ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 287-294

Developmental pruning of synapses and category learning

Roberto Viviani and Manfred Spitzer
Universitätsklinikum Ulm, Abteilung Psychiatrie III, Leimgrubenweg 12-14,
89075 Ulm, Germany

Abstract After an initial peak, the number of synapses in mammalian cerebral cortex decreases in the formative period and throughout adult life. However, if synapses are taken to reflect circuit complexity, the issue arises of how to reconcile pruning with the increasing complexity of the representations acquired in successive stages of development. Taking these two conflicting requirements as an architectural constraint, we show here that a simple topographic self-organization process can learn increasingly complex representations when some of its synapses are progressively pruned. By addressing the learning-theoretic properties of increasing complexity, the model indicates how pruning may be computationally advantageous. This suggests a novel interpretation of the interplay between biological and acquired patterns of neuronal activation determining topographic organization in the cortex.

1. Introduction

Several studies on the development of the cortex of primates (1, 2) and of humans (3, 4) have investigated the time course of synaptic development. After the first phase of explosive proliferation, the number of synapses gradually decreases, with an acceleration around puberty. With the possible exception of the prefrontal areas, proportions and timing of synaptic development appear to be similar across the cortex (5-7).

The exact functional significance of synaptic pruning remains unclear. While there is some consensus that early synaptogenesis is endogenously regulated, it has been proposed that sensory experience directs synaptic pruning, for example as a selective mechanism driven by experience (8). In the human being, the prolonged learning phase means that representations acquired in successive stages of development become increasingly complex (9, 10). Within the connectionist framework, Quinn and Johnson (11) have shown that the features of generalizations that are typical of semantic categories in children can be explained by the gradual formation of internal representations whose initial complexity is low (for a review, see 12). However, synaptic pruning and increasing complexity are difficult to reconcile. In the network architectures often used to model cognitive processes, such as multi-layer perceptrons, high degrees of complexity of the network and of the functions it can approximate necessarily depend on large numbers of weights (13). Since these architectures are highly idealized models of a biological network, one might assume that in real brains there are plenty of ways to prune synapses and improve the quality of learning. However, intuitions such as this do not take into account that very little is required from the weights of a network to increase the complexity of the functions that can be represented. Consider, for example, the case of a layer computing a nonlinear transformation of the input (such as a Gaussian receptive field). Random weights modifying the input will contribute to the overall computational power of the network, and hence to the complexity of the patterns that can be learned. In general, in the

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 287-294

presence of nonlinear transformations it is not necessary that the weights be modifiable or chosen in any particular way; instead, it is enough that there be many of them (14). Indeed, even a set of fixed random weights plugged into a linear perceptron forms a learning machine that overcomes the complexity limits of the perceptron (15). In this study we will show that a simple topographic self-organizing process allows complexity to grow under pruning (16). Our network implements a minimal topographic self-organizing algorithm, and constitutes a variation on well-known schemes such as SOM (17, 18). Its distinctive feature consists in its capacity to characterize the input also at intermediate stages of learning. Like other self-organizing architectures, this network can be used to discover clusters or distortion-minimizing 'prototypes' from the input distribution, a task also known as 'quantization' (19). Each output unit stores a prototype of the input in the weights of the connections from the input layer ('code vector'). This architecture has many points of contact with models of category learning in cognitive psychology, where the prototypes model conceptual categories (20), but in slightly different form has also been used to model primary sensory cortical areas (21, 22). In our study, however, we will be primarily concerned with learning processes in the isocortex, for which there is also evidence of topographic organization (23).

2. Network architecture

The network is composed of an input and an output layer. The units in the output layer are arranged to form a lattice structure that defines the distance between them. To avoid edge effects, units at the borders of the lattice are considered adjacent. In the simulations that follow, the lattice is one-dimensional. The network is composed of three groups of weights. The first group consists of the weights connecting the input to the output layer. These are the only weights that learn adaptively during training, eventually storing the prototypes. The remaining weights constitute the 'intrinsic' connections between units within the output layer. They can be either excitatory or inhibitory, thus forming the second and third groups. Weights of the first group are initialized to small random values. Weights of the second and third groups are initialized according to an exponential function of the distance between the units they connect:

$$w_{j'j}^{\text{intr}} = \frac{a}{1+\exp\left[-b\ \left(r-\left|j-j'\right|\right)\right]},$$

where $|j - j'|$ is the distance of the two units in the output map. For excitatory weights, we used $a = 1.8$, $b = 0.2$, $r = 11.0$; for inhibitory weights, $a = -1.3$, $b = 0.1$, $r = 31.0$. (In this and the equations that follow, an output unit is written y_j when we refer to its activation, j when we refer to its position in the output map. The intrinsic weights in the output layer are denoted by the symbol w^{intr}, and when we distinguish between excitatory and inhibitory weights, w^{exc} and w^{inh}. The symbols w without the special superscript refer to the weights between input and output layer that are adapted during training. Vectors are in bold; hence, $\mathbf{w}_j$, $\mathbf{w}_{j'}$ are the prototypes attached to the units j and j' in the map.) The slope of the exponential function initializing the excitatory weights is steeper than that of the inhibitory weights. Thus, the combined effect of the second and third groups of weights is that units in the output layer excite

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 287-294

their immediate neighbors and inhibit units further away. The result is the well-known 'Mexican hat' architecture.

The spread of activation is accomplished in two successive phases. In the first phase, units in the output layer behave as receptive fields tuned to the Euclidean distance between each weight vector and the input:

$$y_j = \exp\left(\frac{-\left\| \mathbf{w}_j - \mathbf{x} \right\|^2}{r_I^2} \right),$$

where $\mathbf{x}$ is the input vector, and r_I the width of the receptive fields. In all simulations, r_I was set to 4.0. At this point, the 'winner unit' is defined as the unit with the largest activation (nearest neighbor rule), and its index in the output lattice as the 'code' of the current input. In the second phase, the output layer is updated through its own intrinsic connections:

$$y_{j'} = \frac{2}{1 + \exp\left[-g \cdot \sum_{j=0}^{N} y_j \left(w_j^{\text{exc}} + w_j^{\text{inh}} \right) \right]} - 1,$$

which is the logistic function with a range between –1 and 1 and gain g, which was set to 8.0 in the simulations. Because they are a function of the distance between output units, the intrinsic connections are symmetric and translation invariant. Hence, the effect of updating in the second phase is qualitatively described by an attractor dynamic (24), with all attractors having the same shape, but being located in different positions in the output map.

To ensure that the attractor in the output layer is located in the region where the receptive fields are most active, we updated the output units in order of their activation as it was acquired in the first phase, starting from the least activated unit. This corresponds to the assumption that the more active units reach their discharge threshold earlier than less active units. The attractor then looks like a blob of activation around the winner. This more realistic modification marks the originality of our network. Without it, the network becomes dysfunctional after moderate degrees of pruning (the activation pattern in the output layer is governed by an attractor dynamic, but the attractor is not situated in correspondence of the winner). As the simulations will show, this modification also determines the distribution of the code vectors at intermediate stages of training. Unlike what is observed in ordinary topographic self-organizing maps (18), in these phases the code vectors are located at the center of large input clusters instead of being scattered around in the input space following the topology of the map (16).

Because of their configuration around the winner, the activation values in the output map after the second phase define a set of neighboring candidate code vectors, which are used to implement a soft competition scheme for the design of a quantization algorithm (25). Hence, training of the first group of weights is accomplished by a Hebbian competitive learning rule, in which the amount of adaptation is weighted by the activation in the output layer:

$$\Delta \mathbf{w}_j = h \; y_j (\mathbf{x} - \mathbf{w}_j),$$

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 287-294

where the learning rate h was kept at 0.01 during the whole training process. This rule attempts to minimize the Euclidean distance between the input pattern and the closest code vector, i.e. the average distortion between the input pattern and its prototype. During training, the second and third groups of weights do not learn. Instead, the excitatory weights of the second group are gradually eliminated starting from those with smaller magnitude:

$$w_{t+1}^{\text{exc}} = \begin{cases} w_t^{\text{exc}}, & \text{if } w > k \\ 0, & \text{if } r < \dfrac{1}{E \cdot N} \text{ or } w > k \end{cases},$$

where k is a pruning factor that was gradually increased from 0.4 to 1.8, E the number of steps in the train epoch, r a random value between 0 and 1, and N is the size of the output layer. The weights of the third group are left unaltered.

3. Results

In all simulations we will pursue the somewhat canonical learning goal of finding prototypes for clusters of points located on a hypersphere of unitary radius (26). Each cluster, representing a category, is constituted by a Gaussian distribution around a centroid. Points are generated for each cluster with equal probabilities.

The purpose of the first simulation is to visualize the behavior of the pruning algorithm. For this reason, the input clusters are bi-dimensional and regularly grouped. Figure 1 shows the location of thirty code vectors (black spots) relative to the input patterns (small gray points) at progressive stages of the learning process. In the figure, the code vectors appear to increase in number during learning. In reality, there are always the same number of code vectors, clustered together so closely to become

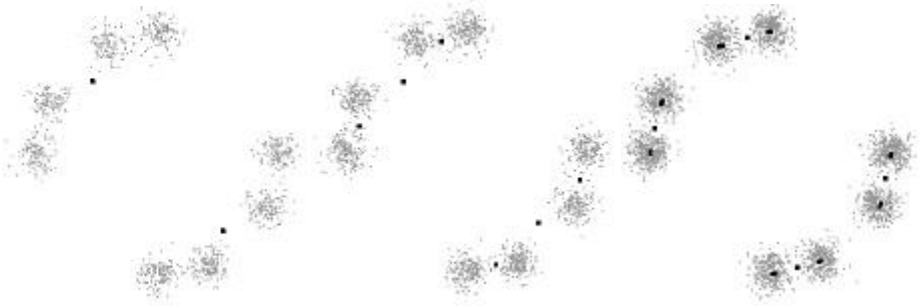

Figure 1. Location of 30 code vectors at intermediate stages of training (from left to right). In their progressive adaptation to the input distribution, the code vectors maintain topographic order. The average MTP index (a measure of topological preservation ranging from –1 to 1, with 0 indicating no preservation, devised by Bezdek and Pal, 27) of 15 networks thus trained was .63 (std. dev. .08), compared to .58 (std. dev. .06) obtained with topographic SOM.

undistinguishable. The number and location of the clusters of code vectors increase progressively adapting to the details of the input. Thus, the initial two clusters encode the ‘global’ two categories corresponding to the broadest two groupings of the input. Progressively smaller groupings are represented, corresponding to learning categories from the ‘global’ to the ‘basic’ level (10).

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 287-294

Intuitively, we may regard an object as simple if a short description of it can be formulated (28). We will approximate a measure of complexity by measuring the entropy of the set of codes (29). An important requirement on our measurement of complexity is also that it mirror the increase in categories that occurs in human cognitive development. For memory processes of semantic nature, a natural approximate measure of complexity is constituted by the number of distinct representations that are in storage. Hence, we will also measure the number of representations that are formed at different levels of pruning. Because the entropy has an upper bound that is a function of the number of codes, we expect these two measures to be related; the simulations will show that this is effectively the case.

To demonstrate the increase of complexity of the category set we will challenge the network with more complicated multidimensional patterns, constituted by a 14-dimensional Gaussian mixture of 30 centroids as input. The centroids were initialised randomly according to a uniform, isotropic distribution. The network had 30 output units arranged in a one-dimensional lattice. We estimated the entropy and number of different used prototypes by averaging over 400 networks trained on the same input distributions, but at increasing pruning levels. The result of these experiments is displayed in Figure 2, which demonstrates that the complexity of the network raises almost steadily with increasing levels of pruning.

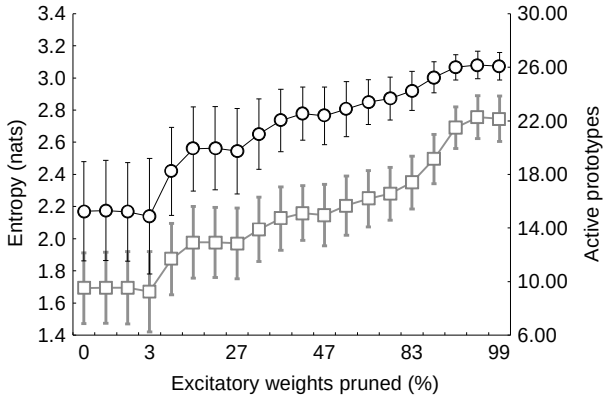

Figure 2. Entropy (black, circles) and used prototypes (gray, squares) at increasing degrees of pruning. For 30 prototypes in total, 3.40 nats is the maximal entropy that can be achieved. The correlation coefficient between pruning and entropy is r = 0.79; between pruning and the number of active prototypes, r = 0.84.

For each experiment we created 20 independent training data sets D_k (k = 1, 2, ... 20) of 120 records each from the same Gaussian mixture distribution. We then trained 20 networks on each data set independently, and estimated the entropy H of each. To obtain an estimate of H over the distribution of the codes, we averaged it over 5,000 new realizations of the input:

$$H = E\left[\ln\frac{1}{p(j)}\right] \approx \frac{1}{5000}\sum_{i=1}^{5000} p(j_i)\ln\frac{1}{p(j_i)}.$$

Thus, each experiment produces an estimate of entropy given an input distribution and a pruning level. To control for effects due to the random initialisation of the Gaussian

mixture, the experiment was repeated 20 times with new realization of the uniform centroid distribution. This procedure was repeated in 20 different conditions characterised by increasing degrees of final pruning. The number of used code vectors was obtained similarly, in each experiment averaging over the total number of different codes that were assigned when the centroids were used as input in turn.

4. Discussion

Our results are remarkable for three reasons. Firstly, satisfaction of either of these constraints (increasing complexity and decreasing number of synapses) is common among existing architectures, but not of both. It is not difficult to find in the existing literature on self-organization examples of models where complexity increases during training, as for example in ART2 (30) or in dynamic versions of Kohonen's self organizing map (31). However, in these architectures the number of synapses increases during training. By converse, well-known clustering techniques, such as principal component analysis, derive a large number of clusters, from which a subset can be selected. However, in such cases the complexity is initially high, and subsequently decreases when a smaller subset of clusters is selected.

Secondly, unlike selective elimination, complexity increase constitutes a learning strategy with demonstrable inferential properties (32). In the framework of statistical learning theory, the input-output mapping learned from a finite number of examples cannot be confidently used to generalize to unobserved cases if the complexity of the network is not limited in relation to the size of the training set (33). Hence, in on-line learning settings it is important to adjust network complexity so as to allow gradually more complex representations during the progression of training.

Thirdly, the fact that this inferential strategy may be implemented though an anatomical structure represents a novel interpretation of topographic organization of certain cortical areas. Existing interpretations point out the *biological* advantage of keeping often activated connections short (34), but fall short of individuating any computational advantage of topographicity.

Current empirical evidence is consistent with the model presented here. In the macaque monkey, pruning disproportionately affects synapses in layer II and III of the isocortex (1, 6). There is increasing evidence that layer II and III, where interneurons with horizontal connections are hosted, exercise a directive influence over the topographic organization of the cortex, as their reorganization after deafferentiation precedes that of the granular layers (35). Furthermore, as a result of the pruning process the ratio between synapses of the symmetric type (inhibitory) to the asymmetric type (excitatory) is inverted in layer II and III, with inhibitory synapses taking the lead in adulthood (6). In contrast with the synaptic selection theory, the development of synapses in layers II and III does not depend on the presence of external stimuli, as neither stimulation nor sensory deprivation prevent the formation of topographic organization (36-38).

References

1. Rakic P, Bourgeois JP, Eckenoff MF, Zecevic N, Goldman-Rakic PS (1986). Concurrent overproduction of synapses in diverse regions of the primate cerebral cortex. Science 232:232-235

2. Rakic P, Bourgeois J-P, Goldman-Rakic PS (1994). Synaptic development of the cerebral cortex: Implications for learning, memory, and mental illness. Prog. Brain. Res. 102:227-243
3. Huttenlocher PR (1979). Synpatic density in human frontal cortex: Developmental changes and effects of aging. Brain Res. 163:159-189
4. Huttenlocher PR, Dabholkar AS (1997). Regional differences in synaptogenesis in human cerebral cortex. J. Comp. Neurol. 387:167-178
5. Zecevic N, Rakic P (1991). Synaptogenesis in monkey somatosensory cortex. Cereb. Cortex 1:510-523
6. Bourgeois JP, Rakic P (1993). Changes of synaptic density in the primary visual cortex of the macaque monkey from fetal to adult stage. J. Neurosci. 13:2801-2820
7. Bourgeois JP, Goldman-Rakic PS, Rakic P (1994). Synaptogenesis in the prefrontal cortex of rhesus monkey. Cereb. Cortex 4:78-96
8. Changeux JP, Danchin A (1976). Selective stabilization of developing synapses as a mechanism for the specification of neural network. Nature 264:705-712
9. Keil FC (1983). On the emergence of semantic and conceptual distinctions. J. Exp. Psychol., Gen. 112:357-385
10. Mandler JM, Bauer PJ, McDonough L (1991). Separating the sheep from the goats: Differentiating global categories. Cognitive Psychol. 23:263-298
11. Quinn PC, Johnson MH (1997). The emergence of perceptual category representations in young infants: A connectionist analysis. J. Exp. Child Psychol. 66:236-263
12. Mareschal D, Quinn PC (2001). Categorization in infancy. Trends Cogn. Sci. 5:443-450
13. Hornik K, Stinchcombe M, White H (1989). Multilayer feedforward networks are universal approximators. Neural Netw. 2:359-366
14. Cover TM (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Trans. Electron. Comput. EC-14:326-334
15. Gallant SI (1990). A connectionist learning algorithm with provable generalization and scaling bounds. Neural Netw. 3:191-201
16. Viviani R (2002). Lateral interactions in self-organizing maps In: Dorronsoro JR (ed). Artificial Neural Networks - ICANN 2002 (Lecture Notes in Computer Science n. 2415), pp. 920-926
17. Willshaw DJ, von der Malsburg C (1976). How patterned neural connections can be set up by self-organization. Proc. R. Soc. Lond. B 194:431-445
18. Kohonen T (1982). Self-organized formation of topologically correct feature maps. Biol. Cybern. 43:59-69
19. Gersho A, Gray RM (1991). Vector Quantization and Signal Compression. Dordrecht: Kluwer
20. Nosofsky RM, Kruschke JK, McKinley SC (1992). Combining exemplar-based category representations and connectionist learning rules. J. Exp. Psychol., Learning Mem. Cogn. 18:211-233

21. Obermayer K, Ritter H, Schulten K (1990). A principle for the formation of the spatial structure of cortical feature maps. Proc. Natl. Acad. Sci. USA 87:8345-8349
22. Spitzer M, Böhler P, Weisbrod M, Kischka U (1995). A neural network model of phantom limbs. Biol. Cybern. 72:197-206
23. Harris JA, Harris IM, Diamond ME (2001). The topography of tactile learning in humans. J. Neurosci. 21:1058-1061
24. Hopfield JJ (1982). Neural networks and physical systems with emergent collective computational abilites. Proc. Natl. Acad. Sci. USA 79:2554-2558
25. Yair E, Zeger K, Gersho A (1992). Competitive learning and soft competition for vector quantizer design. IEEE Trans. Sign. Process. 40:294-308
26. Rumelhart DE, Zipser D (1985). Feature discovery by competitive learning. Cogn. Sci. 9:75-112
27. Bezdek JC, Pal NR (1995). An index of topological preservation for feature extraction. Pattern Recognition 28:381-391
28. Li M, Vitányi PMB (1992). Inductive reasoning and Kolmogorov complexity. J. Computer Syst. Sci. 44:343-384
29. Cover TM, Thomas JA (1991). Elements of Information Theory. New York: John Wiley & Sons
30. Carpenter GA, Grossberg S, Rosen DB (1991). ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition. Neural Netw. 4:493-504
31. Fritzke B (1994). Growing cell structures. A self-organizing network for unsupervised and supervised learning. Neural Netw. 7:1441-1460
32. Devroye L, Györfi L, Lugosi G (1996). A Probabilistic Theory of Pattern Recognition. Berlin: Springer
33. Vapnik VN (1995). The Nature of Statistical Learning Theory. Berlin: Springer
34. Durbin R, Mitchison G (1990). A dimension reduction framework for understanding cortical maps. Nature 343:644-647
35. Trachtenberg JT, Stryker MP (2001). Rapid anatomical plasticity of horizontal connections in the developing visual cortex. J. Neurosci. 21:3476-3482
36. Bourgeois JP, Jastreboff PJ, Rakic P (1989). Synaptogenesis in visual cortex of normal and preterm monkeys: Evidence for intrinsic regulation of synaptic overproduction. Proc. Natl. Acad. Sci. USA 86:4297-4301
37. Bourgeois JP, Rakic P (1996). Synaptogenesis in the occipital cortex of macaque monkey devoid of retinal input from early embryonic stages. Eur. J. Neurosci. 8:942-950
38. Murphy KM, Duffy KR, Jones DG, Mitchell DE (2001). Development of cytochrome oxidase blobs in visual cortex of normal and visually deprived cats. Cereb. Cortex 11:122-135