Online Spectral Clustering and the Neural Mechanisms of Concept Formation

Stefano Rovetta[1] and Francesco Masulli[1,2]

[1] DIBRIS – University of Genova, Via Dodecaneso 35, 16146 Genova, Italy
[2] Temple University, Philadelphia PA, USA
{stefano.rovetta,francesco.masulli}@unige.it

Abstract. Spectral clustering can provide surprising performances. As all kernel methods, is uses a similarity matrix, whose size grows with n^2, and it requires to solve a possibly large eigenproblem. In this paper we focus on a method for spectral embedding of stream data, modeled as an unbounded quantity of input observation. A second purpose of this work is to analyze the proposed method and compare it with traditional neural network implementations: current knowledge about computations in neurons and the brain does not contrast with the computing primitives required for a local implementation of the proposed technique. A hypothesis stemming from this work could be that concept formation and discrimination in neurons and the brain could be explained by a spectral embedding framework.

Keywords: Spectral clustering, Online learning, Concept formation, Unsupervised learning, Neural networks.

1 Introduction

Spectral clustering is a family of unsupervised machine learning techniques capable of providing surprising performances, such as the detection of clusters of more or less arbitrary shape [26]. From the cognitive modeling standpoint, two weaknesses of spectral clustering are that solutions in non-clusterable data sets tend to be less meaningful than with other, more traditional techniques such as k means; and that, especially with data embedded in $\mathbb{R}^d$ where the heat kernel

$$K(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|/\sigma^2} \tag{1}$$

is commonly used, in the presence of clusters of different densities the choice of the parameter σ is critical and it may not be possible to find a unique optimal value [28,15]. Generalization, or "out-of-sample extension", is not directly provided by these methods, but several techniques can be used to this purpose [4,10,8]

The present work, however, is concerned with one specific computational limitation of the method. Spectral clustering, as all kernel methods [9], is based on the use of a Gram (or similarity) matrix, whose size grows with n^2 (where n is the data set cardinality), and therefore computations usually scale as n^3. The methods require solving an eigenproblem, with related computational complexity.

S. Bassis et al. (eds.), *Recent Advances of Neural Networks Models and Applications,*
Smart Innovation, Systems and Technologies 37, DOI: 10.1007/978-3-319-18164-6_7

In this paper we focus on the problem of providing a spectral embedding solution to the problem of clustering stream data, which can be modeled as an unbounded quantity of input observation ($n \to \infty$). This is motivated by the growth of available raw stream data. For instance, some applications currently receiving a lot of attention are wearable sensors for health monitoring, data from mobile devices in crowd and traffic management in "smart city" projects, and sensors for ambient-assisted living. Clearly, solving this problem requires some approximations to the method, which will be introduced in Section 4. We provide a technique that, while not directly tested here on stream data, is nevertheless optimized by online training, making it suitable in this framework.

A second purpose of this work is to analyze the proposed method and compare it with traditional neural network implementations. We will see that current knowledge about computations in neurons and about the structure of several brain areas, for instance those related to early vision, does not contrast with the computing primitives required for a local implementation of the proposed technique.

A hypothesis stemming from this work could be that, contrary to what is suggested by many "canonical" models of computation in the brain based on simple maximum similarity matching (e.g., the "prototype effect" [12] in perception and recognition), concept formation and discrimination could be explained by a more flexible and powerful spectral embedding framework.

2 Spectral Clustering

Spectral clustering is a clustering criterion that can be justified as arising from spectral graph partitioning [6] or from several other principles, such as random walks, diffusion phenomena and the heat equation, or Laplacian-of-Gaussian filters for edge detection. In the case of data embedded in $\mathbb{R}^d$, it uses a similarity matrix W to construct a neighborhood graph, and then it analyzes the spectral properties of this graph by studying the eigenvalues and eigenvectors of the graph Laplacian $L = D - W$ or one of its normalized versions, $L_{\text{rw}} = I - D^{-1}W$ or $L_{\text{sym}} = I - D^{-1/2}WD^{-1/2}$, where D is the diagonal degree matrix whose values are the row (or column) sums of W.

The eigenvectors and eigenvalues of graph Laplacians provide information about the number of connected components of the graph, although in different forms depending on the normalization. In particular, L and L_{rw} have piece-wise constant eigenvectors, all corresponding to eigenvalue 0, and whose multiplicity equals the number of connected components of the graph. These eigenvectors are indicator vectors of the connected components, being 0/1 valued. If clusters are defined in a more general and realistic way, i.e., as sub-graphs which are not connected components, but have stronger within-group connectivity than between-group connectivity, then the eigenvectors are still approximate indicator vectors.

The algorithm by Ng, Jordan and Weiss [16] is slightly different in that it uses a complementary but equivalent definition of L_{sym}, which is $L'_{\text{sym}} = I - L_{\text{sym}}$ and has the same eigenvectors; moreover if λ_i is an eigenvalue for L_{sym} then $1 - \lambda_i$ is an eigenvalue for L'_{sym}, so that this approach studies the largest-valued (ideally 1), as opposed to smallest-valued (ideally 0), eigenvalues of the Laplacian spectrum.

After computing the eigendecomposition of the Laplacian, the top eigenvectors are arranged as the columns of a matrix; then a spectral embedding is performed, where the l-th data point is represented by the l-th row of the matrix. Finally these representations, after having been row-normalized, are clustered with a simple method (often k means).

A very good introduction to the different flavors of spectral clustering is provided in [26]. In ref. [9] an overall survey of the properties of spectral, as well as kernel-based clustering is provided.

3 Online and Incremental Versions of Spectral Clustering

Due to the mentioned computational limitations, spectral clustering has been the subject of several modifications. These fall into two typical broad categories. The first one is that of exact algorithms which exploit the sparsity of graph data, i.e., the fact that the number of edges is less than $n(n-1)/2$. The second category is approximated algorithms, where the approximation may apply to the data (not all data are kept), to the similarity matrix, or to other aspects. Many of these modifications are iterative and can be used for online training, although not all are suitable for the clustering of stream data.

An instance of the first category is the approach by H. Ning [17] which directly updates the eigenvectors and eigenvalues by decomposing the graph and identifying only the individual elements that need updating. It applies only to cases of graph with limited connectivity; the approach is suitable for instance for studying the Internet graph, although it relies on some hypotheses on the extent of modifications required at each updating step, which should be limited.

The second category is represented by the “Nyström method”, i.e., the use of the Nyström formula to obtain a reduced-rank approximation of a Gram (similarity) matrix, which has been proposed for use in spectral clustering in [10]. The “fast approximate” method from [27] approximates the data rather than the eigensystem by using a pre-clustering step, to which spectral clustering is then applied.

4 An Approximated, Online Spectral Clustering Method

Rather than approximating the data or the eigensystem, in this work we propose to approximate the matrix W and therefore the normalized Laplacian. The technique includes two steps: approximation and eigendecomposition.

For the first step, approximation, we assume that the input data $\mathbf{x}$ are satisfactorily described as realizations of a stationary, discrete-time, stochastic vector process

$$\mathbf{x} = \mathbf{c}_j + \nu, \qquad j \in \{1 \ldots m\}, \tag{2}$$

where j is a random integer between 1 and m and ν is a random (noise) term for which we assume a reasonable probability distribution, i.e., unimodal, symmetric, and zero-centered. Therefore we approximate the data with a set $\{\mathbf{c}_1, \ldots, \mathbf{c}_c\}$ of reference or landmark [7] points which are optimized to minimize a mean squared distortion criterion

$$J = \int (\mathbf{x} - \mathbf{c}(\mathbf{x}))^2 \, p(\mathbf{x}) d\mathbf{x}, \tag{3}$$

where $\mathbf{c}(\mathbf{x})$ is the nearest landmark to data point $\mathbf{x}$. This is a vector quantization problem. We require that its solution approximately reflect the data distribution $p(\mathbf{x})$, but it does not necessarily have to pinpoint any structure (clusters) within it. A vector quantization problem is usually solved by stochastic approximation methods.

The landmark points are then used for approximating the asymmetric normalized Laplacian

$$L_{\mathrm{rw}} = D^{-1}W \tag{4}$$

by replacing the computation of the similarity $K(\mathbf{x},\mathbf{x}')$ of any given data point to the remaining points $\mathbf{x}'$, a set of possibly infinite cardinality in our hypotheses, with the similarity of the same point $\mathbf{x}$ to each landmark $\mathbf{c}_j$, a set of finite cardinality m. The chosen similarity function $K(\cdot,\cdot)$ can be for instance the heat kernel (1). The task is therefore that of identifying the similarity matrix $W_{jk} = K(\mathbf{c}_j,\mathbf{c}_k)$ from a possibly unbounded sequence of observed samples $\mathbf{x}$ generated according to model (2). Note that, according to the asymmetric normalization chosen, if the current sample is $\mathbf{x} = \mathbf{c}_j + \nu$ we have

$$L_{jk} = \frac{W_{jk}}{\sum_h W_{jh}} \approx \frac{K(\mathbf{x},\mathbf{c}_k)}{\sum_h K(\mathbf{x},\mathbf{c}_h)}, \tag{5}$$

which depends only on $\mathbf{x}$, not on other samples as it would in the case of the symmetrical normalization $D^{-1/2}WD^{-1/2}$.

In the second step the structure of the data distribution is analyzed by means of the eigendecomposition of the normalized Laplacian. Since in this setting the Laplacian is noisy (random) and given by a sequence of row vectors, each of the form

$$\left[\frac{K(\mathbf{x},\mathbf{c}_1)}{\sum_h K(\mathbf{x},\mathbf{c}_h)}, \dots, \frac{K(\mathbf{x},\mathbf{c}_m)}{\sum_h K(\mathbf{x},\mathbf{c}_h)}\right],$$

then we may use Oja's subspace rule [19,18] which gives the eigendecomposition of a matrix $S^T S$, known through a sequence of noisy samples of S; the eigenvectors of $S^T S$ are the same as the right eigenvectors of S (while the eigenvalues are squared w.r.t. those of S). Note that, since this algorithm requires centered data, the mean input vector is also learned as a set of bias terms, and then subtracted.

To complete the spectral clustering process, the embedded data should be clustered, usually by k means. However, due to the properties of the spectral embedding, this last step is usually almost trivial. In this work we will mainly focus on embedding.

To sum up, the following is an outline of the proposed online algorithm:

1. Input one pattern $\mathbf{x}$
2. Compute similarities from landmarks: $K(\mathbf{x},\mathbf{c}_j)$
3. Compute the corresponding row of the normalized Laplacian: $\lambda_j = \frac{K(\mathbf{x},\mathbf{c}_j)}{\sum_h K(\mathbf{x},\mathbf{c}_h)}$
4. Update landmarks
5. Compute one subspace step using Laplacian row as input
6. Update subspace projection
7. Go to step 1

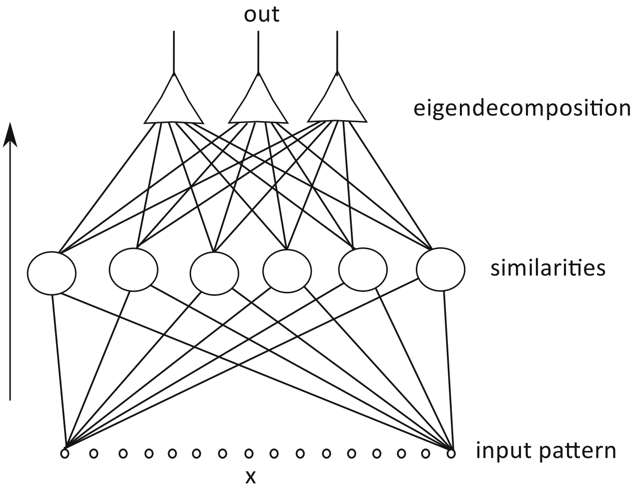

Fig. 1. Graphical representation of the method. Small circles are inputs; large circles compute similarities, each unit storing one prototype $\mathbf{c}_j$; triangles compute eigendecomposition of the approximated Laplacian, each unit computing the projection on one dimension of the embedding space. For clarity not all connections between inputs and similarity units are shown.

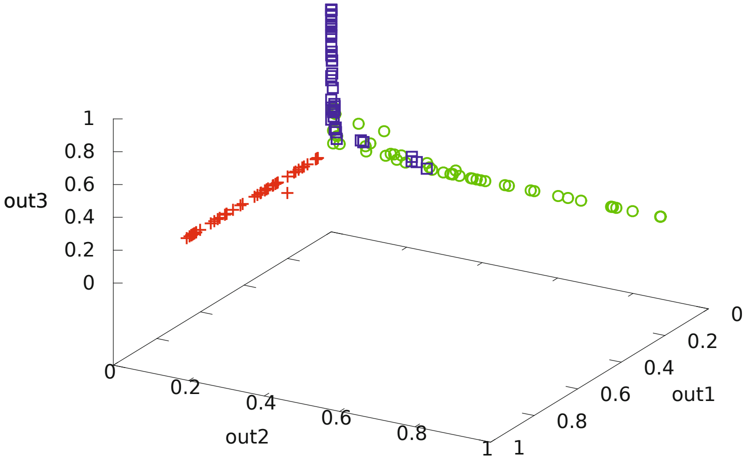

Fig. 2. Iris data: The spectral embedding obtained. Crosses is *Setosa*, circles is *Virginica*, and squares is *Versicolor*; ”out1”, ”out2” and ”out3” indicate three output components.

5 Experimental Results

Some experiments have been performed to check the consistency of the proposed method with standard approaches, rather than proving its quality in absolute terms. The data sets used are Anderson's Iris data [2], a data set composed of two concentric circles ("circles", see left of Fig. 3), and a data set of random samples from the letters W I R N ("WIRN", see top of Fig. 4).

Anderson's Iris data needs little presentation. It is a three-class dataset with 4 inputs (petal width, petal length, sepal width, sepal length, all expressed in centimeters) and 50 instances per class, for a total of 150 patterns. The data set was downloaded from the UCI Machine Learning repository [3].

The circles and WIRN data were generated by randomly sampling points from 2-dimensional geometrical structures within the (dimensionless) square $[0,1] \times [0,1]$, as shown in the respective figures, so they are both 2-input data sets. The circles data has two clusters and 200 instances per cluster (total cardinality: 400), whereas the WIRN data has four clusters and instances distributed as follows: 499 in cluster "W", 226 in "I", 469 in "R" and 376 in "N" (total cardinality: 1570). The different cardinalities are due to sampling different shapes and sizes with uniform random density.

The latter two experiments can be directly compared with the results presented in [16] on similar data. Note, however, that both in traditional approaches and in the online version proposed here the results are strongly dependent on the choice of σ, therefore comparisons, even those contained in [16], are not necessarily fair.

The method was implemented in C++. Vector quantization was performed with a centroid optimization heuristic similar to online k means, with some degree of interaction between the best-matching vector and the remaining ones, to reduce the risk of false minima; the interaction degree was annealed during training with an exponential

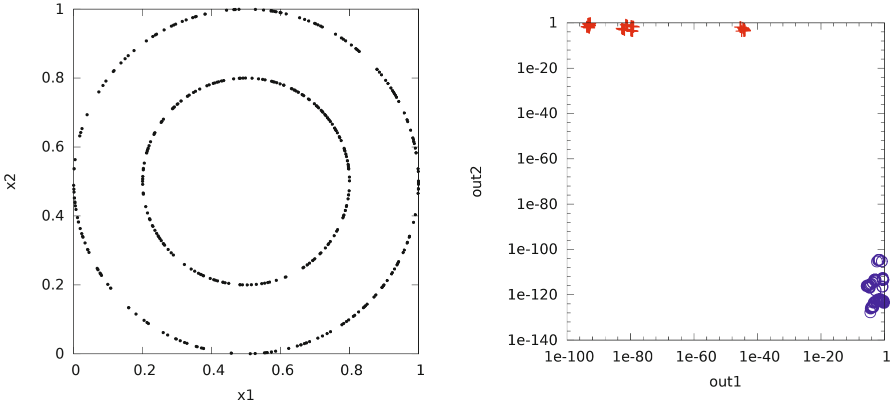

Fig. 3. The "Circles" data set. Left, data; "x1" and "x2" indicate two input components. Right: Spectral embedding (learned representation), log-log scale; "out1" and "out2" indicate two output components.

rule. Eigendecomposition and mapping was performed with Oja's generalized Hebb rules plus an orthogonalization step (a method roughly equivalent to Sanger's GHA rule [24]). Both steps are purely online, with no memory required in addition to the already described quantities: landmarks, eigenvectors, bias terms.

The figures show the data and the results of the spectral embeddings obtained. These are graphs with axes representing the two or three components of the embeddings themselves (actually the plotted values are the outputs of the network corresponding to each input pattern presented). From the figures it is clear that the subsequent, actual clustering is trivial with circles and WIRN, while for Iris, which only contains two separable clusters, some misattributed data remain, as it is to be expected.

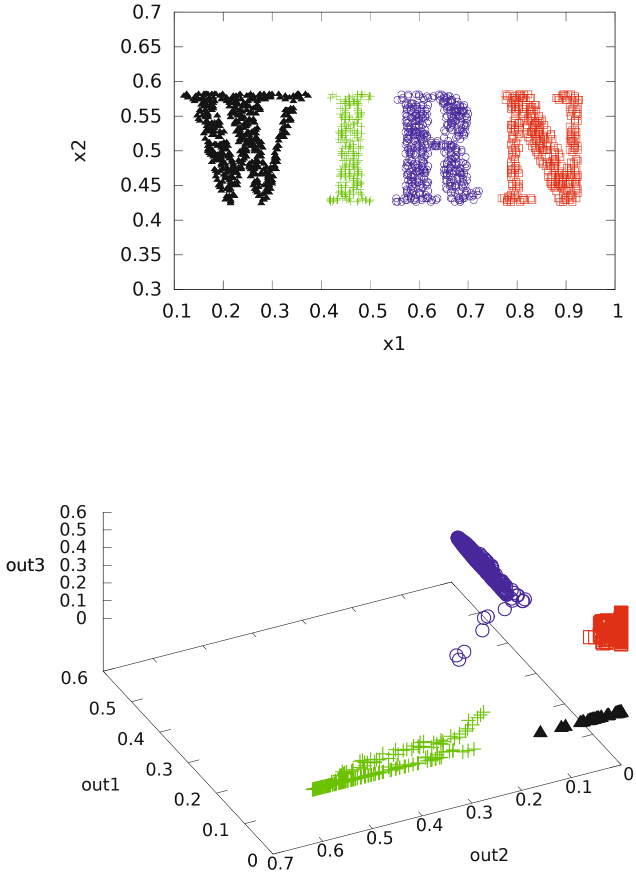

Fig. 4. The "WIRN" data set. Top, data labeled after the clustering result; "x1" and "x2" indicate two input components. Bottom, spectral embedding (learned representation); "out1", "out2" and "out3" indicate three output components.

For the WIRN data also the result of the final clustering step, using k means on the normalized output patterns, is presented.

In general, the embeddings have a dimensionality that equals the number of clusters sought, so for Iris all three components are shown. For WIRN we show three of the four components.

6 A Neural Implementation

An intriguing property of the method presented is that it is completely local and, as already noted, it requires constant memory w.r.t. data cardinality, therefore it is a good candidate for a distributed implementation. But we will go as far as showing that the required computational primitives are indeed *not incompatible* with commonly accepted input-output response models found in the nervous system, for instance in early vision stages. This suggests that the operating mechanism of some areas in the nervous system could be actually implementing a form of spectral embedding for learning representations.

Regarding the landmark set $\{\mathbf{c}_j\}$, a basic competitive update rule was used, as described in the experimental section.

As the similarity function $K(\cdot,\cdot)$, up to now we have referred to the Gaussian similarity or heat kernel (1). However, we note that a similar function can be obtained with a standard linear threshold formal unit, modified to include a sum-of-squared-inputs term as follows:

$$r(\mathbf{x}) = w_q \mathbf{x}\cdot\mathbf{x} + \mathbf{w}\cdot\mathbf{x} + w_0 \qquad a(r) = \frac{1}{1+e^{-r}}, \tag{6}$$

where $\mathbf{x}\cdot\mathbf{x} = \sum_{i=1}^{d} x_i^2 = \|\mathbf{x}\|^2$, r is the net stimulus and a the activation value, obtaining what has been termed a *circular perceptron* [22] due to the hyper-spherical shape of its discriminant surface. With suitable constraints on the weights w_q, $\mathbf{w} = [w_1,\dots,w_d]$, and w_q this model can be interpreted as implementing the formal neural function

$$r(\mathbf{x}) = \frac{\|\mathbf{x}-\mathbf{c}\|^2+\theta}{\sigma^2} \qquad a(r) = \frac{1}{1+e^{-r}} \tag{7}$$

by means of the following conversions:

$$\begin{cases} w_q = \frac{1}{\sigma^2} \\ \mathbf{w} = -\frac{\mathbf{c}}{\sigma^2} \\ w_0 = \frac{\|\mathbf{c}\|^2-\theta}{\sigma^2} \end{cases} \Leftrightarrow \begin{cases} \sigma = \frac{1}{\sqrt{w_q}} \\ \mathbf{c} = -\frac{\mathbf{w}}{w_q} \\ \theta = \frac{\|\mathbf{w}\|^2-w_0}{w_q} \end{cases}$$

A network including these “circular neurons” has been shown [23] to be equivalent in several respects to a vector quantization network. The presence of a quadratic term introduces a (biologically plausible [11]) dependence of the output response on the overall input intensity, not only on the net input. Figure 5 shows the excellent degree of coincidence between the circular activation function and the Gaussian one, attainable with suitable parameter values.

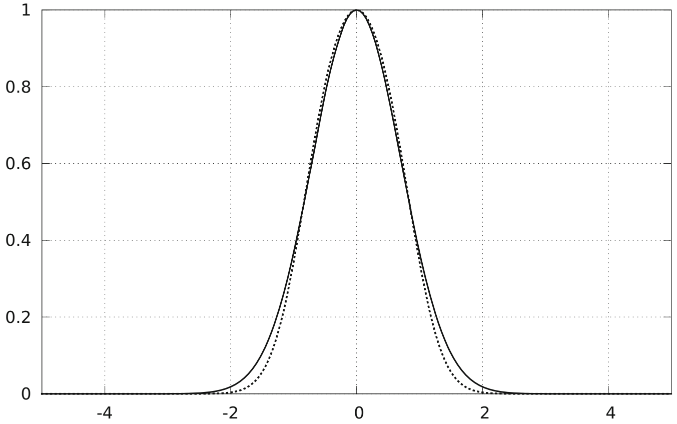

Fig. 5. Comparison of circular unit (dotted) and Gaussian (continuous) activations as functions of their net stimulus.

Competitive learning rules and orthogonalization can also be explained by means of Heeger and Carandini's normalization model [5], which explains experimental data that indicate the presence of a net inhibitory effect of neurons within a group, even in the absence of inhibitory synapses. This effect in turn can be explained in the framework of retrograde signaling [1], neural backpropagation [25], and neuromodulation. These considerations support and reinforce the common idea that both competitive updating rules and the Hebbian learning rule are biologically plausible and indeed may be a model of some mechanisms of synaptic plasticity in the central nervous system.

7 Concept Formation

During the experiments it has been observed that, after an initial phase of "ramping up", adaptation proceeds quite smoothly. As soon as some structure appears in the set of landmarks, the second layer can start learning meaningful eigencomponents. These usually do not change abruptly; it appears that the evolution stays smooth as long as there is no change in attraction basins of landmarks in the first layer. This makes the selection of prototypes in the first layer not critical, as also observed in [7] with experiments on the landmark MDS, a method that shares some elements with the present one.

By performing several experiments it has additionally been observed that, for a suitable selection of σ, or equivalent parameters in case of different formulations of the prototype units, for well separated clusters the embedding tends to be binary, i.e., most coordinates of the embedded point are zero and only one is significantly different from zero. This tendency to form sparse representation is a known property of the Laplacian eigensystem. Moreover, except for possible arbitrary axis permutations, the locus of embedding points tends to stay about the same regardless of the actual location of landmarks, which depends on the random initialization.

The proposed model therefore shows some very interesting properties:

- Similarly to its standard counterparts (spectral clustering methods), it automatically points out interesting structure from fairly complex data distributions, with higher flexibility than prototype-based models.
- It intrinsically produces sparse internal representations starting from distributed intermediate representations, that tend to move from analog to binary.
- Different exemplars produce similar internal models. This may provide them with an ability to develop a “theory of mind”, i.e., a shared representation for concepts and consequently the possibility to model other individual’s internal state.

It can be noted that all these properties are obtained by only using computational primitives that are commonly accepted as biologically plausible and have been in use for decades.

It is interesting to note how the observed behaviour is also compatible with experiments and theories of concept formation as observed in the human brain, as well as in primates and other mammals. For instance, in the case of vision, the representation shift from distributed to localized is a well-known organizational aspect: the initial representation is obviously completely distributed on the individual receptors of the retina, then it gets organized into receptive fields [13]; in specific areas of the visual cortex it is gradually made more selective, where neurons traditionally termed ”complex” and ”hypercomplex” cells [14] implement a hierarchy of sparse distributed representations; finally, there is substantial experimental evidence [21] of the existence of ”concept cells” in the medial temporal lobe, that implement a completely localist representation individually elicited by abstract cognitive tasks such as recognizing the face, or even just reading the name, of a known person (e.g., ”Jennifer Aniston neurons” [20]).

8 Conclusions and Future Work

The work presented in this paper is just an initial proposal, and several issues should be further investigated. Among these, the technical problems of selecting a proper value for σ and a proper scheduling (or online modulation) for the learning steps. This latter point can be an opportunity to develop a supervised version of the network, as well as to incorporate some form of novelty detection to modulate the stability/plasticity dilemma.

A combined landmark-eigensystem learning method can also be investigated, to possibly reduce the number of model parameters.

Finally, the model is especially well suited for being employed in modular and multilayer, possibly deep structures, since it is unsupervised and for training it only uses information local to each layer.

References

1. Alger, B., Pitler, T.: Retrograde signaling at GABAA-receptor synapses in the mammalian CNS. Trends in Neurosciences 18(8), 333–340 (1995)
2. Anderson, E.: The irises of the gaspe peninsula. Bulletin of the American Iris Society 59, 25 (1935)
3. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
4. Bengio, Y., Paiement, J.F., Vincent, P., Delalleau, O., Le Roux, N., Ouimet, M.: Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. Mij 1, 2 (2003)
5. Carandini, M., Heeger, D.J.: Normalization as a canonical neural computation. Nature Reviews Neuroscience 13(1), 51–62 (2012)
6. Chung, F.R.K.: Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92). American Mathematical Society (February 1997)
7. De Silva, V., Tenenbaum, J.B.: Sparse multidimensional scaling using landmark points. Tech. rep., Technical report, Stanford University (2004)
8. Drineas, P., Mahoney, M.W.: On the nyström method for approximating a gram matrix for improved kernel-based learning. The Journal of Machine Learning Research 6, 2153–2175 (2005)
9. Filippone, M., Camastra, F., Masulli, F., Rovetta, S.: A survey of kernel and spectral methods for clustering. Pattern Recognition 40(1), 176–190 (2008)
10. Fowlkes, C., Belongie, S., Chung, F., Malik, J.: Spectral grouping using the nystrom method. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(2), 214–225 (2004)
11. Heeger, D.J.: Half-squaring in responses of cat striate cells. Visual Neuroscience 9, 427–443 (1992)
12. Homa, D., Cornell, D., Goldman, D., Shwartz, S.: Prototype abstraction and classification of new instances as a function of number of instances defining the prototype. Journal of Experimental Psychology 101(1), 116 (1973)
13. Hubel, D.H., Wiesel, T.N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology 160(1), 106 (1962)
14. Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. The Journal of Neurophysiology 28(2), 229 (1965)
15. Nadler, B., Galun, M.: Fundamental limitations of spectral clustering. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems 19, pp. 1017–1024. MIT Press, Cambridge (2007)
16. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems 14. MIT Press, Cambridge (2002)
17. Ning, H., Xu, W., Chi, Y., Gong, Y., Huang, T.S.: Incremental spectral clustering by efficiently updating the eigen-system. Pattern Recognition 43(1), 113–127 (2010)
18. Oja, E.: Neural networks, principal components, and subspaces. International Journal of Neural Systems 01(01), 61–68 (1989)
19. Oja, E., Karhunen, J.: On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. Journal of Mathematical Analysis and Applications 106(1), 69–84 (1985)
20. Quiroga, R.Q., Reddy, L., Kreiman, G., Koch, C., Fried, I.: Invariant visual representation by single neurons in the human brain. Nature 435(7045), 1102–1107 (2005)
21. Quiroga, R.Q.: Concept cells: the building blocks of declarative memory functions. Nature Reviews Neuroscience 13(8), 587–597 (2012)
22. Ridella, S., Rovetta, S., Zunino, R.: Circular back–propagation networks for classification. IEEE Transactions on Neural Networks 8(1), 84–97 (1997)

23. Rovetta, S., Zunino, R.: Circular backpropagation networks embed vector quantization. IEEE Transactions on Neural Networks 10(4), 972–975 (1999)
24. Sanger, T.D.: Optimal unsupervised learning in a single-layer linear feedforward neural network. Neural Networks 2(6), 459–473 (1989)
25. Stuart, G., Spruston, N., Sakmann, B., Häusser, M.: Action potential initiation and backpropagation in neurons of the mammalian CNS. Trends in Neurosciences 20(3), 125–131 (1997)
26. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
27. Yan, D., Huang, L., Jordan, M.I.: Fast approximate spectral clustering. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 907–916. ACM (2009)
28. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in Neural Information Processing Systems, pp. 1601–1608 (2004)