

# Pattern analysis and synthesis in attractor neural networks

H. Sebastian Seung  
Bell Laboratories  
Lucent Technologies  
Murray Hill, NJ 07974 USA  
seung@bell-labs.com

October, 1997

## Abstract

The representation of hidden variable models by attractor neural networks is studied. Memories are stored in a dynamical attractor that is a continuous manifold of fixed points, as illustrated by linear and nonlinear networks with hidden neurons. Pattern analysis and synthesis are forms of pattern completion by recall of a stored memory. Analysis and synthesis in the linear network are performed by bottom-up and top-down connections. In the nonlinear network, the analysis computation additionally requires rectification nonlinearity and inner product inhibition between hidden neurons.

One popular approach to sensory processing is based on generative models, which assume that sensory input patterns are synthesized from some underlying hidden variables. For example, the sounds of speech can be synthesized from a sequence of phonemes, and images of a face can be synthesized from pose and lighting variables. Hidden variables are useful because they constitute a simpler representation of the variables that are visible in the sensory input.

Using a generative model for sensory processing requires a method of pattern analysis. Given a sensory input pattern, analysis is the recovery of the hidden variables from which it was synthesized. In other words, analysis and synthesis are inverses of each other. There are a number of approaches to pattern analysis. In analysis-by-synthesis, the synthetic model is embedded inside a negative feedback loop[1]. Another approach is to construct a separate analysis model[2].

This paper explores a third approach, in which visible-hidden pairs are embedded as attractive fixed points, or attractors, in the state space of a recurrent neural network. The attractors can be regarded as memories stored in the network, and analysis and synthesis as forms of pattern completion by recall of a memory. The approach is illustrated with linear and nonlinear network architectures. In both networks, the synthetic model is linear, as in principal

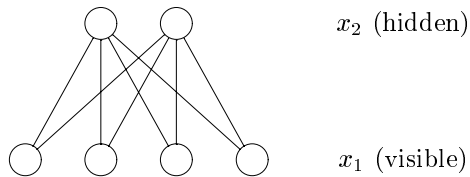


Figure 1: Linear network. The links between nodes are drawn undirected, because the top-down and bottom-up connections are symmetrical.

component analysis and its variants such as **Conic**[3]. Analysis is linear and feedforward in the linear network, but involves a rectification nonlinearity and lateral inhibition in the nonlinear network.

The idea of using attractor neural networks for content-addressable memory is not new[4]. The novel idea here is the use of attractors that are continuous manifolds, rather than discrete points. Furthermore, the relationship between attractor neural networks and hidden variable models has been relatively unexplored, with the exception of work on the Boltzmann machine, a stochastic generalization of attractor neural networks. The present work focuses on the problem of *representing* hidden variable models with attractor dynamics. The problem of learning continuous attractors from examples is addressed elsewhere[5].

The networks studied here are related to some recent brain models. To name just two examples, the neural integrator of the oculomotor system has been modeled as a linear network with an attractive line of fixed points[6], and a hypercolumn of visual cortex has been modeled as a network with rectification nonlinearity and a one-dimensional attractive manifold[7].

## 1 Linear network

The network of Figure 1 contains two layers of linear neurons. The activities of the bottom and top layers represent visible and hidden variables. It is assumed that there are fewer hidden variables than visible variables. The synaptic connections between the layers determine the relationships between visible and hidden variables.

The state of the network is described by two vectors  $x_1$  and  $x_2$ , and evolves in time according to the equations

$$\begin{aligned} \dot{x}_1 + x_1 &= W_{12}x_2, \\ \dot{x}_2 + x_2 &= W_{21}x_1. \end{aligned} \tag{1}$$

The  $x_1$  and  $x_2$  vectors are of dimension  $n_1$  and  $n_2$ . The  $n_2 \times n_1$  matrix  $W_{21}$  and the  $n_1 \times n_2$  matrix  $W_{12}$  contain the bottom-up and top-down connections, respectively.

Two conditions are placed on these synaptic connections:

$$W_{21}W_{12} = I , \tag{2}$$

$$W_{12} = W_{21}^T . \tag{3}$$

The first condition tunes the strength of the interlayer feedback loop. As will be explained shortly, this feedback loop causes the dynamics (1) to have a continuous manifold of fixed points. By the second condition of symmetric interactions, the dynamics (1) is gradient descent on the energy function

$$E = \frac{1}{2}|x_1|^2 + \frac{1}{2}|x_2|^2 - x_1^T W_{12} x_2 \tag{4}$$

$$= \frac{1}{2}|x_1 - W_{12} x_2|^2 \tag{5}$$

This follows from differentiation of  $E$ , and comparison with (1).

**Memory** The set of zero energy states

$$Z = \{(x_1, x_2) : x_1 = W_{12} x_2\} \tag{6}$$

is an  $n_2$ -dimensional linear space embedded in the  $n_1 + n_2$  dimensional subspace of all  $(x_1, x_2)$  pairs. It consists of fixed points of the dynamics (1), as can be shown using condition (2). Since gradient descent on  $E$  must converge to its minimal value, it follows that  $Z$  is an attractor of the dynamics (1).

It is worth examining more carefully the mathematical reasons for the existence of the continuous attractor  $Z$ . Fixed points of (1) are found by solving a homogeneous system of  $n_1 + n_2$  linear equations in as many unknowns. Generically, the only solution is the trivial  $x_1 = 0$  and  $x_2 = 0$ , which is a point attractor provided that the dynamics is stable. But in the special case of tuned feedback (2), the equations are underdetermined, and there is a whole space of solutions  $Z$ .

From the computational point of view, each visible-hidden pair  $(x_1, x_2)$  in  $Z$  is a *memory* stored in the network. Recall of a memory can be triggered by clamping a sufficiently large number (at least  $n_2$ ) of the neurons at the values of the memorized pattern. When the rest of the neurons are allowed to evolve freely, then the dynamics converges so that the rest of the values are “filled in.” This phenomenon is known as *pattern completion*, and illustrates the content-addressability of the memories in  $Z$ .

There are two types of pattern completion that are of particular interest, because they elucidate the computational roles played by the top-down and bottom-up connections. They are variants of the basic dynamics (1) in which either  $x_1$  or  $x_2$  is clamped, and are called pattern analysis and synthesis.

**Synthesis** If  $x_2$  is clamped, only the  $x_1$  dynamics is operative, driven by the top-down connections. The result is simple:  $x_1$  converges to  $x_1 = W_{12} x_2$ . In

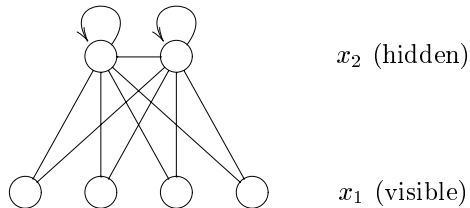


Figure 2: Nonlinear network. As in the linear network, there are top-down and bottom-up connections. In addition, each hidden neuron receives lateral inhibition and self-feedback. The nonlinearity of the neurons is a simple rectification.

other words, the pattern that appears in the visible layer is a linear superposition of columns of the matrix  $W_{12}$ , where the coefficients of the superposition are given by the hidden layer  $x_2$ . This operation, called pattern synthesis or generation, can be regarded as recall of the memory that matches  $x_2$ . Patterns that can be synthesized are those that lie in the column space of  $W_{12}$ , which is an  $n_2$ -dimensional subspace of  $R^{n_1}$ .

**Analysis** If  $x_1$  is clamped, the  $x_2$  dynamics converges to  $x_2 = W_{21}x_1$ , driven by the bottom-up connections. This computation is best understood as an optimization. The  $x_2$  dynamics (1) minimizes  $E$  with respect to  $x_2$ ,

$$\min_{x_2} \frac{1}{2} |x_1 - W_{12}x_2|^2 \quad (7)$$

with  $x_1$  held fixed. If there is a memory that matches  $x_1$ , it is recalled, and the minimal value of  $E$  is zero. If there is no exact match, the final state of the network is not one of the memories in  $Z$ . Instead, the optimization finds hidden variables  $x_2$  that could be used to synthesize a good approximation to  $x_1$ . The optimization (7) is called pattern analysis, because it is the inverse of pattern synthesis.

## 2 Nonlinear network

The network of Figure 1 is very simple: both analysis and synthesis are simple linear transformations. The network of Figure 2 is more complex: the neurons have a rectification nonlinearity, and the network architecture has lateral connections within the hidden layer.

The network dynamics is given by

$$\begin{aligned} \dot{x}_1 + x_1 &= [W_{12}x_2]^+ , \\ \dot{x}_2 + x_2 &= [W_{21}x_1 + W_{22}x_2]^+ . \end{aligned} \quad (8)$$

The only nonlinearity is the rectification  $[x]^+ = \max\{x, 0\}$ . In addition to the top-down and bottom-up connections, there are the lateral connections in the  $n_2 \times n_2$  matrix  $W_{22}$ . Three conditions are placed on the synaptic matrices,

$$W_{12} = W_{21}^T, \quad (9)$$

$$W_{12} \geq 0, \quad (10)$$

$$W_{22} = I - W_{21}W_{12}. \quad (11)$$

The first condition of symmetry between bottom-up and top-down interactions is familiar from before. The second condition is that the connections between the two layers are excitatory, rather than of mixed sign.

The third condition defines a form of lateral interaction called *inner product inhibition*. Each off-diagonal element of  $W_{22}$  is an inhibitory connection strength. The inhibition between two hidden neurons is equal and opposite to the inner product of the corresponding row of  $W_{21}$  and column of  $W_{12}$ . This inner product measures the strength of the indirect excitatory interaction between the two hidden neurons that is mediated by the hidden-visible feedback loop. The diagonal elements of  $W_{22}$  are the strengths of self-feedback for the hidden neurons.

The architecture of interlayer excitation and lateral inhibition lends (8) more biological realism than (1). Furthermore, like the activities of biological neurons,  $x_1$  and  $x_2$  are constrained to be nonnegative, as long as they are initialized nonnegative. More formally, the nonnegative orthant  $x_1 \geq 0, x_2 \geq 0$  is an invariant set of the dynamics (8).

The equations (8) do not constitute a gradient descent dynamics. But it can be shown that

$$E = \frac{1}{2}|x_1 - W_{12}x_2|^2 \quad (12)$$

is nonincreasing in the nonnegative orthant. This follows from differentiation and substitution of the equations of motion. Therefore the nonlinear network optimizes the energy function (12), subject to constraints that will be explained shortly.

**Memory** The  $n_2$ -dimensional linear manifold

$$Z^+ = \{(x_1, x_2) : x_2 \geq 0, x_1 = W_{12}x_2\} \quad (13)$$

consists of fixed points of the dynamics (8). Because of the rectification nonlinearity,  $x_1$  and  $x_2$  are constrained to be nonnegative. However, the constraint  $x_1 \geq 0$  is redundant, given that  $W_{12} \geq 0$  and  $x_2 \geq 0$ , so it has been omitted from the definition of  $Z^+$ . Since the network dynamics (8) minimizes the energy function (12), it must converge to a zero energy state, and therefore the set  $Z^+$  is an attractor of the dynamics.

**Synthesis** As before, the pairs in  $Z^+$  can be regarded as memories that are content-addressable. If  $x_2$  is clamped, then  $x_1$  converges to  $x_1 = W_{12}x_2$ . So pattern synthesis is a linear operation, just as it was for the linear network.

**Analysis** If  $x_1$  is clamped, then the  $x_2$  dynamics converges to

$$\operatorname{argmin}_{x_2 \geq 0} \frac{1}{2} |x_1 - W_{12}x_2|^2 . \quad (14)$$

This constrained optimization problem is known as nonnegative least squares. The visible pattern is approximated by a linear combination of the columns of  $W_{12}$ , as in the analysis computation (7) of the linear network. The difference is that the coefficients of the combination are constrained to be nonnegative.

### 3 Lateral inhibition

In the traditional interpretation of feedforward neural networks, a hidden neuron is regarded as a *feature detector*. Its response is triggered by input patterns that are similar to the pattern in its bottom-up connections. The interpretation of the analysis dynamics in (8) is more complicated, because of the presence of lateral interactions. The analysis computation (14) can be interpreted as a *feature decomposition*. If a pattern is composed of a mixture of features, the pattern analysis decomposes that mixture into its constituents.

With inner product inhibition, hidden neurons for similar features inhibit each other strongly, while hidden neurons for different features inhibit each other weakly. This gradation is critical for feature decomposition. The graded inhibition mediates competitive interactions, but still allows more than one hidden neuron to be active.

This is in contrast to global inhibition, which is uniform in strength. Global inhibition causes so much competition between neurons that winner-take-all behavior tends to result[8], in which just one neuron is active and all the rest are silent. This is a localized representation, quite different from the distributed representation needed in feature decomposition.

In topographic feature maps, the lateral interactions are excitatory at short-range, inhibitory at long-range, and eventually vanish at large distances[9]. Since neurons with similar features are found close together in the map, the gradation of lateral interactions correlates with feature similarity. However, inner product inhibition has a clearer computational interpretation in terms of the cost function (12).

Competitive interactions between hidden variables also arise in probabilistic reasoning[10]. Since hidden variables are “hidden causes” of the visible variables, they have an inhibitory influence on each other. If one cause adequately explains the data, then there is no need to invoke another cause. This competition between “explanations” is known as “explaining away.”

### 4 Rectification nonlinearity

The use of rectification nonlinearity in (8) is unusual; the sigmoid function is more conventional in artificial neural networks. There are two main computational advantages of using rectification. First, the piecewise linear and analog

nature of the network is suitable for representing continuous attractors. Second, the network is invariant to changes in input intensity.

Intensity invariance actually depends on lateral inhibition, as well as rectification. The lateral inhibition sets an activity-dependent threshold, rather than the fixed threshold seen in most artificial neural networks. Neurons with fixed thresholds often lose selectivity with increasing input intensity, a phenomenon known as the “iceberg effect.” The “iceberg effect” is not seen in the network (8). If the intensity is changed, the inactive neurons remain inactive, while the neurons with nonzero activity are scaled up uniformly. This can be seen by verifying that if  $(x_1, x_2)$  is a memory, then so is  $(\lambda x_1, \lambda x_2)$ , for any  $\lambda > 0$ . Similarly, if analysis maps  $x_1 \rightarrow x_2$ , then  $\lambda x_1 \rightarrow \lambda x_2$ . This property is also known as proportional response.

Another motivation for using rectification nonlinearity is biological realism. For neurons in many brain areas, the relationship between firing rate and current is relatively linear above threshold. Although neurons eventually do saturate at high firing rates, they normally operate well below this regime. So saturation nonlinearity may not be so important for normal brain function, which is one reason why rectification nonlinearity was used in (8) instead of the conventional sigmoid.

The intensity invariance seen here is related to the contrast invariance exhibited by recent models of visual cortex[7]. As will be explained in detail elsewhere, proportional response is a general property of neural networks with rectification nonlinearity, as long as the neurons lack fixed thresholds.

## 5 Conclusion

Two networks, linear and nonlinear, have illustrated the use of hidden variables in attractor neural networks. In both networks, the dynamics converges to a continuous attractor, a manifold of fixed points. The global stability of the attractor is shown by construction of an energy function. The points in the attractor can be regarded as stored memories.

Pattern analysis and synthesis are both types of pattern completion by recall of a memory. The analysis computation is a least squares optimization. In the linear network, it is a simple linear transformation performed by the bottom-up connections. In the nonlinear network, it involves lateral inhibitory feedback, in addition to bottom-up input.

This paper has focused on the problem of *representing* hidden variable models as attractor neural networks. The problem of *learning* continuous attractors from examples is discussed elsewhere[5]. The linear network is suitable for learning PCA and its variants. The nonlinear network is suitable for the **Conic** algorithm, in which the hidden variables are constrained to be nonnegative[3].

## Acknowledgments

I am indebted to Dan Lee for extensive discussions on all the topics in this paper. I have also benefited from discussions with L. Saul.

## References

- [1] J.-H. Oh and H. S. Seung. Learning generative models with the up-propagation algorithm. *Adv. Neural Info. Proc. Syst.*, 10, 1998.
- [2] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The “wake-sleep” algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.
- [3] D. D. Lee and H. S. Seung. Unsupervised learning by convex and conic coding. *Adv. Neural Info. Proc. Syst.*, 9, 1997.
- [4] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. USA*, 79:2554–2558, 1982.
- [5] H. S. Seung. Learning continuous attractors in recurrent networks. *Adv. Neural Info. Proc. Syst.*, 10, 1998.
- [6] H. S. Seung. How the brain keeps the eyes still. *Proc. Natl. Acad. Sci. USA*, 93:13339–13344, 1996.
- [7] R. Ben-Yishai, R. L. Bar-Or, and H. Sompolinsky. Theory of orientation tuning in visual cortex. *Proc. Nat. Acad. Sci. USA*, 92:3844–3848, 1995.
- [8] S. Amari and M. A. Arbib. *Competition and cooperation in neural nets*, pages 119–165. Academic Press, New York, 1977.
- [9] T. Kohonen. The self-organizing map. *Proc. IEEE*, 78:1464–1480, 1990.
- [10] G. E. Hinton and Z. Ghahramani. Generative models for discovering sparse distributed representations. *Phil. Trans. Roy. Soc.*, B352:1177–1190, 1997.