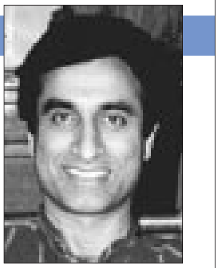


Better Web searches and prediction with instantaneously trained neural networks

By Subhash Kak
Louisiana State University
kak@ee.lsu.edu

Consider a robot reconnaissance plane looking for man-made objects on unfamiliar terrain. To be effective, the vehicle needs to process images and generalize based on prestored primitives in real time. Because rule-based statistical methods and neural network methods—such as the backpropagation algorithm or self-organizing maps, the standard techniques for generalization—are notoriously slow, fast pattern-learning techniques are becoming increasingly necessary.

Standard artificial neural networks can serve as models of biological memory embodied as strongly connected and layered networks of processing units. These feedback (Hopfield with delta learning) and feedforward (backpropagation) networks learn patterns slowly: the network must adjust weights connecting links between input and output layers (see Figure 1) until it obtains the correct response to the training patterns. But biological learning is not a single process: some forms are very quick and others relatively slow. Short-term biological memory, in particular, works very quickly, so slow neural network models are not plausible candidates in this case.

Over the past few years, my colleagues and I have developed new neural network designs that model working memory in their ability to learn and generalize instantaneously.[1–3] These networks are almost as good as backpropagation in the quality of their generalization.[4] With their speed advantage, they will work in many real-time signal-processing, data-compression, forecasting, and pattern-recognition applications. In this report, I describe the networks and their applications to two problems: time-series prediction and an intelligent Web metasearch engine design. My descriptions should indicate how these designs could work in other situations.

Different kinds of memory

To provide a context for examining instantaneous learning, let's first consider different types of biological memory. Although described separately, different memory types appear to be fundamentally interrelated. The classification of memory types can take a variety of forms.[5]

First, many kinds of sensory memory systems help us perceive the world. For example, visual memory includes components that let a memory trace persist for about one-tenth of a second. This persistence lets us see continuous motion in the discrete frames of a television broadcast. Another component to this memory, more sensitive to shape than brightness, integrates information arriving from the two retinas. Like visual persistence, a memory related to auditory persistence creates an echo that lingers after the item has been spoken. That's why we remember the later words in a series better if we hear them rather than read them.

There are memories about facts, events, skills, and habits as well. Some are based on language, others aren't. Fact and event memory is distinct from other kinds of memory, such as the memory forming the basis of skills and habits. Declarative *(explicit)* memory refers to facts and events. Such memory can form after single events. Although we generally acquire nondeclarative *(implicit)* memory across several presentations of the stimulus, in situations such as taste aversion, a person might acquire it after a single event. Declarative memory is flexible and can be readily applied to novel situations, while nondeclarative memory tends to be inflexible and defined in the context of the learning situation. Implicit knowledge is not readily accessed by response systems that did not participate in the original learning.

Implicit memory lets us recall and make judgments about words, objects, and images without any conscious recall of prior experience. That is, we have an aspect of memory that is not a component of conscious recall. Consequently, we cannot view memory as something that is stored somewhere. Rather, we need to see it as part of the brain's reorganization process.

When a subject sees or hears a word or object several times, that subject will see or hear it more readily on second or later occasions. This *priming* phenomenon operates across a wide range of sensory and motor systems, at various levels of processing. Implicit memory is a manifestation of priming. In perceptual representational learning, the experience of an object on one occasion facilitates the perception of the same or a similar object on a subsequent occasion.

Semantic memory represents the individual's general knowledge of the world, whereas episodic memory structures our personal experiences. Thus semantic memory includes the meaning of words, formulas of different kinds, and geographical knowledge, whereas *episodic memory* deals with particular incidents, such as a visit to the doctor last week.

If we consider the distinction between short- and long-term memories, we note that in human amnesia, short-term memory is usually intact. The problem therefore relates to the storage of the information and not an impairment in perception or rule learning.

Working memory

More than 100 years ago, psychologists used the *digit span* to estimate working memory capacity.[6] To determine digit span, the subject views a sequence of digits that are to be repeated back in the same order. The length of the sequence is increased to the point where the subject always fails. The sequence at which the subject is right half the time is the digit span. For most people, the digit span is seven digits plus or minus two, with the actual values distributed from four or five to 10 or more.

If the digits are parceled into chunks, the number of digits memorized can increase greatly. Apparently, the number of chunks rather than the number of digits determines the capacity of immediate memory. By clever chunking of the parts of a digit sequence, a subject can memorize a very large sequence, running into thousands of digits.

If the immediate memory has many components, there must be a central executive where the processing by the subsystems comes together. The central executive is an attentional system that controls various visual and auditory subsystems, relating them to long-term memory. One auditory system is the *phonological loop*, which involves some process of rehearsal with subvocal speech to maintain the memory trace. Likewise, there appears to be a visio-spatial sketchpad that helps us memorize images.

The ability to remember also depends on mood and the level of physiological arousal. Performance appears to improve as arousal increases, up to some peak, beyond which it deteriorates. Different tasks are optimally performed at different levels of arousal.

Memories are also lost with time. Although learning appears to be linearly related to time, forgetting has a logarithmic relationship: the information loss is very rapid at first, then it slowly levels off. Short-term retention is influenced by events experienced during the retention interval (*retroactive interference*) and those occurring prior to the event that is to be remembered (*proactive interference*). Retroactive interference involves memory impairment caused by events between learning and testing—a new memory can supersede or otherwise impact an older one. In *proactive inhibition*, the reverse process occurs: an old memory interferes with our ability to learn new information.

A model for instantaneously learned working memory

For our purposes here, the most significant aspects of biological working memory are its instantaneous or near-instantaneous nature and its limited capacity.

Our model of an *instantaneously trained neural network* (ITNN) builds on the idea of the dedicated "hardware" of the phonological loop or visio-spatial sketchpad. We take it that the sketchpad-type system faithfully represents all the training samples by allocating a unique neuron to each training sample. Such an allocation implies that the memory will be limited by the sketchpad's size.

Basically, to achieve this allocation, we need a network in which each hidden node acts as a sharp filter for its training sample. This node's output should be correlated with this sample and anticorrelated with all other training samples.

Consider the mapping $Y = f(X)$, where X and Y are n- and m-dimensional binary vectors. We consider binary neurons that output 1 if and only if the sum of the inputs exceeds 0. To realize the filter, we have it act as a hyperplane to separate the corner of the n-dimensional cube represented by the training vector. That's why we call our technique a *corner-classification* (CC) technique.[7] We have various versions of the method, one of the more effective of which is called CC4.[8,9] The weight is –1 if the input is 0 and +1 if the input is 1.

To provide for effective nonzero thresholds to the hyperplane realized by the node, our technique assumes an extra input $x_{n+1} = 1$. The weight of the link from this node to a hidden neuron is $r - s + 1$, where r is the radius of generalization and s is the number of ones in the input sequence. The weights in the output layer are equal to 1 if the output value is 1 and –1 if the output value is 0. This amounts to learning both the input class and its complement. By assigning an input vector in the training sample to a unique hidden neuron, we can claim that the hidden neuron "stores" the training vector.

That the value of r defines the radius of generalization can be seen by considering the all-zero input vectors for which $w_{n+1} = r + 1$. Because all the other weights are –1 each, at most there can be r different +1s in the input vector for this vector to be recognized by its hidden neuron.

The choice of r will depend on the nature of generalization sought. If no generalization is needed, $r = 0$. For exemplar patterns, the choice of r defines the degree of error correction. But the choice will also depend on the number of training samples. Figure 2 shows a network that maps three five-component input vectors into two-component output vectors.

I have described the algorithm for binary inputs; we have also devised versions that

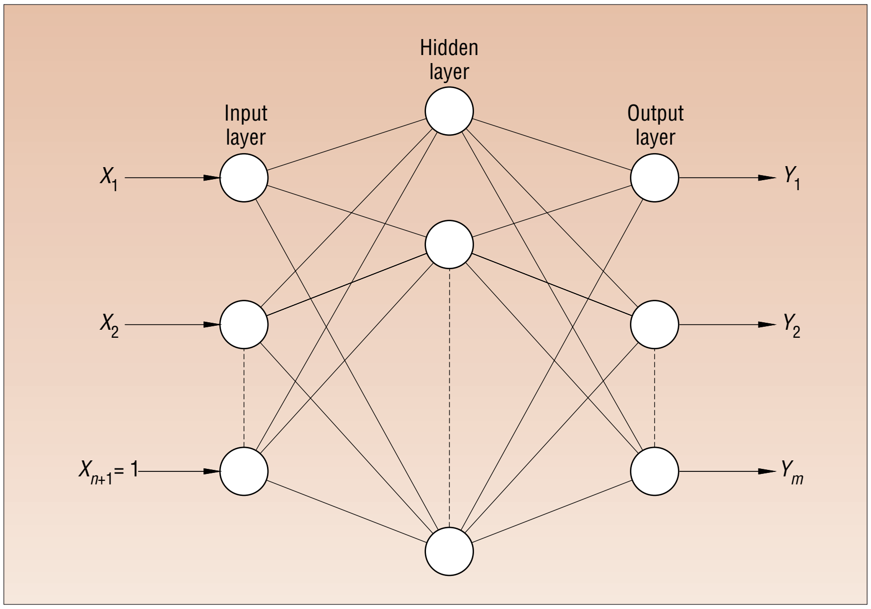


Figure 1. A general network architecture.

work directly for analog inputs. Likewise, we have created versions where the radius of generalization changes adaptively for different points of the training routine, necessitating some recursive training.

Forecasting a time series

Time-series prediction is useful for error-suppression in audio and video signals, in control, and for a variety of business applications. Part of the time series serves for finding the weights; the rest operates in a real-time application involving the time series. In most cases, the active size of the network remains fixed, just like the sketchpad of human working memory.

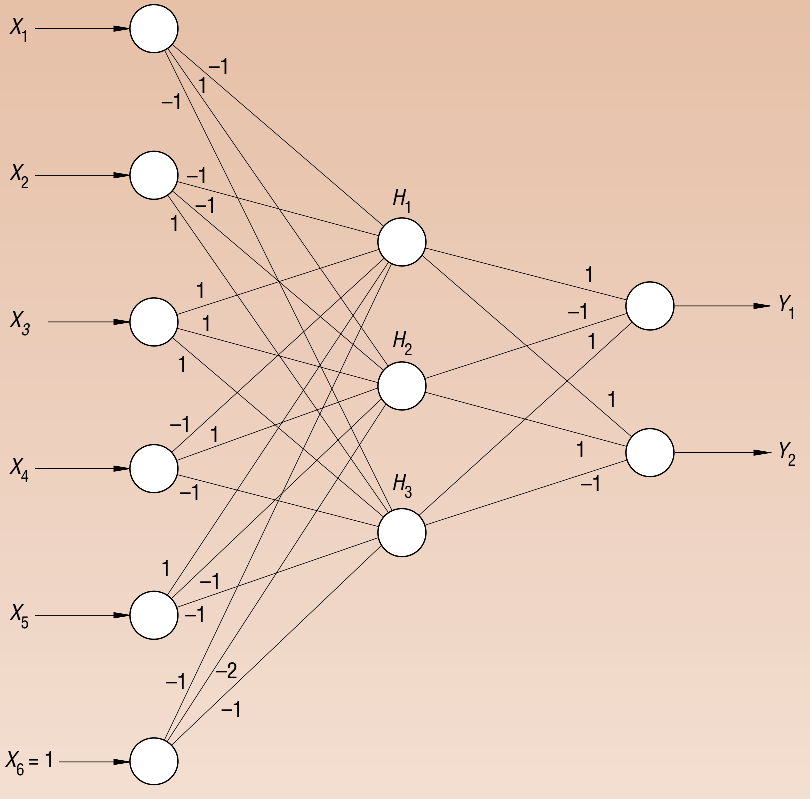
A sliding window of *w* preceding points serves to predict the next point in the time series. Figures 3 and 4 present results for a chaotic time series based on the logistic map.

In these examples, which are part of an ongoing study in my group, the data was quantized and unary-coded. Analog versions of the algorithms produce even better performance.

Design of an intelligent metasearch engine

Recent studies have shown that as indexable Web pages have increased to about 800 million (early 1999), the coverage of single standard search engines has decreased.[10,11] This coverage ranges from the lows of 2.2% for Euroseek and 2.5% for Lycos, to a high of 16% for Northern Light.

Standard search engines provide unequal access to Web pages, picking out those that have more links to them and preferring commercial sites to educational ones. Because they are also out of date, indexing new or modified pages can take months.



Input	s	Weights	Input to			Output of			Input to		Output	
			H_1	H_2	H_3	H_1	H_2	H_3	Y_1	Y_2	Y_1	Y_2
0 0 1 0 1 1	2	–1–1 1–1 1–1	1	–2	–1	1	0	0	1	1	1	1
1 0 1 1 0 1	3	1 1 1 1–1–2	–2	1	–2	0	1	0	–1	1	0	1
0 1 1 0 0 1	2	–1 1 1–1–1–1	–1	–2	1	0	0	1	1	–1	1	0

Figure 2. Network architecture for an example mapping.

Metasearch engines have emerged for just this reason. A metasearch engine submits the query term to several standard search engines, obtains the search results, throws away the redundant results, and combines and displays the rest in a consistent user interface.

But metasearch engines inherit the main drawbacks of limited precision and vulnerability to keyword spamming that plague the standard engines. Besides, they display their returns bunched in terms of the returns from the standard engines. So, irrelevant returns of Engine A appear above the more relevant returns of Engine B, solely because of the sequencing chosen.

In a recent study, we proposed a method of merging the returns using a classification method based on CC4.[12] Because CC4 learns instantaneously, it lets us classify a metasearch engine's responses in terms of different relevance values.

In the metasearch neural network, the relevant Web pages obtained from different standard search engines are more similar to each other than to irrelevant pages and vice versa. The similarity could be best determined if the complete contents of the Web pages were reviewed. But this would be too complex and time-consuming, so we built a system that uses the titles and the summaries alone. This system considers two Web pages as similar in content if there are more common keywords in their titles and contents.

The method begins by taking the top few Web pages and the last few Web pages from each search engine; it assumes that their classification—relevant or irrelevant—is already known. The neural network is built using keywords from the Web pages. Each keyword maps into a 0 or 1, so the length of the input vector equals the number of the keywords chosen. The output is 1 if the Web page comes from the top of the list and 0 if it comes from the bottom. The neural network can assign all the Web pages to either of these two classes or assign a relevancy value that is a fraction between 1 and 0.

This method's performance would improve if we used HTML keywords and description metatags as well as the Dublin Core metadata standard in the classification. Currently, only about a third of all home pages use any keywords or description tags, and less than a percent use the Dublin Core metadata standard. But a neural-network-based system that generates based on the full text information could be devised.

We expect that instantaneously trained neural networks will find increasing uses in engineering and business applications. As a model of working or short-term memory, it can help provide a closer linkage with the findings of psychologists and neuroscientists, demonstrating that AI techniques that combine quick generalization with rule-based processing are powerful and versatile.

Further research on the use of ITNNs in prediction and intelligent Web search is continuing in my group at LSU. Neural Technologies, LLC, based in Kansas City, Missouri, is developing commercial products based on this technology. These will include Solosearch.com, a new metasearch engine.

References

1. S. Kak, “On Training Feedforward Neural Networks,” *Pramana J. Physics*, Vol. 40, 1993, pp. 35–42.

2. S. Kak, “New Algorithms for Training Feedforward Neural Networks,” *Pattern Recognition Letters*, Vol. 15, 1994, pp. 295–298.

3. S. Kak and J. Pastor, “Neural Networks and Methods for Training Neural Networks,” US Patent No. 5,426,721, 20 June 1995.

4. P. Raina, “Comparison of Learning and Generalization Capabilities of the Kak and the Backpropagation Algorithms,” *Information Sciences*, Vol. 81, 1994, pp. 261–274.

5. M.S. Gazzaniga, *The Cognitive Neurosciences*, MIT Press, Cambridge, Mass., 1995.

6. A. Baddeley and S. Della Sala, “Working Memory and Executive Control,” *Phil. Trans. Royal Society of London*, Vol. 351, 1996, pp. 1397–1406.

7. S. Kak, “The Three Languages of the Brain: Quantum, Reorganizational, and Associative,” *Learning as Self-Organization*, K. Pribram and J. King, eds., Lawrence Erlbaum, Mahwah, N.J., 1996, pp. 185–219.

8. K.-W. Tang and S. Kak, “A New Corner Classification Approach to Neural Network Training,” *Circuits, Systems, and Signal Processing*, Vol. 17, 1998, pp. 459–469.

9. S. Kak, “On Generalization by Neural Networks,” *Information Sciences*, Vol. 111, 1998, pp. 293–302.

10. S. Lawrence and C.L. Giles, “Searching the World Wide Web,” *Science*, Vol. 280, 1998, pp. 98–100.

11. S. Lawrence and C.L. Giles, “Accessibility of Information on the Web,” *Nature*, Vol. 400, 1999, pp. 107–109.

12. B. Shu and S. Kak, “A Neural-Network Based Intelligent Metasearch Engine,” *Information Sciences*, Vol. 120, 1999, pp. 1–11.

Subhash Kak is a professor in the Department of Electrical and Computer Engineering at Louisiana State University in Baton Rouge. His research interests include quantum computing, neural networks, information technology, and the history of science. He obtained his PhD from the Indian Institute of Technology, Delhi. He was one of the first to do research on bringing together quantum mechanics and information. Most recently, he has worked on the theory and applications of instantaneously trained artificial neural networks. Contact him at the Dept. of Electrical and Computer Eng., LSU, Baton Rouge, LA 70803-5901; kak@ee.lsu.edu; http://www.ee.lsu.edu/kak.

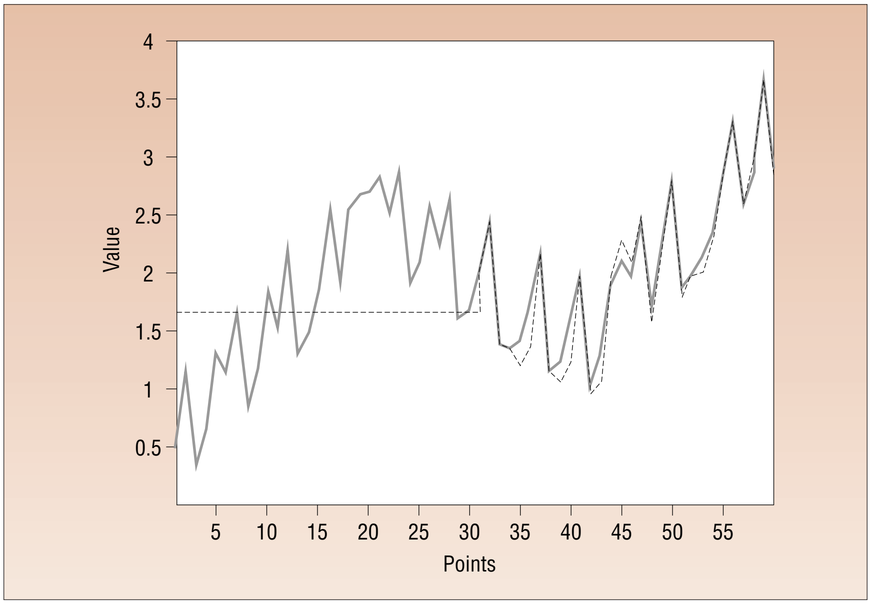


Figure 3. Prediction of an undulatory logistic map with a chaotic time series of which the last 60 points are shown and 30 points (broken line) predicted.

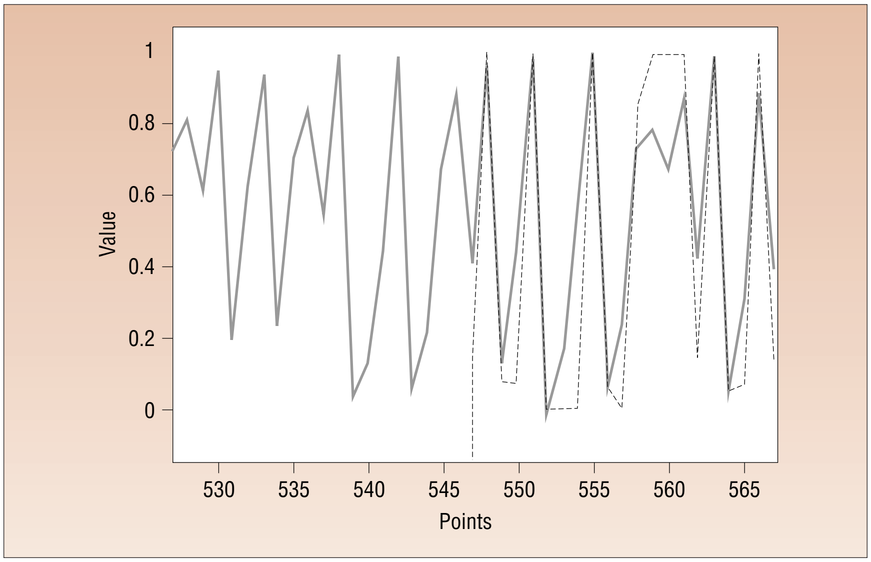


Figure 4. Prediction of a logistic map with a chaotic time series of length 567 for which the last 30 points are predicted (broken line).