Regarding the temporal requirements of a hierarchical Willshaw network

João Sacramento[a,*], Francisco Burnay[b], Andreas Wichert[a]

[a] *INESC-ID Lisboa and Instituto Superior Técnico, Technical University of Lisbon,*
Av. Prof. Dr. Aníbal Cavaco Silva, 2744-016 Porto Salvo, Portugal
[b] *Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Technical University of Lisbon,*
Av. Rovisco Pais, 1049-001 Lisboa, Portugal

Abstract

In a recent communication, Sacramento & Wichert (2011) proposed a hierarchical retrieval prescription for Willshaw-type associative networks. Through simulation it was shown that one could make use of low resolution descriptor patterns to decrease the total time requirements of recalling a learnt association. However, such method introduced a dependence on a set of new parameters which define the structure of the hierarchy. In this work we compute the expected retrieval time for the random neural activity regime which maximises the capacity of the Willshaw model and we study the task of finding the optimal hierarchy parametrisation with respect to the derived temporal expectation. Still in regard to this performance measure, we investigate some asymptotic properties of the algorithm.

Keywords:
Associative memory, Willshaw model, Retrieval time, Hierarchical neural network, Sparse coding

1. Introduction

In the strictest technical sense, an associative memory model is designed to solve a variation of the classical nearest neighbour determination problem. Instead of finding a solution for the original labelled classification task formulation (Fix & Hodges, 1951; Cover & Hart, 1967; Minsky & Papert, 1969), an associative memory is a system that stores information about a finite set of M associations of the form

$$S := \{(\mathbf{x}^\mu \mapsto \mathbf{y}^\mu) : \mu = 1, \ldots, M\}, \tag{1}$$

with most memory models assuming the patterns are binary vectors, i.e., $\mathbf{x} \in \{0,1\}^m$ and $\mathbf{y} \in \{0,1\}^n$. Given a possibly corrupt or incomplete pattern $\tilde{\mathbf{x}} \in \{0,1\}^m$, the system

[*] Corresponding author. Tel.: +351 21 423 32 31; fax: +351 21 314 48 43.
Email addresses: `joao.sacramento@ist.utl.pt` (João Sacramento), `francisco.burnay@ist.utl.pt` (Francisco Burnay), `andreas.wichert@ist.utl.pt` (Andreas Wichert)

should be able to find the best-matching (or rather, the 'nearest neighbour') $\mathbf{x}^\mu$ with respect to a desired similarity metric and then return a pattern $\hat{\mathbf{y}} \in \{0,1\}^n$ ideally corresponding to the originally stored $\mathbf{y}^\mu$. Thus, the original association ($\mathbf{x}^\mu \mapsto \mathbf{y}^\mu$) ought to be restored through a robust recall process.

Three different yet closely related tasks are usually identified with the above process: when $n = 1$ and $m \gg n$ the memory solves a binary classification problem over the labels 'known' and 'unknown' (learnt patterns being associated with the former) and is said to perform familiarity discrimination (Bogacz et al., 2001; Bogacz & Brown, 2003; Greve et al., 2009); when $m = n$ and $\forall\mu : \mathbf{x}^\mu = \mathbf{y}^\mu$ an autoassociative function is carried out and the memory is expected to perform pattern completion or correction; finally, the case of arbitrary m, n and $\mathbf{x}^\mu, \mathbf{y}^\mu$ is called heteroassociation. The latter is most easily comparable with a standard von Neumann computer memory.

The general quality of a neural associative memory implementation can be assessed with respect to several quantities. The most addressed in the literature is the storage capacity (Willshaw et al., 1969; Palm, 1980; Amit et al., 1985; Gardner, 1988; Palm & Sommer, 1992; Knoblauch et al., 2010), which is typically measured through the critical pattern capacity α_c (simply given by a normalisation of the number of patterns M over the number of content neurons n) or through the more general network capacity C, measured in bits per synapse (bps) and defined as the maximal mutual Shannon information (Shannon, 1948; Cover & Thomas, 2006) between stored and retrieved vectors $I(\mathbf{x}^1, \ldots, \mathbf{x}^M; \hat{\mathbf{y}}^1, \ldots, \hat{\mathbf{y}}^M)$ normalised over the count of synaptic contacts required by the network. The latter is usually preferable since it takes into account both the required resources and the information content of the patterns.

Another quantity of interest is the expected time necessary for the learning and retrieval processes to terminate, generally presented as a count of elementary operations and as a function of some of the parameters which define the memory task. The temporal requirements of an associative memory model deserve attention from both the technical and biophysical perspectives, partly determining the model's efficiency. From a purely algorithmic point of view, the pattern recognition community is currently facing the challenge of solving as quickly as possible the nearest neighbour determination problem in high dimensional space (large m, n) and for high pattern loads (large M), to cope with the increase in size of modern data sets. When the analogy with biological neural networks is to be taken in consideration, the temporal efficiency is equally important as it is linked to the energetic requirements of membrane potential determination (Knoblauch et al., 2010).

While attempting to solve the associative memory task using a neural computation approach a lot of effort has been placed in developing and studying recurrent networks (i.e., with feedback couplings), as for finite systems they can provide stronger error tolerance to pattern noise through an increase in the size of the basins of attraction (Gardner-Medwin, 1976; Hopfield, 1982; Amit et al., 1985; Derrida et al., 1987; Kosko, 1987; Gardner, 1988; Golomb et al., 1990; Palm & Sommer, 1992; Sommer & Palm, 1999), in exchange for additional iterations during the retrieval process.

If our aim is the least computational effort, it is known that under the sparse random coding regime (Barlow, 1972; Field, 1994; Olshausen & Field, 2004) the simpler Willshaw net achieves a high storage capacity (viz., $C = \log 2 \approx 0.7$ bit per synapse in the limit of $m, n \to \infty$) using a biologically plausible local learning rule of the Hebbian type and a parallel 'single-shot' retrieval prescription (Steinbuch, 1961; Willshaw et al., 1969;

Palm, 1980; Amari, 1989; Nadal & Toulouse, 1990; Knoblauch et al., 2010). Besides allowing for high storage capacities, the coding restriction on the ℓ_0 pseudo-norm of the pattern vectors (or, equivalently, on the ℓ_1 norm since we assume the patterns are binary) imposed by the sparseness requirement also reduces the temporal complexity of learning and retrieval and seems to be in accordance with the signalling and maintenance energy budget of the mammalian brain (Levy & Baxter, 1996; Lennie, 2003; Laughlin & Sejnowski, 2003).

Due to the inherently parallel synchronous update mechanism, the temporal benefits of the single-shot retrieval procedure employed by the Willshaw model can only be fully exploited using specialised hardware constructs. Attempting to decrease the retrieval time on sequential computer implementations, a recent communication suggested the use of a hierarchical retrieval prescription in order to take advantage of the sparse structure of the stored patterns (Sacramento & Wichert, 2011).

However, the proposed model introduced a dependence on a new set of integer parameters which defined the hierarchy, and were obtained through exhaustive combinatorial search. It remained unclear whether this problem was tractable for high dimensional pattern spaces, and whether a heuristic approach could be derived in order to avoid the integer constrained optimisation. In this work we address these issues and compute refined time expectations for finite memories. En passant, we also show that asymptotically the hierarchical refinement procedure reduces the temporal complexity of the retrieval process when compared with the original single-layer network.

The rest of this paper is organised as follows. In section 2, we review the network model presented in (Sacramento & Wichert, 2011) and derive exact expectations for the time requirements of learning and retrieval. Then, in section 3, we analyse the optimisation task of determining the hierarchical configuration which minimises the time expectation we obtain. We show that even though the problem is difficult to solve analytically, the solution space grows with a polynomial of the pattern space dimension and can thus be tackled through enumeration. We also provide a heuristic method to solve the task and verify its validity empirically for several network configurations.

2. Model characterisation

On the first part of this work we will see how associative networks of the Willshaw type use a kind of plasticity (namely synaptic) and a local Hebbian learning rule to store and recall memory traces. After defining the equations which govern the learning and retrieval processes of the original single-layer model and its hierarchical variant, we will change focus to the statistical characterisation of their temporal requirements. Following the algorithm analysis tradition, we will adopt as our time measure a simple count of the number of necessary operations that either a sequential computer or a specialised hardware construct can perform in constant $O(1)$ time. Asymptotic comparisons using Bachmann-Landau notation can then be made, as well as finite numerical evaluations for particular cases. This approach has found widespread use across the literature, as it is mathematically tractable and abstract enough to establish a comparison between different models and implementation architectures.

2.1. Network equations for learning and retrieval

The original Willshaw model is a single-layer neural network comprising two populations of McCulloch-Pitts binary threshold neurons (McCulloch & Pitts, 1943): an address population of m neurons capable of establishing synaptic connections with n neurons which form the content population. We can then interpret the silent-firing (0-1) activity patterns of each set of neurons at a given synchronous time frame as our binary input (address) and output (content) vectors.

During the learning phase, we assume each pair $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ from S is presented to the network independently at time μ. A Hebbian-type learning rule is applied and formation of synapses can occur as a consequence of new stimuli. Willshaw networks employ a particular non-additive clipped Hebb rule, where the synaptic strength factor is disregarded, i.e., we only care to check whether a synapse between any two neurons i and j is either present or not. Thus, the entire state of a network can be represented by a binary weight matrix $\mathbf{W} \in \{0,1\}^{m\times n}$, where $W_{ij} = 0$ denotes an absent synapse from pre-synaptic neuron i to post-synaptic neuron j and $W_{ij} = 1$ a present one. After learning the M associations of S, the entries of the synaptic connectivity matrix are then given by

$$W_{ij} = \min\left(1, \sum_{\mu=1}^{M} x_i^\mu y_j^\mu\right), \tag{2}$$

which results in bidirectional synapses $\forall i, j,\ W_{ij} = W_{ji}$ for the autoassociative case of $m = n$ and $\forall \mu,\ \mathbf{x}^\mu = \mathbf{y}^\mu$.

Notice how this learning prescription leads to distributed storage, in the sense that each synaptic contact W_{ij} can store information about more than one pattern pair. It is also the simplest possible realisation of the hypothesis of Hebb (1949), as the synaptic update procedure is local and bounded. Note that due to the nonlinearity of the rule, S cannot in general be recovered from $\mathbf{W}$.

The retrieval process starts when the address population fires according to a certain cue pattern $\tilde{\mathbf{x}}$. The activity state $\hat{\mathbf{y}}$ of the content neuron population which will yield the output pattern of the network must then be updated. Each unit j computes (locally) its dendritic potential, corresponding to the sum of the incoming excitatory signals,

$$s_j = \sum_{i=1}^{m} W_{ij}\tilde{x}_i, \tag{3}$$

over which a nonlinear activation function is applied

$$\hat{y}_j = H[s_j - \Theta], \tag{4}$$

where H is the Heaviside step function. Note that the parameter Θ determines the highly nonlinear threshold operation which denoises the output and its choice will critically influence the quality of the recovered pattern $\hat{\mathbf{y}}$. Optimal threshold determination, i.e., finding the Θ which minimises a penalty function such as the expected Hamming distance between stored and retrieved patterns $d_H(\mathbf{y}^\mu, \hat{\mathbf{y}}) = \sum_{j=1}^{n} |y_j^\mu - \hat{y}_j|$, cannot be solved in general without taking statistical assumptions on S (Buckingham & Willshaw, 1993; Graham & Willshaw, 1995). However, when the address cue $\tilde{\mathbf{x}}$ is exact or its active components are a subset of those present in the learnt $\mathbf{x}^\mu$, the memory is said to perform pattern-part-retrieval, and the threshold setting strategy $\Theta = ||\tilde{\mathbf{x}}||_0$ originally proposed by Willshaw et al. (1969) becomes optimal. This choice not only simplifies the model's analytical treatment, but also has a biologically plausible implementation through feed-forward inhibition (Knoblauch et al., 2010).

Instead of a single-layer of neurons, the model extension proposed in (Sacramento & Wichert, 2011) is composed by R Willshaw layers which share a (fixed) address neuron population of dimension m and are ordered by content space dimension. We can then represent the state of the network by an ordered set of R synaptic connectivity matrices

$$\mathcal{W} = (\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(r)}, \ldots, \mathbf{W}^{(R)}), \tag{5}$$

where each $\mathbf{W}^{(r)}$ is $m \times n_r$ dimensional, with $n_1 < \ldots < n_r < \ldots < n_R$. The entries of the r-th synaptic matrix $\mathbf{W}^{(r)}$ are of the same form of (2), except that each of the original content patterns $\mathbf{y}^\mu$ is replaced by a transformed version $\zeta_r(\mathbf{y}^\mu)$:

$$W_{ij}^{(r)} = \min\left(1, \sum_{\mu=1}^{M} x_i^\mu [\zeta_r(\mathbf{y}^\mu)]_j\right). \tag{6}$$

The essential functioning scheme is that the R-th layer stores the provided set of associations S as prescribed above, i.e., $\mathbf{W}^{(R)} = \mathbf{W}$ and $n_R = n$, while the preceding populations learn content patterns of increasingly lower resolution (in inverse order, i.e., from $R-1$ to 1, the latter storing the coarsest approximations). When a cue is presented to the network, each level r produces a low-dimensional output pattern $\hat{\mathbf{y}}^{(r)}$ which serves as a coarse filter to reduce the set of potentially interesting content neurons which have to perform the dendritic check operation at level $r+1$.

After the pair $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ is associated at the R-th layer, the learning process continues until each of the memories from $r = R-1$ to $r = 1$ has been presented with a pair of patterns $(\mathbf{x}^\mu, \zeta_r(\mathbf{y}^\mu))$. Here $\zeta_r \colon \{0,1\}^{n_{r+1}} \to \{0,1\}^{n_r}$ is a family of functions which recursively approximate the content patterns. The recurrence is defined component-wise from level $r+1$ to level r,

$$[\zeta_r(\mathbf{y}^\mu)]_i = \bigvee_{j=i\cdot a_r-(a_r-1)}^{i\cdot a_r} [\zeta_{r+1}(\mathbf{y}^\mu)]_j, \tag{7}$$

where the 'aggregation factor' parameter $a_r \in \mathbf{N}$ relates the dimensions n_{r+1} and n_r of two consecutive content pattern spaces:

$$n_r = \begin{cases} \lceil n_{r+1}/a_r \rceil & \text{if } 1 \leq r < R, \\ n & \text{if } r = R. \end{cases} \tag{8}$$

Retrieval from an address cue $\tilde{\mathbf{x}}$ starts at the smallest content neuron population, i.e., at level $r = 1$, where the Willshaw prescription given by (3) and (4) is applied to compute $\hat{\mathbf{y}}^{(1)}$. For layer $r+1$ (when $1 \leq r \leq R-1$), the recall cue is again $\tilde{\mathbf{x}}$, but hopefully by making use of the preceding coarse output $\hat{\mathbf{y}}^{(r)}$, only a fraction of the content neuron population will have to perform the dendritic sum operation.

As the approximation in (7) is computed using the Boolean OR operator, a given '1' component j of $\hat{\mathbf{y}}^{(r)}$ corresponds to an index set $Y_j^{(r)}$ with a_r elements that can be used to identify the neurons at level $r+1$ which possibly contain information about the original content pattern $\mathbf{y}^\mu$ we ought to recover:

$$Y_j^{(r)} = \{j \cdot a_r, j \cdot a_r - 1, \ldots, j \cdot a_r - (a_r - 1)\}. \tag{9}$$

Conversely, we have $Y_j^{(r)} = \emptyset$ when $\hat{y}_j^{(r)} = 0$. Then, the union of the $||\hat{\mathbf{y}}^{(r)}||_0$ non-empty sets yields the complete set $\mathcal{Y}_{r+1}$ of indices for which the dendritic sum must be calculated at level $r+1$:

$$\mathcal{Y}_{r+1} = \bigcup_{j\colon \hat{y}_j^{(r)}=1} Y_j^{(r)}. \tag{10}$$

Thus, for the $r+1$-th layer, the dendritic sum operation defined in (3) is kept unchanged, but we only have to apply it to the members of $\mathcal{Y}_{r+1}$:

$$s_j^{(r+1)} = \begin{cases} \sum_i W_{ij}^{(r+1)} \tilde{x}_i & \text{if } j \in \mathcal{Y}_{r+1}, \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

the same holding true for the output transfer function, which we can now define as

$$\hat{y}_j^{(r+1)} = \begin{cases} H[s_j^{(r+1)} - \Theta] & \text{if } j \in \mathcal{Y}_{r+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

2.2. Time requirements of single-layer networks

In this section we briefly review the temporal requirements of the original single-layer Willshaw model, which is remarkably straight-forward to analyse in terms of learning and retrieval efficiency. In fact, expressions for both cases can be derived from simple inspection of equations 2, 3 and 4.

When presented with a finite set of patterns for learning, a naive sequential computer implementation requires $m \cdot n$ synaptic weight adjustments for each of the M patterns. However, as described in (Bentz et al., 1989; Knoblauch et al., 2010), an immediate performance gain can be achieved if the patterns are encoded in the so-called 'pointer format' where a given binary vector $\mathbf{v}$ is represented by a $||\mathbf{v}||_0$-sized vector of natural numbers containing the indices of the '1' components. One should note that from the sparseness requirement follows $||\mathbf{x}^\mu||_0 \ll m$ and $||\mathbf{y}^\mu||_0 \ll n$ for all $\mu = 1, \ldots, M$, making this representation extremely convenient. Thus, the total number of sequential computations required for learning M associations on a Willshaw net is given by:

$$t_{\text{learn}}^{\text{W}} = \sum_{\mu=1}^{M} ||\mathbf{x}^\mu||_0 \cdot ||\mathbf{y}^\mu||_0 \approx M \cdot k \cdot l, \tag{13}$$

where k and l are the mean activity levels of the presented address and content patterns (respectively), i.e.,

$$k := \frac{1}{M} \sum_{\mu=1}^{M} ||\mathbf{x}^\mu||_0, \tag{14}$$

and

$$l := \frac{1}{M} \sum_{\mu=1}^{M} ||\mathbf{y}^{\mu}||_0. \tag{15}$$

Of course, if the patterns have fixed activity levels as in the scenario analysed by Palm (1980), then the approximation in equation 13 becomes an exact equality. For the optimal setting where $k \sim \log m$ and $l \sim \log n$ and if we further consider the typical case of $m \sim n$, then asymptotically it holds that learning a single association is in $\Theta(\log^2 n)$. This bound illustrates one of the strengths of the Hebbian 'one-shot' learning prescription — its quickness. When $k, l \to 0$ this procedure leads to capacities near the theoretical maximum of $C = 1$ (Palm, 1992), while requiring only a single pass on each pattern.

When performing a retrieval with a given input cue $\tilde{\mathbf{x}}$ with activity level $z := ||\tilde{\mathbf{x}}||_0$, similar conditions apply, and the pointer format is again highly desirable. Using such an encoding scheme, the sequential retrieval cost (in number of operations) is given by

$$\begin{aligned} t^{\mathrm{W}}_{\mathrm{retr}} &= z \cdot n + n && (16)\\ &= (z+1) \cdot n, && (17) \end{aligned}$$

since n synaptic contacts (weight matrix positions) of z address neurons (weight matrix rows) have to be checked and the entire population of n content neurons (weight matrix columns) has to perform a final threshold cut. Once again considering the optimal setting of $z \sim \log n$, we have the asymptotic tight bound $t^{\mathrm{W}}_{\mathrm{retr}} = \Theta(n \log n)$.

2.3. Time requirements of the hierarchical model

Deriving an upper bound for the learning time expression $t^{\mathrm{hier}}_{\mathrm{learn}}$ of the hierarchical Willshaw model is straight-forward as the synaptic weight update rule is kept unchanged, and is simply consecutively applied to every layer $1 \leq r \leq R$. We note that the activity level of an aggregated vector cannot be greater than that of the original vector, i.e., $||\zeta_r(\mathbf{y})||_0 \leq ||\mathbf{y}||_0$. As there are R levels, we know that

$$t^{\mathrm{hier}}_{\mathrm{learn}} \leq R \cdot t^{\mathrm{W}}_{\mathrm{learn}}. \tag{18}$$

Finding an exact expression for $t^{\mathrm{hier}}_{\mathrm{learn}}$ is more laborious and requires some knowledge on the statistical properties of the stored patterns, as $||\zeta_r(\mathbf{y})||_0$ depends on the positioning of the '1' components of the original vector $\mathbf{y}$. We will now analyse the aggregation process in the light of elementary probability theory, as this analysis will be helpful not only now but also later when dealing with the retrieval process.

2.3.1. Aggregation probabilities

To make the analysis tractable we follow an assumption similar to Marr's (Marr, 1971) which states that the stored patterns are randomly drawn from the sets of $\binom{m}{k}$ and $\binom{n}{l}$ possible patterns but have activity levels with fixed mean values k (for address patterns) and l (for content patterns). We assume that the activity level of each pattern is determined through the inspection of a binomial variable with its characteristic probability set to k/m or l/n, respectively. Then, $||\mathbf{x}^{\mu}||_0$ and $||\mathbf{y}^{\mu}||_0$ become binomially distributed random variables with a priori known expectations $\mathrm{E}(||\mathbf{x}^{\mu}||_0) = k$ and $\mathrm{E}(||\mathbf{y}^{\mu}||_0) = l$.

This scenario is quite common in the associative memory literature and has already been thoroughly covered (Buckingham, 1991; Buckingham & Willshaw, 1992; Sommer &

Palm, 1999; Knoblauch, 2008), in part thanks to its mathematical tractability. Due to the binomially distributed activity levels, the generated pattern components are independent and have the same firing probability, i.e., for each generated pattern $\mathbf{v}$ we have

$$\begin{aligned} \mathrm{P}(v_i = 1 \wedge v_j = 1) &= \mathrm{P}(v_i = 1) \cdot \mathrm{P}(v_j = 1) \ \forall_{i,j} &(19)\\ &= \mathrm{P}(v_i = 1)^2 \ \forall_{i,j}, &(20)\end{aligned}$$

where $\mathrm{P}(\cdot)$ denotes probability.

The described set up is also interesting because it has a reasonable biological interpretation. Unlike the traditional alternative covered in the analysis of Palm (1980), where k and l are constant, here the Hebbian cell assembly sizes (which correspond to pattern activities) are allowed to vary, even if according to a simplified statistical model. Such a scenario is more realistic as it seems unlikely that the formed synaptic cliques would all have exactly the same size (Knoblauch, 2008).

Essentially, we are interested in computing the expected activity level $\bar{l}_r := \mathrm{E}(||\zeta_r(\mathbf{y}^\mu)||_0)$ of a given content pattern $\zeta_r(\mathbf{y}^\mu)$ which results from the application of the windowed aggregation function, at a given depth r. To do so, we first solve the problem of a single-step (i.e., from $r-1$ to r) application of the aggregation function to a vector $\mathbf{y}$ with a given window length of a, denoting $\bar{l}_r$ simply by $\bar{l}$; then, we can recurrently apply the rule for the subsequent levels of the hierarchy.

We note that a window (which is an a-sized subset of contiguous components of y) will not result in a '1' after aggregation only if all of its components are set to '0'. The probability of this event is simple to determine:

$$\mathrm{P}(\forall i \in \mathfrak{W} : y_j = 0) = \mathrm{P}(y_j = 0)^a = \left(\frac{n-l}{n}\right)^a, \tag{21}$$

where $\mathfrak{W}$ denotes the set of indices which compose the window. We have taken pattern component independence into account on the first equality (note that $|\mathfrak{W}| = a$). From this equation we can immediately derive the probability of a firing window as both events are complementary. Multiplying by the total number of windows (which is n/a), we arrive to the desired equation:

$$\bar{l}(n, l, a) = \frac{n}{a}\left(1 - \left(\frac{n-l}{n}\right)^a\right). \tag{22}$$

From this point, determining the equation for the general case of R memories is a matter of defining a recursion, as the patterns presented for learning to the r-th memory are successively aggregated using factors $a_R = 1, a_{R-1}, ..., a_r$. By definition, when $r = R$, no aggregation is performed, thus $\bar{l}_R = l$ is the stopping condition. Hence, we have

$$\bar{l}_r = \begin{cases} \bar{l}(n_{r+1}, \bar{l}_{r+1}, a_r) & \text{if } r < R, \\ l & \text{if } r = R. \end{cases} \tag{23}$$

2.3.2. Computing the expected retrieval time

After learning a desired number M of pattern associations, the network is ready to perform retrieval operations. At this point it might be of interest to note that unlike the standard single-layer Willshaw net, where retrieval time is only indirectly a function of the memory load[1], the neuron filtering scheme depends on the outputs of each level, which in turn can be affected by the memory load.

Before proceeding we should clearly define what is meant with 'memory load'. One possibility which has been used since the first analyses (Willshaw et al., 1969; Palm, 1980) to ascertain it is $p := \mathrm{P}(W_{ij} = 1)$, i.e., the density of '1' entries in the synaptic matrix $\mathbf{W}$. It is especially useful since for random patterns we can define an expression related to the storage parameters m, n, k, l, M which are known a priori. As in Willshaw et al. (1969); Palm (1980) we determine p resorting to the complementary event:

$$\begin{aligned} p &= 1 - \mathrm{P}(W_{ij} = 0) \\ &= 1 - \mathrm{P}\left(\forall (\mathbf{x}^\mu, \mathbf{y}^\mu)(i \notin \mathbf{x}^\mu \wedge j \notin \mathbf{y}^\mu)\right) \\ &= 1 - \left(1 - \frac{kl}{mn}\right)^M . \end{aligned} \tag{24}$$

Similarly, we can calculate the probability p_r for a given level r of the hierarchy if instead of l we use $\bar{l}_r$ and instead of n the correspondingly adjusted population size $n_r = \lceil n / \prod_{i=r}^{R-1} a_i \rceil$, giving

$$p_r = 1 - \left(1 - \frac{k\bar{l}_r}{mn_r}\right)^M . \tag{25}$$

One should note that there is a correspondence between a given set of parameters (m, n, k, l) and the maximal values of $p_{\max}$ and $M_{\max}$ which can be reached within the error-free (or high fidelity) retrieval regime. Storing additional patterns beyond the limit of $M_{\max}$ affects the retrieval process, either through 'add-errors' or 'miss-errors'. Determining such operating limits of Willshaw nets has been a major concern in the literature; for a historical perspective and a general treatment see the recent work of Knoblauch et al. (2010).

At this point we should also note that creating an approximation of the original memory using the windowed Boolean OR operation corresponds to storing an equal number of patterns M with roughly the same activity $\bar{l}_r \sim l \sim \log n$ but on a much smaller network — at level r we have only n_r content neurons at our disposal instead of n. The resulting weight matrix is obviously denser: by inspection of equation 25 it is clear that $p_r > p$. The aggregated matrices will easily become overloaded (i.e., with p_r beyond $p_{\max}$) and will perform the recall function with errors, which have to be refined with the progressive retrieval prescription.

Given an input pattern $\tilde{\mathbf{x}}$, we would like to determine for each memory the number of firing neurons, whose hierarchical 'siblings' will have to be checked on the next level. In other words, recalling that we denote by $\hat{\mathbf{y}}^{(r)}$ the output of the r-th memory, we want to compute the expected output activity level $\mathrm{E}(||\hat{\mathbf{y}}^{(r)}||_0)$. Unlike $||\zeta_r(\mathbf{y})^\mu||_0$, the probability mass function of this random variable is no longer trivial to compute, and we can expect it to take values other than $||\zeta_r(\mathbf{y})^\mu||$ except when $r = R$. Ideally, we would have on average $\bar{l}_r$ firing and $n_r - \bar{l}_r$ silent neurons, but due to the overload factor, a fraction of undesired (or 'spurious') firing units can appear.

[1] Note that the content neuron population size n can be seen as a function of the pattern load M; being able to store a desired number of associations while requiring high output quality imposes a restriction on the minimal allowed value for n.

The simplest approach to estimate the number of such units is to follow the well-known synaptic independence assumption (Willshaw et al., 1969; Palm, 1980, 1992). If we consider that the active synapses of a memory are generated independently, then the probability that a spurious unit fires is

$$q_r := \mathrm{P}\left(\hat{y}_j^{(r)} = 1 \mid [\zeta_r(\mathbf{y}^\mu)]_j = 0\right) \approx {p_r}^{\Theta} = {p_r}^{z}, \tag{26}$$

where we have employed the Willshaw threshold, i.e., we have set Θ to the number of correct bits on the input cue $\tilde{\mathbf{x}}$. The equality is valid for the noise-free case, when all the provided bits are correct but possibly some are missing.

With q_r at hand, we can finally derive the expected total (spurious and correct) number of firing units on a given stage r of our hierarchy:

$$u_r := \mathrm{E}(||\hat{\mathbf{y}}^{(r)}||_0) = q_r \left(n_r - \bar{l}_r\right) + \bar{l}_r. \tag{27}$$

Using u_r and our knowledge of the temporal costs of retrieval on a single-layer network (following equation 17) we are able to write the exact average time expression for the hierarchical model:

$$t := t_{\mathrm{retr}}^{\mathrm{hier}} = (z+1)\left(n_1 + \sum_{r=1}^{R-1} a_r u_r\right), \tag{28}$$

where the first member of the sum represents the cost of performing a full lookup on the memory at $r = 1$, while the following memories need only to refine $a_r \cdot u_r$ neurons thanks to the filtered dendritic sum prescription.

Of course, since we rely on equation 26 (which is an approximation) to calculate u_r, equation 28 also becomes inexact. However, it has been shown that the binomial approximation we have used for q_r is good enough under the sublinear address sparseness regime, becoming asymptotically correct e.g. when we require that $k = O(n/\log^4 n)$ (c.f. Knoblauch (2008); Knoblauch et al. (2010)). This fact is of particular importance since we will need the simplest possible form of $t_{\mathrm{retr}}^{\mathrm{hier}}$ (from now on simply referred to as t) to carry out further analytical treatment. In any case, the exact combinatorial expressions p_{Ph} or p_{Wh} derived in Knoblauch (2008) can be used when maximal accuracy is desirable.

3. Time-optimal hierarchical configurations

In the previous sections we have analysed a recently proposed hierarchical variant of the Willshaw network model (Sacramento & Wichert, 2011) in the light of common associative memory theory. We have seen that unlike standard single-layer nets, where retrieval time is a simple function of z and n (as given by equation 16), the performance of the hierarchical memory depends on a larger set of parameters. The majority of these (namely m, n, z, k, l, M) describe the network structure and pattern configuration and are a common staple in associative memory literature. However, the impact of the hierarchy depth R and the corresponding aggregation factors $a_1, \ldots, a_R$ on retrieval time has still not been fully discussed and deserves further attention.

3.1. Mathematical programming problem formulation

The essential question we should answer now is how to optimally choose such free parameters. Here we define optimal in terms of retrieval efficiency, i.e., the values of R and $a_1, \ldots, a_R$ which lead to a minimal expected temporal cost for solving the memory task defined by the remaining parameters m, n, z, k, l, M.

For compactness we will denote a hierarchy configuration choice by a single R-dimensional vector $\mathbf{a} = (a_1, ..., a_R)$. We will further denote by $\mathcal{A}_R$ the set of all valid hierarchy configurations with dimension R. Any vector of natural numbers is a candidate if a_R is equal to 1 and the remaining $a_{r \neq R}$ are greater than 1. However, this definition is loose as it allows aggregation factors combinations which are obviously uninteresting: at the last aggregation step (when $r = 1$), we can choose a value for a_1 larger than n_2 thanks to the ceiling operator in the specification of n_r, but this choice will be clearly useless since n_1 must be greater than 1. For a given depth R we can formulate this restricted set as

$$\begin{aligned} \mathcal{A}_R \;\; &= \Big\{ \mathbf{a} = (a_1 > 1, \ldots, a_r > 1, \ldots, a_R = 1) : \\ &\qquad a_r \in \mathbf{N} \wedge a_1 \leq n_2 = \left\lceil \tfrac{n}{\prod_{r=2}^{R-1} a_r} \right\rceil \Big\}. \end{aligned} \tag{29}$$

From $\mathcal{A}_R$ we can define the full search space which comprises vectors of variable dimension:

$$\mathcal{A} = \bigcup_{R=1}^{\infty} A_R. \tag{30}$$

Of course, since n_r is computed from the successive division of n by the aggregation factors $a_{R-1}, \ldots, a_{r+1}$, we know that the hierarchy depth is trivially bounded, as the lengthiest combination satisfying (29) is attained when every aggregation factor a_r (with $1 < r < R$) is equal to 2:

$$1 \leq R \leq \lceil \mathrm{ld}\, n \rceil + 1 =: \rho, \tag{31}$$

where $\mathrm{ld}\, n$ denotes the logarithm of n to the base 2.

For a given memory task specified by the parameters (m, n, z, k, l, M), the set of configurations $\mathcal{A}$ is finite and we can state our optimisation task as a set of minimisation sub-problems. For every depth $R \leq \rho$, we ought to minimise $t(\mathbf{a})$ on $\mathcal{A}_R$:

$$\underset{\mathbf{a} \in \mathcal{A}_R}{\text{minimise}} \quad t(\mathbf{a}), \tag{32}$$

which is a general integer programming problem.

Fortunately, we are able to relax the integer division constraint employed in the definition of $\mathcal{A}_R$ and then show that the cardinal of the solution space $\mathcal{A}$ is bounded from above by a polynomial of n. At every aggregation step r, if $n_{r+1}/a_r \notin \mathbf{N}$ an additional window will appear due to the use of the floor operator in the definition of n_r. However, due to the non-linear increase in the add error probability q_r with respect to $\prod_r a_r$, it is expectable that aggregation factor choices with the highest values of $\prod_r a_r$ lead to uninteresting memory configurations. As such, we redefine $\mathcal{A}_R$ while leaving aside the integer division restriction:

$$\mathcal{A}_R \;\; = \Big\{ \mathbf{a} = (a_1, \ldots, a_r > 1, \ldots, a_R = 1) : a_r \in \mathbf{N} \wedge \textstyle\prod_r a_r \leq n \Big\}. \tag{33}$$

It turns out that the determination of $|\mathcal{A}|$ when $\mathcal{A}_R$ assumes this new form is a variation of a famous problem from number theory generally referred to as the Kalmár problem on *factorisatio numerorum* (Kalmár, 1931; Hille, 1936; Knopfmacher & Mays, 2005), the only difference lying on the kind of relation which in our case is an inequality instead of a strict equality. The classical version of the problem — which has found application in other tasks such as computational biology (Newberg & Naor, 1993) — is concerned with finding bounds and algorithms to compute a certain counting function $H(n)$, defined as the number of ordered factorisations of a natural number n composed of factors greater than or equal to 2. We note that $|\mathcal{A}|$ can be defined in terms of $H(n)$:

$$|\mathcal{A}| = H(n) + H(n-1) + H(n-2) + \ldots + H(1), \tag{34}$$

where $H(1)$ is taken to be 1 by definition. Determining good approximations for $H(n)$ is still a challenging problem, but it is known that in the worst case (for highly factorable numbers) the counting function grows with $O(n^b)$, with $b = \zeta^{-1}(2) \approx 1.73$, ζ in this case being the well-known Riemann zeta function (Hille, 1936; Chor et al., 2000). Our cardinal $|\mathcal{A}|$ then clearly behaves as a polynomial of n.

Although a tractable problem, the search space cardinality $|\mathcal{A}|$ is still supralinear in n. Computing exact solutions for the optimisation task in high-dimensional spaces could be problematic, as solving the integer programming problem through enumeration implies measuring the number of comparisons performed by a memory model implementation or evaluating t for each $\mathbf{a} \in \mathcal{A}$. Our aim on the next section is to analyse an approximated (simplified) version of the problem in order to further restrict the optimisation space to a subset of $\mathcal{A}$ with sublinear cardinality in n, so as to reduce the time complexity of the minimisation problem at hand.

3.2. Approximate analysis using real-valued parameters

A common procedure to gain further insight on integer programming problems is to perform a relaxation, i.e., one lifts the integrality constraints on the optimisation variables and then uses tools from real calculus to study the objective function. In our case, the total synaptic activity $t(\mathbf{a})$ can be regarded as a function of real variables if we let the aggregation factors a_r be real. We may then resort to differentiation to analyse t, and proceed with numerical experimentation to investigate if the real approximation is close enough to be useful. Unfortunately, performing an analytical minimisation of (28) would imply finding a closed-form solution for $\frac{\partial t}{\partial \mathbf{a}} = 0$, which by itself is difficult to accomplish and as such some approximations are in order.

We consider the case of a simplistic ideal hierarchical memory with a density p low enough to maintain the aggregated memories capable of operating in a high fidelity regime. That is to say that q_r is small enough to allow for the approximation $u_r \approx l$. This approximation also implies $\forall r,\ \bar{l}_r = l$, i.e., we assume that the probability of a reduction of the number of '1' entries of each content pattern $\mathbf{y}^\mu$ throughout the aggregation process is negligible, which is a reasonable assumption for sparse uniform random patterns. Since $n_1 \geq l$ it is always possible to handcraft such patterns, and for large enough M, n we expect the uniform random variable to produce indices such that $||\zeta_r(\mathbf{y}^\mu)||_0 \approx l$. Therefore, rewriting (28) we obtain

$$t \approx (z+1)\left(n_1 + l\sum_{r=1}^{R-1} a_r\right) =: t^*. \tag{35}$$

We can now easily minimise t^* to derive a set of ideal $a_r^* \in \mathbf{R}^+$ and $R^* \in \mathbf{R}^+$ (cf. Appendix A):

$$\begin{aligned} R^* &= \log\left(\frac{n}{l}\right), \\ a_r^* &= e = 2.71828+, \forall r < R^*, \end{aligned} \tag{36}$$

which suggest a hierarchical structure resembling a tree.

The minimising depth $\log(n/l)$ we have just computed can as well be reached without explicitly analysing the approximation t^*. An asymptotic upper bound for R can be immediately derived if we take into account the sparseness requirement of $l \sim \log n$. As the content patterns are uniformly distributed across the $\{0,1\}^n$ pattern space, to perform correctly every level r must have at least $\bar{l}_r$ neurons. Choosing the lengthiest combination of factors asymptotically corresponds to letting $\forall r,\ a_r = 2 = O(1)$. Assuming $\bar{l}_r \sim l$, we can enforce the first level to have at least l neurons:

$$n_1 \sim \frac{n}{a^{R-1}} \gtrsim l. \tag{37}$$

or, in terms of R,

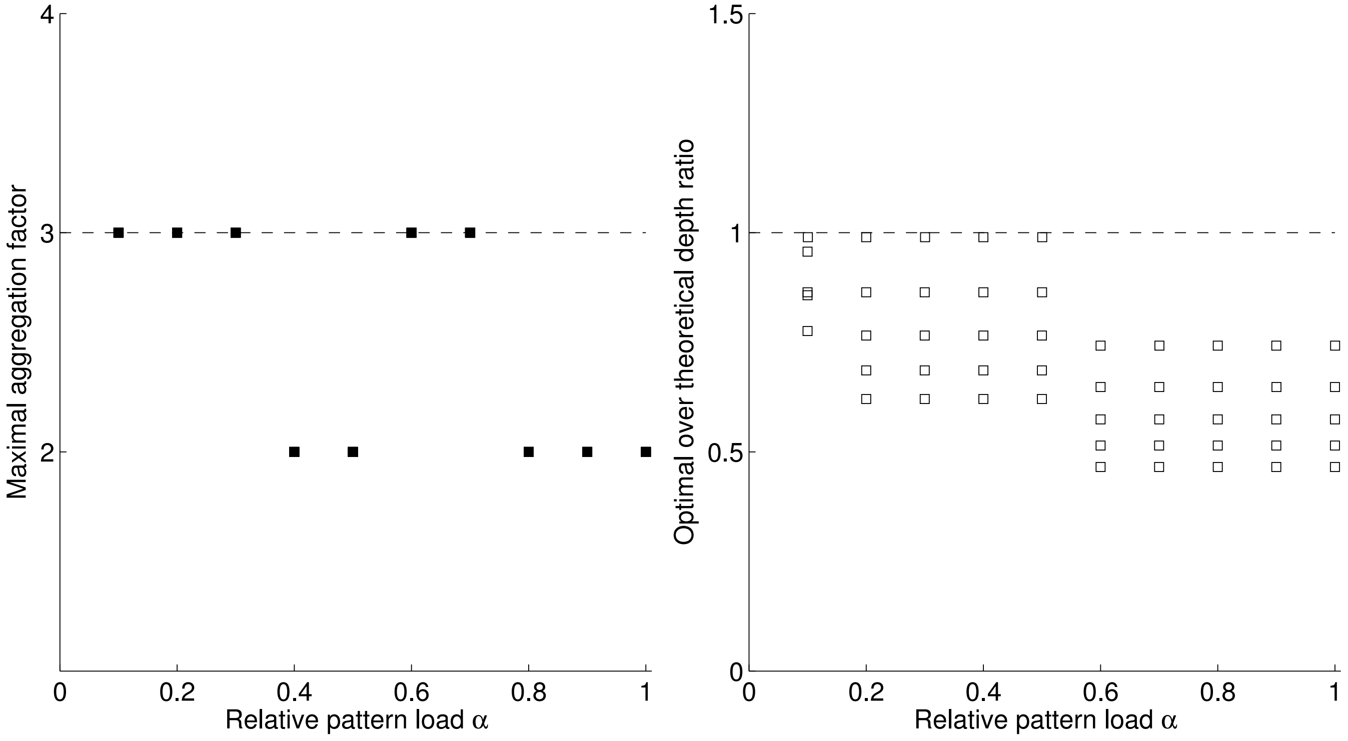
$$R = O\left(\log\left(n/l\right)\right). \tag{38}$$

This bound is as tight as possible if we do not possess any knowledge on the pattern load M.

The fact that R^* corresponds exactly to the maximal possible (useful) value for R is actually not surprising. We note that t^* closely describes the expected retrieval time when the pattern load M is low. In fact, for sufficiently large $m \sim n$, we can show that $t(\mathbf{a}) \geq t^*(\mathbf{a})$, the equality holding if $M = 1$. As when $M = 1$ there is no synaptic interference (only one pattern is stored in each layer), we have $\forall r,\ q_r = 0$, yielding $u_r = \bar{l}_r = l$. Since q_r is a monotone increasing function of M, and t is a monotone increasing function of q_r, we can state that $t \geq t^*$.

Figure 1 shows the results of minimising the average retrieval time required by a sequential computer implementation of the hierarchical model. For a sequence of network characterisations where $m = n$, $z = k = l = \operatorname{ld} n$ and varying pattern loads given by a simple step progression $M = \alpha \cdot M_{\max}$ with $\alpha \in \{1/10, 2/10, \ldots, 1\}$, we have measured the retrieval time required by each hierarchy configuration drawn from $\mathcal{A}$ averaged over every learnt association pair $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ and then registered the optimal aggregation factor combination $\mathbf{a}^{\text{opt}}$. For every characterisation not only the (now integer-valued) optimal depth verified $R^{\text{opt}} \leq \lceil R^* \rceil$, but also each optimising aggregation factor component did indeed observe $\forall r,\ \max(a_r^{\text{opt}}) = \lceil e \rceil = 3$. Thus, in every experiment we have carried out, $\mathbf{a}^{\text{opt}}$ would have been found by minimising t through enumeration in the subset of $\mathcal{A}$ containing only configurations such that $R \in \{2, 3, \ldots, R^*\}$ and $a_r \in \{2, 3\}$. Note that the cardinal of this restricted search space is $\sum_{i=1}^{R-1} 2^i = 2^R - 2 = O(n/l)$, where $R = R^*$.

However, the empirical nature of the former results should be stressed. Technically, the derivation which led to the optimal parametrisation stated in (36) requires a pattern load much below the theoretical maximum (which is a function $M_{\max}(m,n)$ of network size, among other parameters) achievable by the R-th network, so that the aggregated memory layers can remain in the high-fidelity regime where $q_r \approx 0$. Taking $a_r = a_r^*$ and $R = R^*$ as in (36) implies that the first layer will have a content population dimension of



(a)

(b)

Figure 1: The maximal observed value of a_r^{opt} (shown in Figure 1a) and the optimal over theoretical depth ratio $R^{\mathrm{opt}}/\lceil R^* \rceil$ (shown in Figure 1b), both as a function of the relative pattern load $\alpha = M/M_{\max}$. The minimisation task of determining the optimal configuration $\mathbf{a}^{\mathrm{opt}}$ for each network characterisation was solved through enumeration: for every $\mathbf{a} \in \mathcal{A}$, the corresponding hierarchical memory was implemented and the set S of M random pattern associations learnt; then, the configuration $\mathbf{a}$ which led to the minimal observed retrieval time (averaged over all $\mathbf{x}^\mu \in S$) was chosen as $\mathbf{a}^{\mathrm{opt}}$. This task was carried out for a sequence of memory settings where $m = n$, $z = k = l = \mathrm{ld}\, n$ and $M = \alpha \cdot M_{\max}$, while letting $n \in \{512, 1024, 2048, 4096, 8192\}$ and $\alpha \in \{1/10, 2/10, \ldots, 1\}$. To compute $M_{\max}$ for each setting we used the formulae for finite memories derived in section 3.2 of (Knoblauch et al., 2010). Notice how for any of the above we have $a_r^{\mathrm{opt}} \leq \lceil e \rceil = 3$ and $R^{\mathrm{opt}} \leq \lceil R^* \rceil$. It is also interesting to observe how, unlike R^{opt}, $\max(a_r^{\mathrm{opt}})$ does not depend on n or l (for each α, all points are overlapping), as predicted by the approximate analysis.

size $n_1 = n/e^{\log n/l} = l$; considering the activity level $\bar{l}_1 \approx l$, to maintain $q_1 \approx 0$ we can then expect to be able to store only an extremely small number of patterns M (Knoblauch et al., 2010). Fortunately, numerical optimisation across a variety of network settings has shown that in practice the integer a_r factors lie always very close to the theoretical solution $a_r^* = e$, while the true (integer) optimal depth R^{opt} slowly decreases from $\lceil R^* \rceil$ to 2 as M increases up to $M_{\max}$.

3.3. Worst-case asymptotic comparison with single-layer nets

In a similar fashion to Palm (1980) we can derive simpler expressions if we only require our results to be strictly valid when $m \sim n \to \infty$. Finite memories can operate with a near-zero level of errors when $M < M_{\max} := cmn/\operatorname{ld}(n)^2, c < \ln 2$ and $p < 1/2$. Numerical optimisation methods reveal approximate values for c and p for a given set of remaining network parameters (Palm, 1980; Knoblauch et al., 2010). Unfortunately, to our knowledge no tractable closed-form expressions have been derived until now. However, under the asymptotically optimal regime (i.e., with $m = n \to \infty$ and $k = l = \operatorname{ld} n$), it has been shown that $c = \ln 2$ and $p = 1/2$. Resorting to these bounds we can try to analyse and optimise t with respect to $\mathbf{a}$ for the case of maximally loaded hierarchical memories.

We analyse the case of $R = 2$, which is the most tractable, even if not necessarily the optimal. To shorten the notation we let $a := a_1$. Expanding equation 28 we get

$$\begin{aligned} t &= (z+1)\left(\frac{n}{a} + au_1\right) && (39)\\ &= (z+1)\left[\frac{n}{a} + a\left(q_1\left(\frac{n}{a} - \bar{l}_1\right) + \bar{l}_1\right)\right] && (40)\\ &\approx z\left(\frac{n}{a} + aq_1\frac{n}{a}\right) = zn\left(\frac{1}{a} + q_1\right), && (41)\end{aligned}$$

where we have discarded the threshold count and we have used the approximation $u_1 \approx q_1 n/a$. Of course, this is not strictly true but since the aggregation of the maximally loaded memory at $r = R$ will generate a highly erroneous memory at $r = 1$ (even when $a = 2$) we assume that the spurious unit count is the dominating factor.

Now, employing the binomial approximation as in equation 26 we get $q_1 \approx {p_1}^z$. However, the expression for p_1 of (25) is not trivial to analyse, and we would rather prefer to compute p_1 as a function of p_R to take advantage of the asymptotic bound $p_R = 1/2$. This can be done if instead of applying the aggregation process to each content pattern μ one by one we consider the event of aggregating each row of the original weight matrix $\mathbf{W}$. Similarly to (24), we resort to the probability of the complementary event of not finding a '1' component in the entire synaptic connectivity row:

$$p_1 = 1 - \mathrm{P}(W_{ij} = 0)^a = 1 - (1 - p_R)^a. \qquad (42)$$

Inserting (42) into (41) using the approximation $q_1 \approx {p_1}^z$, we reach

$$t \approx zn\left(\frac{1}{a} + (1 - (1 - p_R)^a)^z\right). \qquad (43)$$

When $m, n \to \infty$, we obtain $p_1 = 1 - 2^{-a}$ and reevaluating the latter equation,

$$t \approx zn\left(\frac{1}{a} + (1 - 2^{-a})^z\right). \qquad (44)$$

Even from (44) it is difficult to derive a closed-form expression for the optimal a value. However, through numerical inspection, we can see that this value is (extremely) slowly growing with n; for instance, when $n = 10^3$, the optimal choice is $a = 2$, but when $n = 10^6$ we should rather choose $a = 3$. This rather surprising fact is related with the likewise slowly growing c originally computed by Palm — larger memories display better capacities.

Asymptotically, $a \sim f(n)$ becomes clear. Under the maximal load assumption, how does the $R = 2$ hierarchical retrieval time compare with that of the original single-layer memory? We can analyse how the ratio $\tau := t/t^{\mathrm{W}}_{\mathrm{retr}}$ behaves in the limit of $n \to \infty$:

$$\begin{aligned}\lim_{n\to\infty} \tau = \lim_{n\to\infty} \frac{t}{t^{\mathrm{W}}_{\mathrm{retr}}} \quad &\approx \quad \lim_{n\to\infty} \frac{zn\left(1/a + (1-2^{-a})^z\right)}{zn} \\ &= \quad \lim_{n\to\infty} \left(\frac{1}{a} + (1-2^{-a})^z\right). \end{aligned} \tag{45}$$

Choosing the optimal aggregation factor (asymptotically) corresponds to taking a value of a such that $\tau \to 0$, which requires vanishing $(1-2^{-a})^z$ or, equivalently, diverging $z2^{-a}$ towards ∞. Taking logarithms yields $\log z + \log 2^{-a} \sim \log z - a$, implying $a \ll \log z$. As $z \sim \log n$ by the sparseness requirement, we obtain an asymptotic value for a as a function f of network size:

$$a \sim f(n) = \log\log n. \tag{46}$$

By construction, for $a \sim f(n)$, when $n \to \infty$ the term $(1 - 2^{-a})^z$ in equation 45 vanishes, and the ratio τ becomes $1/a = 1/\log\log n$, implying that the hierarchical search method is asymptotically superior in terms of retrieval time when compared to the single-layer model even for maximally loaded memories, where the aggregation process is highly limited, but only for an improvement margin of the order of $\log\log n$. Figure 2 illustrates the slowly decreasing τ when we optimise t for growing n.

4. Discussion

As with most associative memory models, Willshaw networks benefit from fully parallel (i.e., at the synapse level) or n-parallel (one processing unit per content neuron) implementations in specialised hardware, where the distributed nature of the storage model can be entirely exploited and constant $O(1)$ or logarithmic $O(\log n)$ retrieval times can be achieved. However, in practice such specific realisations may not be available and one may be restrained to a sequential procedure capable of being executed on a typical von Neumann computer.

It has been argued whether for binary sparse patterns a Willshaw-type network is able to outperform strictly artificial nearest neighbour (NN) determination techniques (such as locality-sensitive hashing algorithms; see e.g. the survey of Andoni & Indyk (2008)) on sequential hardware (Palm, 1987). This debate is still interesting since many technical applications rely on NN for pattern recognition or classification over large data sets, and the so-called 'curse of dimensionality' has been undermining more sophisticated exact algorithms when the content or address space dimensions become very large; most methods simply degenerate into Hamming-distance lookup tables or perform even worse (Weber et al., 1998). As strong retrieval error guarantees have been computed for finite-sized Willshaw-type networks under various regimes (see Knoblauch et al. (2010) for an

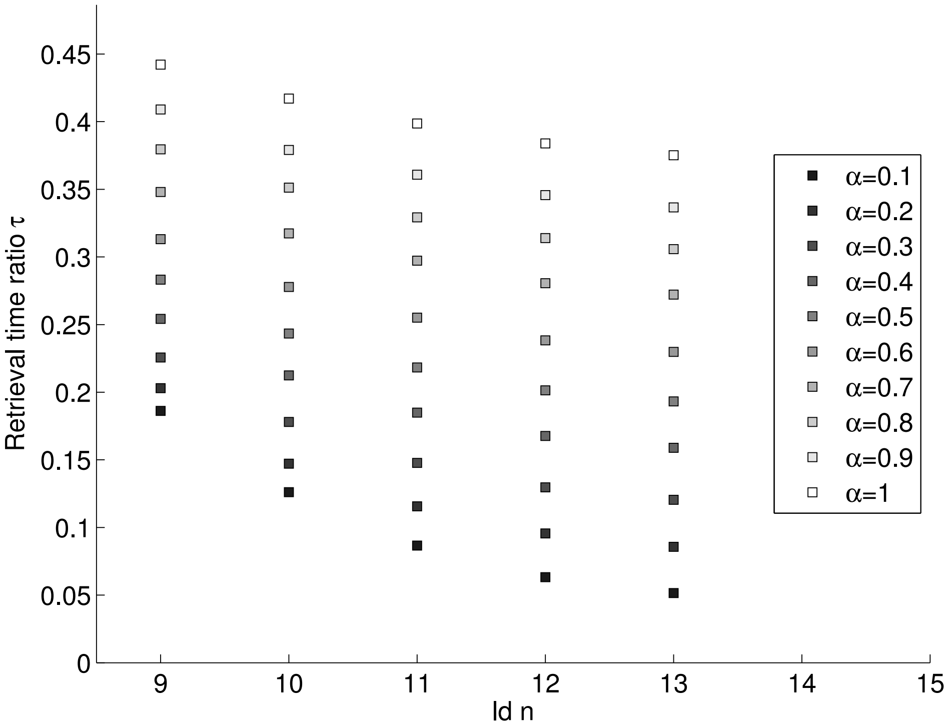


Figure 2: The evolution of the ratio $\tau = t/t_{\mathrm{retr}}^{\mathrm{W}}$ for increasing values of n. For each setting (α, n), where $\alpha = M/M_{\max}$ is the relative memory load, we determined the optimal configuration $\mathbf{a}$ and then used the resulting hierarchical time t^{measured} to compute $\tau = t^{\mathrm{measured}}/t_{\mathrm{retr}}^{\mathrm{W}}$. Once again, for all points, $z = k = l = \mathrm{ld}\, n$ and $m = n$, and $M_{\max}$ was derived using the results of section 3.2 of (Knoblauch et al., 2010).

up-to-date analysis), accelerating the original sequential lookup procedure could increase the appeal of such networks as technical sparse pattern recognition devices.

In this article we have analysed in greater detail the R-layered progressive recall extension proposed in (Sacramento & Wichert, 2011). It was shown that the method displays some interesting properties arising from the distributed Hebbian storage rule of the Willshaw net. In fact, due to the sparse coding requirement, the fraction of firing content neurons $l/n \to 0$ as $n \to \infty$ and so does the required time of the method relatively to the original single-layer prescription, reinforcing the fact that the (heavy) coding restriction is indeed an advantage in terms of retrieval. This result should hold even for the maximal load scenario analysed in section 3.3 (where aggregation might have seemed unreasonable), albeit with a modest improvement of the order of $\log \log n$.

An interesting open path is to explore the application of a similar hierarchical retrieval procedure to neural associative networks that are more robust (in terms of noise) and flexible (in terms of the sparseness requirement). It is known that a storage capacity of 0.72 bps can be achieved in the noisy-inputs regime for sublinear, yet supralogarithmic activity levels by resorting to real-valued synapses and more elaborate learning prescriptions, such as the optimal local linear synaptic modification rule (Sejnowski, 1977; Willshaw & Dayan, 1990; Dayan & Willshaw, 1991; Palm & Sommer, 1996) — usually referred to as the covariance rule — or the more general optimal local non-linear rule derived from a Bayesian-probabilistic perspective (Knoblauch, 2011). Quite promising results have also been achieved recently with 'zip nets' which lift the need for continuous synapses by employing discrete connectivity levels (favoured from the implementation point-of-view) controlled by a synaptic thresholding mechanism (Knoblauch, 2010). Determining activity level orders and a threshold setting so that a hierarchy of these networks would outperform the simpler model that we have considered here, both in terms of retrieval time and query noise tolerance, seems an interesting future direction. For zip nets (as a matter a fact, for Willshaw nets as well), one could even try to apply a combination of compression schemes as described in (Knoblauch et al., 2010) with hierarchical retrieval, to further increase the model's efficiency.

Another approach rather different from the one followed here would be to consider a biophysically constrained scenario where the interactions of neurons of a more realistic kind would be studied in the light of their energetic requirements. It would be interesting to investigate whether the abstract filtering mechanism resembles or is in some sense related to a biological transmission process, as it is now believed that the balance between raw computing performance (e.g., in the form of time or capacity) and energy consumption has taken a major role in the development of the mammalian brain (Levy & Baxter, 1996; Laughlin, 2001; Laughlin & Sejnowski, 2003). We leave these questions open in the speculation realm, while hoping to address them in future work.

Acknowledgements

The authors are indebted to an anonymous reviewer which contributed with many detailed comments, including an improved asymptotic bound $f(n)$ on a, and to Ângelo Cardoso and Francisco S. Melo for their suggestions on the first version of this manuscript. This work was supported by *Fundação para a Ciência e Tecnologia* (INESC-ID multiannual funding) through the PIDDAC Program funds and through an individual doctoral grant awarded to the first author (contract SFRH/BD/66398/2009).

Appendix A. Derivation of closed-form expressions for a_r^* and R^* under the error-free memory approximation

According to equation 35, the approximate total synaptic activity is given by

$$t^*(a_1, a_2, \ldots, a_{R-1}) = \frac{zn}{\prod_{r=1}^{R-1} a_r} + zl \sum_{r=1}^{R-1} a_r, \tag{A.1}$$

which can be easily minimised for a given a_i,

$$\frac{\partial t^*(a_i)}{\partial a_i} = -\frac{zn}{a_i \prod_{r=1}^{R-1} a_r} + zl = 0, \tag{A.2}$$

where a_i can be any of the $R-1$ aggregation factors. Rearranging (A.2) yields

$$a_i = \frac{n}{l} \frac{1}{\prod_{r=1}^{R-1} a_r}, \forall i, \tag{A.3}$$

and since the product is the same for all a_i, it is therefore necessary that $a_i = a$, $\forall i$. As such,

$$\begin{aligned} a &= \frac{n}{la^{R-1}} \\ a &= \left(\frac{n}{l}\right)^{1/R} \Leftrightarrow R = \frac{\log(n/l)}{\log a}. \end{aligned} \tag{A.4}$$

It is now necessary to find the optimal R that minimises t^*. The variable R is not, at this point, a variable of the synaptic activity. But since all factors are equal between them, one can redefine the function, as

$$t^*(a, R) = zna^{1-R} + zl(R-1)a, \tag{A.5}$$

without loss of generality for local minima. Its derivative in terms of R will be

$$\begin{aligned} \frac{\partial t^*(a,R)}{\partial R} &= -zn \log a a^{1-R} + zla = 0 \\ R &= \frac{\log((n/l)\log a)}{\log a}, \end{aligned} \tag{A.6}$$

and a simple comparison with the result for R given by (A.4) shows that

$$\begin{aligned} a^* &= e = 2.718281+, \\ R^* &= \log\left(\frac{n}{l}\right). \end{aligned} \tag{A.7}$$

We must finally show that this is, indeed, the global minimum point of t^*. Second partial derivatives yield

$$\begin{aligned} \left.\frac{\partial^2 t^*(a,R)}{\partial a^2}\right|_{(a^*,R^*)} &= \frac{zl}{e} \log\left(\frac{n}{l}\right)\left[\log\left(\frac{n}{l}\right) - 1\right] \\ \left.\frac{\partial^2 t^*(a,R)}{\partial R^2}\right|_{(a^*,R^*)} &= zle > 0 \\ \left.\frac{\partial^2 t^*(a,R)}{\partial a \partial R}\right|_{(a^*,R^*)} &= zl\left[\log\left(\frac{n}{l}\right) - 1\right], \end{aligned} \tag{A.8}$$

and the Hessian determinant is

$$\det H(a^*, R^*) = z^2 l^2 \left[\log\left(\frac{n}{l}\right) - 1\right] > 0. \tag{A.9}$$

References

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

Amit, D. J., Gutfreund, H., & Sompolinsky, H. (1985). Spin-glass models of neural networks. *Physical Review A*, 32(2), 1007–1018.

Andoni, A. & Indyk, P. (2008). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions". *Communications of the ACM*, 51(1), 117–122.

Barlow, H. B. (1972). Single units and sensation: A neuron doctrine for perceptual psychology? *Perception*, 1(4), 371–394.

Bentz, H. J., Hagström, M., & Palm, G. (1989). Information storage and effective data retrieval in sparse matrices. *Neural Networks*, 2(4), 289–293.

Bogacz, R. & Brown, M. W. (2003). Comparison of computational models of familiarity discrimination in the perirhinal cortex. *Hippocampus*, 13(4), 494–524.

Bogacz, R., Brown, M. W., & Giraud-Carrier, C. (2001). Model of familiarity discrimination in the perirhinal cortex. *Journal of Computational Neuroscience*, 10(1), 5–23.

Buckingham, J. (1991). *Delicate nets, faint recollections: a study of partially connected associative network memories*. PhD thesis, University of Edinburgh.

Buckingham, J. & Willshaw, D. (1992). Performance characteristics of the associative net. *Network: Computation in Neural Systems*, 3(4), 407–414.

Buckingham, J. & Willshaw, D. (1993). On setting unit thresholds in an incompletely connected associative net. *Network: Computation in Neural Systems*, 4(4), 441–459.

Chor, B., Lemke, P., & Mador, Z. (2000). On the number of ordered factorizations of natural numbers. *Discrete Mathematics*, 214(1-3), 123–133.

Cover, T. & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21–27.

Cover, T. M. & Thomas, J. A. (2006). *Elements of information theory*. New York, NY: Wiley-Interscience, second edition.

Dayan, P. & Willshaw, D. (1991). Optimising synaptic learning rules in linear associative memories. *Biological Cybernetics*, 65(4), 253–265.

Derrida, B., Gardner, E., & Zippelius, A. (1987). An exactly solvable asymmetric neural network model. *Europhysics Letters*, 4(2), 167–178.

Field, D. J. (1994). What is the goal of sensory coding? *Neural Computation*, 6(4), 559–601.

Fix, E. & Hodges, J. L. (1951). *Discriminatory analysis, nonparametric discrimination: consistency properties*. Project 21-49-004, Rept. 4, Contract AF41(128)-31, USAF School of Avation Medicine.

Gardner, E. (1988). The space of interactions in neural network models. *Journal of Physics A: Mathematical and General*, 21(1), 257–270.

Gardner-Medwin, A. R. (1976). The recall of events through the learning of associations between their parts. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 194(1116), 375–402.

Golomb, D., Rubin, N., & Sompolinsky, H. (1990). Willshaw model: Associative memory with sparse coding and low firing rates. *Physical Review A*, 41(4), 1843–1854.

Graham, B. & Willshaw, D. (1995). Improving recall from an associative memory. *Biological Cybernetics*, 72(4), 337–346.

Greve, A., Sterratt, D., Donaldson, D., Willshaw, D., & van Rossum, M. (2009). Optimal learning rules for familiarity detection. *Biological Cybernetics*, 100(1), 11–19.

Hebb, D. O. (1949). *The organization of behavior: a neuropsychological theory*. New York, NY: Wiley-Interscience.

Hille, E. (1936). A problem in "factorisatio numerorum". *Acta Arithmetica*, 2, 134–144.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8), 2554–2558.

Kalmár, L. (1931). Über die mittlere Anzahl der Produktdarstellungen der Zahlen. *Acta Litterarum ac Scientarum Szeged*, 5, 95–107.

Knoblauch, A. (2008). Neural associative memory and the Willshaw–Palm probability distribution. *SIAM Journal on Applied Mathematics*, 69(1), 169–196.

Knoblauch, A. (2010). Zip nets: Efficient associative computation with binary synapses. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)* (pp. 4271–4278). Piscataway, NJ: IEEE Press.

Knoblauch, A. (2011). Neural associative memory with optimal bayesian learning. *Neural Computation*, 23(6), 1393–1451.

Knoblauch, A., Palm, G., & Sommer, F. T. (2010). Memory capacities for synaptic and structural plasticity. *Neural Computation*, 22(2), 289–341.

Knopfmacher, A. & Mays, M. E. (2005). A survey of factorization counting functions. *International Journal of Number Theory*, 1(4), 563–581.

Kosko, B. (1987). Adaptive bidirectional associative memories. *Applied Optics*, 26(23), 4947–4960.

Laughlin, S. B. (2001). Energy as a constraint on the coding and processing of sensory information. *Current Opinion in Neurobiology*, 11(4), 475–480.

Laughlin, S. B. & Sejnowski, T. J. (2003). Communication in neuronal networks. *Science*, 301(5641), 1870–1874.

Lennie, P. (2003). The cost of cortical computation. *Current Biology*, 13(6), 493–497.

Levy, W. B. & Baxter, R. A. (1996). Energy efficient neural codes. *Neural Computation*, 8(3), 531–543.

Marr, D. (1971). Simple Memory: A Theory for Archicortex. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 262(841), 23–81.

McCulloch, W. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5(4), 115–133.

Minsky, M. L. & Papert, S. A. (1969). *Perceptrons: an introduction to computational geometry*. Cambridge, MA: MIT Press.

Nadal, J.-P. & Toulouse, G. (1990). Information storage in sparsely coded memory nets. *Network: Computation in Neural Systems*, 1, 61–74(14).

Newberg, L. A. & Naor, D. (1993). A lower bound on the number of solutions to the probed partial digest problem. *Advances in Applied Mathematics*, 14(2), 172–183.

Olshausen, B. A. & Field, D. J. (2004). Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14(4), 481 – 487.

Palm, G. (1980). On associative memory. *Biological Cybernetics*, 36(1), 19–31.

Palm, G. (1987). Computing with neural networks. *Science*, 235(4793), 1227b–1228.

Palm, G. (1992). On the information storage capacity of local learning rules. *Neural Computation*, 4(5), 703–711.

Palm, G. & Sommer, F. T. (1992). Information capacity in recurrent McCulloch-Pitts networks with sparsely coded memory states. *Network: Computation in Neural Systems*, 3(10), 177–186.

Palm, G. & Sommer, F. T. (1996). Associative data storage and retrieval in neural networks. In *Models of Neural Networks: Association, Generalization, and Representation (Physics of Neural Networks)*, volume 3 (pp. 79–118). New York, NY: Springer.

Sacramento, J. & Wichert, A. (2011). Tree-like hierarchical associative memory structures. *Neural Networks*, 24(2), 143–147.

Sejnowski, T. J. (1977). Storing covariance with nonlinearly interacting neurons. *Journal of Mathematical Biology*, 4(4), 303–321.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27, 379–423, 623–656.

Sommer, F. T. & Palm, G. (1999). Improved bidirectional retrieval of sparse patterns stored by Hebbian learning. *Neural Networks*, 12(2), 281–297.

Steinbuch, K. (1961). Die lernmatrix. *Biological Cybernetics*, 1(1), 36–45.

Weber, R., Schek, H.-J., & Blott, S. (1998). A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In A. Gupta, O. Shmueli, & J. Widom (Eds.), *Proceedings of 24rd International Conference on Very Large Data Bases* (pp. 194–205). New York, NY: Morgan Kaufmann.

Willshaw, D. & Dayan, P. (1990). Optimal plasticity from matrix memories: what goes up must come down. *Neural Computation*, 2(1), 85–93.

Willshaw, D. J., Buneman, O. P., & Longuet-Higgins, H. C. (1969). Non-holographic associative memory. *Nature*, 222(5197), 960–962.