Proc. of the 5th Joint Symposium on Neural Computation, San Diego, CA, pp. 196-203, (1998)
(two errors have been marked and corrected in this version)
(due to font problems some labels in the figures are wrong, read 'H'='(', 'L'=')', 'h'='η')

Learning Invariance Manifolds

Laurenz Wiskott
Computational Neurobiology Laboratory
The Salk Institute for Biological Studies
San Diego, CA 92186-5800
wiskott@salk.edu, http://www.cnl.salk.edu/CNL/

Abstract

A new algorithm for learning invariance manifolds is introduced that allows a neuron to learn a non-linear transfer function to extract invariant or rather slowly varying features from a vectorial input sequence. This is generalized to a group of neurons, referred to as a Gibson-clique, to learn slowly varying features that are uncorrelated. Since the transfer functions are non-linear, this technique can be applied iteratively. Four examples demonstrating the properties of the learning algorithm include learning complex cell response with one Gibson-clique and learning translation invariance in a hierarchical network of Gibson-cliques.

1 Introduction

> *Third [...], the process of perception must be described. This is not the processing of sensory inputs, however, but the extracting of invariants from the stimulus flux.* (Gibson, 1986, p. 2)

Learning invariant representations is one of the major problems in neural systems. The approach described in this paper is conceptually most closely related to work by Földiák (1991), Mitchison (1991), Becker & Hinton (1995) and Stone (1996). The idea is that while an input signal may change quickly due to changes in the sensing conditions, e.g. scale, location, and pose of the object, certain aspects of the input signal change slowly or rarely only, e.g. the presence of a feature or object. The task of a neural system in learning invariances is therefore the extracting of slow aspects from the input signal.

On an abstract level, the input $\mathbf{x} = \mathbf{x}(t)$ of a sensor array can be viewed as a trajectory in a high-dimensional input space. Many points in this space can represent the same feature if they only differ in their sensing conditions. One can imagine that these points lie on a manifold (cf. Lu et al., 1996), which may be called *invariance manifold.* Looking at an object under varying sensing conditions means that the trajectory lies within the invariance manifold. Saccading to a new object, for instance, will cause a jump in the trajectory with a component perpendicular to the manifold. Here, a single manifold is defined by an equipotential surface of a scalar transfer function $g(\mathbf{x})$ in the high-dimensional space. The set of all equipotential surfaces defines a (continuous) family of manifolds. This can be extended to a set of transfer functions $g_i(\mathbf{x})$ providing a set of manifold families.

The proposed algorithm differs from the work by Földiák (1991), Mitchison (1991), Becker & Hinton (1995) and Stone (1996) in the mathematical formulation, one distinct feature being that input signals are individually combined in a non-linear fashion, which follows the idea that complex non-linear computation can be performed by the dendritic tree (Mel, 1994). Furthermore, the system is formulated as a learning algorithm rather than an online learning rule, and it is naturally generalized to a group of output neurons, here referred to as a *Gibson-clique.*

2 The Learning Algorithm

Consider a neuron that receives an N-dimensional input signal $\mathbf{x} = \mathbf{x}(t)$ where t indicates time and $\mathbf{x} = [x_1, ..., x_N]^T$ is a vector. The neuron is able to perform a non-linear transformation on this input defined as a weighted sum over a set $\mathbf{h} = [h_1, ..., h_M]^T$ of M non-linear functions $h_m = h_m(\mathbf{x})$

(usually $M > N$). Here polynomials of order two are used, but other sets of non-linear functions could be used as well. Applying $\mathbf{h}$ to the input signal yields the non-linearly expanded signal $\mathbf{h}(t) \equiv \mathbf{h}(\mathbf{x}(t))$. The set of weights $\mathbf{w} = [w_1, ..., w_M]^T$ is subject to learning and the final output of the neuron is given by $y(t) \equiv g(\mathbf{x}(t)) \equiv \mathbf{w}^T\mathbf{h}(\mathbf{x}(t))$. g is called the transfer function of the neuron. Notice that it is much more complex than the common sigmoidal functions employed in conventional model neurons, since it combines individual components of the input signal in a non-linear fashion.

The objective is to optimize the weights such that the output is as invariant as possible, i.e. the weights minimize the mean square of the time derivative

$$\Delta(y) \equiv \langle \dot{y}^2 \rangle = \mathbf{w}^T \langle \dot{\mathbf{h}}\dot{\mathbf{h}}^T \rangle \mathbf{w}, \tag{1}$$

a quantity that will be referred to as the Δ-value. $\langle \cdot \rangle$ indicates the temporal mean.

This objective alone would lead to a system that learns constant features that do not change over the input signal or, more likely, a system that learns no features by setting $\mathbf{w} = \mathbf{0}$. Thus a constraint needs to be imposed such that the output signal conveys some information. This implies that a feature needs to change at least slowly or rarely, so that it is actually not invariant features that are learned but slowly varying features or, for short, slow features. However, the manifolds themselves defined by $g = \text{const}$ can still be considered invariance manifolds. The constraint that features have to convey some information is here formalized by requiring that the variance of the output signal be unity,

$$\langle y^2 \rangle - \langle y \rangle^2 = \mathbf{w}^T \underbrace{\left(\langle \mathbf{h}\mathbf{h}^T \rangle - \langle \mathbf{h} \rangle \langle \mathbf{h}^T \rangle\right)}_{=\mathbf{I}} \mathbf{w} = \mathbf{w}^T\mathbf{w} = 1. \tag{2}$$

It is assumed here that the signals produced by the non-linear functions h_m have zero mean and a unit covariance matrix. A sphering stage has to be applied to an arbitrary set of non-linear functions $\mathbf{h}'$ to derive the set $\mathbf{h}$ with these properties.

Minimizing the Δ-value under this constraint is equivalent to finding the normalized eigenvector with minimal eigenvalue for matrix $\langle \dot{\mathbf{h}}\dot{\mathbf{h}}^T \rangle$ (cf. Mitchison, 1991). The minimal eigenvalue is equal to $\langle \dot{y}^2 \rangle$. The eigenvectors of the next higher eigenvalues produce uncorrelated neurons with the next higher Δ-values. These can be useful if several uncorrelated features need to be extracted. They can also be used to propagate enough information through a cascade of transfer functions (cf. the third example in Section 3).

It is useful to measure the invariance of signals not by the Δ-value directly but by a measure that has a more intuitive interpretation. A good measure may be an index η defined by $\eta(y) \equiv$ ~~$\sqrt{\Delta(y)\,T/(4\pi^2)}$~~ if $t \in [0, T]$. For a pure sine wave ~~sin~~$(n\, 2\pi\, t/T)$ with an integer number of oscillations n the index η is just the number of oscillations, i.e. $\eta = n$. Thus the index η of an arbitrary signal indicates what the number of oscillations would be for a pure sine wave of same Δ-value, at least for integer values of η.

$\frac{T}{2\pi}\sqrt{\Delta(y)}$

$\sqrt{2}$ sin(...)

3 Examples

The properties of the learning algorithm are now illustrated by four examples. The first example is about learning complex cell behavior based on simple cell outputs. The second one is similar but also includes disparity estimation. One Gibson-clique of second order polynomials is sufficient for these two examples. The third example is abstract and requires a more complicated transfer function, which can be approximated by three Gibson-cliques in succession. This illustrates how Gibson-cliques can learn invariances in an iterative scheme. A hierarchical network of Gibson-cliques is considered in the last example, which is a model of a visual system learning translation invariance.

Example 1: The example for learning complex cell behavior based on simple cell outputs follows the view that simple cells can be modeled by Gabor wavelets (Jones & Palmer, 1987), while a complex cell combines the responses of several simple cells of same orientation and location but different phase, e.g. taking the square sum of a cosine and a sine Gabor-wavelet. The response of a Gabor-type simple cell to a visual stimulus smoothly moving across the visual field can be modeled by a combination of time varying amplitude $a(t)$ and phase $\phi(t)$. Two simple cells of same orientation and location have same amplitude and phase modulation but a constant phase difference depending on the phase shift of their receptive fields (cf. Fig. 1). In this example the signals of three simple cells, two at same location and one at a different location, are modeled by $x_1(t) \equiv (4+a_1(t))\sin(t+2\phi_1(t))$, $x_2(t) \equiv (4+a_1(t))\sin(t+2\phi_1(t)+\pi/4)$, and $x_3(t) \equiv (4+a_2(t))\sin(t+2\phi_2(t))$, $t \in [0, 4\pi]$. All signals have a length of 512 data-points. The amplitude and phase modulation signals $a_1(t)$, $a_2(t)$, $\phi_1(t)$, and $\phi_2(t)$ are low-pass filtered Gaussian noise with zero mean and unit variance. The width σ of the Gaussian low-pass filter is 10 data-points for all of these signals. Notice that the phase difference between the first two simple cells is 45° and not 90°. The latter would be more convenient, but is not necessary. The complex-cell response that can be extracted from these three simple cells is the amplitude modulation signal $a_1(t)$.

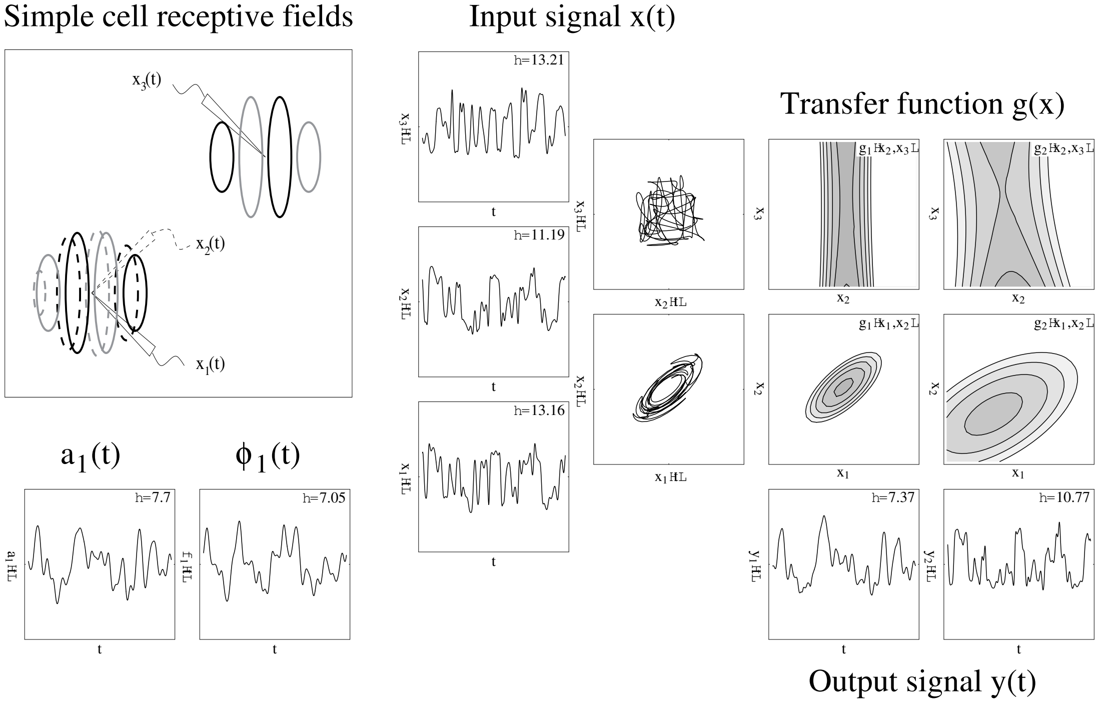
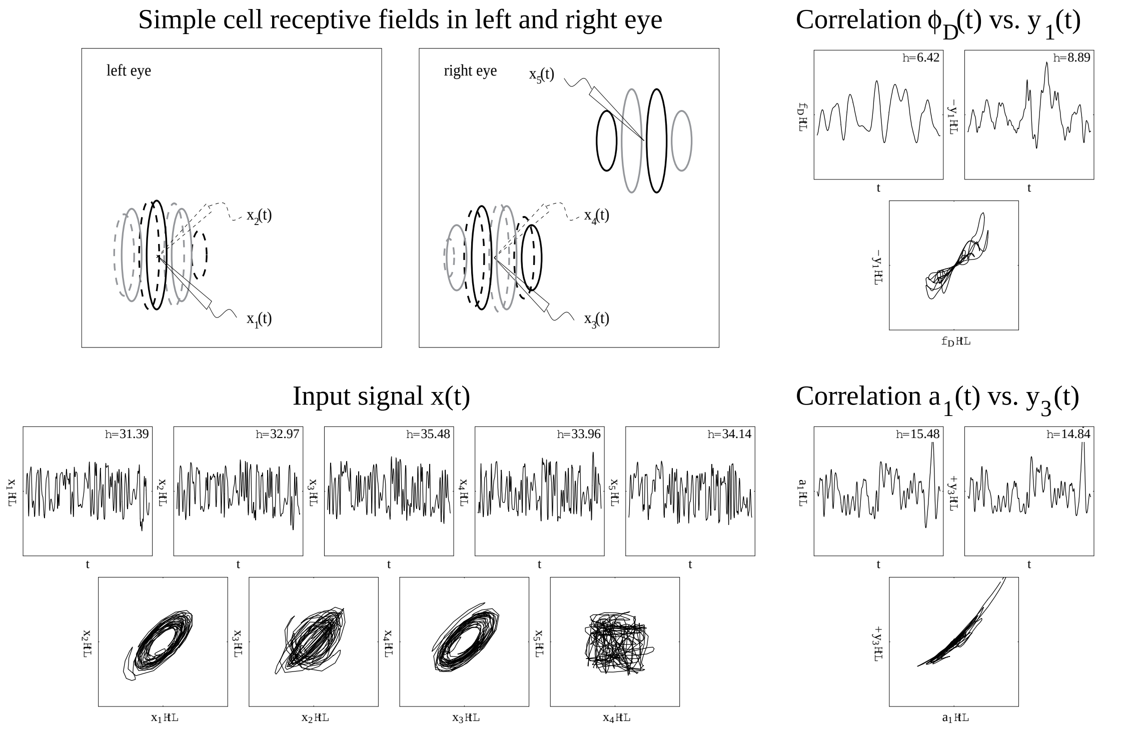


Figure 1: Learning complex cell response with one Gibson-clique.

Figure 1 upper left shows a sketch of the receptive fields of the three hypothetical simple cells. The generating amplitude and phase signals $a_1(t)$ and $\phi_1(t)$ are shown below. On the right is shown the input signal $\mathbf{x}(t)$, cross-sections through the learned transfer function $\mathbf{g}(\mathbf{x})$ (arguments not varied are set to zero, e.g. $g_1(x_2, x_3)$ means $g_1(0, x_2, x_3)$), and the extracted output signal $\mathbf{y}(t)$. All signals have unit variance and all graphs range from -4 to $+4$, including the grey value scale of the contour plots. Time axes range from 0 to 4π.

The amplitude comodulation and 45° phase relationship of the first two simple cells is reflected in the elliptic form of trajectory plot $x_2(t)$ vs. $x_1(t)$. The slow component of this signal is its distance from the center weighted according to the elliptic shape. The third simple cell has no relationship to the first two (see, for example, trajectory plot $x_3(t)$ vs. $x_2(t)$). The first component of the learned transfer function, $g_1(\mathbf{x})$, correctly represents the elliptic shape of the $x_2(t)$ vs. $x_1(t)$ trajectories and ignores signal $x_3(t)$. It therefore extracts the amplitude signal $a_1(t)$ as desired (compare signals $a_1(t)$ and $y_1(t)$). The correlation between $a_1(t)$ and $y_1(t)$ is 0.97. The η-index of $y_2(t)$ is almost as high as the one of $x_2(t)$ which indicates that $y_2(t)$ does not represent another slow feature (as one would also expect from the way the input signal was generated). Thus, $g_2(\mathbf{x})$ could be discarded. When trained on signals which are 2048 data-points long, the correlation between $a_1(t)$ and $y_1(t)$ is 0.981±0.004 on training data and 0.93±0.04 on test data (mean over 10 runs ± standard deviation).

Example 2: The previous example can be extended to two eyes (see Fig. 2). The input signal is given by a pair of simple cells in the left eye, a pair of simple cells in the right eye at a corresponding location of the visual field, and a fifth unrelated simple cell. As in the previous example, it is assumed that some visual stimulus is smoothly moved across the receptive fields. Left and right eye are assumed to receive identical input but slightly shifted relative to each other depending on the disparity. The phase relationship between related simple cell signals is determined by the relative phase shift of their receptive fields. In addition, the phase relationship between two simple cell signals from different eyes is modulated by the disparity, indicated by the phase modulation $\phi_D(t)$. The input signal is modeled by $x_1(t) \equiv (4+a_1(t))\sin(t+2\phi_1(t))$, $x_2(t) \equiv (4+a_1(t))\sin(t+2\phi_1(t)+\pi/4)$, $x_3(t) \equiv (4+a_1(t))\sin(t+2\phi_1(t)+\pi/2+0.5\phi_D(t))$, $x_4(t) \equiv (4+a_1(t))\sin(t+2\phi_1(t)+3\pi/4+0.5\phi_D(t))$, and $x_5(t) \equiv (4+a_2(t))\sin(t+2\phi_2(t))$, $t \in [0, 8\pi]$. All signals have a length of 1024 data-points. The amplitude and phase modulation signals are low-pass filtered Gaussian noise with zero mean and unit variance. The width σ of the Gaussian low-pass filters is 30 data-points for $\phi_D(t)$ and 10 for the other four signals, $a_1(t)$, $a_2(t)$, $\phi_1(t)$, and $\phi_2(t)$.


Simple cell receptive fields in left and right eye

Correlation ϕ_D(t) vs. y_1(t)

Input signal x(t)

Correlation a_1(t) vs. y_3(t)

Figure 2: Learning disparity and complex cell response with one Gibson-clique.

Figure 2 upper left shows a sketch of the receptive fields of the five hypothetical simple cells. The input signal coming from the five simple cells is shown below. The generating disparity phase modulation signal $\phi_D(t)$ and amplitude modulation signal $a_1(t)$ are shown on the right, plotted vs. the first and third output signal components, respectively. The correlation coefficients are 0.91 and 0.97 for $\phi_D(t)$ vs. $y_1(t)$ and $a_1(t)$ vs. $y_3(t)$, respectively. When trained on signals which are 2048 data-points long, the correlation between $\phi_D(t)$ and $y_1(t)$ is 0.87±0.03 on training data and 0.86±0.03 on test data (mean over 10 runs ± standard deviation). The correlation between $a_1(t)$ and $y_3(t)$ is 0.92±0.04 and 0.89±0.06 on training and test data, respectively. These high correlations demonstrate that a Gibson-clique is able to extract several meaningful invariants at the same time, here disparity and complex cell response. However, this requires that the invariants are clearly separated by their different time scale on which they vary.

Example 3: The first two examples were particularly easy because a second order polynomial was sufficient to recover the slow feature well. The third example is more complex. First generate a slowly and a fast varying random time series $x_s(t)$ ($\sigma = 20$ data-points) and $x_f(t)$ ($\sigma = 6$ data-points), respectively. Both signals have a length of 1024 data-points, zero mean, and unit variance. They are then mixed to provide the input signal $\mathbf{x}(t) \equiv [x_f, \sin(2x_f) + 0.5x_s]^T$. The task is to extract the slowly varying random time series $x_s(t)$

The transfer function required to extract the slow feature x_s cannot be well approximated by a second order polynomial. One might therefore use third or higher order polynomials. However, one can also iterate the learning algorithm, applying it with second order polynomials repeatedly, leading to transfer functions of second, fourth, eighth, sixteenth order etc. To avoid an explosion of the signal dimensionality only the first components of the output signal of one Gibson-clique should be used as an input for the next clique. In this example only three components of an output signal are transfered to the next Gibson-clique. This cuts down the computational cost of this iterative scheme significantly compared to the direct scheme of using higher order polynomials.

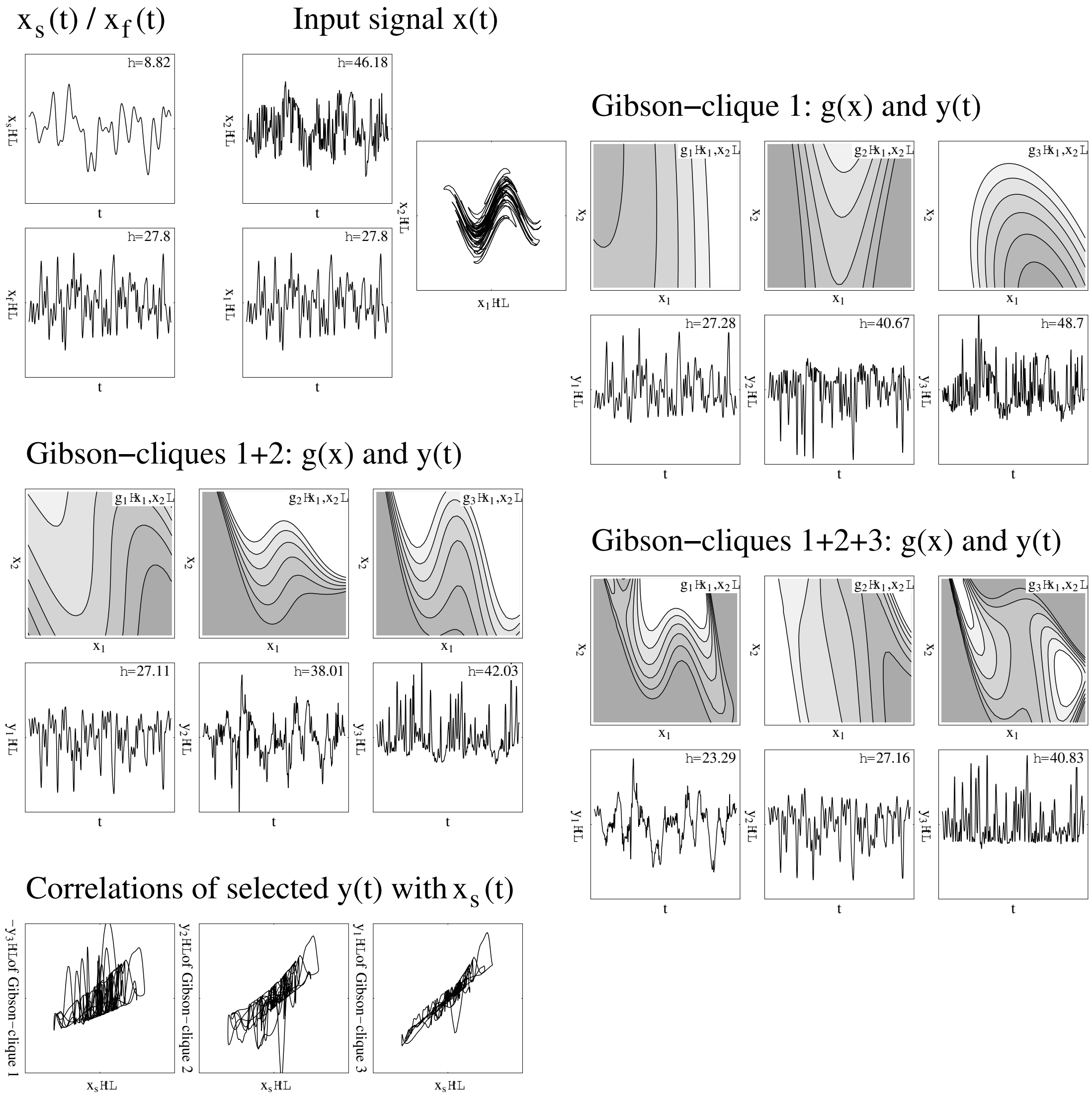


Figure 3: Hidden slow signal discovered by three Gibson-cliques in succession.

Figure 3 shows slow and fast random time series, input signal, and transfer functions as well as output signals for three Gibson-cliques in succession. The plotted transfer functions always include the transformations done by previous Gibson-cliques, too. The correlations between selected output signals and $x_s(t)$ are illustrated at the bottom left. These are the output signals with the highest correlation with $x_s(t)$, namely $y_3(t)$, $y_2(t)$, and $y_1(t)$ for the first, second, and third Gibson-clique, respectively. The corresponding correlation coefficients are 0.48, 0.76, and 0.94. Several Gibson-cliques are necessary to approximate the wave-like invariance manifolds that can be seen in the $x_2(t)$

vs. $x_1(t)$ trajectory plot. Notice that each Gibson-clique is trained in an unsupervised manner and that no back-propagation of any kind of error-signal is required. The correlation between $y_2(t)$ of the third Gibson-clique and the fast random time series $x_f(t)$ is 0.98, which means that also $x_f(t)$ is represented. Even though this signal is easy to extract, it is not obvious why it should be extracted in pure form and not mixed with other aspects of the input signal.

When trained on signals of length 8192 data-points, the maximum correlation between $x_s(t)$ and the first three components of the output signal is 0.85±0.10 on training data and 0.74±0.20 on test data (mean over 10 runs ± standard deviation). The maximum correlation is taken because occasionally the slow signal gets extracted by the second or third component of the output signal. The generalization to test data is worse than in the previous examples. The maximum correlation between $x_f(t)$ and the first three components of the output signal is 0.93±0.09 on training as well as test data.

Example 4: Consider now a hierarchical architecture as illustrated in Figure 4 (upper left). A one-dimensional retina (R) is densely packed with sensor neurons. The distance between two sensor neurons defines one spatial unit. Layer 1a consists of Gibson-cliques with 9 neurons each. Each clique receives input from nine adjacent retinal sensors, and the learning algorithm is applied independently to each clique, generating a nine-dimensional output signal per clique (these are only the first nine components of an output signal that could be potentially 54-dimensional for second order polynomials). Layer 1b consists of Gibson-cliques with 3 neurons, receiving the nine-dimensional signals from corresponding Layer 1a cliques. Each of the upper Layers 2, 3, and 4 consists of Gibson-cliques with 3 (or 4) neurons receiving three-dimensional inputs from three adjacent cliques in the lower layer. The number of neurons per Gibson-clique is shown to the right of the rightmost clique in each layer. Due to this hierarchical architecture the receptive field size increases as 1, 9, 9, 17, 35, up to 65 as one proceeds from retina to upper layers. Receptive field size is indicated to the left of the leftmost clique in each layer.

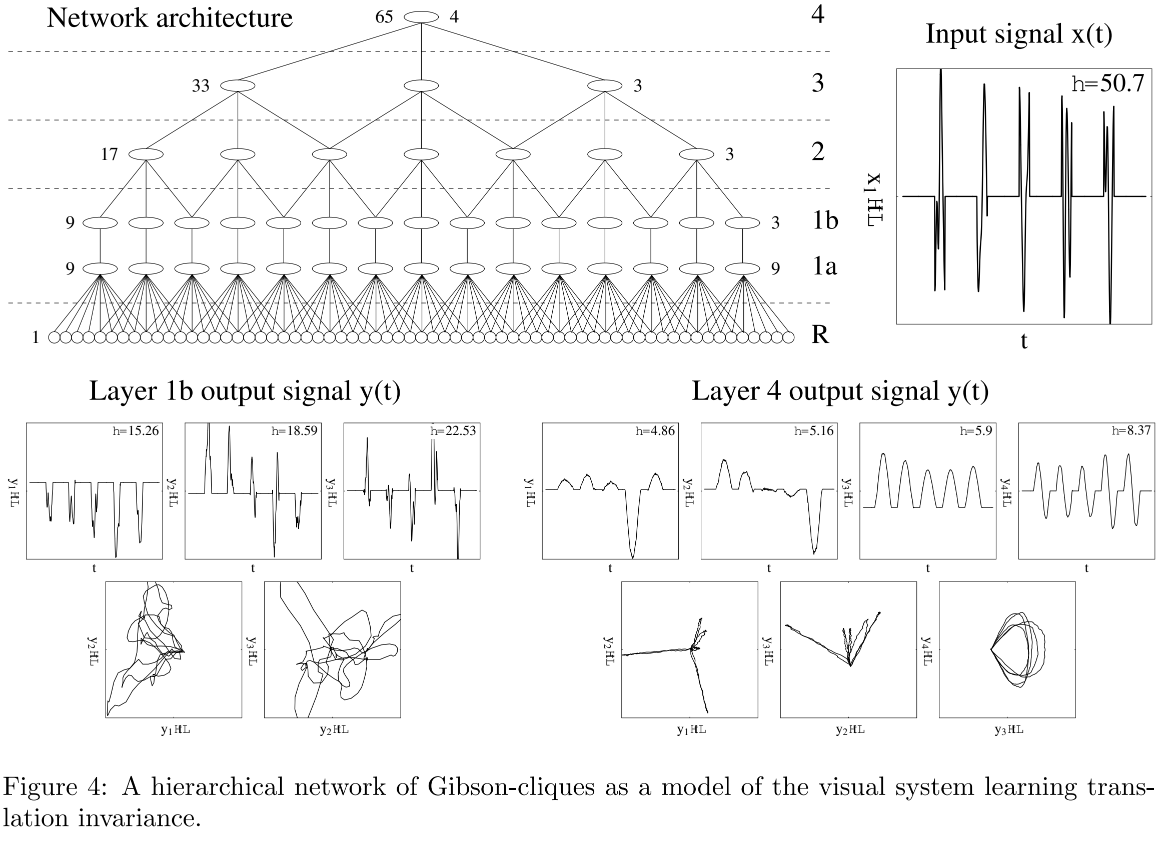


Figure 4: A hierarchical network of Gibson-cliques as a model of the visual system learning translation invariance.

The stimulus is a spatial pattern moving across the retina with a constant velocity of one spatial unit per time unit, so that the temporal stimulus for a single retinal sensor is identical to the spatial pattern, except for a shift dependent on the position of the sensor. The spatial (or temporal) pattern consists of five objects of low-pass filtered Gaussian noise (Fig. 4, upper right). Each object has a width of 32. Two objects are separated by a gap of 100. The whole stimulus is 760 units long and presented cyclically to the retina.

The response of Layer 1b Gibson-cliques and the Layer 4 Gibson-clique are shown at the bottom of Figure 4. Notice that the time axes can also be used as spatial axes, since the stimulus has been moved across the retina with velocity one. The neurons begin to respond to an object as soon as it moves into the receptive field and continue to respond as long as parts of it are still within the receptive field. This leads to a longer response of the Layer 4 neurons than of the Layer 1b neurons.

Since the system tends to extract slowly varying aspects of the input signal, one can expect that it focuses only on the presence and absence of objects, so that the output signal does not follow any details of the pattern. The output signal of Gibson-cliques in Layer 1b is already more slowly varying than the raw object signals (compare the η-values 15.26 and 50.7). However, the Layer 4 Gibson-clique generates a still smoother output signal, the first component of which has an η-value of less than five. The shape of the responses is approximately half a sine-wave per object, which would be the optimal invariant response with respect to the above objective. One might expect that the most invariant response would be constant within a certain range with a sharp on- and off-set as an object enters and leaves the receptive field. However, such a signal would be highly variant at the on- and off-set and therefore suboptimal.

The output signal of the Layer 4 Gibson-clique is not only highly invariant, it also discriminates between the five objects. This becomes clear from the trajectory plot $y_2(t)$ vs. $y_1(t)$. The response directions in the (y_1, y_2)-space relative to the null-response is different for all five objects. Although, the response to the third object is very weak. This tendency to discriminate between different objects despite the translation invariant response is not built into the system but results implicitly from the invariance objective and the constraints. It is interesting that the system also generates output signal components that do not differentiate well between objects, namely $y_3(t)$ and $y_4(t)$. These components only convey information as to where the object is located in the receptive field. They could therefore be interpreted as a *where*-signal while the first two component are a *what*-signal. However, it is not clear whether this *what* vs. *where* distinction of output signal components is not merely an accidental feature of this particular example.

Training on five objects allows the network to generate an ideal response. Figure 5 shows the Layer 4 output signal if the network was trained on 50 objects. On the left is shown the response to part of the training data, while on the right is shown the response to ten novel objects not contained in the training data (the signals in this figure are twice as long as in Figure 4). Even though the response is not as clean as in Figure 4 and the output components have a different order, the qualitative behavior is the same. It can also be seen that the system generalizes translation invariance well for the given class of objects.

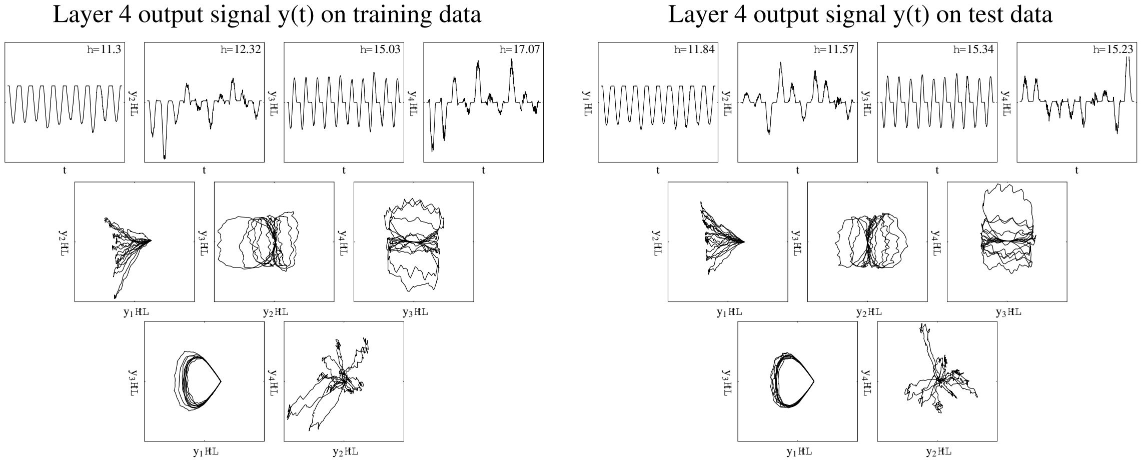


Figure 5: Generalization of translation invariance in a hierarchical network of Gibson-cliques.

There are two types of translation invariances inherent in this system. One is generated by the learning algorithm, which leads to the smooth and translation invariant response of single neurons to the stimulus pattern. Another one is due to the homogeneous design of the system. Since every retinal sensor gets the same input pattern (though shifted dependent on its location) and since the initial connectivity is translation invariant, the learned transfer functions are identical for all Gibson-cliques within one layer. This is as if one had applied a weight sharing constraint. However, this only means that the response of neurons is the same over the whole layer given the same local input. It does not mean that the response of an individual neuron is translation invariant, i.e. insensitive to a shift of the stimulus. Only if one would average over a spatial neighborhood, such as in the Neocognitron (Fukushima et al., 1983), this weight sharing property would lead to translation invariant responses of individual neurons. Since no spatial averaging is built in here, the system does not rely on the weight sharing property. The translation invariance of the output signals is due only to the learning algorithm. In fact, the learning algorithm should be able to learn translation invariant responses even if the architecture were designed in-homogeneously such that implicit weight sharing were prevented.

4 Conclusion

A new unsupervised learning algorithm has been presented and tested on several examples. With the algorithm a group of neurons, referred to as a Gibson-clique, can be trained to learn a high-dimensional non-linear transfer function to extract slow components from a vectorial input signal. Since the learned transfer functions are non-linear, the algorithm can be applied iteratively, so that complex transfer functions can be learned in a hierarchical network of Gibson-cliques with limited computational effort. Gibson-cliques may therefore be a useful component in developing a self-organizational model of sensory systems.

Acknowledgment

I am grateful to Terrence Sejnowski for his support and valuable feedback. I have been partially supported by a Feodor-Lynen fellowship by the Alexander von Humboldt-Foundation, Bonn, Germany.

References

Becker, S. and Hinton, G. E. (1995). Spatial coherence as an internal teacher for a neural network. In Chauvin, Y. and Rumelhart, D. E., editors, *Backpropagation: Theory, Architecture and Applications.*, pages 313–349. Hillsdale, N.J. : Lawrence Erlbaum Associates.

Földiák, P. (1991). Learning invariance from transformation sequences. *Neural Computation*, 3:194–200.

Fukushima, K., Miyake, S., and Ito, T. (1983). Neocognitron: a neural network model for a mechanism of visual pattern recognition. *IEEE Trans. on Systems, Man, and Cybernetics*, 13:826–834. Reprinted in *Neurocomputing*, J. A. Anderson and E. Rosenfeld, Eds., MIT Press, Massachusetts, pp. 526–534.

Gibson, J. J. (1986). *The Ecological Approach to Visual Perception.* Lawrence Erlbaum Associates, London. Originally published in 1979.

Jones, J. P. and Palmer, L. A. (1987). An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *J. of Neurophysiology*, 58:1233–1258.

Lu, H.-M., Hecht-Nielsen, R., and Fainman, S. (1996). Geometric properties of image manifolds. In *Proc. of the 3rd Joint Symp. on Neural Comp.*, volume 6, pages 53–60, San Diego, CA. Univ. of California.

Mel, B. W. (1994). Information processing in dendritic trees. *Neural Computation*, 6:1031–1085.

Mitchison, G. (1991). Removing time variation with the anti-Hebbian differential synapse. *Neural Computation*, 3(3):312–320.

Stone, J. V. (1996). Learning perceptually salient visual parameters using spatiotemporal smoothness constraints. *Neural Computation*, 8(7):1463–1492.