Submitted to: Proceedings of the International Joint Conference on Neural Networks 2000

Structured models from structured data: emergence of modular information processing within one sheet of neurons

Cornelius Weber and Klaus Obermayer

Dept. of Computer Science, FR2-1, Technische Universität Berlin
Franklinstr. 28/29, D-10587 Berlin, Germany. Email: cweber@cs.tu-berlin.de

Abstract

In our contribution we investigate how structured information processing within a neural net can emerge as a result of unsupervised learning from data. Our model consists of input neurons and hidden neurons which are recurrently connected and which represent the thalamus and the cortex, respectively. On the basis of a maximum likelihood framework the task is to generate given input data using the code of the hidden units. Hidden neurons are fully connected allowing for different roles to play within the unfolding time-dynamics of this data generation process. One parameter which is related to the sparsity of neuronal activation varies across the hidden neurons. As a result of training the net captures the structure of the data generation process. Trained on data which are generated by different mechanisms acting in parallel, the more active neurons will code for the more frequent input features. Trained on hierarchically generated data, the more active neurons will code on the higher level where each feature integrates several lower level features. The results imply that the division of the cortex into laterally and hierarchically organized areas can evolve to a certain degree as an adaptation to the environment.

Introduction

It is often neglected in relation to hierarchical information processing in the brain that each half of the cortex is a two-dimensional sheet of tissue. It hosts as many as 73 areas in the macaque [16]. They can be distinguised by architectonics, connectivity to other areas and topographic organization as measurable e.g. at half of the areas of the visual system with respect to visual field [3]. Recent findings suggest that a "neurochemical fingerprint" [5] determines the earliest compartmentalization in corticogenesis [2].

Information between these areas is known from functional [17] and connectivity analysis [3] to be processed in parallelly as well as hierarchically organized pathways. Due to the large number of areas and an even 10-fold larger number of connections between them [15] genetic information can hardly specify the connectivity pattern fully.

A model which explains corticocortical connections purely on the basis of topological neighborhood covers roughly half of the global existing connectivity pattern [15]. Genetic information as well as metabolic costs and wiring volume are hereby minimized. However, the model makes systematic errors at the borders of functionally dissimilar groups of cortical areas (e.g. between visual and auditory areas) and ignores hierarchical relationships.

There is evidence for activity driven self organization of cortical areas as well as of cortico-cortical connections. *(i)* Lesion studies [8] demonstrate that as an adaptation to a changed input cortical neurons can belong to different areas. *(ii)* Even in normal animals of the same species, there is a 2-fold variability of single areas from one brain to the next [3]. *(iii)* In *in vitro* cocultures thalamic axons grow to any region of the cortex, indistinguishably from their behavior at the proper target [12]. *(iv)* Axons are known to sprout exuberantly across non-proper target areas and refine later to the proper target areas [9].

Our aim is not to explain the emergence of all cortical areas neglecting intrinsic mechanisms. But we want to show first, that by an activity driven process with only faint influence by intrinsic mechanisms two areas can emerge in parallel such that each of them processes different parts of the input. Secondly, that areas at different hierarchical levels can emerge, again without geometrical or strong intrinsic constraints.

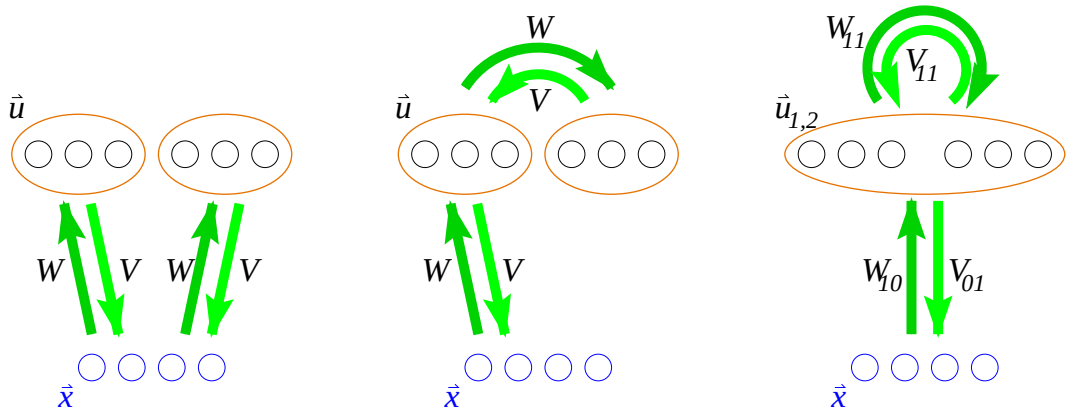

Figure 1: Three different model architectures. In each of them the activations $\vec{x}$ on the input units are represented by hidden unit activations $\vec{u}$. W are recognition weights, V are generative weights, indexed in the right figure with the number of the layer of termination and origin. **Left:** parallel organization and **middle:** hierarchical organization of hidden units which are segregated into two areas. **Right:** architecture of our model. The lateral weights W_{11} and V_{11} (top) allow each hidden neuron to take part in a representation $\hat{u}_1$ on a lower and $\hat{u}_2$ on a higher hierarchical level. Dependent on the structure of the data training will result in one of the two other architectures shown.

Finally, that both adaptations can occur within the same network and with unchanged parameters, as an adaptation to differently generated data only.

Previous related computational or theoretic work is scarce. Dailey and Cottrell [1] investigate the division of labour between two experts in a kind of mixtures of experts model [10]. In their model, one expert receives low-frequency, the other high-frequency information of the same input data which was taken from objects and faces. As a result of supervised training on identifying the faces and classifying the objects, the module which receives low-frequency information specializes for face recognition. An explanation, however, how each module can filter its input in such a way, remains external to the model.

Our idea is that neurons possess different intrinsic functional properties in different regions of the cortex. Upon learning the neurons should code for those elements of the data which its function is best adapted to. In our net the hidden neurons are distinguished into two classes, one with highly active neurons, the other with sparsely active neurons. This property can be scaled by a real valued parameter of the neuronal transfer function. Hereby neurons are expected to specialize on data elements which are less or more sparsely distributed.

Theory and Methods

We use a recurrent model which learns an internal representation of the data by a maximum-likelihood framework. Using one hidden layer [4][6][13], the role of the feedback projection is to reconstruct the data from hidden unit activations. The forward connections are used to transmit the reconstruction error to the hidden neurons which they try to minimize by correcting their activation towards an optimal representation.

Reconstruction of the data with as many hidden units as input units is trivial and demands for some kind of bottleneck. A sparse activation prior on hidden neurons makes them represent sparsely occuring input features, e.g. if natural images are used for training then localized edge detectors emerge [4][13].

As an effect of sparse hidden neuron activations the data are under-estimated during reconstruction. The weights will compensate by learning larger values in order to increase the activations. To preserve sparse coding a weight constraint must be introduced.

The hidden representation can span more than just one hierarchical level [14][11]. Then a given neuron not only adjusts its activation to minimize the reconstruction error on the lower level but also to match the feedback ("prediction") from a higher level representation.

Concatenation of levels In a model for the evolution of a two-level hierarchy all hidden neurons have to choose whether to take the computational role of the first hidden layer, the second hidden layer, or both. Here we concatenate the first and the second hidden layer which renders weights between them lateral (Fig. 1). Different activations on these units, however, can belong logically either to the first hidden layer ($\hat{u}_1$) or to the second hidden layer ($\hat{u}_2$). An activity-dependent weight constraint introduces competition between all incoming weights of a hidden neuron (besides preserving sparse coding). This encourages a hidden neuron to receive input from the input neurons via W_{10} or from other lateral neurons via W_{11} only.

Relaxation of activations For each data point $\vec{x}$ which is presented on the input layer the algorithm iteratively relaxates hidden unit activations after starting with zero values.

The following computations are done in the input, the logically first hidden layer and the logically second hidden layer. First, the negative reconstruction error $\tilde{x}_0$ in the input neurons is the difference between the data $\vec{x}$ and the reconstruction from the hidden code $\hat{u}_1$ via feedback weights V_{01}:

$$\tilde{x}_0(t) \;=\; \vec{x} \;-\; V_{01}\hat{u}_1(t) \tag{1}$$

The negative reconstruction error measured in the logically first layer is the difference between the bottom-up input and the top-down reconstruction:

$$\tilde{x}_1(t) \;=\; W_{10}\tilde{x}_0(t) \;-\; V_{11}\vec{u}_2(t) \tag{2}$$

The hidden code vector $\hat{u}_1$ is the hidden units' representation of the data on the first hierarchical level. It is adjusted *(i)* to account for the negative error $\tilde{x}_0$ via recognition weights W_{10} and *(ii)* to account for the prediction from the next higher level activations $\hat{u}_2$ via generative weights V_{11}:

$$\begin{aligned} \vec{h}_1(t) &= \hat{u}_1(t) \;+\; \varepsilon_u\,(\beta\, W_{10}\tilde{x}_0(t) \;-\; (1-\beta)\,\tilde{x}_1(t)) \\ &= \hat{u}_1(t) \;+\; \varepsilon_u\,((2\beta-1)\, W_{10}\tilde{x}_0(t) \;+\; (1-\beta)\, V_{11}\,\vec{u}_2(t) \\ \hat{u}_1(t+1) &= \vec{f}(\vec{h}_1(t)) \end{aligned} \tag{3}$$

where ε_u is the update step size, β handles the tradeoff between bottom-up and top-down information and f is the the transfer function which transfers the inner activations $\vec{h}_1$ to the hidden code at time $t+1$.

The hidden code $\hat{u}_2$ is the hidden units' representation on the logically second hierarchical level. It adjusts to the code $\hat{u}_1$ on the first level via the lateral recognition weights W_{11} but has no feedback from a higher hierarchical level.

$$\begin{aligned} \vec{h}_2(t) &= \hat{u}_2(t) \;+\; \varepsilon_u\, W_{11}\tilde{x}_1(t) \\ \hat{u}_2(t+1) &= \vec{f}(\vec{h}_2(t)) \end{aligned} \tag{4}$$

The transfer function enforces sparse coding by reducing small activations in particular:

$$f_i(h_i) \;=\; h_i \;-\; \lambda^u \cdot \frac{2\,h_i}{1+h_i^2} \tag{5}$$

where λ^u scales the sparseness constraint. After relaxation of Eqs. (1,2,3,4) towards a stationary state we have found the optimal code $\hat{u}_{1,2}$ to reconstruct the data point, under a sparseness prior on the hidden unit activations from which the sparseness constraint can be derived [13].

We train recognition weights w_{10}^{ij} from input neuron j to hidden neuron i and lateral recognition weights w_{11}^{ik} from hidden neuron k to hidden neuron i by the following update rules:

$$\begin{aligned} \Delta w_{10}^{ij} &= \varepsilon_w\left(u_1^i\,\tilde{x}_0^j \;-\; \lambda^w\,|\bar{h}^i|\,w_{10}^{ij}\,||\vec{w}^i||^2\right) \\ \Delta w_{11}^{ik} &= \varepsilon_w\left(u_2^i\,\tilde{x}_1^k \;-\; \lambda^w\,|\bar{h}^i|\,w_{11}^{ik}\,||\vec{w}^i||^2\right) \end{aligned} \tag{6}$$

where ε_w is the learn step size. The first term on the right hand side implements Hebbian learning. The second term which is scaled by λ^w is a soft activity dependent weight constraint. $||\vec{w}^i||^2 = \sum_l^N (w_{10}^{il})^2 + \sum_l^H (w_{11}^{il})^2$ is the sum of the squared weights to all N input units and all H hidden units and $|\bar{h}^i| = |h_1^i| + |h_2^i|$ is the mean of absolute values of the inner activations of hidden neuron i at the final relaxation time step. The weight constraint scales the length but does not change the direction of a hidden neuron weight vector. It is local in the sense that it does not depend on any weight of any other hidden neuron. Generative weights are made symmetric to the recognition weights, i.e. $V_{01} = W_{10}^T$ and $V_{11} = W_{11}^T$.

Results

Generation of the data Artificial data are generated by two different paradigms. First, in a non-hierarchicel manner, data consist of discrete, sparsely generated elements. These elements are lines of 4 different orientations on a 5×5 grid of input units resulting in a total number of 20 different elements. For the first experiment each line is chosen with a fixed probability independently of any other. Thus there is no structure among the code elements. For the purpose of structuring our network into two distinct groups of hidden neurons we form two groups of the elements. One group, horizontal and 45° lines are generated with probability 0.1 whereas the other group, vertical and 135° lines are generated half as often, with probability 0.05 each. Fig. 2 a), left, shows example data.

For the second experiment data are generated in a hierarchical manner [7]. First, one of 4 orientations are choosen, which represents a decision process within a higher hierarchical level. Then, on the lower level, lines from the formerly chosen orientation only are generated with probability 0.3 each. Fig. 2 b), left, shows example data.

Training Weights were initialized with small random values with mean zero. Then the following on-line learning procedure was repeated $5 \cdot 10^6$ times. A data point was shown to the input neurons and Eqs. (1,2,3,4) were iterated 10 times after initialization of hidden unit activations with zero to obtain the best estimate for the hidden code. Using these values the weights were trained according to Eqs. (6).

The parameters were: stepsize for the update of activations $\varepsilon_u = 0.1$, tradeoff for bottom-up/top-down input $\beta = 0.9$, learning stepsizes $\varepsilon_{w\,01} = 0.03$, $\varepsilon_{w\,11} = 0.003$, constraint on the weights $d_w = 0.03$, sparsity parameter $\lambda^u = 0.1$ for one half of the hidden neurons, $\lambda^u = 0.2$ for the others.

Areas organize in parallel The net which had been trained on the parallel data extracted all lines from the data. Fig. 2 a) shows some example data and the weights after training. Each code element (one of the 5×4 possible lines) is represented by a weight vector of the matrix W_{10}. Due to overcomplete coding a small number of neurons are not connected to the input and some lines are represented by two hidden neurons. The bottom half of the hidden neurons which have stronger activations ($\lambda^u = 0.1$) specialize on the input features which occur more often (lines of 0° and 45°). The upper neurons which have weaker activations ($\lambda^u = 0.2$) specialize on the rare features (lines of 90° and 135°). On both layers, there are a small number of exceptions from the rule.

If prior knowledge that the data has no hierarchical structure was assumed then $\hat{u}_2$ would not have to be computed by Eq. (4). For consistency with the next experiment, however, we included the second hierarchical level and $\hat{u}_1$ takes into account $\hat{u}_2$. Hereby lateral weights W_{11} emerge through which in general the activation $\hat{u}_2$ of one neuron supports the activation $\hat{u}_1$ of an arbitrary other neuron. The core result of this experiment as decribed above is unchanged if a second hierarchical level is omitted (results not shown).

Areas organize hierarchically The net shown in Fig. 2 b) was trained on the hierarchical data set. It has structure among the weights W_{10} to the input in the area with less active neurons, and has structure among the lateral recognition weights W_{11} in the area with stronger activation. The former neurons have not discovered single lines as input elements because a larger number of them were presented which have the same direction compared to the first experiment (the average number of lines in each stimulus is 1.5 in both settings). Thus, elementary features could as well be the dark lines in between. Another argument for a different representation is that the less active neurons form an undercomplete representation of the input.

Neurons in the more active area join together those neurons in the less active region which code for the same orientation by the lateral weights W_{11}. As the more active neurons they code for the orientation of a stimulus because one of four orientations is statistically chosen more often than a single line.

We did not adjust the sizes of the areas to the expected outcome which we could have done by setting the parameters of exactly four neurons to have strong activations. As a consequence of a too large number of highly active neurons more neurons redundantly represent the second hierarchical level.

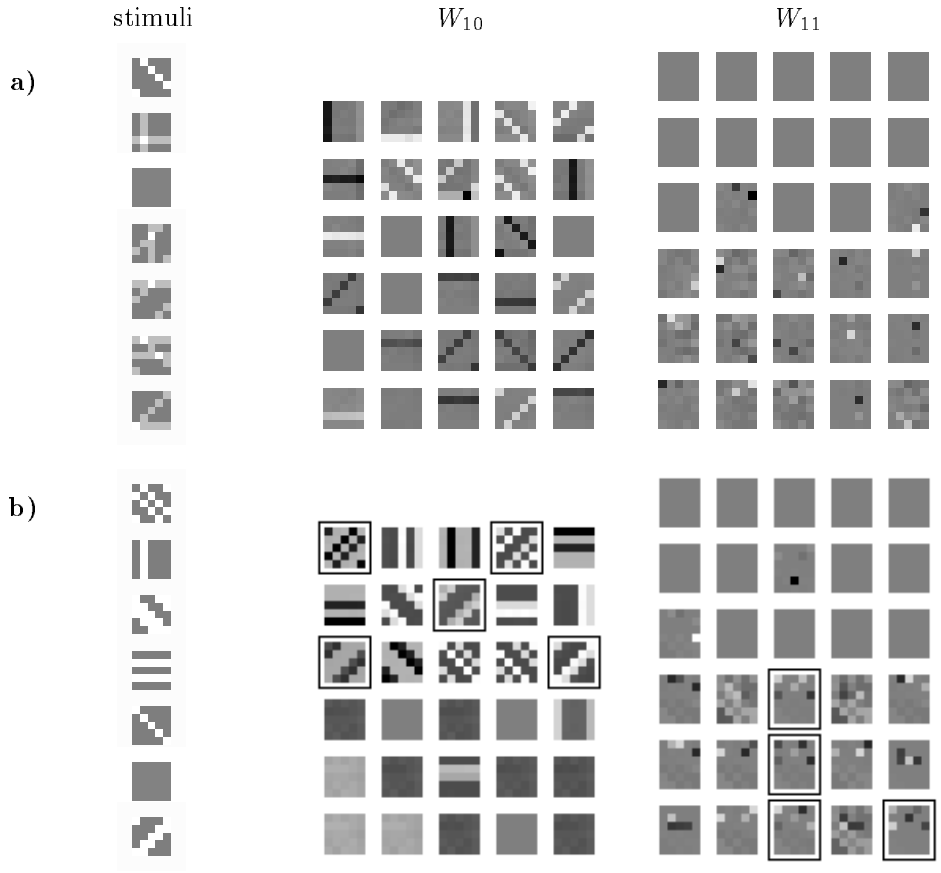

Figure 2: **Left:** examples of stimuli $\vec{x}$ used for training. **Middle:** the recognition weight matrices W_{10} and **right:** the lateral recognition weight matrices W_{11} after training. Each square of the weight matrices shows the receptive field of one of the 5×6 hidden neurons, black indicating negative, white positive weights. **Middle**, weights to one of the 5×5 input neurons and **right**, lateral weights. A larger number of hidden neurons than input neurons allow for an overcomplete representation.
a) Parallel organization of areas: weights W_{10} to the inputs concentrate on 0° and 45° lines in the lower half and on 90° and 135° lines in the upper half. **b)** Hierarchical organization of areas: neurons in the upper half code for the input via W_{10} while neurons in the lower half organize the code from the upper units via W_{11}. Neurons which code for lines of 45° orientation are marked by a frame.

Discussion

We have shown that a net trained according to a maximum likelihood framework can organize in *a)* two parallelly or *b)* two hierarchically organized areas dependent on statistics of the data. Parameters are unchanged between these experiments. The hidden area with more frequently active neurons *a)* looks at input features which occur more often or *b)* takes the function of the hierarchically higher area in our setting.

We take this as a model for the development of connection patterns and relations between cortical areas like the parallel segregation of the visual stream into a lateral and a dorsal pathway or the hierarchical relation between V1 and V2. Different functional properties, e.g. in a spike coding model the refractory period or burst duration, could be varied across areas, to account for a broader parallel organization and for more hierarchical stages.

For simplicity we have neglected many additional learning mechanisms of the biological system. Most important are topographic constraints which favor neighboring cortical areas to be connected [15]. These determine however the gross connectivity pattern between cortical areas only. Within an area a topographic

constraint can force a minority of neurons to code conform with the majority. Such a constraint could eleminate the outliers in our results. Time behavior like a possible flow of a stimulus induced signal from the back of the cortex towards the front could alternatively or as an additional mechanism give rise to hierarchical relationships.

By our approach which omits as much as possible internal mechanisms we could display a large potential which the data has in the organization of structure. On a high level of abstraction it demonstrates how much influence the structure of the data may have on the emerging structure of the brain.

References

[1] M.N. Dailey and G.W. Cottrell. Organization of face and object recognition in modular neural networks. *Neural Networks*, 12(7-8):1053–1073, 1999.

[2] M. J. Donoghue and P. Rakic. Molecular evidence for the early specification of presumptive functional domains in the embryonic primate cerebral cortex. *J. Neurosci.*, 19(14):5967–79, 1999.

[3] D.J. Felleman and D.C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1:1–47, 1991.

[4] P. Földiák. Forming sparse representations by local anti-hebbian learning. *Biol. Cybern.*, 64:165–170, 1990.

[5] S. Geyer, M. Matelli, G. Luppino, A. Schleicher, Y. Jansen, N. Palomero-Gallagher, and K. Zilles. Receptor autoradiographic mapping of the mesial motor and premotor cortex of the macaque monkey. *J. Comp. Neurol.*, 397:231–250, 1998.

[6] G. Harpur. Development of low entropy coding in a recurrent network. *Network – Computation in Neural Systems*, 7(2):277–284, 1995.

[7] G. E. Hinton, P. Dayan, B. J. Frey, and R. Neal. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.

[8] K.J. Huffman, Z. Molnár, A. Van Dellen, D.M. Kahn, C. Blakemore, and L. Krubitzer. Formation of cortical fields on a reduced cortical sheet. *J. Neurosci.*, 19(22):9939–9952, 1999.

[9] G.M. Innocenti. Exuberant development of connections, and its possible permissive role in cortical evolution. *Trends in Neurosci.*, 18(9):397–402, 1995.

[10] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neur. Comp.*, 3:79–87, 1991.

[11] M. Kawato, H. Hayakawa, and T. Inui. A forward-inverse optics model of reciprocal connections between visual cortical areas. *Network*, 4:415–422, 1993.

[12] Z. Molnar and C. Blakemore. Development of signals influencing the growth and termination of thalamocortical axons in organotypic culture. *Experimental Neurology*, 156(2):363–393, 1999.

[13] B.A. Olshausen and D.J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

[14] R.P.N. Rao and D.H. Ballard. Dynamic model of visual recognition predicts neural response properties of the visual cortex. *Neur. Comp.*, 9(4):721–763, 1997.

[15] J.W. Scannell, C. Blakemore, and M.P. Young. Analysis of connectivity in the cat cerebral cortex. *J. Neurosci.*, 15(2):1463–1483, 1995.

[16] M.P. Young. The organization of neural systems in the primate cerebral cortex. *Proc. R. Soc. Lond. B*, 252:13–18, 1993.

[17] S. Zeki and S. Shipp. The functional logic of cortical connections. *Nature*, 335:311–317, 1988.