Warsaw University
Faculty of Mathematics, Informatics and Mechanics

Filip Piękniewski

Spontaneous Scale-free Structures in Spike Flow Graphs for Recurrent Neural Networks

PhD dissertation

Supervisor

dr hab. Tomasz Schreiber

Faculty of Mathematics & Computer Science
Nicolaus Copernicus University

October 2008

Author's declaration:
aware of legal responsibility I hereby declare that I have written this dissertation myself and all the contents of the dissertation have been obtained by legal means.

October 1, 2008 .....

date *Filip Piękniewski*

Supervisor's declaration:
the dissertation is ready to be reviewed

October 1, 2008 .....

date *dr hab. Tomasz Schreiber*

Abstract

In this thesis we present both theoretical and numerical results, revealing self-organization in graphs of functional connections for a number of recurrent neural network models. The consequence of studied self-organization is the spontaneous emergence of scale-free connectivity. The concept of the scale-free network (whose vertex degree distribution obeys a power law) discussed in detail in the thesis is an intensively studied part of random graph theory. These studies are particularly motivated by a number of recent discoveries which show that many networks found in the real world are indeed scale-free. To date, such a structure has been found in the Internet, scientific/artistic collaboration networks, social networks, linguistic networks, metabolic networks and many others. The results related to neural networks however, are not evident. Some findings show that the neural networks (on the level of synapses) of small organisms exhibit rather exponential decay of degree distribution, while other experiments conducted with functional magnetic resonance imaging (fMRI) on the human brain reveal that the networks connecting correlated functional centers of activity are scale-free, with a power law exponent consistent with our theoretical results. In further chapters of this thesis we introduce a mathematically tractable model of an asynchronous spiking neural network which generalizes the concept of a Boltzmann machine. We analyze energy minima (ground states) of the presented model and the graph of charge transfers between the units in the course of the dynamics. We argue that under certain natural assumptions about the Hamiltonian at low enough temperatures the large-scale behavior of the system admits an accurate description in terms of a winner-take-all type dynamics. This can be used to show that the resulting graph of charge transfers, referred to as the *spike flow graph* in the sequel, has scale-free properties with power law exponent $\gamma = 2$. Additionally, we demonstrate the results of numeric studies which show that the introduced model accurately describes the interaction between isolated neuronal groups based on the phenomenological model of Eugene M. Izhikevich. We point out possible further research directions based on spontaneously developing neuronal groups and large scale brain simulations.

Keywords: scale-free graph; stochastic neural network; winner-take-all dynamics

AMS Mathematics Subject Classification 2000: 05C80, 82C32, 92B20, 68T99

Streszczenie

W pracy przedstawiamy wyniki, zarówno teoretyczne jak i numeryczne, opisujące samoorganizację funkcjonalnych grafów połączeń dla kilku modeli rekurencyjnych sieci neuronowych. Rezultatem badanej samoorganizacji jest przede wszystkim spontaniczne powstawanie w takich grafach struktur bezskalowych, czyli takich w których rozkład stopni wierzchołków spełnia prawo potęgowe. Dyskutowane szerzej w pracy sieci bezskalowe stanowią intensywnie eksplorowaną obecnie gałąź badań grafów losowych, szczególnie w kontekście doniesień, iż bardzo wiele grafów występujących w otaczającym nas świecie ma cechy bezskalowości. Do tej pory cechy takie stwierdzono w strukturze Internetu, sieciach współpracy naukowej/artystycznej, społecznych, lingwistycznych, metabolicznych i wielu innych. Tymczasem wyniki dotyczące struktury połączeń sieci neuronowych nie są jednoznaczne. Pewne doniesienia wskazują na wykładniczy zanik ogona rozkładu stopni wierzchołków w układach neuronowych niewielkich organizmów (na poziomie synaps), inne natomiast badania przeprowadzone na mózgu ludzkim za pomocą funkcjonalnego rezonansu magnetycznego (fMRI) wskazują, iż grafy indukowane przez centra aktywności są istotnie bezskalowe, z wykładnikiem zgodnym z naszymi przewidywaniami teoretycznymi. W dalszych rozdziałach tej pracy prezentujemy model matematyczny asynchronicznej impulsującej sieci neuronowej uogólniającej koncepcję maszyny Boltzmanna. Analizujemy minima energetyczne (stany bazowe) prezentowanego modelu, oraz graf indukowany przez przepływy potencjału pomiędzy jednostkami. Dowodzimy, że dla pewnych naturalnych założeń dotyczących funkcji energetycznej i w odpowiednio niskiej temperaturze wielkoskalowy opis układu sprowadza się do dynamiki typu "zwycięzca bierze wszystko" z czego można dalej wnioskować, iż indukowany graf przepływu impulsów jest bezskalowy z wykładnikiem $\gamma = 2$. Ponadto prezentujemy wyniki badań symulacyjnych, które wskazują, iż zaproponowany model dobrze odzwierciedla interakcje między izolowanymi grupami neuronów opartych o fenomenologiczny model Eugene M. Izhikevicha. Wskazujemy dalsze kierunki badań oparte o spontanicznie wykształcające się grupy neuronów oraz wielkoskalowe symulacje układu nerwowego.

Słowa kluczowe: graf bezskalowy; stochastyczna sieć neuronowa; dynamika "zwycięzca bierze wszystko"

Klasyfikacja tematyczna AMS 2000: 05C80, 82C32, 92B20, 68T99

Contents

List of Figures **ix**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Structure of this thesis . . . 2

2 Random and power law graphs **5**
2.1 Introduction . . . 5
2.2 Erdős-Rényi random graphs . . . 6
2.2.1 Onset of the giant component - phase transition . . . 6
2.2.2 Erdős-Rényi random graphs and the real world . . . 9
2.3 Fixed degree distribution models . . . 10
2.4 Models based on real world phenomena . . . 11
2.4.1 Small World Graphs . . . 11
2.4.2 Preferential attachment and power law graphs . . . 13
2.4.3 Duplication models for biological networks . . . 17
2.4.4 Connectivity of the brain . . . 18
2.5 Appendix: Mathematical tools . . . 19
2.5.1 Spectral methods . . . 20
2.5.2 Tree matrix theorem . . . 21
2.5.3 Clustering coefficient . . . 22

3 Contemporary neural models **23**
3.1 Introduction . . . 23
3.2 Hodgkin-Huxley model . . . 24
3.2.1 Structural dynamics of neural cell . . . 27
3.3 Firing rate models . . . 28
3.3.1 McCulloch-Pitts neuron and the perceptron . . . 29
3.3.2 Multi Layer Perceptrons . . . 29
3.3.3 Hopfield network . . . 30
3.4 Reduced ionic models . . . 31

3.5 Phenomenological spiking models . . . 34
3.5.1 Integrate and fire models . . . 34
3.5.2 Fithugh-Nagumo model . . . 36
3.5.3 Simple model by Eugene M. Izhikevich . . . 37

4 Spike flow graphs **39**
4.1 Basic model . . . 40
4.2 Winner-take-all dynamics and ground states . . . 42
4.3 Power law for spike-flow in-degrees . . . 46
4.4 Numeric results . . . 48

5 Scale-freeness of dynamical spiking networks **53**
5.1 Introduction . . . 53
5.2 Model details . . . 56
5.3 Results . . . 58

6 Further research **63**
6.1 Introduction . . . 63
6.2 The model . . . 63
6.2.1 Anatomy . . . 64
6.2.2 Short term synaptic dynamics . . . 65
6.2.3 Long term synaptic dynamics . . . 66
6.2.4 Neuronal dynamics . . . 67
6.3 Implementation details . . . 68
6.3.1 Short term dynamics . . . 68
6.3.2 STDP . . . 69
6.3.3 Neuronal dynamics . . . 72
6.3.4 Code structure and parallelization . . . 73
6.4 Results . . . 76

7 Conclusions **83**

Bibliography **87**

Index **99**

List of Figures

2.1 A sample Erdős-Rényi graph with 10000 vertices just above the critical probability. There is a large component and a couple of tiny isolated components. . . . 7

2.2 Phase transition on a sample Erdős-Rényi graph with 1000 vertices. Edge probability values $p = 0.9/1000, 1/1000, 1.1/1000$ from left to right respectively. As predicted by theory the graph below the critical probability is composed of a number of small trees (leftmost figure), at the critical regime the number of small trees decreases and cycles emerge as well as the giant component (middle). A little above the critical regime the graph becomes connected (right). . . . 8

2.3 From a fully localized graph (left) to a completely random graph (right). Small world networks interpolate between the two extremes. Good localization combined with a small diameter makes these graphs a good model of real world networks. . . . 12

2.4 Power law degree distribution found in many empirical graphs (blue) and Poissonian distribution expected in large Erdős-Rényi random graphs (pink). One of the features of power laws is that they appear as straight lines in log-log plots and the exponent can be estimated as the negative slope. This property makes power laws easy to distinguish in various empirical data. . . . 14

2.5 Pareto distribution $\frac{kx_m^k}{x^{k+1}}$ with mode $x_m = 1$ and scaling parameters $k = 1, 2, 3$. . . . 16

2.6 (Left) an illustration of the Zipfs law which says that the most frequent word will occur approximately twice as often as the second most frequent word, which will occur approximately twice as often as the third most frequent word etc. This sample is based on a fragment of Hamlet by Shakespeare. (Right) similar dependency of sizes of US cities. . . . 17

2.7 Graphs: random Erdős-Rényi, small world Watts-Strogatz, scale-free Barabási-Albert. Note in the last example an existence of a hub node with exceptional connectivity and a fair number of poorly connected nodes. . . . 18

3.1 Complete neuron cell diagram. Image courtesy of Mariana Ruiz Villarreal (Available at: http://en.wikipedia.org/wiki/Image:Complete_neuron_cell_diagram_en.svg) . . . 24

3.2 Example evolution of membrane potential (left), sodium ion (middle) and potassium ion (right) conductance in response to 6.2 millivolt pulse. . . . 27

3.3 Schematic diagram of the artificial neuron (left) and a geometrical interpretation of its activation in 2d space (the plotted function is smooth sigmoidal for better readability). . . . 30

3.4 Phase portrait of the instantaneous "persistent sodium plus potassium" model. The nullclines are depicted with bold colored lines, while a number of trajectories are depicted with thin black lines. There are equilibria (stable and unstable) on the intersections of nullclines responsible for rest state, and a mechanism of excitability (a "shadow" of the limit cycle). . . . 33

3.5 Schema of the neuromime pulse generator. . . . 35

3.6 Phase portrait of the FitzHugh-Nagumo model. Notice that the phase portrait resembles the one of $Na + K$ model (figure 3.4). . . . 36

3.7 The phase plane for the simple spiking neuron model. Nullclines and a number of sample trajectories are plotted. . . . 37

3.8 Diagram of the dynamical model (Izhikevich, 2003) and possible spiking regimes that the model is capable of reproducing. Note the subthreshold oscillations (resonator), which are missing in simpler integrate and fire models. Image courtesy of Eugene M. Izhikevich (http://www.izhikevich.com). . . . 38

4.1 Schematic presentation of the spike flow model . . . 43

LIST OF FIGURES

4.2 Typical evolution of the charge stored in seven units of the highest support. The above figure is a result of a simulation run of 3000 units (left) and 4000 units (right). Note that while in the beginning of the simulation all seven elite members compete for charge, by the end the single best unit gets everything. 48

4.3 Percentage of charge jumps leading to a unit of higher support (sampled every 100 jumps) in a simulation run of 3000 units. The plot on the left is scaled linearly, the one on the right is semi-log. Note that after initial unstable phase (about 10^4 steps), the plot increases steadily until about $2 * 10^6$ steps where again some fluctuations occur. These fluctuations are caused by increased significance of stochastic term (tiny energy modifications leaving room for thermal fluctuations). It is worth noting that by that time jumps are already infrequent while the state is near the energy optimum. By step $2 * 10^7$ the system freezes completely in the ground state. . 49

4.4 Cumulative distribution function (CDF) of the out-degree in the spike flow graph (in-degree yields a similar plot) of 3000 units (left) and 4000 units (right). Presented CDF slopes correspond to the power law exponent $\gamma = 2 + / - 0.03$. The slopes were approximated by the least squares method. 50

4.5 The charge stored in 2% units of highest support (left) and number of units storing 98% of total charge. These figures give strong support to the idea of dividing the units into elite and bulk, and treating these groups separately. 51

5.1 A schematic presentation of the model investigated in this chapter. The model consists of a number (about 1000-3000) of neuronal groups, connected randomly (weights chosen from Gaussian distribution $\mathcal{N}(0, 1)$) by the group leaders - neurons chosen to interconnect every group with others. The group's synchronization depends on the input received from the group leader and, on the other hand, the activity of the leader reflects the activity of the group. 55

5.2 Example of spike activity plot of 3000 group leaders during 6000ms timeframe (only the activity of group leaders is plotted). Note the global synchrony episodes as well as bursting of single units. 56

5.3 Magnified segment of figure 5.2 showing neurons 1000 to 1500 within 2000ms-3000ms timeframe. Please note the similarities of these plots (actually it is a self-similarity). The number and length of straight horizontal lines (each symbolizing a bursting activity) in both plots is approximately the same. 57

5.4 Example degree distributions of synchronization graphs produced by a network of spiking neurons for different coupling regimes. Certainly these graphs do not exhibit a scale-free property. 58

5.5 The degree distributions of a network received from the model (left), and one obtained by thresholding the original network at an average weight (right) in order to produce an unweighted graph. In either case the presence of a power law with exponent of about 2 is quite clear. Note the significant difference between these plots and the ones showed in figure 5.4. 59

5.6 The dependency between spiking activity and vertex degree before (left) and after thresholding the graph (right). In either case we observe a clearly monotone dependency which supports our preferential attachment hypothesis. 60

5.7 Clustering coefficient as a function of degree in the investigated network. This plot is interesting, since such behavior of clustering coefficient is rather rare - the nodes of small degrees are well clustered whereas those with high degree are not. Compare with box 2 in (Barabási & Oltvai, 2004). The black-circle plot depicts this dependency obtained from Erdős-Rényi random graph with similar number of edges. 61

5.8 Spike activity convolved with $\exp\left(-\left(\frac{x}{10}\right)^2\right)$ kernel. This procedure blurs the spike train significantly, but lets one receive non zero product of two such trains even if corresponding spikes are shifted. The product is later integrated to obtain *synchronization strength*, a measure we introduced to describe similarity between spike trains. 61

6.1 A sketch of the connectivity in the neuronal sphere model investigated in this chapter. Only 200 out of 100 000 neurons are plotted. 64

6.2 A plot of changes of membrane voltage in a randomly selected excitatory neuron (left) and inhibitory neuron (right). Each figure shows the activity over a period of 2s. Notice a number of sub-threshold jumps. 68

6.3 A sample spike train obtained from the model, that used the "resetting" implementation of STDP. The firing regime is a bit different than the one in the original model, the firing to high, there is a lot of bursting. 70

6.4 A sample spike train obtained from the model using the "incrementing" implementation of STDP. The firing regime here resembles the regime from the original model. 70

6.5 Structural presentation of the code used in simulation. The same structure can be extended to an arbitrary number of cores. 74

6.6 The firing rate versus time at the beginning of the simulation for excitatory neurons (black) and inhibitory neurons (blue). The plot on right shows the same data in semi logarithmic scale. Clearly the system converges into an equilibrium state after about 1000s of model time. 76
6.7 Variation of activity (spiking rate) in the sphere model. The hue represents excitatory activity in logarithmic scaling. The figure demonstrates that the activity is very homogenous. 77
6.8 Distribution of excitatory weights (left) and corresponding distribution of excitatory postsynaptic potentials (EPSP). EPSPs are computed by taking into account the state of synaptic short term depression/facilitation and assuming that the postsynaptic neuron is at rest (compare with original model Izhikevich *et al.* (2004), the results are very similar.) 78
6.9 Schema of the group finding algorithm. A is the anchor neuron, B,C its immediate descendants. We assume that B and C fire after excitation coming from the anchor. Next we seek for common postsynaptic targets having converging delays (D). E and F are discarded from the group. 79
6.10 A fraction of the groups fund in one of the simulations (total number of groups in this case was about 13000). The left figure shows 2% of total number of groups, while the right figure shows 10%. 80
6.11 Sample histograms of group activity. Left figure shows the output (number of spikes departing from the group versus number of groups), the right figure shows the same statistics divided by the activity of an anchor neuron. In the lower row are the logarithmic plots. 81

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

The concept of a power law network has gathered a lot of attention in recent years providing a unified description of a wide variety of complex network topologies all of which display evidence of strong structuring principles co-existent with a considerable degree of randomness (see Albert & Barabási (2002) for a comprehensive survey). A distinctive feature of a power law network is that its random subgraphs with great probability have degree distributions that follow a power law with the same exponent as in the original graph. Hence the shape of degree distribution is invariant over scale and thus such networks are often referred to as *scale-free*. The presence of such power laws has been observed for a broad class of networks, prominent examples including the World Wide Web (Albert *et al.* , 1999), science collaboration networks (Barabási *et al.* , 2002), citation networks (Redner, 1998), ecological networks (Montoya & V., 2002), linguistic networks (i Cancho & Solé, 2001) as well as cellular metabolic networks (Jeong *et al.* , 2000) and many others, see (Albert & Barabási, 2002) and chapter 2 of this thesis. Many instances of structuring principles resulting in scale-free networks have been proposed, with a prominent collection of examples stemming from Barabási-Albert model (Albert & Barabási, 2002; Barabási & Albert, 1999) and its variants modeling a variety of scale-free networks with different power law exponents by exploiting the fundamental properties of growth and preferential attachment.

Recently, considerable interest has been attracted by neural networks built on scale-free graphs and it turned out that a hierarchical scale-free network architecture is in many cases beneficial for efficiency of neuronal information processing. A scale-free graph is relatively sparse, and so the memory needed to store a neural network built on such a graph is significantly reduced, as is the computational effort required to perform certain tasks(see Perotti *et al.* (2006); Stauffer *et al.* (2003)). In this context it is natural to ask whether these advantages are reflected

in some mechanisms which are inherent to the usual recurrent neural network dynamics and which result in emergence of power laws. In general this does not seem to be necessarily the case for neural networks with simple processing units. For instance the neural network of C.elegans worm exhibits rather exponential decay (Amaral *et al.* , 2000; Koch & Laurent, 1999). Numerical simulation based on the simple model by Eugene M. Izhikevich (Izhikevich, 2003) also did not reveal any scale-free structure, as long as the processing units were single neurons (Piękniewski, 2007). However, things can be very different when more complicated individual unit architecture is assumed, in which case a single *formal neuron* can be interpreted as modeling a computational unit exhibiting some non-trivial internal structure and memory, for instance a group of biological or artificial neurons, see (Piękniewski, 2007) for related numerical study. In this thesis we study circumstances in which scale-free connectivity may emerge naturally as a consequence of neural like dynamics. We introduce a simple and mathematically tractable model (Piękniewski & Schreiber, 2008) which in many ways resembles the classical Boltzmann machine (Aarts & Korst, 1989), yet admitting a richer state space and assuming rather different dynamics for individual neurons which are simple spiking units here. We analyze the ground states of the model and argue it can be represented via a kind of a winner-take-all dynamics whose particular features enable us to establish explicit results on the scale-free properties of the spike-flow graph. These results seem to be in agreement (including power law exponent) with empirical results based on fMRI presented in (Eguíluz *et al.* , 2005). We also present related numerical studies based on the explicit model as well as dynamical spiking neurons and investigate what conditions are necessary in such networks for the power law connectivity to emerge.

1.2 Structure of this thesis

This thesis is organized as follows: in chapter 2 we introduce various concepts of random graphs, in particular Erdős-Rényi random graphs (section 2.2) and generalizations of this model (section 2.3). In section 2.4 we discuss several examples of empirical graphs studied in recent years as well as random models aimed at examining their structure. In subsection 2.4.4 we discuss the present knowledge about the low level connectivity of the nervous system. In chapter 3 we give a brief introduction into contemporary models of neurons in a somewhat different fashion than usually found in the literature. In particular we focus on dynamical spiking models that are more related to neuroscience than to typical applications (sections 3.2, 3.4, 3.5), while only briefly covering classical models and algorithms (section 3.3). In chapter 4 based on (Piękniewski & Schreiber, 2008) we introduce a novel model resembling a spiking neural network. This chapter contains the

main results of the thesis, stating that under appropriate assumptions about the Hamiltonian, the presented model reveals a power law graph structure as an outcome of its dynamical behavior. This model is significantly different compared to existing models of random power law graphs discussed in chapter 2. Apart from formal derivation, chapter 4 also contains the results of direct numerical simulation of the presented model (section 4.4). In the next chapter 5 based on (Piękniewski, 2007) we present another numerical study founded on dynamical spiking neurons (discussed in section 3.5) revealing the general mechanisms which may lead to power law distributions in neural systems. In chapter 6 we present a more advanced numerical study based on (Izhikevich *et al.* , 2004). We reconstructed the model in search of power law distributions, but the numeric results obtained are rather obscure and therefore we regard the chapter as a direction for further research. In particular in section 6.4 we discuss the possible perspectives of further studies based on most recent large scale neuro-simulations. Finally, in chapter 7 we conclude the thesis summarizing the achieved results.

Chapter 2

Random and power law graphs

2.1 Introduction

Graphs can be formally defined within one line of text yet they posses huge abilities in providing models for certain far more complex mathematical entities. Any binary relation on a finite set is a graph, which suggests a rather controversial statement that almost everything in mathematics can be seen as a graph. Mathematics itself can be represented as a graph whose vertices are statements and edges are steps of proof leading from one statement to another. Category theory offers yet a higher perspective, stating that in fact whole theories in mathematics can be interpreted as vertices with homomorphisms playing the role of edges. But graphs are not exclusively associated with mathematics. The real world is full of graphs whose structure can tell us a lot about physical reality. These graphs can be explicit like a science collaboration network, the Internet, the neuronal structure of a human brain, or implicit like graphs of chemical reactions forming the metabolism of a cell, or a semantic network of words in a given language. These examples encourage people to study graph theory, in particular to study properties of a *typical graph*. Yet the notion of a *typical graph* is problematic. The confusion is hidden in the meaning of the word "typical". Typical in what sense? There are infinitely many graphs, and even if we restrict ourselves to graphs of a fixed size, their number could easily become astronomical[1]. What probabilistic measure should we attribute to a set of graphs of a fixed size? Even if we choose the probabilistic measure, will we be able to study the expected properties of a graph with respect to that measure? Will we be able to efficiently select a random graph with respect to that measure? And most of all: is the introduced random model a good replica of the graphs found in the real world? These seemingly

[1]Depending on whether for example $2^{2^{1000}}$ is still astronomical or rather something more than that.

simple questions may have surprisingly difficult answers.

2.2 Erdős-Rényi random graphs

The study of random graphs began with the pioneering work of Paul Erdős and Alfréd Rényi (Erdős & Rényi, 1959, 1960). They have introduced two models of random graphs:

- Set the number of vertices n and a number of edges m. For consecutive edges choose their endpoints randomly among all of the vertices.
- Set the number of vertices n and a probability p. For every pair of vertices insert an edge with probability p.

At first sight the models look equivalent, especially when $p = m/\binom{n}{2}$, but in fact the first model has a small degree of dependence due to a fixed amount of edges. The second one is more convenient for analysis, because edges appear independently of each other. Obviously the probability measure resulting from such a setup is concentrated among graphs with expected number of edges $p\binom{n}{2}$, yet any other graph is achievable with non zero probability (though decreasing exponentially with the divergence from the expected number of edges). By varying parameter p different regions of the space can be studied. Surprisingly for $n \to \infty$ there is a sharp threshold in p above which the graph has the so called giant component with probability close to 1, and below which it is composed of small tree like components. A variety of interesting things can happen in the critical regime, the study of which is beyond the scope of this thesis (see Bollobas (2001, 1998))

2.2.1 Onset of the giant component - phase transition

The presence of a giant component is an important property studied in the random graph theory. As mentioned above for Erdős-Rényi random model, there exists a sharp threshold (function depending on n), below which a typical random graph is a set of disjoint trees (or the so called unicyclic components) of size at most $\log n$. Only after the probability p reaches the threshold, a large connected component emerges (whose size is of the same order as the size of the entire graph). Here we will present a sketch of argumentation motivating the above discussion (details can be found in chapter 2 of Durrett (2007)). The key idea is simple: assume we start from some randomly chosen vertex v and look at its neighbors. They form the first generation of descendants of v. We can then examine the second generation, that is to say the neighbors of the neighbors, and so on. Such a process can be approximated by a procedure defined as

follows: consider a sequence ζ_i^t i.i.d nonnegative integer valued random variables and define Z_t, $t > 0$ as $Z_0 = 1$ and:

$$Z_{t+1} = \begin{cases} \zeta_1^{t+1} + ... + \zeta_{Z_t}^{t+1}; & Z_t > 0 \\ 0 & Z_t = 0 \end{cases} \tag{2.1}$$

Z_t is called *Galton-Watson* process or a branching process. The probability dis-

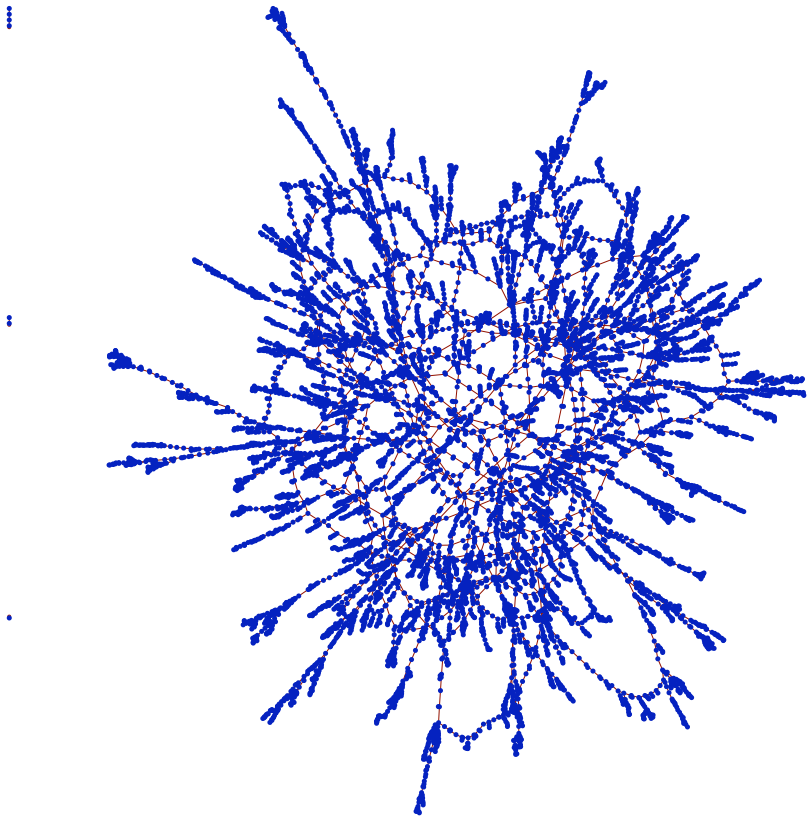

Figure 2.1: A sample Erdős-Rényi graph with 10000 vertices just above the critical probability. There is a large component and a couple of tiny isolated components.

tribution p_k of ζ_i^t is called the *offspring distribution*. The idea is simple: Z_t is the number of individuals in t-th generation. Each individual gives birth to some number of offsprings according to p_k. Under appropriate delicate assumptions the branching process accurately approximates the local structure of connected components in E-R graphs, at least as long as they are small enough, each occupying a negligible fraction of vertices. In this context the offspring probability law is simply the binomial distribution induced by the constructing principles of

the graph. Note that when $p = \lambda/n$ for some λ, each vertex has a binomial number of neighbors $B(n-1, \lambda/n)$ whose expected value tends to λ. When $\lambda < 1$ the process dies out (expected number of descendants of any individual is less than 1), when $\lambda > 1$ the process is sustained with high probability (the case $\lambda = 1$ is delicate, but it can be shown that in this case the branching process dies out as well unless the offspring probability distribution is trivial). This approximation begins to break down when the size of the component approaches the size of the graph, which is already far enough to establish the existence of giant component (though it requires delicate calculation). It is fairly easy to show that a branching process whose expected number of children for each individual is less than 1 will die out exponentially fast. Note that $E(Z_t) = \lambda^t$ (each new generation multiplies the existing by λ). $Z_t \geq 1$ on $\{Z_t > 0\}$, so $P(Z_t > 0) \leq E(Z_t; Z_t > 0) = E(Z_t) = \lambda^t \to 0$ when $t \to \infty$ (furthermore the convergence is exponential). In the case when $\lambda = 1$ and $P(\zeta_i^t = 1) < 1$ the vanishing of the process can also be proven using the fact that Z_t/λ_t is a martingale. The proof that the process persists when $\lambda > 1$ is somewhat longer and uses properties of generating functions of the offspring distribution. In fact the

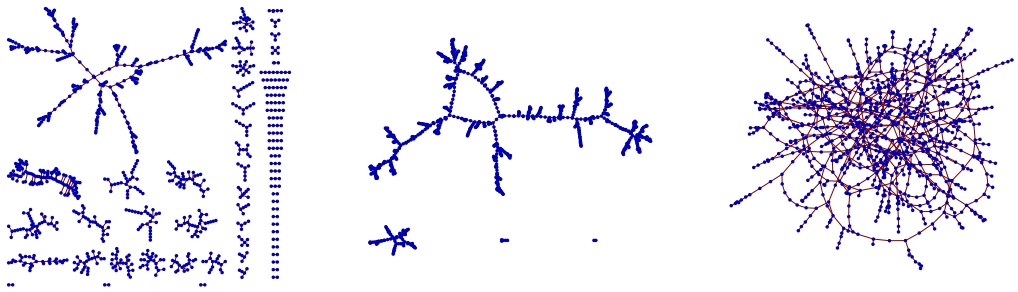

Figure 2.2: Phase transition on a sample Erdős-Rényi graph with 1000 vertices. Edge probability values $p = 0.9/1000, 1/1000, 1.1/1000$ from left to right respectively. As predicted by theory the graph below the critical probability is composed of a number of small trees (leftmost figure), at the critical regime the number of small trees decreases and cycles emerge as well as the giant component (middle). A little above the critical regime the graph becomes connected (right).

phase transitions in Erdős-Rényi model are now (after many years of research) characterized more precisely as follows (see section 5.1 in Chung & Lu (2006)):

- $p = o(1/n)$ the graph is a disjoint union of trees.
- $p \approx \lambda/n$ for $\lambda \in (0, 1)$ the graph contains cycles of any length with positive probability (in the limit). All connected components are either trees or unicyclic components of size $O(\log n)$ (see figure 2.2).

- $p = (1+\varepsilon)/n$ there suddenly appears the giant component. For $\varepsilon = 0$ (the critical regime) the largest component is of size $n^{2/3}$. With probability 0.9325 (as $n \to \infty$) the graph consists of trees, cycles and double cycles. With probability 0.9957 the graph consists of trees, cycles and at most one more complex component. With probability between 0.987 and 0.9998 it is planar. For $\varepsilon > 0$ most of the small components get connected and for a giant component of size αn, where $\varepsilon = -\frac{1}{\alpha}\log(1-\alpha) - 1$.

- $p = \lambda/n$ for $\lambda > 1$ except for the giant component, all the other components are of logarithmic size. Most of them are trees (see figures 2.2 and 2.1) though unicyclic components may be present as well.

- $p = \lambda \log(n)/n$ for $\lambda > 1$ as $n \to \infty$ the graph becomes almost surely connected.

- $p = \omega(n)\log(n)/n$ with $\omega(n) \xrightarrow[n\to\infty]{} \infty$ the graph is almost surely connected and the degrees of almost all nodes are asymptotically equal.

2.2.2 Erdős-Rényi random graphs and the real world

Apart from being a very successful mathematical model Erdős-Rényi random graphs have certain drawbacks which became apparent by the end of XX century. At this time, computational power reached level appropriate for the numerical study of large graphs that appear in the real world. Other sciences like biology and social sciences collected numerous databases, whose structural properties became important for further research. Within a short period of time a number of papers appeared, revealing, step by step, striking new features of empirical graphs. These can be summarized as follows:

- empirical graphs are sparse. Most of the graphs found in engineering, biology, social sciences have a small number of edges, scaling linearly with the number of vertices (with small linear coefficient). Corresponding Erdős-Rényi graphs (with approximately the same number of edges) are almost surely composed of trees and unicyclic components, whereas empirical graphs form complex giant components(see chapter 2 in Chung & Lu (2006), chapter II in Albert & Barabási (2002) and reprinted papers in chapter 3 in Newman *et al.* (2006)).

- empirical graphs have a nontrivial local structure. In most of the graphs found in nature, the fact that two vertices have common neighbors greatly increases the probability that these two are neighbors themselves. This property is captured by the notion of clustering coefficient (see appendix at

the end of this chapter) which, for every vertex, calculates the actual number of triangles divided by the number of all possible triangles among the vertex neighbors. Sparse Erdős-Rényi almost completely lack the local structure. This phenomenon combined with the short average path characteristic (that is also present in E-R graphs once they get connected) is sometimes referred to as the *small world property* .

- empirical graphs often have heavy tail distributions of their vertex degrees. These tails usually resemble power laws, but even if the distributions are essentially exponential in nature, they are far less concentrated than in the corresponding Erdős-Rényi graph. In a large E-R graph all nodes have fairly equal degree (the degree distribution approaches Poisson distribution with $n \to \infty$), while in empirical graphs there are huge discrepancies giving rise to formation of the so called hubs (vertices having exceptional connectivity).

Among others these three are the main reasons for seeking different model of a random graph, that would inherit some of the analysis techniques of E-R model, and yet would be more adequate for real world phenomena.

2.3 Fixed degree distribution models

One of the ideas is to generalize the concept of Erdős-Rényi graph by introducing expected degree sequence w (see chapter 5 in Chung & Lu (2006)). In this setup each vertex receives its expected degree w_i. An edge between vertices i and j is added with probability sw_iw_j where s is the global scaling parameter $s = 1/\sum_{i=1}^{n} w_i$. In this setup the expected number of edges adjacent to vertex i is $\sum_{j=1}^{n} sw_iw_j = 1/\left(\sum_{j=1}^{n} w_j\right) \cdot \sum_{j=1}^{n} w_iw_j = w_i$. A lot of interesting properties of such graphs have been studied in numerous papers (summarized in chapters 5-7 in Chung & Lu (2006)), and in particular, properties of a typical graph whose expected degree sequence follows a power law. Among other results the most interesting ones include:

- A typical graph with desired expected degree sequence almost surely has a giant component if the average expected degree is greater than one (theorem 6.14 in Chung & Lu (2006)) .
- The volume (a volume of a subset of vertices is the sum of their degrees) of the giant component is $\lambda_0 + O\left(\sqrt{n}\frac{\log^{3.5} n}{Vol(G)}\right)$ where λ_0 is the unique positive root of the following equation

$$\sum_{i=1}^{n} w_i e^{-w_i\lambda} = (1-\lambda)\sum_{i=1}^{n} w_i \tag{2.2}$$

which is a significant generalization of results of Erdős & Rényi (1959) (again theorem 6.14 in Chung & Lu (2006)).

- If the expected degree sequence follows a power law with exponent β, the exponent (and the power law) is with great probability preserved in random subgraphs of such a graph (section 5.8 in Chung & Lu (2006)). The random subgraph can be chosen by either fixing random vertices or random edges (the preservation of power law works in both cases). This gives a hint as to why power law graphs are sometimes referred to as *scale-free* graphs.

- Random graphs, even with the expected power law degree sequence, lack the local structure. For real world graphs it seems that such a structure is imposed by the notion of distance in physical space (see 2.4.1), while randomly constructed graphs, as discussed here, are not embedded into any space (though there are such models as well (Toroczkai & Guclu, 2007) often related to percolation theory and epidemic). One may think of a real world network as a sum of a local network and a global network (chapter 12 in Chung & Lu (2006)). Since the global and local mechanism of network construction might be different, they could result in discrepancies between the degree distributions of empirical graphs and exact power law estimations.

Many of the estimates presented in Chung & Lu (2006) are compared to the data from real world graphs (exhibiting surprisingly good agreement).

2.4 Models based on real world phenomena

2.4.1 Small World Graphs

The notion of Small World networks (short average path length combined with high locality, see chapter 12 in Chung & Lu (2006)) was a part of public knowledge for years, sometimes referred to as the *six degrees of separation* phenomenon. The concept was popularized by an experiment conducted by Stanley Milgram who examined the average path in sociological network of citizens of Omaha and Wichita (Travers & Milgram, 1969). The procedure was as follows:

- Random individuals in U.S. cities of Omaha, Nebraska and Wichita, Kansas were chosen as the starting points and Boston, Massachusetts to be the end point of a chain of correspondence. These cities were selected because they represented a great distance in the United States, both socially and geographically.

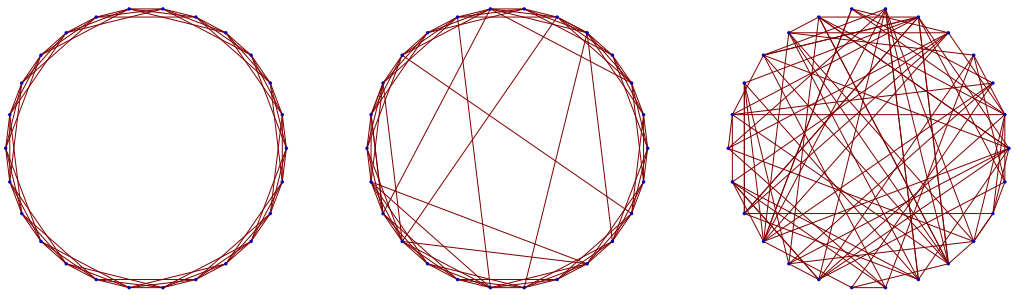

Figure 2.3: From a fully localized graph (left) to a completely random graph (right). Small world networks interpolate between the two extremes. Good localization combined with a small diameter makes these graphs a good model of real world networks.

- Information packets were initially sent to random starting points. The detailed study purpose was included, and an information about the target person in Boston (without the address).
- Upon receiving the invitation to participate, the recipient was asked whether he or she personally knew the contact person described in the letter. If so, the person was to forward the letter directly to that person. For the purposes of this study, knowing someone "personally" is defined as knowing them on a first-name basis.
- In the more likely case that the person did not personally know the target, the person was to think of a friend or relative they know personally that is more likely to know the target. They were then directed to sign their name on the roster and forward the packet to that person. A postcard was also mailed to the researchers at Harvard so that they could track the chain's progression toward the target.
- When and if the package eventually reached the contact person in Boston, the researchers could examine the roster to count the number of times it had been forwarded from person to person. Additionally, for packages that never reached the destination, the incoming postcards helped identify the break point in the chain.

Shortly after the experiment started, letters began to arrive at their destinations and researchers could analyze the chain of contacts between randomly chosen people. Among those that actually reached their targets the average path was about 6, however the experiment was criticized, since many of the letters never

reached their journey's end. Nevertheless, the results were quite surprising and attained public interest. Similar concepts are related to the famous Erdős number for mathematicians. Erdős had his number 0, any of his collaborators (coauthors of papers) have Erdős number 1, collaborators of collaborators 2 and so on[2]. Later analogous experiments gave rise to the Kevin Bacon[3] game which is based on finding a shortest movie collaboration path between a given actor and Kevin Bacon.

These phenomena, though funny and grabbing public interest, were beyond the reach of precise mathematical analysis for many years (Erdős-Rényi graphs do have short average paths, but completely lack the local structure) until 1998 when Duncan J. Watts and Steven H. Strogatz published their paper (Watts & Strogatz, 1998) in which they captured Small World phenomena in a simple and elegant mathematical model. The idea is remarkably simple: start with a locally connected graph and rewire a small number of edges randomly. These rewired edges provide global shortcuts dramatically decreasing the average path length, while the rest of the edges are responsible for highly localized structure (revealed by the high clustering coefficient). The key issue here is that by rewiring edges, average distance drops dramatically, while clustering coefficient decreases slowly, which leaves a huge space for small world networks of the interest. This elegant model, though very useful, ignored another silent feature of real world networks, which became apparent at that time as well: power law degree distributions.

2.4.2 Preferential attachment and power law graphs

Power law degree distributions are among the most striking features of real world networks. It is fairly understandable that many real world graphs coming from totally different disciplines would share some basic common features like size, sparseness, locality, but why on earth would they have similar degree distributions (much different than those of Erdős-Rényi graphs)? Nevertheless power law distributions are present in a huge variety of graphs including the World Wide Web (Albert *et al.* , 1999), science collaboration networks (Barabási *et al.* , 2002), Hollywood graph (Barabási & Albert, 1999), citation networks (Redner, 1998), ecological networks (Montoya & V., 2002), linguistic networks (i Cancho & Solé,

[2]The author estimated his Erdős number to be 4 as he coauthored with Tomasz Schreiber (Pięknieswski & Schreiber, 2008) who coauthored with Joseph Elliott Yukich who coauthored with Svante Janson who coauthored with Paul Erdős...

[3]Kevin Bacon is neither the most famous, neither the richest actor in Hollywood, but he played so many roles that he seemed to be a good candidate for a hub in an actor collaboration network. In fact thoughtful analysis of Hollywood actor collaboration graph revealed that there are more than a thousand better candidates for being the *centers of Hollywood*, Rod Steiger was the best one while writing this text. An up to date list can be found on:
`http://oracleofbacon.org/`

2001), telephone call network (Abello *et al.* , 1998; Aiello *et al.* , 2000) as well as cellular metabolic networks (Bhalla & Iyengar, 1999; Jeong *et al.* , 2000), and many others, see (Albert & Barabási, 2002) for a survey. Right from the beginning it seemed evident that some basic organizing principles present in all those network are responsible for the power law. In 1999 Barabási & Albert (1999)

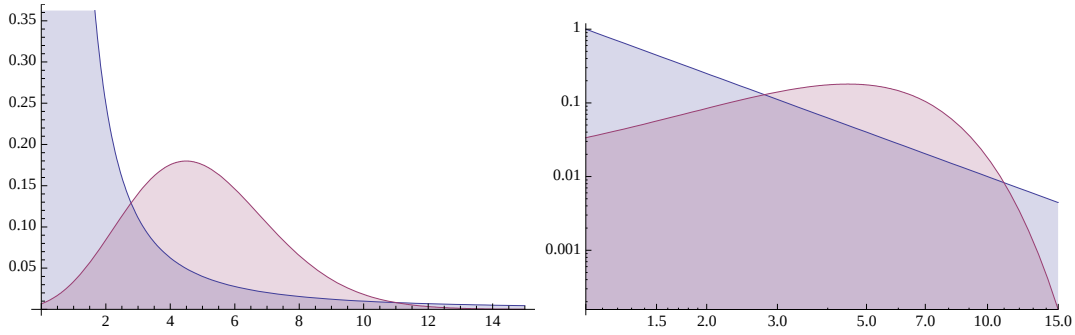

Figure 2.4: Power law degree distribution found in many empirical graphs (blue) and Poissonian distribution expected in large Erdős-Rényi random graphs (pink). One of the features of power laws is that they appear as straight lines in log-log plots and the exponent can be estimated as the negative slope. This property makes power laws easy to distinguish in various empirical data.

introduced a model based on two principles: growth and preferential attachment. They have shown that such a process results in a power law graph (often referred to as a scale-free graph). In detail their model looks as follows:

- Start with a small number of vertices (usually just a single vertex)
- In each step add a vertex, and connect it with existing ones giving preference to those already well connected (that is the probability that an edge will end up in some vertex v is proportional to the present degree d_v of v). The number of edges added in each step is fixed as a model parameter m.

Such a mechanism leads to a power law graph with exponent $\gamma = 3$. The argument proceeds as follows:

- Note that vertices with high degree acquire new connections faster than others. The rate at which a vertex i collects edges is approximately

$$\frac{\partial k_i}{\partial t} = \frac{k_i}{2t} \tag{2.3}$$

  which yields $k_i(t) = m\sqrt{t/t_i}$ where t_i is the time at which vertex i was added to the system.

- Using the above, one can estimate the probability that vertex i has connectivity smaller than k. We have

$$P(k_i < k) = P(t_i > m^2 t/k^2) \tag{2.4}$$

(the connectivity depends directly on the time the vertex was allowed to collect edges). Assuming quite naturally that vertices are added at a constant rate we obtain:

$$P(t_i > m^2 t/k^2) = 1 - P(t_i \leq m^2 t/k^2) = 1 - \frac{m^2 t}{k^2(t + m_0)} \tag{2.5}$$

where m_0 is the initial number of vertices (e.g. $m_0 = 1$). For large time scales $t/(t + m_0) \approx 1$. To obtain probability density one only has to differentiate $P(k) = \partial P(k_i(t) < k)/\partial k$ which gives

$$P(k) \approx \frac{2m^2}{k^3} \tag{2.6}$$

and so $\gamma = 3$ is independent of m and t

By varying the dependence of the attachment probability on connectivity (by introducing nonlinear dependency for example), other exponents from $[2, 3]$ are also achievable. It is fairly understandable that such a process is present in citation networks (a popular cited paper has a higher chance of hitting reference list than a new article), the World Wide Web (newly created page is likely to contain links to those already popular) or social networks (where being popular makes it more likely to gather even more popularity) but it is far more surprising that such principles are present in ecological, linguistic or telephone call networks. Biological networks however, often have exponents below 2, which suggests there is some other mechanism responsible for the network structure (see further section 2.4.3).

The *rich gets richer* phenomenon was first observed by Italian economist Vilfredo Pareto[4] (Pareto, 1896-1897) in his lecture notes, and hence power law distribution are now named Pareto distributions. Lotka (1926) found that the number of authors who published n papers is inversely proportional to n^2 in decennial index of chemical abstracts. Later Zipf (1932) found power laws in frequencies of occurrences of words in English language (similar results were afterwards found in other languages). Simon (1955) analyzed power laws in distributions of words in prose samples, distribution of scientists by the number of papers they published, distributions of cities by population, distribution of income by size and

[4]Probably many people in the past suspected this principle, but Pareto was the first to give it a formal statement in the framework of probability theory.

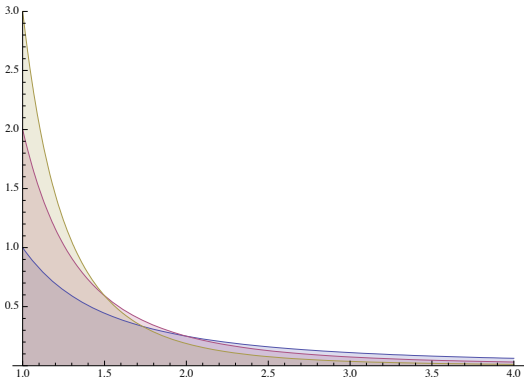

Figure 2.5: Pareto distribution $\frac{kx_m^k}{x^{k+1}}$ with mode $x_m = 1$ and scaling parameters $k = 1, 2, 3$.

distributions of biological genera by number of species. At the very beginning of his paper Simon notes: *"Its (the distribution's) appearance is so frequent and the phenomena in which it appears so diverse, that one is led to conjecture that if these phenomena have any property in common it can only be a similarity in underlying probability mechanisms"*. Simon's model based on preferential attachment caused some excitement and was attacked by Mandelbrot. Things had to wait until the very end of XX century when various research groups reported power laws in many real world graphs and the avalanche had started. Probably the best empirical graph to study today is the Internet, since it is huge (for practical study it is almost unbounded), and fairly easy to track on many levels. Furthermore the availability of the Internet encourages people to organize into various social networking services, which in turn provide further data for study graphs of social relationships. Again, these graphs are large, easy to obtain and fairly well reflect social networks which are otherwise hard to study. The fact that power laws are present in such large graphs ensures that this phenomenon is not some temporary fluctuation, but rather something quite universal. In particular the World Wide Web was extensively studied by independent research groups, which all found that the indegree exponent of the Web is 2.1 and outdegree 2.7. These figures are stable over a couple of years and sizes (Albert & Barabási, 2002; Barabási *et al.* , 2000; Broder *et al.* , 2000; Kumar *et al.* , 1999). This consistency over time and space is the reason why power law graphs are frequently referred to as *scale-free* graphs.

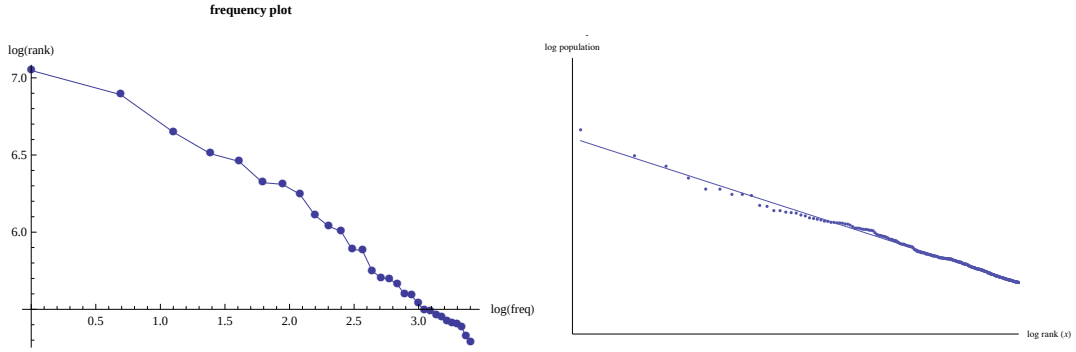

Figure 2.6: (Left) an illustration of the Zipfs law which says that the most frequent word will occur approximately twice as often as the second most frequent word, which will occur approximately twice as often as the third most frequent word etc. This sample is based on a fragment of Hamlet by Shakespeare. (Right) similar dependency of sizes of US cities.

2.4.3 Duplication models for biological networks

The exponent ranges found in diverse real world graphs differ depending on the field of study. Various communication networks have exponent between 2 and 3. There are examples of networks of higher exponents but they are rare (in fact networks with exponents above 3 are in many respects different than those below, see chapter 4 in Chung & Lu (2006)). Biological (in particular biochemical) networks however, frequently have exponents below 2 and such networks are again somewhat different than those with exponents in $[2, 3]$ (see Seyed-allaei *et al.* (2006)). The key process present in biological networks is duplication. Essentially most of the processes related to metabolism are based on continuous duplication working against the destructive force of disorder and growth of entropy. It turns out that duplication is indeed responsible for low power law exponents. Consider the following model:

- Start with some initial graph
- Select a random vertex v of the current graph. Create a new vertex u and connect u to v
- For each edge adjacent to v with probability p copy the edge to u (that is u inherits a number of neighbors of v)
- Repeat the above steps sequentially

The new vertex u can be regarded as a descendant of v. The initial graph of the process can be any graph, the process that runs for long enough will end up in

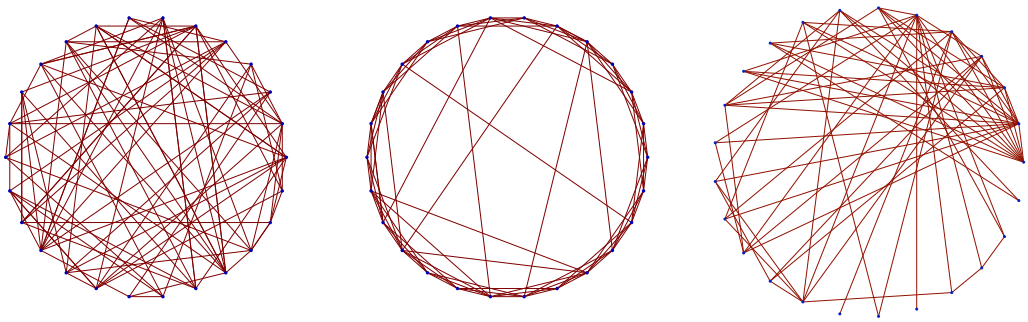

Figure 2.7: Graphs: random Erdős-Rényi, small world Watts-Strogatz, scale-free Barabási-Albert. Note in the last example an existence of a hub node with exceptional connectivity and a fair number of poorly connected nodes.

a power law graph with exponent depending on the probability p. The detailed estimation of the degree sequences of the above model is in chapter 4 of (Chung & Lu, 2006) and in (Chung *et al.* , 2003). Indeed depending on p power law graphs with exponents below 2 can be constructed which indicates that this model is suitable for biochemical networks.

2.4.4 Connectivity of the brain

The brain is probably the most sophisticated network we have right at our disposal, and yet it is very difficult to study. Neuronal fibers are of microscopic size and their number is astronomical, and most of all it is very hard to investigate neural connectivity without destroying its fragile structure. For that reason detailed empirical connectivity data of a nervous system is only known for simple organisms like Caenorhabditis elegans worm (there are also multiple partial results on connectivity of single cortical microcolumns and various thalamic structures but the global picture is still missing). It is still unknown what kind of a graph we will find in the brain. Will it be a power law network? Or will it resemble some hierarchic structure or will it rather be more like E-R random graph? There are even problems with defining the actual unit (a vertex) and the principles of connectivity. The neuron could be regarded as a basic unit, taking its synapses as edges. But on the other hand, the neuron could be regarded as a long wire, with the synapses being the information processing units. Or perhaps we should rather look at neuronal groups of some kind? Maybe each axonal branching point should be treated as a vertex? These questions are of a rather fundamental character but on the other hand the theoretical models of random networks studied at present are not usually based on any of the phenomena found

in the nervous system. Neurons certainly don't wire entirely randomly like the E-R graphs. Clearly there is a local structure and global connections, but it does not fit well into the Watts-Strogatz rewiring model. Finally, even though there certainly is a feedback loop which makes an active neuron more likely to wire to others, there is no such thing as growth and preferential attachment like in the Barabási-Albert model. Are there any other more realistic principles related to the brain structure that would lead to power law graphs?

Empirical studies of Caenorhabditis elegans worm (Amaral *et al.* , 2000; Koch & Laurent, 1999) imply that the graph of neural connections with neurons as vertices and synapses as edges has an exponentially decaying tail. This result, although important, does not answer many of the above questions. Some synapses can be weak and unimportant while other ones can be strong and essential. Investigating static morphology of neural connections might not be the right way to seek power law connectivity, which might be hidden in the networks dynamics (which is far more difficult to study experimentally). Finally, the notion of a single neuron might not be the right information processing unit of the brain (in particular fMRI studies of Eguíluz *et al.* (2005) show that functional networks connecting correlated centers of activity in the brain are in fact both scale-free and small world).

On the other hand, artificial neural networks are usually either built on fully connected (Hopfield network), well-structured and layered (Multi Layer Perceptrons) or completely local (Pulse Coupled Networks in many applications) graphs. Fully connected graphs are inefficient in large applications and furthermore inconsistent with biological prototype. Layered connectivity is somewhat artificially imposed by the drawbacks of learning algorithms, in particular the inability of a back-propagation algorithm to deal with recurrent connections. Local connectivity results in interesting phenomena (Piękniewski, 2006), but again lacks biological motivation. Some interesting results were obtained for Small World Networks (Kwok *et al.* , 2006; N.Davey *et al.* , 2004) and scale-free graphs (Perotti *et al.* , 2006; Stauffer *et al.* , 2003), but in this case the connectivity was not a result of neural activity but was rather imposed as a background for already existing models. In the current state of the art to our best knowledge there is no clear mechanism of existing neural models that would lead to scale-free connectivity. The results presented in further chapters (4 and 5) of this thesis are aimed at showing such a mechanism and studying related phenomena.

2.5 Appendix: Mathematical tools

The mathematical arsenal in dealing with graphs is diverse. There are various combinatorial results, that fit well into regular graph set-up (frequently found in

algebra or combinatorics), but random graphs usually fail to meet the rigorous conditions[5] required by these theorems[6]. Furthermore, algorithmic tools that work for graphs of moderate size are unsuccessful with huge graphs (frequently found in surrounding reality). The problems investigated in random graph theory may seem trivial (like whether a graph is connected or not), but they turn out to be pretty difficult when the examined graph has millions of vertices (or when we want to establish whether a typical random graph out of an astronomical number of graphs satisfying certain conditions has some property or not). In this section we shortly introduce some concepts that play an important role in random graph analysis.

2.5.1 Spectral methods

Any graph G with n nodes can be represented by its adjacency matrix A of $n \times n$ elements A_{ij} whose values are $A_{ij} = 1$ if there is an edge from node i to j and $A_{ij} = 0$ otherwise. For undirected graphs adjacency matrices are symmetric and therefore have real eigenvalues. The set of eigenvalues of the adjacency matrix A corresponding to a graph G is called the spectrum of the graph G. Eigenvalues contain a lot of interesting information about the original graph compressed into a fairly small sequence of real numbers. Recent findings of Czerwinski (2007) (not yet published in a peer reviewed journal but available at arXiv) indicate that graph eigenvalues together with their eigenvectors contain enough information to solve the graph isomorphism problem in polynomial time[7]. An important fact is that the spectra are fairly easy to approximate numerically. Most interesting properties of graph spectra are that the largest eigenvalue of adjacency matrix gives the idea of growth of number of alternative paths between vertices. Many other enumeration problems can be stated in terms of adjacency matrix eigenvalues (see (Godsil & Royle, 2001) for a survey). To get a general idea of the spectral properties of a graph it is useful to define its spectral density:

$$p(\lambda) = \frac{1}{n} \sum_{j=1}^{n} \delta(\lambda - \lambda_j) \tag{2.7}$$

[5] In fact graphs with a non trivial group of automorphisms are rare, see chapter 2 in Godsil & Royle (2001).

[6] Many interesting results are obtained for regular or nearly regular graphs. In particular large and decently connected Erdős-Rényi graphs are fairly regular (meaning that the degrees of vertices do not differ too much from their respective expectations). In contrast, power law graphs display nearly the opposite of such a definition of regularity, in that they have a large variety of node degrees, and so require quite different mathematical tools to study.

[7] It is known that the subgraph isomorphism problem is NP-complete, but the graph isomorphism problem is likely to be easier.

which may approach a continuous function as $n \to \infty$. Spectral density[8] has interesting links with graphs properties, for example $k-th$ moment of spectral density is the number of paths of length k that return to the their origin (possibly visiting other nodes multiple times). An interesting result was established by Wigner (1955, 1957, 1958), stating that for a wide range of matrices whose entries are random variables, spectral density approaches semicircular function (known as Wigner's semi-circle law). In particular random Erdős-Rényi graph fits this framework (Füredi & Komlós, 1981).

Alternatively (but not equivalently) graph spectra can be defined as sets of eigenvalues of a (normalized) Laplacian matrix. A Laplacian matrix is constructed as follows:

$$L = D - A \tag{2.8}$$

where A is the adjacency matrix and D is a diagonal matrix with $(i,i)-th$ entry containing the degree of node i. Matrix L can be normalized by

$$\mathcal{L} = D^{-1/2} L D^{-1/2} \tag{2.9}$$

so that diagonal values of $\mathcal{L}$ are all equal 1. This definition relates better to some graph invariants. For examples the spectral gap (the absolute value of the difference between the first and second consecutive eigenvalues of $\mathcal{L}$) gives a lot of information about expansion properties, in particular the Cheeger constant (Alon, 1986; Alon & Milman, 1985). These notions are of particular interest in the study of expander graphs, which are very useful tools for theoretical computer science (see Chung (1997) for more details).

2.5.2 Tree matrix theorem

In this context it is worth recalling the tree matrix theorem by Gustav Kirchhoff (1847) which states that the number of nonidentical spanning trees of a graph is equal to any cofactor of the (not normalized) combinatorial Laplacian. Kirchhoff found this relation when studying the flow of current in electrical networks represented by graphs, and related the total current flow to the number of possible spanning trees (see chapter II in Bollobas (1998) for details). This theorem has a simple and elegant proof (based on the fact that L is the product of incidence matrix and its transpose, and application of Cauchy-Binet formula) which gives the flavor of relationships between determinants and various graph properties that are the object of extensive study.

[8] Formally we mean spectral measure, which may not have a density in the exact sense, but the term "spectral density" is used frequently in the literature.

2.5.3 Clustering coefficient

Clustering coefficient was introduced in Watts & Strogatz (1998) to reveal the local structure of a graph. For a node i its clustering coefficient is defined as follows:

$$C_i = \frac{2|\{e_{j,k}\}|}{d_i(d_i - 1)} \tag{2.10}$$

where nodes j and k are adjacent to i, and d_i denotes degree of i. In other words it is the number of connections between the neighbors of the node i divided by the number of all possible connections between the neighbors of i. Clustering coefficient can be alternatively defined as

$$C_i = \frac{2\lambda_G(i)}{d_i(d_i - 1)} \tag{2.11}$$

where λ_G is the number of triangles on vertex i. This definition suggests quite an efficient algorithm of computing the clustering coefficient, since the number of triangles coming from a node (assuming there are no loops of length one at vertices) is simply the corresponding entry on the diagonal of A^3, where A is the adjacency matrix. We assume that $C_i = 0$ if $d_i = 0$ and $C_i = 1$ if $d_i = 1$. The average clustering coefficient of the whole graph is simply the average of coefficients computed for every node and hence it coincides with the suitably normalized trace of A^3. High clustering coefficient is a typical feature in many empirical graphs (see section 2.4.1) while is very low in E-R random graphs. Other interesting features of a graph include the information on how clustering coefficient is distributed over vertices of different degrees and, likewise, how the degree is distributed over nodes having clustering coefficient within a fixed range etc.

Chapter 3

Contemporary neural models

3.1 Introduction

It is widely believed that the origin of magnificent variety of brain activities lies in the activity of neurons. Like any other cells in a living tissue, neurons metabolize, have their cell body, intracellular vesicles, mitochondria, nuclei and other structures. What makes neurons different is their ability to generate action potentials – electro-chemical waves on the cell's membrane. Such action potentials (often referred to as spikes) propagate through the membrane until they end up in a synapse - a terminal between two neurons. Synapses are complex chemical devices[1] which are responsible for exchanging action potentials between different neurons. This electrical activity of neurons suggests their importance in generating brain activity, yet there are other cells present in the nervous tissue as well and the role that some of them play remains a mystery. Empirical data shows that there are about 10^{14} neurons in a typical human brain, each of them can be connected with up to 10000 others. Many of these connections are probably redundant, yet still these numbers are impressive. The connections are mostly local, but some neurons could be up to a 1 meter long, and spread their axons in completely different parts of the brain or body (in fact much of the nervous tissue is composed of neuronal fibers). Systems of this size and complexity are beyond the range of current computer simulations unless massive simplifications are made. This circumstance drives researchers into extracting the most important computational properties of neurons and creating simple empirical models, which in many cases lack the morphological details and have ridiculously simplified dynamics. Even these simplified models often provide a great challenge for theoretical approach, as well as useful tools for engineering and practical com-

[1] There are also other types of synapses based on direct flow of electrical current, referred to as "gap junctions" or "electrical synapses".

puter science (computational intelligence in particular).

In this chapter we briefly introduce the general mechanism leading to an action potential (at least as far as we understand these mechanisms today). We start with the classical Hodgkin-Huxley model (section 3.2) and then depart into various directions introducing reduced ionic models and phenomenological models (sections 3.4 and 3.5). In particular we cover only briefly the classical models utilized in engineering (section 3.3), based on the firing rate idea. These models have been thoughtfully discussed in the literature (see Rojas (1996) for example) and share a common feature that the temporal structure of neural signals is to a large extent ignored. For that reason, these models are computationally efficient, but lack many, possibly important features of biological neurons.

3.2 Hodgkin-Huxley model

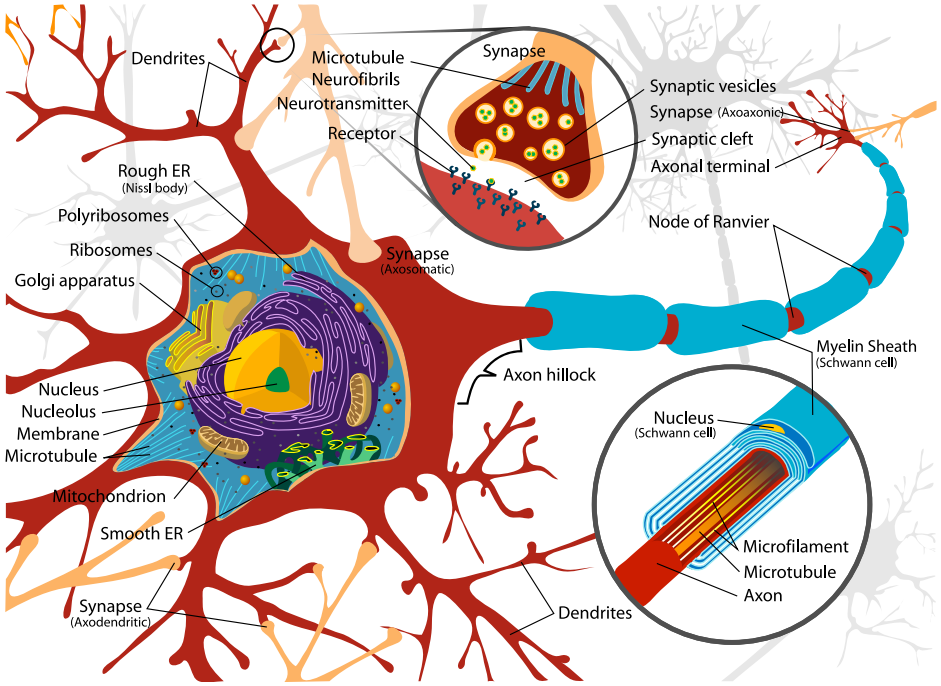

Figure 3.1: Complete neuron cell diagram. Image courtesy of Mariana Ruiz Villarreal (Available at: `http://en.wikipedia.org/wiki/Image:Complete_neuron_cell_diagram_en.svg`)

Theoretical research on neuronal dynamics received strong input in 1952 with the seminal work of Sir Alan Lloyd Hodgkin and Sir Andrew Fielding Huxley (Hodgkin & Huxley, 1952), who studied electrochemical properties of the giant

squid neurons (earlier empirical studies of Sir Edgar Douglas Adrian (1926, 1928) should also be noted). They were able to create an electrical circuit that resembled the electrical properties they observed. The key to understand the mechanism of spike propagation in axons is to understand the properties of infinitesimal piece of neuron's membrane. One such piece consists of the membrane (of nonzero capacitance), extracellular and intracellular media both filled with a family of ions and ion pumps. Ions are able to diffuse through the membrane. There are three forces that drive the diffusion of ions:

- Electric force, since ions are charged
- Ordinary diffusion which tries to equalize the saturation of ions on both sides of the membrane
- Ionic pumps which continuously pump certain ionic species across the membrane

Eventually each ion's concentration on both sides of the membrane reaches an equilibrium, which may result in a non zero electric potential (Nernst potential) across. All in all the membrane is usually polarized with rest potential of $\approx -65mV$. The Hodgkin-Huxley model takes into account sodium (g_{Na}) and potassium (g_K) ionic channels, leak channel (g_L). There are other ions which also play some role in the membrane dynamics (like calcium) but they are less important. Eventually the model can be described in terms of the following set of differential equations:

$$\begin{aligned} C_m \frac{dV}{dt} &= -g_L(V - V_L) - g_{Na} m^3 h (V - V_{Na}) - g_K n^4 (V - V_K) \\ \frac{dm}{dt} &= \alpha_m(V)(1 - m) - \beta_m(V) m \\ \frac{dh}{dt} &= \alpha_h(V)(1 - h) - \beta_h(V) h \\ \frac{dn}{dt} &= \alpha_n(V)(1 - n) - \beta_n(V) n \end{aligned} \tag{3.1}$$

where

$$\begin{aligned} \alpha_n(V) &= 0.01 \frac{10 - V}{e^{\frac{10-V}{10}} - 1}, & \beta_n(V) &= 0.125 e^{\frac{-V}{80}}, \\ \alpha_m(V) &= 0.1 \frac{25 - V}{e^{\frac{25-V}{10}} - 1}, & \beta_m(V) &= 4 e^{\frac{-V}{18}}, \\ \alpha_h(V) &= 0.07 e^{\frac{-V}{20}}, & \beta_h(V) &= \frac{1}{1 + e^{\frac{30-V}{10}}} \end{aligned}$$

and typically:

$$\begin{array}{lll} E_k = -12mV, & E_{Na} = 120mV, & E_L = 10.6mV, \\ g_K = 36mS/cm^2, & g_{Na} = 120mS/cm^2, 7 & g_L 7 = 0.3mS/cm^2, \\ C = 1\mu F/cm^2 & & \end{array}$$

The first line in equation 3.1 describes the changes of the total membrane potential, whereas the latter three describe the dynamics of ionic channels. The variables m and n are the statuses of activation gates of sodium and potassium channels respectively (0 - fully closed, 1 - fully open), whereas h is inactivation gate status for sodium (0 - inactivated, 1 - deinactivated). That is, increase in V causes increase in m, which increases sodium current and so on. The gate variable m is taken to a power of 3 in the voltage equation since there are 3 activation gates and one inactivation gate h (and there are 4 activation gates for potassium respectively). If some external stimuli (possibly nearby membrane compartment) changes the V variable driving it away from the rest state, it starts a whole cascade of events. Since V changes, also m, h and n change (they depend on voltage through α and β functions), and so ionic conductances change etc. Different ions have different response rates (α and β functions) and so, a strong enough impulse may cause sodium conductance to grow and in turn the system will be amplifying V until sodium inactivates and potassium ions come to play. In the early stage of a spike, increase in V causes increase in g_{Na}, which causes Na^+ ions inflow, which in turn causes further increase in V. This amplifying feedback goes on for a couple of milliseconds (V reaches values beyond $30mV$), when slow K ions catch up. While sodium channel eventually inactivates, potassium instantly re-polarizes the membrane which enters the so called refractory period, and then slowly converges to the equilibrium back again. The above formulation 3.1 is presented for historical reasons, it is more convenient to express Hudgkin-Huxley equation a bit differently:

$$\begin{aligned} C_m \frac{dV}{dt} &= -g_L(V - V_L) - g_{Na} m^3 h(V - V_{Na}) - g_K n^4 (V - V_K) \\ \frac{dm}{dt} &= (m_\infty(V) - m)/\tau_m(V) \\ \frac{dh}{dt} &= (h_\infty(V) - h)/\tau_h(V) \\ \frac{dn}{dt} &= (n_\infty(V) - n)/\tau_n(V) \end{aligned} \tag{3.2}$$

and refer to $m_\infty, h_\infty, n_\infty$ as steady state activation functions and $\tau_m(V), \tau_h(V), \tau_n(V)$

voltage dependent time constants. In this case:

$$\begin{aligned} n_\infty &= \alpha_n/(\alpha_n+\beta_n), & \tau_n &= 1/(\alpha_n+\beta_n) \\ m_\infty &= \alpha_m/(\alpha_m+\beta_m), & \tau_m &= 1/(\alpha_m+\beta_m) \\ h_\infty &= \alpha_h/(\alpha_h+\beta_h), & \tau_h &= 1/(\alpha_h+\beta_h) \end{aligned}$$

These functions are obtained empirically for different ionic channels and approximated appropriately by sigmoids, gaussoids or other elementary functions. For example $n_\infty(V)$ denotes the asymptotic conductance of potassium channel at voltage V, which is reached with relaxation speed proportional to $\tau_n(V)$. In the event of an action potential the voltage constantly changes, and so asymptotic conductances change, as well as the rate of their relaxation, which makes the dynamics pretty complex (see figure 3.2) and computationally demanding. Consequently Hudgkin-Huxley equation is not suitable for large scale simulations of neurons, even on contemporary computers.

3.2.1 Structural dynamics of neural cell

The Hudgkin-Huxley equation describes the dynamics of a infinitesimal piece of the neuronal membrane, whereas neurons have a rather complex anatomy (see figure 3.1). In order to accurately reproduce the neuronal activity one has to reconstruct the spatial details of the cell (that may include hundreds of dendrites, synapses, dendritic spines and internal structures) and simulate the spatial version of the equation over such a complex domain. The spatial equation is a parabolic second order equation of the form:

$$CV_t = \frac{a}{2R}\frac{\partial^2 V}{\partial x^2} + I - I_K - I_{Na} - I_L \tag{3.3}$$

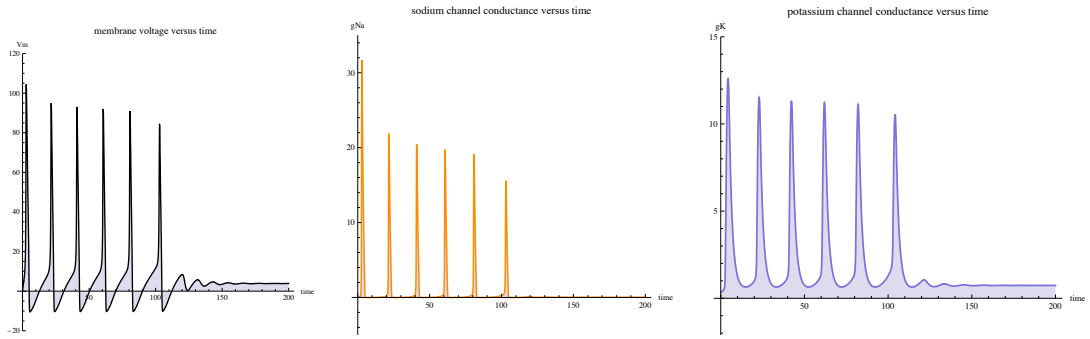

Figure 3.2: Example evolution of membrane potential (left), sodium ion (middle) and potassium ion (right) conductance in response to 6.2 millivolt pulse.

referred to as the Hudgkin-Huxley cable equation. R is the intracellular resistivity and a is the cell radius (both may vary). The typical practice is to divide a neuron into set of uniform compartments connected via conductances. Compartments themselves have the Hudgkin-Huxley dynamics (also the spatial version thereof when the compartments are large e.g. long dendrites). This method allows for simulation of dendritic tree branches and complex geometry, but obviously is computationally demanding - depending on the geometrical complexity (a pyramidal neuron may have thousands of dendrites), a simulation of a single neuron could be a challenge for contemporary computers (especially if it is supposed to be a real time simulation). In simulations of large populations of neurons one usually uses single compartment models, while spike propagation dynamics is squeezed into a simple synaptic conduction delay (see chapter 6 for an example of such a model).

3.3 Firing rate models

Since electrophysiology of neurons is complex, researchers in 40'ies and early 50'ies of XX^{th} century looked for something less computationally demanding. In those days recent development of digital computers launched high hopes for quick occurrence of the artificial intelligence, and consequently some immense simplifications into neural models had to be made. Many researchers believed that neurons perform tasks analogous to those of logical gates in electronic circuits of computers, and the whole complexity of action potentials is some biological throwback rather than something of essential importance. The key to simplification of neural models were the following observations[2]:

- A neuron which is not stimulated remains quiescent
- When the level of stimulation reaches certain threshold the neuron begins to produce action potentials and becomes active

In this setup the neuron can be attributed with two states - quiescence and activity - depending on the level of external stimuli, ignoring the temporal structure of action potentials. The input consists of signals from dozens of synapses. In this coarse grained setup one can assume that this input is simply summed linearly. Such simplified models resemble to some extent logical gates of electronic circuits. In the classical model (next subsection) the state of a neuron is binary (firing - 1, quiet - 0). In later developments by Frank Rosenblatt (1958, 1988) introduced "synaptic" weights and continuous activation functions (the differentiability of

[2]As discussed in other sections of this thesis, both of these observations are rough approximations, that are frequently violated in biological neurons.

activation function became important with the development of back-propagation algorithm (Rumelhart *et al.* , 1986; Werbos, 1974))

3.3.1 McCulloch-Pitts neuron and the perceptron

The model informally introduced above is called the McCulloch-Pitts artificial neuron (in honour of Warren McCulloch and Walter Pitts) and was introduced in (McCulloch & Pitts, 1943) nearly a decade earlier than Hudgkin-Huxley equations. Formally the neuron is a simple entity consisting of a number of binary inputs that sum up to $I = \sum_{i=1}^{n} E_i$, a threshold θ and a transfer function (activation function), which in this simple model is:

$$f(I) = \begin{cases} 1; & \text{whenever activation exceeds the threshold } \theta \\ 0; & \text{otherwise} \end{cases} \tag{3.4}$$

In the case of McCulloch-Pitts neuron the input $E_i \in \{0, 1\}$. Later, around 1958 Frank Rosenblatt combined the findings of Donald Hebb (1949) with the McCulloch-Pitts neuron and introduced the perceptron (Rosenblatt, 1958, 1988). The perceptron allowed for continuous inputs, included synaptic weights, more complex activation functions and was equipped with an algorithm that found a solution to a classification problem (if one existed). The algorithm however worked only for one unit. In the case of the perceptron the input is the dot product $I = \sum_{i=1}^{n} E_i w_i$, and the weights w_i are variable parameters which change in the course of learning. Geometrically such a neuron is capable of recognizing linearly separable sets of data which is not very impressive unless many such units are joined together (a single neuron is not even capable of recognising the simple xor problem), and for such structures there was no learning algorithm. For many years this drawback slowed down the development of artificial neural networks (and was a source of strong criticism of the whole idea (Minsky & Papert, 1969)), since in order to solve more complex problems networks had to be constructed either manually or via one of the constructive algorithms (tower algorithm, tiling algorithm etc. (Gallant, 1990, 1993)). This was a major drawback since networks created via constructive algorithms could not be easily controlled in terms of regularization, and manually tuned networks could easily fail on new data. Other models like the ADALINE (Widrow & M.E. Hoff, 1960) were also derived from McCulloch-Pitts neuron, but the field remained in a stale until the back-propagation algorithm came to play.

3.3.2 Multi Layer Perceptrons

The long anticipated revival of the discipline occurred in the 80'ies of the XX^{th} century, with the back-propagation algorithm (Rumelhart *et al.* , 1986), although

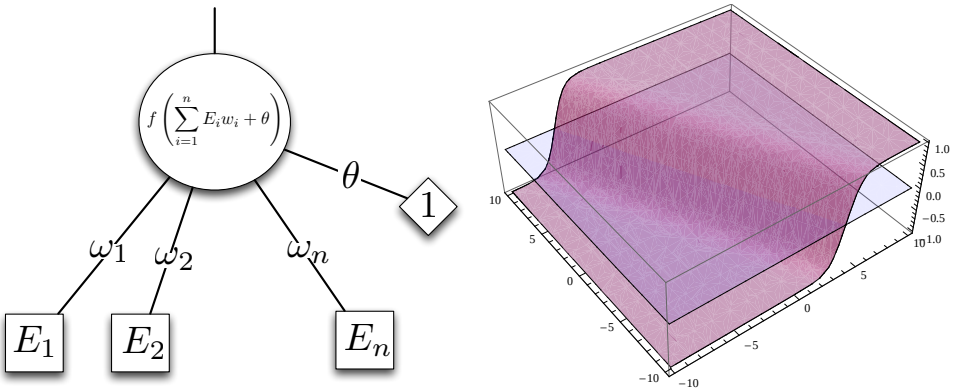

Figure 3.3: Schematic diagram of the artificial neuron (left) and a geometrical interpretation of its activation in 2d space (the plotted function is smooth sigmoidal for better readability).

the idea dates back to 1974 work of Paul Werbos (1974). Rumelhart, Hinton and Williams rediscovered the algorithm a decade later, and this time their paper gathered proper attention of the society. The Back-propagation algorithm for the first time allowed for completely automatic construction and training of multi layered networks consisting of sigmoidal perceptrons (MLP) to solve complex categorization tasks. A lot of practical implementations of the idea emerged in the following years (like associative memories, data-mining systems) and a cascade of interesting research results was started. In particular, computational properties of such networks were studied. It has been shown that a three layer network solves any problem that is solvable by a network with arbitrary number of layers (Cybenko, 1989; Hornik, 1991) (this result is known as the universal approximation theorem). It also turned out that the general problem of deciding whether for a given set of training data and a given network architecture there exists a set of weights that would allow for good classification is NP-complete (Blum & Rivest, 1993). This result did not stop the development, since in practical applications networks' structure can be dynamically adjusted (ontogenic neural networks), and the classification does not necessarily have to be ideal. The result shows however, that even such a vastly simplified neuronal model can be difficult for computational analysis.

3.3.3 Hopfield network

Multi layer perceptrons are examples of feed forward networks, that is they don't contain any feedback connections. This simplifies the analysis (and enables the straightforward implementation of the error back-propagation), but is biologically inaccurate. Most of the brain structures are heavily recurrent – this fact estab-

lished a need for different models (and, more important: learning algorithms). This request was answered in 1982 with the seminal paper by John Hopfield (1982). The Hopfield network in the original setup is also based on McCulloch-Pitts-like neurons but is completely recurrent, and requires entirely different approach in the analysis. In this case the appropriate formalism turned out to be the statistical mechanics. The learning algorithm for the Hopfield network is based on Hebbian learning for pattern recognition tasks, or derived directly from the form of Hamiltonian for combinatorial optimization problems. Later Hopfield network got equipped with stochastic dynamics(Ackley *et al.* (1985); Hinton & Sejnowski (1986, 1983) and independently Hofstadter (1984)) which resulted in the so called Boltzmann machine (Aarts & Korst, 1989) (combination of the Hopfield architecture with simulated annealing). The model proved to be useful in pattern recognition, computer vision, robot control but in general had not obtained particularly good results in combinatorial optimization due to emergence local minima and slow convergence. In general methods based on simulated annealing are weak in hard combinatorial optimization unless the solution has a carefully selected structure of search space which results in smoother energy landscape (see chapter "The traveling salesman problem: a case study" in (Aarts & Lenstra, 1997) for a comprehensive study based on the TSP).

Hopfiled networks generally fit into the spin-glass theory framework, while locally connected Hopfield networks are similar to Ising model with respect to their statistical mechanics and can be studied in terms of Pirogov-Sinai theory and related methods (Zahradník, 1984). A study of phase diagrams in such locally connected Hopfield networks can be found in (Piękniewski, 2005, 2006; Piękniewski & Schreiber, 2005).

3.4 Reduced ionic models

The Hudgkin-Huxley model is computationally demanding, theoretically difficult (four variables, nonlinear kinetics) and yet not entirely biologically accurate. There are dozens of other ionic channels either voltage or calcium gated, and there are more being discovered almost every year. It is certainly not obvious whether all this variety is a necessity, or is it some biological/evolutional artifact. It seems however that the key issue in neurocomputation is excitability, that is the ability of neurons to generate action potentials (other mechanisms possibly play some regulatory role). It turns out that for this single task, even the Hudgkin-Huxley model contains a lot of unnecessary features. This led to a development of minimal conduction based models, that is the reduction of Hudgkin-Huxley model as much as possible with the constraint that it still retains excitability. Surprisingly even extensive simplifications retain their ability to generate action potentials

as long as there are at least two variables (which is a necessary condition, since one dimensional dynamical systems do not have limit cycles). One of the most prominent models is the so called "persistent sodium plus potassium"($Na + K$) (see section 5.1.2 in Izhikevich (2006a)). The model:

$$\begin{aligned} C_m \frac{dV}{dt} &= -g_L(V - V_L) - g_{Na} m(V - V_{Na}) - g_K n(V - V_K) \\ \frac{dm}{dt} &= (m_\infty(V) - m)/\tau_m(V) \\ \frac{dn}{dt} &= (n_\infty(V) - n)/\tau_n(V) \end{aligned} \tag{3.5}$$

is mathematically equivalent to the model of Morris & Lecar (1981). Since sodium activation is almost instantaneous, the model could be further reduced to

$$\begin{aligned} C_m \frac{dV}{dt} &= -g_L(V - V_L) - g_{Na} m_\infty(V - V_{Na}) - g_K n(V - V_K) \\ \frac{dn}{dt} &= (n_\infty(V) - n)/\tau_n(V) \end{aligned} \tag{3.6}$$

Two dimensional dynamical system can be conveniently analyzed in terms of phase plane, nullclines[3] etc. Figure 3.4 shows a phase portrait of $Na + K$ model with sample parameters $C = 1$, $E_L = -80$, $E_{Na} = 60$, $E_K = -90$, $g_L = 8$, $g_{Na} = 20$, $g_K = 10$, $I = 0$ and $m_\infty(V) = \frac{1}{1+e^{\frac{-20-V}{15}}}$, $n_\infty(V) = \frac{1}{1+e^{\frac{-25-V}{5}}}$ and $\tau(V) = 1$. These parameters are biologically reasonable, taken from chapter 4 in (Izhikevich, 2006a). When I increases the system approaches a bifurcation and a limit cycle is created responsible for spiking. Obviously any further reduction of $Na + K$ model would strip its most important property, that is excitability, therefore this model is minimal. There are other combinations possible:

- "transient sodium model" (see section 5.1.3 in Izhikevich (2006a)) - an interesting model that includes only transient sodium and leak current. Surprisingly this ridiculously simple model is still capable of generating action potentials!

$$\begin{aligned} C_m \frac{dV}{dt} &= -g_L(V - V_L) - g_{Na} m^3 h(V - V_{Na}) \\ \frac{dh}{dt} &= (h_\infty(V) - h)/\tau_h(V) \\ \frac{dm}{dt} &= (m_\infty(V) - m)/\tau_m(V) \end{aligned} \tag{3.7}$$

[3] A nullcline is a locus of points where the vector field vanishes in one of the base (x/y) directions.

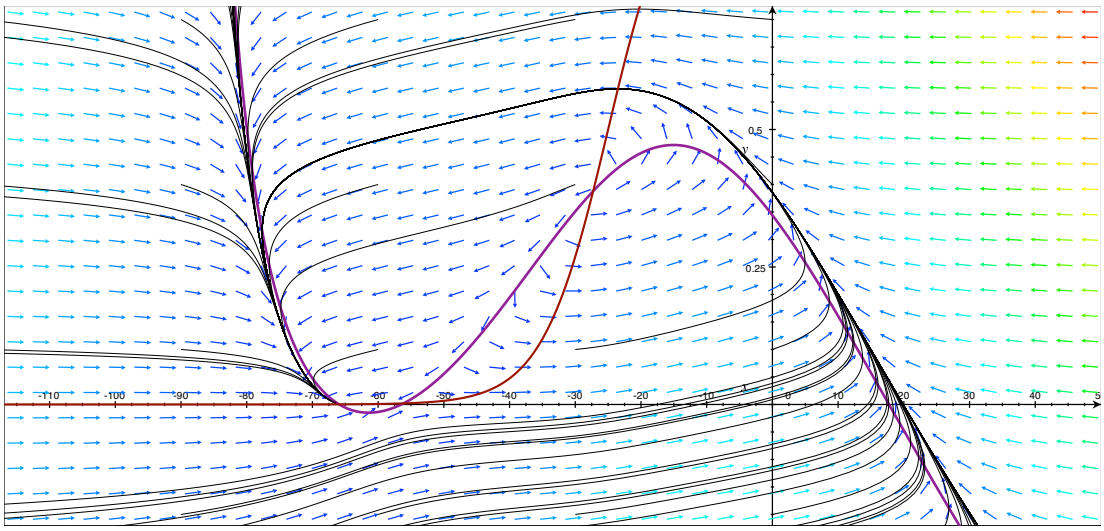

Figure 3.4: Phase portrait of the instantaneous "persistent sodium plus potassium" model. The nullclines are depicted with bold colored lines, while a number of trajectories are depicted with thin black lines. There are equilibria (stable and unstable) on the intersections of nullclines responsible for rest state, and a mechanism of excitability (a "shadow" of the limit cycle).

Again the sodium activation gating variable can be replaced with m_∞ for simplicity. This model can produce spikes solely depending on sodium activation/inactivation and leakage. The spikes, apart from having rather slow re-polarization, are not very different than those of other models.

- "persistent sodium plus h-current model" (see section 5.1.4 in Izhikevich (2006a)) - used to model sub-threshold voltage oscillations in some thalamic and cortical neurons

- "h-current plus inwardly rectifying potassium model" (see section 5.1.5 in Izhikevich (2006a)) - a rather weird model

- "persistent plus inwardly rectifying potassium model", "transient potassium model" (see sections 5.1.6 and 5.1.7 in Izhikevich (2006a)) - models consisting of only K^+ currents, yet still able to produce sustained oscillations.

All of these models are conveniently discussed in chapter 5 of Izhikevich (2006a). There are also other approaches to simplification of the Hudgkin-Huxley equation. In early computer simulation by Russian scientists Krinskii & Kokoz (1973) it came out that

$$n(t) + h(t) \approx 0.84 \tag{3.8}$$

The analysis of (n, h) plane reveals that the values of n and h lie near $h = 0.89 - 1.1n$ line (see section 5.2 in Izhikevich (2006a)). By substituting m gating variable with m_∞ (instantaneous sodium kinetics), one can obtain two dimensional reduced Hudgkin-Huxley model as follows:

$$\begin{aligned} C_m \frac{dV}{dt} &= -g_L(V - V_L) - g_{Na} m_\infty^3 (0.89 - 1.1n)(V - V_{Na}) - g_K n^4 (V - V_K) \\ \frac{dn}{dt} &= (n_\infty(V) - n)/\tau_n(V) \end{aligned} \tag{3.9}$$

3.5 Phenomenological spiking models

Since the key feature of neurons is the excitability, then why bother with the ionic zoo? Wouldn't it be better to create models that are simply excitable? This paradigm is the foundation for creating the so called phenomenological models, which simply mimic the phenomenon of excitability ignoring the biological and electrochemical details. The pros and cons of these models are discussed in (Izhikevich, 2004).

3.5.1 Integrate and fire models

The key property of neurons is that they accumulate some input, and when exceeding a certain threshold they fire. Simplified models of McCulloch-Pits perform "spatial" integration while completely ignoring temporal structure of inputs (time is discretized, and the neuron does not have any memory of its previous state). Conduction based models on the other hand, rely heavily on the temporal structure of inputs, but are difficult in analysis. Between these two extremes there are a number of "integrate and fire" models, the simplest of which can be expressed:

$$C_m \frac{dV}{dt} = I(t) \tag{3.10}$$

that is the membrane voltage at time t is simply an integral of input $I(t)$ over time. When V reaches a certain threshold value V_{th} a delta function action potential is fired and everything is reset to the initial state. This model was investigated in 1907 year by Lapicque (Abbott, 1999). Basically this model resembles a capacitor being constantly charged by an input current, instantly discharged after reaching the threshold. The only difference is that there is a spike signal sent as an output. Presented mechanism is not biologically accurate, since it can obtain arbitrary high spiking frequencies with high input which is not the case for biological neuron (lack of refractory period). Another issue, is that the neuron "remembers" its

history for arbitrary long time before it gets discharged, and this again is in contrast with biological neurons which have leaks, and return to steady state some time after excitation. A more biologically plausible is the leaky integrate and fire model:

$$C_m \frac{dV}{dt} = I(t) - \frac{V}{R} \tag{3.11}$$

which converges to the equilibrium (zero charge) with rate proportional to R. Both of these models can be simulated in an "event driven" fashion (assuming that the input is actually relevant only when there is an action potential coming), that is the parameters of neuron can be recomputed when some input actually comes in, and don't have to be constantly updated (which is the case for more complex dynamical models), which renders them useful in large scale simulations. In particular these models are extensively used in the concept of Liquid State Machines (Maass & Markram, 2004; Maass *et al.* , 2002), a fairly new notion of neuronal microcircuits performing calculations via "submerging" an input into a high dimensional dynamical system (liquid) and reading of the properties of resulting dynamical attractor.

Another interesting model called *neuromime* based on leaky integrators was introduced in (French & Stein, 1970) (see figure 3.5). In this case the firing threshold is variable and depends on previous activity (which is far more plausible from the biological point of view). An input for neuromime is supplied to the first leaky integrator. Its output is then compared with the dynamical threshold Θ. If it exceeds Θ a spike is generated. The output spike is supplied back to another leaky integrator which is responsible for providing Θ. Therefore Θ gets increased, which causes further spikes less probable. If the neuron doesn't fire, Θ converges slowly to Θ_0 (minimal threshold). This model was explored in (Eckhorn *et al.* , 1990) and is the basic unit for Pulse Coupled Neural Networks (PCNN) (Johnson & Padgett, 1999). Unlike integrate and fire neurons, neuromime accommodates its firing rate to the magnitude of input, but in contrast to biological neurons it lacks important dynamical features like subthreshold oscillations or excitation

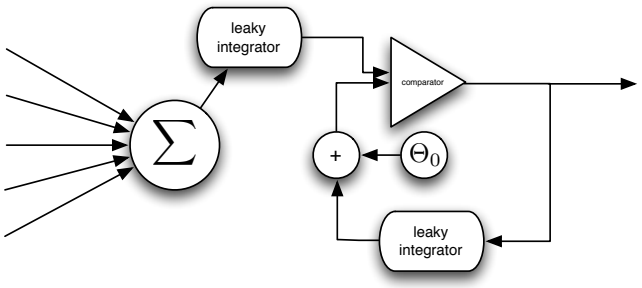

Figure 3.5: Schema of the neuromime pulse generator.

block.

3.5.2 Fithugh-Nagumo model

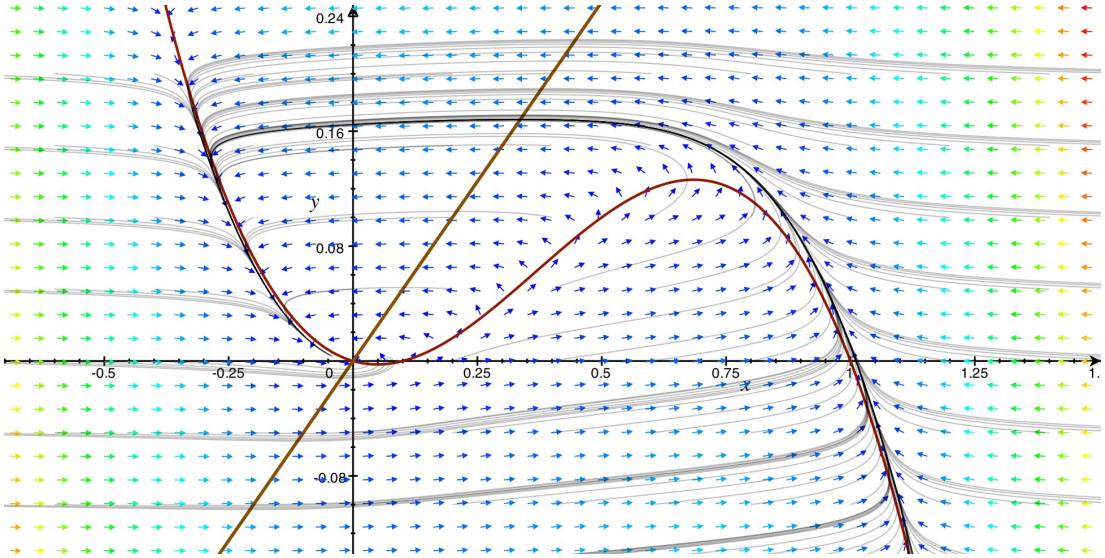

Figure 3.6: Phase portrait of the FitzHugh-Nagumo model. Notice that the phase portrait resembles the one of $Na + K$ model (figure 3.4).

The FitzHugh-Nagumo model introduced and explored in (FitzHugh, 1961, 1969, 1955; Nagumo *et al.* , 1962) is defined as follows:

$$\begin{aligned} \frac{dV}{dt} &= V - V^3 - w + I \\ \tau\frac{dw}{dt} &= V - a - bw \end{aligned} \tag{3.12}$$

where V is the voltage-like variable and w is referred to as the recovery variable. This model abstracts from the ionic conductances and focuses on key features of neuronal dynamics. Notice that the phase portrait of FitzHugh-Nagumo model resembles that of $Na + K$ (compare figures 3.6 and 3.4). The important feature of this model is the interplay of cubic nullcline with the second nullcline (that is sigmoidal in $Na + K$ model, here approximated by a linear function). The magnitude of input influences the relative shift of cubic nullcline with respect to linear nullcline, changing the topology of phase space and causing various dynamical events.

The FitzHugh-Nagumo model was very successful in explaining many of dynamical phenomena of the Hudgkin-Huxley equation, like the absence of threshold, excitation block (the new stable equilibrium emerges at the right knee of the

cubic nullcline eliminating stable cycle responsible for spiking), post-inhibitory spikes (sudden stop of inhibition causes a spike, since the cubic nullcline jumps upwards, and the system suddenly falls into a limit cycle) and spike accommodation (slow increase in the input current shifts the cubic nullcline but the state remains in the stable equilibrium whereas sudden increase by the same magnitude causes instantaneous action potential) in terms of phase plane analysis (see Izhikevich (2006b)).

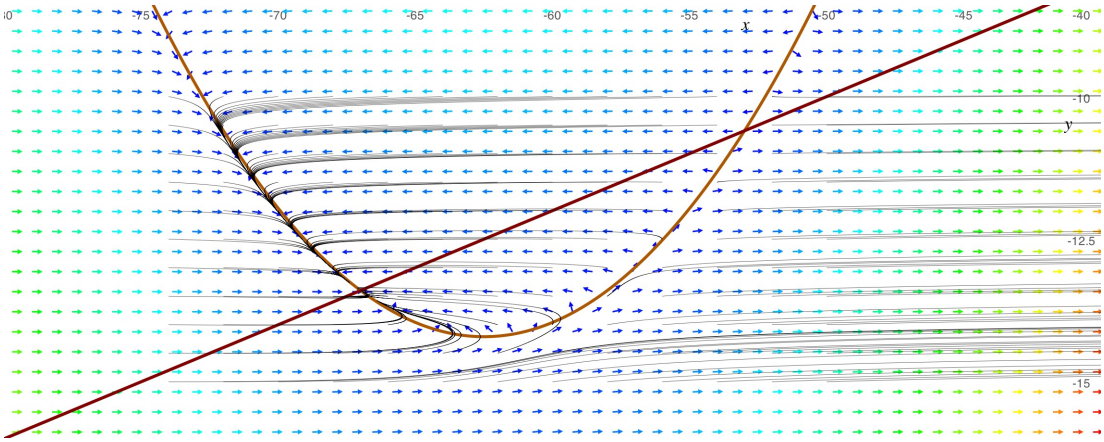

Figure 3.7: The phase plane for the simple spiking neuron model. Nullclines and a number of sample trajectories are plotted.

3.5.3 Simple model by Eugene M. Izhikevich

In 2003 Eugene M. Izhikevich went a step further and introduced (Izhikevich, 2003) simple model defined as follows:

$$\begin{aligned} \frac{dV}{dt} &= 0.04V^2 + 5V + 140 - u + I \\ \frac{du}{dt} &= a(bV - u) \end{aligned} \tag{3.13}$$

This model focuses near the point of intersection of nullclines. Here cubic nullcline is approximated by a quadratic function, second nullcline is linear (see figure 3.7). In general trajectories in this model are unbounded, so they are manually clipped, that is: $V := c; u := u + d$ whenever V exceeds $30mV$ (which marks the occurrence of a spike). This imitates the behavior of more complex systems near the steady state equilibrium, but the exact shape of an action potential is different than that of Hodgkin-Huxley model or experimental data. This model,

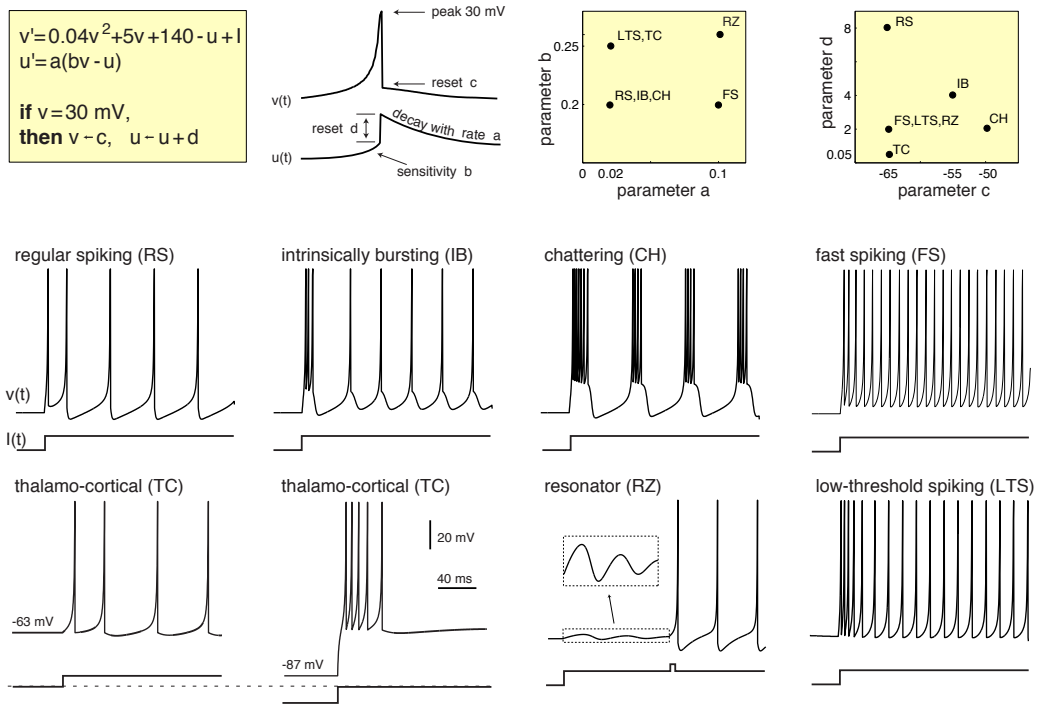

Figure 3.8: Diagram of the dynamical model (Izhikevich, 2003) and possible spiking regimes that the model is capable of reproducing. Note the subthreshold oscillations (resonator), which are missing in simpler integrate and fire models. Image courtesy of Eugene M. Izhikevich (`http://www.izhikevich.com`).

although it might seem rather artificial, is useful in modeling of large populations of neurons in a coarse-grained setup (although should not be used in more detailed studies of small neuronal circuits). Simple dynamics and fair numerical stability allows for large simulations of hundreds of thousand or even millions such neurons (including simulations of the whole brain (Izhikevich & Edelman, 2008)). By varying dimensionless parameters a, b, c, d a variety of behaviors observed in biological neurons can be mimicked (see figure 3.8). This model is nearly as simple as integrate and fire models, yet capable of producing oscillations (and for the same reason has to be simulated continuously, and cannot be run in event driven fashion). More on studying collective dynamics of neural networks in dynamical system setup can be found in (Hoppensteadt & Izhikevich, 1997) and (Izhikevich, 2006a).

Chapter 4

Spike flow graphs

In this chapter (presenting the material of Piękniewski & Schreiber (2008)) we propose a simple and tractable mathematical model for a situation in which a single processing unit is able to store some information related to its previous incoming and outgoing activity. Such a unit is an adequate description of activity of a neuronal group (as supported by empirical evidence of Eguíluz *et al.* (2005)) with heavy recurrent connectivity (recurrence is very important here as shown in further chapters). Any entity being able to store incoming signals (tokens) and forwarding them without high loss would also fit this framework. In particular it can also be interesting for economists as modeling the flow of capital in the society (which also has scale-free properties as noted by Vilfredo Pareto (1896-1897)) as well as many other phenomena.

The model consists of a number of neuron like units allowed to exchange charge under stochastic dynamics, which is modeled as neuronal spikes being transmitted along the edges of a fully connected network. Next, each edge is labelled with the count of spikes it transmitted, which results in a graph with weighed edges, called the *spike flow graph* in the sequel. Our theoretical results below, further confirmed by numeric evidence, state roughly speaking that if we remove those neural connections which are only relatively seldom used for spike transfers and we keep only those often used and relevant to the dynamics, the resulting graph is with overwhelming probability scale free with power law exponent $\gamma = 2$. The proof goes by showing that in low enough temperatures the large-scale behavior of the system admits an accurate description in terms of a particular winner-take-all type dynamics. Whereas the considered neural network model may be regarded to some extent simplistic, its asymptotic description in terms of a winner-take-all type dynamics and hence also the scale-free nature of the spike flow graph seem to be rather universal, as suggested by numeric evidence (Piękniewski, 2007).

We find the results of this chapter important and interesting as showing how scale-free structures spontaneously emerge in neural information processing, ar-

guably for rather general models and with no special assumptions aimed at stimulating this kind of self-organization. Apart from their theoretical value the results established in this chapter provide a further justification for considering neural architectures based on small-world and scale-free graphs, as has become popular in the literature in recent years, see (Perotti *et al.* , 2006; Stauffer *et al.* , 2003) and the references therein.

The remaining part of this chapter is organized as follows. In Section 4.1 below we introduce our basic theoretical model sharing certain features with the standard Boltzmann machines (Aarts & Korst, 1989), yet admitting a richer state space and assuming a rather different dynamics for individual neurons which are simple spiking units here. Next, in Section 4.2 we describe the behavior of this model in large system size and long evolution time limit and argue it can be represented via a kind of a winner-take-all dynamics whose particular features enable us to establish explicit results on the scale-free properties of the spike-flow graph in the following Section 4.3.

4.1 Basic model

In our research we sought a model whose dynamics would in its essence resemble that encountered in usual recurrent neural networks and, while being simple in terms of its statistical mechanics, would exhibit a scale-free structure as a natural consequence of its construction. These considerations resulted in the following *spike flow model* originally introduced in (Piękniewski & Schreiber, 2007). We consider a simple stochastic recurrent neural network consisting of N neurons assuming states labeled by natural numbers $\sigma_i \in \{0, 1, \ldots, M_i\}$, $i = 1, \ldots, N$, interpreted as *neuronal charges* below, and with natural or possibly infinite numbers M_i standing for maximum admissible values for the respective charges σ_i, $i = 1, \ldots, N$. The network is built on a complete graph in that there is a connection between each pair of neurons σ_i, σ_j, $i \neq j$, carrying a real-valued weight $w_{ij} \in \mathbb{R}$ satisfying the usual symmetry condition $w_{ij} = w_{ji}$, moreover $w_{ii} := 0$. The values of w_{ij} are drawn independently from the standard Gaussian distribution $\mathcal{N}(0, 1)$ and are assumed to remain fixed in the course of the network dynamics. A configuration $\bar{\sigma} = (\sigma_i)_{i \leq N}$ of the network is assigned its Hamiltonian given by

$$\mathcal{H}(\bar{\sigma}) := \frac{1}{2} \sum_{i \neq j} w_{ij} |\sigma_i - \sigma_j| \tag{4.1}$$

if $0 \leq \sigma_i \leq M_i$, $i = 1, \ldots, N$, and $\mathcal{H}(\bar{\sigma}) = +\infty$ otherwise. The dynamics of the network is defined as follows: at each step we randomly choose a pair of neurons (σ_i, σ_j), $i \neq j$, and denote by $\bar{\sigma}^*$ the network configuration resulting from the original configuration $\bar{\sigma}$ by decreasing σ_i by one and increasing σ_j by one, that is to

say by *letting a unit charge transfer from σ_i to σ_j*, whenever $\sigma_i > 0$ and $\sigma_j < M_j$. Next, if $\mathcal{H}(\bar{\sigma}^*) \leq \mathcal{H}(\bar{\sigma})$ we accept $\bar{\sigma}^*$ as the new configuration of the network whereas if $\mathcal{H}(\bar{\sigma}*) > \mathcal{H}(\bar{\sigma})$ we accept the new configuration $\bar{\sigma}^*$ with probability $\exp(-\beta[\mathcal{H}(\bar{\sigma}^*) - \mathcal{H}(\bar{\sigma})])$, $\beta > 0$, and reject it keeping the original configuration $\bar{\sigma}$ otherwise, with $\beta > 0$ standing for an extra parameter of the dynamics, in the sequel referred to as the inverse temperature conforming to the usual language of statistical mechanics and assumed fixed and large (low temperature) throughout. Observe that the sum $\sum_i \sigma_i$ of neuronal charges is preserved by the dynamics and that, in the course of dynamics with some initial configuration $\bar{\sigma}^0$, any other $\bar{\sigma}$ with $\sum_i \sigma_i^0 = \sum_i \sigma_i$ is eventually reached with positive (although possibly very small) probability. Consequently, upon standard verification of the usual detailed balance conditions, we readily see that the collection of stationary states of the above dynamics are precisely the distributions

$$\mathbb{P}_n(\bar{\sigma}) = \begin{cases} \frac{\exp(-\beta\mathcal{H}(\bar{\sigma}))}{\left(\sum_{\bar{\sigma}',\, \sum_i \sigma_i' = n} \exp(-\beta\mathcal{H}(\bar{\sigma}'))\right)}, & \text{if } \sum_i \sigma_i = n, \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

and their convex combinations. In particular, our model bears some resemblance to the usual stochastic Boltzmann machines (Aarts & Korst, 1989), with the weights w_{ij} indicating the extent to which the system favors the agreement (for positive w_{ij}) or disagreement (for negative w_{ij}) of the neuronal states σ_i and σ_j. There are evident differences though, one of them being the possibly unbounded state space whenever $M_i = \infty$, the other one that precisely two neurons are affected in each update with clearly determines the source and destination of the charge flow. Whereas the latter difference does not lead far away from the concept of a Boltzmann machine, as yielding a rather similar form of the stationary distribution, the former one is crucial – indeed, if M_i is a large number, the behavior of the corresponding i-th unit becomes quite complex and arguably it can be regarded as exhibiting some kind of *memory* of charge transfers undergone in the course of the dynamics. In this thesis we shall concentrate on the cases where M_i's are all infinite or of order only slightly smaller than the overall charge stored in the system, thus conforming to our leading assumption of a complex neuronal structure. We will prove below that in this set-up the induced network exhibits natural scale-free features. On the other extreme one can impose all M_i's very small, which makes our model resemble classical Bolztmann machines. Taking all $M_i \equiv 1$ and i.i.d. Gaussian weights yields a network which can be regarded as a somewhat modified version of the well-known Sherrington-Kirkpatrick spin glass model, see Chapter 2 in (Talagrand, 2003). In general, our model interpolates between both extremes and can exhibit a wide range of behaviors depending on the choice of the M_i's. For the network dynamics running during a period $[0, T]$ we are now in a position to define the *spike flow graph* to be a directed graph with

vertices corresponding to the neurons σ_i, $i = 1, \ldots, N$ and whose edges carry numbers (edge multiplicities) $F_{i\to j}$ indicating how many times in the course of the dynamics the charge flow occurred from σ_i to σ_j. If β is large, which is always going to be assumed in this chapter, after a long enough simulation run the system freezes in some ground state whereupon any further charge flow becomes very rare and consequently the numbers $F_{i\to j}$ also freeze undergoing virtually no further changes. The in-degree of a neuron σ_i is now defined as $d_{\text{in}}(j) := \sum_i F_{i\to j}$. The main question considered in this chapter is whether the so-defined spike flow graph is scale-free in that its in- degree distribution follows a power law, that is to say $\mathbb{P}(d_{\text{in}}(i) \approx x) \sim c_{\text{in}} x^{-\gamma_{\text{in}}}$ for a randomly picked node i. Choosing M_i's large enough we shall establish a positive answer to this question. It should be noted at this point, as will become clear from our discussion below, that the asymptotic behavior of the corresponding out-degree distribution is the same as that of the in-degrees.

4.2 Winner-take-all dynamics and ground states

For the scope of this research we will limit ourselves to very low temperature regime, which amounts to assuming that the overwhelming majority of network updates are just jumps towards lower energy configurations, as in the corresponding zero-temperature (infinite β) approximation.

We assume first that all M_i's are infinite and thus no upper bounds are imposed on individual charges. In this extreme set-up we argue that with overwhelming probability with respect to the choice of the weights w_{ij}, the unique ground state (lowest energy state) of the network, and hence also the unique attractor of its dynamics, is a configuration in which all charge present in the system is stored in a single *best* unit with all the remaining units devoid of charge. To see this, for each unit σ_i consider the *support* S_i it gets from the remaining units, given by

$$S_i := -\sum_{j\neq i} w_{ij}.$$

Clearly, all the S_i's so defined are Gaussian random variables $\mathcal{N}(0, N-1)$ and are virtually independent – indeed, S_i and S_j for $i \neq j$ share just one summand w_{ij} whereas the remaining ones are independent. With $S_{:k}$ standing for the k-th largest value among S_i's, it is known by extreme value theory, see e.g. Section 1.2 in (Talagrand, 2003), that the order statistics $S_{:k}$ are well approximated by

$$S_{:k} \approx \sqrt{2N \log_2 N}\left(\sqrt{\log 2} + \frac{\xi_k}{\sqrt{\log_2 N}}\right) \tag{4.3}$$

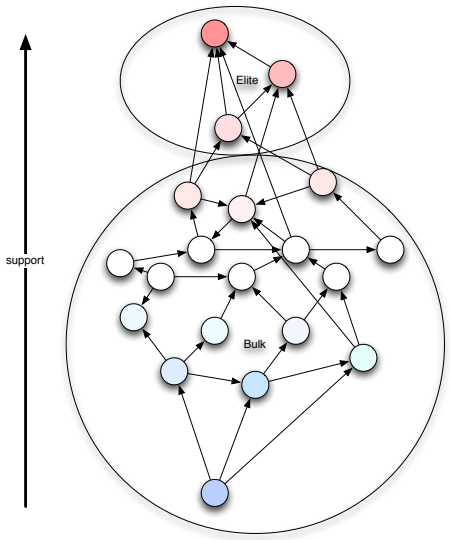

Figure 4.1: Schematic presentation of the spike flow model

where the sequence $\xi_1 > \xi_2 > \dots$ is chosen according to a Poisson point process with intensity $\frac{1}{\pi}\exp(-2t\sqrt{\log 2})$, $t \in \mathbb{R}$, in particular the p-th ξ_i above 0 is of order $\log p$ and $S_{:k}$'s are of order $\sqrt{N \log N}$ which is much higher than the order of the typical S_i being $\sqrt{N}$. To proceed, assume we run our spike-flow dynamics for some long enough amount of time to get close to equilibrium, whereupon we consider a small number $o(N)$ of neurons which store the highest charge, considerably higher than the remaining units, and we call these *elite neurons* while granting the term *bulk neurons* to the remaining units in the system. Since the number of elite neurons is a negligible fraction of N, the formula (4.1) becomes then

$$\mathcal{H}(\bar{\sigma}) \approx -\sum_{i \in \text{elite}} \sigma_i S_i + \frac{1}{2} \sum_{j,l \in \text{bulk}} w_{jl} |\sigma_j - \sigma_l|. \tag{4.4}$$

Thus, whenever in the course of the network dynamics a charge transfer is proposed from a bulk neuron σ_j to an elite neuron σ_i, the resulting energy change is seen to be well approximated by $-S_i$ plus a term due to the interaction between σ_j and other bulk neurons. In general, we have no control of this term, yet if σ_i is one of the neurons with the highest support as in (4.3), this offending term of order at most $\sqrt{N}$ is very likely to be negligible compared to $-S_i$ which is of order $\sqrt{N \log N}$, thus making the energy change strongly negative and the proposed transfer extremely likely to be accepted. Clearly, the inverse transfer becomes then almost impossible. Consequently, whenever a neuron with a very high support enters the elite, it virtually never leaves it; moreover it continuously drains charge from the bulk losing it only to other elite members if at all (see

Figure 4.2 for numerical support of these claims). Furthermore, should a neuron with a small support value happen to enter the elite at the early stages of the dynamics, it will soon leave it having its charge drained by other higher supported neurons. Thus, after running our dynamics long enough we end up with a picture where the elite consists of neurons with the highest support. Although the elite neurons do struggle for charge between themselves, they cooperate in draining it from the bulk. Therefore eventually almost no charge will be present in the bulk and hence the Hamiltonian will admit a particularly simple approximation

$$\mathcal{H}(\bar{\sigma}) \approx -\sum_{i\in\text{elite}} \sigma_i S_i \tag{4.5}$$

and all further updates in the system will only happen due to charge transfers within the elite. Note now that the interactions between elite neurons as determined by their connectivities (weights) are of order $o(\sqrt{N})$ since the cardinality of the elite is $o(N)$ whereas the differences between the highest consecutive support values are of higher order $\Theta(\sqrt{N})$ in view of (4.3) which makes the former negligible compared to the latter. Thus, the dynamics between the elite neurons takes eventually a particularly simple form: a pair of elite neurons is chosen by random and if the one with smaller support attempts to transfer a unit charge to the one with higher support, the attempt is accepted, otherwise it is rejected. The only ground state of the system is then obtained by putting all charge into the unit of the highest support. It should be noted that at intermediate stages of the dynamics it may happen that elite members show up with charges whose order is inverse to that determined by the supports rather than consistent with it. This is an artifact due to the fact that if we admitted negative charges here, a twofold sign-flip symmetry would be present in the system in full analogy to usual networks with no external field and such inverse ordering would compete with the standard one on equal rights. This is not the case here though because negative charges are not allowed and therefore such inversely ordered structures are unstable and do not persist in the course of the dynamics.

In view of the above discussion, the highest in-degrees of the spike-flow graph are observed in elite units enjoying the highest support from the system, and the corresponding charge flows $F_{i\to j}$ are mainly due to the internal charge transfers within the elite. Thus, we have shown that the asymptotic behavior of our network model is accurately described by the following *winner-take-all* model:

- the system consists of K neurons $u_i,\ i = 1,\dots,K$, representing the elite units and ordered according to decreasing supports,

- n units of charge are sequentially introduced into the system, each time according to the following dynamics

  - first, a unit charge is transferred to a randomly chosen neuron u_{k_0}, $k_0 \leq K$,
  - thereupon it starts jumping to further neurons u_{k_l}, where $k_{l+1} < k_l$ is randomly chosen in $\{1, \ldots, k_l - 1\}$,
  - eventually the unit charge reaches u_1 and gets frozen there.

- the in-degrees of the elite neurons in the original network are approximated by the numbers D_i indicating how many charge units have visited u_i on their way to u_1.

In other words, in this model the charge transfers always occur from a neuron with smaller support to a randomly chosen better supported one, whence the term *winner-take-all* dynamics. Curiously enough, the winner-take-model is easily seen to exhibit a *consistency property* – if we take some $K' < K$ and observe the behavior of the model restricted to K' neurons of highest support only, this exactly coincides with what we get if we run our original dynamics on the restricted set $\{u_1, \ldots, u_{K'}\}$ of units. Consequently, from the viewpoint of our asymptotic analysis of the in-degree sequence $D_1, D_2, \ldots$ the precise value of K is irrelevant as long as $K \ll N$ but $K \to \infty$ as $N \to \infty$.

Now, repeating the argument presented in this section for M_i's large but finite, we end up with the following modification of the above winner-take-all dynamics. Assume first that the elite neurons of the highest support are not yet saturated, that is to say their capacity has not yet been reached. In such a case the dynamics follows exactly as previously. Once a certain elite neuron gets saturated, it becomes *inactive*, since it cannot accept any more incoming charge. If this happens to be a neuron of high support, then it is very unlikely to get unblocked prior than possibly at the very final stages of the dynamics, since with overwhelming probability only units of higher support drain any charge from the considered unit, and their number is negligible compared to that of lower supported neurons pumping their charges upwards the support hierarchy. Therefore, should a neuron of a very high support get saturated, it will most likely stay inactive for the most of the simulation thereafter, and it could be removed from any additional consideration as playing no relevant role anymore. Further evolution of the so reduced system follows the same pattern: at any stage the winner-take-all dynamics is present among the set of best unsaturated neurons. Consequently, in case M_i's are of the same order as total charge present in the system, the deviation from the *unbounded* version of the dynamics is negligible, which can be easily noted in simulations. If M_i's are much smaller though, the saturation factor becomes significant and the *winner-take-all* dynamics breaks down. Some models exhibiting this property will also be discussed in the next Section 4.3.

4.3 Power law for spike-flow in-degrees

We are now in a position to exactly characterize the asymptotic behavior of the in-degree sequence $D_i,\ i \geq 1$. Again, we begin with the extreme set-up $M_i \equiv \infty$ first, passing to more general choice of charge constraints thereafter. It is worth noting that asymptotically the out-degree sequence behaves in exactly the same way as the in-degrees since for most units save the highest support neuron and very low support neurons their in- and out-degrees are almost equal. Some insignificant disagreements may occur in finite numerical simulations, where all units start with some fixed amount of charge and proceed according the dynamics. In such case the out-degree sequence is disturbed by the single (therefore insignificant) unit that eventually receives and keeps the whole charge present in the system, whereas the in-degree sequence is disturbed within some range of low degrees (units which received far less charge than they gave away to others). We can avoid these fluctuations by only looking at the tail of the distribution (in practice, say, degrees higher than 5-10 times the initial charge per neuron) or by simulating larger systems.

To proceed, consider a single charge unit introduced into the system and denote k_l for the number of neuron u_{k_l} it visits after its l-th jump, $l = 0, 1, \ldots$. Recall from Section 4.2 that k_0 is drawn uniformly from $\{1, \ldots, K\}$. Further, consider also a sequence $X_0, X_1, X_2, \ldots$ of continuous $(0, 1)$-valued random variables such that X_0 is uniform in $(0, 1)$ and X_{l+1} is chosen uniformly from $(0, X_l)$ for all $l \geq 0$. Then it is easily seen that for K large enough we can safely approximate in law

$$k_l = \lceil K X_l \rceil$$

with $\lceil \cdot \rceil$ standing for the upper integer value of its argument. In particular, defining $\pi_i,\ i = 1, \ldots$ to be the probability that the charge unit visits u_i, we have $\pi_i = \mathbb{P}(\exists_l k_l = i)$ and hence for K large enough we get the approximation

$$\pi_i \approx \mathbb{E}|\{l,\ X_l \in [(i-1)/K, i/K]\}|,\ \ i > 1 \tag{4.6}$$

and, clearly, $\pi_1 = 1$. The values of in-degrees D_i are then binomially distributed $b(\pi_i, n)$ with parameters π_i and n, the latter standing for the number of charge units present in the system.

To proceed with our asymptotic analysis we observe that X_l's form a so-called *record sequence* in the sense of classical extreme-value theory, see Chapter 4 in (Resnick, 1987). Consequently, by Section 4.1 ibidem, the sequence $T_l := -\log X_l$ is simply a unit intensity homogeneous Poisson point process in $\mathbb{R}_+$. Thus, using (4.6) we get

$$\pi_i \approx \mathbb{E}|\{l,\ T_l \in [-\log(i/K), -\log((i-1)/K)]\}| \approx 1/i.$$

Hence, for large values of n we have by the law of large numbers

$$D_i \approx n/i.$$

It means that, for large k,

$$|\{i,\ D_i > k\}| \approx n/k$$

or,

$$|\{i,\ D_i \approx k\}| \approx n/k^2.$$

We have thus proven the main result of this thesis.

Theorem 1 *For the basic spike-flow model with $M_i \equiv \infty$ the resulting spike-flow graph is scale-free with exponent $\gamma = 2$.*

In analogy to this argument assume now that M_i's instead of being infinite are finite, indepedent from the weights w_{ij}, independent among themselves, and drawn from a power-law distribution

$$\mathbb{P}(M_i > k) \approx ck^{-\alpha} \tag{4.7}$$

for some $\alpha > 0$. In such a set-up, if c in (4.7) is not very large, a non-negligible fraction of units will get saturated in the course of the dynamics and therefore would stop accepting any more incoming charge at some stage of the network evolution. This may considerably alter the behavior characterized by Theorem 1. In fact, it is natural to expect that three groups of units will emerge:

- Units of highest support, *elite of the elite*, which can be sure to reach their capacities. By the independence of M_i's from the weights w_{ij}'s and hence also from the supports S_i's, when choosing at random among such units the probability of exceeding in-degree k is given by the product of the probability of the chosen unit exceeding the in-degree k in the unconstrained dynamics ($M_i \equiv +\infty$) times the probability of its capacity being higher than k. Consequently, in view of Theorem 1 the in-degrees of the highest support units should follow a power law with exponent $\alpha + \gamma = \alpha + 2$.

- Units of intermediate supports, *lower elite*, still falling into the elite and reaching rather high in-degrees, but not exceeding or even reaching their capacities. Such units do not feel the constraints M_i's and constitute a portion of the network where in-degrees should follow a power law with exponent $\gamma = 2$ as in the unconstrained dynamics.

- Units of rather high but not highest supports, *medium elite*, for which the capacity and in-degrees they would reach under the unconstrained dynamics are of a comparable order. Their behavior should interpolate between the above two extremes.

These observations would suggest that two principal regimes should be observable for the in- degree distribution of such networks: highest in-degrees should follow a power law with exponent $\alpha + 2$ whereas the lower elite in-degrees should behave as in Theorem 1 stating power law with exponent 2. The region separating these regimes should interpolate between these two behaviors, possibly exhibiting very complicated properties due to the presence to *traffic jams* at medium elite units, which are no more negligible unlike in case of lower elite units, but which only temporarily disable the blocked units and may be eventually discharged in contrast to the case of high elite units. We cannot claim to have confirmed these conjectures by numerical results though because the realistic system sizes we were able to reach in our simulation were too small to ensure statistically significant collection of units in each of the afore-mentioned regimes.

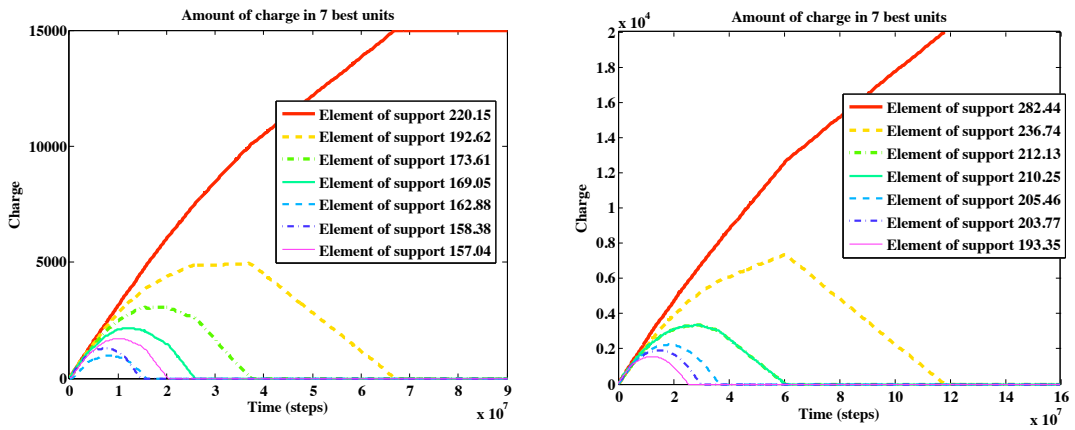

Figure 4.2: Typical evolution of the charge stored in seven units of the highest support. The above figure is a result of a simulation run of 3000 units (left) and 4000 units (right). Note that while in the beginning of the simulation all seven elite members compete for charge, by the end the single best unit gets everything.

4.4 Numeric results

The above considerations were accompanied by a numerical simulation implemented in Matlab, letting us continuously verify our assumptions, and giving valuable hints for further investigation. The simulations were usually carried out for systems of about 3000-6000 neurons with the basic dynamics (as described in Section 4.1) – the only speed-up was that the neuron to pass a unit charge to some other one was chosen randomly only among those containing any charge at all. The total energy computation required a quadratic time with regard to the

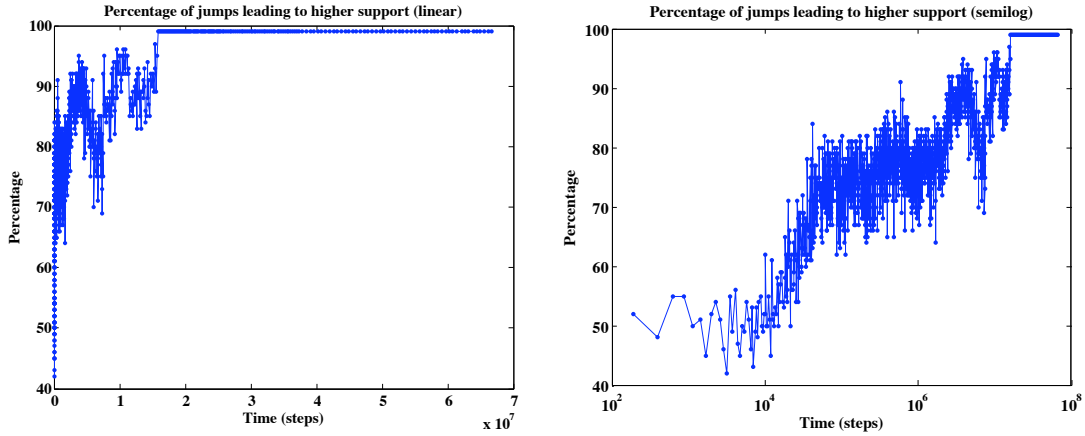

Figure 4.3: Percentage of charge jumps leading to a unit of higher support (sampled every 100 jumps) in a simulation run of 3000 units. The plot on the left is scaled linearly, the one on the right is semi-log. Note that after initial unstable phase (about 10^4 steps), the plot increases steadily until about $2 * 10^6$ steps where again some fluctuations occur. These fluctuations are caused by increased significance of stochastic term (tiny energy modifications leaving room for thermal fluctuations). It is worth noting that by that time jumps are already infrequent while the state is near the energy optimum. By step $2 * 10^7$ the system freezes completely in the ground state.

number of neurons, but during the simulation we only needed to compute local energy updates, which took only linear time. Despite of these straightforward enhancements, larger systems (≈ 10000 units) became problematic due to memory consumption and did not give any qualitatively better results. In the future we plan to simulate much bigger systems based on the simplified version of dynamics (the winner-take-all asymptotic version) that would allow us to avoid the need for explicit connectivity matrix, in order to confirm the intuitions described in final paragraphs of Section 4.3. In the course of the present simulation β (the inverse temperature) was fixed at $\beta = 10$ which, since the average energy updates in the simulation were of order ≈ 1 per step, places us in the low temperature regime. The temperature only became more significant by the end of the simulations when the energy modifications were of much smaller order leaving place for thermal fluctuations, but by that time the system usually had already converged to the expected "winner-take-all" configuration (Figure 4.3 gives some insight into temperature based fluctuations). The results of the simulations confirmed our theoretical predictions about the "winner-take-all" dynamics (see Figures 4.2,4.3,4.5), as well as the scale-free properties of the *spike flow graph* (Figure 4.4). The number of steps was 10 times the number of neurons squared, which

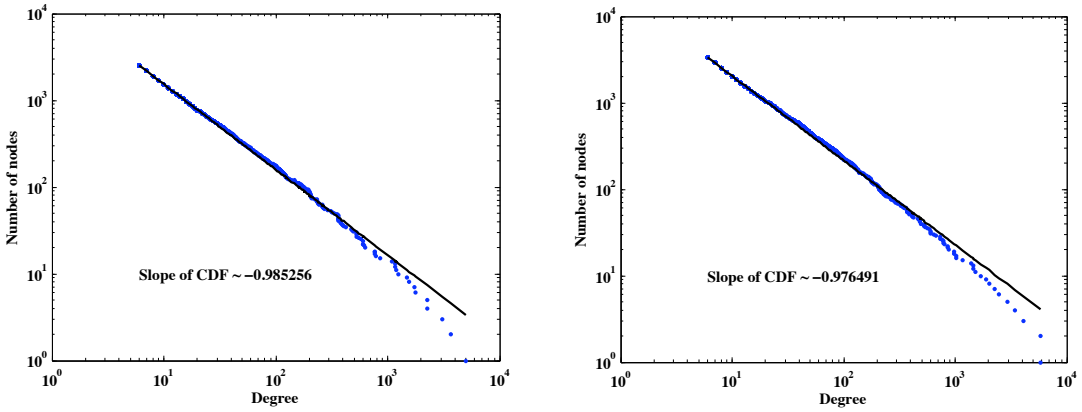

Figure 4.4: Cumulative distribution function (CDF) of the out-degree in the spike flow graph (in-degree yields a similar plot) of 3000 units (left) and 4000 units (right). Presented CDF slopes correspond to the power law exponent $\gamma = 2 + / - 0.03$. The slopes were approximated by the least squares method.

was about the number of steps required for full convergence to the ground state. Rarely the system converged to a state in which two units of highest support shared the whole charge. This is possible, whenever the weight between the two competing units is comparable to the difference of their supports, thus forming a local energy minimum (pumping charge to the better unit requires temporary energy increase). Since the experimental system is finite, such unusual configurations may appear with some small probability. Evidently as the system size increases, such energy minima become less probable (asymptotically negligible).

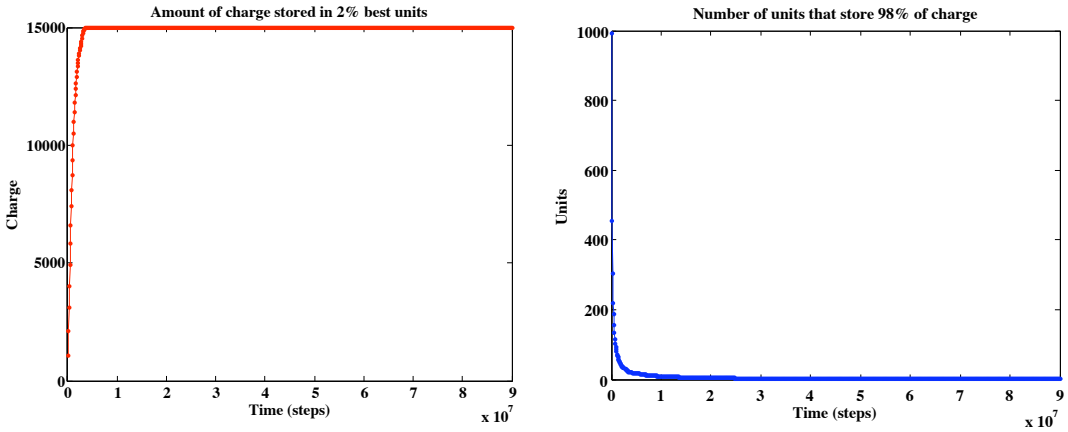

Figure 4.5: The charge stored in 2% units of highest support (left) and number of units storing 98% of total charge. These figures give strong support to the idea of dividing the units into elite and bulk, and treating these groups separately.

Chapter 5

Scale-freeness of dynamical spiking networks

5.1 Introduction

In this chapter we present the material of (Piękniewski, 2007), a numerical study aimed at finding scale-free connectivity in systems based on dynamical spiking neural networks. As mentioned in previous chapters, the simplest property leading to scale-freeness of a network is preferential attachment combined with model growth (see Albert & Barabási (2002)). Preferential attachment imposes that units already well connected should have a higher probability of being attached to other nodes. In terms of neurons and their spike flow graphs this property might be translated as follows: *the more activity a unit receives, the more active it becomes and retains this activity for some period of time, depending on the initial excitation.* It is essential in this formulation that a unit possesses a non trivial memory of its state (amount of activity already received, and therefore ability to become active). Although single neurons (even single compartment dynamical models) do possess some amount of state memory (stored in current vector of parameters in the phase space), this is not enough to clearly exhibit "preferential attachment" in the sense defined above. Single neurons act more like wires, though the excitation mechanism is active, they do not have any significant capacity to store incoming signals. In the following sections we argue that things change if we move from single neurons into neuronal groups of certain kind. Such groups have opportunity to stay active for a longer period of time, and have some ability to *store* received activity by continuous excitation (of a number of units within the group) and therefore have some sort of memory of their past excitation. This property (although not easy to clearly distinguish from other dynamical behaviors of such a group), should lead to a certain kind

of synchronization that would result in a scale-free network of synchronizations between the groups. These informal statements will be specified more precisely in following subsections. Theoretically speaking, for infinite simulation runs the model growth is required, because otherwise the network would saturate with spike flow and lose scale-free property. With the models discussed in this chapter though, the oversaturation can be to a large extent neglected due to relatively short time of simulation, therefore our model is static, although it is also worth noting that in biological reality there are processes of decay and growth that prevent neural networks from saturation. In the previous chapter we examined a model of a *spike flow graph*, with simple units whose states were in $\mathbb{N}$. We have shown that under appropriate assumptions the neuronal dynamics of that model imposes scale-free structure on the induced graph of potential transfers. Encouraged by this result we asked a question - is this behavior general and can it be reproduced with dynamical spiking neural models? For this investigation we have chosen the simple spiking model introduced by Eugene M. Izhikevich in (Izhikevich, 2003) (see section 3.5.3). We selected this model due to computational simplicity which lets one carry out simulations with a large number of these fairly complex (in terms of possible dynamical behaviors) spiking neurons reasonably fast, and therefore it fits perfectly the requirements of the presented research assignment.

One of the first objectives of this study was to confirm that single neurons do not posses enough memory[1] to exhibit a scale-free synchronization graph (informally, the weight of an edge in synchronization graph[2] is high if the spike trains of two units are similar and low in the other case, this concept is explained more formally in next section). We carried out a number of simulations with different weight matrices, and did not obtain any graph that would exhibit a scale-free nature (figure 5.4), either before or after thresholding[3]. It seems that such simple systems either synchronize too well, or don't synchronize at all, whereas scale-free property requires something in between.

Since, as expected, single neurons are not complex enough, our second step was to construct a model of neuronal groups. This approach is not very far from real life, since it is well known that neurons form well connected groups, and such phenomena were observed before even in the model we use in the present chapter (see Izhikevich *et al.* (2004) for example). For simplicity we have constructed the groups randomly, with respect to some basic properties like distinction between excitatory and inhibitory neurons etc. The connectivity within a single group (about 10 to 20 neurons) was quite similar to that presented in (Izhikevich, 2003),

[1] In the sense discussed above.

[2] This notion is similar to the functional network studied empirically by Eguíluz *et al.* (2005)

[3] The obtained graph is weighted, thresholding is a way of creating corresponding unweighted graph.

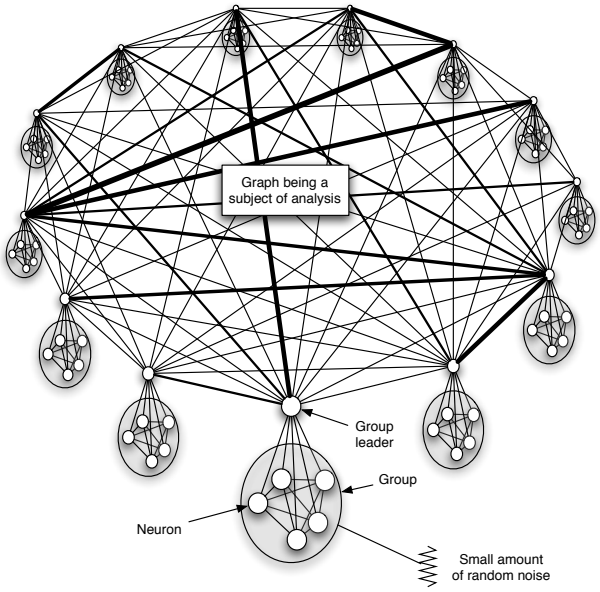

Figure 5.1: A schematic presentation of the model investigated in this chapter. The model consists of a number (about 1000-3000) of neuronal groups, connected randomly (weights chosen from Gaussian distribution $\mathcal{N}(0,1)$) by the group leaders - neurons chosen to interconnect every group with others. The group's synchronization depends on the input received from the group leader and, on the other hand, the activity of the leader reflects the activity of the group.

with appropriately scaled weights to ensure sustained activity within a group. In every group, one neuron was chosen to play a special role (in the sequel we will refer to it as the *group leader*). This special neuron connects the group with other groups, it forms a kind of a gateway between the group and the rest of the model. Group leaders were connected randomly with normally distributed weights (see figure 5.1 for conceptual schema). The simulation was carried out with about 3000 groups (that gives about 45000 neurons) for more than 10000 steps (the coefficients in the model are tuned, so that each step corresponds to approximately 1 ms in real time, this however is not a key issue, since the presented model does not resemble any particular biological network). The output seemed quite promising right from the beginning - figures 5.2 and 5.3 reveal rich neuronal behavior with global synchronization episodes, some of the units were exhibiting bursting activity. The striking feature of this plot is its self-similarity - in some ways it looks like a fractal. This detail, although interesting, was not a part of this research project - the goal was to translate synchronizations into real numbers, use these numbers as graph weights, and determine whether this graph has a scale-free property or not. The details are yet to be described in further

sections, however this section can be concluded by giving a positive answer to the preceding question - figure 5.5 strongly supports the claim that the degree distribution of a graph received from this numerical experiment follows a power law (similar plot was obtained in a number of simulations).

5.2 Model details

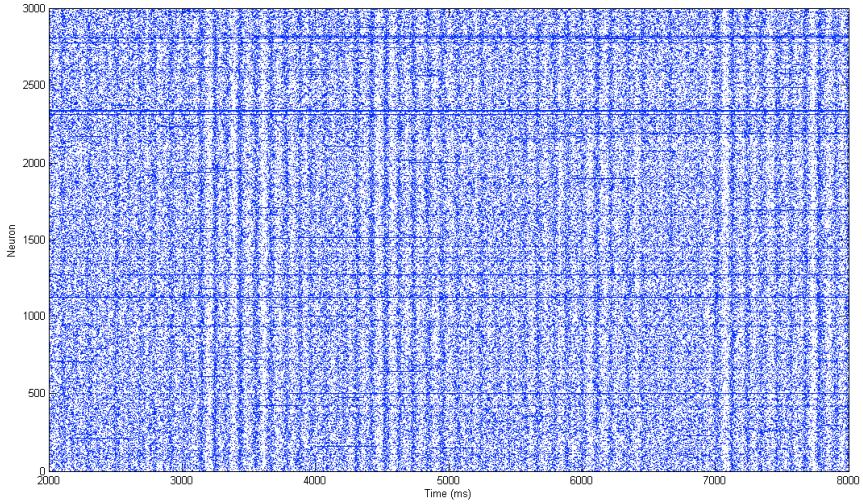

Figure 5.2: Example of spike activity plot of 3000 group leaders during 6000ms timeframe (only the activity of group leaders is plotted). Note the global synchrony episodes as well as bursting of single units.

As mentioned in the preceding section, the simulation was carried out for a set of 3000 groups (each consisting of 10-20 neurons - the number was chosen randomly with uniform distribution), represented by the group leaders. The connectivity within a single group was quite similar to the one from (Izhikevich, 2003), with appropriately scaled weights, to ensure synchronization. The ratio of excitatory/inhibitory neurons was also chosen randomly from uniform distribution. Note that this construction was not based on any particular biological inspiration since the goal of this research was rather to find a link between the discrete model discussed in previous chapter and more complex continuos dynamical one, than mimicking the biological complexity. The next steps of this project would be to create more and more biologically feasible models that would still exhibit the scale-free property, to this end however it is essential to know what dynamical features of these models are responsible for emergence the scale-free phenomenon (see next chapter). The simulation was carried out on two levels, on both of them synchronously:

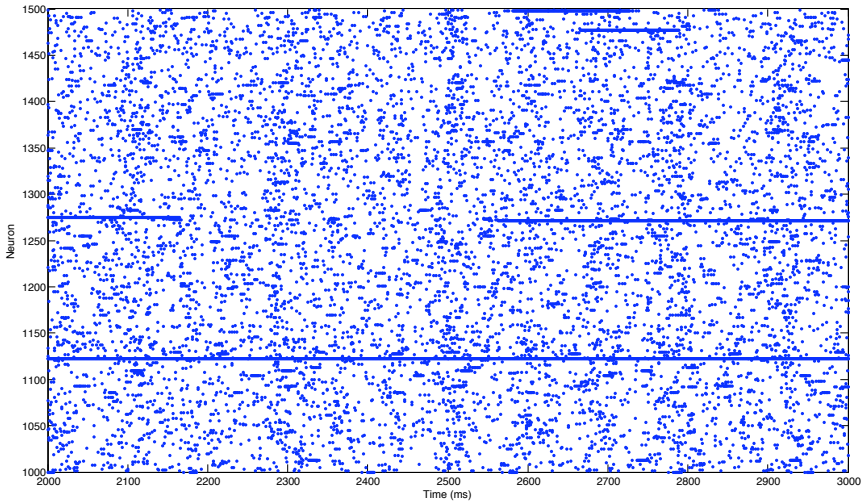

Figure 5.3: Magnified segment of figure 5.2 showing neurons 1000 to 1500 within 2000ms-3000ms timeframe. Please note the similarities of these plots (actually it is a self-similarity). The number and length of straight horizontal lines (each symbolizing a bursting activity) in both plots is approximately the same.

1. Initialization phase - each group was simulated synchronously over one time step (1ms). The initial input to every group was 0 plus some slight Gaussian noise (applied to every neuron independently) that simulated external excitation.

2. After this phase, weighted summation of group leader output activities is performed and given as input activity to group leaders in the next step.

3. Each group was simulated synchronously over one time step, with the group leader activity and a slight Gaussian noise as an input for every neuron.

4. Steps 2 and 3 were repeated until the end of simulation (in this case up to 12000 steps).

As the output, the simulation produced a significant number of spike trains (3000 neurons, each over more than 10000 time steps) that had to be compared with respect to a measure of synchronization computed in the following manner:

1. Each spike train was blurred by a convolution with $\exp\left(-\left(\frac{x}{10}\right)^2\right)$ kernel, see figure 5.8.

2. The transformed spike train of every two neurons was then multiplied and integrated. The integral (real number) was interpreted as a measure of synchronization.

The blur was necessary, to assure similarity between two spike trains that were in fact roughly similar, but corresponding spikes were shifted by several time steps in either direction (in this model we did not include any axonal delays). It is worth noting that this measure strongly supports bursting - two units giving continuous spike response in the same time gain much similarity in the sense above. Note that this measure is significant only if spikes actually do occur, two empty spike trains are similar in some sense, but in terms of a proposed measure their similarity is zero.

Based on the similarity measure above a symmetric weight matrix corresponding to a weighted graph was obtained. The resulting graph was subject to further analysis, based on typical tools from random graph theory (section 2.5) like degree distribution, average path, number of connected components and clustering coefficient.

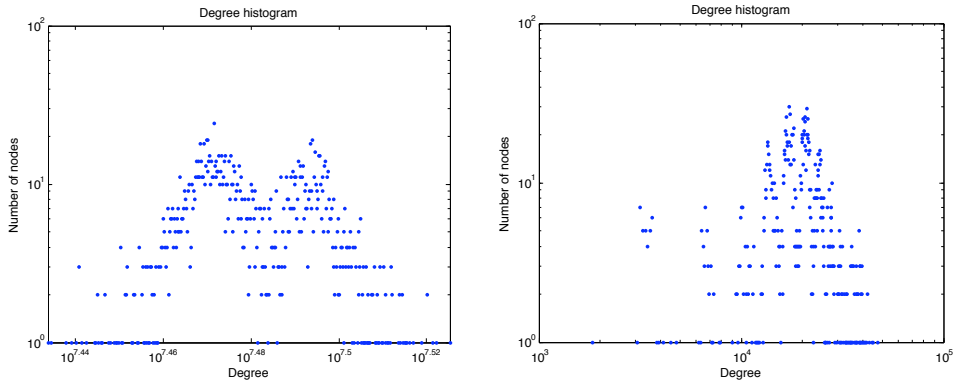

Figure 5.4: Example degree distributions of synchronization graphs produced by a network of spiking neurons for different coupling regimes. Certainly these graphs do not exhibit a scale-free property.

5.3 Results

The obtained graph was weighted which had its advantages and disadvantages. We used some analytical tools on the weighted graph, and then continued with an unweighted one created by thresholding the original graph over a certain value (in this case the average weight in the graph). The essential feature in the focus of this chapter – the scale-free property – was observed in either case. For the weighted graph the node degree was defined simply as the sum of weights of edges adjacent to that node. For the unweighted graph we used the usual definition (number of edges adjacent to the node). In both graphs the degree distribution followed a

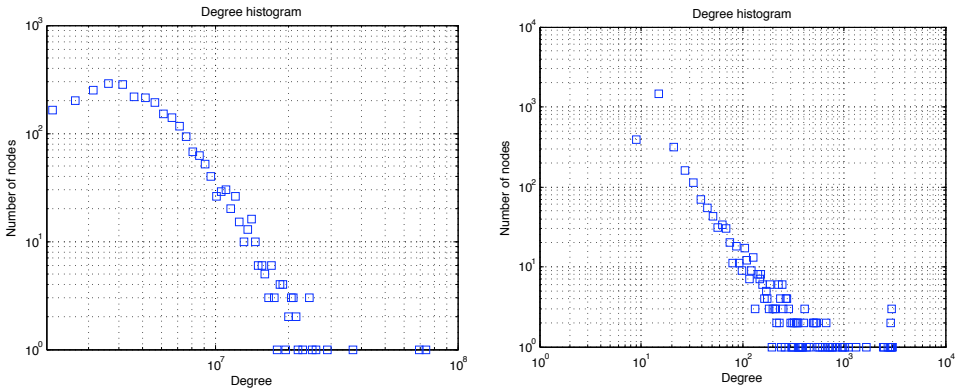

Figure 5.5: The degree distributions of a network received from the model (left), and one obtained by thresholding the original network at an average weight (right) in order to produce an unweighted graph. In either case the presence of a power law with exponent of about 2 is quite clear. Note the significant difference between these plots and the ones showed in figure 5.4.

power law, with exponent of about 2, in the weighted case there were some slight deviations - the power law was slightly violated near the plot limits. This however is not very surprising, since such disturbances are present in a number of other scale free networks, especially of medium size graphs like the one investigated in this chapter. These artifacts appear because of under/over saturation of high/low degree nodes, due to finite time of simulation. In the unweighted case these fluctuations are even less significant possibly caused by thresholding. In order to gain confidence (and avoid possible statistical disturbance), the simulation was repeated a number of times (about 30 for each set of parameters), the results were always similar.

The power law exponent is roughly 2 in agreement with the results of the previous chapter. To provide a better estimate, far larger graphs should be simulated, which does not seem particularly reasonable since this particular model is rather a proof of concept than a biologically plausible simulation of practical significance.

Note that as expected - more active units (the activity is measured as an integral of spike train convolved with $\exp\left(-\left(\frac{x}{10}\right)^2\right)$ kernel, as previously) gain more neighbors in the output graph (either weighted or unweighted). This is clearly visible in figure 5.6 and evidently supports our hypothesis of the presence in the model of a form of preferential attachment principle as discussed earlier.

The unweighted graph formed a singlegiant component[4] (possibly with some

[4]This obviously depends on the thresholding level. As the threshold grew, more and more

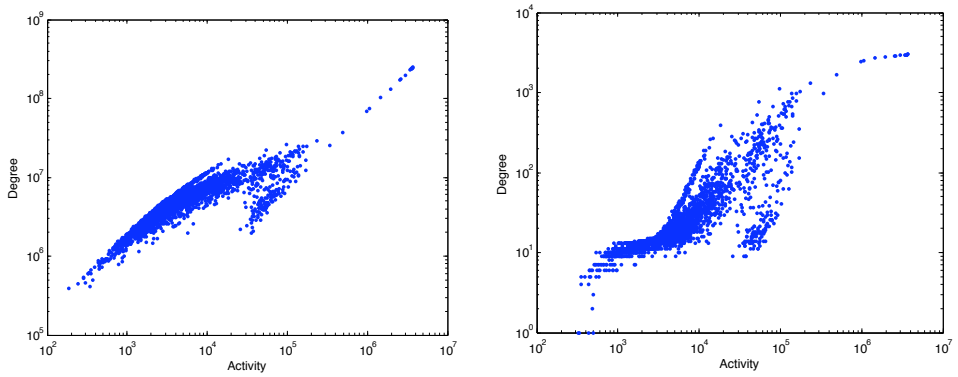

Figure 5.6: The dependency between spiking activity and vertex degree before (left) and after thresholding the graph (right). In either case we observe a clearly monotone dependency which supports our preferential attachment hypothesis.

number of isolated nodes).

The interesting feature of the thresholded graph is its very high clustering coefficient - this graph is very well clustered. What's more, the clustering coefficient exhibits a surprisingly regular dependence on degree (see figure 5.7, which displays this dependency as well as the corresponding dependency obtained from Erdős-Rényi random graph with similar connection density), which suggests that lower degree nodes are nearly fully clustered but after reaching a certain degree threshold (in the case of presented simulation the threshold is of about 10^2, but this value most probably depends on the system size), the clustering coefficient drops dramatically leaving high degree nodes almost unclustered. This gives an interesting insight into graph structure, but it is not yet obvious whether this dependency is an artifact of graph thresholding or is it some general property of these networks[5].

It is worth noting that the resulting graph also had a small world property - high clustering coefficient as discussed above (about a magnitude higher than in corresponding Erdős-Rényi random graph) combined with short average path length (depending on the thresholding level, the average path varied from about 2 to 4 nodes, so the connectivity is very good). Again this might not be very surprising in the context of already published results (Kwok *et al.* , 2006) and the fact that scale-free networks exhibit short average paths quite naturally.

nodes became isolated, but still a single giant connected component was present.

[5]It seems reasonable though, from sociological point of view, that people having only a few friends have higher probability that these friends also know each other, than people with a huge number of friends (hubs) possibly distributed over a large area. These considerations are obviously informal.

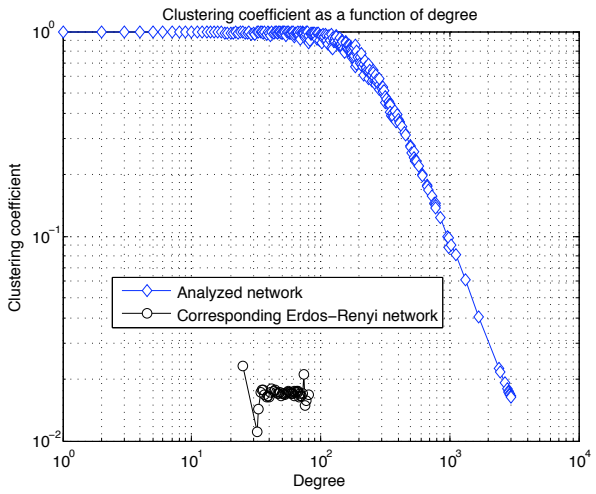

Figure 5.7: Clustering coefficient as a function of degree in the investigated network. This plot is interesting, since such behavior of clustering coefficient is rather rare - the nodes of small degrees are well clustered whereas those with high degree are not. Compare with box 2 in (Barabási & Oltvai, 2004). The black-circle plot depicts this dependency obtained from Erdős-Rényi random graph with similar number of edges.

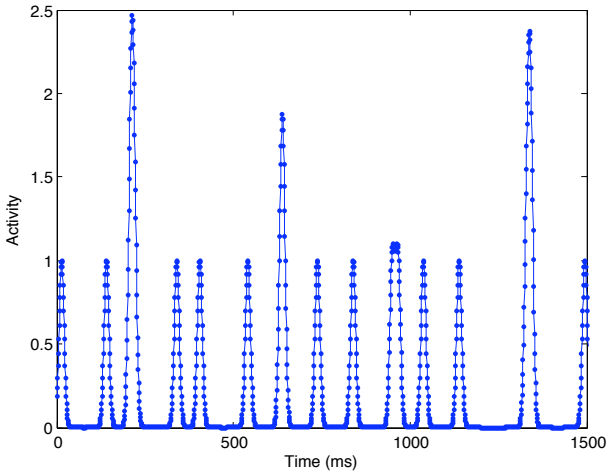

Figure 5.8: Spike activity convolved with $\exp\left(-\left(\frac{x}{10}\right)^2\right)$ kernel. This procedure blurs the spike train significantly, but lets one receive non zero product of two such trains even if corresponding spikes are shifted. The product is later integrated to obtain *synchronization strength*, a measure we introduced to describe similarity between spike trains.

Chapter 6

Further research

6.1 Introduction

The model discussed in the previous chapter is somewhat artificial. The neuronal groups presented there, are not biologically motivated (though one might think of them as of cortical columns), and only exhibit the appearance of mechanism discussed earlier in the dynamical systems setup, though the methodology of creating a graph based on synchronization much resembles that used in empirical studies of Eguíluz *et al.* (2005), which revealed scale-free structure in functional MRI data. We made efforts to seek for power law structures directly in a more biologically plausible model, but the results are not evident. Nevertheless we decided it is worth discussing these simulations (simulating 100000 dynamical neurons is a challenging task), pointing out the possible reasons for the failure of finding power law structures and examining possible further directions.

6.2 The model

The model we investigated was introduced in the well known paper of Izhikevich *et al.* (2004) which appeared in *Cerebral Cortex*. It consists of a neuronal sphere filled with 100000 excitatory and inhibitory neurons having both local and remote connections. Each neuronal connection has a delay that is proportional to the distance (long range connections have different propagation velocities though). The synapses model four types of synaptic receptors having linear kinetics. The synapse strength is determined both by short term depression-facilitation as well as long range changes controlled by *spike timing dependent plasticity* (STDP). In the course of the simulation STDP causes neurons to self-organize into spontaneous groups. We were able to reproduce these groups and study their statistical features.

6.2.1 Anatomy

The model consisted of 80000 excitatory neurons and 20000 inhibitory neurons spread uniformly on the surface of a sphere of radius $8mm$. The 4/1 ratio of excitatory/inhibitory neurons is observed in mammalian brains. Each neuron was modeled as a single compartment unit using the simple phenomenological model (Izhikevich, 2003). Each excitatory neuron had 75 local connections spread uniformly within a radius of $1.5mm$ and 25 long distance connections. The targets of these long range connections lied within a circle of radius $0.5mm$ with a randomly chosen center. One can think of these long range connections as of a single axonal fiber, branching at its end into 25 collaterals targeting randomly chosen neurons within the range of $0.5mm$. These long range connections (that model long myelinated axons) had spike propagation speed of $1m/s$ whereas the local connections of $0.15m/s$. Each inhibitory neuron had 25 local targets spread uniformly within a radius of $0.5mm$. This setup resulted in synaptic delays varying from 1 to 12 ms of model time (the model was simulated with $1ms$ resolution as discussed in further sections). As in the original model (Izhikevich

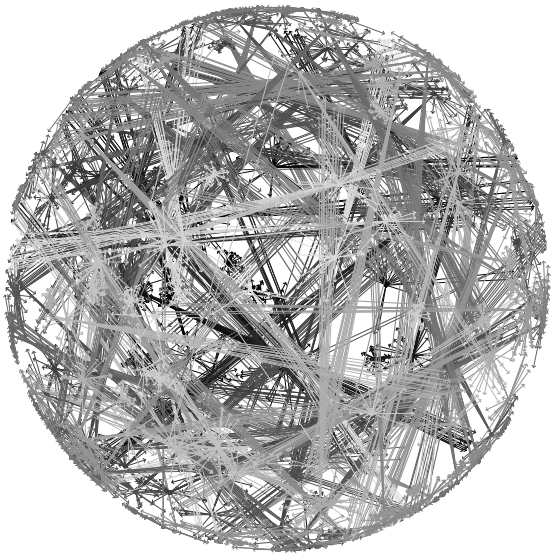

Figure 6.1: A sketch of the connectivity in the neuronal sphere model investigated in this chapter. Only 200 out of 100 000 neurons are plotted.

et al. , 2004), to assure heterogeneity neural parameters a, b, c, d (see section 3.5.3 in particular figure 3.8) were varied. In general, excitatory neurons had $(a, b) = (0.02, 0.2)$ and $(c, d) = (-65, 8) + (15, -6)r^2$ where r is a uniform random variable (single copy). Such a choice of parameters results in most of the excitatory neurons falling into regular spiking (RS) regime with some fraction

tending to chattering regime (CH) (compare with figure 3.8). Inhibitory neurons had $(a, b) = (0.02, 0.25) + (0.08, -0.05)r$ and $(c, d) = (-65, 2)$ and therefore, as explained above, they interpolate between low threshold spiking (LTS) and fast spiking (FS) regimes. The typical neuronal density on the surface of the sphere was 125 neurons/mm^2

6.2.2 Short term synaptic dynamics

The model consisted of 8.5 millions of synapses, and their efficient implementation was crucial. Each synapse incorporated short term depression/facilitation proposed by Markram *et al.* (1998) as follows:

$$\begin{aligned} \frac{dR}{dt} &= \frac{1-R}{D} - Rw\delta(t - t_{spike}) \\ \frac{dw}{dt} &= \frac{U-w}{F} + U(1-w)\delta(t - t_{spike}) \end{aligned} \tag{6.1}$$

where $\delta()$ is Dirac delta function, U, D, F are parameters, t_{spike} is the time of spike of the presynaptic neuron. That is whenever presynaptic neuron is quiescent, R and w variables evolve according to linear differential equation:

$$\begin{aligned} \frac{dR}{dt} &= \frac{1-R}{D} \\ \frac{dw}{dt} &= \frac{U-w}{F} \end{aligned} \tag{6.2}$$

When the spike occurs, R gets decreased by Rw and w gets increased by $U(1-w)$. The product Rw models in the sequel the fractional amount of neurotransmitter (affecting the respective conductance), therefore rate of change of postsynaptic potential (PSP) is proportional to Rw. The parameters U, D, F in general can be different for every synapse which would require simulating 8.5 millions such equations. However synapses found in most biological neurons are rather homogenous and it is reasonable to assume that each neuron's synapses are identical with respect to the short term dynamics (in this model the synapse parameters only differ between excitatory and inhibitory neurons, as discussed below) and consequently the equation 6.1 can be simulated per neuron not per synapse (the solution of 6.1 depends entirely on intervals between presynaptic spikes and the presynaptic neuron type, so values Rw are identical for every outgoing synapse of the same neuron, but may arrive at different time steps depending on the axonal delay). The parameters were $U = 0.5$, $F = 1000$ and $D = 800$ for excitatory neurons and $U = 0.2$, $F = 20$, $D = 700$ for inhibitory neurons. Consequently inhibitory synapses facilitate a lot faster than excitatory ones. In general these parameters correspond to synapses exhibiting depression, so when a couple of spikes arrives

at the synapse, the first one may produce a high PSP but consecutive ones will be weaker and weaker.

6.2.3 Long term synaptic dynamics

Apart from short term dynamics excitatory synapses have synaptic weights controlled by long term dynamics. Each excitatory synaptic weight $c_{i\to j}$ from neuron i to j can take values in $[0, 0.5]$. Short and long term dynamics both contribute to the rate of change of excitatory postsynaptic potential, which is proportional to the product $c_{i\to j}R_i w_i$. This term for excitatory synapses can be at most 0.25 (and 0.2 for inhibitory synapses which have all weights constant and equal 1). The weights are controlled by a differential equation (simulated and here scaled for time step 1s):

$$\frac{dc}{dt} = 0.001 + c^{\Delta}(t) \tag{6.3}$$

where $c^{\Delta}(t)$ is an auxiliary function following its own dynamics (discussed further). The 0.001 term is responsible for slow, activity independent amplification of synaptic weights, in particular of those neurons that remain quiescent in the course of the simulations (such weights are no amplified by the spike timing dependent plasticity). Values of c are clipped in order to stay in $[0, 0.5]$. $c^{\Delta}(t)$ is controlled by a linear differential equation:

$$\frac{dc^{\Delta}}{dt} = -\frac{c^{\Delta}(t)}{10} \tag{6.4}$$

simulated with time step 1s and here scaled appropriately. Apart from the above equation, another mechanism contributing to the values od c^{Δ} and hence to the rate of change of c is the so-called *spike timing dependent plasticity* (STDP). The STDP is a novel form of Hebbian learning taking into account temporal structure of pre and postsynaptic spikes (Markram *et al.* , 1997). The paradigm of the idea is that if the presynaptic spike precedes the postsynaptic one (and that way may have contributed to the occurrence of postsynaptic spike), then the weight leading from pre to postsynaptic neuron should be potentiated. If on the other hand the presynaptic spike comes after postsynaptic spike, then the weigh should be depressed. The rate of depression/potentiation depends on time interval between the spikes. Assume $\triangle t = t_{post} - t_{pre}$, then *long term potentiation* is $LTP = A \cdot e^{\frac{\triangle t}{\tau_{LTP}}}$ for positive $\triangle t$ and *long term depression* $LTD = A \cdot e^{\frac{-\triangle t}{\tau_{LTD}}}$ for negative $\triangle t$. In this model $\tau_{LTP} = 15$, $\tau_{LTD} = 20$, $A = 0.004$, therefore the depression curve has a higher time constant so in general depression is stronger than potentiation. Values of LTP/LTD are added/substracted from c^{Δ} when the spikes are processed. The detailed discussion of this mechanism is in subsection 6.3.2 related to implementation details.

6.2.4 Neuronal dynamics

Each neuron was simulated as a single compartment dynamical system based on the simple model discussed in 3.5.3. Recall that the model is based on a quadratic/linear set of equations:

$$\begin{aligned} \frac{dV}{dt} &= 0.04V^2 + 5V + 140 - u + I_{syn} \\ \frac{du}{dt} &= a(bV - u) \end{aligned} \tag{6.5}$$

where parameter I_{syn} is the total synaptic input of the system. The total synaptic input consists of modeled synaptic transmitters:

$$\begin{aligned} I_{syn} = &- g_{AMPA} \cdot V + \\ &- g_{NMDA} \cdot \frac{\left(\frac{(V+80)}{60}\right)^2}{1 + \left(\frac{(V+80)}{60}\right)^2} \cdot V + \\ &- g_{GABA_A} \cdot (V + 70) + \\ &- g_{GABA_B} \cdot (V + 90) \end{aligned} \tag{6.6}$$

where V stands for the postsynaptic membrane potential. Therefore the synaptic input depends on the membrane voltage (this dependency is a bit weird for NMDA). The g_{AMPA},g_{NMDA}, g_{GABA_A} and g_{GABA_B} model conductances of alpha-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid, N-methyl-D-aspartic acid and gamma-aminobutyric acid respectively and have linear kinetics:

$$\frac{dg_{AMPA/NMDA/GABA_{A/B}}}{dt} = -\frac{g_{AMPA/NMDA/GABA_{A/B}}}{\tau_{AMPA/NMDA/GABA_{A/B}}} \tag{6.7}$$

The appropriate time constants are $\tau = 5$, 150, 6 and 150 for $AMPA$, $NMDA$, $GABA_A$ and $GABA_B$ receptors respectively (see Dayan & Abbott (2001)). Since V is typically negative at the rest state, so g_{AMPA} and g_{NMDA} act as excitatory transmitters and g_{GABA_A} and g_{GABA_B} usually act as inhibitory transmitters (unless V gets below $-70mV$). Each time a presynaptic excitatory spike reaches the synapse, g_{AMPA} and g_{NMDA} are increased by $c_{i\to j}R_iw_i$ (with R_iw_i being the short term depression/facilitation variables of presynaptic neuron i as described above) . Each time a presynaptic inhibitory spike reaches the synapse, g_{GABA_A} and g_{GABA_B} are increased by R_iw_i (inhibitory connections are not plastic, and have fixed weight equal 1). Apart from synaptic stimuli, the model incorporated a random noisy input (without it, the activity of the model would not persist). The input was simulated as a forced spiking steered by the Poisson point process with intensity of $1/1000ms$. On average one spike per second for each neuron can

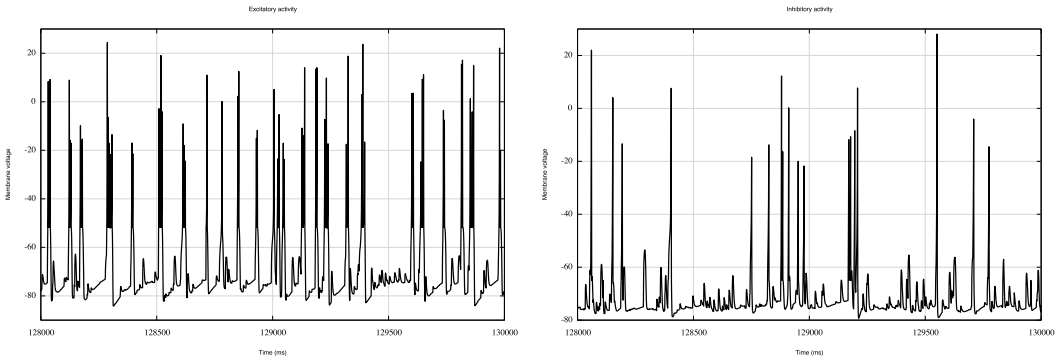

Figure 6.2: A plot of changes of membrane voltage in a randomly selected excitatory neuron (left) and inhibitory neuron (right). Each figure shows the activity over a period of 2s. Notice a number of sub-threshold jumps.

be attributed to the noisy input, the rest of them (about 7 spikes per second) were due to synaptic interactions. There were some discrepancies when forced spiking was replaced by a super-threshold pulse, or fake synapse, especially at the beginning of the simulation where rhythms were more significantly pronounced. Forced spiking leads, however, quickly to a steady Poissonian spike train desirable for the simulation.

6.3 Implementation details

In this section we examine some of the issues already discussed previously, this time focusing on implementation details. Since neurosimulations of such a scale and complexity have been developed fairly recently, whereas scientific papers published in various journals frequently do not include detailed implementation information, we decided this section might be valuable.

6.3.1 Short term dynamics

Recall from subsection 6.2.2, that the short term dynamics defined in equation 6.1 was computed per neuron, not per synapse. That way the number of equations to solve is reduced from nearly 8.5 million to 100000 (almost two orders of magnitude). This reduction is possible if we assume that each neuron has homogenous synapses (in terms of short term dynamics). Nevertheless there is still some space for improvements: note that most of the time neurons do not spike and so it is a good idea to optimize numerical solution to 6.2. This could be done

by a variable substitution:

$$\begin{aligned} R_{fast} &= 1 - R \\ w_{fast} &= w - U = -(U - w) \end{aligned} \tag{6.8}$$

that way equation 6.2 becomes:

$$\begin{aligned} \frac{dR_{fast}}{dt} &= -\frac{R_{fast}}{D} \\ \frac{dw_{fast}}{dt} &= -\frac{w_{fast}}{F} \end{aligned} \tag{6.9}$$

which is particularly easy for numerical simulation, since Euler scheme can be replaced by a single multiplication (this saves a couple of valuable processor cycles). The rest of the equation 6.1 becomes a bit more complicated:

$$Rw = (1 - R_{fast}) * (U + w_{fast}); \tag{6.10}$$

but it only needs to be computed when presynaptic neuron actually spikes (and that happens usually once per a 100 simulation steps). Whenever presynaptic neuron decides to spike, the product of R and w from that particular moment is computed and forwarded to the synapses, where it arrives after respective axonal conduction delay and contributes to PSP.

6.3.2 STDP

The efficient implementation of STDP is essential for the model since this mechanism has to be triggered for every pair of spikes emerging at two connected neurons. Things get more complicated, since there are at least several of implementations of STDP varying by selection of spikes that contribute to plasticity (Izhikevich & Desai, 2003). As suggested in the paper, in the original model the authors used an implementation which takes only temporarily nearest pairs of spikes into account (that means LTP/LTD functions are being completely reset in the event of a spike). In our implementation this resulted in a bit too high firing rate (about 25 spikes/s) and too strong potentiation (see figure 6.3). After switching the implementation to the one that increments LTP/LTD, we reached a spiking regime that better resembles the one (about 8 spikes/s) obtained in the original implementation (compare figure 6.4). It remains unclear why in our implementation of the model with "resetting" STDP converged to the other spiking regime. The model of such a complexity however, is sensitive to certain parameters and implementation details (like the mechanism of random stimulation) and it is very difficult to get identical behavior. The correspondence with the author of (Izhikevich *et al.* , 2004) was very useful in dealing with many implementation

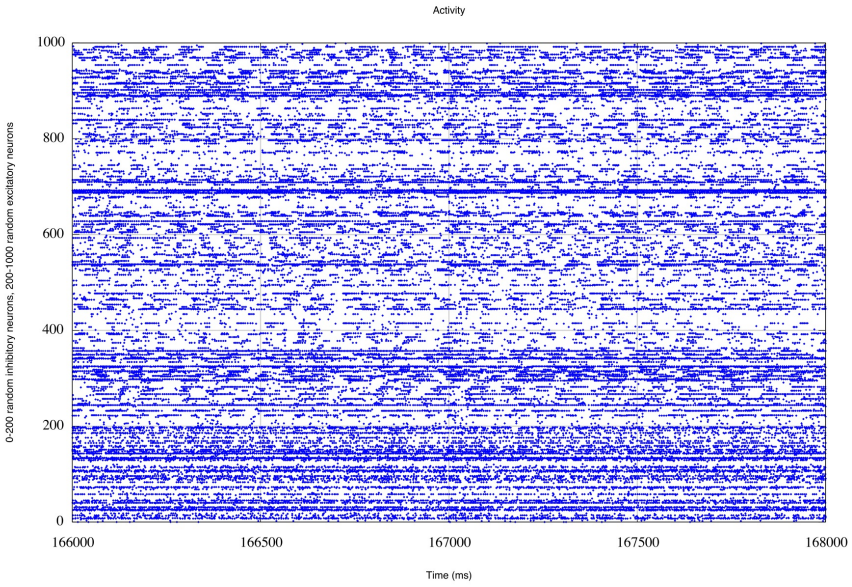

Figure 6.3: A sample spike train obtained from the model, that used the "resetting" implementation of STDP. The firing regime is a bit different than the one in the original model, the firing to high, there is a lot of bursting.

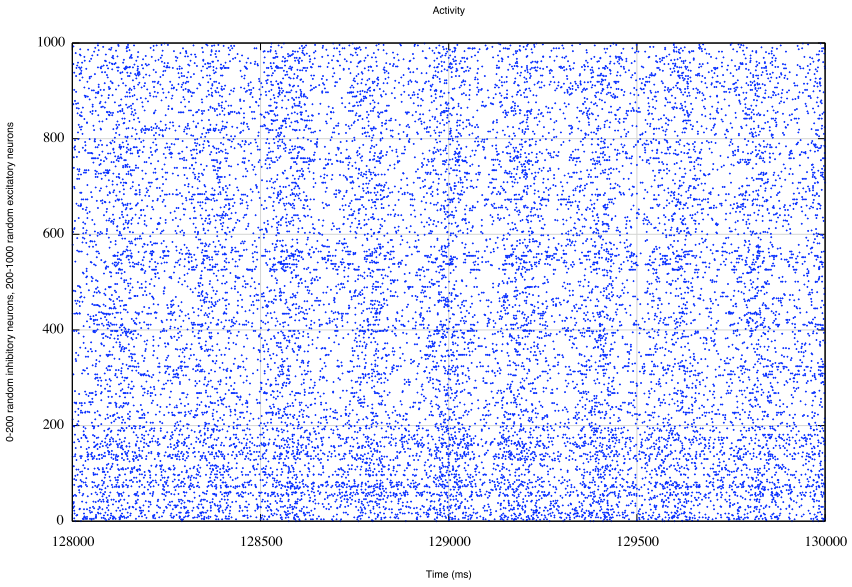

Figure 6.4: A sample spike train obtained from the model using the "incrementing" implementation of STDP. The firing regime here resembles the regime from the original model.

details[1] but by the moment of writing we still haven't figured out the STDP issue (it is worth noting that the emergence of neuronal groups was also observable in the other spiking regime). Nevertheless "incrementing" implementation is not much different and results in approximately right spiking regime, so the rest of the simulation was done in the following setup:

- Each neuron keeps LDP/LTP values. It needs only instantaneous (current) value of LTD and a piece of LTP curve extending to the past up to the maximal conduction delay (this guarantees that any target neuron that receives a spike will be able to use the LTP value of the presynaptic neuron recorded at the time it spiked). LTP is kept in a cyclic buffer.

- If the neuron i spikes, it alters the rate of change of all its excitatory presynaptic weights by increasing c^{Δ} by values of LTP kept in the presynaptic neurons (these values become significant if presynaptic neuron spiked recently prior to the conduction delay, and so might have contributed to the current spike of i). Recall that LTP is kept in a cyclic buffer, thus previous values extending up to maximal conduction delay are available. If we denote the current time by t and conduction delay from neuron j to i $D_{j\to i}$ we have:
$$c^{\Delta}_{j\to i} := c^{\Delta}_{j\to i} + LTP_j[(t - D_{j\to i})] \tag{6.11}$$
for every presynaptic neuron j of i ($(t - D_{j\to i})$ is reduced modulo maximal conduction delay). After updating presynaptic c^{Δ}, neuron i updates its own LTP_i/LTD_i functions, increasing[2] them by $A = 0.004$.

- If the neuron i receives a spike from another neuron j it decreases $c^{\Delta}_{j\to i}$ by its own value of LTD_i (note this value will be significant if neuron i spiked recently).

- Changes to $c_{j\to i}$ are not instantaneous, in fact all STDP contributions are accumulated in $c^{\Delta}_{j\to i}$ for a $1000ms$ (time steps) and impact $c_{j\to i}$ once per $1s$ (model time) according to long term dynamics described in subsection 6.2.3.

LTP is implemented as a cyclic buffer, in each step of simulation next value (modulo maximal conduction delay) is computed from previous one. The same applies to LTD but in this case previous value can be deleted. Since STDP curve

[1]The author would like to greatly acknowledge the help of Dr Eugene M. Izhikevich who without any hesitation answered a number of emails containing an even greater number of questions.

[2]The other implementation of STDP discussed in this section differs only in this point: the values of LTP/LTD are being reset to $A = 0.004$ instead of being increased.

is an exponential curve, the computation is particularly simple, the next value is obtained from previous by multiplication by a constant multiplier.

6.3.3 Neuronal dynamics

Recall from 6.2.4 that the dynamics of each neuron is controlled by:

$$\begin{aligned}\frac{dV}{dt} = 0.04V^2 + 5V + 140 - u - g_{AMPA} \cdot V + \\ - g_{NMDA} \cdot \frac{\left(\frac{(V+80)}{60}\right)^2}{1 + \left(\frac{(V+80)}{60}\right)^2} \cdot V + \\ - g_{GABA_A} \cdot (V + 70) + \\ - g_{GABA_B} \cdot (V + 90) \\ \frac{du}{dt} = a(bV - u)\end{aligned} \tag{6.12}$$

The above setup suggests a rather straightforward method of numerical solution based in the first order Euler method, but this could be tricky. The incorporation of inhibitory synaptic receptors in the model introduces a numerical instability. Note that when V is positive, inhibitory conductances give it a huge inhibitory kick. In the next step V is strongly negative and gets a strong excitatory push instead. This oscillatory mechanism may amplify under certain conditions[3] (which are not very rare to occur in the system of 100000 neurons) and lead to numerical instability (V becomes so large, that recovery variable becomes NaN; because recovery is not being reset after the spike, the instability remains conserved once it appears). This undesirable phenomenon can be avoided by using mixed open/closed Euler scheme[4]. The open part includes the voltage and recovery dependent part of the equation, whereas the closed part includes voltage dependent synaptic conductances. Since they depend linearly on V (with the exception of $NMDA$ which is "nearly" linear), the closed scheme has a uncom-

[3]The exact mechanism leading to instability is somewhat knotty and depends on the details of numeric implementation. In general the instability emerges, when an inhibitory spiking mechanism gets activated, which gradually increases the recovery variable. Once the recovery variable is very large, V may become numeric infinity, and then u becomes *not a number (NaN)*. The process takes some time, and is not a problem of single iteration. In a huge system however (like the one discussed), it turned out to eliminate about ≈ 40 neurons each 1000 ms, which is obviously not acceptable. Increasing the precision wouldn't improve the situation significantly.

[4]This method was suggested by Dr. E. Izhikevich, who faced similar problems while implementing the original model.

plicated solution:

$$V_{t+1} = \frac{(V_t + \Delta_t \cdot ((0.04 \cdot V_t + 5) \cdot V_t - u + 140 - 70 \cdot g_{GABA_A} - 90 \cdot g_{GABA_B}))}{1 + \Delta_t \cdot (g_{AMPA} + g_{NMDA} \cdot (V_{NMDA}) + g_{GABA_A} + g_{GABA_B})} \tag{6.13}$$

where we assume:

$$V_{NMDA} = \frac{\left(\frac{(V_t+80)}{60}\right)^2}{1 + \left(\frac{(V_t+80)}{60}\right)^2} \tag{6.14}$$

is computed with the same time step as synaptic conductances (that is 1ms for discussed simulation) and is kept constant (as well as other synaptic parameters) while the finer (that is 0.5 ms) time step solution of voltage is computed.

6.3.4 Code structure and parallelization

Simulating 100000 neurons is a demanding task for contemporary computers, even if the neurons are fairly simple. Each second of simulated model time requires billions of unavoidable floating point operations, and interactions. To some extent the problems found in neuro-simulations related to code parallelization are similar to those of computational fluid dynamics with the neuronal activity resembling fluid flow over a very complex domain[5]. This particular model however is well suited for parallelization and does not incorporate any global conservation principles like those found in inviscid fluid simulations[6]. The original simulation was performed (according to Izhikevich *et al.* (2004)) on a Pentium 1 GHz with 1.5 GB of RAM. At the time of writing this thesis we could utilize two machines: 4 core Xeon 2.0 GHz with 4GB of RAM and an 8 core Xeon 2.8 GHz with 10 GB of RAM[7]. In order to exploit the multi-core architecture the OpenMP (OpenMP (2008)) language extensions available in GCC 4.2 and above were used. OpenMP is a very convenient parallelizing language extension which lets one use contemporary shared memory parallel systems while skipping laborious thread maintenance sections. The parallelization is achieved by a system of pragmas inserted before important sections of code e.g. `#pragma parallel for` inserted in the program before `for` loop, instantly parallelizes the loop, by default the

[5]In fact a detailed brain simulation would have to take into account a flow of oxidized blood and diffusion of nutritional substances etc., which would incorporate CFD as well...

[6]In such case the fluid flow (velocity) vector field has to have a zero divergence over the whole domain, which requires solving a global system of linear equations. In the case of neurosimulations such global principles can be imposed for example by a limit on global activity depending on the inflow of substances required for metabolism etc.

[7]The author would like to thank Mr Leszek Rybicki for the opportunity of using his 8 core Mac Pro.

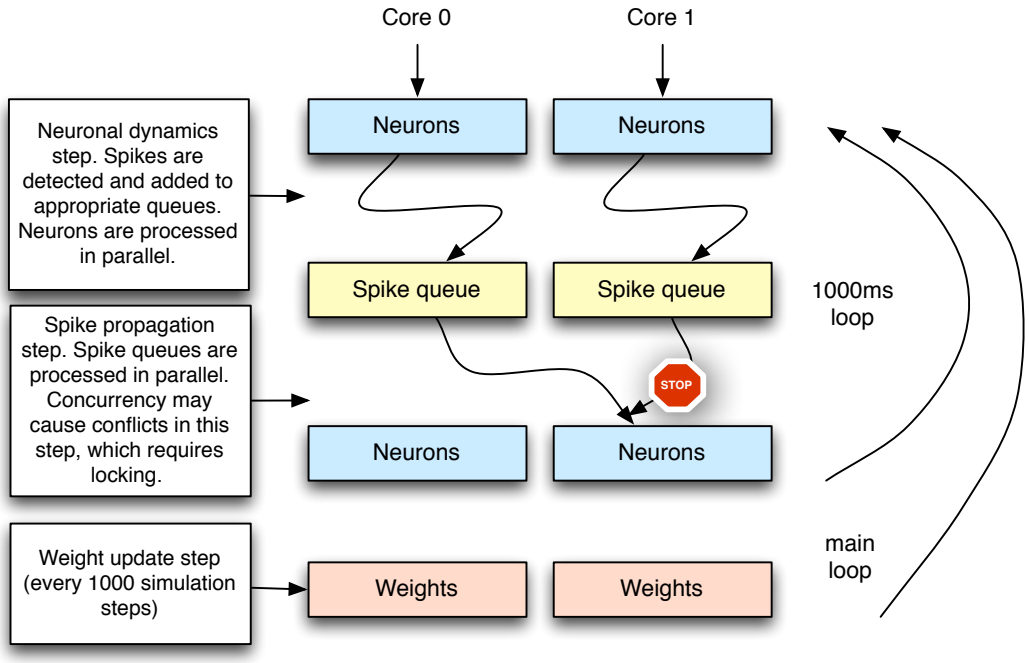

Figure 6.5: Structural presentation of the code used in simulation. The same structure can be extended to an arbitrary number of cores.

number of threads created depends on the number of available cores in the machine. But parallelizing code is far from being that simple, there is a lot of work left with the synchronization of threads and effective load balance which are not trivial. The crucial step in case of neuro simulations (as well as CFD etc.) is the propagation of information across the domain (that is propagation of spikes in this case) which inherently requires synchronization of multithread/distributed systems and therefore its efficient implementation is critical. Each spike has a set of targets associated with it, every target has an established delay. In the discussed model, the spikes are handled as follows:

- When a neuron spikes (V jumps above 30mv and is being reset), the short term depression/facilitation is computed, and a spike is added to the spike queue. The queue (that consists in reality of a couple of queues) contains the source neuron number, depression/facilitation value, index of the processed synapse (0 initially) and the time elapsed since the spike (0 initially). The important issue here is that the list of synapses of every neuron is sorted (ascending) with respect to axonal delay.

- After all neurons are processed, the next step is to take care of the spikes. The spike queue is examined one by one. First, the time elapsed since spike

occurrence is incremented. Next, all the synapses that match the current delay are processed (the fact that the synapses are sorted with respect to axonal delay guarantees that all the synapses are processed on time). If there are no more synapses with matching delays, next spike is being processed.

- Finally the queue is inspected once again from the beginning, deleting all the spikes that reached their final synapse (until an unfinished spike is found). Eventually every spike will end up in this initial segment of the queue and get deleted, not necessarily right after it reaches it's final synapse, but at worse a couple of time steps later.

The above algorithm is optimal, since every event of spike reaching its destination has to be processed, and in the presented setup each such event is processed exactly once, and exactly when it should. It is impossible to process all the arrivals right at spike time, since synaptic conductances depend on future postsynaptic voltage to become available after axonal delay. In our implementation the presented mechanism had to be adapted to parallelization issues. This requires that the spike queue be locked while a spike is being added (since many threads might have reached the spiking neuron at the same time). Adding a semaphore would solve the problem but could introduce significant lags (in fact it does). A better idea is to split the spike queue into many independent queues, each attached to a certain thread id. That way, locking is avoided and all the threads can process their neurons independently. After neurons have been processed, the spike queues can again be processed in parallel. There is unfortunately an unavoidable critical section here. When a spike is processed, it alters synaptic conductances of the postsynaptic neuron, but the same neuron might be a target of another spike, being processed by different thread. The most efficient solution is to prepare a lock for every neuron and move the modifying code into protected area. It requires some extra memory for the locks, but is certainly more efficient than creating a global critical section (since there are 100000 neurons and just a couple of parallel threads the concurrency is rare and the time the threads spend on locks is not ruinous). One tiny improvement would be to create two locks per neuron – an excitatory lock and inhibitory lock, since excitatory and inhibitory spikes modify different conductances and therefore are independent.

In the presented setup the program was able to utilize about 750% of processor time[8] on an 8 core machine (the remaining 50% idle was lost on synchronization and memory access). Any further improvements of the code would have to take into account costly access to the systems memory which would become a real bottleneck for systems with more than 8 cores. Most of such improvements would

[8]Which is not bad, taking into account Amdahls law.

be low level and depend heavily on particular architecture (loop unrolling, cache optimization etc.).

6.4 Results

The simulation was run for a couple of days which corresponds to several hours of model time (the simulation was carried out for a number of times in different regimes). Initial excitatory weights were in all cases set to 0.1, the model needed about a 1000 seconds (model time) to approach asymptotic firing rate (see figure 6.6). In the setup discussed above excitatory neurons fired between 8 and 9 spikes per second. Inhibitory neurons had higher firing rates (between 11 and 13 spikes per second) that balanced their smaller number and connectivity. The model converged into a homeostasis – even though weight continued to change, the the global behavior remained constant. By looking at figure 6.7 it is easy to see that

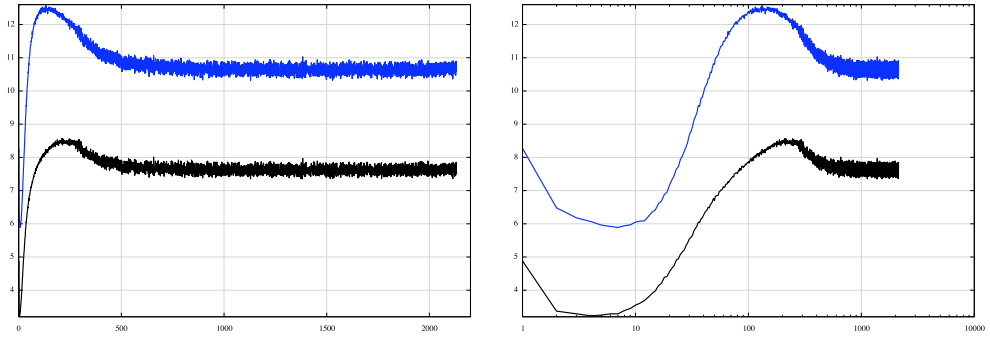

Figure 6.6: The firing rate versus time at the beginning of the simulation for excitatory neurons (black) and inhibitory neurons (blue). The plot on right shows the same data in semi logarithmic scale. Clearly the system converges into an equilibrium state after about $1000s$ of model time.

the spatial distribution of activity was rather homogenous as expected. There were some variations on the small scale though, note a number of "hot" neurons on the sphere picture, but they were evenly distributed all over the sphere. Small number of fairly quiescent neurons is also visible. Distribution of synaptic weights and EPSP (figure 6.8) was a good marker indicating whether the model is in the right spiking regime (corresponding to the original model). The most interesting part of this numerical study was the emergence of spontaneous neuronal groups. In each such group neurons are organized so that they fire in time locked intervals (not necessarily synchronously). In (Izhikevich, 2006c) Eugene M. Izhikevich coined a term *polychronous* for such a behavior. In his own words *"Since the firings of these neurons are not synchronous but time-locked to each other, we*

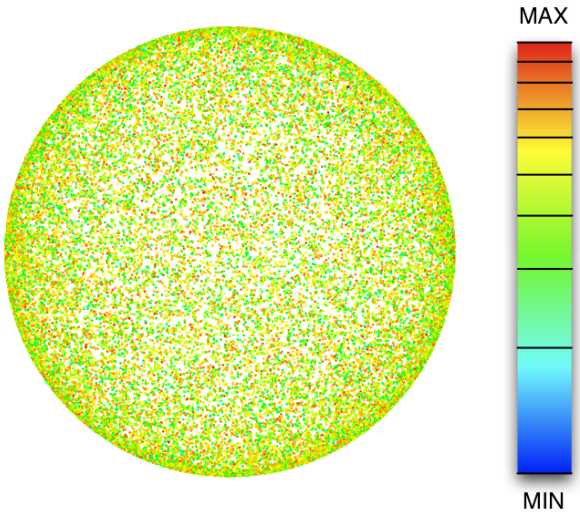

Figure 6.7: Variation of activity (spiking rate) in the sphere model. The hue represents excitatory activity in logarithmic scaling. The figure demonstrates that the activity is very homogenous.

refer to such groups as polychronous, where poly means many and chronous means time or clock in Greek. Polychrony should be distinguished from asynchrony, since the latter does not imply a reproducible time-locking pattern, but usually describes noisy, random, nonsynchronous events.". Polychrony is an interesting phenomenon since these groups can overlap, and each neuron could be a member of many such groups. An analogy in classical circuits could be that a single logical gate be a potential part of completely different computations depending on a current cycle (which sounds inconceivable with present technology). Nevertheless neurons are not as simple as logical gates, and could be rather described as functional circuits. In this setting the participation in such a polychronous group sounds a bit like switching a context in a microprocessor.

Emergence of polychronous groups was studied in (Izhikevich *et al.* , 2004) using a simple algorithm based on the concept of anchor neuron. We used this algorithm in present study as well, although in later work (Izhikevich, 2006c) other method of detecting groups was incorporated. The anchor-based algorithm works as follows:

- Select an anchor excitatory neuron having two or more strong (within 5% of the strongest connections) connections to other excitatory neurons. This ensures that firing of the anchor neuron increases the probability that target neurons will fire in the appropriate time, resulting from axonal delays.

- For the descendants of the anchor neuron, find any common postsynap-

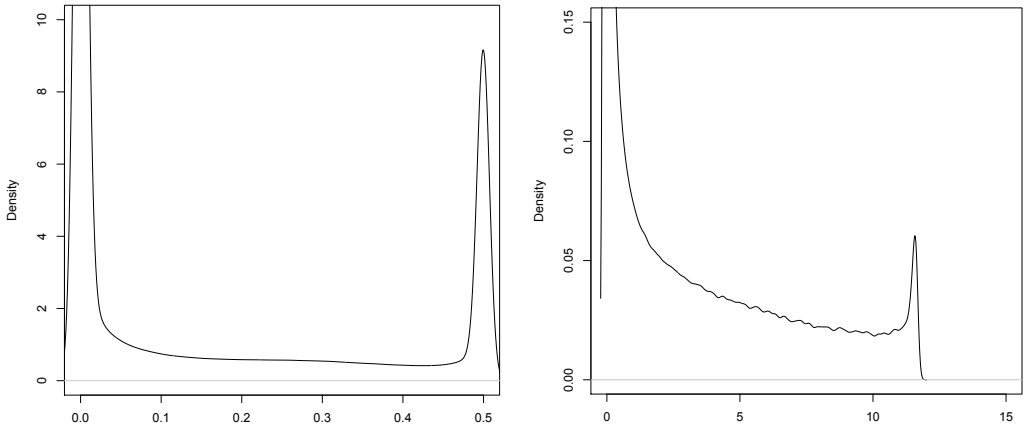

Figure 6.8: Distribution of excitatory weights (left) and corresponding distribution of excitatory postsynaptic potentials (EPSP). EPSPs are computed by taking into account the state of synaptic short term depression/facilitation and assuming that the postsynaptic neuron is at rest (compare with original model Izhikevich *et al.* (2004), the results are very similar.)

tic targets that have strong connections and matching delays, that is to say if the descendants of the anchor neuron fire excited by the anchor, the resulting spikes will reach the common target synchronously (within 2ms interval). If no such common targets are found, the anchor neuron is discarded.

- If there are such common targets, they are added to the group. The process is repeated, that is to say common postsynaptic targets with matching delays for all the current members of the group are found. The process finishes when there are no more such targets.

The algorithm, though rather straightforward, has some details which may severely influence the results. First of all, in the original model, the possible targets were restricted to the *local circuitry* neurons, that is long myelinated connections were not taken into account. Secondly not every target with converging delays is selected as a new group member, only those whose excitation was strong enough. This leaves some space for ambiguity. Authors in (Izhikevich *et al.* , 2004) used estimated EPSP as a discriminating factor, but the estimation itself does not take into account the state of the postsynaptic neuron (see figure 6.8) which makes this estimate rather artificial. Furthermore, EPSPs change all the time which may cause some further distortions. In the original paper authors found about 1.5 thousand polychronous groups, whereas in the presented implementation depending on the parameters of group finding algorithm we found between 4000

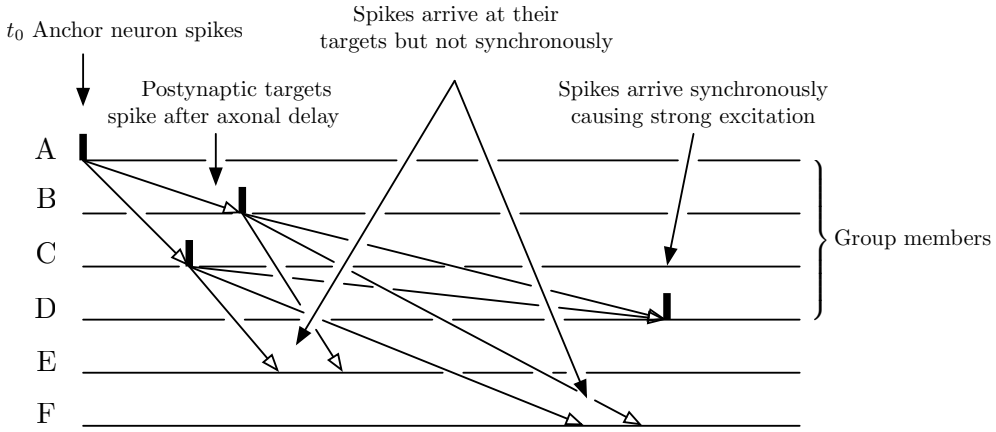

Figure 6.9: Schema of the group finding algorithm. A is the anchor neuron, B,C its immediate descendants. We assume that B and C fire after excitation coming from the anchor. Next we seek for common postsynaptic targets having converging delays (D). E and F are discarded from the group.

up to 13000 such groups. These figures are not frightening though since later developments with other group finding algorithms show that the number of polychronous groups could easily exceed the number of neurons in the system (this system contains 80000 excitatory neurons!). The differences are also by no means inconsistent with the original results presented in (Izhikevich *et al.* , 2004) and may be due to some tiny details of the algorithm the authors of the original model actually used (unfortunately the exact code they used is not available[9]). On the contrary they show that the concept of spontaneously developing neuronal group is robust and manifests itself with variable intensity in wide range of models and group search algorithms. In the end we decided not to worry about the huge number of the groups and decided to build them in the following setup:

- Ignore the global connections - they have rather long axonal delays and provide feedback (reentrant) connections not much correlated with the local activity.

- The decision of whether a neuron should be added to the group is based on the value of the synaptic weights, not by estimated EPSP. Large weight usually means large EPSP (certainly tiny weights force small EPSPs). In this setup the groups should be more stable, though the algorithm finds far more groups than the original one.

[9]Dr Eugene M. Izhikevich was so kind to provide some pieces of code and valuable hints for which the author is grateful.

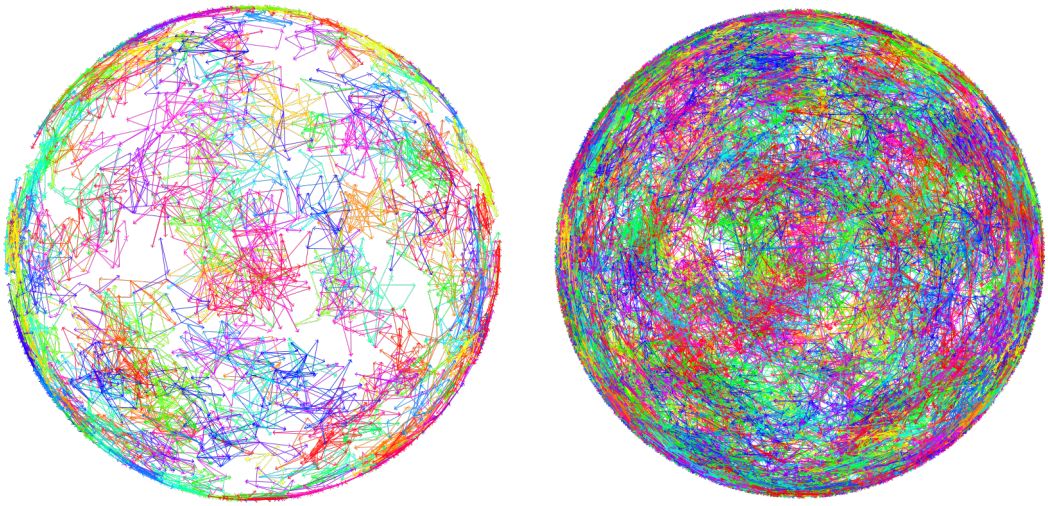

Figure 6.10: A fraction of the groups fund in one of the simulations (total number of groups in this case was about 13000). The left figure shows 2% of total number of groups, while the right figure shows 10%.

The groups found in our simulations are plotted in figure 6.10. The picture resembles the original figure in (Izhikevich *et al.* , 2004).

The whole numerical simulation was conducted in order to seek for power law distributions among the signals between the groups. Note that this setup is entirely different compared to the one presented in chapter 5: the groups are not isolated, furthermore they even frequently overlap. These circumstances caused difficulties in deciding what actually should be regarded as the group input, and what should be regarded as output. Nevertheless we provided a couple of simulations, calculating the number of spikes incoming and outgoing to/from every group and analyzed the output histograms. Unfortunately on the current state of the art we have to conclude that we cannot confirm the existence of power law distributions in this system.

There are a number of reasons why the histograms presented in figure 6.11 could be questionable. First of all there is some level of noise in the system. At least 10% of the spikes that appear are forced by random process and many others could be attributed to reentrant inputs, which in this statistics would also act as noise (since these connections come from long distances, have significant delays and usually are not correlated with local activity). The noise level in this case is difficult to estimate but it is easy to imagine that it could be very significant. Secondly the groups are constantly changing (some appear and some disappear), and even though many of the groups are persistent it is questionable on how to treat those volatile groups in the statistics (in the above setup they were ignored).

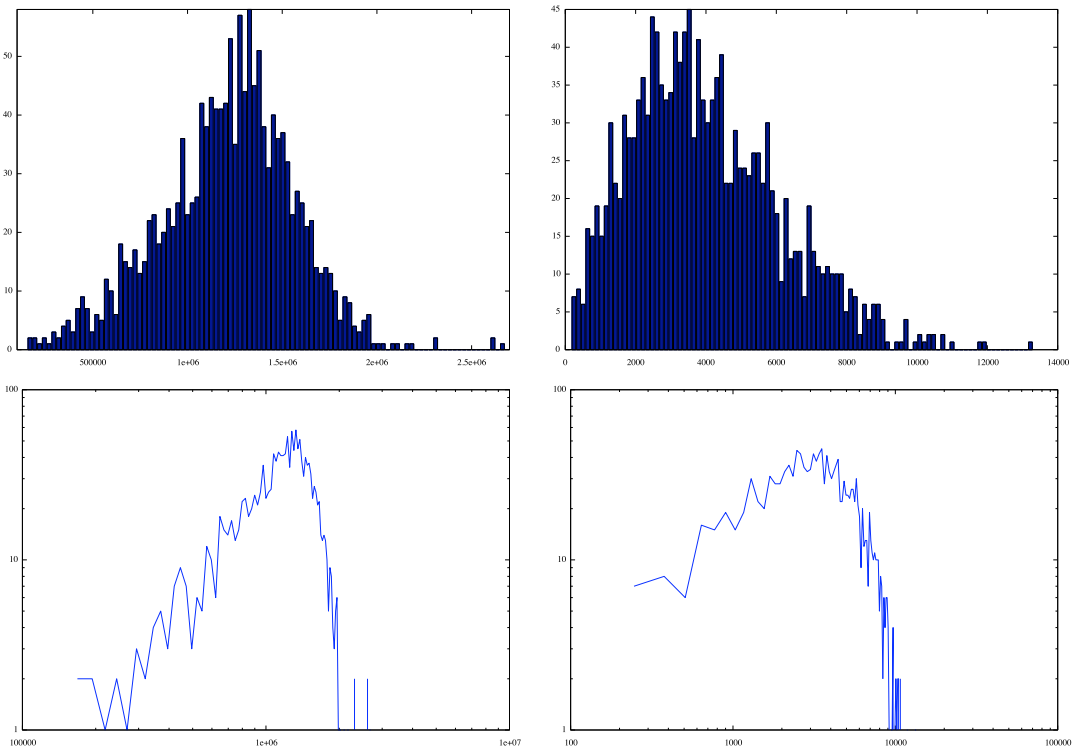

Figure 6.11: Sample histograms of group activity. Left figure shows the output (number of spikes departing from the group versus number of groups), the right figure shows the same statistics divided by the activity of an anchor neuron. In the lower row are the logarithmic plots.

Even though at short intervals the number of volatile groups is small, in the long turn they may become very significant.

The numerical experiment described above should be considered as a warm up task before the real challenge which is simulating a whole functional brain in a computer system. First efforts to run such an extensive simulation are already being made (Izhikevich & Edelman, 2008). The idea is to reconstruct the large scale connectivity of a fully sized brain based on DTI (Diffusion Tensor Imaging) and anatomical data. The small scale connectivity can be then reconstructed randomly with respect to known distributions found in various parts of the brain. In further developments the model should be equipped with the possibility of small scale rewiring based on neurogenesis. The results already published are promising, although in general the presented project is a daunting task and has a high risk of failure[10]. Such an experiment would also be an invaluable opportunity

[10]Hardly anybody believes that such simulations will lead us to strong AI in immediate future,

to study the connectivity of the brain in a global setup without the restrictions of the empirical approach.

but they could be extremely valuable from the medical point of view and teach as a lot about brain disorders and related issues.

Chapter 7

Conclusions

In this thesis we briefly discussed the current state of the art in random graph theory, in particular the recent development in scale-free and small world graphs. We examined dynamical spiking neural models and their relevance for mimicking phenomena found in biological neurons. In Section 4.1 we introduced a new spiking network model that is sufficiently simple to allow for theoretical approach, yet its dynamics is rich enough to clearly exhibit scale-free properties. The *spike flow model* provides an interesting theoretical hint linking together neurodynamics with power law graphs. Moreover, the argument in the crucial Section 4.2 providing a description of the large-size behavior of our model in terms of a winner-take-all dynamics, seems to be rather robust and independent of many specific features of the model, which makes us believe it is universal for a broad class of similar models which do not necessarily have Gaussian and or i.i.d. weights, and even admitting possible modifications to the dynamics etc. A further important issue, already signalled in the previous chapters above, is that our results are by no means contradictory to the lack of scale-free properties reported in certain real-world neural networks, e.g. negative results for C.elegans worm nervous system, whose connectivity reveals rather exponential decay – see Amaral *et al.* (2000); Koch & Laurent (1999). The point is that we only expect scale-free properties to arise in presence of sufficiently complicated information processing units, corresponding for instance to neuronal groups rather than single neurons, and exhibiting a kind of (collective) state memory. This is often not the case for individual biological neurons or their models, where only the presence of a short refractory period carries some information about the history of previous excitations which cannot be stored for later use (roughly speaking the arrival of the system at the stable rest equilibrium practically erases all information about previous activity). Things can be different however if a recurrent group of neurons is taken into account as a single computational unit. In this case the rich dynamics inside the group can develop a kind of a collective state memory we

have in mind (the phase space of the system is so complex that once thrown out of equilibrium it requires a significant amount of time to get back to the rest state). In chapter 5 (and a recent paper Piękniewski (2007)) we show a numerical experiment based on the Eugene M. Izhikevich simple spiking neuron model (Izhikevich, 2003) that supports the above claims. In brief, the experiment shows that by substituting small neuronal groups in place of single neurons, the resulting activity graph (analogous to the spike-flow graph discussed above) becomes a scale-free network, even for fairly small neuronal groups (less than 20 neurons per group). In chapter 6 we investigate possible further studies giving first results. These results did not give a clear support for our thesis and require further investigation, possibly related to the very concept of spontaneous group (or the definition of group activity combined with the methodology of obtaining the related "functional network"). These early studies also give an idea about difficulties that can be encountered with more biologically plausible models. Last but not least our results are supported by the important empirical findings of Eguíluz *et al.* (2005), which show that the functional network recorded by fMRI in the human brain is not only both scale-free and small-world, but also that the power law exponent is approximately ≈ 2, in good agreement with our theoretical and numerical results. The methodology of reconstructing the functional graph in (Eguíluz *et al.* , 2005) much resembles that used in chapter 5 (which stands in contrast to the methodology used in chapter 6, where the spikes were counted directly – which is not achievable in empirical studies with current technology) and gives important clues on further research directions, especially those related to large scale brain simulation (Izhikevich & Edelman, 2008).

Acknowledgements

The author gratefully acknowledges the support from the Polish Minister of Scientific Research and Higher Education grant N N201 385234 (2008-2010). This work was partially supported by the Marshall of Kuyavian-Pomeranian Voivodeship (Wojewdztwo Kujawsko-Pomorskie) in Poland with the funds from European Social Fund (EFS) (a part of integrated operational program for regional development, activity 2.6 – w ramach Zintegrowanego Programu Rozwoju Regionalnego ZPORR, działanie 2.6) in the form of a stipend for PhD students (*Step in the future program, first edition – Krok w przyszłość stypendia dla doktorantów I edycja.*). The author would like to greatly acknowledge the support of his advisor Dr hab. Tomasz Schreiber, and other members of the Faculty of Mathematics and Computer Science in Toruń, in particular Prof. dr hab. Wojciech Rytter for directing his attention into various graphical problems. Leszek Rybicki (the author's faculty room-mate and friend) supported this work by his 8 core Mac Pro and valuable discussions on various subjects. The staff of Department of Informatics in particular Prof. dr hab. Włodzisław Duch, Dr Norbert Jankowski and Dr Krzysztof Grąbczewski enriched the author's knowledge in various beneficial debates related to neural networks and other subjects. Special gratitude should be expressed to Dr Eugene M. Izhikevich from Neuroscience Institute in San Diego for answering a number of emails and providing some very useful pieces of his simulation code.

Bibliography

Aarts, E., & Lenstra, J.K. (eds). 1997. *Local search in combinatorial optimization*. London: John Wiley and Sons. 31

Aarts, Emile, & Korst, Jan. 1989. *Simulated annealing and boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*. New York, NY, USA: John Wiley & Sons, Inc. 2, 31, 40, 41

Abbott, L. F. 1999. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain research bulletin*, **50**(5-6), 303–304. Available from: http://dx.doi.org/10.1016/S0361-9230(99)00161-6, doi:10.1016/S0361-9230(99)00161-6. 34

Abello, James, Buchsbaum, Adam, & Westbrook, Jeffery. 1998. A functional approach to external graph algorithms. *Esa '98: Proceedings of the 6th annual european symposium on algorithms*, 332–343. Available from: http://dx.doi.org/10.1007/3-540-68530-8˙28. 14

Ackley, D., Hinton, G., & Sejnowski, T. 1985. A learning algorithm for boltzmann machines. *Cognitive science*, **9**, 147–169. 31

Adrian, E. D. 1926. The impulses produced by sensory nerve endings. *The journal of physiology*, **61**, 49–72. 25

Adrian, E.D. 1928. *At the basis of sensation*. Christopher (London). 25

Aiello, William, Chung, Fan, & Lu, Linyuan. 2000. A random graph model for massive graphs. *Pages 171–180 of: Stoc '00: Proceedings of the thirty-second annual acm symposium on theory of computing*. New York, NY, USA: ACM. doi:http://doi.acm.org/10.1145/335305.335326. 14

Albert, Réka, & Barabási, Albert-László. 2002. Statistical mechanics of complex networks. *Reviews of modern physics*, January, 47–97. 1, 9, 14, 16, 53

Albert, Réka, Jeong, Hawoong, & Barabási, Albert-László. 1999. Diameter of the world-wide web. *Science*, **401**(Septmeber), 130–131. 1, 13

BIBLIOGRAPHY

Alon, Noga. 1986. Eigenvalues and expanders. *Combinatorica*, **6**(2), 83–96. Available from: http://portal.acm.org/citation.cfm?id=18498. 21

Alon, Noga, & Milman, Vitali D. 1985. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of combinatorial theory. series b*, **38**(1), 73–88. 21

Amaral, L. A., Scala, A., Barthelemy, M., & Stanley, H. E. 2000. Classes of small-world networks. *Proc natl acad sci u s a*, **97**(21), 11149–11152. Available from: http://dx.doi.org/10.1073/pnas.200327197, doi:10.1073/pnas.200327197. 2, 19, 83

Barabási, Albert-László, & Albert, Réka. 1999. Emergence of scaling in random networks. *Science*, October, 509–512. 1, 13, 14

Barabási, Albert-László, & Oltvai, Zoltan N. 2004. Network biology: Understanding the cell's functional organization. *Nature reviews genetics*, February, 101–113. Available from: http://www.nd.edu/~networks/Publication%20Categories/01%20Review%20Articles/NetworkBio˙Nature%20Rev%20Genetics%205,%20101-113%20(2004).pdf. x, 61

Barabási, Albert-László, Réka, Albert, & Hawoong, Jeong. 2000. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica a: Statistical mechanics and its applications*, **281**(1-4), 69–77. Available from: http://dx.doi.org/10.1016/S0378-4371(00)00018-2, doi:10.1016/S0378-4371(00)00018-2. 16

Barabási, Albert-László, Jeong, Hawoong, Néda, Zoltan, Ravasz, Erzsebet, Schubert, A., & Vicsek, Tamas. 2002. Evolution of the social network of scientific collaborations. *Physica a*, **311**(4), 590–614. 1, 13

Bhalla, Upinder S., & Iyengar, Ravi. 1999. Emergent properties of networks of biological signaling pathways. *Science*, **283**(5400), 381–387. Available from: http://dx.doi.org/10.1126/science.283.5400.381, doi:10.1126/science.283.5400.381. 14

Blum, Avrim, & Rivest, Ronald L. 1993. Training a 3-node neural network is NP-complete. *Pages 9–28 of: Machine learning: From theory to applications*. Available from: citeseer.ist.psu.edu/blum92training.html. 30

Bollobas, B. 2001. *Random graphs*. Cambridge University Press. 6

Bollobas, Bela. 1998. *Modern graph theory*. Springer. Available from: http://www.msci.memphis.edu/faculty/bollobasb.html. 6, 21

Broder, Andrei, Kumar, Ravi, Maghoul, Farzin, Raghavan, Prabhakar, Rajagopalan, Sridhar, Stata, Raymie, Tomkins, Andrew, & Wiener, Janet. 2000. Graph structure in the web. *Comput. netw.*, **33**(1-6), 309–320. doi:http://dx.doi.org/10.1016/S1389-1286(00)00083-9. 16

Chung, Fan, & Lu, Linyuan. 2006. *Complex graphs and networks (cbms regional conference series in mathematics)*. Boston, MA, USA: American Mathematical Society. 8, 9, 10, 11, 17, 18

Chung, Fan R. K. 1997. *Spectral graph theory (cbms regional conference series in mathematics, no. 92) (cbms regional conference series in mathematics)*. American Mathematical Society. Available from: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20"&path=ASIN/0821803158. 21

Chung, Fan R. K., Lu, Linyuan, Dewey, T. Gregory, & Galas, David J. 2003. Duplication models for biological networks. *Journal of computational biology*, **10**(5), 677–687. 18

Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals, and systems*, **2**, 303–314. 30

Czerwinski, Reiner. 2007 (Nov). *A polynominal time algorithm for graph isomorphism*. Available from: http://arxiv.org/abs/0711.2010, arXiv:0711.2010. 20

Dayan, Peter, & Abbott, L. F. 2001. *Theoretical neuroscience*. Cambridge, Mass.: MIT Press. 67

Durrett, Rick. 2007. *Random graph dynamics*. Cambridge: Cambridge University Press. Available from: http://www.math.cornell.edu/~durrett/RGD/RGD.html. 6

Eckhorn, R., Reitboeck, H. J., Arndt, M., & Dicke, P. 1990. Feature linking via synchronization among distributed assemblies: Simulations of results from cat visual cortex. *Neural computation*, **2**(3), 293–307. Available from: http://www.mitpressjournals.org/doi/abs/10.1162/neco.1990.2.3.293, arXiv:http://www.mitpressjournals.org/doi/pdf/10.1162/neco.1990.2.3.293, doi:10.1162/neco.1990.2.3.293. 35

BIBLIOGRAPHY

Eguíluz, V. M., Chialvo, D. R., Cecchi, G. A., Baliki, M., & Apkarian, A. V. 2005. Scale-free brain functional networks. *Phys rev lett*, **94**(1). Available from: http://view.ncbi.nlm.nih.gov/pubmed/15698136. 2, 19, 39, 54, 63, 84

Erdős, Paul, & Rényi, Alfred. 1959. On random graphs. i. *Publicationes mathematicae*, **6**, 290–297. 6, 11

Erdős, Paul, & Rényi, Alfred. 1960. The evolution of random graphs. *Magyar tud. akad. mat. kutató int. közl.*, **5**, 17–61. 6

FitzHugh, R. 1961. Impulses and physiological states in models of nerve membrane. *Biophysics journal*, **1**, 445–466. 36

FitzHugh, R. 1969. Mathematical models of excitation and propagation in theoretical models of nerve. *Pages 1–85 of:* Schwan, Herman P., *et al.* (eds), *Biological engineering*. Inter-university electronic series, vol. 9. New York, NY, USA: McGraw-Hill. 36

FitzHugh, Richard. 1955. Mathematical models of threshold phenomena in the nerve membrane. *Bulletin of mathematical biology*, **17**(4), 257–278. Available from: http://dx.doi.org/10.1007/BF02477753. 36

French, Andrew S., & Stein, Richard B. 1970. A flexible neural analog using integrated circuits. *Biomedical engineering, ieee transactions on*, **BME-17**(3), 248–253. doi:10.1109/TBME.1970.4502739. 35

Füredi, Z., & Komlós, J. 1981. The eigenvalues of random symmetric matrices. *Combinatorica*, **1**(3), 233–241. 21

Gallant, S.I. 1990. Perceptron-based learning algorithms. *Neural networks, ieee transactions on*, **1**(2), 179–191. doi:10.1109/72.80230. 29

Gallant, Stephen I. 1993. *Neural network learning and expert systems*. Cambridge, MA, USA: MIT Press. 29

Godsil, Chris, & Royle, Gordon. 2001. *Algebraic graph theory*. Graduate Texts in Mathematics, vol. 207. Springer. 20

Hebb, Donald O. 1949. *The organization of behavior*. New York: Wiley. 29

Hinton, G. E., & Sejnowski, T. J. 1986. Learning and relearning in boltzmann machines. *Parallel distributed processing: Explorations in the microstructures of cognition*, **I and II**, 282–317. 31

Hinton, G.E., & Sejnowski, T.J. 1983. Optimal perceptual inference. *Pages 448–453 of: Proceedings of the ieee computer society conference on computer vision and pattern recognition*. Washington DC: IEEE, New York, NY. 31

Hodgkin, A. L, & Huxley, A. F. 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of physiology*, **117**, 500–544. 24

Hofstadter, Douglas R. 1984. The copycat project: An experiment in nondeterminism and creative analogies. *Massachusetts institute of technology*, **755**. Available from: http://hdl.handle.net/1721.1/5648. 31

Hopfield, J. J. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of science*, **79**(April), 2554–2558. Available from: http://adsabs.harvard.edu/cgi-bin/nph-bib˙query?bibcode=1982PNAS...79.2554H. 31

Hoppensteadt, Frank C., & Izhikevich, Eugene M. 1997. *Weakly connected neural networks*. Berlin: Springer. 38

Hornik, Kurt. 1991. Approximation capabilities of multilayer feedforward networks. *Neural netw.*, **4**(2), 251–257. doi:http://dx.doi.org/10.1016/0893-6080(91)90009-T. 30

i Cancho, Ramon Ferrer, & Solé, Ricard V. 2001. The small-world of human language. *Proceedings of the royal society of london b*, **268**(1482), 2261–2265. 1, 13

Izhikevich, Eugene M. 2003. Simple model of spiking neurons. *Ieee transactions on neural networks*, 1569– 1572. Available from: http://www.nsi.edu/users/izhikevich/publications/spikes.pdf. ix, 2, 37, 38, 54, 56, 64, 84

Izhikevich, Eugene .M. 2004. Which model to use for cortical spiking neurons? *Neural networks, ieee transactions on*, **15**(5), 1063–1070. doi:10.1109/TNN.2004.832719. 34

Izhikevich, Eugene M. 2006a. *Dynamical systems in neuroscience: The geometry of excitability and bursting*. Boston: MIT Press. 32, 33, 34, 38

Izhikevich, Eugene M. Fitzhugh-nagumo model [online]. September 2006. Available from: http://www.scholarpedia.org/article/FitzHugh-Nagumo˙model. 37

BIBLIOGRAPHY

Izhikevich, Eugene M. 2006c. Polychronization: Computation with spikes. *Neural comput.*, **18**(2), 245–282. doi:http://dx.doi.org/10.1162/089976606775093882. 76, 77

Izhikevich, Eugene M., & Desai, Niraj S. 2003. Relating STDP to BCM. *Neural comp.*, **15**(7), 1511–1523. Available from: http://neco.mitpress.org/cgi/content/abstract/15/7/1511, arXiv:http://neco.mitpress.org/cgi/reprint/15/7/1511.pdf. 69

Izhikevich, Eugene M., & Edelman, Gerald M. 2008. Large-scale model of mammalian thalamocortical systems. *Proceedings of the national academy of sciences*, February, 3593–3598. Available from: http://dx.doi.org/10.1073/pnas.0712231105, doi:10.1073/pnas.0712231105. 38, 81, 84

Izhikevich, Eugene M., Gally, Joe A., & Edelman, Gerald M. 2004. Spike-timing dynamics of neuronal groups. *Cerebral cortex*, 933– 944. Available from: http://vesicle.nsi.edu/users/izhikevich/publications/reentry.pdf. xi, 3, 54, 63, 64, 69, 73, 77, 78, 79, 80

Jeong, Hawoong, Tombor, B., Albert, Réka, Oltvai, Zoltan N., & Barabási, Albert-László. 2000. The large-scale organization of metabolic networks. *Nature*, **407**(6804), 651–653. 1, 14

Johnson, J.L., & Padgett, M.L. 1999. Pcnn models and applications. *Neural networks, ieee transactions on*, **10**(3), 480–498. doi:10.1109/72.761706. 35

Kirchhoff, Gustav. 1847. über die Auflösung der Gleichungen auf welche man bei der untersuchung der Lineare Vertheilung galvanischer Ströme gführt wird. *Ann. phys. chem.*, **72**, 497–508. 21

Koch, Christof, & Laurent, Gilles. 1999. Complexity and the Nervous System. *Science*, **284**(5411), 96–98. Available from: http://www.sciencemag.org/cgi/content/abstract/284/5411/96, arXiv:http://www.sciencemag.org/cgi/reprint/284/5411/96.pdf, doi:10.1126/science.284.5411.96. 2, 19, 83

Krinskii, VI., & Kokoz, IuM. 1973. Analysis of the equations of excitable membranes. i. reduction of the hodgkins-huxley equations to a 2d order system. *Biofizyka*, **18**(3), 506–511. 33

Kumar, Ravi, Raghavan, Prabhakar, Rajagopalan, Sridhar, & Tomkins, Andrew. 1999. Trawling the web for emerging cyber-communities. *Pages 1481–1493 of: Computer networks*. 16

Kwok, Hoi Fei, Jurica, Peter, Raffone, Antonino, & van Leeuwen, Cees. 2006. Robust emergence of small-world structure in networks of spiking neurons. *Cognitive neurodynamics*, december. Available from: http://www.springerlink.com/content/57241p34x7317641/fulltext.pdf, doi:10.1007/s11571-006-9006-5. 19, 60

Lotka, Alfred J. 1926. The frequency distribution of scientific productivity. *Journal of the washington academy of sciences*, **16**(12), 317–323. 15

Maass, W., & Markram, H. 2004. On the computational power of recurrent circuits of spiking neurons. *Journal of computer and system sciences*, **69(4)**(4), 593–616. Available from: http://www.igi.tugraz.at/maass/psfiles/135-revised-elsart.pdf. 35

Maass, W., Natschläger, T., & Markram, H. 2002. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, **14**(11), 2531–2560. Available from: http://www.igi.tugraz.at/papers/lsm-nc-130.pdf. 35

Markram, H., Wang, Y., & Tsodyks, M. 1998. Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the national academy of sciences of the united states of america*, **95**(9), 5323–5328. Available from: http://view.ncbi.nlm.nih.gov/pubmed/9560274. 65

Markram, Henry, Lübke, Joachim, Frotscher, Michael, & Sakmann, Bert. 1997. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, **275**, 213–215. Available from: http://www.sciencemag.org/cgi/content/abstract/275/5297/213. 66

McCulloch, Warren, & Pitts, Walter. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biology*, **5**(4), 115–133. Available from: http://dx.doi.org/10.1007/BF02478259, doi:10.1007/BF02478259. 29

Minsky, M., & Papert, S. 1969. *Perceptrons*. MA: MIT Press. 29

Montoya, Jose M., & V., Ricard V. Solé. 2002. Small world patterns in food webs. *Journal of theoretical biology*, **214**(3), 405–412. 1, 13

Morris, C, & Lecar, H. 1981. Voltage oscillations in the barnacle giant muscle fiber. *Biophys. j.*, **35**(1), 193–213. Available from: http://www.biophysj.org/cgi/content/abstract/35/1/193, arXiv:http://www.biophysj.org/cgi/reprint/35/1/193.pdf. 32

BIBLIOGRAPHY

Nagumo, J., Arimoto, S., & Yoshizawa, S. 1962. An active pulse transmission line simulating nerve axon. *Proceedings of the ire*, **50**(10), 2061–2070. doi:10.1109/JRPROC.1962.288235. 36

N.Davey, Christianson, B., & Adams, R. 2004 (July). High capacity associative memories and small world networks. *Pages 159–165 of: Proc. IEEE international joint conference on neural networks (IJCNN 05)*. 19

Newman, Mark, Barabasi, Albert-Laszlo, & Watts, Duncan J. 2006. *The structure and dynamics of networks: (princeton studies in complexity)*. Princeton, NJ, USA: Princeton University Press. Available from: http://press.princeton.edu/titles/8114.html. 9

Pareto, Vilfredo. 1896-1897. *Cours d'économie politique*. Rouge, Lausanne. 15, 39

Perotti, J. I., Tamarit, F. A., & Cannas, S. A. 2006. A scale-free neural network for modelling neurogenesis. *Physica a statistical mechanics and its applications*, **371**(Nov.), 71–75. doi:10.1016/j.physa.2006.04.079. 1, 19, 40

Piękniewski, F. 2005. *Analiza stabilności i diagramów fazowych dla lokalnie hopfieldowskich sieci neuronowych w obecności wysoce skorelowanych wzorców okresowych*. M.Sc. thesis, Uniwersytet Mikołaja Kopernika, Toruń. Available from: http://www-users.mat.uni.torun.pl/~philip/msc.pdf. 31

Piękniewski, F. 2006 (July). Mesoscopic approach to locally hopfield neural networks in presence of correlated patterns. *Pages 5899–5905 of: Proc. IEEE world congress in computational intelligence (WCCI 2006)*. Available from: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1716543. 19, 31

Piękniewski, F. 2007. Emergence of scale-free graphs in dynamical spiking neural networks. *Pages 755–759 of: Proc. IEEE international joint conference on neural networks*. Orlando, Florida, USA: IEEE Press. Available from: http://www.ieeexplore.ieee.org/xpl/freeabs_all.jsp?isnumber=4370891&arnumber=4371052&count=569&index=160. 2, 3, 39, 53, 84

Piękniewski, F., & Schreiber, T. 2005 (July). Phase diagrams for locally hopfield neural networks in presence of correlated patterns. *Pages 776–781 of: Proc. IEEE international joint conference on neural networks (IJCNN 06)*, vol. 2. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1555950. 31

Piękniewski, F., & Schreiber, T. 2007 (April). Emergence of scale-free spike flow graphs in recurrent neural networks. *Pages 357–362 of: Proc. IEEE symposium series in computational intelligence - foundations of computational intelligence*. Available from: http://doi.ieeecomputersociety.org/10.1109/FOCI.2007.371496. 40

Piękniewski, Filip, & Schreiber, Tomasz. 2008. Spontaneous scale-free structure of spike flow graphs in recurrent neural networks;. *Neural networks*, **In Press, Corrected Proof**, –. Available from: http://www.sciencedirect.com/science/article/B6T08-4SVKSNF-4/2/0363e0991d6ebe0fadcdf859b906018a, doi:doi:10.1016/j.neunet.2008.06.011. 2, 13, 39

Redner, S. 1998. How popular is your paper? an empirical study of the citation distribution. *European physical journal b*, **4**(2), 131–134. 1, 13

Resnick, Sidney I. 1987. *Extreme values, regular variation, and point processes*. Springer Series in Operations Research and Financial Engineering. Secaucus, NJ, USA: Springer-Verlag New York, Inc. 46

Rojas, R. 1996. *Neural networks: A systematic introduction*. Springer. Available from: http://page.mi.fu-berlin.de/rojas/neural/index.html.html. 24

Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psych. rev.*, **65**, 386–407. (Reprinted in *Neurocomputing* (MIT Press, 1988).). 28, 29

Rosenblatt, F. 1988. *The perception: a probabilistic model for information storage and organization in the brain*. Cambridge, MA, USA: MIT Press. 28, 29

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. 1986. Learning internal representations by error propagation. *Pages 318–362 of:* Rumelhart, D., & McClelland, J. (eds), *Parallel distributed processing*, vol. 1. Cambridge, MA: MIT Press. 29

Seyed-allaei, Hamed, Bianconi, Ginestra, & Marsili, Matteo. 2006. Scale-free networks with an exponent less than two. *Physical review e*, **73**, 5. Available from: doi:10.1103/PhysRevE.73.046113. 17

Simon, Herbert A. 1955. On a class of skew distribution functions. *Biometrika*, **42**, 425–440. Available from: http://dx.doi.org/doi:10.1093/biomet/42.3-4.425, doi:doi:10.1093/biomet/42.3-4.425. 15

BIBLIOGRAPHY

Stauffer, Dietrich, Aharony, Amnon, da Fontoura Costa, Luciano, & Adler, Joan. 2003. Efficient hopfield pattern recognition on a scale-free neural network. *The european physical journal b*, 395–399. doi:10.1140/epjb/e2003-00114-7. 1, 19, 40

Talagrand, Michel. 2003. *Spin glasses: A challenge for mathematicians, cavity and mean field models*. Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge / A Series of Modern Surveys in Mathematics, vol. 46. Secaucus, NJ, USA: Springer-Verlag New York, Inc. 41, 42

The, OpenMP Architecture Review Board. 2008 (August). *The openmp application program interface*. http://openmp.org/wp/. Available from: http://openmp.org/wp/. 73

Toroczkai, Zoltán, & Guclu, Hasan. 2007. Proximity networks and epidemics. *Physica a: Statistical mechanics and its applications*, **378**(1), 68–75. Available from: http://www.sciencedirect.com/science/article/B6TVG-4MP5JGV-1/1/0b2f4809a1396b21d1c8c1827abfb9c4. 11

Travers, Jeffrey, & Milgram, Stanley. 1969. An experimental study of the small world problem. *Sociometry*, **32**(4), 425–443. Available from: http://www.jstor.org/stable/2786545. 11

Watts, Duncan J., & Strogatz, Steven H. 1998. Collective dynamics of small-world networks. *Nature*, June, 440–442. 13, 22

Werbos, P. 1974. *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. Ph.D. thesis, Harvard. 29, 30

Widrow, B., & M.E. Hoff, Jr. 1960. Adaptive switching circuits. *1960 ire western electric show and convention record*, August, 96–104. 29

Wigner, Eugene P. 1955. Characteristic vectors of bordered matrices with infinite dimensions. *Annals of mathematics*, **62**(3), 548–564. Available from: http://www.jstor.org/view/0003486x/di961748/96p0020o/0. 21

Wigner, Eugene P. 1957. Characteristic vectors of bordered matrices with infinite dimensions ii. *Annals of mathematics*, **65**(2), 203–207. Available from: http://www.jstor.org/view/0003486x/di961756/96p00014/0. 21

Wigner, Eugene P. 1958. On the distribution of the roots of certain symmetric matrices. *Annals of mathematics*, **67**(2), 325–327. Available from: http://www.jstor.org/view/0003486x/di961762/96p00057/0. 21

Zahradník, Miloš. 1984. An alternate version of pirogov-sinai theory. *Communications in mathematical physics*, **93**(4), 559–581. Available from: http://dx.doi.org/10.1007/BF01212295. 31

Zipf, GK. 1932. Selective studies and the principle of relative frequency in language. *Pages 3544–3557 of: Icslp96: Fourth international conference on spoken language processing*. 15

Index

Albert, Réka, 14

Back-propagation, 19, 29, 30
Barabási, Albert-László, 14
Boltzmann machine, iii, 2, 31, 40, 41
brain, 18

C.elegans, 2, 18, 83
cable equation, 28
clustering coefficient, 10, 13, 22, 58, 60, 61
combinatorial optimization, 31

degree distribution, iii, 1, 10, 11, 13, 14, 42, 48, 56, 58, 59
dynamics, 2
    stochastic, 31, 39, 40
    winner-take-all, 2, 39, 40

Erdős, Paul, 6

FitzHugh-Nagumo model, 36

Galton-Watson process, 7
giant component, 6, 8–10
graph, 5
    empirical, 2, 9–11, 14, 16, 22
    random, 6

Hebb, Donald, 29
Hodgkin-Huxley model, 24, 25, 37
Hopfield
    ,John, 31
    network, 19, 31

hub, 10

integrate and fire, 34
Izhikevich, Eugene M., 37, 76, 84

Kirchhoff, Gustav, 21

Laplacian, 21
Liquid state machine, 35
local structure, 9, 11, 13

McCulloch-Pitts model, 29
membrane, 25
Milgram, Stanley, 11
multi compartment model, 28

network, 1
    biochemical, 17
    scale-free, 1, 11, 14
neuromime, 35
neuron, 2, 18, 19, 23, 24, 28–30, 34, 35, 37, 39, 42–46, 48, 55, 57, 64–69, 71, 72, 74, 75, 77–79, 83, 84
nullcline, 32, 36, 37
numerical stability, 72

parallelization, 73
Pareto, Vilfredo, 15
PCNN, 19, 35
Perceptron, 19, 29, 30
percolation, 11
power law, 1, 10, 11, 13, 15
preferential attachment, 1, 14, 19, 53

Rényi, Alfréd, 6
rewiring, 13
Rosenblatt, Frank, 29

Simon, Herbert A., 16
simulated annealing, 31
small world, 10, 11
spectrum, 20
spike flow graph, 39, 49
spike-flow graph, 2
spiking
    network, iii, 2, 53, 54
    neuron, 2, 3, 32, 34, 37, 40, 54, 64, 75
    regime, 69, 71, 76
spin glass, 31
Strogatz, Steven H., 13

unicyclic component, 9

Watts, Duncan J. , 13

Zipfs law, 15