A Personal Revisitation of Neural Nets

If a program, good and timely,
An author does supply,
On due and measured credit
That author may rely.

But if his chosen model
Has been a neural net,
A much inflated credit
He certainly will get.

What a brain, will people say;
What a magic touch!
T'were churlish to be critical;
No praise can be too much.

Neural nets have a long history. The first paper on the subject, by McCulloch Pitts, appeared in May 1943 in the Bulletin of Mathematical Biophysics [1]. This was before even the ENIAC had been built. The paper was written in the jargon of neurology, but what it actually contained was the proof of a number of results in formal logic which could be applied to the description of threshold switching circuits. Von Neumann refers to this work in his EDVAC report and knowledge of it evidently conditioned his reaction to what he learned about the ENIAC project at the Moore School.

I saw some of the earliest practical work on neural nets at MIT in the late 1960s when I was a fairly frequent visitor. The work led to a book by Minsky and Pappert [2] in which it was shown that the simple perceptrons, then in use to realise neural nets, were incapable of dealing with problems involving certain classes of functions, notably exclusive OR functions. Minsky and Pappert offered many insights but few clarifications. It was perhaps for this reason that their book did not have the effect of stimulating further theoretical work; rather it had the effect of damping down enthusiasm for neural nets altogether.

It was not until around 1980 that interest revived. The development of new training algorithms overcame earlier limitations, notably, by making it possible to work with (and train) perceptrons having more than one hidden layer. Equally important was the fact that vastly more computing power had by then become readily available. Neural nets are very hungry for computation, particularly in the training phase.

In January 1990, I attended some presentations on neural nets at the Triumph-Adler Laboratory in Nuremberg (then associated with Olivetti) given by well-known European experts. These were in the context of speech analysis and synthesis, subjects in which the Olivetti Research Directorate then had a major interest. The use of neural nets for these purposes was being promoted, but it soon became clear to me that the resulting systems could not match what was being done by conventional methods. I concluded that I could forget about neural nets, at any rate for the time being. I still believe that this conclusion was correct at the time. The presentations at Nuremberg were of a practical nature, and there was no mention of the theory of neural nets, which was was advancing rapidly at the time. For a good discussion of the present state of the theory see reference [3].

Neural nets are useful for problems such as character recognition in which sets of data, represented by matrices, are to be classified according to examples given. The net must be trained using a set of matrices which have already been classified by a human agent. Its performance is then assessed on another set of matrices, independent of the first but drawn from the same statistical population. The time taken for training depends on the size of the net, varying from a time measured in minutes to one measured in days.

The training process is essentially an iterative adjustment of the parameters of the net so as to minimise a certain non-linear function of the input variables. The function is not written down explicitly, but it is implicit in the design of the net.

The process is closely analogous to curve fitting by least squares and has a similar statistical basis. While all training algorithms have the effect of minimizing a function by some form of gradient descent, the details of the algorithms vary and every worker in the field has his favourites. Similarly, there are many devices that can be used to avoid settling on a false minimum. The design of the net itself is at present an art rather than a science. There is limited prior experience on which a designer can draw and he is largely thrown back on his own ingenuity and intuition.

Neural nets have a special appeal to those attracted towards artificial intelligence. Many of them are associated with, or have been trained in, artificial intelligence departments in universities, or in computer science departments where artificial intelligence is emphasised. However, the romantic appeal of artificial intelligence, is widespread, and includes many project managers and others who work in industry. The result is that there is a tendency for neural nets to be looked on with special favour in the computer world at large. The piece of doggerel printed above makes this point in a lighthearted way.

There is, and will no doubt continue to be, no shortage of people who are drawn by the connection with artificial intelligence and become specialists in neural nets *per se*. On the other hand, there are others, whose interest lies in the general area of classification including pattern recognition and scene analysis. To them neural nets provide one line of approach only; other important approaches are, for example, by way of hidden Markov models or rule-based systems. These various approaches are very similar from the mathematical point of view and have their basis in the same kind of statistical theory. Such people tend to distance themselves from artificial intelligence or at any rate from neurology.

I asked the various people whom I consulted to give me examples of neural nets that had proved to be of real utility in scientific or industrial applications. Several examples were mentioned, but my informants could not give me details. An example mentioned more than once was the use, by the US Postal Service, of a neural net for the recognition of hand-written Zip codes on letters. This application sounded very impressive. I had visions of an envelope being put under a scanner and of the Zip code flashing up on the screen. Although my informants could not give me details of this application, they were able to provide references which I resolved to follow up. This led to my exposure to the fascinating field of postal automation.

The first reference was to work done by a group at Bell Laboratories, Holmdel [4]. The group worked with a sample of real mail but they did not, as I had expected, start with a scan of the whole envelope and proceed to locate the Zip code. That difficult task had already been done by another organisation and the workers at Bell Laboratories started with Zip cods that had already been isolated. However, they succeeded in designing and building a neural net which proved capable of deciphering the written Zip code with a useful degree of accuracy.

The above paper contained a reference to another paper dealing specifically with postal automation in the United States [5]. The US Postal Service had long been working towards the ambitious goal of generating automatically and printing on an envelope a machine-readable bar-code that would contain sufficient information to enable the letter to be directed, without further human intervention, to the point at which it passes into the hands of the postman for final delivery. The work had gone well and large-scale deployment of a practical system began in the early 1980s.

The information in the bar-code consists of the basic Zip code (5 digits) and the 4 'add-on' digits that the US public are now urged to include, but frequently do not; in addition there are 2 extra digits used only within the Postal Service. The system makes the best use it can of the Zip code digits supplied by the sender. It obtains further information by locating and deciphering the destination address and referring to a file of all addresses recognised as valid by the Postal Service. If this information does not enable the system to generate the bar-code, then an image of the envelope is transmitted, via a T1 connection, to an off-site human operator who reads the envelope and keys in the missing information.

The substantial computing power required for generating the bar-code is provided, by a a rack containing seven Power PC boards on a VME bus, one for system control and six for character recognition. There is also an RS6000 for operator intervention and control. Each recognition processor has 64 MB of memory and a 4 GB hard disk for holding the systems software and the national Zip code directory.

This system proved to be very effective on mail with printed or type written addresses, but the software for interpreting hand writing was not sufficiently powerful to enable it to deal successfully in the time available with the hand written envelopes which form 60% of the total work load. An effort began in March 1996 to incorporate an improved hand written address recognition system developed at SUNY Buffalo [6]. This involved upgrading the hardware to ten processors and adding a symmetric multi-processors system with slots for eight processors although, initially, three only of these slots were populated.

Non-automated processing of mail is highly labour intensive and the ultimate value, in economic terms, of an automated system depends on how many human operators can be dispensed with. Even if the system can handle successfully only a small part of the mail offered to it, the savings can be considerable. From the point of view of the quality of the postal service, it does not matter if the system rejects a substantial proportion of the mail, provided that it rejects it outright; only very rarely should it print an erroneous bar-code that would cause a letter to be sent to a completely wrong part of the country. The proportion of false bar-codes can be adjusted by setting the threshold for rejection.

In the system as enhanced there are three hand-written digit recognition modules each with its strength and weaknesses. The system is capable of exploiting these three modules, in a flexible manner, in response to the difficulties presented by individual pieces of mail. Similarly there are two word recognition modules.

One of the digit recognition modules is based on the neural net developed at Holmdel. It may be said, therefore, that neural nets have, in fair competition, managed to establish a small but honourable niche for themselves in the postal application. It is limited to the interpretation of individual hand written digits or groups of digits, and there are no sign that neural nets are on the point of coming to play a central role in the overall analysis of the writing on an envelope as a structured entity. I suspect that a similar picture might emerge from a study of other application fields in which neural nets can be used, although the postal application is the only one that I have followed up.

General Comments on Neural Nets

It is apparent that neural nets are still in their infancy. A better understanding is needed of what problems lend themselves to approach by way of neural nets and of how the designer of a net may build into it—by means of a preprocessor or otherwise—features that reflect what he knows of the problem and its structure. It is sometimes said that neural nets provide the second best solution to a problem; if they work in spite of having very little human-type knowledge of the application built into them then, it is claimed, a method with more such knowledge would surely be superior. On the other hand, a neural net can be surprisingly fast at run time, since the information incorporated into it during training is very efficiently encoded. Moreover, if a drift in the workload takes place a neural net can be retrained with a new training set. A system based on a hidden Markov model or a rule-based system would need human intellectual effort to make it meet the new conditions.

How far the development of neural nets will go, either on their own or combined with other methods, is impossible to assess. While we may certainly expect progress, it is not likely to be rapid. There are no signs that neural nets will come to dominate the field of character and scene recognition to the exclusion of other methods.

I have been concerned solely with neural nets as a practical weapon in the armoury of the software engineer. Their importance for the philosophical study of machine learning is not in doubt.

Advice to Managers

The question arises as to how a software engineer should go about giving advice to a research manager on issues like the one discussed here. If the manager is strong technically, he may not need advising being able himself to carry through the kind of investigation that I have described. If he has not time to do the investigation himself and requests a briefing, then he will at least be able to follow the argument and to read the key references if necessary.

A non-technical manager will need to consult experts weighing their opinions according to his assessment of them as people. If he is not a good judge of people, he has no right to be in a managerial position at all.

Managers are as liable as anyone else to respond to the lure of artificial intelligence and a consultant will need to proceed with tact. Perhaps my little piece of doggerel will help him.

References

1. W. S. McCulloch and W. Pitts, ‘A Logical Calculus of the Ideas Immanent in Nervous Activity’, *Bulletin of Mathematical Biophysics,* vol 5 p 115, 1943

2. M. L. Minsky and S. A. Pappert, *Perceptrons,* MIT Press, 1969

3. C. M. Bishop, *Neural Networks for Pattern Recognition,* Oxford, 1995

4. Y. LeCun, *et al,* ‘Backpropagation Applied to Handwritten Zip Code Recognition’, *Neural Computation,* vol 1 pp 541–551, 1989

5. Hing-Huei Wong and S. N. Srihari, 'A Framework for Object Recognition in a Visually Complex Environment and its Application to Locating Address Blocks on Mail Pieces', *International Journal of Computer Vision,* vol 2 pp 125–151, 1988

6. S. N. Srihari and E. J. Kuebert, 'Integration of Hand-Written Address Interpretation into the United States Postal Service Remote Computer Reader System,' *Proc 4th International Conference on Document Analysis and Recognition,* pp 892–896, 1997

Maurice V. Wilkes
Version dated 11:53 hours on 10 August 2001