In: Fara, P., and Patterson K. (Eds.),
Memory, MIT Press, Cambridge, MA

8 Memory and Neural Networks

TERRENCE J. SEJNOWSKI

The brain's operation depends on networks of nerve cells, called neurons, connected with each other by synapses. Scientists can now mimic some of the brain's behaviours with computer-based models of neural networks. One major domain of behaviour to which this approach has been applied is the question of how the brain acquires and maintains new information; that is, what we would call learning and memory. Neural networks employ various learning algorithms, which are recipes for how to change the network to store new information, and this chapter surveys learning algorithms that have been explored over the last decade. A few representative examples are presented here to illustrate the basic types of learning algorithms; the interested reader is encouraged to consult recent books listed in the section on Further reading, which present these algorithms in greater detail and provide a more complete survey.

It is important to keep in mind that the models of neural networks that are simulated in computers are far simpler than the highly complex and often messy neural systems that nature has devised. Although much simpler, neural network models capture some of the most basic features of the brain and may share some general properties that will allow us to understand better the operation of the more complex system. Neural network models are built from simple processing units that work together in parallel, each concerned with only a small piece of information. The representation of an object, for example, is typically represented as a pattern of activity over many of these processing units, and learning to recognize a new object occurs by changing the connection strengths or weights on the links between units. Long-term memories are therefore stored in the neural network, resulting in a close

relationship between how information is represented, processed, stored and recalled.

If there is no external supervision, learning in a neural network is said to be unsupervised; when there is an external 'teacher', the learning is supervised. If the teacher provides only a scalar feedback (a single number indicating how well the network has performed), it is called reinforcement learning. A distinction should also be made between a learning *mechanism* and a learning *algorithm*. For example, a Hebbian synapse (named after the Canadian neuropsychologist, Donald Hebb) is a learning *mechanism* that has been found in many parts of the brain, in which the strength of the connection across a neural synapse changes according to the correlation between activity in the presynaptic and post-synaptic neurons. In other words, if neurons fire together frequently, then the synaptic link between them is strengthened. Hebbian synapses are often thought to reflect a particular type of learning algorithm, but it is possible to use a Hebbian mechanism to implement any or all of supervised, reinforcement and unsupervised learning algorithms.

The time scale for learning can vary over a wide range, and different values of this range are suited to different types of learning. A fast learning rate is good for one-shot memorization, which is appropriate for remembering unique items or events that may only be presented once. Representational learning is slower, since many exemplars from each category need to be compared before the salient distinctions can be extracted that characterize the category. An example of the former is remembering a specific episode involving an elephant, one time when you were at the zoo. What you know about an elephant (what it looks like, where it comes from, how it differs from a rhinoceros or a giraffe), on the other hand, is representational knowledge acquired gradually. Reinforcement learning is typically slow, because the amount of information gathered on any one trial may be quite small and must be integrated over trials before the network can converge on the desired knowledge or behaviour.

The processing units used in most neural networks are simplified caricatures of neurons that lack many features of real neurons in the brain, such as their anatomical structure (e.g. dendrites that form synapses with nearby axons from other neurons, as in Figure 1, *left*). Two

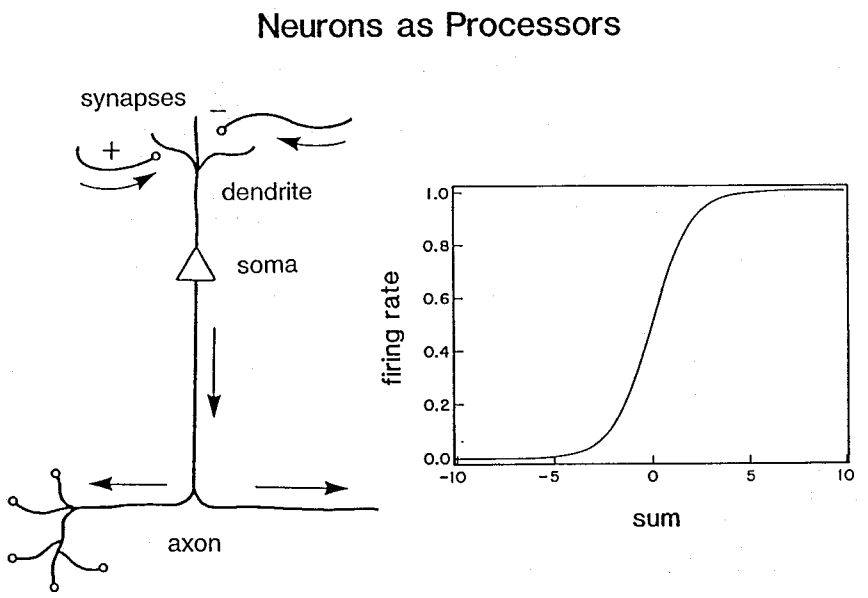

Figure 1 Model of a neuron that serves as a processing unit in a neural network model. *Left*: Schematic model of a processing unit receiving inputs from other processing units on the dendrites, and sending information out along the axon. Neurons are connected together through synapses that can be excitatory (positive weight) or inhibitory (negative weight). *Right*: Input–output function for a processing unit. The total synaptic input at a given moment in time is transformed by the sigmoidal function into an output firing rate.

essential features of real brain systems have, however, been retained. First, the links or connections between processing units in an artificial neural network are a direct analogy of synapses between neurons, whose weighted activity is often linearly summed. Secondly, the non-linear output function that communicates the integrated activity to other network units, typically through a threshold function (Figure 1, *right*), is also based on our understanding of the interaction of real neurons. These simplified models have the virtue of supporting rigorous mathematical analysis that is essential for achieving a deeper understanding of the performance of the network. Simple network models are also the starting point for more sophisticated models that more accurately reflect further properties of real neurons.

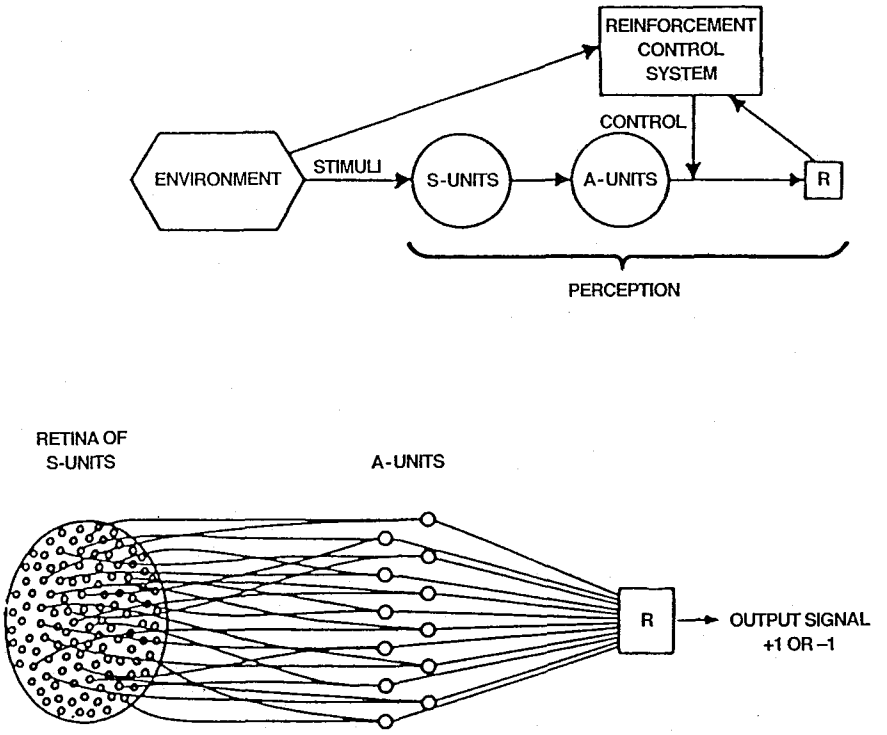

Figure 2 Most of the essential properties of modern multilayer 'perceptrons' can already be found in the earliest versions. *Top*: Learning control system used for training a perceptron. The environment provides sensory stimuli that impinge on the sensory receptors (S-units) and a teaching signal that is used to change the strengths of the connections to the single output unit (R). *Bottom*: There is a fixed set of connections between the sensory receptors and the association units (A-units) and a set of variable connections from the A-units to the output unit. The goal of the perceptron was to classify all sensory patterns into two classes, those that belong to a category, such as the numeral '2', and those that do not.

Supervised learning

A simple learning procedure exists for automatically determining the weights in a feedforward network (i.e. one in which activity flows from input to output without feedback loops) having a single layer of variable connection strengths between the input and output layers (Figure 2). It is an incremental learning procedure that requires a teacher to provide the

network with correct outputs for a set of input patterns; with each input pattern, the weights in the network are slightly altered to improve its performance. If a set of weights exists that can solve the problem, then a convergence theorem guarantees that such a set of weights will be found; this has been demonstrated for the simple two-layer feedforward networks of binary (two-valued) units known as 'perceptrons' (Figure 2), introduced by Frank Rosenblatt in his book on *Principles of Neurodynamics* (1959), and for networks with continuous-valued units as shown by Bernard Widrow and Ted Hoff who introduced the least-mean-squared (LMS) algorithm.

These learning procedures are error-correcting in the sense that only information about the discrepancy between the desired output provided by the teacher and the actual output given by the network is used to update the weights. A serious limitation of a feedforward network with one layer of modifiable weights is that only simple problems can be learned. Fortunately, learning algorithms have recently been discovered for more complex networks that can support non-linear mappings, like those commonly found in neural processing. In 1983, Geoffrey Hinton (a computer scientist and psychologist at University College London) and I generalized the perceptron learning algorithm to a multilayer network with feedback connections, called the Boltzmann machine. Shortly thereafter, David Rumelhart, a psychologist at Stanford University, together with Geoffrey Hinton and Ronald Williams, generalized the LMS learning rule to a multilayered feedforward architecture by the backpropagation of errors.

In the most typical case of a multilayered network, there is an extra layer of 'hidden' units, so called because they do not correspond directly to either input or output but rather learn a non-linear mapping between them; the network therefore has two sets of weights rather than one. A concrete example of a feedforward network with hidden units is illustrated in the next section. In error backpropagation, the error (discrepancy between the correct and actual response of the network) is first used to adjust the weights on connections from the hidden layer to the output layer. Next, the error contribution is backpropagated by the chain rule of differential calculus to the hidden layer, where it can be used to adjust the weights on connections from the input units. Although

backpropagation has not been demonstrated in the brain, it is perhaps the best-known network learning algorithm and has been used to solve many problems of practical and/or theoretical interest. Such networks are nearly impossible to design by explicit computer programming. By examining successful networks that have learned by backpropagation, we have discovered new ways of thinking about distributed representations, in which each unit or neuron represents a tiny part of the solution. This is illustrated by NETtalk, a network that was trained by Charles Rosenberg and me to pronounce English text in 1985.

Lessons from NETtalk

English orthography is characterized by a strikingly inconsistent relationship between the spellings and the pronunciations of words. For example, the *a* in almost all words ending in *-ave*, such as *brave* and *gave*, is a long vowel, but *have* deviates from this typical spelling-sound pattern. There are also words such as *read* or *bass* whose pronunciation can vary with their grammatical role or contextual meaning. NETtalk demonstrates that even a small single network can learn to capture most of the regularities in English pronunciation while at the same time absorbing many of the exceptions. The feedforward architecture of this network is shown in Figure 3. First, a sequence of seven letters in an English text was used to activate units on the input layer, then the states of the hidden units were determined by activity feeding forward from the input to the hidden layer, and, finally, the states of the phoneme units at the top were calculated by converging inputs from the hidden layer. The hidden units neither received direct input nor had direct output, but were used by the network to form internal representations that were appropriate for solving the problem of mapping letters to phonemes. The goal of the backpropagation learning algorithm was to search the space of all possible weights on connections in the network for a set that performed the mapping with a minimum of error.

The network was trained on a subset of the words from a 20 012 word dictionary, and new words were used to test its ability to generalize what it had learned. Learning proceeded one letter at a time. Early in the training period, the network learned to distinguish between vowels and consonants; however, it tended to translate all vowel letters to the same

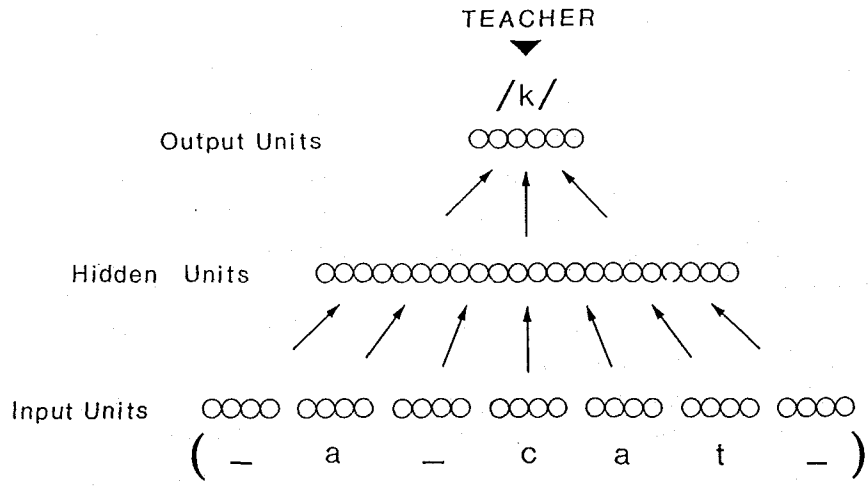

Figure 3 Schematic drawing of the NETtalk network architecture. A window of letters in an English text is fed to seven groups of input units. Information from these units is transformed by an intermediate layer of 80 'hidden' units, which project to the output layer corresponding to the sound of the centre letter. There was a total of 309 units and 18 629 weights in the network. In this example, the 'teacher' instructs the output that the correct phoneme for the letter 'c' in 'cat' is the 'hard c' sound /k/.

vowel sound and all consonant letters to the same consonant sound, which resulted in a babbling effect. A second stage occurred when word boundaries were recognized, and the output then resembled pseudo-words. After just a few passes through the dictionary, most words were intelligible.

By studying the patterns of activation amongst the hidden units, it was possible to understand some of the coding methods that the network had discovered. On average, for any given input, about 20% of the hidden units were highly activated while most of the remaining hidden units had little or no activation. Thus, the coding scheme could be described as neither a local representation, which would have activated only one or two hidden units specific to a given input, nor a 'holographic' representation in which all hidden units would have participated to some extent. When the hidden units were analysed with statistical techniques, it became apparent that the most important distinction was the complete separation of consonants and vowels. However, within each of these two

groups the clustering had a different pattern. For the vowels, the next most important variable was the individual vowel letter (*a*, *e*, *i*, etc.), whereas consonants were clustered according to the similarity of their sounds.

It was surprising to find that, when NETtalk was trained on new words, it displayed a phenomenon called the 'spacing effect' in human learning. In 1885, Hermann Ebbinghaus, the pioneering German psychologist, in his book *Memory: A Contribution to Experimental Psychology*, noted that 'with any considerable number of repetitions, a suitable distribution of them over a space of time is decidedly more advantageous than the massing of them at a single time'. The spacing effect has been found to apply to learning across a wide range of stimulus materials and tasks, semantic as well as perceptual/motor, and has even been found when the repetitions of information occur across different modalities (such as auditory and visual presentation) or across different languages. The ubiquity of this phenomenon suggests that spacing reflects something of central importance in learning and memory. However, despite over 100 years of research, there is no adequate, or at least simple, explanation for the spacing effect. In the case of NETtalk, the spacing effect arose because new words needed to be integrated into the representation of the previously learned words in a distributed and incremental way that would not interfere with previous learning, and this provides a hypothesis about the nature of the spacing effect in human learning.

NETtalk is an illustration in miniature of many aspects of learning. First, the network starts out with considerable 'innate' knowledge in the form of input and output representations that were chosen by the experimenters, but with no knowledge specific for English – the network could have been trained on any language with the same set of letters and phonemes. Secondly, the network acquired its competence through practice, went through several distinct stages, and finally reached a significant level of correct performance. Thirdly, and very important as a simulation of real memory in the brain, the learned information was distributed throughout the network such that no single unit or link was essential. If the trained network were subsequently damaged, in an analogue to brain damage from stroke or neurodegenerative conditions

like Alzheimer's disease, then the network behaved in a fashion that has come to be known as *graceful degradation*. That is, its responses became 'noisier' or more error prone but the network's performance did not simply collapse; in particular, correct responses to specific inputs (words) were not suddenly lost. Finally, the effect of temporal ordering and spacing during learning of new information was remarkably similar to that in humans.

Despite these similarities to human learning and memory, NETtalk is too simple to serve as a good model for the acquisition of reading skills in humans. The network attempts to accomplish in one phase what occurs in two major phases of human development. Children learn to talk first; only after representations for spoken words and their meanings are well developed does the child learn to read. It is also possible that, in addition to our ability to use context-dependent letter-to-sound correspondences, we have access to articulatory representations for whole words, but there are no whole-word level representations in the network. It is perhaps surprising that the network was capable of reaching a significant level of performance using a window of only seven letters. Despite these limitations, we learned from NETtalk and many similar examples that a relatively simple learning algorithm is capable of acquiring much more complex behaviour than we had previously imagined. This gave us confidence that other learning algorithms could be found that were more biologically plausible and even more powerful. The algorithms discussed below describe several staging posts along the way towards this goal.

Reinforcement learning

The external teacher in a supervised learning algorithm should not be taken too literally, since one brain area could serve as teacher and provide the information needed to train another brain area. One mechanism for this might be to have a sensory system 'teach' the motor system to mimic a sensory input. For example, when songbirds learn to sing, there is an auditory phase during which the young bird listens to and becomes familiar with the tutor song of an adult, followed by a sensorimotor phase of successive approximation and improvement leading to adult song. It is believed that a 'template' of the tutor song is formed in the auditory system during the first phase and that, during the sub-

sequent sensorimotor learning, the developing song of the bird is compared to the stored template, and changes are made to the motor system to improve the song.

There are specialized circuits in the bird's brain that are used to process birdsong. An 'anterior forebrain pathway' is needed for song learning but is not required for adult performance after learning is completed. Thus, in this form of mimicry, the 'teacher' may be a sensory representation stored in one part of the brain that is used as a reference to train the motor system through incremental learning. Recently, Kenji Doya and I developed a model of birdsong learning that uses a form of supervised learning called reinforcement learning. First, small changes were made to the connection strengths that control the song production. The degree of match between the song produced and the stored auditory template was then used to determine whether performance had improved, and whether to maintain or further refine the changes made to the connection strengths.

Often, only a crude feedback signal is available from the external world after a sequence of behaviours. In classical conditioning, a well-studied form of reinforcement learning that is found in a wide range of species, an animal may have a particular sequence of behaviours strengthened by finding a food reward, or weakened if the behaviour results in an aversive experience. In psychology, conditioning has been studied mainly through observing the behaviour of animals such as rats or birds, as summarized by Nick Mackintosh of Cambridge University in his book *The Psychology of Animal Learning.* Recently, however, it has been possible to follow the sensory and motor pathways inside the brain that are responsible for this form of learning. Classical conditioning is regulated by diffuse ascending neurotransmitter systems, arising from a relatively small number of neurons in the brainstem, that project throughout large regions of the forebrain. Similar diffuse neurotransmitter systems are also found in invertebrate animals, as illustrated next for the bee.

Bee foraging and temporal difference learning

Bees are among the most intelligent insects and survive as a colony by foraging for nectar in plants that may be at some distance from the hive. In particular, Randolf Menzel at the Free University in Berlin has shown

that honeybees can learn to associate a sucrose reward with an odour using proboscis extension as a behavioural assay for learning. The presence of sugar in nectar will automatically produce a proboscis extension in the bee. When paired with sucrose, an odour that previously had no effect on proboscis extension reliably elicits this response; this is straightforward classical conditioning. Martin Hammer, also in Berlin, using a technique of intracellular recording, has identified a single neuron in honeybees (called VUMmx1) that may mediate this simple form of reinforcement learning. Hammer's new discovery was that the nerve-cell firing of VUMmx1 by injection of electrical current can substitute for sucrose in a conditioning experiment: after pairing between VUMmx1 activation and an odour, the odour by itself elicits proboscis extension. Thus the activity of VUMmx1 can substitute for a reinforcing stimulus like nectar. In fact, direct application of the neurotransmitter released by VUMmx1 can also substitute for a rewarding stimulus.

Together with Read Montague, at the Baylor College of Medicine in Houston, and Peter Dayan, at University College London, I recently modelled the foraging behaviour of bees using a reinforcement learning algorithm (Figure 4). The model is based on the principle of prediction by temporal differences introduced by Richard Sutton, a psychologist, and Andrew Barto, a computer scientist. There is a central role in the model for VUMmx1, which is responsible for teaching the bee to predict the future reward consequent on sensory stimuli. As indicated above, if an odour temporally precedes the delivery of nectar, then, through a predictive form of the Hebbian learning algorithm, the inputs to VUMmx1 are strengthened. This learning algorithm is Hebbian because it depends on the conjunction of pre- and postsynaptic activity; however, the postsynaptic activity is not the summed synaptic input but the *prediction* error, calculated as the difference between the actual unconditioned reward stimulus (nectar) and the amount of reward expected on the basis of changing predictions about the future. This model accounts for data obtained from a large number of behavioural experiments on bee learning, including risk aversion behaviour.

Wolfram Schultz, a neurophysiologist at the University of Fribourg, Switzerland, has recorded from single neurons in primates and found firing patterns in response to rewarding stimuli that are analogous to

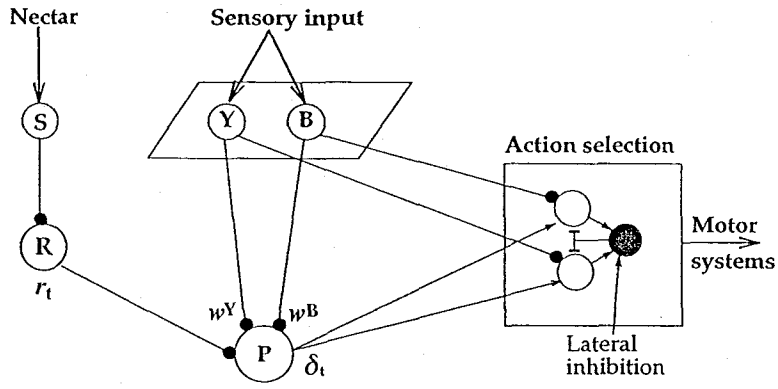

Figure 4 Neural architecture for a model of bee foraging. Unit P learns to makes predictions about future expected reinforcement. Sensory input (units B and Y) are activated by blue and yellow flowers. S is activated while the bee sips the nectar. Learning takes place at the weights w after comparing expected reward with the actual reward r_t originating from unit R. During foraging, the sensory impression of a flower evokes a predicted reward δ_t from P, which is used by the action selection network to decide whether to approach or to avoid the flower. In the decision network, lateral inhibition is used to ensure that only one action is selected between the two possibilities, a form of winner-take-all circuit.

those observed in VUMmx1 in bees. Early in learning, these neurons fired reliably to reward, but later in learning they no longer responded directly to the reward; instead they were activated by the earlier sensory stimuli that reliably predicted the reward. This is exactly what would be produced by a predictive Hebbian learning rule that modifies the inputs to these neurons, and underlies the model's explanation for phenomena of classical conditioning. The output from such a cell may therefore carry information about errors in the prediction of future rewards consequent on specific stimuli. This interpretation was supported in an experiment in which the expected reward was withheld: neurons that no longer responded directly to the reward showed a *decrease* in activity when the reward should have occurred. The predictive principle of temporal difference learning provides computational explanations for many otherwise puzzling facts about learning in the brain.

Predictive Hebbian learning may be a universal mechanism that is important for orienting animals to stimuli that lead to future reward.

The brain is probably organized to make predictions about the importance of sensory stimuli for survival in an uncertain world, and to use these predictions as a basis for appropriate action. In the example given here, only a single action was trained; but temporal difference learning can be used to learn a sequence of actions leading to a delayed reward. For example, Gerald Tesauro at the IBM T. J. Watson Laboratories in Yorktown Heights, New York, has used temporal difference learning to train a network to play backgammon at a level that is competitive with the best human players. The network improved from random moves to master level by playing against itself, and the only reward that it received was the outcome of each game – won or lost. This is a remarkable achievement that illustrates the potential for temporal difference learning to master complex tasks.

Unsupervised learning

When there is no explicit teacher, there is still much to learn. As the former New York Yankee baseball player Yogi Berra once said, 'You can observe a lot just by watching'. The world is information rich, and the problem of the learner is to extract from the constant stream of sensory information a summary of the regularities in this information that will help to improve the recognition of familiar items or events and the detection of new ones. One good strategy for unsupervised learning is to try to predict future sensory states as a basis for developing predictive representations. This type of representational learning is not guided by a specific task, but may none the less be biased in directions that are either innate or influenced by experience. People who are amnesic following damage to the hippocampal formation (see Wilson, Chapter 6) are profoundly impaired on explicit learning, but have significantly preserved ability to learn tasks that might be based on implicit memory at an early representational level. The development of the nervous system also offers important clues to unsupervised learning, since many of the same mechanisms that organize neural interactions during development are also used, in modified forms, in adult brains. In particular, there are several diffuse systems of neuromodulators ascending from the brainstem that support neural plasticity during development.

An example of an unsupervised learning algorithm that can be

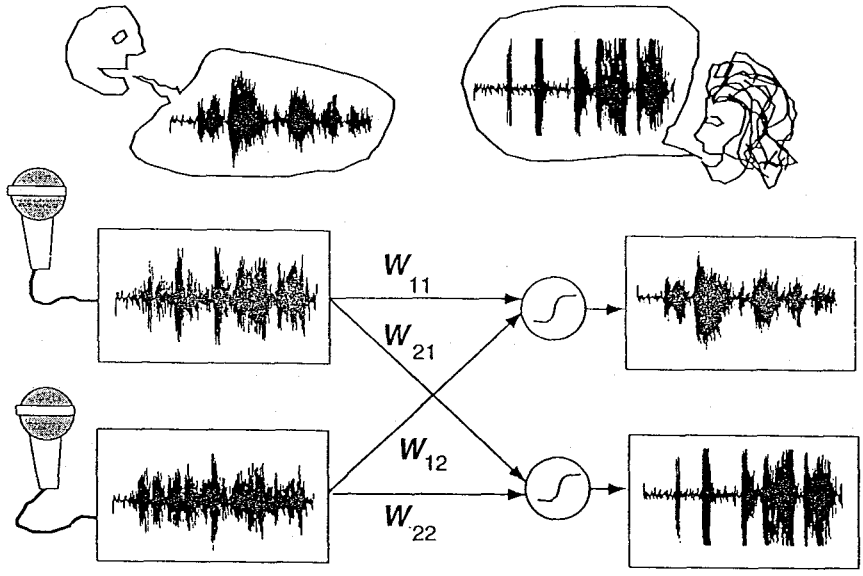

Figure 5 Illustration of the blind separation problem. Two microphones record signals from two people speaking simultaneously, but with different amplitudes at each microphone. The task is to find a set of weights w_{ij} that invert the linear mixtures on the inputs to produce outputs that are the original independent signals from each speaker. No specific information is given about the signals. The independent component analysis algorithm can solve this problem.

implemented by a neural network is a form of statistical analysis closely related to Hebbian synaptic plasticity known as principal component analysis (PCA). A new class of unsupervised algorithms, called independent component analysis (ICA), can extract higher-order information, and has been the subject of intense investigation by a group of researchers worldwide. Anthony Bell and I have proposed a framework for 'blind' source segregation, based on an idea introduced by Simon Laughlin at Cambridge University in the context of visual information transfer, that can separate signals into their independent components, as illustrated in Figure 5. The ICA learning algorithm we derived has proven effective in analysing many types of data, including natural images, natural sounds, and electroencephalographic data in which the independent sources correspond to electrical activity in different regions of the

brain. One result from analysing visual images in this framework will be summarized here.

Edge detectors in visual cortex

The classic experiments of David Hubel and Torsten Wiesel on neurons in the visual cortex showed that many cells preferred specific features, such as edges and lines of particular orientation in a small part of the visual field, different for each neuron, called its receptive field. This led Horace Barlow at Cambridge University to search for coding principles that would predict the formation of such feature detectors. Barlow suggested that edges are suspicious coincidences that arise at the end of a redundancy reduction process, the purpose of which is to make the activation of each feature detector as statistically *independent* of all others as possible.

Anthony Bell and I recently applied our ICA learning algorithm to images of natural scenes including trees, foliage and bushes. The training set consisted of 17 595 input patterns consisting of 12 pixel × 12 pixel patches from the images. The filters and basis functions resulting from training on natural scenes are displayed in Figure 6. The general result is that the filters found by ICA learning are localized and mostly oriented. These filters produce output distributions of very high sparseness; that is, when an image is convolved with all of the ICA filters, only a few of them are highly activated and most of them have nearly zero output. The general properties of the ICA filters, which were designed to be as statistically independent as possible, are similar to those of oriented simple cells in primary visual cortex, supporting Barlow's intuition regarding the independence of cortical feature-detector cells.

ICA is a powerful unsupervised learning algorithm, but it is limited to one layer of processing units, and the output is a linear transformation of the inputs. The architecture of cerebral cortex is organized in a hierarchy of processing layers, starting with the primary sensory areas and ascending to more complex and eventually multimodal representations. At the top of the hierarchy, neural activity from all the sensory modalities (vision, hearing, touch, balance, etc.) funnels into the hippocampus, a part of the brain thought to be necessary for creating long-term mem-

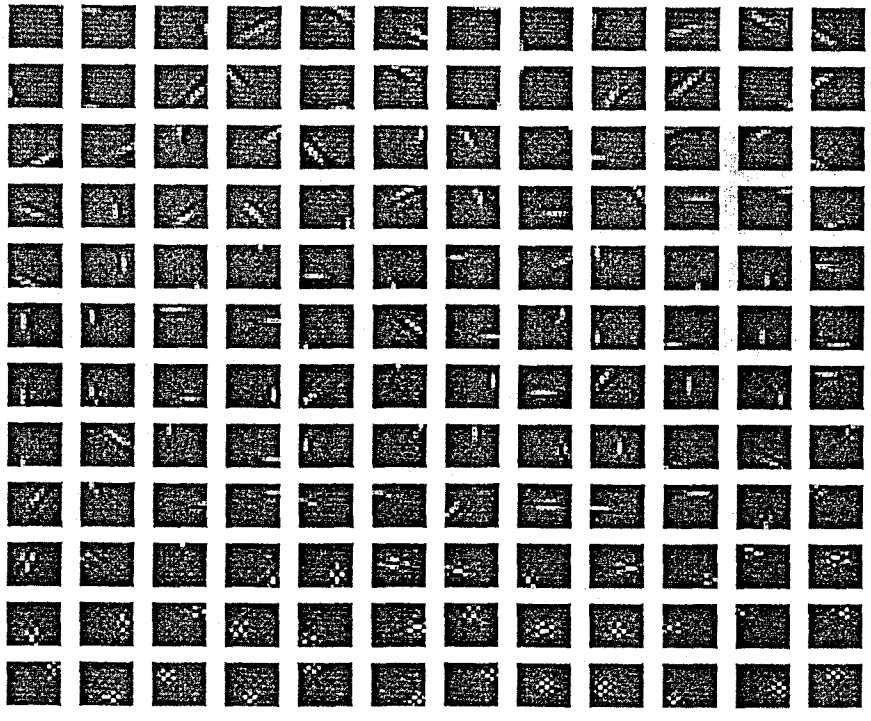

Figure 6 The matrix of 144 filters obtained by independent component analysis training. Each filter is a 12 × 12 patch that produces an output depending on how well the excitatory regions of the patch (white) match the high intensity regions of the image and avoid the inhibitory parts of the filter (black). Most of the filters have a preferred orientation, which resemble the response properties of simple cells in visual cortex, though some are localized chequerboard patterns that produce responses in two perpendicular directions. Only a few of these filters will be highly activated by any given image, which produces a sparse encoding of the image by the filters. These filters form a complete set in the sense that any patch of image can be exactly reconstructed knowing only the activity levels of each filter.

ories of specific objects and events. Damage to the hippocampus (which is common, for example, in Alzheimer's disease) interferes with new and recent learning; but if the hippocampal damage occurs some substantial time after a learned experience, then there is little or no significant disruption of the memory. This implies that, although the hippocampus mediates consolidation of memory, the neural patterns corresponding to

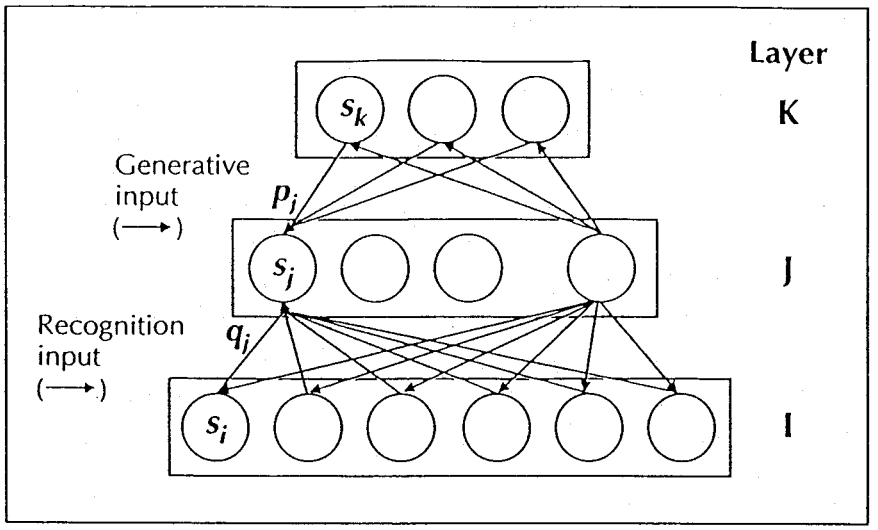

Figure 7 Wake–sleep network model. Sensory inputs (s_i) in layer I originate in the bottom layer and feedforward connections (q_j) carry this information through a layer J of hidden units (s_j) to the output units (s_k) top layer K in the recognition or wake phase. The feedback connections (p_i) from top to bottom provide a generative input to the bottom layer during the sleep phase.

long-lasting memories are stored in other brain regions. Unsupervised learning algorithms are needed that can create a hierarchy of sensory representations similar to those found in the brain.

Wake–sleep learning

Geoffrey Hinton, Peter Dayan and their colleagues have recently presented a powerful new theoretical framework for creating efficient memory representations in hierarchical feedforward neural networks (Figure 7). The framework suggests computational explanations for why we may need to sleep and why the memory consolidation process can take so long. In this model, when the external inputs to the network have been suppressed (just as external stimulus inputs to the brain are suppressed when we are asleep), then the feedback connections generate patterns on the input layers of the network that correspond to representations at the higher level. During this generative 'sleep' stage, the strengths of the feedforward synaptic connections are altered. Con-

versely, during the 'awake' state, the sensory inputs drive the feedforward system; in this phase, weights on the feedback connections can be modified. This two-phase process produces an internal, hierarchical representation of an experience that is economical and able to generate typical input patterns. The learning mechanisms are biologically possible since, unlike supervised learning algorithms such as backpropagation that require a detailed teacher, wake–sleep learning depends only on locally available signals and there is no external teacher.

The wake–sleep learning algorithm attempts to capture the regularities of sensory inputs with an internal code that is componential: it is capable of representing not just whole individual items but components and features that are common to many objects. Because these statistical components are not apparent without comparing many sensory experiences, the training process is gradual, in the sense that only small changes are made during any one wake–sleep cycle. From this point of view, the hippocampus could serve as the highest level of representation. Through a generative process involving feedback connections from the hippocampus to the cortex, feedforward connections in the cortex that are responsible for the primary representations of sensory information are modified. The forward recognition network and the feedback generative networks are the mutual statistical inverses of each other.

The wake–sleep model makes testable predictions for the function of feedback connections and the times at which they should be modifiable. It has proved difficult to assign any function for cortical feedback connections, and the wake–sleep model explains why this might be. Experiments designed to test the effects of these connections have been performed during the awake state; but according to the model, they should drive physiological activity only during sleep. Also, the model predicts that the feedback and feedforward projections should be modifiable at different times: feedforward connections during sleep, and feedback connections during the awake state. The types of experiment that are now possible using multielectrode recordings should allow these predictions to be tested directly.

Although the links between sleep and memory are intriguing, the different phases of wake–sleep learning may not correspond in such a straightforward fashion to awake or sleeping states. There are several

different phases of sleep, one of them being dream, or rapid eye-movement (REM), sleep, so it is not clear which sleep state should be associated with the 'sleep' phase of the wake–sleep learning algorithm. It is also possible that both phases alternate even during states of consciousness, with the sleep phase of the algorithm corresponding to lapses of attention and daydreaming. None the less, it is worthwhile to explore the possibility that the same neural network may have quite different properties in different states, including differences in the forms of neural plasticity. The state of the cortex is influenced by neuromodulators, which are known to affect neural plasticity, so mechanisms are already known that could implement these state changes.

Biological and mathematical foundations

Many different types of memory systems have been found in different parts of the brain. Even the retina can adapt to light intensities over a wide range of temporal and spatial scales, a form of short-term sensory memory. Therefore we should expect not a single, universal model of memory but rather a diversity of memory models, and this is reflected in a diversity of learning algorithms. In this chapter, only a few, representative examples of neural network models of learning and memory were presented to illustrate this richness.

The focus has been on learning algorithms in which the parameters that represent long-term changes in memory are the strengths of the synapses. There are other variables and parameters in network models that may also be important, including the number of units (neurons) in the model, the short-term dynamics of connections (synapses), and the properties of the non-linear input–output function. For example, it is possible for a dynamical network model with feedback connections to exhibit long-term memory without changing any of the weights on connections, by changing only the time courses of the synaptic activities.

A memory system can be considered a form of information compression for an ensemble of sensory inputs or sensorimotor transformations. Thus each memory system in the brain can be analysed to discover how well it is able to perform information compression, and what types of information are most efficiently represented. A mathematical principle, called Minimum Description Length (MDL),

has been helpful in developing new learning algorithms and provides a useful framework for organizing existing algorithms. The MDL framework may also help us to understand changes that occur in the developing brain.

One of the most exciting aspects of neural network research is the extent to which it crosses disciplines, including neuroscientists interested in understanding how brain systems are organized, cognitive scientists interested in the properties of minds, and engineers who want to mimic some of the brain's impressive capabilities. We are just beginning to understand learning and memory in the simplest neural network systems, and to appreciate the complexity of the many memory systems found in brains.

FURTHER READING

Arbib, M. A., *The Handbook of Brain Theory and Neural Networks*, Cambridge, MA: MIT Press, 1995.

Ballard, D. H., *An Introduction to Natural Computation*, Cambridge, M.A.: MIT Press, 1997.

Bishop, C. M., *Neural Networks for Pattern Recognition*, New York: Oxford University Press, 1995.

Hammer, M. and Menzel, R., 'Learning and memory in the honeybee', *Journal of Neuroscience* **15** (1995), 1617–1630.

Hebb, D. O., *The Organization of Behavior*, New York: Wiley, 1949.

Mackintosh, N. J., *Conditioning and Associative Learning*, Oxford: Oxford University Press, 1983.

Montague, P. R. and Sejnowski, T., 'The predictive brain: temporal coincidence and temporal order in synaptic learning mechanisms', *Learning and Memory* **1** (1994), 1–33.

Quartz, S. and Sejnowski, T. J., 'The neural basis of cognitive development: a constructivist manifesto', *Behavioral and Brain Sciences*, **20** (1997), 537–96.

Tesauro, G., 'Temporal difference learning and TD-Gammon', *Communications of the ACM* **38** (1995), 58–68.