An Introduction to Neural Computing

What Is Neural Computing?

Did you ever wonder how we as humans could perform tasks like recognizing a friend's face in a crowd of people, even if part of their face is hidden behind a tree?

Computers are typically not very good at tasks at which humans excel, such as image and speech recognition, reasoning, understanding, and acting in the face of uncertainty. This difference cannot be due directly to a lack of speed, since a computer actually manipulates data thousands of times faster than neurons in the brain. However, computer processors have a structure that is very different from the human brain, and it is natural to wonder whether human superiority in these areas might be related to the difference. Could a computer whose internal operations mimic those of the human brain really "think"? And if it could, would it be as versatile as the brain?

Neural computing is a concept in advanced computing technology that attempts to emulate the basic structure and functionality of the human brain. This is obviously a daunting task, but research on designing a computer based on the neurons and synapses within the human brain dates back to the early days of computing in the 1950s.

The desire to develop a computer that behaves more like a human brain was largely driven by the desire to make a computer that *acts* more human. The hope for the neural computer was that it might show human-like intelligence, at least at a rudimentary level.

We as humans have the unique ability to *learn*. This ability allows us to adapt and survive in an ever-changing environment. In some sense, learning is what life is all about. Learning gives us the ability to amass knowledge, and apply this knowledge to new situations. We call this reasoning, drawing conclusions or inferences from observations, facts, or hypotheses.

Conventional computing requires someone to work out a step-by-step solution to a problem, then program these steps into the computer. By contrast, a neural computer can be *trained* to solve a problem. By assuming the responsibility for *learning* how solutions are reached, a neural computer effectively programs itself.

Central to this technology is a software system known as a *neural network*, which is a simulation of a large number of simple processing units connected together into a complex network-like structure. Such a network can take the form of actual electronic hardware or be simulated in computer software.

Training a neural network involves presenting it with data that contains a series of examples of a problem and the desired solution(s) for each case. Given sufficient training material, the neural network is able to learn the underlying principles involved in the solution, which it can then use to tackle similar problems. In essence, we are showing the neural network examples of cause and effect, and expecting it to learn these relationships. We call this type of learning *Supervised Learning*.

As humans, we learn in a similar way. All of the complex tasks that we do have been learned through our experiences. For example, a Stock Market Analyst learns through experience the subtle indications of when to buy or sell a stock. He may be influenced by a large number of factors, some of which he may not even be aware of, but his decision is based on examples of similar stock movements that he has seen before. When we learn based on association with things we have seen before, we call this type of learning *Unsupervised Learning*. With Unsupervised Learning, we learn by association, or by drawing analogies to experiences we have seen before. We do not have an explicit cause and effect relationship as in Supervised Learning, but instead infer based on similar situations.

If we could treat a computer as a young child and train it to do what we wished by simply showing it examples of what we wanted it to do, we could overcome the computing bottleneck that keeps most of us from using our computers to their real potential.

Neural computing is a method for doing just that. Because it learns by example, from lessons that you provide, it allows a software solution that is adaptive, self-improving, flexible, and capable of performance approaching that of an experienced expert.

The Computing Bottleneck Today

The factor that limits the use of computers in most areas today is no longer speed or program size, or anything related to the computer hardware. The PC on your desk has more raw power than the large mainframe computers of a decade ago. The central problem now has become the difficulty of developing software that will get the computer to do what we want it to do.

Programs are complex, and are expensive to produce and maintain, even in those areas where the algorithm - the recipe of steps that becomes the program when coded in a computer language - is known.

Yet for many problems there is now an even more fundamental problem: we do not know what to tell the computer to do. If we don't know (or can't describe) how we recognize a friend's face in a crowd, or exactly how we make a business decision, then we cannot write down the sequence of steps necessary to tell a computer how to do it. Interestingly however, our own brains manage to do these things quite easily, and without being programmed explicitly.

As mentioned previously, there are two types of learning used in neural computing: supervised learning and unsupervised learning. In supervised learning, each training example has the correct answer associated with it. Using a loan application as an example, details from the application form describing the applicant are accompanied by knowledge of whether or not the loan was granted, and how the resulting loan worked out. You can think of supervised learning like a lesson we learn in school: for each problem there is an associated set of answers.

In unsupervised learning there is no teacher. The task here is to learn by association, or to group together patterns that are similar in some way; and to find the common threads in a morass of data. As you will appreciate from your own experience, unsupervised learning may be more difficult, in that a clear-cut cause and effect relationship has not been stated.

Yet, unsupervised learning is actually how we as humans do the majority of our learning. When you were very young, your Mother may have said to you, "If you touch the stove, you will burn your hand". This is an example of supervised leaning, because you were given both sides of the cause and effect equation. However, certainly your Mother never told you everything that you could touch that would result in you burning your hand. Instead, you learned to associate things that looked like flames or felt hot as you got close with the likelihood of burning your hand if you touched it.

Imagine the complexity of attempting to program a computer to warn you every time you were about to burn your hand. It would be hard to even list all the situations that could arise that might result in a burned hand. Yet, as humans, the task of keeping our hands away from danger or an accidental burn seems trivial.

How will Neural Computing Affect Me?

Neural computing is a technology that enables exciting new business solutions for a company, and gives practical commercial benefits.

Neural computing can:

- Provide an easy way to analyze large and complex amounts of data, and actually extract information that results in new knowledge or insight (Data in – Rules out)
- Learn nonlinear and/or discontinuous functions, which would be very difficult to express within an equation (Data in – Model out)
- Learn how a dynamic system changes over time, and predict where the system is likely to go next
- Add adaptive modeling functionality to an existing system at a very low cost
- Capture expertise in a particular domain, and make that knowledge available to others
- Provide autonomous control for well-defined processes and decision-making situations
- Dramatically reduce the time and expense required to find the optimal solution to a problem which seemingly may have many solutions

The nature of the solutions provided by neural computing is such that it is a complementary technology, and can be used alongside, and with, techniques that you may already be using.

The Benefits of Neural Computing

Neural computing offers a new type of solution to forecasting and data analysis problems, and provides better results than traditional techniques, especially when the problem is complex and there is unknown interaction among elements of data.

Neural computing solutions recognize patterns in your data, giving them the ability to deal with real world problems directly and with minimum guidance or programming. While people can find patterns easily, it is difficult to make computers do so. For example, recognizing faces is something we do every day but which computers find very difficult to do. The same is true of recognizing trends and patterns in data.

Neural computing solutions are flexible, and easy to maintain. Since cause and effect relationships can change over time, a neural net can adapt to these changes through re-training, rather than by the expensive and time-consuming method of having to re-write software.

Solutions can be implemented rapidly since the neural network does not need to be explicitly programmed. This can result in shorter design cycles, and makes the technology cheaper to implement than some other techniques.

Neural networks are robust, both in their ability to handle noisy or corrupt data, and in the gentle degradation they exhibit if individual "neurons" within the neural net fails. This means that in the event of a problem the results produced by a neural network will get worse gradually, rather than catastrophically, giving time for the error to be corrected. However, this fault tolerant ability to degrade slowly may only be important in an actual hardware implementation of a neural net.

The Applications for Neural Nets

There are many diverse applications for neural computing technology. In general, this technology can be applied to any problem where there is a relationship to be discovered and analyzed between cause and effect. However, as one might expect, some applications are better than others are. Neural Nets are most valuable when applied to a problem in which:

- A clearly defined problem has been identified, and it is understood how the solution will be used
- A clear business case can be made about why it is advantageous to discover cause and effect relationships within data
- Historical data is readily available, providing numerous training examples
- Individual fields of information within the data may be incomplete or unreliable, so that a deterministic program would find it difficult to draw a conclusion or reach a solution
- The rules which define how causal situations trigger particular outcomes are incomplete, unknown, or not easily stated (and maintained)
- An explanation of how a decision is reached is not required

Since 1985, the AI WARE Division of Computer Associates successfully developed and marketed software solutions utilizing advanced computer technologies such as neural nets, fuzzy logic, and genetic algorithms to the following areas:

> **Formulated Product Design and Optimization Within The Chemical Process Industry** – design of chemically formulated products such as industrial and household chemicals, paints and coatings, rubber, plastics, adhesives, and food products requires an expert knowledge of the relationships between the recipe and processing of the ingredients and the resulting product properties achieved in the final product. The product designer must understand the interaction of all the ingredients and process parameters, and knows that finding the optimal recipe to achieve all of the desired product properties is a balancing act requiring much trial-and-error research.
>
> Neural nets have proved extremely successful at learning the relationship between formulation variables and the resulting product properties and cost.

Within the past seven years, we have worked with most of the major companies within the CPI to cut product development time (time to market), improve product designs and reduce overall R&D expense.

By presenting the neural net with data that represents past formulation tests, a model is built that determines exactly how each of the independent formulation variables affects the properties of the product being produced. Once this model is built and validated, it may then be consulted to:

- answer "What-If" questions about how changes to the formulation affect resulting properties and cost
- determine the optimal combination of ingredients and processing parameters that must be used to achieve a given set of new or improved product properties.

The total list of companies we worked with in this area is quite extensive, including:

E. I. DuPont	Avery Dennison
Miles Laboratories	Goodyear Tire & Rubber
Mitsubishi Chemical	LTV Steel
Procter & Gamble	Boise Cascade
Unilever	Armstrong
Kraft General Foods	BP Chemical
Borden Chemical	Gillette
Xerox	Glidden Paints
BFGoodrich	Air Products and Chemicals
Dow Chemical	BASF
Monsanto	Dow Corning
DuPont – Dow Elastomers	NASA
Allied Signal	Nike
Warner Lambert	- - - and hundreds of others

Database Marketing - where products and services have been targeted to specific markets by analysis of a customer database. The assumption behind Targeted Marketing is that if a profile can be developed of buying patterns of certain types of individuals, then the neural net can predict the likelihood of similar individuals to respond to a given marketing solicitation.

Process Control – Process monitoring and control with neural networks has been proven successful for many different industries and applications. Real time, closed-loop control is possible with neural nets because the model may be consulted very quickly. Process models can be learned automatically by simply observing the controllable, uncontrollable process parameters, and the resulting properties characterizing the efficiency of the process. The complex linear or nonlinear forces driving changes to the process can be discovered and controlled, even as equipment ages, retrofits are done, and when raw material and operating conditions change. Because the neural net learns from experience, the model adapts as the state of the system changes.

Business and Financial Applications – in the areas of Finance, Sales Analysis, Marketing, Strategic Planning, and Operations, where large amounts of data are generated and stored, but little knowledge may actually be known about the driving cause-and-effect relationships within the data.

Neural Nets have proven effective in financial applications such as:

Finance
Risk Management
Budgeting
Financial Forecasting
Investment Optimization
Fraud Detection
ROI Analysis
Continuous Auditing

Strategic Planning
Market Price Sensitivity
Site Selection
Revenue Optimization
Inventory Levels
Strategic Assessment

Marketing
E-Commerce
Target Market Identification
Survey Response Analysis
Demographic Analysis
Product Demand Forecasting
Customer Satisfaction Studies

Sales
Sales Forecasting
Lost Sales Analysis
Transaction Forecasting

Production
Quality Control
Yield Forecasting/Optimization
Staffing Levels
Predictive Maintenance
Resource Planning

While the movements of stock markets may be challenging to predict because the causal factors may be difficult to identify and capture, the neural network provides a tool to help analyze and make sense of seemingly unrelated factors. The most important factors will be identified, and business objectives can be maximized through an improved understanding of the complex forces driving business decisions. Forecasting of future events allows the user to take steps now to optimize a strategy to meet future business objectives.

Data Visualization - Neural computing techniques have been used to visualize the correlation's between data, and has allowed new trends to be recognized and exploited.

Database Enhancement - Many companies have databases that are either incomplete or have incorrect entries. Neural computing has been used to fill in missing data, and to amend errors. Neural Nets are unique in that they can still be used to develop effective models from data that is incomplete, sparse, or noisy.

Customer Satisfaction Analysis – Even when data is more subjective than qualitative, neural nets provide better predictive capabilities than other types of modeling. The cosmetics industry will often conduct a panel review of impressions about a cosmetic product such as suntan lotion. While lab test results measure properties such as SPF rating or viscosity, these panel reviews try to evaluate properties such as feel, smell, and overall satisfaction. These subjective characteristics are difficult to measure and analyze, but neural nets are used not only to make sense of this data, but also to determine how to modify the product to improve on the results.

When Not To Use Neural Computing

Neural Computing solutions are not suitable for solving all modeling problems. There may be cases where data is just not available for training. Or data may be available but a cause-and-effect relationship can not be found.

Remember, the good news with using neural networks is that "**all you need is data",** while the bad news may also be that "**all you need is data**".

Some people are concerned by the 'black box' nature of neural solutions, in that it may be difficult to understand why certain results are given by a neural network. You must have confidence that the data you have trained your neural network on is representative of the problem. This fear can typically be avoided by validating the neural net model with data that was withheld from the training process, but used only to test the accuracy of the model.

There is also the more general problem of understanding enough about the problem you are trying to solve, so that you believe you can collect data that in fact does contain a cause-and-effect relationship to be discovered. It is possible to bring data to a neural net that can not be trained, because there is no relationships to be found. A neural net can not predict the next roll of the dice, or any other truly random event.

How Does Neural Computing Work? (Lets Get More Technical)

As already said, the technology behind neural networks as we employ them today was originally developed by researchers to mathematically emulate the behavior of neurons within living organisms. While this approach proved to be too simplistic for biological researchers, neural networks were later adopted by a much wider community when they were shown to have powerful computational capabilities in function approximation and pattern recognition. The biological origins of the technology account for continued use of terms like "training" and "learning" to describe how neural networks operate, although they refer to mathematical procedures.

Internally, a (supervised learning) neural network is an interconnected lattice of rather simple, nonlinear *processing elements*, or nodes. From the outside, a neural network appears as a function that maps an input vector space to an output vector space. The ability to represent a variety of functions or systems results from this complex architecture built of very basic components.

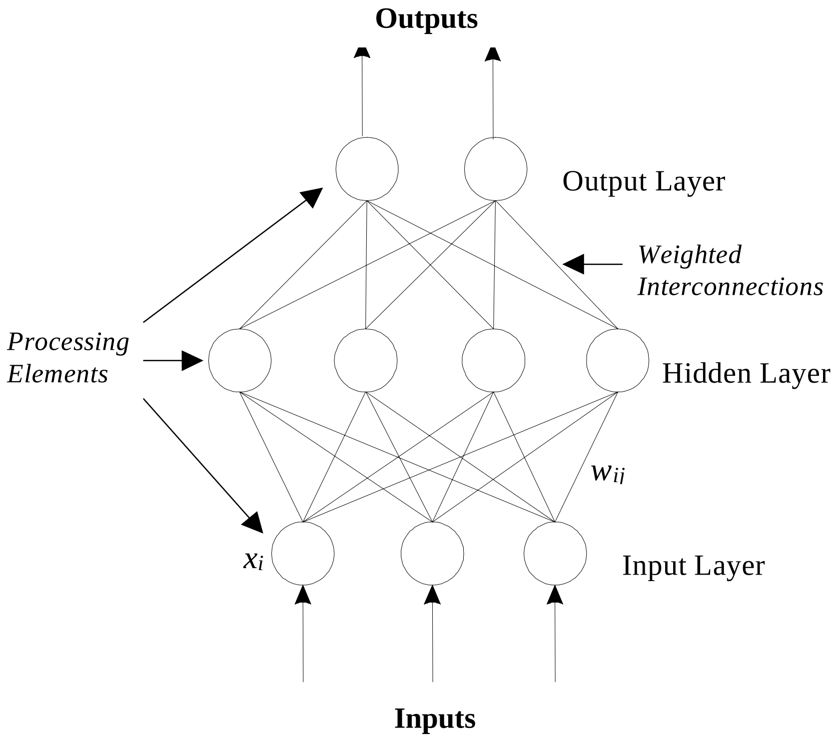

Figure 1 A classic neural net architecture representation. The Input layer contains the independent variable values. This layer is connected to the Hidden Layer, which can have a variable number of *processing elements*. All processing elements in the Hidden Layer are connected to the elements in the Output layer, which contain the dependent variables for the given modeling problem. The *interconnections* that connect all the processing elements in this structure are weighted, and it is the weights that are adjusted during the training process.

Given a set of input/output data for the network to reproduce, a training or learning algorithm is used to configure the internal connections of the network. This algorithm is an optimization procedure in its own right which seeks to minimize the error (difference in values) between the actual data and the predictions of the neural network, by modifying the internal connections within the net. This process is known as *training* the network, and the algorithm is commonly called a *learning* algorithm.

Because of the simplicity of the processing elements, the time it takes to execute a trained model prediction, or ‘consult’ the model, is typically very fast (measured in milliseconds). This makes neural network models suitable for complex problem solving, such as sensitivity analysis and real-time process control.

To take a closer look at what goes on inside a neural net, we will consider each of the components that make up the structure of the neural net.

A neural network is composed of processing elements that can be organized in different ways. Much research has gone into determining how many processing elements should be used in hidden layer(s), and how the interconnections should be configured.

PROCESSING ELEMENTS – A neural net is composed of artificial neurons that we call processing elements (PEs). Each of these processing elements receives input(s), processes the input(s), and delivers a single output. This process is shown in Figure 2. The input can be raw data or output from of other processing elements. The output can be the final output number to the output layer, or it can be an input to another processing element.

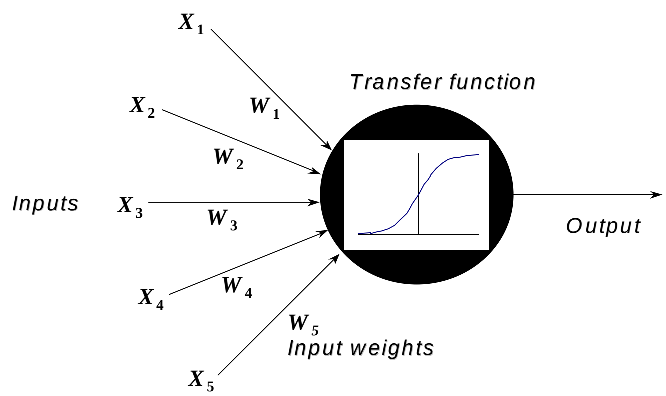

Figure 2 A typical Processing Element, which includes both a summation function and a transformation (transfer) function.

The **summation function** multiplies the input values (*X*s) by the weights (*W*s) and totals them together for a weighted sum. It then computes the output value for the processing element, by filtering the weighted sum through a nonlinear transfer function. Based on this level, the neuron may or may not produce an output. The transfer function is usually nonlinear (sigmoidal), but may be configured to be linear. Note that it is the nonlinear transfer function which gives the neural net its ability to model nonlinear relationships.

You can find many commercial software packages that can build and train a neural network model from user-supplied data. Researchers have reported on the successful development of predictive, neural network models for many diverse processes, from the stock market to pollution control. However, when the number of inputs to the model and the number of records available for training becomes large, the training procedure for conventional neural network architectures becomes increasingly more time-consuming. The task of developing the best neural net model often requires some trial-and-error using different neural network architectures, and a slow training process can make this model comparison unduly tedious.

Computer Associates owns a patent on a unique new neural network architecture, the Functional-Link Net™ (FLN). FLN is a simplified neural network architecture (see Figure 3) that has been shown as computationally capable as more complex structures, yet trains in far less time. The Functional-link Net enables more rapid development of successful neural network models in cases where large numbers of inputs, or large amounts of data, must be processed.

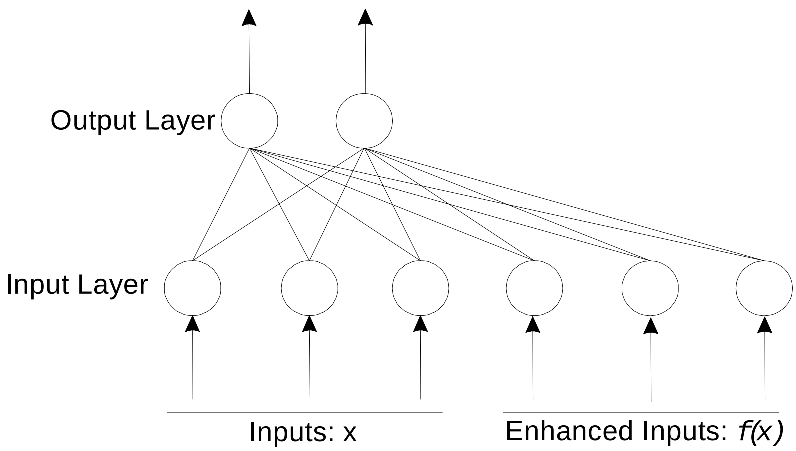

Figure 3 An example of Computer Associate's Functional Link Net architecture. Notice the omission of the hidden layer, and the addition of extra nodes on the Input layer, which are functional transformations of the original inputs. Although not shown, weights are associated with each interconnection, as in the classic Hidden Layer architecture.

A Comparison with Other Methods

Neural computing is an approach to solving problems that avoids explicitly programming solutions. However, the techniques that they use are not completely new, and are closely related to two other major areas of computing expert systems and statistical pattern recognition methods.

Expert systems are programs that mimic the reasoning processes used by humans. To produce an expert system, you must find out what an expert does in a given situation and codify their knowledge as a set of rules.

Statistical pattern recognition methods use linear or multiple linear mathematical techniques to uncover patterns in data. To get good results from them you must have a good idea of the underlying pattern that exists. The statistician must make assumptions about these relationships before a model is generated.

Neural network analysis shares many characteristics with statistical analysis, but with somewhat different focus. Both techniques aim at discovering relationships between inputs and outputs which lie hidden in the data, with the goal of being able to make reasonably accurate predictions about system behavior using the model.

Statistical analysis can rely heavily on the structure or distribution of the available data set, while neural networks can derive meaningful information even from unplanned historical data. This also allows neural network models to be developed from interim or aborted designed experiments. These models can be easily updated or “retrained” with new experimental data as it becomes available, continually improving the predictive capabilities of the model. In effect, neural net models can become ‘smarter’ with age.

Perhaps the most common criticism of neural networks is that they are a "black box", from which little global information on the underlying system model can be extracted. However, this criticism is based on assumptions regarding *a priori* knowledge of the model dynamics, that certain interrelations between inputs exist, and that they are uniform throughout the input space. Neural network analysis is essentially non-parametric regression, which requires no *a priori* assumptions or constraints on the internal structure of the model. These assumptions are integral to a parametric regression model (such as linear or quadratic regression). They may tend to confine the analyst to a type of model that is not capable of fitting the non-linear dynamics that govern a particular, real-world system.

If you have little *a priori* knowledge about your process to begin with, a “black box” approach, with thoroughly validated models, has distinct advantages. It provides the means for immediate improvement, without additional investment in time and research to determine underlying process dynamics. A few, sophisticated, neural net modeling approaches and optimization techniques can incorporate “expert” knowledge about the process, combining the best of both worlds.

Neural networks, with their ability to generalize and to learn by example, make good pattern recognition systems. Generalization, achieved by selecting combinations of internal features to represent the problem, gives them some advantages over the more common statistical techniques; while the ability to learn and adapt to new data gives them an edge over expert systems for many tasks.

Glossary of Terms

Artificial Intelligence
Artificial Intelligence, or "AI", is an attempt to reproduce intelligent reasoning in computers. It is inspired by the desire to make computers do things which humans typically do better.

Expert Systems
Software uses a rule-based approach of capturing expertise on a specific well-defined area. Very similar to IF-THEN-ELSE type logic. By using rules to combine knowledge with information gathered from an expert, they draw conclusions, provide advice, and help to choose between alternatives.

Genetic Algorithms
The genetic algorithm is an optimization technique based on evolutionary principles. Genetic algorithms offer an intelligent way of searching a large multi-dimensional space for the optimum solution to a problem. The GA is a stochastic algorithm that works with a 'population' of individuals, each of which is a candidate solution to a problem; these individuals 'mate' with each other, 'mutate', and 'reproduce' and in this way evolve through successive generations towards an optimum solution.

Heuristics
A heuristic is a rule of thumb, which gives guidance about how to solve a particular problem. Generally, there is a probability associated with heuristics. Heuristics are mainly used when data is imprecise or a level of uncertainty exists.

Knowledge Discovery
A multi-stage business process leading to the automated detection of regularities in data, which will be useful in decision making in new situations.

Modeling
Modeling in its broadest sense is a mathematical approximation or generalization of something. In the case of neural net modeling, we adaptively adjust a matrix of coefficients to minimize an error function, until we have sufficiently represented a mapping of an input vector to an output vector.
The model is used to represent reality for the given purpose. It is an abstraction of reality within a well-defined area.

Model Validation
Once a model has been constructed, it is imperative to check to make sure it represents the actual relationship the model was meant to capture. This procedure checks the generalizations assumed by the model, to ensure it represents the real system accurately and completely within the application domain and for the purpose intended. This process is called model validation.

Neural Network
A computational procedure which emulates the biological activity and processing capabilities of the human brain. Often described as "learning systems", a neural network, or neural net, captures relationships in data.
Neural Networks model data, by performing a nonlinear transformation from an *m*-variable input vector to an *n*-variable output vector.

The biological inspiration of neural networks shows up quite clearly in the terminology used to describe their behavior. Neural nets don't execute programs as much as they "behave", given a specific input. They are "trained", "react", "self-organize", "learn", and "forget".

Optimization
The process of seeking the best solution to a given problem, where many solutions must be considered.

Prediction
To state, tell about, or make known in advance, especially on the basis of special knowledge. Prediction as used in this paper is querying the model to ask questions of the form "What if... or, prediction can be used to forecast a future event.

Training
The task of acquiring and processing the body of numeric data to a neural network, to develop a model.