2. An Artificial Neural Networks' Primer

2.1 Chronicle of Artificial Neural Networks Development

According to Nelson and Illingworth [1991], the earliest attempt to understand the human brain goes back centuries. They cite information given by Fischler and Firschein [1987] who refer to the work of Hippocrates, and the less familiar Edward Smith Papyrus; a treatise written around 3000 BC that described the location of certain sensory and motor control areas in the brain. For the most part of history, since the days of ancient Greek philosophers such as Plato and Aristotle, the study of the brain has been limited to the philosophical question of whether the mind and the body are one. As Rich and Knight state in the beginning of their book, *Artificial Intelligence*, "Philosophy has always been the study of those branches of knowledge that were so poorly understood that they had not yet become separate disciplines in their own right". This was certainly true with modern brain theory and the eventual development of Artificial Neural Networks (ANNs). Technology to enable the study of the workings of the brain was not available until the late nineteenth century. Since then, ANNs have had a very rocky climb to fame. There are four distinct periods of their development to their current status. Eberhart and Dobbins [1990] classified them in the following order:

- 1890-1969 The Age of Camelot
- 1969-1982 The Dark Age (Depression Age)
- 1982-1986 The Renaissance
- 1986-Current The Age of Neoconnectionism

The first period began in the late nineteenth century with the advent of modern science and the pursuit for better understanding of the workings of the brain. As technology improved, psychologists and biologist were able to start hypothesizing on *how* rather than *why* the human brain functions. Most ANNs literature places the beginning of the ANNs and modern brain theory era with the publication of a text by William James[6] entitled "Psychology (Briefer Course)" [James 1890]. The text contained many insights into brain activities and was the precursor of many of the current theories.

It was some fifty years later before the next major breakthrough came in 1943, when McCulloch and Pitts presented their first model of a biological neuron [McCulloch and Pitts 1943]. They developed theorems related to models of neuronal systems based on the knowledge of the biological structure at the time. Their models could solve any finite logical expressions, and, since James, they were the first authors who proposed a massively parallel neural model. However, their models could not "learn" as they used only fixed weights. Donald Hebb [1949], an eminent psychologist, added to this knowledge with his hypothesis of how the neurons communicated and stored knowledge in the brain structure. This hypothesis became known as *Hebbian Learning Rule* and enabled the eventual development of learning rules for the McCulloch-Pitts neural models.

This period peaked in 1958 when Frank Rosenblatt published his landmark paper [Rosenblatt 1958] that defined a neural network structure called the *perceptron*. Rosenblatt was inspired by the way the eye functioned and built his *perceptron* model based on it. He

[6] According to Eberhart and Dobbins [1990], James was considered by many to be the greatest American.

incorporated learning based on the Hebbian Learning Rule into the McCulloch-Pitts neural model. The tasks that he used the perceptron to solve were identifying simple pattern recognition problems such as differentiating sets of geometric patterns and alphabets. The Artificial Intelligence community was excited with the initial success of the perceptron and expectations were generally very high with the perception[7] of the perceptron being the panacea for all the known computer problems of that time. Bernard Widrow and Marcian Hoff contributed to this optimism when they published a paper [Widrow and Hoff 1960] on ANNs from the engineering perspective and introduced a single neuron model called ADALINE that became the first ANN to be used in a commercial application. It has been used since then as an adaptive filter for telecommunication to cancel out echoes on phone lines. The ADALINE used a learning algorithm that became known as the delta rule[8]. It involves using an error reduction method known as gradient descent or steepest descent.

However, in 1969, Marvin Minsky and Samuel Papert, two well renown researchers in the Artificial Intelligence field, published a book entitled 'Perceptron' [Minsky and Papert 1969], criticizing the perceptron model, concluding that it (and ANNs as a whole) could not solve any real problems of interest. They proved that the perceptron model, being a simple linear model with no hidden layers, could only solve a class of problems known as *linearly separable* problems. One example of a non-linearly separable problem that they proved the perceptron model was incapable of solving is the now infamous exclusive-or[9] and its generalization, the parity detection problem. Rosenblatt did consider multilayer perceptron models but at that time, a learning algorithm to train such models was not available.

This critique, coupled with the death of Rosenblatt in a boat accident in 1971 [Masters 1993], cast doubt on the minds of research sponsors and researchers alike on the viability of developing practical applications from Artificial Neural Networks. Funds for ANNs research dried up, and many researchers went on to pursue other more conventional Artificial Intelligence technology. In the prologue of the recent reprint of 'Perceptron', Minsky and Papert [1988, pp. vii-xv][10] justified their criticism of the perceptron model and pessimism of the ANNs field at that time by claiming that the redirection of research was "no arbitrary diversion but a necessary interlude". They felt that more time was needed to develop adequate ideas about the representation of knowledge before the field could progress further. They further claimed that the result of this diversion of resources brought about many new and powerful ideas in symbolic AI such as relational databases, frames and production systems which in turned, benefited many other research areas in psychology, brain science, and applied expert systems. They hailed the 1970s as a golden age of a new field of research into the representation of knowledge. Ironically, this signaled the end of the second period of ANN development and the beginning of the Dark Ages for ANNs research.

---

[7] Pardon the pun!

[8] This algorithm is also known as the Widrow-Hoff or Least Mean Squares method. An extension of this algorithm is used today in the back-propagation algorithm.

[9] The exclusive-or (XOR) problem and linear separability issue is discussed in more detail in Chapter 3.

[10] Interestingly, the reprint of the 'Perceptron' was dedicated by the authors to the memory of Frank Rosenblatt.

However, pockets of researchers such as David Rumelhart at UC San Diego (now at Stanford University), Stephen Grossberg at Boston University, Teuvo Kohonen in Finland and Kunihiko Fukushima in Japan, persisted with their research into Artificial Neural Networks. Their work came into fruition in the early 1980s, an era that many deemed as the Renaissance period of ANNs. John Hopfield of the California Institute of Technology, a prominent scientist, presented a paper [Hopfield 1984] at the Academy of Science on applying ANNs to the infamous 'traveling salesman problem'. It was his ability to describe his work from the point of a scientist coupled with his credibility, that heralded the gradual re-acceptance of ANNs. Interest grew from researchers from a multitude of fields, ranging from biologists to bankers, and engineers to psychologists. This era culminated with the publication of the first of the three volume, now famous reference text on ANNs, 'Parallel Data Processing' by Rumelhart et al. [1986b]. The authors had proposed the 'back-propagation' learning algorithm in an earlier publication [1986a] that was popularized by the text. The back-propagation algorithm overcame some of the pitfalls of the perceptron model that were pointed out by Minsky and Papert by allowing multi-layer perceptron models to learn. According to Ripley [1993], the back-propagation algorithm was originally discovered by Bryson and Ho [1969] and Werbos [1974] but did not gain prominence until it was rediscovered and popularized by Rumelhart et al. According to Eberhart and Dobbins [1990], it is hard to overstate the effect the Parallel Data Processing (PDP) books had on neural network research and development. They attribute the success of the books in one sentence: "The books presented everything practical there was to know about neural networks in 1986 in an understandable, usable and interesting way; in fact, 1986 seemed to mark the point at which a 'critical mass' of neural network information became available".

The current era begins where the PDP books left off and has been called the Age of Neoconnectionism by Cowan and Sharp [1988]. In this era, there has being a growing number of commercial ANN applications as well as continued prolific research interest from a wide field of disciplines in ANNs, as evident by the number of publications and conferences on ANNs. Sejnowski and Rosenburg's [1987] success on their NETtalk ANN-based speech generation program that teaches itself to read out aloud and subsequent work by Martin [1990] on an ANN-based handwriting recognition to recognize zip codes for the US Post Office, spurred on the prominence of ANNs as a potential application tool for handling difficult tasks. The significant improvements in computer technology as well as the rapid reduction in the cost of high powered computers have resulted in making the development of ANNs applications a universally attractive and affordable option

2.2 Biological Background

ANNs were inspired by the biological sciences, particularly the neurological sciences, as discussed in the section on the chronicle of their development. However, ANNs resemblance to their biological counterparts are limited to some borrowed concepts from the biological networks, mainly for their architecture. They are still far from resembling the workings of the simplest biological networks, due to the enormous complexity of the biological networks.

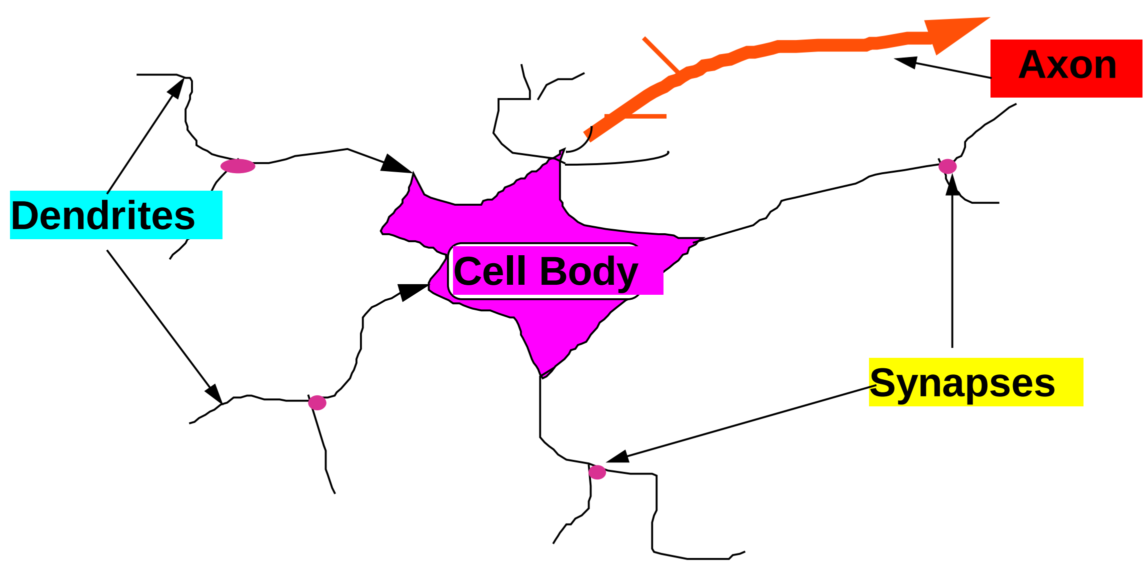
The cells found in the human brain and nervous system are known as *neurons*. Information or signals are transmitted out unidirectionally through connections between neurons known as *axons*. Information is received by a neuron through its *dendrites*. The human brain consist of around 100 billion neurons and over 10^{14} synapses. Neurons communicate with each other through *synapses* which are gaps or junctions between the connections. The transmitting side of the synapses release neurotransmitters which are paired to the neuroreceptors on the receiving side of the synapses. Learning is usually done by adjusting existing synapses, though some learning and memory functions are carried out by creating new synapses. In the human brain, neurons are organized in clusters and only several thousands or hundred of thousands participate in any given task. Figure 2-1 shows a sample neurobiological structure of a neuron and its connections.

The axon of a neuron is the output path of a neuron that branches out through axon collaterals which in turn connect to the dendrites or input paths of neurons through a junction or a gap known as the synapse. It is through these synapses that most learning is carried out by either exciting or inhibiting their associated neuron activity. However, not all neurons are adaptive or plastic. Synapses contain neurotransmitters that are released according to the incoming signals. The synapses excite or inhibit their associated neuron activity depending on the neurotransmitters released. A biological neuron will add up all the activating signals and subtract all the inhibiting signals from all of its synapses. It will only send out a signal to its axon if the difference is higher than its threshold of activation.

The processing in the biological brain is highly parallel and is also very fault tolerant. The fault tolerance characteristic is a result of the neural pathways being very redundant and information being spread throughout synapses in the brain. This wide distribution of information also allows the neural pathways to deal well with noisy data.

A biological neuron is so complex that current super computers cannot even model a single neuron. Researchers have therefore simplified neuron models in designing ANNs.

Figure 2-1
A typical biological neuron



2.3 Comparison to Conventional Computational Techniques

ANNs differ from conventional computational techniques in that the system builder of an ANN is not required to write programs, hence, there is no necessity for system builder to know a priori the necessary rules or models that are required to perform the desired task. Instead, a system builder trains an ANN to 'learn' from previous samples of data in much the same way that a teacher would teach a child to recognize shapes, colors, alphabets, etc. The ANN builds an internal representation of the data and by doing so 'creates' an internal model that can be used with new data that it has not seen before.

Existing computers process information in a serial fashion while ANNs process information in parallel. This is why even though a human brain neuron transfers information in the milliseconds (10^{-3}) range while current computer logic gates operate in the nanosecond (10^{-9}) range, about a million times faster, a human brain can still process a pattern recognition task much faster and more efficiently than the fastest currently available computer. The brain has approximately 10^{11} neurons and each of these neurons acts as a simple processor that processes data concurrently; i.e. in parallel.

Tasks such as walking and cycling seem to be easy to humans once they have learned them and certainly not much thought is needed to perform these tasks once they are learnt. However, writing a conventional computer program to allow a robot to perform these tasks is very complex. This is due to the enormous quantity of data that must be processed in order to cope with the constantly changing surrounding environment. These changes require frequent computation and dynamic real time processing. A human child learns these tasks by trial and error. For example, in learning to walk, a child gets up, staggers and falls, and keeps repeating the actions over and over until he/she has learned to walk. The child effectively 'models' the walking task in the human brain through constant adjustments of the synaptic strengths or weights until a stable model is achieved.

Humans (and neural networks) are very good at pattern recognition tasks. This explains why one can usually guess a tune from just hearing a few bars of it or how a letter carrier can read a wide variety of handwritten address without much difficulty. In fact, people tend to always associate their senses with their experiences. For example, in the 'Wheel of Fortune' game show, the contestants and viewers are usually able to guess a phrase correctly from only a few visible letters in a phrase. The eyes tend to look at the whole phrase, leaving the brains to fill in the missing letters in the phrase and associate it with a known phrase. Now, if we were to process this information sequentially like a serial computer; i.e., look at one visible character at a time; and try to work out the phrase, it would be very difficult. This suggests that pattern recognition tasks are easier to perform by looking at a whole pattern (which is more akin to neural network's parallel processing) rather than in sequential manner (as in a conventional computer's serial processing).

In contrast, tasks that involve many numerical computations are still done faster by computers because most numerical computations can be reduced to binary representations that allow fast serial processing. Most of today's ANN programs are being simulated by serial computers, which is why speed is still a major issue for ANNs, specifically the training time. There are a growing number of ANN hardware[11] available in the market today including personal computer-based ones like the Intel's Ni1000 and the Electronically Trainable Artificial Neural Network (ETANN), the IBM's ZISC/ISA Accelerator for PC and the Brainmaker Professional CNAPS™ Accelerator System. These ANN hardware process information in parallel, but the costs and the learning curves

[11] See Lindsey and Lindblad [1994, 1995] and Lindsey et. al.'s [1996] for a comprehensive listing of commercial ANN hardware.

required to use them are still quite prohibitive. Most researchers are of the view that in the near future, a special ANN chip will be sitting next to the more familiar CPU chip in personal computers, performing pattern recognition tasks such as voice and optical character recognition.

2.4 ANN Strengths and Weaknesses

ANNs are easy to construct and deal very well with large amounts of noisy data. They are especially suited to solving nonlinear problems. They work well for problems where domain experts may be unavailable or where there are no known rules. ANNs are also adaptive in nature. This makes them particularly useful in fields such as finance where the environment is potentially volatile and dynamic.

They are also very tolerant of noisy and incomplete data sets. Their robustness in storing and processing data, earned them some applications in space exploration by NASA, where fault tolerant types of equipment are required. This flexibility derives from the fact that information is duplicated many times over in the many complex and intricate network connections in ANNs, just like in the human brain. This feature of ANNs is, in contrast to the serial computer[12] where if one piece of information is lost, the entire information set may be corrupted.

The training process of an ANN itself is relatively simple. The pre-processing of the data, however, including the data selection and representation to the ANN and the post-processing of the outputs (required for interpretation of the output and performance evaluation) require a significant amount of work[13]. However, constructing a problem with ANNs is still perceived to be easier than modeling with conventional statistical methods. There are many statisticians who argue that ANNs are nothing more than special cases of statistical models, and thus the rigid restrictions that apply to those models must also be applied to ANNs as well. However, there are probably more successful novel applications using ANNs than conventional statistical tools. The prolific number of ANNs applications in a relatively short time could be explained by the universal appeal of the relatively easy methodology in setting up an ANN to solve a problem. The restrictions imposed by many equivalent statistical models is probably less appealing to many researchers without a strong statistical background. ANN software packages are also relatively easier to use than the typical statistical packages. Researchers can successfully use ANNs software packages without requiring full understanding of the learning algorithms. This makes them more accessible to a wider variety of researchers. ANN researchers are more likely to learn from experience rather than be guided by statistical rules in constructing a model and thus they may be implicitly aware of the statistical restrictions of their ANN models.

---

[12] Serial computers are also called Von Neumann computers in computer literature.

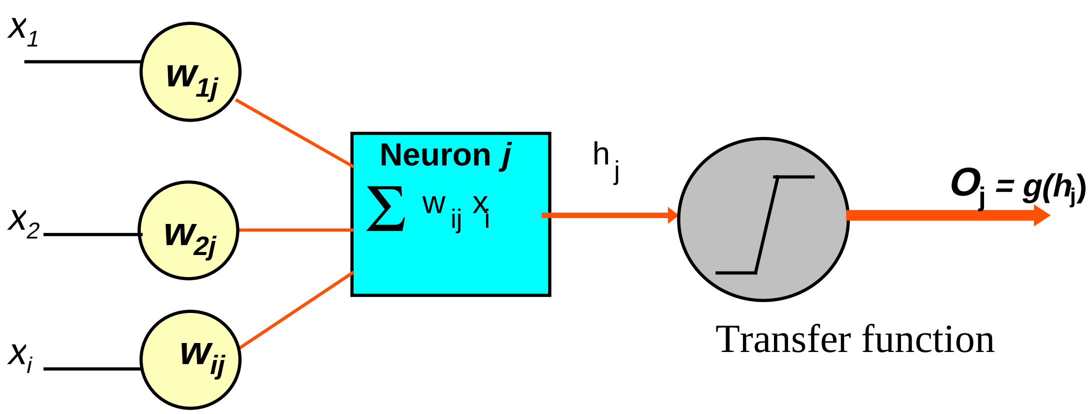
[13] The old adage of garbage in, garbage out holds especially true for ANN modeling. A well-known case in which an ANN learned the incorrect model involved the identification of a person's sex from a picture of his/her face. The ANN application was trained to identify a person as either male or female by being shown various pictures of different persons' faces. At first, researchers thought that the ANN had learnt to differentiate the face of a male from a female by identifying the visual features of a person's face. However it was later discovered that the pictures used as input data showed all the male persons' heads nearer to the edge of the top end of the pictures, presumably due to a bias of taller males in the data than females. The ANN model had therefore learned to differentiate the sex of a person by the distance his/her head is from the top edge of a picture rather than by identifying his/her visual facial features.

The major weakness of ANNs is their lack of explanation for the models that they create. Research is currently being conducted to unravel the complex network structures that are created by ANN. Even though ANNs are easy to construct, finding a good ANN structure, as well as the pre-processing and post processing of the data, is a very time consuming processes. Ripley [1993] states 'the design and learning for feed-forward networks are hard'. He further quoted research by Judd [1990] and Blum and River [1992] that showed this problem to be NP-complete[14].

2.5 Basic Structure of an ANN

The basic structure of an ANN consists of artificial *neurons*[15] (similar to biological neurons in the human brain) that are grouped into *layers*[16]. The most common ANN structure consists of an input layer, one or more hidden layers and an output layer. A modified simple model of an artificial neuron is shown in Figure 2-2.

Figure 2-2
An Artificial Neuron



In the human brain, neurons communicate by sending signals to each other through complex connections. ANNs are based on the same principle in an attempt to simulate the learning process of the human brain by using complex algorithms. Every connection has a weight attached which may have either a positive or a negative value associated with it. Positive weights activate the neuron while negative weights inhibit it. Figure 1 shows a network structure with *inputs* (x_1, x_2, ...x_i) being connected to neuron *j* with weights (w_{1j}, w_{2j},...w_{ij}) on each connection. The neuron sums all the signals it receives, with each signal being multiplied by its associated weights on the connection.

---

[14] NP (Non-Polynomial)-complete problems as mentioned in Chapter 1, are a set of very difficult problems.

[15] There is no standardization of terminology in the artificial neural network field. However, the Institute of Electrical and Electronic Engineers currently have a committee looking into it. Other terminology that has been used to describe the artificial neuron include processing elements, nodes, neurodes, units, etc.

[16] In some ANN literature the layers are also called *slabs*.

This output (h_j) is then passed through a transfer (activation) function, *g(h)*, that is normally non-linear to give the final output O_j. The most commonly used function is the sigmoid (logistic function) because of its easily differentiable properties[17], which is very convenient when the back-propagation algorithm is applied. The whole process is discussed in more detail in chapter 3.

The back-propagation ANN is a feed-forward neural network structure that takes the input to the network and multiplies it by the weights on the connections between neurons or nodes; summing their products before passing it through a threshold function to produce an output. The back-propagation algorithm works by minimizing the error between the output and the target (actual) by propagating the error back into the network. The weights on each of the connections between the neurons are changed according to the size of the initial error. The input data are then fed forward again, producing a new output and error. The process is reiterated until an acceptable minimized error is obtained. Each of the neurons uses a transfer function[18] and is fully connected to nodes on the next layer. Once the error reaches an acceptable value, the training is halted. The resulting model is a function that is an internal representation of the output in terms of the inputs at that point. A more detailed discussion of the back-propagation algorithm is given in chapter 3.

2.6 Constructing the ANN

Setting up an ANN is essentially a six step procedure.

Firstly, the data to be used need to be defined and presented to the ANN as a pattern of input data with the desired outcome or target.

Secondly, the data are categorized to be either in the training set or validation (also called test and out-of-sample) set. The ANN only uses the training set in its learning process in developing the model. The validation set is used to test the model for its predictive ability and when to stop the training of the ANN.

Thirdly, the ANN structure is defined by selecting the number of hidden layers to be constructed and the number of neurons for each hidden layer.

Fourthly, all the ANN parameters are set before starting the training process. The ANN parameters are discussed briefly in the next section and in more detail in chapter 3.

Next, the training process is started. The training process involves the computation of the output from the input data and the weights. The backpropagation algorithm is used to

---

[17] The sigmoid (logistic) function is defined as $O_{pj} = \frac{1}{1+e^{-net_{pj}}}$. In the ANN context, O_{pj} is the output of a neuron j given an input pattern p and net_{pj} is the total input to the ANN. The derivative of the output function to the total input is required to update the weights in the back-propagation algorithm. Thus we have: $\frac{\partial O_{pj}}{\partial net_{pj}} = O_{pj}(1-O_{pj})$, a trivial derivation. For a more detailed discussion on the back-propagation algorithm, see Chapter 3.

[18] A sigmoid function like the logistic function is most common transfer function in ANNs. Transfer functions are discussed in more detail in Chapter 3.

'train' the ANN by adjusting its weights to minimize the difference between the current ANN output and the desired output.

Finally, an evaluation process has to be conducted to determine if the ANN has 'learned' to solve the task at hand. This evaluation process may involve periodically halting the training process and testing its performance until an acceptable result is obtained. When an acceptable result is obtained, the ANN is then deemed to have been trained and ready to be used.

As there are no fixed rules in determining the ANN structure or its parameter values, a large number of ANNs may have to be constructed with different structures and parameters before determining an acceptable model. The trial and error process can be tedious and the experience of the ANN user in constructing the networks is invaluable in the search for a good model.

Determining when the training process needs to be halted is of vital importance in obtaining a good model. If an ANN is overtrained, a curve-fitting problem may occur whereby the ANN starts to fit itself to the training set instead of creating a generalized model. This typically results in poor predictions of the test and validation data set. On the other hand, if the ANN is not trained for long enough, it may settle at a local minimum, rather than the global minimum solution. This typically generates a sub-optimal model. By performing periodic testing of the ANN on the test set and recording both the results of the training and test data set results, the number of iterations that produce the best model can be obtained. All that is needed is to reset the ANN and train the network up to that number of iterations.

2.7 A Brief Description of the ANN Parameters

This section gives a brief introductory non-technical description of the ANN parameters. The mathematical descriptions of the parameters and learning process are discussed in more detail in chapter 3.

2.7.1 Learning Rate

The learning rate determines the amount of correction term that is applied to adjust the neuron weights during training. Small values of the learning rate increase learning time but tend to decrease the chance of overshooting the optimal solution. At the same time, they increase the likelihood of becoming stuck at local minima. Large values of the learning rate may train the network faster, but may result in no learning occurring at all. The adaptive learning rate varies according to the amount of error being generated. The larger the error, the smaller the values and vice-versa. Therefore, if the ANN is heading towards the optimal solution it will accelerate. Correspondingly, it will decelerate when it is heading away from the optimal solution.

2.7.2 Momentum

The momentum value determines how much of the previous corrective term should be remembered and carried on in the current training. The larger the momentum value, the more emphasis is placed on the current correction term and the less on previous terms. It serves as a smoothing process that 'brakes' the learning process from heading in an undesirable direction.

2.7.3 Input Noise

Random noise is used to perturb the error surface of the neural net to jolt it out of local minima. It also helps the ANN to generalize and avoid curve fitting.

2.7.4 Training and Testing Tolerances

The training tolerance is the amount of accuracy that the network is required to achieve during its learning stage on the training data set. The testing tolerance is the accuracy that will determine the predictive result of the ANN on the test data set.

2.8 Determining an Evaluation Criteria

It is not always easy to determine proper evaluation criteria in designing an ANN model to solve a particular problem. In designing an ANN to solve a particular problem, special attention needs to be taken in determining the evaluation criteria. This can be done by careful analysis of the problem at hand, the main objective of the whole process and the ANN role in the process.

For example, in designing an ANN to perform the task of designing a trading system for the foreign exchange market, there are many ways to evaluate the ANN model. The most obvious is to determine the forecast accuracy in terms of forecast error. However, in this particular problem, the accuracy of the forecast is not as important as the ability of the ANN model to generate profit in the trading system. Thus the evaluation criteria in this case is the profit made in trading the out of sample data period.

In the task of designing an early warning predictor of credit unions in distress in chapter 4, the evaluation criteria is based on the number of Type I errors committed, i.e., the number of credit unions actually in distress that were predicted to be not in distress. The ANN forecast was a number between zero and one with zero indicating no distress and 1 being in distress. However, in this case, in developing the evaluation criteria, an acceptable cut-off value has to be determined in differentiating distress from non-distress. The obvious choice is to use 0.5 but on further analysis, a value of 0.1 is determined to be a better value for this task.

2.9 References

1. Blum, A. L. and Rivers, R.L., "Training a 3-node Neural Network is NP-complete", *Neural Networks 5*, pp. 117-127, 1992.
2. Bryson, A. E., Ho, Y, -C., *Applied Optimal Control*, Blaisdell, 1969.
3. Cowan, J. D. and Sharp, D. H., "Neural Nets and Artificial Intelligence", *Daedalus*, 117(1), pp. 85-121, 1988.
4. Fischler and Firschein, *Intelligence: The Eye, the Brain, and the Computer*, Reading, MA, Addison-Wesley, p. 23, April 1987.
5. Hebb, D. O., *The Organization of Behavior*, John Wiley, New York, 1949.
6. James, W., *Psychology (Briefer Course)*, Holt, New York, 1890.
7. Judd, J. S., Neural Network Design and Complexity of Learning, MIT Press, USA, 1990.
8. Lindsey, C. S. and. Lindblad, T., "Review of Hardware Neural Networks: A User's Perspective", *Proceedings of ELBA94.*, 1994.
9. Lindsey, C. S. and. Lindblad, T., "Survey of Neural Network Hardware", *Proceedings of SPIE95*, 1995.
10. Lindsey, C. S., Denby, B. and Lindblad, T., June 11, 1996, *Neural Network Hardware*, [Online], Artificial Neural Networks in High Energy Physics, Available: http://www1.cern.ch/NeuralNets/nnwInHepHard.html, [1996, August 30].
11. Masters, T., *Practical Neural Network Recipes in C++*, Academic Press Inc., San Diego, CA., USA, ISBN: 0-12-479040-2, p.6, 1993.
12. McCartor, H., "Back Propagation Implementation on the Adaptive Solutions CNAPS Neurocomputer", Advances in Neural Information Processing Systems 3, USA, 1991.
13. McCulloch, W. S. and Pitts, W., "A Logical Calculus of Ideas Immanent in Nervous Activity:, *Bulletin of Mathematical Biophysics*, pp. 5:115-33, 1943.
14. Minsky, M. and Papert, S. A., *Perceptrons*, MIT Press, Cambridge, MA, USA,1969.
15. Minsky, M. and Papert, S. A., *Perceptrons. Expanded Edition*, MIT Press, Cambridge, MA, USA, ISBN: 0-262-63111-3, 1988.
16. Nelson, M. M. and Illingworth, W. T., *A Practical Guide to Neural Nets*, Addison-Wesley Publishing Company, Inc., ISBN: 0-201-52376-0/0-201-56309-6, USA, 1991.
17. *Neural Computing: NeuralWorks Professional II/Plus and NeuralWorks Explorer*, NeuralWare Inc. Technical Publishing group, Pittsburgh, PA, USA, 1991.
18. Ripley, B. D., "Statistical Aspects of Neural Networks", Networks and Chaos: Statistical and Probabilistic Aspects edited by Barndoff-Nielsen, O. E., Jensen, J.L. and Kendall, W.S., Chapman and Hall, London, United Kingdom, 1993.
19. Rosenblatt, F., "The perceptron: a probabilistic model for information storage and organization in the brain", *Psychological Review*, 65:pp.386-408, 1958.
20. Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Back-Propagating Errors", *Nature*, No. 323: pp.533-536, 1986a.

21. Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Error Propagation", *Parallel Distributed Processing: Explorations in the microstructure of Cognition edited by Rumelhart, McClelland and the PDP Research Groups Vol.1*, pp. 216-271, MIT Press, Cambridge Mass., USA, ISBN: 0-262-18120-7, 1986b.

22. Sejnowski, T. J. and Rosenburg, C. R., "Parallel Networks that Learn to Pronounce English Text, *Complex Systems*, No. 1, pp. 145-168, 1987.

23. Shih, Y., *Neuralyst™ User's Guide*, Cheshire Engineering Corporation, USA, p. 21, 1994.

24. Werbos, P., *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph.D. thesis, Harvard University*, 1974.

25. Widrow, B. and Hoff, M. D., "Adaptive Switching Circuits", *1960 IRE WESCON Convention Record*, Part 4, pp. 96-104, 1960.