Emergence of Modularity within One Sheet of Neurons: A Model Comparison

Cornelius Weber and Klaus Obermayer

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

Abstract. We investigate how structured information processing within a neural net can emerge as a result of unsupervised learning from data. The model consists of input neurons and hidden neurons which are recurrently connected. On the basis of a maximum likelihood framework the task is to reconstruct given input data using the code of the hidden units. Hidden neurons are fully connected and they may code on different hierarchical levels. The hidden neurons are separated into two groups by their intrinsic parameters which control their firing properties. These differential properties encourage the two groups to code on two different hierarchical levels. We train the net using data which are either generated by two linear models acting in parallel or by a hierarchical process. As a result of training the net captures the structure of the data generation process. Simulations were performed with two different neural network models, both trained to be maximum likelihood predictors of the training data. A (non-linear) hierarchical Kalman filter model and a Helmholtz machine. Here we compare both models to the neural circuitry in the cortex. 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.

1 Introduction

The cortex is the largest organ of the human brain. However, a mammal can survive without a cortex and lower animals do not even have a cortex. Essential functions like controlling inner organs and basic instincts reside in other parts of the brain. So what does the cortex do? An other way to put this question is: what can the lower animals **not** do? If lower animals cannot learn a complex behavior we may infer that they cannot understand a complex environment. In other words: the cortex may provide a representation of a complex environment to mammals.

This suggests that the cortex is a highly organized structure. On a macroscopic scale, the cortex can be structured into dozens of areas, anatomically and physiologically. These are interconnected in a non-trivial manner, making neurons in the cortex receive signals primarily from other cortical neurons rather than directly from sensory inputs. In the absence of input the cortex can still generate dreams and imagery from intrinsic spontaneous activity. Recurrent connectivity between its areas may be the key to these capabilities.

S. Wermter et al. (Eds.): Emergent Neural Computational Architectures, LNAI 2036, pp. 53–67, 2001.

The cortex is, nevertheless, a sheet of neuronal tissue. Across its two dimensions, it hosts many functionally distinct areas (e.g. 65 areas in cat [12]) which process information in parallel as well as via hierarchically organized pathways [5]. The earliest manifestations of areas during corticogenesis are regionally restricted molecular patterns ("neurochemical fingerprints" [6]) which appear before the formation of thalamo-cortical connections [4].

Considering connectivity, there are up to ten times more area-to-area connections than areas. Thus, the description of cortico-cortical connectivity is more complex and requires modeling to be understood. Abstract geometrical models [12] suggest that topological neighborhood plays an important but not exclusive role in determining these connections.

Recently we presented computational models [14][13] in which the connections between areas are trained from neuronal activity. The maximum likelihood framework makes the network develop an internal representation of the environment, i.e. of the causes generating the training data.

Our recent models belong to two groups: one [14] we may call a non-linear Kalman filter model [11][9] in which neurons are deterministic and have a continuous transfer function. The other [13] is a Helmholtz machine [3] where hidden neurons are stochastic and binary. The dynamics of neuronal activations also differ: in the deterministic model, neurons integrate the inputs from several sources and over time. In the stochastic model, computations are separated and do not need to be integrated over time. The activation terms at different times are then summed up in the learning rules.

The model areas are determined a priori by the intrinsic functional properties of their neurons. More precisely, hidden neurons are divided into two groups which differed in their firing properties by corresponding parameter changes. Thus one group responds with stronger activity to a given input than the other group. In consequence the first group learns to process activity patterns which occur more frequently. The input space can thereby be divided into two groups. Using lateral weights among the hidden neurons, these two models allow hidden neurons to code using two hierarchical levels. When presented with hierarchically generated data, some of the hidden neurons establish a second hierarchical level by grouping together other neurons via their lateral weights while their weights to the input neurons decline.

In this contribution we want to inquire principles according to which the connectivity between cortical areas arises. We set up a model for the development of the connectivity between connectionist neurons and consider the macroscopic areas to be made up of a (small) group of microscopic neurons. A key idea is that such groups are distinguished by their neuronal properties prior to learning. We thereby do not address the possibility that intrinsic neuronal properties (other than those determined by their weights) can be dynamically changed during development.

Fig. 1. The cortico-cortical connection matrix of the cat. On both axes, the 65 cortical areas are arranged into the four complexes, visual system, auditory system, somatosensory system and fronto-limbic system. Connections along the horizontal axis are afferent, and those along the vertical axis are efferent. Strong connections are depicted dark, weaker connections are depicted in a lighter grey, missing connections white. Self connections (diagonal) are omitted. Data was taken from [12].

1.1 Review of Biological Data

Cortical areas can be hard to distinguish. For this reason, different criteria may be combined:

- Chemical properties: recent findings suggest that a "neurochemical fingerprint" [6] determines the earliest compartmentalization in corticogenesis [4].
- Architecture: staining can reveal a different structure. This method is reliable only for a few areas [5].
- Physiological properties. As an example, topographic organization is measurable w.r.t. the visual field in half of all visual cortical areas [5].
- The connectivity "fingerprint", i.e. the connectivity pattern to other cortical areas. So, if two areas had similar connectivity patterns, then they would be the same.

An estimate of the number of cortical areas is 65 in the cat [12] and 73 in the macaque [15]. The number of connections reported by Young [12] between these areas is 1139 in the cat (Fig. 1) which represents 27.4% of all possible connections (only ipsilateral connections are considered). The number of connections reported in the macaque [15] is 758 which represents 15% of all possible connections. Most of the connections are bidirectional: they consist of axons going into both directions. There are only 136 reported one-way connections which is 18%

of the total (macaque). Together there is a mean of approximately 10 input- and 10 output-connections per area.

The cortex can be divided into 4 complexes (see Fig. 1 for the cat data) with different functionality and sizes: visual cortex 55%, somato-sensory 11%, motor 8%, auditory 3% and 23% for the rest [5]. There are 25 areas plus 7 visual-association areas in the visual system of the macaque. The number of connections within this system is 305, i.e. 31% of all possible connections. The somato-sensory/motor system has 13 areas with 62 connections, i.e. 40% of all possible connections. Thus, connectivity between areas within one complex is higher than average.

The connection strengths between areas (i.e. density of fibers) comprise two orders of magnitude [5]. Only 30-50% of connections may be reliably found across different animals. Sizes of areas also vary: V1 and V2 take each 11-12% of the surface area (macaque [5]), the smallest areas are 50-times smaller. There is even a 2-fold variability in the size of single areas from one brain to another [5] within animals of the same species and age.

Finally, it should be noted that every visual area is connected to non-cortical areas. The number of these connections may out-range the number of cortico-cortical connections.

Table 1. The roles of intrinsic and activity dependent developmental mechanisms. Neural connections as well as areas are the result of an interplay between both intrinsic and activity dependent mechanisms.

	intrinsic	**activity dependent**
what is meant	genetic description	learning
how does it work	chemical markers	Hebbian learning
when does it appear	early	late
its targets	cell movement, cell differentiation, connections	connections
its results	layers, areas	receptive field properties, barrels, areas

2 Methods

Commonly used model architectures have a pre-defined structure because the order in which neuronal activations are computed depends on the architecture. An architecture as in Fig. 2 **b)** for example does not allow a two-stage hierarchy to develop. An architecture as in Fig. 2 **c)** does not allow the hierarchically topmost units (the three dark units) to develop connections to the input.

A more general structure is a full connectivity between all neurons, as in Fig. 2 **a)**. The only restriction we have chosen here is that input units are not

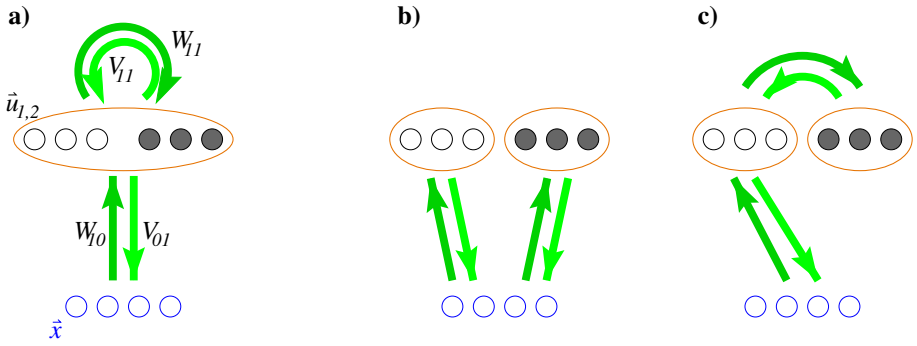

Fig. 2. Three different model architectures. In each of them the activations $\boldsymbol{x}$ on the input units are represented by hidden unit activations $\boldsymbol{u}$. W are recognition weights, V are generative weights, indexed in the left figure with the number of the layer of termination and origin. **a)** architecture of our model. The lateral recognition weights W_{11} and generative weights V_{11} (top) allow each hidden neuron to take part in a representation $\boldsymbol{u}_1$ on a lower and $\boldsymbol{u}_2$ on a higher hierarchical level. Dark units differ from the white hidden units by a parameter of their transfer function only. Depending on the structure of the data training will result in one of the two other architectures shown: **b)** a parallel organization and **c)** a hierarchical organization of the two areas.

connected to other input units. Learning rules for such architectures exist, like the well-known Boltzmann machine learning rule. The purpose of our study is to let structures as in Fig. 2 **b),c)** self-organize from this general, full connectivity.

2.1 Different Approaches to Maximum Likelihood

The goal of maximum likelihood is to find the model which can best explain, i.e. generate, the whole data set $\{\boldsymbol{x}^\mu\}$. If the data are independent, we have:

$$p(\{\boldsymbol{x}^\mu\}|V) \;=\; \prod_\mu p(\boldsymbol{x}^\mu|V) \;=\; \int \prod_\mu p(\boldsymbol{x}^\mu, \boldsymbol{u}|V)\, d\boldsymbol{u}$$

where $\boldsymbol{u}$ is the hidden unit activation vector. The data $\boldsymbol{x}^\mu$ and the hidden representation $\boldsymbol{u}$ are related to each other, so $\boldsymbol{u}$ is an internal representation of a data point. Through learning the model parameters V adjust to the whole data set, so the weights are a representation of the whole environment.

The integration across the hidden code $\boldsymbol{u}$ which is necessary to obtain the probability distribution of the model generating the data is computationally infeasible. Different approximations therefore must be done which lead to different models. Examples are shown in Fig. 3.

On the left branch of Fig. 3, the integral over the hidden code is replaced by the "optimal" hidden code vector $\boldsymbol{u}^{opt}$ which generates a certain data item with highest probability. This estimates a current system state $\boldsymbol{u}$ which relates to an observed process through $\boldsymbol{x} = V\boldsymbol{u} + e$ with noise e. With a linear transform

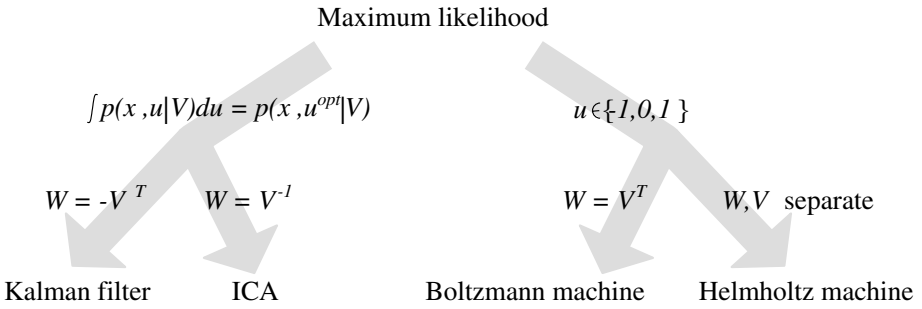

Fig. 3. Approximations used to arrive at different models.

V and in the case of Gaussian noise e we would arrive at a Kalman filter [11]. However, we will assume a non-Gaussian "sparse" prior in our model which we then term a non-linear Kalman filter. In the special case of an invertible weight matrix, one arrives at an ICA algorithm [2] (see [10] for comparison).

On the right branch, the hidden code vector is discretized such that the integral may be computed. Having weight symmetry and a strict gradient ascent algorithm for training the weights which maximizes the likelihood we arrive at the Boltzmann machine (standard literature, e.g. [7]). Separate treatment of recognition weights and generative weights leads to the Helmholtz machine [3]. A practical but heuristic algorithm to train this network is the wake-sleep algorithm [8].

For gradient based training of the weights, the performance of the network in generating the data is permanently measured. The way in which the data set is used for the generation of the data throughout learning distinguishes the two models which we explore in this contribution:

- **Kalman filter model:** when the model generates data, a "target" data point $\boldsymbol{x}^{\mu}$ is always selected which has to be reconstructed by the model.
- **Helmholtz machine:** without any data point selected, the model has to generate any of the given data points with the appropriate probability.

Thus, whether there is or there is no data point present influences the way in which the fast changing model parameters, the activations, are used:

- **Kalman filter model:** the goal is to find, given a data point, the posterior distribution of the fast changing model variables, i.e. the hidden unit activations. As an approximation, the optimal representation $\boldsymbol{u}^{opt}$ which maximizes the posterior probability to generate that data point is selected.
- **Helmholtz machine:** generation of the data is separated in time from the process of recognition: a "wake phase" characterized by presence of data is distinguished from a "sleep phase" during which the net generates its own "fantasy" data.

2.2 Architecture and Notation

Weights: The network architecture as well as some of the variables are depicted in Fig. 2, left. Input units (below) are linear and receive the data. Recognition weights W_{10} transfer information from the input units to all hidden units. Generative weights V_{01} transfer information from the hidden units to the input units. Lateral weights W_{11} and V_{11} transfer information from all hidden units to all hidden units. They are also distinguished into recognition weights, W_{11}, and generative weights, V_{11}, depending on whether they are used to transfer information upwards within the functional hierarchy or downwards towards the input units, respectively. In the Kalman filter model, $W_{10} = V_{01}^T$ and $W_{11} = V_{11}^T$.

Activations: The data cause an activation vector $\boldsymbol{x}$ on the input neurons. Hidden unit activation vectors have to be assigned to a virtual hierarchical level: in a hierarchical organization, the order of neuronal activation updates is a series where the number of update steps which are needed to propagate information from a data point to a neuron defines its hierarchical level. A hidden neuron is regarded to code on the first hierarchical level if it is activated by the input units. We denote this activation $\boldsymbol{u}_1$. A hidden neuron is regarded to code on the second hierarchical level if it is activated by its input from other hidden neurons via lateral weights only but not from the input units directly. We denote this activation $\boldsymbol{u}_2$. Note that both activations, $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, occur on all hidden units. Activations are distinguished by the way they arise on a neuron and may coexist in the Kalman filter model.

Parameters: Please see [14] and [13] for the values of all parameters in the non-linear Kalman filter model and the Helmholtz machine, respectively.

2.3 The Kalman-Filter Algorithm

The Kalman-filter algorithm can be separated into two steps: *(i)* calculation of the reconstruction errors $\tilde{x}_0(t)$ on the input units and $\tilde{x}_1(t)$ on the first hidden level and *(ii)* adjustment of activations in order to decrease these errors. Both steps are repeated until equilibrium is reached in which case the optimal hidden representation of the data has been found.

(i) compute reconstruction errors:

$$\begin{aligned} \tilde{x}_0(t) &= \boldsymbol{x} - V_{01}\boldsymbol{u}_1(t) \\ \tilde{x}_1(t) &= W_{10}\tilde{x}_0(t) - V_{11}\boldsymbol{u}_2(t) \end{aligned}$$

(ii) adjust hidden unit activations:

$$\begin{aligned} \boldsymbol{h}_1(t) &= \boldsymbol{u}_1(t) + \varepsilon_u((2\beta - 1)\, W_{10}\tilde{x}_0(t) \\ &\qquad + (1 - \beta)\, V_{11}\boldsymbol{u}_2(t)) \\ \boldsymbol{h}_2(t) &= \boldsymbol{u}_2(t) + \varepsilon_u\, W_{11}\tilde{x}_1(t) \end{aligned}$$

where transfer functions $\boldsymbol{u}_1(t+1) = \boldsymbol{f}(\boldsymbol{h}_1(t))$ and $\boldsymbol{u}_2(t+1) = \boldsymbol{f}(\boldsymbol{h}_2(t))$ are

used. ε_u denotes the activation update step size and β denotes the trade-off between bottom-up and top-down influence.

After repetition of these two steps until convergence the weights are updated (with step sizes ε^w) according to:

$$\Delta w_{10}^{ij} \;=\; \varepsilon_{10}^w u_1^i \, \tilde{x}_0^j \qquad\qquad \Delta w_{11}^{ik} \;=\; \varepsilon_{11}^w u_2^i \, \tilde{x}_1^k$$

2.4 The Wake-Sleep Algorithm (Helmholtz Machine)

The wake-sleep algorithm consists of two phases.

First, in the wake phase, data are presented. Then the net finds a hidden representation of the data and based on this representation, the net re-estimates the data.

Wake phase	infer hidden code:	$\boldsymbol{u}_1^w \;=\; \boldsymbol{f}_m^w(W_{10}\boldsymbol{x})$	$\boldsymbol{u}_2^w \;=\; \boldsymbol{f}_m^w(W_{11}\boldsymbol{u}_1^w)$
	reconstruct input:	$\boldsymbol{s}_1^w \;=\; V_{11}\boldsymbol{u}_2^w$	$\boldsymbol{s}_0^w \;=\; V_{01}\boldsymbol{u}_1^w$

After one-sweep computation of these equations the difference between the data and the re-estimation is used to train the generative weights (ε are the respective learning step sizes):

$$\Delta V_{11} \;=\; \varepsilon_{11}\,(\boldsymbol{u}_1^w - \boldsymbol{s}_1^w)\cdot(\boldsymbol{u}_2^w)^T \qquad\qquad \Delta V_{01} \;=\; \varepsilon_{01}\,(\boldsymbol{x} - \boldsymbol{s}_0^w)\cdot(\boldsymbol{u}_1^w)^T$$

Secondly, in the sleep phase, a random "fantasy" activation vector is produced in the highest level. The net then generates the corresponding "fantasy" data point and based on this, the net re-estimates the activation vector on the highest level.

	initiate hidden code at highest level:	$\boldsymbol{s}_2^s \;=\; \boldsymbol{f}_b^s(0)$	
Sleep phase	generate input code:	$\boldsymbol{s}_1^s \;=\; \boldsymbol{f}_b^s(V_{11}\boldsymbol{s}_2^s)$	$\boldsymbol{s}_0^s \;=\; V_{01}\boldsymbol{s}_1^s$
	reconstruct hidden code:	$\boldsymbol{u}_1^s \;=\; \boldsymbol{f}_m^s(W_{10}\boldsymbol{s}_0^s)$	$\boldsymbol{u}_2^s \;=\; \boldsymbol{f}_m^s(W_{11}\boldsymbol{s}_1^s)$

After obtaining these variables the difference between the original activation vector and the re-estimation is used to train the recognition weights:

$$\Delta W_{10} \;=\; \varepsilon_{10}\,(\boldsymbol{s}_1^s - \boldsymbol{u}_1^s)\cdot(\boldsymbol{s}_0^s)^T \qquad\qquad \Delta W_{11} \;=\; \varepsilon_{11}\,(\boldsymbol{s}_2^s - \boldsymbol{u}_2^s)\cdot(\boldsymbol{s}_1^s)^T$$

2.5 Weight Constraints

In order to discourage a hidden neuron to be active at all processing steps, competition between all incoming weights of one neuron is introduced by an activity-dependent weight constraint for both models. This encourages a hidden neuron to receive input from the input neurons via W_{10} or from other lateral neurons via W_{11} but not both.

– In the Kalman filter model, 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 receive the following activity dependent weight constraint which is added to the weight learning rules:

$$\begin{aligned} \Delta w_{10}^{ij^{constr}} &= -\lambda^w \, |\bar{h}^i| \, w_{10}^{ij} \, \|\boldsymbol{w}^i\|^2 \\ \Delta w_{11}^{ik^{constr}} &= -\lambda^w \, |\bar{h}^i| \, w_{11}^{ik} \, \|\boldsymbol{w}^i\|^2 \end{aligned}$$

where λ^w is a scaling factor. $\|\boldsymbol{w}^i\|^2 = \sum_l^N (w_{10}^{il})^2 + \sum_l^H (w_{11}^{il})^2$ 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. Generative weights are made symmetric to the recognition weights, i.e. $V_{01} = W_{10}^T$ and $V_{11} = W_{11}^T$.

– For the Helmholtz machine, the weight constraint term which is added to the learning rule treats positive and negative weights separately. Using the Heaviside function $\Theta(x) = 1$, if $x > 0$, otherwise 0, we can write:

$$\Delta w^{ij^{constr}} \;=\; -\lambda^w \, \bar{h}^i \, \Theta(w^{ij}) \, w^{ij} \sum_{j'} \Theta(w^{ij'}) \, (w^{ij'})^2$$

where $\bar{h} = \boldsymbol{u}_1^w + \boldsymbol{u}_2^w + \boldsymbol{u}_1^s + \boldsymbol{u}_2^s$ is the sum of all activations which have been induced by the recognition weights. The indices j, j' extend over all input and hidden units. The wake-sleep algorithm is not a gradient descent in an energy space. It easily gets stuck in local minima. To improve the solutions found, generative weights V_{01} and V_{11} as well as lateral recognition weights W_{11} were rectified, i.e. negative weights were set to zero.

The weight constraints scale the length but do not change the direction of a hidden neuron weight vector. They are local in the sense that they do not depend on any weight of any other hidden neuron.

2.6 Distinguishing the Modules

In order to help two distinct areas to evolve, the hidden neurons are separated into two groups by assigning them different intrinsic parameters of their transfer functions. The key idea is that neurons from one area are more active than neurons from the other. The more active neurons will then respond to the more frequent features that can be extracted from the data; the less active neurons are expected to learn those features which do not occur as often. Note that these differences can distinguish hierarchical levels.

– In the Kalman filter model, a difference in one parameter among the hidden neurons will be sufficient to initiate the segregation process, either in parallel or hierarchically, depending on the data. The transfer function

$$f(h_i) \;=\; h_i \;-\; \lambda \cdot \frac{2h_i}{1 + h_i^2}$$

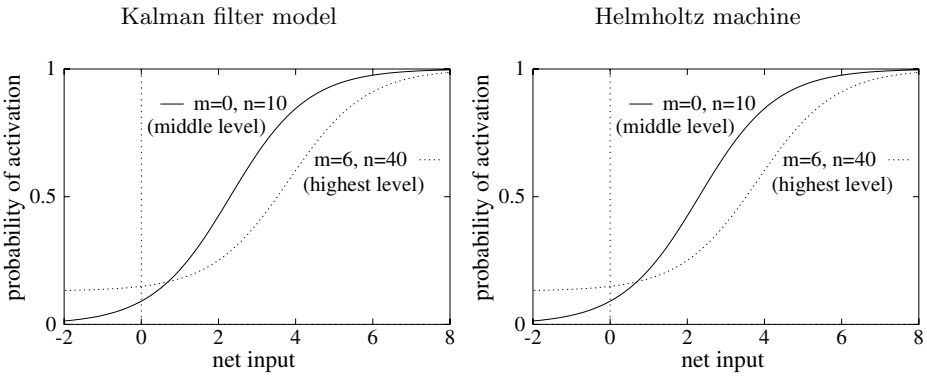

Fig. 4. Neuronal transfer functions.

is depicted in Fig. 4, left, with the two different values of the parameter λ which controls the sparseness of neuronal firing. Larger values of λ make a neuron respond with smaller activation to a given input.

– For the Helmholtz machine a stochastic transfer function is chosen for the hidden neurons (Fig. 4, right):

$$f_m(h_i) = \frac{e^{h_i} + m}{e^{h_i} + m + n}$$

By this function, the stochastic "ON"-state can be traced back to two distinguished sources. First, the activation from the neurons input, h_i and second, the parameter m. Both increase the probability for the neuron to be "ON". We refer to the latter as spontaneous or endogenous activity. The parameter n adds some probability for the neuron to be "OFF", thus encourages sparse coding.

We chose the parameters such that they matched the precise role the neurons should play in the wake-sleep algorithm. Especially the hierarchical setting has to be considered as it is more difficult to achieve this kind of structure. Two distinct physiological properties of the neurons are important for the role they play in the wake-sleep algorithm and thus two parameters are varied. *(i)* Neurons in the highest hierarchical level are responsible for initiation of the hidden code, i.e. they have to be spontaneously active without any primary input. For this reason we assigned high resting activity to the units which are designed for the higher level by setting the parameter m to a high value. *(ii)* Lower level neurons should respond to input. This applies both to recognition when there is input from the input neurons and to generation when there is input from the highest units. High responsitivity is achieved by a strong gain – or a small sparseness parameter n.

3 Results

3.1 Generation of the Training Data

The data consist of discrete, sparsely generated elements. These are lines of 4 different orientations on a 5×5 grid of input neurons. In the parallel paradigm, horizontal and 45° lines are generated with probability 0.05 whereas the other group, vertical and 135° lines are generated twice as often, with probability 0.1 each (Fig. 5, **a)**). As a result of training, the group of more active neurons is expected to preferably code for the more frequent data elements, and vice versa (cf. [14][13]).

Fig. 5. Examples of stimuli $\boldsymbol{x}$ used for training. **a)** Stimuli generated by two models in parallel. **b)** Stimuli generated by a hierarchical model. White means a positive signal on a grey zero value background.

In the hierarchical paradigm, one of 4 orientations are chosen, 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). See Fig. 5, **b)**.

3.2 Results of Training

Parallel Structure: Parts **a)** of Figs. 6 and 7 show the resulting weight matrices after both nets have been trained on the parallelly generated data. The neurons have extracted the independent lines from the data. In general, neurons in the upper half (Kalman filter) or upper third (Helmholtz machine) code for the more frequent lines (90° and 135°, in polar coordinates), whereas neurons in the lower parts code for the less frequent lines (0° and 45°).

The reason for this division of labor is that neurons in the upper part are more active. In case of the Helmholtz machine, the critical factor is the resting activity, more than the responsitivity to input. The reason for this may be that in the initial phase of training, input is generally low and the resting activity accounts for most of the learning.

Hierarchical Structure: Parts **b)** of Figs. 6 and 7 show the resulting weight matrices after training on the hierarchically generated data. Only in the case of the Helmholtz machine, the input is decomposed into the independent lines; the Kalman filter model generates the data using a superposition of several, more complex features each of which reflect one orientation but not an individual line.

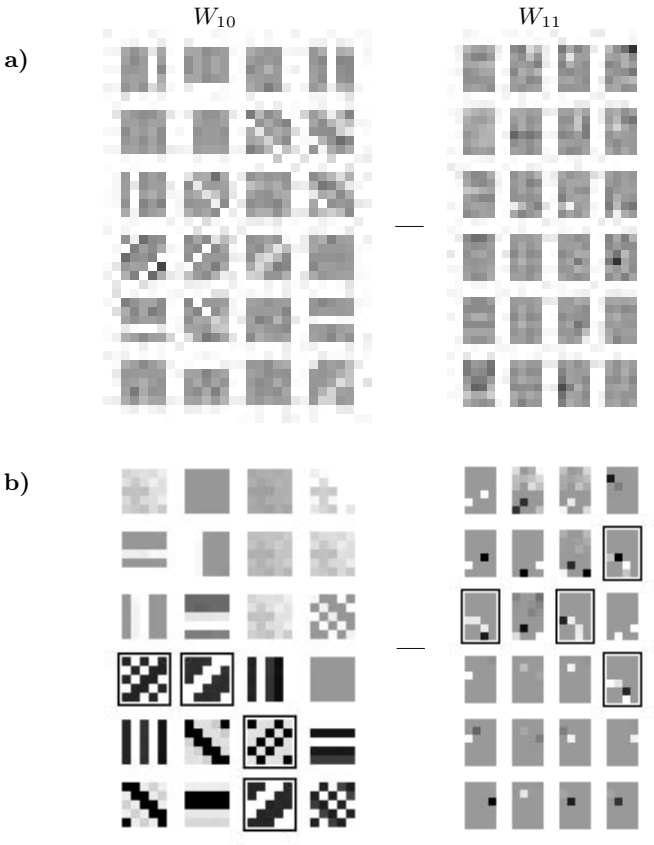

Fig. 6. Kalman filter model results. **Left:** the recognition weight matrices W_{10} and **right:** the lateral recognition weight matrices W_{11} after training. Each square of the weight matrices shows a receptive field of one of the 4×6 hidden neurons. Each neuron has weights to one of the 5×5 input neurons (**left**) and lateral weights (**right**). Here, black indicates negative, white positive weights. Contrast is sharpened by a piecewise linear function such that weights weaker than 10 percent of the maximum weight value are not distinguished from zero and weights stronger than 60 percent of it appear like the maximum weight value. Short lines between left and right matrices indicate the area boundaries.
a) Parallel organization of areas: weights W_{10} to the inputs generally code for 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 lower half code for the input via W_{10} while mainly neurons in the upper half group together some of the neurons in the lower half via W_{11}. Receptive fields which code for lines of 45° orientation are marked by a frame.

In the case of the Helmholtz machine, such a superposition cannot generate the data because the generative weights V are constrained to be positive.

Hierarchical structure has emerged such that, first, the less active neurons in the lower parts have pronounced weights to the input, and second, the more

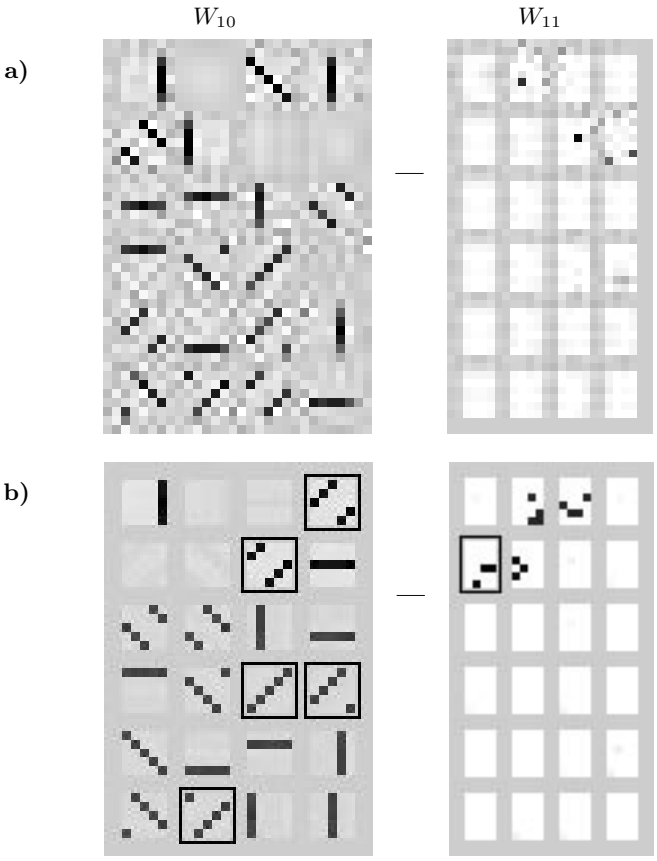

Fig. 7. Helmholtz machine model results. Architecture and display as in Fig. 6. Negative weights are brighter than the background (frame), positive weights are darker. For lateral weights (**right**), zero weights are depicted white (there are no negative weights). **a)** Areas have organized in parallel: weights W_{10} to the inputs code for 90° and 135° lines in the upper third and predominantly for 0° and 45° lines in the lower two thirds. **b)** Areas have organized hierarchically: neurons in the lower two thirds code for the input via W_{10} while four neurons in the upper third each integrate via W_{11} units from the lower two thirds which code stimuli of one direction. Neurons which code for lines of 45° orientation are marked by a frame.

active neurons in the upper parts have more lateral weights to the less active hidden neurons. These lateral connections group together neurons which code for the same orientation. In the Kalman filter model, results are fuzzy, because even neurons on the second level code in a distributed manner. Some neurons also code on both the first and second hierarchical level.

Outliers are found in both models. Note that it is only the data that changed between the parallel and vertical setting. All parameters and also initial conditions like randomized weights were the same in both settings for each model.

4 Discussion

Fig. 8 overlays the model computations onto the neural circuitry of the cortex (cf. [9]). Model weights W and V are identified with bottom-up and top-down inter area connections, respectively. Computations which are local in the model are identified to computations within a cortical "column". This means that they are localized along the surface of the cortex but extend through the six layers. We identify layer 4 as the locus of bottom-up information reception and layer 6 as the locus of top-down information reception. Layers 2/3 integrate both of these inputs and are the source of the outputs (after transmission through a transfer function). These go directly to a higher area and via layer 5 (not considered in the model) to a lower area.

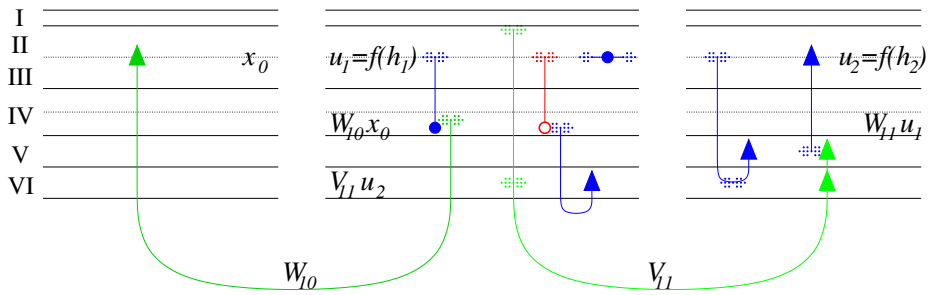

Fig. 8. Proposed computations of neurons, projected into the six cortical layers. The hierarchical level of the three areas increases from left to right. The drawn connections are proposed to be the main connections collected from anatomical literature. The area depicted to the left corresponds to the input layer of the models. There, data $\boldsymbol{x}_0$ origin from layers 2/3. Alternatively, one can treat this as thalamic input.

Both models, the non-linear Kalman filter model and the Helmholtz machine, can be identified with this connection scheme and the notations of the model equations can be applied to Fig. 8. Both models use basic computations like the scalar product between the inter-area weights and the activations from the source area of an input. These computations follow directly from the anatomy.

The differences between the models lie in the way bottom-up and top-down input is integrated within one area. In the Kalman filter model, all activation values $\boldsymbol{u}_1$, $\boldsymbol{u}_2$ which are needed for training are computed incrementally, thus, they are maintained over a whole relaxation period. It is biologically implausible that each hidden neuron keeps track of both values. In the Helmholtz machine (wake-sleep algorithm), activations from the bottom-up and the top-down path are computed serially, they flow from one hierarchical level to the next. Even though a neuron belongs logically to two hierarchical levels, when it is activated on a certain level it can forget the previous activation on another level. The update dynamics is biologically plausible and reminiscent of a synfire chain model [1]. However, during recognition there is no feedback from higher cortical areas.

In the case of the Kalman filter model, when the learning rule is evaluated, all relevant terms are present. In the case of the Helmholtz machine, the full learning rule is split into two parts which are evaluated at separate times, a "wake" mode when data activate the input units and a "sleep" mode when hidden neurons become spontaneously active. If data is missing, we have shown that spontaneous activity on its own can in principle mould the internal structure of the network [13] in a restricted way.

More subtle differences between the two models certainly cannot show up in the anatomy: the time order of the neuronal computations, the initialization of activities and the learning rules. On the other hand, there are restrictions in the models: only the main streams are considered, lateral connections are omitted. Furthermore, in the models, no learning takes place within a cortical "column".

Acknowledgements. The writing-up was done in the lab of Alexandre Pouget and Sophie Deneve and Suliann Ben Hamed corrected this work.

References

1. M. Abeles. *Corticonics. Neural Cirquits of the Cerebral Cortex.* Cambridge University Press, 1991.
2. Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neur. Comp.*, 7(6):1129–1159, 1995.
3. P. Dayan, G. E. Hinton, R. Neal, and R. S. Zemel. The Helmholtz machine. *Neur. Comp.*, 7:1022–1037, 1995.
4. 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.
5. D.J. Felleman and D.C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1:1–47, 1991.
6. 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.
7. S. Haykin. *Neural Networks. A Comprehensive Foundation.* Macmillan College Publishing Company, 1994.
8. G. E. Hinton, P. Dayan, B. J. Frey, and R. Neal. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.
9. M. Kawato, H. Hayakawa, and T. Inui. A forward-inverse optics model of reciprocal connections between visual cortical areas. *Network*, 4:415–422, 1993.
10. B.A. Olshausen. Learning linear, sparse, factorial codes. A.I. Memo 1580, Massachusetts Institute of Technology., 1996.
11. 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.
12. J.W. Scannell, C. Blakemore, and M.P. Young. Analysis of connectivity in the cat cerebral cortex. *J. Neurosci.*, 15(2):1463–1483, 1995.
13. C. Weber and K. Obermayer. Emergence of modularity within one sheet of intrinsically active stochastic neurons. In *Proceedings ICONIP*, 2000.
14. C. Weber and K. Obermayer. Structured models from structured data: emergence of modular information processing within one sheet of neurons. In *Proceedings IJCNN*, 2000.
15. M.P. Young. The organization of neural systems in the primate cerebral cortex. *Proc. R. Soc. Lond. B*, 252:13–18, 1993.