ARTICLE Communicated by Terrence Sejnowski

Understanding Emergent Dynamics: Using a Collective Activity Coordinate of a Neural Network to Recognize Time-Varying Patterns

John J. Hopfield
hopfield@princeton.edu
Princeton Neuroscience Institute, Princeton University, Princeton, NJ 08544, U.S.A.

In higher animals, complex and robust behaviors are produced by the microscopic details of large structured ensembles of neurons. I describe how the emergent computational dynamics of a biologically based neural network generates a robust natural solution to the problem of categorizing time-varying stimulus patterns such as spoken words or animal stereotypical behaviors. The recognition of these patterns is made difficult by their substantial variation in cadence and duration. The neural circuit behaviors used are similar to those associated with brain neural integrators. In the larger context described here, this kind of circuit becomes a building block of an entirely different computational algorithm for solving complex problems. While the network behavior is simulated in detail, a collective view is essential to understanding the results. A closed equation of motion for the collective variable describes an algorithm that quantitatively accounts for many aspects of the emergent network computation. The feedback connections and ongoing activity in the network shape the collective dynamics onto a reduced dimensionality manifold of activity space, which defines the algorithm and computation actually performed. The external inputs are weak and are not the dominant drivers of network activity.

1 Introduction

Animal sensory systems commonly recognize familiar patterns that are spread out in time. Humans rapidly recognize a spoken word by hearing it or lip-reading, and they visually recognize a particular friend at a distance by the way he or she walks. Such stimulus patterns will be called dynamic patterns, to distinguish them from static patterns such as photographs. Most natural dynamic patterns are not clocked at some fixed pace, but have substantial variation of both duration and internal cadence. The lack of a fast and effective algorithm to characterize the similarity of two dynamic patterns is one of the reasons that voice-to-text speech processing programs are computation intensive and still do not perform well in natural environments.

Neural Computation **27, 2011–2038** (2015)
doi:10.1162/NECO_a_00768

All devices capable of extended computations (whether human brains or digital computers) are dynamical systems, in which the future state of the system is implicitly determined by the present state and a set of equations of motion (Hopfield, 1994; Mytkowitcz, Diwan, & Bradley, 2009; Dambre, Verstraeten, Schrauwen, & Massar, 2012). Computation makes iterative use of the equations of motion to transform from "input data" to "desired answers." Computations are easy to implement when the equations of motion directly resemble the desired transformation. Higher animals seem to have little difficulty solving dynamic pattern recognition problems very rapidly, even in the presence of considerable background noise. Humans solve them nonconsciously. These facts suggest that the dynamics of neural circuits in a brain are very effective in implementing an approximate solution to this ubiquitous problem.

Large physical systems with many similar components often display phenomena that are accurately described by collective variables. The equations of motion of the collective variables can look entirely different from the equations of motion of the underlying microscopic system. For example, a small water droplet of a fog is stable in size and shape, and moves slowly downward under the influence of gravity and the viscous drag forces of the surrounding air. This collective description of falling drops is utterly unlike the underlying dynamical variables describing the collisions of a large number of water, oxygen, and nitrogen molecules interacting with atomic-scale molecular forces and obeying Newton's laws of motion. When neural collective dynamics has equations of motion that are used in a computation but do not closely resemble the behaviors of individual neurons, it will be necessary to understand the collective variables to understand how the computation is performed. Such collective operation will lend a spontaneous robustness against neuronal failure or circuitry errors.

This article describes how an artificial neural network model, related to models that have been invoked to explain navigational path integration in rats (McNaughton et al., 1996) and for head direction cells (Zhang, 1996; Xie, Hahnloser, & Seung, 2002), can implement a computation of dynamic pattern recognition. The neural network develops its useful computational behavior through the dynamics of a collective variable. A closed-form equation of motion for the collective variable is developed. This allows the collective variable response of the network to a complex time-dependent stimulus to be predicted without the need to evaluate the behavior of individual neurons. Prior use of collective variables is typified in using a response vector computed from the activity of many neurons to predict behavior (Shadlen, Britten, Newsome, & Movshon, 1996). But there has been no way to predict the response vector of a general time-varying stimulus except by knowing the activity of the many cells involved. Seung (private communication, 2015) has pointed out that when the synapse connectivity pattern of N neurons is less than rank N and one particular variant of neural dynamics is chosen, the activity pattern factors into a product space, and

thus can be described in lower dimensions. This mathematical fact depends on details and is quite distinct from the robust emergent dynamics of this article, where the fine details of the intra-area connectivity pattern and of the neural dynamics do not matter. Eliasmith and Anderson (2003) note that from an engineering viewpoint, low-dimensional dynamical manifolds can be built into structured networks of neurons, but their approach does not seem related to the idea of emergent collective variables in large physical systems.

The principles of operation will be illustrated by designing a network for the practical problem of recognizing spoken words. Using examples of a problem avoids the necessity of statistically characterizing the natural variations of real-world dynamic patterns, otherwise a daunting task. Speech has been chosen for a demonstration example because it is easy to use real data and known to be difficult, and one can directly experience the cadence diversity of the patterns being classified. The collective neural network approach to this ubiquitous problem does not require implicit serial-to-parallel conversion of the data stream, discrete feature detection, segmentation, long cellular time constants, or temporary storage of detailed intermediate computation. The interesting contrasts between the ideas presented here and earlier neural approaches to this same computational problem (Gutig & Sompolinsky, 2006, 2009; Hopfield & Brody, 2000, 2001) are presented in the discussion.

2 Network Model and Computing Paradigm

To do dynamical pattern recognition requires a reliable initial activity state of the network at the unknown time when the pattern begins. This is most easily achieved by making this the state to which the network relaxes in the absence of sensory input. The initial state is most often chosen to be a state of no neural activity. In the model presented in this article, this is not the case. There is substantial activity in the absence of a sensory input. The total activity of the network changes very little during the computational process; the network activity is redistributed by the sensory input, not created by it. (This can also be true for networks in chaos or poised near criticality (Maass, Natschlaeger, & Markram, 2002; Barak, Sussillo, Roma, Tsodyks, & Abbott, 2013), neither of which is the case in this article.) As in conventional network modeling, recognition of a pattern is indicated by the simultaneous activity of a set of neurons that would otherwise not be simultaneously active.

The computation is carried out by a network of N neurons whose intrinsic feedback connections implement a one-dimensional bump attractor (Wu, Hamaguchi, & Amari, 2008) that is biased. In brief, in a bump attractor, the neurons are (in concept) arranged in a line. In steady state, there is a stereotyped bump of activity centered somewhere along the line of neurons, and the bump can be located anywhere along the line. The bump location

is a single collective coordinate whose value determines the activity of each neuron. The N-dimensional activity state of the network in steady state is thus restricted to a one-dimensional manifold (a curved line) in the N-dimensional activity space. By inserting a small bias (described in further detail below) into the conventional bump attractor network, the bump can be made to slowly drift to the left-hand end of the line, providing a unique standard initial state for the network in the absence of input signals.

Except for Figure 7, all results in this article are from a detailed neuron dynamics on N neurons that does not itself contain the notion of a bump or translation invariance. However, our ability to conceptually plan a network and qualitatively understand what might result from simulations is entirely based on insights into likely bump behavior. While an idealized bump attractor on a line of neurons is often described in mathematics using a ring of neurons connected in a translation-invariant fashion, the simulations carried out here are done on a finite line of neurons. The dominant effect of opening the ring to become a line of length N is merely to restrict the possible locations of the center of the bump to lie in the region between the ends (minus the bump half-width).

Synapses from sensory signal axons to the network neurons allow the time-dependent sensory signal to influence the bump. Amari (1977) first explored the existence of a one-dimensional bump dynamics with time-independent inputs using a field theory on a line of neurons and perturbation theory. Because the neural circuit he described was more general and had no underlying Lyapunov function, his analysis needed to be far more detailed, a tour de force of neural dynamical mathematics.

The sensory signal dynamical pattern will be said to be recognized by the network if and only if the sensory signal succeeds in moving the bump from its initial location to the right-hand end of the line attractor; that is, the neurons near the right-hand end of the line become simultaneously active. All elements of this paradigm are elementary except for the design of the connections from the signal to the computing network. This is an entirely novel design problem, and there is no guarantee that any set of connections could transport the bump to the other end for correct input dynamical patterns but not for incorrect ones. This section describes the detailed network model and the synapses used to make a biased bump attractor. The more difficult problem of the signal-to-network connections is described in the following section.

All modeling uses a rate-based description of spiking neurons. The rate of firing of neuron k is $f(i_k)$ where i_k is the total input current to neuron k and $f(i)$ is the input-output relation of all neurons. The details of spike trains could have been included, but since the modeling does not involve action potential timing per se, this would add an unnecessary complication. The fundamental insight is gained from the mathematics of a rate-based description. All synapses have a conductance that rises instantaneously at the time of an implicit presynaptic action potential and then decays

exponentially with a time constant τ. The synaptic current i_n into network cell n from other network cells k and from signal input axons m thus obeys

$$\frac{di_n}{dt} = -\frac{i_n}{\tau} + \Sigma_k T_{nk} f(i_k) + \Sigma_m W_{nm} S_m(t), \tag{2.1}$$

where T_{nk} is the strength and sign of the synaptic connection from cell k to cell n and W_{nm} describes the synaptic connections from signal input line m, which is driven by an input neuron with firing rate $S_m(t)$.

A bump attractor in one dimension is built on a line of neurons. A local bump of activity is kept stable by having positive feedback at short distances to keep the bump turned on and longer-range inhibitory interactions to keep the bump from spreading. In the head direction bump attractor model, the detailed inhibitory structure is important to driving the system with a vestibular signal, and including explicit inhibitory neurons is important (Zhang, 1996). For present purposes, allowing both excitatory and inhibitory synapses from a single neuron simplifies both the mathematics and the simulation while losing nothing essential to function. Thus, our processing network contains only a single kind of neurons capable of both excitatory and inhibitory outputs

The matrix T for an unbiased bump was defined except for a scale factor by

$$T_{nn} = 0$$
$$\text{for } n \neq k \text{ and } |n-k| < p \quad T_{nk} = 1 - \varepsilon$$
$$\text{for } |n-k| \geq p \quad T_{nk} = -\varepsilon,$$

where ε is a positive number representing global inhibition and p determines the range of the excitatory interactions and the bump width. Figure 1 illustrates T, the gain function $f(i)$ synapse current in response to an implicit action potential, and stable bump shape typical of the simulations. The connection pattern is symmetric, so the convergence to a stable state is guaranteed by the dynamics of equation 2.1 if $\mathbf{S}$ is does not depend on time. The bump is equivalently stable at any location along the line of neurons for $\boldsymbol{S} = 0$. The exact shape of the bump is not conceptually important, so a very simple connection matrix T_{nk} has been used even though it produces a crude and doubtless nonoptimal form of the bump.

Weak perturbations of this system can result in the bump moving as an almost rigid object, preserving the integrity of the bump. For example, if a weak gradient of input currents along the chain is added, the bump will move with a velocity proportional to that gradient (see the appendix). Or if the excitatory connections are made not quite symmetric by making the

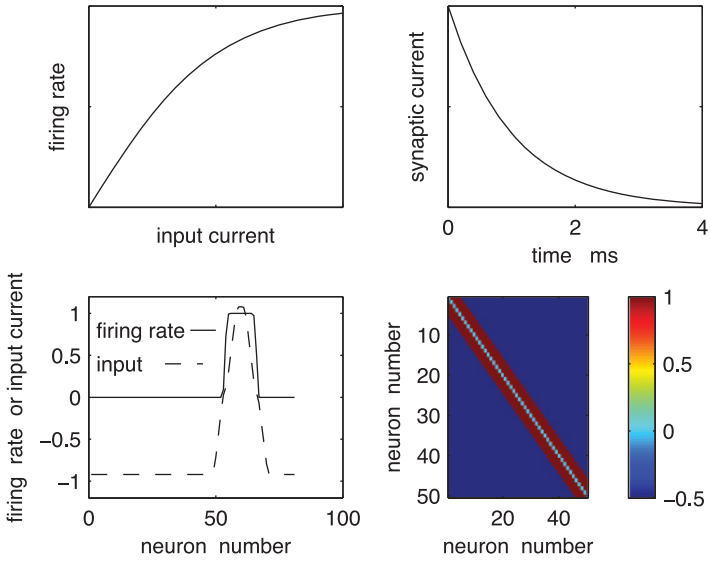

Figure 1: (a) Neuron firing rate $f(i)$ as a function of input current i. (b) Synaptic current pulse due to an implicit presynaptic action potential. (c) Typical stable bump shape: dashed line input current, solid line firing rate. (d) Intranetwork synaptic connection pattern.

connections

$$\text{for } n \neq k \text{ and } n - k < p \quad T_{nk} = 1 + \delta - \varepsilon$$
$$\text{for } n \neq k \text{ and } k - n < p \quad T_{nk} = 1 - \delta - \varepsilon$$
$$\text{for } |n - k| \geq p \quad T_{nk} = -\varepsilon,$$

for small values of δ, the bump will move along the line with a velocity proportional to δ. A δ in the range 0.01 to 0.02 was used to bias the attractor and provide a unique initial state with the bump at the beginning of the chain of neurons. A full simulation of the neural equations of motion is carried out, so the actual results of the simulation do not involve a weak perturbation approximation. However, having the intuition that the bump is a rigid object that can respond to forces due to input current gradients and synapse asymmetry creates an understanding of the computational process.

The network cells n also receive input synapses from neurons m whose firing rates $S_m(t)$ are determined by the power of the auditory sound signal within frequency bands m. The synapses between these sensory cells and the processing neurons are denoted by W_{nm} and will, for conceptual and computational simplicity, have the same synaptic time constant τ. In the overall equations of motion, equation 2.1, the first summation describes the effect of the network's internal synapses that (alone) would result in a

simple biased bump attractor. The second summation describes the currents due to synapses from cells responding to external time-dependent sensory information.

3 Network Sensory Inputs from Speech

The spoken digits zero, one, two, . . . , nine have been used as the dynamic patterns to be recognized, using recordings available from the Texas Instruments speaker-independent connected-digit database (Garofolo et al., 1990). Early auditory centers in mammals have a tonotopic map with each neuron tuned to a narrow frequency band. These frequency-specific responses are modeled by a filtering that projects an acoustic waveform into a set of 20 frequency bands covering the range from 200 to 5000 Hz, spread uniformly on the mel scale (approximately logarithmic in frequency) used in auditory psychophysics. These 20 intensity-within-frequency-band signals were low-pass-filtered to remove responses above 10 Hz. Channel intensity was normalized so that all channels had similar maximal responses to speech. The logarithm of the power in the filtered signal (above a threshold) was taken, in keeping with the logarithmic response characteristics of vertebrate auditory systems. This signal processing is a surrogate for some of the processing of early auditory centers in the brain. The net processing result is a 20-dimensional time-dependent vector $S^{20}(t)$, where the superscript 20 is a reminder of the dimensionality of the vector. Typical speech spectrogram displays of $S^{20}(t)$ are shown in Figure 2.[1]

Other methods of generating sonograms should work equally well. Much useful information about the word spoken has been thrown away in this processing, but easily recognizable speech can be reconstructed from $S^{20}(t)$. Most engineering word recognition programs begin with a similar representation of the speech signal, generally computed by using a fast Fourier transform of the temporally windowed speech waveform.

Early auditory areas in mammals have, in addition, cells that respond to changes in the auditory signal with time, such as tonal sweeps or a rapid onset or offset of sound power in a frequency channel. The existence of such features is already implied by the power sonogram itself. Presumably the advantage of this explicit redundancy is to emphasize behaviorally significant features and add sparseness to the total representation. We will similarly use an overcomplete representation of the information in a sonogram (but without extracting categorical features such as onset time), combining $S^{20}(t)$ with its time derivative $\lambda dS^{20}(t)/dt$ to provide a 40-dimensional analog vector $S(t)$, where λ is a scaling constant chosen to create a similar weighting to the two kinds of inputs. Use of both together is analogous to the use of both cepstral coefficients and their time derivatives in hidden

[1] Relevant brief sound files for this and other figures are provided in the online supplement available at http://www.mitpressjournals.org/doi/suppl/10.1162/NECO_a_00768.

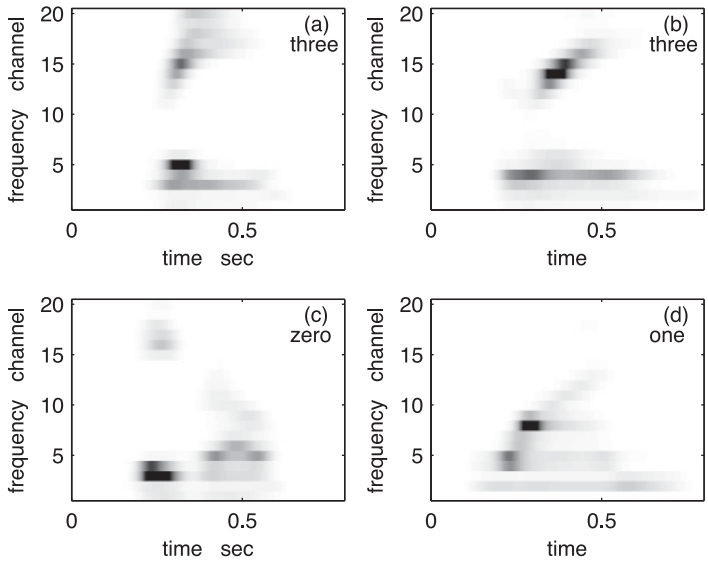

Figure 2: $S^{20}(t)$ displayed as a sonogram for examples of spoken words *three* (a,b) (soundfile1 and soundfile2) , *zero* (c) (soundfile3), and *one* (d) (soundfile4). (a,c) From a single female speaker. Their manifest differences are chiefly due to different words being spoken. (b) From another female speakers saying *three*. Both panels a and b are to be recognized as *three* in spite of the difference in cadence.

Markov models (HMM) for speech recognition, a very effective empirical procedure. High-performance word recognition systems sometimes make use of more complex transforms of the inputs (Furui, 1986).

4 Designing Synaptic Connections W_{nm} for Recognizing a Word

The principles of network function can often be more clearly seen by suppressing some of the details necessary to biological neural circuits. For that reason, our implementation of a bump attractor ignored Dale's law and allowed a neuron to connect to another with either an excitatory or an inhibitory synapse. We make a similar simplification here, allowing any synapse W_{nm} to be excitatory or inhibitory. In addition, while the components of $S^{20}(t)$ are all positive, the components of $\lambda dS^{20}(t)/dt$ can have either sign. Since firing rates are positive, a biological system carrying this signal must carry it along two pathways in parallel—one for positive and one for negative values. In constructing W_{nm}, we will ignore that fact and implicitly allow negative firing rates, realizing that using a separate pathway with inhibitory connections can make an equivalent implementation in biology using only positive firing rates.

I will develop an understanding of how such a system can work by using one particular example of a digit as a template for designing the connections W. Learning the best possible set of connections for word recognition using a large database is a separate research problem. An utterance of the word *three* is used in the illustration. This utterance had a duration of 480 msec with appreciable signal power and can be described as a 40-dimensional time-dependent signal vector $\boldsymbol{Stemplate}(t)$ of duration 480 msec. Construct a one-dimensional bump attractor L neurons long. Pick a set of L times t_k, $k = 0$ to $(L-1)$ in order along the template. Uniform spacing is convenient but not optimal. In this example, the time points were chosen uniformly 8 msec apart, and $L = 61$. The normalized weights to the interior neurons for this template are chosen to be

$$W_{kn} = Stemplate(t_k)_n / (\boldsymbol{Stemplate}(t_k) * \boldsymbol{Stemplate}(t_k)), \tag{4.1}$$

where the dot product normalizes the weights. For this normalization, the last term in equation 2.1 obeys

$$\Sigma_m W_{nm} S_m(t) = 1 \text{ if the signal } \boldsymbol{S}(t) = \boldsymbol{Stemplate}(t), t = t_k \textit{ and } n = k.$$

This means that if the signal is the actual template, neuron k has a sensory input near 1 for times near $t = t_k$. A common multiplicative normalization factor not relevant to the pattern of weights has been omitted for clarity. The discussion of why these connections have the desired effect on the bump motion is in the next section, where simulation results lead to an intuitive understanding of the successful network response selectivity.

Two minor technical points should be noted. First, in order to be able to avoid line-end distortions, the bump attractor line of neurons with synapses W is extended to the left by including a few neurons without W connections, thus allowing an initial condition of the bump centered just before the beginning of the W connections. Second, noise is included in the simulations. The weak bias in the bump attractor network suppresses diffusional drift, preventing noise alone from causing the bump to move very far. Neither of these technical points is essential to understanding the computing principle behind the operation of the network.

5 Results from Network Simulations

When there is no input signal, the bias in the network positions the bump at the left-hand end of the line of neurons, where the bump is stable. An input signal can cause the bump to move. The utterance is said to have been recognized as an example of the template word if it transports the bump to the region of neuron 60 near the end of the signal. (After the signal ends,

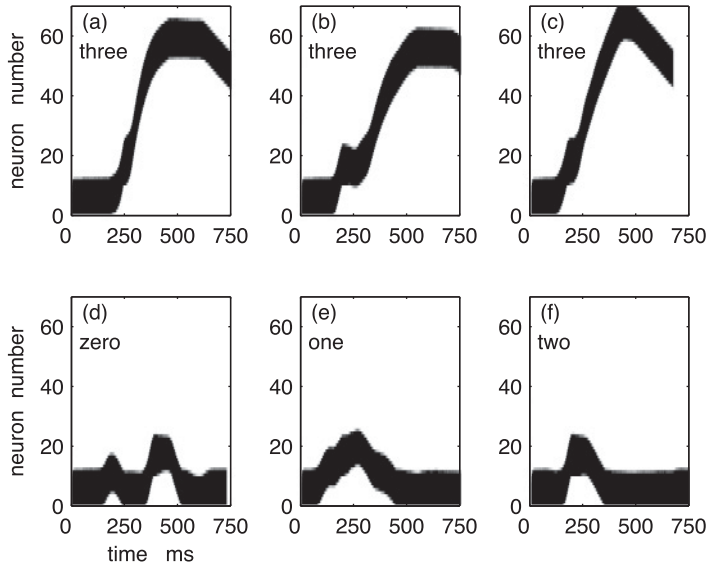

Figure 3: The response (gray scale) as a function of time of 70 neurons arranged in a line attractor with signal input connections W designed to recognize a spoken *three* for neurons 10 to 70 and no signal input to neurons 1 to 9. The initial state is a bump in the 0 to 12 region. Sound input in each case begins somewhere in the 50 to 250 msec region and ends 150 to 400 msec later, depending on the utterance. In each horizontal row, the activity of one neuron is plotted, with white being inactive and black being strongly active. (a) Response to the template *three* from which the connections W were designed (soundfile1). (b,c) Responses to *three* spoken by two other speakers (soundfile2 and soundfile5). (d–f) Responses to spoken *zero* (soundfile3), *one* (soundfile4), and *two* (soundfile6).

the bump will again drift back toward the end where it started due to the asymmetry into the connections.)

Figures 3a to 3f show responses of all 70 neurons with connections designed for a *three* detector, obtained by integrating the equations of motion, equation 2.1. In all cases and at all times, the pattern of activity is a localized bump about 12 neurons wide. The effect of a weak input is merely to make the bump move, with little distortion.

When the input signal is a spoken *three*, the bump moves to the vicinity of neuron 60 or beyond. Figures 3a to 3c are three different female voices, with a duration of the slowest spoken *three* more than twice that of the fastest. When the input is from other spoken digits, the center of the bump remains below neuron 20. While Figure 3 is only an anecdote, it is sufficient to illustrate that this structure of network has interesting computing potential.

Networks were designed for *zero*, *one*, *two*, and *three*, choosing as a template a single example from a single speaker and tested on five female speakers, each saying the digits *zero*, *one*, . . . , *nine*. For *zero*, *one*, and *three*, four of five correct utterances were identified, with no false positives on the 45 incorrect examples. The network for detecting *three* recognized equally well the rapid *three* of Figure 2a and the slow *three* of Figure 2b, in spite of the extreme difference in length and cadence of these two files. The template-based network for the short utterance *two* was less effective, with many false-positive errors. There is not enough structure (using this spectral representation) in this short pattern over time to distinguish the template-based model of *two* from short fragments of longer words.

6 Understanding the Computational Principle

When there is no input signal and the bias parameter δ is set to zero, the dynamical equations for this network have the Lyapunov function (Hopfield, 1984)

$$L_o = -\frac{1}{2}\Sigma_k \Sigma_n T_{nk} f(i_k) f(i_n) + \frac{1}{\tau}\Sigma_k \int_0^{f(i_k)} f^{-1}(x)dx. \tag{6.1}$$

The network state without input evolves in time until it reaches a local minimum of this function. The connections have been designed so that the stable states are a broad stereotype bump of activity centered anywhere along the line of neurons, so this function has the same value for any bump location.

If an input signal I_k that is constant in time is added to each neuron k, the network dynamics is then controlled by the Lyapunov function:

$$L = L_o - \Sigma_k f(i_k) I_k. \tag{6.2}$$

For nonzero (and unequal) inputs, the position equivalence is broken, in general causing the bump-like state to shift and distort. If the feedback is strong and the inputs I_k are weak, the bump will move with little distortion. L_o is independent of bump location, so the second term in equation 6.2 controls the slow bump motion.

It is intuitively useful to regard the bump shape and the input vector as functions of a continuous position variable x. For zero input, the stable bump shape is $B(x - x_b)$, where x_b is the bump center location and can take on any value along the continuous line attractor. Similarly, the total signal input current I_k to neuron k can be thought of as a continuous variable $I(x)$. Since the bump moves with little distortion, the second term in equation 6.2 is a bump-shape-weighted sum of the input. If the input I_k varies slowly

with k, this term is proportional to $-I(x_b)$. The bump will move until it rests at local minimum of a terrain whose height is $(-I(x_b))$. Sign conventions are arbitrary, and I have chosen the signs such that going downhill is the natural spontaneous behavior.

Let fixed input currents be chosen like the signal inputs to the network for the template utterance at some fixed time t_p:

$$I_n = \Sigma_m W_{nm} S_m(t_p).$$

Since the synapses to neuron k have been chosen (see equation 4.1) to make neuron k maximally responsive to the signal at time t_k, we can expect that there will be a minimum (perhaps a local minimum) in $-I_n$ for a value of n near p, when t_p is the chosen time. In the operational system, I_n will slowly change in time because the natural S is time dependent. As time progresses, and now the real time is t_{p+1}, the valley will have moved over to a value of n near $p+1$. The dynamics will thus be characterized by a bump that is moving downhill in a one-dimensional terrain containing a moving valley. The bump can be captured by that valley and will then follow as the location of the valley slowly shifts. When the input is due to the template utterance, this valley should move smoothly from one end of the line of neurons to the other end during the utterance.

Figure 4a shows the network activity as a function of time when the connections are for a template of *three* and the template utterance was used for the signal input. Figure 4e, below panel a, shows the input I_k to each of the neurons as a function of time during this utterance. At any time, the input pattern I_k as a function of neuron number k has a single peak, whose center moves smoothly from neuron 10 (when the utterance begins) to the vicinity of neuron 65 when the utterance ends. When the utterance begins, the terrain $-I_k$ thus has a local valley (local minimum) near $k = 10$. (Neurons 1 to 9 are for end padding and have no signal inputs.) This valley moves steadily along the line of neurons, reaching a value somewhere in the 60 to 70 range at the end of the spoken word. The bump is trapped in the minimum and is dragged along, lagging a bit behind, while the location of that minimum changes from one end of the line of neurons to the other end over the course of the spoken word.

Figure 4b and its corresponding panel, Figure 4f below it, are the same kind of presentation for a *three* spoken by a different speaker. The upper panel displays the same transport of activity to the 60 region at the end of the utterance. The duration of this particular spoken word is longer than the template, and the time it takes for the bump to move to the 60 region is correspondingly longer. The panel below it shows a moving valley in $-I_k$. This valley is not so well defined or simple as in the case when the utterance tested was the template itself, but it still drags the bump from one end to the other.

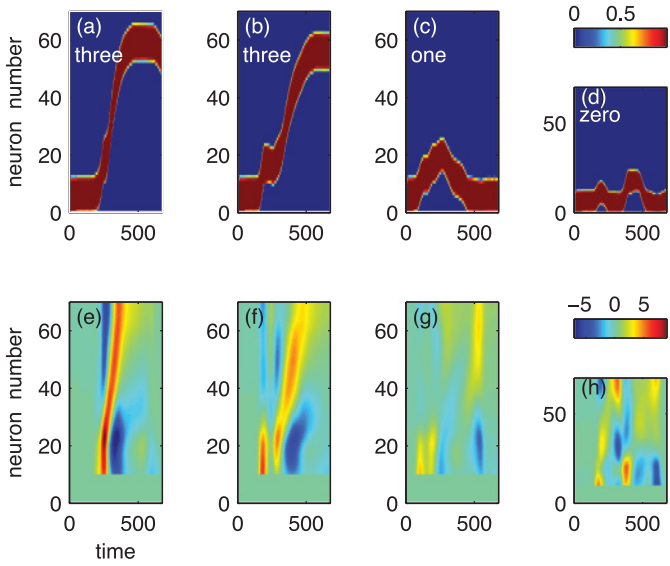

Figure 4: The pattern of neuron activity and the signal inputs as a function of time (in milliseconds) for the 70 neurons in the line attractor network designed to respond to *three*. Time zero is the beginning of the speech file, which has 50 to 300 msec of silence before the sound of the word begins. (a) Response to the template *three* (soundfile1). (b) Response to another *three* (soundfile2). (c) Response to *one* (soundfile4) . (d) Response to *zero* (soundfile3), displayed as in Figure 3. The corresponding panels (e–h) display in false color the pattern of signal input current as a function of time to these neurons via the connections W. Neurons 1 to 9 have no synapses W and receive no signal input. The sign convention makes red indicate a terrain valley and blue a terrain peak.

By contrast, Figures 4c and 4d are similar plots for connections based on a *three* template, but when the spoken examples are *one* and *zero*. In neither case is the activity bump transported to the neuron 60 region, so neither is recognized as being an example of *three*. In both cases, the lower panel shows why. There is no organized moving valley structure to drag the bump effectively, and while the bump moves, the disorganized time-dependent terrain does not move it very far.

Figure 5 shows the inputs to typical neurons in the network as a function of time. At most times, the total input current is an order of magnitude greater than current I_k from the sensory input, so the feedback currents dominate the activity of neurons. There are, however, a few short time intervals when the magnitude of the feedback current is almost zero and the sensory signal is particularly influential, as in the region near 230 msec in Figure 5a or 370 msec in Figure 5b.

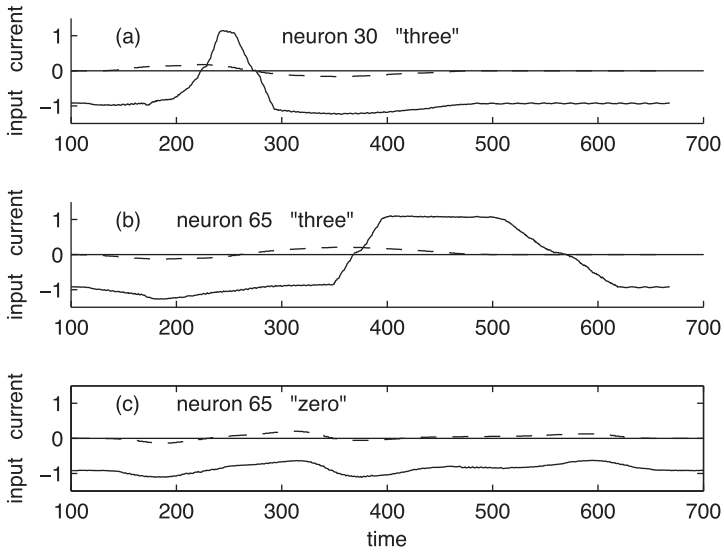

Figure 5: Components of the input current to typical network neurons as a function of time (in msec). Dashed line: input I_k directly from the sensory signal via the synapses W. Solid line: total input, including feedback currents from the excitatory internal connections and the global inhibition. The synapses W are designed to recognize *three*. (a) Neuron 30, input from *three*. (b) Same as panel a but for neuron 65. (c) Same as panel b, but for input from spoken *zero*. Because in this case the bump remains near neuron 10, the feedback to neuron 65 is strongly inhibitory for all times.

The importance of having only weak connections from the signal can be seen by looking at the response of the *three* detector to the utterance *zero*. By design, neurons having numbers in the range 55 to 70 have a broadly selective sensitivity to signal inputs that sound like *ee*, the end of *three*. *Zero* has a strong *ee* sound near its beginning. Why don't the 55 to 70 neurons respond near the beginning of *zero*, since figure 4h shows that at time 300 msec, these neurons are the ones most strongly driven by the signal input? The answer lies in the fact that the signal input is not strong enough to override the network-based feedback current inhibiting these neurons, as can be seen in Figure 5c. The only way these *ee* neurons can be made active by weak signal inputs is by moving the existing bump of activity to their location through a coordinated signal over time.

The intuition of moving downhill on a time-varying energy function is not quite right when δ is not zero and the connection matrix is not symmetric. There is then a second source of force on the bump. If the line of neurons has open ends, this force due to asymmetry is equivalent to a force

generated by an additional term in the energy function, linear in position along the line of neurons and proportional to δ. Such a transformation cannot be made if the line of neurons is closed into a ring, as might be done in order to produce an automatic reset capability. For reset, if a word is recognized, the bump is rapidly transported to the starting position at the beginning of the chain of neurons. Connecting the neurons into a ring, with the ring-closure region having a strong forward bias asymmetry, achieves this result without altering the rest of the network dynamics.

7 Tempo Insensitivity

The basic reason for insensitivity to time duration in recognizing words can be thought about in terms of ability to deal with time warp. The term $I_n(t) = \Sigma_m W_{nm} S_m(t)$ in equation 2.1 is been designed to produce a moving valley structure in the $n - t$ plane, and the dynamics locks a bump of activity into that valley and drags it along as time progresses. Consider the response of the system to a time-scaled version of S_m, $S_m(\alpha t)$. The moving valley structure of $I_n(t)$ will replaced by $I_n(\alpha t)$ and is unchanged, except that the time axis will be scaled by α. Thus, the computational result—bump dragged from one end to the other (correct recognition) or not dragged to the other end (correct treatment of a 'wrong' input stream)—should be the same for $S_m(\alpha t)$ and for $S_m(t)$. Indeed, the same argument can be made if the scaling constant α is replaced by a time-dependent positive scaling function $\alpha(t)$. If differences between two utterances of the same word can be ascribed purely to a time warp function, then these two utterances should be identically categorized. This line of argument breaks down if α is so large that the bump cannot be dragged sufficiently fast by the available forces.

However, some sensory signals we have used are time derivatives, and while there is an underlying process $P(t)$, the sensory signals are proportional to dP/dt. In this case, when the underlying physical process $P(t)$ is replaced by $P(\alpha t)$, the terrain $I_n(t)$ will replaced by $\alpha I_n(\alpha t)$. The terrain becomes scaled in height by α, and thus should produce a similar bump motion. In this case, the force and the necessary slew rate both scale with α.

The signal vector S contains two sets of 20 components, one of which was a direct signal and the other a derivative signal. Either alone can be argued as above to produce good time-warp invariance. When both are used together, there is no simple scaling relationship. Rapid and slow speech have systematic differences not accounted for merely by local time dilation, so the general line of argument can provide only intuitive guidance.

8 Interference Rejection

In many environments the voices of other speakers are present as background. Humans do an amazing job of ignoring such a background. Since at any time such background may dominate some frequency bands of the

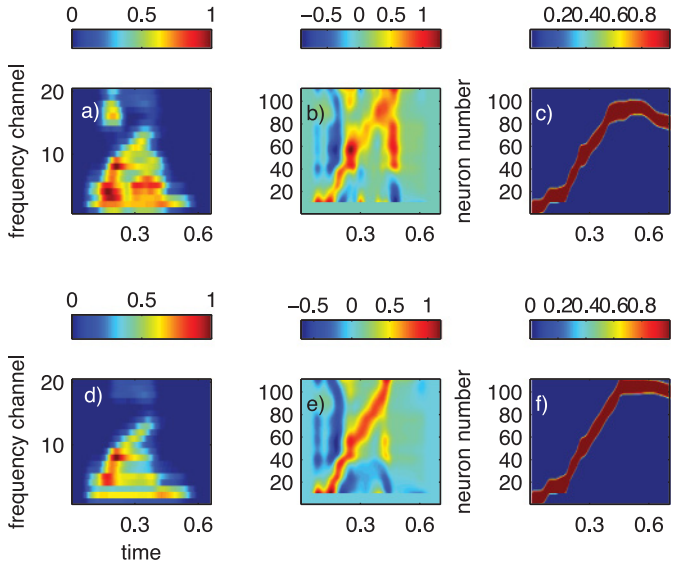

Figure 6: (a) The sonogram produced by the simultaneous presence of *one* and *zero* (soumdfile7). (b) The inputs to a network designed to recognize *one* for this simultaneous sound file. (c) Response of the network designed to recognize *one* to the simultaneous presentation of *zero* and *one*. (d–f) Like panels a to c except the input sound is only *one*. The template for *one* is of longer duration than the template for *three* in Figure 3, and correspondingly more neurons were used.

sound stream, background rejection requires an ability to integrate information over time and across information channels. To examine whether the computational network presented here is likely to have good background rejection properties, we tested a network designed to respond to *one* on inputs of two digits simultaneously spoken at equal loudness. The recognition network identified this sound file as *one* and did not recognize other combined sound files not containing *one* or any of the other spoken single digits. Figure 6c shows the response of this network to this speech file, moving the bump from one end of the line of neurons to the other end. The signal input to the network in Figure 6b shows a corrupted valley leading from neurons in the 0 to 15 range at the beginning of the utterance to neurons in the 105 to 120 range near the end of the utterance. Although reduced in depth and continuity compared to the valleys of Figure 6e, this weak valley was sufficient to drag the bump from one end of the line of neurons to the other. A network designed to recognize *three* showed a similar ability to work with simultaneous files.

This interference rejection is possible because in the design of the network for recognizing *one*, only features present in *one* were used. The network

did not require negative design (or learning) to not respond to other words. Thus the presence of a feature in *zero* that is not in *one* is not ipso facto evidence against recognizing *one* in the combined file.

9 The Collective Coordinate Equation of Motion: Successes and Limitations

In the absence of inputs and synaptic asymmetry, a stable bump can be centered on any neuron n. The bump shape can be described in terms of the input currents to the neurons k along the chain. Since all possible bump center locations n are equivalent, this shape depends only on $k - n$ and will be denoted by i^o_{k-n}. The shape is illustrated in Figure 1c. This current has an asymptotic value i_{asym} for neurons far from the center of the bump, and it is convenient to define a bump shape K_s so that when the bump is centered at n,

$$i^o_k = i_{asym} + K_{k-n}. \tag{9.1}$$

Thus, K_m is a description of the bump shape in input space, centered at an index zero. The bump has reflection symmetry: $K_m = K_{-m}$. K_m is defined only for integer values of m, but because the bump comprises many neurons and is smooth in m, a function $K(s)$ can be defined, where for s any integer, $K(s) = K_s$. For s not an integer, the value of $K(s)$ is obtained by linear interpolation from the nearest integer. Let y be a continuous bump center location variable. Bump shapes and positions are now described as $K(x - y)$, the rigid bump approximation.

N is the number of neurons in the line attractor, and n is the half-width of the bump. y will be restricted to the range $n < y < N - n$. Within this range and within the approximation that bump is rigid object, an equation of motion for y can be written (see the appendix):

$$\frac{dy}{dt} = -\upsilon + \lambda \sum_n \{K(y - n + 1) - K(y - n)\} \Sigma_m W_{nm} S_m(t). \tag{9.2}$$

The term in braces is the finite difference version of a derivative. When the synapses are smooth in the variable n, the quantity

$$I_n = \Sigma_m W_{nm} S_m(t) \tag{9.3}$$

can be generalized on n to a continuous but time-dependent variable $I(y,t)$. Replacing the finite difference by a derivative yields

$$\frac{dy}{dt} = -\upsilon + \lambda d/dy \int K(x) I(y - x, t) dx, \tag{9.4}$$

where υ is a constant proportional to the asymmetry parameter δ and λ is a normalization constant. The second term in equation 9.2 can be intuitively understood through equation 9.4 as the derivative with respect to y (or n) of a smoothed (on y or n) version of the terrain $\Sigma_m W_{nm} S_m(t)$.

Equation 9.2 is a closed-form approximate equation of motion for the center of the bump, a collective coordinate. Using it, bump motion can be found without calculating the detailed behavior of the activity of N neurons and by solving one differential equation instead of N.

The motion of the bump can be rigorously computed by solving the N neuron equations of motion, equation 2.1, and using the resultant activities to evaluate the center of gravity of the activity by

$$y(t) = \Sigma_k k f(i_k) / \Sigma_k f(i_k), \tag{9.5}$$

Figure 7 compares the motion of the collective variable $y(t)$ as computed from equations 9.2 and 9.3 with that obtained by solving the underlying detailed neuron equations of motion, equation 2.1, and then computing $y(t)$ directly from equation 9.5. The collective variable equation of motion clearly describes the bump location generated from the full neural activity equations in great detail.

The collective variable equations of motion, equations 9.2 to 9.4, embody a new computer algorithm for dealing with dynamical pattern recognition. It can be used to design or learn synaptic weights that could be more effective than the single-template synaptic weights used in this article. However, the full equations of motion also contain effects and possibilities not present in the simple collective variable equation of motion:

1. If $\Sigma_m W_{nm} S_m(t)$ is too large, the bump can die at one location, with another bump be simultaneously born at a quite different location when the neuronal equations of motion are solved. Such jumps are not contained in equations 9.2 to 9.4, and their elimination may be helpful for the purpose of applications.
2. Y-shaped line attractors can be easily constructed using the full neuronal equations of motion. A bump can then be made to move up along the stem of the Y and then take the right- or left-hand branch according to the input when the junction is reached. In this way, a branched detector can be built, perhaps useful in building recognizers when there are shared initial dynamical segments, as in *wonder* and *oneself*. Equations 9.2 to 9.4 do not intrinsically contain such a possibility.
3. When the number of neurons is large, synaptic connections T can be build in such a way that the N neurons contain many line segment attractors. Each neuron may belong to several such lines, just as a given hippocampal cell has place fields in multiple environments (Tsodyks & Sejnowski, 1995; Tsodyks, 1999; Monasson & Rosay, 2013).

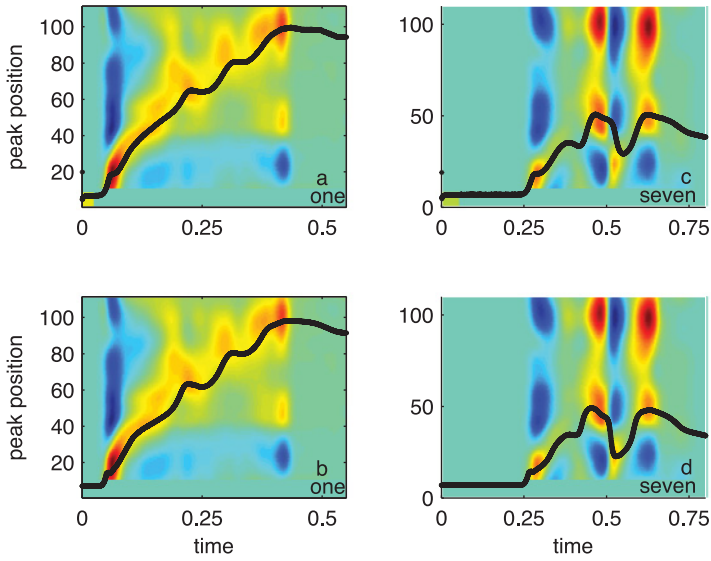

Figure 7: Comparisons of $y(t)$ as evaluated from the full neural equations of motion (a,c) and from the equations of motion of the collective variable (b,d). The weights W were designed from a relatively short example of *one*. The line attractor had 110 neurons. Panels a and b show the response to the very long *one* of Figure 2d and represent recognition by taking y from its initial (and smallest possible) value of 8 to a final value very close to the maximum possible value. Panels c and d show the response to *seven*, for which y remains below 50, indicating nonrecognition. The background false color is a map of $\Sigma_m W_{nm} S_m(t)$. The equations of motion, equations 9.2 to 9.4, contain the spatial derivative with respect to the neuron position coordinate of a slightly smoothed version of this map. The similarity of the terrain of panel a near 0.4 s and panel c near 0.6 s is due to the similarity of the *nn* sound at the end of *one* and of *seven*.

> Mind-set or expectation is then easily expressed through multistability. This greatly increases the flexibility and usefulness of the network.

Equations 9.2 to 9.4 have two implicit scales that must be appropriately chosen for good performance. First, the scale of W controls how rapidly y can follow a moving valley. If it is too small, even a well-shaped valley will not be followed. If it is too large and the valley structure is somewhat confused and noisy, this permits the system to follow sharp noise structure and make unreasonable transits in its $y(t)$ trajectory. The size of υ sets an effective threshold for following a badly structured valley by representing an opposing force to keep y near zero. It serves as a stringency control. If $\upsilon = 0$, even a purely noisy landscape will eventually drive y to a value representing recognition.

10 Discussion

Robust emergent behaviors of large systems (whether physical or biological) are a logical consequence of the behaviors and interactions of the more microscopic elements making up the large system. One of the biggest challenges to neuroscience is to understand the causal relationship between different levels—for example, how language acquisition is a consequence of the structure of a brain and the dynamical properties of its neurons and synapses. Multiple intermediate levels of emergence may be required to bridge such a large gap. Emergence is fully understood when the equations or principles that are involved in a higher-level description have been derived from a lower-level description and are self-contained without further reference to the lower system. For example, the laws of hydrodynamics used to design airplane wings have been derived from the underlying physics of colliding molecules, are mathematically self-contained, describe air as a fluid, and have no reference to the underlying physics of molecular collisions.

Equations 9.2 to 9.4 describe the emergent equation of motion of a dynamical pattern recognition network. It was derived from the dynamical activity of a network of neurons but has a completely different mathematical structure from the underlying individual neuron dynamics and does not contain neuronal activity variables. Nonetheless, it encapsulates an implicit collective algorithm describing dynamical pattern recognition by the underlying network, an algorithm in which the activity of individual neurons has disappeared. While we tested the capabilities of the network by a full neuronal simulation, Figure 7 shows how completely the collective description captures what the underlying network is capable of and allows us to understand how it recognizes dynamical patterns.

Section 1 noted that "computations are easy to implement [in hardware or neurons] when the equations of motion [of the computing machine] directly resemble the desired transformation." I can now give explicit substance to this statement. The problem of dynamical pattern recognition can be approximately solved by using an HMM and implementing this algorithm on a general-purpose digital machine, with a synchronizing clock and huge number of very fast transistors doing binary logic. There is no direct relationship between the low-level computer hardware dynamical system and the HMM algorithm. Indeed, the very absence of such a relationship is what makes general-purpose digital machines useful. No one is likely to propose a neural network to implement HMM or that our brains make use of the HMM algorithm. In contrast, equations 9.2 to 9.4 implicitly describe an alternative algorithm for dynamical pattern recognition, an algorithm whose basis involves solving a time-dependent differential equation. There is a 1:1 correspondence between the solution of the mathematical differential equation defining the algorithm and the behavior of the collective variables of a simple neural network. This algorithm can therefore be

collectively and robustly implemented by a small network of neurons, which will appear mysteriously effective in this task.

It is useful to contrast this collective variable approach to dynamical pattern recognition with the earlier approaches of the tempotron (Gutig & Sompolinsky, 2006, 2009) and transient synchrony (Hopfield & Brody, 2000, 2001). Applied to the word recognition problem, these approaches all begin with speech spectrograms, but then differ in the way that the spectrogram is preprocessed by a feedforward designed network or procedure. The tempotron and transient synchrony invoke recognition of the times at which analog signals in frequency channels cross onset and offset thresholds as the fundamental information fed on to the recognition system. The tempotron system generates a spike at that time; the transient synchrony system initiates decaying ramps of activity. By contrast, the collective variable approach uses the raw analog signal and its time derivative and does not invoke a preprocessing network for event detection.

All three systems have a neural dynamical system driven by synaptic inputs from the preprocessed signal that make the final computation and whose operation is the key to cadence flexibility. The tempotron is intrinsically a one-neuron dynamical system, with the required complex dynamics based on spiking activity and a detailed structured diversity of synapse dynamics that has been learned by experience. By contrast, collective variable and transient synchrony models involve the dynamics of many interacting neurons, and no single neuron is either indispensable or sufficient. Transient synchrony has weak collective dynamics of spiking activity, a collective effect that exists only under special input circumstances that produce action potential synchronization. Collective variable has strong collective dynamics that dominate the dynamics at all times and do not involve any details of action potential timing.

Finally, the three systems differ in their default knowledge and learned knowledge. The tempotron, the most flexible of the systems, has no intrinsic view of cadence variability. It knows about time warp only from the fact that cadence variability is present in a training set of examples. All information about time warp and the values of synapse strengths and dynamical parameters are acquired through a supervised learning procedure based on a large database of labeled examples. By contrast, the dynamics of the transient synchrony system contains implicit knowledge and classifies together a pattern over time and the same pattern uniformly stretched in time without the need to learn from time-warped examples. The collective variables system has even more range of variation captured in its implicit knowledge as to how to deal with cadence variability, but there is no tidy description of this knowledge.

In the collective variables approach, each neuron was used in a single linear bump attractor, and thus each was dedicated to recognizing a single word. For large numbers of neurons, multiple bump attractors can be simultaneously embedded in such a fashion that each neuron is involved

in several bump manifolds (Monasson & Rosay, 2013). If this is done, a set of neurons can embed many recognition subnetworks without the need for dedicating each neuron to a particular pattern to be recognized.

The vestibular-ocular system also has a single collective coordinate and a one-dimensional manifold of stable states (Arnold & Robinson, 1997; Aksay, Gamkrelidze, Seung, Baker, & Tank, 2001). The circuit design of the vestibular-ocular integrator system is entirely different from that of a bump attractor (Machens & Brody, 2008). The key fact that we have made use of in the dynamic pattern recognition system was that the value of the collective coordinate (bump location) determines whether the input of a neuron has an effect or is disregarded. The Seung model (see Seung, Lee, Reis, & Tank, 2000) description of the vestibular-ocular system has a related feature, for different neurons have different thresholds and the set of neurons that is above threshold (and thus is responsive) depends on the collective coordinate. In addition, the synapses are driven by a neuron saturate, so that driving an already strongly firing neuron to fire more rapidly does not alter the collective coordinate. This suggests that this circuit also might be useful in computations involving dynamic inputs, because the value of the collective coordinate determines whether an input to a neuron is effective or ignored.

The network studied here has considerable ongoing activity in the absence of sensory input. When the sensory input to any particular neuron in the network is turned off, the overall performance of the network is little changed, as expected, because the network performance is a collective property. However, the feedback within the network is so large that there is rather little visible effect of cutting off the sensory input even on the activity pattern of the neuron that was deprived of sensory input. One is reminded of the olfactory bulb in that regard (Adrian, 1950), where in awake behaving rats, each mitral cell, although driven directly by sensory neurons, shows considerable activity in the absence of an odor and an almost imperceptible change in activity due to odors (Rinberg, Koulakov, & Gelperin, 2006). In neocortex, the inter-area excitatory synapses to input layer IV comprise less than 20% of the intra-area excitatory synapses within that area, a structure that would permit the spiking background activity in the absence of a stimulus to be a defining aspect of the computational process. Further roles for such cortical ongoing activity have been recently proposed (Duarte & Morrison, 2014).

When input signals are small, the network state remains very close to a low-dimensional manifold produced by the ongoing activity in the feedback network. In most artificial neural networks designed to recognize patterns (and in the most common interpretational paradigm of neurobiology), input drives network activity, and in the absence of inputs, the system is quiet. In the present system, the network is always active, and the sensory signals redistribute the activity rather than create it.

Effective pattern recognition of realistic patterns with natural backgrounds is difficult. It generally requires both extensive learning based on a large database of examples and domain-specific cleverness in preprocessing a signal prior to the classifier. Neither has been done here. Until this is done, the potential engineering usefulness of the algorithm is difficult to evaluate.

Some of the ideas in this article may be useful in formulating and understanding experiments involving rodent hippocampal place cells. Consider a rat in virtual reality moving forward along a track, with visual patterns abcabcabcabc along the left wall and $\alpha\beta\gamma\delta\varepsilon\zeta\eta\theta\iota\kappa$ along the right wall. Each spatial location is visually unique, so many place cells will have learned to respond at unique locations. Now slowly turn off the input from the right wall. Will these place cells still have a unique response to position along the track even though the visual input is now ambiguous? If hippocampal place cell response is described by an input-driven motion of a bump attractor, a unique response should be expected. The situation is like that of a spoken word with repeated syllables but with spatial location replacing time.

In computer engineering, magnetic bubble domain (Bobeck & Scovil, 2001) shift-register memory chips exploited degeneracy in two dimensions to store a memory bit (a reversed-polarization bubble in a thin film of magnetic garnet) and to move it from one physical location to another by applying weak local magnetic field gradients. These bubbles are close 2D analogs of the 1D neural activity bumps studied here. However, neural circuitry has far more flexibility in its construction than physical magnetic systems have, and therefore far greater potential for computational exploitation of low-dimensional manifolds and collective behaviors.

Appendix: Motion in the Rigid Bump Approximation (RBA)

Let the fundamental neural equations of motion be

$$\frac{di_j}{dt} = -\frac{i_j}{\tau} + \Sigma_k J_{jk} f(i_k) + \Sigma_k A_{jk} f(i_k) + I_j, \tag{A.1}$$

where A_{jk} is a small perturbation representing an antisymmetric part of the connections and I_j is a small perturbation current. As in the simulations, J_{jk} comprises short-range excitation and longer-range inhibition (infinite range for simplicity), is translationally invariant, and is symmetric. J_{jk} has been chosen to produce a one-dimensional bump attractor 10 to 20 neurons wide in the simulations. $f(i)$ is a sigmoid with range 0–1.

Because the neurons are discretely located along a line, the actual stable states of the network in the absence of the perturbation are a set of stable points. For simplicity, we consider an infinite line of neurons with periodic

boundary conditions and will comment on the effect of truncating this line to produce ends. Each stable point is a bump spread over many neurons and centered on one of the neurons. There is an equivalent stable state centered on each neuron n. There is a Lyapunov (energy) function E for the dynamics of such symmetric networks, expressible in terms of the activity pattern of the neurons, and the stable states of the network are local minima of this function.

We begin with the empirical fact that in simulations, when there is a weak gradient in the input current (so that the input current I_j is a constant plus a term linear in j), the bump moves with a velocity proportional to that gradient, with a shape that closely resembles a fixed envelope of activity sliding across the neurons. This is a manifestation of the fact that although the energy function without input currents has the set of stable points described above, there is a continuous low-lying one-dimensional manifold of states connecting these minima. When the bumps are broad, the height of the barriers between these minima is so small that even a small gradient in the input current can push a bump over the barriers. In this limit, the bump drifts with uniform velocity proportional to the input gradient. In the following, we find the equation of motion for the bump location when the input current I_k varies smoothly with the index k, for a bump of finite extent.

Let i_k^m be the value of i_k for neuron k when the bump is at equilibrium and centered on neuron location m. The translational symmetry of the system makes $i_k^{m+1} = i_{k-1}^m$.

Consider the interpolated state of the input currents to neurons k for fixed m:

$$i_k^m(1-y) + i_k^{m+1}y$$

for $0 \le y \le 1$. This state has the appearance of shifting the bump smoothly from being centered at neuron m for $y = 0$ to being located at neuron $m+1$ for $y = 1$. Let v^m be the bump velocity when the bump is at m. Over time interval $0 \le t \le 1/v^m$, the variable $y = v^m t$ runs from 0 to 1. The rigid bump approximation uses this sliding bump shape as the approximate solution to equation A.1 while the bump is centered between m and $m+1$, and from this ansatz yields an expression for the drift velocity due to input currents as a function of bump location. In carrying out the analysis of the drift velocity due to input currents, the effect of the small connection anisotropy **A** will be neglected.

Define the discrete derivative of the bump shape with respect to the position of the bump as

$$\delta i_k^m = i_k^m - i_k^{m-1} = i_k^m - i_{k+1}^m,$$

where m is the position of the bump center. Then while the bump center is moving from m to $m + 1$:

$$i_i(t) = i_i^o + v^m t * \delta i_i^m. \tag{A.2}$$

Differentiating equation A.2 with respect to time, substituting the result into equation A.1, and evaluating at $t = 0$ yields

$$v^m \delta i_i^m = \left\{ -\frac{i_i^0}{\tau} + \sum_k J_{ik} f(i_k^0) \right\} + I_i = I_i.$$

The RBA is not exact for finite bump extents and infinitesimal I_i, so each equation for a particular neuron i yields a slightly different value of v^m. A suitable simple average involves multiplying each equation by δi_i^m to weight the velocity estimates for neurons that are changing state most rapidly. The result is

$$v^m = \Sigma_i \delta i_i^m I_i / \Sigma_i \delta i_i^m \delta i_i^m. \tag{A.3}$$

When the bump is centered at location m, δi_i is an odd function of $(m - i)$, which vanishes for large values of $|m - i|$, and represents the discrete derivative of the bump shape in input-current variables. The denominator is a normalizing factor that does not depend on m. A bump shape $K(m - i)$ in input currents is defined (for integer values of its argument) by subtracting the asymptotic value of the input current for neurons distant from the bump center. This allows the sum in equation A.3 (discrete form of an integral) to be integrated by parts.

We now pass to the limit of smooth variables. Let the variable $y(t)$ describe the bump-center position as a function of time. dy/dt agrees with v^m when the value of y is the integer m and interpolates smoothly between these values. The bump shape function is smoothly interpolated over the continuous variable x. The equation for bump motion due to weak input currents, when variables are smooth in the their indices, becomes

$$\frac{dy}{dt} = \lambda d/dy \int K(x) I(y(t) - x, t) dx, \tag{A.4}$$

where the constant λand the form of the kernel K are λ are described by equation A.3.

Simulations involving a connection range of 4 (to neighbors on each side) and a bump width of about 15 and an I_k that depends linearly on k showed a bump of approximately fixed shape moving smoothly at a speed proportional to the constant gradient of I_k and close to that expected from equation A.3. Krotov (personal communication, 2014) has verified that this

equation is exact in the limit of a field theory, where the number of neurons in a bump is first taken to infinity and then the effect of a perturbation is evaluated in lowest order.

A similar analysis was used to evaluate the role of the antisymmetric part of the connection matrix represented by A_{jk} in equation A.1. In this article, A_{jk} was both antisymmetric and translational invariant, so A_{jk} is an odd function of the variable $(k - j)$. Simulations were carried out on a connection matrix having a small antisymmetric term of the same range as the excitatory connections used to stabilize the bump. These showed a bump of almost fixed shape moving at constant speed with a velocity proportional to the size of the antisymmetric A. If the antisymmetric part of J were spatially dependent, this drift velocity would also depend on position.

While this analysis is based on an infinite chain of neurons, the simulations are based on a finite chain with ends. Experimentally, the effect of terminating the chain merely constrains the bump to lie within the interior of the chain, an effect that can be achieved by many different termination models as long as the termination pattern chosen does not result in binding the bump to an end. The situation is qualitatively like the motions of a droplet of mercury free to move on a horizontal glass surface in response to small forces. If bounding walls are now added to confine the region of motion, the droplet motion will simply be constrained to the interior area. The details of the interaction with the bounding walls simply do not matter as long as they do not result in the droplet binding to the wall.

Acknowledgements

I thank Carlos Brody for remarks on a draft, Sebastian Seung for general discussions of neural dynamics and emergent properties, and Dmitri Krotov for working out a field theory of weakly perturbed bump motion.

References

Adrian, E. D. (1950). The electrical activity of the mammalian olfactory bulb. *EEG Clinical Neurophysiology*, *2*, 377–388.

Aksay, E., Gamkrelidze, G., Seung, H. S., Baker, R., & Tank, D. W. (2001). In vivo intracellular recording and perturbation of persistent activity in a neural integrator. *Nat. Neurosci.*, *4*, 184–193.

Amari, S-I. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biol. Cybernetics*, *27*, 77–87.

Arnold, D. B., & Robinson, D. A. (1997). The oculomotor integrator: Testing of a neural network model. *Exp. Brain Research*, *113*, 57–74.

Barak, O., Sussillo, D., Roma, R., Tsodyks, M., & Abbott, L. F. (2013) From fixed points to chaos: Three models of delayed discrimination. *Progress in Neurobiology*, *103*, 214–222. http://dx.doi.org/10.1016/j.pneurobio.2013.02.002

Bobeck, A. H., & Scovil, H. E. D. (2001). Magnetic bubbles. *Scientific American*, *78*, 224.

Dambre, J., Verstraeten, D., Schrauwen, B., & Massar, S. (2012). Information processing capacity of dynamical systems. *Scientific Reports*, *2*, art. 514. doi:10.1038/srep00514

Duarte, R. C. F., & Morrison, A. (2014). Dynamic stability of sequential stimulus representations in adapting neuronal networks. *Frontiers in Computer Neuroscience*, *22*, doi:10.3389/fncom.2014.00124

Eliasmith, C., & Anderson, C. H. (2003). *Neural engineering*. Cambridge MA: MIT Press.

Furui, S. (1986). Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions in Acoustics, Speech, and Signal Processing*, *34*, 52–59.

Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallett, D., & Dahlgren, N. (1990). *DARPA, TIMIT Acoustic-Phonetic Continuous Speech Corpus* (CD-ROM). Gaithersburg, MD: National Institute of Standards and Technology.

Gutig, R., & Sompolinsky, H. (2006). The tempotron: A neuron that learns spike timing–based decisions. *Nat. Neuroscience*, *9*, 420–428. doi:10.1038/nn1643

Gutig, R., & Sompolinsky, H. (2009). Time-warp invariant neural processing. *PLOS Biology*, *7*(7), e100041. doi: 10.1371/journal.pbio.1000141

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences, USA*, *81*, 3088–3092.

Hopfield, J. J. (1994). Neurons, dynamics, and computation. *Physics Today*, *47*, 40–46. doi:10.1063/1.881412

Hopfield, J. J., & Brody, C. D. (2000). What is a moment? "Cortical" sensory integration over a brief interval. *Proceedings of the National Academy of Sciences, USA*, *97*, 13919–13924.

Hopfield, J. J., & Brody, C. D. (2001). What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration. *Proceedings of the National Academy of Sciences, USA*, *98*, 1282–1287.

Maass, W., Natschlaeger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, *14*, 2531–2560.

Machens, C. K., & Brody, C. D. (2008). Design of continuous attractor networks with monotoic tuning curves using a symmetry principle. *Neural Computation*, *20*, 452–485. PMID:1804741.

McNaughton, B. L., Barnes, C. A., Gerrard, J. L., Gothard, K., Jung, M. W., Knierim, J. J., ... Weaver, K. L. (1996). Deciphering the hippocampal polyglot: The hippocampus as a path integration system. *Journal of Experimental Biology*, *199*, 173–185.

Monasson, R., & Rosay, S. (2013). Crosstalk and transitions between multiple spatial maps in an attractor neural network model of the hippocampus: Phase diagram. *Physical Review E*, *87*. doi:10.1103/PhysRevE.87.062813

Mytkowicz, T., Diwan, A., & Bradley, E. (2009). Computer systems are dynamical systems. *Chaos*, *19*, 033124. http://dx.doi.org/10.1063/1.3187791

Rinberg, D., Koulakov, A., & Gelperin, A. (2006). Sparse odor coding in behaving mouse. *Journal of Neuroscience*, *26*, 8857–8865.

Seung, H. S., Lee, D., D., Reis, B., Y., & Tank, D. W. (2000). Stability of the memory of eye position in a recurrent network of conductance-based model neurons. *Neuron.*, *26*, 259–271.

Shadlen, M. N., Britten, K. H., Newsome, W. T., & Movshon, J. A. (1996). A computational analysis of the relationship between neuronal and behavioral responses to visual motion. *Journal of Neuroscience*, *16*, 1486–1510.

Tsodyks, M. (1999). Attractor neural network models of spatial maps in hippocampus. *Hippocampus*, *9*, 481–489. doi:10.1002/(SICI)1098-1063(1999)9:4<481::AID-HIPO14>3.0.CO;2-S

Tsodyks, M., & Sejnowski, T. (1995). Associative memory and hippocampal place cells, *Int. J. Neural Syst.*, *6*, 81–86.

Wu, S., Hamaguchi, K., & Amari, S. (2008). Dynamics and computation of continuous attractors. *Neural Computation*, *20*, 994–1025.

Xie, X., Hahnloser, R. H. R., & Seung, H. S. (2002). Double-ring network model of the head direction system. *Phys. Rev. E*, *66*, 041902–1–9.

Zhang, K. (1996). Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: A theory. *Journal of Neuroscience*, *16*, 2112–2126.

Received January 11, 2015; accepted May 27, 2015.