Information recall using relative spike timing in a spiking neural network

Philip Sterne
Cavendish Laboratory, University of Cambridge

Abstract

We present a neural network that is capable of completing and correcting a spiking pattern given only a partial, noisy version. It operates in continuous time and represents information using the relative timing of individual spikes. The network is capable of correcting and recalling multiple patterns simultaneously.

We analyse the network's performance in terms of information recall. We explore two measures of the capacity of the network, one that values the accurate recall of individual spike times, and another that values only the presence or absence of complete patterns. Both measures of information are found to scale linearly in both the number of neurons and the period of the patterns, suggesting these are natural measures of network information. We show that there is a smooth transition from encodings which provide precise spike times to flexible encodings that can encode many scenes. This makes it plausible that many diverse tasks could be learnt with such an encoding.

1 Introduction

Biological neurons have many sources of randomness inherent in their structure (Faisal et al., 2008), yet biological neural networks are able to form robust representations which can be recalled using only a partial cue. Using a neural model that is intrinsically noisy, we use spatiotemporal patterns of spikes to create a robust representation that can be activated using only a small subset of the spikes that comprise it. Our representations are formed from many patterns and each pattern is stored in the network using the delays inherent in neural signal propagation. Our neural model uses many dendrites per neuron, with each dendrite detecting the presence of a particular pattern involving that neuron. The main thrust of this paper will be to quantify the performance of the network as an associative memory using concepts from information theory. With these performance measures we can also focus on secondary issues such as the tradeoffs between representations involving many small patterns and representations with only a few large patterns. The performance measures also allow us to compare recall performance of networks which use a rate-like encoding mechanism, and we can quantify the differences in information between rate-coding and spike-timing for continuous-time synfire chain models.

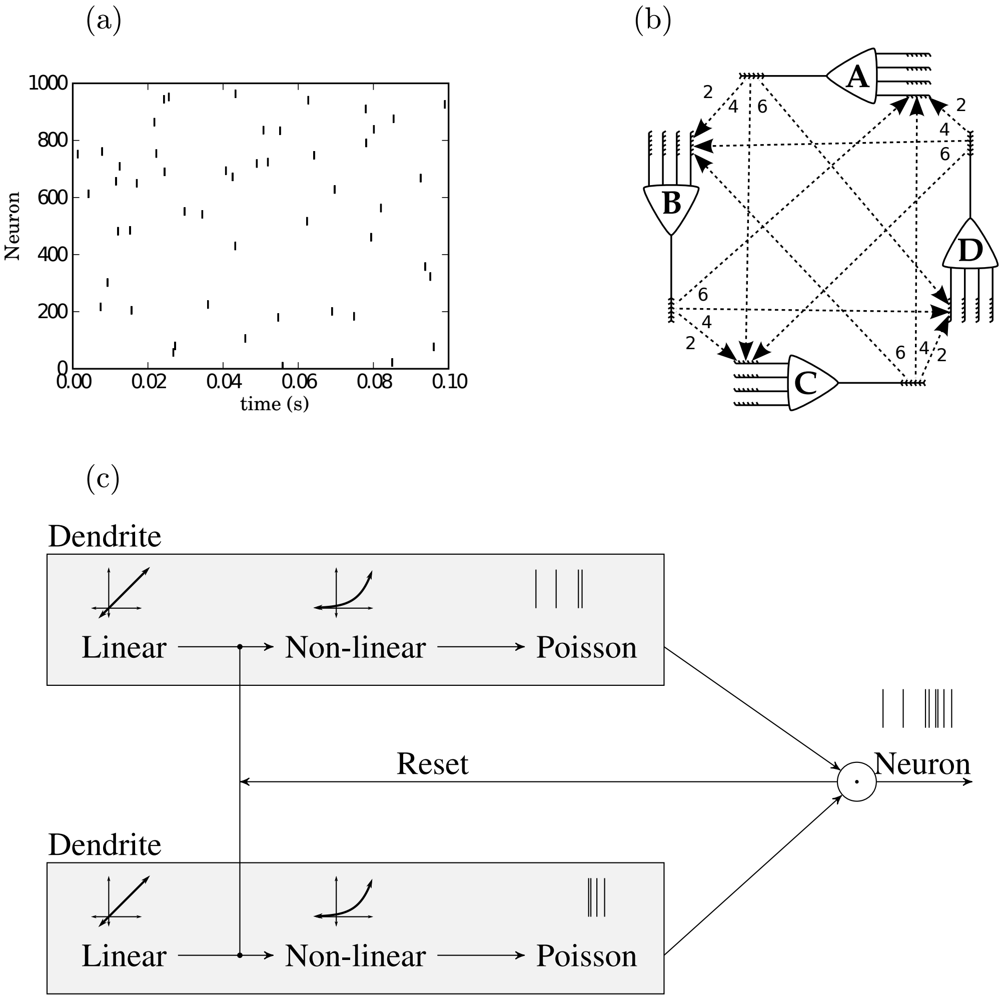


Figure 1: Overview of the patterns, how they are stored, and the neural model used in this paper. In (a) we show an example firing pattern. It is a sparse code as only 50 neurons fire in a 100 millisecond period out of a population of 1000 neurons. This paper examines how many such patterns can be stored and recalled in a single population of neurons. In (b) we show a schematic of the wiring of firing pattern *ABCD*, with each spike occurring after 2ms after the previous spike. Each neuron has a single dendrite dedicated to the detection of the pattern. Once the pattern is started it will repeat indefinitely. The transmission delays must be set up so that spikes from the three preceding neurons will arrive simultaneously at the fourth. (c) The structure of the neural model. Each dendrite has an LNP model, and the neuron spikes whenever an individual dendrite spikes. All dendrites on a neuron are reset whenever the neuron spikes. After Simoncelli et al. (2004).

As a motivating example consider a simplified visual world consisting of M possible features. In this simplified world, a visual scene always has exactly m uncorrelated features present. Each feature has a particular pattern of spikes associated with it (see figure 1(a)). The representation of this scene is simply the union of all the spikes of the features present. However, sensory input does not give an exact encoding: We only have a noisy cue with missing and jittered spikes present. The network must then fill in any missing spikes and correctly synchronise all the spikes within each active pattern. The network will then have restored a more faithful representation of the visual scene than the scene created by the original noisy sensory neurons.

For our purposes we create random periodic patterns so that we know what the ideal patterns should look like. To test the recall performance of the network a noisy, incomplete cue is used to start the recall and after a time equal to several periods of the pattern we examine how closely the network spikes resemble the ground truth. For the purposes of this paper there are two measures of performance. The first measure is related to the richness of the representation. While $\binom{M}{m}$ scenes can be represented, the fact that the scene has to be represented by the neural network places some constraints on m given M. The second measure of performance is related to how much information can be added by correctly aligning noisy spikes and introducing missing spikes. These measures are somewhat at odds with each other as a flexible representation will require a larger, more complete cue, leaving less room for information to be filled in.

A key idea of this scheme is to use the transmission delay from presynaptic to post-synaptic neuron to detect a particular pattern, (Wills, 2004; Hopfield, 1995; Gerstner et al., 1993). This is similar to Polychronization (Izhikevich, 2006; Vertes and Duke, 2010) and Synfire chains (Abeles, 1991; Bienenstock, 1995). Our contribution is to quantify how many bits such methods can convey and closely examine the tradeoffs between the two measures of information when using different parameter settings. We measure these using random patterns that we have created. Since we have complete access to the ideal patterns it is easy to store the pattern in the network. We can then contrast the actual performance with the ideal pattern.

We insert a pattern into the network by adding a dendrite to each neuron which participates in the pattern. Each dendrite is wired up to a subset of the participating neurons with a set of delays such that when the pattern is present many post-synaptic potentials will arrive simultaneously. Using a neural model which detects coincident spikes we can then detect that the particular pattern is active and the current neuron should spike. Since the patterns are periodic, then once a critical threshold is reached the pattern will become self-sustaining (see figure 1(b)).

2 Methods

Measuring the recall performance of a network that has many patterns stored required the design of a new neural model, which we describe in section 2.1. Detection of recalled patterns using grandmother neurons is covered in section 2.2 (and it must be emphasized that the grandmother neurons are used only for measuring recall, they do not form part of the proposed architecture). We quantify how accurate the recall is in section 2.3 before finally discussing how to improve the performance by slightly modifying the synaptic delays.

2.1 Description of the model

In this paper a neuron is viewed as a collection of dendrites, each of which generate spikes individually (see figure 1(c)). There is evidence that dendrites can be functionally separate (Liu, 2004), (London and Häusser, 2005), (Polsky et al., 2004), (Softky, 1994), (König et al., 1996), (Golding et al., 2002), but our choice of functionally-separate dendrites is primarily to aid the easy insertion and detection of a pattern into the network.

Each dendrite generates spikes according to a linear-nonlinear-Poisson (LNP) model (Simoncelli et al., 2004). When a single dendrite generates a spike, the whole neuron spikes and all the dendrites enter a refractory period. Other than this collective reset behaviour all dendrites are considered separately.

We define ψ_n to be the set of all spike times of neuron n:

$$\psi_n = \{\tau | \text{neuron } n \text{ spiked at time } \tau\}. \tag{1}$$

We define β to be a sum of Dirac delta functions, with each function positioned on a spike time plus a delay ϕ:

$$\beta(n, \phi, t) = \sum_{k \in \psi_n} \delta(k + \phi - t). \tag{2}$$

Equation 2 represents the post-synaptic input of a delay line (dendrite) connected to neuron n with delay ϕ. We define n_{ij} to be the presynaptic neuron, and ϕ_{ij} to be the transmission delay of the j^{th} synapse on the i^{th} dendrite. The voltage of the ith dendrite changes according to:

$$\frac{dv_i}{dt} = \frac{1}{\lambda}\left(v_{\text{EQ}} - v_i\left(t\right)\right) + \sum_{j=1}^{g} w_{ij}\beta(n_{ij}, \phi_{ij}, t), \tag{3}$$

where g is the number of synapses for each dendrite and w_{ij} is the synaptic weight. In this paper we use a single constant for all the synaptic weights and denote this by w. If some spikes were more reliable indicators of other patterns they could receive a stronger synaptic weight. Equation 3 therefore represents simple exponential decay to an equilibrium voltage (v_{EQ})with a time constant of λ, and whenever a postsynaptic potential is received the voltage is increased by w_{ij}.[1]

In initial experiments the network was extraordinarily accurate, often exceeding microsecond precision. This precision resulted in a large amount of information in the spike times (roughly four times as much as the results shown in section 3.2), but is biologically unrealistic. We therefore added transmission noise to each ϕ_{ij}, so that the transmission time varies with a standard deviation of 2% of ϕ_{ij}[2]. Note that adding noise limits the accuracy of the recalled patterns but makes the results more relevant to

[1] Appendix C has been provided to help with the many symbols used in this paper

[2] In a delay line of 50ms the standard deviation will be 1ms. Almost all transmissions along this delay line will be within 2 standard deviations: (48ms, 52ms).

biological networks. The instantaneous rate of spiking of the dendrite is computed by exponentiating the voltage $v_i(t)$:

$$r_i(t) = \gamma e^{\alpha v_i(t)}, \tag{4}$$

where $\alpha = 1\,\mathrm{V}^{-1}$ and $\gamma = 1\,\mathrm{s}^{-1}$ in this paper.

Once the dendrite generates an event the neuron spikes and the voltage of all the neuron's dendrites are reset to a large negative voltage (v_{RESET}) which decays away in the same manner as all other synaptic inputs. This simulates the neuron's refractory period. While this violates the strict definition of a Poisson process (spikes are not independent of each other) we will still refer to the spikes being generated by a Poisson process since, conditioned on the voltage, the spikes are independent of each other. In other words, given the current voltage one can immediately calculate the instantaneous firing rate $r_i(t)$ which gives the probability of spiking in the next infinitesimal time-slice. Knowing the entire past spiking history does not provide any further information about the instantaneous rate of spiking, and in this sense is memoryless (or Poisson).

Informally in our model, each dendrite is a coincidence detector (Barlow, 1985). The baseline voltage can be set so that spontaneous firing happens infrequently (here the baseline rate is once every 500 seconds). For each excitatory post-synaptic potential (EPSP) the instantaneous rate of spiking increases by a factor of $e^{\alpha w}$ for a short time (λ). If several spikes are received in a time shorter than λ, then the rate of spiking increases exponentially in the number of spikes and, for sufficiently many spikes, the dendrite will spike on a functionally-relevant timescale with high probability. This can be seen in figure 2 in which a single dendrite is simulated for a short period of time. In the beginning five spikes are received over a relatively long timescale, giving the voltage time to decay away and the dendrite spikes with relatively low precision (there is space between the fifth spike and the post-synaptic response). Near the end of the simulation four spikes are received in quick succession and the dendritic voltage is driven to a high value as there is no time to decay, the firing rate increases exponentially and the postsynaptic spike occurs very soon after the fourth spike was received.

When wiring up a dendrite to detect a particular pattern we are not required to connect to all the spikes in the pattern. Each synapse in a dendrite has a delay-line and neuron associated with it. The number of spikes in a pattern we denote by G and the number of synapses in any dendrite is g (since we allocate a single dendrite to detect a single pattern and do not allow self-connections then necessarily $g < G$). For computational reasons we found $g = 20$ a fast default setting, with performance improving only gradually with larger g. One might imagine in a biological system that maintaining long delay lines incurs a large cost and hence favour short delay lines. Throughout this paper we ignore such considerations and simply connect the dendrite to a random subset of the pattern.

In figure 3, 500 patterns have been stored in a network of 1000 neurons. Each pattern has $G = 50$ spikes. This means that each neuron participates in approximately 25 patterns. For every pattern that a neuron participates in, a single dendrite is added to the neuron, and hence each neuron has approximately 25 dendrites. Each pattern has a period of 100ms, and the voltage decay half-life is 5ms. (This implies $\lambda = \frac{5}{\log(2)}$ms.)

Each dendrite is connected to a subset of the other neurons which participate in the pattern (here each dendrite synapses onto 20 of the 49 other neurons which form the

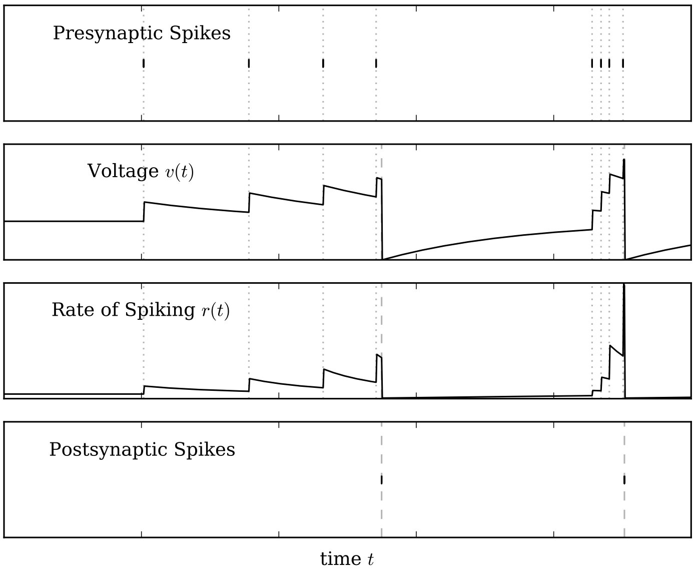


Figure 2: The dynamics of a single dendrite. As presynaptic spikes are received, the membrane potential is increased by a fixed amount and gradually decays back to the equilibrium. If several presynaptic spikes are received in quick succession, there is insufficient time for the voltage to decay away and the rate of spiking is exponentially high, resulting in postsynaptic spikes that are extremely accurately timed. See text for more details.

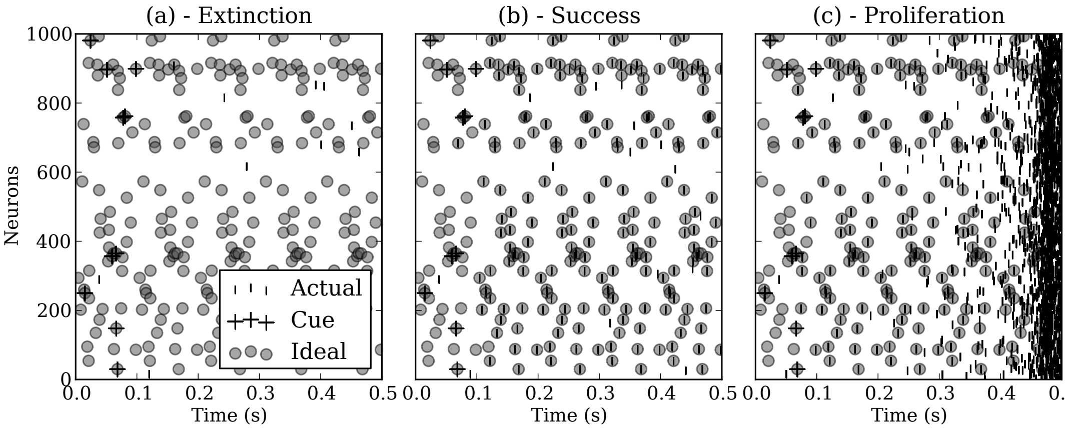


Figure 3: The capabilities and failures of a concurrent recall network with 1000 neurons. 500 periodic patterns (each consisting of 50 spikes and a period of 100ms) have been stored in the network. Recall is attempted by giving a starting cue of 10 spikes from one of the patterns (20% of the pattern). Panels (a)-(c) shows the result for changing synaptic strengths w=2, 3, and 4. Panel (a) shows the pattern dying out because synaptic strengths are too weak. In (b) there is successful recall. In (c) the network saturates because spurious spikes proliferate.

pattern). There are only 49 neurons to choose from as self-connections are not allowed (auto synapses with long transmission times are not biologically plausible). The 20 synapses are randomly selected from the 49 neurons and are given the correct delays.

To test the recall of the network, 10 spikes are selected at random from a pattern and injected into the network. Figure 3 shows the three possibilities: either the pattern is successfully recalled, the spiking dies out, or the pattern starts to be recalled but as more spurious spikes occur it leads to the proliferation of spikes and the network becomes saturated.

2.2 Detecting successful retrieval

Once we have wired up the network to store many patterns, we are able to recall several of the patterns simultaneously as long as the network does not saturate. We can give the network several partial, corrupted patterns and see which patterns have been successfully cleaned up. However, as shown in figure 4, the resulting mass of spikes can make detection difficult. Simple grandmother cells are used here to detect precise spiking patterns. Sigma-Pi neurons (Wills, 2004) were wired up so that if 3 successive spikes from a particular pattern are detected with the correct temporal relationship (within a window of 3 milliseconds) then the grandmother neuron will spike. Unlike the LNP dendrites the grandmother neurons never generate spikes spuriously.

Since a Sigma-Pi neuron can have many dendrites, we wire each dendrite to detect a different phase of the pattern. Thus if a pattern is correctly recalled, the associated grandmother neuron will be characterised by a steady spiking pattern.

In figure 4, 30 patterns have been stored in a network; all the patterns have periods ranging from 30 milliseconds to 300 milliseconds. 3 patterns are selected to be recalled and 10 spikes from each pattern are injected into the network (representing 20% of the pattern). The correct response for the grandmother neurons representing those patterns is to spike regularly.

A pattern is successfully recalled if more than half the expected grandmother spikes are present. In figure 4 all three patterns have been successfully recalled, but we can see that the grandmother neurons hardly fired for the first period of recall. In the early stages of recall it can therefore be ambiguous as to whether the recall will be successful or not. To avoid this problem all the experiments are run for at least 10 periods to ensure that there is sufficient time for recall or failure to be unambiguous.

Having covered the neural model and how patterns are detected in the network we now discuss how to evaluate the performance of the network after a noisy cue has been recalled.

2.3 Measuring Information

This section looks at the different types of information that can be encoded in the network. Accurate individual spike times convey a lot of information that could be used for further processing by higher-level areas. The network is also capable of recalling multiple patterns simultaneously. This allows us to measure the flexibility of the encoding - the number of possible subsets of stored patterns that can be simultaneously recalled. This measure is related to the intrinsic information from a lower-level processing area,

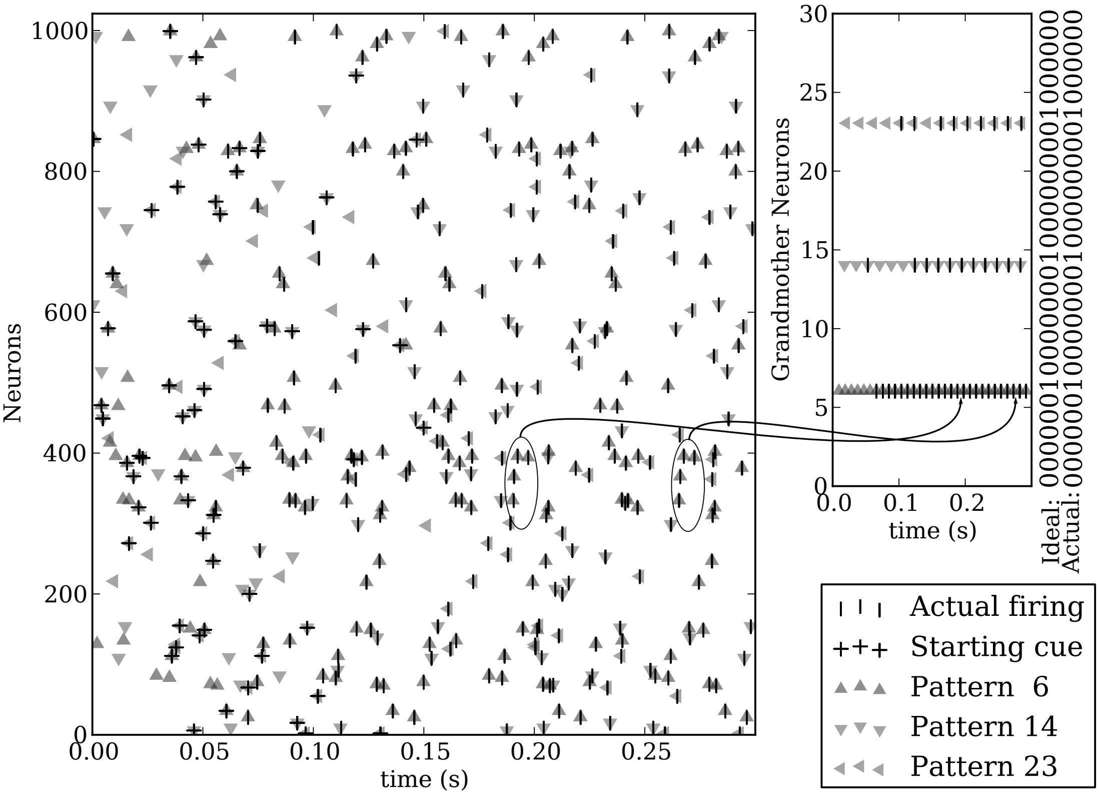


Figure 4: Multiple patterns can be completed simultaneously. Here we store 30 patterns and recall 3. To detect which patterns are recalled we use a grandmother neuron for each pattern. We threshold the number of grandmother spikes to estimate whether a pattern is present. This gives us the actual encoding, which we can compare with the ideal encoding.

e.g. if a visual system can extract M possible features from a scene and the encoding is such that only m features are present then the task has an intrinsic information of $\log_2 \binom{M}{m}$ bits, as $\binom{M}{m}$ different scenes are possible.

In some sense the information capacity of a concurrent recall network is infinite, as one can increase the capacity without bound by decreasing λ, decreasing v_{EQ} and increasing the synaptic strength. More specifically as $c \to 0$:

$$\lambda' \to c\lambda, \tag{5}$$

$$v'_{\mathrm{EQ}} \to v_{\mathrm{EQ}} + \alpha \log c, \tag{6}$$

$$w' \to w - \alpha \log c. \tag{7}$$

The above transformations correspond to a scaling of time, and each pattern will require ever finer precision to be recalled successfully. This means more patterns can be stored and simultaneously recalled. However in realistic physical systems infinite precision is impossible and for the rest of this paper we assume that we are unable to control λ.

2.3.1 Information of whole patterns

One measure of information is the total number of subsets of patterns that the network can recall. To determine which patterns are being recalled we measure the spiking output of the grandmother cells. If the number of spikes in a certain period exceeds a threshold then the pattern is deemed to have been recalled (see section 2.2).

In some sense, the information contained in which patterns are active represents the information transmitted by the grandmother neurons. We are not proposing that grandmother neurons would be found or used in a real neural architecture; rather that by measuring the grandmother neurons' output we can get insight into the representational capabilities of the network.

To obtain a numerical measure of how well the network is performing we estimate the number of bits communicated (ignoring precise spike times for this section). This is the mutual information I_1 between the ideal bit vector and the recalled bit vector (as shown in figure 4) that is created from observing the grandmother cells. If the network was perfectly reliable, then after recalling m patterns from M stored patterns $\log_2 \binom{M}{m}$ bits would have been communicated. However the network is not perfectly reliable, and some patterns might be recalled that should not be while other patterns might not be recalled that should be.

We can estimate the information contained in the set of patterns actually recalled by first estimating how many bits need to be added before we obtain the ideal encoding in which all the desired m patterns are active and all the remaining patterns are inactive. If there are $m_\uparrow$ patterns which need to be turned on, out of a possible M_0 patterns that are off, then the total number of possible subsets is $\binom{M_0}{m_\uparrow}$, and picking one of these will require $\log_2 \binom{M_0}{m_\uparrow}$ bits. Similarly for the active patterns M_1 that require turning off $m_\downarrow$ we incur a cost of $\log_2 \binom{M_1}{m_\downarrow}$ bits.

Subtracting the correction information that is required from the information of the ideal message, gives us the actual number of bits communicated:

$$I_1 = \log_2 \binom{M}{m} - \log_2 \binom{M_0}{m_\uparrow} - \log_2 \binom{M_1}{m_\downarrow}. \tag{8}$$

However this is only an estimate of the mutual information as one also needs to communicate the number of errors for both conditions ($m_\uparrow$ and $m_\downarrow$). The number of bits needed to communicate this in practice is small compared to I_1. It also requires a large number of trials to estimate the entropy of $m_\uparrow$ and $m_\downarrow$ accurately, so for this paper it is ignored entirely.

In this paper we consider the parameter regime where only a few patterns are active out of many possible patterns. This leads to a sparse vector representation. If the network is operating fairly reliably, then both error conditions are infrequent ($m_\uparrow \approx m_\downarrow \approx 0$). In this case the penalty for incorrectly turning a pattern on is approximately $\log_2 m$ and the penalty for failing to recall a pattern is approximately $\log_2 M$.

Since incorrectly turning patterns on incurs a lower penalty, one might favour network settings where a small number of spurious patterns are turned on. However in a sparse codes the relative increase in network activity ($\propto \frac{1}{m}$) can be large and this tended to result in proliferation.

2.3.2 Information of individual spike times

Another measure of information is the precise timing of individual spikes within each pattern that the network is able to reproduce. To measure this we look at the mutual information between the ideal pattern which we store, and the noisy pattern that is recalled.

To calculate the information transmitted by the neural network we imagine that we are trying to send a message of the exact times of perfectly recalled patterns. We then compare the difference in message length between two observers. An ignorant observer would distribute probability mass equally over all possible times and neurons. An attentive observer who has seen the spikes of our network is able to place blobs of higher probability on areas where the network has spiked and smaller probability everywhere else. If the network is doing a good job of recall, then this strategy will produce shorter messages.

Whenever a spike is seen, the attentive observer then believes in a high rate of spiking (r^+) centered around the spike time, with a width 2Δ and lowers the belief in the rate of spiking in remaining areas(r^-). If n_+ spikes actually occur within the high rate of spiking and n^- within the low rate of spiking then the network has transmitted

$$I_2 = n^+ \log_2 \left(\frac{r^+}{r} \right) + n^- \log_2 \left(\frac{r^-}{r} \right), \tag{9}$$

bits. To calculate the information transmitted we need to know how wide Δ should be and how to distribute probability mass between r^+ and r^-. Rather than an analytic approach we search through all possible values and choose the optimal for each simulation. Technically this is cheating, as we are using the correct data to help decode the output, however on successive runs the optimal parameters change very little. This means that very little information is actually being leaked. As a thought experiment, imagine that before decoding spikes from a network, you run many similar simulations of the network and use those simulations to decide on optimal values for Δ, r^+ and r^-. These will be close to the optimal parameters for your particular decoding task, and hence your performance will be close to the optimal performance reported here.

2.4 Synaptic strength and transmission delays

When performing recall the synaptic strength (w) should be as large as possible while avoiding proliferation. A large synaptic strength allows the initial cue to be as small as possible, while still ensuring successful recall. However strong synapses can create biased recall times. This is shown in figure 5(a). For illustrative purposes we ignore the discrete nature of individual spikes in figure 5(a) and plot the behaviour of a dendrite for various synaptic strengths. While the rate of receiving spikes is constant for each synaptic strength, the average voltage is affected by stronger synapses. The higher average voltage creates an exponentially higher rate of spiking[3], and this in turn means that the distribution of the first spike moves earlier. If on average each neuron spikes early then there will be a phase precession. The resulting bias in spike times will affect the information measured by I_2 as the spike times are not as precise as is possible.

Any bias in the spike times can be compensated for by adjusting the delay lines; we could subtract out the bias, or multiplicatively correct for the bias. Ideally we would like each delay line to have an equal probability in contributing to the spike. Since the noise grows proportionally to the delay, the correct way to achieve this is by multiplying each delay line by a constant value. Note that this assumes that each EPSP merely has to arrive before the correct time to contribute to the spike. If a delay line was large then it is possible that a spike could arrive early and its influence would have decayed away by the correct time. As a rule of thumb we ensure that two standard deviations of the noise in the longest possible delay line is still less than the time constant of the decay:

$$2\sigma < \lambda. \tag{10}$$

In our simulations, $\sigma = 0.02T = 0.002\,\mathrm{s}$ and $\lambda = \frac{0.005}{\ln 2}$ s, so this requirement is satisfied.

The optimal scaling for a pattern with $G = 50$ spikes and $g = 20$ synapses per dendrite is shown in figure 5(b). Each point in figure 5(b) was created by storing a single pattern with a given synaptic strength and scaling all delays by a constant factor. When the pattern was recalled it could be examined for evidence of bias in the recall times (e.g. the network could be consistently firing earlier than in the ideal pattern). If the recall was biased earlier, then we would increase the synaptic delays and vice versa. The optimal empirical scaling factors produce unbiased spike times and as such maximise the spike time information (I_2).

3 Results

In the methods section we have shown how to wire a network so it is able to perform recall. We also introduced two different performance measures that measure different aspects of recall; I_1 measures the information contained in the selection of patterns to recall, while I_2 measures the information contained in individual spike times. We now show results that illuminate how different aspects of the network design can influence the performance of both of these measures. The most crucial decision is how many patterns to store. As more patterns are stored, the network can recall fewer patterns

[3] While the rate might be exponentially higher the voltage gets reset after each spike, so there is never more than a single spike

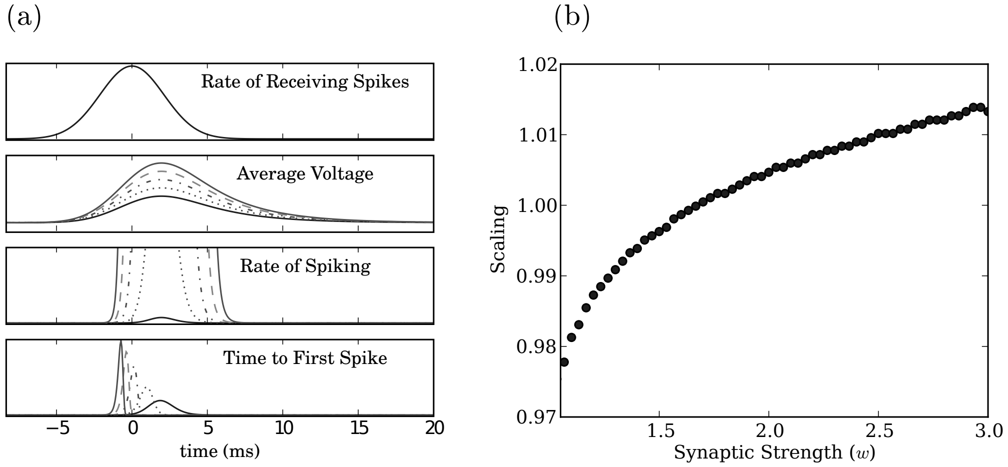


Figure 5: The synaptic strength can introduce a bias into the spike times which needs to be corrected for optimal recall performance. In (a) we assume that the arriving spikes are distributed around the correct time ($t = 0$) but have intrinsic noise. The average voltage will depend on the synaptic strength, and we plot the averages for several different synaptic strengths (w). The different synaptic strengths affect the average rate of spiking and hence the distribution of the first spike time. For a given synaptic strength we show the empirically-optimal scaling of delay lines for unbiased spike times in (b). See text for more details.

simultaneously. In section 3.2 we examine the recall properties of the network when given a noisy cue. We also explore the effect of a pattern's size on the performance measures in section 3.3.

For the final section we define a rate code as a pattern with a period equal to the membrane time constant (λ). In this setting the exact timing of the spike becomes less important and the code is decoded mainly through the presence or absence of spikes. We show evidence that a rate code is significantly less efficient than a timing code for both performance measures.

3.1 Stored versus recalled patterns

When encoding information one is free to decide how many patterns to store and how many of those patterns should be simultaneously recalled. If many patterns are stored then fewer can be simultaneously recalled before proliferation occurs.

Wills (2004) analyses a model in which a dendrite will fire if D spikes are received in a short time period. Their analysis predicts that the maximum number of patterns that can be recalled (m) as a function of the number of patterns stored (M) is

$$m \propto M^{-\frac{1}{D-1}}. \tag{11}$$

Since our neural model is similar, one might expect the same conclusion to hold here. In figure 6 we fit curves to both measures of the performance of our network and there is a good fit. Figure 6 also shows the number of bits communicated by the network for

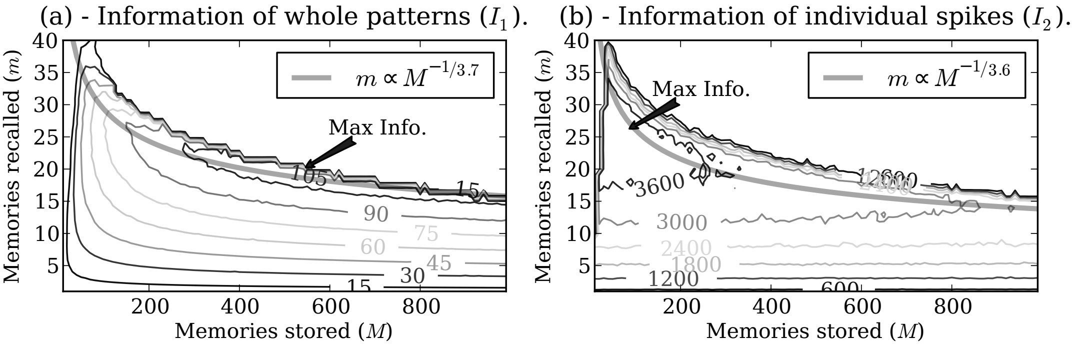


Figure 6: The number of bits per period (shown in contours) that can be communicated in the network as a function of the number of patterns that have been stored (M) and the number of patterns simultaneously recalled (m). For a given number of stored patterns the maximum information is approximated by $m \propto M^{\frac{1}{D-1}}$ for both information measures.

each information measure. To start each selected pattern, neurons representing half the pattern fire. The network is left to run for 1 second (10 periods of recall). If proliferation is detected, then there is no information transmitted (proliferation is defined as twice the number of expected spikes). It is important to note that figure 6 also fixes the number of spikes in a pattern $G = 50$ and the number of synapses in a dendrite $g = 20$. In section 3.3 we explore the effect of changing the pattern size.

Figure 6(a) shows that the whole-pattern information (as defined in equation 8) is maximized when many patterns are stored. Precise recall is not important, as long as the patterns are recalled sufficiently well.

In figure 6(b) we see that the information provided by spike times (as defined in equation 9) is roughly independent of the number of patterns stored (M). Instead the information only depends on the number of patterns being recalled (m). Since the proliferation boundary satisfies $m \propto M^{-\frac{1}{D-1}}$, we can recall more patterns simultaneously by only storing comparatively few patterns. Note that the fitted line of best performance is not on the edge of the proliferation boundary as many spikes unrelated to a particular pattern begin to appear near the proliferation boundary.

3.2 Information Retrieval

For the network to be useful we need to show that the network can improve an initial noisy cue which has jittered and missing spikes. We compare the results when storing a few large patterns and storing many small patterns. This provides insight into some of the tradeoffs of group size when cleaning up a noisy cue.

To construct the network with large patterns, we store $M = 10$ patterns of $G = 500$ spikes each, in a network of 1000 neurons. Each dendrite connects to a subset of $g = 20$ preceding spikes. We start by measuring the information in an initial set of spikes (the noisy cue). The initial set is constructed by selecting one of the patterns and choosing a subset of the spikes to activate (n). We jitter these initial spikes using a

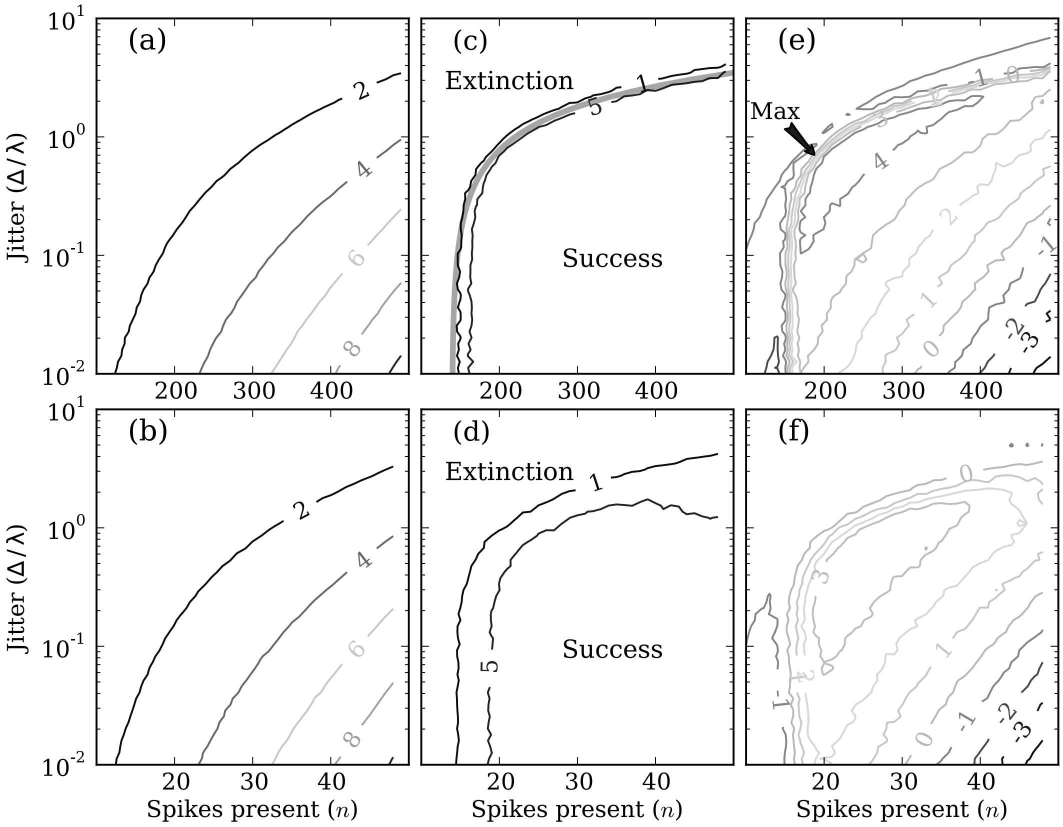


Figure 7: The contrast in spike-time information (I_2) between the recall of one large pattern of 500 spikes (top row), versus recalling 10 small patterns of 50 spikes each (bottom row). The information is a function of n (the number of spikes in the initial cue) and Δ (the amount by which the spikes are jittered). (a)–(b) The information in a given cue before–, and (c)–(d) after– giving it to the network. The difference between initial and final information is shown in (e)–(f). The contours are labeled in units of bits per expected spike. There are 500 expected spikes so if the network provides of 4 bits per spike then the network is providing 2000 bits of total information. The predicted recall boundary is plotted in light grey in (c) and the theoretical maximum information is shown in (e), both agree well with experiment (and are derived in the appendices).

uniform distribution with a width 2Δ centered on the correct time. These initial spikes contain information about the underlying correct patterns and we show the information contained (as a function of n and Δ) in figure 7(a).

The set of spikes in the initial cue is fed into the network and run for 10 periods. The information contained in the network's final period of spikes is then measured (and plotted in the middle column). There is a clear boundary between successful recall and extinction in figure 7(c). This clear boundary is also predicted by (Diesmann et al., 1999) which finds that a single step of neural processing will either increase or decrease the accuracy of the initial cue, and this difference will be further magnified in future rounds of neural processing, leading to complete recall, or extinction.

The difference in information between the initial and final set of spikes represents the ability of the network to recall information. This difference is shown in figure 7(e). In spite of the noise inherent in the network (i.e. spontaneous spikes and transmission noise) there is still a large positive region. This represents the network cleaning up the signal, by providing lower noise and ensuring more spikes are present.

The noisy dendrites account for the negative region in the lower right of figure 7(e). Here the initial set has many spikes with high accuracy. The network is unable to reproduce such a high quality signal and results in a loss of information. With better hardware (less noise in the delay lines, smaller time constant for voltage decay or a lower rate of spontaneous spiking) the network would be able to reproduce a higher quality signal and more information would be communicated.

For the second row of figure 7 we examine the performance of a network with small patterns. We now store 10 times as many patterns ($M = 100$) but each pattern contains 10 times fewer spikes ($G = 50$). This ensures the total number of dendrites is constant. We now select 10 patterns to activate with a comparable number of starting spikes (i.e. the cue has a comparable amount of information – compare figures 7(a) and 7(b).

When starting 10 patterns there is a broader transition between completely successful recall and completely unsuccessful recall as shown in figure 7(d). This is because it is possible for several of the patterns to be unsuccessfully recalled, which is not possible when only a single large pattern is being recalled. To ensure all patterns are reliably recalled we need a better cue, which has an effect on the maximum performance of the network. In figure 7(e) the network is capable of providing approximately 2250 bits of information over the initial starting cue, however, when recalling 10 small patterns the network can only manage 1900 bits. This reduction is in part also due to increased spike-time jitter in small patterns. The typical successful recall of a large pattern results in approximately 3000 bits of information. However successfully recalling 10 smaller patterns typically results in 2750 bits of information. We believe smaller patterns are affected more by individual errant spikes.

While there is a reduction in the maximum performance when measured in individual spike times, there is an increase in the information contained in the state of the network. With 10 large patterns the network can only recall one of them, hence it can only represent 10 states and will only transmit $\log_2 10 \approx 3.3$ bits. However with more numerous smaller patterns the network can recall any 10 of 100 patterns, thus it can represents $\binom{100}{10}$ states and transmit $\log_2 \binom{100}{10} \approx 43$ bits.

It is also useful to know the form of the least informative cue that still results in

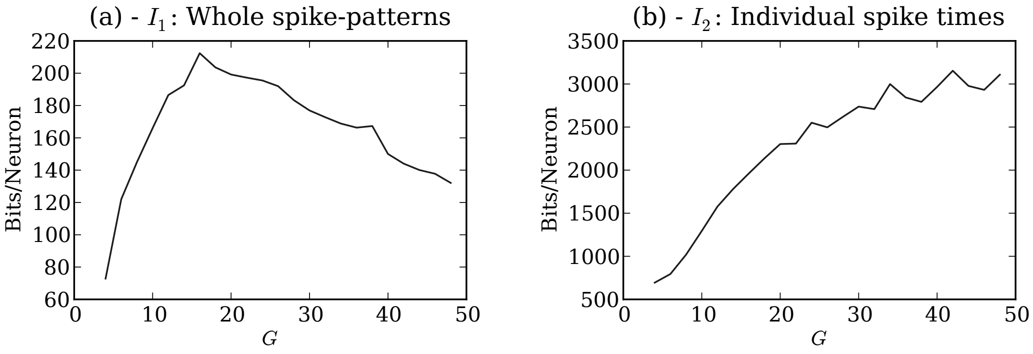


Figure 8: Optimal capacity of the network as a function of the number of spikes per pattern G.

successful recall. We are able to characterize the boundary layer between extinction and successful pattern recall (this is plotted in a gray line in figure 7(c)) and the derivation is given in appendix A. Since we have an analytic form for the information contained in an initial cue we can then find the cue with minimal information that still results in successful recall (the derivation is given in appendix B). This is shown in figure 7(e) and agrees closely with the empirically optimal point.

3.3 Pattern Size

The tradeoff between I_1 and I_2 for patterns of different sizes is more clearly seen in figure 8. Here we created many nets with patterns of different sizes (G). For a given G we search all combinations of patterns stored (M) and simultaneously recalled (m), as shown in figure 6. We plot the maximum of $I_1(G)$ in figure 8(a) and the maximum of $I_2(G)$ in 8(b). We set the number of synapses on a dendrite $g = \max(G-1, 20)$ as the performance did not appear to significantly improve with more synapses, yet the computational costs for $g > 20$ were significant. The maximum possible spike time accuracy would occur in the limit of an infinitely large group of spikes. The network would not achieve arbitrary precision, but would be limited by the equilibrium voltage as well as the inherent noise in the delay lines.

While the information in individual spike times keeps increasing with group size, the information for whole groups of spikes peaks for relatively small groups and then declines. This is as a result of two opposing forces. Small patterns are more likely to turn on spontaneously or fail to turn on correctly; as pattern sizes get larger they also get more reliable. However patterns that are too large begin to negatively affect the total number of patterns that can be stored and recalled. The ideal pattern size will be a function of the equilibrium voltage and the noise in the delay lines, but we did not investigate this further.

3.4 Linear Capacity and Implications for Rate Codes

In figure 9 the capacity of a network is plotted as a function of the period. By doubling the period, the network is able to store approximately double the number of patterns and simultaneously recall double the number of patterns as well. This results in a doubling of the capacity. Similarly if the number of neurons is doubled then twice as many patterns can be stored and recalled. The linear relationship between our information measures for both the number of neurons and the period of the patterns strongly argues that these are natural performance measures for the network.

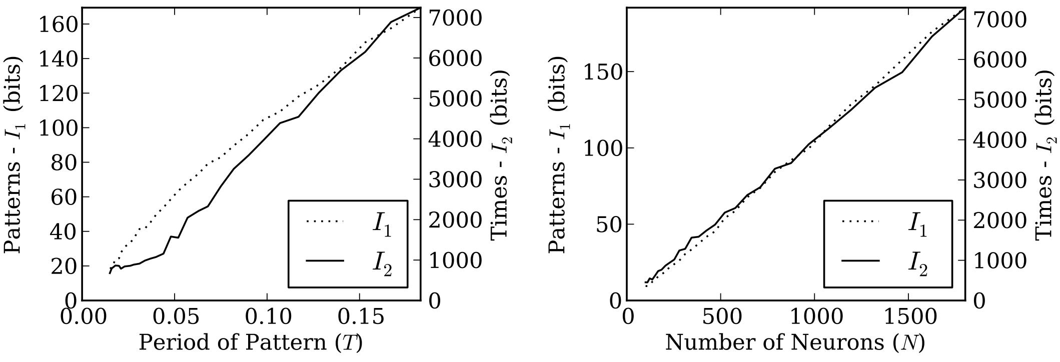
Note that the linear capacity only holds approximately for the period (T). As T increases the transmission noise also increases and equation 10 no longer holds. This means the dendritic voltage decays significantly between the first presynaptic spike and the ideal spike time. Accurate spiking then requires significantly more presynaptic spikes and the relative performance will degrade. It is also unlikely that the linear relationship will hold for periods that are significantly shorter than the voltage decay constant. This is due to the insufficient time between spikes in successive periods. Another aspect that can affect short-period performance are the neglected terms in equation 8, which convey how much entropy there is in the number of recall failures (either spontaneous false recall, or true pattern extinction). For large, reliable networks the entropy terms are dwarfed by the other factors, but can be significant for small networks with parameter regimes where performance is more variable.

A linear relationship means each spike carries the same amount of information regardless of network size or period. This is much better than a naïve rate encoding. In rate encoding the number of spikes a neuron produces in a given time is the quantity of interest. However if the neurons are noisy then there is some uncertainty regarding the intended number of spikes as opposed to the observed number of spikes. If one wants to halve the uncertainty of a neural rate then a four-fold increase in the time is required. This is a result of the standard deviation of the rate estimate growing as $O(\frac{1}{\sqrt{n}})$ (Gerstner et al., 1996). Naïve population codes that function as parallel rate codes are are similarly inefficient.

A much better rate code can be created by modifying our spike-timing codes so that the voltage time constant is approximately the same as the period of the pattern. In this situation the exact time of a spike becomes less important, as it can happen anytime and will (approximately) have the same effect. Such a network will be characterised by the steady spiking rates of neurons. Now the capacity of a rate-coded network can be doubled by doubling the number of neurons (i.e. not a four-fold increase). This characterization also allows an easy comparison between the performance of synfire-like networks and rate-coded networks. With a membrane time constant of 7ms, a network with synfire-like patterns that are 70ms long can code ten times as much as information as is possible in a rate-coded network.

4 Discussion

We will present a brief summary of our results in section 4.1 which leads naturally to a discussion of possible extensions to the work in section 4.1. Finally the main



(a) Linear in the period.

(b) Linear in the number of neurons.

Figure 9: For both measures I_1 and I_2, the network capacity is approximately linear in both period and neurons

conclusions of the work are presented in section 4.3.

4.1 Summary of results

This paper presented a method for encoding information in the relative spike times of a neural network using delay lines. Functionally-separate dendrites were used to simplify the insertion of a new pattern into the network; with each dendrite detecting a single pattern. We showed how to store patterns in the network, and the task was to recall the complete pattern given an initial noisy pattern containing noisy and jittered spikes. The spiking network was capable of cleaning up multiple noisy cues simultaneously. Each possible set of patterns defined a single representation and this paper explored how much information could be recalled in such a representation.

Two measures of information were defined which indicate how much information is recalled with a particular representation. The whole-pattern measure I_1 indicated how many different representations could be recalled, which measures the flexibility of the encoding. The accuracy of individual spikes gave rise to the spike-time measure I_2. Accurate individual spikes could provide information for further processing. For a given pattern size I_2 was maximised by storing a small number of patterns and recalling nearly all of them simultaneously. This is in contrast to I_1 which is maximized by storing many patterns and recalling fewer of them.

If the pattern size is allowed to vary then I_2 performs better with large groups as an individual mis-timed spike has a lesser effect on the entire pattern. I_1 is maximized with a pattern size that is as small as possible yet still reliable, as this allows more patterns to be stored and recalled.

We chose to define a rate code as a spatiotemporal pattern of spikes, in which the period of the pattern is equal to the membrane time constant. In this setting the exact spike time becomes less important than the presence or absence of the spike (although there is still some information contained in spike times, so it is debatable whether this is a true rate code or not). Our results suggest that these rate codes contain less information than spatiotemporal patterns which have a long period. Both measures of network information scale linearly in both the number of neurons and the period of the patterns, lending support to the idea that these are natural measures of network information.

4.2 Future work

In our simulations the noise has been proportional to the length of the delay line. This implies that longer connections are less reliable. In future work it would be interesting to see whether capacity can be increased by giving lower synaptic weights to long delays and higher synaptic weights to small delays. Similarly a neuron that participates in only a few patterns is a more reliable indicator of those patterns than a neuron which participates in many patterns. It is likely that a stronger synaptic strength for quiet neurons would also increase capacity.

Some of the connections we consider are implausibly long (some delays will be nearly 100ms). In Izhikevich (2006) a survey of the literature finds that cortico-cortical connection delays range from 1ms to 35ms. To add biological plausibility one should eliminate excessively-long connections, and see if accurate spike timing in these patterns is maintained. We would predict the accuracy to remain the same, or even improve, since the noise is not allowed to grow, but is instead averaged out over many short intervals. This would introduce another restriction on the number of spikes needed for patterns with long periodicity.

Enforcing shorter delay lines would also enable successful recall for smaller cues. Rather than randomly selecting spikes from a pattern to present as a cue, we could present the beginning of the pattern, and the network would recall the rest of the pattern.

This work can be contrasted with Izhikevich (2006) and Vertes and Duke (2010). Our work uses a different neural model which allows many dendritic compartments, and each compartment codes for a single pattern. It is unclear how our capacity results will transfer to point neuron models, which might suffer from increased interference between patterns. We have also avoided any discussion of how such patterns may be learned, although Izhikevich (2006) finds evidence of polychronous groups spontaneously forming with STDP. It may be the case that it is easier for point models to learn such a spatiotemporal code.

In Faisal and Laughlin (2007) there is a detailed discussion of the types of noise that can affect transmission times. There are many nonlinear effects observed, but biological neurons can deal with transmission time noise of around 10%, as opposed to the 2% used here. In part this extra noise could be averaged out by using larger patterns, but interesting further work would be to investigate the optimal coding strategies in increasing noise conditions. We believe there would be a natural progression from a spike-time code in low noise conditions to a code which is more rate-like in high noise conditions. Rate-like codes also develop when the membrane time-constant is similar to period of the pattern. This suggests that a natural measure of how rate-like the code is might be the ratio of spike-time jitter to the period.

This work can be viewed as extending the work of Diesmann et al. (1999) in several directions; patterns are recalled simultaneously, and the network performance has been given a solid quantitative basis using information theory. However figure 7 shows similar results to Diesmann et al. (1999) figure 3(c) with a small number of accurately-timed spikes able to recall a pattern as well as a large number of less-accurate spikes. By examining the information contained in the initial cue, we are able to find the optimal cue which has minimal information, yet still results in reliable recall. Diesmann et al. (1999) also suggests that temporal accuracy is limited by the upslope of the excitatory postsynaptic potential (EPSP). In our model the upslope is infinite, since the EPSP is modeled with a Dirac delta function. This suggests that temporal accuracy is potentially unbounded and motivates adding delay-line noise. Models with finite upslope of the EPSP may not require delay line noise, and researching this further could provide insight into the factors which limit temporal accuracy.

4.3 Conclusions

The different measures of information correspond to different tasks that might be required of a brain. If one wishes to maximize the amount of information stored in individual spike times (I_2), then one should store only a few patterns, but make them well connected to allow many spikes to accurately determine the next spike.

A task like visual perception requires a large amount of flexibility in the representation, and would require a large amount of information to be encoded in I_1 with the activation of whole patterns (e.g. a single pattern could signal the presence of a coherent edge). In a single visual scene there would be many edges that would need to be simultaneously encoded and transmitted. The edges would still represent a small fraction of all possible edges one might perceive and an encoding that favours I_1 will be more suitable.

One also needs to decide on the required degree of pattern completion. If recall is required from small, noisy cues then the synapses need to be strong and this will limit the maximum number of simultaneously recalled patterns. If the task requires whole-pattern information then the total capacity of the system is reduced.

It should also be noted that although this paper deals exclusively with periodic patterns, there is nothing to stop the same concepts being applied in a layered architecture where a noisy pattern is received by a lower layer, and partially cleaned up as it is passed to a higher layer.

The encoding of information into discrete groups of spatiotemporal spikes is flexible; we can smoothly move between rich “scene” encodings which have small patterns and “spike time” encodings where large groups provide accurate spikes. The amount of information we can recall is also determined by the period of the patterns we store. The smoothness and flexibility suggests that many different tasks could be achieved using such an encoding and hence merits further study.

Acknowledgments

We would like to thank David MacKay, Sebastian Wills, Máté Lengyel, Jean-Pascal Pfister and the Inference group for interesting ideas and discussions. We would also like to thank Emli-Mari Nel and Keith Vertanen for proof-reading drafts of this paper. The detailed reviewer comments also contributed to a significantly more coherent paper. This work was partly funded by a Mandela Cambridge Scholarship and partly funded by LOEWE Neuronale Koordination Forschungsschwerpunkt Frankfurt (NeFF).

A Successful recall boundary

Knowing the boundary for successful recall will enable us to find the cue with minimal information that still results in recall. We are thus interested in finding the transition from extinction to recall as a function of n the number of spikes in the cue and Δ the width about which the spikes are jittered. For a dendrite to successfully spike within λ of the correct time requires that the voltage be driven approximately to:

$$v_{\text{crit}} = -\alpha^{-1} \log(\gamma\lambda), \tag{12}$$

where α and γ are defined in equation 4 on page 5. We start by approximating discrete spikes with a continuous input. After time Δ the dendrite will receive an average continuous input of $\left(\frac{ngw}{G\Delta}\right)$ Vs^{-1}, where $\frac{n}{G}$ is the proportion of activated synapses, g is the number of synapses in the dendrite and $\frac{w}{\Delta}$ is the intensity of incoming voltage. Starting from equilibrium the voltage behaves as:

$$v(t) = \left(\frac{ngw}{G\Delta}\right) \lambda \left(1 - e^{-\frac{t}{\lambda}}\right) + v_{\text{EQ}}. \tag{13}$$

Setting $v(\Delta) = v_{\text{crit}}$, we can obtain the relationship between Δ and n:

$$n = \frac{G(v_{\text{crit}} - v_{\text{EQ}})\left(\frac{\Delta}{\lambda}\right)}{gw\left(1 - e^{-\frac{\Delta}{\lambda}}\right)}. \tag{14}$$

This boundary has been plotted in figure 7(c) as a thick grey line and lies almost precisely over the boundary.

B Maximum Information Recalled

If the boundary of successful recall is known then it is possible to determine the cue with the smallest information which will still lead to successful recall. This means the network is filling in the most information and is operating at its most efficient.

If n spikes are present and have a jitter of width Δ then the information (from equation 9 on page 10) is:

$$I_2 = n \log_2\left(\frac{r^+}{r}\right) + (G - n)\log_2\left(\frac{r^-}{r}\right) \tag{15}$$

The missing spikes could occur at any time so that the low rate of spiking is given by:

$$r^- = \frac{G - n}{NT}. \tag{16}$$

The jittered spikes each occur within a window Δ wide and will also contain some of the missing spikes, which makes the high rate of spiking:

$$r^+ = \frac{1}{\Delta} + \frac{G - n}{NT}. \tag{17}$$

The completely ignorant rate is :

$$r = \frac{G}{NT}. \tag{18}$$

Substituting the above equations and equation 14 into equation 15, and differentiating with respect to Δ we obtain a lengthy equation which we are able to solve numerically. This maximum is plotted in figure 7(e) and is close to the empirical maximum.

C List of Symbols

Symbol	**Default**	**Description**
G	50	Number of spikes in a Group (a.k.a. pattern,memory)
g	20	Number of synapses in a dendrite ($< G$ as self-connections are not allowed)
M		Number of Memories stored in the network
m		The number of patterns actively recalled
N	1000	Number of Neurons
w	2.0 V	Synaptic strength - when a spike is received the voltage of the dendrite is increased by s
T	0.1 s	Period of a repeating memory (It also represents the maximum delay in synaptic transmission)
λ	0.007 ms	Time constant of voltage decay
σ	0.02ϕ	Standard Deviation of delays
Δ		Maximum jitter time
ϕ_{ij}		Time delay of the jth synapse on the ith dendrite

References

Abeles, M. (1991). *Corticonics: Neural circuits of the cerebral cortex.* Cambridge University Press.

Barlow, H. (1985). Cerebral cortex as model builder. *Models of the visual cortex*, pages 37–46.

Bienenstock, E. (1995). A model of neocortex. *Network: Computation in Neural Systems*, 6(2):179–224.

Diesmann, M., Gewaltig, M., and Aertsen, A. (1999). Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 402(6761):529–533.

Faisal, A. and Laughlin, S. (2007). Stochastic simulations on the reliability of action potential propagation in thin axons. *PLoS Comput Biol*, 3:783–795.

Faisal, A., Selen, L., and Wolpert, D. (2008). Noise in the nervous system. *Nature reviews. Neuroscience*, 9(4):292.

Gerstner, W., Kempter, R., Van Hemmen, J., and Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383(6595):76–78.

Gerstner, W., Ritz, R., and van Hemmen, J. (1993). Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns. *Biological Cybernetics*, 69(5):503–515.

Golding, N., Staff, N., and Spruston, N. (2002). Dendritic spikes as a mechanism for cooperative long-term potentiation. *Nature*, 418(6895):326–331.

Hopfield, J. J. (1995). Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376(6535):33–36.

Izhikevich, E. (2006). Polychronization: Computation with spikes. *Neural Computation*, 18(2):245–282.

König, P., Engel, A., and Singer, W. (1996). Integrator or coincidence detector? The role of the cortical neuron revisited. *Trends in Neurosciences*, 19(4):130–137.

Liu, G. (2004). Local structural balance and functional interaction of excitatory and inhibitory synapses in hippocampal dendrites. *Nature Neuroscience*, 7(4):373–379.

London, M. and Häusser, M. (2005). Dendritic computation. *Annual Review of Neuroscience*, 28(1):503–532.

Polsky, A., Mel, B., and Schiller, J. (2004). Computational subunits in thin dendrites of pyramidal cells. *Nature Neuroscience*, 7(6):621–7.

Simoncelli, E., Paninski, L., Pillow, J., and Schwartz, O. (2004). Characterization of neural responses with stochastic stimuli. *The Cognitive Neurosciences*, pages 327–338.

Softky, W. (1994). Sub-millisecond coincidence detection in active dendritic trees. *Neuroscience*, 58(1):13–41.

Vertes, P. and Duke, T. (2010). Effect of network topology on neuronal encoding based on spatiotemporal patterns of spikes. *Human Frontier Science Program*, 4(3-4):153–163.

Wills, S. (2004). *Computation with spiking neurons*. PhD thesis, Cambridge University.