An Introduction to the Study of Brain

Zhengzhong Liang

ECE 521

2016.11.16

Instructor: Miklos N. Szilagyi

Content

1, Introduction

1.1 The Overview of This Dissertation

Brain is one of the most fascinating parts of our body. Its complicated structure enables it to achieve amazing cognitive tasks and produce complex behavior. Since brain is so complicated, starting to study brain is not an easy task. There are too many relevant theories, and one is easy to be overwhelmed by the large amount of information.

This is the reason why I write this dissertation. It can serve as an introduction to the study of brain. In this chapter, I will begin with trying to tidy up the entangled disciplines of brain study. In later chapters, some selected topics about brain are showed and discussed. These topics range from psychology to computer science, covering a wide range of topics. As I have said, I hope this dissertation can serve as an introduction to students who are interested in brain study and want to start doing some work. After reading this dissertation, I think reader can establish a rather clear envisage of brain and relevant topics.

1.2 Outline of Academic Disciplines

If we look up “academic discipline” on the Internet, a wiki page will be showed to us, in which almost all of the academic disciplines that human create are listed [1]. All academic disciplines are broadly divided into 5 base categories: arts, humanities, social sciences, sciences and technology. Within each base category, there are many sub-disciplines. For example, arts consist of performing arts and visual arts. The righter column are the more detailed fields in the discipline. For example, within sub-discipline “Languages and Literature”, there is one filed called linguistics.

Table 1.1: Outline of Academic Discipline

Arts	Performing arts		
	Visual arts		
Humanities	Geography		
	History		
	Languages and Literature	linguistics	
	Philosophy		
Social sciences	Economics		
	Law		
	Political science		
	Psychology	neuropsychology	
	Sociology		
Sciences	Biology	neuroscience	
		physiology	
	Chemistry	neurochemistry	
	Earth and space sciences		
	Mathematics	dynamic system	
		information theory	
		graph theory	
	Physics		

Technology	Agronomy		
	Computer science	artificial intelligence	cognitive science
		computing	computational neuroscience
	Engineering	system science	dynamic system
			network theory
			system neuroscience
			other
	Medicine	neurology	
		neurosurgery	

As we can see, topics about brain exist in all 5 base categories. Table 1.1 is a good illustration of how fascinating brain is. The mysterious brain attracts scholars from largely variated fields to study it. Also, it is a good illustration that shows how difficult it is to figure out all problems about brain. Since the study of brain covers a wide range of disciplines, it is impossible for one person to master all knowledge of brain.

1.3 Brain and Relevant Discipline

Although scholars from variated fields all study brain, their perspectives and focuses are different.

1.3.1 Psychology [2]

A major topic in psychology is cognitive psychology. This subject studies human's cognitive behavior from psychological perspective. Some classical topic in this filed includes the human's memory, learning and reasoning. Pure psychological studies usually do not analyze human's nervous structure. They only focus on the manifestation of human's cognitive activity. Another topic in psychology about brain is neuropsychology [3]. This subject deals with the functional and structure of brain.

1.3.2 Biology

A main topic about brain in biology is neuroscience [4]. Traditional neuroscience is considered to be a sub-field of biology. However, now it is an inter-discipline field, which shares theories from mathematics, computer science and engineering. Traditional neuroscience handles problems about the anatomical structure of nervous system, including brain. Another filed in this discipline is physiology [5]. It concentrates on how the organs and cells work to fulfill the functional activity.

1.3.3 Chemistry

Neurochemistry [6] studies the chemical materials in nervous system. These materials include neurotransmitter and other molecules.

1.3.4 Mathematics

Many theories in mathematics can be used to interpret the discoveries about brain from other disciplines. For example, dynamic system theory is used for modeling neuron activity. Recently, graph theory is also applied to studying the functional parts of brain. Using graph theory, scientists are able to link brain's functional structure and anatomical structure.

1.3.5 Computer Science

Computer science is playing a more and more important role in the study of brain. Previously, computer scientists concentrated more on artificial intelligence. Many artificial intelligence models are inspired by the discoveries of brain. However, recently, computer technology is also used to study brain itself. Using computer, scientists are able to simulate large-scale brain activity. Through simulation, scientists can understand brain better.

1.3.6 Engineering

Engineers try to interpret brain from a system perspective. Scholars in this field try to map how individual areas of brain affect the whole brain's function. Theories about dynamic system, graph theory and system neuroscience are used.

1.3.7 Medicine

Study about brain in this field deals with clinical problems about brain. For example, neurology is concerned about the disorder in human's nervous system [7]. Similarly, neurosurgery deals with surgery problems about human's nervous system [8].

1.4 Inter-discipline Trends

One trend in the study of brain is the growth of inter-discipline research. The most typical inter-discipline form is to combine science and technology. This is shown in figure 1.1. The emergence of computational neuroscience is a perfect example [9]. Traditional neuroscience deals with problem about the biological phenomenon and facts. Then mathematical theories are applied to model those facts and phenomenon quantitatively. Recently, due to the great progress in computer science and engineering, technology in this field is also applied to the study of brain. One major consequence is the emergence of computational neuroscience.

According to Wikipedia, computational neuroscience is a discipline that deals with the function of brain in terms of information processing. Major topics include sensory processing, memory and synaptic plasticity and other relevant topics. As we have seen, enormous computational power of computer enables scientists to build realistic neural network in a large scale. We may expect that more and more important discoveries about brain will be introduced by research of computational neuroscience.

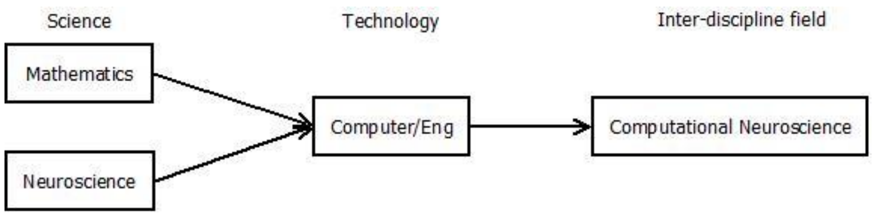


Figure 1.1 Inter-Discipline Trend of Brain Study

2, Neuroscience Basics

In this chapter we will concentrate on the basic neuroscience facts about nervous system and brain. These facts are foundation for our further discussion. Most of the content in this chapter is from Eric Kandel's book "Principles of Neural Science" [10 11].

2.1 Composition of Human's Nervous System

Human's neural system is an extremely complicated system, consisting of multiple sub-systems. "Brain" is only part of this amazing system. Furthermore, when we talk about brain, we are actually referring to a bunch of organs and structures. Thus, before we can delve deeper into the properties of brain, it is necessary to get to know the overview of human's neural system. The composition of human's central nervous system is shown in table 2.1.

Table 2.1 Composition of Human's Central Nervous System

Category	Sub-system	Structures in sub-system
Central Nervous System	Cerebral hemisphere	Cerebral cortex
		Basal ganglia
		Hippocampus
		Amygdala nuclei
	Corpus callosum	
	Diencephalon	Thalamus
		Hypothalamus
	Brain stem	Mid brain
		Pons
		Medulla oblongata
	Cerebellum	
	Spinal cord	Cervical
		Thoracic
		Lumbar
		Sacral regions

Human's nervous system consists of two major systems, the central nervous system and the peripheral nervous system. The table above shows the sub-systems and components of human's central nervous system. Human's central nervous systems can be divided into 6 parts, cerebral hemisphere, corpus callosum, diencephalon, brain stem, cerebellum and spinal cord. These sub-systems are called central nervous systems because they integrate information from all parts of human's body. In contrast, peripheral nervous system is responsible for connecting central nervous system with organs. Since brain does not belong to peripheral nervous system, we do not include peripheral nervous system in the table.

As you can see, the term "brain" does not show up in this list, indicating that "brain" is not a scientific terminology. In most cases, the term "brain" refers to cerebral hemisphere, cerebellum and brain stem, which are the most important parts of human's neural system.

2.2 Functions of the Structures in Human's Central Nervous System

Among the structures of brain, the cerebral cortex is the most complicated structure. Also, it is the foundation of human's cognitive function. These functions include memory, learning, language, reasoning, sensing and other cognitive functions. Thus we will introduce cerebral cortex in the next section. In this section will briefly discuss the functions of other structures. These structures in central nervous system are also relevant to some cognitive functions, although they may not be as important as cerebral cortex.

Basal ganglia are composed of multiple subcortical nuclei, and it is connected with cerebral cortex, thalamus and brain stem. It is believed that basal ganglia are relevant to human's learning, routine behaviors, emotion and movement control. Because disease in basal ganglia could cause disorder in these functions. Hippocampus also lies under cerebral cortex. Hippocampus is important to the information integrating from short-term memory to long-term memory. Amygdala nuclei is crucial to the processing of memory, decision-making and emotional reactions.

As you can see, the structures in cerebral hemisphere are relevant to brain's senior function. In contrast, the structures outside of cerebral hemisphere are more related to brain's primary functions. For example, diencephalon, consisting of thalamus and hypothalamus, are crucial to human's visceral activities and autonomic nervous system. Brainstem plays an important role in controlling human's cardiovascular system, respiratory, pain and awareness. Cerebellum is crucial to human's motor control. And finally, the spinal cord is responsible for transmitting the signals between central nervous system and peripheral nervous system.

2.3 Cerebral Cortex

Cerebral cortex should be discussed separately because it is itself complicated enough. Cerebral cortex is the tissue on the outside of cerebral hemisphere, covered by a layer of gray matter. Many cognitive functions can be localized to cerebral cortex.

Anatomically, cerebral cortex is divided into for lobes—frontal lobe, parietal lobe, occipital lobe and temporal lobe. This classification is based on the anatomical variation of these four areas. A sketch of the lobes of the brain is shown in the figure 2.1.

The function of these 4 areas are also distinct. The frontal lobe highly relevant to planning future and controlling movement. The parietal lobe is crucial to somatic sensation. The temporal lobe is responsible for hearing and the occipital lobe is essential to vision.

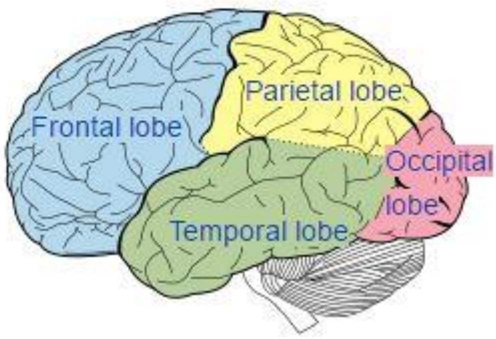


Figure 2.1 Lobes of Brain

The four lobes can be further divided into more functional areas. According to Korbinian Brodmann, cerebral cortex can be further divided into about 50 areas. He thought that every individual area of the cerebral cortex is responsible for some specific tasks. For example, in the following picture, blue areas are relevant to vison and brown areas are relevant to olfactory sensation. However, this theory is challenged by some recent discoveries in neuroscience. These discoveries show that some certain behaviors of human beings are not undertook by a specific area of brain. Instead, they are a result of the cooperation of many different function areas of brain. Even a single activity is accomplished by a bunch of elementary neural activities. Each elementary neural activity may be concerned with only one area, whereas the whole cognitive activity may encompass multiple areas. Information from different brain areas are integrated and processed together. The functional areas of brain are shown in figure 2.2.

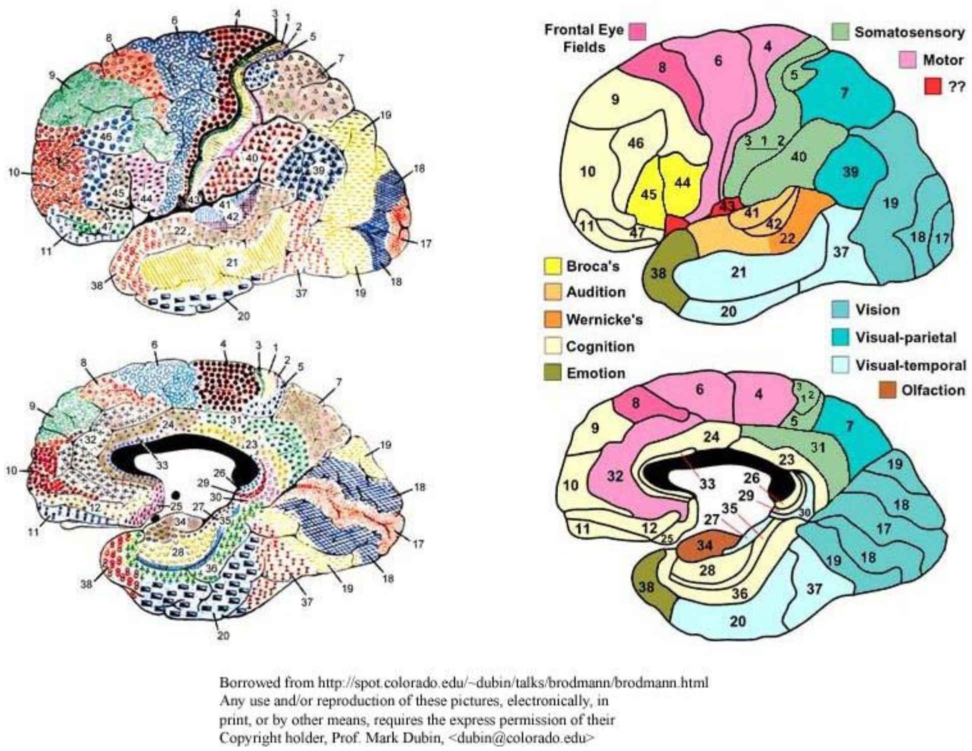


Figure 2.2 Functional Areas of Brain

2.4 Neurons of Brain

Brain contains about 100 billion nerve cells, and these cells belong to 1000 different types. However, all these types can be categorized into 2 classes. The first class is called nerve cell, which is also called “neuron”. Nerve cells are the cells that receive, process and send signals. They are cells which process information in our brain. The other class of cell is called glial cells. Glial cells do not participate in the processing of information. Instead, they are merely supportive cells. For example, the glial cells form the spatial structure of brain. The glial cells may be also relevant to nutritive functions.

Nerve cells can be further divided into unipolar neuron, bi-polar neuron and multipolar neuron. Figure 2.3 shows this classification [12]. Multipolar neurons distribute widely in human’s nervous system, from central nervous system or peripheral nervous system. There are many multipolar neurons in human’s cerebral cortex. They are very important to the information processing of brain. Bipolar neurons serve as signal transmitter. For example, the retinal cell connects receptor and other neurons. Usually, receptors in retina can detect light and color of object, then these signals are transmitted to retinal neuron. The signals will be transmitted by bi-polar neuron to neurons in cerebral cortex. Unipolar neurons are usually found in human's sensory system. For example, the touch sensation on skin is transmitted by unipolar neuron. The function of anaxonic neuron is not very clear [13]. Further study about this type of neuron maybe carried out in the future.

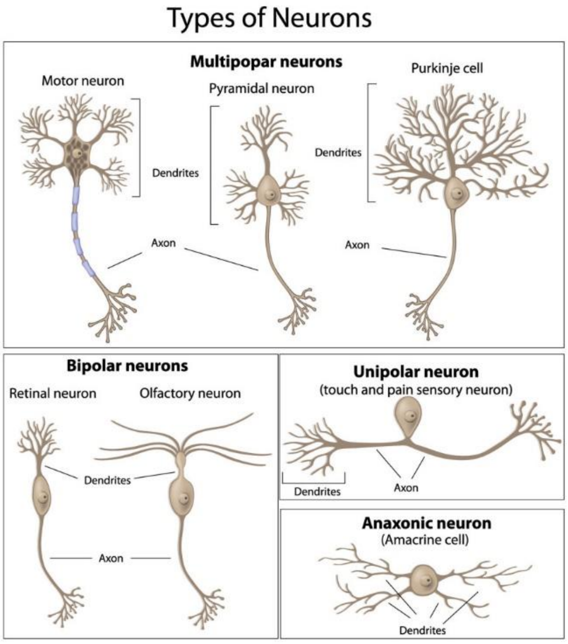


Figure 2.3 Types of Cells in Nervous System

3, Neuron Dynamics and Mathematical Models

3.1 Overview

In chapter 1 and 2, we cover some anatomical facts of brain as well as human's nervous system. Those facts help us to understand the most fundamental aspects of human's neural system. However, using mere that kind of knowledge, we are still far from explaining how brain works.

In all, human's neural system consists of neurons. Thus, figuring out how every individual neuron works would largely help us to reveal brain's mechanism.

In this chapter, we will first discuss some properties of neuron and synapse. For example, the phenomenon of resting potential and action potential. However, we will not delve into the details and rationales about how these properties emerges. The details are biological topics, thus they will be just mentioned instead of discussed thoroughly in this dissertation.

Our concentration will be the properties and the mathematical descriptions of these properties. We will start from the mathematical model of a single neuron, then move to the model of synapse. In later chapters, the discussion about a small-scale neural network will also be included.

3.2 Neuron Dynamics-Anatomy and Physiology of Single Neurons

3.2.1 Anatomical Structure

A single neuron contains 3 major parts: soma, dendrites and axons. Soma is the body of a neuron, also where its nucleus lies. Dendrites are small branches which grow from the soma and reach other neurons' soma or axons. An axon is a long branch produced by the soma and reach to other neurons' somas.

Except the 3 major parts, another microstructure of neurons is also essential to the signal transmission in neurons. That is synapse. As is shown in Figure 3.1, synapse is a structure which is between the presynaptic neuron and the postsynaptic neuron. It can transmit information from the former neuron to the latter one. Different from how signals are transmitted within a neuron, the signals between neurons are transmitted by chemical material. When there is a signal in the presynaptic neuron to be transmitted to a latter neuron, the synapse of presynaptic neuron can release neurotransmitter to the postsynaptic neuron. After the postsynaptic neuron receives the neurotransmitter, it can produce electrical signal, which can be transmitted along the axon of the postsynaptic neuron and transmitted to another neuron.

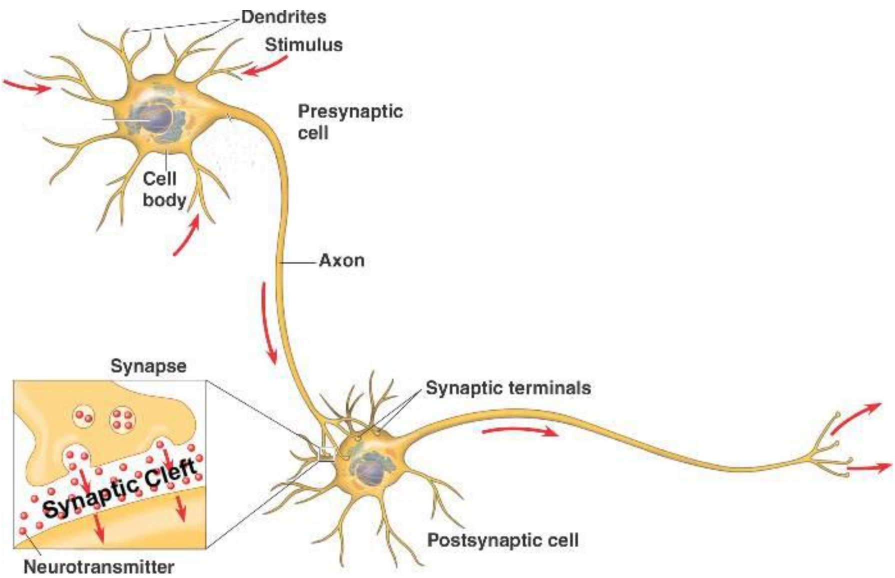


Figure 3.1 Two Neurons

3.2.2 Resting Potential and Action Potential

When the neuron is not stimulated, the inner part of a neuron is at a resting potential, which is about -65 mv comparted with the outer environment. Depending on the types of neuron, this value can fluctuate a little. Three types of ions can account for the resting voltage and action potential. The functions of these ions and channels are thoroughly discussed in Eric Kandel's book. The underlying mechanism of the whole process how neurons maintain resting potential or action potential is pretty complicated. They will not be discussed in this dissertation.

When the neuron is stimulated, a neuron may produce action potential [14 15]. A perfect illustration of how a neuron can be stimulated is figure 3.2. The left panel shows how a neuron can be stimulated. The stimulus comes from dendrites. Stimulus signal is current, usually several Nano-Ampere. The stimulus can come from different neurons. In this figure, stimulus comes from 4 sources. When the stimulus reaches neuron soma, a potential will be excited. However, this potential is not action potential yet. Only when the excited potential reaches a value (which is called a threshold), the action potential can be produced.

This process is illustrated in the right panel of figure 3.2. The resting potential is about -65mv. From time 0, a stimulus is placed to neuron, and a minor potential is excited. When this excited potential reaches -50mv, action potential is produced. Action potential can only maintain for a very short period of time. Almost immediately, the action potential ends and neuron enters the refractory period, during which its potential drops below resting potential and then returns to resting potential again, waiting for producing next action potential.

The process of producing action potential is called depolarization. And in contrast, the process when action potential returns to resting potential is called repolarization. Action potential can travel along the axon of neuron. The speed is about 100m/s. In this way neurons can transmit signal to another neuron.

The way neuron works is very likely to Moose code and today's Internet. In these system, information is represented by either 0 or 1. There may be some underlying relationship among these 3 systems, but unfortunately I do not have time to cover this.

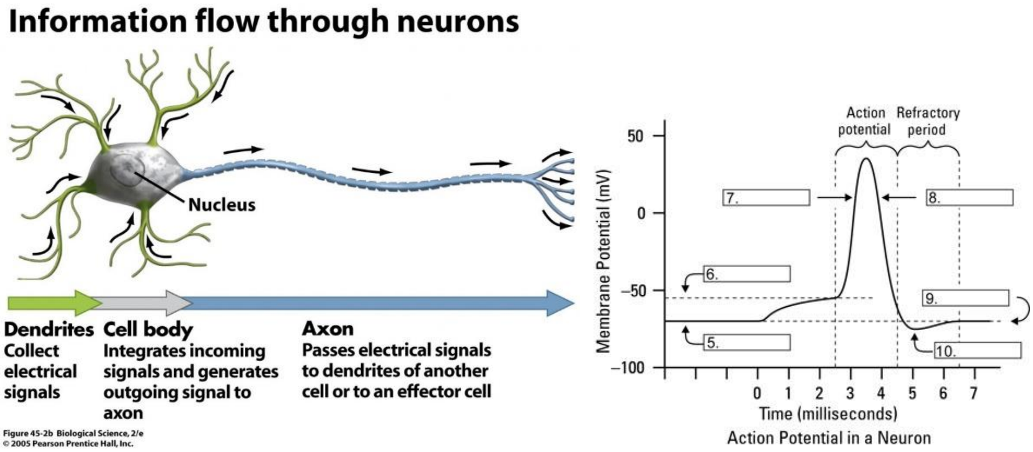


Figure 3.2 Action Potential in Neurons

3.3 Mathematical Models of Single Neuron

Scholars have always been trying to formulate the activity of neurons using mathematical equations. As the research of human's neural system proceeds, more and more mathematical models are proposed. These models have their own focus. Some models are intended to describe the physiological phenomenon of neurons as accurately as possible, while others are dedicated to reducing the complexity of neuron models and only retain the most essential features of neurons. This section will introduce three types of models, which have totally different aims. Then there will be a comparison of the models introduced.

3.3.1 Leaky Integrate-and-fire Model

Integrate-and-fire model was proposed by Louis Lapicque in 1907 [16]. Lapicque did not try to consider the ions and pumps in the membrane of neuron. Instead, he observed how the neuron as a whole performs under stimulus. Then he devised a model which can roughly describes the activity of neuron.

Another idea behind this model is that, although the shapes of the action potential of different neurons are different, the difference is so unobvious that we can ignore the difference and use a general model to describe the common feature of these neurons. So in this model, different information is not distinguished by different shapes of action potential. Instead, it is distinguished by the time at which action potential fires.

A sketch of a Leaky Integrate-and-Fire model is shown in figure 3.3 [17 18]. The upper space in this figure represent the outside of neuron, and the lower space of this figure denotes the inner part of neuron. Outer part and inner part is separated by neuron membrane, which consists of membrane resistance, membrane capacitor, resting potential and gate voltage in this figure. The current source on the left represents stimulus, which comes from the dendrites of the neuron.

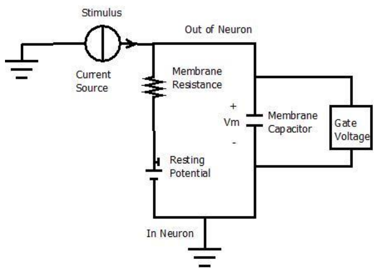


Figure 3.3 Equivalent Circuit of LIF Model

When there are no signals from other neurons, there will be no spikes produced from this neuron. There is no current in this circuit. So the membrane potential V_m should be around -70 mV, which is the value of "Resting Potential" in this figure. When there are signals from other neurons, the dendrites of this neuron, which in this figure is the current source, can produce current. The current flows throw the membrane and charges the capacitor. When the voltage of this capacitor reaches a "Gate Voltage", the neuron spikes.

The following equation represents the voltage of capacitor. In this equation, u(t) is the voltage of capacitor, it is a function of time. U_{rest} is the resting potential of neuron, which is about -65 mv. R is the membrane resistance of neuron and I is the stimulus current.

$$u(t) = u_{\text{rest}} + R\, I_0 \left[1 - \exp\left(-\frac{t}{\tau_m}\right)\right] .$$

Why is this model called integrate-and-fire model? The term "integrate" and "fire" have specific meaning. "Integrate" represents that stimulus current charges the capacitor before firing. The voltage of capacitor can be obtained by integrate current over time. "fire" represent that when the voltage of capacitor reaches a threshold, this model will fire (produces action potential).

The simulation of Integrate-and-fire model is performed on computer. This simulation program is written in python, and relevant code is attached in appendix. Simulation tests the performance of model under different stimulus (figure 3.4). In left top one, the stimulus is negative, so the membrane potential is below resting potential. This is the situation where neuron is inhibited. In right top, a positive stimulus is placed, The amplitude is 10 uA. Compared with the bottom one, in which the stimulus is 20 uA, the spikes in the right top is sparser.

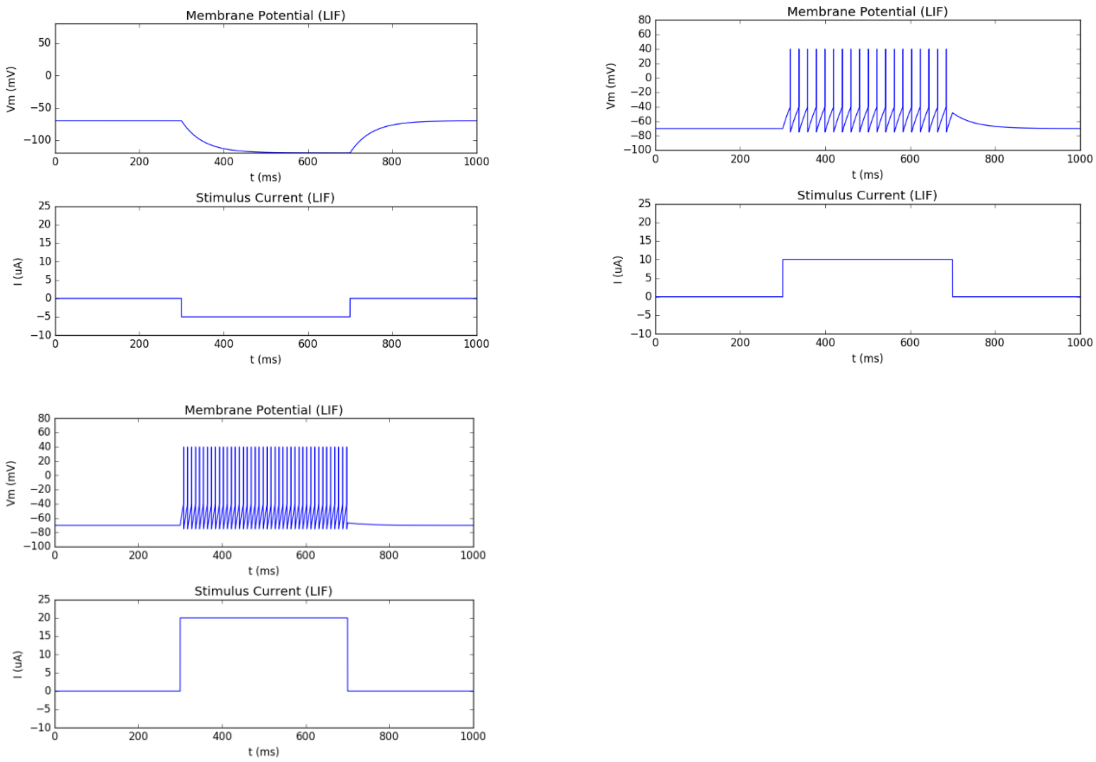


Figure 3.4 Simulation of LIF Model

3.3.2 Hudgkin-Huxley Model

Hudgkin-Huxley model was proposed by Alan Lloyd Hodgkin and Andrew Fielding Huxley in 1952 [19]. The ideology of this model is to describe the behavior of the components on neuron's membrane. Because it is these components on neuron's membrane that account for the behavior of a neuron. From a mathematical perspective, this model uses a set of non-linear differential equations to depict these components.

Although Hudgkin-Huxley model was proposed more than 60 years ago, it is still widely used by many scholars all over the world today, because it can perfectly simulate a vast range of behaviors of neurons. There are more than 1000 types of neurons in our body, and these neurons have different properties and behaviors. Generally, these behaviors can be divided into 20 categories. Izhikevich in 2004 compared 11 different neuron models which are popular in various field. He chose 22 types of behavior can studied whether these models can simulate those different behaviors. It turns out that Hudgkin-Huxley model can simulate 19 behaviors out of all 22 behaviors. It is obvious that choosing this model enables researchers to study many essential characteristics of neurons.

The circuit of hudgkin-huxley model is shown below. C stands for the membrane capacitor.

Three types of ion can influence the membrane potential, they are K^+, Na^+ and other negative ions (mainly Cl^-). The ion channel of these three types of ion are included in this model. The voltage source in figure 3.5 shows the ion concentration of the ions. For example, the concentration of Na^+ is higher outside the membrane, so the direction of voltage source is upwards. The resistor shows the resistance that the ions will encounter when they are traversing membrane. It should be pointed that the resistance of K^+ and Na^+ channels is not a constant. The resistance depends on the current membrane potential.

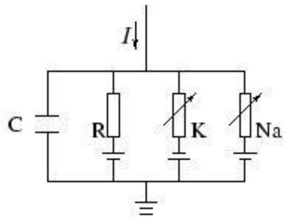


Figure 3.5 Equivalent Circuit of Hudgkin-Huxley Model

The dynamics of Hudgkin-Huxley model can be described by a set of non-linear differential equations. $\mathrm{m}, \mathrm{n}, \mathrm{h}$ are coefficients which describe the electrical property of ion channels. They also depend on the value of current membrane potential.

$$I(t) = I_C(t) + \sum_k I_k(t)$$

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = -\sum_k I_k(t) + I(t).$$

$$\sum_k I_k = g_{Na}\, m^3 h\,(u - E_{Na}) + g_K\, n^4\,(u - E_K) + g_L\,(u - E_L).$$

$$\dot{m} = \alpha_m(u)\,(1 - m) - \beta_m(u)\,m$$

$$\dot{n} = \alpha_n(u)\,(1 - n) - \beta_n(u)\,n$$

$$\dot{h} = \alpha_h(u)\,(1 - h) - \beta_h(u)\,h$$

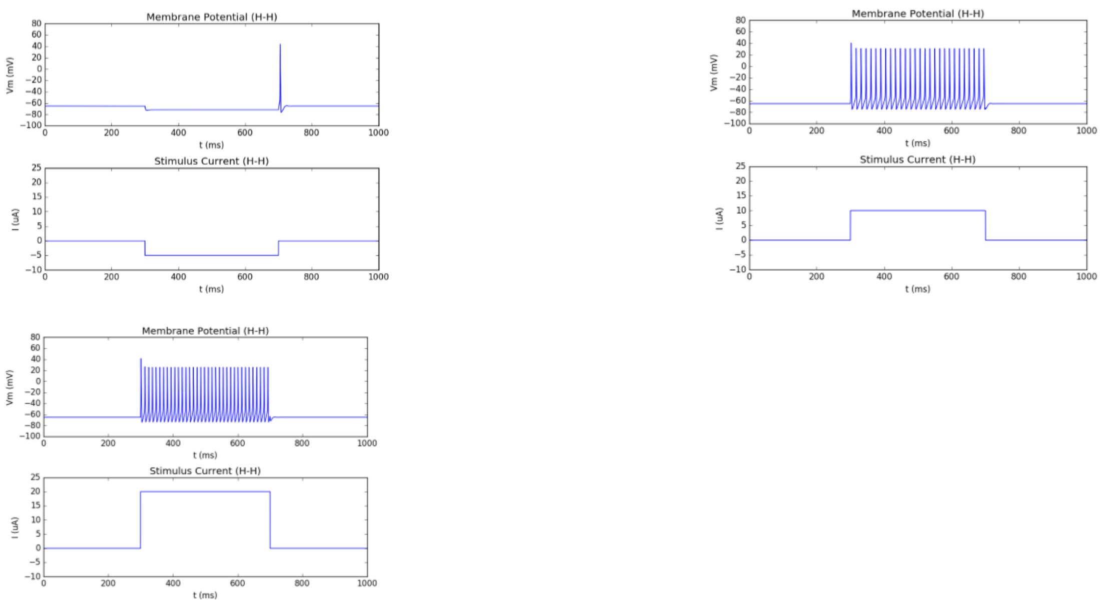


Figure 3.6 Simulation of Hudgkin-Huxley Model

It is a set of pretty complicated non-linear differential equation, which makes it hard to solve the answer manually. Luckily, it is not so hard to run a simulation program on computer. Figure 3.6 shows the behavior of neuron activity using this model. This model can perfectly simulate the inhibition and excitation of a neuron. Also, if stimulus is enlarged, the fire rate of neuron will increase, which is a pretty realistic property.

3.3.3 Izhikevich Model

Izhikevich in 2003 proposed an amazing spiking neuron model [20]. This model is also described by differential equation. However, differential equations are much simpler than Hudgkin-Huxley model.

$$
\begin{aligned}
&v' = 0.04v^2 + 5v + 140 - u + I \\
&u' = a(bv - u) \\
&\text{if } v \geq 30\,\text{mV},\ \text{then} \begin{cases} v \leftarrow c \\ u \leftarrow u + d. \end{cases}
\end{aligned}
$$

In the equation, $\mathbf{v}$ is the membrane potential and $\mathbf{u}$ is a membrane recovery variable. Also this model is pretty simple, by changing the value of $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$, this model can simulate the behavior of many different types of neurons. Simulation is carried out on computer. The result shows that this model can perfectly simulate the inhibition of excitation of a neuron, and the shape of membrane potential is pretty realistic (figure 3.7).

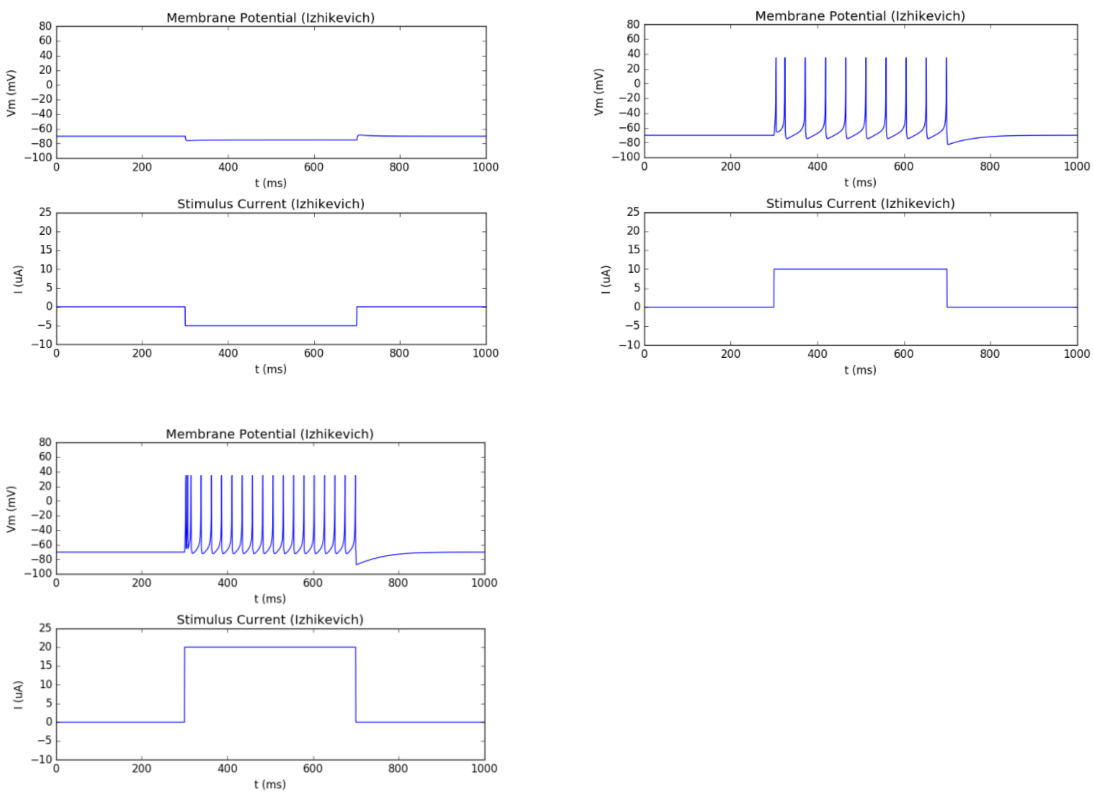
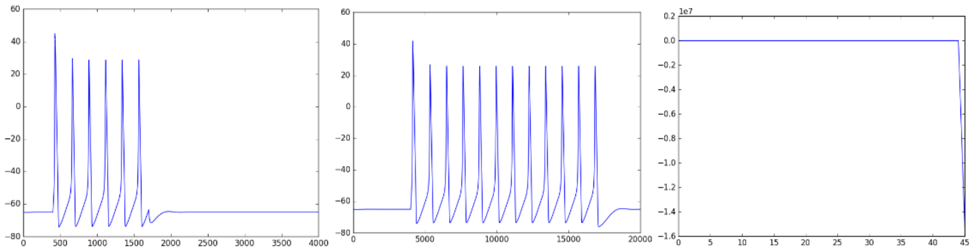


Figure 3.7 Simulation of Izhikevich Model

3.3.4 Comparison of Different Models

Izhikevich in 2006 wrote a paper discussing the advantages and disadvantages of different models which describe neuron activity [21]. The hudgekin-Huxley model is the most computationally expensive one. Integrate-and-fire model and Izhikevich model requires the same computational resources, but voltage wave of Izhikevich model is more realistic than Integrate-and-fire model.

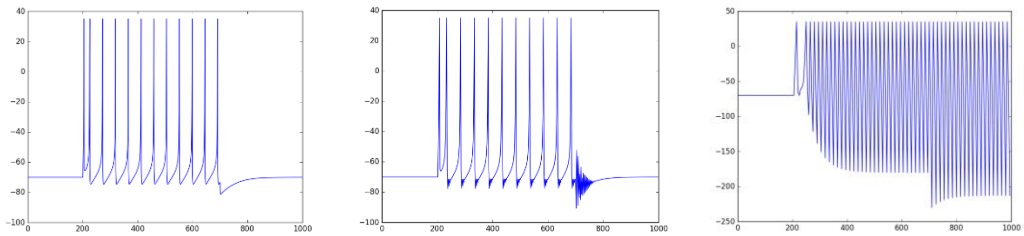
Simulation is also carries out one computer. The first model is Hudgkin-Huxley model. These are the graphs when step “dt” = 0.01 ms, 0.05 ms and 0.5 ms. In the following discussion, dt means iteration step (figure 3.8).



(a) (b) (c)

Figure 3.8 Hudgkin-Huxley Model Under Different Iteration Step

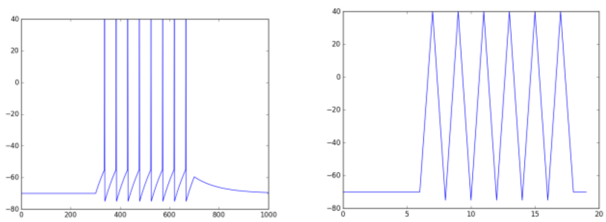
Graph (a) shows the wave when dt = 0.01 ms. In this case the model works well. The same situation happens when dt=0.05 ms, which is shown in (b). However, when dt =0.5 ms, the value of potential goes to minus infinity, which is shown in (c). So the iteration step for H-H model should be rather small. Otherwise the simulation won’t converge.



(a) (b) (c)

Figure 3.9 Izhikevich Model Under Different Iteration Step

The situation is similar when it comes to Izhikevich model (figure 3.9). (a) shows the graph when dt = 0.5 ms. (b) shows the situation when dt = 2 ms. The model works well in these iterations. However, when dt = 5 ms, the model won’t converge. This is shown in (c). Compared with HH model, the iteration step of Izhikevich model could be relatively large, which means it requires less computational resources.



(a) (b)

Figure 3.10 LIF Model Under Different Iteration Step

LIF model requires the least computational resources (figure 3.10). (a) shows the situation

when dt=0.5 and (b) shows it when dt =5. As you can see, under both circumstances the simulation converges. Although the shape is pretty ugly when dt is too large.
From simulation we can conclude that the iteration step of Hudgekin-Huxley model should be very small. Whereas the iteration step of Izhikevich model and Integrate-and-fire model could be relatively large. If the simulation duration is the same. For example, if we want to simulate the neuron's activity for 1 seconds, then the Hudgekin-Huxley model may have to carry out 20000 times of iteration. However, if we use Izhikevich model or Integrate-and-fire model, we may just have to carry out 1000 times of computation. This is a grate computation advantage. This is why Hudgekin-Huxley model is seldom used for simulating large-scale neural network. Because it requires too much computational resources.

4, A Typical Neural Sub-System

4.1 Somatic System

Human sense the world through somatic system. The beginning of each sensation process is always the sensory receptor. 4 types of sensation can be detected by sensory receptor: touch-pressure sensation, position sensation, thermal sensation and pain sensation. These 4 types of sensation are detected by different types of sensory receptor. In addition, the representation of these 4 types sensation in cerebral cortex is different. Some areas in cerebral cortex could form pain sensation while others may form touch sensation. Here we take touch sensation as an example and study human's somatic system.

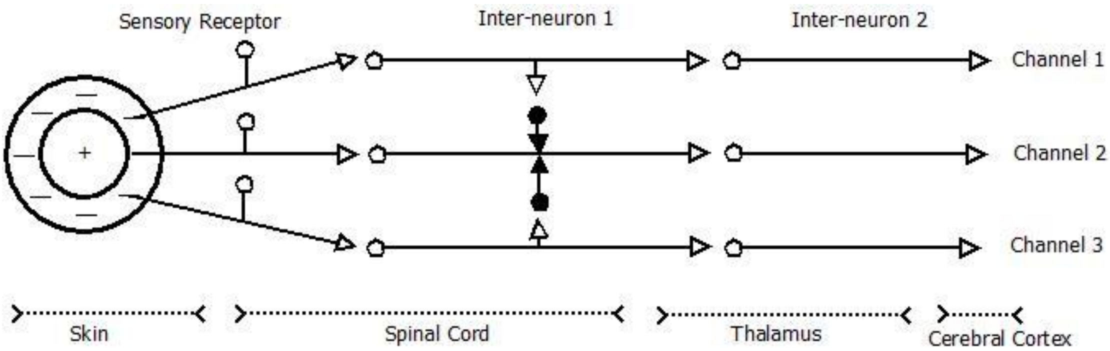


Figure 4.1 Somatic System

Figure 4.1 shows an example of how signal of touch sensation ascends through nervous system in our body. Each circle in this figure indicates a neuron body. Empty circle means that this neuron will excite next neuron, whereas black circle indicates that this neuron will inhibit the activity of next neuron. There are 11 neurons in total in this figure. The lines with arrows means the axons of these neurons.

The two circles on the left of this figure represent a receptive field. When there is a stimulus in the area with "+", then all of the neurons in channel 2 will be activated. From left to right, the signal is transmitted through neurons and axons. The dashed line on the bottom of this figure indicates where these neurons are lying. In this example, receptive field lies in skin. Then the signal is transmitted through the neurons in spinal cord and it keeps ascending. After several milliseconds, the signal reaches the neuron in thalamus, where it is transmitted to another neuron and continue to ascend. Finally, the signal is transmitted to cerebral cortex. The neural network in cerebral cortex is not plotted, because the structure of it is too complicated. What is sure is that when the stimulus signal finally reaches cerebral cortex, many neurons in cerebral cortex will be activated. The synchronized firing of these neurons in cerebral cortex is the way touch sensation is produced.

The structure of receptive field enables it to sense the position of stimulus. As we have discussed, if the stimulus is placed in the central area (with "+"), then only channel 2 will be activated. There are no signals in channel 1 and channel 3. However, if the peripheral area of receptive field is placed with a stimulus, then channel 1 and channel 3 will be activated. Meanwhile, signals of channel 1 and channel 3 will inhibit the activity of channel 2. This is the function of inhibitory neurons, which are represented by black circles and black arrows in this figure. In summary, when stimulus is placed in different areas, various channels and groups of neuron will be activated. In this way, brain can locate the position of stimulus.

4.2 Coding Pattern of Somatic System

How neurons represent touching sensation is an interesting question. This is the field of neuron coding, which also concerns about information theory. Since brain is efficient in processing all kinds of information, it is natural to consider applying information theory to the analysis of brain. A good overview of this field is the book "Information Theory and the Brain" [22]. It is a book that includes multiple papers regarding to using information theory to study brain. Unfortunately, I was unable to go through that book. So the discussion about that book will be included in dissertation. Rather, I will focus on the most basic aspects about how neurons encode information.

There are 2 types of signal in human's nervous system, graded signal and all-or-non signal. Let me take the sensation system as example (figure 4.2). The sensation receptor can produce graded voltage signal. The more intense the stimulus is, the larger the amplitude of output voltage is. In other word, information in sensory receptor is coded by the amplitude of voltage.

However, in most neurons of human's nervous system, information is not encoded by the amplitude of voltage. Rather, information is encoded by the frequency and timing of spiking. Inter-neuron 1 is a good example. In the voltage graph, each line represents a "spike". A spike is an impulse of voltage. As you can see, when the amplitude of the signal from sensory receptor is small, the frequency of spikes is low. Whereas when the amplitude of the signal from sensory receptor is large, the frequency of spikes is high. The amplitude of each spike is the same, and the only difference is the frequency. This example illustrates how information can be coded by frequency in our nervous system.

Some recently research shows that the timing of spikes also plays a crucial role in neural coding. A typical theory is called Spike-Timing Dependent Plasticity. The main idea of this theory is that the relevant spiking time of 2 neurons can affect the synapse growth between 2 neurons. And this is the foundation of learning behavior. We will cover this topic in later chapters.

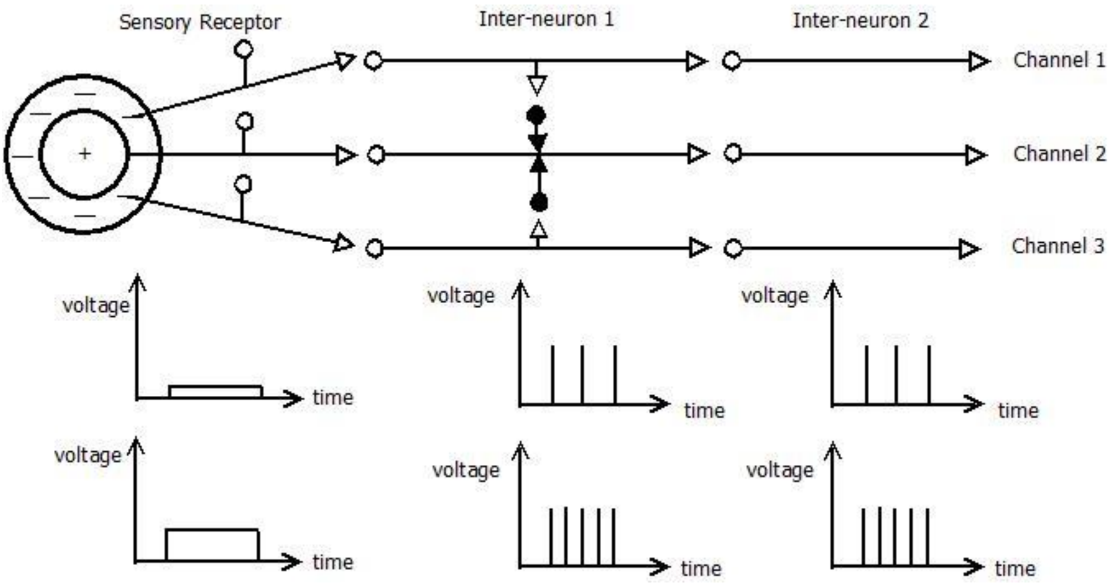


Figure 4.2 Neural Coding in Somatic System

5, Graph Theory

Using network theory to study brain is a pretty new scientific field, and there are not many books concerned about this topic. However, there are 2 high-quality papers reviewing this topic. Thus, most content in this chapter is from these 2 papers [23 24].

5.1 Inspiration to Use Graph Theory to Study Brain

Bavelas (1948) studied how 2 groups of people communicated ideas, and he found that a complex system could not be understood by only studying the individual components of the system. Rather, in order to understand the behavior of the system, how each component interacts with other components must be studied. This can be viewed as the start of using graph theory to study complex system.

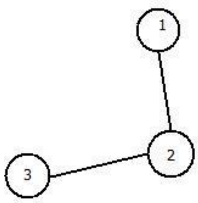
However, due to some reasons, applying graph theory to brain study did not get popular as expected in the following several decades. Until 1998, Watts and Strogatz proposed the small-world model of network, and they further showed that a group of nodes have some specific functions. This specialization has a high efficiency in transmitting information globally. This property of small-world network is similar to some properties of brain. Then relevant theories were applied to brain studies. Many important concepts in graph theory were used to analyze brain. This section will list some important concepts of graph theory. And in the next section, I will show how these concepts are relevant to brain.

5.2 Nodes, Edges and Network Representation

A graph consists of nodes and edges. Regarding to the representation of brain, each node in a graph usually represents a functional area of brain, and the edges often represent the connections between different areas. These connections can be either functional connections or anatomical connections. If these are anatomical connections, the topology of network is usually obtained by the time of response of different areas. If 2 areas are activated almost at the same time, then the functional connection between them is strong. In contrast, if 2 areas are not activated at the same time, then the connections between them are weak. There are also anatomical connections. Anatomical connection means the physical connection, often referring to nerve fibers. If there is a major nerve fiber between 2 areas of brain, then the physical connection between them is strong.

The node in a network seldom represents a single neuron. In most cases, each node represents a functional area of brain. The reason is that there are about 100 billion neurons in human's brain, and each neuron has about 7000 synapses in average. If we want to analyze the structure of the whole brain meanwhile retaining such details, the scale of network will be too large to handle. In addition, it has been proved that cognitive functions are based on large-scale activation. Thus, it is reasonable that we discard some details and model a functional area as a node in a graph.

In computer, a network is represented by matrix. A simple example is shown below (figure 5.1). This network consists of 2 edges and 3 nodes. The matrix on the right depicts the network in a mathematical way. If connections exist between 2 nodes, then the value of that particular position is one. The value of numbers in this matrix is either 0 or 1. However, this is not necessary. Actually, these values can be any values between 0 and 1. 0 Indicates there is no connection at all, and 1 indicates the connection has reached the maximum intensity. Any value between 0 and 1 indicates there are connections but somehow do not reach the maximum value.



Node	1	2	3
1	0	1	0
2	1	0	1
3	0	1	0

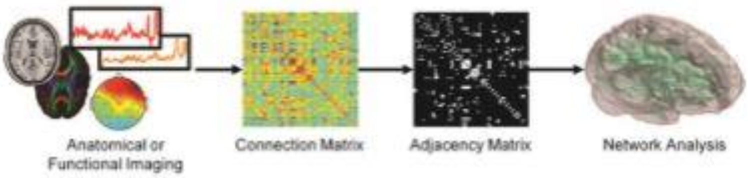


Figure 5.1 Representing Network Using Matrix

This figure illustrates how scientists map the structure of brain to a matrix. The color in this figure indicates the intensity of connection. Connection matrix stands for the anatomical or functional structure of brain. An adjacency matrix can be obtained by applying threshold to a connection matrix. For example, if connection matrix represents the anatomical structure of brain, then the elements in connection matrix represent the strength of physical connections (fibers) among nodes. Applying threshold means to omit those connections whose strength is below a specific level. For instance, we may ignore the connections which has less than 5 fibers. In this way, only major connections in the network will be maintained in adjacency matrix.

The reason to apply threshold is that this operation can tremendously reduce the scale of network. After applying threshold, many elements in matrix will be 0, turning the matrix to a sparse matrix. And computation of sparse matrix requires less computation resource than ordinary matrix.

5.3 Use Graph Theory to Interpret Brain

After turning the brain to a network, the next step is to use some metrics to study that network. In network theory, there are many measures to evaluate a network. In this section, I will show these measures and how these measures can be used to analyze the structure of brain.

5.3.1 Node degree, Degree Distribution and Assortativity

Node degree is the number of edges that are connected to a node. Degree distribution describes the degrees of all nodes. In some network, the degree distribution obeys Gaussian rule, while some may not. Degree distribution depicts some very important properties of network. And complex networks usually have non-Gaussian distribution. Assortativity measures the likelihood that nodes with similar degrees are connected. If assortativity is positive, it means that high-degree nodes are likely to be connected to each other meanwhile low-degree nodes are likely to be connected to each other. In contrast, if assortativity is negative, it means that high-degree nodes are likely to be connected with low-degree nodes.

5.3.2 Clustering Coefficient and Motifs

Clustering coefficient is a ratio. It can be obtained by dividing the degree of the node by the number of total connections. For example, Figure 5.2 shows a simple network. There are 7 nodes, so the total number of possible connections is 21. Given this, since the degree of node 7 is 6, the clustering coefficient is 6/21. In contrast, the clustering coefficient of node 1 is only 1/21. So the more connections have, the larger the coefficient of this network is. Studies shows that complex network has larger clustering coefficient, indicating there are more connections in complex network. Motifs are some repeated connection patterns in a network. In this following picture, the connection pattern of 1,6,7 is totally the same as the pattern of 1,2,7. Thus, the pattern is a motif.

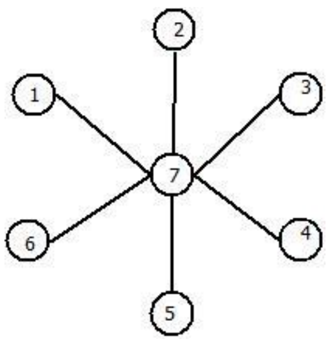


Figure 5.2 A Simple Network

5.3.3 Path Length and Efficiency

Path length is the minimum number of edges between 2 nodes. Random network and complex network have small average path length. Efficiency is inversely related to average path length.

5.3.4 Connection Density

Connection density is the ratio of the number of actual edges and the number of all potential edges. Take the above figure as an example, there are 6 connections in network, the number of all potential edges is 21. Thus the connection density of that network is 6/21.

5.3.5 Hubs, Centrality and Robustness

Hubs in network have more edges than other nodes. Centrality measures how many shortest path must go through that node. If the centrality of a node is high, it means that this node a is must path for many connections, thus being important to the communication of nodes in this network. For example, in figure 5.3, there are 36 possible pairs of connections (1-2, 1-3, 1-4, ……, 7-8, 7-9, 8-9). Among these pairs, 16 shortest paths are bound to through 9. In contrast, only 8 shortest paths must go through node 1. Thus, node 9 has higher centrality, and it is more important to the communication of network. Intuitively, we can come to the same conclusion. Node 9 is the only node that communicates two groups of nodes. Node 9 is called a hub. Study shows that elimination of non-hub nodes has little influence on the whole network, whereas the elimination of hubs can largely affect the network's robustness.

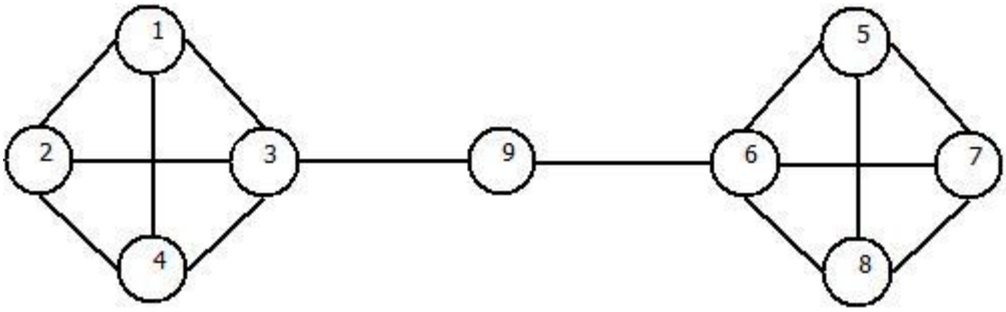


Figure 5.3 A Simple Network 2

5.3.6 Random Network, Small-world Network and Scale-free Network

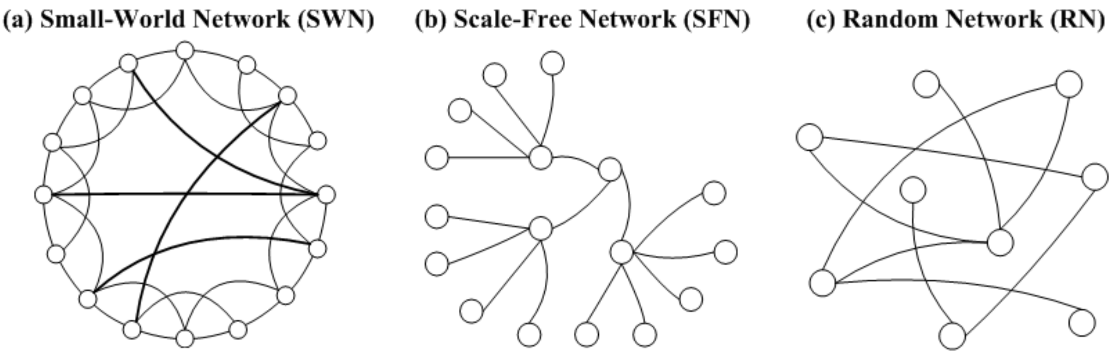


Figure 5.4 Different Types of Network

A random network is the most ordinary network (figure 5.4). The feature of this network is that there seems to be no feature in the network. The probability that 2 nodes are connected is random, and node's degree distribution obeys Gaussian law. However, in scale-free network, the degree distribution obeys power law. The feature of scale-free network is that it has multiple hubs. In real world, internet network is like a scale-free network. Actually, there is a third network, which is called regular network. In regular network, only adjacent nodes are connected.

Small-world network is a forth network. The feature of small-world network is high local clustering. Meanwhile different clusters can communicate through some edges. In other words, small-world network inherits some features from regular network, random network and scale-free network.

5.3.7 The Property of Brain Network

The study to brain's functional network show that it manifests both small-world property and scale-free property.

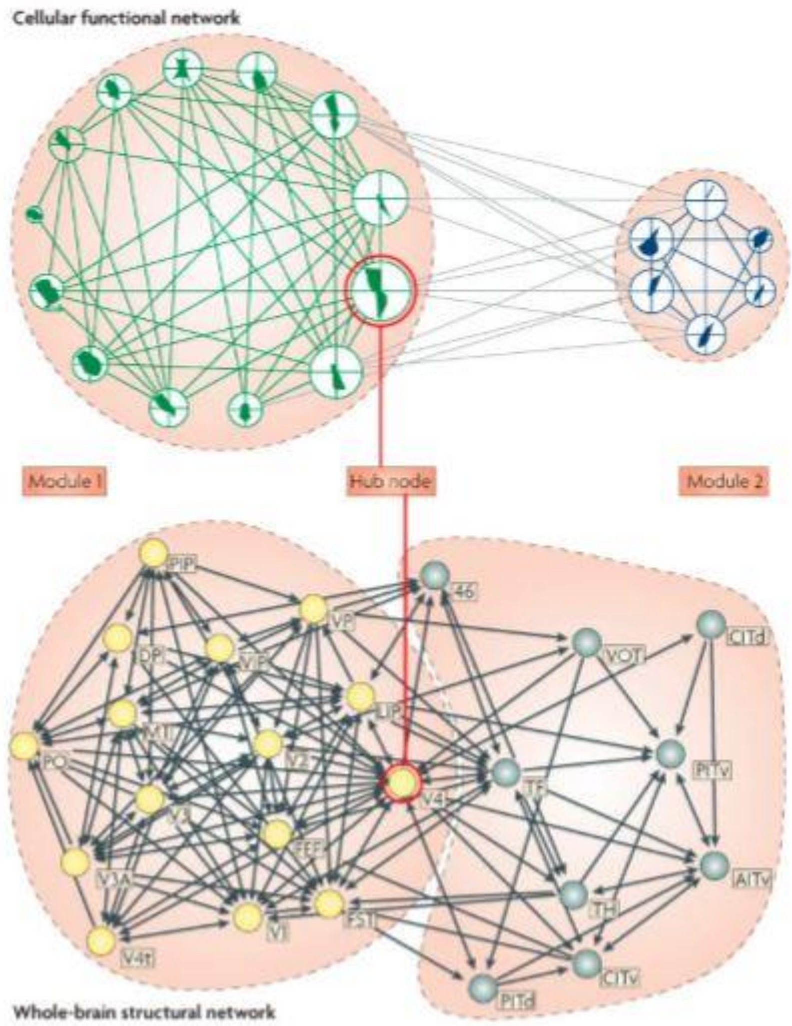


Figure 5.5 Network of Brain

Figure 5.5 shows the network structure of a cat's brain and macaque brain. The upper panel is functional network of cat's brain. Each circle denotes a functional region. The size of the circle denotes the degree of a node. The illustration in each circle represents the direction of response. The meaning of this illustration is not thoroughly discussed in paper. I guess the illustration denotes to what region this region has the largest response. For example, the largest circle responses to the regions at 11 o' clock position and 6 o' clock position more intensely than to regions in other directions. The lower panel shows the structure network of macaque cortex. Each node represents specific brain area. Both of the network manifest small-world property. And the degrees of nodes in both networks obey power law.

6, Brain and Cognitive Science

6.1 Overview

In this chapter, we focus on the cognitive functions of brain. Most of the theories come from cognitive science and psychology. However, since every single cognitive function relies on some specific neural structure, we will also cover some theories about neuroscience. We will try to discover the connections between brain's cognitive function and its underlying neural structure.

The book "cognitive psychology and its implications" by John Anderson [25] is a rather satisfactory introduction to this field. This book covers the most fundamental cognition questions about brain, including attention, perception, memory, learning and problem solving. Besides, this book listed and introduced many relevant experiments. However, I am not able to use this book as my main reference. Because too much content is included in this book. Rather, I will use this book as a directory to my study of brain's cognitive function. I will only use this book to figure out what topics should be included. My main reference is Wikipedia.

6.2 Perception and Attention

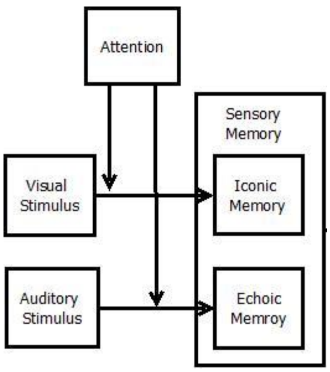


Figure 6.1 Sensory Memory

Figure 6.1 shows the process perception and the role of attention in perception process. Brain receives visual stimulus from outside world and store the information to iconic memory. Auditory information is stored in echoic memory. Both of iconic memory and echoic memory belong to sensory memory. A major property of sensory memory is that information written in this memory will not be stored for a long time. These signals will soon vanish in sensory memory. Some of them may be transmitted to short-term memory, others may just disappear. The different properties of memories indicate a different biological structure in brain. The mechanism and brain structure used in sensory memory must be different from those used in short-memory. We will cover this topic in later chapter.

Attention is a limited resource, and it can affect the process of perception. Attention can be assigned to limited perception process. However, if a perception process is frequently practiced, less attention is needed. The perception process which requires no attention is called automatic, and the perception process which requires attention is called deliberate.

One interesting topic about attention and perception is pattern recognition (figure 6.2). It is believed that 2 types of processes are involved in pattern recognition. The first type if called Bottom-Up process. This idea of this process is that brain form the concept in our brain by integrating pieces of scattered concepts. The other type is called Top-Down process, which means that the brain forms a concept of an object by comparing this concept with other existed concepts. Psychological studies show that both types are involved in pattern recognition process.

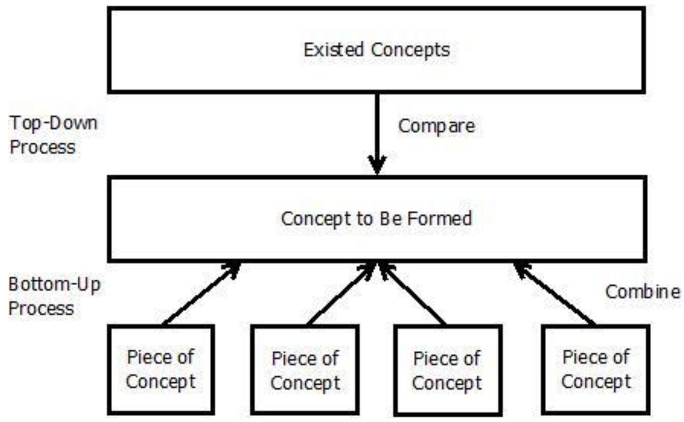


Figure 6.2 Process of Perception

6.3 Knowledge Representation

There are two different theories that try to explain how knowledge is stored in our brain. One is called “dual-code theory”. This theory holds that knowledge is represented by visual images and verbal representations. The other theory is called “propositional code theory”. In contrast with “dual-code theory”, “propositional code theory” holds that knowledge is not represented by specific visual or verbal representation. Instead, knowledge is presented by abstract concepts. And according to this theory, the meaning of one concept may be represented by its connections to other concepts.

Several experiments can partly prove the correctness of propositional code theory. One experiment is accomplished by Wanner in 1968, in which the researcher shows that when subjects are asked to remember a sentence or word, subjects tend to memorize the abstract meaning of it instead of the exact word. The experiment about visual memory came to the same conclusion. One experiment by Shepard in 1967 showed that when subjects are asked to remember a picture, they tend to remember some abstract representations of that picture instead of the exact picture.

Another idea of propositional code theory is that knowledge can be represented by propositional network, which basically consists of concepts and relationships. An example of propositional network is shown below. It should be pointed that propositional code theory also largely affected the study of artificial intelligence. One branch in AI field is called symbolic branch, who deems that knowledge in intelligence is represented by a propositional network. Since every concept is represented by a symbol in computer and these concepts could be understood by human, this method is called symbolic method.

Figure 6.3 is an example of propositional code theory. Number “1” in the ellipse is a proposition. The meaning of this proposition is “Professor gives me an A+ to my dissertation after my presentation”. The arrows connect the arguments which have some relation with the proposition.

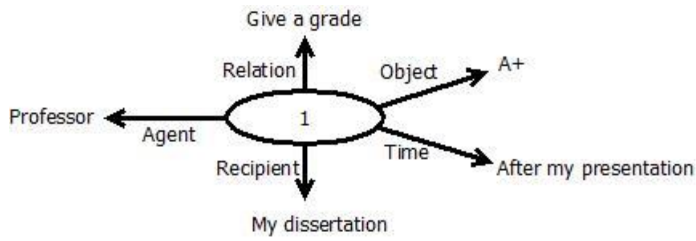


Figure 6.3 Proposition

Complex knowledge units are called schema, and typical instances of schema are called prototypes. Schema can represent many things, include a sequence of actions. More discussion about schema and prototype is included in John Anderson’s book.

6.4 Memory Type

There are different types of memory in our brain. A good classification can be found in [26]. Human’s memory can be roughly divided into 2 types, long-term memory and short-term memory. Within each type, there are several sub-types.

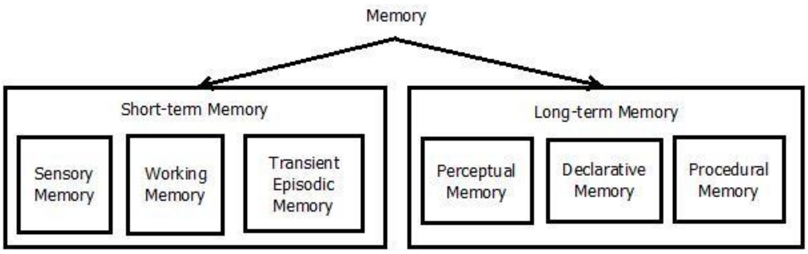


Figure 6.4 Types of Memory

Figure 6.4 a list of the types of memory in our brain. Discussion about working memory, transient episodic memory and perceptual memory will not be included in this dissertation.

Short-term memory can store knowledge only when it is activated. And the time that short-term memory is activated is limited. Thus, the knowledge in short-term memory will usually be stored for a very short period of time. In contrast, the knowledge in long-term memory can be stored for a relatively long period of time, like a few days or many years.

Although knowledge in long-term memory can be stored for a long period of time, recalling that knowledge consumes more time. The reason is that when we want to recall the knowledge in long-term memory, it must be firstly transmitted to short-term memory. In other words, we can not access long-term memory directly.

7, Connections between Brain's Cognitive Function and Its Underlying Neural Activity

An important property of complex system is the disconnection between local structure and the manifestation of overall system. It is hard to connect these 2. However, if we can make a bridge between a system's structure and its function, this system is somehow not complex anymore. In this chapter, I will try to study the connection between human's nervous structure and its function.

7.1 Hebbian Learning

A good introduction to the learning mechanism of brain can be found in the paper "A History of Spike-Timing Dependent Plasticity". This paper reviews the history about how people understood the learning process of brain [27]. Back to 11^{th} century, some people thought that the brain is blank when a baby is born. As time goes by, orphan studies the knowledge and "put" them in brain, so that the brain is not blank anymore. This is a naïve theory about brain's plasticity, but this theory can somehow reveal the fact that brain is constantly changing.

A great summary of brain's plasticity came from Donald Hebb. His idea is illustrated in figure 7.1. He postulated that if cell A is constantly and repeatedly exciting cell B, then the connection between 2 cell will be strengthened. Although Hebb himself did not provide a mathematical formulation of this learning mechanism, his postulation is pretty close to what we know about the learning mechanism in our brain.

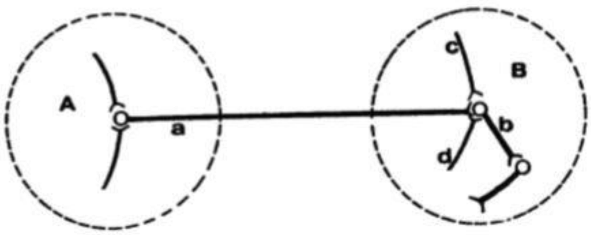


Figure 7.1 Hebb's Postulate

In the following decades, many scientists devoted themselves to this field. Now this learning process has a more accurate definition: spike-timing dependent plasticity (STDP). The idea is that the strength of connection between neurons is determined not only by the spike, but also determined by the timing of spikes. Gerstner in 2002 composed a paper discussing the mathematical formulation of STDP [28]. The assumption of his model is shown in figure 7.2.

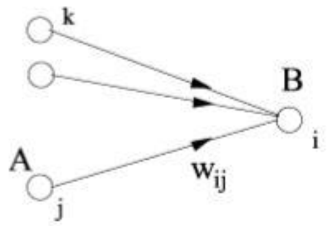


Figure 7.2 Hebbian Learning Between Neurons

There are a basic assumption regarding to how this network works. Our subject is the changing of w_{ij}. The changing speed of w_{ij} is determined by the spikes of only A and B. The spike of B is not due to A. Rather, the spikes of B is caused by other k neurons, which are plotted on the top of this figure. There are two types of formulation, one is rate-based. Rate-based simulation means the strength of connection is determined by the average spike rates of A and B. Another formulation is called spike based. This formulation argues that the connection is determined the actual number of spikes produced by A and B in the past.

7.1.1 Rate-Based Hebbian Learning

A general mathematical formulation of Hebbian Learning can be written like the following equation. The learning rate is determined by the average firing rate of neuron i and neuron j. Some other coefficient can also influence the learning rate. These coefficients are determined by the current strength of connection w_{ij}.

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t} w_{ij} \approx\ & c_2^{\mathrm{corr}}(w_{ij}) v_i v_j + c_2^{\mathrm{post}}(w_{ij}) v_i^2 + c_2^{\mathrm{pre}}(w_{ij}) v_j^2 \\ & + c_1^{\mathrm{pre}}(w_{ij}) v_j + c_1^{\mathrm{post}}(w_{ij}) v_i + c_0(w_{ij}) \\ & + \mathcal{O}(v)\ .\end{aligned}$$

In practice, some of the items in the equation above is omitted. If we omit some of the items, the equation can be reduced to the following form.

$$\frac{\mathrm{d}}{\mathrm{d}t} w_{ij} = c_2^{\mathrm{corr}}(w_{ij}) v_i v_j$$

The changing speed of synapse is determined by the spike rates of A and B (which are neuron i and neuron j). c_2^{corr} is a correlation coefficient. It is a function of the current connection strength w_{ij}. In some cases, people consider c_2^{corr} as a constant. Sometimes people will give an upper bound of this changing rate. To do that, we shall set c_2^{corr} as the following form. As long as the strength of connection w_{ij} exceeds the upper bound w^{max}, the correlation coefficient will turn to be negative, which will result in turning the changing rate to negative value. Then the strength of connection will be undermined.

$$c_2^{\mathrm{corr}}(w_{ij}) = \eta_0 (w^{\mathrm{max}} - w_{ij})$$

7.1.2 Spike-Based Hebbian Learning

The idea of spike-based Hebbian learning is that the learning rate is determined by both the number of spikes in the past as well as the relative spike timing of the two neurons. A mathematical formulation is like this.

$$\begin{aligned}\frac{dw_{ij}}{dt} = c_0(w_{ij}) &+ \sum_{t_j^f} \alpha_1^{\text{pre}}(w_{ij}; t - t_j^f) \\ &+ \sum_{t_i^f} \alpha_1^{\text{post}}(w_{ij}; t - t_i^f) \\ &+ \sum_{t_j^f} \sum_{t_i^f} \alpha_2^{\text{corr}}(w_{ij}; t - t_i^f, t - t_j^f) + \dots \; .\end{aligned}$$

The correlation term is the third term. However, Gerstner did not provide the form of correlation term of spike-based learning. This term can be found in other scholar's paper. For example, Hosaka [29] in his paper used the following form.

$$\Delta w(\Delta t) = \begin{cases} -\lambda f_{-}(w) \times \exp(-|\Delta t|/\tau_p) & (\Delta t \geq 0), \\ \lambda f_{+}(w) \times \exp(-|\Delta t|/\tau_d) & (\Delta t < 0), \end{cases}$$

In this equation, Δt is the relative spike timing of two neurons. If the presynaptic neuron fires before the postsynaptic one, then the connection will be strengthened. Otherwise it will be undermined. Figure 7.3 can illustrate this relationship perfectly. In the figure 7.3, if Δt is less than 0, then it indicates the presynaptic neuron fires before the postsynaptic one. Under this circumstance, the learning rate is positive, which means the connection will be strengthened. This strengthening is called long-term potentiation. On the contrary, if the Δt is larger than one, it means postsynaptic neuron fires at first. It will cause the undermining of connection, which is called long-term depression.

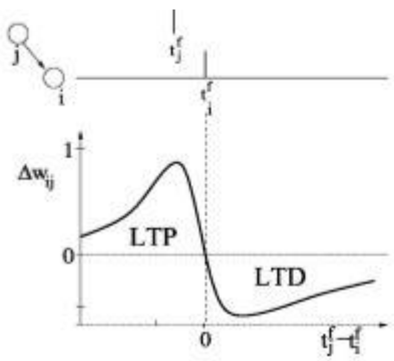


Figure 7.3 Learning Window

7.2 Long-Term Memory and Pattern Recognition

The learning behavior of neurons is a perfect demonstration of brain's self-organization and complexity. The behavior of a single neuron is simple, and the learning rule is not such complex. However, when billions of neurons using this learning rule are put together, the whole system can manifest complicated behaviors. In addition, although the learning rule is pretty simple, the adaptability of brain is amazing. The learning rule enables the structure of brain to constantly change. So the brain itself can always alter its structure to make it adapt new environment. This is why learning behavior is so important to brain. It is the source of brain's self-organization ability. And it is the source of human's intelligence.

Learning rule is also important to memory and pattern recognition. This is confirmed by computer simulation. In Hosaka's paper, a neural network with STDP is simulated on computer [29]. The procedure is to first input repeatedly some stimulus to the neural network. The pattern of each stimulus is the same. Several times of stimulus later, the structure of neural network will change and can respond to this specific stimulus (figure 7.4).

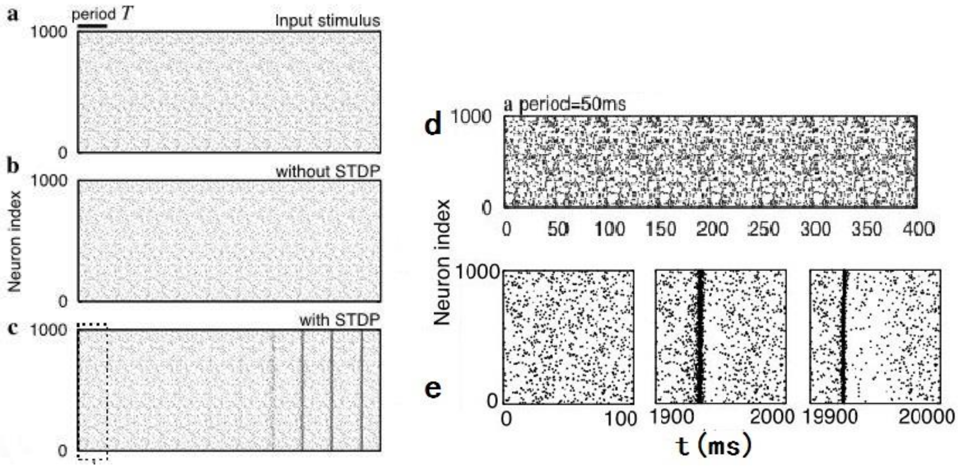


Figure 7.4 Synchronized Fire of Neurons

(a) shows the input signal of this neural network. (d) is a detailed illustration of input signal. As you can see, there is an obvious pattern in that input signal. The period of that pattern is 50 ms, and that pattern is repeatedly input to neural network. After some time, the structure of network is modified due to the existence of STDP. And the output pattern of network also changes. (c) shows the change of output pattern. At the beginning, there is no synchronized fire. However, after learning, when the input pattern shows up again, the neuron will fire at almost the same time. This is called "syn-fire".

(e) is a detailed illustration of output signal. As you can see, the line is not strictly vertical, which indicates that all neurons don't actually fire at the same fire. However, the firing time is pretty close.

Simulation is carried out on computer. The figure 7.5 shows the result of simulation. This source code is written by Toby Lightheart. The syn-fire phenomenon is not so obvious as the experiment in Hosaka's paper. Maybe this is caused by the specific learning rules.

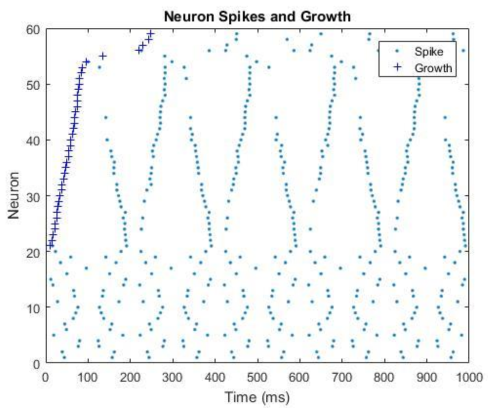


Figure 7.5 Simulation of Network with STDP

What is the meaning of this syn-fire phenomenon? At least it tells us the following things. Firstly, STDP can surely change the structure of brain. Secondly, the change of brain structure makes able able to recognize some fixed pattern. Thus we could say that STDP is the underlying neural mechanism of pattern recognition. Thirdly, since a pattern can be represented by syn-fire in brain, we can postulate that other concepts can also be represented by syn-fire of brain. We when recall a concept, maybe some specific group of neurons will fire at the same time.

The representation of concept is a major function of memory, as we have introduced. Thus, we can say the STDP is the mechanism of the formation of long-term memory. Long-term memory is achieved by the modification of neural structure. This is confirmed by several studies. And a fixed pattern of concept is stored in brain by syn-fire of a group relevant neurons [30].

7.3 Short-Term Memory

We have shown that long-term memory is achieved by the modification of neural structure which is caused by STDP. However, there is another type of memory called short-term memory. As we have shown, short-term memory is temporary, and it could not be retrieved. This is pretty different from the features of long-term memory. This difference implicates the distinct neural mechanisms behind them.

It has been confirmed that short-term memory is due to the special structures in brain. The paper by Stephen Grossberg analyzed this type of structure [31] (figure7.6). Input 1 and input 2 can receive stimulus from outside world. When N1 is activated by stimulus, it will stimulate itself too. Meanwhile, it will spontaneously inhibit the activity of another neuron. So only M1 will be activated. M1 could be consider as some specific region in cerebral cortex. We have discussed that concepts are represented by the syn-fire of particular groups of neurons in brain. Combining with figure, when there is stimulus, the output of N1 will last for a period because it is activating itself. This output may cause the activation of other neurons in M1 region. However, since N2 is inhibited, neurons in M2 will not activated. In this way, some groups of neuron are activated and others are not.

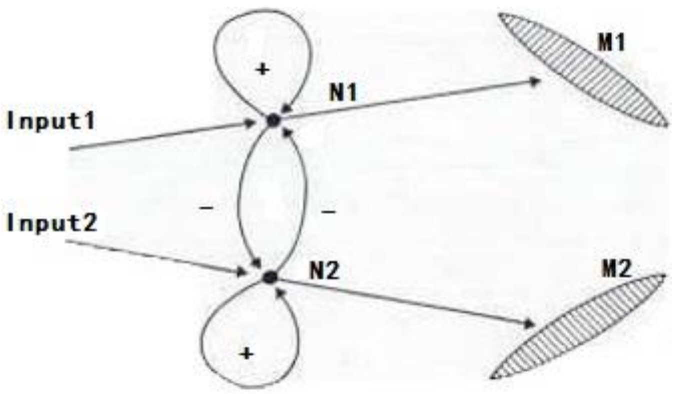


Figure 7.6 Short-Term Memory

Since the signal in N1 will decay itself, this self-activation will not last for a long time. When this self-activation disappears, short-term memory ends. And the information can be retrieved anymore.

8, Brain and Artificial Intelligence

8.1 Artificial Intelligence Overview

Understanding the essence of intelligence and further building intelligent systems have always been the dream of many scholars. Along history, a tremendous number of scholars have devoted themselves to this charming question, thus forming largely variated methods to build intelligent systems. In general, all of these methods can be divided into 3 categories: symbolic method, sub-symbolic method and hybrid method. A rather comprehensive review of these 3 types of methods in artificial intelligence can be found at Goertzel's paper in 2011[33].

Scholars of symbolic branch believe that the nearly omnipotent power of brain comes from its ability to represent and manipulate symbols. Thus, these scholars tried to build agents that can represent and manipulate symbols, in a similar way as our brain does. Symbols are stored as abstract data in computer's storage. These symbols are meaningful, which means human can understand the meaning of these symbols. They could be logic symbols, terms, numbers or other things, but they have concrete meaning.

On the contrary, the sub-symbolic branch believes that the symbols in our mind does not need to be represented explicitly. Instead, the neurons themselves are symbols. With the connections between neurons, our brain can produce the many complex concepts which are familiar to us, although we could not know explicitly what every neuron's meaning is. Thus, in order to build an agent, scholars in this field try just to connect many neurons, then regulate their behavior and learning rule. Artificial neural network is a typical method among sub-symbolic methods. Each neuron does not have a concrete meaning. But with connections, these neurons compose a human and complex network, which has the ability to represent meaningful concepts that are understandable to human beings. For example, the abstract information can be extracted and represented by hierarchy neural network [32]

Actually, from an application perspective, methods of these two field have their advantages and disadvantages individually. For example, symbolic methods may perform well in solving some types of problems while do badly solving other types of problems. The same situation also happens to sub-symbolic methods. It is widely accepted that symbolic methods are famous for solving complicated problems, especially reasoning problems, yet they perform limitedly on pattern recognition [33 34]. On the contrary, sub-symbolic methods perform well on pattern recognition problems, but they are limited when being used for complicated reasoning problems. Given this situation, some scholars try to combine these 2 different types of methods, forming a so-called hybrid method.

In the following content in this chapter, I will discuss some classical theories from these three fields. It is obvious that I could not cover all theories in this article. I will choose some typical theories in different categories. I will use artificial neural network to demonstrate the theory of sub-symbolic methods. A superset of "artificial neural network" is "machine learning". Artificial neural network is only one method among numerous methods in machine learning field. However, since other methods in machine learning is not so relevant to brain, I will not cover them in this dissertation.

8.2 Artificial Neural Network: History

Since human beings are considered the most intelligent creature ever discovered, it is natural to try to understand intelligence by studying human beings, or more specifically, by studying human beings' brain. Enlightened by this idea, some scholars proposed artificial neural network.

The basic philosophy underlying artificial neural network is to imitate the structure and mechanism of brain. The formed network could be used for data processing and other tasks. However, the term "artificial neural network" is a little bit tricky. Because actually, the mechanism and structure of artificial neural network are quite different from real neural network. This section will be focused on introducing the history and trace of development of artificial neural network. A more detailed review can be found in John Hertz's book [35] and Jurgen Schmidhuber's paper [36]. John Hertz's book is written in 1991, which is a little bit old for today's research. It is a good introduction to learn the history of artificial neural network, but not an appropriate book to learn recent theories about artificial neural network. A recent review was released by Michael I Jordan in 2015 [37]. This paper discusses the trends in machine learning, meanwhile also covering the content of artificial neural network. As for studying artificial neural network from a statistical perspective and studying the theoretical theories underlying artificial neural network, "Pattern Recognition and Machine Learning" by Christopher Bishop is a good choice [38]. It starts from linear regression and covers the most classical neural network models and relevant mathematical theories. Many classical questions regarding to ANN are also discussed comprehensively in that book. It is hard to finish that book because too much content is included in it. More often, it serves as a reference book, like a dictionary. For example, when you have designed a ANN model but did not get a satisfactory result, you may turn to that book for an explanation. Table 8.1 shows the most classical models in the history of artificial neural network.

Table 8.1 Classic Models of Artificial Neural Network

Achievement	Time	Proposed by	Category
McCulloch & Pitts Neuron	1943	Warren S. McCulloch & Walter Pitts [39]	Neuron Model
Hebbian Learning	1949	Donald Old. Hebb [40]	Learning Rule
Perceptron	1958	Frank Rosenblatt	Neuron Model
Multi-Layer Perceptron & Back-propagation	1974	Werbos[41]	Network Model
Convolutional Neural Network	1979	Fukushima	Network Model

8.2.1 McCulloch & Pitts Neuron Model

This neuron model was proposed in 1943. The idea behind this model is that, there is an "all-or-none" property in our neuron. There are only two values of a voltage of a neuron, the lower one or the higher one. There are no intermediate states. Thus, McCulloch and Pitts reproduces this feature in their model. The output of a neuron will be either a higher one or a lower one. A illustration of this model is shown in figure 8.1.

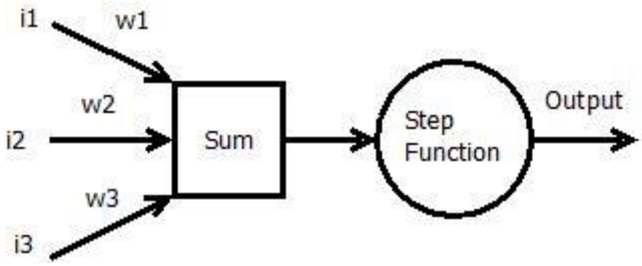


Figure 8.1 M-P Neuron Model

In McCulloch's paper, he concluded that for any logical expression, there is an appropriate network that can complete the same function using this neuron model. Although the properties of this model are quite different from a realistic neuron, its powerful computational ability demonstrated the magic of neural network to some extent. Furthermore, this model has vastly influenced neuron models proposed after this. In his paper, the author did not discuss how to modify the weights in this model. In other words, the author did not discuss the "learning ability" of this model. Instead, he concentrated on its computational universality. In later researcher, other scholars proposed many learning rules, which are used for modifying the weight of this model and thus making a network capable of solving different problems.

8.2.2 Hebbian Learning

In Hebb's book "the organization of behavior", Hebb proposed for the first time that learning is due to the strengthening of connection between perceptions. And this strengthening could only happen when 2 perceptions are activated independently but be activated close enough in time.

Also it was Hebb that for the first time proposed the information in our brain by neurons and connections. He postulated that the forming and changing of the connections between neurons accounts for the complex function of our brain. Today, this idea is widely accepted.

Hebb's theory at that time is only a physiological and psychological theory. Hebb did not formulate in mathematics. After his postulation on synapse's plasticity, many properties of synapses were discovered by other scientists. And the theories of synapse plasticity are constantly accumulating new discoveries, thus constantly changing. Now, Hebbian's first postulation on learning have been formally formulized by the term STDP (spiking-timing dependent plasticity). STDP theory can perfectly account for how the signals of neuron affect the structure of neural network, thus affecting learning process. A comprehensive review of Hebbian learning and STDP can be found in Henry Markram's paper [27]

8.2.3 Multi-Layer Perceptron

Based on McCulloch & Pitts' neuron model, AI researchers made many modifications and applied new artificial neural network to many practical problems. One of the most famous model is multi-perceptron model (figure 8.2).

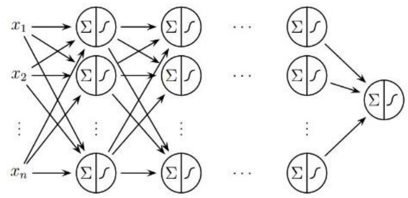


Figure 8.2 Multi-Layer Perceptron

$x_1, x_2, \dots, x_n$ are inputs of network. The inputs are passed to the hidden layer units, and then the next hidden layer, until the output. Then back-propagation algorithm is applied. This algorithm can detect the error between actual output and expected output. Then the weights of network are modified according to the error. After several times of iteration, the weights of network will converge and the "training" is completed. [41]

8.2.4 Convolutional Neural Network

The history of convolutional neural network can be traced back to 1979. It was proposed by Fukushima for the first time [42]. Two major features of convolutional neural network (CNN) that are different from traditional multi-layer perceptron are weight replication and subsampling, which largely reduced the number of weights in the network. Since more weights requires more computational resources, traditional multi-layer perceptron has a limited scale-if there are too many neurons, then there should be many weights which need to be computed. Since the number of weights in CNN is largely reduced, the scale of CNN network can be pretty large, which enhances the network's ability of processing information.

However, this CNN prototype promoted by Fukushima did not gain much popularity. According to Schmidhuber, it is because the potential of CNN structure is untapped under the learning rule that Fukushima proposed. Fukushima chose an unsupervised learning rule, whose performance was later proved to be less satisfactory compared with the learning rules used in recent CNN networks.

With new learning rules, today's CNN has much more layers than earlier ones. In some cases, the number of layers in CNN reaches 50. The increase of network's depths results in his enhancement of extracting abstract information from input data [32]. Due to the special properties of CNN, it is mostly used in computer vision field (figure 8.3).

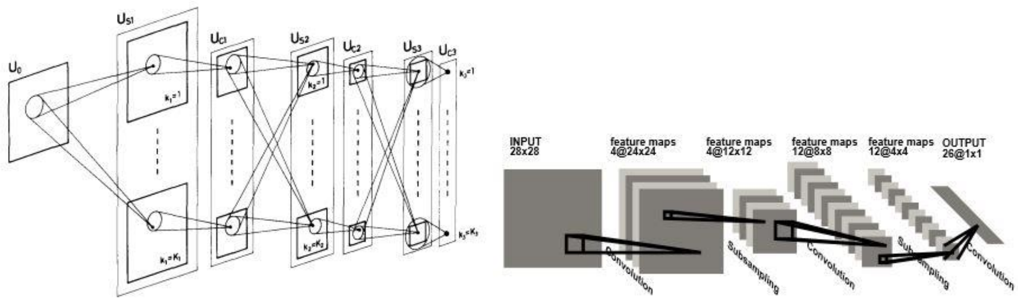


Figure 8.3 Convolutional Neural Network

8.2.5 Spiking Neuron Model

Spiking neural network lies in the third generation of artificial neural network [43]. Many sub-types of spiking neuron models have been invented, including LIF model, Hudgkin-Huxley model and Izhikevich model, which are included in chapter 4. I will not discuss in detail about the property of each sub-type. Rather, I will explain why spiking neuron models are becoming increasing popular.

Recent study shows that information in our brain is represented not only by the voltage, but also represented by the timing that a voltage occurs [44]. However, traditional McCulloch-Pitts based neuron models could not represent time. Thus, systems using traditional neuron models is less powerful than spiking models when they are used for information processing. In order to enable the artificial neural system to handle more amount of information, many scholars turn to spiking neuron models.

The proposal of spiking neuron can be seen as a reunion of AI community and biology community. Because the computing model, which is adopted by AI community, is now pretty close to realistic model which is used by biology community.

8.2.6 Relationship between Artificial Neural Network and Neuroscience

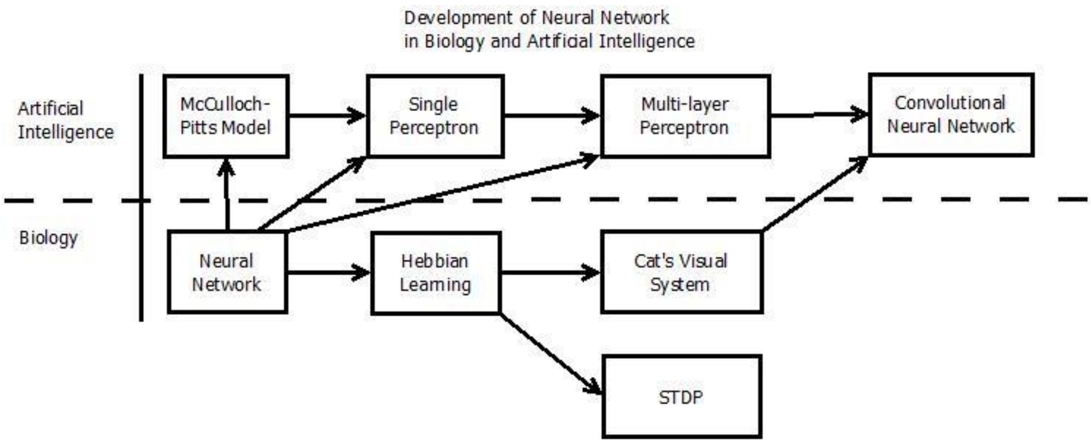


Figure 8.4 Artificial Neural Network and Biological Neural Network

McCulloch wrote in his paper that his idea was enlightened by physiological ideas. Actually, every great progress made in artificial neural network were influenced, slightly or largely, by biological neural network. Convolutional neural network, which becomes famous nowadays for its amazing ability of processing image, was also inspired by biological discoveries. The convolutional neural network resembles the structure of cat's visual system [45] (figure 8.4).

Although the relationship between artificial neural network and biological neural network is pretty tight, it is not until recently that they become pretty much the same. In other words, in a history about 50 years, artificial neural networks and biological neural networks are totally different concepts. Artificial neural networks, as introduced above, were invented to solve computational problems. Scholars made much simplification to these artificial neuron models in order to reduce computation load. As a result, many important properties of realistic neurons are not included in an artificial neural network. John Hertz concluded some major differences between artificial neuron model (which refers to McCulloch-Pitts model) and realistic neuron [35].

From the history of the development of artificial neural network, we could easily conclude that as long as manmade intelligent agent is not as perfect as our brain, we will have to get more inspiration from biological discoveries. And this is why computational neuroscience is so popular nowadays. The giant leap in computer science has made it possible to explore our brain more efficiently than any other time in history. We could expect that more and more facts of our brain will be unveiled.

8.3 Symbolic Methods in Artificial Intelligence

The concept of symbolic method was at first introduced by Newell and Simon in 1976 [46 47]. As Newell has proposed, the architecture of a symbolic system includes three parts: input/output part, control part and memory part. And I perceive the working mechanism as the following.

There are 4 concepts in a symbolic system: entity, judgement, action and expression. Expression is an executable command. It is like programming language. When you input the language to the computer, the computer can execute as you demand. However, the difference is, here the expression is not given by human, it is produced automatically by the symbolic system. This system can produce the expression according to the current environment and its memory about the past. After the system produces this expression, it will do as the expression says. Then it will receive the feedback from environment and produce the next expression. Expression consists of action, entity and judgement, or at least one of them.

Entity is an abstract concept, and it can represent various things, according to various situation. It can be a value, a position [48], a person or something else. But is should be noted that, this entity was given by human. Human defines what an entity means. Moreover, this entity could not be split. It is the minimum unit that a symbolic system can handle. Action is what the symbolic agent can do. For example, it can copy an expression and put expression in its memory, or it can move to a specific position. Judgement is the judging process implemented by the symbolic system. For example, the agent judges whether it is in position 1 or in position 2. If it is in position 1, then it will move to position 3; if it is in position 2, then it will move to position 1.

A very significant feature of symbolic method is that we know how the system works. We know how the system "thinks", what each entity represents and how it makes a decision. Because we define the entities for a symbolic system, the entities are readable to us. In other words, knowledge is represented by entities and combination of entities.

There are several classic models in symbolic cognition field, including SOAR by John E Laird [49], ACT-R by John R Anderson [50] and ICARUS by Pat Langley [51]. These models inherit the most important features from Newell's symbolic system: entity, expression and problem space. Furthermore, they improve Newell's symbolic system. One major improvement is the architecture of these models. The architecture of these models are more complicated and well-designed than Newell's model. For example, in ACT-R model, memory was divided into long-term memory and short-term memory. And the long-term memory was further divided into procedural memory and declarative memory. The structure of memory was more complicated and delicate than that in Newell' model.

9, Nonsense

9.1 Nostalgia

We all have experience like this: when we see some old things or scenario, we will have some special, familiar, cheerful but somehow melancholy feelings. This is called nostalgia. This phenomenon can be easily explained by the theories we have studied.

The thing that shows must be something that you constantly dealt with in the past. It formed some long-term memory in your brain. This formation was a result of STDP. The network structure in your brain was modified so that your brain can react to that thing. As time goes by, some connections between neurons may decay, but many or some of them remain there. Now you see that thing again, and some of the "old" neurons produce syn-fire again. Then the memory is recalled.

9.2 Development of Artificial Intelligence

It is widely accepted that symbolic methods are famous for solving complicated problems, especially reasoning problems, yet they perform limitedly on pattern recognition [34]. On the contrary, sub-symbolic methods performs well on pattern recognition, but they are limited when being used for complicated reasoning problems.

Given this situation, it is natural to combine the two methods together. Actually, it is one of the trends in artificial general intelligence. One vivid example is SOAR model. Although it was proposed as a symbolic model, in its recent version, some sub-symbolic methods were also included [52]. In the recent version of SOAR, non-symbolic method was added to perform visual imagery. Besides, there are mounting number of hybrid cognitive architectures recently [53 54]. These models inherit the most important findings of the earlier symbolic models, for example the memory represented by symbolic entity. But they also incorporate the non-symbolic methods to extend their function.

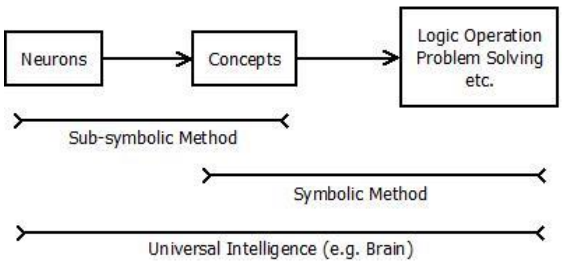


Figure 9.1 Relationship Between Symbolic and Sub-Symbolic Method

Actually, sub-symbolic methods and symbolic methods do not contradict with each other in essence. Sub-symbolic methods are well known for using nodes and connections to represent concepts. Whereas symbolic methods are well known for using concepts to solve problems, which can be perceived as "thinking".

Universal intelligent agent could combine these two types of methods. Brain is a good example (figure 9.1). Concepts are stored in many different regions. However, these concepts are not isolated, because there are hubs and fibers that connect different regions. So that concepts from different regions can be processed together. The neural activity within a region can be seen as implementation of sub-symbolic method, whereas inter-region activity can be seen as an implementation of symbolic method.

9.3 How AI Will Affect Our Society

One day if universal intelligent agent is achieved, what is the meaning of existence of human? All works that could be done by human can be done by those intelligent agents. Maybe on that day human can only be pets to those agents, just like the cats or dogs today. Human is not the supreme species anymore on earth. It maybe another type of evolution.

Reference:

[1] https://en.wikipedia.org/wiki/Outline_of_academic_disciplines
[2] https://en.wikipedia.org/wiki/Psychology
[3] https://en.wikipedia.org/wiki/Neuropsychology
[4] https://en.wikipedia.org/wiki/Neuroscience
[5] https://en.wikipedia.org/wiki/Physiology
[6] https://en.wikipedia.org/wiki/Neurochemistry
[7] https://en.wikipedia.org/wiki/Neurology
[8] https://en.wikipedia.org/wiki/Neurosurgery
[9] https://en.wikipedia.org/wiki/Neurosurgery
[10] Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell. *Principles of neural science*. 1st Edition. Elsevier Science Publishing Company, 1981.
[11] Siegelbaum, Steven A., and A. J. Hudspeth. *Principles of neural science*. Eds. Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell. Vol. 4. New York: McGraw-hill, 2000.
[12] http://www.interactive-biology.com/3247/the-neuron-external-structure-and-classification/
[13] https://en.wikipedia.org/wiki/Anaxonic_neuron
[14] http://www.dummies.com/education/science/biology/action-potential-of-neurons/
[15] http://www.psychologyinaction.org/2011/04/01/conventional-wisdom-upset-persistent-action-potential-firing-in-distal-axons/
[16] https://en.wikipedia.org/wiki/Biological_neuron_model
[17] http://icwww.epfl.ch/~gerstner/SPNM/node26.html
[18] http://neuronaldynamics.epfl.ch/online/Ch1.S3.html
[19] http://icwww.epfl.ch/~gerstner/SPNM/node14.html
[20] Izhikevich, Eugene M. "Simple model of spiking neurons." *IEEE Transactions on neural networks* 14.6 (2003): 1569-1572.
[21] Izhikevich, Eugene M. "Which model to use for cortical spiking neurons?" *IEEE transactions on neural networks* 15.5 (2004): 1063-1070.
[22] Baddeley, R., Hancock, P., Földiák, P. *"Information Theory and the Brain"*. U.K. Cambridge, 2000.
[23] Telesford, Qawi K., et al. "The brain as a complex system: using network science as a tool for understanding the brain." *Brain connectivity* 1.4 (2011): 295-308.
[24] Bullmore, Ed, and Olaf Sporns. "Complex brain networks: graph theoretical analysis of structural and functional systems." *Nature Reviews Neuroscience* 10.3 (2009): 186-198.
[25] Anderson, John R. *Cognitive psychology and its implications*. WH Freeman/Times Books/Henry Holt & Co, 1990.
[26] http://www.brains-minds-media.org/archive/150
[27] Markram, Henry, Wulfram Gerstner, and Per Jesper Sjöström. "A history of spike-timing-

dependent plasticity." *Spike-timing dependent plasticity* (2011): 11.
[28] Gerstner, Wulfram, and Werner M. Kistler. "Mathematical formulations of Hebbian learning." *Biological cybernetics* 87.5-6 (2002): 404-415.
[29] Hosaka, Ryosuke, Osamu Araki, and Tohru Ikeguchi. "STDP provides the substrate for igniting synfire chains by spatiotemporal input patterns." *Neural computation* 20.2 (2008): 415-435.
[30] http://www.human-memory.net/processes_storage.html
[31] Grossberg, Stephen. "Contour enhancement, short term memory, and constancies in reverberating neural networks." *Studies of mind and brain*. Springer Netherlands, 1982. 332-378.
[32] LeCun, Yann, Koray Kavukcuoglu, and Clément Farabet. "Convolutional networks and applications in vision." *ISCAS*. 2010.
[33] Goertzel, Ben. "Artificial general intelligence: concept, state of the art, and future prospects." *Journal of Artificial General Intelligence* 5.1 (2014): 1-48.
[34] Besold, T. R. 2015. Same same, but different? A research program exploring differences in complexity between logic and neural networks. *Proceedings of the 10th International Workshop on Neural-Symbolic Learning and Reasoning NeSy'15*.
[35] Hertz, John, Anders Krogh, and Richard G. Palmer. *Introduction to the theory of neural computation*. Vol. 1. Basic Books, 1991.
[36] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." *Neural Networks* 61 (2015): 85-117.
[37] Jordan, M. I., and T. M. Mitchell. "Machine learning: Trends, perspectives, and prospects." *Science* 349.6245 (2015): 255-260.
[38] Bishop, C. "Pattern Recognition and Machine Learning (Information Science and Statistics), 1st edn. 2006. corr. 2nd printing edn." (2007).
[39] McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." *The bulletin of mathematical biophysics* 5.4 (1943): 115-133.
[40] Hebb, Donald Olding. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.
[41] http://ml.informatik.uni-freiburg.de/_media/teaching/ss10/05_mlps.printer.pdf
[42] Fukushima, Kunihiko. "Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position- Neocognitron." *ELECTRON. & COMMUN. JAPAN* 62.10 (1979): 11-18.
[43] Maass, Wolfgang. "Networks of spiking neurons: the third generation of neural network models." *Neural networks* 10.9 (1997): 1659-1671.
[44] Vreeken, Jilles. "Spiking neural networks, an introduction." *Institute for Information and Computing Sciences, Utrecht University Technical Report UU-CS-2003-008* (2002).
[45] LeCun, Yann, and Yoshua Bengio. "Convolutional networks for images, speech, and time series." *The handbook of brain theory and neural networks*3361.10 (1995): 1995.
[46] Newell, A., Simon, H. A. 1976. Computer Science as Empirical Inquiry: Symbols and Search. *Communications of ACM*, 19 (3), 113-126.
[47] Sun, R. 2000. Artificial Intelligence: Connectionist and Symbolic Approaches. *Citeseer*.
[48] Newell, A. 1980. Physical Symbol Systems. *Cognitive Science*, 4, 135-138.
[49] Laird, J. E., Newell, A., Rosenbloom, P. S. 1987. SOAR: AN ARCHITECTURE FOR GENERAL

INTELLIGENCE. *Artificial Intelligence*, 33 (1), 1-64.
[50] Anderson, J. R. 1995. ACT: A Simple Theory of Complex Cognition. *American Psychologist*, 51 (4), 355-365.
[51] Langley, P. et al. 1991. A Design for the ICARUS Architecture. *ACM SIGART Bulletin*, 2 (4), 104-109.
[52] Laird, J. E. 2008. Extending the Soar Cognitive Architecture. *Proceedings of the 2008 conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference*, 224-235.
[53] Franklin, S. et at. 2012. Global Workshop Theory, its LIDA model and the underlying neuroscience. *Biologically Inspired Cognitive Architectures*, 1, 32-43.
[54] Goertzel, B. et al. 2011. Cognitive Synergy between Procedural and Declarative Learning in the Control of Animated and Robotic Agents Using the OpenCogPrime AGI Architecture. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 1436-1441.

Appendix:

Source Code 1: Integrate-And-Fire Model, Python Implementation

```
from pylab import *

tmax=1000.0
dt=0.1
T=ceil(tmax/dt)  #create a 1d array
exStart=ceil(T*0.3)
exEnd=ceil(T*0.7)
Iext=zeros(T)
Iext[0]=0.0

Vvect=zeros(T)  #membrane potential of the neuron
R=10.0  #membrane resistance = 10 Mohm
tau=50.0 #decaying time constant = 10ms
Vthre=-40.0  #threshold voltage = -55 mv
Vrest=-70.0  #resting potential = -70 mv
Vspike=40.0  #spike voltage = 40mv
Vreset=-75.0  #reset voltage = -75mv
Vinf=0.0  # voltage when t = infinite
Vvect[0]=-70.0
Iext=zeros(T)
for i in arange(T-1):
    if i>exStart and i<exEnd:
        Iext[i]=-5.0
    else:
        Iext[i]=0

for t in arange(T-1): #  loop. interval = 1
    if Vvect[t+1] != Vreset:
        Vinf=Iext[t]*R+Vrest
        Vvect[t+1]=Vinf+(Vvect[t]-Vinf)*exp(-dt/tau)
    if Vvect[t+1]>Vthre:
        Vvect[t+1]=Vspike
        Vvect[t+2]=Vreset

tvec=arange(0.,tmax,dt)
```

```
plt.figure()

plt.subplot(2,1,1)
plt.title('Membrane Potential (LIF)')
plt.plot(tvec, Vvect)
plt.xlabel('t (ms)')
plt.ylabel('Vm (mV)')
plt.axis([0,1000,-120,80])

plt.subplot(2,1,2)
plt.title('Stimulus Current (LIF)')
plt.plot(tvec,Iext)
plt.xlabel('t (ms)')
plt.ylabel('I (uA)')
plt.axis([0,1000,-10,25])

plt.tight_layout()   #can be used to arrange the layout,avoiding overlap
show()
```

Source Code 2: Hudgkin-Huxley Model, Python Implementation

```
from pylab import *

# constants
# Cm,gNa,gKa,gL, Ena,EK,EL
# parameters of constants can be found at:
# http://icwww.epfl.ch/~gerstner/SPNM/node14.html
Cm=1.0
gNa=120.0
gK=36.0
gL=0.3
ENa=50.0
EK=-77.0
EL=-54.387


# time and time interval
tmax=1000.0
dt=0.01       #iteration step, which is very important. If step is larger than 0.01s,
iteration will not converge.
T=ceil(tmax/dt)
exStart=ceil(T*0.3)
exEnd=ceil(T*0.7)

# variables initialization
# m,h,n,Vm,Iext
# alphaM,betaM,alphaN,betaN,alphaH,betaH
# dm, dh, dn
# parameters of alpha and beta can be found in M. E. Nelson
# the reference direction of these variables matter
m=zeros(T)
m=0.05
h=zeros(T)
h=0.6
n=zeros(T)
n=0.32
```

```python
Vm=zeros(T)
Vm[0]=-65.0
Iext=zeros(T)
Iext[0]=0.0
alphaM=0.0
alphaN=0.0
alphaH=0.0
betaM=0.0
betaN=0.0
betaH=0.0


# updating loops
# updating sequence: Vm-> m-> h-> n

for t in arange(T-1):
    if t>exStart and t<exEnd:
        Iext[t]=-5.0      #Iext=excitatory current. starting from t=40s, ending at t=170s. 
The strength is 20uA
    else:
        Iext[t]=0.0

    v=Vm[t]
    Iion=gNa*m**3.0*h*(v-ENa)+gK*n**4.0*(v-EK)+gL*(v-EL)
    dVm=(Iext[t]-Iion)*dt/Cm
    Vm[t+1]=Vm[t]+dVm

    v=Vm[t+1]
    alphaM=0.1*(v+40.0)/(1.0-exp(-(v+40.0)/10.0))
    betaM=4.0*exp(-(v+65.0)/18.0)
    alphaH=0.07*exp((-65.0-v)/20.0)
    betaH=1.0/(exp((-35.0-v)/10.0)+1.0)
    alphaN=0.01*(55.0+v)/(1.0-exp(-0.1*v-5.5))
    betaN=0.125*exp(-(v+65.0)/80.0)

    m=m+dt*(alphaM*(1.0-m)-betaM*m)
    h=h+dt*(alphaH*(1.0-h)-betaH*h)
    n=n+dt*(alphaN*(1.0-n)-betaN*n)

tvec=arange(0.,tmax,dt)

plt.figure()

plt.subplot(2,1,1)
plt.title('Membrane Potential (H-H)')
plt.plot(tvec, Vm)
plt.xlabel('t (ms)')
plt.ylabel('Vm (mV)')
plt.axis([0,1000,-100,80])

plt.subplot(2,1,2)
plt.title('Stimulus Current (H-H)')
plt.plot(tvec,Iext)
plt.xlabel('t (ms)')
plt.ylabel('I (uA)')
plt.axis([0,1000,-10,25])
```

```
plt.tight_layout()    #can be used to arrange the layout,avoiding overlap
show()
```

Source Code 3: Izhikevich Model, Python Implementation

```
#from numpy import array
#from numpy import ceil
#from numpy import zeros
#from numpy import arange
#from matplotlib import plot
from pylab import *

tmax=1000.0
dt=0.5
T=ceil(tmax/dt)    #total simulation length
exStart=ceil(T*0.3)  #excitatory current start time
exEnd=ceil(T*0.7)    #excitatory current end time

a=0.02
b=0.2
c=-65
d=8.

Iapp=20.0    # the amplitude of excitatory current
i =zeros(T)   #the vector used to store time course of excitatory current

v=zeros(T)
u=zeros(T)
v[0]=-70
u[0]=-14

for t in arange(T-1):
    if t>exStart and t<exEnd:
        i[t]=Iapp
    else:
        I=0

    if v[t]<35:
        dv=(0.04*v[t]+5)*v[t]+140-u[t]
        v[t+1]=v[t]+(dv+i[t])*dt
        du=a*(b*v[t]-u[t])
        u[t+1]=u[t]+dt*du
    else:
        v[t]=35
        v[t+1]=c
        u[t+1]=u[t]+d

tvec=arange(0.,tmax,dt)

plt.figure()

plt.subplot(2,1,1)
plt.title('Membrane Potential (Izhikevich)')
plt.plot(tvec, v)
plt.xlabel('t (ms)')
plt.ylabel('Vm (mV)')
plt.axis([0,1000,-100,80])
```

```
plt.subplot(2,1,2)
plt.title('Stimulus Current (Izhikevich)')
plt.plot(tvec,i)
plt.xlabel('t (ms)')
plt.ylabel('I (uA)')
plt.axis([0,1000,-10,25])

plt.tight_layout()    #can be used to arrange the layout,avoiding overlap
show()
```

Source Code 4: STDP Simulation Program, MATLAB Implementation

The following source code is written by Toby Lightheart. Code should run in MATLAB environment.

```
function stdc_sim()
%stdc_sim() - Spike timing-dependent construction simulation
%
%  function [firings, growth, SNN, inputParam] = stdc_sim()
%  Uncomment alternative function and comment out 'function stdc_sim()' to
%  get optional simulation outputs.
%
%  This function is an example implementation of the 'spike timing-
%  dependent construction' (STDC) learning algorithm; a constructive
%  approach to spike timing-dependent plasticity (STDP).  Starting with a
%  minimal network (only input neurons), neurons and connection are
%  constructed in response to the detection of temporally correlated input
%  spikes.
%
%  In this example STDC is applied to a 2-dimensional random neural field
%  that is excited by a vertical strip that travels cyclically across the
%  field in both horizontal directions.  The output of this simulation are
%  plots of the spatial location and the growth and firing of neurons.
%
%  Optional simulation outputs:
%   firings - matrix of spike time and neuron index
%   growth - matrix of growth time and neuron index
%   SNN - a data structure containing the spiking neural network
%     parameters (Izhikevich model)
%   inputParam - a data structure containing the input

%  Author: Toby Lightheart
%  Date Last Modified: 09/12/2011
%  Contact: toby.lightheart@adelaide.edu.au
%  Institution: School of Mechanical Engineering,
%       University of Adelaide, Australia

%% Initialize the simulation and algorithm variables

nMax = 100;        % Maximum number of neurons, default nMax = 100;
Ne = 20;           % Initial excitatory neurons, default Ne = 20;
n = Ne;            % Number of neurons

inputNeurons = [ones(n,1); zeros(nMax-n,1)];  % input neurons
```

```
firings = [];       % record of all spikes
growth = [];        % record growth of neurons

% Initialize spiking neural network
SNN = spiking_net_init();

% Initialize input parameters
inputParam = input_init(n);

% Initialize constructive algorithm
Tc = 10;            % time threshold for construction, default Tc = 10;

f = false(nMax,1);              % current active neurons
ft = -Tc*ones(nMax,1);          % time of last spike
C = get_binary_connections();   % binary matrix of network connections


%% Simulate spiking network with spike timing dependent construction (STDC)

TS = 1;             % time step of 1 ms - also used for input update
for t=1:TS:1000     % simulation of 1000 ms

    % update network
    spiking_net_update();

    % get neuron spike vector
    f = get_spikes(inputNeurons);      % binary vector of spikes

    % spike timing-dependent construction search
    ft = ft - f.*ft + f*t;      % update neuron spike time vector
    r = ft > (t-Tc);            % binary vector of 'recent' input spikes
    x = sum(1*((C*r - C*~r)==sum(r))); % find connection combination

    if ~x
        % add the neuron to the network
        create_neuron(r);

        % add to binary matrix of network connections
        C = get_binary_connections();
    end
end

% plot the neural field and the network growth and spikes
display_results();


%% Spiking neural network simulation and interface functions

% Spiking neural network initialisation
    function SNN = spiking_net_init()
        % SNN = spiking_net_init() - initialise the simulated 'SNN'
        %  Initialise a spiking neural network (SNN) using the 'simple
        %  model' created by E. M. Izhikevich (2003)

        % 'regular spiking' neurons
        %  default parameter values: a=0.02, b=0.2, c=-65, d=8
        a=[0.02*ones(Ne,1);     zeros(nMax-n, 1)];  % Recovery rate
```

```
        b=[0.2*ones(Ne,1);      zeros(nMax-n, 1)];  % Recovery sensitivity
        c=[-65*ones(Ne,1);      zeros(nMax-n, 1)];  % Refractory reset
        d=[8*ones(Ne,1);       zeros(nMax-n, 1)];  % Refractory recovery

        S=[zeros(nMax, n)      zeros(nMax, nMax-n)]; % Synapse weights

        % initial membrane potential, default v=-65
        v=[-65*ones(n,1);      zeros(nMax-n,1)];   % Initial values of v
        u=b.*v;                              % Initial values of u

        % Return the spiking neural network data structure
        SNN = struct('a',a, 'b',b, 'c',c, 'd',d, 'S',S, 'v',v, 'u',u);
    end

% Spiking neural network input
    function in = spiking_net_input()
        % in = spiking_net_input() - generate spiking network input 'in'
        %  Interface function between spiking neural network and the
        %  an input signal generation function:
        %    in = update_input()

        in = update_input();
    end

% Spiking neural network update
    function spiking_net_update()
        % spiking_net_update() - update the SNN for a single time step
        %  Update spiking neural network data (stored in 'SNN') using the
        %  'simple model' created by E. M. Izhikevich (2003)

        % neuron spike threshold potential
        SP_T = 30;  % default SP_T = 30;

        % update post-spike potentials
        s = (SNN.v >= SP_T);       % find neurons that have spiked
        SNN.v(s) = SNN.c(s);       % reduce potential to post-spike value
        SNN.u(s) = SNN.u(s) + SNN.d(s); % add refractoriness

        % update spiking network inputs
        I = spiking_net_input();
        I = [I; zeros(nMax-size(I,1),1)]+sum(SNN.S(:,s),2);

        % standard differential equation for neuron potential update
        %  (Izhikevich 2003): v' = 0.04*v^2 + 5*v + 140 - u + I
        %  Izhikevich implements 0.5 ms step for numerical stability
        SNN.v = SNN.v+0.5*(0.04*SNN.v.^2+5*SNN.v+140-SNN.u+I);
        SNN.v = SNN.v+0.5*(0.04*SNN.v.^2+5*SNN.v+140-SNN.u+I);

        % prevent unusual behaviour in empty network matrix locations
        SNN.v = SNN.v.*[ones(n,1); zeros(nMax-n,1)];

        % update refractoriness of neurons
        SNN.u = SNN.u+SNN.a.*(SNN.b.*SNN.v-SNN.u);

        % store spikes for graphical output
        firings = [firings; t+0*find(s>0),find(s>0)];
    end
```

```matlab
% Get network spikes from network
    function spikes = get_spikes(neurons)
        % spikes = get_spikes(neurons) - test if 'neurons' have spiked
        %  Return a binary vector 'spikes' indicating whether
        %  input parameter 'neurons' have spiked

        % neuron spike threshold potential
        SP_T = 30;  % default SP_T = 30;

        spikes = (SNN.v).*neurons >= SP_T;  % return spikes
    end

% Get binary connections of the neural network
    function bin = get_binary_connections()
        % bin = get_binary_connections() - create a binary matrix of
        % SNN connections
        %  Find all network connections that are above the threshold and
        %  return them as a binary matrix 'bin'

        % synapse weight threshold
        SW_T = 0.3;   % default SW_T = 0;

        bin = 1*(SNN.S > SW_T);    % create a binary matrix of synapses > 0
    end

% Create a new spiking neuron with given connections
    function create_neuron(req)
        % create_neuron(req) - create a neuron with 'req' connections
        %  Create a neuron with connections from 'req' in spiking neural
        %  network 'SNN'

        % Initial weight constant to control total input to new neuron
        INIT_W = 40;    % default INIT_W = 40;

        if n<nMax
            % increment the number of neurons
            n=n+1;

            % add new attributes to neural network property vectors
            %  'regular spiking' neuron
            %  default a=0.02, b=0.2, c=-65, d=8
            SNN.a(n) = 0.02;    % Recovery rate
            SNN.b(n) = 0.2;     % Recovery sensitivity
            SNN.c(n) = -65;     % Refractory reset
            SNN.d(n) = 8;       % Refractory recovery

            % initial neuron potential and refractoriness
            %  set SNN.v(n)=0 for immediate spiking after creation
            SNN.v(n)= -65;      % Initial value of v, default SNN.v(n)=-65;
            SNN.u(n)= 0; %SNN.b(n).*SNN.v(n);   % Initial value of u

            % add new neuron connection weights to synapse matrix
            SNN.S(n,:) = INIT_W/sum(req).*req';

            % record growth for graphical output
            growth=[growth; t+0*n,n];
```

```
        end
    end

%% Neuron field with spatial input

 % initialize neural network input
   function [param] = input_init(numIn)
      % [param] = input_init(numIn) - initialise network input
      %  Initialise the spiking neural network input parameters 'param'
      %  for a given number of inputs 'numIn'

      % neuron field parameters
      field = gen_neuron_field(numIn);

      % line input parameters
      line = gen_input_line();

      % return field and line
      param.f = field;
      param.l = line;
   end

 % generate a spatial representation of input neurons
   function field = gen_neuron_field(numIn)
      % field = gen_neuron_field(numIn) - generate a 2D field of neurons
      %  Generate a 2D spatial field of a total of 'numIn' neurons and
      %  return the 'field' parameters

      field.NUM = numIn;  % number of neurons
      field.SIZE = 1;     % size of the distribution area
      field.loc = field.SIZE*rand(numIn,2);   % randomise locations
   end

 % generate an activation pattern (line) for the neuron field
   function line = gen_input_line()
      % line = gen_input_line() - generate an excitatory input line
      %  Create a finite width 'line' to operate as an activation region

      line.WI = 0.1;      % width of the input line, default 0.1
      line.SP = 0.01;     % speed line travels, default 0.01
      line.SL = [0; 0.1]; % slope of line [x; y], default [0; 0.1]
      line.DI = [1; 0];   % direction of movement [x; y], default [1; 0]

      line.ST = [0; 0];   % start/restart line location, default [0; 0]
      line.loc = line.ST; % initialise the location of the line [x; y]
   end

 % update the input neuron activation
   function [in] = update_input()
      % [in, inputParam] = update_input(inputParam) - Update network
      % input line position
      %  Update the input parameters 'inputParam' and neuron activation
      %  'in'

      INC = 10;   % potential increase from input, default 10

      % update the location of the input line
```

```
        inputParam.l.loc = inputParam.l.loc ...
                          + TS * inputParam.l.DI * inputParam.l.SP;

        % change input properties at t=500 - uncomment to observe effect of
        % changing input region orientation and direction
%         if t==500
%             inputParam.l.SL = [0.1; 0];    % change line orientation
%             inputParam.l.DI = [0; 1];      % change line direction
%             inputParam.l.loc = inputParam.l.ST;    % change line location
%         end

        % reset line if exceeding field boundaries
        if any(inputParam.l.loc>inputParam.f.SIZE | inputParam.l.loc<[0;0])
            inputParam.l.DI = -inputParam.l.DI; % reverse line direction
        end

        % calculate distance using two point description of line
        p1 = inputParam.l.loc;
        p2 = inputParam.l.loc + inputParam.l.SL;

        % calculate the distance of the neuron to the line
        dist = abs((p2(1)-p1(1))*(p1(2)-inputParam.f.loc(:,2)) ...
                 - (p2(2)-p1(2))*(p1(1)-inputParam.f.loc(:,1)));
        dist = dist/sqrt((p2(1)-p1(1))^2 + (p2(2)-p1(2))^2);

        % input to neurons that are within activation distance
        in = INC*(dist < inputParam.l.WI/2);
    end

%% Plot output from the operation of the constructive algorithm

    function display_results()
        % display_results() - Display results of the simulation
        %  Plot the spatial location of input neurons and the pattern of
        %  growth and activation in the network population

        % plot the spatial location of input neurons
        figure(1);
        cla;
        plot(inputParam.f.loc(:,1),inputParam.f.loc(:,2), 'o');
        title('Neural Field');
        ylabel('y Position');
        xlabel('x Position');
        legend('Neuron');

        % plot the timing and index of neuron spikes
        figure(2);
        hold off;
        if ~isempty(firings)
            plot(firings(:,1),firings(:,2),'.');
        end

        % plot the timing and index of neuron growth
        hold on;
        if ~isempty(growth)
            plot(growth(:,1),growth(:,2),'b+', 'MarkerSize', 5);
        end
```

```matlab
        title('Neuron Spikes and Growth');
        ylabel('Neuron');
        xlabel('Time (ms)');
        legend('Spike', 'Growth');
    end

%% End of STDC
% Reference:
%   E. M. Izhikevich, Simple model of spiking neurons, IEEE Trans.
%   Neural Netw., vol. 14, no. 6, pp. 15691572, Nov. 2003.
end
```