NEW SCOPES IN ARTIFICIAL NEURAL NETWORK

AMIT BOTKE, GOURAV PATHAK

Abstract: In this paper the author describe the use of neural network in various related fields artificial neural networks are parallel computational models, comprised of densely interconnected adaptive processing units. These networks are fine-grained parallel implementations of nonlinear static or dynamic systems. A very important feature of these networks is their adaptive nature where "learning by example" replaces "programming" in solving problems. This feature makes such computational models very appealing in application domains where one has little or incomplete understanding of the problem to be solved, but where training data is available. Another key feature is the intrinsic parallel architecture which allows for fast computation of solutions when these networks are implemented on parallel digital computers or, ultimately, when implemented in customized hardware.

Keywords: Image Compression, SOM, MJ Futures, CATCH

Introduction: Artificial neural networks are viable computational models for a wide variety of problems. These include pattern classification, speech synthesis and recognition, adaptive interfaces between humans and complex physical systems, function approximation, image compression, associative memory, clustering, forecasting and prediction, combinatorial optimization, nonlinear system modeling, and control. These networks are "neural" in the sense that they may have been inspired by neuroscience, but not necessarily because they are faithful models of biological neural or cognitive phenomena. In fact, the majority of the networks covered in this book are more closely related to traditional mathematical and/or statistical models such as non-parametric pattern classifiers, clustering algorithms, nonlinear filters, and statistical regression models than they do with neurobiological models.

Historical background: The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not allow them to do too much. Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras. Many important advances have been boosted by the use of inexpensive computer emulations. Following an initial period of enthusiasm, the field survived a period of frustration and disrepute. During this period when funding and professional support was minimal, important advances were made by relatively few researchers. These pioneers were able to develop convincing technology which surpassed the limitations identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding.

Why use neural networks?: Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2. Self-Organization: An ANN can create its own organisation or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Neural networks versus conventional computers: Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do. Neural networks process

information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurones) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable. On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to solved must be known and stated in small unambiguous instructions. These instructions are then converted to a high level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault. Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks, require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

Applications of neural networks:

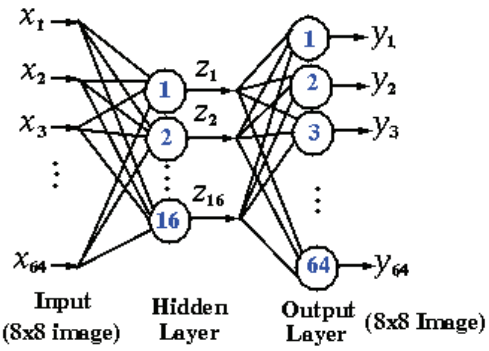
Character Recognition: The idea of character recognition has become very important as handheld devices like the Palm Pilot are becoming increasingly popular. Neural networks can be used to recognize handwritten characters.

Feed-forward networks: character recognition: The idea of using feedforward networks to recognize handwritten characters is rather straightforward. As in most supervised training, the bitmap pattern of the handwritten character is treated as an input, with the correct letter or digit as the desired output. Normally such programs require the user to train the network by providing the program with their handwritten patterns.

Image Compression: Neural networks can receive and process vast amounts of information at once, making them useful in image compression. With the Internet explosion and more sites using more images on their sites, using neural networks for image compression is worth a look.

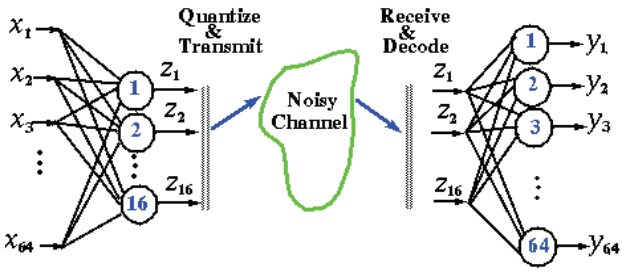
Image Compression using Backprop: Computer images are extremely data intensive and hence require large amounts of memory for storage. As a result, the transmission of an image from one machine to another can be very time consuming. By using data compression techniques, it is possible to remove some of the redundant information contained in images, requiring less storage space and less time to transmit. Neural nets can be used for the purpose of image compression, as shown in the following demonstration. A neural net architecture suitable for solving the image compression problem is shown below. This type of structure--a large input layer feeding into a small hidden layer, which then feeds into a large output layer is referred to as a bottleneck type network. The idea is this: suppose that the neural net shown below had been trained to implement the identity map. Then, a tiny image presented to the network as input would appear exactly the same at the output layer.

Bottleneck-type Neural Net Architecture for Image Compression:



In this case, the network could be used for image compression by breaking it in two as shown in the Figure below. The transmitter encodes and then transmits the output of the hidden layer (only 16 values as compared to the 64 values of the original image).The receiver receives and decodes the 16 hidden outputs and generates the 64 outputs. Since the network is implementing an identity map, the output at the receiver is an exact reconstruction of the original image.

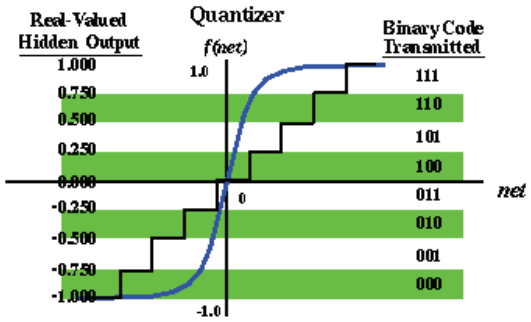
The Image Compression Scheme using the Trained Neural Net



Actually, even though the bottleneck takes us from 64 nodes down to 16 nodes, no real compression has occurred because unlike the 64 original inputs which are 8-bit pixel values, the outputs of the hidden layer are real-valued (between -1 and 1), which requires

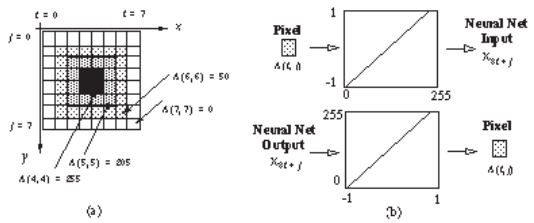
possibly an infinite number of bits to transmit. True image compression occurs when the hidden layer outputs are quantized before transmission. The Figure below shows a typical quantization scheme using 3 bits to encode each input. In this case, there are 8 possible binary codes which may be formed: 000, 001, 010, 011, 100, 101, 110, 111. Each of these codes represents a range of values for a hidden unit output. For example, consider the first hidden output. When the value of is between -1.0 and -0.75, then the code 000 is transmitted; when is between 0.25 and 0.5, then 101 is transmitted. To compute the amount of image compression (measured in bits-per-pixel) for this level of quantization, we compute the ratio of the total number of bits transmitted: to the total number of pixels in the original image: 64; so in this case, the compression rate is given as bits/pixel. Using 8 bit quantization of the hidden units gives a compression rate of bits/pixel.

The Quanitization of Hidden Unit Outputs



The training of the neural net proceeds as follows; a 256x256 training image is used to train the bottleneck type network to learn the required identity map. Training input-output pairs are produced from the training image by extracting small 8x8 chunks of the image chosen at a uniformly random location in the image. The easiest way to extract such a random chunk i s to generate a pair of random integers to serve as the upper left hand corner of the extracted chunk. In this case, we choose random integers *i* and *j*, each between 0 and 248, and then (*i*,*j*) is the coordinate of the upper left hand corner of the extracted chunk. The pixel values of the extracted image chunk are sent (left to right, top to bottom) through the pixel-to-real mapping shown in the Figure below to construct the 64-dimensional neural net input. Since the goal is to learn the identity map, the desired target for the constructed input is itself; hence, the training pair is used to update the weights of the network.

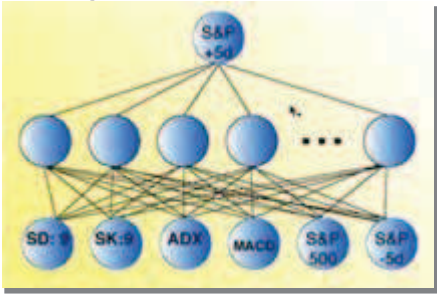
The Pixel-to-Real and Real-to-Pixel Conversions:



Once training is complete, image compression is demonstrated in the recall phase. In this case, we still present the neural net with 8x8 chunks of the image, but now instead of randomly selecting the location of each chunk, we select the chunks in sequence from left to right and from top to bottom. For each such 8x8 chunk, the output the network can be computed and displayed on the screen to visually observe the performance of neural net image compression. In addition, the 16 outputs of the hidden layer can be grouped into a 4x4 "compressed image", which can be displayed as well.

Stock Market Prediction: The day-to-day business of the stock market is extremely complicated. Many factors weigh in whether a given stock will go up or down on any given day. Since neural networks can examine a lot of information quickly and sort it all out, they can be used to predict stock prices. Neural networks have been touted as all-powerful tools in stock-market prediction. Companies such as MJ Futures claim amazing 199.2% returns over a 2-year period using their neural network prediction methods. They also claim great ease of use; as technical editor John Sweeney said in a 1995 issue of "Technical Analysis of Stocks and Commodities," "you can skip developing complex rules (and redeveloping them as their effectiveness fades) . . . just define the price series and indicators you want to use, and the neural network does the rest." These may be exaggerated claims, and, indeed, neural networks may be easy to use once the network is set up, but the setup and training of the network requires skill, experience, and patience. It's not all hype, though; neural networks have shown success at prediction of market trends. The idea of stock market prediction is not new, of course. Business people often attempt to anticipate the market by interpreting external parameters, such as economic indicators, public opinion, and current political climate. The question is, though, if neural networks can discover trends in data that humans might not notice, and successfully use these trends in their predictions. Good results have been achieved by Dean Barr and Walter Loick at LBS Capital Management using a relatively simple neural network with just 6 financial indicators as inputs. These inputs include the ADX, which indicates the average directional movement over the previous 18 days, the current value of the S&P 500,

and the net change in the S&P 500 value from five days prior (see David Skapura's book "Building Neural Networks," p129-154, for more detailed information). This is a simple back-propagation network of three layers, and it is trained and tested on a high volume of historical market data. The challenge here is not in the network architecture itself, but instead in the choice of variables and the information used for training. I could not find the accuracy rates for this network, but my



source claimed it achieved "remarkable success" (this source was a textbook, not a NN-prediction-selling website!). Even better results have been achieved with a back-propagated neural network with 2 hidden layers and many more than 6 variables. I have not been able to find more details on these network architectures, however; the companies that work with them seem to want to keep their details secret.

Additional Neural Network Applications in the financial world:

- Currency prediction
- Futures prediction
- Bond ratings
- Business failure prediction
- Debt risk assessment
- Credit approval
- Bank theft
- Bank failure

Traveling Saleman's Problem: Interestingly enough, neural networks can solve the traveling salesman problem, but only to a certain degree of approximation.

The travelling salesman's problem: Definition: *Given a number of cities on a plane, find the shortest path through which one can visit all of the cities.*

In contrast to using recursion to try all the different possibilities, we are going to approximate the solution using a Kohonen SOM, which organizes itself like a elastic rubber band. The basic idea of this solution is to start out with a elastic net in a random orientation:

The algorithm is to choose a random city each time, and pull the point on the net that is closest to the city towards the city:

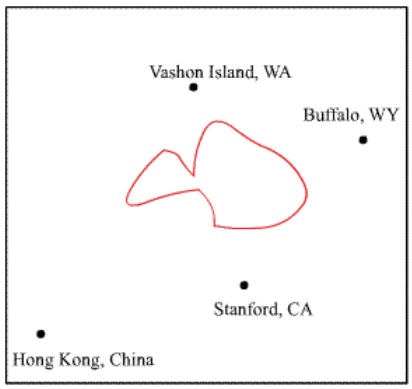


Fig.1: A point pulled towards Vashon Island brings the net closer to the city.

Since the net is elastic, each pull changes the shape of the net. After many pulls for different cities, the net eventually converges into a ring around the cities. Given the elastic property of the ring, the length of the ring tends to be minimized. This is how the TSP can be approximated.

The details: As in simple competitive networks, the weight vectors of the networks are randomly assigned at the beginning in SOM's. Another similarity is that SOM also identifies the winner (the perceptron whose weight vector is closest to the input vector) every time an input vector is presented. The only difference is that while in simple competitive networks only the winner learns, in SOM's all nodes learn, but the rate of learning varies inversely with the node's physical distance from the winner. The Kohonen SOM used in solving the TSP has the following structure:

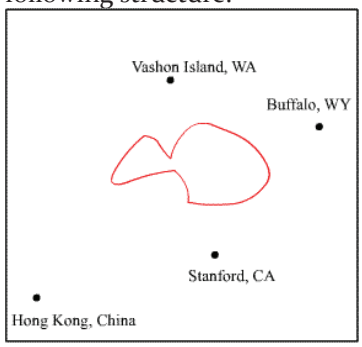


Fig.2 A randon ring.

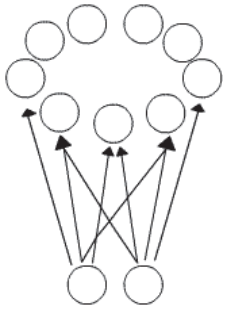


Fig.3 A Kohonen net with a ring-shaped top layer.

Recall the elastic rubber-band model mentioned above, such a ring shaped map simulates a rubber-band if we consider the weight vectors as points on a plane. We can join these points together according to the position of their respective perceptron in the ring of the top layer of the network. Suppose the coordinates of a city (x, y) is presented as the input vector of the network, the network will identify the weight vector closest to the city and move it and its neighbors closer to the city. In this manner, the weight vectors behave as points on a rubber band and each "learning" is analogous to pulling the closest point of the band towards a city. The rule that the amount of learning varies inversely with the physical distance between the node and the winner is what leads to the elastic property of the rubber band.

Medicine, Electronic Nose, Security, and Loan Applications: These are some applications that are in their proof-of-concept stage, with the acception of a neural network that will decide whether or not to grant a loan, something that has already been used more successfully than many humans.

Medicine: One of the areas that has gained attention is in cardiopulmonary diagnostics. The ways neural networks work in this area or other areas of medical diagnosis is by the comparison of many different models. A patient may have regular checkups in a particular area, increasing the possibility of detecting a disease or dysfunction. The data may include heart rate, blood pressure, breathing rate, etc. to different models. The models may include variations for age, sex, and level of physical activity. Each individual's physiological data is compared to previous physiological data and/or data of the various generic models. The deviations from the norm are compared to the known causes of deviations for each medical condition. The neural network can learn by studying the different conditions and models, merging them to form a complete conceptual picture, and then diagnose a patient's condition based upon the models.

Electronic Noses:

An actual electronic "nose"
Image courtesy Pacific Northwest Laboratory

The idea of a chemical nose may seem a bit absurd, but it has several real-world applications. The electronic nose is composed of a chemical sensing system (such as a spectrometer) and an artificial neural network, which recognizes certain patterns of chemicals. An odor is passed over the chemical sensor array, these chemicals are then translated into a format that the computer can understand, and the artificial neural network identifies the chemical. A list at the Pacific Northwest Laboratory has several different applications in the environment, medical, and food industries. Environment: identification of toxic wastes, analysis of fuel mixtures (7-11 example), detection of oil leaks, identification of household odors, monitoring air quality, monitoring factory emission, and testing ground water for odors. Medical: The idea of using these in the medical field is to examine odors from the body to identify and diagnose problems. Odors in the breath, infected wounds, and body fluids all can indicate problems. Artificial neural networks have even been used to detect tuberculosis. Food: The food industry is perhaps the biggest practical market for electronic noses, assisting or replacing entirely humans. Inspection of food, grading quality of food, fish inspection, fermentation control, checking mayonnaise for rancidity, automated flavor control, monitoring cheese ripening, verifying if orange juice is natural, beverage container inspection, and grading whiskey.

Security: One program that has already been started is the CATCH program. CATCH, an acronym for Computer Aided Tracking and Characterization of Homicides. It learns about an existing crime, the location of the crime, and the particular characteristics of the offense. The program is subdivided into different tools, each of which place an emphasis on a certain characteristic or group of characteristics. This allows the user to remove certain characteristics which humans determine are unrelated.

Loans and credit cards: Loan granting is one area in which neural networks can aid humans, as it is an area not based on a predetermined and pre-weighted criteria, but answers are instead nebulous. Banks want to make as much money as they can, and one way to do this is to lower the failure rate by using neural networks to decide whether the bank should approve the loan. Neural networks are particularly useful in this area since no process will guarantee 100% accuracy. Even an 85-90% accuracy would be an improvement over the methods humans use. In fact, in some banks, the failure rate of loans approved using neural networks is lower than that of some of their best traditional methods. Some credit card companies are now beginning to use neural networks in deciding whether to grant an application. The process works by analyzing past failures and making current decisions based upon past experience. Nonetheless, this creates its own problems. For

example, the bank or credit company must justify their decision to the applicant. The reason "my neural network computer recommended against it" simply isn't enough for people to accept. The process of explaining how the network learned and on what characteristics the neural network made its decision is difficult. As we alluded to earlier in the history of neural networks, self-modifying code is very difficult to debug and thus difficult to trace. Recording the steps it went through isn't enough, as it might be using conventional computing, because even the individual steps the neural network went through have to be analyzed by human beings, or possibly the network itself, to determine that a particular piece of data was crucial in the decision-making process.

Conclusion: The computing world has a lot to gain from neural networks. Their ability to learn by example makes them very flexible and powerful. Furthermore there is no need to devise an algorithm in order to perform a specific task; i.e. there is no need to understand the internal mechanisms of that task. They are also very well suited for real time systems because of their fast response and computational times which are due to their parallel architecture. Neural networks also contribute to other areas of research such as neurology and psychology. They are regularly used to model parts of living organisms and to investigate the internal mechanisms of the brain. Perhaps the most exciting aspect of neural networks is the possibility that some day 'conscious' networks might be produced. There is a number of scientists arguing that consciousness is a 'mechanical' property and that 'conscious' neural networks are a realistic possibility. Finally, I would like to state that even though neural networks have a huge potential we will only get the best of them when they are integrated with computing, AI, fuzzy logic and related subjects.

References:

1. An introduction to neural computing. Aleksander, I. and Morton, H. 2nd edition
2. Neural Networks at Pacific Northwest National Laboratory http://www.emsl.pnl.gov:2080/docs/cie/neural/neural.homepage.html
3. Industrial Applications of Neural Networks (research reports Esprit, I.F.Croall, J.P.Mason)
4. A Novel Approach to Modelling and Diagnosing the Cardiovascular System http://www.emsl.pnl.gov:2080/docs/cie/neural/papers2/keller.wcnn95.abs.html
5. Artificial Neural Networks in Medicine http://www.emsl.pnl.gov:2080/docs/cie/techbrief/NN.techbrief.ht
6. M. Venkateswarlu , B. Rajini Kanth, Viscoelastic Properties of Nanoparticulated Magnetite and Cobalt Ferrite Based Mr Fluids; Engineering Sciences international Research Journal: ISSN 2320-4338 Volume 3 Issue 1 (2015), Pg 12-16
7. Neural Networks by Eric Davalo and Patrick Naim
8. Learning internal representations by error propagation by Rumelhart, Hinton and Williams (1986).
9. Klimasauskas, CC. (1989). The 1989 Neuro Computing Bibliography. Hammerstrom, D. (1986). A Connectionist/Neural Network Bibliography.
10. DARPA Neural Network Study (October, 1987-February, 1989). MIT Lincoln Lab. Neural Networks, Eric Davalo and Patrick Naim
11. Assimov, I (1984, 1950), Robot, Ballatine, New York.
12. Electronic Noses for Telemedicine http://www.emsl.pnl.gov:2080/docs/cie/neural/papers2/keller.ccc95.abs.html
13. Pattern Recognition of Pathology Images http://kopernik-eth.npac.syr.edu:1200/Task4/pattern.html
14. http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Introduction to neural networks
15. Dr.K.V.Radha, Mass Transfer Correlation for Decolorization Studies in Immobilized Packed Bed Bioreactor; Engineering Sciences international Research Journal: ISSN 2320-4338 Volume 3 Issue 1 (2015), Pg 17-19
16. http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Appendix A - Historical background in detail
17. http://www.willamette.edu/~gorr/classes/cs449/brain.html
18. http://www.willamette.edu/~gorr/classes/cs449/intro.html

***

Amit Botke/ Student/
Gourav Pathak/ Student/