Annales Univ. Sci. Budapest., Sect. Comp. 47 (2018) 261–272

ENUMERATION OF CONNECTED FEEDFORWARD NETWORKS

Nicolas Doyon and Anthony Richard

(Québec, Canada)

Communicated by Jean-Marie De Koninck

(Received March 1, 2018; accepted April 18, 2018)

Abstract. In their seminal work, Erdős and Rényi [3] uncovered a sharp transition in the size of the largest connected component of random graphs. Here, we consider feedforward networks which have important applications in artificial intelligence and sensory neural networks [9] [13] . We obtain generating functions for both the total number of feedforward networks and the number of connected feedforward networks as well as an asymptotic expression for the total number of feedforward networks. Moreover, considering a random family of feedforward networks, we uncover a sharp threshold in their probability of being connected.

1. Introduction

Exact enumaration of graphs satisfying a given set of conditions is in itself a rich problem [10], [6], [7] with applications in many scientific fields [1], [2]. Previously, we enumerated connected networks [11]. In the present paper, we focus on the enumeration of *feedforward* networks which play important roles both in biological systems such as sensory systems and in artifical intelligence [12], [9]. Many networks can be considered as directed graphs and often contain

Key words and phrases: Erdős-Rényi, Graph enumeration, Pólya Theorem.
2010 Mathematics Subject Classification: Primary 11Y55, Secondary 05C80.
The Project is supported by NSERC Canada
https://doi.org/10.71352/ac.47.261

two privileged sets of nodes, the *input* nodes which receive external stimuli and *output* nodes which communicate the network computation to the outside world. We focus here on feedforward networks in which the nodes can be partitioned into disjoint sets $L_0, L_2, \ldots, L_k$ such that connections are possible only from nodes in L_j to nodes in L_{j+1}, $0 \leq j \leq k-1$.

1.1. Definitions and notation

Definition 1.1. A *network* is a directed graph $N = (V, V_{\text{Inp}}, V_{\text{Out}}, E)$ where V is the set of nodes, $V_{\text{Inp}} \subset V$ is the set of input nodes, $V_{\text{Out}} \subset V$ is the set of output nodes and $E \subset V \times V$ is the set of connections where (v_i, v_j) is a connection from node j to node i. The input and output nodes satisfy: $v_i \in V_{\text{Inp}} \Rightarrow (v_i, v_j) \notin E, \forall j$ and $v_i \in V_{\text{Out}} \Rightarrow (v_j, v_i) \notin E, \forall j$. Furthermore, $(v_i, v_i) \notin E, \forall i$. We define the set of internodes as $V_{\text{Inter}} := V \backslash (V_{\text{Inp}} \cup V_{\text{Out}})$.

Definition 1.2. A network is a *feedforward* network if there exists an integer $k \geq 1$ and a function f such that $f : V \to \{0, 1, \ldots, k\}$ with:

1) $v \in V_{\text{Inp}} \Leftrightarrow f(v) = 0$,

2) $v \in V_{\text{Out}} \Leftrightarrow f(v) = k$,

3) $(v_i, v_j) \in E \Rightarrow f(v_i) = f(v_j) + 1$.

If it is so, for j, $0 \leq j \leq k$, we define $\mathbf{L}_j := \{v \in V : f(v) = j\}$. We define the layer structure of a feedforward network by the vector $L \in \mathbb{N}^k$, $L = (L_1, L_2, \ldots, L_k)$ with $L_j := \#\mathbf{L}_j$.

Definition 1.3. We say that two networks $N^{(j)} = \left(V^{(j)}, V^{(j)}_{\text{Inp}}, V^{(j)}_{\text{Out}}, E^{(j)}\right)$, $j = 1, 2$ are *isomorphic* if $V^{(1)}_{\text{Inp}} = V^{(2)}_{\text{Inp}}$, $V^{(1)}_{\text{Out}} = V^{(2)}_{\text{Out}}$ and if there exists a bijective function $H : V^{(1)} \to V^{(2)}$ such that $(v_i, v_j) \in E^{(1)} \Leftrightarrow (H(v_i), H(v_j)) \in E^{(2)}$.

This definition implies that input and output nodes are considered distinguishable while internodes are considered indistinguishable.

Definition 1.4. We say that a feedforward network is connected if for each j such that $v_j \notin V_{\text{Inp}}$, $\exists v_i$ such that $(v_j, v_i) \in E$ and if for each j such that $v_j \notin V_{\text{Out}}$, $\exists v_i$ such that $(v_i, v_j) \in E$.

Definition 1.5. We say that two internodes v_i, v_j $(i \neq j)$ are *redundant* if $(v_i, v_\ell) \in E \Leftrightarrow (v_j, v_\ell) \in E$ and that a network is redundant if it contains at least two nodes which are redundant.

Notation. We write $\lambda \vdash n$ if λ is an integer partition of n.

1.2. The Pólya Enumeration Theorem

The Pólya Enumeration Theorem is a flexible tool to enumerate structures such as graphs or molecule conformations [8]. Letting G stand for the group of permutations between indistinguishable nodes of a graph and N for the number of nodes of this graph, it can be expressed as

$$F(x) = \frac{1}{\#G} \sum_{s \in G} P_s(x), \qquad P_s(x) = \prod_{j=1}^{\infty} \left(1 + x^j\right)^{o_s(j)} \tag{1.1}$$

with $o_s(j)$ the number of orbits of length j induced on the possible connections by the node permutation s. Here and below, observe that the product is actually a finite one since the length of an orbit is smaller or equal than the number of possible connections which is $\leq N^2$. The function $F(x)$ is the generating function for the number of networks where the coefficient of x^n is the number of networks with exactly n connections. Since $P_s(x)$ depends only on the *type* of the permutation s, (1.1) can be written as

$$F(x) = \frac{1}{\#G} \sum_{\lambda \in \mathcal{B}} \#\{s \in G : \mathrm{type}(s) = \lambda\} \prod_{j=1}^{\infty} \left(1 + x^j\right)^{o_\lambda(j)} \tag{1.2}$$

where $\mathcal{B}$ is the set of all possible permutation types. For instance, if we consider permutations of N elements and impose no restriction, $\mathcal{B}$ is the set of all partitions of N.

1.3. Statement of results

Write $F_x(L)$ for the generating function of all feedforwad networks with structure layer $L = (L_0, L_1, \ldots, L_k)$.

Theorem 1.1. *Let k be a positive integer and let $L = (L_0, L_1, \ldots, L_k)$ be a vector of positive integers, then*

$$F_x(L) = \sum_{\substack{\lambda = (\lambda_1, \ldots, \lambda_{k-1}) \\ \lambda_j \vdash L_j,\, 1 \leq j \leq k-1}} \left(\prod_{j=1}^{k-1} \prod_{\ell=1}^{L_j} \frac{1}{\ell^{a(\ell,j)} a(\ell,j)!} \right) P_x(\lambda_1) P_x(\lambda_{k-1}) \prod_{j=1}^{k-2} Q_x(\lambda_j, \lambda_{j+1})$$

where $a(\ell, j)$ is the number of occurrences of the integer ℓ in the partition λ_j. The auxiliary polynomials P and Q are defined by

$$P_x(\lambda) = \prod_{\ell=1}^{\infty} \left(1 + x^\ell\right)^{a(\ell)}$$

where $a(\ell)$ is the number of occurrence of ℓ in the partition λ and

$$Q_x(\lambda_j,\lambda_\ell)=\prod_{\substack{1\le r\le\infty\\1\le s\le\infty}}\left(1+x^{\mathrm{LCM}(r,s)}\right)^{rs\cdot a(r,j)a(s,\ell)/\mathrm{LCM}(r,s)}.$$

From the generating function $F_x(L)$, we obtain

Theorem 1.2.

$$F_1(L)=\frac{2^{\sum_{j=0}^{k-1}L_jL_{j+1}}}{\prod_{j=1}^{k-1}L_j!}\left(1+O\left(k^2B^22^{-2A}\right)\right) \tag{1.3}$$

provided that $kB2^{-2A}<1$ where $A=\min_{j\in\{0,\dots,k\}}L_j$ and $B=\max_{j\in\{0,\dots,k\}}L_j$.

We define $C_x(L)$ as the generating function for the number of feedforward networks with structure layer L in which every node sends at least a connection but doesn't necessarily receive one and we define implicitly $C_x(L,\lambda)$ by the contribution to $C_x(L)$ of a permutation of type λ.

Given $\lambda=(\lambda_1,\lambda_2,\dots,\lambda_{k-1})$, we define $a_\lambda(t,\ell)$ as the number of occurrences of the integer ℓ in λ_t. We say that $\nu=(\nu_1,\nu_2,\dots,\nu_{k-1})$, satisfies $\nu\subseteq\lambda$ if $a_\nu(t,\ell)\le a_\lambda(t,\ell)$ for all t,ℓ. If $\nu\subseteq\lambda$ we define

$$J(\nu,\lambda)=\prod_{t=1}^{k-1}\prod_{\ell=1}^{L_t}\binom{a_\lambda(t,\ell)}{a_\nu(t,\ell)}.$$

Theorem 1.3. *Let $L=(L_0,L_1,\dots,L_k)$ be a vector of positive integers. Let $K_x(L)$ be the generating function for the number of connected feedforward networks having layer structure L, we have*

$$K_x(L)=\sum_{\substack{\lambda=(\lambda_1,\lambda_2,\dots,\lambda_{k-1})\\\lambda_j\vdash L_j,\,1\le j\le k-1}}K_x(L,\lambda),\qquad \textit{with}$$

$$K_x(L,\lambda)=C_x(L,\lambda)-\sum_{\substack{\nu\subset\lambda\\\nu\neq\phi}}J(\nu,\lambda)K_x(L,\lambda-\nu)R_x(L,\nu,\lambda) \tag{1.4}$$

where

$$R_x(L,\lambda,\nu)=\prod_{t=1}^{k-1}\prod_{\substack{1\le j<\infty\\1\le\ell<\infty}}\left(\left(1+x^{\mathrm{LCM}(j,\ell)}\right)^{a_\nu(t+1,\ell)}-1\right)^{a_\lambda(t,j)j\cdot\ell/\mathrm{LCM}(j,\ell)}.$$

We define $M(L)$ as the number of feedforward networks with layer structure L that are both connected and non redundant. We obtain

Theorem 1.4.

$$\begin{aligned} M(L) &= L_k! \prod_{j=0}^{k-1} \left(\sum_{m=0}^{L_j} (-1)^m \binom{L_j}{m} \binom{2^{L_j} - m - 1}{L_{j+1}} \right) = \\ &= \frac{2^{\sum_{j=0}^{k-1} L_j L_{j+1}}}{\prod_{j=1}^{k-1} L_j!} \left(1 + O\left(\sum_{j=0}^{k-1} \left(\frac{L_j^2}{2^{L_{j+1}}} + \frac{L_{j+1}^2}{2^{L_j}} \right) \right) \right). \end{aligned}$$

Finally, we define $\mathcal{G}(n,k,p)$ as the family of random feedforward networks with k layers of n nodes in which each possible connection actually exists with an independent probability of p. We obtain the following.

Theorem 1.5. *For a positive real number b, the probability that a network in $\mathcal{G}(n,k,p)$ is strongly connected when $p = \frac{\log(nk/b)}{n}$ and as n tends to infinity is equal to*

$$e^{-2b} \left(1 + O\left(\frac{k \log^2 n}{n} \right) \right).$$

2. Proof of the Theorems

2.1. Enumeration of all feedforward networks

Let k be a positive integer and let $L = (L_0, L_1, \ldots, L_k)$ be a vector of positive integers. Let $G = G(L)$ be the group of permutations between indistinguishable nodes. We have, $G = S_{L_1} \times \ldots \times S_{L_{k-1}}$ and

$$\#G = \prod_{j=1}^{k-1} L_j!. \tag{2.1}$$

For $s \in G$, write $s = (s_1, s_2, \ldots, s_{k-1})$ where s_j is the permutation between the nodes of layer j and set $\ell_{r,j}(s)$ as the number of disjoint cycles of length r in s_j. We say that two permutations s and s^* are of the same type if

$\ell_{r,j}(s) = \ell_{r,j}(s^*), \quad \forall r, j$. We write type$(s) = \lambda = (\lambda_1, \ldots, \lambda_{k-1})$ with $\lambda_j \vdash L_j$. Equation (1.2) yields

$$F_x(L) = \frac{1}{\prod_{u=1}^{k-1} L_u!} \sum_{\substack{\lambda=(\lambda_1,\ldots,\lambda_{k-1}),\\ \lambda_j \vdash L_j, 1\leq j\leq k-1}} \#\{s \in G, \text{type}(s) = \lambda\} P_\lambda(x) \tag{2.2}$$

where $P_\lambda(x)$ is the contribution to $F_x(L)$ from a permutation of type λ. Let $a(r,j)$ be the number of occurences of r in λ_j. The number of permutations of L_j elements with $a(r,j)$ cycles of length r for all r is

$$\frac{L_j!}{\prod_{r=1}^{L_j} r^{a(r,j)} a(r,j)!}.$$

Taking the product over all layers j, we get from (2.2)

$$F_x(L) = \sum_{\substack{\lambda=(\lambda_1,\ldots,\lambda_{k-1}),\\ \lambda_j \vdash L_j,\, 1\leq j\leq k-1}} P_\lambda(x) \prod_{j=1}^{k-1} \prod_{r=1}^{L_j} \frac{1}{r^{a(r,j)} a(r,j)!}. \tag{2.3}$$

Now consider a cycle of length v on nodes in layer j and one of length w on nodes in layer $j+1$, these induce orbits of lengths LCM(v,w) on connections from nodes in layer j to nodes in layer $j+1$. Given that the total of possible connections is $v \cdot w$, the number of orbits is $\frac{v \cdot w}{\mathrm{LCM}(v,w)}$. The corresponding contribution to $P_\lambda(x)$ from these connections is thus

$$\left(1 + x^{\mathrm{LCM}(v,w)}\right)^{v\cdot w/\mathrm{LCM}(v,w)}. \tag{2.4}$$

We write $P_\lambda(x) = P_\lambda^{(1)}(x) P_\lambda^{(2)}(x) P_\lambda^{(3)}(x)$ where $P_\lambda^{(1)}(x)$ accounts for the connections between the input nodes and the first layer of internodes, $P_\lambda^{(2)}(x)$ for the connections between internodes and $P_\lambda^{(3)}(x)$ for the connections between the last layer of the internodes and the output nodes. We straightforwardly have from (2.4)

$$P_\lambda^{(1)}(x) = \prod_{r=1}^{\infty} (1 + x^r)^{a(1,r)} \quad \text{and} \quad P_\lambda^{(3)}(x) = \prod_{r=1}^{\infty} (1 + x^r)^{a(k-1,r)} \tag{2.5}$$

where $a(j,r)$ is the number of cycles of length r in the permutation of the jth internode layer. Similarly, we have

$$P_\lambda^{(2)}(x) = \prod_{j=1}^{k-2} \prod_{\substack{1\leq r<\infty\\ 1\leq s<\infty}} \left(1 + x^{\mathrm{LCM}(r,s)}\right)^{a(j,r)a(j+1,s)rs/\mathrm{LCM}(r,s)}. \tag{2.6}$$

Putting equations (2.5) and (2.6) in (2.3), we prove Theorem 1.1. ■

2.1.1. An asymptotic formula for $F_1(L)$

We show that the main contribution to $F_1(L)$ comes from the identity permutation. This contribution is

$$q_1 := \frac{2^{\sum_{j=0}^{k-1} L_j L_{j+1}}}{\prod_{j=1}^{k-1} L_j!}. \tag{2.7}$$

Now let s be a permutation leaving all but two nodes of layer j^* fixed. Let q_2 be the contribution of this permutation to $F_1(L)$. We have

$$\frac{q_2}{q_1} = \binom{L_{j^*}}{2} 2^{-L_{j^*+1} - L_{j^*-1}}.$$

Let Q_m stand for the total contribution from permutations leaving all but m nodes fixed. In particular, we have assuming that $A \leq L_j \leq B$ for each j,

$$\frac{Q_2}{q_1} = O\left(kB^2 2^{-2A}\right). \tag{2.8}$$

We now want to obtain an upper bound for Q_m for each integer $m \geq 3$. Let $D_m = (d_1, \ldots, d_{k-1})$ be a vector of non negative integers summing up to m and let $W(D_m)$ be the number of networks counted by permutations leaving all but d_j interneurons fixed in layer j. We then have

$$\begin{aligned}\frac{W(D_m)}{q_1} &\leq \prod_{j=1}^{k-1} \binom{L_j}{d_j} d_j! 2^{-d_j(L_{j+1}+L_{j-1})} \leq \prod_{j=1}^{k-1} L_j^{d_j} 2^{-d_j(L_{j+1}+L_{j-1})} \leq \\ &\leq B^m 2^{-Am}.\end{aligned}$$

Summing this over all admissible vectors D_m, we obtain

$$\frac{Q_m}{q_1} \leq \left((kB)2^{-2A}\right)^m. \tag{2.9}$$

Combining equations (2.7), (2.8) and (2.9), completes the proof Theorem 1.2. ∎

2.2. Enumerating connected networks

We now give a generating function for the number of connected feedforward networks, thus proving Theorem 1.3. Let $C_x(L)$ be the generating function for the number of networks with layer structure L in which each node (except output nodes) sends a forward connection and let $K_x(L)$ be the generating function for the number of networks with layer structure L satisfying the supplemental condition that each node (except input nodes) receives a connection. Similarly, let $C_x(L,\lambda)$ and $K_x(L,\lambda)$ be generating functions for the number of networks left invariant by a permutation of type λ . We have

$$C_x(L,\lambda)=\prod_{s=0}^{k-1}\prod_{\substack{1\le j<\infty\\ 1\le \ell<\infty}}\left((1+x^{\mathrm{LCM}(j,\ell)})^{a(s+1,\ell)}-1\right)^{a(s,j)j\cdot\ell/\mathrm{LCM}(j,\ell)} \tag{2.10}$$

where $a(s,j)$ is the number of occurrences of the integer j in the partition λ_s. The minus 1 term is necessary to exclude the case where the permuted nodes in layer s send no forward connection. We obtain $K_x(L,\lambda)$ from $C_x(L,\lambda)$ by subtracting the generating function associated to networks in which each node sends a forward connection but in which some nodes don't receive any connection. We divide the internodes of a network into two sets, those who receive a path from at least one input node and those who don't. Given a permutation s of type λ leaving connections invariant, we let s' be the subset of this permutation acting on internodes that don't receive a path from input nodes and denote ν the type of the permutation s'. Between two consecutive layers, connections from a nodes permuted by $s\backslash s'$ to a neode permuted by s' are not allowed. We thus have

$$K_x(L,\lambda)=C_x(L,\lambda)-\sum_{\substack{\nu\subset\lambda\\ \nu\neq\phi}}J(\nu,\lambda)K_x(L,\lambda-\nu)R_x(L,\nu,\lambda),\qquad\text{with}$$

$$R_x(L,\lambda,\nu)=\prod_{t=1}^{k-1}\prod_{\substack{1\le j<\infty\\ 1\le \ell<\infty}}\left(\left(1+x^{\mathrm{LCM}(j,\ell)}\right)^{a_\lambda(t+1,\ell)}-1\right)^{a_\nu(t,j)j\cdot\ell/\mathrm{LCM}(j,\ell)}$$

where $a_\lambda(t,j)$ is the number of occurrences of j in λ_t. The coefficient $J(\nu,\lambda)$ counts the number of possible ways of choosing a subpermutation of type ν in a permutation of type λ. Explicitly, we have

$$J(\nu,\lambda)=\prod_{t=1}^{k-1}\prod_{j\ge1}\binom{a_\lambda(t,j)}{a_\nu(t,j)}.$$

2.3. Enumerating connected, non redundant networks

We let $M(L)$ be the number connected and non redundant networks having layer structure L. The inward connections received by a node in layer j can be seen as a binary sequence where the ℓ^{th} entry is equal to one if and only if the node receives a connection from the ℓ^{th} node of layer $j-1$. From the non redundancy condition, the L_j corresponding sequences must take distinct values so the number of possible ways of choosing the connections from nodes in layer $j-1$ to nodes in layer j is given by $\binom{2^{L_{j-1}}}{L_j}$. Assuming that each node in layer j receives at least one input from a node in layer $j-1$ reduces this number to $\binom{2^{L_{j-1}}-1}{L_j}$. However this also counts networks in which some nodes don't send any connection. To correct this, we use the inclusion-exclusion principle and obtain the number of connected subgraphs between nodes of adjacent layers L_j and L_{j+1}, namely

$$\sum_{m=0}^{L_j}(-1)^m\binom{L_j}{m}\binom{2^{L_j-m}-1}{L_{j+1}}. \tag{2.11}$$

Taking the product over all layers of nodes we get from (2.11)

$$M(L)=L_k!\prod_{j=0}^{k-1}\left(\sum_{m=0}^{L_j}(-1)^k\binom{L_j}{m}\binom{2^{L_j-m}-1}{L_{j+1}}\right). \tag{2.12}$$

where the factor $L_k!$ comes from the fact that the nodes in the output layer are by construction distinguishable. Equation (2.12) thus proves the first part of Theorem 1.4. In order to prove the second part of Theorem 1.4, first observe that

$$\begin{aligned}\binom{2^{L_j}-1}{L_{j+1}} &= \frac{(2^{L_j}-1)!}{L_{j+1}!(2^{L_j}-1-L_{j+1})!}=\\ &=\frac{(2^{L_j}-1)^{L_{j+1}}}{L_{j+1}!}\left(1+O\left(\frac{L_{j+1}}{2^{L_j}}\right)\right)^{L_{j+1}}=\\ &=\frac{2^{L_jL_{j+1}}}{L_{j+1}!}\left(1+O\left(\frac{L_{j+1}^2}{2^{L_j}}\right)\right).\end{aligned} \tag{2.13}$$

We also have

$$\begin{aligned}\sum_{m=1}^{L_j}(-1)^m\binom{L_j}{m}\binom{2^{L_j-m}-1}{L_{j+1}} &\le \sum_{m=1}^{L_j}\binom{L_j}{m}\binom{2^{L_j-m}-1}{L_{j+1}}=\\ &=\frac{2^{L_jL_{j+1}}}{L_{j+1}!}O\left(\frac{L_j}{2^{L_{j+1}}}\right).\end{aligned} \tag{2.14}$$

Putting (2.13) and (2.14) together and taking the product over all layers, we obtain from 2.11

$$M(L)=\frac{2^{\sum_{j=0}^{k-1}L_jL_{j+1}}}{\prod_{j=1}^{k-1}L_j!}\left(1+O\left(\sum_{j=0}^{k-1}\left(\frac{L_j^2}{2^{L_{j+1}}}+\frac{L_{j+1}^2}{2^{L_j}}\right)\right)\right),$$

thereby proving the second part of Theorem 1.4. ∎

3. The probability of being connected

As is often the case in the study of graphs and networks [5], [4], we define a family of random networks and compute the probability that a network satisfies a relevant property. Given positive integers n and k as well as a parameter p, $0 < p < 1$, we define the family of random networks $\mathcal{G}(n, k, p)$ as the family of feedforward networks with k layers of n nodes such that each possible connection has an independant probability p of actually existing.

Let $P(a, p, n)$ stand for the probability that two adjacent layers each of n nodes are connected and that the total number of connections between the two layers is equal to a. Since the total number of possible connections between the two layers is n^2, we have by the inclusion-exclusion principle

$$P(a,p,n) = \sum_{s=0}^{n}\sum_{r=0}^{n}(-1)^{s+r}\binom{n}{s}\binom{n}{r}\binom{(n-s)(n-r)}{a}p^a(1-p)^{n^2-a}.$$

Summing the above over a, we get

$$\sum_{a=1}^{n^2} P(a,p,n) = \sum_{s=0}^{n}\sum_{r=0}^{n}(-1)^{r+s}\binom{n}{s}\binom{n}{r}(1-p)^{nr+ns-rs}. \tag{3.1}$$

Assuming that $p = \frac{\log(nk/b)}{n}$ for a positive constant b, we have

$$\begin{aligned}(1-p)^n &= \exp\left(n\log(1-p)\right) = \\ &= \exp\left(-\log n - \log(nk/b) + O\left(\frac{\log^2 n}{n}\right)\right) = \\ &= \frac{b}{nk}\exp\left(O\left(\frac{\log^2 n}{n}\right)\right).\end{aligned}$$

Thus we get from 3.1

$$\begin{aligned}&\sum_{a=1}^{n^2} P(a,p,n) = \\ &= \sum_{s=0}^{n}\sum_{r=0}^{n}(-1)^{r+s}\binom{n}{s}\binom{n}{r}\frac{b^{r+s}}{(nk)^{r+s}}\left(1+O\left(\frac{\log^2 n}{n}+p(r+s)\right)\right).\end{aligned} \tag{3.2}$$

Using the relation

$$\binom{n}{r} = \frac{n^r}{r!}\left(1+O\left(\frac{r^2}{n}\right)\right)$$

in (3.2) we obtain that $\sum_{a=1}^{n^2} P(a,p,n)$ is equal to

$$\sum_{s=0}^{n}\sum_{r=0}^{n}(-1)^{r+s}\frac{(b/k)^{r+s}}{r!s!}\left(1+O\left(\frac{\log^2 n + r^2 + s^2}{n} + p(r+s)\right)\right).$$

If R and S tend to infinity with n, the above is equal to

$$\sum_{s=0}^{S}\frac{(-1)^s(b/k)^s}{s!}\times$$
$$\times\sum_{r=0}^{R}\frac{(-1)^r(b/k)^r}{r!}\left(1+O\left(\frac{\log^2 n + R^2 + S^2 + \log n(R+S)}{n}\right)\right)+$$
$$+O\left(e^{-S}+e^{-R}\right).$$

Choosing $S = R = \log n$, we obtain

$$\sum_{a=1}^{n^2} P(a,p,n) = \exp(-2b/k)\left(1+O\left(\frac{\log^2 n}{n}\right)\right).$$

Finally, taking the product over all layers, we obtain

$$\mathcal{G}(n,k,p) = e^{-2b}\left(1+O\left(\frac{k\log^2 n}{n}\right)\right)$$

completing the proof of Theorem 1.5. ∎

References

[1] **Akutsua, T. and H. Nagamochi,** Comparison and Enumeration of Chemical Graphs, *Computational and Structural Biotechnology Journal,* **5** (2013).

[2] **Dale, M.R.T.,** *Applying Graph Theory in Ecological Research*, Cambridge University Press, Cambridge, 2017.

[3] **Erdős, P. and A. Rényi,** On random graphs I, *Publicationes Mathematicae*, **6** (1959), 290–297.

[4] **Erdős, P. and A. Rényi,** On the evolution of random graphs, *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, **5** (1960), 17–61.

[5] **Gilbert, E.N.,** Random graphs, *Annals of Mathematical Statistics*, **30** no. 4 (1959), 1141–1144.

[6] **Harary, F.,** The number of oriented graphs, *Michigan Math. J.*, **4** no. 3 (1957), 221–224.

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

[8] **Horvath, A.L.,** *Molecular Design: Chemical Structure Generation from the Properties of Pure Organic Compounds*, Elservier Science, Series: Studies in Physical Theoretical Chemistry, **75**, 1992.

[9] **El-Melegy, M.T., M.H. Essai and A.A. Ali,** *Robust Training of Artificial Feedforward Neural Networks*, Springer, Foundations of Computational Intelligence, **201**, 2009.

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

[11] **Richard, A., P. Desrosiers, S. Hardy and N. Doyon,** Counting hidden neural networks, *Journal of Integer Sequences*, 16.4.7 vol. 19. (2016).

[12] **Sinha, D.B., N.M. Ledbetter and D.L. Barbour,** Spike-timing computation properties of a feed-forward neural network model, *Frontiers in Computational Neuroscience,* **8** no. 5, (2014).

[13] **Zhang, T. and T.O. Sharpee,** A robust feedforward model of the olfactory system, *PLoS Comput Biol.*, **12** no. 4, (2016).

N. Doyon and A. Richard
Université Laval
Québec
Canada
`nicolas.doyon@mat.ulaval.ca`
`anthony.richard.1@ulaval.ca`