An Overview of Neural Networks

Brad Morantz

Introduction

What is a neural network? Where can they be found? What does it do and how do you use it? Is it really intelligent? What is intelligence anyways? How does this method compare with other AI Methods, i.e. expert systems?

Biological Neural Networks

The brain of a monkey, a snail, porpoise, or a homo sapien is a neural network. In fact, brains in most living creatures are neural networks. They have been researched extensively, and while an exact understanding is not current knowledge, much has been discovered. They are composed of units called neurons, and they are interconnected. Hence the name neural network. And the larger the animal, usually the larger the brain.

The bat (the small flying rodent) is a good example. Its brain is the size of a plum. And it not only operates the involuntary processes of its host, but also normal functioning, flying, and controls navigation of the creature using its sonar as its guide. This tiny little neural network computer outperforms our greatest radar systems and super computers. This animal is capable of flying through an electric fan without being injured. It does not need a large air-conditioned room, 440V power, or millions of dollars. This exceptional performance allows it to find its prey and capture it. The United States Air Force and its defense contractors would be ecstatic to attain the performance of this tiny neural computer.

From years of study, scientists and researchers have determined that the brain is made of millions of interconnected neurons. Each neuron has an input area (dendrite), a body (soma) and an output (axon). Connections are made by the axon of one neuron laying on the dendrite of another. A unidirectional electrochemical junction is formed, called a synapse. The intensity of the connection is called the weight. Knowledge in the brain is stored in the set of weights. We do not yet understand how the brain sets these weights.

There are from 4×10^{10} to 10^{11} neurons in the average brain of a human being. There can be as many as 10^4 interconnections per neuron, allowing a maximum of 10^{15} interconnections. Some recent publications now suggest the number of interconnections per neuron to approach 10^5. This is a tremendous amount of interconnectivity.

These junctions have an operating speed of one kilohertz, over a million times slower than a typical personal computer. If it is so much slower, then how do these biological neural networks grossly outperform our greatest supercomputers? The massively parallel structure is how it can accomplish these tasks.

What can a Computer Artificial Neural Network do?

There are several things that a neural network computer emulation program can do. It can perform a modeling task similar to regression, except that it is not constrained to fitting a straight line through the data points. This function can also be used for time series forecasting. The artificial neural network (ANN) can also work with multiple dependent or criterion variables, which is comparable to the general linear model that is implemented by canonical correlation.

Classification is a common use of an ANN. Common problems are credit-scoring people to see if they qualify for credit cards or bank loans. Credit inputs can be things like years on the job, debt ratio, payment history, etc. Fuzzy logic can be used to help convey information to the ANN. The military has used these to identify snipers hidden in the environment or submarines under water. Quite often these systems are used in conjunction with optical sensors, visible and/or infrared. They have been able to discriminate objects that were not visible to the human eye.

Pattern recognition is an important application area. While much like classification, rather than grouping the results into classes (e.g.. poor credit risks, good credit risks), this is a more exacting process. When we look at photographs of historical figure, we don't just say that there is "some old man with long hair", but rather we say that this is a photograph of George Washington, or Ben Franklin, or for that matter, you might recognize your friend Mary. The answer is very specific, rather than a classification.

Iris and retina scans for identification use this to identify people accurately for various purposes. A neural network works well here, as we use our brains to recognize something from an image and possibly other inputs. Ever notice how you see someone out in public, you know that you know them, but cannot remember exactly who it is as they are out of context. You do not recall exactly who it is, until you walk into the style shop for your weekly trim.

Lastly, neural networks can be self organizing, where they group things into clusters based on the values in each of the dimensions or variables. This is very similar to cluster analysis, and as in the other comparisons, neural networks are not constrained to linear relationships. This process can be used for classification, as similar items tend to be close together on the same dimensions.

Another important use of ANN emulations is brain research. Medical and psychological researchers model the brain to learn more about how it works and how they can control or cure various ailments and conditions. Since the brain controls the entire body and all functions within it, holistic medical researchers and practitioners also have a strong interest in modeling the brain.

The power of a neural network

Neural networks do the same things as established and accepted statistical programs. In some instances, where the underlying theory is well understood and the causal factors are known, statistics might be the better choice. The power of neural networks is that they learn input to output relationships, and they are not constrained to being linear. They are also not affected by multicolinearity, and are more immune to noise than their statistical counterparts.

This is advantageous, as the world has been shown to be nonlinear and correlated. Multicolinearity is when one thing happens; one or more others are affected. For example, if one is working on credit scoring, if inputs were income level, years of education, among others, one would find that more educated people tend to have a larger income (except in the case of engineers in the last few years). This connection or correlation between the predictor variables, called multicolinearity, would cause a statistical method to be less accurate.

Linearity is a relationship that changes linearly with each other. If a roll of wire costs X dollars, then 2 rolls should cost 2X dollars. But if one is a good bargainer, and is in need of several rolls of wire, one might haggle with the seller for a better deal, and then the linear relationship would be gone. This is the way the real world works.

Neural networks learn the input to output relationship without building any underlying logic or reasoning. They do not need to know how good of a bargainer you are, or how desperate the seller is or is not to sell his wares. They just learn from what has happened and expect to see the same thing happen again.

The exact situation did not need to have happened in the past. An ANN can interpolate between various learning experiences and predict or interpolate what it expects to happen for a specific input vector or tuple. There might have been conflicts in the historical data, where two or more observations had the same data, but different outputs. From the entire picture it has created of the relationship, it knows which is the correct answer.

Compared to some other methods, ANN's are easier to get started. Experts and knowledge engineer are not needed as they do not require underlying theory or models. It learns strictly from the data. Therefore, it does not contain any bias that a human expert might have. An expert chef might not like strawberry flavor, and this bias would be hidden throughout his knowledge and affect the rules or information that he would submit.

To sum it up, the neural network is a data driven general function approximator. It discovers relationships based purely upon the historical data used for training. It then mimics this relationship. It does not replicate the relationship, as might occur in regression when supplied with sufficient theory and expert support. It just creates an output (or very close) that one would get from the real world system. It generalizes based upon the given training data to interpolate for an unknown input. Is it intelligent? No, it just appears to be intelligent, hence the expression artificial intelligence.

How do we make a computer emulation of an Artificial Neural Network?

There are programs available, or one can program their own artificial neural network. But one must realize that they are trying to reproduce a massively parallel system on a serial "Von Neumann" machine. Some of the newer CPU's have 2 or 3 math processors on board, which is negligible compared to the number of neurons in the brain. A few more powerful machines, usually used as servers, may have 2 or 4 CPU's, a step in the right direction, but still insignificant. Some of the big Supercomputers (SGI, IBM, etc) may have 24 or 32 CPU's, which is still insignificant when compared to a biological neural network. IBM Research Center in Yorktown Heights NY had started on a multiprocessing machine with 1 Meg (1,048,576) processors, which if completed, would be an interesting machine upon which to build am ANN.

The number of neurons in current emulations is very small. A typical program may have 30 to 50 neurons, which is infinitely small when compared to a biological brain. One type of sea snail, commonly used for research because its neurons are physically large and are easier to observe has about 100 neurons, about double to triple our computer program.

The neurons or nodes must also be modeled. At present, we only model them as mathematical functions, as we have not learned enough about the synaptic chemistry.
There are three types of activation functions:

1. Linear activation – this activation is typically used for the input nodes, and output nodes if the system will be used for non-discrete or non-categorical output. In other words, if the ANN will be used to generate a real number, a predicted value that is not an integer, then linear activation must be used for the output nodes. An example of this might be the price of a certain stock or the cost to produce a widget. Classification or a type of categorical response would work best with one of the logistic functions (following). Problems requiring integer or whole number answers fall into this class (i.e. the number of cashiers at a grocery store is an integer as we cannot hire a half-person).
2. Sigmoidal activation is the most common type of logistic (can be in only one of two states) response. The output is either a 1 or zero, corresponding to a "fire" or not. This is generated using a mathematical sigmoid function. This has been called a squashing or clamping function, as it compresses all of the inputs into a zero or one output.

   Sigmoidal function: $1.0/(1.0+e^{-S})$ where $s = \Sigma$ inputs
   0 or +1 result
3. Hyperbolic tangent is also a logistic function, but in using this trigonometric function, the output is a -1 or +1. Some researchers claim that this outperforms the sigmoidal function,

but this has not been substantiated sufficiently. Additionally, it does not correlate as well to the biological neural network. Hence, it is not as common as the other two types.

Hyperbolic tangent function: $(e^s - e^{-s}) / (e^s + e^{-s})$ where $s = \Sigma$ inputs
-1 or +1 result

Architecture

Setting the parameters of the ANN requires the greatest amount of operator input of all of the control interface demands. This human in the loop (HITL) process requires selecting the number of input, hidden, and output nodes, and organizing the training process. Proper architecture can make the difference between accurate and worthless results. This operator dependence is one of the reasons why neural networks are not highly accepted. Research is being done in autonomous learning and machine cognition to alleviate this problem. Hopefully in a few years this problem will be solved. There has been some research into using genetic algorithms for designing the optimal (or near optimal) architecture.

Input layer or number of input nodes: this is the number of input variables or number of lagged values for a time series. Inputs can be numerical values (such as years on the job, debt ratio, temperature, etc.), categorical values (i.e. hair color, make of car, breed of dog, etc), or fuzzy (hot, fast, old, etc.).

Hidden layer or number of hidden nodes: This is optional; some ANN's do not need these additional neurons. Other models need multiple hidden layers to better fit the data. Research has shown that one hidden layer is sufficient, with possibly a greater number of nodes than in multiple layers. This layer adds complexity that allows it to fit nonlinear relationships. Today there is no exact science for setting the number of hidden nodes. It is currently a project under investigation. A few heuristics have been suggested, but have not been proven.

Output layer or number of output nodes: This number equals the number of outputs. In most instances, there is one output, the predicted value. In some cases there can be multiple output nodes, for classification or pattern recognition, multiple horizon time series, or multi objective problems. Time series is predicting future values, as in a stock price or CD rate using only past or historical values as input. Multi horizon is for next step and the one after that, i.e. tomorrow's value and the day after that. Multi-objective is when there are two (or more) output values, i.e. length and width or sale price and profit margin.

There are two basic types of structure:

1. Feed forward

The input starts at one side and goes towards the output side. There is no feedback (where output from one layer goes back and is connected to an earlier layer). Process/information always moves in the direction of input to output. See drawing. Feed forward is more common because it is far easier to program and requires far less computation time. When multiprocessing computers (machines with a large number of CPU's) becomes more prevalent, then this might change.

2. Feedback neural network

Like the feed forward network, the flow starts at the input and moves towards the output. The difference is that sometimes there is some connection that feeds from a hidden or output node back to an earlier node. System control theory states that this will add performance or accuracy.

Training

Just as we humans have to go to school, so do ANN's, so to speak. They need to be trained, to get their knowledge; in this case it is showing them historical data. A method to use the training data to set the weights is required. This is a key component to the operation of the network because the more effective the training, the better the ANN will perform. This sounds just like humans.

Many questions arise. How big of a training set? How old is the data? Which window method? Some practitioners and researchers talk about over-training. Their argument is that if the system is trained too much, then it will fit the training data too well, and may not be able to generalize to fit the new data, the ones wishing an answer from the neural net.

There are various window methods that take a portion of the data and use that for the training. The window is the portion that is used for training. This window can move or change during the process, especially when doing time series. At first, one may think that all of the data should be used for training, but there are several reasons why this may not be the best idea. The most obvious reason is that there is just too much historical data, and it might take too long. Maybe it could be repetitive, and the extra data might not introduce any new knowledge. It also might be too old, and while it was accurate when it happened, things have changed and it no longer represents the truth. Consider styles and how they change, and how what was then is not now.

Another concern is anomaly detection. The historical data has to be somehow collected. It is possible that mistakes were made during collection, some abnormal situations were observed, errors were made while recording the data, or any one of a million other possibilities. The neural network is somewhat immune to noise, but it can go astray with too much incorrect data. Think about the times in history when children were educated with wrong information, and they grew up to make poor decisions. So the better the knowledge that we give our ANN, the better that it will fare.

There are many ways to deal with this problem. Some of them include things like knowing where one obtains the data set, being careful of disreputable sources. Looking over the data for obvious errors is another thing that can be done. Maybe one could hire an expert to look at the data. Random sampling of the data to verify the results can give an overall indication of the quality. Statistical analysis can look for outliers, values that are more than 3 standard deviations away from the mean. (This may not mean that the observation is erroneous, just that it might need to be checked. Some statisticians claim that the greatest information is contained in the outliers).

If the data set is large enough to justify the extra work, automated anomaly detection can combine many of the above things into an automated computer program that eliminates duplicates and either clearly erroneous or suspicious data. Suspicious data might also be marked for human examination. This process can contain expert rules, statistical process, and general data scrubbing methods. Results can be quite good, when compared to unprocessed data sets.

Multiple outputs may complicate training. Sometimes when the output is more than one value, as in the case of multi-objective problems, the true best answer is not known. When looking at the case of the output having two values, sales price and profit margin, what would be the best solution? How would you train the system? The output is really a tuple or vector. One could try and find a scalar (single value) that would indicate the optimal solution, but in many cases, this is not so easy. In the just mentioned sales problem, one expert might say that net profit is the object of the optimization, but another expert might disagree.

There are two main classes of training. The first, and most popular, is supervised training, where the training set has the input information and the output answer as well. This way the net can see the inputs and what each associated output should be. This allows the system to learn the input to output relationship, which is the heart of a neural network's power. This is the way we learned in school; here is a picture and this is what or who it is, here is a problem and this is the answer.

Unsupervised training is when the data does not include the answer and all that the training program can do is to try and organize the data along the variables, clustering them into similar groups.

Training methods:

1. Back propagation is the most widely used today. It is a form of gradient descent.
2. Gradient descent is a mathematical class of optimization methods and includes numerous algorithms written by a number of researchers and mathematicians.
3. Simulated annealing is a computer emulation of a physical process and attempts to optimize a solution using laws of physics.
4. Other optimization methods including GRG (generalized reduced gradient), and a host of other methods including commercial packages and public domain programs.
5. Genetic algorithms (GA) have been employed to set the weights and research is being done on using them to design the network architecture. Genetic algorithms or evolutionary computing is an exploited search in N space (multi dimensional search where the system is told if it is getting 'hot' or 'cold' in the search direction). This does not guarantee optimal results, but near optimal, which may be good enough. These are effective with large problems, but because of the overhead in setting up a GA, they are slower than other methods when working with smaller less complex data sets.
6. Exhaustive search is where all possible values and combinations are tried. This has great results, but because of the possible number of trials, the time required for to do this can be excessive (a large system might take 2 years to do an exhaustive search, far beyond a reasonable time). For a small system this is good as it does guarantee optimality.
7. Good guessing can be employed, but lots of luck. Performance and accuracy will be highly dependent upon the skill of the operator or person doing the guessing. This is not a recommended method.
8. Research is currently being conducted to discover if there is some way that knowledge about the problem, as well as some new mathematical process or investigation of the data set can provide the answers.
9. There are new methods under investigation that working in a very narrow domain for a specific process might be able to be self-architecting and optimizing.

As with almost any task that we must attend to, knowing which tool to use is very important. The right tool must be picked to fit the problem at hand. There are no reliable and proven heuristics to say which method to use for a certain problem. This is another place where an artificial neural network is operator dependent.

Problems/disadvantages:

1. We don't know what the ANN is looking at. A famous anecdote is when the military was trying to detect and identify enemy tanks in the field. The training set included friendly forces in a daylight setting and enemy forces in lowlight conditions. The trained system then decided based upon lighting. The ANN was looking at the lighting instead of the tank.
2. We don't know what it knows. It has learned much from the data, yet we do not know what this knowledge is. A neural network uses many interconnections and weights, which currently cannot be interpreted. Software engineers call a process such as this where they can't look into and understand it, a 'black box'. If this is being used for credit scoring, the person has the right to know why they were turned down. Yet, without knowledge, or knowing what it was looking at, you could not reply to the customer, creating a legal exposure and possible lawsuit.
3. The ANN is too operator dependent. As pointed out earlier, the operator designs the architectures, organizes and selects the training, and makes numerous decisions in preparing the system. The shortfall is obvious, as the performance and results obtained are very much a function of the operator and the kind of a day that he/she is having.
4. The system works in a black box. It is doing its thing, and we have no idea of what it is doing or why.

Future

Hybrid systems encompassing fuzzy logic for inputs and outputs, genetic algorithms for training, and genetic algorithms or other optimization techniques for setting network architecture are already under investigation. Other work to make neural networks more autonomous and take the human operator out of the loop is also being researched and will bring these systems more to common and accepted use.

Rule extraction to learn what knowledge is in the data is another current research area. There are a few packages currently available and a search on the Internet will find several academic articles on the subject. This process allows one to look at the data and give us some indications as to what the neural network is looking at and tell us a little of what is going on in the system where the data was recorded.

As mentioned earlier, parallel processing computers, as they start to have a larger number of processors, will start to be able to run neural networks in near real time, and will have the power to run larger networks, feedback networks, and perform autonomous learning and machine cognition. This has the potential to have the power and performance of some of the machines in science fiction (Like Hal of *2001* or Colossus of *the Forbin Project*).

Some interest has been shown internationally for the development of a dedicated chip neural network. A few semiconductor companies had some 8-bit MPU's (micro processor units) with software code for emulating neural networks. There have been a few actual neural network chips planned and built. One proposal out of the Far East was for such a chip to be built that would power user friendly appliances, such as a coffee pot or toaster that learned how you liked your coffee or toast, respectively.

Summary

Artificial neural networks are not truly intelligent, but do exhibit intelligent behavior. They are data driven general function approximators that are presently highly dependent upon human interaction to design their architecture and train them. This article is a short overview and is not inclusive of all information about neural networks, as that would require a small encyclopedia.

About the Author

Dr Morantz received his doctorate and masters from Georgia State University in (Mathematical) Decision Sciences with a minor in Computer Science. His BS is from Cleveland State University in Computer Information Systems and Electrical Engineering. He also attended CWRU for Electrical Engineering and Arizona State University for Computational BioScience. He has worked in defense, biomedical engineering, and automotive electronics. Dr Morantz has published in academic journals and conference proceedings on artificial neural networks, machine cognition, automated anomaly detection, data mining, and similar subjects.

References

The Author's web page www.geocities.com/bradscientist
The IEEE www.ieee.org
IEEE Transactions on Neural Networks
IEEE Intelligent Systems Journal
IEEE Neural Network Society
American Association for Artificial Intelligence www.aaai.org
Z Solutions has trial software www.zsolutions.com
International Joint Conference on Neural networks www.ijcnn.net