{0, 0.1, ..., 1\}$ and for each value of ϕ 25 iterations have been performed. Moreover, the normally distributed noise mean and standard deviation have been set to 0 and 1, respectively.

3.4. Scalar product invariance of synaptic weights and quantisation

In this paragraph a new scenario **has been** supposed, regarding the quantisation of synaptic weights over a finite number of levels. This ought to mean that resolution affects the goodness of classification; moreover, because of the previous results about noise rejection manifold reduction should improve the quality of classification even if the network undergoes quantisation

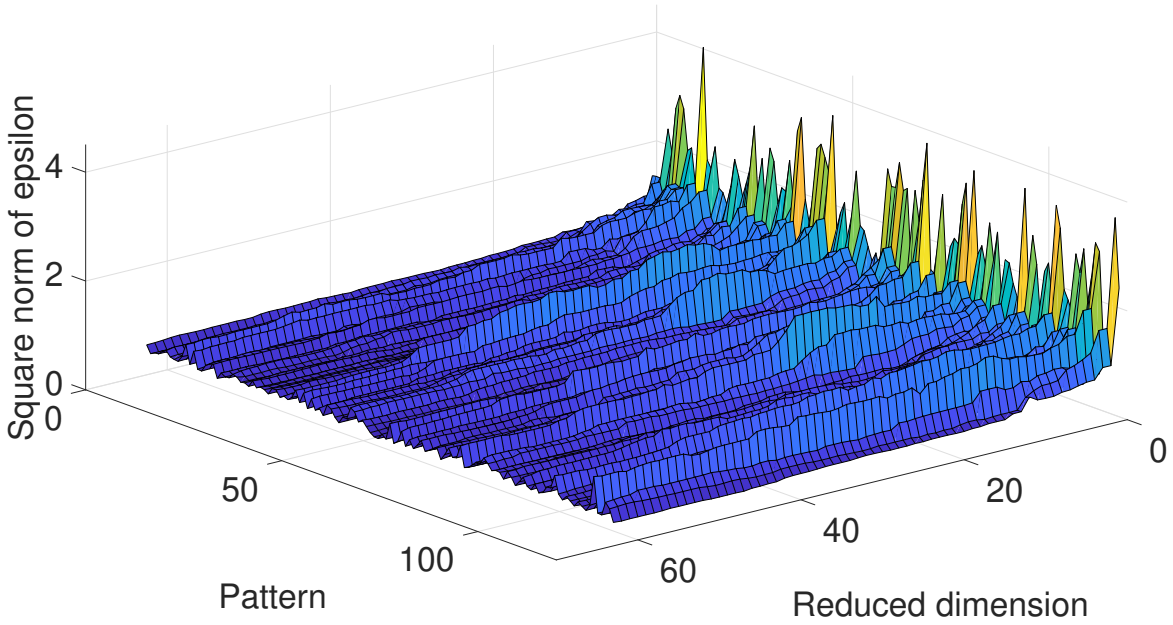

Figure 5: **Typical trend of the square** norm of residual ϵ, computed as Equation 12 states. The overall error is initially high, but the increasing reduced dimension leads to lower residuals and better performance. This usually happens because of both the presence of multiple connected components within $\mathcal{G}$ and the poor quality of the reduced manifold which cannot deal with the precision requirements.

noise (which is ineradicable when implementing the network in a low resolution digital environment). First, let us note that the computation procedure for the winning class is based on the calculation of a set of time averages and the evaluation of the the maximum of this set. Consequently the results of the classification should be scalar product invariant, whatever the (positive) scalar quantity is. This fact has been empirically validated and exploited to discretise all the weights. Discretisation procedure has been accomplished in two steps: optimal weights have been first normalised and then quantised. If $\mathbf{W}$ is a generic matrix containing all the synaptic weights (no matter if it is obtained without or with manifold reduction), whose maximum and minimum are named $w_{\max}$ and $w_{\min}$, respectively, then normalised, discretised weights have been computed as follows:

$$\mathbf{W} \leftarrow \left[\frac{2^{\text{bits}-1}}{\max\left(|w_{\max}|, |w_{\min}|\right)} \mathbf{W} \right] \tag{14}$$

where $[\cdot]$ refers to round operation. This procedure allows to have a finite number of weights over a finite number of intervals determined by the resolution, expressed in terms of binary digits. By applying Equation 14 in our simulations, results have revealed an intrinsic robustness of Laplacian Eigenmaps to quantisation in terms of goodness of classification. **Further details are reported in the next sections.**

4. Simulations

This section summarises the results obtained after testing the capabilities of the Laplacian Eigenmaps-based algorithm for reduction in classification tasks. In particular, two distinct cases have been analysed to show how the network behaves with data having different sizes and characteristics, whose properties are reported below.

4.1. Case study 1: Iris dataset

***Iris* dataset** was introduced by Fisher [14] in 1936 and consists of $n_p = 150$ patterns, which can belong to $m = 3$ classes only: *Iris setosa*, *Iris versicolor* and *Iris virginica*. Each pattern comprises $n_i = 4$ features, which express some salient characteristics of this kind of flower. The percentage of patterns dedicated to learning has been set to 80%.

Our results have shown that for the 35-th percentile, which has guaranteed an average reduced dimension equal to 52 over 5 reservoir matrices, noise rejection has been more pronounced when manifold reduction is applied, no matter if the noise is either normally or uniformly distributed (Figure 6 and 7). Regarding the quantisation of the trained weights, Figure 8 shows that it can be possible to have high performance with a more reduced number of bits when Laplacian Eigenmaps are applied. In terms of success rates, results obtained by fixing the percentile equal to 90 have been similar.

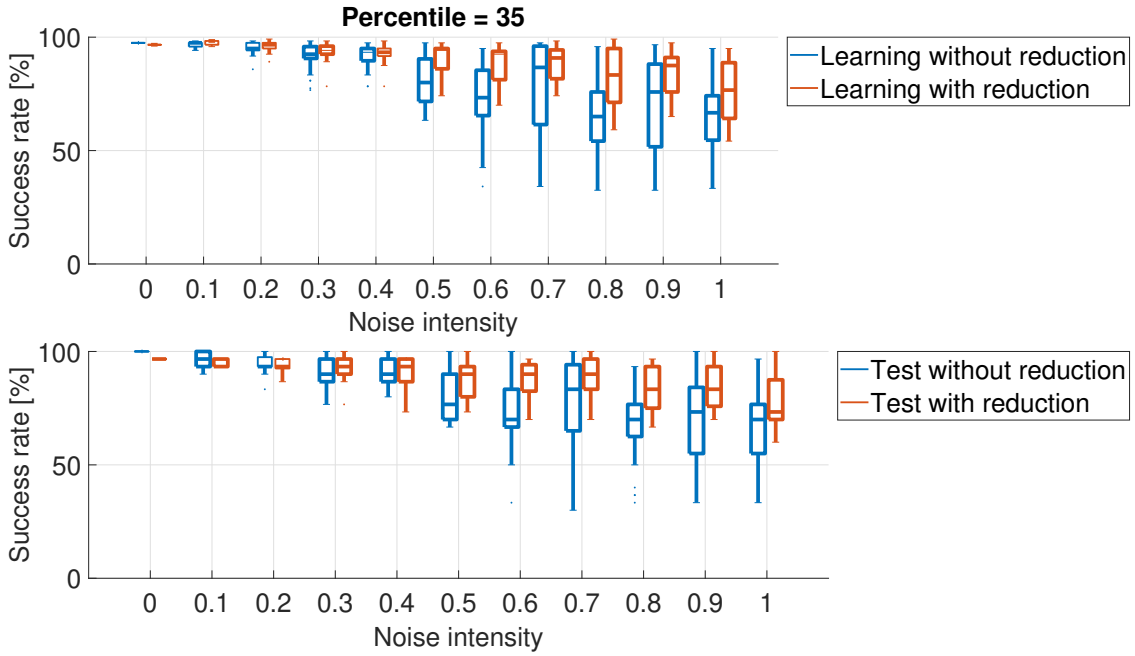

Figure 6: Normally distributed noise on success rates (35-th percentile) (*Iris* dataset). When Laplacian Eigenmaps are applied, the network is more robust to noise and success rates are often higher than those without reduction, in both learning (upper plot) and test (bottom plot).

4.2. Case study 2: Wisconsin Breast Cancer dataset

We have carried out other tests by adopting another very common and widely used dataset, the so-called *Wisconsin Breast Cancer* dataset (diagnostic version) [30, 24]. It describes some characteristics of the cell nuclei present in an image of a fine needle aspirate of a breast mass, organised in $n_p = 569$ patterns whose length is equal to $n_i = 30$. Furthermore, the number of classes is $m = 2$ and they may be either "malignant" or "benign". Again, the percentage of patterns dedicated to learning has been set to 80%. Additionally, the 35-th percentile has produced similar results to those obtained with the *Iris* dataset, concisely summarised in Figures 9 and 10, for which an average reduced dimension has been equal to 45 over 5 simulations. We have performed other tests to evaluate quantisation noise rejection as well and the results are shown in Figure 11.

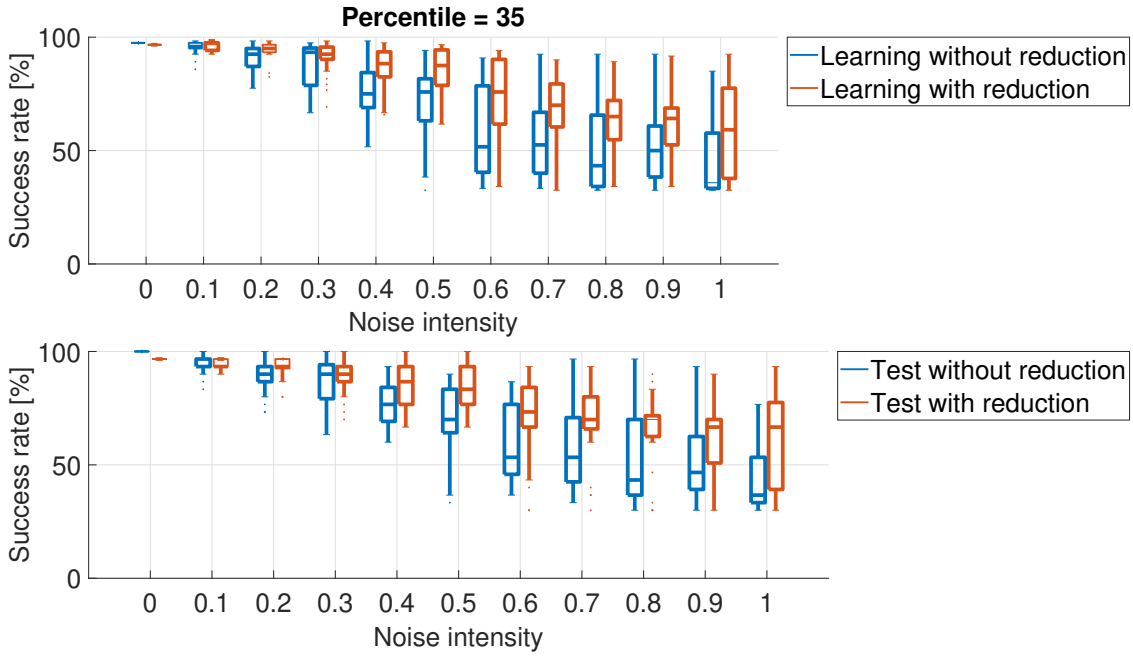

Figure 7: Uniformly distributed noise on success rates (35-th percentile) (*Iris* dataset). When Laplacian Eigenmaps are applied, the network is more robust to noise and success rates are often higher than those without reduction, in both learning (upper plot) and test (bottom plot).

5. Discussion

The approach presented in this paper is semi-empirical and data driven, but carries some useful hints for further investigations. Dimensionality reduction has shown a side advantage consisting in its robustness, in particular to quantisation effects. By analysing **both Figures 8 and 11**, it is possible to appreciate how reduced weights by means of 3-4 bits have led to a negligible decrease of the success rate in classification, even by applying the 35-th percentile. Considering that, the information of the same liquid layer, in front of the same pattern set, can be exploited for different purposes concurrently and large number of readout maps could be implemented, each one serving a specific task at the same time. In such a case, the proposed approach allows a huge memory saving when implementing the whole architecture.

Laplacian Eigenmaps method has shown good results in manifold reduction, at least for nonlinear networks like the one presented in this paper.

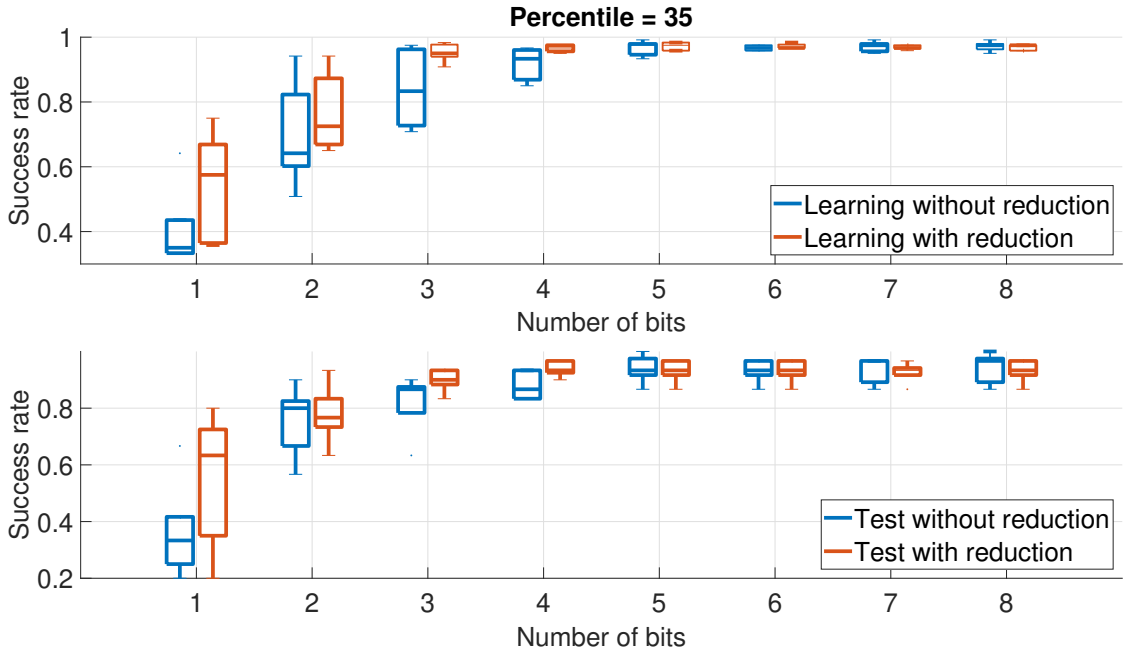

Figure 8: Statistics on success rate trends with respect several resolution values (35-th percentile) (*Iris* dataset). No sooner is the resolution increased than trained weights fall within the finite set of values $\{0, 1, ..., 2^{\text{bits}-1}\}$. Consequently, the higher the number of bits is, the wider the range in which weights can vary is.

However, something new should be highlighted for further considerations. The method we have implemented takes into account the statistical distribution of Euclidean distances among all the nodes constituting the graph over the entire set of patterns. This procedure is compulsory to build the adjacency matrix and thus the Laplacian matrix of the graph, but the result depends on what kind of similarity measure is chosen. Generally speaking, similarity measures have their own pros and cons [29], but authors consider of primary interest for further investigations the possibility of adopting different ways to cluster two or more dynamic nodes. In fact, when two nodes have some kind of time-dependent dynamics like in our case, it could be reasonable to cluster them into one single group if they are temporally similar. Such a concept could be better captured by computing the cross-correlation between these two signals. Then, a significant analysis could be referred to the employment of cross-correlation among the signals produced for each pattern, instead of considering all the patterns: this should lead both to a

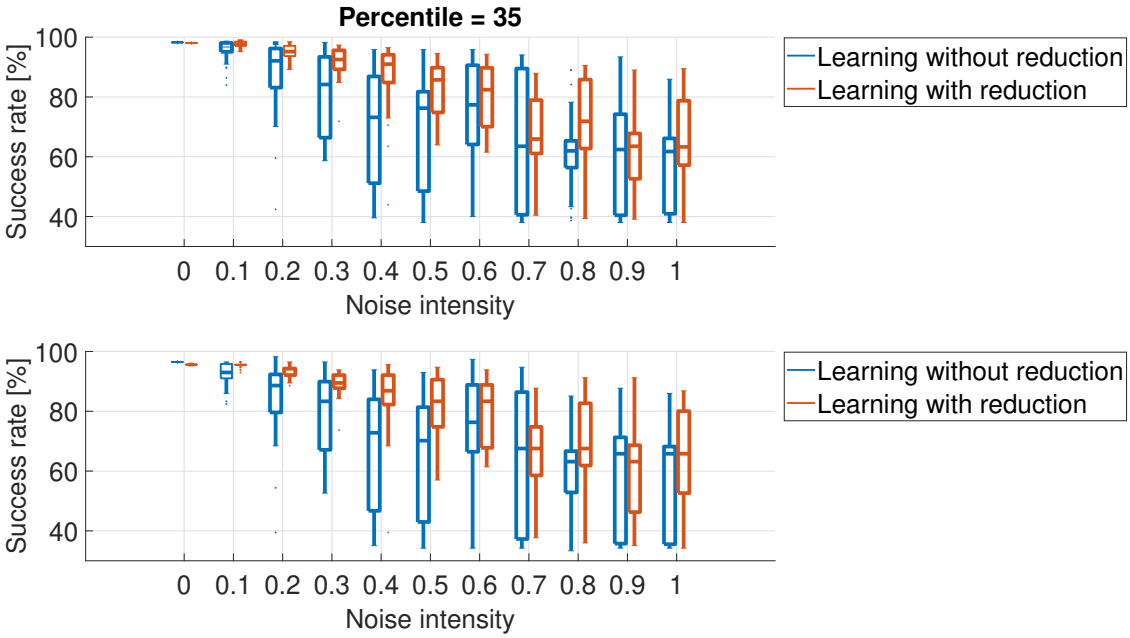

Figure 9: Normally distributed noise on success rates (35-th percentile) (*Wisconsin Breast Cancer* dataset). When Laplacian Eigenmaps are applied, the network is more robust to noise and success rates are often higher than those without reduction, in both learning (upper plot) and test (bottom plot).

dynamic graph where similarity measure changes with respect time and to a perspective analysis based on time characteristics.

6. Conclusions

In this paper, a data-based analysis focused on Laplacian eigenmaps for manifold reduction has been developed and applied for solving supervised classification tasks. In particular, a simple three-layer network, whose computational core comprises locally connected neurons forming a liquid state machine, **has been used to process both the *Iris* and the *Wisconsin Breast Cancer* datasets, which are commonly used as benchmarks to test classifier performance**. First, the understanding of how Hannan-Quinn index behaves with respect the selected percentile has been fundamental for further considerations. In particular in our percentile-based HQ calculation the best reduced manifold has often been a $q - n_{\text{zero}}$-dimensional

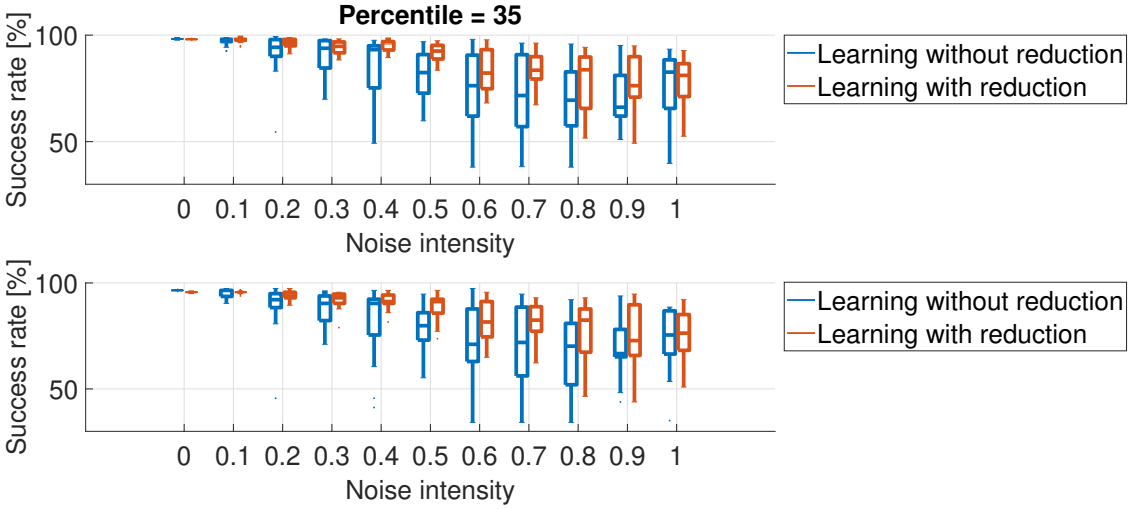

Figure 10: Uniformly distributed noise on success rates (35-th percentile) (*Wisconsin Breast Cancer* dataset). When Laplacian Eigenmaps are applied, the network is more robust to noise and success rates are often higher than those without reduction, in both learning (upper plot) and test (bottom plot).

map. However, success rates during classification have been analysed deeply and we have noticed that a lower dimension is achieved for lower percentiles, but lower percentiles do not necessarily undermine classification results. This has justified our choice to select the 35-th percentile which is able to sensibly reduce the manifold. Manifold reduction has been semi-automised in a data-driven way, according to results obtained from several simulations. Moreover, the robustness of the algebraic transformation allowing dimensionality reduction has been studied supposing two forms of noise: normally distributed noise and uniformly distributed noise. Results suggest that rejection to noise is more pronounced when the reduction transformation is applied. Furthermore, scalar invariance to product of reduction transformation has been empirically proved and exploited for introducing the possibility of quantising trained synaptic weights; this aspect could be surely further developed and exploited because once a complete discretisation of all the values is done, digital implementations are easier to assess.

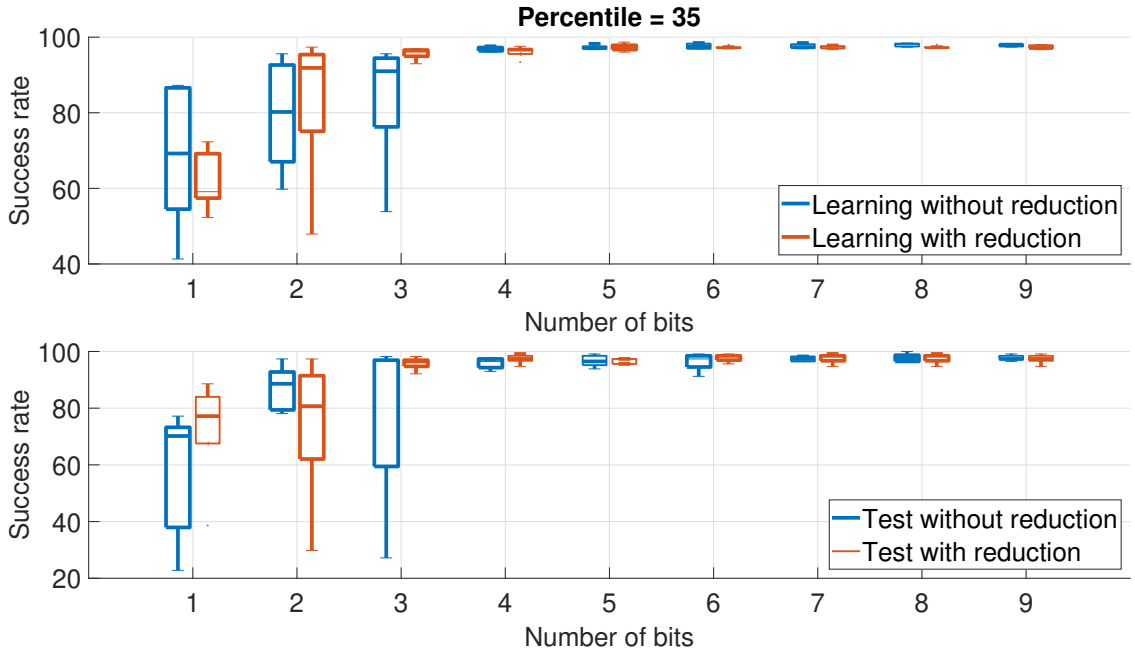

Figure 11: Statistics on success rate trends with respect several resolution values (35-th percentile) (*Wisconsin Breast Cancer* dataset). No sooner is the resolution increased than trained weights fall within the finite set of values $\{0, 1, ..., 2^{\text{bits}-1}\}$. Consequently, the higher the number of bits is, the wider the range in which weights can vary is.

Acknowledgement

This work was partially supported by the UNICT-FIR 2014 Project and by MIUR Project CLARA.

References

[1] M. L. Anderson, Neural reuse: A fundamental organizational principle of the brain, Behavioral and Brain Sciences 33 (4) (2010) 245–266.

[2] E. Arena, P. Arena, R. Strauss, L. Patané, Motor-Skill Learning in an Insect Inspired Neuro-Computational Control System, Frontiers in Neurorobotics 11 (2017) 12.

[3] P. Arena, M. Calí, L. Patané, A. Portera, R. Strauss, Modelling the insect Mushroom Bodies: Application to sequence learning, Neural Networks 67 (2015) 37 – 53, ISSN 0893-6080.

[4] P. Arena, M. Calí, L. Patané, A. Portera, R. Strauss, A Fly-Inspired Mushroom Bodies Model for Sensory-Motor Control Through Sequence and Subsequence Learning, International Journal of Neural Systems 26 (06) (2016) 1650035, pMID: 27354193.

[5] P. Arena, L. Patané, Spatial Temporal Patterns for Action-Oriented Perception in Roving Robots II: An Insect Brain Computational Model, Springer Publishing Company, Incorporated, 1st edn., ISBN 3319346954, 9783319346953, 2016.

[6] P. Arena, L. Patané, A. Spinosa, Insect inspired spatial-temporal cellular processing for feature-action learning, in: 2017 European Conference on Circuit Theory and Design (ECCTD), 1–4, 2017.

[7] M. Belkin, P. Niyogi, Laplacian Eigenmaps for Dimensionality Reduction and Data Representation, Neural Comput. 15 (6) (2003) 1373–1396, ISSN 0899-7667.

[8] R. E. Bellman, Dynamic Programming, Dover Publications, Incorporated, ISBN 0486428095, 2003.

[9] S. S. Bhople, M. A. Gaikwad, Design of Mesh and Torus Topologies for Network-On-Chip Application 2.

[10] L. Büsing, B. Schrauwen, R. Legenstein, Connectivity, Dynamics, and Memory in Reservoir Computing with Binary and Analog Neurons, Neural Comput. 22 (5) (2010) 1272–1311, ISSN 0899-7667.

[11] M. Coli, P. Palazzari, R. Rughi, The toroidal neural networks, in: 2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No.00CH36353), vol. 4, 137–140, 2000.

[12] D. L. Donoho, C. Grimes, Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data, Proceedings of the National Academy of Sciences 100 (10) (2003) 5591–5596.

[13] A. Dresden, The fourteenth western meeting of the American Mathematical Society, Bull. Amer. Math. Soc. 26 (9) (1920) 385–396.

[14] R. A. Fisher, The Use of Multiple Measurements in Taxonomic Problems, Annals of Eugenics 7 (7) (1936) 179–188.

[15] M. Han, M. Xu, Laplacian Echo State Network for Multivariate Time Series Prediction, IEEE Transactions on Neural Networks and Learning Systems 29 (1) (2018) 238–244, ISSN 2162-237X.

[16] E. J. Hannan, B. G. Quinn, The Determination of the Order of an Autoregression, Journal of the Royal Statistical Society. Series B (Methodological) 41 (2) (1979) 190–195, ISSN 00359246.

[17] H. Jaeger, Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach 5.

[18] Y. Jiang, P. Guo, Regularization Versus Dimension Reduction, Which Is Better? (2007) 474–482.

[19] I. Kupfermann, K. R. Weiss, The command neuron concept, The Behavioral And Brain Sciences 1 (1) (1978) 3 – 39.

[20] B. Leitch, G. Laurent, GABAergic synapses in the antennal lobe and mushroom body of the locust olfactory system, The Journal of Comparative Neurology 372 (4) (1996) 487–514, ISSN 1096-9861.

[21] M. Lukoševičius, A Practical Guide to Applying Echo State Networks .

[22] Q. Ma, L. Shen, W. Chen, J. Wang, J. Wei, Z. Yu, Functional echo state network for time series classification, Information Sciences 373 (2016) 1 – 20, ISSN 0020-0255.

[23] G. Manganaro, P. Arena, L. Fortuna, Springer Series in Adv. Microelectronics, ISBN 978-3-642-60044-9, 1999.

[24] O. L. Mangasarian, W. N. Street, W. H. Wolberg, Breast Cancer Diagnosis and Prognosis Via Linear Programming, Operations Research 43 (4) (1995) 570–577.

[25] L. M. Masuda-Nakagawa, K. Ito, T. Awasaki, C. J. O'Kane, A single GABAergic neuron mediates feedback of odor-evoked signals in the mushroom body of larval Drosophila, Frontiers in Neural Circuits 8 (2014) 35, ISSN 1662-5110.

[26] A. Neumaier, Solving Ill-Conditioned and Singular Linear Systems: A Tutorial on Regularization, SIAM Review 40 (3) (1998) 636–666.

[27] T. G. Robertazzi, Toroidal networks, IEEE Communications Magazine 26 (6) (1988) 45–50, ISSN 0163-6804.

[28] N. Sakthivel, B. Nair, M. Elangovan, V. Sugumaran, S. Saravanmurugan, Comparison of dimensionality reduction techniques for the fault diagnosis of mono block centrifugal pump using vibration signals 17 (2014) 30–38.

[29] A. S. Shirkhorshidi, S. Aghabozorgi, T. Y. Wah, A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data, PLOS ONE 10 (12) (2015) 1–20.

[30] W. N. Street, W. H. Wolberg, O. L. Mangasarian, Nuclear feature extraction for breast tumor diagnosis, 1993.

[31] L. J. P. van der Maaten, E. O. Postma, H. J. van den Herik, Dimensionality Reduction: A Comparative Review, 2008.

[32] M.-H. Yusoff, J. Chrol-Cannon, Y. Jin, Modeling neural plasticity in echo state networks for classification and regression, Information Sciences 364-365 (2016) 184 – 196, ISSN 0020-0255.