Random Sketching, Clustering, and Short-Term Memory in Spiking Neural Networks

Yael Hitron
Weizmann Institute of Science, Rehovot, Israel
yael.hitron@weizmann.ac.il

Nancy Lynch
Massachusetts Institute of Technology, Cambridge, MA, USA
lynch@csail.mit.edu

Cameron Musco
University of Massachusetts, Amherst, MA, USA
cmusco@cs.umass.edu

Merav Parter
Weizmann Institute of Science, Rehovot, Israel
merav.parter@weizmann.ac.il

Abstract

We study input compression in a biologically inspired model of neural computation. We demonstrate that a network consisting of a random projection step (implemented via random synaptic connectivity) followed by a sparsification step (implemented via winner-take-all competition) can reduce well-separated high-dimensional input vectors to well-separated low-dimensional vectors. By augmenting our network with a third module, we can efficiently map each input (along with any small perturbations of the input) to a unique *representative neuron*, solving a neural clustering problem.

Both the size of our network and its processing time, i.e., the time it takes the network to compute the compressed output given a presented input, are independent of the (potentially large) dimension of the input patterns and depend only on the number of distinct inputs that the network must encode and the pairwise relative Hamming distance between these inputs. The first two steps of our construction mirror known biological networks, for example, in the fruit fly olfactory system [9, 29, 17]. Our analysis helps provide a theoretical understanding of these networks and lay a foundation for how random compression and input memorization may be implemented in biological neural networks.

Technically, a contribution in our network design is the implementation of a *short-term* memory. Our network can be given a desired *memory time* t_m as an input parameter and satisfies the following with high probability: any pattern presented several times within a time window of t_m rounds will be mapped to a single representative output neuron. However, a pattern not presented for $c \cdot t_m$ rounds for some constant $c > 1$ will be "forgotten", and its representative output neuron will be released, to accommodate newly introduced patterns.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases biological distributed computing, spiking neural networks, compressed sensing, clustering, random projection, dimensionality reduction, winner-take-all

Digital Object Identifier 10.4230/LIPIcs.ITCS.2020.23

1 Introduction

In this work we study brain-like networks that receive potentially complex and high-dimensional inputs (e.g., from sensory neurons representing odors, faces, or sounds) and encode these inputs in a very compressed way. Specifically, we consider networks with n input neurons and k output neurons, where n may be very large. When presented with up to k sufficiently different but otherwise arbitrary input patterns, the goal of the network is

11th Innovations in Theoretical Computer Science Conference (ITCS 2020).
Editor: Thomas Vidick; Article No. 23; pp. 23:1–23:31
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to represent the inputs in such a way that they can be recognized when presented again: each input should be uniquely mapped to a single *representative output neuron* that fires if that input pattern is reintroduced. Further, any small perturbations of a presented input should be recognized by the same representative neuron. We call the above problem the *neural clustering problem.*

Clustering, input memorization, and compression are fundamental problems in biological neural networks. Our work is also inspired by the important *novelty detection* problem [25, 41]. Novelty detection requires detecting inputs that differ significantly from previously seen inputs. It is easy to see that this problem can be solved with a neural clustering network, in which all sufficiently far inputs are mapped to different representative neurons and all sufficiently close inputs are mapped to the same neuron. A novel input is detected whenever a new representative neuron is assigned. The novelty detection problem has been considered recently in the fruit fly olfactory system [16], where it is believed to be solved using a random projection based method. The high level structure of this method closely resembles the initial stages of our clustering algorithm, and we see a major contribution of our work as providing a theoretical understanding of how random projection can be implemented in biologically inspired neural networks. For further discussion about the connection to fruit fly novelty detection see Section 1.2.

1.1 Our Results

We study the neural clustering problem in a biologically inspired model of *stochastic spiking neural networks* (stochastic SNNs), which was previously defined in [33, 34, 35]. In these networks, computation proceeds in discrete rounds with each neuron either firing (spiking) in a round or remaining silent. Each neuron spikes randomly, with probability determined by its membrane potential. This potential is induced by spikes from neighboring neurons, which can have either an excitatory or inhibitory effect (increasing or decreasing the potential). In general, the input to an SNN is a stream of binary vectors, corresponding to spikes of the input neurons. In our setting we will consider a single binary vector as the input pattern and assume that each input vector is presented for a certain number of consecutive rounds before changing. This allows the network time to stabilize to the correct output associated with the given input.

We demonstrate that clustering can be solved efficiently in these networks, where the cost is measured by (i) the number of *auxiliary neurons*, besides the input and output neurons, that are required to solve the clustering task and (ii) the number of rounds required to converge to the correct output for a given input, which corresponds to the number of rounds for which the input must be presented for before moving to the next input.

In the clustering problem, we consider a (potentially large) set of n-length patterns that are clustered around k base patterns. It is then required to map all patterns in the same cluster to a unique output in $[k]$.

Clustering with Output Reassignment. We also want our network to be reusable, with a *memory duration* t_m that is given as an input parameter. Instead of considering a single infinite input stream with at most k distinct patterns (or clusters of patterns), our memory module allows one having many distinct patterns, as long as their presentation times are sufficiently spaced out. That is, in any window of $\Theta(t_m)$ rounds, the network is presented at most k distinct patterns. To handle distinct patterns in each $\Theta(t_m)$-round window, the network must *forget* patterns that have not been introduced for a while and release their allocated outputs so that they can be assigned to new inputs. Specifically, for some fixed constant c, our network remembers a pattern (its cluster) for at least t_m rounds and at most $c \cdot t_m$ round. The output of any pattern not introduced for $c \cdot t_m$ rounds is released with high probability and can be reassigned to represent another input.

The Neural Clustering Problem. We now formally define the neural clustering problem, which is parametrized by several parameters: the input dimension n, the number of distinct input patterns k, the memory duration t_m, a bound on the relative distance of input patterns Δ, and the allowed failure probability δ. We require that every pattern introduced as input, remains the input pattern for at least $t_p = \text{poly}(k, 1/\Delta, \log(1/\delta))$ (i.e., independent of n) consecutive rounds. The t_p parameter is the processing time or mapping time, i.e., the time it takes for the network to converge to the output neuron. Throughout, we will assume that all patterns have p non-zero entries. We conjecture that this assumption can be easily removed however keep it to simplify our arguments.

Define the *relative Hamming distance* between two inputs $\overline{X}_i, \overline{X}_j \in \{0,1\}^n$ to be:

$$\mathcal{RD}(\overline{X}_i, \overline{X}_j) = \frac{||\overline{X}_i - \overline{X}_j||_1}{\max\{||\overline{X}_i||_1, ||\overline{X}_j||_1\}} \ .$$

In the basic clustering problem, the network is introduced to a possibly large number of distinct patterns that are *clustered* around k-centers. That is, in every window of t_m rounds, the patterns introduced are clustered around a base-set of k patterns $\overline{X}_1, \ldots, \overline{X}_k \in \{0,1\}^n$ such that the relative difference between each pair in the base-set is at least Δ, and any other pattern introduced is sufficiently close to one of the patterns in the base-set (with relative distance $\leq \Delta/\alpha$ for some $\alpha = \tilde{O}(1)$). In the clustering problem the network maps *similar* patterns to the same unique output q_i for $i \in [k]$ (i.e., the cluster name) and *non-similar* patterns to distinct names. Formally:

▶ **Definition 1** (Clustering Input Condition). *An infinite input sequence $\overline{Z}_1, \overline{Z}_2, \ldots$ is a well-behaved clustering input sequence with input size n, output size k, memory duration t_m, relative distance parameter Δ, closeness parameter α, and input persistence time t_p if:*

- *For any set of t_m rounds $T = \{t, t+1, \ldots, t+(t_m-1)\}$ there exist $\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_k \in \{0,1\}^n$ such that $\mathcal{RD}(\overline{X}_i, \overline{X}_j) \geq \Delta$ for all $i \neq j$ and for all $i \in T$, $\mathcal{RD}(\overline{Z}_i, \overline{X}_j) \leq \Delta/\alpha$ for some $j \in [k]$.*
- *If $\overline{Z}_i \neq \overline{Z}_{i-1}$, then $\overline{Z}_i = \overline{Z}_{i+1} = \ldots = \overline{Z}_{i+t_p}$.*

▶ **Definition 2** (Clustering Network). *A network $\mathcal{N}$ solves the clustering problem for input size n, output size k, memory duration t_m, relative distance parameter Δ, closeness parameter α, input duration t_p, and failure probability δ if, on a well-behaved input sequence for the same parameters (Definition 1), on any fixed window of t_m rounds, with probability $\geq 1-\delta$:*

- *Each input pattern $\overline{Z}$ is* mapped *to some output q_j for $j \in [k]$. That is, whenever the input changes to $\overline{Z}$ round i (so $\overline{Z}_{i-1} \neq \overline{Z}$ but $\overline{Z}_i = \overline{Z}$), there is a unique output neuron q_j that fires at round $i + t_p$ and continues to fire as long as the input remains fixed to $\overline{Z}$.*
- *Any pair of far patterns $\overline{Z}, \overline{Z}'$ with $\mathcal{RD}(\overline{Z}, \overline{Z}') \geq \Delta$ introduced within the t_m time window are mapped to different outputs.*
- *Any pair of closed-patterns $\overline{Z}, \overline{Z}'$ with $\mathcal{RD}(\overline{Z}, \overline{Z}') \leq \Delta/\alpha$ introduced within the same t_m time window will be mapped to the same output neuron.*

Our goal is to design a clustering network that uses small number of auxiliary neurons and requires small input persistence time t_p. We show the following theorem.

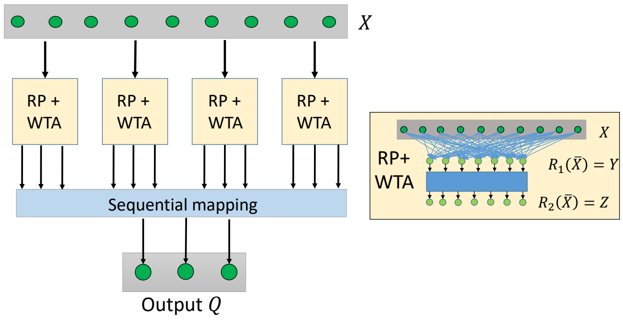

Figure 1 High level illustration of the clustering network. Right: The input pattern $\bar{X} \in \{0,1\}^n$ is mapped to an intermediate sparser vector in two steps: random projection and WTA sparsification. Left: In the clustering network, the input $\bar{X}$ is mapped by applying $O(\log(k/\delta))$ parallel repetitions of the random projection + WTA mapping. As a result, $\bar{X}$ is mapped to a vector $\bar{Z}$ with $O(\frac{\log(k/\delta)}{\Delta})$ neurons. This vector is mapped to the output unit vector in $\{0,1\}^k$ via a sequential mapping module.

▶ **Theorem 3.** *For any parameters n, k, t_m, δ and Δ, there is a network $\mathcal{N}$ with $O\left(\frac{\log(1/\Delta)^3 \log(t_m/\delta)\log(1/\delta)}{\Delta^{3/2}}\right)$ auxiliary neurons that solves the clustering problem with these parameters, input persistence time $t_p = O\left(\frac{\log(1/\Delta)^2 \log(t_m/\delta)}{\Delta}\right)$ and closeness parameter $\alpha = O(\log(1/\Delta)^4)$.*

Note that the number of auxiliary neurons and the convergence time of Theorem 3 are *independent of the input dimension n*, which may potentially be very large. The spiking neural network construction that achieves Theorem 3 involves in three steps. The first two steps reduce the input from n neurons to $m \ll n$ neurons, while approximately preserving the relative distances between inputs. These steps use a biologically inspired construction that mirrors circuits seen, for example, in the fruit fly olfactory system [9, 29, 17]. In particular the first step maps the input to a set of intermediate neurons via random projection, and the second step sparsifies the outputs of these intermediate neurons to yield a sparse code representing the input. The final *sequential mapping* step then solves the clustering problem given these m intermediate neurons as inputs, avoiding the high cost of directly solving the problem on the n-dimensional input. See Figure 1 for an illustration.

1.2 Comparison to Previous Work

1.2.1 Broader Agenda: Algorithmic Theory for Brain Networks

Understanding how the brain works, as a computational device, is a central challenge of modern neuroscience and artificial intelligence. Different research communities tackle this problem in different ways, ranging from studies that examine neural network structure as a clue to computational function [43, 3], to functional imaging that studies neural activation patterns [40, 31], to theoretical work on simplified models of neural computation [23, 36], to the engineering of complex neural-inspired machine learning architectures [21, 27]. This paper joins a recent line of work [44, 37, 45, 38, 33, 30, 46, 34, 28, 32, 39, 10, 22] that approach this problem using techniques from *distributed computing theory and other branches of theoretical computer science*. The ultimate goal of this research direction is to develop an *algorithmic theory for brain networks*, based on stochastic graph-based neural network models. To understand neural behavior from a theory of computing point of view, we design networks to solve abstract problems that are inspired by tasks that seem to be solved in actual brains. We believe that the rigorous analysis of such networks in terms of static costs (e.g., the number of neurons), and dynamic costs (e.g., the time to converge to a solution) will lead to a better understanding of how these tasks may be performed in biological neural networks.

1.2.2 Connections to Sparse Recovery

Our work is closely related to sparse recovery (compressed sensing), where the goal is to map high-dimensional but sparse vectors (with dimension n and $s \ll n$ nonzero entries) into a much lower dimensional space, such that the vectors can be uniquely identified and efficiently recovered [19]. We can see that this goal is essentially identical to that of our first two network layers, before the sequential mapping step. Two different s-sparse binary vectors have relative hamming distance $\geq 1/s$. Additionally there are $k = O(n^s)$ s-sparse binary vectors in n dimensions. Thus, as a Corollary of Lemma 12, our first two layers can uniquely compress all such vectors with high probability into dimension $m = \tilde{O}\left(\frac{\log k}{\Delta^{1/2}}\right) = O(s^{3/2} \log n)$.

It is known that optimal sparse recovery reducing the dimension to $O(s \log n)$ can be achieved using random projections [13]. However, unlike in our setting, these random projections have real valued outputs, which cannot be directly represented by binary spiking neurons. The case when output of the random projection is thresholded to be a binary value has been studied extensively, under the name "one-bit compressed sensing" [6]. In this setting, it is known that dimension $\tilde{\Theta}\left(s^2 \log n\right)$ can be achieved and is required for general sparse recovery [1]. If the input vectors are restricted to be binary (as in our case), dimension $O(s^{3/2} \log n)$ is possible [24, 1]. Our results match this bound up to logarithmic factors.

1.2.3 Connections to Fruit Fly Novelty Detection via Bloom Filters

Recently, Dasgupta et al. [16] demonstrated that the fruit fly olfactory circuit implements a variant of a classic Bloom filter [5] to assess the novelty of odors. A bloom filter is a data structure that maintains a set of items, allowing for membership queries, with the possibility of occasional false positives. The filter has m bits and r random hash functions mapping the input space to integers in $1, ..., m$. When an item is inserted, it is hashed using these r functions and the bits corresponding to the hashed values are set to 1. A membership query is answered by hashing the input in question and checking that all r bits corresponding to its hashed values are set to 1.

Such a filter can be used to implement novelty detection – a novel pattern is detected whenever an insertion operation sets a new bit to 1 or an membership query returns false. Dasgupta et al. [16] demonstrate that a such a scheme is used in the fly olfactory circuit. The hashing step consists of a random projection followed by winner-take-all sparsification, which maps each input into a r-sparse binary vector. The r entries of this vector represent the r hash function outputs. This step closely resembles the first two layers of our clustering network.

Our third layer operates differently than a bloom filter, associating each sparsified intermediate vector to with unique output via sequential mapping rather than simply marking the bits corresponding to its entries. However, it can implement the same functionality (and correspondingly can implement novelty detection). Specifically, to implement insertion and deletion operations we can make the following modifications to the sequential mapping sub-network:

- The input layer contains an extra neuron that is set to 1 if the operation is insertion, and 0 if the operation is a membership query.
- In the sequential mapping step the output layer fires only if this extra neuron fires. In this way, new outputs will only be mapped during insertion operations and not membership queries.
- For query operations, we add an output neuron that fires only if there exists an index $j \in [k]$ for which many memory modules m_{ji}, as well as an association neuron a_{ji} fire.
- Novelty detection can be implemented via an additional output neuron that responds when an insertion causes a new output to be mapped or when a query operation returns false.

2 Computational Model and Preliminaries

We start by defining our model of stochastic spiking neural networks.

Network Structure. A *Spiking Neural Network* (SNN) $\mathcal{N} = \langle X, Q, A, w, b \rangle$ consists of n input neurons $X = \{x_1, \ldots, x_n\}$, m output neurons $Q = \{q_1, \ldots, q_m\}$, and ℓ auxiliary neurons $A = \{a_1, ..., a_\ell\}$. The directed, weighted synaptic connections between X, Q, and A are described by the weight function $w : [X \cup Q \cup A] \times [X \cup Q \cup A] \to \mathbb{R}$. A weight $w(u, v) = 0$ indicates that a connection is not present between neurons u and v. Finally, for any neuron v, $\beta(v) \in \mathbb{R}_{\geq 0}$ is the activation bias – as we will see, roughly, v's membrane potential must reach $\beta(v)$ for a spike to occur with good probability.

The in-degree of every input neuron x_i is zero. That is, $w(u, x) = 0$ for all $u \in [X \cup Q \cup A]$ and $x \in X$. Additionally, each neuron is either *inhibitory* or *excitatory*: if v is inhibitory, then $w(v, u) \leq 0$ for every u, and if v is excitatory, then $w(v, u) \geq 0$ for every u.

Neuron Chains. In our implementation, we make use of *chains* of neurons to create a delay in a response. For a neuron u, and integer ℓ, let $C_\ell(u)$ be a directed path of length ℓ starting with u. All neurons on the chain are excitatory. We then say that a chain $C_\ell(u)$ is connected to v if each neuron $w \in C_\ell(u)$ has an outgoing edge to v.

The SNN Model. An SNN evolves in discrete, synchronous rounds as a Markov chain. The firing probability of every neuron at time t depends on the firing status of its neighbors at time $t - 1$, via a standard sigmoid function, with details given below. For each neuron u, and each time $t \geq 0$, let $u^t = 1$ if u fires (i.e., generates a spike) at time t. Let u^0 denote the initial firing state of the neuron. Our results will specify the initial input firing states $x_j^0 = 1$ and assume that $u^0 = 0$ for all $u \in [Q \cup A]$. The firing state of each input neuron x_j in each round is the input to the network, and our results will specify to which sequences of input firing patterns they apply.

For each non-input neuron u and every $t \geq 1$, let $\text{pot}(u, t)$ denote the membrane potential at round t and $p(u, t)$ denote the corresponding firing probability ($\Pr[u^t = 1]$). These values are calculated as:

$$\text{pot}(u,t) = \sum_{v \in X \cup Q \cup A} w_{v,u} \cdot v^{t-1} - \beta(u) \text{ and } p(u,t) = \frac{1}{1 + e^{-\text{pot}(u,t)/\lambda}} \tag{1}$$

where $\lambda > 0$ is a *temperature parameter*, which determines the steepness of the sigmoid. It is easy to see that λ does not affect the computational power of the network. A network can be made to work with any λ simply by scaling the synapse weights and biases appropriately.

For simplicity we assume throughout that $\lambda = \frac{1}{\Omega(\log(n \cdot k \cdot \Delta \cdot t_m \cdot 1/\delta))}$, where $n, k, \delta, \Delta, t_m$ are the parameters of the clustering problem, defined in Section 1.1. Thus by (1), if $\text{pot}(u,t) \geq 1$, then $u^t = 1$ w.h.p. and if $\text{pot}(u,t) \leq -1$, $u^t = 0$ w.h.p. , where w.h.p. denotes with probability at least $1 - (1/\delta \cdot n \cdot k \cdot \Delta \cdot t_m)^{-c}$ for some constant c.

The remainder of the paper is devoted to proving Theorem 3. Our analysis considers the three stages of the network in sequence: random projection, sparsification, and sequential mapping to the final outputs.

3 Layer 1: Random Projection

The goal of this step is to reduce the input size from n input neurons to $m \ll n$ neurons while ensuring that the relative distance between any two n-length input vectors is approximately preserved after the mapping. In this way, we can solve the clustering problem working with the much smaller m neuron representation instead of the original n neuron input. While there are many ways in which distance may be preserved, we consider one in particular, based on the membrane potentials induced on the intermediate neurons by the inputs:

▶ **Definition 4** (Distance Preserving Dimensionality Reduction)**.** *Consider $\overline{X}_1, \ldots, \overline{X}_k \in \{0,1\}^n$ with $\mathcal{RD}(\overline{X}_i, \overline{X}_j) \geq \Delta$ for $i \neq j$. Consider a network $\mathcal{N}$ mapping n input neurons to m intermediate neurons, which are split into b buckets each containing m/b neurons. $\mathcal{N}$ is distance preserving for $\overline{X}_1, \ldots, \overline{X}_k$ if, for any two $\overline{X}_i, \overline{X}_j$, and any round t, in the large majority of buckets, the identity of the neuron that in round $t+1$* has the highest membrane potential below a fixed threshold τ *is different if $\overline{X}_i$ is presented at round t than if $\overline{X}_j$ were presented.*[1]

Our network satisfies Definition 4 with parameters $m = \tilde{O}\left(\frac{1}{\sqrt{\Delta}}\right)$ and $b = \tilde{O}(1)$. We implement the dimensionality reduction step via *random projection*. We note that random projection has been studied extensively as a dimensionality reduction tool in computer science, with applications in data analysis [4, 7, 12], fast linear algebraic computation [42, 11], and sparse recovery [8]. See [47] for a survey. In neuroscience, it is thought that random projection may play a key role in neural dimensionality reduction [20, 2]. Random projection for example, underlies sparse coding of inputs in the fruit fly olfactory circuit [9, 17]. Random connections have also been studied in theoretical models for memory formation, in which inputs are mapped to representative output neurons [45, 38, 28].

We start with describing the construction and then analyzing its properties. The main outputs of this section are Corollaries 9 and 11 which show that, with high probability, the identities of the neurons with maximum membrane potential below some threshold τ in each bucket of the intermediate layer share little overlap for far inputs (with relative distance $\geq \Delta$) and significant overlap for close inputs (with relative distance $\leq \Delta/\alpha$ for $\alpha = O(\log(1/\Delta)^4)$). That is, the network satisfies the distance preserving dimensionality reduction guarantee of Definition 4 for far inputs, along with an analogous guarantee for close inputs.

Our mapping can be understood as an example of *local sensitive hashing* [18, 15, 17]. In each bucket, the input is hashed to the identity of the maximum potential neuron below τ in that bucket. Near inputs have many hash collisions, and thus there is significant overlap in the identities of the mapped neurons. Far input have fewer collisions and thus less overlap.

[1] We formally define how the membrane potential is calculated in Section 2. "Large majority" will be a constant fraction of the buckets significantly larger than 1/2, which will be specified in our bounds.

Layer Description. The random projection layer consists of $m \cdot \ell$ intermediate auxiliary neurons for $m = \Theta\left(\frac{\log(1/\Delta)}{\sqrt{\Delta}}\right)$ and $\ell = \Theta(\log(t_m/\delta))$. The layer is subdivided into ℓ buckets $b_1, ..., b_\ell$ containing m neurons each. Each input neuron has an excitatory connection to each neuron in the intermediate layer with weight sampled as a Chi-squared random variable (with one degree of freedom). We denote this random weight matrix connecting the two layers by $\mathrm{A} \in \mathbb{R}^{m \cdot \ell \times n}$. For $b \in 1, ..., \ell$, we let $\mathrm{A}_b \in \mathbb{R}^{m \times n}$ denote the rows of A corresponding to the intermediate neurons in bucket b. In typical applications of random projection, the entries of A are most commonly either Gaussian or Rademacher random variable. Here we use Chi-squared random variables as they are non-negative, a requirement in our setting where the outgoing edge weights from each neuron (corresponding to the entries in A) must be either all positive or all negative.

Layer Analysis. When the input neurons X fire with input pattern $\overline{X}_i \in \{0, 1\}^n$ at time t, by (1) the vector of membrane potentials of the intermediate neurons at time $t + 1$ is given by $\mathrm{A}\,\overline{X}_i \in \mathbb{R}^{m \times \ell}$. Our analysis will focus on the properties of this vector of potentials, which can be viewed as a real valued compressed representation of the input $\overline{X}_i$. Later, we will show how these properties lead to desirable properties of the spiking patterns of the intermediate neurons.

For technical reasons, we will not focus on the actual largest entry of $A_b\overline{X}_i$, but on the *largest entry bounded by some fixed threshold* τ which can still be identified via a minor modification to a traditional WTA circuit. We begin with a preliminary lemma showing that a Chi-squared distribution (the distribution of each entry in $A_b\overline{X}_i$) is roughly uniform around its mean. We give a proof in Appendix A.

▶ **Lemma 5** (Chi-squared uniformity). *Let $\mathcal{D}_p$ be the Chi-squared distribution with p degrees of freedom. For any c with $1 \leq c < p^{1/2}$ there are constants c_ℓ, c_u (depending on c) such that, for any interval $[r_1, r_2] \subseteq \left[p - cp^{1/2}, p + cp^{1/2}\right]$, we have: $\frac{c_\ell(r_2 - r_1)}{p^{1/2}} \leq \Pr_{x \sim \mathcal{D}_p}\left[x \in [r_1, r_2]\right] \leq \frac{c_u(r_2 - r_1)}{p^{1/2}}$. That is, $\mathcal{D}_p$ is roughly uniform on the range $\left[p - cp^{1/2}, p + cp^{1/2}\right]$.*

We also use the fact that the Chi-squared distribution decays far from its mean, which follows from standard sub-exponential concentration bounds.

▶ **Lemma 6** (Chi-squared decay). *Let $\mathcal{D}_p$ be the Chi-squared distribution with p degrees of freedom. For any $c \leq 1$ there is a constant c_1 (depending on c) such that:*

$$\Pr_{x \sim \mathcal{D}_p}\left[x \notin [p - c_1 p^{1/2}, p + c_1 p^{1/2}]\right] \leq c.$$

Using the near-uniform distribution property of Lemma 5, we can show that with good probability, for every compressed vector $\mathrm{A}_b\,\overline{X}_i \in \mathbb{R}^m$ the gap between the two largest entries (bounded by the threshold) is $\Omega(p^{1/2}/m)$ – since there are m entries roughly uniformly distributed in a range of size $O(p^{1/2})$. This gap will be necessary for the neuron with the largest membrane potential (and hence the highest firing probability) to be reliably identified in the second sparsification layer of our network. We remark that in non-neural applications of random projection such a gap would not be necessary: the largest entry in the bucket can be typically be identified exactly.

The complete proof is given in Appendix A.

▶ **Lemma 7** (Sufficient Gap). *Consider our construction with bucket size $m = c_1$. Let $\overline{X} \in \{0,1\}^n$ be any input vector with $\|\overline{X}\| = p$ for $p \geq 5$. Let $\tau = p + 2p^{1/2}$ and for any $b \in [\ell]$ define:*

$$i_{1,b}(\overline{X}) = \operatorname*{arg\,max}_{i \in [m]: [A_b \overline{X}](i) \leq \tau} [A_b \overline{X}](i) \quad \textit{and} \quad i_{2,b}(\overline{X}) = \operatorname*{arg\,max}_{i \in [m] \setminus i_{1,b}(\overline{X}): [A_b \overline{X}](i) \leq \tau} [A_b \overline{X}](i),$$

where we set $i_{1,b}(\overline{X}), i_{2,b}(\overline{X}) = 0$ in the case that no index satisfies the constraint. For sufficiently large constants c_1, c_2, with probability $\geq 99/100$ over the random choice of A_b, $i_{1,b}(\overline{X}) \neq 0$, $[\mathrm{A}_b\,\overline{X}](i_{1,b}(\overline{X})) \geq p$, and either $[\mathrm{A}_b\,\overline{X}](i_{1,b}(\overline{X})) - [\mathrm{A}_b\,\overline{X}](i_{2,b}(\overline{X})) \geq \frac{p^{1/2}}{c_2 m}$ or $i_{2,b}(\overline{X}) = 0$.

Along with Lemma 7 we prove that, with good probability, the neuron with the maximum potential below τ in each bucket differs for any two far inputs.

▶ **Lemma 8** (Low Collision Probability – Far Inputs). *Let $\overline{X}_1, \overline{X}_2 \in \{0,1\}^n$ be two vectors with $\|\overline{X}_1\| = \|\overline{X}_2\| = p$* [2] *and $\mathcal{RD}(\overline{X}_1, \overline{X}_2) \geq \Delta$. Assume that $p \geq c$ for some sufficiently large constant c. Consider our construction with bucket size $m = \frac{c_1 \log(1/\Delta)}{\sqrt{\Delta}}$. Then for sufficiently large constants c_1, c_2, for any $b \in [\ell]$, defining $i_{1,b}(\cdot)$ and $i_{2,b}(\cdot)$ as in Lemma 7, with probability ≥ 0.9165:*

- *$i_{1,b}(\overline{X}_1) \neq i_{1,b}(\overline{X}_2)$.*
- *For both $j = 1, 2$: $i_{1,b}(\overline{X}_j) \neq 0$, $[A_b \overline{X}_j](i_{1,b}(\overline{X}_j)) \geq p$, and $[A_b \overline{X}_j](i_{1,b}(\overline{X}_j)) - [A_b \overline{X}_j](i_{2,b}(\overline{X}_j)) \geq \frac{p^{1/2}}{c_2 \cdot m}$ or $i_{2,b}(\overline{X}_j) = 0$.*

See Appendix A for the complete proof of Lemma 8. Intuitively, if $\overline{X}_1$ and $\overline{X}_2$ each have Hamming weight p and relative distance Δ they differ on $\Omega(\Delta p)$ entries. If *just the shared entries* of these vectors fired as inputs, by Lemma 5 each intermediate neuron in the bucket of size m would have its potential distributed roughly uniformly in a range of width $O([(1-\Delta)p]^{1/2}) = O(p^{1/2})$. On average these potentials would be spaced out by $O(p^{1/2}/m)$. By setting $m = \tilde{O}(1/\sqrt{\Delta})$ we have average spacing $\tilde{O}(\Delta^{1/2} p^{1/2})$. This is a small enough spacing, so that when we consider the $\Omega(\Delta p)$ non-shared neurons in the inputs, their contribution to the potential will be large enough to significantly reorder the potentials of the intermediate neurons, so that the neuron with maximum potential is unlikely to be the same for the two different inputs.

From Lemma 8 we can show that our network satisfies the distance preserving dimensionality reduction guarantee of Definition 4, along with the additional condition that the gap between the membrane potentials of the neurons with the largest potentials under $\tau = p + 2p^{1/2}$ is sufficiently large, so that these neurons can be distinguished reliably in the second sparsification layer:

▶ **Corollary 9** (Overall Success – Far Inputs). *For $m = O\left(\frac{\log(1/\Delta)}{\sqrt{\Delta}}\right)$, and $\ell = O(\log(t_m/\delta))$, for any window of t_m rounds, with probability $\geq 1 - \delta$, for all pairs of inputs $\overline{X}_1, \overline{X}_2$ presented during these rounds with $\mathcal{RD}(\overline{X}_1, \overline{X}_2) \geq \Delta$, on at least $91/100 \cdot \ell$ of the ℓ buckets, letting $\tau = p + 2p^{1/2}$ and defining $i_{1,b}(\cdot)$ and $i_{2,b}(\cdot)$ as in Lemma 7:*

- *$i_{1,b}(\overline{X}_1) \neq i_{1,b}(\overline{X}_2)$*
- *For both $j = 1, 2$: $i_{1,b}(\overline{X}_j) \neq 0$, $[A_b \overline{X}_j](i_{1,b}(\overline{X}_j)) \geq p$, and $[A_b \overline{X}_j](i_{1,b}(\overline{X}_j)) - [A_b \overline{X}_j](i_{2,b}(\overline{X}_j)) = \Omega\left(\frac{p^{1/2}}{m}\right)$ or $i_{2,b}(\overline{X}_j) = 0$.*

[2] When $\overline{X}$ is binary we often drop the subscript and just use $\|\overline{X}\|$ to denote the ℓ_1 norm which is equal to the number of nonzero entries, $|\mathrm{supp}(\overline{X})|$.

Proof. By Lemma 8 and a Chernoff bound, since $\ell = \Theta(\log(t_m/\delta)) = \Theta(\log(t_m^2/\delta))$, for any fixed pair of inputs with $\mathcal{RD}(\overline{X}_1, \overline{X}_2) \geq \Delta$, the conditions hold on at least $91/100 \cdot \ell$ buckets with probability $\geq 1 - \delta/t_m^2$. The corollary follows since at most t_m inputs can be presented in t_m rounds, and so we can union bound over at most t_m^2 pairs of far inputs. ◀

We can give a complementary statement to Lemma 8: if $\overline{X}_1$ and $\overline{X}_2$ are close to each other, it is relatively likely that the index of the largest value of $\mathrm{A}_b \overline{X}_1$ and $\mathrm{A}_b \overline{X}_2$ are the same. We defer the proof to Appendix A.

▶ **Lemma 10** (High Collision Probability – Close Inputs). *Let $\overline{X}_1, \overline{X}_2 \in \{0,1\}^n$ be two vectors with $\mathcal{RD}(\overline{X}_1, \overline{X}_2) \leq \Delta/\alpha$. Consider our construction with bucket size $m = \frac{c_1 \log(1/\Delta)}{\sqrt{\Delta}}$. Then for sufficiently large constants c_1, c_2 and $\alpha = O(\log(1/\Delta)^4)$, for any $b \in [\ell]$, defining $i_{1,b}(\cdot)$ and $i_{2,b}(\cdot)$ as in Lemma 7, with probability ≥ 0.97:*

- *$i_{1,b}(\overline{X}_1) = i_{1,b}(\overline{X}_2)$.*
- *For both $j = 1, 2$: $i_{1,b}(\overline{X}_j) \neq 0$, $[A_b \overline{X}_j](i_{1,b}(\overline{X}_j)) \geq p$, and $[A_b \overline{X}_j](i_{1,b}(\overline{X}_j)) - [A_b \overline{X}_j](i_{2,b}(\overline{X}_j)) \geq \frac{p^{1/2}}{c_2 \cdot m}$ or $i_{2,b}(\overline{X}_j) = 0$.*

Lemma 10 yields an analogous corollary to Corollary 9, which follows via a Chernoff bound and a union bound over at most t_m^2 pairs of close inputs that may be presented over t_m rounds.

▶ **Corollary 11** (Overall Success – Close Inputs). *For $m = O\left(\frac{\log(1/\Delta)}{\sqrt{\Delta}}\right)$, $\ell = O(\log(t_m/\delta))$, and $\alpha = O(\log(1/\Delta)^4)$, for any window of t_m rounds, with probability $\geq 1 - \delta$, for all pairs of inputs $\overline{X}_1, \overline{X}_2$ presented during these rounds with $\mathcal{RD}(\overline{X}_1, \overline{X}_2) \leq \Delta/\alpha$, on at least $96/100 \cdot \ell$ of the ℓ buckets, letting $\tau = p + 2p^{1/2}$ and defining $i_{1,b}(\cdot)$ and $i_{2,b}(\cdot)$ as in Lemma 7:*

- *$i_{1,b}(\overline{X}_1) = i_{1,b}(\overline{X}_2)$*
- *For both $j = 1, 2$: $i_{1,b}(\overline{X}_j) \neq 0$, $[A_b \overline{X}_j](i_{1,b}(\overline{X}_j)) \geq p$, and $[A_b \overline{X}_j](i_{1,b}(\overline{X}_j)) - [A_b \overline{X}_j](i_{2,b}(\overline{X}_j)) = \Omega\left(\frac{p^{1/2}}{m}\right)$ or $i_{2,b}(\overline{X}_j) = 0$.*

4 Layer 2: Sparsification via Winner Takes All

Corollaries 9 and 11 show that the random projection step preserves significant information about input distance, encoded in the membrane potentials of the intermediate neurons, which correspond to the entries of $\mathrm{A}\,\overline{X}$ when the network is given input $\overline{X}$. These membrane potentials cause the intermediate neurons to fire randomly, as Bernoulli processes with different rates. The goal of our second layer is to convert this random behavior to a uniquely identifying sparse code for each input. We achieve this through a winner-takes-all (WTA) based sparsification process, which is thought to play a major role in neural computation [26, 14, 37]. A separate winner-take-all instance is applied to each bucket, “selecting” the neuron with the highest membrane potential below τ by inducing its corresponding neuron in the sparsification layer to fire with high probability while all other neurons in the bucket do not fire.Let $\overline{Y} \in \mathbb{R}^m$ denote the vector of membrane potentials of a single bucket of the intermediate layer: $\overline{Y} = \mathrm{A}_b\,\overline{X}$. Our WTA layer maps each $\overline{Y}$ into a binary unit-vector $\overline{Z}$ of the same length, in which the only firing neuron corresponds to the neuron with the largest potential in $\overline{Y}$ that is bounded by the threshold parameter τ. As explained in Section 3, the random projection step produces $\ell = O(\log(t_m/\delta))$ random compressed vectors, one for each of the ℓ buckets. Each such copy is an input to an independent WTA circuit and thus, in this section, we focus on our construction restricted to just a single bucket, bearing in mind that in fact our network consists of ℓ repetitions of this module.

The first part of the WTA circuit is devoted to *reading*: the circuit collects firing statistics for a period of $T = \widetilde{O}(m^2)$ rounds to obtain a good estimate of the neuron in the bucket that 1) has potential $\leq \tau$ and 2) has the largest firing rate. This neuron corresponds to the neuron with the highest potential in $\overline{Y}$ bounded by τ. This is done by augmenting each neuron i in the bucket with a directed chain H_i of neurons of length T. The j^{th} neuron in the chain triggers the firing of the $(j+1)^{th}$ neuron with high probability. As a result, after T rounds, the number of firing neurons in the chain H_i is equal to the number of times i fired within the last T rounds, with high probability. We thus refer to this H_i chain as the *history* chain of the i^{th} neuron in the bucket. The second part of the circuit first excludes all neurons with potential $\geq \tau$ and then applies a standard WTA circuit to pick the neuron remaining that fires the most in this T-length time interval. See Fig. 2 for an illustration of the overall clustering network and the WTA module. The main result of this section is as follows.

▶ **Lemma 12.** *For every pair of input patterns $\overline{X}_i, \overline{X}_j$ presented over a period of t_m rounds, with probability at least $1 - \delta$ the following hold:*

(I) *If $\mathcal{RD}(\overline{X}_i, \overline{X}_j) \geq \Delta$, then $\text{supp}(\overline{Z}_i) \setminus \text{supp}(\overline{Z}_j) \geq 0.9 \cdot \ell$.*

(II) *If $\mathcal{RD}(\overline{X}_i, \overline{X}_j) \leq \Delta/\alpha$, then $\text{supp}(\overline{Z}_i) \cap \text{supp}(\overline{Z}_j) \geq 0.9 \cdot \ell$.*

We first give a detailed description of the specification step via WTA (see Figure 2). We focus on a single bucket, bearing in mind that in fact our network consists of ℓ repetitions of this module.

Reading via History Chain. Every neuron $i \in \{1, \ldots, m\}$ in the bucket is connected to a chain H_i of length $T = \Theta(\log(1/\delta) \cdot m^2)$ of neurons where the j^{th} neuron in this chain fires in round t with high probability iff its incoming neighbor on that chain fires in round $t-1$. This is done by setting the bias value of each neuron to 1 and the edge weights to be $1/2$. As a result we get that the number of firing neurons in this chain equals to the number of times i fires within the last T rounds with high probability.

Omitting the Neurons Exceeding the Threshold Value. For every neuron $i \in \{1, \ldots, m\}$ we introduce an inhibitor copy r_i that has the same incoming weights as i and therefore also has the same potential. We set the bias of r_i such that with high probability it fires iff its potential exceeds the threshold value τ. We then connect r_i to all neurons in the chain H_i with large negative weight. As a result, if the potential of neuron i exceeds τ with high probability all neurons in H_i will not fire.

Selecting the Maximum Firing Rate with Pairwise Comparisons. For every ordered pair of neurons $i, j \in [1, m]$, we have a designated (threshold gate) neuron $y_{i,j}$ that fires iff the i^{th} neuron in the bucket fires more than the j^{th} neuron within the last T rounds. To accomplish this, each of the neurons in the chain H_i (respectively, H_j) is connected to $y_{i,j}$ with a positive (respectively, negative) edge weight of ± 1. Hence, the total weighted sum incoming to $y_{i,j}$ is exactly the difference between $R(i)$ and $R(j)$ where $\overline{R}(i), \overline{R}(j)$ are the number of times that the i^{th} and j^{th} neurons fired in the last T rounds. We set the bias of $y_{i,j}$ such that it fires with high probability iff $\overline{R}(i) - \overline{R}(j) \geq 1$. The i^{th} output neuron in $\overline{Z}$ computes the AND-gate of the threshold-gates $y_{i,1}, \ldots, y_{i,m}$. That is, z_i fires in round t only if every $y_{i,1}, \ldots, y_{i,m}$ fired in round $t-1$. The AND-gate can be implemented by setting the incoming edge weight from each $y_{i,j}$ to z_i to be $1/m$, and the bias of $\overline{Z}_i$ to $1 - 1/(2m)$.

Analysis. The requirement from the WTA module is that the firing frequency vector $\overline{R}$ has its largest entry in the same position as the largest entry of $\overline{Y}$ that is $\leq \tau$. If this is the case, the WTA circuit indeed selects the neuron corresponding to the largest firing rate $\leq \tau$, and the only entry in the support of $\overline{Z}$ is the one corresponding to this entry. For the largest entry in $\overline{R}$ to reflect the largest entry in $\overline{Y} \leq \tau$ with probability $\geq 1-\delta$, the gap between the largest and second largest firing rates must be $\Omega\left(\sqrt{\log(1/\delta)/T}\right)$. Using the gap condition of Corollary 9 we will show that this gap is $\Omega(1/m)$, letting us set $T = O(\log(1/\delta) \cdot m^2)$. The desired gap is achieved in a large fraction of the buckets, this implies that the WTA picks the maximal entry in most of the buckets as well.

▷ Claim 13. Let $\overline{Y}$ be a vector with $i = \arg\max_{j:\overline{Y}(j)\leq\tau} \overline{Y}(j)$ and $\overline{Y}(i) - \overline{Y}(j) = \Omega(p^{1/2}/m)$ for every[3] $j \neq i$ with $\overline{Y}(j) \leq \tau$. Then in the output vector $\overline{Z}$, $\overline{Z}(i) = 1$ and $\overline{Z}(j) = 0$ for every $j \neq i$ with probability at least 99/100. If $\overline{Y}$ is first introduced in round t, the desired output vector $\overline{Z}$ fires in round $t + T + 2$ w.h.p.

The proof of Claim 13 and the complete proof of Lemma 12 is given in Appendix B.

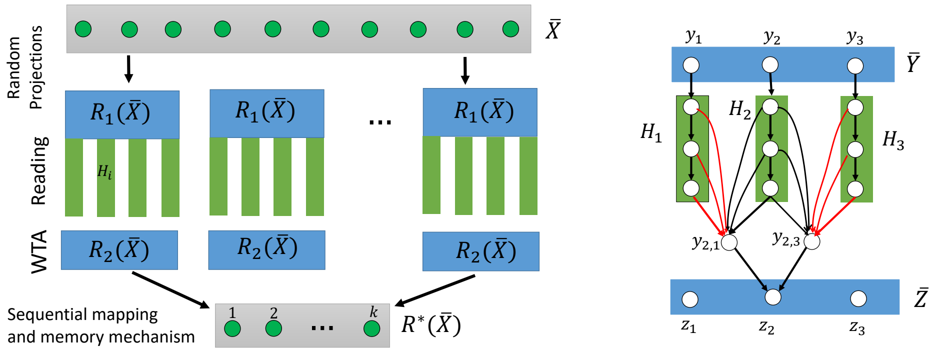

Figure 2 Left: Overall network description, the input pattern $\overline{X}$ is mapped to unique output neuron in $[1,k]$ via three main steps. Right: Description of the WTA circuit. For clarity we only show the connections for the second output neuron, but same holds for all k output neurons. Every input neuron i in $\overline{Y}$ is connected to a history chain H_i of length T that is used to collect firing statistics. For each pair of input neurons i,j, there is a threshold gate $q_{i,j}$ that fires only if i fired at least $T/2m$ more times than j within T rounds. Each history neuron in H_i, H_j is connected with weight 1 (respectively -1) to $q_{i,j}$ and the bias of $q_{i,j}$ is $T/2m$. Finally, each output neuron q_i computes the AND gate of $q_{i,1},\ldots,q_{i,m}$, i.e., fires only if all these gates fire in the previous round. As a result a winner q_i is selected only if $y_i - y_j = \Omega(1/m)$ for every $j \neq i$.

5 Layer 3: Sequential Mapping

We conclude by discussing the final sequential mapping layer of our network, which maps the *binary* patterns $\overline{Z}_i$ of length $r = O(\ell \cdot m)$ to a single output neuron. The inputs to the third layer are the r neurons $Z = \{z_1, z_2, \ldots, z_r\}$ and its outputs are the k output neurons $Q = \{q_1, q_2, \ldots, q_k\}$. The r-length patterns will be mapped to their unique output neuron in a sequential manner, where at each given round, a newly introduced pattern will be mapped to the available output with the smallest index. The mapping will satisfy the following properties: (1) patterns $\overline{Z}_i, \overline{Z}_j$ that correspond to *far* input patterns $\overline{X}_i, \overline{X}_j$ respectively will be mapped to distinct outputs, (2) patterns $\overline{Z}_i, \overline{Z}_j$ that correspond to *close* input patterns

[3] This required gap is based on Lemma 7/Corollary 9.

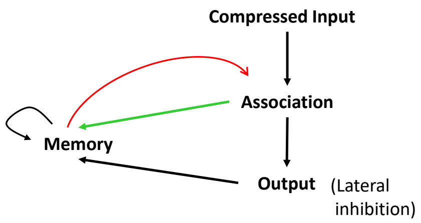

Figure 3 Schematic description.

presented within the same time window of $\Theta(t_m)$ rounds will be mapped to the same outputs. Recall that t_m is the memory duration which is a parameter of the network. A key component in our network is the *memory module* that remembers the association between each previously introduced pattern and its selected output for $\Theta(t_m)$ rounds. Roughly speaking, our network has two intermediate layers: an *association* layer and a *memory* layer (see Figure 3), which we describe below.

We first describe the construction by considering the case where a new pattern $\overline{Z}$ is introduced (and no close pattern to it was introduced before). When $\overline{Z}$ is presented to the network for the first time, it activates the association layer which contains r neurons $a_{i,1}, \ldots, a_{i,r}$ for each output q_i. Let $\mathrm{supp}(\overline{Z})$ be the non-zero entries of $\overline{Z}$. Since[4] $|\mathrm{supp}(\overline{Z})| \leq \ell$ it can activate at most $\ell \cdot k$ many neurons $a_{i,j}$ for every $j \in \mathrm{supp}(\overline{Z})$ and $i \in \{1, \ldots, k\}$. Every output q_i is connected to its association neurons $a_{i,1}, \ldots, a_{i,r}$ and fires only if *many* of them fire.

Our construction will make sure that the number of active *association* neurons of a *taken* output (i.e., output already mapped to other pattern, far from $\overline{Z}$) will be small, which will prevent the firing of these outputs when a far pattern is presented. This will be provided due to the *memory module* appended to each output q_i which remembers the pattern (in fact the cluster of patterns) that were mapped to q_i in the *past*. For each j in the support of the pattern associated with q_i, the memory module corresponding to q_i and z_j inhibits all other association neurons associated with q_i, while activating the association a_{ij}. This association will be remembered – by the memory module – for at least $c_1 \cdot t_m$ rounds and at most $c_2 \cdot t_m$ rounds, for $c_1 < c_2$ with high probability.

For every available output q_i, all its association neurons $a_{i,j}$ for $j \in \mathrm{supp}(\overline{Z})$ will start firing once $\overline{Z}$ is presented, which will in turn activate q_i. To select exactly one output neuron among all the available ones, the output layer is connected via a lateral inhibition, where every neuron q_i inhibits all q_j for $j \geq i + 1$.

Overall, our sequential mapping module satisfies:

▶ **Theorem 14** (The Sequential Mapping Module). *There exists a sequential mapping module with r input neurons, $\widetilde{\Theta}(r \cdot k)$ auxiliary neurons, and k output neurons that for every pattern $\overline{Z}$ that is introduced in round t satisfy the following with probability $1 - \delta$:*

(1) *The pattern $\overline{Z}$ is mapped to one of the outputs $q_1, \ldots, q_k$ in round $t + 6$.*

(2) *Any pair of* close *patterns $\overline{Z}, \overline{Z}'$ introduced within a span of $c_1 \cdot t_m$ rounds are mapped to the same output neuron.*

(3) *Any pair of far patterns $\overline{Z}, \overline{Z}'$ introduced within a span of $c_1 \cdot t_m$ rounds are mapped to different output neurons.*

In addition, if a pattern $\overline{Z}$ (or a pattern close to it) is not introduced for t_m rounds, then its unique mapped output q_j is released after $c \cdot t_m$ rounds, for some constant $c \geq 1$.

[4] As the WTA module picks at most one winning entry in each of the ℓ buckets.

5.1 Complete Network Description of the Sequential Mapping

Next we precisely describe the neurons and edge weights of the sequential mapping subnetwork.

The association layer. For each neuron z_i in the input layer, and each neuron q_j in the output layer, we introduce an *association neuron* denoted as $a_{j,i}$. The neuron $a_{j,i}$ has positive and negative incoming edges from the memory modules that is described in the next paragraph. It also has a *positive* incoming edge from the neuron z_i with weight $w(z_i, a_{j,i}) = 2\ell$, and bias $\beta(a_{j,i}) = (19/10)\ell - 1$. We set the connections to this neuron in a way that guarantees it fires only if z_i fired in the previous round, and no other (far) pattern is already mapped to q_j.

The memory modules. For each neuron z_i in the input layer and output neuron q_j we introduce a *memory of association* module $M_{j,i}$ which remembers the association of neuron z_i and q_j. The memory module $M_{j,i}$ contains $c \cdot \log(\frac{1}{\delta'})$ excitatory neurons denoted as $M^+_{j,i}$ where $\delta' = \delta/\ell$ and c is chosen to be a sufficiently large constant. For every $m \in M^+_{j,i}$ we introduce a *feedback* neuron f_m which starts exciting m once the memory module is being activated.In addition, we introduce a delay chain $C^M_j = C_5(q_j)$ that starts at the output q_j and ends at each of the neurons $m \in M^+_{j,i}$. Finally, the memory module contains two head neurons, an excitatory neuron $m^+_{j,i}$ and an inhibitory neuron $m^-_{j,i}$.

Each excitatory neuron $m \in M^+_{j,i}$ has positive incoming edges from $a_{j,i}, q_j, C^M_j$, as well as from the corresponding feedback neuron f_m with the following weights and bias

$$w(a_{j,i}, m) = 2\lambda, \;\; \forall u \in C^M_j \;\; w(u, m) = 2 \;\;, \; w(f_m, m) = \lambda \cdot (\chi + 2) + 9 \;\;\; \beta(m) = 9 + 2 \cdot \lambda \;,$$

where $\chi = \log(t_m - 1)$. Note that if the feedback neuron f_m fired in the previous round, the memory neuron m fires with probability at least $\frac{1}{1+e^{-\chi}} = 1 - 1/t_m$. The feedback neuron f_m for $m \in M^+_{j,i}$ has *positive* incoming edges from m and $m^+_{j,i}$ with weights $w(m, f_m) = 2$, $w(m^+_{j,i}, f_m) = 2$, and bias $\beta(f_m) = 3$. Hence, w.h.p. f_m fires iff m and $m^+_{j,i}$ fired in the previous round. The excitatory head neuron $m^+_{j,i}$ has *positive* incoming edges from all $m \in M^+_{j,i}$ with weights $w(m, m^+_{j,i}) = 2$ and bias $\beta(m^+_{j,i}) = c \cdot \log(1/\delta') + 1$. The head neuron $m^-_{j,i}$ is an inhibitory copy of $m^+_{j,i}$ with the same incoming edges, bias and potential function.

Each association neuron $a_{j,i}$ has a *positive* incoming edge from the head memory neuron $m^+_{j,i}$ with weight $w(m^+_{j,i}, a_{j,i}) = \ell$. In addition, $a_{j,i}$ has *negative* incoming edges from the inhibitory memory neurons $m^-_{j,i'}$ for every $i' = \{1, 2, \ldots k\} \setminus \{i\}$ with weights $w(m^-_{j,i'}, a_{j,i}) = -1$. Note that w.h.p. $a_{j,i}$ fires in round t only if $z_i = 1$ in round $t-1$. In case where there are at least $1/10\ell$ memory modules $m_{j,i'}$ that inhibit $a_{j,i}$, it fires only if its own memory module, namely, $M_{j,i}$ is active. To prevent a situation of partial memory where only part of the memory modules associated with a pattern are released, if at most 0.9ℓ of the memory modules $M_{j,1}, \ldots, M_{j,r}$ are active, we activate the inhibition of these firing modules. For that purpose, for every output q_j, we introduce 3 *deletion* neurons d^1_j, d^2_j, d^3_j. The neurons d^1_j, d^2_j detect this situation and the inhibitor d^3_j kills the partial memory. The deletion neuron d^1_j has incoming edges from all head neurons $m^+_{j,i}$ for $i = 1 \ldots r$ with weights $w(m^+_{j,i}, d^1_j) = 2$ and bias $\beta(d^1_j) = 1$. Hence, w.h.p. d^1_j fires in round t iff at least one memory module fired in round $t-1$. The second deletion neuron has incoming edges from all the inhibitor head neurons $m^-_{j,i}$ for $i = 1 \ldots r$ with weights $w(m^-_{j,i}, d^2_j) = -1$ and bias $\beta(d^2_j) = -0.9\ell + 1$. Thus, w.h.p. d^2_j fires in round t iff at most 0.9ℓ memory module fired in round $t-1$. The third deletion neuron d^3_j has incoming edges from d^1_j and d^2_j with weights $w(d^1_j, d^3_j) = w(d^2_j, d^3_j) = 2$ and bias $\beta(d^3_j) = 3$. Hence, d^3_j fires in round t iff d^1_j and d^2_j fired in round $t-1$. In addition, the head neurons $m^+_{j,i}, m^-_{j,i}$ have a *negative* incoming edge from d^3_j with weight $w(d^3_j, m^+_{j,i}) = w(d^3_j, m^-_{j,i}) = -2c\log(1/\delta')$.

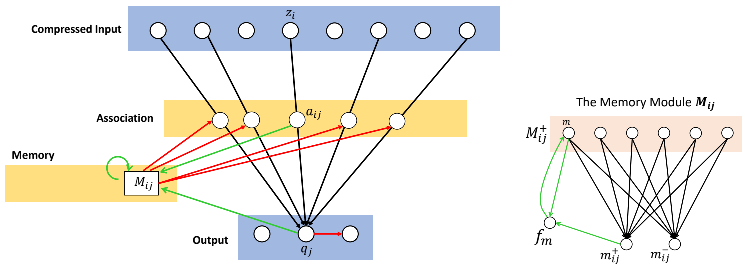

Figure 4 Left: an illustration of the network. The green edges correspond to edges with positive weight where the red edges correspond to negative weights. For simplicity we omitted the history and deletion neurons as well as the rest on the association and memory modules. Right: The memory module and the feedback loop mechanism.

History neurons. If an input pattern $\overline{Z}$ is already mapped to an output neuron, our goal is to map every pattern close to $\overline{Z}$ to the same output. To make sure that close patterns $\overline{Z}, \overline{Z}'$ are indeed mapped to the same output, for each output neuron q_j we introduce an inhibitory *history* neuron h_j. The role of the history neuron is to take care of a situation where a pattern $\overline{Z}$ is mapped to output q_j, but when a close pattern $\overline{Z}'$ is presented later on, an output q_i for $i < j$ is free. Recall that in our construction, each pattern is mapped to the first available output. To do that, the network parameters of the history neurons are defined as follows. Each history neuron h_j has *positive* incoming edges from all associated excitatory memory neurons $m^+_{j,i}$ for $i = 1 \dots r$ with weights $w(m^+_{j,i}, h_j) = 1$. In addition, it has a *positive* incoming edge from the output neuron q_j with weight $w(q_j, h_j) = \ell$ and bias $\beta(h_j) = -(3/2)\ell - 1$. Thus, the history neuron h_j fires if the output neuron q_j fired and at least a large fraction of the memory modules corresponding to q_j are active (the latter indicates that q_j is indeed taken). The history neuron h_j then inhibits all the preceding output neurons $q_1, \dots, q_{j-1}$, preventing the input pattern from being mapped to a different output.

The output layer. The output layer Q consists of excitatory neurons. In order to map the input pattern sequentially, for each $q_j \in Q$ we introduce an inhibitor output neuron q_j^- which inhibits the output neurons $q_{j'}$ for $j' \in \{j+1, \dots, k\}$. The neuron q_j is connected to q_j^- via a delay chain of length 3 denoted as $C_j^I = C_3(q_j)$. The neuron q_j^- has incoming edges from C_j^I with weights 2, and a negative bias of $\beta(q_j^-) = 5$. Hence, w.h.p. q_j^- fires iff q_j fired for 3 consecutive rounds.

Each output neuron q_j has *positive* incoming edges from the association neuron $a_{j,i}$ for every $i = \{1, 2, \dots k\}$. In addition, q_j has *negative* incoming edges from all preceding neurons q_i^- for $i < j$ and all successive history neurons h_i where $i > j$. The weights and bias are given by

$$w(a_{j,i}, q_j) = 2 \ \ \forall i \in [r], \quad w(q_i^-, q_j) = -3\ell \ \ \forall i < j, \quad w(h_i, q_j) = -3\ell \ \ \forall i > j, \quad \beta(q_j) = \ell - 1$$

Note that q_j fires in round t only if at least $(1/2)\ell$ association neurons fired in round $t-1$, and no history or inhibitor output neuron inhibit it.

As in previous sections, we assume that before the first round no neuron fires (i.e $v^0 = 0$ for every neuron v in the network). Figure 4 illustrates the structure of the network and Figure 5 demonstrates the network flow with two inputs.

5.2 Network Dynamics

Before providing the detailed analysis of the network, we give a more detailed description of the network behavior in the two orthogonal cases: mapping close patterns to the same output and mapping far patterns to distinct outputs.

Introduction of a New Pattern $\overline{X}_j$. A pattern $\overline{X}_j$ is introduced in round t where $q_1, \ldots, q_{j-1}$ are already allocated. We will describe how $\overline{X}_j$ is mapped to q_j. First, in Step (1), $\overline{X}_j$ is mapped to a vector $\overline{Y}_j = R_1(\overline{X}_j)$. In Step (2), $\overline{Y}_j$ is mapped to a *binary* vector $\overline{Z}_j$ which is the input to the sequential mapping sub-network. Let t' be the time in which $\overline{Z}_j$ fires. This will cause the firing of the association layer in the following manner. Let $\overline{X}_1, \ldots, \overline{X}_{j-1}$ be the patterns mapped to $q_1, \ldots, q_{j-1}$.

- For every allocated neuron q_i, $i \leq j-1$, and every entry $i_1 \in \text{supp}(Z_j) \setminus \text{supp}(Z_i)$, their association neuron a_{i,i_1} is inhibited by the memory modules M_{i,i_2} for every $i_2 \in \text{supp}(Z_i)$.
- Thus, for every allocated neuron q_i, when introducing Z_j, at most $|\text{supp}(Z_i) \cap \text{supp}(Z_j)| \leq 0.1 \cdot \ell$ association neurons $a_{i,j'}$ are active.
- Since an output q_i fires only if at least $1/2\ell$ association neurons are active, q_i would not fire.
- For every free output q_i for $i \in \{j, \ldots, k\}$, all the association neurons a_{i,i_1} for every $i_1 \in \text{supp}(Z_j)$ are now active. Hence, in the next round, all $q_j, \ldots, q_k$ fire.
- Since we have a lateral inhibition, q_j inhibits[5] all other q_i for $i \in \{j+1, \ldots, k\}$.
- Only at the point where $q_{j+1}, \ldots, q_k$ are inhibited, the memory modules M_{j,i_1} of the winner output q_j start being active, for every $i_1 \in \text{supp}(Z_j)$. This memory module continues firing from that point on for $\Theta(t_m)$ rounds, even when X_j is not introduced.
- Each activated module M_{j,i_1} for every $i_1 \in \text{supp}(Z_j)$ inhibits each of the other association neurons a_{j,i_2} for every $i_2 \neq i_1$. In addition, each M_{j,i_1} excites its own association neuron a_{j,i_1} for $i_1 \in \text{supp}(Z_j)$, thus canceling the inhibition from the other M_{j,i_2} modules. As a result, the only inhibited association neurons are a_{j,i_2} for $i_2 \notin \text{supp}(Z_j)$.

Re-Introduction of a Close-Pattern $\overline{X}_j$. We now consider the situation where $\overline{X}_j$ is introduced in round t, and a close-pattern $\overline{X}_{j'}$ was introduced in the past (e.g., in a window of $\Theta(t_m)$ rounds). We would like to show that $\overline{X}_j$ will be mapped to the exact same output neuron $q_{j'}$ as $\overline{X}_{j'}$.

- For every allocated neuron q_i and every entry $i_1 \in \text{supp}(Z_j) \setminus \text{supp}(Z_i)$, their association neuron a_{i,i_1} is inhibited by the memory modules M_{i,i_2} for every $i_2 \in \text{supp}(Z_i)$.
- Thus, for every allocated neuron q_i for $i \neq j'$, when introducing Z_j, at most $|\text{supp}(Z_i) \cap \text{supp}(Z_j)| \leq 0.1 \cdot \ell$ association neurons a_{i,i_1} are active. As a result, q_{i_1} will not fire.
- In contrast, for the desired output neuron $q_{j'}$, only $|\text{supp}(Z_j) \setminus \text{supp}(Z_{j'})|$ association neurons are inhibited, while the remaining ones, namely, a_{j',i_1} for $i_1 \in \text{supp}(Z_j) \cap \text{supp}(Z_{j'})$ are active. Since $|\text{supp}(Z_j) \cap \text{supp}(Z_{j'})|$ is sufficiently large, $q_{j'}$ will fire.
- Due to lateral inhibition of $q_{j'}$, all other free outputs $q_{i'}$ for $i' \geq j'+1$ will not fire.
- It remains to show that all other *free* outputs q_i for $i \leq j'-1$ will not be active. Recall that these outputs have a lateral inhibition on $q_{j'}$ that starts inhibiting $q_{j'}$ within a small number of rounds since the activation of q_i. It is therefore important to neutralize these outputs before their inhibition on $q_{j'}$ comes into play. Indeed this is the reason for introducing the delay to the lateral inhibition mechanism.

[5] In fact, its inhibitor copy will do this inhibition.

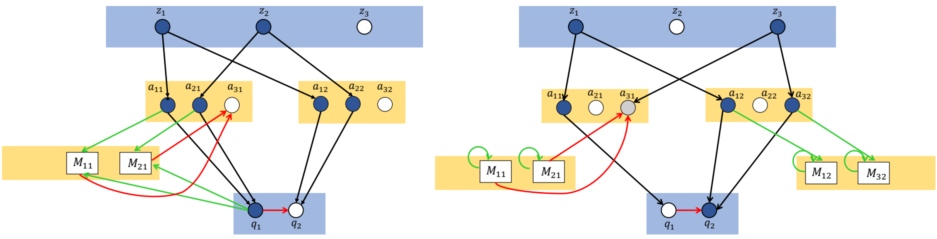

Figure 5 Left: network's state where first pattern $(1,1,0)$ is presented. Since all outputs are free at that point, the pattern is mapped to the first output q_1, which activates all its memory modules. Right: network description when the second input $(1,0,1)$ is presented. Because the memory modules $M_{1,1}$ and $M_{1,2}$ are active, the association neuron $a_{1,3}$ is inhibited and this q_1 will not fire. As a result, $(1,0,1)$ is mapped to q_2, activating corresponding memory modules $M_{2,1}$ and $M_{2,3}$.

- To indicate the fact that $q_{j'}$ was already allocated to a pattern close to X_j, we have a history neuron $h_{j'}$ that works as follows. It gets input from all the memory modules of $q_{j'}$, as well as from $q_{j'}$ itself. Since the close patterns X_j and $X_{j'}$ have many entries in common, sufficiently many memory modules of $q_{j'}$ will activate $h_{j'}$. For a free output q_i for $i \leq j'-1$, the memory modules of q_i are not active and hence the history neuron would not be active.
- The history neuron $h_{j'}$ then inhibits all prior outputs q_i for $i \leq j'-1$ just before their lateral inhibition chain affects $q_{j'}$. In addition, the inhibition on q_i also occurs before the memory modules of q_i start being active. That is, since we want to remember only the association to the correct output $q_{j'}$, we delay the activation of the memory model. The latter starts only after $q_{j'}$ fires for a consecutive constant number of rounds.

5.2.1 Correctness

The following definitions are useful in our context.

▶ **Definition 15.** *A pattern $\overline{Z}$ is* mapped *to an output neuron q_j in round t if when presenting $\overline{Z}$ to the sequential mapping network in round $t-1$, q_j is the only firing output neuron in round t.*

▶ **Definition 16.** *$M_{j,i}$ is* active *in round t, if its head neurons $m^+_{j,i}$, $m^-_{j,i}$ fired in round t.*

In order to prove the main Theorem 14, we start by establishing useful auxiliary claims and observations.

▶ **Observation 17.** *For every output neuron q_j if the number of active memory modules M_{ij} in round t is between 1 and 0.9ℓ, then w.h.p. there are no active memory modules in round $t+3$.*

Proof. For output neuron q_j if the number of active memory modules M_{ij} in round t is at least 1 w.h.p. the deletion neuron d^1_j fires in round $t+1$. If the number of active memory modules M_{ij} is also less than 0.9ℓ then w.h.p. d^2_j fires in round $t+1$ and therefore d^3_j fires in round $t+2$, inhibiting all memory modules M_{ij} for $i = 1,\ldots,r$. ◀

▶ **Observation 18.** *Given that the deletion neurons of output q_j did not fire in round $t-1$, w.h.p. a memory module $M_{j,i}$ is active in round t iff at least $(c/2)\log(1/\delta')$ neurons $m \in M^+_{j,i}$ fired in round $t-1$.*

Proof. Recall that a memory module $M_{j,i}$ is *active* in round t if the excitatory neuron $m^+_{j,i}$ fired. The potential function of $m^+_{j,i}$ is given by

$$pot(m^+_{j,i}, t) = \sum_{m \in M^+_{j,i}} 2 \cdot m^{t-1} - 2c \log(1/\delta')(d^3_j)^{t-1} - c \log(1/\delta') + 1 \ .$$

If at least $\frac{c}{2} \log(1/\delta')$ neurons in $M^+_{j,i}$ fired in round $t-1$, the potential of $m^+_{j,i}$ in round t is at least 1 and the probability that $m^+_{j,i}$ fire in round t is at least $\frac{1}{1+e^{-1/\lambda}} \geq 1 - \Theta(\frac{\delta}{n \cdot k \cdot \Delta \cdot t_m \cdot \log 1/\delta})$. On the other hand, if less than $\frac{c}{2} \log(1/\delta')$ neurons in $M^+_{j,i}$ fired, the potential of $m^+_{j,i}$ is at most -1 and the probability that $m^+_{j,i}$ fired in round t is at most $\frac{1}{1+e^{1/\lambda}} \leq \Theta(\frac{\delta}{n \cdot k \cdot \Delta \cdot t_m \cdot \log 1/\delta})$. ◀

▷ Claim 19. If $\overline{Z}_1, \overline{Z}_2$ are close and $\overline{Z}_2, \overline{Z}_3$ are close, then $\overline{Z}_1, \overline{Z}_3$ are close.

Proof. Let $\overline{X}_1, \overline{X}_2, \overline{X}_3$ be the corresponding input patterns, where $\overline{Z}_i = R_2(\overline{X}_i)$ for $i \in \{1, 2, 3\}$. By the definition of the clustering instance, every pair of patterns $\overline{X}_i, \overline{X}_j$ are either with relative distance at least $\Delta/2$ (i.e., if these patterns belong to different clusters), or have relative distance at most Δ/α (i.e., if they belong to the same cluster) for $\alpha = \Omega(\log(1/\Delta))$.

By Lemma 12, input patterns $\overline{X}_i, \overline{X}_j$ that belong to different (resp., same) clusters are mapped to far (resp., close) vectors $\overline{Z}_i, \overline{Z}_j$. We therefore have that $\overline{X}_1, \overline{X}_2$ are in the same cluster, and also $\overline{X}_2, \overline{X}_3$ are in the same cluster, concluding that $\overline{X}_1, \overline{X}_2, \overline{X}_3$ are all in the same cluster. ◁

▷ Claim 20. For every $j \in [k]$ and $i \in [\ell]$ w.h.p. the memory module $M_{j,i}$ is active in round t given that it was not active in round $t-3$, only if C^M_j and $a_{j,i}$ fired in round $t-2$.

Proof. By Observation 18 $M_{j,i}$ is activated in round t only if at least $(c/2) \log(1/\delta')$ neurons $m \in M^+_{j,i}$ fire in round $t-1$. Since $M_{j,i}$ was not active in round $t-3$ all feedback neurons f_m for $m \in M^t_{j,i}$ was not active in round $t-2$ and the potential of each $m \in M^+_{j,i}$ in round $t-1$ is $\sum_{u \in C^M_j} 2 \cdot (u)^{t-2} + 2\lambda \cdot (a_{ij})^{t-2} - 9 - 2\lambda$. Hence, if C^M_j and $a_{j,i}$ fired in round $t-2$, in the next round the potential of each $m \in M^+_{j,i}$ is at least 1 and m fires in round $t-1$ with probability at least $1 - \Theta(\frac{\delta}{n \cdot k \cdot \Delta \cdot t_m \cdot \log 1/\delta})$. Thus, by Chernoff bound w.h.p. at least $(c/2) \log(1/\delta')$ neurons $m \in M^+_{j,i}$ fired in round $t-1$.

On the other hand if C^M_j and $a_{j,i}$ did not fire together in round $t-2$, the potential of every $m \in M^t_{j,i}$ in round $t-1$ is at most -2λ and m fires with probability at most $\frac{1}{1+e^2}$. Using Chernoff bound and choosing c to be sufficiently large, we conclude that $(c/2) \log(1/\delta')$ neurons $m \in M^+_{j,i}$ fire in round $t-1$ with probability at most δ'. ◁

▶ **Observation 21.** *For every output q_j at each round w.h.p. the number of memory modules $M_{j,i}$ that are active is at most ℓ.*

Proof. Since every pattern has at most ℓ non zero entries, in each round at most ℓ association neurons $a_{j,i}$ fire. By Claim 20, at each round at most ℓ memory modules $M_{j,i}$ are activated for the first time. If in round $t-3$ more then 0.1ℓ memory modules were active, the only association neurons firing in round $t-2$ correspond to the activated memory modules and therefore w.h.p. no new modules are activated in round t. Else, by Observation 17 w.h.p. the deletion neuron d^3_j kills the active memory modules and no memory module is active in round t. ◀

Using the same arguments, since the deletion neurons erase the partial memory, we can also conclude that for every output neuron all its active memory modules correspond to the same input pattern.

▶ **Observation 22.** *For each output neuron q_i in each round if it has active memory modules, there exists an input pattern $\overline{Z}$ s.t if $M_{i,j}$ is active then $j \in \text{supp}(\overline{Z})$.*

▷ Claim 23. If $\overline{Z}$ is mapped to q_j in round t, with probability greater than $1-\delta$ at least 0.8ℓ memory modules $M_{j,i}$ where $i \in \text{supp}(\overline{Z})$ are active for $c_1 \cdot t_m$ consecutive round starting from round $t+8$.

Proof. Let $\overline{Z}$ be a pattern mapped to q_j in round t. Recall that we assume persistence and therefore w.h.p. $\overline{Z}$ is also mapped to q_j in rounds $t+1$ to $t+8$.

- First we argue that at least 0.8ℓ of the association neurons $a_{j,i}$ for $i \in \text{sup}(\overline{Z})$ fire in round $t+6$. From Observation 17 either there where no memory modules corresponding to q_j active before $\overline{Z}$ was introduces or at least 0.9ℓ [6]. If there where no memory modules active, all association neurons $a_{j,i}$ for $i \in \text{sup}(\overline{Z})$ fire starting round $t+1$ ahead as long as $\overline{Z}$ persist. Otherwise, since q_j fired in round $t+7$, we conclude that at least 0.5ℓ association neurons $a_{j,i}$ fired in round $t+6$. The association neurons that fired are from the support of $\overline{Z}$ and together with Observation 22 we conclude that the pattern previously mapped to q_j is close to $\overline{Z}$ and at least 0.8ℓ association neurons fired in rounds $t+6$ (due to Lemma 12).
- For $i \in sup(\overline{Z})$ for which a_{ij} fired in rounds $t+6$, we now calculate the probability that M_{ij} is active in round $t+8$. By Observation 18 its enough to calculate the probability that at least $(c/2)\log(1/\delta')$ neurons $m \in M^+_{j,i}$ fired in round $t+7$. The potential function of every $m \in M^+_{j,i}$ is given by

  $$\text{pot}(m,t) = \sum_{u \in C^M_j} 2 \cdot (u)^{t-1} + 2\lambda \cdot (a_{ij})^{t-1} + (9+\lambda \cdot (2+\chi)) \cdot (f_m)^{t-1} - 9 - 2\lambda \ .$$

  Since q_j fires in rounds t to $t+7$, the delay chain C^M_j fired in round $t+6$, and the probability m fires in round $t+7$ is at least $1-\Theta(\frac{\delta}{n\cdot k\cdot\Delta\cdot t_m\cdot\log 1/\delta})$. Using Chernoff bound with probability greater than $1-\delta/3\ell$ at least $\frac{c\log(1/\delta')}{2}$ neurons in $M_{j,i}$ fire in round $t+7$ and the head memory neuron $m^+_{j,i}$ fires in round $t+8$.
- Next we calculate the probability that $m \in M^+_{j,i}$ fires $c_1 t_m$ consecutive rounds starting round $t+8$ given that $m^+_{j,i}$ fires in round $t+8$. Since $m^+_{j,i}$ fired in round $t+8$, for every $m \in M^+_{j,i}$ that fired in round $t{+}8$, the feedback neuron f_m is activated in round $t{+}9$ and m fires in round $t{+}10$ with probability at least $1{-}1/t_m$. Hence, the probability $m \in M^+_{j,i}$ fires in rounds $t{+}8$, $t{+}9$ and $c_1 t_m$ consecutive rounds is at least $(1-\Theta(\frac{\delta}{n\cdot k\cdot\Delta\cdot t_m\cdot\log 1/\delta}))^2\cdot(\frac{1}{e^{c_1}})$. We chose c_1 such that this is greater than $1/2$. Thus, using Chernoff bound and a large enough c (depending on c_1) the probability that at least $\frac{c\log(1/\delta')}{2}$ neurons $m \in M^+_{j,i}$ fire in rounds $t+8$, $t+9$ and then for $c_1 t_m$ consecutive rounds is at least $1-\delta'/3 = 1-\delta/3\ell$.
- Summing things up, the probability $M_{j,i}$ is active for $c_1 \cdot t_m$ consecutive rounds from round $t+8$ ahead is at least the probability that $m^+_{j,i}$ fired in round $t+8$ and $\frac{c\log(1/\delta')}{2}$ neurons in $M^+_{j,i}$ fires $c_1 t_m$ consecutive rounds starting round $t+8$. By union bound this probability is greater than

  $$1 - 3\cdot(\delta/3\ell) = 1-\delta/\ell.$$

Thus, we conclude that the probability all 0.8ℓ modules $M_{j,i}$ s.t $a_{j,i}$ fired in round $t+6$ are active for $c_1 \cdot t_m$ consecutive rounds is greater than $1-\delta$. ◁

We are now ready to prove the correctness of the sequential mapping step.

[6] up too ±3 rounds, but since we assume persistence its ok.

Proof of Theorem 14

Proof. We start by proving the 3 main properties of the network. Given a pattern $\overline{Z}$ introduced in round t we will show:

(1) $\overline{Z}$ is <u>mapped</u> to one of the outputs $q_1, \ldots, q_k$ in round $t+6$

(2) For any pattern $\overline{Z}'$ which is *close* to $\overline{Z}$ and was introduced within a span of $c_1 \cdot t_m$ rounds from t, $\overline{Z}$ and $\overline{Z}'$ are mapped to the same output neuron.

(3) For any pattern $\overline{Z}'$ which is *far* from $\overline{Z}$ and was introduced within a span of $c_1 \cdot t_m$ rounds from t, $\overline{Z}$ and $\overline{Z}'$ are mapped to a different output neuron.

By induction on the order of arrival of the patterns. Let $\overline{Z}$ be the first pattern arrived in round 0. We show that $\overline{Z}$ is mapped to the first (available) neuron q_1 in round 6. For every $i \in \sup(\overline{Z})$ the potential function of the association neuron $a_{1,i}$ is given by:

$$pot(a_{1,i}, t) = 2\ell(z_i)^{t-1} + \ell(m^+_{1,i})^{t-1} - \sum_{j \neq i}(m^-_{1,j})^{t-1} - (19/10)\ell - 1 \ .$$

Since $\overline{Z}$ is the first pattern seen, no neuron has fired in round zero and $\mathrm{pot}(a_{1,i}, 1) = (1/10)\ell - 1 > 1$, and w.h.p. each $a_{1,i}$ for $i \in \sup(\overline{Z})$ fires in round 1.

Since q_1 is the first output, no preceding output neuron inhibits it, and its potential is:

$$\mathrm{pot}(q_1, t) = \sum_{i=1}^{r}(2 \cdot a_{1,i})^{t-1} - \sum_{i=2}^{k} 3\ell h_i - \ell + 1 \ .$$

By Claim 13, w.h.p. each input pattern $\overline{Z}$ (to the sequential mapping network) has at least 0.98ℓ non-zero entries (and at most ℓ). Therefore, at least 0.98ℓ association neurons $a_{1,i}$ excite q_1 in round 2. Recall that the history neurons h_i fire only if at least $1/2$ of the corresponding memory modules are active in the previous round. Hence w.h.p. in round 1, no history neuron fires.

We conclude that q_1 fires in round 2 w.h.p. By Claim 20 every memory module $M_{i,j}$ becomes active only after having q_i firing for 5 consecutive rounds (due to the delay chain C^M_i). For that reason, no memory module fires before round 5. Since the memory neurons are not active, $a_{1,i}$ keeps firing in rounds 1 to 6, and q_1 keeps firing in rounds 2 to 7. Since q_1 is connected to q^-_1 via a delay chain C^I_1 of length 3, starting round 5 (and as long as q_1 fires), the inhibitor q^-_1 inhibits all other output neuron q_i for $i \geq 2$. Thus, for every $i \geq 2$ the potential of q_i in round 6 is at most $\ell - 2\ell - 1/2\ell + 1 < -1$. As a result, for $i \geq 2$ neuron q_i does not fire starting round 6.

We next argue that at this point, no memory modules are yet active and consequently the history neurons are inactive as well. This is due to the fact that the delay in the inhibition of q_i by q_1 is *shorter* then the delay chain C^M_i that starts at q_i and ends at the memory modules. Thus q_i is inhibited before its memory modules are activated. We conclude that if $\overline{Z}$ is observed, starting from round 6, the output neuron q_1 is the only active output neuron, and $\overline{Z}$ is *mapped* to q_1.

Assume the claim holds for the first $i-1$ presented patterns, we next consider the i^{th} pattern $\overline{Z}$ presented in round t.

- We first show that for every $\overline{Z}'$ that is far from $\overline{Z}$ introduced in round $t' \in [t - c_1 \cdot t_m, t-1]$, the pattern $\overline{Z}$ will be mapped to a different output. By the induction assumption, $\overline{Z}'$ is mapped in round $t'+6$ to some output neuron, q_j. Since $\overline{Z}'$ was introduced within $c_1 \cdot t_m$ rounds, by Claim 23 at least 0.8ℓ many memory modules $M_{j,i}$ for $i \in \sup(\overline{Z}')$ are active it round t.

Let $\overline{X}$, $\overline{X}'$ be the inputs corresponding to $\overline{Z}$ and $\overline{Z}'$ respectively. From Lemma 12, $\|\sup(\overline{Z}) \cap \sup(\overline{Z'})\| \leq 0.1\ell$. By Observation 21, the number of active memory modules $M_{j,i}$ in round t is at most ℓ. Thus, the number of active memory modules $M_{j,i}$ in round t for $i \in \sup(\overline{Z})$ is at most $0.2\ell + 0.1\ell = 0.3\ell$.

For every association neuron $a_{j,i}$ whose memory module $M_{j,i}$ is inactive in round t, there are at least 0.8ℓ memory modules that inhibit it. Therefore its potential is $2\ell - 8/10\ell - (19/10)\ell + 1 < -1$, and w.h.p. it does not fire in round $t+1$. Overall, at most 0.3ℓ association neurons $a_{j,i}$ start firing from round $t+1$ and as long as the pattern persists. We conclude that q_j will stop firing from round $t+2$.

- Next we show that two close patterns $\overline{Z}'$ and $\overline{Z}$, introduced within a span of $c_1 \cdot t_m$ rounds are mapped to the same output. First note that by Claim 19, all patterns that are close to $\overline{Z}$ are close to each other. Hence, by the induction assumption, all patterns close to $\overline{Z}$ introduced within the last $c_1 t_m$ rounds were mapped to the same output. We now consider a pattern $\overline{Z}'$ close to $\overline{Z}$ introduced in round $t' \in [t - c_1 \cdot t_m, t - 10]$. By the induction assumption, the pattern $\overline{Z}'$ was mapped to output q_j in round $t' + 6$. By claim 23, 0.8ℓ many memory modules $M_{j,i}$ are active in round $t' + 7 < t$ onward (i.e., for $\Theta(t_m)$ rounds). Combining with Lemma 12 because $\overline{Z}$ is close to $\overline{Z}'$, at least 0.7ℓ many memory modules $M_{j,i}$, $i \in \sup(\overline{Z})$ are active in round t. Since there are at most ℓ many active memory modules associated with q_j in round t, the potential of the association neuron $a_{j,i}$ for which the memory neuron is active in round $t+1$ is at least $2\ell + \ell - (\ell - 1) - 1.9\ell - 1 = 0.1\ell > 1$. We have that at least 0.7ℓ many association neuron $a_{j,i}$ fire in round $t+1$, leading to the firing of q_j in round $t+2$.

  Next, because at least 0.8ℓ memory modules $M_{j,i}$ are active from round $t+3$ ahead, the history neuron h_j fires, and by that inhibits all output neurons q_i for $i \leq j-1$ starting from round $t+4$. Recall that every inhibitor neuron q_i^- for $i \leq j-1$ starts firing only after the delay chain C_i^I fired, i.e. after 3 rounds that q_i fired. Hence the history neuron h_j inhibits every q_i for $i \leq j-1$, just before q_i^- starts firing. We next show that no other q_i fires for $i \geq j+1$. This holds since q_j^- inhibits any such q_i in round $t+6$ via the delay chain C_j^I. Finally, we show that q_j continues firing as long the pattern persists. Because at least 0.5ℓ of the association neurons $a_{j,i}$ are firing, and no preceding inhibitor q_i^- is currently firing (thanks to the history neuron h_j), it remains to show that no other history neuron h_i for $i \neq j$ inhibits q_j. By the induction assumption, all patterns close to $\overline{Z}$ were mapped to q_j. Hence if for some other output q_i for $i \neq j$, at least 0.5ℓ of its associated memory modules are active, by Observation 17 at least 0.9ℓ memory modules are active. Since the pattern $\overline{Z}''$ that was mapped to q_i is far from $\overline{Z}$, we have that at most 0.2ℓ many memory modules $M_{i,i'}$ for $i' \in sup(\overline{Z})$ are active, thus q_i does not fire and consequently h_i does not fire.

- We now consider the case of a newly presented pattern, i.e., no close pattern to it has been presented in the last $\Theta(t_m)$ rounds. We will show that in such a case, $\overline{Z}$ will be mapped to the left-most available output q_j, where by available we mean that no memory module $M_{j,i}$ is active in round t. Let $q_{i_1}, q_{i_2} \ldots q_{i_s}$ be the available output neurons in round t. Hence all the association neurons $a_{i_1,i}$ for $i \in \sup(\overline{Z})$ start firing in round $t+1$. This is because no memory module $M_{i_1,j}$ is active. Thus, in round $t+2$ the output neuron q_{i_1} starts firing. As for the unavailable neurons q_j, by Observation 17 at least 0.9ℓ memory modules M_{ji} are active and by Observation 22 they are associated with a pattern $\overline{Z''}$ which was mapped to q_j. The pattern $\overline{Z''}$ is far from $\overline{Z}$ ($\overline{Z}$ is a new pattern) and therefore at most 0.2ℓ memory module $M_{j,i}$ for $i \in \sup(\overline{Z})$ are active in round t. Hence, at most 0.2ℓ association neurons $a_{j,i}$ fire starting round $t+1$ and w.h.p. q_j will not fire starting round $t+2$ (and also no history neuron inhibits q_{i_1}). Since we assume persistence, and due to the delay in the activation of the memory modules, q_{i_1} fires also in rounds $t+3$ and $t+4$, and in round $t+5$ the inhibitor $q^-_{i_1}$ starts firing, inhibiting all the successive output neurons q_j for $j \geq i_1 + 1$.

In order to finish the proof of Theorem 14 we will prove th following Lemma.

▶ **Lemma 24** (Reset, Clearance of Memory). *Let $\overline{Z}$ be a pattern last introduced in round t and mapped to q_j. If no close pattern $\overline{Z}'$ is introduced in rounds $[t, t + c_2 \cdot t_m]$, then q_j is released in some round $\tau \leq t + c_2 t_m$, i.e., all memory modules $M_{j,i}$ stop firing with probability greater than $1 - \delta$.*

Proof. Let $\overline{Z}$ be a pattern last introduced in round t and mapped to q_j. By Observation 17 if in some round τ there are less than 0.9ℓ memory modules corresponding to q_j firing, after 3 round 0 memory modules are active and w.h.p. q_j is released. As long as there are at least 0.9ℓ memory modules firing, since all patterns introduced in rounds t to $t + c_2 t_m$ are far from $\overline{Z}$ by the same arguments used in Lemma 14 starting from round $t+1$ less than 0.5ℓ association neurons associated with q_j fire and q_j will not fire for $c_2 t_m$ consecutive rounds starting from round $t+2$ (as long as it is not already released). Thus, from round $t+7$ ahead w.h.p. all neurons in the delay chain C^M_j do not fire.

Therefore, the probability neuron $m \in M^+_{j,i}$ fires in round $\tau \in [t+8, t+c_2 t_m]$ given that f_m did not fire in round $\tau - 1$ is at most $\Theta(\frac{\delta}{\log 1/\delta \cdot n \cdot k \cdot \Delta \cdot t_m})$. Moreover, by union bound the probability that there exists a neuron $m \in M^+_{j,i}$ that fired in some round $\tau \in [t+8, t+c_2 t_m]$ given that f_m did not fire in round $\tau - 1$ is at most $\delta/2\ell$.

Next we calculate the probability at least half of the neurons $m \in M^+_{j,i}$ fire for $c_2 t_m$ consecutive rounds. Because the delay chain C^M_j do not fire starting round $t+7$ the potential of each neuron $m \in M^+_{j,i}$ in round $t' \in [t+7, t+c_2 t_m]$ is bounded by $\lambda \cdot (\chi + 2) = \lambda(\log(t_m - 1) + 2)$. Hence, the probability $m \in M^+_{j,i}$ fires in round t' is at most $1 - \frac{1}{e^2(t_m-1)+1} < 1 - \frac{1}{e^2 t_m}$. We conclude that the probability a neuron $m \in M^+_{j,i}$ fires for $c_2 t_m$ consecutive rounds is at most e^{c_2/e^2} which for $c_2 > e^2 log(3)$ is less than $1/3$. Using Chernoff bound and a sufficient large c (constant depending on c_2) the probability that at least $(c/2)\log(1/\delta')$ neurons in $M^+_{j,i}$ fire for $c_2 \cdot t_m$ consecutive rounds starting round $t+7$ is at most $\delta/2\ell$.

If $m^+_{j,i}$ fires for $c_2 t_m$ consecutive rounds starting round $t+8$, be Observation 18 at each round at least $1/2$ of the neurons in $M^+_{j,i}$ fired. Given that no neuron $m \in M^+_{j,i}$ fires in round $\tau \in [t+8, t+c_2 t_m]$ unless f_m fired in round $\tau - 1$, the head neuron $m^+_{j,i}$ fires for $c_2 t_m$ consecutive rounds only if at least $1/2$ of the neurons in $M^+_{j,i}$ fires for $c_2 t_m$ consecutive rounds. Thus we conclude that $m^+_{j,i}$ fired for $c_2 t_m$ consecutive rounds starting round $t+8$ with probability at most $\delta/(2\ell) + \delta/(2\ell) = \delta/\ell$. Note that by Observation 21 at most ℓ memory neurons M_{ij} are active at each round and using union bound we conclude that with probability at least $1 - \delta$ the output neuron q_j is release in round $\tau < t + c_2 t_m$ ◀

This concludes Theorem 14 and therefore also Theorem 3. ◀

References

1 Jayadev Acharya, Arnab Bhattacharyya, and Pritish Kamath. Improved bounds for universal one-bit compressive sensing. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 2353–2357, 2017.

2 Zeyuan Allen-Zhu, Rati Gelashvili, Silvio Micali, and Nir Shavit. Sparse sign-consistent Johnson–Lindenstrauss matrices: Compression with neuroscience-based constraints. *PNAS*, 2014.

3 Cornelia I Bargmann and Eve Marder. From the connectome to brain function. *Nature Methods*, 10(6):483–490, 2013.

4 Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM, 2001.

5 Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

6 Petros T Boufounos and Richard G Baraniuk. 1-bit compressive sensing. In *42nd Annual Conference on Information Sciences and Systems (CISS 2008).*, pages 16–21. IEEE, 2008.

7 Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. Random projections for k-means clustering. In *Advances in Neural Information Processing Systems 23 (NIPS)*, 2010.

8 Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12):5406–5425, 2006.

9 Sophie JC Caron, Vanessa Ruta, LF Abbott, and Richard Axel. Random convergence of olfactory inputs in the Drosophila mushroom body. *Nature*, 497(7447):113, 2013.

10 Chi-Ning Chou, Kai-Min Chung, and Chi-Jen Lu. On the algorithmic power of spiking neural networks. *arXiv preprint*, 2018. arXiv:1803.10375.

11 Kenneth L Clarkson and David P Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, 2013.

12 Michael B Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, 2015.

13 Graham Cormode and Shan Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

14 Robert Coultrip, Richard Granger, and Gary Lynch. A cortical model of winner-take-all competition via lateral inhibition. *Neural Networks*, 5(1):47–54, 1992.

15 Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. Fast locality-sensitive hashing. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1073–1081. ACM, 2011.

16 Sanjoy Dasgupta, Timothy C Sheehan, Charles F Stevens, and Saket Navlakha. A neural data structure for novelty detection. *Proceedings of the National Academy of Sciences*, 115(51):13093–13098, 2018.

17 Sanjoy Dasgupta, Charles F Stevens, and Saket Navlakha. A neural algorithm for a fundamental computing problem. *Science*, 358(6364):793–796, 2017.

18 Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004.

19 David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

20 Surya Ganguli and Haim Sompolinsky. Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis. *Annual Review of Neuroscience*, 2012.

21 Simon S Haykin. *Neural networks and learning machines*, volume 3. Pearson, 2009.

22 Yael Hitron and Merav Parter. Counting to Ten with Two Fingers: Compressed Counting with Spiking Neurons. *ESA*, 2019. arXiv:1902.10369.

23 John J Hopfield, David W Tank, et al. Computing with neural circuits- A model. *Science*, 233(4764):625–633, 1986.

24 Laurent Jacques, Jason N Laska, Petros T Boufounos, and Richard G Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE Transactions on Information Theory*, 59(4):2082–2102, 2013.

25 Robert T Knight. Contribution of human hippocampal region to novelty detection. *Nature*, 383(6597):256, 1996.

26 Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987.

27 Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 2015.

28 Robert A. Legenstein, Wolfgang Maass, Christos H. Papadimitriou, and Santosh Srinivas Vempala. Long Term Memory and the Densest K-Subgraph Problem. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 57:1–57:15, 2018.

29 Andrew C Lin, Alexei M Bygrave, Alix De Calignon, Tzumin Lee, and Gero Miesenböck. Sparse, decorrelated odor coding in the mushroom body enhances learned odor discrimination. *Nature neuroscience*, 17(4):559, 2014.

30 Adi Livnat and Christos Papadimitriou. Evolution and learning: used together, fused together. A response to Watson and Szathmáry. *Trends in Ecology & Evolution*, 31(12):894–896, 2016.

31 Nikos K Logothetis. What we can do and what we cannot do with fMRI. *Nature*, 453(7197):869, 2008.

32 Nancy Lynch and Cameron Musco. A Basic Compositional Model for Spiking Neural Networks. *arXiv preprint*, 2018. arXiv:1808.03884.

33 Nancy Lynch, Cameron Musco, and Merav Parter. Computational Tradeoffs in Biological Neural Networks: Self-Stabilizing Winner-Take-All Networks. In *Proceedings of the 8th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2017.

34 Nancy Lynch, Cameron Musco, and Merav Parter. Neuro-RAM Unit with Applications to Similarity Testing and Compression in Spiking Neural Networks. In *Proceedings of the 31st International Symposium on Distributed Computing (DISC)*, 2017.

35 Nancy Lynch, Cameron Musco, and Merav Parter. Spiking Neural Networks: An Algorithmic Perspective. In *5th Workshop on Biological Distributed Algorithms (BDA 2017)*, July 2017.

36 Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997.

37 Wolfgang Maass. On the computational power of winner-take-all. *Neural computation*, 12(11):2519–2535, 2000.

38 Christos H Papadimitriou and Santosh S Vempala. Cortical learning via prediction. In *Conference on Learning Theory*, pages 1402–1422, 2015.

39 Christos H Papadimitriou and Santosh S Vempala. Random Projection in the Brain and Computation with Assemblies of Neurons. In *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

40 Narender Ramnani and Adrian M Owen. Anterior prefrontal cortex: insights into function from anatomy and neuroimaging. *Nature Reviews. Neuroscience*, 5(3):184, 2004.

41 Charan Ranganath and Gregor Rainer. Cognitive neuroscience: Neural mechanisms for detecting and remembering novel events. *Nature Reviews Neuroscience*, 4(3):193, 2003.

42 Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2006.

43 Olaf Sporns, Giulio Tononi, and Rolf Kötter. The human connectome: a structural description of the human brain. *PLoS Computational Biology*, 1(4):e42, 2005.

44 Leslie G Valiant. *Circuits of the Mind*. Oxford University Press on Demand, 2000.

45 Leslie G Valiant. Memorization and association on a realistic neural model. *Neural computation*, 17(3):527–555, 2005.

46 Leslie G Valiant. Capacity of Neural Networks for Lifelong Learning of Composable Tasks. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 367–378, 2017.

47 Santosh S Vempala. *The random projection method*, volume 65. American Mathematical Society, 2005.

A Additional Proofs: Random Projection

We first prove Lemma 5, that a Chi-squared distribution is nearly uniform within a constant number of standard deviations from its mean.

▶ **Lemma 5.** *Let $\mathcal{D}_p$ be the Chi-squared distribution with p degrees of freedom. For any c with $1 \le c < p^{1/2}$ there are constants c_ℓ, c_u (depending on c) such that, for any interval $[r_1, r_2] \subseteq \left[p - cp^{1/2}, p + cp^{1/2}\right]$, we have:*

$$\frac{c_\ell (r_2 - r_1)}{p^{1/2}} \le \Pr_{x \sim \mathcal{D}_p} [x \in [r_1, r_2]] \le \frac{c_u (r_2 - r_1)}{p^{1/2}}$$

That is, $\mathcal{D}_p$ is roughly uniform on the range $\left[p - cp^{1/2}, p + cp^{1/2}\right]$.

Proof. It is well known that $\mathcal{D}_p$ has mean p, density $d(x) = \frac{1}{2^{p/2}\Gamma(p/2)} x^{p/2-1} e^{-x/2}$. Since we assume $p^{1/2} > c \ge 1$ we have $p \ge 2$ and the distribution has mode $p - 2$. Additionally, we have $p - cp^{1/2} > 0$. So for $x \in \left[p - cp^{1/2}, p + cp^{1/2}\right]$ we can bound:

$$d(x) \le d(p-2) = \frac{1}{2^{p/2}\Gamma(p/2)} (p-2)^{p/2-1} e^{-p/2+1} \le \frac{1}{\Gamma(p/2)} \cdot \left(\frac{p}{2e}\right)^{p/2-1}$$

By Stirling's approximation, $\Gamma(p/2) \ge \sqrt{\frac{2\pi}{p/2}} \left(\frac{p}{2e}\right)^{p/2}$ which gives:

$$d(x) \le \sqrt{\frac{p}{4\pi}} \cdot \frac{2e}{p} = \frac{e}{\sqrt{\pi} \cdot p^{1/2}}. \tag{2}$$

On the other side, since $p - cp^{1/2} > 0$, and since the density of the Chi-squared distribution is monotonically decreasing as x moves further from the mode $p - 2$ either left or right:

$$d(x) \ge \min(d(p - cp^{1/2}), d(p + cp^{1/2})). \tag{3}$$

We lower bound each term in the minimum.

$$\begin{aligned} d(p - cp^{1/2}) &= \frac{1}{2^{p/2}\Gamma(p/2)} (p - cp^{1/2})^{p/2-1} e^{-p/2+(c/2)p^{1/2}} \\ &= \frac{1}{2\Gamma(p/2)} \left(\frac{p}{2e}\right)^{p/2-1} \cdot \left(1 - \frac{c}{p^{1/2}}\right)^{p/2-1} \cdot e^{(c/2)p^{1/2}-1} \end{aligned}$$

Again using Stirling's approximation, and a similar argument to the proof of (2), for some constant c_1, $\frac{1}{2\Gamma(p/2)} \left(\frac{p}{2e}\right)^{p/2-1}$ is lower bounded by $\frac{c_1}{p^{1/2}}$. Thus,

$$\begin{aligned} d(p - cp^{1/2}) &\ge \frac{c_1}{p^{1/2}} \cdot \left(1 - \frac{c}{p^{1/2}}\right)^{p/2-1} \cdot e^{(c/2)p^{1/2}-1} \\ &\ge \frac{c_1}{p^{1/2}} \cdot \left(1 - \frac{c}{p^{1/2}}\right)^{\left(\frac{p^{1/2}}{c}-1\right)\cdot(c/2)p^{1/2}} \cdot e^{(c/2)p^{1/2}-1} \cdot \left(1 - \frac{c}{p^{1/2}}\right)^{(c/2)p^{1/2}-1} \\ &\ge \frac{c_1}{p^{1/2}} \frac{1}{e^{(c/2)p^{1/2}}} \cdot e^{(c/2)p^{1/2}-1} \cdot \left(1 - \frac{c}{p^{1/2}}\right)^{(c/2)p^{1/2}-1} \\ &\ge \frac{c_1}{ep^{1/2}} \cdot \left(1 - \frac{c}{p^{1/2}}\right)^{\left(\frac{p^{1/2}}{c}-1\right)\cdot(c^2/2)+(c^2/2-1)} \\ &\ge \frac{c_1 \cdot e^{c^2/2}}{ep^{1/2}} \cdot \left(1 - \frac{c}{p^{1/2}}\right)^{c^2/2-1} \ge \frac{c'}{p^{1/2}} \end{aligned} \tag{4}$$

for some constant c' that depends on c. We give a similar bound for $p + cp^{1/2}$.

$$\begin{aligned}
d(p+cp^{1/2}) &= \frac{1}{2\Gamma(p/2)}\left(\frac{p}{2e}\right)^{p/2-1}\cdot\left(1+\frac{c}{p^{1/2}}\right)^{-p/2-1}\cdot e^{(c/2)p^{1/2}-1}\\
&\geq \frac{c_1}{p^{1/2}}\cdot\left(1+\frac{c}{p^{1/2}}\right)^{-p/2-1}\cdot e^{(c/2)p^{1/2}-1}\\
&= \frac{c_1}{p^{1/2}}\cdot\left(1+\frac{c}{p^{1/2}}\right)^{-\frac{p^{1/2}}{c}\cdot(c/2)p^{1/2}}\cdot e^{(c/2)p^{1/2}-1}\cdot\left(1+\frac{c}{p^{1/2}}\right)^{-1}\\
&\geq \frac{c}{ep^{1/2}}\cdot\left(1+\frac{c}{p^{1/2}}\right)^{-1}\geq\frac{c'}{p^{1/2}} \qquad (5)
\end{aligned}$$

for some c'. Combining (4) and (5) with (3) and (2) gives that there exist constants c_ℓ, c_u such that for all $x \in [p - cp^{1/2}, p + cp^{1/2}]$,

$$\frac{c_\ell}{p^{1/2}} \leq d(x) \leq \frac{c_u}{p^{1/2}}.$$

Thus for any r_1, r_2:

$$\frac{c_\ell(r_2 - r_1)}{p^{1/2}} \leq \Pr_{x\sim\mathcal{D}_p}\left[x \in [r_1, r_2]\right] \leq \frac{c_u(r_2 - r_1)}{p^{1/2}},$$

completing the lemma. ◀

We next give a complete proof of Lemma 7.

A.1 Proof of Lemma 7

Since each $[A_b\overline{X}](i)$ is a Chi-squared random variable with p degrees of freedom, which has median $\leq p$, each $[A_b\overline{X}](i)$ is upper bounded by $p \leq \tau = p + 2p^{1/2}$ with probability $\geq 1/2$. Thus, by Lemma 5 applied with $c = 2$, conditioned on $[A_b\overline{X}](i) \leq p + 2p^{1/2}$, there is some c_ℓ with:

$$\Pr\left[[A_b\overline{X}](i) \in \left[p, p+2p^{1/2}\right] \big| [A_b\overline{X}](i) \leq p + 2p^{1/2}\right] \geq \frac{c_\ell \cdot 2p^{1/2}}{p^{1/2}} = 2c_\ell.$$

Thus, for large enough constant c_1 and $m = c_1$, with probability at least $\frac{199}{200}$, we have $i_{1,b}(\overline{X}) \neq 0$ and $[A_b\overline{X}](i_{1,b}(\overline{X})) \geq p$. Call this event $\mathcal{E}_1$. Condition on the event that $\mathcal{E}_1$ occurs and, in particular, that $[A_b\overline{X}](i_{1,b}(\overline{X})) = x$ for any $x \in [p, p + 2p^{1/2}]$. Call this event $\mathcal{E}_{1,x}$. Then for all $j \neq i_{1,b}(\overline{X})$, $[A_b\overline{X}](j)$ is an independent Chi-squared random variable with p degrees of freedom conditioned on either 1) $[A_b\overline{X}](j) \leq x$ or 2) $[A_b\overline{X}](j) \geq p + 2p^{1/2}$. Since $[A_b\overline{X}](j) \leq p \leq x$ with probability at least 1/2, this conditioning at most doubles the density at any one value. Thus, by Lemma 5,

$$\Pr\left[[A_b\overline{X}](j) \in \left[x - \frac{p^{1/2}}{c_2 m}, x\right] \big| \mathcal{E}_{1,x}\right] \leq \frac{2c_u \cdot \frac{p^{1/2}}{c_2 m}}{p^{1/2}}.$$

By a union bound, we thus have:

$$\Pr\left[\exists j:\ [A_b\overline{X}](j) \in \left[x - \frac{p^{1/2}}{c_2 m}, x\right] \big| \mathcal{E}_{1,x}\right] \leq \frac{2c_u \cdot \frac{p^{1/2}}{c_2}}{p^{1/2}} = \frac{2c_u}{c_2}.$$

Setting c_2 sufficiently large ensures that this quantity is bounded by $\frac{1}{200}$. Thus, by a union bound with the probability that $\mathcal{E}_1$ occurs, with probability $\geq \frac{99}{100}$: $i_{1,b}(\overline{X}) \neq 0$ and 2) no $[A_b\overline{X}](j)$ falls in $\left[x - \frac{p^{1/2}}{mc_2}, x\right] = \left[[A_b\overline{X}](i_{1,b}(\overline{X})) - \frac{p^{1/2}}{mc_2}, [A_b\overline{X}](i_{1,b}(\overline{X}))\right]$. This completes the proof.

A.2 Proof of Lemma 8

We first use the relative distance assumption to give a basic claim:

▷ Claim 25. Write $\overline{X}_1, \overline{X}_2$ as $\overline{X}_1 = \chi + \delta_1$ and $\overline{X}_2 = \chi + \delta_2$ where $\chi \in \{0,1\}^n$ is the common vector with $\chi(i) = 1$ iff $\overline{X}_1(i) = \overline{X}_2(i) = 1$. Note that since $\|X_1\| = \|X_2\| = p$ we have $\|\delta_1\| = \|\delta_2\|$. Letting $\Delta = \mathcal{RD}(\overline{X}_1, \overline{X}_2)$,

$$\frac{\|\delta_1\|}{p} = \frac{\Delta}{2}.$$

Proof. We can write:

$$\Delta = \mathcal{RD}(\overline{X}_1, \overline{X}_2) = \frac{\|\overline{X}_1 - \overline{X}_2\|}{p} = \frac{\|\delta_1 - \delta_2\|}{p} = \frac{\|\delta_1\| + \|\delta_2\|}{p}.$$

The claim follows since $\|\delta_1\| = \|\delta_2\|$. ◁

▷ Claim 26. For $i \in [2]$ and $j \in [m] \cup 0$ let $\mathcal{E}_j$ be the event that $j = i_{1,b}(\overline{X}_1)$. With probability $\geq 999/1000$ over the choice of $\mathrm{A}_b\,\chi$, for all j we have:

$$\Pr[\mathcal{E}_j \mid \mathrm{A}_b\,\chi] \leq \frac{1}{16}.$$

Proof. Let $\Delta = \mathcal{RD}(\overline{X}_1, \overline{X}_2)$ and assume for simplicity that $\Delta \leq 1$ (we will latter see that it is easy to remove this assumption). By Claim 25, $\|\delta_1\| = \|\delta_2\| \leq \frac{p}{2}$ and thus $\|\chi\| \geq \frac{p}{2}$. For a constant c_3 (to be set later) sub-divide the range $\left[\|\chi\| - c_3 p^{1/2}, \|\chi\| + c_3 p^{1/2}\right]$ into $\frac{1}{\Delta^{1/2}}$ subranges of width:

$$2c_3 p^{1/2}\Delta^{1/2} = 2\sqrt{2}c_3\|\delta_1\|^{1/2},$$

where the equality follows from Claim 25. By Lemma 5 (applied with the constant c in the Lemma set to c_3) for any $i \in [m]$, $[A_b\chi](i)$ falls into each range with probability $\Theta(c_3 \cdot \Delta^{1/2})$. Thus, by a standard Chernoff bound, for $m = \frac{c_1 \log 1/\Delta}{\sqrt{\Delta}}$ for sufficiently large c_1, with probability 1999/2000 over the choice of A_b, at least c_4 indices of $A_b\chi$ fall within each bucket where c_4 is a constant to be set later. Note that c_1 depends on c_3, c_4. Call the event that c_4 indices fall into each bucket $\mathcal{E}_{full-buckets}$. Additionally, as argued in Lemma 7, for sufficiently large m, the maximum value $[A_b\bar{X}_1](i_{1,b}(\bar{X}_1))$ below $p + p^{1/2}$ satisfies $[A_b\bar{X}_1](i_{1,b}(\bar{X}_1)) \geq p$ with probability at least 1999/2000. Thus, with probability 1999/2000 over the choice of A_b,

$$\Pr\left[[A_b\bar{X}_1](i_{1,b}(\bar{X}_1)) \geq p \middle| A_b\chi\right] \geq 1999/2000. \tag{6}$$

Let $\mathcal{E}_{good}$ be the event that both $\mathcal{E}_{full-buckets}$ and (6) hold. $\mathcal{E}_{good}$ holds with probability $\geq 999/1000$ over the choice of A_b. First note that conditioning on $\mathcal{E}_{good}$, $\Pr[\mathcal{E}_0 | \mathrm{A}_b\,\chi] \leq \frac{1}{2000}$, easily giving the claim for $j = 0$. We now consider $j \in [m]$. We consider any bucket,

$$R = \left\{j : [A_b\chi](j) \in \left[r, r + 2\sqrt{2}c_3\|\delta_1\|^{1/2}\right]\right\},$$

where r is be some integer multiple of $2\sqrt{2}c_3\|\delta_1\|^{1/2}$. Roughly, since each index in R has a very similar value in $A_b\chi$, each has nearly the same likelihood of being the largest entry in A_bX_1 below $\tau = p + 2p^{1/2}$. Since $\mathcal{E}_{full-buckets}$ occurs, there are at least c_4 of these indices and thus if c_4 is large, none has very high probability of being the largest entry. Formally, we will show that, assuming $\mathcal{E}_{good}$ holds, for each $j \in R$,

$$\Pr[\mathcal{E}_j \mid \mathrm{A}_b\,\chi] \leq \frac{1}{16}. \tag{7}$$

Since this bound holds for all buckets in the range $[\|\chi\| - c_3 p^{1/2}, \|\chi\| + c_3 p^{1/2}]$, it will give the claim after arguing that no index with $A_b\chi$ falling outside this range is likely to have $\mathcal{E}(1,j)$ occur either.

Indices in Buckets. Each entry of $A_b\delta_1$ is identically distributed as an independent Chi-squared random variable with $\|\delta_1\|$ degrees of freedom. Additionally, $A_b\delta_1$ is independent of $A_b\chi$ since δ_1 and χ have disjoint supports. Consider $j \in R$ with $\Pr[[A_b\bar{X}_1](j) \geq \tau | A_b\chi] \geq 15/16$. In this case, since $\mathcal{E}(1,j)$ can only hold if $[A_b\bar{X}_1](j) \leq \tau$, (7) trivially holds.

Next consider $j \in R$ with $\Pr[[A_b\bar{X}_1](j) \geq \tau | A_b\chi] \leq 15/16$. By Lemma 6 there is some c with:

$$\Pr\left[[A_b\delta_1](j) \geq \|\delta_1\| + c\|\delta_1\|^{1/2}\right] = \frac{1}{64},$$

or equivalently since $[A_b\overline{X}_1](j) = [A_b\chi](j) + [A_b\delta_1](j)$:

$$\Pr\left[[A_b\overline{X}_1](j) \geq [A_b\chi](j) + \|\delta_1\| + c\|\delta_1\|^{1/2} \middle| A_b\chi\right] = \frac{1}{64},$$

Setting $r_2 = \min\left([A_b\chi](j) + \|\delta_1\| + c\|\delta_1\|^{1/2}, \tau\right)$ we thus have that

$$\Pr[[A_b\overline{X}_1](j) \in [r_2, \tau] \big| A_b\chi] \leq \frac{1}{64}. \tag{8}$$

Additionally, by Lemma 5 there is some r_1 with $r_2 - r_1 = \Theta(\|\delta_1\|^{1/2})$ such that:

$$\Pr\left[[A_b\overline{X}_1](j) \in [r_1, r_2] \big| A_b\chi\right] = \frac{1}{32}.$$

Since for all $j' \in R$, $|[A_b\chi](j) - [A_b\chi](j')| \leq 2\sqrt{2}c_3\|\delta_1\|^{1/2} = O(\|\delta_1\|^{1/2})$ we have $r_2 = [A_b\chi](j') + O(\|\delta_1\|^{1/2})$ and thus again by Lemma 5, for all $j' \in R$:

$$\Pr\left[[A_b\overline{X}_1](j') \in [r_1, r_2] \big| A_b\chi\right] = \Omega(1).$$

If we set the constant c_4 large enough, since assuming $\mathcal{E}_{good}$, $|R| \geq c_4$ we have:

$$\Pr\left[\exists\, j' \in R \setminus j : \ [A_b\overline{X}_1](j') \in [r_1, r_2] \big| A_b\chi\right] \geq \frac{31}{32}.$$

If this event holds, we can only have $\mathcal{E}(1,j)$ occur if $[A_b\overline{X}_1](j)$ falls in $[r_2, \tau]$, which by (8) occurs with probability $\leq \frac{1}{64}$ conditioned on $A_b\chi$. Thus by a union bound we have:

$$\Pr[\mathcal{E}_j \mid \mathrm{A}_b\,\chi] \leq \frac{1}{16},$$

giving (7) in this case.

Indices Outside Buckets. We now consider indices not falling in any bucket: that is, j with $[A_b\chi](j) \leq \|\chi\| - c_3p^{1/2}$ or $[A_b\chi](j) \geq \|\chi\| + c_3p^{1/2}$. For the later, to have $\mathcal{E}_b(1,j)$ occur we must have $[A_b\overline{X}_1](j) \leq \tau = p + 2p^{1/2}$ and thus $[A_b\delta_1](j) \leq \|\delta_1\| - (c_3-2)p^{1/2} \leq \|\delta_1\| - (c_3-2)\sqrt{2}\|\delta_1\|$. By Lemma 6, this occurs with probability $< 1/16$ for all $\|\delta_1\|$ as long as we set c_3 large enough. Similarly, for j with $[A_b\chi](j) \leq \|\chi\| - c_3p^{1/2}$, with probability $\geq 15/16$, we will have $[A_b\delta_1](j) \leq \|\delta_1\| + c_3p^{1/2}$ and thus $[A_b\overline{X}_1](j) \leq p$. Since assuming $\mathcal{E}_{good}$, the maximum value of $A_b\overline{X}_1$ bounded by $\leq \tau$ is $\geq p$, if $[A_b\overline{X}_1](j) \leq p$, $\mathcal{E}_b(1,j)$ will not occur. Thus completes the argument in this case, giving that for all j with $[A_b\chi](j) \leq \|\chi\| - c_3p^{1/2}$ or $[A_b\chi](j) \geq \|\chi\| + c_3p^{1/2}$, $\Pr[\mathcal{E}_b(1,j) \mid \mathrm{A}_b\,\chi] \leq \frac{1}{16}$.

Removing Bound on Δ. Finally, we note that we can remove the assumption that $\Delta \leq 1$. If $\Delta \geq 1$ we can simply have χ encompass some of the non-shared entries in $\overline{X}_1$ until $\|\chi\| \geq \frac{p}{2}$ and $\|\delta_1\| \leq \frac{p}{2}$ as desired. The bound will go through as argued up to constants, since we will still have $\frac{\|\delta\|_1}{p} = \Theta(\Delta)$ as in Claim 25 (note that we always have $\Delta \leq 2$). ◁

We can now complete the proof of Lemma 8. We have:

$$\begin{aligned}\Pr[i_{1,b}(\overline{X}_1) = i_{1,b}(\overline{X}_2) \mid \mathrm{A}_b\,\chi] &= \sum_{j=0}^{m} \Pr\left[i_{1,b}(\overline{X}_1) = i_{1,b}(\overline{X}_2) = j \mid \mathrm{A}_b\,\chi\right]\\ &= \sum_{j=0}^{m} \Pr\left[i_{1,b}(\overline{X}_1) = j \mid \mathrm{A}_b\,\chi\right] \cdot \Pr\left[i_{1,b}(\overline{X}_2) = j | \mathrm{A}_b\,\chi\right]\end{aligned} \tag{9}$$

where the second line follows from the fact that $\mathrm{A}_b\,\overline{X}_1$ and $\mathrm{A}\,\overline{X}_2$ are independent conditioned on $\mathrm{A}_b\,\chi$ since δ_1, δ_2 are disjoint vectors. By Claim 26, with probability $\geq 999/1000$ over the choice of $\mathrm{A}_b\,\chi$ we can bound (9) by:

$$\Pr[i_{1,b}(\overline{X}_1) = i_{1,b}(\overline{X}_2) \mid \mathrm{A}_b\,\chi] \leq \sum_{j=0}^{m} \Pr\left[i_{1,b}(\overline{X}_2) = j \mid \mathrm{A}_b\,\chi\right] \cdot 1/16 = 1/16 \tag{10}$$

where the last line follows simply since $\sum_{j=0}^{m} \Pr\left[i_{1,b}(\overline{X}_2) = j \mid \mathrm{A}_b\,\chi\right] = 1$. Since (10) holds with probability $\geq 999/1000$ over the choice of $\mathrm{A}_b\,\chi$, overall $\Pr[i_{1,b}(\overline{X}_1) = i_{1,b}(\overline{X}_2)] \leq 1/16 + 1/1000$.

Applying Claim 7 and a union bound gives that $i_{1,b}(\overline{X}_1) \neq i_{1,b}(\overline{X}_2)$ and the gaps between the largest and second largest entries of $\mathrm{A}_b\,\overline{X}_1$ and $\mathrm{A}_b\,\overline{X}_2$ (bounded by τ) are both at least $\geq \frac{p^{1/2}}{c_2 \cdot m}$ (or there is at most one such entry), and $[\mathrm{A}_b\,\overline{X}_1](i_1(\overline{X}_1)), [\mathrm{A}_b\,\overline{X}_2](i_1(\overline{X}_2)) \geq p$ with probability $\geq 1 - (1/16 + 1/1000) - 2/100 = .9165$, giving the lemma.

A.3 Proof of Lemma 10

We now give the deferred proof of Lemma 10, which shows that two close inputs are likely to have the same intermediate neuron with the maximum potential $\leq \tau$ in each bucket. We restate the lemma below.

▶ **Lemma 10.** *Let $\overline{X}_1, \overline{X}_2 \in \{0,1\}^n$ be two vectors with $\mathcal{RD}(\overline{X}_1, \overline{X}_2) \leq \Delta/\alpha$. Consider our construction with bucket size $m = \frac{c_1 \log(1/\Delta)}{\sqrt{\Delta}}$. Then for sufficiently large constants c_1, c_2 and $\alpha = O(\log(1/\Delta)^4)$, for any $b \in [\ell]$, defining $i_{1,b}(\cdot)$ and $i_{2,b}(\cdot)$ as in Lemma 7, with probability ≥ 0.97:*

- *$i_{1,b}(\overline{X}_1) = i_{1,b}(\overline{X}_2)$.*
- *For both $j = 1, 2$: $i_{1,b}(\overline{X}_j) \neq 0$, $[A_b\overline{X}_j](i_{1,b}(\overline{X}_j)) \geq p$, and $[A_b\overline{X}_j](i_{1,b}(\overline{X}_j)) - [A_b\overline{X}_j](i_{2,b}(\overline{X}_j)) \geq \frac{p^{1/2}}{c_2 \cdot m}$ or $i_{2,b}(\overline{X}_j) = 0$.*

Proof. By Lemma 7, with probability $\geq 99/100$, for all $i \in [m] \setminus i_{1,b}(\overline{X}_1)$ with $[\mathrm{A}_b\,\overline{X_1}](i) \leq \tau$:

$$[\mathrm{A}_b\,\overline{X}_1](i_{1,b}(\overline{X}_1)) - [\mathrm{A}_b\,\overline{X_1}](i) = \Omega\left(\frac{p^{1/2}}{m}\right), \tag{11}$$

By a similar argument, with probability $\geq 99/100$, for all $i \in [m]$,

$$|\tau - (\mathrm{A}_b\,\overline{X}_1)_i| = \Omega\left(\frac{p^{1/2}}{m}\right). \tag{12}$$

Additionally, by standard sub-exponential concentration (as used in Lemma 6) with probability $\geq 99/100$, for both $j = 1, 2$ and all $i \in m$ we have $[\mathrm{A}_b\,\delta_j](i) \in \|\delta_i\| \pm O(\log m \cdot \|\delta_i\|^{1/2})$. Note that $\log m = O(\log(1/\Delta))$. Additionally, by Claim 25, since $\mathcal{RD}(\overline{X}_1, \overline{X}_2) \leq \Delta/\alpha$ for $\alpha = O(\log(1/\Delta)^4)$, we have for both $i = 1, 2$, $\frac{\|\delta_i\|}{p} \leq \frac{\Delta}{2\alpha} = O\left(\frac{\Delta}{\log(1/\Delta)^4}\right)$. This gives that

$$O(\log m \cdot \|\delta_1\|^{1/2}) = O\left(\frac{\Delta^{1/2} p^{1/2}}{\log(1/\Delta)}\right) = O\left(\frac{p^{1/2}}{m}\right).$$

So for both $j = 1, 2$ and all $i \in m$, $(\mathrm{A}_b\,\delta_j)_i \in \|\delta_i\| \pm O\left(\frac{p^{1/2}}{m}\right)$. So by (11) we have for all $i \neq i_{1,b}(\overline{X}_1)$ with $[\mathrm{A}_b\,\overline{X_1}](i) \leq \tau$:

$$[\mathrm{A}_b\,\overline{X}_2](i_{1,b}(\overline{X}_1)) - [\mathrm{A}_b\,\overline{X}_1](i) =$$
$$[\mathrm{A}_b\,\overline{X}_1](i_{1,b}(\overline{X}_1)) - [\mathrm{A}_b\,\delta_1](i_{1,b}(\overline{X}_1)) + [\mathrm{A}_b\,\delta_2](i_{1,b}(\overline{X}_1)) - [\mathrm{A}_b\,\overline{X}_1](i) = \Omega\left(\frac{p^{1/2}}{m}\right).$$

By (12) we also have,

$$[\mathrm{A}_b\,\overline{X}_2](i_{1,b}(\overline{X}_1)) = [\mathrm{A}_b\,\overline{X}_1](i_{1,b}(\overline{X}_1)) - [\mathrm{A}_b\,\delta_1)](i_{1,b}(\overline{X}_1)) + [\mathrm{A}_b\,\delta_2](i_{1,b}(\overline{X}_1)) \leq \tau.$$

and similarly, for all $i \neq i_{1,b}(\overline{X}_1)$ with $[\mathrm{A}_b\,\overline{X_1}](i) \geq \tau$:

$$[\mathrm{A}_b\,\overline{X}_2](i) \geq \tau.$$

That is, $i_{1,b}(\overline{X}_1)$ is the largest entry of $\mathrm{A}_b\,\overline{X}_2$ under τ, and thus $i_{1,b}(\overline{X}_1) = i_{1,b}(\overline{X}_2)$.

Applying Lemma 7 and a union bound gives the second claim with overall probability $1 - 1/100 - 1/100 - 1/100 = 97/100$. ◀

B Detailed Analysis of the Sparsification Step via WTA

Proof of Claim 13

Proof. Let $i = \arg\max_{j:\overline{Y}(j) \leq \tau} \overline{Y}(j)$. For every neuron $j \in \{1, \ldots, m\}$ in the input vector $\overline{Y}$, let $\overline{R}(j)$ be the random variable that counts the number of rounds in which j fires in a window of $T = \Theta(m^2 \log m)$ rounds. By the construction described above in which all j with $\overline{Y}(j) \geq \tau$ are inhibited with very strong weight, $\overline{R}(j) = 0$ w.h.p. for all such j. Thus we focus on j with $\overline{Y}(j) \leq \tau$. We show that if $\overline{Y}(i) - \overline{Y}(j) = \Omega\left(\frac{p^{1/2}}{m}\right)$ for $j \neq i$, then $\overline{R}(i) \gg \overline{R}(j)$ with probability at least $1 - \Theta(1/m)$.

First, let $\overline{P}$ be the vector of firing probabilities of each intermediate neuron induced by the potentials in $\overline{Y}$ (ignoring the entries that have been zero'd out since $\overline{Y}(j) \geq \tau$). By (1) we have $\overline{P}(i) = \frac{1}{1+e^{-\overline{Y}(i)}}$. Letting $s(x) = 1/(1 + e^{-x})$, we have $s'(x) \in \left[\frac{1}{2}, \frac{3}{4}\right]$ for $x \in [0, 1]$

and can see that if $\overline{Y}(i)-\overline{Y}(j)=\Omega(1/m)$, then also $s(\overline{Y}(i))-s(\overline{Y}(j))=\Omega(1/m)$. That is, a gap of $\Omega(1/m)$ between $\overline{Y}(i)$ and $\overline{Y}(j)$ translates to a gap of $\Omega(1/m)$ between the firing probabilities $\overline{P}(i)$ and $\overline{P}(j)$. To ensure that $\overline{Y}(i),\overline{Y}(j)$ are in $[0,1]$ we can simply rescale the weights of the random connection matrix A by $\frac{1}{2p}$ and shift them by p by adding a bias of p to each intermediate neuron. By Corollary 9, before this shift and scaling, $\overline{Y}(i)\in[p,p+2p^{1/2}]$, so afterwards, $\overline{Y}(i)\in[0,1]$. For all $j\neq i$, since by Corollary 9 we had $\overline{Y}(i)-\overline{Y}(j)=\Omega\left(\frac{p^{1/2}}{m}\right)$ we still have $\overline{Y}(i)-\overline{Y}(j)=\Omega\left(\frac{1}{m}\right)$ as required and thus $\overline{P}(i)-\overline{P}(j)=\Omega\left(\frac{1}{m}\right)$.

By Chernoff bound, with probability of at least $1-c/m$,

$$\overline{R}(i)\geq T\cdot\overline{P}(i)-\sqrt{T\cdot\overline{P}(i)\cdot c\log m}\ \text{ and }\ \overline{R}(j)\leq T\cdot\overline{P}(j)+\sqrt{T\cdot\overline{P}(j)\cdot c\log m)}\ .$$

Hence, with probability $1-2c/m$ we get that

$$\begin{aligned}\overline{R}(i)-\overline{R}(j) &\geq T\cdot(\overline{P}(i)-\overline{P}(j))-\sqrt{T\cdot\overline{P}(i)\cdot c\log m)}-\sqrt{T\cdot\overline{P}(j)\cdot c\log m}.\\ &\geq T\cdot(\overline{P}(i)-\overline{P}(j))-2\sqrt{T\cdot\overline{P}(i)\cdot\log m}=\Omega(T/m)-O(\sqrt{T\cdot\log m})\\ &= \Omega(T/m)\ ,\end{aligned}$$

by taking $T=c'\cdot m^2\log m$ for a sufficiently large constant c'.

Since the incoming weight of each neuron $y_{i,j}$ is $\overline{R}(i)-\overline{R}(j)=\omega(1)$, we get that $y_{i,j}$ fires with probability of $1-\Theta(1/m)$. By doing a union bound over all $m-1$ neurons, and taking large enough constants, we get that with probability at least $99/100$, all neurons $y_{i,j}$ fire for every $j\neq i$. Hence, z_i is the only firing neuron in $\overline{Z}$. ◀

Recall that in Step (1), every input vector $\overline{X}_i$ is projected into ℓ vectors $\overline{Y}_{i,b}=\mathrm{A}_b\cdot\overline{X}_i$ for every $b\in\{1,\ldots,m\}$. On each such vector $\overline{Y}_{i,b}$ we apply the WTA circuit and get a vector $\overline{Z}_{i,b}$. Let $\overline{Z}_i=\overline{Z}_{i,1}\circ\overline{Z}_{i,2}\circ\ldots\circ\overline{Z}_{i,\ell}$ for $\ell=O(\log(t_m/\delta))$, where $\circ$ denotes vector concatenation.

We conclude this section by showing that the relative gap between input patterns $\overline{X}_i,\overline{X}_j$ is reflected in their output vectors of the WTA circuit. By combining Claim 13 with Cor. 9 and 11, we prove Lemma 12 which completes the correctness of Step (II).

Proof of Lemma 12

Proof. First observe that for every input $\overline{X}_i$, there are at most ℓ non-zero entries in $\overline{Z}_i$ since the threshold gates fire only if there is a sufficient gap in the firing rates. (I) For a fixed pair $\overline{X}_i,\overline{X}_j$ of far patterns, let $B_{i,j}$ be the set of all buckets b where $\arg\max_{r\in[m]:\overline{Y}_{i,b}(r)\leq\tau}\overline{Y}_{i,b}(r)\neq\arg\max_{r\in[m]:\overline{Y}_{j,b}(r)\leq\tau}\overline{Y}_{j,b}(r)$ and the gap between largest and second largest entries in both vectors $\overline{Y}_{i,b}$ and $\overline{Y}_{j,b}$ is $\Omega(p^{1/2}/m)$. By Cor. 9, with probability $1-\delta$, for every pair of far patterns $\overline{X}_i,\overline{X}_j$, $|B_{i,j}|\geq 0.9\cdot\ell$.

By Claim 13, if $\overline{Y}_{i,b}$ has a desired gap between the largest entry and other entries then, with probability $p=99/100$, $\overline{Z}_{i,b}$ has exactly one winning entry corresponding to $\arg\max(\overline{Y}_{i,b})$. In expectation the vectors $\overline{Z}_{i,b}$ differ in $p\cdot|B_{i,j}|$ buckets. Thus by applying Chernoff bound overall k^2 pairs, in 0.9ℓ of the buckets, the WTA picks a distinct winner for the $\overline{X}_i$ and $\overline{X}_j$ patterns. Thus, $\mathrm{supp}(\overline{Z}_i)\setminus\mathrm{supp}(\overline{Z}_j)\geq 0.9\ell$.

(II) For a fixed pair $\overline{X}_i,\overline{X}_j$ of close patterns, let $B_{i,j}$ be the set of all buckets b where $\arg\max_{r\in[m]:\overline{Y}_{i,b}(r)\leq\tau}\overline{Y}_{i,b}(r)=\arg\max_{r\in[m]:\overline{Y}_{j,b}(r)\leq\tau}\overline{Y}_{j,b}(r)$ and the gap between largest and second largest entries in both vectors $\overline{Y}_{i,b}$ and $\overline{Y}_{j,b}$ is $\Omega(p^{1/2}/m)$. By Cor. 11 with probability $1-\delta$, for every pair of close patterns $\overline{X}_i,\overline{X}_j$, $|B_{i,j}|\geq 0.91\cdot\ell$. By applying Claim 13 and Chernoff bound overall k^2 pairs, in at least $0.9\cdot\ell$ of the buckets, the selected winner is the same with probability of $1-\delta$, implying that $\mathrm{supp}(\overline{Z}_i)\cap\mathrm{supp}(\overline{Z}_j)\geq 0.9\cdot\ell$. ◀