



João Pinheiro Neto

A STUDY ON THE STRUCTURE AND DYNAMICS OF
COMPLEX NETWORKS

ESTUDO SOBRE A ESTRUTURA E DINÂMICA DE REDES
COMPLEXAS

Campinas
2015



UNIVERSITY OF CAMPINAS
INSTITUTE OF PHYSICS GLEB WATAGHIN

João Pinheiro Neto

A STUDY ON THE STRUCTURE AND DYNAMICS OF COMPLEX NETWORKS

ESTUDO SOBRE A ESTRUTURA E DINÂMICA DE REDES COMPLEXAS

Dissertation presented to the Institute of Physics “Gleb Wataghin” of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Physics.

Dissertação apresentada ao Instituto de Física “Gleb Wataghin” da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Física.

Supervisor/Orientador: Prof. Dr. José Antônio Brum

Co-supervisor/Coorientador: Prof. Dr. Marcus Aloizio Martinez de Aguiar

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO
DEFENDIDA PELO ALUNO JOÃO PINHEIRO NETO E ORIENTADA
PELO PROF. DR. JOSÉ ANTÔNIO BRUM

Assinatura do Orientador(a)

A handwritten signature in blue ink, appearing to read "JAB".

Assinatura do Coorientador(a)

A handwritten signature in blue ink, appearing to read "MAM".

CAMPINAS
2015

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Física Gleb Wataghin
Valkíria Succi Vicente - CRB 8/5398

Pinheiro Neto, João, 1989-
P655s A study on the structure and dynamics of complex networks / João Pinheiro Neto. – Campinas, SP : [s.n.], 2014.

Orientador: José Antônio Brum.
Coorientador: Marcus Aloizio Martinez de Aguiar.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Física Gleb Wataghin.

1. Redes complexas. 2. Criticalidade auto-organizada. 3. Redes neurais. I. Brum, José Antônio, 1959-. II. Aguiar, Marcus Aloizio Martinez de, 1960-. III. Universidade Estadual de Campinas. Instituto de Física Gleb Wataghin. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Estudo sobre a estrutura e dinâmica de redes complexas

Palavras-chave em inglês:

Complex networks

Self-organized criticality

Neural networks

Área de concentração: Física

Titulação: Mestre em Física

Banca examinadora:

José Antônio Brum [Orientador]

Camilo Rodrigues Neto

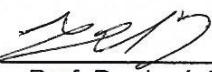
Gabriela Castellano

Data de defesa: 15-12-2014

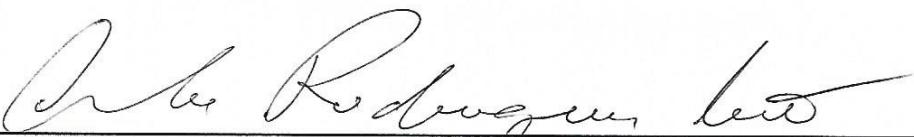
Programa de Pós-Graduação: Física

MEMBROS DA COMISSÃO JULGADORA DA DISSERTAÇÃO DE MESTRADO DE
JOÃO PINHEIRO NETO – R.A. 081750 APRESENTADA E APROVADA AO
INSTITUTO DE FÍSICA “GLEB WATAGHIN”, DA UNIVERSIDADE ESTADUAL DE
CAMPINAS, EM 15 / 12 / 2014.

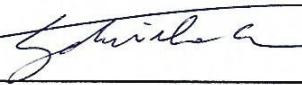
COMISSÃO JULGADORA:



Prof. Dr. José Antônio Brum
Orientador do Candidato - DFMC/IFGW/UNICAMP



Prof. Dr. Camilo Rodrigues Neto – EACH / USP



Profa. Dra. Gabriela Castellano – DRCC/IFGW/UNICAMP

Abstract

In this Masters Dissertation we study the structure and dynamics of complex networks. We start with a revision of the literature of complex networks, presenting the most common network metrics and models of network connectivity. We then study in detail the dynamics of the Random Threshold Network (RTN) model. We develop a new mean-field approximation for the RTN dynamics that is considerably more simple than previous results. This new approximation is useful from a practical standpoint, since it allows the generation of RTNs where the average activity of the network is controlled. We then review the literature of Adaptive Networks, explaining some of the adaptive models with interesting characteristics. At last, we develop two models of adaptive networks inspired by the evolution of neuronal structure in the brain. The first model uses simple rules and a link-removing evolution to control the activity on the network. The inspiration is the removal of neurons and neuronal connections after infancy. This model can also control the activity of individual groups within the same network. We explore a variant of this model in a bi-dimensional space, where we are able to generate modular and small-world networks. The second model uses external inputs to control the topological evolution of the network. The inspiration in this case is the development of neuronal connections during the infancy, which is influenced by interactions with the environment. The model generates finite avalanches of activity, and is capable of generating specific and modular topologies using simple rules.

Keywords: Complex networks, Self-organized criticality, Neural networks.

Resumo

Nesta dissertação de mestrado estudamos a dinâmica e estrutura de redes complexas. Começamos com um revisão da literatura de redes complexas, apresentando as métricas de rede e modelos de conectividade mais comuns. Estudamos então em detalhe a dinâmica do modelo das Random Threshold Networks (RTN). Desenvolvemos uma nova aproximação de campo médio para a dinâmica de RTNs, consideravelmente mais simples que aproximações anteriores. Esta nova aproximação é útil de um ponto de vista prático, pois permite a geração de RTNs onde a atividade média da rede é controlável. Fazemos então uma revisão da literatura de redes adaptativas, apresentando alguns modelos de redes adaptativas com características interessantes. Por fim, desenvolvemos dois modelos de redes adaptativas inspirados pela evolução da estrutura neuronal no cérebro. O primeiro modelo utiliza regras simples e uma evolução baseada na remoção de links para controlar a atividade sobre a rede. A inspiração é a remoção de neurônios e conexões neuronais após a infância. Este modelo também consegue controlar a atividade de grupos individuais dentro de uma mesma rede. Exploramos uma variante deste modelo em um espaço bi-dimensional, onde conseguimos gerar redes modulares e small-world. O segundo modelo utiliza inputs externos para controlar a evolução da topologia da rede. A inspiração neste caso é o desenvolvimento das conexões neuronais durante a infância, que é influenciado por interações com o ambiente. O modelo gera avalanches finitas de atividade, e é capaz de gerar topologias específicas e modulares utilizando regras simples.

Palavras-Chave: Redes complexas, Criticalidade auto-organizada, Redes neurais.

Contents

Abstract	vii
Resumo	ix
Acknowledgements	xiii
List of Figures	xxii
List of Symbols	xxiii
1 Introduction	1
2 Basics of Network Theory	3
2.1 Basic definitions	3
2.2 Measures of network topology	4
2.3 Community detection and partitioning	6
2.4 Network models	13
3 Theory of Random Threshold Networks	21
3.1 Definition	21
3.2 Dynamics of Random Threshold Networks	23
3.3 Annealed approximation	25
3.4 Phase transition of A_∞	31
3.5 Measures of damage spreading	33
4 Adaptive Networks	35
4.1 Definition	35
4.2 Complex dynamics and topology	36

4.3	Self-Organized Criticality in Boolean models	41
5	An Adaptive Model with Stable Dynamics	49
5.1	Motivation and model definition	49
5.2	Simulation results	52
5.3	Spatial embedding	54
5.4	Summary	59
6	A Model with Controllable Topology from External Inputs	61
6.1	Motivation and model definition	61
6.2	Activity avalanches from external inputs	64
6.3	Controlling the topology of a spatial network	67
6.4	Summary	74
7	Conclusion	77
Bibliography		80
Index		91

Acknowledgements

I thank FAPESP for supporting this work through the grants 2012/18550-4 and 2013/15683-6. I also thank CNPq for financing the initial portion of the project.

I thank Prof. Jose Antonio Brum, my advisor during the project. During our meetings and countless hours talking, Prof. Brum taught me much Physics. But more than that, he also taught me a lot about Science, and about life in general. He is one of my examples on how to be a researcher, and I am very grateful for the opportunity to work with him.

I thank Prof. Marcus A. M. de Aguiar, for accepting to be the co-advisor of this project. I also thank Professors Eduardo Miranda, Gabriela Castellano, Rickson Mesquita and Maurice de Koning for the feedback given at various stages of the project. I also thank Prof. Stefan Bornholdt, from the University of Bremen, for hosting me during a 5-month stay and helping to develop one of the models presented in this work. I also thank Dr. Tiago Peixoto for helping me in many aspects during my stay in Bremen, and for the feedback on the models.

I thank my friends and colleagues in Campinas, in particular César, Heitor, Kevin e Shadi. Through thick and thin, they were always there to help.

I thank Ana, for standing by my side and making me happier.

I thank my parents, for more than I can describe.

Agradecimentos

Agradeço à FAPESP, pelo apoio dado a este trabalho através dos processos 2012/18550-4 e 2013/15683-6. Agradeço também ao CNPq pelo financiamento durante o período inicial do projeto.

Agradeço ao Prof. Jose Antonio Brum, meu orientador durante o projeto. Durante as reuniões e horas incontáveis conversando, o Prof. Brum me ensinou muita Física. Mas mais que isso ele me ensinou muito sobre Ciência, e sobre a vida como um todo. Ele é um dos meus exemplos de como ser pesquisador, e sou muito grato pela oportunidade de trabalhar com ele.

Agradeço ao Prof. Marcus A. M. de Aguiar, por ter aceito ser co-orientador deste projeto. Agradeço também aos professores Eduardo Miranda, Gabriela Castellano, Rickson Mesquita e Maurice de Koning pelo feedback dado ao trabalho em diferentes etapas deste. Agradeço ao Prof. Stefan Bornholdt, da Universidade de Bremen, por ter me recepcionado durante uma estadia de 5 meses e ter ajudado a desenvolver um dos modelos apresentados neste trabalho. Agradeço ao Dr. Tiago Peixoto por ter me ajudado em vários aspectos durante a minha estadia em Bremen, e pelo feedback nos modelos.

Agradeço aos meus amigos e colegas de Campinas, em particular César, Heitor, Kevin e Shadi. Eles sempre estiveram lá para me ajudar, nos bons e maus momentos.

Agradeço à Ana, por estar ao meu lado e me fazer mais feliz.

Agradeço aos meus pais, por mais do que eu posso descrever.

List of Figures

2.1 Examples of (a) an undirected unweighted network and its adjacency matrix A , and (b) a directed weighted network and its weight matrix W . Reproduced from [11].	3
2.2 The Zachary karate club network is an example of a small network. The color and node shapes represent a division of the network in two communities. Reproduced from [5] (left). An human protein interaction network is an example of a large network. Reproduced from [12] (right).	4
2.3 Example of a network division in three communities. The division minimizes the number of inter-community links. Reproduced from [16] (left). Example of a dendrogram, showing the hierarchical community division of a network. The circles represent nodes, and the tree represents the community membership of the nodes at different levels, from bottom (single nodes) to top (one community). Reproduced from [22] (right).	7
2.4 Visualization of the Louvain algorithm. At each pass the network is re-sized, with the new nodes being the communities of the previous network. Inside each pass the modularity is optimized. Reproduced from [32].	12
2.5 Size of the largest connected component in Erdos-Renyi networks. . .	14
2.6 Examples of real-world networks with power-law degree distributions $p_k \sim k^{-\gamma}$. (A) Actor collaboration network, with $\gamma_{actor} = 2.3$. (B) World Wide Web, with $\gamma_{www} = 2.1$. (C) Power grid network, with $\gamma_{power} = 4$. Reproduced from [35].	15

2.7	Network robustness against random failure (blue squares) and targeted attack (red circles), by removing a fraction f of the nodes. S (empty points) is the size of the largest cluster in the network, and $\langle s \rangle$ (filled points) is the average size of the other clusters. The metrics are calculated for (a) ER networks and (b) Barabási-Albert networks. Both networks have the same average degree. Adapted from [42].	17
2.8	Graphical representation of the Watts-Strogatz model. By varying p the model is able to go from an ordered ring network ($p = 0$) to a random network ($p = 1$), with a small-world network for intermediate values. Reproduced from [14].	18
2.9	Normalized average path length $L(p)/L(0)$ and clustering coefficient $C(p)/C(0)$ for varying shuffling probability p . Reproduced from [14].	19
3.1	Network active fraction for Barabási-Albert (left) and Erdos-Rényi (right) topologies, with many initial conditions $A(0)$. The parameters are $n = 10^4$, $K = 50$, $h = 1$ and $F_+ = 0.45$ for both cases.	24
3.2	Stable activity A_∞ color map for simulation data with $h = 0$ (left) and $h = 1$ (right). The other parameters are $n = 10^4$ and $A(0) = 0.1$ for both cases. Each of the 900 data points is an average of 10 networks.	24
3.3	Minimum number of active nodes nA_0^C (left) so that $A_\infty > 0$ and the corresponding A_∞ (right). If the network could not be activated with any $A(0)$ we set $A_\infty = 0$. Parameters are $n = 500$ and $h = 1$	25
3.4	Stable activity $A_\infty(K, F_+)$ color map of the mean-field approximation for $h = 0$ (left) and $h = 1$ (right).	29
3.5	Comparison between the theoretical prediction and simulation results with fixed degree K for $h = 0$ (left) and $h = 1$ (right).	29
3.6	Comparison between the theoretical prediction and simulation results ($n = 1000$) with a fixed positive fraction F_+ for $h = 0$ (left) and $h = 1$ (right).	30
3.7	First-order phase transition of A_∞ for $h = 1$ (left) and $h = 2$ (right). The other parameters are $n = 1000$, $K \in \{10, 20, 30\}$ and $A(0) = 0.5$. Each curve is the result of 10^4 simulations.	31
3.8	Minimum activity A_∞^{min} for any F_+ . The parameters are $10 < K < 200$ and $n \in \{200, 500, 1000\}$. The threshold is $h = 0$ and every point is the average of 20 simulations.	32

3.9	Minimum activity A_∞^{min} for any F_+ . The parameters are $10 < K < 200$ and $n = 200$. The threshold is $h \in \{1, 2, 3\}$ (left) and $h \in \{4, 5, 6\}$ (right). Every point is the average of 100 simulations.	32
4.1	Evolutionary mechanism of an adaptive network, showing the interplay between topology and dynamics. Reproduced from [64].	36
4.2	Degree distribution p_k of the SIS adaptive model. The black dots are for the infected, and the circles for the susceptible nodes. Model with random rewiring independent of state (left). Model without infection and recovery ($p = r = 0$) (center). Adaptive network model with $w = 0.3$, $r = 0.002$ and $p = 0.008$. Other parameters are the number of nodes $n = 10^5$ and average degree $K = 20$. Adapted from [66].	38
4.3	Illustration of the co-evolution of the model by Aoki <i>et al</i> (a). Complementary cumulative distribution of the resource x_i (b), link weights w_{ij} (c) and strength s_i (d) for the evolved network. The underlying network is an ER network with $n = 16384$ and $K = 10$. Other parameters are $\alpha = 1$, $\kappa = 0.05$, $D = 0.34$ and $\epsilon = 0.01$. Reproduced from [67]	40
4.4	Network evolution for $\kappa = 0$ in the model by Aoki <i>et al</i> . Simulation of the resource distribution as a function of α , with $n = 512$, $K = 5$ and $\alpha \in \{0.5, 1.25\}$ (a). Consequences on the topology for $\alpha \in \{0.75, 1, 1.25\}$. The other parameters are $n = 16384$, $K = 10$, $D = 0.02$ and $\epsilon = 0.01$. Reproduced from [67].	41
4.5	Evolution of the average degree K_{ev} with $N = 1024$ and varying initial degree $K_{ini} \in \{1.5, 3.0\}$ (left). Scaling of K_{ev} as a function of N (right). Reproduced from [73].	42
4.6	Frozen component C of the dynamics on a random network, as a function of the number of nodes N and average degree K . Each point is an average of 10^3 simulations. Reproduced from [73].	43
4.7	Time evolution of the network in the model by Rybarsch & Bornholdt. The parameters are $\beta = 5$ and $\Theta_i = 0 \forall i$. The initial network is a random network with average degree $K_{ini} = 0$ (left) and $K_{ini} = 4$ (right). Reproduced from [75].	45

4.8	Avalanches of activity in the model by Rybarsch & Bornholdt. The avalanches were created by activating a single node on the evolved network. (A) Cumulative frequency of the avalanche size S . (B) cumulative frequency of the avalanche duration T . Results from 10^5 avalanches on 10 different networks of $n = 1024$ nodes. Parameters are $\beta \rightarrow \infty$ and $\Theta_i = 0 \forall i$. Reproduced from [75].	46
4.9	Network response to a change in threshold in the model by Rybarsch & Bornholdt. The network is evolved with a threshold $\Theta_i = 1$, resulting in a higher degree $\langle K \rangle$. A sudden change in the threshold to $\Theta_i = 0$ causes the network to respond, decreasing $\langle K \rangle$ and stabilizing in the new critical state. Reproduced from [75].	47
5.1	Representation of the membrane electrical potential of a neuron. The neuron requires stimuli above a certain minimum threshold in order to fire. Reproduced from the Wikimedia Commons file “File:Action potential.svg”. Available at http://commons.wikimedia.org/wiki/File:Action_potential.svg	50
5.2	Schematic representation of the adaptive algorithm. A node i loses a positive in-link w_{ij} if its average activity $\langle \sigma_i \rangle$ is higher than α . Otherwise, i loses a negative in-link.	52
5.3	Network activity for initial $F_+(0) \in \{0.1, 0.5, 0.9\}$ and $\alpha \in \{0.25, 0.75\}$. Each point is the average value of $A(t)$ during the cycle of $t_0 + \tau$ time steps. (top) Corresponding evolution in the parameter space. The arrows indicate the direction of the evolution. (bottom) Other parameters are $h = 0$, $n = 1024$, and $A(0) = 0.1$	53
5.4	Network activity for many $\alpha_i \in \{0, 0.25, 0.75, 1\}$ in the same network, with $n = 1000$, $F_+(0) = 0.5$, $K_0 = 200$ and $K = 50$	54
5.5	Topological evolution for a network without spatial constraints (left) and with spatial constraints (right). The parameters are $\alpha = 0.8$, $n = 1000$, $K_0 = 100$ and $K = 10$	56
5.6	Evolved topology of a network and measures of it. The color of each node corresponds to its assigned module. The parameters are $n = 1000$, $\alpha = 0.8$, $\beta = 10$ and $K = 8.35$	57
5.7	Modularity Q for the evolved and randomized networks (left) and relative average path length $L(\beta)/L(0)$ and clustering $C(\beta)/C(0)$ (right). The other parameters are $n = 1000$, $\alpha = 0.1$ and $K = 10$	58

5.8	Standard deviations of $A(t)$ (left) and $\langle \sigma_i \rangle$ (right) for the evolved and Erdos-Renyi networks with the same activity, as a function of α . Parameters are $n = 500$, $K = 20$ and $K_0 = 100$ for the evolved network.	59
6.1	Density of synapses in a portion of the human brain. Samples from 21 different brains. Reproduced from [7].	61
6.2	Schematic representation of the adaptive algorithm.	63
6.3	Evolution of a network with a fixed input pattern. Topological measures and average activity (left) and dynamical measures (right). Parameters are $n = 1024$, $n_A = 0.01$ and $\Theta_0 = 0$	64
6.4	Avalanche probability distribution from random perturbations on an evolved network. Data from 10^6 avalanches. (left) Derrida plot of the network avalanches. Data from 10^4 avalanches, perturbed at different time steps (right). The network was evolved with $n = 1024$, $n_A = 10^{-3}$ and $\Theta_0 = 0$	65
6.5	Hamming distance for the original network and randomized networks for $t = T_h \in \{1, 5\}$ (left) and Hamming distance divided by the network activity with full input at $t = T_h$ (right). Each point is an average of 10^4 runs for the original network and 3.10^3 runs for each of the 20 random networks.	66
6.6	Example of a network evolved with spatial embedding. The network was evolved by activating a small number of nodes at the center. The node colors corresponds to the community division found by maximizing the modularity. Parameters are $n = 1000$, $\Theta = 1$ and $\beta = 10$	68
6.7	Dynamics of the evolved network. The red points represent activated nodes, and the black points represent inactive nodes. Dynamical states at $t = 1$, showing the nodes with external input (left), and at $t = 8$, showing how the activity spreads through the network.	69
6.8	Network evolution as a function of β . In the upper figures we have the degree K and avalanche length (upper left), and average correlation C , average activity during the avalanche and average final $\Theta = \langle \Theta_i \rangle$ (upper right). In the lower figures we have the relative clustering C/C_r and relative path length L/L_r (lower left), and the modularity Q for the evolved and randomized networks (lower right). Parameters are $n = 300$, $\Theta_0 = 1$ and 6 initial input nodes ($n_A = 0.02$).	70

6.9	Derrida plot of the evolved network, for various β . Both networks have $\Theta_0 = 1$. Results for an ensemble of networks with $n = 300$ and $n_A = 0.02$ (left) and $n = 512$ and $n_A = 0.012$ (right).	71
6.10	Topology of a network evolved with three input regions at coordinates $(0, 0)$, $(1, 0)$ and $(0.5, 1)$. The colors represent the community detection found by a modularity maximization method. Parameters are $n = 1000$, $\Theta_0 = 1$ and $n_A = 0.002$ for each site.	72
6.11	Topology of the evolved network with three input patterns at three different stages. The temperature (color) represents the degree of a node. Topology at $t = 1000$ (left), $t = 3000$ (center) and $t = 6000$ (right).	72
6.12	Avalanche dynamics of the evolved network starting from the input site at $(0.5, 1)$. Dynamics at $t = 1$ (left), during the maximum activity at $t = 6$ (center) and at $t = 9$ (right).	73
6.13	Dynamics of the evolved network starting from the input site at $(1, 0)$. Dynamics at $t = 1$ (left), during the maximum activity at $t = 7$ (center) and at $t = 9$ (right).	73
6.14	Avalanche dynamics of the evolved network starting from the input site at $(0, 0)$. Dynamics at $t = 1$ (left), during the maximum activity at $t = 6$ (center) and at $t = 9$ (right).	74

List of Symbols

$A(t)$ Fraction of nodes active at time t

A_∞ Stable activity of the network

F_+ Fraction of excitatory links in the network

H Hamming distance

K Average degree of a network

m Number of links in the network

n Number of nodes in the network

Q Modularity

BA Barabási-Albert

ER Erdős-Rényi

WS Watts-Strogatz

”The truth may be puzzling [...]. It may not be consonant with what we desperately want to be true. But our preferences do not determine what’s true.”

Carl Sagan

”The first principle is that you must not fool yourself - and you are the easiest person to fool.”

Richard Feynman

Introduction

We live in a time of vast amounts of data. Genome databases have the mapped DNA of over 10000 species[1], while Twitter had an average of 340 million messages sent per day in 2012[2]. Many of these data sets are formed by discrete agents that interact between them, such as people in a social network. Network Science (also known as Complex Networks) emerged as a way to study and seek hidden patterns in these systems[3, 4, 5]. It borrows concepts from many fields of knowledge, including Physics, Mathematics and Computer Science. As such, it is an interdisciplinary field with applications in a wide variety of areas, ranging from Genetics to Sociology.

Besides studying the *structure* of a network, Network Science also studies the *dynamics* on a network. In most cases, agents in the network have some behavior besides interacting to one another. For instance, people can be healthy or sick in an epidemiological network, and spins can be up or down in a spin network. The topic of dynamical processes on networks aims the study of dynamical models and how the underlying network structure affects these models. This has parallels in the real-world. To cite an example, the spreading of an airborne disease on a city like Hong Kong is different from the spreading in a sparse, rural community. In other words, how the agents interact (i.e., the structure of the network) tends to have a strong effect on the dynamics of the system. In some cases, there is also a change in the structure of the network that is influenced by the dynamics. This is the object of study of a subfield of Network Science, called Adaptive Networks.

Neuroscience is an interdisciplinary field of study concerned with understanding the nervous system[6]. It has improved our knowledge of the human brain dramatically, specially in the last few decades. The basic functional unit of the brain is the neuron, which can be active (firing) or inactive. The neuron interacts with other neurons using connections called synapses. One important discovery from

Neuroscience is how the brain develops. After birth, the newborn child's brain goes through a synaptic growth phase. During this phase, the number of synapses increases dramatically. After it the number of synapses diminishes, in a process called synaptic pruning[7, 8, 9].

The focus of this Dissertation is twofold: to review some of the current understanding of networks, focusing on both dynamics and structure, and present our contributions to the field of Complex Networks. These contributions are in the form of a study of the dynamical model called Random Threshold Network, and two adaptive network models. Our two models are inspired by the development of the brain. With them, we attempt to control certain aspects of network dynamics and structure while using simple rules and a minimal number of parameters. Our first model is inspired by the synaptic pruning in the brain, while the second model is influenced by the synaptic growth.

The organization of this Dissertation is the following. In Chapter 2 we review the basics of Network Theory, presenting the most common metrics used to characterize the structure of a network. We pay special attention to the problem of community detection in a network. We also review the most common models of network structure. In Chapter 3 we explore the Random Threshold Network dynamical model. We focus on the balance between excitatory and inhibitory links, and develop a mean-field approximation for the dynamics. In Chapter 4 we present an introduction to the field of Adaptive Networks, and explain some of the common properties found in adaptive models. We do this by briefly reviewing models that exemplify these properties. In Chapter 5 we present the first of our adaptive models. In this model, we are able to control the activity of a network using simple rules. We also present a simple rule capable of controlling certain structural properties in spatial networks. In Chapter 6 we present our second model, where external inputs are used to control the growth of a network. Finally in Chapter 7 we have the conclusion of this Dissertation.

Basics of Network Theory

2.1 Basic definitions

A *network*, also called a graph in mathematical literature, is a representation of a set of n objects, called *nodes*, that interact pairwise[10]. The interaction between two of these nodes is called a *link*, and we denote the link between nodes i and j by a_{ij} . We denote by m the number of links in the network. If the interaction between the nodes is symmetric then $a_{ij} = a_{ji}$ and we call the network *undirected*. If the interaction is directed - such as in a network of computers where information is sent from one computer to another - then the network is directed and we can have $a_{ij} \neq a_{ji}$. In this case, we denote by a_{ij} the interaction from node j to node i . This notation is used in order to simplify calculations. It is worth noting that some authors use a reversed notation, with a_{ij} denoting the link from i to j . The networks in Fig. 2.1 are such examples. The structure (or topology) of the network can be encoded in the *adjacency matrix* \mathbf{A} , whose elements are the links a_{ij} .

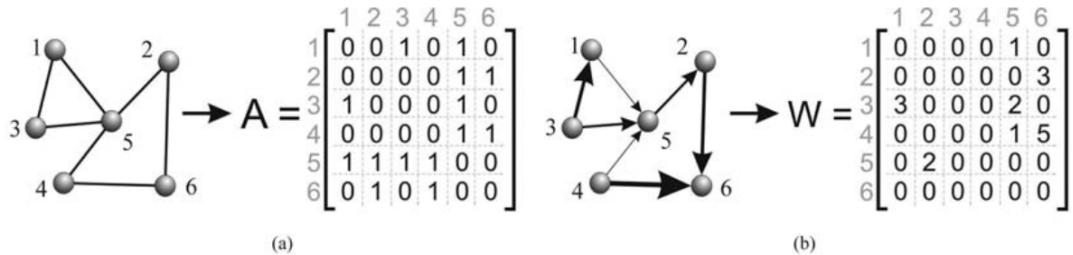


Figure 2.1: Examples of (a) an undirected unweighted network and its adjacency matrix \mathbf{A} , and (b) a directed weighted network and its weight matrix \mathbf{W} . Reproduced from [11].

A second classification category is regarding the value of a_{ij} . We call the network

unweighted when $a_{ij} = 1$ if the link exists and $a_{ij} = 0$ if it does not. If the link possesses a different value than zero and one, then we call it a *weighted network* and denote the link between i and j by w_{ij} . The matrix \mathbf{W} whose elements are the weights w_{ij} is called the *weight matrix*. One example of a weighted network is a power grid network, where nodes represent power stations and links represent transmission lines. The transmission lines generally have different capacities, so different weights can be assigned to links to represent different transmission capacities.

2.2 Measures of network topology

Small networks can be analyzed visually, for instance by representing the nodes with points and the links with lines joining the connected nodes. Large networks, however, require the use of *network metrics* in order to be characterized. This contrast can be seen in Fig. 2.2. In this section we describe some of the metrics and techniques used to study the topology of a network.

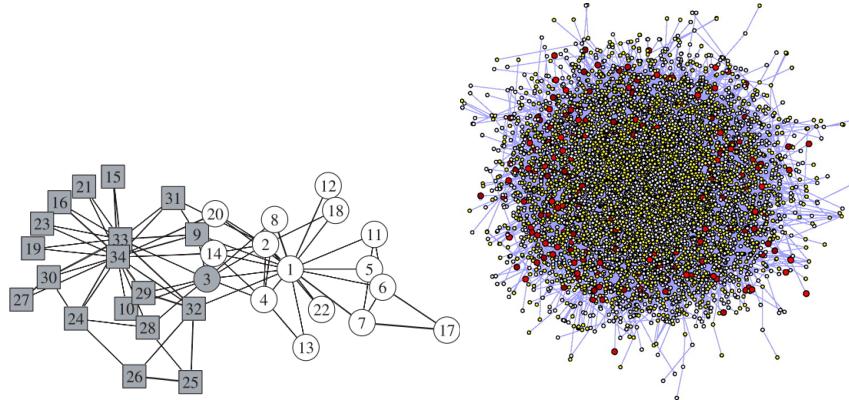


Figure 2.2: The Zachary karate club network is an example of a small network. The color and node shapes represent a division of the network in two communities. Reproduced from [5] (left). An human protein interaction network is an example of a large network. Reproduced from [12] (right).

One of the most basic metrics is the *degree*. The degree k_i of a node i is defined as the number of connections ($a_{ij} \neq 0$) the node possesses, calculated as $k_i = \sum_j a_{ij}$ for an undirected network. For a weighted network, we define the node *strength* as $S_i = \sum_j w_{ij}$. For a directed network, we define the *in-degree* as the number of links a node receives (given by $k_i^{in} = \sum_j a_{ij}$), and the *out-degree* is the number of links it sends away (given by $k_i^{out} = \sum_j a_{ji}$). The *average degree* K in a network with n nodes is given by

$$K = \frac{1}{n} \sum_{i,j} a_{ij} \quad (2.1)$$

and is the average number of connections per node in the network. Since every link needs to end somewhere, there is only one average degree in directed networks, as opposed to an “average in-degree” and “average out-degree”. The total number of links in the network is given by $m = nK/2$ for undirected networks and $m_{dir} = nK$ for directed ones. Besides the average degree, another important metric is the *degree distribution*. The degree distribution p_k is the probability distribution of a random node having its degree equal to k . In particular, a network with a degree distribution of the form $p_k \sim k^{-\gamma}$ (called a power-law) is labeled a *scale-free network*.

The degree is a local measure, in the sense that it gives information about the properties of individual nodes. We may also be interested in how these nodes are organized. A question one may ask is if the neighbors of a node are also neighbors of each other. This question is encoded in the *clustering coefficient*. The local clustering coefficient c_i is defined as

$$c_i = \frac{(\text{number of triangles connected to node } i)}{(\text{number of triples centered on node } i)} \quad (2.2)$$

where a triple is any selection of three nodes and a triangle is a triple that forms a loop. The clustering coefficient C is defined as the average of c_i

$$C = \langle c \rangle = \frac{1}{n} \sum_{i=1}^n c_i \quad (2.3)$$

In other words, the clustering coefficient measures how tightly grouped nodes are in the network. In terms of the adjacency matrix, C can be calculated as

$$C = \frac{1}{n} \sum_{i=1}^n \left[\frac{1}{k_i(k_i - 1)} \sum_{j,m} a_{ij} a_{jm} a_{mi} \right] \quad (2.4)$$

While simple, the clustering coefficient is a widely used measure, and many real-world networks have been shown to possess a high clustering[3].

Another reasonable question is how far apart two nodes i and j are. This is measured by the shortest path length L_{ij} , defined as the minimum number of nodes (or *distance*) one needs to go through in order to reach node j starting from node i . For large networks, we are interested in the *characteristic path length* $L = \langle L_{ij} \rangle$, taken as the average distance¹ between any two nodes. A low L means that, starting from

¹By “distance” we always mean the number of nodes one needs to pass through in order to go from one node to another. For the physical distance between nodes embedded into a 2D or 3D space we will use the term “Euclidean distance”.

a random node, most of the network is easily reachable. Thus, the characteristic path length is a measure of network integration[13]. There must be a relationship between L and the network size n , with L increasing as n increases. If the characteristic path length scales as $L \sim \ln(n)$ or slower, we call the network a *small-world network*. In small-world networks the distance between two nodes is small, even if the network is very large. The small-world property has been observed in a wide variety of real-world networks, from power grid networks[14] to brain networks[15].

2.3 Community detection and partitioning

In most networks nodes are organized in groups, also known as *communities*. While the clustering coefficient gives an idea of how tightly connected the nodes are, it does not describe the size and composition of the groups in the network. The problem of community detection and partitioning remains one of the most active topics in network science, and many methods have been proposed to address it[16, 17, 18, 19]. Communities can be overlapping, when a node can be member of two or more communities at the same time, or non-overlapping[20, 21]. The number of communities in the network can be known or unknown. Some methods may also yield better results than others, but may not be computationally feasible for large networks. Therefore, the choice of algorithm to be used depends on which aspect of community division one wants to describe.

We can subdivide the problem of community division in two, depending on whether we know the number of communities *a priori*. It is called *graph partitioning* if the number of communities is known, and it is a classic problem in computer science. The idea is to divide the nodes in non-overlapping groups and minimize the number of links between groups. One example of application is parallelizing a computation. In a computer with many processing units (or cores), a task can be divided into sub-tasks and distributed among the cores. We can represent this with a graph, where nodes are the sub-tasks and links are the interactions between the sub-tasks. Communication between the cores is usually slow, which defeats the purpose of parallelization. Therefore we want to minimize the number of inter-community links in our graph representation, which can be accomplished with a variety of partitioning algorithms[10].

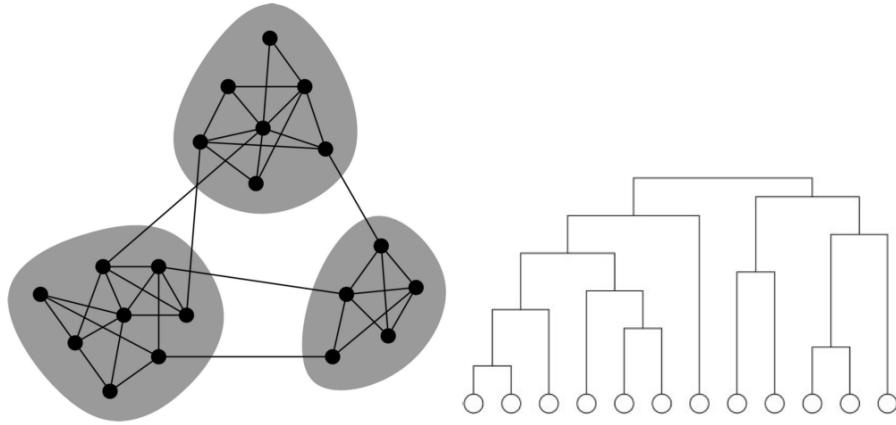


Figure 2.3: Example of a network division in three communities. The division minimizes the number of inter-community links. Reproduced from [16] (left). Example of a dendrogram, showing the hierarchical community division of a network. The circles represent nodes, and the tree represents the community membership of the nodes at different levels, from bottom (single nodes) to top (one community). Reproduced from [22] (right).

If we do not have prior information on the expected number of communities, we call it *community detection*. At the center of every community detection algorithm is an implicit definition of what constitutes a community. These community definitions are not equivalent, making community detection a less well-defined problem than graph partitioning. Let us first define the possible outputs of these algorithms. A community detection algorithm can give two types of output: the community membership(s) of each node, and/or a dendrogram expressing the hierarchical community division of the network (see Fig. 2.3). Community detection is generally considered a NP-hard problem[23], meaning that it cannot be solved exactly except for very small networks. Therefore, most algorithms of community detection are approximation, or heuristic, algorithms.

The first class of algorithms we will describe is the one based on *betweenness centrality* (BC)[24]. The concept of BC refers to how central a network element is regarding a certain property. For instance, the shortest-path betweenness b_i of a node i is defined as

$$b_i = \frac{1}{(n-1)(n-2)} \sum_{\{j,h\} \neq i} \frac{\rho_{hj}(i)}{\rho_{hj}} \quad (2.5)$$

where ρ_{hj} is the number of shortest paths between nodes h and j , and $\rho_{hj}(i)$ is the number of such paths that pass through node i . In other words, the shortest-path betweenness measures how central a node is regarding paths in the network. While shortest-path betweenness is the most used metric in BC algorithms, some variants

use metrics such as random-walk betweenness[25] and current-flow betweenness[26]. The BC algorithm is the following: we calculate the betweenness metric for each link in the network, and remove the link with the largest betweenness. The process is iterated until all links are removed. This eventually starts breaking the network into isolated components, generating a dendrogram of communities. The implicit idea is that inter-community links are important in carrying information in the network and possess large betweenness. By removing them we are then easily able to identify the communities. Betweenness centrality algorithms tend to yield good results but are slow, with a time complexity of $O(mn(m + n))$ ².

Another class of algorithms is the one based on *hierarchical clustering*. These algorithms use a definition of similarity between nodes, such as cosine similarity[23]. The algorithm starts with every node in its own group. The two most similar groups are joined together, and this is iterated until there is only one group. Therefore, hierarchical clustering methods also yield a dendrogram of community structure. Considering a pair of nodes from two different groups, there are three ways to define the similarity between groups: the similarity of the *most* similar pair (“single linkage”), the *least* similar pair (“complete linkage”), and the *average* of all pairs (“average linkage”). The time complexity of the hierarchical clustering depends heavily on the cost to compute the similarity. For networks embedded into space, where the Euclidean distance can be used, the algorithm goes as $O(n^2) \sim O(n^2 \log n)$. If the network is not embedded, however, the time complexity can go much higher[23].

The last class of algorithms we will describe is the one based on *modularity*. Modularity[16] can be viewed as a fitness function of a particular community partition, and is one of the most used metrics of community detection. Its basic idea is that, in a network with communities, most links should join members of the same group. Modularity differs from graph partitioning since it does not simply minimize the number of inter-community links. Instead, it compares the number of links inside the communities with what would be expected by chance. The modularity Q is defined as

$$Q = (\text{fraction of links within communities}) - (\text{expected fraction of such links}) \quad (2.6)$$

In other words, for every pair of nodes $\{i, j\}$ in the same community, Q calculates the fraction of links $\sum_{i,j} a_{ij}/2m$ in the community and subtracts from it what would

²The time complexity measures how the computational time of an algorithm scales with the size of its inputs. In this case, a network with the same average degree as another but with twice the nodes will demand around nine times the calculation time.

be expected for a “null model” network. For an undirected randomized network³, the probability of a link between two nodes i and j is given by $k_i k_j / 2m$. Therefore, the modularity Q can be written as

$$Q = \frac{1}{2m} \sum_{i,j} \left(a_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \quad (2.7)$$

where $\delta(c_i, c_j) = 1$ if the communities c_i of i and c_j of j are the same, and $\delta(c_i, c_j) = 0$ otherwise. It can be shown that $Q \in [-\frac{1}{2}, 1]$, and we expect $Q \approx 0$ for a community division on a network without community structure. One possible extension for directed networks is given by[17]

$$Q_{dir} = \frac{1}{m} \sum_{i,j} \left(a_{ij} - \frac{k_i^{in} k_j^{out}}{m} \right) \delta(c_i, c_j) \quad (2.8)$$

where the normalization in Eq. 2.7 and 2.8 follows from the fact that $\sum_{i,j} a_{ij} = 2m$ in undirected networks but $\sum_{i,j} a_{ij} = m$ in directed ones. Finally, a straightforward extension of modularity for weighted networks is given by[27]

$$Q_{weight} = \frac{1}{2\omega} \sum_{i,j} \left(w_{ij} - \frac{w_i w_j}{2\omega} \right) \delta(c_i, c_j) \quad (2.9)$$

where $w_i = \sum_j w_{ij}$ is the strength of node i and $\omega = \sum_{i,j} w_{ij}/2$ is the total network strength.

In order to use modularity to study network structure, one must find the community division that maximizes modularity. This can be accomplished through a large number of heuristic algorithms. Examples include algorithms based on simulated annealing[28], extremal optimization[29], genetic algorithms[30], greedy algorithms[31] and an adaptation of the Kernighan-Lin algorithm[16]. Here we will describe two algorithms, one that is based on linear algebra and easily deployed, and a fast algorithm used in large networks.

The linear algebra algorithm is called the spectral modularity maximization (SMM) algorithm, and it can be used to divide a network into two modules. Let us first define the modularity matrix \mathbf{B} , whose elements B_{ij} are

$$B_{ij} = a_{ij} - \frac{k_i k_j}{2m} \quad (2.10)$$

³More specifically, we use the configuration model[17]. In this model, every link is divided in two stubs, and the pair of every stub is then randomized. Thus, the configuration model generates a randomized version of a network while preserving the degree distribution.

From Eqs. 2.7 and 2.10, we can write

$$Q = \frac{1}{2m} \sum_{i,j} B_{ij} \delta(c_i, c_j) \quad (2.11)$$

Now let us define the group vector s with

$$s_i = \begin{cases} +1 & , \text{ if node } i \text{ belongs to module 1} \\ -1 & , \text{ if node } i \text{ belongs to module 2} \end{cases} \quad (2.12)$$

We can write the Kronecker delta $\delta(c_i, c_j)$ as $\delta(c_i, c_j) = \frac{1}{2}(s_i s_j + 1)$ and rewrite Eq. 2.11 as

$$Q = \frac{1}{4m} \sum_{i,j} B_{ij} (s_i s_j + 1) = \frac{1}{4m} \sum_{i,j} B_{ij} s_i s_j \quad (2.13)$$

where we use the fact that $\sum_i B_{ij} = \sum_j B_{ij} = 0$. In matrix form, Eq. 2.13 is given by

$$Q = \frac{1}{4m} s^T \mathbf{B} s \quad (2.14)$$

and therefore we must find the vector s that maximizes Q . For that, we use a relaxation method. Instead of $s_i = \pm 1$, we allow s_i to take any real value, with the restriction that $s^T s = \sum_i s_i^2 = n$. Geometrically, we are allowing s to point anywhere in space, instead of just the corners of a n -dimensional hypercube. Using a Lagrange multiplier β' , we can write

$$\frac{\partial}{\partial s_i} \left[\sum_{j,k} B_{jk} s_j s_k + \beta' \left(n - \sum_j s_j^2 \right) \right] = 0 \quad (2.15)$$

and obtain

$$\sum_j B_{ij} s_j = 2\beta' s_i \quad (2.16)$$

$$\mathbf{B} s = \beta s \quad (2.17)$$

where we define $\beta = 2\beta'$ for simplicity. From Eq. 2.17 we see that s is an eigenvector of \mathbf{B} . Since \mathbf{B} is Hermitian, the corresponding eigenvalue β is real. Applying Eq. 2.17 to Eq. 2.14 we have

$$Q = \frac{1}{4m} s^T \beta s = \frac{n}{4m} \beta \quad (2.18)$$

and therefore we reduce the problem of maximizing Q to finding the largest eigenvalue β_1 of \mathbf{B} . Note that Eq. 2.18 does not give the modularity of the network, as

$s_i = \pm 1$ in the real network. However we can approximate the answer by choosing a vector s as close to the corresponding eigenvector u_1 of β_1 as possible. This is done by maximizing the product $s^T u_1$, which can be done by choosing s_i as

$$s_i = \begin{cases} +1 & , \text{if } [u_1]_i > 0 \\ -1 & , \text{if } [u_1]_i < 0 \end{cases} \quad (2.19)$$

where $[u_1]_i$ is the i th element of u_1 . This solves the problem, as it gives us the community membership of each node after the bisection. The value of Q can then be calculated from Eq. 2.14. The algorithm can be adapted to find more than two communities by a process of repeated bi-sectioning. At each step, we can check if the division of a community increases or decreases the modularity Q of the entire network. The division is performed whenever it increases Q , and we stop the algorithm when no further division is possible. The main drawback of this method is that sometimes the repeated bi-sectioning fails to find the best community division[10].

The second algorithm of modularity maximization we will describe is the Louvain algorithm[32]. The Louvain algorithm is a very fast modularity maximization algorithm, with a linear time complexity in sparse data. This allows the algorithm to be used to analyze very large networks, such as a Twitter network with 2.4M nodes[33]. Despite being fast, the algorithm is reliable[34], making it one of the most widespread algorithms to detect communities on a network. The algorithm is divided in two phases, iterated repeatedly. First, we start with every node a member of its own community. Then, for a chosen node i , we calculate the modularity gain of moving i from its community to the community of every neighbor of i . The node i is then moved to the community that maximizes the modularity gain. If the gain is negative, i stays at its original community. This is repeated for all nodes and several times, until no modularity gain can be found and the modularity reaches a local maxima. The second phase consists in changing the scale by generating a new network, where the nodes are the communities found in the previous step. This new network will be weighted and possess self-links ($a_{ii} \neq 0$), even if the original network does not. This two-step process constitutes a “pass”, and the algorithm is run a (generally small) number of passes until no more changes are possible and a maximum of modularity is obtained. In Fig. 2.4 we have a visual representation of the Louvain algorithm.

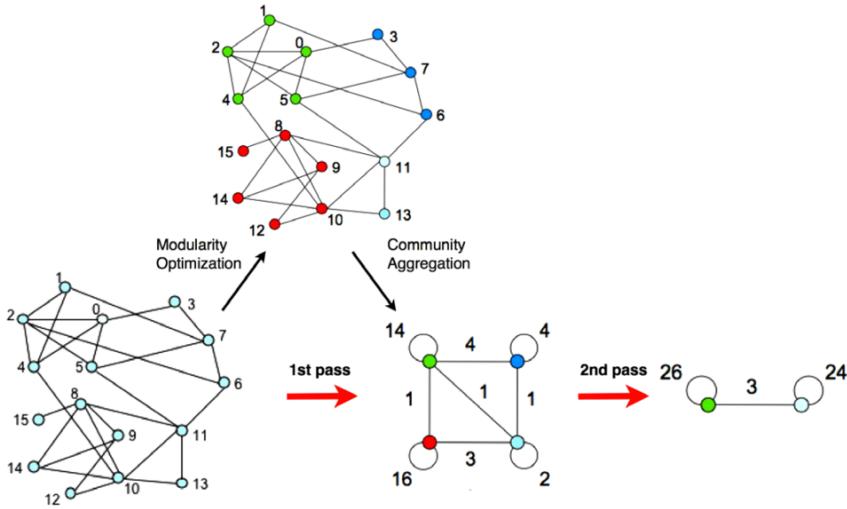


Figure 2.4: Visualization of the Louvain algorithm. At each pass the network is re-sized, with the new nodes being the communities of the previous network. Inside each pass the modularity is optimized. Reproduced from [32].

The efficiency of the algorithm is due primarily to two factors. First, after each pass the network is shrunk, cutting down the time to find the modularity maxima. More importantly, the modularity gain by changing the community of a link is a fast calculation. The gain ΔQ_{weight}^C by moving an isolated node i to a community C can be computed as

$$\Delta Q_{weight}^C = \left[\frac{2w_i^C + \Sigma^C}{2\omega} - \left(\frac{w_i + \Sigma^{tot}}{2\omega} \right)^2 \right] - \left[\frac{\Sigma^C}{2\omega} - \left(\frac{\Sigma^{tot}}{2\omega} \right)^2 - \left(\frac{w_i}{2\omega} \right)^2 \right] \quad (2.20)$$

where w_i^C is the strength of i pointing to nodes inside C , Σ^C is the total strength of links inside C , Σ^{tot} is the total strength of external links pointing to C , and w_i and ω were defined previously. A similar expression gives the change in modularity ΔQ_{weight}^- of removing i from its original community. Therefore one only needs to calculate $\Delta Q_{weight}^C - \Delta Q_{weight}^-$ for every community neighbor of i and chose the one that maximizes the gain. For the sake of comparison, in Table 2.1 we summarize the time complexity of the algorithms explained in this section, and of the other algorithms we mentioned.

Algorithm	Time complexity
Betweenness centrality	$O(mn(m+n))$
Hierarchical clustering (embedded)	$O(n^2) \sim O(n^2\log n)$
Simulated annealing	$O(n^3)$
Extremal optimization	$O(n^2\log n)$
Genetic algorithm	$O(mn^2)$
Greedy algorithm	$O(n^2) \sim O(n^2\log n)$
Kernighan-Lin	$O(mn)$ by step
Spectral modularity	$O(n^2) \sim O(n^3)$
Louvain algorithm (sparse)	$O(n)$

Table 2.1: Time complexity of various community detection algorithms.

2.4 Network models

Models of network connectivity are useful in studying the properties of real-world networks. For instance, the Barabási-Albert network proposes a “rich gets richer” algorithm of link generation whose degree distribution is seen in many real-world networks[35]. They are also useful in generating underlying network topologies to study dynamical models. Different topologies impact dynamics in different ways. For example, scale-free networks lack an epidemic threshold in epidemic spreading models, meaning they are very susceptible to disease spreading[36].

The first model we will explore is the Erdos-Rényi (ER) network. One of the most ubiquitous network models, it is usually the model referred to when the term “random network” is used. It is named after mathematicians Paul Erdos and Alfréd Rényi, who first studied its properties in 1959[37]. It is made of two closely variants: the $G(n, p)$ graph and the $G(n, m)$ graph. In the $G(n, p)$ graph an unweighted, undirected network with n nodes is constructed. Each of the possible $\binom{n}{2}$ links exist with probability p and independently of each other. In the $G(n, m)$ variant, a graph is chosen at random from the set of all graphs with n nodes and m links. As $n \rightarrow \infty$, the fluctuation in the number of links in the $G(n, p)$ graph decreases and both variants are equivalent for $m = \binom{n}{2}p$. Here we will study the $G(n, p)$ variant (and call it the ER network from now on), as it is both easier to work with analytically and to generate on a computer. The link generation in the ER network is a binomial process with probability p , and therefore the degree distribution is given by

$$p_k^{ER} = \binom{n-1}{k} p^k (1-p)^{n-1-k} \quad (2.21)$$

In the limit of large n we can take the Poisson approximation to the binomial

distribution. In this limit, we have $(1 - p)^{n-1-k} \simeq e^{-(n-1)p}$ and $\binom{n-1}{k} \simeq (n-1)^k/k!$, and with $K = (n-1)p$ we can write

$$p_k^{ER} = \frac{(n-1)^k}{k!} p^k e^{-K} = \frac{K^k}{k!} e^{-K} \quad (2.22)$$

which is a Poisson distribution . That is the reason the ER network is sometimes called the Poisson random graph.

Perhaps the most interesting property of the ER network is regarding its *giant component*. The largest component of a network is the biggest set of nodes in the network where a path exists between any two nodes. In other words, it is the biggest group of nodes where, starting from a random node, you can reach any other node in the group by jumping from one node to another. The size of the largest component is an important feature, as usually in applications it is crucial to have a network where a large portion of the nodes are reachable from one another. In the ER network, the size of the largest component is equal to 1 for $p = 0$ and n for $p = 1$. Therefore, in terms of the network size, the largest component is an intensive quantity for $p = 0$, but extensive for $p = 1$. A transition must occur from one regime to the other. When the largest component scaling is extensive we call it a giant component. For $n \rightarrow \infty$, the fraction S of nodes of the largest component in ER networks can be obtained[10] from

$$S = 1 - e^{-KS} \quad (2.23)$$

which can be solved graphically or numerically as a function of K . In Fig. 2.5 we have the result of the numerical solution of Eq. 2.23.

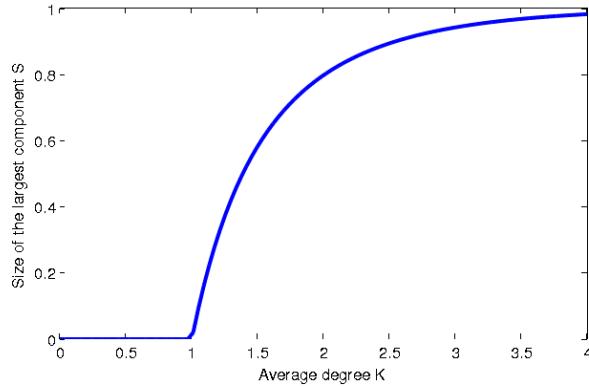


Figure 2.5: Size of the largest connected component in Erdos-Renyi networks.

The size of the largest component goes through a second-order phase transition, with a giant component appearing at $K = 1$ (or $p \simeq 1/n$). In other words, at

the thermodynamic limit the number of links needed for the entire network to be reachable is one⁴. This is very similar to the problem of percolation in infinite dimensions[4]. The percolation threshold in a Cayley tree is given by $p_c = 1/(z - 1)$, where z is the tree's coordination number. The coordination number in a random network of n nodes is given by $z = n - 1$. This results in $p_c \simeq 1/n$, which is the same critical probability where the giant component appears in ER networks.

The second model of network connectivity we will describe is the Barabási-Albert network[35]. The model proposes a simple mechanism of link generation capable of generating scale-free networks. The relevance of the BA model is that many real-world networks possess power-law degree distributions, like the ones in Fig. 2.6.

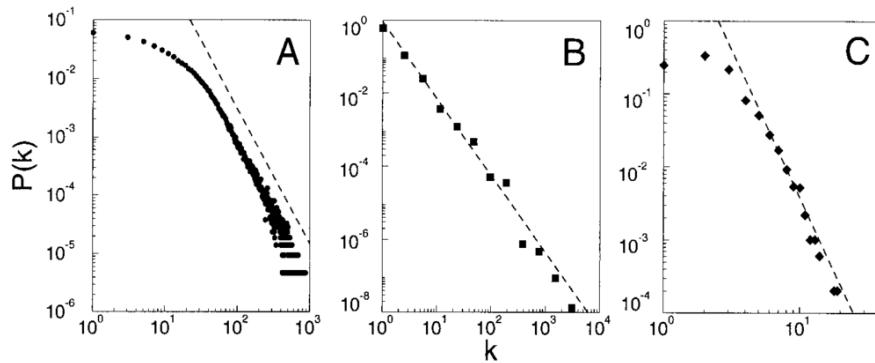


Figure 2.6: Examples of real-world networks with power-law degree distributions $p_k \sim k^{-\gamma}$. (A) Actor collaboration network, with $\gamma_{\text{actor}} = 2.3$. (B) World Wide Web, with $\gamma_{\text{WWW}} = 2.1$. (C) Power grid network, with $\gamma_{\text{power}} = 4$. Reproduced from [35].

The algorithm for generating a BA network is the following: starting with a completely connected network of m_0 nodes, at each time step a new node with m new links is added, with $m \leq m_0$. The probability p_i of a previously existent node i being chosen to connect with the new node is given by $p_i = k_i / \sum_j k_j$, for each new link. The process continues until a desired network size n is reached. In other words, the BA model uses a “rich gets richer” concept of network growth. This means that an already highly connected node is more likely to increase its degree than a node with fewer links. This idea, also known as *preferential attachment*, encounters parallels in many real-world networks with a power-law distribution. For instance in an actor collaboration network, such as in Fig. 2.6(A), it is most likely that a new actor will be cast in a supporting role with a more established actor.

⁴Strictly speaking, one would also need to prove that the solution $S > 0$ is the one chosen for $K > 1$, as opposed to $S = 0$ which is always a solution of Eq. 2.23. This can be done by considering how the periphery of a component grows with K [10].

The degree distribution of the BA network as described above can be calculated, in the $n \rightarrow \infty$ limit, from a continuum approach as

$$p_k^{BA} = \frac{2m^2}{k^3} \quad (2.24)$$

which gives⁵ a power-law with an exponent of $\gamma_{BA} = 3$. Therefore, preferential attachment can be suggested as an explanation of the scale-free behavior seen in some systems. It is important to note that both preferential attachment and network growth (addition of one node at a time) are essential for the appearance of power-law scaling. In a model variant without preferential attachment the degree distribution decays as $p_k \sim \exp(-k/m)$. In a model with all the nodes present at the beginning the degree distribution is not stationary, with the network eventually becoming fully connected.

The model studied by Barabási & Albert is, in fact, a special case of what is known as the Price model[39], after Derek Price. In this model, the probability of a node i receiving a link from a new node is given by $p_i \sim k_i^{in} + a$, where k_i^{in} is the node's in-degree and a is an attachment parameter. This models yields the degree distribution

$$p_k^{Price} \sim k^{-(2+a/K)} \quad (2.25)$$

which is a power-law distribution with exponent $\gamma_{Price} = 2+a/K$. The BA model can be obtained by setting $a = m$ and noting that $K \simeq m$ for BA networks. Therefore, while BA networks have a fixed power-law exponent $\gamma_{BA} = 3$, in the Price model the exponent can be controlled with a parameter. Many modifications to the BA model have been proposed to address shortcomings of the model regarding real-world networks. These include a node fitness function in addition to the degree[40] and a node aging process[41].

Scale-free networks, besides being common in nature, possess interesting properties. As already mentioned, scale-free networks are very susceptible to disease spreading[36]. They are also at the same time robust against random failure and weak against targeted attack[42]. In both processes, a fraction f of the nodes is removed and the topology of the resulting network is analyzed. In random failures the nodes selected for removal are chosen at random, whereas in a targeted attack the nodes with highest degree are removed. As can be seen in Fig. 2.7, ER networks are equally susceptible to random and targeted removal. The size S of the largest

⁵A master equation approach[38] yields $p_k = 2m(m+1)/[k(k+1)(k+2)]$, which is very similar to Eq. 2.24.

cluster goes through a second-order phase transition with the same critical fraction f_c for both cases. This is a result of the degree distribution of these networks. The standard deviation of the distribution of Eq. 2.22 is only \sqrt{K} , meaning there is not much difference between choosing the node with the highest degree and a random node. The situation is different for BA networks, however. For a random attack S decreases slowly, while for a targeted attack it has the same behavior of ER networks, albeit with a lower f_c . This means that BA networks are more robust than ER networks against a random attack, but more susceptible against a targeted attack. This can be explained by the fact that in BA networks most nodes have low degree, with a few nodes possessing very high degree. Therefore, a random failure is likely to remove a node with low degree, incapable of disrupting a large number of paths in the network. On the other hand, a targeted attack will remove a much larger number of links, larger than of a ER network of similar connectivity. Thus, a targeted attack will heavily disrupt the connectivity of a BA network.

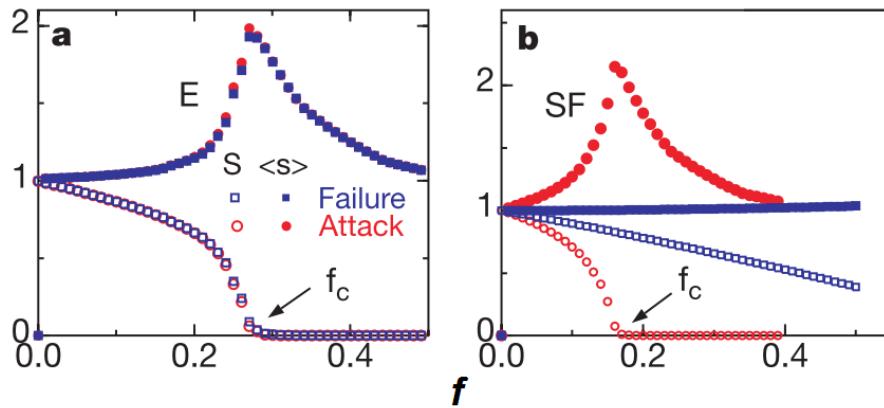


Figure 2.7: Network robustness against random failure (blue squares) and targeted attack (red circles), by removing a fraction f of the nodes. S (empty points) is the size of the largest cluster in the network, and $\langle s \rangle$ (filled points) is the average size of the other clusters. The metrics are calculated for (a) ER networks and (b) Barabási-Albert networks. Both networks have the same average degree. Adapted from [42].

The last model of connectivity we will address is the Watts-Strogatz (WS) network[14]. While ER networks use a simple linking rule, and BA networks explore preferential attachment, WS networks examine the small-world concept. Small-world networks are functionally integrated, as the average distance between nodes is small. On the other hand, highly clustered networks are functionally segregated, as nodes are divided in small groups. The WS model is capable of generating a small-world network that also maintains a high clustering coefficient C .

The WS network is generated in the following way: first we start with a ring network⁶ with n nodes and K neighbors per node. Each link is then rewired randomly⁷ with probability p . The average degree K is chosen so that $n \gg K \gg \ln(n)$, which guarantees a connected network ($S = 1$) in the $p \rightarrow 1$ limit, where we obtain an ER network. The clustering coefficient of the ring network ($p = 0$) can be calculated as $C(0) = (3K - 6)/(4K - 4)$, which converges to the high value of $3/4$ as K increases. However, its average path length $L(p)$ is $L(0) \simeq n/2K$, scaling linearly with network size. In the limit $p = 1$ we have an ER network, whose path length scales logarithmically as $L(1) \simeq \ln(n)/\ln(K)$ but with a low clustering $C(1) \simeq K/n$. The rewiring process is depicted in Fig. 2.8.

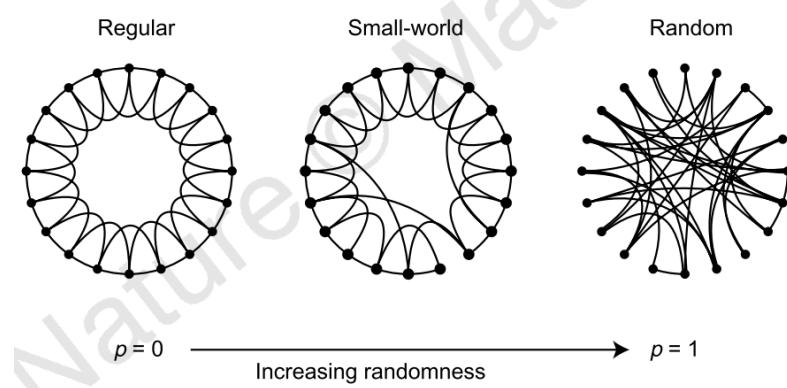


Figure 2.8: Graphical representation of the Watts-Strogatz model. By varying p the model is able to go from an ordered ring network ($p = 0$) to a random network ($p = 1$), with a small-world network for intermediate values. Reproduced from [14].

The idea behind the WS network is that, by shuffling a small number of links, these links can serve as shortcuts between distant regions. Therefore, while the clustering coefficient C would not be very affected, the average path length L would be reduced. In Fig. 2.9 we have the normalized measures $L(p)/L(0)$ and $C(p)/C(0)$ for varying p . Note that $L(p)$ decays faster than $C(p)$. The WS model was the first model to propose a mechanism for the generation of small-world networks with high clustering, and sparked much interest in the properties of such networks[43, 44].

⁶A ring network is a type of network topology where nodes are placed in a circular configuration and linked to their nearest neighbors in the circle.

⁷More precisely, both edges of the link are considered separately.

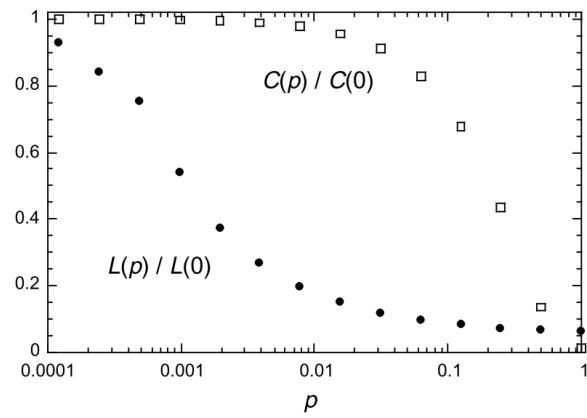


Figure 2.9: Normalized average path length $L(p)/L(0)$ and clustering coefficient $C(p)/C(0)$ for varying shuffling probability p . Reproduced from [14].

Theory of Random Threshold Networks³

3.1 Definition

Random threshold networks (RTN) have been used to model a vast array of phenomena, from neural networks[45] to genetic regulatory networks[46, 47]. A RTN is a dynamical system where each element is associated with a Boolean variable (“on” or “off”) and the activation of an element depends on a threshold. Let us consider a network with n randomly connected nodes and average degree K . Each existing link from node j to node i is assigned a weight $w_{ij} \in \mathbb{R}$, and the dynamical state of each node is given by a Boolean variable. The state $\sigma_i(t+1)$ of node i at time $t+1$ is then given by

$$\sigma_i(t+1) = \begin{cases} 1 & , \text{ if } \sum_{j=1}^n w_{ij}\sigma_j(t) > h \\ 0 & , \text{ if } \sum_{j=1}^n w_{ij}\sigma_j(t) \leq h \end{cases} \quad (3.1)$$

where $h \geq 0$ is a threshold parameter. The matrix W , whose entries are the links w_{ij} , is not necessarily symmetric, and the link weights are not necessarily unitary. In other words, a RTN can be a directed and/or weighted network. A network with the dynamics defined in Eq. 3.1 constitutes a Random Threshold Network.

Large-scale networks with Boolean variables were first introduced by Kauffman[48] in the context of gene regulation networks, and have been extensively studied since[49]. In this model, called a Random Boolean Network (RBN), the dynamical state $\sigma'_i(t+1) \in \{0, 1\}$ of a node i is given by

$$\sigma'_i(t+1) = F_i(\sigma'_{i,1}(t), \sigma'_{i,2}(t), \dots, \sigma'_{i,k}(t)) \quad (3.2)$$

where F_i is a Boolean function and $\sigma'_{i,j}$ denote the neighbors of i . Since the threshold function is a Boolean function, threshold networks are a subset of the ensemble of all

possible RBNs. The behavior of RTNs differ considerably from their general RBN counterparts, however[50].

From Eq. 3.1 we can see that RTNs are memoryless (or Markovian) deterministic systems with discrete time and updated in parallel. Information is only propagated by active nodes ($\sigma_i = 1$). This contrasts with the behavior of spin models such as the Ising model, where both ± 1 states propagate information. Since both Boolean and spin are two-state models, the dynamics of Eq. 3.1 can be mapped to spin values with

$$\sigma_i(t+1) = \begin{cases} 1 & , \text{ if } \sum_{j=1}^n w_{ij}\sigma_j(t) > h - \sum_{j=1}^n w_{ij} \\ -1 & , \text{ if } \sum_{j=1}^n w_{ij}\sigma_j(t) \leq h - \sum_{j=1}^n w_{ij} \end{cases} \quad (3.3)$$

which equates to every node possessing its own individual threshold $h_i = h - \sum_j w_{ij}$. Therefore, the dynamics of Boolean threshold models can be very different from the dynamics of spin threshold models.

Alternative variations of random threshold networks exist. Some authors add a third, self-regulating possibility to Eq. 3.1, given by $\sigma_i(t+1) = \sigma_i(t)$ if $\sum_{j=1}^n w_{ij}\sigma_j(t) = h$ [51, 52]. This can break the ergodicity of the dynamics by introducing long-term temporal correlations between the dynamical states[50]. While this change adds new rich dynamics, it is more complicated and may not be important to some applications. For instance, in biological networks it may not be reasonable for the input sum of a node to be exactly equal to the threshold. Another possible variation is to consider sequential node updates instead of parallel updates. In this case, only one node is updated at each time step according to some sequence. This type of sequential update can impact the dynamics, for instance changing the number of attractors and attractor lengths in the network[53]. However it also makes for a more complicated model, as it becomes necessary to analyze the impact of different update sequences¹.

In this dissertation we will study and use the simpler RTN model, as defined in Eq. 3.1 and with a parallel update rule. The link weights are taken as $w_{ij} = \pm 1$, meaning a node can either activate or inhibit its neighbors. The control parameter of the topology is the fraction of positive links F_+ , defined as

$$F_+ = \sum_{i,j} \operatorname{sgn}(w_{ij} + 1) / \sum_{i,j} |w_{ij}| \quad (3.4)$$

¹One can also make the update sequence random. However, this changes the dynamics from deterministic to stochastic, which presents its own set of problems.

The dynamics starts with an initial state $\vec{\sigma}(0) = \{\sigma_1(0), \sigma_2(0), \dots, \sigma_n(0)\}$ and is updated for each node in parallel afterwards. We are interested mainly in the network collective state, measured by the fraction of active nodes $A(t)$ given by

$$A(t) = \frac{1}{n} \sum_i \sigma_i(t) \quad (3.5)$$

We also focus on densely connected networks ($K > 10$), since many real-world networks are highly connected. For instance, neurons in the human cortex possess around ~ 60000 synapses each[54]. This high number of connections is thought to increase the information processing capacity in the brain[55].

3.2 Dynamics of Random Threshold Networks

In this section we will explore the dynamics of Random Threshold Networks with variable positive link fraction F_+ . While some dynamical properties of RTNs, such as sensitivity[50] and number of attractors[53] have been extensively studied, we are unaware of a study on the effect of the variable F_+ on the fraction of active nodes $A(t)$. In most RTN and spin models, links are either always excitatory ($F_+ = 1$) or excitatory and inhibitory with equal probability ($F_+ = 0.5$). However, in many applications the balance of excitation and inhibition is known to be different from these cases. In particular, the fraction of excitatory neurons in the mammalian cortex is around 70-80%, with inhibitory neurons thought to be essential in regulating brain activity[56].

A typical simulation result of the active fraction $A(t)$ is shown in Fig. 3.1. for Barabási-Albert[35] (left) and Erdos-Rényi[10] (right) network topologies, with $n = 10^4$ nodes. After an activation of a fraction $A(0)$ of its nodes, chosen randomly, the network quickly reaches a stable activity $A_\infty = A(t \rightarrow \infty)$. The resulting dynamics is largely independent of initial conditions, provided $A(0)$ is larger than a certain value A_0^C necessary for $A_\infty > 0$. Since the degree distribution for an Erdos-Rényi network is a Poisson distribution and the Barabási-Albert one is a power law, the active fraction is also largely independent from the degree distribution. Therefore, for the rest of this section we will assume the topology to be of an Erdos-Rényi network².

²There are some slight changes in the parameter space color map for a Barabási-Albert network for low degree. However the color map's general behavior is the same.

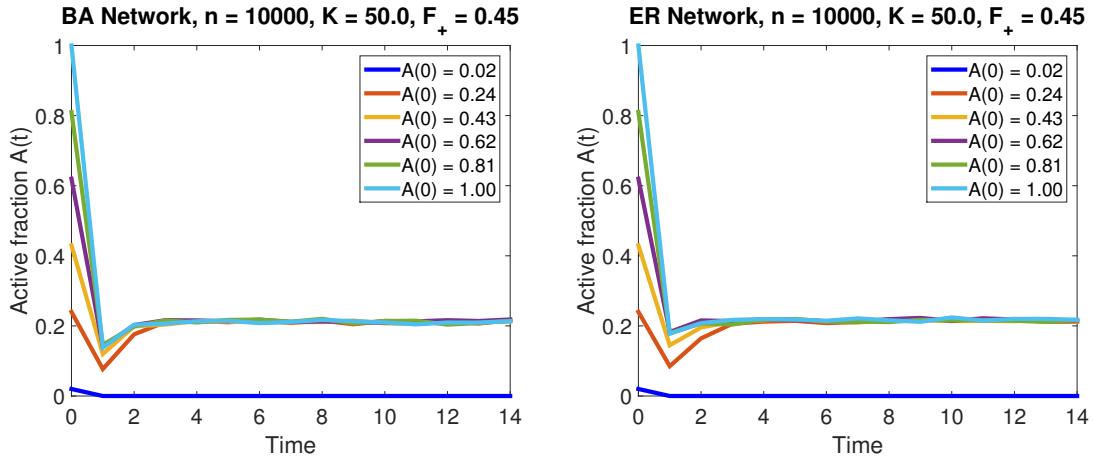


Figure 3.1: Network active fraction for Barabási-Albert (left) and Erdos-Rényi (right) topologies, with many initial conditions $A(0)$. The parameters are $n = 10^4$, $K = 50$, $h = 1$ and $F_+ = 0.45$ for both cases.

The dynamics is very sensitive to the value of F_+ , however. In Fig. 3.2 we have the value of A_∞ for the (F_+, K) parameter space with $h = 0$ and 1. For small F_+ (mostly inhibitory links) the activity dies out, and for high F_+ it takes over the network. There is a clear transition in A_∞ around $F_+ = 0.5$, and the transition range widens with smaller average degree K . It is important to note that the value of A_∞ does not depend on network size.

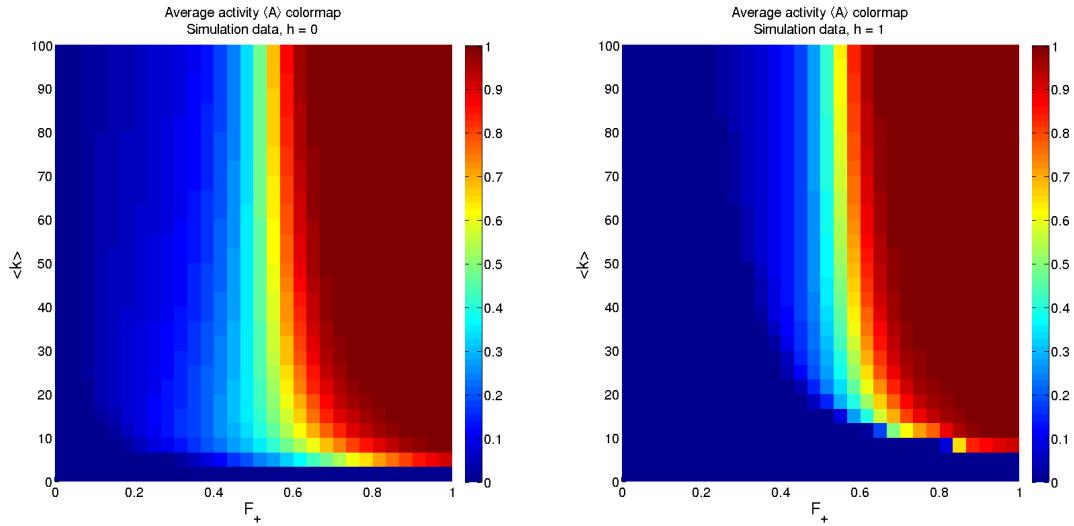


Figure 3.2: Stable activity A_∞ color map for simulation data with $h = 0$ (left) and $h = 1$ (right). The other parameters are $n = 10^4$ and $A(0) = 0.1$ for both cases. Each of the 900 data points is an average of 10 networks.

In Fig. 3.3 we have a color map for the average³ minimum activation A_0^C with $A_\infty > 0$ and the corresponding A_∞ . The value of A_0^C depends mainly on the average degree K , and can be very low for higher K . There is also no dependence on A_0^C for A_∞ . This comes from the overcritical nature of high-degree RTNs with low h . If we view the initial activation as a perturbation, in these networks a perturbation tends to quickly take over the entire network. Therefore, the value of an $A(0)$ larger than A_0^C basically dictates the spreading speed. The only exception is the extreme case of $A(0) \approx 1$ and low F_+ . In this case sometimes we get $A(1) = 0$ even if $A_\infty \neq 0$ for a lower $A(0)$.

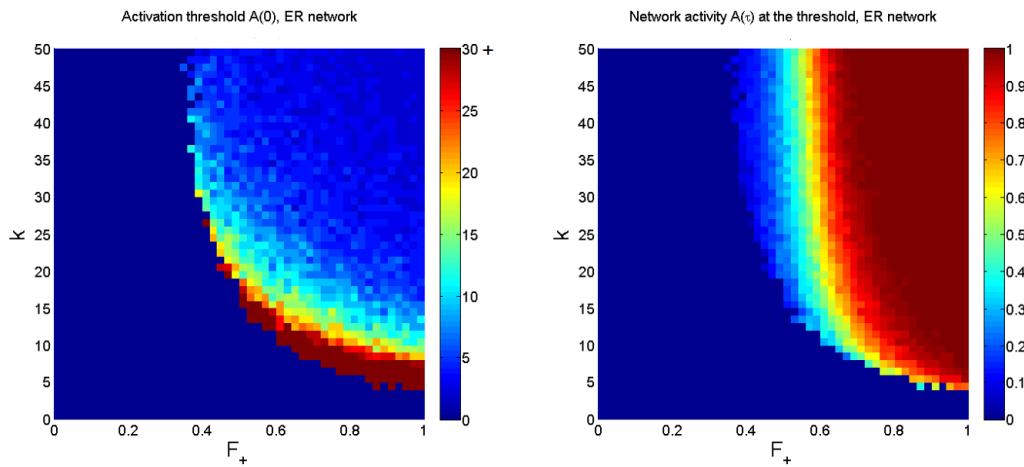


Figure 3.3: Minimum number of active nodes nA_0^C (left) so that $A_\infty > 0$ and the corresponding A_∞ (right). If the network could not be activated with any $A(0)$ we set $A_\infty = 0$. Parameters are $n = 500$ and $h = 1$.

3.3 Annealed approximation

To obtain a better insight on the dynamics of RTN we develop an expression for the stable network activity $A_\infty = A(t \rightarrow \infty)$. We start with the annealed approximation, introduced by Derrida & Pomeau[57]. The idea of this approximation is to ignore temporal correlations between nodes, effectively making each node independent. The activation probability of a given node with $m > h$ active in-degree

³The probability of survival of the dynamics after the first few timesteps is a sigmoid function of $A(0)$. As $n \rightarrow \infty$, it becomes a step function, and we can identify the activation threshold A_0^C . By taking the average threshold of a large number of simulations we are able to obtain A_0^C numerically.

neighbors, $P_{active}(m)$, is given by the binomial cumulative distribution

$$P_{active}(m) = \sum_{l=\lfloor \frac{m+h}{2} \rfloor + 1}^m \binom{m}{l} F_+^l (1 - F_+)^{m-l} \quad (3.6)$$

where F_+ is the network's fraction of positive links and h the activation threshold, with $\lfloor x \rfloor$ denoting the floor function⁴ of x . The probability of a node i with in-degree k_i being active at time $t + 1$ is given by

$$P[\sigma_i(t+1) = 1, k_i] = \sum_{m=h+1}^{k_i} \left\{ \binom{k_i}{m} A(t)^m (1 - A(t))^{k_i-m} P_{active}(m) \right\} \quad (3.7)$$

Summing over k_i , we have

$$P[\sigma_i(t+1) = 1] = \sum_{k=1}^{n-1} p_k \sum_{m=h+1}^k \left\{ \binom{k}{m} A(t)^m (1 - A(t))^{k-m} P_{active}(m) \right\} \quad (3.8)$$

where p_k is the degree distribution of the network. Since we assume the nodes to be independent, the number of active nodes at time $t + 1$, $nA(t + 1)$, will be given by the expected value of a binomial distribution with probability $P[\sigma_i(t + 1) = 1]$ and n trials:

$$nA(t + 1) = nP[\sigma_i(t + 1) = 1] \quad (3.9)$$

$$A(t + 1) = \sum_{k=1}^{n-1} p_k \sum_{m=h+1}^k \left\{ \binom{k}{m} A(t)^m (1 - A(t))^{k-m} P_{active}(m) \right\} \quad (3.10)$$

The expression in Eq. 3.10 is the annealed approximation, and can be found in [50]. It can be interpreted as a map from $A(t)$ to $A(t + 1)$, meaning that from an initial condition $A(0)$ we can calculate $A(t)$ for any t . If we consider the limit $t \rightarrow \infty$, we have the self-consistent equation

$$A_\infty = \sum_{k=1}^{n-1} p_k \sum_{m=h+1}^k \left\{ \binom{k}{m} A_\infty^m (1 - A_\infty)^{k-m} P_{active}(m) \right\} \quad (3.11)$$

⁴The floor function of x is defined as the integer part of x .

We now make further approximations to Eq. 3.11 in order to get a more manageable expression. The regularized incomplete beta function is defined as

$$I_z(a, b) = \frac{\int_0^z t^{a-1}(1-t)^{b-1} dt}{\int_0^1 t^{a-1}(1-t)^{b-1} dt}, \quad (3.12)$$

for $a > 0$, $b > 0$ and $z \in [0, 1]$. Integrating by parts, one can show that

$$I_p(n+1, N-n) = \sum_{i=n+1}^N \binom{N}{n} p^i (1-p)^{N-i} \quad (3.13)$$

In terms of the regularized incomplete beta function, Eq. 3.6 can be written as

$$P_{active}(m) = I_{F_+} \left(\lfloor \frac{m+h}{2} \rfloor + 1, m - \lfloor \frac{m+h}{2} \rfloor \right) \quad (3.14)$$

Approximating $\lfloor x/2 \rfloor \approx x/2 - 1/4$, we have

$$P_{active}(m) = I_{F_+} \left(\frac{m+h+3/2}{2}, \frac{m-h+1/2}{2} \right) \quad (3.15)$$

Substituting Eq. 3.15 into Eq. 3.11, we have

$$A_\infty = \sum_{k=1}^{n-1} p_k \sum_{m=h+1}^k \left\{ \binom{k}{m} A_\infty^m (1-A_\infty)^{k-m} \times I_{F_+} \left(\frac{m+h+3/2}{2}, \frac{m-h+1/2}{2} \right) \right\} \quad (3.16)$$

Simulations indicate that the behavior of A_∞ depends little on the degree distribution. Therefore we can replace the sum over p_K for the average degree K

$$A_\infty = \sum_{m=h+1}^K \binom{K}{m} A_\infty^m (1-A_\infty)^{K-m} \times I_{F_+} \left(\frac{m+h+3/2}{2}, \frac{m-h+1/2}{2} \right) \quad (3.17)$$

For $m \gg 0$, $I_{F_+} \left(\frac{m+h+3/2}{2}, \frac{m-h+1/2}{2} \right)$ is a slow-varying function, so we make another approximation by removing it from the innermost sum. This implies a distance from the transition between activity and no activity. Substituting m for its average value $\bar{m} = KA_\infty$, we have

$$A_\infty = \left[1 - \sum_{m'=0}^h \binom{K}{m'} A_\infty^{m'} (1-A_\infty)^{K-m'} \right] \times I_{F_+} \left(\frac{KA_\infty + h + 3/2}{2}, \frac{KA_\infty - h + 1/2}{2} \right) \quad (3.18)$$

where we used the fact that $\sum_{m=0}^K \binom{K}{m} A_\infty^m (1 - A_\infty)^{K-m} = 1$. If we assume $K \gg h$, which is true especially for $h > 0$, we can approximate $\sum_{m'=0}^h \binom{K}{m'} A_\infty^{m'} (1 - A_\infty)^{K-m'} \approx 0$, resulting in

$$A_\infty = I_{F_+} \left(\frac{KA_\infty + h + 3/2}{2}, \frac{KA_\infty - h + 1/2}{2} \right) \quad (3.19)$$

which can be written as

$$A_\infty = \frac{\int_0^{F_+} t^{(KA_\infty+h+1)/2} (1-t)^{(KA_\infty-h-1)/2} dt}{\int_0^1 t^{(KA_\infty+h+1)/2} (1-t')^{(KA_\infty-h-1)/2} dt}, \quad (3.20)$$

Equation 3.19 can be solved numerically in order to yield A_∞ . In particular, we can use the inverse incomplete beta function⁵ to obtain

$$F_+ = I_{A_\infty}^{-1} \left(\frac{KA_\infty + h + 3/2}{2}, \frac{KA_\infty - h + 1/2}{2} \right) \quad (3.21)$$

which gives the F_+ necessary for the generation of a threshold network with specified A_∞ and K . Most popular algebra packages, for instance MATLAB(R) and Mathematica(R), have built-in functions for the incomplete beta function and its inverse. This makes Eq. 3.19 and Eq. 3.21 easier to use than the previous result of Eq. 3.11.

This annealed approximation explains certain properties of RTNs. First off, the result is independent of network size n . This is seen in simulations, where the network size only dictates the size of the fluctuations around A_∞ . The approximation discards these fluctuations, eliminating the dependence on n . As K increases the non-trivial ($A_\infty \neq 0$ and 1) dynamic states become more centered around $F_+ = 0.5$, which is the behavior of the regularized beta function $I_X(a, a)$ when a is high[58]. For low connectivity there is an asymmetry towards higher F_+ . In terms of the approximation, this has its origin in the asymmetry of $h + 1/2$ in the arguments of I_{F_+} that vanishes as KA_∞ increases. In Fig. 3.4 we plot the phase space of Eq. 3.19 for $h = 0$ and $h = 1$.

⁵If $Y = I_X(a, b)$, then $X = I_Y^{-1}(a, b)$.

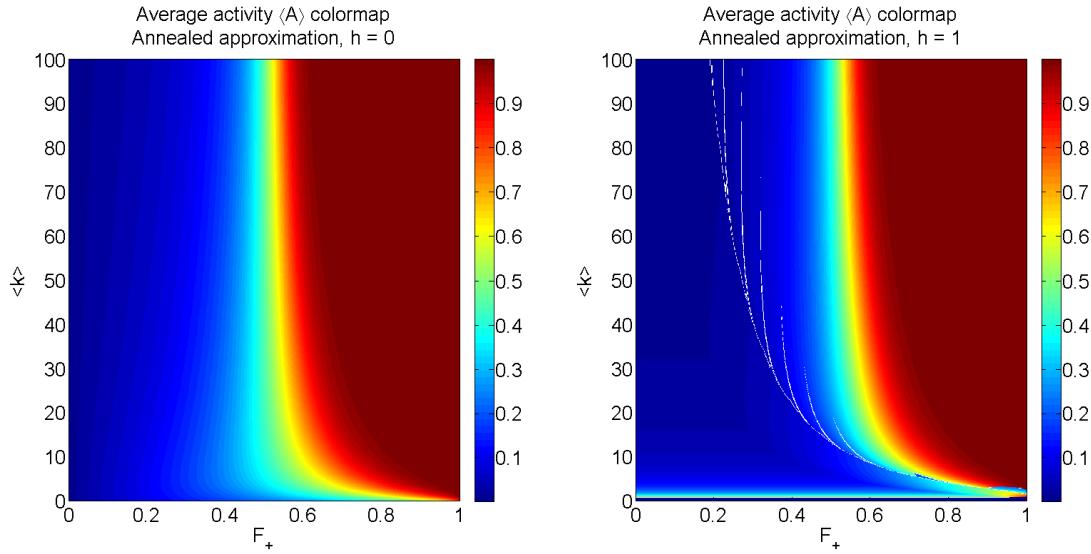


Figure 3.4: Stable activity $A_\infty(K, F_+)$ color map of the mean-field approximation for $h = 0$ (left) and $h = 1$ (right).

The mean-field results qualitatively match the simulation results, with some divergence for low K . This can be seen in Eq. 3.19, where $h = 0$ and $K = 0$ results in a non-zero $A_\infty = I_{F_+}(3/2, 1/2)$. For a more quantitative comparison, in Fig. 3.5 we compare the analytic and simulation results for a fixed degree K . It is important to note that in the approximation we assumed a high degree K . Therefore, the lowest degree we will use when comparing the approximation to simulation data will be $K = 10$.

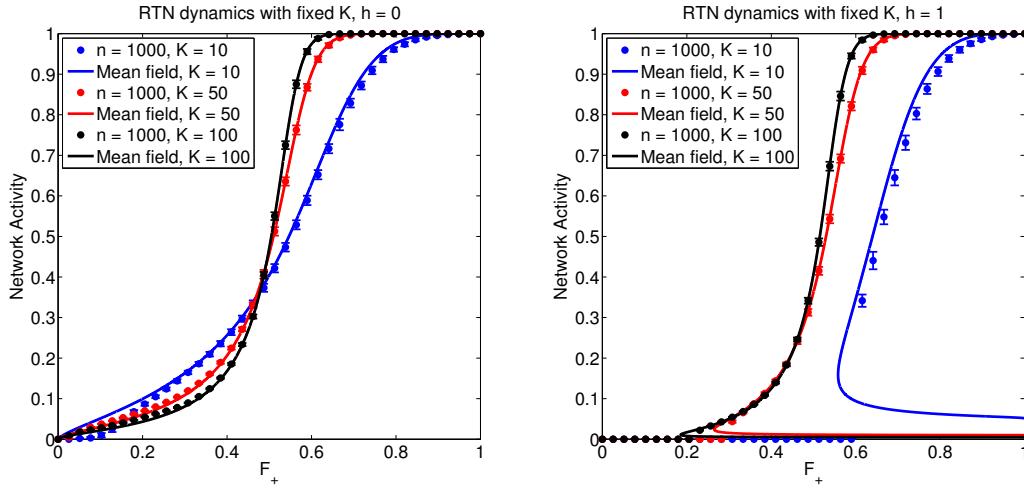


Figure 3.5: Comparison between the theoretical prediction and simulation results with fixed degree K for $h = 0$ (left) and $h = 1$ (right).

For $h = 0$ there is an excellent match between the approximation and the simulation. The result is less clear for $h = 1$. As we approach the transition between activity and no activity a low A_∞ solution for Eq. 3.19 appears. Before addressing why this solution shows up, let us analyze the results for a fixed F_+ in Fig. 3.6.

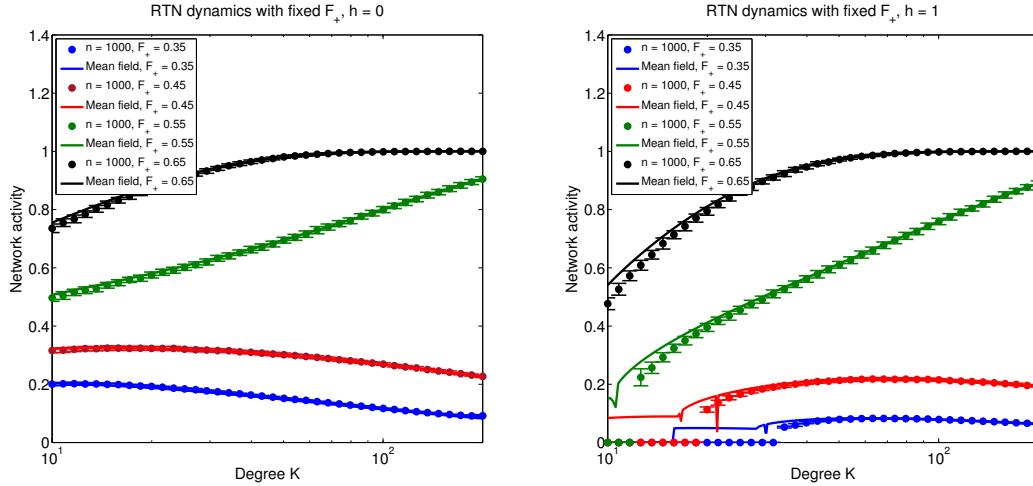


Figure 3.6: Comparison between the theoretical prediction and simulation results ($n = 1000$) with a fixed positive fraction F_+ for $h = 0$ (left) and $h = 1$ (right).

In this case the approximation also matches well the simulations for $h = 0$, with some divergence near the activity transition for $h = 1$. Since Eq. 3.19 is a transcendental equation, we used an optimization method in order to solve it numerically. This led to some numerical errors near and before (lower K) the transition between $A_\infty = 0$ and $A_\infty > 0$ in Fig. 3.6. After this transition, however, the approximation fits well the simulation results.

The discrepancy between the simulation and mean-field results for $h = 1$ and low A_∞ is expected, as our assumption that $P_{\text{active}}(m)$ (Eq. 3.15) is slow-varying breaks down at $m \approx 0$. The low A_∞ solution in Fig. 3.5 (right) is therefore not an unstable fixed point, and is not present in Eq. 3.17. Since Eq. 3.21 is defined for every choice of (A_∞, K) with $KA_\infty - h + 1/2 > 0$, it will provide an F_+ even when such a choice is not possible. This happens when A_∞ goes through a first-order phase transition, which is more visible for $h = 1$ and low K , as seen in Fig. 3.5 (right) for $K = 10$. In other words, the approximation fails at the transition between $A_\infty = 0$ and $A_\infty > 0$ because it is a first-order phase transition. However, this failure is only of one type: predicting a non-zero A_∞ for a network with $A_\infty = 0$. When the network has a non-zero A_∞ , our approximation is accurate.

3.4 Phase transition of A_∞

In order for our approximation to be practical, we must find its domain of validity. This will give a range of values where it can be applied. Therefore, we must find the lowest activity $A_\infty^{min} > 0$, for any F_+ , that the network can have for a fixed K . With this in hand, we can safely use Eq. 3.21 to predict the F_+ a network must possess to have an activity $A_\infty \geq A_\infty^{min}$ with a certain degree K . In other words, we must study the phase transition between activity and no activity in the network. First, we must show that A_∞ indeed goes through a first-order phase transition. To do so, we plot the results of a large number of simulations with random F_+ and $h > 0$. In Fig. 3.7 we show the results for $h = 1$ and $h = 2$.

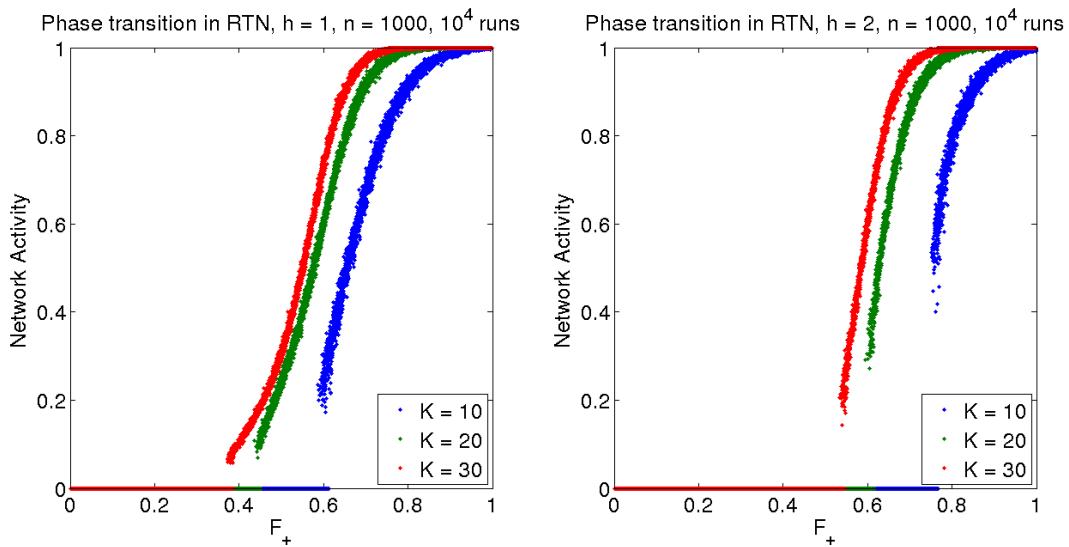


Figure 3.7: First-order phase transition of A_∞ for $h = 1$ (left) and $h = 2$ (right). The other parameters are $n = 1000$, $K \in \{10, 20, 30\}$ and $A(0) = 0.5$. Each curve is the result of 10^4 simulations.

Since A_∞^{min} depends only on K , one way to find it is to run simulations covering $F_+ \in [0, 1]$ and obtain $A_\infty^{min} = A_\infty^{min}(K)$ for any desired h . The behavior of the network appears to differ substantially between $h = 0$ and $h > 0$, so we will analyze these cases separately. The simulations are done in the following way: for a certain size n and degree K , we generate networks with F_+ starting at zero and increasing in steps of 0.01. When a network with $A_\infty > 0$ is found, we stop the process and register the activity as A_∞^{min} . The reasoning is that if $A_\infty > 0$ for a certain F_+ , then $A'_\infty > 0$ for another $F'_+ > F_+$, and $A'_\infty \geq A_\infty$. This behavior is supported by simulations. For completeness, the initial condition for every simulation is $A(0) = 0.5$. In Fig. 3.8 we have the results of A_∞^{min} for $h = 0$, and $K \in [10, 200]$.

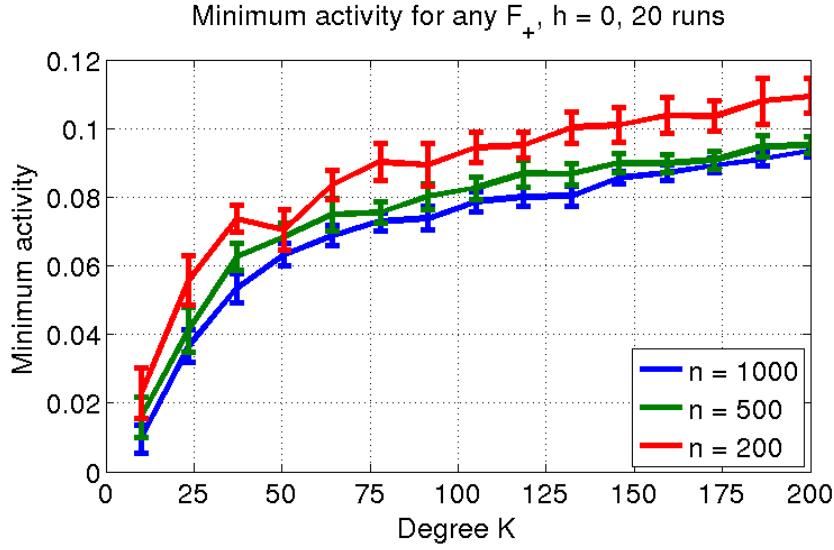


Figure 3.8: Minimum activity A_∞^{\min} for any F_+ . The parameters are $10 < K < 200$ and $n \in \{200, 500, 1000\}$. The threshold is $h = 0$ and every point is the average of 20 simulations.

The minimum activity A_∞^{\min} increases with larger K , but it is small for $K < 200$. For $n = 500$ and $K = 10$, for instance, the minimum number of active nodes is around 5. Therefore, for $h = 0$ and $A_\infty > 0.15$ one can safely use Eq. 3.21. There is also a small dependence on the network size, with A_∞^{\min} decreasing with larger n . In Fig. 3.9 we have the behavior for various $h > 0$ and $n = 200$. This size n was chosen because it establishes a higher bound in A_∞^{\min} for higher n - if a certain desired A_∞ is larger than A_∞^{\min} for $n = 200$, it will also be for $n > 200$.

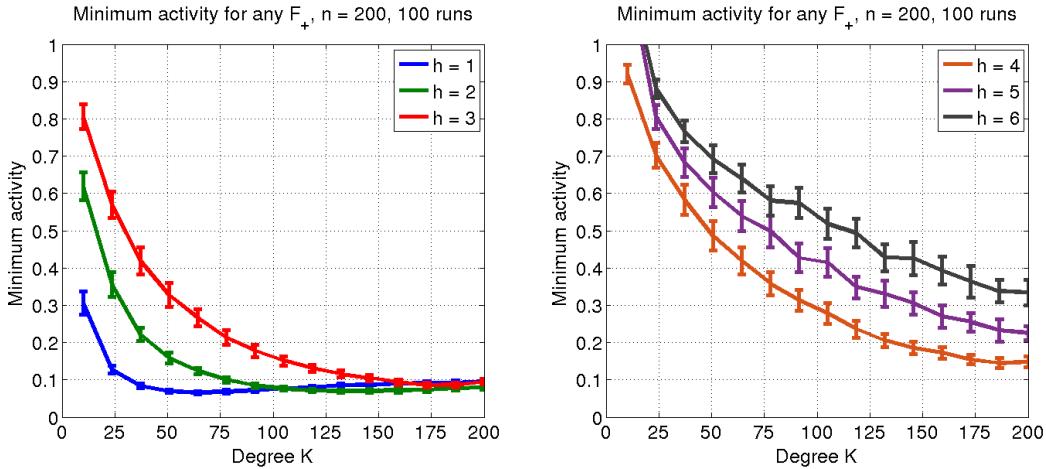


Figure 3.9: Minimum activity A_∞^{\min} for any F_+ . The parameters are $10 < K < 200$ and $n = 200$. The threshold is $h \in \{1, 2, 3\}$ (left) and $h \in \{4, 5, 6\}$ (right). Every point is the average of 100 simulations.

For $h > 0$, the dynamic exclusion region can be very large. Contrary to the case $h = 0$, for $h > 0$ the value of A_∞^{\min} decreases with larger K . This means that the dynamic range $\Delta_A = 1 - A_\infty^{\min}$ decreases with K for $h = 0$ but increases for $h > 0$. This difference in behavior can be seen, less clearly, in Fig. 3.2 for $h = 0$ and $h = 1$. Whereas for $h = 0$ we can usually assume $A_\infty^{\min} < 0.15$, for higher h it depends strongly on K . In particular, for $h = 5$ and $K \leq 10$ we have no activity whatsoever with $n = 200$, while for $K = 200$ every $A_\infty > 0.2$ is available.

The simulations in Figs. 3.8 and 3.9 establish the domain of validity for our mean-field approximation for a wide range of parameters. Therefore, in order to use the approximation one must examine Figs. 3.8 and 3.9 to find out if the desired A'_∞ is larger than the minimum activity. If that is the case, then Eq. 3.21 gives the F_+ required for the generation of a RTN with activity A'_∞ .

3.5 Measures of damage spreading

The dynamics of the RTN model is deterministic. Therefore, if the dynamical states of two identical networks are equal at time t , they will always be equal in the future. In real-world networks, however, the system may suffer a dynamical error or be affected by external influences. Thus, we are interested in studying the resilience of the system to external perturbations. These perturbations come in the form of externally changing the state of a node (or group of nodes) and examining how the change spreads with time.

Our main tool in analyzing the response to perturbations in RTNs is the Hamming distance . We define the Hamming distance between two configurations with node states $\sigma_i(t)$ and $\tilde{\sigma}_i(t)$ as

$$H(t) = \sum_i |\sigma_i(t) - \tilde{\sigma}_i(t)| \quad (3.22)$$

In other words, the Hamming distance measures the number of nodes with different states between two different configurations. If we consider the configuration with $\tilde{\sigma}_i$ as a perturbed version of the one with σ_i , then $H(t)$ measures how the perturbation spreads through the network. Let us define the asymptotic value of $H(t)$, averaged over many different perturbations of the same size⁶ as

$$H_\infty = \langle \lim_{t \rightarrow \infty} H(t) \rangle \quad (3.23)$$

⁶That is, different perturbations that change the value of the same number of node states.

If $H_\infty = 0$, then a perturbation will vanish from the dynamics, and we consider the network in the ordered regime. If $H_\infty > 0$, the perturbation stays in the network and the network is in the chaotic regime. If $H_\infty = 0$ only marginally, then perturbations take a long time to be removed from the network. We call this the critical regime. The value of H_∞ therefore acts as an order parameter of the dynamics[50], and can be computed directly from Eq. 3.23. Another way to compute H_∞ is using the Derrida plot (also called the Derrida Map)[59]. The Derrida Plot relates the size of a perturbation $H(0)$ with itself after one time step, $H(1) = M(H(0))$. In other words, it is the perturbation size one time step after it happens, as a function of the original perturbation size. Let us define the average network sensitivity S as

$$S = \frac{dM(H_0)}{dH_0} \Big|_{H_0=0} \quad (3.24)$$

which is the slope of the Derrida Plot as the size of the perturbation goes to zero. From mean field considerations[50], we see that if $S < 1$ then $H_\infty = 0$ and the network is in the ordered regime. If $S > 1$ we have $H_\infty > 0$ and the network is in the chaotic regime. The critical regime happens for $S = 1$. Therefore, one can only look how the network responds to small perturbations in order to ascertain its dynamical regime. If the network is critical, then the Derrida Plot shall follow the bisection $H(1) = H(0)$ for small perturbations.

Adaptive Networks

4.1 Definition

The study of networks can be divided in two main lines of research. The first is concerned with the characterization of network topology. This is done mainly with metrics, such as those described in Sections 2.2 and 2.3, and models of connectivity, like those in Section 2.4. The second line of research is concerned with dynamics on networks. Examples include models of oscillators like the Kuramoto model[60], spin models[61] and contact processes such as the SIS model[36]. Following this scheme, we study dynamical systems and how the underlying topology of these networks affects such systems.

Real-world networks in general evolve their topology in time. Moreover, in many networks there is an influence of the network's dynamical state on this evolution. A prime example is the brain, where the ability to learn can be thought of as the interplay between the dynamical state (neuronal firing patterns) and the topology (distribution of neurons and synapses)[62]. Other examples are power-grid networks, where new transmission lines are built to remove bottlenecks. Another example are epidemiological networks. The human behavior can change with weather, with people staying in closed environments, and in large epidemics with widespread panic. These changes can alter how the disease spreads.

An *adaptive network* is a network model where there is an interplay between dynamical and topological evolution[63]. As already mentioned, the topology of a network can deeply affect its dynamics. In an adaptive network, there is an effect of the dynamics on the topological evolution, according to some rule. In other words, the topology affects the dynamics, but the dynamics also influences the topology. This feedback mechanism is represented in Fig. 4.1.

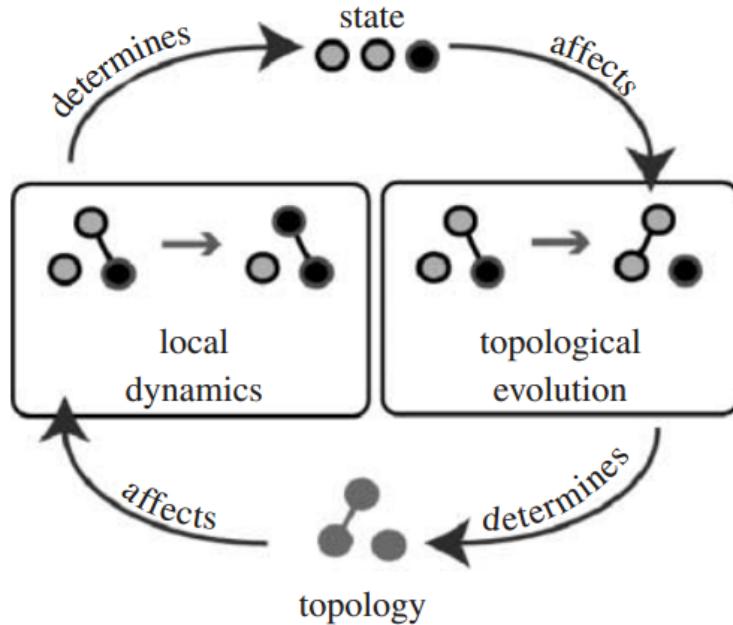


Figure 4.1: Evolutionary mechanism of an adaptive network, showing the interplay between topology and dynamics. Reproduced from [64].

Adaptive network models have been implicitly studied in many fields and under different names, such as network creation games in engineering[65]. The systematic study of adaptive networks and their properties is a recent effort, however. By comparing different models, it became clear that certain properties are common among adaptive networks[64]. These properties include the formation of complex topologies and more complex dynamics than their counterpart models on fixed networks. Another property, associated with the complex topology, is a spontaneous division of labor in the network. For instance, the network can divide itself into leaders, that can strongly affect the network dynamics, and followers, that do not. Lastly, another property seen in many adaptive models is Self-Organized Criticality (SOC). In SOC models, the co-evolution between topology and dynamics leads the network to a critical state. In the next sections we will briefly present models that illustrate these properties.

4.2 Complex dynamics and topology

Let us first consider a simple epidemiological model, the susceptible-infected-susceptible (SIS) model. In the SIS model, each node is an individual that can either be susceptible (S) or infected (I) by a disease. Susceptible nodes can become infected, and infected nodes eventually recover and become susceptible again. In our particular realization, each infected node can pass the disease to each of its neighbors at each

time step with fixed probability p . Each infected node can also recover to susceptible with probability r per time step. The average number of infections an infected node generates, given by R_0 , can be used to predict how the disease spreads. If $R_0 > 1$ then the disease will spread and take over the network, if $R_0 < 1$ then it will eventually die off. For a network with an average degree K we can calculate $R_0 = pK/r$. Setting $R_0 = 1$ we have the critical transmission rate $p^* = r/K$. Therefore, if our disease has a higher transmission rate than p^* it will eventually take over the entire network¹.

In 2006, Gross and collaborators[66] proposed the following modification to the SIS model: at each time step, a susceptible node can cut its link to an infected node with probability w . The cut link is then rewired to another susceptible node. This is an example of a simple adaptive network, and it emulates the behavior of humans regarding infectious diseases. The critical transmission rate for this model can be calculated as

$$p_{AN}^* = \frac{w}{K [1 - \exp(-w/r)]} \quad (4.1)$$

Note that $p_{AN}^* \approx r/K + w/2K + O(w^2)$ and therefore Eq. 4.1 recovers the previous result for $w = 0$. Moreover, we have $p_{AN}^* = w/K$ for $w \gg r$. Therefore, a high rewiring can increase the critical transmission rate, significantly changing the dynamics of the model. The rewiring also changes the topology. To better understand this change, we can compare the model to two limiting cases: a random rewiring independent of the node state (S or I), and a case without infection and recovery ($p = r = 0$). The differences can be seen in the degree distribution p_k , plotted in Fig. 4.2.

¹It is worth noting that this offers a possible explanation for the flu season. During the winter, people tend to stay in closed and crowded environments. This increases K of the social network, increasing R_0 and making the virus more likely to spread.

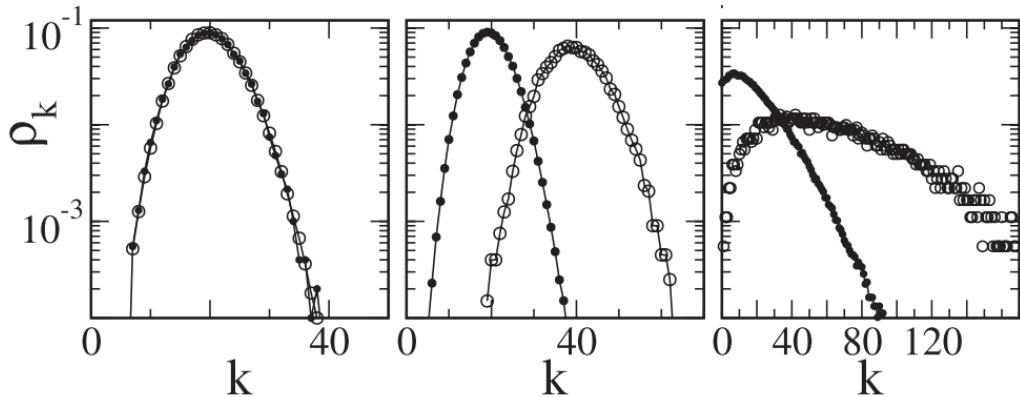


Figure 4.2: Degree distribution p_k of the SIS adaptive model. The black dots are for the infected, and the circles for the susceptible nodes. Model with random rewiring independent of state (left). Model without infection and recovery ($p = r = 0$) (center). Adaptive network model with $w = 0.3$, $r = 0.002$ and $p = 0.008$. Other parameters are the number of nodes $n = 10^5$ and average degree $K = 20$. Adapted from [66].

The topological evolution without the feedback from the local dynamics (Fig. 4.2, left) leads to a Poisson degree distribution. Without the dynamics itself (Fig. 4.2, center) the network breaks into two isolated components: one of susceptible nodes, one of infected nodes. They have different average values (depending on the initial number of infected), but they still follow a Poisson distribution. For the full adaptive model however (Fig. 4.2, right) we see a clear distinction between p_k for the infected and susceptible nodes. Both distributions are broader than the other cases. This is an example of a complex topology resulting from a simple adaptive algorithm. This evolving topology also produces complex dynamics. Infected nodes lose links, which in principle would prevent spreading of the disease. However, the rewired links create densely-connected groups of susceptible links. These groups are vulnerable, and when the disease invades the group by infecting one member, it can quickly infect the entire group[66].

Another model with an interesting co-evolution between topology and dynamics was devised by Aoki & Aoyagi[67]. In this model, each node possesses a quantity $x_i(t)$ of a resource that diffuses through the network in time. The network is undirected but weighted, with weights $w_{ij}(t)$ that change in time depending on the resource of nodes i and j . The underlying topology of the network is fixed, and given by an Erdos-Renyi network with n nodes and average degree K . Only pairs of linked nodes ($a_{ij} = 1$) can have $w_{ij}(t) \neq 0$. The change in resource of a node

$\Delta x_i(t) = x_i(t+1) - x_i(t)$ is given by

$$\Delta x_i(t) = -\kappa(x_i(t) - 1) + D \sum_j \left(\frac{w_{ij}(t)}{s_j(t)} x_j(t) - \frac{w_{ij}(t)}{s_i(t)} x_i(t) \right) \quad (4.2)$$

where $F(x) = -\kappa(x - 1)$ is a self-dissipation term with an equilibrium state $x = 1$, and κ and D are parameters. The strength of node i is $s_i(t) = \sum_j w_{ij}(t)$. The second term in Eq. 4.2 is a diffusive term. It can be obtained by noting that the resource sent by node i to node j is proportional to $w_{ij}x_i/s_i$. The evolution in the link weight $\Delta w_{ij}(t) = w_{ij}(t+1) - w_{ij}(t)$ is given by

$$\Delta w_{ij}(t) = \epsilon [x_i(t)^\alpha x_j(t)^\alpha - w_{ij}(t)] \quad (4.3)$$

where α and ϵ are parameters. The link weight change in Eq. 4.3 is akin to a chemical reaction between the resources x_i and x_j that strengthen the link w_{ij} . In terms of a chemical reaction, it is equivalent to

$$\alpha[x_i] + \alpha[x_j] \rightleftharpoons [w_{ij}] \quad (4.4)$$

where the brackets denote the quantity of a substance, and α the proportion of reagents. From the law of mass action, the forward rate of the reaction will be given by $f_r(t) = \epsilon x_i(t)^\alpha x_j(t)^\alpha$, and the backward rate by $f_b(t) = \epsilon w_{ij}(t)$. In this case, the parameter ϵ is identified with the affinity constant of the reaction. We see that $\Delta w_{ij}(t) = f_r(t) - f_b(t)$, which explains Eq. 4.3, and that at the equilibrium $w_{ij} \rightarrow x_i^\alpha x_j^\alpha$.

This model is inspired by two processes: a diffusion process for the nodes, and a chemical reaction process for the links. Both resource and link distribution at $t = 0$ are taken from a normal distribution with mean 1 and standard deviation 0.1. In Fig. 4.3(a) we have a representation of the adaptation algorithm. In Fig. 4.3(b)-(d) we have the complementary cumulative distribution function (complementary CDF)² for the resource x_i , link weight w_{ij} and node strength s_i for an evolved network.

²Also known as the tail distribution. If a discrete variable has a probability distribution $p(x)$, the complementary CDF is given by $\bar{F}(y) = 1 - \sum_{x < y} p(x)$.

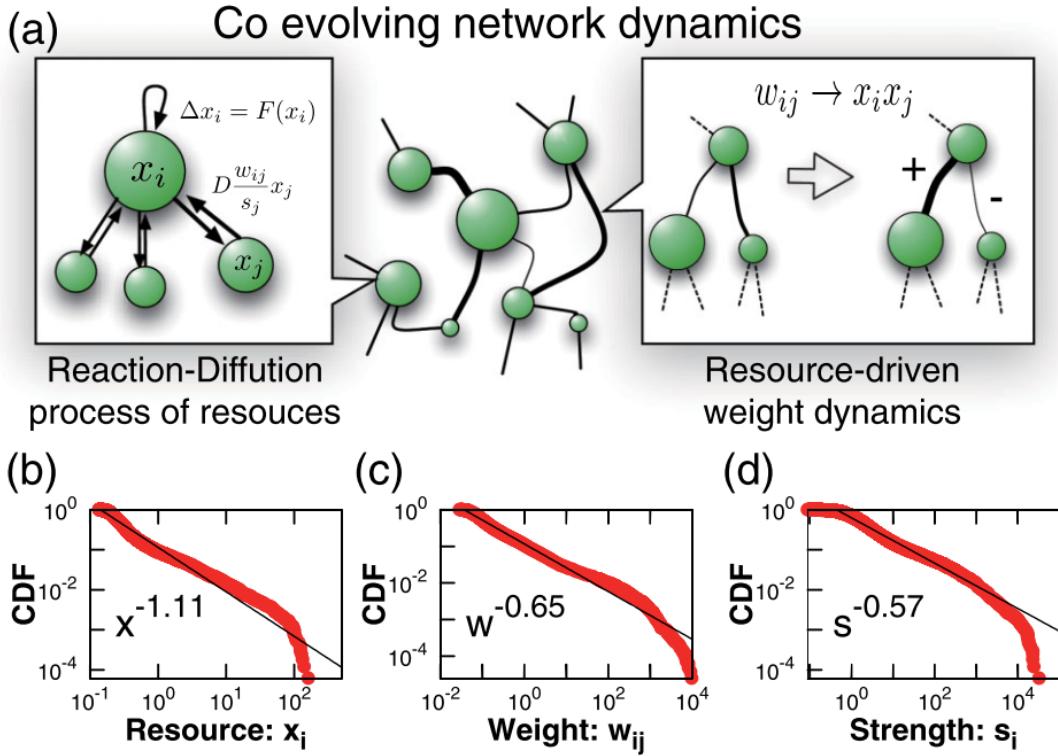


Figure 4.3: Illustration of the co-evolution of the model by Aoki *et al* (a). Complementary cumulative distribution of the resource x_i (b), link weights w_{ij} (c) and strength s_i (d) for the evolved network. The underlying network is an ER network with $n = 16384$ and $K = 10$. Other parameters are $\alpha = 1$, $\kappa = 0.05$, $D = 0.34$ and $\epsilon = 0.01$. Reproduced from [67] .

We see that all the distributions are power-laws with different exponents. Besides being power-laws, the final distributions are completely different from the initial distributions, and are not present without the co-evolution. This means they are generated by the co-evolution. The most important parameter of the model is α , and it regulates the influence of the resource on the link evolution. In Fig. 4.4 we have the results of the adaptive algorithm for varying α and $\kappa = 0$, meaning no self-dissipation or generation of resource.

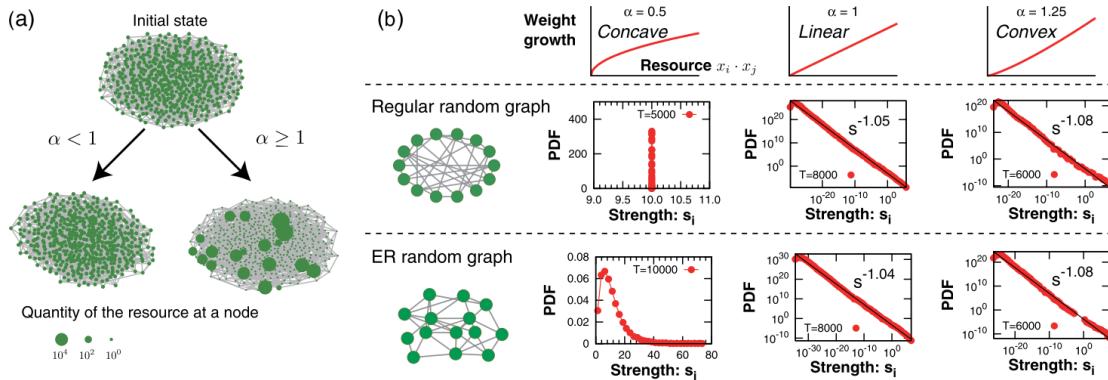


Figure 4.4: Network evolution for $\kappa = 0$ in the model by Aoki *et al.* Simulation of the resource distribution as a function of α , with $n = 512$, $K = 5$ and $\alpha \in \{0.5, 1.25\}$ (a). Consequences on the topology for $\alpha \in \{0.75, 1, 1.25\}$. The other parameters are $n = 16384$, $K = 10$, $D = 0.02$ and $\epsilon = 0.01$. Reproduced from [67].

A very different picture emerges for $\alpha < 1$, with the resource being distributed equally at the end of the evolution (Fig. 4.4a). The strength distribution also changes, and starts to depend on the underlying topology (Fig. 4.4b). This model is therefore an example of the co-evolution between topology and dynamics resulting in a complex topology. It also generates an interesting dynamics, where a diffusion process can have two very different outcomes depending on the influence of the topological evolution.

4.3 Self-Organized Criticality in Boolean models

Models based on Boolean dynamics, such as Random Boolean Networks and Random Threshold Networks (Chapter 3), are good candidates for adaptive models because of their general simplicity. In this section we will review some simple adaptive models based on Boolean states with Self-Organized Criticality (SOC). Introduced by Bak & Tang in 1987[68], a SOC system is a dynamical system which robustly poses itself at a critical state of a phase transition. More precisely, it refers to a class of dynamical systems where a critical point is a dynamical attractor[69]. In other words, it is a system that does not need fine tuning of parameters or initial conditions to be in a critical state. This led to the application of SOC models to explain a wide variety of phenomena, such as earthquakes[70], forest fires[71] and co-evolution of species[72].

The first SOC model we will review was proposed by Bornholdt & Röhl in 2000[73]. In this model, each of the N nodes have a spin value $\sigma_i(t) = \pm 1$, and

undirected links $w_{ij} = \pm 1$ between them. The dynamics of each node is given by

$$\sigma_i(t+1) = \begin{cases} +1 & , \text{ if } \sum_{j=1}^n w_{ij}\sigma_j(t) > 0 \\ -1 & , \text{ if } \sum_{j=1}^n w_{ij}\sigma_j(t) \leq 0 \end{cases} \quad (4.5)$$

where each nodes is updated in parallel at each time step. This is a threshold update rule similar to the one studied in Chapter 3, with a threshold of $h = 0$. However, as noted in Section 3.1, the change from binary (0 or 1) to spin values introduces important changes to the dynamics. The adaptive algorithm is as follows. Starting from a random initial condition, the dynamics runs according to Eq. 4.5 until it reaches an attractor. This can be a fixed point (every node is frozen) or a limit cycle. Then a random node i is chosen. If i remained frozen during the attractor, it receives a new random link $w_{ij} \in \{-1, 1\}$. If not, it loses a random pre-existing link. The evolution in the average degree³ K_{ev} of the network can be seen in Fig. 4.5.

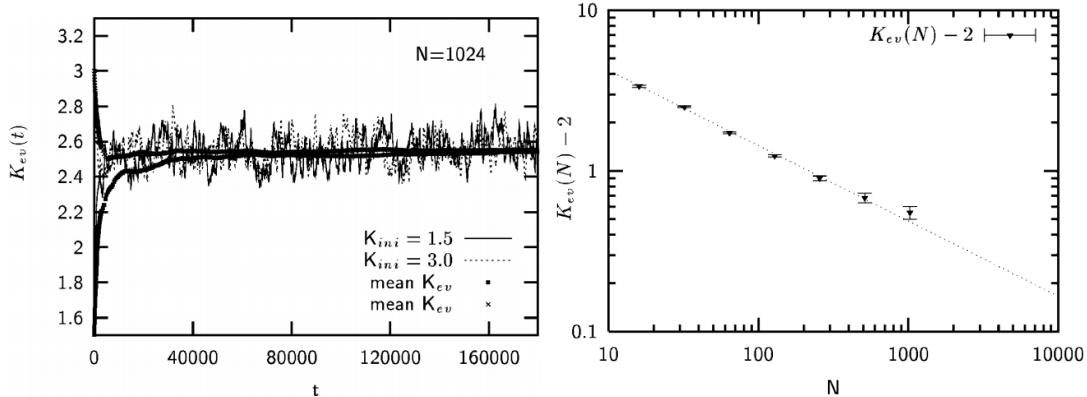


Figure 4.5: Evolution of the average degree K_{ev} with $N = 1024$ and varying initial degree $K_{ini} \in \{1.5, 3.0\}$ (left). Scaling of K_{ev} as a function of N (right). Reproduced from [73].

We see that the algorithm drives K_{ev} towards a certain value K_c independently of the initial degree of the network (Fig. 4.5, left). We also observe that K_{ev} scales as $K_{ev}(N) - 2 = cN^{-\delta}$ (Fig. 4.5, right). The idea behind this rule is that there is a phase transition between frozen and non-frozen dynamics. If the degree is too low, little input is exchanged and a large portion of the network remains frozen in a certain state. A high degree decreases the chance of a node being frozen, since in

³Since we are describing literature results, in this section we will break our naming convention by calling the average degree K_{ev} and $\langle K \rangle$ in two circumstances. This is done in order to maintain coherence with the figures.

this case it receives input from many nodes. This information is encoded in the size of the frozen component C . In Fig. 4.6 we have a plot of C as a function of average degree K and network size N for random networks.

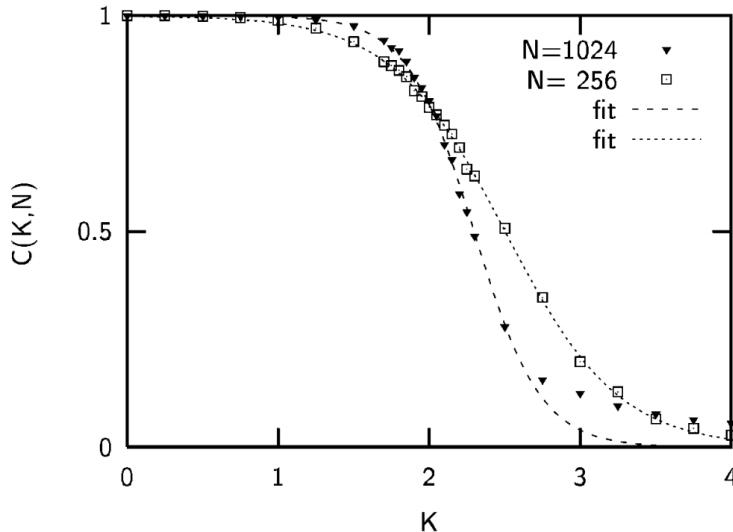


Figure 4.6: Frozen component C of the dynamics on a random network, as a function of the number of nodes N and average degree K . Each point is an average of 10^3 simulations. Reproduced from [73].

A good fit to the frozen component is given by $C(K, N) = \frac{1}{2} \{1 + \tanh [-\alpha(N)(K - K_0(N))]\}$, with the parameter K_0 scaling as $K_0(N) - 2 = \alpha N^{-\beta}$. This is the same scaling relationship of K_{ev} for the evolved network, even with similar parameters. Therefore, the adaptive algorithm drives the network towards the point where $C = 1/2$. Note that for $N \rightarrow \infty$ the transition between frozen and non-frozen dynamics becomes a second-order phase transition, and K_0 becomes the critical connectivity $K_c = 2$. Therefore, in this limit the algorithm drives the network towards the critical state. This self-organization is robust, and is an example of SOC behavior.

In a model related to the previous one, Bornholdt & Röhl explored SOC with a different dynamics[74]. This second model uses a stochastic dynamics instead of a deterministic one. The adaptive algorithm also uses local information instead of global information, and is based on node correlation instead of attractors. The behavior of the model is remarkably similar to the one from Bornholdt & Rohlf, with the critical connectivity now being $K_{ev}^{(2)} = 2.24 \pm 0.03$. This hints at a larger class of SOC models with similar behavior, while possessing different dynamics and details more suitable for different applications.

The last SOC model we will review was recently proposed by Rybarsch &

Bornholdt[75]. Instead of spin variables, the state of each node is given by a binary variable $\sigma_i(t) \in \{0, 1\}$. The probability of node i being at state $\sigma_i = 1$ at time t is given by

$$P[\sigma_i(t+1) = 1] = \frac{1}{1 + \exp[-2\beta(f_i(t) - 0.5)]} \quad (4.6)$$

where

$$f_i(t) = \sum_j w_{ij}\sigma_j(t) - \Theta_i \quad (4.7)$$

with β being an “inverse temperature” parameter and Θ_i a threshold parameter for each node. The probability of a node not being activated is $P[\sigma_i(t+1) = 0] = 1 - P[\sigma_i(t+1) = 1]$, and the link weights are $w_{ij} = \pm 1$. The network is directed. Note that in the limit $\beta \rightarrow \infty$, and for $\Theta_i = \Theta$, we recover the RTN dynamics studied in Chapter 3 with a threshold $h = \Theta + 0.5$. Since the model is probabilistic (with finite β) no initial activation is necessary - a node without input will spontaneously activate with probability $P = (1 + \exp(\beta))^{-1}$.

The model by Rybarsch & Bornholdt uses a local adaptive algorithm based on the Pearson correlation between nodes. The Pearson correlation C_{ij} between nodes i and j is defined as

$$C_{ij} = \frac{\langle \sigma_i(t+1)\sigma_j(t) \rangle - \langle \sigma_i(t+1) \rangle \langle \sigma_j(t) \rangle}{\Delta_i \Delta_j} \quad (4.8)$$

where Δ_i is the standard deviation of the time series of node i , $\vec{\sigma}_i = \{\sigma_i(1), \sigma_i(2), \dots\}$. Equation 4.8 measures the correlation between a node i at time $t+1$ and a node j at time t . A high correlation implies a high synchronization between j firing at time t and i firing at time $t+1$. The adaptive algorithm is described below:

1. Run the dynamics for $\tau = 100$ time steps
2. Select a random node i , and measure the average Pearson correlation of i with its neighbors, $\bar{C}_i = k_i^{-1} \sum_j |w_{ij}| C_{ij}$.
3. With equal probability, either add a new in-link to i (of weight $w_{ij} = \pm 1$) or remove a pre-existing in-link of i .
4. Run the dynamics again for $\tau = 100$ time steps, and measure the new average Pearson correlation of i , \bar{C}'_i .
5. If the correlation increased ($\bar{C}'_i > \bar{C}_i$), keep the rewiring. Otherwise revert the network to the original state.

6. Run the dynamics for more $t = 100$ steps to allow for a transient period, and iterate from step 1.

The idea behind the algorithm is to maximize the correlation between neighbors, taking into account a delay of one time step for signal transmission. It relies on random changes to the topology, that are kept only if they increase the correlation.

Three metrics are used to study the evolution of the network. In terms of topology, the metrics are the average degree $\langle K \rangle$ and the activating links ratio, i.e. the fraction of links in the network that are positive (our F_+ in Chapter 3). The metric used to study the dynamics is the branching parameter. The branching parameter λ is the average number of descendants of a perturbation, averaged over the entire network. More precisely, we calculate the average number of changes in the network dynamics at $t + 1$ after changing the state of a node i at time t , denoted by λ_i . The branching parameter is then $\lambda = \sum_i \lambda_i / n$. The branching parameter allows us to estimate the dynamical phase of the network. For $\lambda < 1$, perturbations tend to die out and the network is in an ordered, or sub-critical regime. For $\lambda > 1$ perturbations eventually take over the network and the network is in a chaotic, or super-critical regime. The case $\lambda = 1$ corresponds to the critical state, where a perturbation propagates but doesn't take over the network. In Fig. 4.7 we have the results for the algorithm with a zero threshold ($\Theta_i = 0 \forall i$) and $\beta = 5$, corresponding to a spontaneous activation probability of 0.7%.

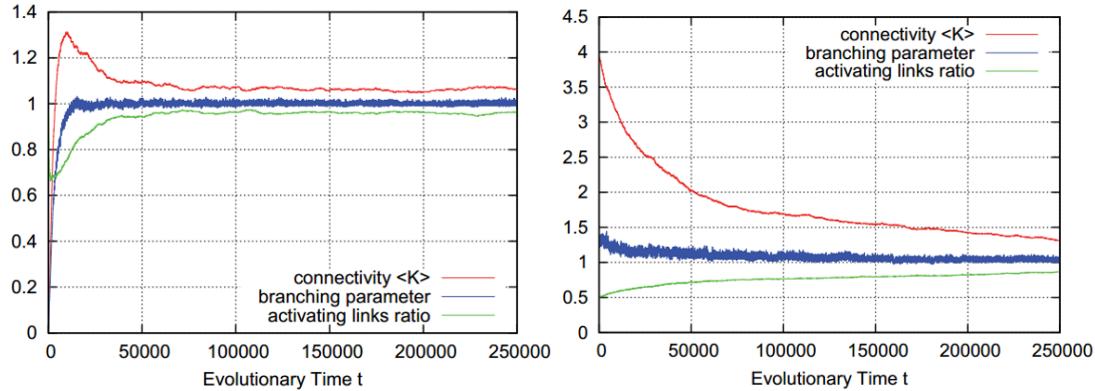


Figure 4.7: Time evolution of the network in the model by Rybarsch & Bornholdt. The parameters are $\beta = 5$ and $\Theta_i = 0 \forall i$. The initial network is a random network with average degree $K_{ini} = 0$ (left) and $K_{ini} = 4$ (right). Reproduced from [75].

We see that the network evolves towards a degree K_c higher than one, while the branching parameter stabilizes around $\lambda = 1$. The evolution is also independent of the initial network, and the activating links ratio p evolves towards $p = 1$. A

variation where p is kept fixed at $p = 0.8$ yields similar results, albeit with a higher degree[75]. One feature commonly associated with critical networks is avalanches of activity with a power-law distribution[68]. To study that, we can run the dynamics on the evolved network in the RTN limit ($\beta \rightarrow \infty$) and with $\Theta_i = 0 \forall i$. We start the dynamics by activating a random node at $t = 0$, and wait until the dynamics dies out (i.e., $\sigma_i(t) = 0 \forall i$). We measure the avalanche duration and the avalanche size. The avalanche duration T is defined as the number of time steps it stays alive. The avalanche size S is defined as the number of nodes that are activated at least once during the avalanche. In Fig. 4.8 we have the results of the avalanche analysis for the evolved network.

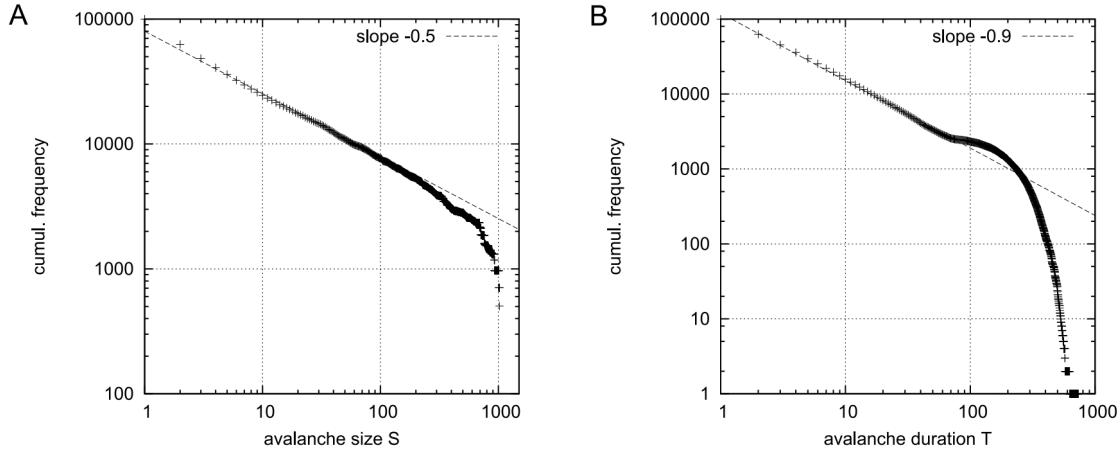


Figure 4.8: Avalanches of activity in the model by Rybarsch & Bornholdt. The avalanches were created by activating a single node on the evolved network. (A) Cumulative frequency of the avalanche size S . (B) cumulative frequency of the avalanche duration T . Results from 10^5 avalanches on 10 different networks of $n = 1024$ nodes. Parameters are $\beta \rightarrow \infty$ and $\Theta_i = 0 \forall i$. Reproduced from [75].

The probability distributions of avalanche size and duration follow power-laws $p(x) \sim x^{-\gamma}$. Moreover, the exponents γ are $\gamma_S = 1.5$ for the avalanche size and $\gamma_T = 1.9$. These are very close to the exponents expected for a critical network[76], which appear in a wide variety of critical models[68, 77, 78, 79].

Avalanches with power-law size distributions also appear in many natural phenomena, such as earthquakes[80]. In particular, the brain is hypothesized to be in a critical state[62]. In a 2003 paper, Beggs and Plenz[81] showed that spontaneous neural activity in cultured slices of rat cortex showed power law distributions for their avalanches. Since the avalanches were obtained without any specific external stimulation, the result suggested that the rat cortex was not only critical, but even organized itself to a critical state. This led to the hypothesis that the brain shows

Self-Organized Criticality. Similar results were later obtained *in vivo* for the rat cortex[82] and in awake rhesus monkeys[83].

The model by Rybarsch & Bornholdt is therefore an interesting minimal model to simulate large-scale brain dynamics. It possesses realistic properties such as critical dynamics, a Hebbian adaptive algorithm[84] and robustness (plasticity) to changes in the network. This robustness can be seen in Fig. 4.9, where a network is evolved with a higher threshold $\Theta_i = 1$, resulting in a higher degree $\langle K \rangle \approx 10$ and a branching parameter that fluctuates more around $\lambda = 1$. A sudden change in the threshold to $\Theta_i = 0$ causes a response from the network, decreasing $\langle K \rangle$ in order to reach the new critical state. The usefulness and simplicity of this model makes it a good starting point to more complex models. We develop this in Chapter 6, where we consider a similar model with deterministic (RTN) dynamics and spatial embedding.

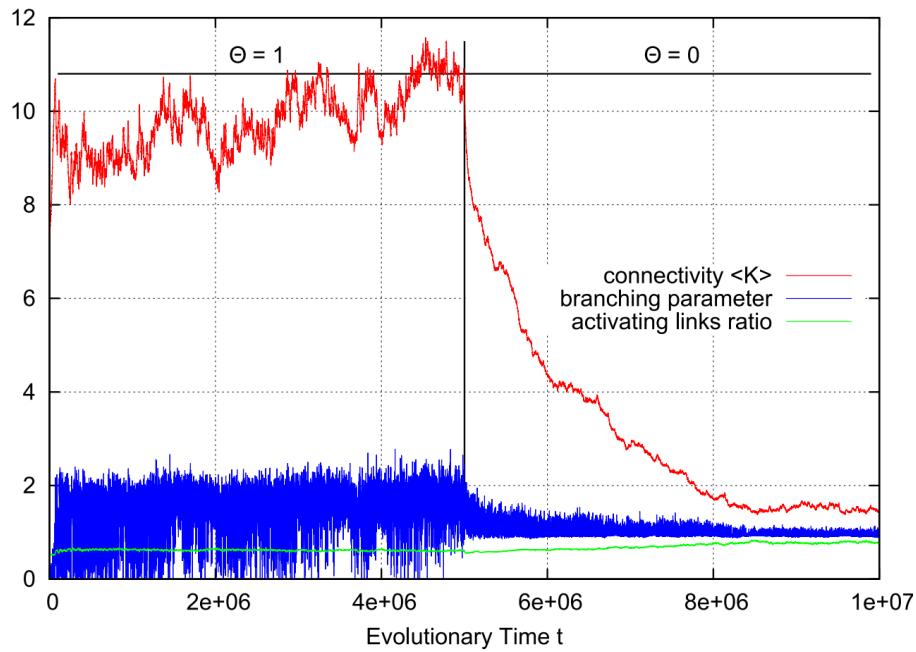


Figure 4.9: Network response to a change in threshold in the model by Rybarsch & Bornholdt. The network is evolved with a threshold $\Theta_i = 1$, resulting in a higher degree $\langle K \rangle$. A sudden change in the threshold to $\Theta_i = 0$ causes the network to respond, decreasing $\langle K \rangle$ and stabilizing in the new critical state. Reproduced from [75].

An Adaptive Model with Stable Dynamics⁵

5.1 Motivation and model definition

In this chapter we propose an adaptive network with threshold dynamics and a link-removal topological evolution. Our motivation is the synaptic pruning process in the mammalian brain, where during the adolescence a massive pruning process removes a large portion of the neuronal synapses[8]. This process is considered essential for the development of the brain[85]. Synapses in the brain can be either excitatory or inhibitory[56], depending on the type of neurotransmitter used¹. Moreover, the stimuli received by a neuron must be above a certain threshold in order for the neuron to fire. This is represented in Fig. 5.1, where we have the electrical potential of the neuronal membrane. If the received stimuli is not above the threshold potential (typically -55 mV) the neuron will not fire.

¹Specifically we are referring to chemical synapses, where chemical substances such as glutamate (excitatory) and GABA (inhibitory) are exchanged. Neurons can also communicate using electrical synapses, that use electrical current.

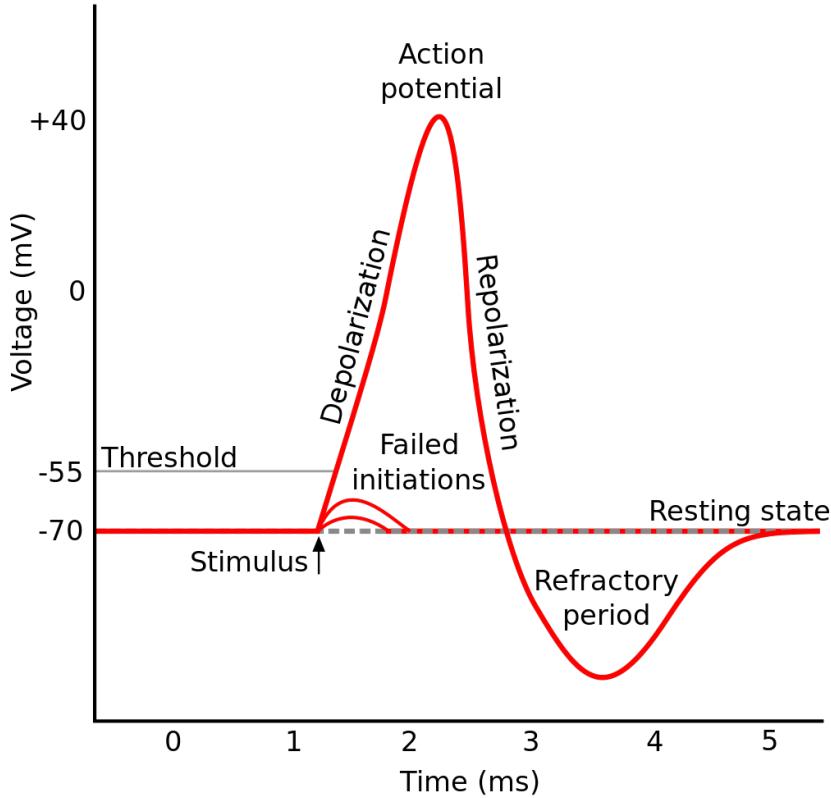


Figure 5.1: Representation of the membrane electrical potential of a neuron. The neuron requires stimuli above a certain minimum threshold in order to fire. Reproduced from the Wikimedia Commons file “File:Action potential.svg”. Available at http://commons.wikimedia.org/wiki/File:Action_potential.svg

Many realistic models of neuronal dynamics exist, such as the Hodgkin–Huxley model[86]. There are also models for the generation of realistic neural networks, for instance NETMORPH[87] and CX3D[88]. These models, however, are generally complex and include many experimental parameters. Our approach, here and in the next chapter, is to develop toy models and study which aspects of network dynamics and topology we can control using simple rules and a minimal number of parameters.

The neuron properties described above make the Random Threshold Network studied in Chapter 3 a good starting point to model neuronal dynamics. For completeness, let us define the dynamics used in the model. Consider a directed network with n randomly connected nodes and average degree K . Each existing link from node j to node i is assigned a weight $w_{ij} \in \{1, -1\}$, and the dynamical state of each node is given by a Boolean variable $\sigma_i(t) \in \{0, 1\}$. The state $\sigma_i(t+1)$ of node i at

time $t + 1$ is then given by

$$\sigma_i(t+1) = \begin{cases} 1 & , \text{ if } \sum_{j=1}^n w_{ij}\sigma_j(t) > h \\ 0 & , \text{ if } \sum_{j=1}^n w_{ij}\sigma_j(t) \leq h \end{cases} \quad (5.1)$$

where h is a threshold parameter. The control parameter of the topology is the fraction of positive links F_+ , calculated as

$$F_+ = \sum_{i,j} \operatorname{sgn}(w_{ij} + 1) / \sum_{i,j} |w_{ij}| \quad (5.2)$$

The dynamics starts with an initial condition $\vec{\sigma}(0) = \{\sigma_1(0), \sigma_2(0), \dots, \sigma_n(0)\}$ and is updated for each node in parallel afterwards. We are interested mainly in the network collective state, measured by the fraction of active nodes $A(t) = n^{-1} \sum_i \sigma_i(t)$. As discussed in Chapter 3, F_+ plays a major role in the dynamics of RTNs. Our algorithm will take advantage of this dependence in order to control the activity of the network. The average activity of node i is defined as

$$\langle \sigma_i \rangle = \frac{1}{\tau} \sum_{t=t_0}^{t=t_0+\tau} \sigma_i(t) \quad (5.3)$$

where τ is a time-scale parameter and t_0 is a transient time parameter. In the adaptive process, each $\langle \sigma_i \rangle$ is compared to a parameter α_i and changes its connectivity according to a certain set of rules. The important question is which set of rules, combined with a specific distribution of α_i , can produce a certain topology and network activity. In other words, the problem amounts to explore the class of adaptive threshold network with local, activity-based topological evolution. Here we take a simple realization of the mechanism with $\alpha_i = \alpha$ for all i , and the following rules: if $\langle \sigma_i \rangle > \alpha$ the node loses a positive in-link ($w_{ij} > 0$), and loses a negative in-link otherwise. In terms of an algorithm, we have:

1. At $t = 0$ activate a fraction $A(0)$ of the network nodes.
2. Run the dynamics for $t_0 + \tau$ time steps, and choose a node i .
3. Calculate the node's average activity $\langle \sigma_i \rangle$. If $\langle \sigma_i \rangle > \alpha$, remove a random positive in-link of i . Otherwise remove a random negative in-link. If there is no suitable link for removal, choose another i .
4. Iterate from step 2.

The above algorithm obviously can only run a finite number of cycles. If we start from a fully connected network with n nodes, after at most $n(n - 1)$ cycles the network will be void of links. We can control the average degree K of the network by running the algorithm an appropriate number of cycles. The algorithm will be run for $\tau_{cycle} = n(K_0 - K)$ cycles, where K_0 is the initial average degree of the network. The choices of t_0 and τ are not critical, and in fact after the initial activation ($t = 0$) t_0 plays no part at all. We therefore have an adaptive network with only two important parameters: the activity parameter α and the average degree K . In Fig. 5.2 we have a schematic representation of the algorithm.

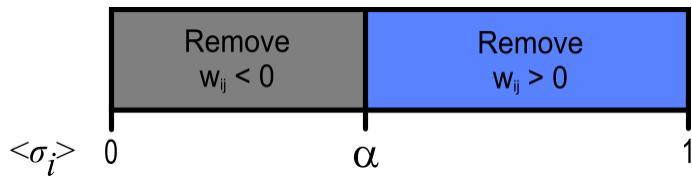


Figure 5.2: Schematic representation of the adaptive algorithm. A node i loses a positive in-link w_{ij} if its average activity $\langle \sigma_i \rangle$ is higher than α . Otherwise, i loses a negative in-link.

5.2 Simulation results

The idea behind these rules is to regulate network activity: the removal of an inhibitory in-link makes the node more likely to activate, and the removal of an excitatory in-link suppresses node activity. Therefore, the rules drive $\langle \sigma_i \rangle \rightarrow \alpha$, and require a highly connected initial network. The question is then if this adaptation produces a network with stable global dynamics (like the random network) or in a chaotic state. In Fig. 5.3 we have the results for a $n = 1024$ fully connected initial network, and varying initial positive link fraction $F_+(0)$ and evolutionary parameter α . We use $A(0) = 0.1$ as initial active network fraction.

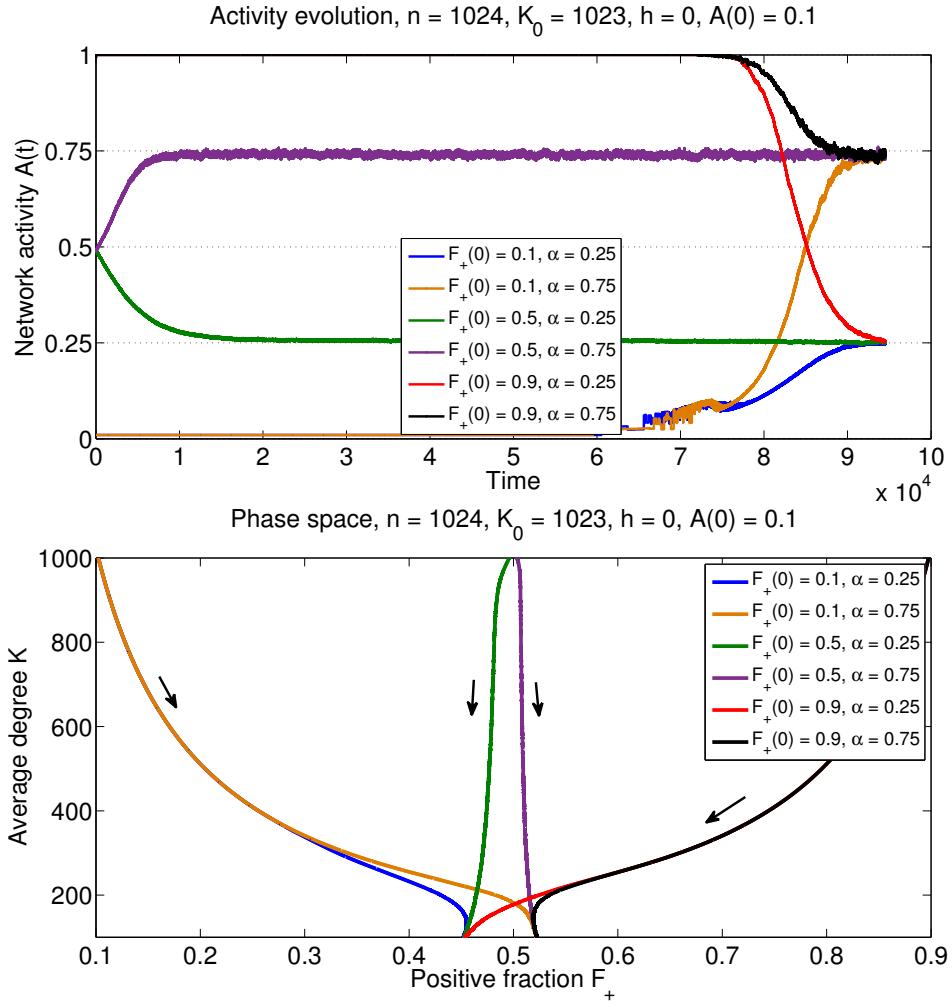


Figure 5.3: Network activity for initial $F_+(0) \in \{0.1, 0.5, 0.9\}$ and $\alpha \in \{0.25, 0.75\}$. Each point is the average value of $A(t)$ during the cycle of $t_0 + \tau$ time steps. (top) Corresponding evolution in the parameter space. The arrows indicate the direction of the evolution. (bottom) Other parameters are $h = 0$, $n = 1024$, and $A(0) = 0.1$.

The evolution robustly drives the network towards a state where $A_\infty = \alpha$. There is a strong resemblance between the path the evolution follows in the $A(t) = \alpha$ phase (Fig. 5.3, bottom) and the activity curves² in the RTN parameter space (Fig. 3.2). This suggests the topology of the evolved network is at least similar to that of a random network. Our simple adaptive rule is therefore capable of controlling an important aspect of threshold networks, namely the average activity A_∞ . By running the algorithm for τ_{cycle} evolutionary steps we are also able to set the network's average degree, given by $K = K_0 - \tau_{cycle}/n$. This is done with only two important parameters and for a vast array of initial networks, provided that

²Defined as the curves in Fig. 3.2 with a fixed A_∞ .

they are sufficiently dense.

We now extend the model to various α_i within the same network. This tests the robustness of the algorithm, and answers the question whether the entire network needs to be at the same A_∞ in order for the dynamics to be stable. In Fig. 5.4 we show the network activity $A(t)_i$ for all the groups with $\alpha_i \in \{0, 0.25, 0.75, 1\}$. The evolution still drives the network towards $A(t \rightarrow \infty)_i = \alpha_i$. This means the algorithm is able to control the activity of individual groups within the network, and not only of the global network state.

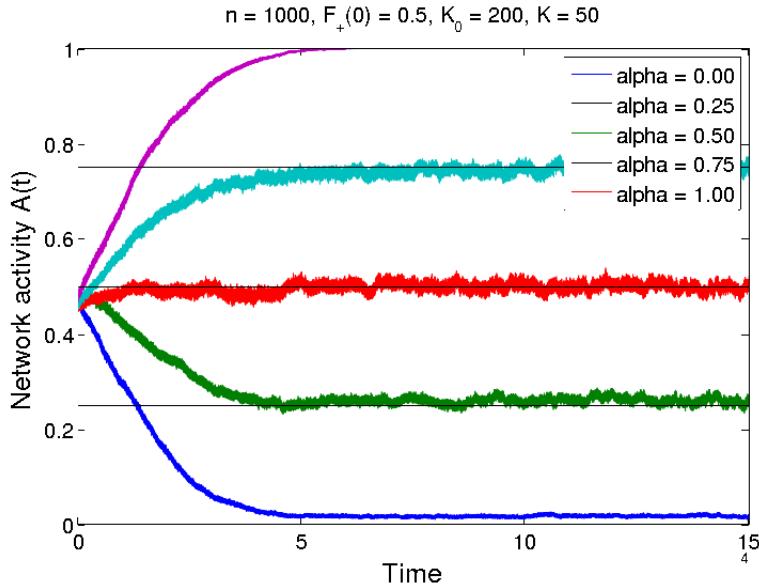


Figure 5.4: Network activity for many $\alpha_i \in \{0, 0.25, 0.75, 1\}$ in the same network, with $n = 1000$, $F_+(0) = 0.5$, $K_0 = 200$ and $K = 50$.

5.3 Spatial embedding

An important aspect of many real-world networks is that they are embedded in space. In brain networks, in particular, spatial embedding is a very important constraint because there is a higher energy expenditure in forming and maintaining long range synapses[62]. Other examples where space is relevant include power grid networks and epidemiological networks[89].

Adaptive network models with spatial constraints are comparatively rare. They usually have a richer and more complicated dynamics, making analysis more difficult. As the model described in Section 5.1 is relatively simple, we can attempt to put it into space and study its emerging properties. To do so we distribute the nodes randomly in a 2D space, assigning coordinates (x_i, y_i) for every node i . This

choice is for simplicity: the nodes can also be put into a lattice or another specific configuration without noticeable changes to the results. The adaptive algorithm is then modified in the following way: when an in-link of i is removed, instead of randomly, the selection of w_{ij} follows a probability $P_i^{rem}(j)$. This probability is given by

$$P_i^{rem}(j) \sim \left(\frac{d_{ij}}{d_{i,max}} \right)^\beta \quad (5.4)$$

where d_{ij} is the Euclidean distance between neighbors i and j , $d_{i,max} = \max\{d_{i,j}\}_{j=1}^n$ is the largest distance to a neighbor and $\beta \geq 0$ is a distance parameter. In other words, the algorithm favors the removal of long range links. The reasoning is that in most applications there is a higher wiring cost associated with larger distances between the nodes. The parameter β mediates how biased this selection is: for $\beta = 0$ we recover the out-of-space case, whereas for $\beta \rightarrow \infty$ we always remove the link from the farthest neighbor.

We are then interested whether the spatial adaptive algorithm maintain the $A_\infty = \alpha$ result from Section 5.2, and if the evolved network has any interesting topological property. In Fig. 5.5 we have the topological evolution comparing the random ($\beta = 0$) and spatial network ($\beta = 10$) cases for $\alpha = 0.8$. We choose a low degree $K = 10$ in order to accentuate any possible difference between the cases. If no difference is found it is unlikely for it to appear for higher K . The spatial constraint does not inhibit the evolution towards $A_\infty = \alpha$ or increase the fluctuations around it. In other words, the spatial embedding does not really affect the dynamics of the evolved network. The important result here is that they are compatible, which is caused by the robustness of the RTN dynamics. A variant of this spatial embedding rule (Eq. 5.4) will play a larger role in our other model in Chapter 6.

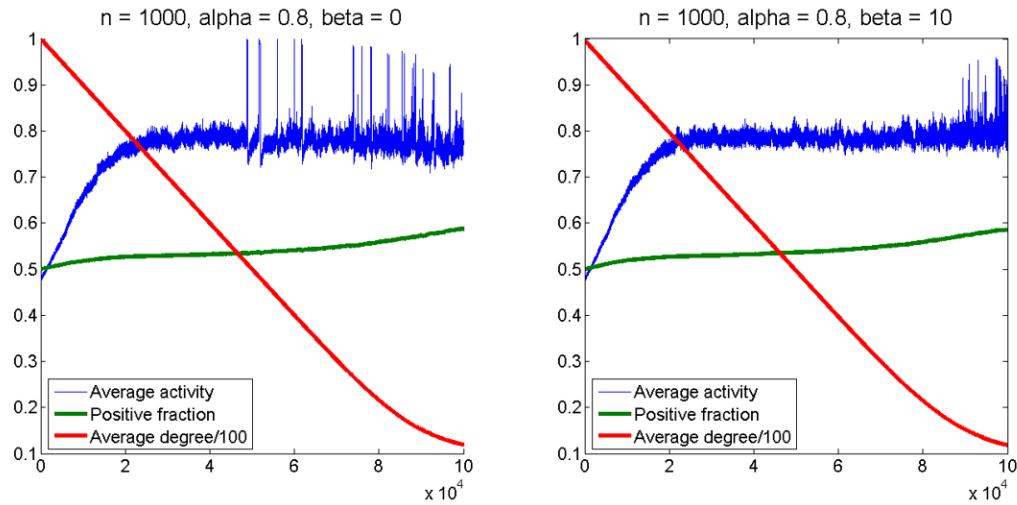


Figure 5.5: Topological evolution for a network without spatial constraints (left) and with spatial constraints (right). The parameters are $\alpha = 0.8$, $n = 1000$, $K_0 = 100$ and $K = 10$.

In many networks the nodes are organized in groups, specially in spatial networks. In order to investigate the communities in the evolved network we will use the modularity Q , clustering coefficient C and average path length L . These measures were defined in Chapter 2. Our interest in Q comes from the fact that many real-world networks have been shown to possess a modular structure. In particular, the (hierarchical) modular structure of the human brain is considered essential for its functionality[62]. In order to understand the results from these metrics they must be compared to the results from a random network. What exactly constitutes a random network varies from case to case. Here we take it to be a randomized version of the original network with the same in and out-degree distribution, in the so-called configuration model.

In Fig. 5.6 we show the topology of an evolved network, with network measures for the network and its randomized version. The modularity Q of the evolved network is very high, but more importantly, it is higher than the one of the randomized network. The evolved network is also a small-world network with high clustering, as $C/C_r = 4$ but $\langle L \rangle / \langle L \rangle_R = 1.16$. This is similar to the result from the Watts-Strogatz model from Section 2.4, but the evolved network is also modular.

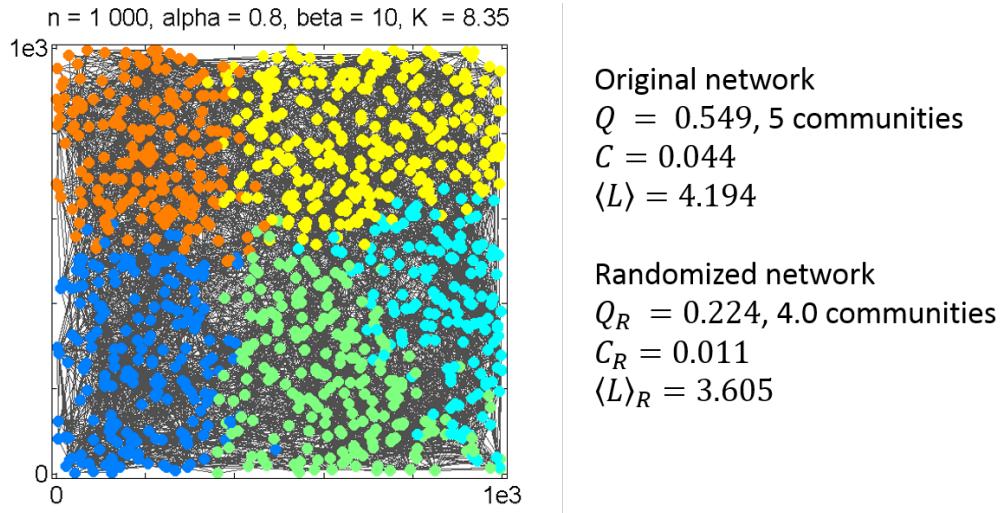


Figure 5.6: Evolved topology of a network and measures of it. The color of each node corresponds to its assigned module. The parameters are $n = 1000$, $\alpha = 0.8$, $\beta = 10$ and $K = 8.35$.

We are interested in how these measures scale as a function of β . In Fig. 5.7 we show Q , C and L for $\beta \in [0, 10]$. As β increases the network becomes increasingly modular, saturating around $\beta = 10$. The modularity for the randomized network is stable at around $Q_R \approx 0.2$, showing that Q is really a feature of the topological structure and not of the degree distribution. It is important to note that this happens for a small number (< 10) of communities. Our algorithm is therefore able to control the modularity of the network with only one parameter. The behaviors for C and L are similar. As β increases the relative clustering $C(\beta)/C(0)$ increases, while the relative average path length $L(\beta)/L(0)$ remains stable. This is an interesting result because brain networks are known to be both modular and small-world[90].

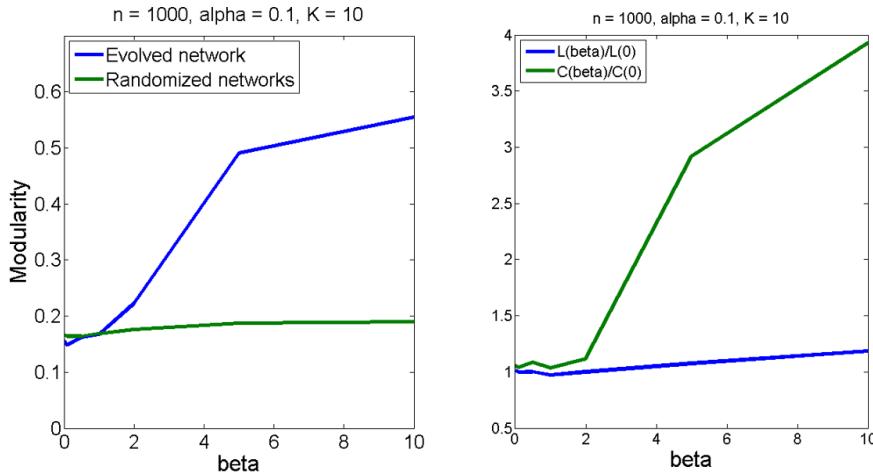


Figure 5.7: Modularity Q for the evolved and randomized networks (left) and relative average path length $L(\beta)/L(0)$ and clustering $C(\beta)/C(0)$ (right). The other parameters are $n = 1000$, $\alpha = 0.1$ and $K = 10$.

In Chapter 3 we studied the dynamics of random threshold networks, and developed a mean-field approximation to generate RTNs with specific activity A_∞ . One could question the usefulness of the model developed here, considering the availability of a simpler method to generate networks with a specific A_∞ . To begin with, the adaptive algorithm is capable of evolving very different initial networks towards $A_\infty = \alpha$, instead of using a network with finely-tuned parameters K and F_+ . The adaptive algorithm is also able to control the activity of individual groups, coexisting within the same network as seen in Fig. 5.4.

There are also differences in the two models regarding the frequency distribution of activity. To explain these differences, let us define the standard deviation of the active fraction as $\Delta_1 = \sqrt{\sum_{t=1}^{\tau} (A(t) - A_\infty)^2 / \tau}$, and the standard deviation of $\langle \sigma_i \rangle$ as $\Delta_2 = \sqrt{\sum_{i=1}^n (\langle \sigma_i \rangle - A_\infty)^2 / n}$. The evolved network was generated using our algorithm with a single $\alpha_i = \alpha$. The ER network was generated by using the mean-field approximation to find the K and F_+ necessary for $A_\infty = \alpha$. We see that Δ_1 measures the fluctuations of $A(t)$, and Δ_2 measures the heterogeneity of activity between nodes. In Fig. 5.8 we compare the evolved and Erdos-Renyi networks for these two measures. We see that the evolved network is less stable regarding the active fraction $A(t)$, with a higher deviation from A_∞ . At the same time, the standard deviation of $\langle \sigma_i \rangle$ is much lower in the evolved network. This means the evolved network has a much more homogeneous distribution of activity than the ER network. Therefore, the most useful model (adaptive or ER) depends on which aspect of network dynamics one wants to emphasize.

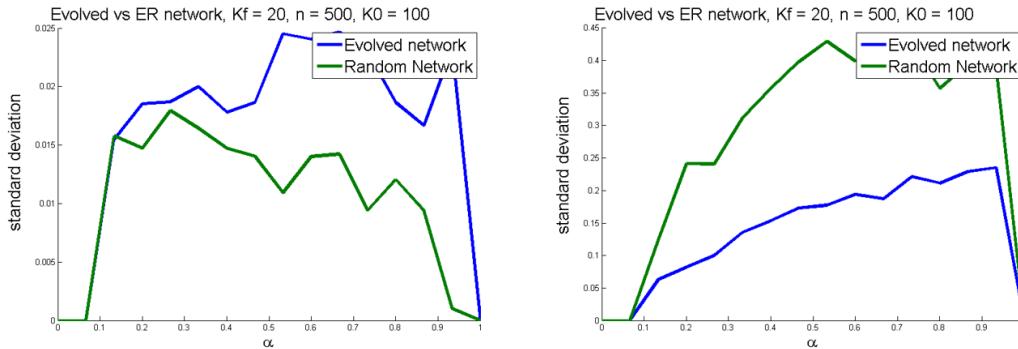


Figure 5.8: Standard deviations of $A(t)$ (left) and $\langle \sigma_i \rangle$ (right) for the evolved and Erdos-Renyi networks with the same activity, as a function of α . Parameters are $n = 500$, $K = 20$ and $K_0 = 100$ for the evolved network.

5.4 Summary

In this Chapter we developed an adaptive network capable of evolving an initial dense network to a certain activity $A_\infty = \alpha$. Our base adaptation algorithm (Section 5.1) is able to control the average activity of the network or groups of nodes in the network (Section 5.2). By running the algorithm a certain number of times τ_{cycle} , we are also able to set the evolved network’s degree K . The algorithm is then extended to spatial networks, where by adding another parameter β we are able to control the modular and small-world properties of the network. Our final model then possesses three relevant parameters, each directly linked to an important property of the evolved network:

- The parameter α controls the average activity $A_\infty = \alpha$.
- The parameter τ_{cycle} gives the evolved network’s degree $K = K_0 - \tau_{cycle}/n$.
- The parameter β controls how modular and clustered the evolved network is, while still being a small-world network.

We consider this a small number of parameters for a model of a spatial, adaptive network. The model can be used to generate threshold networks with specific features. This can be a useful tool for modeling large-scale networks, for instance genetic regulatory networks[52] and neural networks.

6 A Model with Controllable Topology from External Inputs

6.1 Motivation and model definition

In this chapter we will describe a model where external inputs are used in order to influence the topology of the network. While the model in Chapter 6 was inspired by the synaptic pruning process, the model in this chapter is inspired by the synaptic growth in the brain. Newborn babies experience an explosive growth in density of synapses during the first years after birth, as seen in Fig. 6.1. Meanwhile, as already mentioned, many experiments report a mode of activity in the form of avalanches of activity in the brain[81, 82, 83].

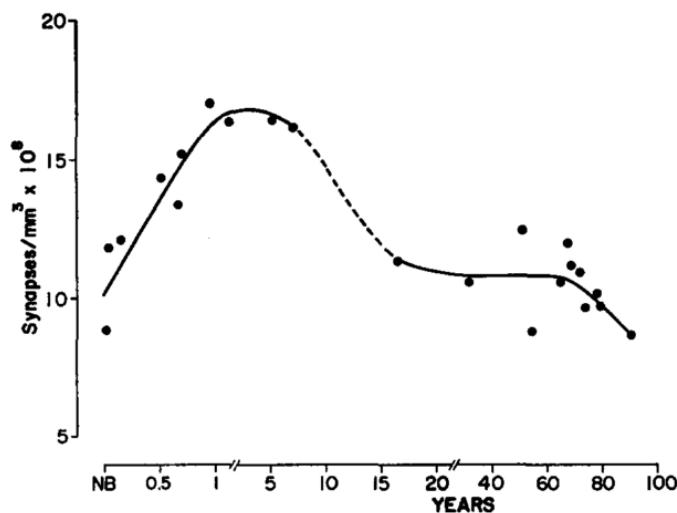


Figure 6.1: Density of synapses in a portion of the human brain. Samples from 21 different brains. Reproduced from [7].

Our goal is to develop an adaptive model where the network connectivity grows from a network without links, and study how can we control its topology and dynamics. In order to influence this growth we will use external inputs. These inputs can be linked to stimuli from the environment in the developing brain. Stimuli during the synaptic growth is considered essential for the development of the brain[91]. In a famous experiment, Hubel & Wiesel[92] deprived newborn kittens of light in one eye for the first three months of life. This induced blindness in that particular eye on the kittens. Similar results were later obtained in rhesus monkeys[93]. These results are caused by the lack of development of synapses in the visual cortex. This lack of synaptic growth has been linked to the absence of stimuli during the developmental phase of the subjects.

For the node dynamics, we will use the Random Threshold Network dynamics explored in Chapter 3. Since we are interested in avalanches of activity, we will modify the RTN dynamics by increasing the threshold after each activation. This guarantees the eventual death of the dynamics in the network, characterizing an avalanche. The dynamics is the following: each node has a binary dynamical state $\sigma_i(t) \in \{0, 1\}$ and an individual activation threshold Θ_i , with a deterministic threshold update rule given by

$$\sigma_i(t+1) = \begin{cases} 1 & , \text{if } \sum_j w_{ij} A_j(t) > \Theta_i(t) \\ 0 & , \text{if } \sum_j w_{ij} A_j(t) \leq \Theta_i(t) \end{cases} \quad (6.1)$$

The threshold Θ_i is also a dynamical variable, with the following dynamics

$$\Theta_i(t+1) = \Theta_0 + \sum_{t'=0}^t \sigma_i(t') \quad (6.2)$$

where Θ_0 is an initial threshold value. This increase of the threshold Θ_i after an activation has a biological motivation. Chemical synapses use neurotransmitters to transfer signal, which can deplete and impede the neuron from firing. Depletion of neurotransmitters has been linked to alertness and sleep deprivation in humans[94].

At $t = 0$ the network receives an external input in the form of spontaneous activation of a fraction n_A of its nodes, and each node is then updated in parallel following Eqs. 6.1 and 6.2. According to Eq. 6.2, whenever a node gets activated its threshold Θ_i will go up by one. This is enough to eventually cease activity on the network. Model variants that maintain the increasing of Θ_i after activations produce very similar results, so the precise form of Eq. 6.2 is not crucial. The two

main dynamical observables are the fraction of active nodes $A(t) = n^{-1} \sum_i \sigma_i(t)$, and the average threshold $\Theta(t) = n^{-1} \sum_i \Theta_i(t)$. Note that the network dynamics (after initial activation) is parameter-less and deterministic.

The topological evolution is based on the Pearson correlation (Eq. 4.8) between a node i at time t and its inputs $j \in \{k \mid w_{ik} \neq 0\}$ at time $t - 1$, and the repeated activation of the network following a certain fixed pattern. The pattern is characterized by the fraction of nodes it activates, $n_A = n^{-1} \sum_i A_i(0)$. The network starts without links, and the adaptive algorithm at each step has the following algorithm:

1. Activate a fraction n_A of the network, and run the dynamics according to Eqs. 6.1 and 6.2.
2. After the avalanche ends, choose a node i and calculate the average Pearson correlation between it and its inputs $j \in \{k \mid w_{ik} \neq 0\}$.
3. With equal probability, add a random in-link $w_{ix} = \pm 1$ to i or remove an existing in-link from i .
4. Run the dynamics again with the same pattern, and calculate the new average Pearson correlation of i .
5. If the rewiring caused a drop in average correlation, revert the change. Otherwise keep it. Iterate from step 1.

Topological and dynamical variables are constantly measured in order to characterize the network state. Since the dynamics starts with an external input, we are interested in the effect of a fixed input pattern versus a random input pattern. In Fig. 6.2 we have a schematic representation of the adaptive algorithm of the model.

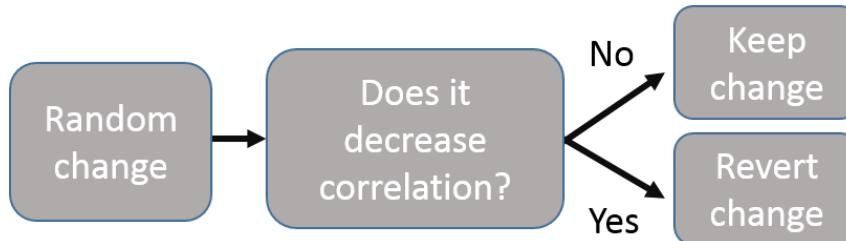


Figure 6.2: Schematic representation of the adaptive algorithm.

Notice the similarity between the adaptive algorithm here and the one from the model by Rybarsch & Bornholdt[75]. The model in this Chapter was actually inspired by the model by Rybarsch & Bornholdt, and started out as an effort to

simplify the mentioned model. To do this we changed the dynamics, using a deterministic rule instead of a stochastic one and removing a parameter. This allowed us to add spatial embedding to the model, which will be explored in Section 6.3. In the next Section we describe the results from the model without spatial embedding.

6.2 Activity avalanches from external inputs

If we keep a fixed input pattern, a network structure where correlation between neighbors is maximal for that specific input emerges. This usually occurs for one neighbor, since with more links it is harder to properly synchronize activity and the correlation drops. In Fig. 6.3 we have the results for such case with $n_A = 0.01$ (i.e. 1% of input nodes). Since the network starts without links every node is equivalent, and the choice of input nodes is not relevant. The average branching parameter λ and average Pearson correlation C (as defined in Section 4.3) are then calculated over $\tau = 20$ time steps. The particular τ used affects the absolute values of λ and C , but should not affect how they change during the evolution.

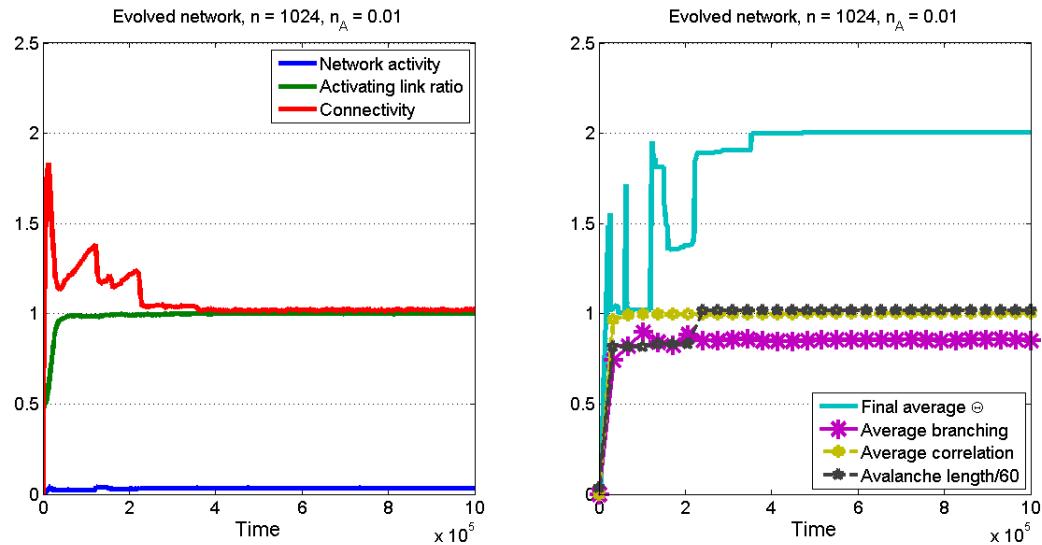


Figure 6.3: Evolution of a network with a fixed input pattern. Topological measures and average activity (left) and dynamical measures (right). Parameters are $n = 1024$, $n_A = 0.01$ and $\Theta_0 = 0$.

After a period of initial growth where the connectivity K reaches a maximum, connectivity drops and eventually stabilizes around $K_C = 1$. This can be explained by the initial correlation. Since C is initially low, every change is accepted and connectivity explodes¹. As C increases, the rewiring becomes more selective. Con-

¹While link addition and removal occur with equal probability, the link needs to exist to be

nectivity drops, and almost all inhibitory links are removed. In terms of dynamics, both correlation and the branching parameter stabilize at a high value. The average threshold Θ at the end of the dynamics stabilizes around two, meaning that on average every node gets activated twice during the avalanche.

In order to focus on the networks response to external stimuli we used a deterministic dynamics without perturbations. It is however interesting to study whether the evolved network has any critical properties regarding random perturbations. To study this we measure the length T (in time steps) and size L (in number of nodes activated) of avalanches caused by activation of a random node. The network used has $n = 1024$ and was evolved with a $n_A = 0.001$ input pattern for 10^6 time steps.

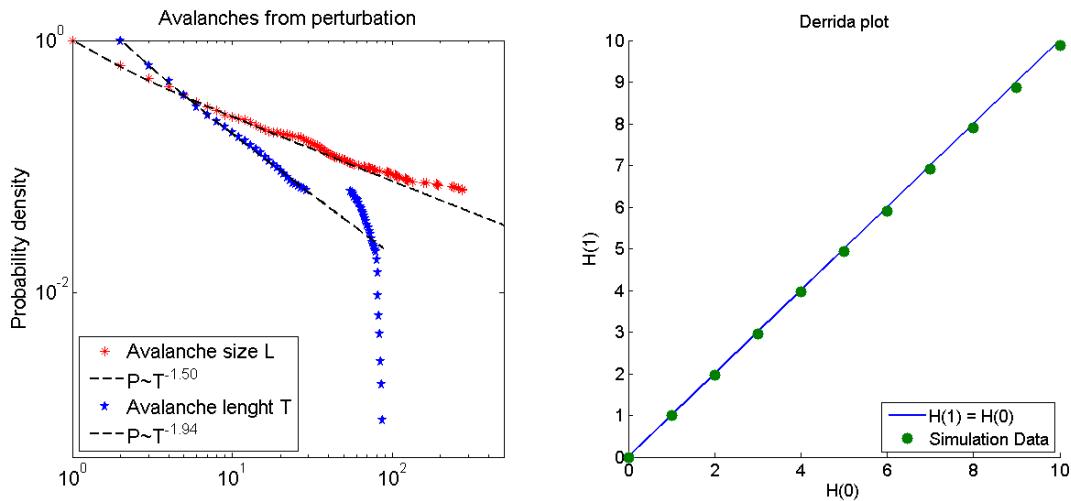


Figure 6.4: Avalanche probability distribution from random perturbations on an evolved network. Data from 10^6 avalanches. (left) Derrida plot of the network avalanches. Data from 10^4 avalanches, perturbed at different time steps (right). The network was evolved with $n = 1024$, $n_A = 10^{-3}$ and $\Theta_0 = 0$.

The results in Fig. 6.4 (left) show that the avalanches follow a power-law with an exponential cutoff. The best-fit exponent values are compatible with the ones predicted for a critical network. Power-laws don't necessarily imply Criticality, however. In order to confirm that the network is critical we can make a Derrida plot of the dynamics, as defined in Section 3.5. We see in Fig. 6.4 (right) that the response to perturbations follows the bisection $H(1) = H(0)$. The match between the data and the bisection confirms the critical nature of the network regarding random perturbations. It is important to note that the critical state is sometimes not found by the evolution. In these cases, the connectivity stabilizes at a higher value removed. This creates a bias towards network growth.

between one and two and the network possesses an overcritical response to random perturbation.

All activity comes from an external input pattern. Therefore it is useful to analyze the network response to noisy patterns, such as partial input patterns. We calculate the Hamming distance $H(T_h)$ between the network with full input and with various degrees of partial inputs, at time $t = T_h$. In order to get more meaningful results, a network with a higher pattern size $n_A = 0.1$ is used. The results are shown in Fig. 6.5.

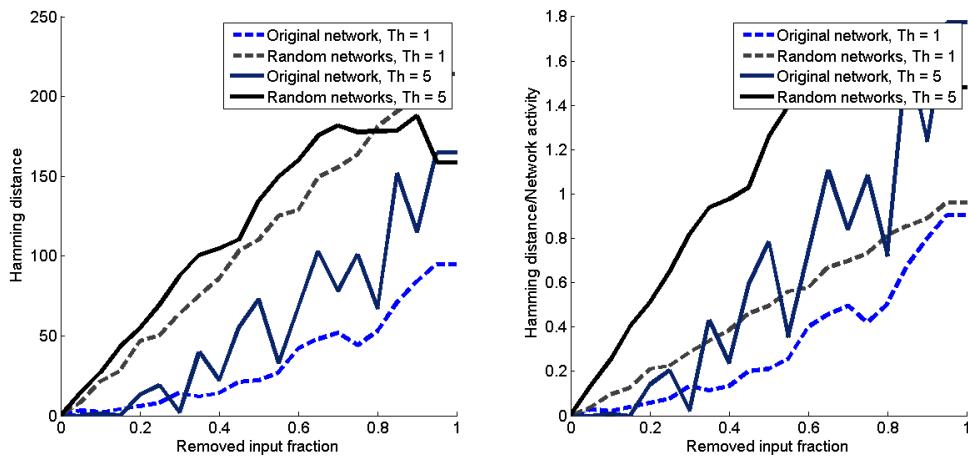


Figure 6.5: Hamming distance for the original network and randomized networks for $t = T_h \in \{1, 5\}$ (left) and Hamming distance divided by the network activity with full input at $t = T_h$ (right). Each point is an average of 10^4 runs for the original network and 3.10^3 runs for each of the 20 random networks.

In order to have a measure of robustness, the network is compared against an ensemble of random networks with the same (in and out) degree distribution. The evolved network generates a configuration considerably closer (lower $H(T_h)$) to the one with full input than the random networks. This is especially true for small perturbations ($\approx 10\%$ of removed input). It is important to note that the results in Fig. 6.5 are for the input pattern used to evolve the network. If instead a random pattern is used, the results from the evolved and random networks are indistinguishable. This shows that the behavior in Fig. 6.5 is a response to a specific pattern, and not a general characteristic of the network structure. Also, if a random (alternating) pattern is used to evolve the network an overcritical and low-correlation topology appears.

6.3 Controlling the topology of a spatial network

The relative simplicity of the model described in Sections 6.1 and 6.2 allows us to include spatial embedding to it. For that, we will use a rule similar to the one in Section 5.3, but that also accounts for link inclusion. Each node i is associated with coordinates (x_i, y_i) , randomly chosen between 0 and 1. We maintain the algorithm from Section 6.1, but include the following change: if a link is to be removed from node i , the selection of which neighbor j will have the link removed is performed with probability $P_i^{rem}(j)$ given by

$$P_i^{rem}(j) \sim \left(\frac{d_{ij}}{d_{i,max}} \right)^\beta \quad (6.3)$$

where $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ is the Euclidean distance between i and j . The distance is divided by the largest distance $d_{i,max}$ to a neighbor². The bias towards long-range removal is done by the parameter β . Setting $\beta = 0$ recovers the out-of-space model, and $\beta \rightarrow \infty$ results in the algorithm always choosing the farthest neighbor. If a link to i is to be added by the adaptive algorithm, the selection of a non-neighbor j is done with probability $P_i^{add}(j)$ given by

$$P_i^{add}(j) \sim \left(1 - \frac{d_{ij}}{d_{i,max}} \right)^\beta \quad (6.4)$$

which is similar to Eq. 6.3. It introduces a bias towards the creation of close-range links, with the same parameter β . We also make the following modification: instead of just not decreasing the correlation, now a change must *increase* the correlation in order to be accepted. This is done in order to remove the explosive growth phase of Fig. 6.3 (left), which allows us to better control the formation of the topology.

The embedding of the network in 2D space facilitates our visualization of the evolving topology and dynamics. Since our algorithm uses an initial input to start the dynamics, it also allows us to study the response of the network regarding different activation regions and patterns. Since each node has a different position (x_i, y_i) , they are not equivalent in the beginning. Therefore, the choice of nodes for initial activation matters. This contrasts with the out-of-space model, where each node is equivalent at $t = 0$ and the only relevant parameter is the fraction of chosen nodes n_A .

²This done for simulation purposes. For high β the value of d_{ij}^β can go below the numerical precision of the computer, generating rounding errors and invalidating the simulation

In Fig. 6.6 we have the results of a network embedded. The evolution was done by activating a small number of nodes at the center (i.e., coordinates $(0.5, 0.5)$) of the plane. The color of the nodes corresponds to a community division found using a modularity maximization method³.

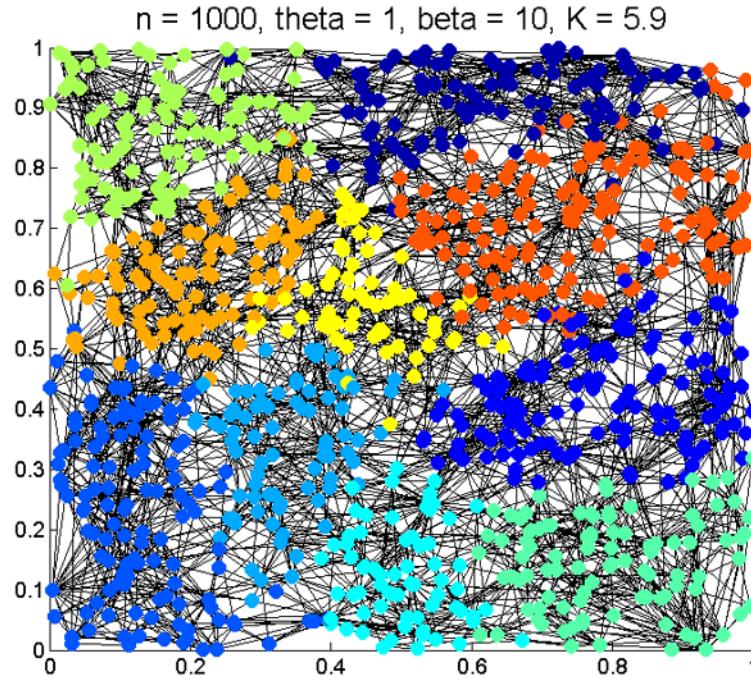


Figure 6.6: Example of a network evolved with spatial embedding. The network was evolved by activating a small number of nodes at the center. The node colors corresponds to the community division found by maximizing the modularity. Parameters are $n = 1000$, $\Theta = 1$ and $\beta = 10$.

We analyzed the modularity Q , clustering coefficient C and average path length L of the network. Similar to Chapter 5, we compare the results to an ensemble of randomized networks created by shuffling the links of the evolved network. Below we summarize the results.

- Modularity: $Q = 0.716$ with 10 communities (evolved network), and $Q_r = 0.272$ with an average of 4.6 communities (randomized networks).
- Clustering coefficient $C = 0.112$ (evolved network), and $C_r = 0.007$ (randomized network)

³Specifically, we used the Louvain algorithm described in Section 2.3 followed by a finetune algorithm[95].

- Path length $L = 6.26$ (evolved network), and $L_r = 4.07$ (randomized network)

Similar to the results from Section 5.3, we find a highly modular community structure. Compared to the randomized network, we find a much higher clustering with a similar average path length. We therefore have a small-world topology⁴ with high clustering, similar to the model in Chapter 5 and to the Watts-Strogatz model[14].

In order to visualize how the dynamics spreads, in Fig. 6.7 we show the active nodes at different time steps. We observe a concentric wave of activity spreading from the nodes activated by external input. The wave eventually reaches the borders of our system, terminating the dynamics. This is a direct result of the spatial constraints - if $\beta = 0$ a node can activate another anywhere in the network and no such pattern of activity is seen.

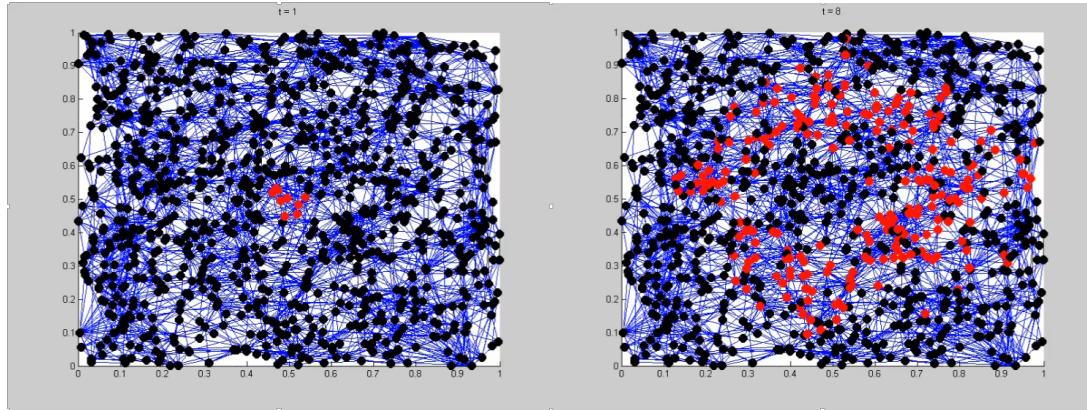


Figure 6.7: Dynamics of the evolved network. The red points represent activated nodes, and the black points represent inactive nodes. Dynamical states at $t = 1$, showing the nodes with external input (left), and at $t = 8$, showing how the activity spreads through the network.

We can also study how the evolved network changes with varying β . This is shown in Fig. 6.8, for a variety of topological and dynamical measures. We see that both average degree K and avalanche length T increase (Fig. 6.8, upper left). The average correlation stays relatively constant, while both the average final threshold $\Theta = \langle \Theta_i(T) \rangle$ and average activity $\bar{A} = T^{-1} \sum_{t=1}^T A(t)$ increase with β (Fig. 6.8, upper right). These results indicate that the avalanches both take longer and result in higher activity for higher β . We also see that the relative clustering C/C_r increases dramatically, whereas the relative path length L/L_r stays constant⁵ (Fig. 6.8, lower

⁴Rigorously, we would also have to show that L scales as $L \sim \ln(n)$ or slower.

⁵We use here the same nomenclature as earlier, where C_r and L_r are the clustering and path length of randomized versions of the evolved network.

left). The modularity Q of the evolved network increases, stabilizing at $Q \approx 0.7$ for high β . At the same time, the modularity of the randomized network decreases with β , which can be attributed to the higher degree (Fig. 6.8, lower right).

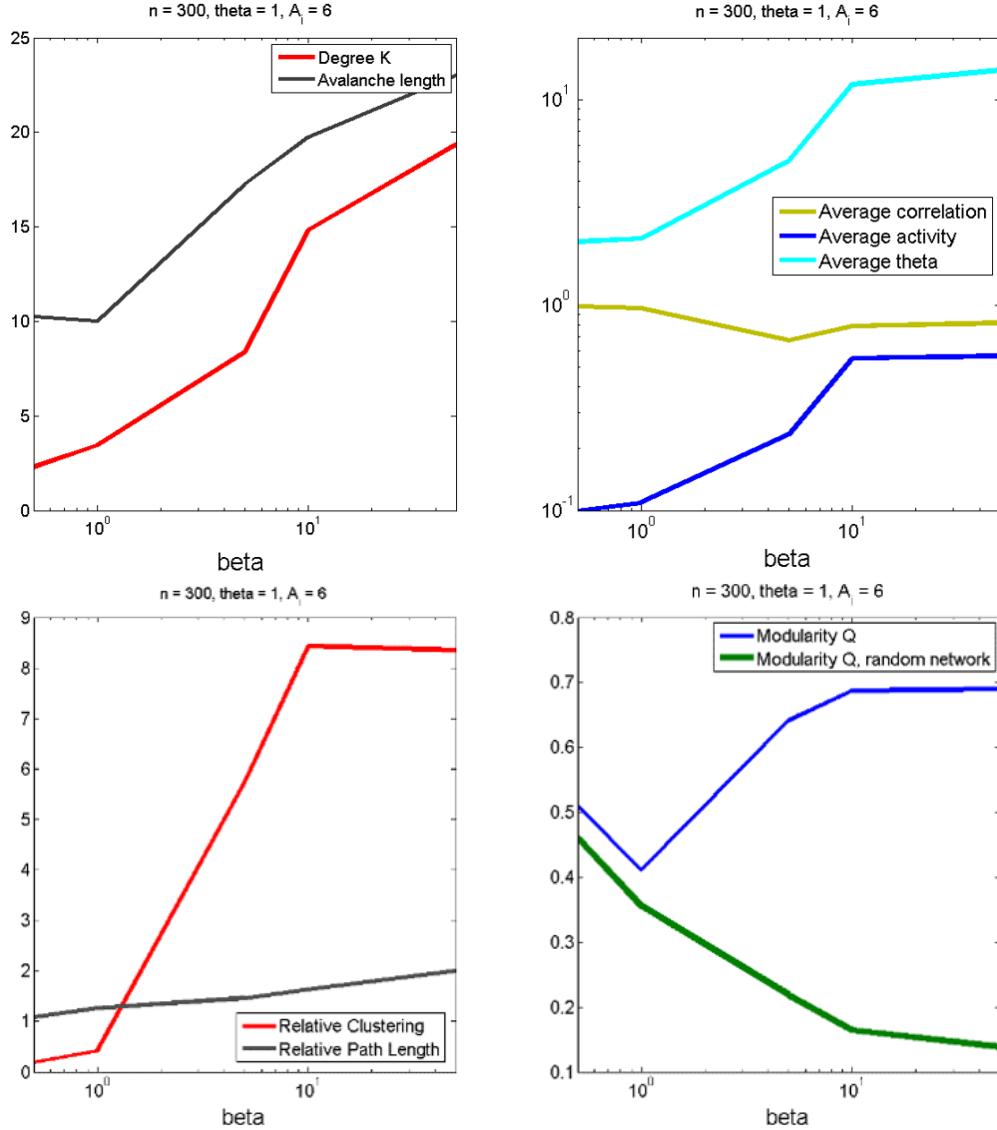


Figure 6.8: Network evolution as a function of β . In the upper figures we have the degree K and avalanche length (upper left), and average correlation C , average activity during the avalanche and average final $\Theta = \langle \Theta_i \rangle$ (upper right). In the lower figures we have the relative clustering C/C_r and relative path length L/L_r (lower left), and the modularity Q for the evolved and randomized networks (lower right). Parameters are $n = 300$, $\Theta_0 = 1$ and 6 initial input nodes ($n_A = 0.02$).

Regarding Criticality, preliminary results indicate that fine-tuning of β can drive the evolving network towards a critical state. This is seen in Fig. 6.9, where we have

the Derrida plot for an ensemble of evolved networks. A higher β implicates in a stronger response to perturbation. In the networks with $n = 500$ and $n_A = 0.012$ the response is critical with $\beta = 1.75$, being undercritical for a lower β and overcritical for higher β (Fig. 6.9, right). At the same time, for networks evolved with $n = 300$ and $n_A = 0.02$ the response still increases with β , but it is always undercritical. The mechanism that leads to these two behaviors remains to be explored. Regardless of this, we see that β is able to heavily influence both the dynamics and topology of the network.

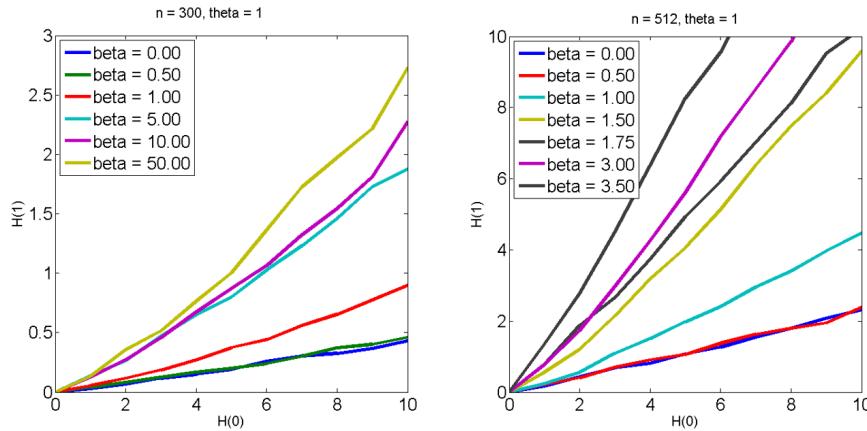


Figure 6.9: Derrida plot of the evolved network, for various β . Both networks have $\Theta_0 = 1$. Results for an ensemble of networks with $n = 300$ and $n_A = 0.02$ (left) and $n = 512$ and $n_A = 0.012$ (right).

We can further extend our model by implementing different patterns for the external input. This is done by alternating between two or more patterns instead of keeping a fixed pattern at each avalanche. In Fig. 6.10 we have the results for a network evolved with three alternating patterns. The input patterns are made of two nodes close to coordinates $(0, 0)$, $(1, 0)$ and $(0.5, 1)$. We see that the alternating patterns create three distinct modules, each corresponding to the input region of a pattern. This shows that the algorithm is able to control the formation of distinct topologies in the network, even with nodes spread randomly in space.

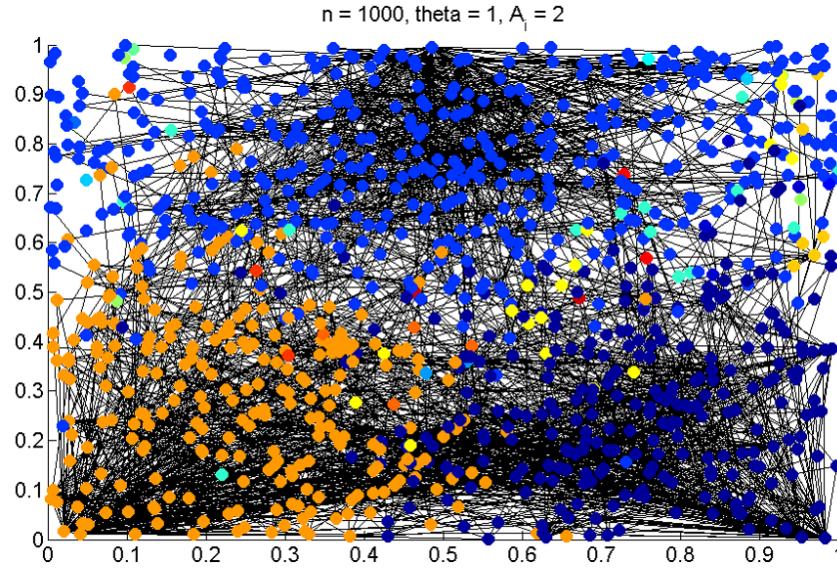


Figure 6.10: Topology of a network evolved with three input regions at coordinates $(0,0)$, $(1,0)$ and $(0.5,1)$. The colors represent the community detection found by a modularity maximization method. Parameters are $n = 1000$, $\Theta_0 = 1$ and $n_A = 0.002$ for each site.

It is also useful to analyze how this topology grows during the evolution of the network. In Fig. 6.11 we show three snapshots of the topology at different stages of the evolution. We clearly see that the network grows from the input regions, and forms a very heterogeneous topology with few very high degree nodes and a large number of low-degree nodes.

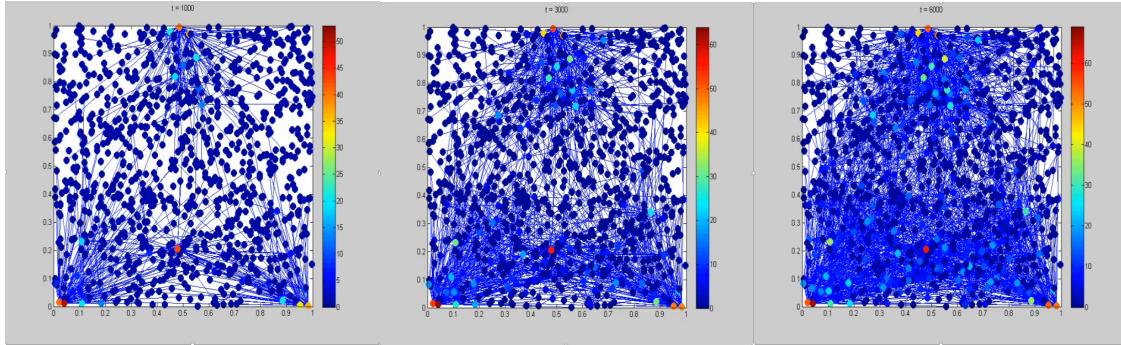


Figure 6.11: Topology of the evolved network with three input patterns at three different stages. The temperature (color) represents the degree of a node. Topology at $t = 1000$ (left), $t = 3000$ (center) and $t = 6000$ (right).

Besides a complex topology, the evolved network also possesses another hallmark of adaptive networks, namely complex dynamics. To show this, we can plot the

avalanches from each of the patterns and compare how they spread. We compare the dynamics at three different time steps: at $t = 0$, at the time when it reaches its maximum activity, and at $t = 9$. In Fig. 6.12 we have the results from the avalanche starting from the input region at $(0.5, 1)$.

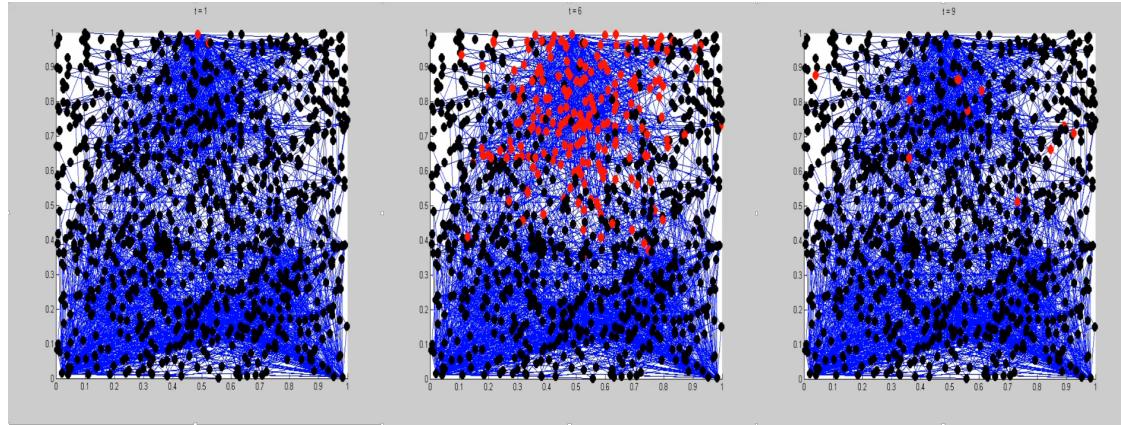


Figure 6.12: Avalanche dynamics of the evolved network starting from the input site at $(0.5, 1)$. Dynamics at $t = 1$ (left), during the maximum activity at $t = 6$ (center) and at $t = 9$ (right).

We see that the dynamics remains largely contained within the community generated by the pattern. In Fig. 6.13 we have the avalanche starting from the input region at $(1, 0)$.

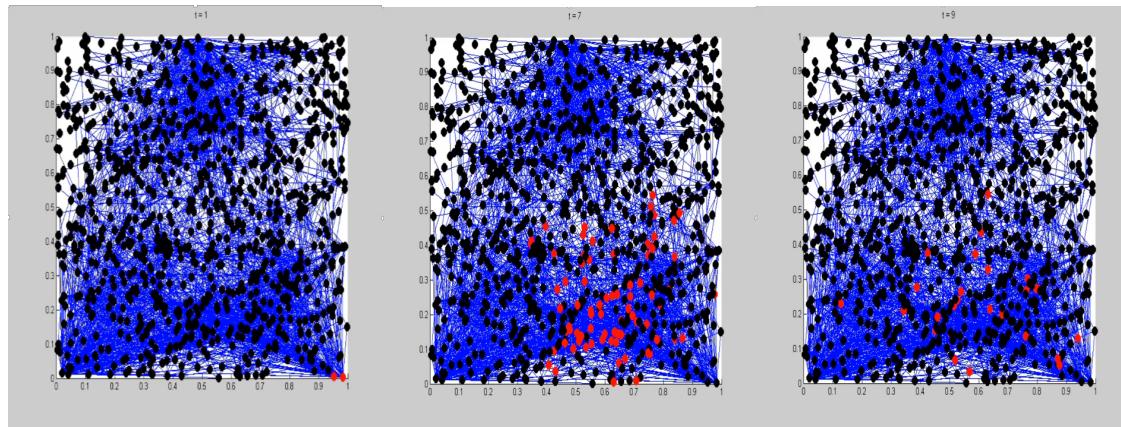


Figure 6.13: Dynamics of the evolved network starting from the input site at $(1, 0)$. Dynamics at $t = 1$ (left), during the maximum activity at $t = 7$ (center) and at $t = 9$ (right).

We also see that the dynamics stays contained within the module generated by the pattern. We also see that the total activity starting from this pattern is lower.

Lastly, in Fig. 6.14 we have the avalanches starting from the pattern with the input region at $(0, 0)$.

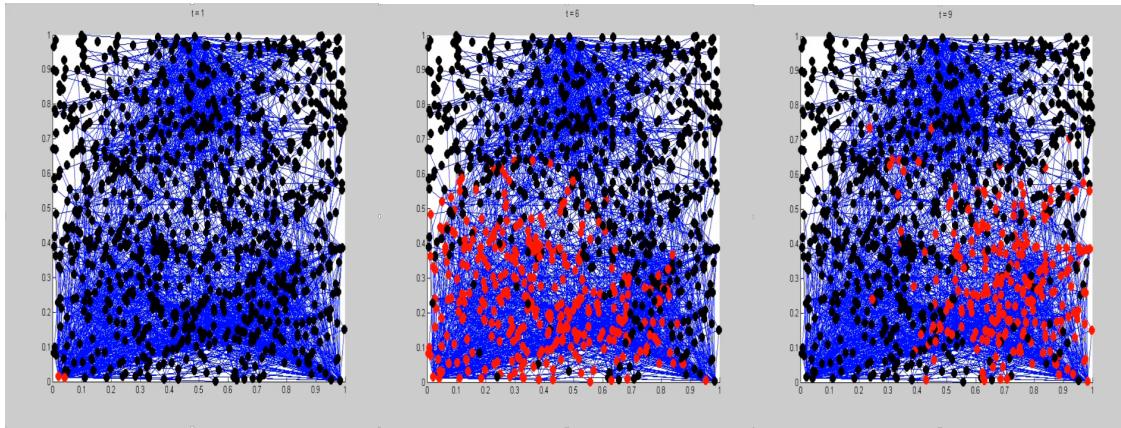


Figure 6.14: Avalanche dynamics of the evolved network starting from the input site at $(0, 0)$. Dynamics at $t = 1$ (left), during the maximum activity at $t = 6$ (center) and at $t = 9$ (right).

We see that the dynamics started from $(0, 0)$ is able to activate a second module of the network. This behavior was a consequence of the interplay between dynamics and topology, and choice of input regions. It is important to note that all of these results were obtained with simple alternating patterns and nodes spread randomly in space. With more clever patterns, and by arranging the positions of the nodes, we speculate that a much more complete control of the topology is possible. This in turn can generate more complex dynamics, all while maintaining the relatively simple adaptive algorithm and dynamics, and minimal number of parameters.

6.4 Summary

In this chapter we proposed a model where avalanches driven by external inputs were used to control the evolution of a network. Without spatial constraints, the evolved networks display some interesting properties. They are adaptive networks with deterministic dynamics and a minimal number of parameters, of which only the base threshold Θ_0 is critical. The evolved networks generally display Self-Organized Criticality (Fig. 6.4), and are also more resilient to partial input than random network (Fig. 6.5).

When put in space, a more interesting picture emerges. With only one new parameter β we are able to control various topological and dynamical properties of the network (Fig. 6.8). These include the degree K , modularity Q and small-world properties. Preliminary results indicate that Criticality may also be possible

with fine-tuning of β (Fig. 6.9). By introducing alternating patterns, we are able to go further and control the formation of modules in the network (Fig. 6.10). This in turn can generate interesting dynamics, as observed in Figs. 6.12-6.14. Concluding, we consider this a promising work-in-progress model, where a relatively simple adaptive algorithm and a minimal number of parameters were already able to generate interesting dynamics and control the topology.

Conclusion

In this work we studied the structure and dynamics of complex networks. The motivation of this study was the search for a simple dynamics that was also capable of describing the neuron or large regions of neurons. We were particularly interested in developing a dynamics of neural networks describing cortical brain areas that could emulate their behavior at early ages when synaptic growth and pruning takes place.

Our first object of investigation was the Random Threshold Network (RTN) model. These networks have a simple dynamics adequate to our goals. More specifically, the dynamics can be controlled by the balance between excitatory and inhibitory links in the network. We also developed a mean-field approximation to our dynamics. This approximation builds upon previous works by being more simple and using functions available in Algebra packages such as Mathematica(R). Thus, our approximation facilitates the generation of Random Threshold Networks with specific activity A_∞ and degree K . We also show that A_∞ goes through a first-order phase transition.

As a second step, based on the RTNs, we considered two adaptive models for the networks. With our first adaptive model we are able to control the dynamics of a network with very simple rules. This model aimed at a simple description of the synaptic pruning process in the human brain. Our goal while developing it was to keep it as simple as possible, while at the same time trying to maintain it compatible with our knowledge of the biological neural network. We also proposed a simple rule of pruning in spatial networks, and show that this rule is able to generate modular topologies. It also generates small-world topologies with high-clustering, similar to the WS model. These are features present in brain networks, so it is interesting that we are able to obtain and control them with a simple pruning rule and one

parameter β .

Our second adaptive model dealt with controlling the topology of a network and its avalanche dynamics. It aimed at a simple description of the synaptic growth of the brain during infancy, as opposed to the synaptic pruning. Using a correlation maximization algorithm, we are able to generate a self-organized critical network. The topology is also somewhat robust against faulty input after being evolved. Our model uses external inputs to generate activity, which is similar to the influence of stimuli from the environment on the brain. We then extended the model by embedding the network in a 2D space. By using a simple distance-based rule, we were able to control the formation of modules in the network. We showed that this complex topology can also give rise to interesting dynamics in the network.

From our study, we conclude that Random Threshold Networks are very robust in terms of their dynamics. Adaptive models based on RTN dynamics benefit from this robustness. These models can alter the network in ways that could throw other less robust dynamical models into a chaotic state. We also conclude that simple rules, coupled with a minimal number of parameters, are able to create interesting and non-linear characteristics. Most importantly, many of these characteristics are seen in real-world networks.

The models developed in this work are intentionally minimal, and can be extended in many directions. The model with ceaseless dynamics (Chapter 5) can be altered to preserve the degree distribution of the base network, or to generate networks with specific degree distribution. For instance, scale-free networks can be generated by using a preferential depletion rule for link removal[96]. This model can also be extended with features like link addition to balance the removal, and deletion of nodes. These features could for instance drive the network towards a stable topology, where the network does not loose more links.

Our second model also offers many possibilities for further extensions. We could explore different input patterns, and the effect of different spatial constraints. For instance, the effect of embedding the network in 3D space. We could change our algorithm to encourage the formation of inhibitory links, which could dramatically change the dynamics. However, it is not clear if it is possible control this formation using simple rules or input patterns.

Brain modeling is a field of its own, and there are many big projects that aim to model and study the human brain[97, 98, 99]. These projects tend to involve very complicated and realistic models, that are run on supercomputers[100]. We believe that one of the most important contributions of this work is to show that some

basic properties of neural networks, such as average activity and modular topology, can be reproduced by simple models. These simple models are useful to acquire an understanding of the general principles in the large networks and we hope will be useful as a complementary tool in the investigation of real structures in the human brain neural network.

Bibliography

- [1] T Ryan Gregory, James a Nicol, Heidi Tamm, Bellis Kullman, Kaur Kullman, Ilia J Leitch, Brian G Murray, Donald F Kapraun, Johann Greilhuber, and Michael D Bennett. Eukaryotic genome size databases. *Nucleic acids research*, 35(Database issue):D332–8, January 2007.
- [2] Eui-Ki Kim, Jong Hyeon Seok, Jang Seok Oh, Hyong Woo Lee, and Kyung Hyun Kim. Use of hangeul twitter to track and predict human influenza infection. *PloS one*, 8(7):e69305, January 2013.
- [3] M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256, January 2003.
- [4] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, January 2002.
- [5] S Boccaletti, V Latora, Y Moreno, M Chavez, and D Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308, February 2006.
- [6] Peter Dayan and L F Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. 2001.
- [7] Huttenlocher Peter R. Synaptic density in human frontal cortex - Developmental changes and effects of aging. *Brain Research*, 163(2):195–205, March 1979.
- [8] P R Huttenlocher and C de Courten. The development of synapses in striate cortex of man. *Human neurobiology*, 6(1):1–9, 1987.

- [9] Peter R Huttenlocher. *Neural plasticity : the effects of environment on the development of the cerebral cortex*. 2002.
- [10] M.E. J. Newman. *Networks*. Oxford University Press, March 2010.
- [11] Luciano da F. Costa, Francisco A Rodrigues, and Alexandre S Cristino. Complex networks: the key to systems biology. *Genetics and Molecular Biology*, 31(3):591–601, 2008.
- [12] Philip M Kim, Jan O Korbel, and Mark B Gerstein. Positive selection at the protein network periphery: evaluation in terms of structural constraints and cellular context. *Proceedings of the National Academy of Sciences of the United States of America*, 104(51):20274–9, December 2007.
- [13] Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: uses and interpretations. *NeuroImage*, 52(3):1059–69, September 2010.
- [14] D J Watts and S H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–2, June 1998.
- [15] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews. Neuroscience*, 10(3):186–98, March 2009.
- [16] M E J Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103(23):8577–82, June 2006.
- [17] E. Leicht and M. Newman. Community Structure in Directed Networks. *Physical Review Letters*, 100(11):118703, March 2008.
- [18] Tiago P. Peixoto. Hierarchical Block Structures and High-Resolution Model Selection in Large Networks. *Physical Review X*, 4(1):011047, March 2014.
- [19] Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, July 2009.
- [20] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–8, June 2005.

- [21] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–4, August 2010.
- [22] M Girvan and M E J Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–6, June 2002.
- [23] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5):75–174, February 2010.
- [24] Ulrik Brandes. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30(2):136–145, May 2008.
- [25] M.E. J. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54, January 2005.
- [26] Ulrik Brandes and Daniel Fleischer. Centrality Measures Based on Current Flow. In Volker Diekert and Bruno Durand, editors, *STACS 2005*, volume 3404 of *Lecture Notes in Computer Science*, pages 533–544. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [27] Sergio Gómez, Pablo Jensen, and Alex Arenas. Analysis of community structure in networks of correlated data. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 80(1 Pt 2):016114, July 2009.
- [28] Roger Guimerà, Marta Sales-Pardo, and Luís Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2):025101, August 2004.
- [29] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72(2):027104, August 2005.
- [30] Mursel Tasgin, Amac Herdagdelen, and Haluk Bingol. Community Detection in Complex Networks Using Genetic Algorithms. page 6, November 2007.
- [31] Ken Wakita and Toshiyuki Tsurumi. Finding Community Structure in Megascale Social Networks. page 9, February 2007.
- [32] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, October 2008.

- [33] Josep M. Pujol, Vijay Erramilli, and Pablo Rodriguez. Divide and Conquer: Partitioning Online Social Networks. page 7, May 2009.
- [34] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5):056117, November 2009.
- [35] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, October 1999.
- [36] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic Spreading in Scale-Free Networks. *Physical Review Letters*, 86(14):3200–3203, April 2001.
- [37] Paul Erdős and Alfred Rényi. On random graphs. *Publicationes Mathematicae*, 6(1):290–297, November 1959.
- [38] S N Dorogovtsev, J F Mendes, and a N Samukhin. Structure of growing networks with preferential linking. *Physical review letters*, 85(21):4633–6, November 2000.
- [39] Derek De Solla Price. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science*, 27(5):292–306, September 1976.
- [40] G Bianconi and Albert-László Barabási. Competition and multiscaling in evolving networks. *Europhysics Letters (EPL)*, 54(4):436–442, May 2001.
- [41] Sn Dorogovtsev and Jf Mendes. Evolution of networks with aging of sites. *Physical Review E*, 62(2 Pt A):1842–5, August 2000.
- [42] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–82, July 2000.
- [43] A. Barrat and M. Weigt. On the properties of small-world network models. *The European Physical Journal B*, 13(3):547–560, January 2000.
- [44] L a Amaral, A Scala, M Barthelemy, and H E Stanley. Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America*, 97(21):11149–52, October 2000.
- [45] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, December 1943.

- [46] Maria Davidich and Stefan Bornholdt. Boolean network model predicts cell cycle sequence of fission yeast. *PLoS one*, 3(2):e1672, January 2008.
- [47] Maria Davidich and Stefan Bornholdt. The transition from differential equations to Boolean networks: a case study in simplifying a regulatory network model. *Journal of theoretical biology*, 255(3):269–77, December 2008.
- [48] S.A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22(3):437–467, March 1969.
- [49] Leo Kadanoff, Susan Coppersmith, and Maximino Aldana. Boolean Dynamics with Random Couplings. *Complexity*, 7(2):69, April 2002.
- [50] Jorge G. T. Zañudo, Maximino Aldana, and Gustavo Martínez-Mekler. Boolean Threshold Networks: Virtues and Limitations for Biological Modeling. In Samuli Niiranen and Andre Ribeiro, editors, *Information Processing and Biological Systems*, volume 11 of *Intelligent Systems Reference Library*, pages 113–135. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [51] Agnes Szejka, Tamara Mihaljev, and Barbara Drossel. The phase diagram of random threshold networks. *New Journal of Physics*, 10(6):063009, June 2008.
- [52] Fangting Li, Tao Long, Ying Lu, Qi Ouyang, and Chao Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, 101(14):4781–6, April 2004.
- [53] Rui-Sheng Wang and Réka Albert. Effects of community structure on the dynamics of random threshold networks. *Physical Review E*, 87(1):012810, January 2013.
- [54] Gordon M. Shepherd, editor. *The Synaptic Organization of the Brain*. Oxford University Press, January 2004.
- [55] G J Brewer, M D Boehler, R A Pearson, A A DeMaris, A N Ide, and B C Wheeler. Neuron network activity scales exponentially with synapse density. *Journal of neural engineering*, 6(1):014001, February 2009.
- [56] Henry Markram, Maria Toledo-Rodriguez, Yun Wang, Anirudh Gupta, Gilad Silberberg, and Caizhi Wu. Interneurons of the neocortical inhibitory system. *Nature reviews. Neuroscience*, 5(10):793–807, October 2004.

- [57] B Derrida and Y Pomeau. Random Networks of Automata: A Simple Annealed Approximation. *Europhysics Letters (EPL)*, 1(2):45–49, January 1986.
- [58] V. R. R. Uppuluri and W. J. Blot. Asymptotic Properties of the Number of Replications of a Paired Comparison. *Journal of Applied Probability*, 11(1):43–52, 1974.
- [59] Christoph Fretter, Agnes Szejka, and Barbara Drossel. Perturbation propagation in random and evolved Boolean networks. *New Journal of Physics*, 11(3):033005, March 2009.
- [60] Steven H. Strogatz. From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D: Nonlinear Phenomena*, 143(1-4):1–20, September 2000.
- [61] B Derrida. Dynamical phase transition in nonsymmetric spin glasses. *Journal of Physics A: Mathematical and General*, 20(11):L721–L725, August 1987.
- [62] Olaf Sporns. *Networks of the Brain*. MIT Press, Cambridge, 2010.
- [63] Thilo Gross and Hiroki Sayama, editors. *Adaptive Networks: Theory, Models and Applications*. Springer, 2009.
- [64] Thilo Gross and Bernd Blasius. Adaptive coevolutionary networks: a review. *Journal of the Royal Society, Interface / the Royal Society*, 5(20):259–71, March 2008.
- [65] Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. *Proceedings of the twenty-second annual symposium on Principles of distributed computing - PODC '03*, pages 347–351, 2003.
- [66] Thilo Gross, Carlos D'Lima, and Bernd Blasius. Epidemic Dynamics on an Adaptive Network. *Physical Review Letters*, 96(20):208701, May 2006.
- [67] Takaaki Aoki and Toshio Aoyagi. Scale-Free Structures Emerging from Co-evolution of a Network and the Distribution of a Diffusive Resource on it. *Physical Review Letters*, 109(20):208702, November 2012.
- [68] Per Bak, Chao Tang, and Kurt Wiesenfeld. Self-organized criticality: An explanation of the 1/f noise. *Physical Review Letters*, 59(4):381–384, July 1987.

- [69] Ronald Dickman, Miguel A. Muñoz, Alessandro Vespignani, and Stefano Zapperi. Paths to self-organized criticality. *Brazilian Journal of Physics*, 30(1):27–41, March 2000.
- [70] Zeev Olami, Hans Feder, and Kim Christensen. Self-organized criticality in a continuous, nonconservative cellular automaton modeling earthquakes. *Physical Review Letters*, 68(8):1244–1247, February 1992.
- [71] Per Bak, Kan Chen, and Chao Tang. A forest-fire model and some thoughts on turbulence. *Physics Letters A*, 147(5-6):297–300, July 1990.
- [72] Kim Sneppen. Self-organized pinning and interface growth in a random medium. *Physical Review Letters*, 69(24):3539–3542, December 1992.
- [73] Stefan Bornholdt and Thimo Röhl. Topological evolution of dynamical networks: global criticality from local dynamics. *Physical review letters*, 84(26 Pt 1):6114–7, June 2000.
- [74] Stefan Bornholdt and Torsten Röhl. Self-organized critical neural networks. *Physical Review E*, 67(6):066118, June 2003.
- [75] Matthias Rybarsch and Stefan Bornholdt. Avalanches in self-organized critical neural networks: a minimal model for the neural SOC universality class. *PLoS one*, 9(4):e93090, January 2014.
- [76] Alessandro Vespignani and Stefano Zapperi. How self-organized criticality works: A unified mean-field picture. *Physical Review E*, 57(6):6345–6362, June 1998.
- [77] Theodore E. Harris. *The Theory of Branching Processes*. Dover, New York, 1989.
- [78] Christel Kamp and Stefan Bornholdt. Critical percolation in self-organized media: A case study on random directed networks. page 4, October 2002.
- [79] Daniel B. Larremore, Woodrow L. Shew, Edward Ott, Francesco Sorrentino, and Juan G. Restrepo. Inhibition Causes Ceaseless Dynamics in Networks of Excitable Nodes. *Physical Review Letters*, 112(13):138103, April 2014.
- [80] P. Bak and Maya Paczuski. Complexity, contingency, and criticality. *Proceedings of the National Academy of Sciences*, 92(15):6689–6696, July 1995.

- [81] John M Beggs and Dietmar Plenz. Neuronal avalanches in neocortical circuits. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 23(35):11167–77, December 2003.
- [82] Elakkat D Gireesh and Dietmar Plenz. Neuronal avalanches organize as nested theta- and beta/gamma-oscillations during development of cortical layer 2/3. *Proceedings of the National Academy of Sciences of the United States of America*, 105(21):7576–81, May 2008.
- [83] Thomas Petermann, Tara C Thiagarajan, Mikhail a Lebedev, Miguel a L Nicolelis, Dante R Chialvo, and Dietmar Plenz. Spontaneous cortical activity in awake monkeys composed of neuronal avalanches. *Proceedings of the National Academy of Sciences of the United States of America*, 106(37):15921–6, September 2009.
- [84] Ole Paulsen. Natural patterns of activity and long-term synaptic plasticity. *Current Opinion in Neurobiology*, 10(2):172–180, April 2000.
- [85] Yang Zhan, Rosa C Paolicelli, Francesco Sforazzini, Laetitia Weinhard, Giulia Bolasco, Francesca Pagani, Alexei L Vyssotski, Angelo Bifone, Alessandro Gozzi, Davide Ragozzino, and Cornelius T Gross. Deficient neuron-microglia signaling results in impaired functional brain connectivity and social behavior. *Nature neuroscience*, 17(3):400–6, 2014.
- [86] A L Hodgkin and A F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–44, August 1952.
- [87] Randal a Koene, Betty Tijms, Peter van Hees, Frank Postma, Alexander de Ridder, Ger J a Ramakers, Jaap van Pelt, and Arjen van Ooyen. NET-MORPH: a framework for the stochastic generation of large scale neuronal networks with realistic neuron morphologies. *Neuroinformatics*, 7(3):195–210, September 2009.
- [88] Frederic Zubler and Rodney Douglas. A framework for modeling the growth and development of neurons and networks. *Frontiers in computational neuroscience*, 3(November):25, January 2009.
- [89] Marc Barthélemy. Spatial networks. *Physics Reports*, 499(1-3):1–101, February 2011.

- [90] Patric Hagmann, Leila Cammoun, Xavier Gigandet, Reto Meuli, Christopher J Honey, Van J Wedeen, and Olaf Sporns. Mapping the structural core of human cerebral cortex. *PLoS biology*, 6(7):e159, July 2008.
- [91] Mark H Johnson. Functional brain development in humans. *Nature reviews. Neuroscience*, 2(7):475–83, July 2001.
- [92] D. H. Hubel and T. N. Wiesel. Effects of Monocular Deprivation in Kittens. *Naunyn Schmiedebergs Arch Exp Pathol Pharmakol*, 497:492–497, 1964.
- [93] D. H. Hubel, T. N. Wiesel, and S. LeVay. Plasticity of Ocular Dominance Columns in Monkey Striate Cortex. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 278(961):377–409, April 1977.
- [94] U D McCann, David M Penetar, Yavin Shaham, Druid R Thorne, Helen C Sing, Mar L Thomas, J C Gillin, and Gregory Belenky. Effects of catecholamine depletion on alertness and mood in rested and sleep deprived normal volunteers. *Neuropsychopharmacology : official publication of the American College of Neuropsychopharmacology*, 8(4):345–56, June 1993.
- [95] Yudong Sun, Bogdan Danila, K. Josić, and Kevin E. Bassler. Improved community structure detection using a modified fine-tuning strategy. *EPL (Europhysics Letters)*, 86(2):28004, April 2009.
- [96] Christian M Schneider, L. de Arcangelis, and H. J. Herrmann. Scale-free networks by preferential depletion. *EPL (Europhysics Letters)*, 95(1):16005, July 2011.
- [97] Henry Markram. The blue brain project. *Nature reviews. Neuroscience*, 7(2):153–60, February 2006.
- [98] Chris Eliasmith, Terrence C Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Yichuan Tang, Charlie Tang, and Daniel Rasmussen. A large-scale model of the functioning brain. *Science (New York, N.Y.)*, 338(6111):1202–5, November 2012.
- [99] P. a. Merolla, J. V. Arthur, R. Alvarez-Icaza, a. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, a. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha. A million spiking-neuron integrated circuit

- with a scalable communication network and interface. *Science*, 345(6197):668–673, August 2014.
- [100] Theodore M Wong, Robert Preissl, Pallab Datta, Myron Flickner, Raghavendra Singh, Steven K Esser, Emmett McQuinn, Rathinakumar Appuswamy, William P Risk, Horst D Simon, Dharmendra S Modha, San Jose, and Lawrence Berkeley. Ten to Power 14. *IBM Research Report*, 10502:13–15, 2012.

Index

- Adaptive Network, 35
- Adaptive SIS model, 37
- Resource diffusion model, 38
- RTN spin SOC model, 41
- Stochastic Boolean SOC model,
43
- Stochastic spin SOC model, 43
- Branching parameter, 45
- Characteristic path length, 5
- Derrida plot, 34
- Hamming distance, 33
- Hierarchical clustering, 8
- Matrix
 - Adjacency, 3
 - Weight, 4
- Modularity, 8
 - Betweenness centrality, 7
 - Modularity Matrix, 9
- Network
- Barabási-Albert, 15
- Erdos-Rényi, 13
- Scale-free, 5
- Small-world, 6
- Spatial embedding, 54
- Watts-Strogatz, 17
- Network metrics
 - Clustering coefficient, 5
 - Degree, 4
 - Degree distribution, 5
 - Giant component, 14
 - Network sensitivity, 34
 - Pearson correlation, 44
 - Preferential attachment, 15
- Random Threshold Network, 21
 - Annealed approximation, 25
 - Parameter space, 24
- Self-Organized Criticality, 41
- Synaptic
 - Growth, 61
 - Pruning, 49