Complex Systems **1** (1987) 145–168

Parallel Networks that Learn to Pronounce English Text

Terrence J. Sejnowski
Department of Biophysics, The Johns Hopkins University, Baltimore, MD 21218, USA

Charles R. Rosenberg
Cognitive Science Laboratory, Princeton University, Princeton, NJ 08542, USA

Abstract. This paper describes NETtalk, a class of massively-parallel network systems that learn to convert English text to speech. The memory representations for pronunciations are learned by practice and are shared among many processing units. The performance of NETtalk has some similarities with observed human performance. *(i)* The learning follows a power law. *(ii)* The more words the network learns, the better it is at generalizing and correctly pronouncing new words, *(iii)* The performance of the network degrades very slowly as connections in the network are damaged: no single link or processing unit is essential. *(iv)* Relearning after damage is much faster than learning during the original training. *(v)* Distributed or spaced practice is more effective for long-term retention than massed practice.

Network models can be constructed that have the same performance and learning characteristics on a particular task, but differ completely at the levels of synaptic strengths and single-unit responses. However, hierarchical clustering techniques applied to NETtalk reveal that these different networks have similar internal representations of letter-to-sound correspondences within groups of processing units. This suggests that invariant internal representations may be found in assemblies of neurons intermediate in size between highly localized and completely distributed representations.

1. Introduction

Expert performance is characterized by speed and effortlessness, but this fluency requires long hours of effortful practice. We are all experts at reading and communicating with language. We forget how long it took to acquire these skills because we are now so good at them and we continue to practice every day. As performance on a difficult task becomes more

automatic, it also becomes more inaccessible to conscious scrutiny. The acquisition of skilled performance by practice is more difficult to study and is not as well understood as memory for specific facts [4,55,78].

The problem of pronouncing written English text illustrates many of the features of skill acquisition and expert performance. In reading aloud, letters and words are first recognized by the visual system from images on the retina. Several words can be processed in one fixation suggesting that a significant amount of parallel processing is involved. At some point in the central nervous system the information encoded visually is transformed into articulatory information about how to produce the correct speech sounds. Finally, intricate patterns of activity occur in the motoneurons which innervate muscles in the larynx and mouth, and sounds are produced. The key step that we are concerned with in this paper is the transformation from the highest sensory representations of the letters to the earliest articulatory representations of the phonemes.

English pronunciation has been extensively studied by linguists and much is known about the correspondences between letters and the elementary speech sounds of English, called phonemes [83]. English is a particularly difficult language to master because of its irregular spelling. For example, the "a" in almost all words ending in "ave", such as "brave" and "gave", is a long vowel, but not in "have", and there are some words such as "read" that can vary in pronunciation with their grammatical role. The problem of reconciling rules and exceptions in converting text to speech shares some characteristics with difficult problems in artificial intelligence that have traditionally been approached with rule-based knowledge representations, such as natural language translation [27].

Another approach to knowledge representation which has recently become popular uses patterns of activity in a large network of simple processing units [22,30,56,42,70,35,36,12,51,19,46,5,82,41,7,85,13,67,50]. This "connectionist" approach emphasizes the importance of the connections between the processing units in solving problems rather than the complexity of processing at the nodes.

The network level of analysis is intermediate between the cognitive and neural levels [11]. Network models are constrained by the general style of processing found in the nervous system [71]. The processing units in a network model share some of the properties of real neurons, but they need not be identified with processing at the level of single neurons. For example, a processing unit might be identified with a group of neurons, such as a column of neurons [14,54,37]. Also, those aspects of performance that depend on the details of input and output data representations in the nervous system may not be captured with the present generation of network models.

A connectionist network is "programmed" by specifying the architectural arrangement of connections between the processing units and the strength of each connection. Recent advances in learning procedures for such networks have been applied to small abstract problems [73,66] and

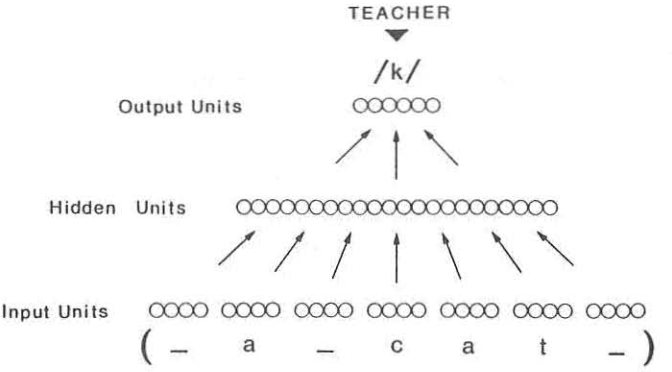

Figure 1: Schematic drawing of the NETtalk network architecture. A window of letters in an English text is fed to an array of 203 input units. Information from these units is transformed by an intermediate layer of 80 "hidden" units to produce patterns of activity in 26 output units. The connections in the network are specified by a total of 18629 weight parameters (including a variable threshold for each unit).

more difficult problems such as forming the past tense of English verbs [68].

In this paper we describe a network that learns to pronounce English text. The system, which we call NETtalk, demonstrates that even a small network can capture a significant fraction of the regularities in English pronunciation as well as absorb many of the irregularities. In commercial text-to-speech systems, such as DECtalk [15], a look-up table (of about a million bits) is used to store the phonetic transcription of common and irregular words, and phonological rules are applied to words that are not in this table [3,40]. The result is a string of phonemes that can then be converted to sounds with digital speech synthesis. NETtalk is designed to perform the task of converting strings of letters to strings of phonemes. Earlier work on NETtalk was described in [74].

2. Network Architecture

Figure 1 shows the schematic arrangement of the NETtalk system. Three layers of processing units are used. Text is fed to units in the input layer. Each of these input units has connections with various strengths to units in an intermediate "hidden" layer. The units in the hidden layer are in turn connected to units in an output layer, whose values determine the output phoneme.

The processing units in sucessive layers of the network are connected by weighted arcs. The output of each processing unit is a nonlinear function

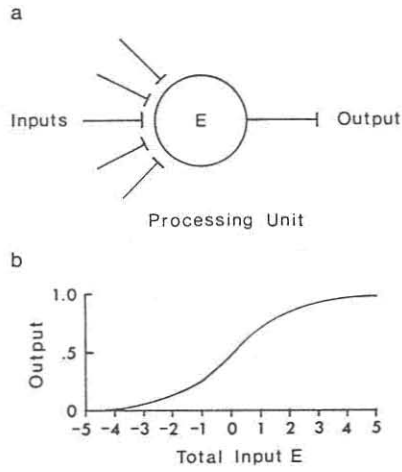

Figure 2: (a) Schematic form of a processing unit receiving inputs from other processing units. (b) The output $P(E)$ of a processing unit as a function of the sum E of its inputs.

of the sum of its inputs, as shown in Figure 2. The output function has a sigmoid shape: it is zero if the input is very negative, then increases monotonically, approaching the value one for large positive inputs. This form roughly approximates the firing rate of a neuron as a function of its integrated input: if the input is below threshold there is no output; the firing rate increases with input, and saturates at a maximum firing rate. The behavior of the network does not depend critically on the details of the sigmoid function, but the explicit one used here is given by

$$s_i = P(E_i) = \frac{1}{1 + e^{-E_i}} \tag{2.1}$$

where s_i is the output of the ith unit. E_i is the total input

$$E_i = \sum_j w_{ij} s_j \tag{2.2}$$

where w_{ij} is the weight from the jth to the ith unit. The weights can have positive or negative real values, representing an excitatory or inhibitory influence.

In addition to the weights connecting them, each unit also has a threshold which can also vary. To make the notation uniform, the threshold was implemented as an ordinary weight from a special unit, called the true unit, that always had an output value of 1. This fixed bias acts like a threshold whose value is the negative of the weight.

Learning algorithm. Learning algorithms are automated procedures that allow networks to improve their performance through practice [63,87,2,75].

Supervised learning algorithms for networks with "hidden units" between the input and output layers have been introduced for Boltzmann machines [31,1,73,59,76], and for feed-forward networks [66,44,57]. These algorithms require a "local teacher" to provide feedback information about the performance of the network. For each input, the teacher must provide the network with the correct value of each unit on the output layer. Human learning is often imitative rather than instructive, so the teacher can be an internal model of the desired behavior rather than an external source of correction. Evidence has been found for error-correction in animal learning and human category learning [60,79,25,80,?]. Changes in the strengths of synapses have been experimentally observed in the mammalian nervous system that could support error-correction learning [28,49,61,39]. The network model studied here should be considered only a small part of a larger system that makes decisions based on the output of the network and compares its performance with a desired goal.

We have applied both the Boltzmann and the back-propagation learning algorithms to the problem of converting text to speech, but only results using back-propagation will be presented here. The back-propagation learning algorithm [66] is an error-correcting learning procedure that generalizes the Widrow-Hoff algorithm [87] to multilayered feedforward networks [23]. A superscript will be used to denote the layer for each unit, so that $s_i^{(n)}$ is the ith unit on the nth layer. The final, output layer is designated the Nth layer.

The first step is to compute the output of the network for a given input. All the units on successive layers are updated. There may be direct connections between the input layer and the output layer as well as through the hidden units. The goal of the learning procedure is to minimize the average squared error between the values of the output units and the correct pattern, s_i^*, provided by a teacher:

$$Error = \sum_{i=1}^{J} (s_i^* - s_i^{(N)})^2 \tag{2.3}$$

where J is the number of units in the output layer. This is accomplished by first computing the error gradient on the output layer:

$$\delta_i^{(N)} = (s_i^* - s_i^{(N)}) P'(E_i^{(N)}) \tag{2.4}$$

and then propagating it backwards through the network, layer by layer:

$$\delta_i^{(n)} = \sum_j \delta_j^{(n+1)} w_{ji}^{(n)} P'(E_i^{(n)}) \tag{2.5}$$

where $P'(E_i)$ is the first derivative of the function $P(E_i)$ in Figure 2(b).

These gradients are the directions that each weights should be altered to reduce the error for a particular item. To reduce the average error for all the input patterns, these gradients must be averaged over all the training

patterns before updating the weights. In practice, it is sufficient to average over several inputs before updating the weights. Another method is to compute a running average of the gradient with an exponentially decaying filter:

$$\Delta w_{ij}^{(n)}(u+1) = \alpha \Delta w_{ij}^{(n)}(u) + (1-\alpha)\delta_i^{(n+1)} s_j^{(n)} \tag{2.6}$$

where α is a smoothing parameter (typically 0.9) and u is the number of input patterns presented. The smoothed weight gradients $\Delta w_{ij}^{(n)}(u)$ can then be used to update the weights:

$$w_{ij}^{(n)}(t+1) = w_{ij}^{(n)}(t) + \epsilon \Delta w_{ij}^{(n)} \tag{2.7}$$

where the t is the number of weight updates and ϵ is the learning rate (typically 1.0). The error signal was back-propagated only when the difference between the actual and desired values of the outputs were greater than a margin of 0.1. This ensured that the network did not overlearn on inputs that it was already getting correct. This learning algorithm can be generalized to networks with feedback connections and multiplicative connection [66], but these extension were were not used in this study.

The definitions of the learning parameters here are somewhat different from those in [66]. In the original algorithm ϵ is used rather than $(1-\alpha)$ in Equation 6. Our parameter α is used to smooth the gradient in a way that is independent of the learning rate, ϵ, which only appears in the weight update Equation 7. Our averaging procedure also makes it unnecessary to scale the learning rate by the number of presentations per weight update.

The back-propagation learning algorithm has been applied to several problems, including knowledge representation in semantic networks [29,65], bandwidth compression by dimensionality reduction [69,89], speech recognition [17,86], computing the shape of an object from its shaded image [45] and backgammon [81]. In the next section a detailed description will be given of how back-propagation was applied to the problem of converting English text to speech.

Representations of letters and phonemes. The standard network had seven groups of units in the input layer, and one group of units in each of the other two layers. Each input group encoded one letter of the input text, so that strings of seven letters are presented to the input units at any one time. The desired output of the network is the correct phoneme, associated with the center, or fourth, letter of this seven letter "window". The other six letters (three on either side of the center letter) provided a partial context for this decision. The text was stepped through the window letter-by-letter. At each step, the network computed a phoneme, and after each word the weights were adjusted according to how closely the computed pronunciation matched the correct one.

We chose a window with seven letters for two reasons. First, [48] have shown that a significant amount of the information needed to correctly pronounce a letter is contributed by the nearby letters (Figure 3). Secondly, we were limited by our computational resources to exploring small networks

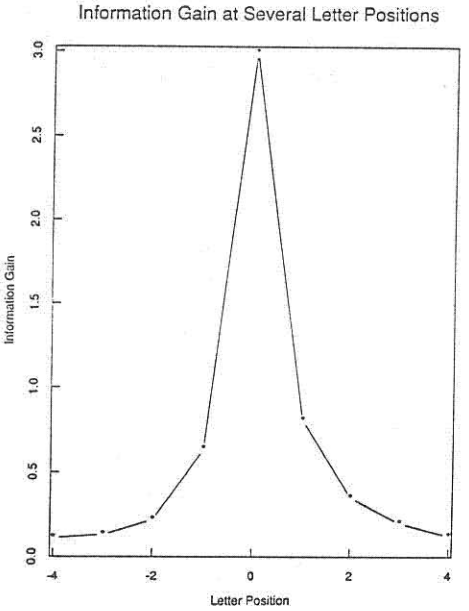

Figure 3: Mutual information provided by neighboring letters and the correct pronunciation of the center letter as a function of distance from the center letter. (Data from [48]).

and it proved possible to train a network with a seven letter window in a few days. The limited size of the window also meant that some important nonlocal information about pronunciation and stress could not be properly taken into account by our model [10]. The main goal of our model was to explore the basic principles of distributed information coding in a real-world domain rather than achieve perfect performance.

The letters and phonemes were represented in different ways. The letters were represented locally within each group by 29 dedicated units, one for each letter of the alphabet, plus an additional 3 units to encode punctuation and word boundaries. Only one unit in each input group was active for a given input. The phonemes, in contrast, were represented in terms of 21 articulatory features, such as point of articulation, voicing, vowel height, and so on, as summarized in the Appendix. Five additional units encoded stress and syllable boundaries, making a total of 26 output units. This was a distributed representation since each output unit participates in the encoding of several phonemes [29].

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 mapping problem of letters to phonemes. The goal of the learning algorithm was to search effectively the space of all possible weights for a network that performed the mapping.

Learning. Two texts were used to train the network: phonetic transcriptions from informal, continuous speech of a child [9] and *Miriam Webster's Pocket Dictionary*. The corresponding letters and phonemes were aligned and a special symbol for continuation, "-", was inserted whenever a letter was silent or part of a graphemic letter combination, as in the conversion from the string of letters "phone" to the string of phonemes /f-on-/ (see Appendix). Two procedures were used to move the text through the window of 7 input groups. For the corpus of informal, continuous speech the text was moved through in order with word boundary symbols between the words. Several words or word fragments could be within the window at the same time. For the dictionary, the words were placed in random order and were moved through the window individually.

The weights were incrementally adjusted during the training according to the discrepancy between the desired and actual values of the output units. For each phoneme, this error was "back-propagated" from the output to the input layer using the learning algorithm introduced by [66] and described above. Each weight in the network was adjusted after every word to minimize its contribution to the total mean squared error between the desired and actual outputs. The weights in the network were always initialized to small random values uniformly distributed between -0.3 and 0.3; this was necessary to differentiate the hidden units.

A simulator was written in the C programming language for configuring a network with arbitrary connectivity, training it on a corpus and collecting statistics on its performance. A network of 10,000 weights had a throughput during learning of about 2 letters/sec on a VAX 11/780 FPA. After every presentation of an input, the inner product of the output vector was computed with the codes for each of the phonemes. The phoneme that made the smallest angle with the output was chosen as the "best guess". Slightly better performance was achieved by choosing the phoneme whose representation had the smallest Euclidean distance from the output vector, but these results are not reported here. All performance figures in this section refer to the percentage of correct phonemes chosen by the network. The performance was also assayed by "playing" the output string of phonemes and stresses through DECtalk, bypassing the part of the machine that converts letters to phonemes.

3. Performance

Continuous informal speech. [9] provide phonetic transcriptions of children and adults that were tape recorded during informal sessions. This was a particularly difficult training corpus because the same word was often pronounced several different ways; phonemes were commonly elided or modified at word boundaries, and adults were about as inconsistent as children.

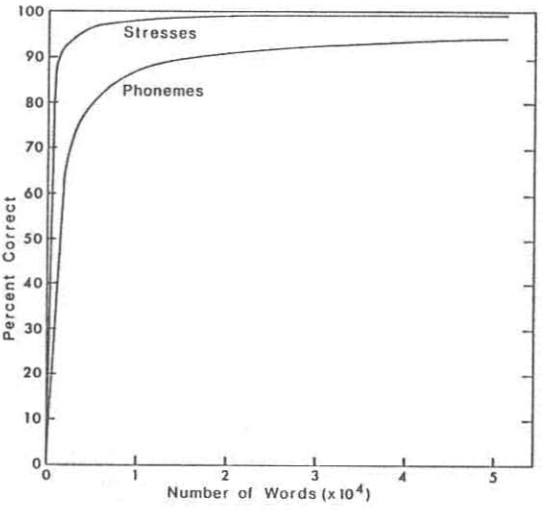

Figure 4: Learning curves for phonemes and stresses during training on the 1024 word corpus of continuous informal speech. The percentage of correct phonemes and stresses are shown as functions of the number of training words.

We used the first two pages of transcriptions, which contained 1024 words from a child in firstgrade. The stresses were assigned to the transcriptions so that the training text sounded natural when played through DECtalk. The learning curve for 1024 words from the informal speech corpus is shown in Figure 4. The percentage of correct phonemes rose rapidly at first and continued to rise at slower rate throughout the learning, reaching 95% after 50 passes through the corpus. Primary and secondary stresses and syllable boundaries were learned very quickly for all words and achieved nearly perfect performance by 5 passes (Figure 4). When the learning curves were plotted as error rates on double logarithmic scales they were approximately straight lines, so that the learning follows a power law, which is characteristic of human skill learning [64].

The distinction between vowels and consonants was made early; however, the network predicted the same vowel for all vowels and the same consonant for all consonants, which resulted in a babbling sound. A second stage occurred when word boundaries are recognized, and the output then resembled pseudowords. After just a few passes through the network many of the words were intelligible, and by 10 passes the text was understandable.

When the network made an error it often substituted phonemes that sounded similar to each other. For example, a common confusion was between the "th" sounds in "thesis" and "these" which differ only in voicing.

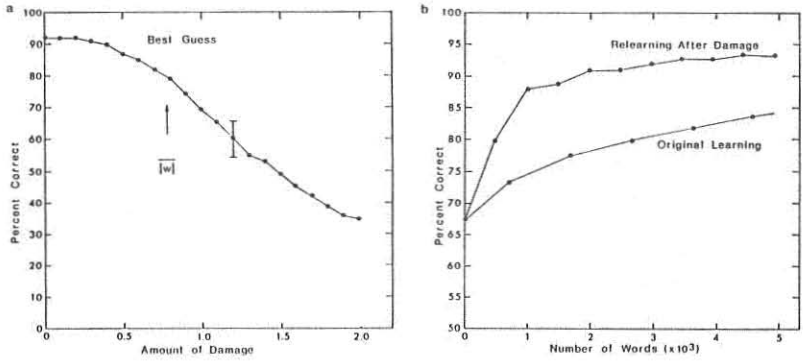

Figure 5: (a) Performance of a network as a function of the amount of damage to the weights. (b) Retraining of a damaged network compared with the original learning curve starting from the same level of performance. The network was damaged by adding a random component to all the weights uniformly distributed on the interval [-1.2, 1.2].

Few errors in a well-trained network were confusions between vowels and consonants. Some errors were actually corrections to inconsistencies in the original training corpus. Overall, the intelligibility of the speech was quite good.

Did the network memorize the training words or did it capture the regular features of pronunciation? As a test of generalization, a network trained on the 1024 word corpus of informal speech was tested without training on a 439 word continuation from the same speaker. The performance was 78%, which indicates that much of the learning was transferred to novel words even after a small sample of English words.

Is the network resistant to damage? We examined performance of a highly-trained network after making random changes of varying size to the weights. As shown in Figure 5(a), random perturbations of the weights uniformly distributed on the interval [-0.5, 0.5] had little effect on the performance of the network, and degradation was gradual with increasing damage. This damage caused the magnitude of each weight to change on average by 0.25; this is the roundoff error that can be tolerated before the performance of the network begins to deteriorate and it can be used to estimate the accuracy with which each weight must be specified. The weights had an average magnitude of 0.8 and almost all had a magnitude of less than 2. With 4 binary bits it is possible to specify 16 possible values, or -2 to +2 in steps of 0.25. Hence, the minimum information needed to specify each weight in the network is only about 4 bits.

If the damage is not too severe, relearning was much faster than the

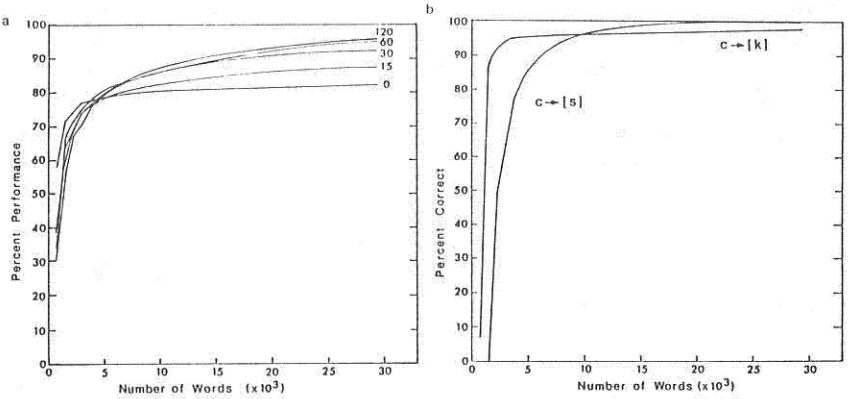

Figure 6: (a) Learning curves for training on a corpus of the 1000 most common words in English using different numbers of hidden units, as indicated beside each curve. (b) Performance during learning of two representative phonological rules, the hard and soft pronunciation of the letter "c".

original learning starting from the same level of performance, as shown in Figure 5(b). Similar fault tolerance and fast recovery from damage has also been observed in networks constructed using the Boltzmann learning algorithm [32].

Dictionary. The *Miriam Webster's Pocket Dictionary* that we used had 20,012 words. A subset of the 1000 most commonly occurring words was selected from this dictionary based on frequency counts in the Brown corpus [43]. The most common English words are also amongst the most irregular, so this was also a test of the capacity of the network to absorb exceptions. We were particularly interested in exploring how the performance of the network and learning rate scaled with the number of hidden units. With no hidden units, only direct connections from the input units to the output units, the performance rose quickly and saturated at 82% as shown in Figure 6(a). This represents the part of the mapping that can be accomplished by linearly separable partitioning of the input space [53]. Hidden units allow more contextual influence by recognizing higher-order features amongst combinations of input units.

The rate of learning and asymptotic performance increased with the number of hidden units, as shown in Figure 6(a). The best performance achieved with 120 hidden units was 98% on the 1000 word corpus, significantly better than the performance achieved with continuous informal speech, which was more difficult because of the variability in real-world speech. Different letter-to-sound correspondences were learned at different rates and two examples are shown in Figure 6(b): the soft "c" takes longer to learn, but eventually achieves perfect accuracy. The hard "c" occurs

about twice as often as the soft "c" in the training corpus. Children shown a similar difficulty with learning to read words with the soft "c" [84].

The ability of a network to generalize was tested on a large dictionary. Using weights from a network with 120 hidden units trained on the 1000 words, the average performance of the network on the dictionary of 20,012 words was 77%. With continued learning, the performance reached 85% at the end of the first pass through the dictionary, indicating a significant improvement in generalization. Following five training passes through the dictionary, the performance increased to 90%.

The number of input groups was varied from three to eleven. Both the speed of learning and the asymptotic level of performance improved with the size of the window. The performance with 11 input groups and 80 hidden units was about 7% higher than a network with 7 input groups and 80 hidden units up to about 25 passes through the corpus, and reached 97.5% after 55 passes compared with 95% for the network with 7 input groups.

Adding an extra layer of hidden units also improved the performance somewhat. A network with 7 input groups and two layers of 80 hidden units each was trained first on the 1000 word dictionary. Its performance after 55 passes was 97% and its generalization was 80% on the 20,012 word dictionary without additional training, and 87% after the first pass through the dictionary with training. The asymptotic performance after 11 passes through the the dictionary was 91%. Compared to the network with 120 hidden units, which had about the same number of weights, the network with two layers of hidden units was better at generalization but about the same in absolute performance.

4. Analysis of the Hidden Units

There are not enough hidden units in even the largest network that we studied to memorize all of the words in the dictionary. The standard network with 80 hidden units had a total of 18,629 weights, including variable thresholds. If we allow 4 bits of accuracy for each weight, as indicated by the damage experiments, the total storage needed to define the network is about 80,000 bits. In comparison, the 20,012 word dictionary, including stress information, required nearly 2,000,000 bits of storage. This data compression is possible because of the redundancy in English pronunciation. By studying the patterns of activation amongst the hidden units, we were able to understand some of the coding methods that the network had discovered.

The standard network used for analysis had 7 input groups and 80 hidden units and had been trained to 95% correct on the 1000 dictionary words. The levels of activation of the hidden units were examined for each letter of each word using the graphical representation shown in Figure 7. On average, about 20% of the hidden units were highly activated for any given input, and most of the remaining hidden units had little or no ac-

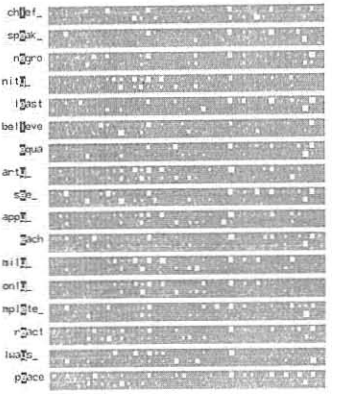

Figure 7: Levels of activation in the layer of hidden units for a variety of words, all of which produce the same phoneme, /E/, on the output. The input string is shown at the left with the center letter emphasized. The level of activity of each hidden unit is shown to the right, in two rows of 40 units each. The area of the white square is proportional to the activity level.

tivation. Thus, the coding scheme could be described neither as a local representation, which would have activated only a few units [6,20,8], or a "holographic" representation [88,?] , in which all of the hidden units would have participated to some extent. It was apparent, even without using statistical techniques, that many hidden units were highly activated only for certain letters, or sounds, or letter-to-sound correspondences. A few of the hidden units could be assigned unequivocal characterizations, such as one unit that responded only to vowels, but most of the units participated in more than one regularity.

To test the hypothesis that letter-to-sound correspondences were the primary organizing variable, we computed the average activation level of each hidden unit for each letter-to-sound correspondence in the training corpus. The result was 79 vectors with 80 components each, one vector for each letter-to-sound correspondence. A hierarchical clustering technique was used to arrange the letter-to-sound vectors in groups based on a Euclidean metric in the 80-dimensional space of hidden units. The overall pattern, as shown in Figure 8, was striking: the most important distinction was the complete separation of consonants and vowels. However, within these two groups the clustering had a different pattern. For the vowels, the next most important variable was the letter, whereas consonants were clustered according to a mixed strategy that was based more on the similarity of their sounds. The same clustering procedure was repeated for three networks starting from different random starting states. The patterns of

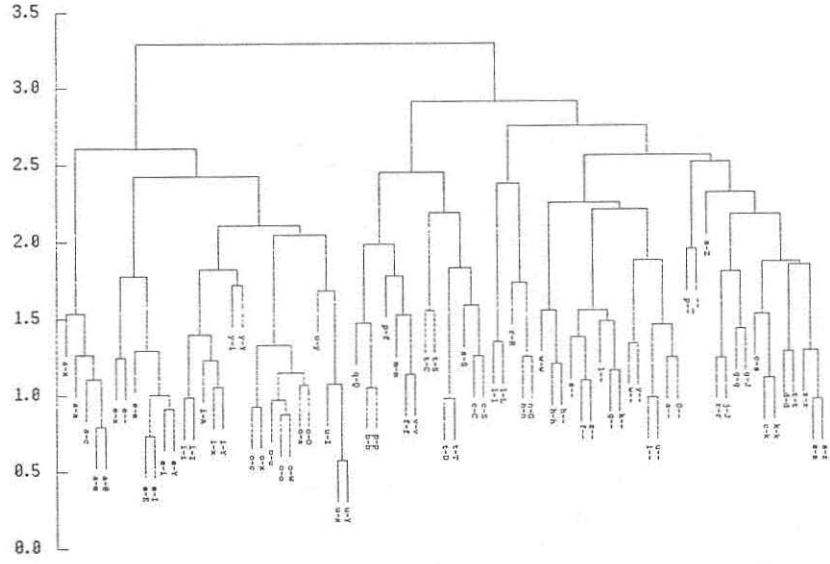

Figure 8: Hierarchical clustering of hidden units for letter-to-sound correspondences. The vectors of average hidden unit activity for each correspondance, shown at the bottom of the binary tree (l-¿p for letter 'l' and phoneme 'p,), were successively grouped according to an agglomerative method using complete linkage ([18]).

weights were completely different but the clustering analysis revealed the same hierarchies, with some differences in the details, for all three networks.

5. Conclusions

NETtalk is an illustration in miniature of many aspects of learning. First, the network starts out without 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. Second, the network acquired its competence through practice, went through several distinct stages, and reached a significant level of performance. Finally, the information was distributed in the network such that no single unit or link was essential. As a consequence, the network was fault tolerant and degraded gracefully with increasing damage. Moreover, the network recovered from damage much more quickly than it took to learn initially.

Despite these similarities with 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 stage what occurs in two stages of human development. Children learn to talk first, and only

after representations for words and their meanings are well developed do they learn to read. It is also very likely that we have access to articulatory representations for whole words, in addition to the our ability to use letter-to-sound correspondences, but there are no 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. This approach would have to be generalized to account for prosodic features in continuous text and a human level of performance would require the integration of information from several words at once.

NETtalk can be used as a research tool to explore many aspects of network coding, scaling, and training in a domain that is far from trivial. Those aspect of the network's performance that are similar to human performance are good candidates for general properties of network models; more progress may be made by studying these aspects in the small test laboratory that NETtalk affords. For example, we have shown elsewhere [62] that the optimal training schedule for teaching NETtalk new words is to alternate training of the new words with old words, a general phenomenon of human memory that was first demonstrated by Ebbinghaus [16] and has since been replicated with a wide range of stimulus materials and tasks [33,34,58,38,77,24]. Our explanation of this spacing effect in NETtalk [62] may generalize to more complex memory systems that use distributed representations to store information.

After training many networks, we concluded that many different sets of weights give about equally good performance. Although it was possible to understand the function of some hidden units, it was not possible to identify units in different networks that had the same function. However, the activity patterns in the hidden units were interpretable in an interesting way. Patterns of activity in groups of hidden units could be identified in different networks that served the same function, such as distinguishing vowels and consonants. This suggests that the detailed synaptic connectivity between neurons in cerebral cortex may not be helpful in revealing the functional properties of a neural network. It is not at the level of the synapse or the neuron that one should expect to find invariant properties of a network, but at the level of functional groupings of cells. We are continuing to analyze the hidden units and have found statistical patterns that are even more detailed than those reported here. Techniques that are developed to uncover these groupings in model neural networks could be of value in uncovering similar cell assemblies in real neural networks.

Acknowledgments

We thank Drs. Alfonso Caramazza, Francis Crick, Stephen Hanson, James McClelland, Geoffrey Hinton, Thomas Landauer, George Miller, David Rumelhart and Stephen Wolfram for helpful discussions about language and learning. We are indebted to Dr. Stephen Hanson and Andrew Olson who made important contributions in the statistical analysis of the hidden

units. Drs. Peter Brown, Edward Carterette, Howard Nusbaum and Alex Waibel assisted in the early stages of development. Bell Communications Research generously provided computational support.

TJS was supported by grants from the National Science Foundation, System Development Foundation, Sloan Foundation, General Electric Corporation, Allied Corporation Foundation, Richard Lounsbery Foundation, Seaver Institute, and the Air Force Office of Scientific Research. CRR was supported in part by grants from the James S. McDonnell foundation, research grant 487906 from IBM, by the Defense Advanced Research Projects Agency of the Department of Defense, the Office of Naval Research under Contracts Nos. N00014-85-C-0456 and N00014-85-K-0465, and by the National Science Foundation under Cooperative Agreement No. DCR-8420948 and grant number IST8503968.

Appendix A. Representation of Phonemes and Punctuations

Phoneme	Sound	Articulatory Features
/a/	father	Low, Tensed, Central2
/b/	*b*et	Voiced, Labial, Stop
/c/	bought	Medium, Velar
/d/	*d*eb	Voiced, Alveolar, Stop
/e/	bake	Medium, Tensed, Front2
/f/	*f*in	Unvoiced, Labial, Fricative
/g/	*g*uess	Voiced, Velar, Stop
/h/	*h*ead	Unvoiced, Glottal, Glide
/i/	Pete	High, Tensed, Front1
/k/	*K*en	Unvoiced, Velar, Stop
/l/	*l*et	Voiced, Dental, Liquid
/m/	*m*et	Voiced, Labial, Nasal
/n/	*n*et	Voiced, Alveolar, Nasal
/o/	boat	Medium, Tensed, Back2
/p/	*p*et	Unvoiced, Labial, Stop
/r/	*r*ed	Voiced, Palatal, Liquid
/s/	*s*it	Unvoiced, Alveolar, Fricative
/t/	*t*est	Unvoiced, Alveolar, Stop
/u/	l*u*te	High, Tensed, Back2
/v/	*v*est	Voiced, Labial, Fricative
/w/	*w*et	Voiced, Labial, Glide
/x/	about	Medium, Central2
/y/	*y*et	Voiced, Palatal, Glide
/z/	*z*oo	Voiced, Alveolar, Fricative
/A/	b*i*te	Medium, Tensed, Front2 + Central1
/C/	*ch*in	Unvoiced, Palatal, Affricative
/D/	*th*is	Voiced, Dental, Fricative
/E/	b*e*t	Medium, Front1 + Front2
/G/	si*ng*	Voiced, Velar, Nasal
/I/	b*i*t	High, Front1
/J/	*g*in	Voiced, Velar, Nasal
/K/	se*x*ual	Unvoiced, Palatal, Fricative + Velar, Affricative
/L/	bott*le*	Voiced, Alveolar, Liquid
/M/	absy*m*	Voiced, Dental, Nasal
/N/	butto*n*	Voiced, Palatal, Nasal
/O/	b*oy*	Medium, Tensed, Central1 + Central2
/Q/	*qu*est	Voiced, Labial + Velar, Affricative, Stop
/R/	b*ir*d	Voiced, Velar, Liquid

Phoneme	Sound	Articulatory Features
/S/	*sh*in	Unvoiced, Palatal, Fricative
/T/	*th*in	Unvoiced, Dental, Fricative
/U/	b*oo*k	High, Back1
/W/	b*ou*t	High + Medium, Tensed, Central2 + Back1
/X/	e*x*cess	Unvoiced, Affricative, Front2 + Central1
/Y/	c*u*te	High, Tensed, Front1 + Front2 + Central1
/Z/	lei*s*ure	Voiced, Palatal, Fricative
/@/	b*a*t	Low, Front2
/!/	Na*z*i	Unvoiced, Labial + Dental, Affricative
/#/	e*x*amine	Voiced, Palatal + Velar, Affricative
/*/	*o*ne	Voiced, Glide, Front1 + Low, Central1
/|/	l*o*gic	High, Front1 + Front2
/^/	b*u*t	Low, Central1
/-/	Continuation	Silent, Elide
/_/	Word Boundary	Pause, Elide
/./	Period	Pause, Full Stop
<	Syllable Boundary	right
>	Syllable Boundary	left
1	Primary Stress	strong, weak
2	Secondary Stress	strong
0	Tertiary Stress	weak
_	Word Boundary	right, left, boundary

Output representations for phonemes, punctuations, and stresses on the 26 output units. The symbols for phonemes in the first column are a superset of ARPAbet and are associated with the sound of the italicized part of the adjacent word. Compound phonemes were introduced when a single letter was associated with more than one primary phoneme. Two or more of the following 21 articulatory feature units were used to represent each phoneme and punctuation: *Position in mouth*: Labial = Front1, Dental = Front2, Alveolar = Central1, Palatal = Central2, Velar = Back1, Glottal = Back2; *Phoneme Type*: Stop, Nasal, Fricative, Affricative, Glide, Liquid, Voiced, Tensed; *Vowel Height*: High, Medium, Low; *Punctuation*: Silent, Elide, Pause, Full Stop. The continuation symbol was used when a letter was silent. Stress and syllable boundaries were represented with combinations of 5 additional units, as shown at the end of the table. Stress was associated with vowels, and arrows were associated with the other letters. The arrows point toward the stress and change direction at syllable boundaries. Thus, the stress assignments for ''atmosphere'' are ''1<>0>>>2<<''. The phoneme and stress assignments were chosen independently

References

[1] D. H. Ackley, G. E. Hinton and T. J. Sejnowski, "A learning algorithm for Boltzmann machines", *Cognitive Science*, **9** (1985) 147-169.

[2] M. A. Arbib, *Brains, Machines & Mathematics*, 2nd edition, (McGraw-Hill Press, 1987).

[3] J. Allen, *From Text to Speech: The MITalk System*, (Cambridge University Press, 1985).

[4] J. R. Anderson, "Acquisition of cognitive skill", *Psychological Review*, **89** (1982) 369-406.

[5] D. H. Ballard, G. E. Hinton, and T. J. Sejnowski, "Parallel visual computation", *Nature*, **306** (1983) 21-26.

[6] H. B. Barlow, "Single units and sensation: A neuron doctrine for perceptual psychology?", *Perception*, **1** (1972) 371-394.

[7] A. G. Barto, "Learning by statistical cooperation of self-interested neuron-like computing elements", *Human Neurobiology* **4** (1985) 229-256.

[8] E. B. Baum, J. Moody, and F. Wilczek, "Internal representations for associative memory", reprint, Institute for Theoretical Physics, University of California, Santa Barbara (1987).

[9] E. C. Carterette and M. G. Jones, *Informal Speech*. (Los Angeles: University of California Press, 1974).

[10] K. Church, "Stress assignment in letter to sound rules for speech synthesis", in *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics* (1985).

[11] P. S. Churchland, *Neurophilosophy*, (MIT Press, 1986).

[12] M. A. Cohen and S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks", *IEEE Transaction on Systems, Man and Cybernetics*, **13** (1983) 815-825.

[13] L. N. Cooper, F. Liberman and E. Oja, "A theory for the acquisition and loss of neuron specificity in visual cortex", *Biological Cybernetics* **33**, (1979) 9-28.

[14] F. H. C. Crick and C. Asanuma, "Certain aspects of the anatomy and physiology of the cerebral cortex", in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 2: Psychological and Biological Models*, edited by J. L. McClelland & D. E. Rumelhart, (MIT Press, 1986).

[15] Digital Equipment Corporation, "DECtalk DTC01 Owner's Manual", (Digital Equipment Corporation, Maynard, Mass.; document number EK-DTC01-OM-002).

[16] H. Ebbinghaus, *Memory: A contribution to Experimental Psychology*, (Reprinted by Dover, New York, 1964; originally published 1885).

[17] J. Ellman and D. Zipser, "University of California at San Diego, Institute for Cognitive Science Technical Report" (1985).

[18] B. Everitt, *Cluster Analysis*, (Heinemann: London, 1974).

[19] S. E. Fahlman, G. E. Hinton and T. J. Sejnowski, "Massively-parallel architectures for AI: NETL, THISTLE and Boltzmann Machines", *Proceedings of the National Conference on Artificial Intelligence*, (Washington, D. C., 1983) 109-113.

[20] J. A. Feldman, "Dynamic connections in neural networks", *Biological Cybernetics*, **46**, (1982) 27-39.

[21] J. A. Feldman, "Neural representation of conceptual knowledge", Technical Report TR-189, University of Rochester Department of Computer Science (1986).

[22] J. A. Feldman and D. H. Ballard, "Connectionist models and their properties", *Cognitive Science*, **6** (1982) 205-254.

[23] A. L. Gamba, G. Gamberini, G. Palmieri, and R. Sanna, "Further experiments with PAPA", *Nuovo Cimento Suppl.*, No. 2, **20** (1961) 221-231.

[24] A. M. Glenberg, "Monotonic and Nonmonotonic Lag Effects in Paired-Associate and Recognition Memory Paradigms", *Journal of Verbal Learning and Verbal Behavior*, **15** (1976) 1-16.

[25] M. A. Gluck and G. H. Bower, "From conditioning to category learning: An adaptive network model", in preparation.

[26] M. A. Gluck and R. F. Thompson, "Modeling the neural substrates of associative learning and memory: A computational approach", *Psychological Review* (1986).

[27] W. Haas, *Phonographic Translation*, (Manchester: Manchester University Press, 1970).

[28] D. O. Hebb, *Organization of Behavior*, (John Wiley & Sons, 1949).

[29] G. E. Hinton, "Learning distributed representations of concepts", *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, (Hillsdale, New Jersey: Erlbaum, 1986) 1-12.

[30] G. E. Hinton and J. A. Anderson, *Parallel models of associative memory*, (Hillsdale, N. J.: Erlbaum Associates, 1981).

[31] G. E. Hinton and T. J. Sejnowski, "Optimal perceptual inference", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Washington, D. C., 1983) 448-453.

[32] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in Boltzmann machines", in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 2: Psychological and Biological Models*, edited by J. L. McClelland & D. E. Rumelhart, (MIT Press, 1986).

[33] D. L. Hintzman, "Theoretical implications of the spacing effect", in *Theories in Cognitive Psychology: The Loyola Symposium*, edited by R.L. Solso, (Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1974).

[34] D. L. Hintzman, "Repetition and memory", in *The Psychology of Learning and Motivation*, edited by G. H. Bower, (Academic Press, 1976).

[35] J. J. Hopfield and D. Tank, "Computing with neural circuits: A model", *Science*, **233** (1986) 624-633.

[36] T. Hogg and B. A. Huberman, "Understanding biological Computation", *Proceedings of the National Academy of Sciences USA*, **81** (1986) 6871-6874.

[37] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interactions, and functional architecture in the cat's visual cortex", *Journal of Physiology*, **160** (1962) 106-154.

[38] L. L. Jacoby, "On Interpreting the Effects of Repetition: Solving a Problem Versus Remembering a Solution", *Journal of Verbal Learning and Verbal Behavior*, **17** (1978) 649-667.

[39] S. R. Kelso, A. H. Ganong, and T. H. Brown, "Hebbian synapses in hippocampus", *Proceedings of the National Academy of Sciences USA*, **83** (1986) 5326-5330.

[40] D. Klatt, "Software for a cascade/parallel formant synthesizer", *Journal of the Acoustical Society of America*, **67** (1980) 971-995.

[41] C. Koch, J. Marroquin, and A. Yuille, *Proceedings of the National Academy of Sciences USA*, **83** (1986) 4263-4267.

[42] T. Kohonen, *Self-Organization and Associative Memory*, (New York: Springer Verlag, 1984).

[43] H. Kuchera, and W. N. Francis, *Computational Analysis of Modern-Day American English*, (Providence, Rhode Island: Brown University Press, 1967).

[44] Y. Le Cun, "A learning procedure for asymmetric network", *Proceedings of Cognitiva (Paris)*, **85** (1985) 599-604.

[45] S. Lehky, and T. J. Sejnowski, "Computing Shape from Shading with a Neural Network Model", in preparation.

[46] W. B. Levy, J. A. Anderson and W. Lehmkuhle, *Synaptic Change in the Nervous System*, (Hillsdale, New Jersey: Erlbaum, 1984).

[47] H. C. Longuet-Higgins, "Holographic model of temporal recall", *Nature*, **217** (1968) 104-107.

[48] J. M. Lucassen and R. L. Mercer, "An information theoretic approach to the automatic determination of phonemic baseforms", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (1984) 42.5.1-42.5.4.

[49] G. Lynch, *Synapses, Circuits, and the Beginnings of Memory*, (MIT Press, 1986).

[50] J. L. McClelland and D. E. Rumelhart, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 2: Psychological and Biological Models*, (MIT Press, 1986).

[51] D. Marr and T Poggio, "Cooperative computation of stereo disparity", *Science*, **194** (1976) 283-287.

[52] W. S. McCulloch and W. H. Pitts, "A logical calculus of ideas immanent in nervous activity", *Bull. Math. Biophysics*, **5** (1943) 115-133.

[53] M. Minsky and S. Papert, *Perceptrons*, (MIT Press, 1969).

[54] V. B. Mountcastle, "An organizing principle for cerebral function: The unit module and the distributed system", in *The Mindful Brain*, edited by G. M. Edelman & V. B. Mountcastle, (MIT Press, 1978).

[55] D. A. Norman, *Learning and Memory*, (San Francisco: W. H. Freeman, 1982).

[56] G. Palm, "On representation and approximation of nonlinar systems, Part II: Discrete time", *Biological Cybernetics*, **34** (1979) 49-52.

[57] D. B Parker, "A comparison of algorithms for neuron-like cells", in *Neural Networks for Computing*, edited by J. S. Denker, (New York: American Institute of Physics, 1986).

[58] L. R. Peterson, R. Wampler, M. Kirkpatrick and D. Saltzman, "Effect of spacing presentations on retention of a paired-associate over short intervals", *Journal of Experimental Psychology*, **66** (1963) 206-209.

[59] R. W. Prager, T. D Harrison, and F. Fallside, "Boltzmann machines for speech recognition", Cambridge University Engineering Department Technical Report TR.260 (1986).

[60] R. A. Rescorla and A. R. Wagner, "A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non- reinforcement", in *Classical Conditioning II: Current Research and Theory*, edited by A. H. Black & W. F. Prokasy, (New York: Appleton-Crofts, 1972).

[61] E. T. Rolls, "Information representation, processing and storage in the brain: Analysis at the single neuron level", in *Neural and Molecular Mechanisms of Learning*, (Berlin: Springer Verlag, 1986).

[62] C. R. Rosenberg and T. J. Sejnowski, "The spacing effect on NETtalk, a massively-parallel network", *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, (Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1986) 72-89.

[63] F. Rosenblatt, *Principles of Neurodynamics*, (New York: Spartan Books, 1959).

[64] P. S. Rosenbloom and A. Newell, "The chunking of goal hierarchies: A generalized model of practice", in *Machine Learning: An Artificial Intelligence Aproach, Vol. II*, edited by R. S. Michalski, J. G. Carbonell & T. M. Mitchell, (Los Altos, California: Morgan Kauffman, 1986).

[65] D. E. Rumelhart, "Presentation at the Symposium on Connectionism: Multiple Agents, Parallelism and Learning", Geneva, Switzerland (September 9-12, 1986).

[66] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error propagation", in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Fundations*, edited by D. E. Rumelhart & J. L. McClelland, (MIT Press, 1986).

[67] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations*, (MIT Press, 1986).

[68] D. E. Rumelhart and J. L. McClelland, "On learning the past tenses of English verbs", in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 2: Psychological and Biological Models*, edited by J. L. McClelland & D. E. Rumelhart, (MIT Press, 1986).

[69] E. Saund, "Abstraction and representation of continuous variables in connectionist networks", *Proceedings of the Fifth National Conference on Artificial Intelligence*, (Los Altos, California: Morgan Kauffmann, 1986) 638-644.

[70] T. J. Sejnowski, "Skeleton filters in the brain", in *Parallel models of associative memory*, edited by G. E. Hinton & J. A. Anderson, (Hillsdale, N. J.: Erlbaum Associates, 1981).

[71] T. J. Sejnowski, "Open questions about computation in cerebral cortex", in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 2: Psychological and Biological Models*, edited by J. L. McClelland & D. E. Rumelhart, (MIT Press, 1986).

[72] T. J. Sejnowski and G. E. Hinton, "Separating figure from ground with a Boltzmann Machine", in *Vision, Brain & Cooperative Computation*, edited by M. A. Arbib & A. R. Hanson (MIT Press, 1987).

[73] T. J. Sejnowski, P. K. Kienker and G. E. Hinton, "Learning symmetry groups with hidden units: Beyond the perceptron", *Physica*, **22D** (1986).

[74] T. J. Sejnowski and C. R. Rosenberg, "NETtalk: A parallel network that learns to read aloud", Johns Hopkins University Department of Electrical Engineering and Computer Science Technical Report 86/01 (1986).

[75] T. J. Sejnowski and C. R. Rosenberg, "Connectionist Models of Learning", in *Perspectives in Memory Research and Training*, edited by M. S. Gazzaniga, (MIT Press, 1986).

[76] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory", in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 2: Psychological and Biological Models*, edited by J. L. McClelland & D. E. Rumelhart, (MIT Press, 1986).

[77] R. D. Sperber, "Developmental changes in effects of spacing of trials in retardate discrimination learning and memory", *Journal of Experimental Psychology*, **103** (1974) 204-210.

[78] L. R. Squire, "Mechanisms of memory", *Science*, **232** (1986) 1612-1619.

[79] R. S. Sutton and A. G. Barto, "Toward a modern theory of adaptive networks: Expectation and prediction", *Psychological Review*, **88** (1981) 135-170.

[80] G. Tesauro, "Simple neural models of classical conditioning", *Biological Cybernetics*, **55** (1986) 187-200.

[81] G. Tesauro and T. J. Sejnowski, "A parallel network that learns to play backgammon", in preparation (1987).

[82] G. Toulouse, S. Dehaene, and J. P. Changeux, "Spin glass model of learning by selection", *Proceedings of the National Academy of Sciences USA*, **83** (1986) 1695-1698.

[83] R. L. Venezky, *The Structure of English Orthography*, (The Hague: Mouton, 1970).

[84] R. L. Venezky and D. Johnson, "Development of two letter- sound patterns in grades on through three", *Journal of Educational Psychology*, **64** (1973) 109-115.

[85] C. von der Malsburg, and E. Bienenstock, "A neural network for the retrieval of superimposed connection patterns", in *Disordered Systems and Biological Organization*, edited by F. Fogelman, F. Weisbuch, & E. Bienenstock, (Springer-Verlag: Berlin, 1986).

[86] R. L. Watrous, L. Shastri, and A. H. Waibel, "Learned phonetic discrimination using connectionist networks", University of Pennsylvania Department of Electrical Engineering and Computer Science Technical Report (1986).

[87] G. Widrow and M. E. Hoff, "Adaptive switching circuits", *Institute of Radio Engineers Western Electronic Show and Convention*, Convention Record 4 (1960) 96-194.

[88] D. Wilshaw, "Holography, associative memory, and inductive generalization", in *Parallel Models of Associative Memory*, edited by G. E. Hinton & J. A. Anderson, (Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1981).

[89] D. Zipser, "Programing networks to compute spatial functions", ICS Technical Report, Institute for Cognitive Science, University of California at San Diego (1986).