Journal of Integer Sequences, Vol. 19 (2016),
Article 16.4.7

Counting Hidden Neural Networks

Anthony Richard
Département de Mathématiques et de Statistique
Université Laval
Québec
Canada

Patrick Desrosiers
Centre de Recherche de L'Institut Universitaire en Santé Mentale de Québec and
Département de Physique, de Génie Physique et d'Optique
Université Laval
Québec
Canada

Simon Hardy
Centre de Recherche de L'Institut Universitaire en Santé Mentale de Québec
and
Département de Biochimie, Microbiologie et Bio-Informatique
and
Département d'Informatique et de Génie Logiciel
Université Laval
Québec
Canada

Nicolas Doyon
Département de Mathématiques et de Statistique
Université Laval
and
Centre de Recherche de L'Institut Universitaire en Santé Mentale de Québec
Québec
Canada
nicolas.doyon@mat.ulaval.ca

Abstract

We apply combinatorial tools, including Pólya's theorem, to enumerate all possible networks for which (1) the network contains distinguishable input and output nodes as well as partially distinguishable intermediate nodes; (2) all connections are directed and for each pair of nodes, there are at most two connections, that is, at most one connection per direction; (3) input nodes send connections but don't receive any, while output nodes receive connections but don't send any; (4) every intermediate node receives a path from an input node and sends a path to at least one output node; and (5) input nodes don't send direct connections to output nodes. We first obtain the generating function for the number of such networks, and then use it to obtain precise estimates for the number of networks. Finally, we develop a computer algorithm that allows us to generate these networks. This work could become useful in the field of neuroscience, in which the problem of deciphering the structure of hidden networks is of the utmost importance, since there are several instances in which the activity of input and output neurons can be directly measured, while no direct access to the intermediate network is possible. Our results can also be used to count the number of finite automata in which each cell plays a relevant role.

1 Introduction

The problem of counting networks, graphs or digraphs with a given set of conditions has a rich mathematical history that was greatly influenced by the seminal work of Pólya and Harary [22, 12, 13]. Results in this field have many applications in scientific fields such as chemistry, resources allocation and number theory [19, 1, 3, 7]. Several versions of the graph (or digraph) enumeration problem have been investigated, depending on whether the vertices are labelled (distinguishable) or not (indistinguishable) or whether directed, connected or simple graphs are considered [5, 21, 6].

In some applications, such as the description of neural networks or cellular automata and network coding theory, graphs describe the flow of information from a set of input nodes to output ones [16, 2, 26]. In such cases, the input and output nodes are often called sources and sinks, respectively. An intermediary node (neuron or cell) is relevant to the global behaviour of the system only if it is (directly or indirectly) connected to both an input and an output element. Enumeration problems of such structures have raised interest in the field of finite cellular automata [24, 11], where the size of a given class of automata can be related to the computing power of this class [10, 23]. Moreover, in some instances of neuroscience, it is possible to directly measure the input and the output of a neural network without having access to direct measurement of the intermediate hidden network as in the investigation of the pain processing network of the spinal cord where [20]. The question as to what extent it is possible to infer this structure from the input-output relation is of great importance to the field of computational neuroscience and poses fascinating problems from a mathematical point of view.

1.1 The problem

In the present paper, we enumerate the possible networks for a given number of input, output and intermediate nodes. Since we assume that interactions (input manipulation or data retrieval) are only possible or relevant with input and output nodes, we consider these as labelled while we treat the intermediate nodes as unlabelled. We also differentiate between two types of intermediate nodes. In the context of neural networks, this could correspond to differentiating between excitatory and inhibitory interneurons for instance. The techniques we develop could be used to treat the cases in which the intermediate nodes are subdivided into an arbitrary number of subtypes which could describe automata composed of many types of fundamental units.

Specifically, we enumerate the networks composed of nodes (neurons or cells) and links (connections) that satisfy the following conditions:

1.1 Every node belongs to one and only one of the following types : input nodes, intermediate node of type I, intermediate node of type II, output node.

1.2 All input and output nodes are distinguishable (labelled).

1.3 All intermediate nodes of type I are indistinguishable and all intermediate nodes of type II are indistinguishable.

2.1 All links are directed.

2.2 There is at most one link from a given node to another node.

3.1 All input nodes send outward links but don't receive inward links.

3.2 All output nodes receive inward links but don't send outward links.

3.3 There are no direct link from input nodes to output nodes.

4.1 Every intermediate node receives a path from at least one input node.

4.2 Every intermediate node sends at least a path to an output node.

Recall that the adjacency matrix A of a network with N nodes is a square matrix containing $N \times N$ elements, $A_{i,j}$, such that $A_{i,j} = 1$ if there is a link from node i to node j and $A_{i,j} = 0$ otherwise. Conditions 3.1, 3.2 and 3.3 thus imply that the adjacency matrix A has the following form:

$$A = \begin{pmatrix} 0_{m\times m} & B_{m\times n} & 0_{m\times k} \\ 0_{n\times m} & C_{n\times n} & D_{n\times k} \\ 0_{k\times m} & 0_{k\times n} & 0_{k\times k} \end{pmatrix},$$

where the subscripts indicate the size of the submatrices, $0_{p\times q}$ stands for a submatrix whose elements are all equal to zero, $B_{m\times n}$ and $D_{n\times k}$ are binary matrices with no special form, and $C_{n\times n}$ is a binary matrix whose diagonal elements are all equal to zero.

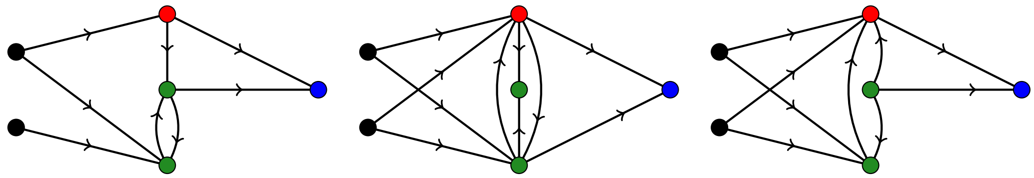


Figure 1: Admissible and non-admissible networks. Black, green, red, and blue represent input nodes, intermediate nodes of type I and II and output nodes respectively. Left: Admissible network. Middle: Non-admissible network. The intermediate node at the center do not send a path to an output node. Right: Non-admissible network. The intermediate node at the center do not receive any path from an input node.

Conditions 4.1 and 4.2 are equivalent to the fact that every intermediate node is relevant in the network. Indeed, the activity of an interneuron that would not receive a path from an input neuron or that would not send a path to an output neuron could not be detected from input/output measurements. Similarly, in the context of finite automata, such a cell would not impact the function computed by the automata. Note that the indistinguishableness of the intermediate node imposes a significant constraint on our enumeration problem : permutations of the intermediate nodes of the same type are only counted once. Any network that complies with all the above conditions is characterized as admissible. Examples of admissible and non-admissible networks are given in Figure 1.

We write $N(m, n_1, n_2, k)$ as the number of admissible networks with m input nodes, n_1 intermediate nodes of type I, n_2 intermediate nodes of type II and k output nodes. For the sake of brevity, we also write $n = n_1 + n_2$. For illustration, we display in Figure 2 all the admissible networks counted by $N(1, 2, 0, 1)$, where the input node is on the left of each network, the intermediate nodes (in green) in the middle and the output one on the right. From this list of examples we conclude that $N(1, 2, 0, 1) = 10$.

1.2 Results summary

We set $N(m, n_1, n_2, k; x)$ as the generating function of the number of admissible networks networks counted by $N(m, n_1, n_2, k)$. Explicitly,

$$N(m, n_1, n_2, k; x) = \sum_j c_j x^j$$

where c_j denotes a function of m, n_1, n_2, and k that counts the number of admissible networks with exactly j connections. We thus have

$$N(m, n_1, n_2, k) = \sum_j c_j.$$

For instance, from the case illustrated in Figure 2, we infer that if $m = 1, n_1 = 2, n_2 = 0$, and $k = 1$, then

$$c_1 = 0, \quad c_2 = 0, \quad c_3 = 1, \quad c_4 = 5, \quad c_5 = 3, \quad c_6 = 1, \quad c_j = 0 \quad (j \geq 7).$$

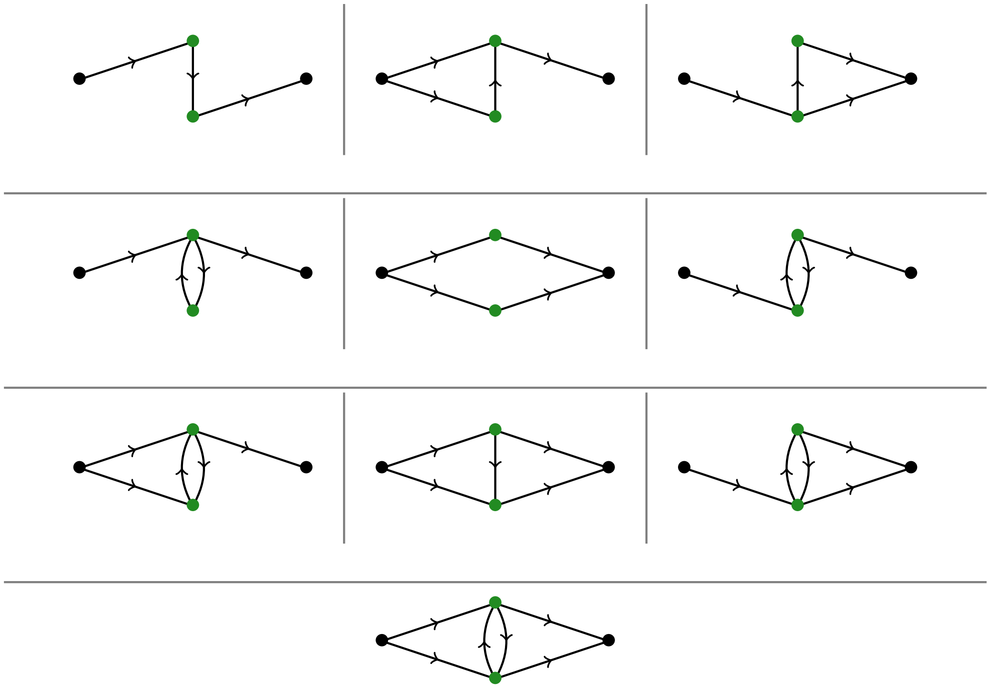


Figure 2: All admissible and non-equivalent networks with one input node (left), two intermediate node of type I (green) and one output node (right).

We need some more notation. We define $N^*(m, n_1, n_2, k)$ as the number of networks satisfying all conditions, except possibly conditions 4.1 and 4.2. We will refer to this quantity as the total number of networks. We also write

$$N^*(m, n_1, n_2, k; x) = \sum_j c_j^* x^j$$

where c_j^* counts the number of such graphs having exactly j connections. Note that $c_j^* \geq c_j$ for all j. We write n for the total number of intermediate nodes, i.e., $n = n_1 + n_2$. For a nonnegative integer n, we write $\lambda \vdash n$ if λ is an integer partition of n and we write $\lambda = (1^{a_1}, 2^{a_2}, \ldots, n^{a_n})$, where $a_j \geq 0$ stand for the number of occurrences of the integer j in the partition λ. Moreover, we write $\lambda_1 \cup \lambda_2$ for the union of the partitions λ_1 and λ_2 such that if $\lambda_1 = (1^{a_1}, 2^{a_2}, \ldots, n^{a_n})$ and $\lambda_2 = (1^{b_1}, 2^{b_2}, \ldots, n^{b_n})$ then $\lambda_1 \cup \lambda_2 = (1^{a_1+b_1}, 2^{a_2+b_2}, \ldots, n^{a_n+b_n})$.

Theorem 1. *The generating function for the total number of networks is*

$$N^*(m, n_1, n_2, k; x) = \sum_{\substack{\lambda_1 \vdash n_1 \\ \lambda_2 \vdash n_2}} \frac{\omega(\lambda_1)\omega(\lambda_2)}{n_1! n_2!} H(\lambda_1 \cup \lambda_2, m, k; x)$$

where

$$\omega(\lambda) = n! \prod_{1 \leq j \leq n} \frac{1}{j^{a_j} a_j!}$$

and

$$H(\lambda, m, k; x) = \prod_{a_j > 0} \left(1 + x^j\right)^{a_j(k+m+j-1)+j(a_j-1)} \prod_{\substack{j \geq 1 \\ \ell > j}} \left(1 + x^{\mathrm{lcm}(j,\ell)}\right)^{2j\ell a_j a_\ell / \mathrm{lcm}(j,\ell)}.$$

In Theorem 13, we obtain a recursive formula for $N(m, n_1, n_2, k; x)$ which we implemented in Python to obtain a list of values of $N(m, n_1, n_2, k)$ given in Table 1. This yields in particular the following striking example:

$$N(5, 5, 5, 5) = 108844524790336539487420588884391944954279893619583184.$$

We also obtain accurate estimates for both $N^*(m, n_1, n_2, k)$ and $N(m, n_1, n_2, k)$.

Theorem 2.

$$\begin{aligned} N^*(m, n_1, n_2, k) \;\; = \;\; & \frac{2^{(n+m+k-1)n}}{n_1! n_2!} \left(1 + \left(\binom{n_1}{2} + \binom{n_2}{2} \right) 2^{-2n-m-k+3} \right. \\ & + \left(2\binom{n_1}{3} + 2\binom{n_2}{3} \right) 2^{-4n-2m-2k+8} \\ & \left. + \left(\binom{n_1}{2}\binom{n_2}{2} + 3\binom{n_1}{4} + 3\binom{n_2}{4} \right) 2^{-4n-2m-2k+10} \right) \\ & + O\left(\frac{2^{(n+m+k-1)n}}{n_1! n_2!} n^7 2^{-6n-3m-3k} \right). \end{aligned}$$

Theorem 3.

$$\begin{aligned} N(m, n_1, n_2, k) \;\; = \;\; & \frac{1}{n_1! n_2!} 2^{(n+m+k-1)n} - \frac{n}{n_1! n_2!} 2^{(m+n+k-1)n-n-m+1} \\ & + \frac{n}{n_1! n_2!} 2^{(m+n+k-1)n-n-k+1} + O\left(\frac{2^{(n+m+k-1)n} n^3}{n_1! n_2!} \left(2^{-2n-\min(m,k)}\right) \right). \end{aligned}$$

The following asymptotic formulas immediately follow from Theorem 2 and Theorem 3.

Corollary 4. *As $n \to \infty$,*

$$N(m, n_1, n_2, k) \sim N^*(m, n_1, n_2, k) \sim \frac{2^{(m+n+k-1)n}}{n_1! n_2!}$$

and

$$N^*(m, n_1, n_2, k) - N(m, n_1, n_2, k) \sim \frac{n \left(2^{(m+n+k-1)n-m-n+1} + 2^{(m+n+k-1)n-k-n+1}\right)}{n_1! n_2!}.$$

In Section 4, we also provide an algorithm that can generate, for a given tuple (m, n_1, n_2, k), all connectivity matrices associated with the admissible networks.

2 Notation and preliminary results

2.1 Notation and the Pólya enumeration

Following the standard notation, we let S_n stand for the set of permutations of $1, 2, \ldots, n$. As in [4], we write $\lambda \vdash n$ if λ is an integer partition of n. We denote partitions by the Greek letters λ or μ and use the frequency representation $\lambda = (1^{a_1}, 2^{a_2}, \ldots)$ to mean that the integer j appears a_j times in the partition λ. For a given permutation $s \in S_n$ and a partition $\lambda \vdash n$, we say that s is of type λ ($\text{type}(s) = \lambda = (1^{a_1}, 2^{a_2}, \ldots)$) if s has precisely a_j disjoint cycles of length j. For example, if $n = 4$ and $s = (2, 1, 3, 4)$ then s can be written as the following product of disjoint cycles $(1, 2)(3)(4)$ so $\text{type}(s) = (1^2, 2^1)$. For a collection of partitions λ_j, we write a^j for the frequency vector associated with the permutation λ_j. We also let ϕ stand for an empty partition.

The Pólya enumeration theorem gives the generating function associated with the counting of graphs satisfying certain conditions by considering the orbits induced on the connections of the graph by the permutations of its nodes. If $H(x)$ is the generating function associated with the counting of graphs satisfying certain conditions (where the coefficient of x^n stands for the number of such graphs with n connections) and if $\mathcal{G}$ stands the group of permitted node permutations, we have

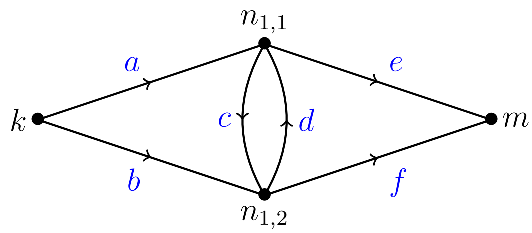
$$H(x) = \frac{1}{\#\mathcal{G}} \sum_{s \in \mathcal{G}} \prod_{j=1}^{\infty} (1 + x^j)^{\alpha(j,s)} \tag{1}$$

where $\alpha(j, s)$ is the number of disjoint orbits of length j induced on the connections of the graphs by the permutations of nodes s. The expression

$$\prod_{j=1}^{\infty} (1 + x^j)^{\alpha(j,s)}$$

can be thought as the generating function of *labelled* graphs left invariant by the permutation s. For example, if we consider the networks with 1 input node, 2 intermediate nodes of type 1 and 1 output node, the only admissible permutations are these of the two indistinguishable intermediate nodes. The admissible permutations are thus $s_1 = (1, 2)$ with $\text{type}(s_1) = (1^2)$ and $s_2 = (2, 1)$ with $\text{type}(s_2) = (2^1)$. There are six possible connections represented in the figure below. Given that s_1 leaves all six connections fixed (induces six disjoint orbits of length one), the term of the generating function associated with s_1 is $(1 + x)^6$. The permutation s_2 induces 3 orbits of length 2 on the connections of the network so the term of the generating function associated with s_2 is $(1 + x^2)^3$. This yields

$$N(1, 2, 0, 1; x) = \frac{1}{2} \left((1 + x)^6 + (1 + x^2)^3 \right).$$



Orbits induced by s_2

$$a \longrightarrow b \longrightarrow a, \quad c \longrightarrow d \longrightarrow c, \quad e \longrightarrow f \longrightarrow e.$$

In many cases, as in the present paper, the values of $\alpha(j, s)$ depend only on the permutation type $\lambda = \text{type}(s)$ so we can write

$$H(x) = \frac{1}{\#\mathcal{G}} \sum_{\lambda \vdash n} \#\{s \in \mathcal{G} : \text{type}(s) = \lambda\} \prod_{j=1}^{\infty} (1 + x^j)^{\alpha(j,\lambda)}. \tag{2}$$

In order to compute $\#\{s \in \mathcal{G} : \text{type}(s) = \lambda\}$, we will make use of the following result:

Lemma 5. *For a positive integer n and a partition type $\lambda \vdash n$, if we define*

$$\omega(\lambda) := \#\{s \in S_n : \text{type}(s) = \lambda\}$$

then

$$\omega(\lambda) = n! \prod_{1 \le j \le n} \frac{1}{j^{a_j} a_j!}.$$

For example, if $n = 4$, the number of permutations of the type $(1^2, 2^1)$ is

$$\frac{4!}{1^2 2! 2^1 1!} = 6.$$

Proof. An elementary proof of this classical result can be found in [14]. □

For two permutations s_1 and s_2 acting on disjoint sets of nodes, we write $s_1 \cdot s_2$ as the usual permutation product. Provided that $\text{type}(s_1) = \lambda_1$ and that $\text{type}(s_2) = \lambda_2$, we also define

$$\lambda_1 \cup \lambda_2 = \text{type}(s_1 \cdot s_2). \tag{3}$$

This is well defined since $\lambda_1 \cup \lambda_2$ doesn't depend on the particular choices of s_1 and s_2. For example, if $\lambda_1 = (1^1, 2^1, 3^2)$ and $\lambda_2 = (1^1, 4^1)$, we have $\lambda_1 \cup \lambda_2 = (1^2, 2, 3^2, 4)$. For an integer $N \ge 2$, we generalize this notation to

$$\lambda_1 \cup \lambda_2 \cup \cdots \cup \lambda_{N+1} = (\lambda_1 \cup \lambda_2 \cup \cdots \cup \lambda_N) \cup \lambda_{N+1}.$$

For two partitions λ_1, λ_2 and any permutation s such that $\text{type}(s) = \lambda_1 \cup \lambda_2$ we define

$$\Omega(\lambda_1, \lambda_2) := \#\{(s_1, s_2) : \text{type}(s_1) = \lambda_1, \text{type}(s_2) = \lambda_2, s_1 \cdot s_2 = s\}.$$

For example, suppose that $\lambda_1 = (1, 2)$ and $\lambda_2 = (1)$ so that $\lambda = (1^2, 2)$. A canonical choice of s (written as a product of disjoint cycles) is $s = (1)(2)(3, 4)$. The possible choices for the pair (s_1, s_2) are $s_1 = (1)(34)$ and $s_2 = (2)$ as well as $s_1 = (2)(34)$ and $s_2 = (1)$. We thus have $\Omega(\lambda_1, \lambda_2) = 2$. For partitions $\lambda_1, \lambda_2, \ldots, \lambda_N$ we recursively extend the definition of Ω to

$$\begin{aligned} & \Omega(\lambda_1, \lambda_2, \ldots, \lambda_N) \\ = \; & \Omega(\lambda_1 \cup \cdots \cup \lambda_{N-1}, \lambda_N)\Omega(\lambda_1 \cup \cdots \cup \lambda_{N-2}, \lambda_{N-1}). \end{aligned}$$

We can obtain the following explicit expression for $\Omega(\lambda_1, \ldots, \lambda_N)$

Lemma 6.

$$\Omega(\lambda_1, \lambda_2, \ldots, \lambda_N) = \prod_j a_j! \prod_{y=1}^{N} \frac{1}{a_j^y!}$$

where a is the frequency vector of $\lambda_1 \cup \lambda_1 \cup \cdots \cup \lambda_N$ and the a^ys are the frequency vectors of λ_y.

Proof. Lemma 6 follows directly from the observation that the a_j cycles of length j have to be distributed between N permutations such that the yth permutation receives exactly a_j^y of these cycles. The number of ways to achieve this is exactly

$$a_j! \prod_{y=1}^{N} \frac{1}{a_j^y!}.$$

□

Remark 7. It is implicit in the definition of Ω that the value of $\Omega(\lambda_1 \cup \lambda_2)$ doesn't depend on the particular choice of the permutation s of type $\lambda_1 \cup \lambda_2$ which indeed follows from Lemma 6 and its proof.

2.2 Auxiliary functions

We derive the components of the generating function $N(m, n_1, n_2, k; x)$ associated with the number of networks counted by $N(m, n_1, n_2, k)$. We let $F(\lambda; x)$ stand for the action of a permutation s of type λ on connections between two nodes permuted by s. In other words, for a permutation s of type λ, $F(\lambda; x)$ is the generating function of *labelled* graphs left invariant by this permutation. We have

Lemma 8.

$$F(\lambda; x) \;=\; \prod_{a_j > 0} \left(1 + x^j\right)^{a_j(j-1) + ja_j(a_j-1)} \prod_{j \geq 1} \prod_{y > j} \left(1 + x^{\text{lcm}(j,y)}\right)^{2jya_ja_y/\,\text{lcm}(j,y)}.$$

Proof. The expression

$$\prod_{a_j>0}\left(1+x^j\right)^{a_j(j-1)}$$

corresponds to the action of a permutation of type λ on edges between two nodes in the same cycle. Indeed, since we are considering oriented connections, the length of the orbit of the connections induced by a node cycle of length j is also j. Given that there are $j(j-1)$ possible connections between nodes permuted by the same cycle, the number of orbits is $j(j-1)/j = j-1$. Since there are a_j ways of choosing a cycle of length j, the exponent must be multiplied by a_j.

On the other hand, the expression

$$\prod_{a_j>0}\left(1+x^j\right)^{ja_j(a_j-1)}$$

corresponds to the action of a permutation of type λ on connections between nodes permuted by different cycles of the same length. Indeed, the length of the orbits induced on the connections is the same as the length of the node cycle: j. For two disjoint cycles of length j, there are $2j^2$ possible connections between a node of the first cycle and one of the second. The number of disjoint connection orbits is thus $2j^2/j = 2j$. Furthermore the number of ways of choosing two node cycles of length j is $\binom{a_j}{2}$ so the exponent is equal to

$$\frac{a_j(a_j-1)}{2}2j = ja_j(a_j-1).$$

Finally, the expression

$$\prod_{j\geq 1}\prod_{y>j}\left(1+x^{\operatorname{lcm}(j,y)}\right)^{2jya_ja_y/\operatorname{lcm}(j,y)}$$

corresponds to the action of a permutation of type λ on connections between two nodes belonging to different cycles of different lengths. Indeed, on connections between a node belonging to a cycle of length j and a node belonging to a cycle of length y, the node permutation will induce orbits of length

$$\operatorname{lcm}(j,y).$$

Given that the number of such possible connections is $2jy$, the number of such orbits is equal to

$$2jy/\operatorname{lcm}(j,y).$$

Moreover, the number of ways of choosing a node cycle of length j and a node cycle of length y is a_ja_y so the exponent is

$$2jya_ja_y/\operatorname{lcm}(j,y).$$

□

If s_1 and s_2 are permutations respectively of type λ_1 and λ_2 acting on disjoint sets of nodes, we define $D(\lambda_1|\lambda_2;x)$ as the generating function counting the number of *labelled* graphs left invariant by $s_1 \cdot s_2$ under the following restriction: Only connections from nodes permuted by s_1 to nodes permuted by s_2 are permitted. Observe that by symmetry, it follows from this definition that $D(\lambda_1|\lambda_2;x) = D(\lambda_2|\lambda_1;x)$. We will use the following result.

Lemma 9.

$$D(\lambda_1|\lambda_2;x) = \prod_{j\geq 1}\prod_{y\geq 1}\left(1+x^{\operatorname{lcm}(j,y)}\right)^{jya_j^1 a_y^2/\operatorname{lcm}(j,y)}.$$

Proof. Cycles of length j and y will induce orbits of length $\operatorname{lcm}(j,y)$ on connections from nodes permuted by the cycle of length j to nodes permuted by the cycle of length y. Given that we only allow connections in one direction, the total number of possible connections is jy. It follows that the number of disjoint orbits is $jy/\operatorname{lcm}(j,y)$. Furthermore, the number of ways of choosing a node cycle of length j in a permutation of type λ_1 and a node cycle of length y in a permutation of type λ_2 is $a_j^1 a_y^2$ from which Lemma 9 follows. □

If s_1 and s_2 (respectively of type λ_1 and λ_2) are two permutations acting on disjoint sets of nodes, we define $I(\lambda_1|\lambda_2;x)$ as the generating function of the number of *labelled* networks left invariant by $s_1 \cdot s_2$ under the following restrictions: 1. Only connections from nodes permuted by s_1 to nodes permuted by s_2 are permitted. 2. Each node permuted by s_1 must send at least one connection to a node permuted by s_2. We have the following explicit expression for $I(\lambda_1|\lambda_2;x)$.

Lemma 10.

$$I(\lambda_1|\lambda_2;x) = \prod_{a_j^1>0}\left(\left(\prod_{a_y^2>0}\left(1+x^{\operatorname{lcm}(j,y)}\right)^{jya_y^2/\operatorname{lcm}(j,y)}\right)-1\right)^{a_j^1}.$$

Proof. Let c_j be a node cycle of length j of s_1 and let s_2 be a permutation of type λ_2 acting on a disjoint set of nodes. The action of $s_2 \cdot c_j$ on connections to nodes permuted by s_2 from nodes permuted by c_j is given by

$$\prod_{a_y^2>0}\left(1+x^{\operatorname{lcm}(j,y)}\right)^{jya_y^2/\operatorname{lcm}(j,y)}. \tag{4}$$

Since we impose that at least one connection actually exists, we must remove from (4) the instance of no connection which corresponds to 1 in the generating function. We thus obtain $I(\lambda_1|\lambda_2;x)$ by performing the product

$$\prod_{a_y>0}\left(1+x^{\operatorname{lcm}(j,y)}\right)^{jya_y^2/\operatorname{lcm}(j,y)} - 1$$

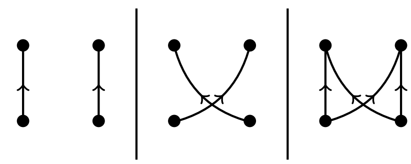
over all cycles of a permutation s_1 of type λ_1 which proves Lemma 10.

□

Observe that by symmetry $I(\lambda_1|\lambda_2;x)$ is also the generating function associated with the number of networks such that: 1. Only connections to nodes permuted by s_1 from nodes permuted by s_2 are permitted. 2. Each node permuted by s_1 must receive at least one connection from a node permuted by s_2. For illustration, if $\lambda_1 = (2^1)$ and $\lambda_2 = (2^1)$, then

$$I(\lambda_1|\lambda_2;x) = \left(1-x^2\right)^2 - 1 = 2x^2 + x^4$$

and the corresponding networks are:



For a permutation type s_1 of type λ_1 and a permutation s_2 of type λ_2 acting on disjoint sets of nodes, we define $T(\lambda_1|\lambda_2;x)$ as the generating function counting the number of *labelled* networks left invariant by $s_1 \cdot s_2$ and satisfying the following properties: 1. The only permitted connections are these between two nodes permuted by s_1 or from a node permuted by s_1 to a node permuted by s_2. 2. Every node permuted by s_1 sends a path to at least a node permuted by s_2. We obtain the following recursive relation for $T(\lambda_1|\lambda_2;x)$

Lemma 11. *The following relation holds:*

$$T(\lambda_1|\lambda_2;x) = I(\lambda_1|\lambda_2;x)F(\lambda_1;x) + \sum_{\substack{\mu_1\cup\mu_2=\lambda_1\\ \mu_1\neq\phi,\mu_2\neq\phi}} \Omega(\mu_1,\mu_2)I(\mu_1|\lambda_2;x)F(\mu_1;x)D(\mu_1|\mu_2;x)T(\mu_2|\mu_1;x),$$

which allows us to compute $T(\lambda_1|\lambda_2;x)$ recursively.

Proof. The term $I(\lambda_1|\lambda_2;x)F(\lambda_1;x)$ corresponds to the situation in which all nodes permuted by s_1 (of type λ_1) send a direct connection to at least a node permuted by s_2 (of type λ_2). Otherwise, we write $s_1 = \ell_1 \cdot \ell_2$ with $\text{type}(\ell_1) = \mu_1$ and $\text{type}(\ell_2) = \mu_2$ where ℓ_1 permutes the nodes sending a *direct* connection to at least a node permuted by s_2. The factor $T(\mu_2|\mu_1;x)$ then imposes that every node permuted by ℓ_2 sends a path to a node permuted by s_2 through a node permuted by ℓ_1. The factor $D(\mu_1|\mu_2;x)$ accounts for the possible connections from nodes permuted by ℓ_1 to the nodes permuted by ℓ_2.

□

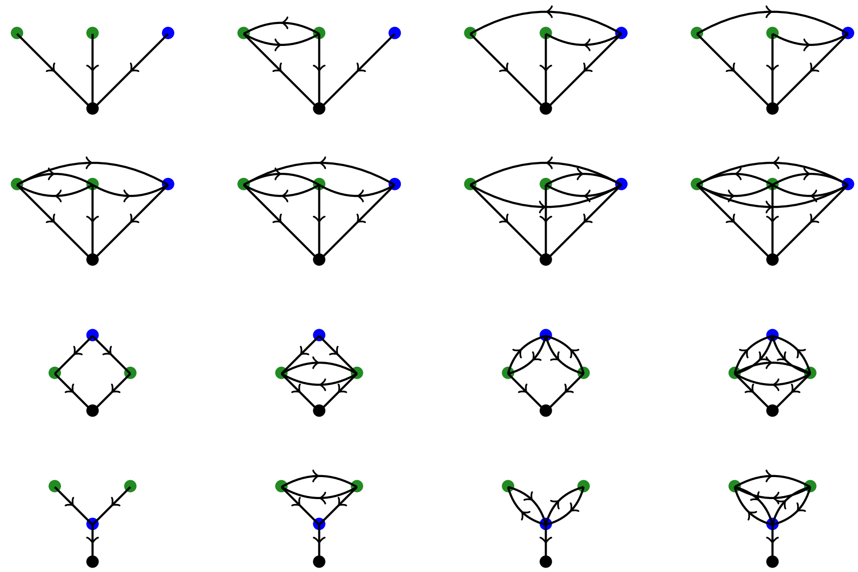
For illustration, we assume that $\lambda_1 = (1^1, 2^1)$ and that $\lambda_2 = (1^1)$. We then have

$$\begin{aligned} T(\lambda_1|\lambda_2;x) \quad = \quad & I\left((1^1,2^1)|(1^1);x\right)F((1^1,2^1);x) \\ & + I\left((2^1)|(1^1);x\right)F\left((2^1);x\right)D\left((2^1)|(1^1);x\right)T\left((1^1)|(2^1);x\right) \\ & + I\left((1^1)|(1^1);x\right)F\left((1^1);x\right)D\left((1^1)|(2^1);x\right)T\left((2^1)|(1^1);x\right). \end{aligned}$$

Using $T\left((1^1)|(2^1);x\right) = I\left((1^1)|(2^1);x\right)F\left((1^1);x\right)$ and $T\left((2^1)|(1^1);x\right) = I\left((2^1)|(1^1);x\right)F\left((2^1);x\right)$ we obtain

$$\begin{aligned}
T(\lambda_1|\lambda_2;x) &= x^3\left(1+x^2\right)^3 \\
&\quad +x^2\left(1+x^2\right)(1+x^2)x^2 + x(1+x^2)x^2\left(1+x^2\right) \\
&= x^9+x^8+4x^7+2x^6+5x^5+x^4+2x^3.
\end{aligned}$$

We draw the corresponding networks under the convention that the node on which s_2 acts is in black while the nodes on the 2-cycle of s_1 are in green and the node left fixed by s_1 is in blue.



3 Main results

We now demonstrate the main results of the article, namely: **1)** An exact expression for the total number of networks **2)** A recursive formula yielding the exact value of $N(m,n_1,n_2,k;x)$, **3)** Sharp analytic bounds for $N^*(m,n_1,n_2,k)$, **4)** Sharp analytic bounds for $N(m,n_1,n_2,k)$.

3.1 Exact series

3.1.1 Total number of networks

Theorem 12.

$$N^*(m,n_1,n_2,k;x) = \sum_{\substack{\lambda_1\vdash n_1\\ \lambda_2\vdash n_2}} \frac{\omega(\lambda_1)\omega(\lambda_2)}{n_1!n_2!}G(\lambda_1\cup\lambda_2;x)$$

with

$$G(\lambda;x) = F(\lambda;x)D(\lambda|(1^m);x)D(\lambda|(1^k);x).$$

Proof. Since the intermediate nodes of type I are distinguishable from these of type II, we only consider partitions $\lambda_1 \cup \lambda_2$ with $\lambda_1 \vdash n_1$ and $\lambda_2 \vdash n_2$. In the expression $G(\lambda; x)$, the factor $F(\lambda; x)$ stands for the action of a permutation of type λ on connections between intermediate nodes. The factor $D(\lambda|(1^m); x)$ stands for the action of a permutation of type λ on the connections from the m input nodes to the intermediate nodes. Finally, the factor $D(\lambda|(1^k); x)$ stands for the action of a permutation of type λ on the connections from the intermediate nodes to the k output nodes. $\square$

3.1.2 Number of admissible networks

The following result gives an effective recursive formula for $N(m, n_1, n_2, k; x)$ which we implemented in Python to obtain the values of $N(m, n_1, n_2, k) = N(m, n_1, n_2, k; 1)$ given in Table 1.

Theorem 13.

$$N(m, n_1, n_2, k; x) = \sum_{\substack{\lambda_1 \vdash n_1 \\ \lambda_2 \vdash n_2}} \frac{\omega(\lambda_1)\omega(\lambda_2)}{n_1! n_2!} C(\lambda_1 \cup \lambda_2; x) \tag{5}$$

where $C(\lambda; x)$ can be obtained recursively through the following equation

$$\begin{aligned} C(\lambda; x) \;&=\; G(\lambda; x) - \sum_{\substack{\mu_1 \cup \mu_2 \cup \mu_3 \cup \mu_4 = \lambda \\ \mu_2 \cup \mu_3 \cup \mu_4 \neq \phi}} \Omega\left(\mu_1, \mu_2, \mu_3, \mu_4\right) C(\mu_1; x) \\ &\quad T(\mu_2|\mu_1 \cup (1^m); x) T(\mu_3|\mu_1 \cup (1^k); x) F(\mu_4; x) D(\mu_2 \cup \mu_4|\mu_3; x) D(\mu_2|\mu_4; x) \end{aligned} \tag{6}$$

with $C(\phi; x) = 1$.

Proof. Theorem 13 follows rather directly from the definitions of the auxiliary functions. The intermediate nodes of a network can be divided in four disjoint groups (regardless if they are of type I or type II) as follows:

1. Intermediate nodes of the first group, tagged as 'connected' intermediate nodes, both receive a path from an input node and send a path to an output neuron. Other intermediate nodes are tagged as 'unconnected'.

2. Unconnected intermediate nodes in the second group receive a path from an input node but send none to an output node.

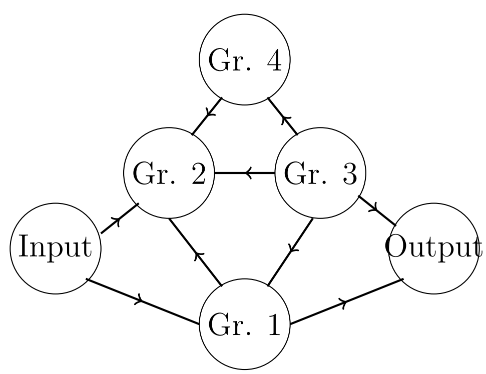
3. Unconnected intermediate nodes of the third group send a path to an output node but receive no path from an input node.

4. Unconnected intermediate nodes of the fourth group neither receive a path from an input node nor send a path to an output node.

We write μ_j, $1 \le j \le 4$ as the types of the permutations acting on the intermediate nodes belonging to group j. Assuming that a network is not counted by $N(m, n_1, n_2, k)$ then $\mu_2 \cup \mu_3 \cup \mu_4 \ne \phi$ since at least one intermediate node doesn't belong to the first group. Equation (6) subtracts from all possible networks these in which some intermediate nodes are unconnected.

The factor $C(\mu_1; x)$ corresponds to the action of a permutation of type μ_1 on the connections between two 'connected' intermediate nodes as well as on connections between connected intermediate nodes and input/output nodes. The factor $T(\mu_2|\mu_1 \cup (1^m); x)$ accounts for the action of permutations types μ_1 and μ_2 on the connections between two intermediate nodes of the second group as well as on connections from a connected intermediate node or from an input node to intermediate nodes of the second group. The factor $T(\mu_3|\mu_1 \cup (1^k); x)$ accounts for the action of permutations of types μ_1 and μ_3 on connections between two intermediate nodes of group 3 as well as on connections from intermediate nodes of group 3 to connected intermediate nodes or to output nodes. The factor $F(\mu_4; x)$ accounts for the action of the permutation type μ_4 on connections between two intermediate nodes of group 4. The factor $D(\mu_2 \cup \mu_4|\mu_3; x)$ accounts for the action of permutation types μ_2, μ_3 and μ_4 on connections from intermediate nodes of group 3 to intermediate nodes of either group 2 or group 4. Finally, the factor $D(\mu_2|\mu_4; x)$ accounts for the action of permutation types μ_2 and μ_4 on connections from an intermediate node of group 4 to an intermediate node of group 2. Given that in formula (6), the partition μ_1 is a partition of an integer which is strictly smaller than the integer partitioned by λ, the recurrence process is convergent. The argument of the proof is illustrated in the schematic below.

□



3.2 Asymptotic series

3.2.1 Total number of networks

Asymptotic formulas for graph enumeration have been obtained for several graph enumeration problems including [15]. In this direction, Theorem 12 allows us to derive the following approximation for $N^*(m, n_1, n_2, k)$.

Theorem 14.

$$\begin{aligned}N^*(m,n_1,n_2,k) &= \frac{2^{(n+m+k-1)n}}{n_1!n_2!}\Bigg(1+\left(\binom{n_1}{2}+\binom{n_2}{2}\right)2^{-2n-m-k+3}\\ &+\left(2\binom{n_1}{3}+2\binom{n_2}{3}\right)2^{-4n-2m-2k+8}\\ &+\left(\binom{n_1}{2}\binom{n_2}{2}+3\binom{n_1}{4}+3\binom{n_2}{4}\right)2^{-4n-2m-2k+10}\Bigg)\\ &+O\left(\frac{2^{(n+m+k-1)n}}{n_1!n_2!}n^7 2^{-6n-3m-3k}\right).\end{aligned}$$

Proof. The proof is technical but rather straightforward. The idea behind it is that the main contributions to $N^*(m,n_1,n_2,k)$ come from the permutations leaving all or almost all intermediate nodes fixed. We will quantify exactly these contributions and show that the others can be neglected. The right hand side in the statement of Theorem 14 could be further expanded, but we limited ourselves to four terms for the sake of simplicity. If we define the partition λ_1 by $\lambda_1 := (1^n)$, we have

$$\omega(\lambda_1)=1 \text{ and } G(\lambda_1;x)=(1+x)^{(n+m+k-1)n} \tag{7}$$

so the contribution of the identity permutation to $N^*(m,n_1,n_2,k)$ is equal to $\frac{2^{(m+n+k-1)n}}{n_1!n_2!}$. We will now show that the contributions of other permutation types are small compared to this.

Let $\lambda_2=(1^{n-2},2^1)$, we have

$$\sum_{\substack{\mu_1\vdash n_1,\mu_2\vdash n_2\\ \mu_1\cup\mu_2=\lambda_2}}\omega(\mu_1)\omega(\mu_2)=\binom{n_1}{2}+\binom{n_2}{2} \tag{8}$$

$$\text{and}\qquad G(\lambda_2;x)=(1+x)^{(n+m+k-3)(n-2)}(1+x^2)^{1+2(n-2)+m+k}.$$

Let $\lambda_3=(1^{n-3},3^1)$, we have

$$\sum_{\substack{\mu_1\vdash n_1,\mu_2\vdash n_2\\ \mu_1\cup\mu_2=\lambda_3}}\omega(\mu_1)\omega(\mu_2)=2\binom{n_1}{3}+2\binom{n_2}{3} \tag{9}$$

$$\text{and}\qquad G(\lambda_3;x)=(1+x)^{(n+m+k-4)(n-3)}(1+x^3)^{2+2(n-3)+m+k}.$$

Let $\lambda_4=(1^{n-4},2^2)$, we have

$$\sum_{\substack{\mu_1\vdash n_1,\mu_2\vdash n_2\\ \mu_1\cup\mu_2=\lambda_4}}\omega(\mu_1)\omega(\mu_2)=\binom{n_1}{2}\binom{n_2}{2}+3\binom{n_1}{4}+3\binom{n_2}{4} \tag{10}$$

$$\text{and}\qquad G(\lambda_4;x)=(1+x)^{(n+m+k-5)(n-4)}(1+x^2)^{6+4(n-4)+2m+2k}.$$

Let $\lambda_5 = (1^{n-4}, 4^1)$, we have

$$\sum_{\substack{\mu_1 \vdash n_1, \mu_2 \vdash n_2 \\ \mu_1 \cup \mu_2 = \lambda_5}} \omega(\mu_1)\omega(\mu_2) = 6\binom{n_1}{4} + 6\binom{n_2}{4} \tag{11}$$

$$\text{and} \qquad G(\lambda_5; x) = (1+x)^{(n+m+k-5)(n-4)}(1+x^4)^{3+2(n-4)+m+k}.$$

Let $\lambda_6 = (1^{n-5}, 2, 3)$, we have

$$\begin{aligned} \sum_{\substack{\mu_1 \vdash n_1, \mu_2 \vdash n_2 \\ \mu_1 \cup \mu_2 = \lambda_6}} \omega(\mu_1)\omega(\mu_2) &= 2\binom{n_1}{2,3} + 2\binom{n_2}{2,3} + 2\binom{n_1}{2}\binom{n_2}{3} + 2\binom{n_1}{3}\binom{n_2}{2} \\ G(\lambda_6; x) &= (1+x)^{(n+m+k-6)(n-5)}(1+x^2)^{1+2(n-5)+m+k}(1+x^3)^{2+2(n-5)+m+k} \\ &\quad \times (1+x^6)^2. \end{aligned} \tag{12}$$

Finally, let $\lambda_7 = (1^{n-5}, 5)$. We have

$$\sum_{\substack{\mu_1 \vdash n_1, \mu_2 \vdash n_2 \\ \mu_1 \cup \mu_2 = \lambda_7}} \omega(\mu_1)\omega(\mu_2) = 24\binom{n_1}{5} + 24\binom{n_2}{5} \tag{13}$$

$$\text{and} \qquad G(\lambda_7; x) = (1+x)^{(n+m+k-6)(n-5)}(1+x^5)^{4+2(n-5)+m+k}.$$

For h a nonnegative integer, we define the set $\gamma(h)$ as

$$\gamma(h) := \{\lambda : \exists \mu_1 \vdash n_1, \mu_2 \vdash n_2, \mu_1 \cup \mu_2 = \lambda, a_1^\lambda = n - h\}$$

that is the set of partitions of n with exactly $n-h$ cycles of length 1. From equation (7) we have

$$\sum_{\substack{\lambda_1 \vdash n_1, \lambda_2 \vdash n_2 \\ \lambda_1 \cup \lambda_2 \in \gamma(0)}} \frac{\omega(\lambda_1)\omega(\lambda_2)}{n_1! n_2!} G(\lambda_1 \cup \lambda_2; 1) = \frac{1}{n_1! n_2!} 2^{(n+m+k-1)n}. \tag{14}$$

From equation (8) we have

$$\sum_{\substack{\lambda_1 \vdash n_1, \lambda_2 \vdash n_2 \\ \lambda_1 \cup \lambda_2 \in \gamma(2)}} \frac{\omega(\lambda_1)\omega(\lambda_2)}{n_1! n_2!} G(\lambda_1 \cup \lambda_2; 1) = \frac{\binom{n_1}{2} + \binom{n_2}{2}}{n_1! n_2!} 2^{(n+m+k-1)n-2n-m-k+3}. \tag{15}$$

From equation (9) we have

$$\sum_{\substack{\lambda_1 \vdash n_1, \lambda_2 \vdash n_2 \\ \lambda_1 \cup \lambda_2 \in \gamma(3)}} \frac{\omega(\lambda_1)\omega(\lambda_2)}{n_1! n_2!} G(\lambda_1 \cup \lambda_2; 1) = \frac{2\binom{n_1}{3} + 2\binom{n_2}{3}}{n_1! n_2!} 2^{(n+m+k-1)n-4n-2m-2k+8}. \tag{16}$$

From equations (10) and (11) we have

$$
\begin{aligned}
&\sum_{\substack{\lambda_1\vdash n_1,\lambda_2\vdash n_2\\ \lambda_1\cup\lambda_2\in\gamma(4)}} \frac{\omega(\lambda_1)\omega(\lambda_2)}{n_1!n_2!}G(\lambda_1\cup\lambda_2;1)\\
=\;& \frac{\binom{n_1}{2}\binom{n_2}{2}+3\binom{n_1}{4}+3\binom{n_2}{4}}{n_1!n_2!}2^{(n+m+k-1)n-4n-2m-2k+10}\\
&+\frac{6\binom{n_1}{4}+6\binom{n_2}{4}}{n_1!n_2!}2^{(n+m+k-1)n-6n-3m-3k+15}. \qquad (17)
\end{aligned}
$$

While finally from equations (12) and (13) we have:

$$
\begin{aligned}
&\sum_{\substack{\lambda_1\vdash n_1,\lambda_2\vdash n_2\\ \lambda_1\cup\lambda_2\in\gamma(5)}} \frac{\omega(\lambda_1)\omega(\lambda_2)}{n_1!n_2!}G(\lambda_1\cup\lambda_2;1)\\
=\;& \frac{2\binom{n_1}{2,3}+2\binom{n_2}{2,3}+2\binom{n_1}{2}\binom{n_2}{3}+2\binom{n_1}{3}\binom{n_2}{2}}{n_1!n_2!}2^{(n+m+k-1)n-6n-3m-3k+13}\\
&+\frac{24\binom{n_1}{5}+24\binom{n_2}{5}}{n_1!n_2!}2^{(n+m+k-1)n-8n-4m-4k+24}. \qquad (18)
\end{aligned}
$$

From equations (14-18) we obtain

$$
\begin{aligned}
N^*(m,n_1,n_2,k) \;=\;& \frac{2^{(n+m+k-1)n}}{n_1!n_2!}\Bigg(1+\left(\binom{n_1}{2}+\binom{n_2}{2}\right)2^{-2n-m-k+3}\\
&+\left(2\binom{n_1}{3}+2\binom{n_2}{3}\right)2^{-4n-2m-2k+8}\\
&+\left(\binom{n_1}{2}\binom{n_2}{2}+3\binom{n_1}{4}+3\binom{n_2}{4}\right)2^{-4n-2m-2k+10}\Bigg)\\
&+O\left(\frac{2^{(n+m+k-1)n}}{n_1!n_2!}n^5 2^{-6n-3m-3k}\right)\\
&+\sum_{h=6}^{n}\sum_{\substack{\lambda_1\vdash n_1,\lambda_2\vdash n_2\\ \lambda_1\cup\lambda\in\gamma(h)}} \frac{\omega(\lambda_1)\omega(\lambda_2)}{n_1!n_2!}G(\lambda_1\cup\lambda_2;1). \qquad (19)
\end{aligned}
$$

Assuming that a permutation s is of type $\lambda\in\gamma(h)$, we can count the number of connections that will not be fixed by s. We have that $h(m+k)$ connections between intermediate nodes and input/output nodes will have orbits of length greater than one while $h(h-1)$ connections between two intermediate nodes which are not fixed by s will have orbits of length greater than one. Finally, $2h(n-h)$ connections between an intermediate node that is fixed by s and one that is not fixed by s will be in orbits of length greater than one. The total number of connections in orbits of length greater than one is thus equal to

$$h(m+k)+h(h-1)+2h(n-h).$$

Given that the total number of connections is equal to $(n+m+k-1)n$, the number of connection orbits induced by a permutation of type $\lambda \in \gamma(h)$ will thus be less or equal than

$$(n+m+k-1)n - \frac{h(m+k)+h(h-1)+2h(n-h)}{2}.$$

Therefore

$$\lambda \in \gamma(h) \Rightarrow G(\lambda;1) \leq 2^{(m+n+k-1)n-h(m+k+2n-h-1)/2}. \tag{20}$$

We also have

$$\sum_{\substack{\lambda_1 \vdash n_1, \lambda_2 \vdash n_2 \\ \lambda_1 \cup \lambda_2 \in \gamma(h)}} \omega(\lambda_1)\omega(\lambda_2) \leq \sum_{h_1=0}^{\min(n_1,h)} \binom{n_1}{h_1}\binom{n_2}{h-h_1} h_1!(h-h_1)! \leq \sum_{h_1=0}^{\min(n_1,h)} n^h \leq n^{h+1}. \tag{21}$$

Equations (20) and (21) imply

$$\begin{aligned}
\sum_{h=6}^{n} \sum_{\substack{\lambda_1 \vdash n_1, \lambda_2 \vdash n_2 \\ \lambda_1 \cup \lambda_2 \in \gamma(h)}} \frac{\omega(\lambda_1)\omega(\lambda_2)}{n_1!n_2!} G(\lambda_1 \cup \lambda_2; 1) \quad &\leq \quad \frac{2^{(m+n+k-1)n}}{n_1!n_2!} \sum_{h=6}^{n} n^{h+1} 2^{-h(m+k+2n-h-1)/2} \\
&\ll \quad \frac{2^{(m+n+k-1)n}}{n_1!n_2!} \sum_{h=6}^{n} n^{h+1} 2^{-h(m+k+n)/2} \qquad (22) \\
&= \quad O\left(\frac{2^{(n+m+k-1)n}}{n_1!n_2!} n^7 2^{-6n-3m-3k}\right).
\end{aligned}$$

And the proof of Theorem 14 follows from (19) and (22). □

3.2.2 Number of admissible networks

Probabilistic interpretation shows that assuming that each potential connection is equally likely to exist or not, almost all graphs are connected as the number of nodes increases [9]. The next result shows that a similar phenomenon occurs in the context of the problem investigated in the present paper as $N(m,n_1,n_2,k) \sim N^*(m,n_1,n_2,k)$ when the number of nodes increases.

Theorem 15. *Assuming that* $\min(m,k) > 1$, *we have*

$$\begin{aligned}
N(m,n_1,n_2,k) \quad &= \quad \frac{1}{n_1!n_2!} 2^{(n+m+k-1)n} - \frac{n}{n_1!n_2!} 2^{(m+n+k-1)n-n-m+1} \\
&\quad - \frac{n}{n_1!n_2!} 2^{(m+n+k-1)n-n-k+1} + O\left(\frac{2^{(n+m+k-1)n} n^3}{n_1!n_2!} \left(2^{-2n-\min(m,k)}\right)\right).
\end{aligned}$$

Proof. We will now provide a proof of Theorem 15 following the same general idea as in the proof of Theorem 14 namely that the contribution to $N(m, n_1, n_2, k; x)$ from permutations other than the identity permutation are relatively small. Furthermore, when subtracting the networks not satisfying the connectivity conditions, the main contribution to the subtracted quantity comes from the case where only one intermediate node belongs to the unconnected group. From equations (20) and (21) we have:

$$\begin{aligned}
\sum_{h\geq 2}\sum_{\substack{\lambda_1\vdash n_1,\lambda_2\vdash n_2\\ \lambda_1\cup\lambda_2\in\gamma(h)}} \frac{\omega(\lambda_1)\omega(\lambda_2)}{n_1!n_2!}C(\lambda_1\cup\lambda_2;1) &\leq \sum_{h\geq 2}\sum_{\substack{\lambda_1\vdash n_1,\lambda_2\vdash n_2\\ \lambda_1\cup\lambda_2\in\gamma(h)}} \frac{\omega(\lambda_1)\omega(\lambda_2)}{n_1!n_2!}G(\lambda_1\cup\lambda_2;1) \\
&\leq \frac{2^{(m+n+n+k-1)n}}{n_1!n_2!}\sum_{h=2}^{n} n^{h+1}2^{-h(m+k+2n-h-1)/2} \\
&= O\left(\frac{2^{(n+m+k-1)n}}{n_1!n_2!}n^3 2^{-2n-m-k}\right). \qquad (23)
\end{aligned}$$

This means that for the purpose of Theorem 15, all permutations except the identity can be neglected. For the remainder of the proof, we will thus assume $\lambda = (1^n)$. We have from Theorem 13

$$\begin{aligned}
C((1^n);x) &= G((1^n);x) \qquad (24)\\
&- \sum_{h=1}^{n}\sum_{\substack{\mu_1\cup\mu_2\cup\mu_3\cup\mu_4=(1^n)\\ \mu_2\cup\mu_3\cup\mu_3=(1^h)}} C(\mu_1;x)\Omega(\mu_1,\mu_2,\mu_3,\mu_4)U((\mu_2,\mu_3,\mu_4),\mu_1;x)
\end{aligned}$$

where the sum on h is performed over all the possible number of unconnected intermediate nodes and

$$U((\mu_2,\mu_3,\mu_4),\mu_1;x) =$$
$$T(\mu_2|\mu_1\cup(1^m);x)T(\mu_3|\mu_1\cup(1^k);x)F(\mu_4;x)D(\mu_2\cup\mu_4|\mu_3;x)D(\mu_2|\mu_4;x).$$

We now show that the contribution of instances with more than 1 unconnected intermediate node is relatively small. Assuming that $\mu_2\cup\mu_3\cup\mu_4 = (1^h)$ we have that $\mu_1 = (1^{n-h})$. The number of forbidden connections is at least

$$h((n-h)+\min(m,k)).$$

Therefore

$$C(\mu_1;1)U((\mu_2,\mu_3,\mu_4),\mu_1;1) \leq 2^{(n+m+k-1)n}2^{-h((n-h)+\min(m,k))}. \qquad (25)$$

The number of ways of choosing the h unconnected intermediate nodes is equal to $\binom{n}{h}$. The number of ways of choosing which of these unconnected intermediate nodes will not send a path to an output neuron is equal to 2^h. We thus have

$$\sum_{\substack{\mu_1\cup\mu_2\cup\mu_3\cup\mu_4=(1^n)\\ \mu_2\cup\mu_2\cup\mu_3=(1^h)}} C((1^{n-h});1)\Omega(\mu_1,\mu_2,\mu_3,\mu_4)U((\mu_2,\mu_3,\mu_4),(1^{n-h})) \tag{26}$$
$$\leq \; 2^{(n+m+k-1)n}\binom{n}{h}2^h 2^{-h((n-h)+\min(m,k))}.$$

Given that under the assumption that $\min(m,k)>1$ we have

$$\sum_{h=2}^{n}\binom{n}{h}2^{-h((n-h)+\min(m,k)-1)} = O\left(n^2 2^{-2n-2\min(m,k)}\right)$$

we can obtain the following from equations (24) and (26)

$$\begin{aligned} C((1^n);1) \;=\;& 2^{(n+m+k-1)n} \\ &- \sum_{\substack{\mu_1\cup\mu_2\cup\mu_3\cup\mu_4=(1^n)\\ \mu_2\cup\mu_2\cup\mu_3=(1)}} C(\mu_1;1)\Omega((\mu_1,\mu_2,\mu_3,\mu_4),(1^n))U((\mu_2,\mu_3,\mu_4),\mu_1;1) \\ &+O\left(n^2 2^{(n+m+k-1)n-2n-2\min(m,k)}\right). \end{aligned} \tag{27}$$

Assuming that only one intermediate node is unconnected, this neuron can be in either group 2, group 3 or group 4. In each instance $\Omega((\mu_1,\mu_2,\mu_3,\mu_4),(1^n)) = n$. If the unconnected intermediate node is in group 2, we have $U(((1),\phi,\phi),(1^{n-1});1) = 2^{n-1+k}$, if it is in group 3 we have, $U((\phi,(1),\phi),(1^{n-1});1) = 2^{n-1+m}$ while if it is in group 4, $U((\phi,\phi,(1)),(1^{n-1});1) = 1$. Using

$$C((1^{n-1});1) = 2^{(n+m+k-2)(n-1)}\left(1+O(n2^{-n})\right)$$

this yields

$$\sum_{\substack{\mu_1\cup\mu_2\cup\mu_3\cup\mu_4=(1^n)\\ \mu_2\cup\mu_2\cup\mu_3=(1)}} C(\mu_1;1)\Omega(\mu_1,\mu_2,\mu_3,\mu_4)U((\mu_2,\mu_3,\mu_4),\mu_1;1) \tag{28}$$
$$= \; 2^{(n+m+k-1)n}n\left(2^{-(n-1+k)}+2^{-(n-1+m)}+2^{-(2n-2+m+k)}\right)+O\left(n^2 2^{-2n-m-k}\right).$$

From equations (27) and (28)we have

$$\begin{aligned} C((1^n);1) \;=\;& 2^{(n+m+k-1)n}\left(1-n\left(2^{-m-n+1}+2^{-k-n+1}\right)\right) \\ &+O\left(2^{(n+m+k-1)n}n^2\left(2^{-2n-\min(m,k)}\right)\right). \end{aligned} \tag{29}$$

The proof of Theorem 15 then follows directly from (23) and (29).

□

4 Algorithms and numerical results

4.1 An algorithm to obtain $N(m, n_1, n_2, k)$

We used the tools developed in the previous sections to compute $N(m, n_1, n_2, k)$ exactly. We implemented the exact formula in Python 3.2 and obtained the values of $N(m, n_1, n_2, k)$ for a total number of up to twenty neurons (see table). The value of $N(5, 5, 5, 5)$ is larger than 10^{50} and was obtained in roughly 1 hour on a personal computer, as follows from Theorem 14 and Theorem 15, the ratio $N^*(m, n_1, n_2, k)/N(m, n_1, n_2, k)$ tends to 1 as the total number of nodes tends to infinity. A potential way to improve the running time of our algorithm would be to generate tables for the recursively defined functions C and T, avoiding multiple computations of the same values, but we didn't observe any practical gain in computation power.

4.2 An algorithm to generate the networks

While counting the networks satisfying a given set of conditions is interesting in its own right, we also provide an algorithm to generate the adjacency matrix associated with the networks counted by $N(m, n_1, n_2, k)$. As discussed below, in future works, generating these networks will allow us to perform several tests on the relationships between their structure and dynamical behaviour.

With each network with N nodes, we associate the adjacency matrix $A = (A_{i,j})_{i,j=1}^{N}$, which is such that $A_{i,j} = 1$ if there is a connection from node i to node j and $A_{i,j} = 0$ otherwise. In our case, $N = m + n + k$ and we adopt the following convention regarding the labelling of the nodes: $i = 1, \ldots, m$ for the input nodes, $i = m + 1, \ldots, m + n$ for the intermediate nodes, and $i = m + n + 1, \ldots, m + n + k$ for the output nodes. We define $\mathcal{E}$ as the following set

$$\mathcal{E} = \{(i, j) : \text{ the connection } i \to j \text{ is permitted } \}.$$

For every pair (i, j), we define the rank of (i, j) as

$$\text{rank}(i, j) = \#\{(i_1, j_1) \in \mathcal{E} : i_1 < i\} + \#\{(i_1, j_1) \in \mathcal{E} : i_1 = i, j_1 < j\}$$

so this function provides an ordering of the pairs (i, j) from 0 to $(n + m + k - 1)n - 1$. We can then assign a single integer index to each adjacency matrix A as follows:

$$\text{Index}(A) := \sum_{(i,j)\in\mathcal{E}} A(i, j)2^{\text{rank}(i,j)}$$

which is the most compact representation for a binary adjacency matrix given the forbidden connections of the present problem.

A Boolean value is attributed to each possible index value depending on whether or not the corresponding matrix is counted by the function $N(m, n_1, n_2, k)$. In order to check if an adjacency matrix A corresponds to a network satisfying the connectivity conditions

(m, n_1, n_2, k)	$N(m, n_1, n_2, k)$, (N/N^*)
$(1, 2, 0, 1)$	10, (0.27778)
$(1, 1, 1, 1)$	18, (0.28125)
$(2, 1, 1, 1)$	102, (0.39844)
$(1, 2, 2, 1)$	142982, (0.53653)
$(2, 2, 2, 1$	2694498, (0.63728)
$(2, 2, 2, 2)$	51866550, (0.76982)
$(3, 2, 2, 2)$	884419998, (0.82206)
$(2, 3, 2, 2)$	78187273596, (0.85166)
$(2, 3, 3, 2)$	455361619832116, (0.90933)
$(3, 3, 3, 2)$	29832388255291748, (0.93118)
$(3, 3, 3, 3)$	1955846953939134676, (0.95407)
$(4, 3, 3, 3)$	126638097270451278260, (0.96531)
$(3, 4, 3, 3)$	13070591789841196037595, (0.97299)
$(3, 4, 4, 3)$	34667056916819114999482177172, (0.98449)
$(4, 4, 4, 3)$	8909372636328795000677433447860, (0.98834)
$(4, 4, 4, 4)$	2289737167377258878090980801663700, (0.9922)
$(5, 4, 4, 4)$	5873165118545144771100744785795825940, (0.99415)
$(4, 5, 4, 4)$	7709370994349444222506223806797006161890, (0.99561)
$(4, 5, 5, 4)$	103675627514386351527569061315028472168329351376, (0.99756)
$(5, 5, 5, 4)$	106228616378925332256701907433326875374372326146256, (0.99817)
$(5, 5, 5, 5)$	108844524790336539487420588884391944954279893619583184, (0.99878)

$k :=$ number of input nodes
$n_1 :=$ number of intermediate nodes of type I
$n_2 :=$ number of intermediate nodes of type II
$m :=$ number of output nodes

Table 1: Total number of connected networks

mentioned in the Introduction, we used two basic facts: (1) ${A^\ell}_{i,j}$ corresponds to the number of paths of length ℓ from the i-th to the j-th vertex, and (2) the shortest path between two nodes has length at most $n+1$. It follows that, if we define

$$\text{connec} = \sum_{\ell=1}^{n+1} A^\ell,$$

then $\text{connec}_{s,t} > 0$ iff there is a path from the s-th node to the t-th neuron.

Lastly, to avoid counting isomorphisms of the same network, we make a list of all permutation matrices different from the identity that permute intermediate nodes of type I among themselves and intermediate nodes of type II among themselves. For all such permutation matrices P, the matrix $P^T AP$ corresponds to the adjacency matrix of a network which is an isomorphism of a network corresponding to the matrix A. We again implemented this in Python as described in Algorithm 1, and on a personal computer we could generate billions of networks in roughly 1 hour.

While many efficient algorithms that check network connectivity have been developed, such as those based on breadth-first search or depth-first search [18], we preferred to use the procedure explained above because it is the simplest, it avoids passing from adjacency matrices to lists of edges, and it does not have a significant impact on the running time of the whole algorithm. The most costly part of the algorithm is obviously the last one, which generates permutations of an adjacency matrix. Consequently, any significant algorithmic improvement would have to involve a conceptually new way to make sure that isomorphism of admissible networks are counted only once.

5 Conclusion

Beyond the stand alone interest of enumerating digraphs with a given set of sources and sinks, the present paper is relevant to structure enumeration in scientific fields such as neuroscience and the study of finite automata. In neuroscience, the encoding capacity of a network depends on the number of possible combinations of network structures, transfer functions and synaptic weights[16, 8, 27]. Hence, network counting can give hints with respect to the computational power of neural systems. Techniques of network enumeration can also be used to investigate the frequency of motif occurrences in neural networks [17] which can help to determine the local impact of global rules [25]. The number of functions that can be computed by a class of automata is also directly related to the size of this class. Since automata not satisfying the conditions given in the present paper would contain cells which behaviour would be irrelevant to the global behaviour of the automaton, we hope that the enumeration techniques here developed will eventually lead to a better understanding of the descriptive power of given classes of finite automata.

A first implication of the results of this paper is that, given the rate of increase of N as a function of the number of nodes, the naive idea of unambiguously inferring the structure of a

Algorithm 1 Generating networks satisfying the connectivity conditions

INPUT: Integers m, n_1, n_2, k.
OUTPUT: A Boolean B of length $2^{n(n+m+k-1)}$, where $n = n_1 + n_2$, such that $B_j = \text{true}$ iff the two following conditions are satisfied: 1) j is the index of an admissible network, 2) If j_1 is the index of an isomorphic network, then $j \leq j_1$.

INITIALIZATION
Set $M = 2^{n(n+m+k-1)}$.
Initialize B, a Boolean vector of length M with all its values equal to None.
Let perm be the list of all the permutation matrices different from the identity permuting intermediate nodes of type I and intermediate nodes of type II.

IDENTIFY ADMISSIBLE MATRICES

for j in range $(0, M)$:
  if B_j is equal to None:
    compute the adjacency matrix A corresponding to the index j
    compute the connectivity matrix connec $= \sum_{h=0}^{n+1} A^h$
    set check $= 1$
    for s in range $(0, n)$:
      update Check with
      check $=$ check $\left(\sum_{u=1}^{m} \text{connec}_{u,m+s}\right) \left(\sum_{u=1}^{k} \text{connec}_{m+s,m+n+u}\right)$
    if check > 0 :
      $B_j = \text{True}$
    else:
      $B_j = \text{False}$
    for P in perm:
      compute ℓ, the index associated with the matrix $P^T A P$
      set $B_\ell = \text{False}$

hidden network from input-output relationships is unrealistic under physiological data. Our results can also be used to derive bounds on the quantity of information needed to derive the network structure. Namely, the number of bits necessary to infer the network structure is at least

$$\frac{\log(N(m, n_1, n_2, k))}{\log(2)}.$$

Finally, we not only counted the admissible networks but also provided an efficient algorithm to generate these networks. In future works, this could be used to perform extensive searches over all admissible networks or to randomly sample networks over the set of all possibilities.

References

[1] A. R. Matamala, Pólya's combinatorial method and the isomer enumeration problem, *Bol. Soc. Chil. Quim.* **47** (2002), 105–112.

[2] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, Network information flow, *IEEE Trans. Info. Theory* **46** (2000), 1204–1216.

[3] N. Alon, P. Dao, I. Hajirasouliha, F. Hormozdiari, and S. C. Sahinalp, Biomolecular network motif counting and discovery by color coding, *Bioinformatics* **24** (2008), 241–249.

[4] G. E. Andrews and F. G. Garvan, Dyson's crank of a partition, *Bull. Amer. Math. Soc.* **18** (1988), 167–171.

[5] E. A. Bender and E. R. Canfield, Enumeration of connected invariant graphs, *J. Comb. Theory B* **34** (1983), 268–278.

[6] R. Birsdorff and J. L. Marichal, Counting non-isomorphic maximal independent sets of the n-cycle graph, *J. Integer Sequences* **11** (2008), Article 08.5.7.

[7] T. C. Craven, An application of Pólya's theory of counting to an enumeration problem arising in quadratic form theory, *J. Comb. Theory A* **29** (1980), 174–181.

[8] A. M. Dubreuil, Y. Amit, and N. Brunel, Memory capacity of networks with stochastic binary synapse, *PLoS Comput. Biol.* **10** (2014), 1–15.

[9] P. Erdős and A. Renyi, On random graphs I, *Publ. Math. Debrecen* (1959), 290–297.

[10] H. Fuks and K. Sullivan, Enumeration of number-conserving cellular automata rules with two inputs, *J. of Cell. Autom.* **2** (2007), 141–148.

[11] P. Grassberger, Some more exact enumeration results for 1D cellular automata, *J. Phys. A-Math. Theor.* **20** (1987), 4029–404.

[12] F. Harary, The number of linear, directed, rooted, and connected graphs, *Trans. Amer. Math. Soc.* **78** (1955), 445–463.

[13] F. Harary, The number of oriented graphs, *Michigan Math. J.* **4** (1957), 221–224.

[14] F. Harary and E. M. Palmer, *Graphical Enumeration*, Academic Press, 1973.

[15] R. van der Hofstad and J. Spencer, Counting connected graphs asymptotically. *Eur. J. Combin.* **27** (2006), 1294–1320.

[16] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Nat. Acad. Sci. USA* **79** (1982), 2554–2558.

[17] R. Itzhack, Y. Mogilevski, and Y. Louzoun, An optimal algorithm for counting network motifs, *Physica A: Stat. Mech. Appl.* **381** (2007), 482–490.

[18] D. Jungnickel, *Graphs, Networks, and Algorithms*, Springer, 2008.

[19] J. Knopfmacher, Asymptotic isomer enumeration, *J. Math. Chem.* **24** (1998), 61–69.

[20] G. Lavertu, S. L. Côte, and Y. De Koninck, Normal and abnormal coding of somatosensory stimuli causing pain, *Brain* **137** (2014), 724–738.

[21] B. Pittel and N. C. Wormald, Counting connected graphs inside-out, *J. Comb. Theory B* **93** (2005), 127–172.

[22] G. Pólya, Combinatorial enumeration of groups, graphs, and chemical compounds, *Acta Math.* **68** (1937), 145–254.

[23] M. Semenovich, B. Gordana, and D. Crnkovic, *Information and Computation: Essays on Scientific and Philosophical Understanding of Foundations of Information and Computation*, World Scientific Publishing, 2011.

[24] C. J. Twining and P.-M. Binder, Enumeration of limit cycles in noncylindrical cellular automata, *J. Stat. Phys.* **66** (1992), 385–401.

[25] E. Vasilaki and M. Giugliano, Emergence of connectivity motifs in networks of model neurons with short- and long-term plastic synapses, *PLoS ONE* **9** (2014), 1–17.

[26] X. Yan, R. W. Yeung, and Z. Zhang, An implicit characterization of the achievable rate region for acyclic multisource multisink network coding, *IEEE Trans. Info. Theory* **58** (2012), 5625–5639.

[27] J. Zunic, On encoding and enumerating threshold functions, *IEEE Trans. Neural Networks* **15** (2004), 261–267.

2010 *Mathematics Subject Classification*: Primary 05A15; Secondary 05A16, 20D60, 92B20.
Keywords: graphical enumeration, Pólya's enumeration theorem, digraph enumeration, hidden neural networks, finite automata enumeration.

Received December 8 2015; revised versions received March 3 2016; April 8 2016. Published in *Journal of Integer Sequences*, May 10 2016.

Return to Journal of Integer Sequences home page.