Information transmission in complex neural networks

Author: Juan José Ojeda Guerrero.
Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain.[*]

Advisor: Maria Ángeles Serrano Moral

Abstract: Computational neural networks are inspired in the brain structure and designed to mimic its intelligence artificially. In particular, they can be used as a tool to explore information transport and processing among interconnected units forming layers. In this article we study how the structure of the network (number of layers) and the state of the connections (edge weights) affect to the information flow. A computational model, that allows us to simulate neurons' collective behaviour, has been used over different neural networks configurations, analysing the variation of some structure features. Results show a strong dependence between the number of connections and the response of the network. Also, we have found a relationship among the edge weights distribution and the propagation of the information from the input node to the output layer.

I. INTRODUCTION

Santiago Ramón y Cajal and *Camillo Golgi* won in 1906 the *Nobel prize* in Medicine due to the discovery of that the central nervous system was composed by several small units called neurons and their behaviour was like a relays in an electric circuit [1]. Ever since that, neuroscience has become a field that attracts the interest of many scientists from various disciplines. For instance, biologists and biophysicists have been working hard to build accurate and complex models that fit well with the conducted experiments about how neurons interact with its neighbours. These experiments tell us that neurons share information through changing the electric potential of the media by means of modifying the flux through its ion channels. So, these models are based mainly in compute the solutions for the ion concentration *differential kinetic equations* and apply stochastic methods for the neurons' response [2]. These kind of models have allowed us to study of the neurons' behaviour, giving some clues for the understanding of brain main activities at neuron level.

At large scale, today the use of computational neural networks (*NNs*) models plays a main role in the AI (*Artificial Intelligence*) development. These models are based on compute an output for a given input after the information have passed through the *NN* with a defined *node topology* (neurons' distribution) and *weighed edges* (synaptic connections' state). These networks can be trained by tuning the edge weights depending on whether the output is correct or not [3]. Therefore, for these models the most important feature are the connection types (long range, self-loop, ...) and its weights.

Then, computational models based on the biological behaviour of the neurons, not *NNs*, claims to recreate the complexity of the human brain by performing detailed computations for enormous number of individual simulated neurons. For instance, more than 20 million neurons conform IBM's brain model [4]. But, these models are too complex to build and execute in an standard computer, due to the huge number of needed computations to simulate even an small portion of the brain. On the other hand, *NNs* used for deep learning usually are fixed fully connected networks (without stochastic processes) to make easier the computation of the learning algorithm. For that reason these models are not able to reflect complex collective phenomenons. So, we have must to choose one model that ensures to reproduce brain key features and allow us to record and analyse the whole process activity. Therefore, the chosen model has to lie in the middle point between both presented kind of models. In this point we found one common feature that belongs to the fundamental nature of a brain: how information flows through it depending on its structure. For that reason, in this article we have focused on the study of information propagation and how its changes are related with the topology features of the computational model.

II. THE NEURAL NETWORK MODEL

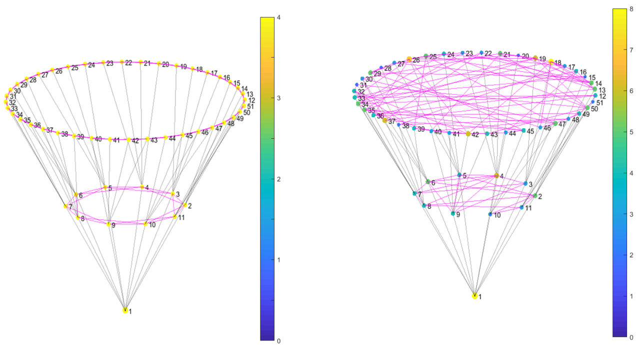
In the paper *Percolation Model of Sensory Transmission and Loss of Consciousness Under General Anesthesia* [5] authors propose a *NN* computational model and they sentence that it is able to reproduce electroencephalographic features associated with loss of consciousness during general anesthesia.

This model has been replicated, but instead of analyse only the impact of the weight edges modification over the information flux, we have also studied the affectation of multiple structure key parameters. This has allowed us to analyse how an input signal is transformed after passing through a network depending on its set-up. Each used configuration represents a portion of a brain under different conditions.

[*] Electronic address: jjo1984@gmail.com

A. Network structure

Cerebral cortex is known to be build in **layers** [6], so nodes in neural networks are distributed in layers as well. This layers are made of nodes that belong to the same generation. That is, all of the predecessor nodes belong to the previous layer. In addition to the single connection with the previous layer and the T connections to the following one, each node has k_i edges with its layer neighbours, depending on the result of apply Watts-Strogartz algorithm [7] to it.
Watts-Strogartz algorithm gives us the possibility of change the intra-layer connections' configuration by varying the parameter β between 0 and 1. Whether the algorithm is applied to a regular ring lattice with *mean degree* $K = \left\langle \sum_{i=1}^{N} k_i \right\rangle$, and the total number of nodes (N) satisfies that $N >> K >> \ln N >> 1$, one obtains the following network features: If $\beta = 0 \rightarrow$ It doesn't change the regular ring lattice, so the degree of each node would be the same ($k_i = K, \ \forall \ i$). If $\beta \approx 0.5 \rightarrow$ One obtains an small-world type network, where $k_i \neq K$ is possible. And, if $\beta \approx 1 \rightarrow$ A random connection distribution is the result of applying the algorithm in this case. In the all three cases, the mean degree K remains constant.
An example of how this algorithm changes the layer edge distribution is shown at *Figure 1.*



(a) $\beta = 0$ (b) $\beta = 0.8$

Figure 1: Comparison of 3-layer network examples with $I_{nod} = 10$, $T = 4$, $K = 4$, and different Watts-Strogatz parameter (β). Nodes are labelled ascending from input to outputs. Connections within the layer itself are coloured pink, and projections to the next layer are drawn in black. The colour-bar legend represent the degree k of connections that have each node with its layer neighbours.

Nodes are the computation units of the *NN*, like neurons in a brain. There the information that surrounds the node is compiled and transformed. Our model is built on the basis of a hierarchical fractal network ascending from an input node to multiple output nodes and structured in layers. It means that each node projects the same number of edges to the next layer, except the input node (layer 1). For our simulations we have built a 1-50-4 networks. This is, the input node projects 50 connections to the 2nd layer, and these 50 nodes project 4 new connections to the following layer, and so on. Therefore, our networks contain $N(L)$ nodes where L is the total number of layers, including the input node. Hence,

$$N(L) = 1 + \sum_{l=1}^{L-1} I_{nod} \cdot T^{(l-1)}, \tag{1}$$

where I_{nod} is the number of nodes at the second layer, in our case $I_{nod} = 50$, and T is the number of edge projections to the next layer, so in our model $T = 4$. This kind of structure tries to replicate thalamocortical structure of mammalian brain.

B. Edge weights

Each node establishes connections, **edges**, with its neighbours to transmit information through them. How much information could be transmitted depends on the weight assigned to the edge, for instance, w_{ij} modulates the information flow from node j to node i. Weights sampling is done stochastically and characterized by a given probability distribution (*CDF*). Moreover, these weights are directional ($w_{ij} \neq w_{ji}$), and in addition, self-loop connections (w_{ii}) are allowed and represent the memory of the past activity in the node. This yields,

$$w_{ij} = \begin{cases} \frac{1}{2} \cdot \left[1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}} \right) \right] & \text{if} \quad i \neq j, \\ c \cdot \exp\left[-\lambda \sum_{k \neq i} w_{ki} \right] & \text{if} \quad i = j, \end{cases} \tag{2}$$

where $0 \leq x \leq 1$ is a random variable uniformly distributed. $\sigma = 0.05$ is the standard deviation of the CDF. $\mu = 1 - p$ is where the CDF is centred, and it depends on a percolation parameter p. Therefore, in our model p represents the probability that a node have to stablish connections with its neighbours and have values between 0 (poor connections) and 1 (excellent connections). So, the bigger p the bigger probability to have a good flux of information between the input node and the output layer. The values for the other parameters have been obtained by replicating the results from [5]: $c = 1 \rightarrow$ Modulates the memory term and it could be a value between 0 and 1. And $\lambda = 0.2 \rightarrow$ Determines the importance of how well connected is the node with its neighbours. The better outsider connections the worst memory term.
In consideration of the synaptic dynamic behaviour we have incorporated the parameter α to module weights resampling periodicity. It would be proportional to $e^{-\alpha \cdot p}$, where p is the percolation parameter.

C. Neural activity

At every time step all nodes' value have to be updated taking into account the value of the information arrived to its surroundings. We need a function that computes the weighed sum of the connected neighbours information values and the internal memory term as well,

$$A_i(t) = \frac{\sum_j w_{ij} P_j(t)}{\sum_j w_{ij}} = \frac{w_{ii} P_i(t) + \sum_{j \neq i} w_{ij} P_j(t)}{w_{ii} \sum_{j \neq i} w_{ij}}. \quad (3)$$

In this equation $A_i(t)$ is the activity of the node i at the time step t, and $P_j(t)$ is a function which contains the activity history of each node from the preceding m time steps, and it is defined as,

$$P_j(t) = \frac{\sum_{\tau=1}^{m} e^{-\tau} A_j(t-\tau)}{\sum_{\tau=1}^{m} e^{-\tau}}, \text{ with } m = t-1. \quad (4)$$

D. Simulations

A simulation consists in passing a sinusoidal signal value to the input node for each time step ($\Delta t = 10^{-3} s$), and perform all the computations (eq. 3 and 4) for the whole network nodes. We have fully developed a program in *Matlab/Octave* language which is able to perform this simulations over any complex *NN* (limited by the computer memory availability) based on the model described at previous sections and characterized by the parameters showed at *Table I*.

We have focused our research on variate the following parameters: $3 \leq L \leq 5$, $4 \leq K \leq 28$ (each 4), $0 \leq \beta \leq 0.8$ (each 0.4), $0 \leq p \leq 1$ (each 0.25) and $4 \leq \alpha \leq 12$ (each 4, and the option without weight resampling). We have injected a 115 Hz input clean cosine signal along 5 seconds, in networks built with different structure set-ups and the all possible combinations of the mentioned parameters. In total we have executed up to 60 different simulations for each one of the 3 studied network models (NM). All of them have $T = 4$ and $I_{nodes} = 50$, but the total number of layers (L) is different. The model 1 has $L = 3$, the model 2 has $L = 4$, and the model 3 has $L = 5$.

III. RESULTS

All networks neural activity (nodes' value) has been recorded, allowing us to recreate and analyse any instant of the process. Furthermore, had been working with a clean input signal has allowed us to take the dispersion of the nodes respect to the mean value of the whole layer activity (the standard deviation S) as an indicator of the layer's coherence. So, for each time-step and each layer we have computed $S_l(t)$ as follows,

$$S_l(t) = \sqrt{\frac{1}{n_l - 1} \sum_{i=1}^{n_l} |A_i(t) - \mu_l(t)|^2}. \quad (5)$$

Where $A_i(t)$ is the activity of the node i at time t, n_l is the total number of nodes at the layer l and $\mu_l(t)$ is the mean value of all layer l nodes at time t,

$$\mu_l(t) = \frac{1}{n_l} \sum_{i=1}^{n_l} A_i(t). \quad (6)$$

At *Figure 2* it is shown how the input signal is gathered at the output layer, depending on the percolation parameter and the chosen model.

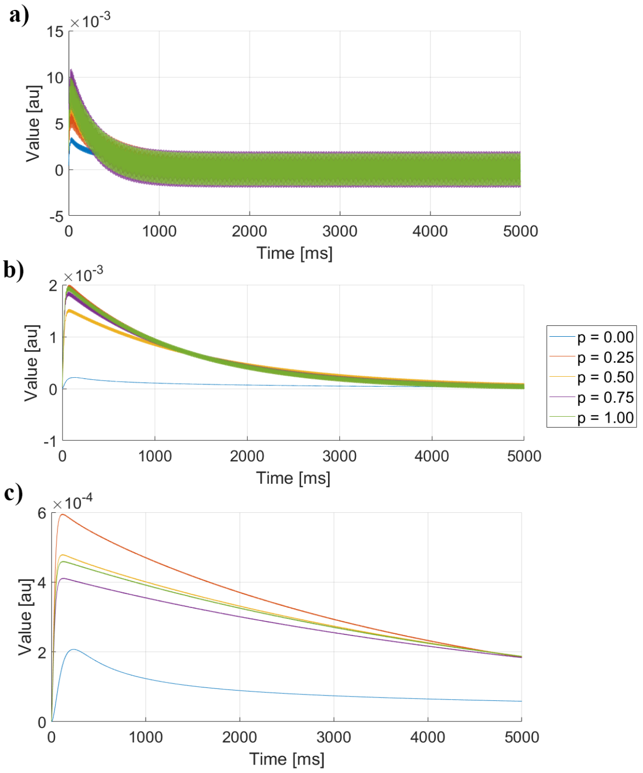


Figure 2: Mean value of all output layer nodes signal over each time step depending on the percolation parameter p. **a)** It corresponds to *NM 1*. **b)** Results for the *NM 2*. **c)** It belongs to *NM 3* simulations. All three plots have been obtained from networks with $K = 28$, $\beta = 0$ and $\alpha = 0$.

In this figure (2) one can see a *first time fill* phenomenon which is produced because nodes are *empty* ($A_i(0) = 0$) at the beginning of each simulation, like a water pipes system in a building. It implies that, at the firsts time-steps, nodes experience a significant value increase until the network is full of information, then it begins to oscillate at the same rate of the input signal. From this inflection point forward the network tends to replicate the input signal behaviour. Analysing *Figure 2* it is clear that the rapidity to reach an stable work rate depends on the number of layers but no significantly on the parameter p.

Figure 3 shows the activity for each output layer node.

In this plots we can observe the effects of p variation over the dispersion of nodes' activity.

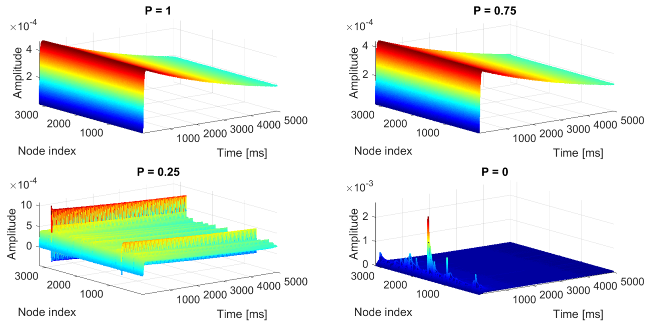


Figure 3: Activity of each node contained at the output layer in a 5-layer network with $K = 28$, $B = 0.4$ and $\alpha = 0$, depending on the chosen percolation parameter p.

We have computed $\overline{S}$ for the output layer of *NM 3* from all computations done without weights resampling. At *Figure 4* we show the results.

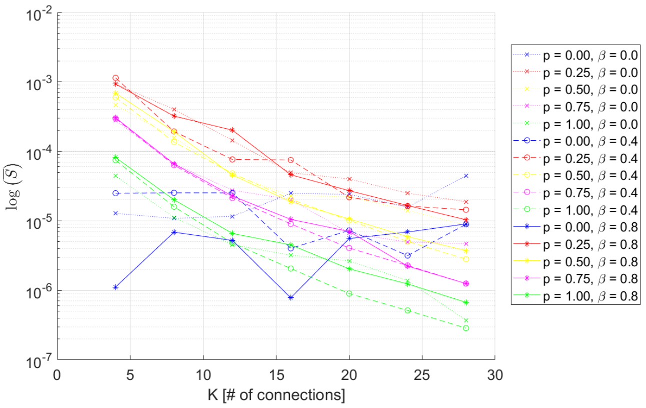


Figure 4: Mean of the standard deviation for the output layer ($l = 5$) in *NM 3* over all simulations performed without weights resampling and modifying p, β and K. The ordinate axis has been represented in logarithmic form to allow us appreciate the small differences for the highest K values.

From *Figure 4*we can see that the behaviour of $\overline{S}$ appears grouped by simulations with same p, and it is shown how $\overline{S}$ diminish when p or K increase. Also, we can observe how the effect of the β variation is more intense for higher K values, but it is not decisive. In the majority of cases, for the same p and K, the best β value for minimize $\overline{S}$ is $\beta = 0.4$. It means that *small-world* network properties contributes to the homogenization of the nodes' activity. With $p = 0$ we see that the behaviour is erratic for small K values, and begins to have the expected values when K reach the higher value. It would be explained due to the very small flux of information in simulations with small number of edges between layer neighbours and also with $w_{ij} \approx 0$. That is, if information doesn't flow the nodes doesn't change their values, so the dispersion between nodes' value would be small.
Once we have found this first relationships we have wondered if we can relate maximum (or minimum) values of $\overline{S}$ with the parameters. At *Figure 5* we have represented for all models the maximum (minimum) values of $\overline{S}$ found for all layers over all simulations, depending on K.

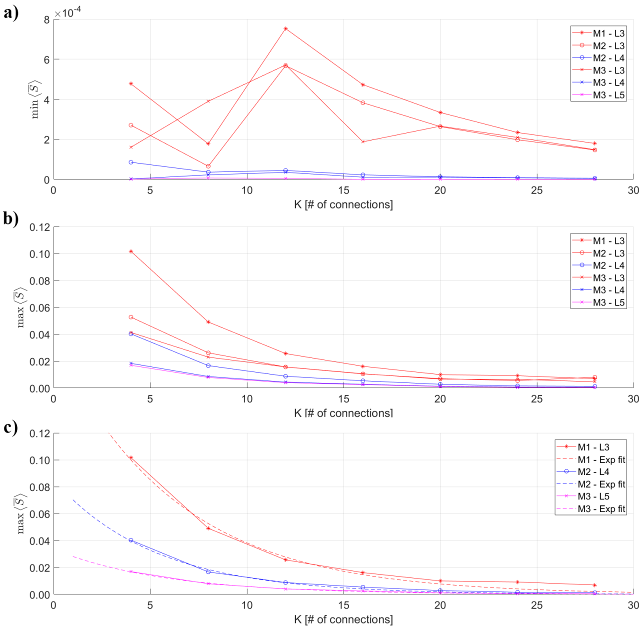


Figure 5: Analysis of the standard deviation extreme values over all simulations. All points represent the most favourable simulation results for the plotted variable depending on the model, the layer and the K value. **a)** Comparison of the minimum values. **b)** Maximum standard deviation mean values for each layer of each model. **c)** Only last (output) layer maximum standard deviation mean for each model. Data have been fitted with an exponential functions ($a \cdot e^{b \cdot x}$). The parameters of the adjusts are the following:
M1→ $a = 0.9891$, $b = 0.18 \pm 0.03$, $R^2 = -0.16 \pm 0.03$;
M2→ $a = 0.9934$, $b = 0.08 \pm 0.01$, $R^2 = -0.19 \pm 0.03$;
M3→ $a = 0.9972$, $b = 0.033 \pm 0.003$, $R^2 = -0.17 \pm 0.02$.

For the minimum values of $\overline{S}$ displayed at *Figure 5a* it is clear that the more layers has the network, the less is the dispersion between nodes. On the other hand, for the maximum values of $\overline{S}$ it is shown at *Figure 5b* how the number of the layer is clearly distinguishable because the smaller layer number the bigger value for $\max \overline{S}$. Also, one can observe how for the *NM 3* values are smaller for the same layer number compared with the other two models. It tells us that a layer between layers has less dispersion than if it were the output layer. We think that it is explained because of these layers have a double input from their both predecessor and successor layers. In *subfigure c* we can see how the relationship between K and $\max \overline{S}$ has an exponential like form, but it is clear that for the layer 3 of *NM 1* the fit doesn't adjust well the higher K values.

IV. CONCLUSIONS

Results have shown us that the variation of the information flux through the *NNs*, built according to the proposed model, depends on the studied parameters with different grades of intensity.

The total number of layers determines the amplitude of the output signal. It is weaker when L is bigger, because of the signal has to travel longer and it is speared in more nodes. L also contributes to make the response smooth as we have shown at *Figure 2*.
The number of connections among nodes in the same layer, K, commands the amplitude and the velocity of response of the layers to the inputs changes. When K increases, the response becomes smoother.
As we have expected, percolation parameter p has a modulating role in how much information is able to reach the output layer, and also in how the information is distributed inside the layers. With low p values it is possible that only an small part of the network contributes to the information propagation, which is compatible with the lost of conscience under anesthesia as it is proposed on [5]. It would be an interesting feature to study in further researches because it is possible to associate with multiple brain diseases, like alzehimer.
The Watts-Strogart parameter β doesn't seems to affect too much to the information flow, but it is significant how when the network has *small-world* attributes the dispersion between the same layer nodes is minimum. It would be related with the appearance of long range connections, that distribute more the information. In fact, the results on *Figure 5a* for the *NM 3* for the higher K values have been obtained from networks with $p = 1$ and $\beta = 0.4$.
Weights resampling introduce noise on the network. It makes that the stochastic processes master over the network properties. So, when α and p are small we found a chaotic information propagation. All the results of *Figure 5c* for *NM 3* have been obtained from networks with $\alpha = 4$ and $0 \leq p \leq 0.25$.
We think that this paper could be an starting point to apply this kind of analysis to trained *NNs* and study how their accuracy results change by changing their topological features.

V. APPENDIX

The source code of the developed *Matlab/Octave* program that builds the *NN* and performs the simulations is available under request to the author via e-mail.

Parameter	**Meaning**
L	Number of layers, including the input node.
T	Number of edge projections from each node to the following layer.
I_{nod}	Number of nodes at the 2nd layer.
K	Number of layer neighbours connected to each node before apply Watts-Strogartz algorithm.
β	Watts-Strogartz parameter.
p	Percolation parameter.
c	Memory parameter.
λ	Memory parameter.
α	Parameter for resampling periodicity.

Table I: Neural network parameters definition.

Acknowledgments

I would like to show my gratitude to my advisor, M. Ángeles Serrano, for the provided opportunity to work in a field that is among my personal interests. Also I would like to recognize all the support gathered from my family and friends along this thrilling journey through the physics degree. Thanks to all of you!.

[1] Jones EG. Cajal's debt to Golgi. Brain Research Reviews. 2011 Jan 7;66(1-2):83-91.

[2] TSODYKS, Misha V.; MARKRAM, Henry. The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. Proceedings of the National Academy of Sciences, 1997, vol. 94, no 2, p. 719-723.

[3] LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. nature, 2015, vol. 521, no 7553, p. 436.

[4] DJURFELDT, Mikael, et al. Brain-scale simulation of the neocortex on the IBM Blue Gene/L supercomputer. IBM Journal of Research and Development, 2008, vol. 52, no 1.2, p. 31-41.

[5] ZHOU, David W., et al. Percolation model of sensory transmission and loss of consciousness under general anesthesia. Physical review letters, 2015, vol. 115, no 10, p. 108103.

[6] RAIZADA, Rajeev DS; GROSSBERG, Stephen. Towards a theory of the laminar architecture of cerebral cortex: Computational clues from the visual system. Cerebral Cortex, 2003, vol. 13, no 1, p. 100-113.

[7] D. J. Watts and S. H. Strogatz, Collective dynamics of 'small-world' networks, Nature (London) 393, 440 (1998).