

Artificial Neural Networks

Andres Chavez
Math 382/L
T/Th 2:00-3:40
April 13, 2010

Abstract

The main interest of this paper is Artificial Neural Networks (ANNs). A brief history of the development of the field will be given. This will be followed by a section dedicated to introducing the reader to the workings of Biological Neural Networks. The concepts of neurons and signal processing will be developed. The aim of this is to provide a basis for understanding the parallels of the Artificial and Biological Neural Nets. With this foundation set, the ideas of what ANNs are and what they consist of will be explored. Network types, learning algorithms and applications will be topics of interest. Some of the underlying mathematical structure will be presented. However, the focus will be conceptual rather than computational in nature. The paper will then conclude with a brief exploration of future possibilities and current research such as Artificial Intelligence and the Blue Brain Project.

We will begin with a synopsis of the history of Artificial Neural Networks (ANNs). “The notion of computing takes many forms. Historically, computing has been dominated by the concept of *programmed computing*, in which (usually procedural) algorithms are designed and subsequently implemented using the currently dominant architecture”(Schalkoff 1). This form of computing has proven to be very powerful. Support for this claim can be drawn from the vast technological advances that surround us today. The CPU’s (Central Processing Units) of today’s computers have procedural architecture. However powerful this form of computing is, as humans push the capabilities of technology, it has become apparent that more powerful computational abilities are necessary. For example, a computer scientist would be hard pressed to develop a “thinking machine” with strictly this programmed computing. Thus an “alternative computational architecture is needed when one considers the ‘computing’ of biological systems” (Schalkoff 1). The computation in the human brain is much different from the programmed computing, in that “the computation of the brain is massively distributed and parallel and learning replaces a priori program development.” These abilities have motivated the development of ANNs. As Schalkoff states, “ANN technology has the potential to be a

dominant computing architecture, and artificial neurons may become the ‘ultimate RISC (reduced-instruction-set computer) building block.’”

The concept of ANN’s has been around for some time. In fact, the beginnings of the field date back to the 1960’s with the work of Frank Rosenblatt, Marvin Minsky, and Seymour Papert. Rosenblatt introduced the idea of Perceptrons, which are “the simplest and most common types of neural networks (Birula135).” Perceptrons will be viewed in greater detail later in the paper. Following Rosenblatt’s work Minsky and Papert proved that “no perceptron without hidden layers can calculate the XOR (exclusive OR) function and conjectured that the same holds for more complicated perceptrons as well, which for a time significantly cooled the interest in neural networks altogether” (Birula136). Fortunately, their proof was not as much of a setback as previously thought because some time later researchers were able to develop simple Multi-Layer Perceptrons (MLP’s) that could calculate the XOR function, which in turn fueled interest again in artificial neural nets.

Now we will begin an exploration of Biological Neural Networks (BNNs) so that the reader will be better able to understand the workings of ANNs by drawing parallels between the biological networks and artificial ones. An apparent example of a BNN is the human brain, which contains over 100 billion neurons. Another example of a BNN is the nervous system of biological organisms. One may be inclined to think of a biological neural network as something of a web-like structure of neurons that communicate with one another. The same idea can be applied to ANNs. This interpretation lends to the understanding of how the network is constructed and is able to communicate. Needless to say, neural networks play a vital role in biological systems and have the potential for achieving great feats in computing. BNNs enable all the biological functions of animals to be executed simultaneously. Similarly, ANNs provide

computers with the ability to multi-task or rather compute in a parallel manner. BNNs are responsible for movement, the beating of a heart, contraction of lungs, sight, pain, emotion, and so on. What makes these networks so effective? After all if one would contrast the impulse reaction time of a single neuron to that of a 1GHz CPU it would be apparent that the CPU is superior. This conclusion would be drawn based on the fact that a single neuron has a reaction time of $1\text{ms}(1 \times 10^{-3}\text{s})$ while the CPU can execute one step of instruction in $1\text{ns}(1 \times 10^{-9}\text{s})$. The neuron however, overcomes this apparent inferiority. It does so because it belongs to a network of neurons and this “affiliation” allows the brain to be superior to current computers. The power of the network lies in its ability to have billions of neurons computing simultaneously. This simultaneous computing is called parallel computing and this style is the strength of ANNs. In this computing style, the neurons do not act individually but instead tackle the same problem together thus drastically increasing their overall performance.

In biological systems there are three types of neurons. Neurons have three artificial counterparts as well and they will be described in later sections. There are sensory neurons, motor neurons, and interneurons. The sensory neurons receive information from external stimuli. Some examples of external stimuli are light hitting the retina in the eye or heat from putting a hand on a hot stove. Interneurons simply pass information from neuron to neuron. These neurons can be thought of as the middleman. For example, when the neurons on a fingertip feel a pencil they transmit signals. However, the neuron in the fingertip does not simply send a signal from its location in the finger directly to the brain through thin air. Instead it is the job of the interneurons to pass the signal along through the finger, to the hand, up the arm, to the shoulder, and so on until the signal reaches the brain where it is then interpreted as a sense of feeling. Lastly, motor neurons pass information to muscles causing them to contract.

Let us explore more in depth what the neuron structure is. This background is motivated by hopes that this may help the reader draw parallels between the artificial and biological neurons thus better understanding their function.

Some of the main constituents of a biological neuron are shown in Figure 2.1 (Gurney8). These structures include the soma (labeled as the cell body), the dendrites, axon hillock, axon, and nodes of Ranvier. The dendrites are “branches” of the neuron and are the “connections” of the neurons. They are responsible for receiving signals from other neurons. These signals can be electrical or chemical. The chemical signals are referred to as neurotransmitters. According to Kevin Gurney, signal transmission is achieved because the neural membrane works to maintain an electrical imbalance of negatively and positively charged ions resulting in a potential difference across the membrane with the inside being negatively polarized by approximately 70 mV with respect to the outside. In other words the neurons work to maintain a voltage of 70 mV across their bodies.

A single neuron can have thousands, even hundreds of thousands of connections to other neurons. Given that there are over 10^{11} (100 billion) neurons in the brain, it is apparent that the neural networks have a capability for very powerful processing. A critical structure of the neuron for signal processing is the dendrite. As previously stated, the dendrites are branch like receiving structures on the neuron. They receive input signals from other neurons at what is called the post-synaptic membrane. Once input is received via the dendrites, the signal travels through the cell body (the soma) and arrives at the axon hillock. This signal is electrical and is referred to as a postsynaptic potential (PSP). The white and darkened arrows in Figure 2.1 show the path of the PSPs: ANNs mirror this signal movement. “The PSPs can serve either to depolarize the membrane from its negative resting state towards 0 volts, or to hyperpolarize the

membrane to an even greater negative potential” (Gurney10). The contributory PSPs at the axon hillock exist for an extended time before they eventually decay such that if two PSPs arrive out of synchronization, they may still interact in the summation process. However, Gurney further explains that some PSPs may travel from distant dendrites and could possibly arrive at the axon hillock after another PSP had already decayed. On this basis he draws the conclusion that a “neuron sums or integrates its PSPs over both space *and* time.” A question that may naturally arise is: how can a neuron sum? The answer is that each PSP contributes to the membrane potential. Thus the sum is resulting changes in the potential of the neuron. The changes occur because some PSPs are inhibitory while others are excitatory. Meaning, the inhibitory PSPs act in a manner that reduce the likelihood of the neuron firing, while the excitatory signals act in a manner that increase the likelihood of the neuron firing. Whether or not the PSP is inhibitory or not is dependent on the physical nature of the neuron that generated it. This “information” regarding the inhibitory nature of the PSP is contained in what is called the synaptic strength of the neuron. These strengths are not set in stone. In fact, they change with normal brain processes and their transient nature is believed to be the basis for learning and memory in the brain. This theory of learning and memory, known as Hebbian theory after Canadian psychologist Donald Hebb, is the foundation for cognitive psychology and plays an important role in Artificial Neural Networks. To continue with the PSP contribution to the membrane potential, the axon hillock as stated previously is the location where the input signals are summed. This summation is represented as a change in the membrane potential and when this potential “exceeds a certain threshold (typically -50 mV), an action potential (i.e. a signal) is generated and propagates down the axon, along any collaterals, eventually reaching axon

terminals resulting in a shower of synaptic events at neighboring neurons ‘downstream’ of the original neuron” (Gurney 11).

A fatty substance called the myelin sheath, which surrounds the axon, aids the propagation of the action potential. Instead of completely covering the axon, the myelin sheath is broken into 1mm intervals where the segments absent of myelin are called nodes of Ranvier. These nodes are crucial to the propagation of the action potential because these nodes allow the signal to jump from node to node thus speeding up the transfer of information down the axon to the next neuron. It is useful now to summarize what has been explained of biological neural networks and neuron function to facilitate the understanding of the structure and function of their artificial counterparts.

First, individually, neurons do not have much processing ability, but instead the power of the network lies in its ability to compute in a parallel manner. Second, “signals are transmitted between neurons by action potentials, which have a stereotypical profile (pulse-like spikes) and display an all-or-nothing character; there is no such thing as half an action potential”(Gurney11). Additionally, the input that neurons receive, affects the memory or synaptic strength of each neuron allowing it to remember and learn. “PSPs may be excitatory or inhibitory and are summed together at the axon hillock with the result expressed as its membrane potential” (Gurney11). Lastly, if this potential exceeds a threshold, an action potential is initiated that proceeds along the axon, which is how neurons communicate.

In beginning our exploration of Artificial Neural Networks, we shall begin by defining what an Artificial Neural Network is and drawing some parallels between the biological nets and the artificial nets. Schalkoff defines ANNs as follows:

“Artificial Neural Network: A structure (network) composed of a number of interconnected units (artificial neurons). Each unit has an input/output (I/O) characteristic and implements a local computation or function. The output of any unit is determined by its I/O characteristic, its interconnection to other units, and (possibly) external inputs. Although “hand crafting” of the network is possible, the network usually develops an overall functionality through one or more forms of training”(Schalkoff2).

This definition represents a system that is very similar to that of a Biological Neural Network. Both network types function on the basis of neurons receiving information (inputs) and communicating with other neurons by sending signals (outputs) through the network. Also, just as there are three biological neuron types, there are three artificial neuron types. The artificial neuron types are referred to as nodes. There are input nodes, output nodes, and internal (hidden) nodes. One can think of the input nodes as the sensory neurons, the output nodes as the motor neurons, and the internal nodes as the interneurons. As defined by Birula there are two difficulties in ANN problem solving. Namely, design and teaching. The former is self evident while the latter may be some what puzzling at the moment. These two notions will be developed in more detail shortly. For now they should remain as after thoughts.

The first ANN we will explore is the Multi-Layered Perceptron (MLP). MLPs are feed-forward ANNs; that is they do not contain any cycles. In feed-forward ANNs information to the input nodes passes through only once until it reaches the output layer. “An MLP is organized into a series of layers: the input layer, a number of hidden layers (layers of internal nodes), and an output layer” (Birula135). Figure 13.3 (Birula136) depicts the direction of information and how each node in a layer is connected to each node in the neighboring layers. An MLP

processes information (signals) according to the algorithm in Figure 13.4 (Birula137). Figure 13.4 (Birula137) introduces the concept of connection weights. These connection weights parallel the ideas of synapse strengths in BNNs and are therefore the way that the artificial net learns and stores memory. The weights are said to be responsible for what the ANN learns and remembers because as the ANN is trained the weights are adjusted in a manner so that the artificial neural network becomes well behaved. In other words “teaching Neural Networks is the process of adjusting the network’s connection weights in order to make it function in the desired way” (Birula138). The weights are important because they literally are multiplied with the input signals. Just as the PSPs are summed at the axon hillock in the biological neuron, so too are the binary inputs (0’s and 1’s) summed at the nodes in the ANN. Thinking about the connection weights in a biological sense, they act to make the binary inputs inhibitory or excitatory. Analogously, once a threshold is exceeded in the artificial neuron, it will fire as the biological neurons do thus contributing its “knowledge” to the rest of the network. We will now address the problem of teaching the network.

The act of teaching is “the adjusting of weights on the basis of some sample input so that the network will learn to serve its purpose” (Birula135). Learning is implemented via learning algorithms which adjust weights automatically. There are three types of learning: supervised, reinforcement, and unsupervised. “Perhaps the simplest learning algorithm comes from John Hopfield and is based on Hebb’s observations of actual processes going on in the brain” (Birula139). Hopfield’s algorithm is based on Hebb’s theory of associative memory-“the memory which associates remembered objects with each other (like one may associate the smell of lilacs with spring)” (Birula135). What Hopfield did was create a network that acts as an autoassociative memory, which means that it associates objects with themselves. Given an input

pattern, it produces the same pattern as output. As useless as this might seem it is in fact the very opposite because “neural networks tend to be fault tolerant. So when a slightly *corrupted* version of a memorized pattern is inputted, the network will still output the original memorized pattern.” (Birula135). The usefulness of this becomes apparent when one tries to scan a document. What happens is that the network removes the error that is introduced during scanning because it recognizes the characters rather than having to individually compare each character, bit by bit to characters stored in memory banks as a procedural architecture would. This advantage saves a lot of time and does not require difficult algorithm creation. In fact Hopfield’s network is a simple single layer Perceptron.

Learning algorithms require training sets and learning rates. The training set is simply a term that refers to the sample data used during the learning process and the learning rate “is a parameter that governs how big the changes to the connection weights are” (Gurney43). In the case of the scanner, its training set would be the alphabet. Another learning algorithm is the Perceptron Learning Rule (PLR), which is in fact the first learning algorithm found and was discovered by American Psychologist Frank Rosenblatt. The training set for the PLR is a set consisting of pairs of inputs and desired outputs. However, the two previously mentioned learning algorithms are limited to Single Layer Perceptrons. Since those learning algorithms are so limited, researchers developed backpropagation. “Backpropagation is similar to the PLR in the sense that it starts from random weights and uses a set of input and desired output pairs to gradually correct the connection weights” (Birula143). The difference is that the weights leading to the output layer are corrected first, then the weights before them, and so on, until the layer at the bottom is reached. “The order of correcting the weights is backwards with respect to the order in which the signals are calculated when the network performs its task. Backpropagation is

the most commonly used learning algorithm for MLPs, and probably for neural networks in general” (Birula143).

In addition to feed-forward ANNs there are Recurrent ANNs. As one might assume, recurrent networks contain cycles, so that information can pass back and forth between nodes. This feature makes recurrent networks more powerful, but simultaneously more difficult to find learning algorithms for. The simplest recurrent network is an Elman network. An Elman network is a two-layer perceptron with additional nodes called context units. These context units connect to all nodes and act as if they were additional input nodes. These context units add more feedback connections and “each time the network processes some input, the state of the hidden layer is stored in the context units. This state is fed together with the input the next time the network processes something” (Birula144). While feedforward networks always produce the same output given the same input, the output of recurrent networks will depend on the current input as well as on the previous inputs. This feature is convenient when operating on data which naturally comes in a series-such as stock market data, for example.

The previously mentioned learning algorithms were supervised type learning algorithms. Supervised learning is when the network is fed examples of input data with corresponding desired outputs. This type of learning allows the network to compare its output with the desired output and make the necessary corrections to the weights so that it may better approximate what it outputs to the desired output. Another training type is reinforcement. Reinforcement is similar to supervised learning in that the network receives input data. The difference is that in a reinforcement algorithm, the network is not presented with the desired output, but is instead graded on its performance. The ANN then takes this grade and appropriately adjusts its weights

until it outputs something that earns it a high enough grade. By high enough grade we mean that the error is within an allowable tolerance.

It is important to note that these algorithms are not implemented once and then the ANN becomes well behaved. On the contrary, the ANN runs through the algorithms many times. Each run through only varies the connection weights slightly, thus the process requires many repetitions and time. Although this seems to be a con of ANNs, the time that one must wait for the ANN to learn is a small penalty in comparison to the benefits gained. Since the network learns, one does not need to spend many arduous hours, days, or weeks developing the perfect program. Instead one has an easier task in that the algorithms that must be developed need not be perfect because the ANN is fault tolerant. In other words because the ANN can “think” it will not crash should there be minor deficiencies in the code or input. Thus ANNs make difficult goals more attainable. Lastly, there is unsupervised learning. Unsupervised training involves inputting data to the ANN, but no additional information is given. Meaning the network does not receive a grade or any hints of what the desired output is. “One might wonder how unsupervised learning can teach the network to behave in the way that we want, since we do not specify what we want at any time. This is just the point; sometimes we do not even know ourselves what we want to achieve when analyzing a set of data” (Birula146). The objective of unsupervised learning is not to train the network to behave as we like, but instead to let it develop its own ideas about the data. Thus this training type can be a powerful tool for researchers because the network can help them find some pattern or classification system for the data. For example, an astronomer may have vast amounts of data from surveying the night sky for many months. It would be a nearly impossible task for him to find a pattern amongst the millions of data entries.

He may however input this data into an ANN and perhaps the ANN will discover a pattern that relates the amount of stars in a galaxy to its shape.

Now we will return to the problem of designing an ANN. By designing the network, it is meant that one must decide on the number of neurons and specify the connections between them. In other words, one must construct a directed graph representing the network. The directed graph is simply an illustration of the nodes with arrows pointing in the direction of information transfer. The task of design can be challenging. One method of overcoming this difficulty in choosing a network topology is to use an evolutionary algorithm. Scientists have found that “evolutionary algorithms prove to be a very good tool for optimizing the topology of a neural network” (Birula146). Evolutionary algorithms are adaptive algorithms based on principles of biological evolution. The way evolutionary algorithms help in the design of ANNs is that they, in a sense, enforce natural selection on the network. They do so by assigning to each node, a fitness score. This score is based on how beneficial the individual node is to the solution of the problem presented to the network. The algorithm views the population of nodes and by giving these scores it weeds out the weak links, thus optimizing the topology of the network.

It would be appropriate now to summarize what we have learned of Artificial Neural Networks. ANNs mirror biological neural systems in that they consist of neurons (nodes in the artificial case) and the nodes communicate via signals (where the signals are binary inputs and outputs). Also, the ANNs can learn and remember. Where these abilities are contained in the connection weights of the nodes

Although ANNs are powerful processors, they are still no match for a human brain. Current ANNs cannot think the way humans do. They are limited in how much they can learn because the learning algorithms that have been developed are specialized training regimens.

Meaning that the algorithms fall short of providing the network with an ability to learn new things throughout its lifetime. However, scientists are vigorously tackling this problem. Researchers on the Blue Brain project are working to develop a microcircuit that mimics the neocortical column (NCC) of a rat. As defined on the project's website "The Blue Brain Project is an attempt to reverse engineer the brain, to explore how it functions and to serve as a tool for neuroscientists and medical researchers." As of today the team has successfully rendered a cellular-level model of the neocortical column. The project is not specifically trying to create AI, but is trying to understand the emergence of mammalian intelligence. The group will be exploring the ability of its modeled NCC to work as a Liquid Computer (a form of analog computer that handles continuous data streams). "This could be used for dynamic vision and scene segmentation, real-time auditory processing, as well as sensory-motor integration for robotics. Another special ability of the neocortex is the ability to anticipate the future based on current data (the birth of cognition) and so we will examine the ability of the NCC to make intelligent predictions on complex data" (Blue Brain). The Blue Brain project promises to make many contributions to the fields of neuroscience, computer science, psychology and many others.

Based on the ANNs obvious mimicry of organic brains it is natural to ponder the idea of Artificial Intelligence (AI). Current ANNs are far from being even remotely close to actually thinking. After all, there are no R2D2's running around, but "the main objectives of AI are to develop methods and systems for solving problems, usually solved by the intellectual activity of humans, for example, image recognition, planning, and prediction ... to develop models which simulate living organisms and the human brain in particular" (Kasabov1). From this we may conclude that the ANNs we have discussed are forms of AI. Thus we are making strides in the direction of developing conscious machines. Needless to say there still remains the opportunity

for someone to leave a mark on the field equivalent to what Einstein did with his theories of Special and General Relativities.

It is hoped by now that the great potential of artificial neural networks has become apparent. Given their wide range of applications from stock market analysis to optical character recognition it should not seem a far stretch to conclude that ANNs will continue to draw the efforts of researchers. Much of their intrigue lies in their potential. It is a peculiarity that we make pieces of silicon and aluminum act as organic tissues do. Not only do Artificial Neural Networks raise interesting questions about their computational abilities and applications to Computer Science or Engineering, but they also raise philosophical issues. For instance, they raise the issue of what it means to be conscious because if machines can be made to be conscious, then perhaps that implies humans are not as special as they would like to believe. From their implementation in scanners to being a foundation for the Internet, ANNs have greatly contributed to the modernization of the globe and if history is any indication, the further advancement of technology will be closely connected to improvements in Artificial Neural Networks.

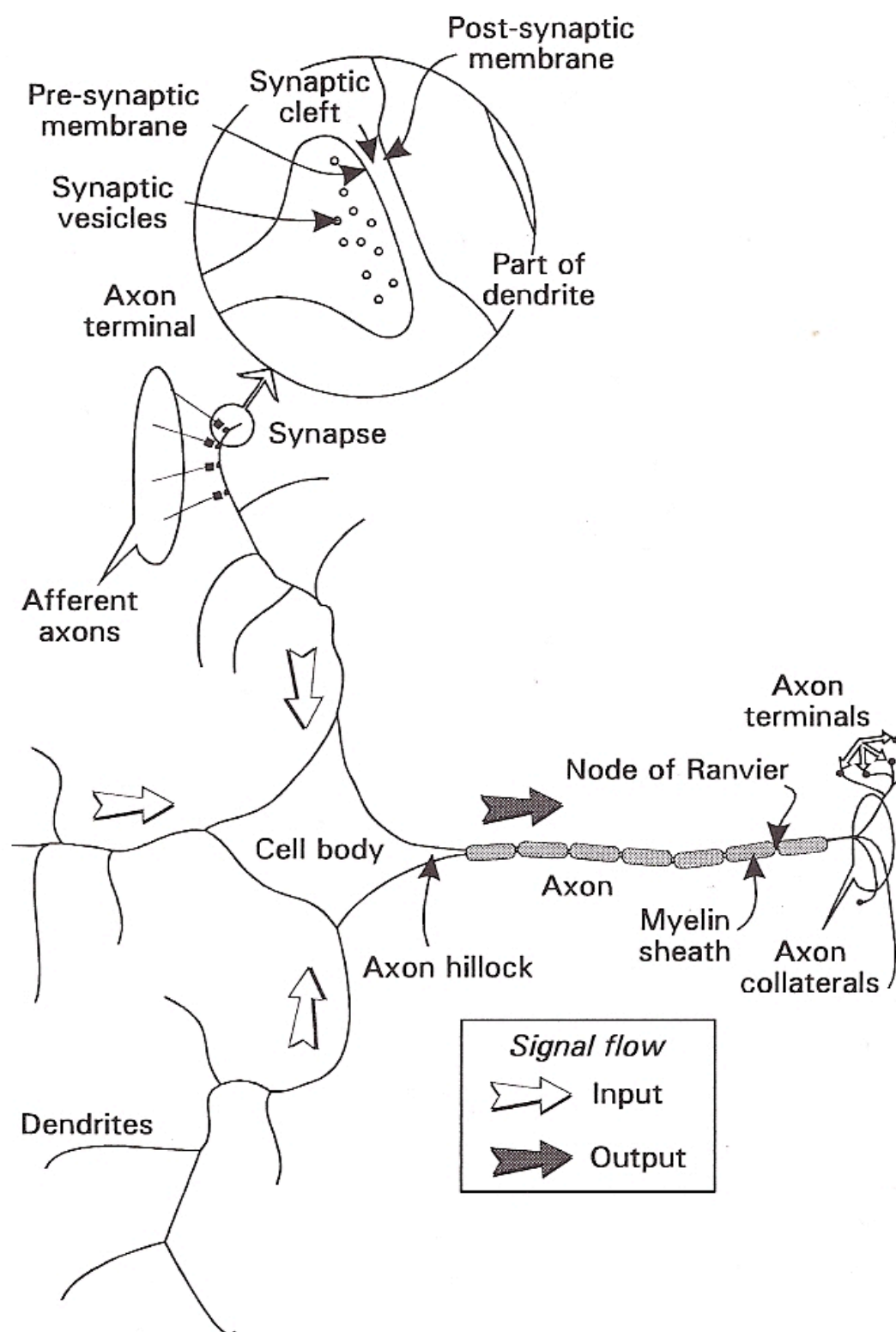


Figure 2.1 Biological neuron.

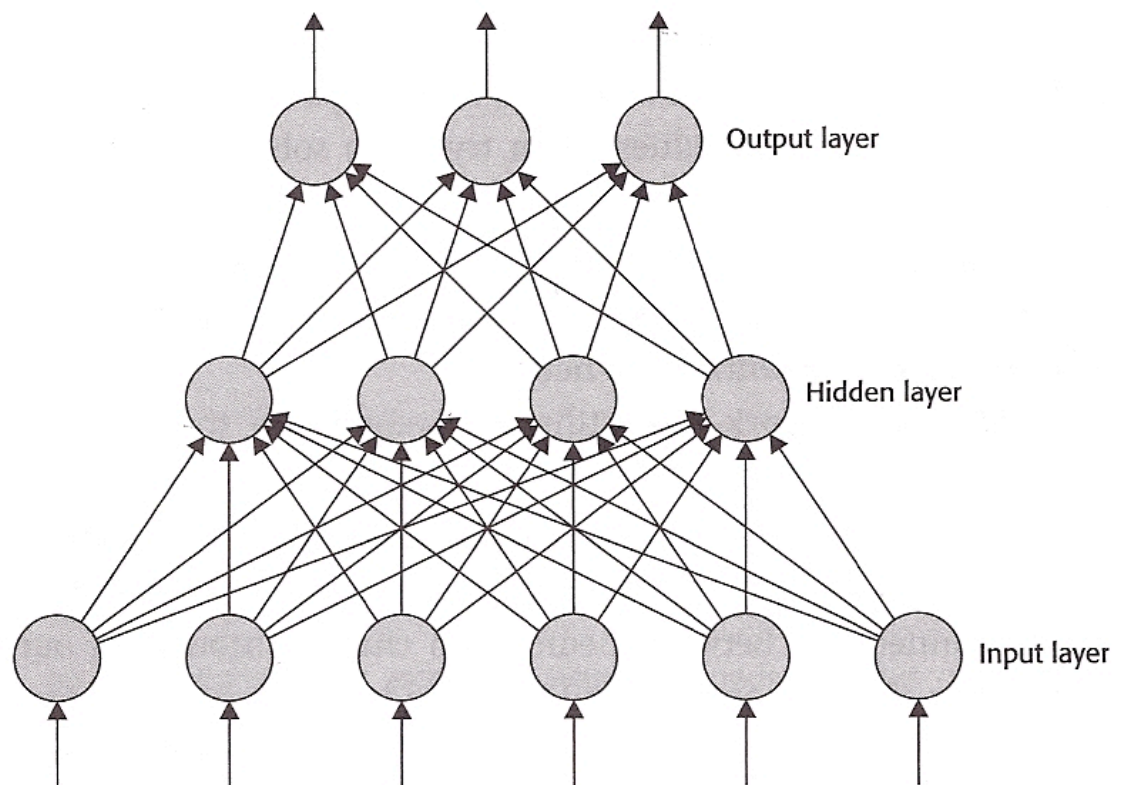


Fig. 13.3 A two-layer perceptron.

- For each node N_i^1 in the input layer (layer number 1), set its signal S_i^1 to I_i , where I_i is the i th component of the input signal.
- For each layer k , starting from the layer after the input layer (layer number 2) and ending with the output layer (layer number M):
 - For each node N_i^k in layer k , sum the incoming signals multiplied by the corresponding connection weights, apply the activation function f , and set the signal of the node S_i^k to the result. In other words, for each node N_i^k in layer k , set its signal to

$$S_i^k = f \left(\sum_{j=1}^{L_{k-1}} w_{j,i}^{k-1} S_j^{k-1} \right),$$

where L_{k-1} is the number of nodes in layer $k - 1$ and $w_{j,i}^{k-1}$ is the weight of the connection from node N_j^{k-1} to node N_i^k .

- For each node N_i^M in the output layer (layer number M), copy its signal S_i^M to O_i , where O_i is the i th component of the output signal.

Fig. 13.4 The algorithm used by an $(M - 1)$ -layer perceptron to transform the L_1 -long input vector $\{I_1, I_2, \dots, I_{L_1}\}$ into the L_M -long output vector $\{O_1, O_2, \dots, O_{L_M}\}$.

Works Cited

Birula-Bialynicki, Iwo and Bialynicka-Birula. Modeling Reality, How Computers Mirror Life.

Oxford University Press Inc. New York 2004

Blue Brain. <http://bluebrain.epfl.ch/>

Gurney, Kevin. An Introduction to Neural Networks. University College London Press 1997.

Kasabov, Nikola K. Foundations of Neural Networks, Fuzzy Systems, and Knowledge

Engineering. Massachusetts Institute of Technology. 1996

Schalkoff, Robert J. Artificial Neural Networks. McGraw-Hill Companies. 1997