# Information Capacity of Binary Weights Associative Memories

Arun Jagota
Department of Computer Science
University of California,
Santa Cruz CA 95064
jagota@cse.ucsc.edu

Giri Narasimhan
Mathematical Sciences
University of Memphis
Memphis TN 38152
giri@next1.msci.memst.edu

Kenneth W. Regan
University at Buffalo
Buffalo NY 14260
regan@cs.buffalo.edu

September 4, 1997

### Abstract

We study the amount of information stored in the fixed points of random instances of two binary weights associative memory models: the Willshaw Model (WM) and the Inverted Neural Network (INN). For these models, we show divergences between the *information capacity* (IC) as defined by Abu-Mostafa and Jacques, and information calculated from the standard notion of *storage capacity* by Palm and Grossman respectively. We prove that the WM has asymptotically optimal IC for nearly the full range of threshold values, the INN likewise for constant threshold values, and both over all degrees of sparseness of the stored vectors. This is contrasted with the result by Palm, which required stored random vectors to be logarithmically sparse to achieve good storage capacity for the WM, and with that of Grossman, which showed that the INN has poor storage capacity for random vectors. We propose Q-state versions of the WM and the INN, and show that they retain asymptotically optimal IC while guaranteeing stable storage. By contrast, we show that the Q-state INN has poor storage capacity for random vectors. Our results indicate that it might be useful to ask analogous questions for other associative memory models. Our techniques are not limited to working with binary weights memories.

**Key words:** Information Capacity, Associative Memories, Graph Counting, Binary Weights, Hopfield Model, Neural Networks.

# 1 Introduction

Abu-Mostafa and Jacques [1] introduced the following concept.

**Definition 1** *The* Information Capacity *(IC) of a memory model is the logarithm (base 2) of the number of its instances that store* distinct *collections of memories.*

The term *instance* in the above definition (and throughout the paper) denotes a configuration of the memory model (for example a neural network with its parameters—weights and thresholds—fixed to certain values). The IC measures the entropy of the memory model under the uniform distribution on its instances, that is, the number of bits of information in a random instance of the memory. As one example [1], consider an $n$-bit memory model in which each location is independent of others. All $2^n$ instances of this model store distinct $n$-bit vectors, hence its information capacity equals $n$, which is the intuitive answer.

The concept of information capacity is useful for memory models in which not all realizable instances store distinct collections of memories. In such cases, the IC, not the logarithm of the total number of realizable instances, gives the true information in a random instance. In this sense, the IC is to such memory models what VC-dimension is to feedforward neural networks: both measure the intrinsic "richness" of a particular architecture.

As an example, consider the discrete Hopfield neural network [15]. For the purposes of associative memories, it is normal to consider the memories in this model to be stored only in its *fixed points.* (The term fixed point as used here is synonymous with *stationary point*, *steady state*, *equilibrium point*, or *local minimum* of network's energy function.) Therefore, it is useful to define two $n$-neuron instances $N_1, N_2$ of this model to be *distinct* if and only if they have differing collections of fixed points, and compute its information capacity under this definition. This is the definition of Information Capacity we employ in the current paper. Such a calculation is not necessarily easy, since many instances may realize the same collection of fixed points.

As a simple example of the utility of this definition, consider storing memories in a Hopfield network employing -1/+1 units, using the Hebb rule [14]. All stored memories can be made stable by making the self-weights sufficiently positive. However, in this case, every network thus constructed has the same collection of fixed points: all the $2^n$ bipolar vectors of length $n$. Hence the information capacity of this family of networks, according to this definition, is zero.

**Relationship to Other Measures of Capacity.** The capacities of various neural associative memories have been studied extensively [28, 1, 25, 21, 10, 2, 6, 9]. The capacity definitions employed most frequently [28, 25, 2, 9] are instances of the following general one, which is, loosely-stated:

**Definition 2** *The* storage capacity *is the largest number of randomly selected memories (binary vectors, or vector pairs) that can be stored so that, for sufficiently large number of neurons $n$, the probability that all (or nearly all) memories are stable (or nearly stable) tends to one.*

From the point of view of the current paper, the different instances of the above general definition have minor differences, and all give very similar results—results, as we shall see, that can differ strikingly from those given by the IC definition.

A related definition, which has also found extensive use, requires that, in addition to the memories being stable, they must also have sufficiently large basins of attraction. Let us refer to these definitions as *stable storage capacity* and *basins capacity* respectively.

The importance of calculating the stable storage and basins capacities of memory models cannot be overstated. These calculations are often done for specific storage rules for the memories [29, 25, 2] (for example, the Hebb rule). From these results, the amount of information stored in an instance of the model, arising from storing random vectors using a particular storage rule, can also often be deduced. For example, it is well known (see [25, 14]) that order of $n/\log n$ $n$-bit random vectors can be stored in the discrete Hopfield memory using the Hebb rule so that, with probability tending to one, every stored vector is stable and has a large basin of attraction. The amount of useful information stored in the fixed points of such a network is then roughly the amount of information in such a collection of vectors. Each random $n$-bit vector has order of $n$ bits of information. Since the size of the collection is small, and the vectors independent, the information can be added to give the cumulative information in the collection as $n^2/\log n$. As a second example, Palm [28] showed that the amount of information stored in the $n$-unit binary (0/1) weights Willshaw model, using a particular storage rule [32], is maximized when each of the stored random vectors contains order of $\log n$ ones. In particular he showed that order of $n^2/\log^2 n$ random vectors of such sparseness can be stored usefully in this memory model. An $n$-bit random vector containing $\Theta(\log n)$ ones has $\Theta(\log^2 n)$ bits of information [2]. The amount of information in a resulting instance is thus $\Theta(n^2)$ bits [28]. (The notation $\Theta(f(n))$ denotes an unspecified function with the same asymptotic growth rate as that of $f(n)$, an increasing function.) Palm's analysis [28] appears to implicitly indicate that the amount of information drops significantly if the stored random vectors are denser. Amari [2] extended the results of Palm as follows. He replaced the binary-weights Willshaw model with the real-valued weights discrete Hopfield model, and employed the Hebb rule for storage. He showed the stable storage capacity to be of the order

$$C(\epsilon) = \begin{cases} n^{2-\epsilon}/\log n & \text{if the stored vectors have order of } n^{\epsilon} \text{ ones, for any } \epsilon : 0 < \epsilon \leq 1 \\ n^2/\log^2 n & \text{if the stored vectors have order of } \log n \text{ ones} \end{cases}$$

Therefore the stable storage capacity is maximized for order of $\log n$ ones (a result consistent with the one of Palm) but degrades gracefully as the density of the ones increases (a result that extends the one of Palm). Amari [2] calculated the information stored in the resulting network instance as of the order

$$C_I(\epsilon) = \begin{cases} n^2/\log n & \text{if the stored vectors have order of } n \text{ ones} \\ (1-\epsilon)n^2 & \text{if the stored vectors have order of } n^{\epsilon} \text{ ones, } 0 < \epsilon < 1 \\ n^2 & \text{if the stored vectors have order of } \log n \text{ ones} \end{cases} \tag{1}$$

This shows that the information remains proportional to the number of links $n^2$ as the density of the vectors is increased from $\log n$ to $n^{\epsilon}$ for $\epsilon < 1$. Only when the density of the ones is of the order $n$ (i.e., $\epsilon = 1$) does the information drop a little. This extends Palm's results to the real-valued weights model.

Interesting and useful as the above results are, there are two (related) ways in which information calculated this way differs from that calculated from the IC definition employed in the current paper.

First, the information in the former case is calculated for a specific storage rule. The IC definition on the other hand calculates information for the architecture, hence gives an upper bound on that realizable by any storage rule. That this distinction is a useful one is given by the fact that, for the discrete Hopfield network, the information realizable by the Hebb rule, as noted earlier, is of the order $n^2/\log n$, whereas the upper bound given by the IC is $n^2$ [1].

Second, the former calculation gives the information stored in an instance arising from a random collection of stored vectors; the IC on the other hand gives the information stored in a random instance of the memory. For some memory models, both calculations give identical results. For others they give widely different results, as we show in this paper.

**Overview of Results.** In this paper, we study two binary-weights neural associative memory models: the *Willshaw Model* (WM) [32], and the *Inverted Neural Network* (INN) [31]. Both may be viewed as special cases of the discrete Hopfield model [15]. The Willshaw model has remained an interesting topic of study since its original design, largely because of its simplicity and because it has good capacity for sparse patterns [14]. The model is simple enough to permit theoretical analyses [28], yet rich enough to capture the essential properties of associative memories. The model is also very attractive from the point of view of hardware implementation [14]. The INN model was developed specifically for optical implementation [31]. A 64-neurons prototype was developed [31]. Previous results seem to indicate that both the Willshaw Model and the INN have poor capacity for non-sparse patterns [28, 12]. Our main results in the current paper are that, in fact, from the Information Capacity point of view, both models have very good capacity.

To preview these results more precisely, from each model, we define a family of models indexed by the threshold value $t$ to a neuron (the same for each neuron). We call these families WM($t$) and INN($t$) respectively. We show that WM($t$) has an asymptotically optimal IC, order of $n^2$, over almost the entire range of reasonable values of $t$ (integer $t \geq 3$ to $t < n/2$). This contrasts with the results of Palm, who required that $t$ be of the order $\log n$, for the stored information derived from storage capacity calculations to be asymptotically optimal [28]. How the asymptotically optimal IC we get for non-logarithmic $t$ can be exploited in practice is a separate question.

We show that INN($t$) has optimal IC of $\binom{n}{2}$ at $t = 0$. At negative integer values of $t$, the IC is at least of the order $\binom{n/(|t|+1)}{2}$, which is asymptotically optimal (i.e., of the order $n^2$) for constant $t$. The exact optimality of the result for $t = 0$ is significant because realizable networks are clearly finite (usually small). Our result contrasts with that of Grossman [12], who showed that the stable storage capacity (and the resulting information derived from it) of INN for $t = 0$ was poor.

We next generalize the vectors to be stored from binary (0/1) to $Q$-state (i.e., a vector $v \in \{0, 1, \ldots, Q-1\}^n$). We have shown earlier that, for any positive integer $Q \geq 2$, an *arbitrary* collection of $Q$-state vectors can be stored in a binary-weights Q-state extension of INN(0), so that all vectors are fixed points [17]. In the current paper, we show that the IC of the Q-state INN(0) model remains asymptotically optimal, that is order of $N^2$, where $N = Q \times n$ is the number of neurons, in the entire range of admissible values of $Q$. As a striking contrast, we show that the stable storage capacity of our storage scheme, for constant Q, is at most of the order $\log n$, which gives the information stored in a resulting Q-state INN instance as at most of the order $n \log n$. Thus, information calculated from two different definitions gives strikingly different results. In particular, a random instance of the Q-state INN(0) model has much more information than an instance emerging from storing random Q-state vectors.

Q-state associative memories have been studied in the past [30, 23]. Rieger [30] extended the two-state Hopfield model to Q-states, by using Q-state neurons. He showed that the stable storage capacity, for random Q-state vectors of length $n$, dropped to $\Theta(n)/Q^2$; hence the information stored in such a network to $\Theta(n) \log_2 Q/Q^2$. Kohring [23] modified Rieger's Q-state model and improved the stable storage capacity to $\Theta(n)/\log_2 Q$; hence the information stored in such a network to $\Theta(n) \log_2 Q/\log_2 Q$, that is, order of $n^2$ bits. Kohring's modification involved recoding a Q-state vector of length n, using $\log_2 Q$ bits for each component value (0 through $Q-1$), and creating $\log_2 Q$ independent attractor networks, to process the bits. This scheme uses $N = n \times \log_2 Q$ neurons overall, and $\log_2 Q \times n^2$ weights. Hence the information stored per weight is of the order $1/\log_2 Q$ bits.

Our Q-state scheme in this paper, based on the INN(0) model and presented in Section 4 (also see [17]), uses $Q$ binary neurons for the Q states. The total number of neurons is thus $N = Q \times n$. We achieve an IC of order $N^2$, which gives one bit of information stored per weight. Furthermore, we give a guarantee that for an arbitrary collection of stored Q-state vectors, every one of them is a fixed point. More recently, another Q-state model is proposed in [24], which also employs $Q$ binary neurons for the Q states (hence $N = Q \times n$ neurons in total). The representation and weight matrix calculation is different however, and stable storage is not guaranteed for an arbitrary collection of stored Q-state vectors.

Next, we restrict the instances of INN(0) to those in which all fixed points are sparse. We show that the IC remains asymptotically optimal, order of $N^2$, where $N$ is the number of neurons, for almost all degrees of sparseness. We give a similar result for the Willshaw model: the IC remains asymptotically optimal for almost all degrees of sparseness. This result contrasts with that of Palm [28], who required the stored random vectors to be logarithmically sparse, in order for the stored information to be asymptotically optimal.

Earlier results on stable storage, basins, or information capacity have relied on statistical [28, 2], coding theory [25], or threshold function counting [1] arguments. By contrast, all our results in the current paper are based on characterizations of the fixed points and graph counting. In the case of the Willshaw model, we characterize the fixed points as certain kinds of induced subgraphs. This characterization is not only useful for our IC results, but also leads to a storage rule for the Willshaw model which guarantees that an *arbitrary* collection of $Q$-state vectors can be stored with every stored vector a fixed point, while retaining asymptotically optimal IC.

Finally, the results in this paper strongly suggest that analogous questions on other associative memory models be studied. As mentioned earlier, most previous results on other associative memory models are based on some variant of the notion of storage capacity, as given by definition 2. Obtaining results using the Information Capacity definition (definition 1) gives a more rounded picture of the capacity of an associative memory model. The techniques we employ in this paper might be useful for similar studies of other associative memory models. The techniques are applicable, without qualification, to real-valued weights models also.

This paper is organized so that all the results are presented first (and some short proofs that do not involve the tools of graph theory). The proofs that rely on graph-theoretic arguments and notation are postponed to a later section, where some useful concepts from graph theory are introduced first, and then the proofs given.

## 2 The Associative Memory Models

The Associative Memory Models that we study here have their roots in [32, 3, 22, 26, 15]. The two models we study in this paper employ binary-valued weights [32, 31], and are restricted to the auto-associative case. Both may be described as special cases of the Hopfield model [15]. This model is composed of $n$ McCulloch-Pitts formal neurons $1, \ldots, n$, connected pair-wise by weights $w_{ij}$. Throughout this paper we will assume that the weights are *symmetric*, that is, $w_{ij} = w_{ji}$. We will use the notation $w_{ij}$ to refer to the single undirected weight between neurons $i$ and $j$ (rather than from $i$ to $j$). The self-weights $w_{ii}$ equal zero. Each neuron has a state $S_i \in \{0, 1\}$. The network evolves according to

$$\mathbf{S}(n+1) := \mathrm{sgn}(W\mathbf{S}(n) - \theta) \tag{2}$$

where sgn is the usual signum function applied component-wise ($\mathrm{sgn}(x) = 1$ if $x \geq 0$; $\mathrm{sgn}(x) = 0$ otherwise) and $\theta = (t)$ is the vector of thresholds of equal value $t$. In this paper, our interest is only in the fixed points $\mathbf{S} = \mathrm{sgn}(W\mathbf{S} - \theta)$. Our results will hold for both synchronous as well as asynchronous updates of (2).

Though the Willshaw model (WM) and the inverted neural network (INN) have the *same* architecture, as described above, they are characterized by a *different* set of weights.

$$\begin{array}{lll} \text{In the WM,} & \text{for all } i \neq j, & w_{ij} \in \{0, 1\}. \\ \text{In the INN,} & \text{for all } i \neq j, & w_{ij} \in \{-1, 0\}. \end{array}$$

(The INN weights described above are based on an equivalent reformulation of the INN description in [31], to make it use the same activation function as does the WM.)

We define WM($t$) and INN($t$) as the family of these two models respectively indexed by the threshold value $t$ to a neuron (the same for each neuron). Though the WM and INN architectures are identical and their weights very similar, they will turn out to have information capacities that differ at certain extremes of the neuronal thresholds $t$.

## 3 Information Capacity of the WM and the INN Models

The results of this section give the information capacity of the Willshaw model and of the inverted neural network model. These results quantify the inherent information capacities of the *models* (characterized by architecture and feasible weights as described in the previous section), not of particular mechanisms for storing memories. Particular mechanisms for storing memories may constrain the feasible weight-spaces even further, so that the resulting information capacities can only be lower or the same, not higher.

**Lemma 1** *When the neuronal threshold $t$ is a positive odd integer with $t+1$ divisible by $n$, a lower bound on the number of WM(t) instances with different collections of fixed points is* $2^{\binom{(t+1)/2}{2}(n/(t+1)-1)^2}$.

**Theorem 2** *When the neuronal threshold $t$ is a positive odd integer with $t+1$ divisible by $n$ and $3 \leq t \leq n/2-1$, the IC of WM(t) is* $\Theta(n^2)$.

**Proof :** For constant $t \geq 3$, $\binom{(t+1)/2}{2} \geq 1$ and $(n/(t+1)-1)^2 = \Theta(n^2)$. For $t$ increasing with $n$ and $t \leq n/2-1$,

$$\binom{(t+1)/2}{2}(n/(t+1)-1)^2 = \Theta(t^2)\Theta(n^2/t^2) = \Theta(n^2)$$

■

**Lemma 3** *When the neuronal threshold $t$ is a negative integer with $|t|$ divisible by $n$, a lower bound on the number of INN(t) instances with different collections of fixed points is* $2^{\binom{n/(t+1)}{2}}$.

This gives the IC of INN(t) as $\geq \binom{n/(t+1)}{2}$, which is $\Theta(n^2)$, asymptotically optimal for constant $t$, and $\binom{n}{2}$, exactly optimal for $t=0$ (the $t=0$ result is also in [16]).

In earlier work [20], we came up with a different complicated expression for a lower bound on the IC of INN(t), whose value decreased monotonically with $t$. This gave the IC of INN(1) as $\Omega(n \log n)$, and that of INN($n/2-1$) as $\Omega(\log \binom{n}{n/2})$. For $t = o(\sqrt{n})$, Lemma 3 is an improvement; for $\sqrt{n} = o(t)$, the result in [20] is better. Overall, the lower bound given by Lemma 3 decreases more gracefully when $t$ increases from 0. (Since two of the authors of [20] are not amongst those of the current paper, it is useful to note that, other than reference to the above result, none of the new ideas, results, or techniques from [20] are used in the current paper.)

### 3.1 Information Capacity Under Particular Storage Rules

The IC results of WM(t) and INN(t) of the previous section, as noted there, apply to the models, not to any particular memory-storage mechanisms for the models. We now examine one particular memory-storage mechanism for the INN and one for the WM and show that the ICs of the INN(0) and the WM(t) under their respective storage mechanisms are not reduced whatsoever from the original ICs.

Consider the following storage rule for the INN(0) model, developed in the context of optical implementation [31], and independently for its attractive associative memory properties [17]. Initially, $w_{ij} = -1$ for all $i \neq j$. A sequence $X^1, \ldots, X^m$ of vectors in $\{0,1\}^n$ is stored as follows. To store $X^\mu$, for all $i \neq j$:

$$w_{ij} := \begin{cases} 0 & \text{if } X_i^\mu = X_j^\mu = 1 \\ w_{ij} & \text{otherwise} \end{cases} \tag{3}$$

Grossman showed that the stable storage capacity of this storage rule for INN(0), for random sparse vectors, is at most $\log_2 n$ [12]. This puts a (weak) upper bound of $n \log_2 n$ on the number of bits of information stored in such a network. It is easy to see that, for each of the $2^{\binom{n}{2}}$ INN(0) instances $\mathcal{N}$, there exists some collection of binary vectors, which when stored via (3), gives the network instance $\mathcal{N}$. By Lemma 3, the IC of INN(0), even for this particular storage rule, is $\binom{n}{2}$, an optimal result that contrasts strikingly with the stable storage capacity result of Grossman.

Consider the following storage rule for the WM(t) model [32, 28]. Its attractive features are simplicity, and the fact that it works well, for suitable $t$, on sparse patterns [14]. Initially, $w_{ij} = 0$ for all $i \neq j$. A sequence $X^1, \ldots, X^m$ of vectors in $\{0,1\}^n$ is stored as follows. To store $X^\mu$, for all $i \neq j$:

$$w_{ij} := \begin{cases} 1 & \text{if } X_i^\mu = X_j^\mu = 1 \\ w_{ij} & \text{otherwise} \end{cases} \tag{4}$$

**Lemma 4** *The result of Theorem 2 is unchanged under storage rule (4).*

Palm showed that, when random vectors are stored in the network using storage rule (4), order of $n^2$ bits of information are stored in the resulting network only if the vectors contain order of $\log_2 n$ ones, and the threshold $t$ is set to order of $\log_2 n$ [28]. By contrast, Lemma 4 shows that the IC is order of $n^2$ bits, under storage rule (4), for almost every value of the threshold.

Finally, it is to be noted that though the architectures, feasible weight-sets, and the storage rules of this section are very similar for both the WM and for the INN, the two network instances produced for the same set of stored patterns can have different fixed points (hence different associative memory properties). For more details on this, see [27] which shows experimentally that the two networks, the WM and the INN, behave differently on the same set of stored patterns. The same paper also characterizes a theoretical condition for stability in the WM—a condition that reveals in which situations the WM has a different set of fixed points than does the INN, when patterns are stored according to the storage rules of this section.

## 4 Q-state Vector Storage and Information Capacity

### 4.1 Q-state Vector Storage in INN

In this paper, we generalize (3) to store Q-state vectors in an INN(0) instance. Q-state vectors are stored in a neural grid of $N = Q \times n$ neurons. A neuron is indexed as $(q, i)$ where $q \in \{0, \ldots, Q-1\}$

and $i \in \{1, \ldots, n\}$. The storage rule is a generalization of (3) [17]. Initially, $w_{(q_1,i_1),(q_2,i_2)} = -1$ for all $(q_1, i_1) \neq (q_2, i_2)$. A sequence $X^1, \ldots, X^m$ of vectors $X^i \in \{0, \ldots, Q-1\}^n$ is stored by presenting each vector sequentially as follows. To store $X^\mu$, for all $(q_1, i_1) \neq (q_2, i_2)$:

$$w_{(q_1,i_1),(q_2,i_2)} := \begin{cases} 0 & \text{if } X^\mu_{i_1} = q_1 \text{ and } X^\mu_{i_2} = q_2 \\ w_{(q_1,i_1),(q_2,i_2)} & \text{otherwise} \end{cases} \tag{5}$$

Figure 1 illustrates storage rule (5) on some Q-state vectors.

**Theorem 5** ([17]) *In any collection of Q-state vectors stored using storage rule (5), all stored vectors are fixed points of the network.*

Notice that when $Q = 2$, this result states that any collection of $n$-bit binary vectors can be stored stably in a $2n$-unit network. By contrast, there are collections of at most $n$ vectors (in fact at most three [10]) that cannot be stored stably in an $n$-unit network (no matter what the storage rule is and even if the weights are real-valued) [1].

Define the *Q-state INN model* as one composed of $N = Q \times n$ neurons, arranged in a grid, in which every instance of the model arises from storing some collection of $Q$-state vectors of length $n$ using storage rule (5). The only modifiable weights in this model are the ones crossing columns (the associated units have different second components $i$). The Q-state INN model comprises of a subset of the N-unit instances of INN(0). An upper bound on the IC of the Q-state INN model is $Q^2\binom{n}{2}$, the number of modifiable weights. The following result gives an asymptotically optimal lower bound.

**Lemma 6** *A lower bound on the number of Q-state INN model instances with different collections of fixed points is* $2^{(Q-1)^2\binom{n}{2}}$.

**Theorem 7** *The Information Capacity of the Q-state INN model is* $\Theta(N^2)$ *for all* $Q = 2, \ldots, N/2$.

**Proof :** For constant $n \geq 2$, $\binom{n}{2} \geq 1$ and $(Q-1)^2 = \Theta(N^2)$. For $n$ increasing with $N$ and $n \leq N/2$,

$$\binom{n}{2}(Q-1)^2 = \Theta(n^2)\Theta(Q^2) = \Theta(N^2)$$

■

This result shows that we are able to store arbitrary collections of $Q$-state vectors stably, without any asymptotic loss of Information Capacity, in the entire range of admissible values of $Q$.

### 4.2 Q-state Vector Storage in WM

The constructive proof of Lemma 1 (see Section 7) has led us to a storage rule for Q-state vectors of length n in the WM(2n-1) model, using a grid of $N = Q \times 2n$ neurons. The storage rule generalizes (4). Notice that the threshold $t$ is set equal to $2n - 1$. A neuron is indexed as $(q, i)$ where $q \in \{0, \ldots, Q-1\}$ and $i \in \{1, \ldots, 2n\}$. Initially, $w_{(q_1,i_1),(q_2,i_2)} = 0$ for all $(q_1, i_1) \neq (q_2, i_2)$. A Q-state vector $X^i = (x_1, x_2, \ldots, x_n)$, $x_j \in \{0, \ldots, Q-1\}$, of length $n$ is recoded as a vector $Y^i = (x_1, x_1, x_2, x_2, \ldots, x_n, x_n)$ of length $2n$. A sequence $Y^1, \ldots, Y^m$ of recoded vectors $X^1, \ldots, X^m$ is stored by presenting each vector $Y^\mu$ sequentially as follows. To store $Y^\mu$, for all $(q_1, i_1) \neq (q_2, i_2)$:

$$w_{(q_1,i_1),(q_2,i_2)} := \begin{cases} 1 & \text{if } Y^\mu_{i_1} = q_1 \text{ and } Y^\mu_{i_2} = q_2 \\ w_{(q_1,i_1),(q_2,i_2)} & \text{otherwise} \end{cases} \tag{6}$$

Figure 2 illustrates storage rule (6) on some Q-state vectors.

**Theorem 8** *In any collection of Q-state vectors stored using storage rule (6), all stored vectors are fixed points of the resulting network.*

When $Q = 2$, this result states that any collection of binary vectors of length $n$ can be stored stably in a $4n$-unit Willshaw model network by recoding the vectors by duplicating each component, and by setting the threshold $t$ to $2n - 1$. In contrast to this, for any positive threshold $t$, we can easily find collections of two binary vectors of length $n$ that are not stored stably in an $n$-unit Willshaw model network with threshold $t$, using storage rule (4). In particular, fix a vector $x \in \{0,1\}^n$ containing at least $t$ ones and generate a vector $y$ from $x$ by adding some ones to the components of $x$. Then if $x$ and $y$ are stored, $x$ becomes unstable.

Define the Q-state WM$(2n-1)$ model as one composed of $N = Q \times 2n$ neurons, arranged in a grid, in which every instance of the model arises from storing some collection of Q-state vectors of length $n$, using the above storage rule (6). Note that the modifiable weights in the model come in groups of four, linking quadruplets of vertices. An upper bound on the IC of the Q-state WM$(2n-1)$ model is thus $Q^2\binom{n}{2}$, the number of independently modifiable quadruplets of weights. The following result gives an asymptotically optimal lower bound.

**Theorem 9** *The Information Capacity of the Q-state WM($2n-1$) model is* $\geq (Q-1)^2\binom{n}{2} = \Theta(N^2)$.

As for Q-state INN, this result shows that we are able to store arbitrary collections of $Q$-state vectors, of length $n$, stably in an WM$(2n-1)$ instance, without any asymptotic loss of IC, in the entire range of admissible values of $Q$.

## 5 Saturation Analysis

In this section, we show that when random Q-state vectors are stored in the INN model, the stored information is very low. This result is in striking contrast with the results of the previous section.

Define a $Q$-state INN instance as *saturated* if every $Q$-state vector of length $n$ is a fixed point of the network.

**Proposition 10** *For any fixed* $\epsilon > 0$, *and for any* $Q \geq 2$, *the probability that presenting* $\log_{Q^2/(Q^2-1)} \binom{n}{2} Q^2 n^\epsilon$ *Q-state vectors of length n creates a saturated Q-state INN instance is at least* $1 - 1/n^\epsilon$, *where the vectors are chosen independently and uniformly at random.*

For constant $Q$ independent of $n$, Proposition 10 shows that $O(\log n)$ random Q-state vectors of length $n$ saturate a Q-state INN instance with high probability. Thus, for constant Q, at most $O(n \log n)$ bits of information are stored in a $Q$-state INN instance arising from storing random $Q$-state vectors. This contrasts with Lemma 6, which showed that $\Theta(n^2)$ bits of information are stored in a random $Q$-state INN instance, for constant Q. Notice however that the upper bound in Proposition 10 increases significantly with $Q$, as the base of the logarithm is $Q^2/(Q^2-1)$.

This opens the question of whether recoding binary vectors ($Q$=2) of length $n$ as Q-state vectors, $Q$ large, increases the number of vectors stored before saturation occurs. The following recoding ideas came from discussion with Bar-Yehuda [4]. Define a k-recoding of a binary vector of length $n$ as a recoding to a Q-state vector, $Q = 2^k$, of length $n/k$ by dividing the $n$ bits of the binary vector into $n/k$ blocks of $k$ bits each. As an extreme case, if we choose $k = n/2$, it may be checked that, by applying our Q-state INN storage rule (5), we can store any collection of $2^{n/2}$-state vectors of length 2 perfectly in a $2^{n/2}$-state INN instance. As an intermediate case, consider $k = \log_2 n$. Then $Q = n$ and a Q-state recoding of an n-bit vector has length $n/\log_2 n$. Thus the bound in Proposition 10 becomes, for $\epsilon = 1$,

$$\log_{n^2/(n^2-1)} \binom{n/\log_2 n}{2} n^2 n \geq n^2$$

This opens the question of whether at least order of $n^2$ random $n$-state vectors of length $n/\log n$ can be stored in such a network, which uses $n^2/\log_2 n$ neurons, before saturation occurs. Answering these questions will require us to obtain a lower bound on the number of random vectors required to saturate a Q-state INN instance with high probability.

Even if saturation capacity were to provably improve with recoding of binary vectors with large $Q = 2^k$, the cost of this must be considered. First, the network size increases to $2^k \times n/k$ neurons. Second, the recoding makes the representation less distributed since a sequence of $k$ bit values in an original binary vector is represented by one neuron.

By contrast, recall that the choice of Q has no bearing on the IC, which remains asymptotically optimal at $\Theta(N^2)$, where $N$ is the number of neurons in a Q-state INN instance, for arbitrary Q. This underscores the importance of the definition in measuring information capacity. It also indicates that, while the network has poor capacity for random Q-state vectors for constant Q, it has inherently good information capacity (by our definition). How to exploit this is a separate issue.

# 6 Sparse Coding and Information Capacity

Sparse coding has been suggested as a mechanism to alleviate the poor storage capacity of neural associative memories [32, 28, 2, 5]. Though our previous results in this paper indicate that sparse coding is not necessary to retain high information capacity, by our definition of information capacity, it is useful to calculate whether sparse coding is sufficient.

## 6.1 The INN Model

Consider first the INN model. We shall restrict our analysis to that of INN(0). Define the *k-sparse* INN(0) model as the set of $n$-unit INN(0) instances in which every fixed point of the instance (an $n$-bit vector) is of cardinality at most $k$ (has at most $k$ ones). Consider any $(Q \times n)$-unit $Q$-state INN(0) instance. Every fixed point in such an instance has cardinality at most $n$ [17]. Thus every such instance is also a $(Q \times n)$-unit $n$-sparse INN(0) instance. From Theorem 7, this gives

**Corollary 11** *The Information Capacity of the k-sparse n-unit INN(0) model, for n divisible by k, is* $\geq (n/k - 1)^2 \binom{k}{2} = \Theta(n^2)$.

Thus, the IC of the k-sparse n-unit INN(0) model remains asymptotically optimal over the entire range of admissible values of k, namely the degree of sparseness. From the Information Capacity point of view, sparse coding neither helps nor hurts, asymptotically. The particular sparse recoding of binary vectors as Q-state vectors, $Q = 2^j$ for some $j$, described in Section 5, retains asymptotically optimal IC for all $j$.

### 6.2 The Willshaw Model

Now consider the Willshaw Model. We do not adopt a definition analogous to the sparse INN(0) model, for the following reason. Consider an arbitrary $(Q \times 2n)$-unit Q-state WM(2n-1) instance. It is not true that every fixed point has cardinality $\leq 2n$. As a counter example, consider a *saturated* instance, one in which every Q-state vector is a fixed point. It is easy to check that the set $V$ of *all* units, of cardinality $Q \times 2n$, is also a fixed point.

Instead, we restrict ourselves to reinterpreting Q-state WM(2n-1) instances as sparse WM(2n-1) instances. Consider any $(Q \times 2n)$-unit $Q$-state WM(2n-1) instance. Such an instance arises from storing some collection of Q-state vectors of length n in a Q-state WM(2n-1) instance, using our storage rule (6). According to our storage rule, the stored Q-state vectors of length $n$ may be reinterpreted as binary vectors of length $Q \times 2n$, each binary vector containing $2n$ ones. For this reason, let us denote a $(Q \times 2n)$-unit Q-state WM(2n-1) instance as also a N-unit k-sparse WM(k-1) instance, with $N = Q \times 2n$ and $k = 2n$. Define the n-unit k-sparse WM(k-1) model as the set of n-unit k-sparse WM(k-1) instances. From Theorem 9, this gives

**Corollary 12** *The Information Capacity of the k-sparse n-unit WM(k-1) model, for n divisible by* $2k$, *is* $\geq (n/(2k) - 1)^2 \binom{k}{2} = \Theta(n^2)$.

Thus, the IC of the k-sparse n-unit WM(k-1) model remains asymptotically optimal, order of $n^2$, over the entire range of admissible values of k, namely the degree of sparseness. In particular, the sparse recoding of a a collection of binary vectors of length $n$ as Q-state vectors, as described in Section 5, neither helps nor hurts, asymptotically.

Recall, from Section 1, Amari's result [2], given by (1), that if random vectors of sparseness $k = o(n)$ are stored in the *real-valued* weights associative memory model, then order of $n^2$ bits of information can be stored; if $k = \Theta(n)$, then $n^2/\log n$ bits of information can be stored. Our result of Corollary 12 leads to similar conclusions, for our definition of IC, for the *binary* weights Willshaw model, and without invoking randomness for the stored vectors. Our result differs when $k = \Theta(n)$ for which the IC, by our definition, remains of the order $n^2$.

## 7 Proofs

The proofs use the standard language of graph theory. We define the terminology we use in this paper here. For a more extensive introduction, the reader is refered to [7]. A graph $G$ is denoted by a pair $(V, E)$ where $V$ is the set of vertices and $E$ the set of edges. Two vertices $u, v$ are called *adjacent* in $G$ if they are connected by an edge. Let $G[U]$ denote the subgraph of $G$ induced by any $U \subseteq V$. ($G[U]$ is restricted to the vertices in $U$ and edges amongst them.) Let $\delta(G)$ denote the minimum degree and $\Delta(G)$ the maximum degree in any graph $G$. For a set $U \subseteq V$ and a vertex $v \in V$, define $d_U(v)$ as the number of vertices of $U$ that $v$ is adjacent to. For a graph $G$, let $G_c$ denote its *complement graph.* $G_c$ has the same vertices as $G$. $(v_i, v_j)$ are adjacent in $G_c$ if and only if $(v_i, v_j)$ are not adjacent in $G$. A set $U \subseteq V$ is called an *independent set* in a graph $G$ if no pairs of vertices in $U$ is adjacent in $G$. An independent set $U$ is *maximal* if no strict superset of $U$ is an independent set. A set $U \subseteq V$ is called a *(maximal) clique* in a graph $G$ if $U$ is a (maximal) independent set in $G$'s complement graph $G_c$.

In our proofs, we will occasionally use the following two elementary facts about graphs. We prove them explicitly to keep the exposition clear and the paper self-contained.

**Fact 1** *Different n-vertex labeled graphs have different collections of maximal independent sets.*

**Proof :** Consider two different graphs. There exists a pair of vertices $u, v$ which is adjacent in one graph and not the other. $\{u, v\}$ is contained in some maximal independent set $I$ in the graph $u, v$ are not adjacent in; $I$ is not an independent set in the other graph. ■

**Fact 2** *Let a graph $H$ be generated from an arbitrary graph $G$ as follows. Initially $H$ is empty ($H$ has no edges) and $V(H) = V(G)$. For every maximal clique $C$ in $G$, $H[C]$ is made a clique. Then $H$ equals $G$.*

**Proof :** Consider a pair of vertices $u, v$ in $G$. If $u$ is adjacent to $v$ then $\{u, v\}$ is contained in some maximal clique of $G$; hence $u$ is adjacent to $v$ in $H$. If $u$ is not adjacent to $v$ then $\{u, v\}$ is not contained in any maximal clique of $G$; hence $u$ is not adjacent to $v$ in $H$. Hence $H$ equals $G$. ■

Fact 2 may be equivalently restated as follows. Let $H$ be generated from $G$ as follows. Initially $H$ is a clique and $V(H) = V(G)$. For every maximal independent set $I$ in $G$, $H[C]$ is made an independent set. Then $H$ equals $G$.

For both models—WM and INN—define a graph $G$ underlying a network instance $N$ as follows. The set of vertices $V$ is the set of neurons. $\{i, j\}$ is an edge in $G$ if and only if $w_{ij} \neq 0$. We identify a vector $\mathbf{S} \in \{0, 1\}^n$ with a set $U = \{i | S_i = 1\} \subseteq V$. We use the set notation from here on.

## 7.1 Proofs of Section 3

We first characterize the fixed points of WM(t) and INN(t) as certain induced subgraphs of their underlying graphs. Inaccurate versions of these characterizations, without proof, are in [11]. For example, in [11], fixed points of an INN(t) instance are characterized as maximal induced subgraphs of the underlying graph with maximum degree at most $t$. The maximal part of this, which is the analog of (ii) in our definition, is wrong. In [11], these characterizations are used to encode combinatorial problems and obtain results on the structural complexity, i.e., results analogous to NP-completeness results, of the networks. This work does not address the issue of associative memories.

**Proposition 13** *For integer $t$, $U$ is a fixed point of an WM(t) instance with underlying graph $G$ if and only if (i) $\delta(G[U]) \geq t$ and (ii) $\forall v \notin U : d_U(v) < t$. (If $U = \emptyset$, (i) is assumed to hold. If $U = V$, (ii) is assumed to hold.)*

For example, if the graph in Figure 4 is the underlying graph of a WM(2) instance, $U = \{1, 2, 3\}$ is a fixed point, since every vertex in $U$ has degree at least 1.

**Proposition 14** ([13]) *For integer $t$, $U$ is a fixed point of an INN(t) instance with underlying graph $G$ if and only if (i) $\Delta(G[U]) \leq t$ and (ii) $\forall v \notin U : d_U(v) > t$. (If $U = \emptyset$, (i) is assumed to hold. If $U = V$, (ii) is assumed to hold.)*

For example, if the graph in Figure 4 is the underlying graph of an INN(1) instance, $U = \{2, 3\}$ is a fixed point, since every vertex in $U$ has degree at most 1, and every vertex not in $U$ (in this case 1) has degree more than 1 in $U$ (in this case 2).

For $t \leq 0$, every WM(t) instance has exactly one fixed point: $V$. For $t \geq n$, every WM(t) instance has exactly one fixed point: $\emptyset$. For integer $t > 0$, $\emptyset$ is a fixed point of every WM(t) instance.

**Proof of Lemma 1.** Let us call a set $U \subseteq V$ satisfying the condition of Proposition 13 a *Maximal high-degree-t subgraph.* We will obtain a lower bound on the number of $n$-vertex labeled graphs with different collections of Maximal high-degree-t subgraphs.

Construct a family of graphs $\{G\}$ as follows. The vertices are arranged in an $n/(t+1) \times (t+1)$ grid, that is with $n/(t+1)$ rows and $t+1$ columns. The vertices are indexed as $v_{i,j}, 1 \leq i \leq n/(t+1), 1 \leq j \leq t+1$. Note that $t+1$ is an even positive integer. The columns are partitioned into $(t+1)/2$ blocks $B_1, \ldots, B_{(t+1)/2}$. Block $B_i$ contains all vertices in columns $2i-1$ and $2i$. Each block is made a complete graph minus a matching of $n/(t+1)$ edges. The missing edges are the ones joining vertices in the same row.

Edges crossing blocks are added as follows. First, a pair of distinct blocks $B_i$ and $B_j$ is chosen. Then a row $r \geq 2$ in block $B_i$ and a row $s \geq 2$ in block $B_j$ is chosen. $r$ may equal $s$ or not. Then the two vertices $v_{r,2i-1}, v_{r,2i}$ in row $r$ of block $B_i$ are joined with the two vertices $v_{s,2j-1}, v_{s,2j}$ in row $s$ of block $B_j$. That is, the four edges $(v_{r,x}, v_{s,y})$, $x \in \{2i-1, 2i\}$, $y \in \{2j-1, 2j\}$, are added. This procedure is repeated for different pairs $r, s \geq 2$ such that $r$ is a row in $B_i$ and $s$ a row in $B_j$, and for different pairs of distinct blocks $B_k, B_l$. Note that no edges crossing blocks are added from vertices in row 1. Figure 3 shows a graph constructed this way.

**Claim 1:** Every maximal independent set in such a graph $G$ has size equal to $t+1$.

**Proof :** Since every column is a clique, every independent set in such a graph $G$ has size $\leq t+1$. Suppose there exists a maximal independent set $I$ of size less than $t+1$. Let $v_{r,2i-1}, v_{r,2i}$ be the pair of vertices in the same row $r$ in block $B_i$. Since from our construction, $v_{r,2i-1}$ and $v_{r,2i}$ are adjacent to the same vertices in $G$, either both $v_{r,2i-1}, v_{r,2i}$ belong to $I$, or neither of them belong to $I$. Hence there exists a block $B_j$, none of whose vertices are in $I$. But then $v_{1,2j-1}$ and $v_{1,2j}$ are not adjacent to any vertex in $I$, which contradicts that $I$ is a maximal independent set. ■

**Claim 2:** The family of maximal independent sets in $G$ is exactly the family of maximal high-degree-t subgraphs in the complement graph $G_c$ of $G$, of cardinality $t+1$.

**Proof :** Let $I$ be a maximal independent set in $G$. In the complement graph $G_c$, $\delta(G_c[I]) = t$. In $G_c$, every vertex $v \notin I$ is not adjacent to at least two vertices in $I$ (the ones in $I$ contained in $v$'s block). Hence, in $G_c$, $\forall v \notin I : d_{G_c[I]}(v) \leq |I| - 2 = t+1-2 < t$. Thus $I$ is a maximal high-degree-t subgraph in $G_c$.

Conversely, let $S$ be a maximal high-degree-t subgraph in $G_c$ of cardinality $t+1$. Since, by definition, $\delta(G_c[S]) \geq t$, $S$ is an independent set of size $t+1$ in $G$, namely a maximal independent set. ■

Since every graph $G$ constructed as above has a different collection of maximal independent sets, the complement graph of every such graph has a different collection of maximal high-degree-t subgraphs of cardinality $t+1$, hence a different collection of maximal high-degree-t subgraphs. The number of such graphs is $2^{\binom{(t+1)/2}{2}(n/(t+1)-1)^2}$, which gives the result of Lemma 1.

**Proof of Lemma 3.** Let us call a set $U \subseteq V$ satisfying the condition of Proposition 14 a *Maximal degree-t subgraph.* A Maximal degree-0 subgraph is a Maximal Independent Set. We will obtain a lower bound on the number of $n$-vertex labeled graphs with different collections of Maximal degree-t subgraphs. For this purpose, we employ the following reduction of the Maximum Independent Set problem to the Maximum degree-t subgraph problem [19].

Given a graph $G = (V, E)$, with $V = \{v_1, \ldots, v_n\}$, construct a graph $G_t = (V_t, E_t)$ as follows. Let $V_t = \{v_{ij}, 0 \leq i \leq t, 1 \leq j \leq n\}$, and $E_t = \{(v_{ij}, v_{kl}) | j = l \text{ or } (v_j, v_l) \in E\}$. Effectively, vertex $v_i$ in $G$ is represented by a "column" $C_i = \{v_{ji}, 0 \leq j \leq t\}$ of vertices in $G_t$ formed into a clique;

edge $(v_i, v_j)$ in $G$ is represented by a *join* of the columns $C_i$ and $C_j$ in $G_t$. Figure 4 illustrates this reduction for a 3-vertex graph $G$, with $t = 2$.

Let $\{I\}$ be the family of maximal independent sets of $G$. Let $\{I_t\}$ be the family of maximal degree-t subgraphs in $G_t$ of the form $C_{i_1} \cup \ldots \cup C_{i_l}$.

**Claim:** There is a one-to-one correspondence between the elements of $\{I\}$ and the elements of $\{I_t\}$.

**Proof :** Let $I = \{i_1, \ldots, i_l\}$ be a maximal independent set in $G$. Let $I_t = C_{i_1} \cup \ldots \cup C_{i_l}$. Since $I$ is an independent set in $G$, from the reduction it is clear that $\Delta(G_t[I_t]) = t$. Since $I$ is a *maximal* independent set in $G$, every vertex not in $I$ is adjacent to at least one vertex in $I$, in $G$. Therefore in $G_t$, for every vertex $v \notin I_t$, $d_{G_t[I_t]}(v) \geq t + 1$. Hence $I_t$ is a maximal degree-t subgraph in $G_t$. Conversely, let $I_t = C_{i_1} \cup \ldots \cup C_{i_l}$ be a maximal degree-t subgraph in $G_t$. Since $\Delta(G_t[I_t]) \leq t$ and the degree of every vertex in $G_t[I_t]$ is at least $t$, there are no edges crossing columns in $I_t$. Hence $I = \{i_1, \ldots, i_l\}$ is an independent set in $G$. Suppose $I$ is not a maximal independent set in $G$. Then there exists a vertex $v_j \notin I$ that is not adjacent to every vertex in $I$, in $G$. Consider any vertex $v \in C_j$ in $G_t$. $v$ is not adjacent to every vertex in $I_t$. This contradicts that $I_t$ is a maximal degree-t subgraph in $G_t$. ■

Since every $n$-vertex labeled graph has a different collection of maximal independent sets, it follows from the Claim that every $(t + 1)n$-vertex labeled graph $G_t$ associated with an $n$-vertex graph has a different collection of maximal degree-t subgraphs of the form $\{I_t\}$, hence a different collection of maximal degree-t subgraphs. The lower bound on the number of $(t+1)n$-vertex labeled graphs with different collections of maximal degree-t subgraphs is thus $2^{\binom{n}{2}}$, the number of $n$-vertex labeled graphs. The result follows.

**Proof of Lemma 4.** Consider any graph $G$ constructed in the proof of Lemma 1. Consider its complement graph $G_c$. From Fact 2, $G_c$ may be reconstructed from its maximal cliques. In other words, if we store the binary vectors associated with the maximal cliques of $G_c$ using the storage rule (4), we get a Willshaw network whose underlying graph is $G_c$.

## 7.2 Proofs of Section 4

Before we proceed with the proofs related to storage of $Q$-state vectors, it is useful to explain the Q-state INN storage rule (5) in terms of operations on the underlying graph associated with the network. Initially, the $Q \times n$-vertex graph is made a clique (every pair of vertices is adjacent). Storing a $Q$-state vector $(q_1, \ldots, q_n)$ makes $\{V_{q_1,1}, \ldots, V_{q_n,n}\}$ a maximal independent set. Figure 1 illustrates, for $Q = n = 3$, the complement $G_c$ of the graph $G$ formed after storing $(0, 0, 0)$, $(0, 1, 2)$, and $(2, 1, 0)$.

**Proof of Lemma 6.** The proof is based on a construction due to [8] for $Q = 2$, that we have extended to arbitrary $Q$.

Construct a family of $(Q \times n)$-vertex graphs as follows. The vertices are arranged into a grid of $Q$ rows $0, \ldots, Q - 1$ and $n$ columns $1, \ldots, n$. Row 0 is kept an independent set. Every column is made a clique. A distinct graph results from choosing the edges among vertices in rows 1 through $Q - 1$. (Every edge chosen links vertices in different columns and is not incident on vertices in Row 0.) Figure 5 shows one graph constructed this way, with $Q = n = 3$.

**Claim 1:** In any such graph, all maximal independent sets have cardinality $n$.

**Proof :** It is clear that all independent sets have cardinality $\leq n$. Suppose there exists a maximal independent set $I$ with cardinality less than $n$. There is a column $j$ with no vertex in $I$. $v_{0,j}$ is not adjacent to any vertex in $I$, which contradicts that $I$ is a maximal independent set. ■

Every such graph $G$ is the underlying graph of some Q-state INN instance. (In particular, from Fact 2, we see that if we store the Q-state vectors associated with the maximal independent sets of such a graph $G$, we get a network whose underlying graph is also $G$.) Every such graph $G$ has a different collection of maximal independent sets. Thus, a lower bound on the number of $Q$-state INN instances with different collections of fixed points is the number of such graphs, which is $2^{(Q-1)^2\binom{n}{2}}$.

**Proof of Theorem 8.** Let $G$ denote the graph underlying the network formed after storage of a sequence $X^1, \ldots, X^m$ of Q-state vectors of length $n$, according to the storage rule (6). (See Figure 2 for an illustration.) Consider vector $X^\mu$. We have to show that the set $S_{X^\mu} = \{v_{(q_i,2i-1)} \cup v_{(q_i,2i)} | X_{(q_i,i)} = 1, i = 1, \ldots, n\}$ is a fixed point of the network. Note that $\delta(G[S_{X^\mu}]) = 2n - 1 \geq t = 2n-1$. Consider any unit $v \notin S_{X^\mu}$. $v$ is not adjacent to the two vertices in $S_{X^\mu}$ in its block (in $v$'s column, or in the appropriate neighboring column). Thus $d_{S_{X^\mu}}(v) \leq 2n - 2 < t$. Hence $S_{X^\mu}$ is a Maximal high-degree-t subgraph of $G$, and $X^\mu$ a fixed point of the associated WM network.

**Proof of Theorem 9.** Consider the family of graphs constructed in the proof of Lemma 1, with $t = 2n - 1$.

**Claim:** The complement graph $G_c$ of every graph $G$ in that family is an underlying graph of some Q-state WM(2n-1) instance.

**Proof :** Consider the Maximal high-degree-t subgraphs of $G_c$ of cardinality $t+1$. From Claim 2 of Lemma 1, these subgraphs are exactly the maximal cliques of $G_c$. From Fact 2, the graph $H$ constructed from these subgraphs equals $G_c$. In other words, the network realized by storing the Q-state vectors of length $2n$ associated with the Maximal high-degree-t subgraphs of $G_c$ of cardinality $t+1$, using storage rule (6), has the underlying graph $G_c$. ■

Every such graph $G_c$ has a different collection of maximal cliques, hence Maximal high-degree-t subgraphs of cardinality $t+1$. Thus a lower bound on the number of Q-state WM(2n-1) instances with different collections of fixed points is the number of graphs in the family, namely $2^{(Q-1)^2\binom{n}{2}}$.

### 7.3 Proofs of Section 5

**Proof of Proposition 10.** Let $X^1, \ldots, X^m$ be $m$ $Q$-state vectors chosen uniformly at random and independently. We want $m$ such that

$$\Pr[\text{after storing } X^1, \ldots, X^m, \text{ the network is not saturated}] \leq 1/n^\epsilon$$

Notice that the network is saturated exactly when there are no edges crossing the columns of its underlying graph. Denote by an edge-slot a pair of vertices in different columns.

$$\begin{aligned}\Pr[\text{network is not saturated}] &= \Pr[\text{there exists an edge-slot that is an edge}] \\ &\leq \textstyle\sum_{\text{edge-slots } \{u,v\}} \Pr[\{u,v\} \text{ is an edge}] \\ &= Q^2\binom{n}{2}(1 - 1/Q^2)^m\end{aligned}$$

We want Pr[network is not saturated] $\leq 1/n^\epsilon$ from which $m = \log_{Q^2/(Q^2-1)} \binom{n}{2} Q^2 n^\epsilon$. □.

## 8 Simulations

As far as information capacity is concerned, the theoretical results of this paper pretty much settle the issue. Experiments to estimate information capacity would not provide new results and in any case would be computationally infeasible.

As far as stable storage and basins capacity is concerned, however, simulations of several of the models studied in this paper have been conducted in the recent past. The results, summarized below, reveal that not only does INN(0) have attractive theoretical properties (optimal information capacity, perfect stable storage) but also works well empirically.

One set of extensive simulations of this kind were conducted in [27]. In that paper, Q-state WM'(3) [1], and Q-state INN(0) [2], the Hopfield outerproduct rule, and its true Hebbian variant were evaluated on the tasks of storing the same sets of $k$ random Q-state vectors of length $n$, where $k$ ranged from $0.16n$ to $0.5n$, Q ranged from 2 to 60, and n ranged from 2 to 60. On stable storage alone, the INN(0) worked best—as predicted by theory—achieving stability of all stored patterns in all experiments. The WM(3) model worked second-best, the true Hebb rule third-best, and the Hopfield outerproduct rule poorest (except when $Q = 2$). On recall of non-spurious fixed points (i.e. valid memories) from noisy probes, the INN(0) once again worked best and the Hopfield outerproduct rule the poorest. This time the true Hebb rule worked second best and the WM(3) third best.

All experiments reported in the previous paragraph were on random patterns. Since, according to the results of the current paper, INN(0) has a poor stable storage capacity on random patterns, while a high information capacity overall, one may speculate that INN(0) would perform even better on structured patterns. Indeed this seems to have been the case in a real-world application of INN(0) that involved storing English words (hence structured patterns) and correcting errors in erroneous probes [18]. In that application, 10,548 words arising from a USPS postal application were stored in a 321-neurons INN(0) network. Printed images of some of the same words were processed by a hardware OCR which output a set of letter-hypotheses, usually with many errors. The task of the network was to correct these errors, given the dictionary stored in the network. It was found that the network performed very well at this task when used in combination with a conventional search mechanism. When either the network or the conventional search mechanism was eliminated from this hybrid scheme, the performance was found to degrade significantly.

## 9 Conclusions

We studied the amount of information stored in the fixed points of random instances of two binary weights associative memory models: the Willshaw Model (WM) and the Inverted Neural Network (INN). We proved that the WM has asymptotically optimal IC for nearly the full range of threshold values, the INN likewise for constant threshold values, and both over all degrees of sparseness of the stored vectors. We contrasted this with the result by Palm, which required stored random vectors to be logarithmically sparse to achieve good storage capacity for the WM, and with that of Grossman, which showed that the INN has poor storage capacity for random vectors. We proposed Q-state versions of the WM and the INN, and showed that they retain asymptotically optimal IC while guaranteeing stable storage. By contrast, we showed that the Q-state INN has poor storage capacity for random vectors.

Our results indicate that measuring the information capacity of associative memory models via the storage capacity alone can sometimes give only part of the picture—it is useful to complement this with information capacity measured according to the IC definition. It might be useful, for example, to measure the IC of other associative memory models (perhaps the *Bidirectional Associative*

[1] Q-state WM' is a variant of Q-state WM in which the Q-state vectors of length $n$ are stored directly via storage rule (6), rather than after the recoding described in Section 4.2. WM' uses half the number of neurons that WM does but has the disadvantage that Theorem 8, guaranteeing stability of all stored vectors, no longer holds.

[2] More accurately, on an exactly equivalent version of INN(0)

*Memory*, or *Sparse Distributed Memory*) and compare it with information capacity obtained from storage capacity considerations. Some of our techniques may be useful in this regard, especially for memory models employing binary-valued weights.

## Acknowledgments

The authors are grateful to anonymous reviewers for their helpful comments that improved the quality of this paper.

## References

[1] Y.S. Abu-Mostafa and J.S. Jacques. Information capacity of the Hopfield model. *IEEE Transactions on Information Theory*, 31(4):461–464, July 1985.

[2] S. Amari. Characteristics of sparsely encoded associative memory. *Neural Networks*, 2(6):451–457, 1989.

[3] J.A. Anderson. A simple neural network generating interactive memory. *Mathematical Biosciences*, 14:197–220, 1972.

[4] R. Bar-Yehuda, 1993. Personal Communication.

[5] Y. Baram. Corrective memory by a symmetric sparsely encoded network. *IEEE Transactions on Information Theory*, 40(2):429–438, 1994.

[6] S. Biswas and S.S. Venkatesh. Codes, sparsity, and capacity in neural associative memory. Technical report, Department of Electrical Engineering, University of Pennsylvania, 1990.

[7] J.A. Bondy and U.S.R Murty. *Graph Theory with Applications.* North-Holland, New York, 1976.

[8] S. Chaudhari and J. Radhakrishnan, 1991. Personal Communication.

[9] T. Chiueh and R.M. Goodman. Recurrent correlation associative memories. *IEEE Transactions on Neural Networks*, 2(2):275–284, 1991.

[10] A. Dembo. On the capacity of associative memories with linear threshold functions. *IEEE Transactions on Information Theory*, 35(4):709–720, 1989.

[11] G.H. Godbeer, J. Lipscomb, and M. Luby. On the computational complexity of finding stable state vectors in connectionist models (Hopfield nets). Technical report, Department of Computer Science, University of Toronto, Toronto, Ontario, 1988.

[12] T. Grossman. The INN model as an associative memory. Technical Report LA-UR-93-4149, Los Alamos National Laboratory, Los Alamos, NM 87545, 1993.

[13] T. Grossman and A. Jagota. On the equivalence of two Hopfield-type networks. In *IEEE International Conference on Neural Networks*, pages 1063–1068. IEEE, 1993.

[14] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation.* Addison-Wesley, 1991.

[15] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, 79, 1982.

[16] A. Jagota. Information capacity of a Hopfield-style memory. In *World Congress on Neural Networks*, volume 2, pages 220–223, New York, July 1993. Portland, IEEE.

[17] A. Jagota. A Hopfield-style network with a graph-theoretic characterization. *Journal of Artificial Neural Networks*, 1(1):145–166, 1994.

[18] A. Jagota. Contextual word recognition with a Hopfield-style net. *Neural, Parallel, & Scientific Computations*, 2(2):245–271, 1994.

[19] A. Jagota and G. Narasimhan. A generalization of maximal independent sets, 1994. Submitted.

[20] A. Jagota, A. Negatu, and D. Kaznachey. Information capacity and fault tolerance of binary weights hopfield nets. In *IEEE International Conference on Neural Networks*, volume 2, pages 1044–1049, New York, 1994. Orlando, FL, IEEE.

[21] J.D. Keeler. Capacity for patterns and sequences in Kanerva's SDM as compared to other associative memory models. Technical report, Research Institute for Advanced Computer Science: RIACS TR 87.29, NASA Ames Research Center, 1987.

[22] T. Kohonen. Correlation matrix memories. *IEEE Transactions on Computers*, C-21:353–359, 1972.

[23] G.A. Kohring. On the problems of neural networks with multi-state neurons. *Journal De Physique I*, 2:1549–1552, August 1992.

[24] R. Kothari, S. Megada, M. Cahay, and G. Qian. Q-state neural associative memories using threshold decomposition. Under Review. Communicated to A. Jagota by R. Kothari in July 1994.

[25] R.J. McEliece, E.C. Posner, E.R. Rodemich, and S.S. Venkatesh. The capacity of the Hopfield associative memory. *IEEE Transactions on Information Theory*, 33:461–482, 1987.

[26] K. Nakano. Associatron—a model of associative memory. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2:381–38, 1972.

[27] A. Negatu and A. Jagota. Q-state associative memory rules and comparisons. In S. Amari, L. Xu, L. Chan, I. King, and K.S. Leung (Eds), *Progress in Neural Information Processing*, volume 1, pages 592–597, Hong Kong, Springer, 1996.

[28] G. Palm. On associative memory. *Biological Cybernetics*, 36:19–31, 1980.

[29] E.M. Palmer. *Graphical evolution.* Wiley, New York, 1985.

[30] H. Rieger. Storing an extensive number of grey-toned patterns in a neural network using multi-state neurons. *Journal of Physics A*, 23:L1273–L1280, 1990.

[31] I. Shariv, T. Grossman, E. Domany, and A.A. Friesem. All-optical implementation of the inverted neural network model. In *Optics in Complex Systems*, volume 1319. SPIE, 1990.

[32] D.J. Willshaw, O.P. Buneman, and H.C. Longuet-Higgins. Non-holographic associative memory. *Nature*, 222, 1969.

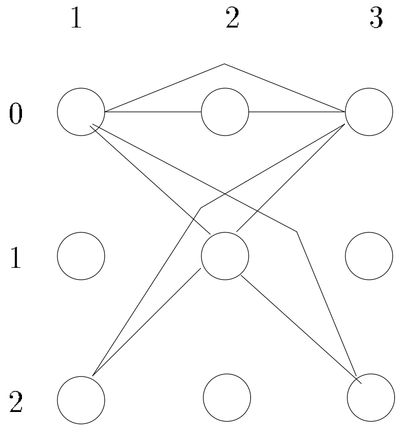


Figure 1: The $Q \times n$ neural grid, where $Q = 3$ and $n = 3$, after three Q-state vectors (0,0,0), (0,1,2), and (2,1,0), each of length $n = 3$, are stored via the INN storage rule. An edge in the graph represents a weight of value zero in the INN network; a non-edge a weight of value -1. Before any of the vectors are stored, there are no edges in the graph, i.e. all the weights are -1. Storing the 3-state vector (0,0,0) introduces the three edges connecting all pairs of neurons in row 0 (i.e., these weights become zero); storing the vector (0,1,2) introduces the three edges on one diagonal on the grid; and storing the vector (2,1,0) introduces the three edges on the other diagonal. The end-result is the graph as shown in the figure.

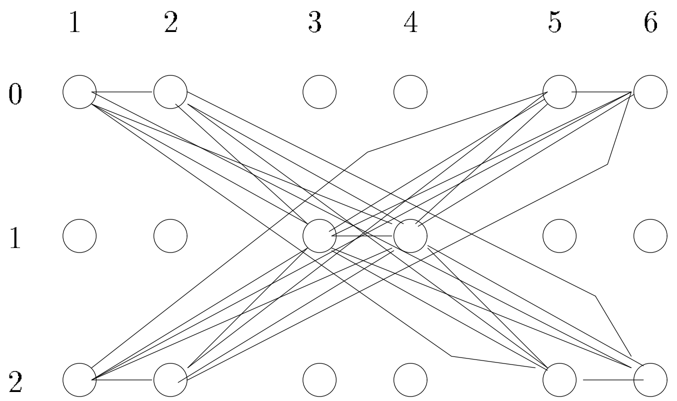


Figure 2: The $Q \times 2n$ neural grid, where $Q = 3$ and $n = 3$, after two Q-state vectors (0,1,2) and (2,1,0), each of length $n = 3$, are first recoded as the vectors (0,0,1,1,2,2) and (2,2,1,1,0,0) respectively; the latter are then stored via the Q-state WM storage rule. An edge in the graph represents a weight of value 1 in the WM network; a non-edge a weight of value 0. Before any of the vectors are stored, there are no edges in the graph, i.e. all the weights are 0. Storing the vectors (0,0,1,1,2,2) and (2,2,1,1,0,0) causes the groups of edges visually seen in the form of the two diagonals to emerge.

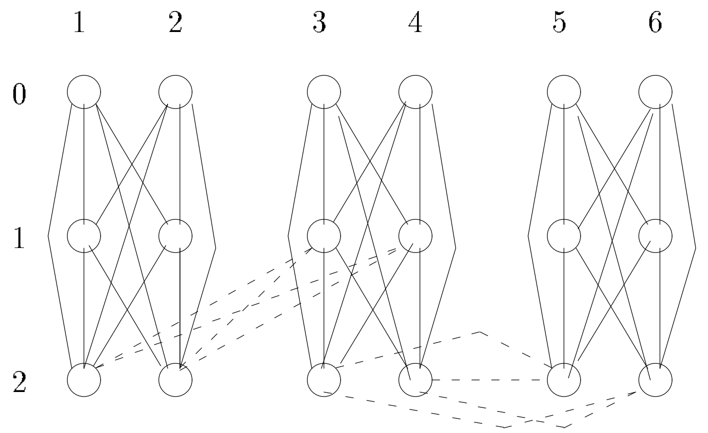
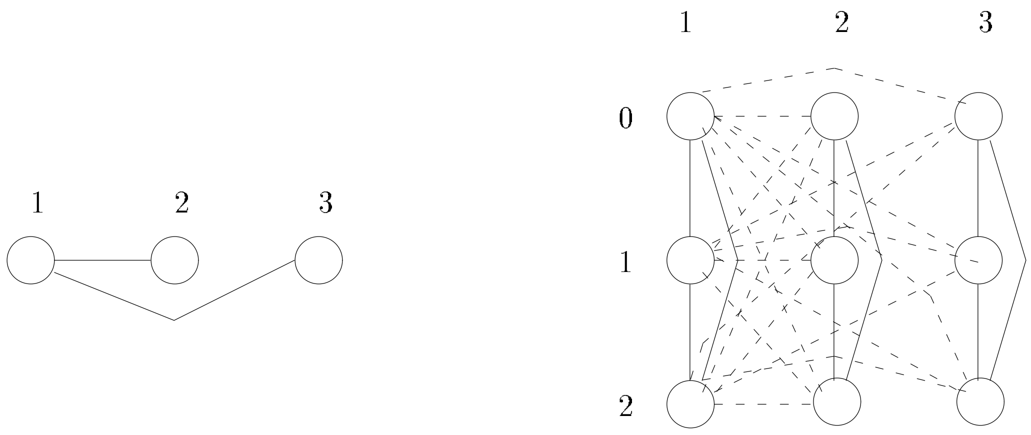


Figure 3: A graph in the family of graphs constructed for the proof of Lemma 1. Solid lines indicate edges fixed in the family; dashed lines indicate edges in this particular graph.



(a) Graph G

(b) Graph Gt

Figure 4: (a) A graph $G$ and (b) its reduced graph $G_t$ for $t = 2$. Dashed lines in (b) denote edges in $G_t$ that correspond to the edges of $G$.

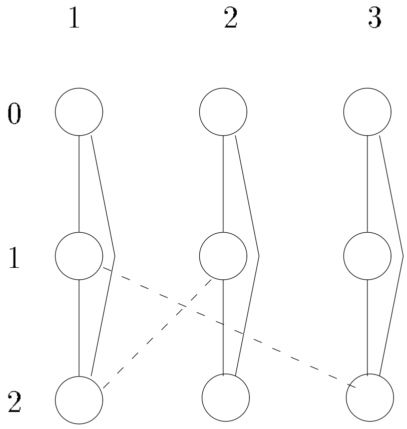


Figure 5: A graph in the family of graphs constructed for the proof of Lemma 6. Solid lines indicate edges fixed in the family; dashed lines indicate edges in this particular graph.