Parallel Networks that Learn to Pronounce English Text

T. J. Sejnowski
C. R. Rosenberg

Qiong Zhao

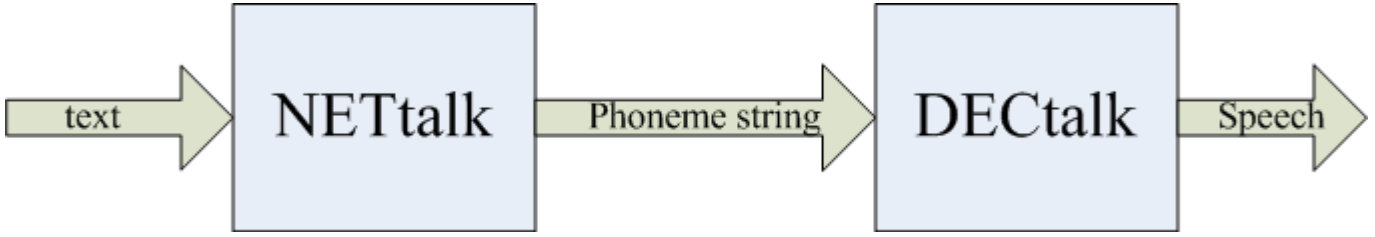
February 15, 2010

Outline

- Main idea
- Background
- How the system works
- Network performance
- Analysis of hidden units
- Conclusion

Main idea

- Train an artificial neural network (NETtalk) to pronounce English words: convert a string of English letters to phonemes



- NETtalk has some similarities with observed human performance

Background

- NETtalk
  - A class of massively-parallel network systems that learn to convert English text to speech
  - First proposed in 1980's
  - Turning point in the history of neural networks

- DECtalk
  - DECtalk is a speech synthesizer and text-to-speech technology
  - Use a lookup table for both common and irregular English words
  - The NETtalk system used part of DECtalk to convert phonemes to sounds

- Novel idea in NETtalk
- Different from DECtalk, NETtalk does not require a large amount of storage space
- It can capture regularities and absorb irregularities in English pronunciation in some way like human nervous system

How does NETtalk work?

How the system works

- How human brain works

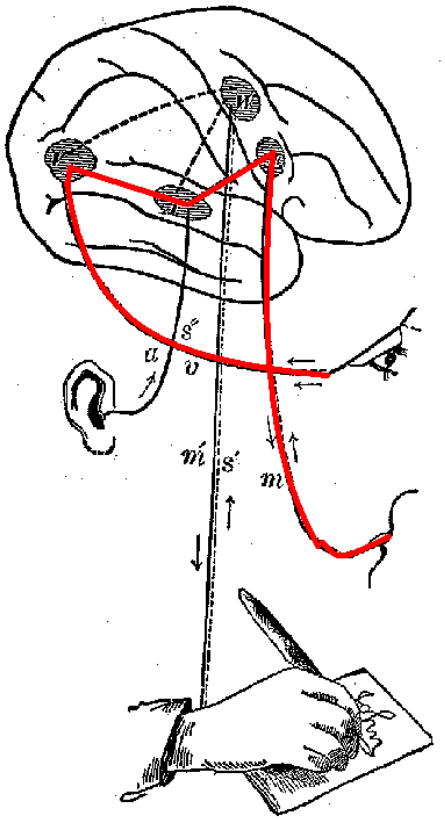
Step 1. words recognized by visual system

→

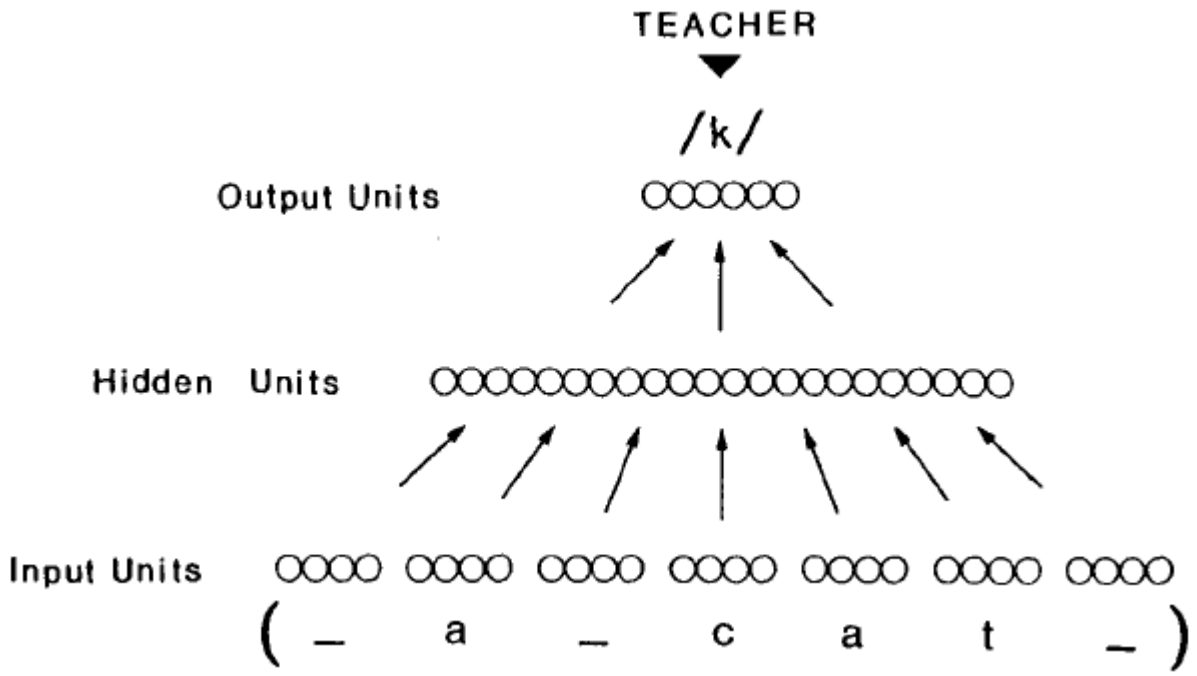
Step 2. visual information transformed into articulatory information

→

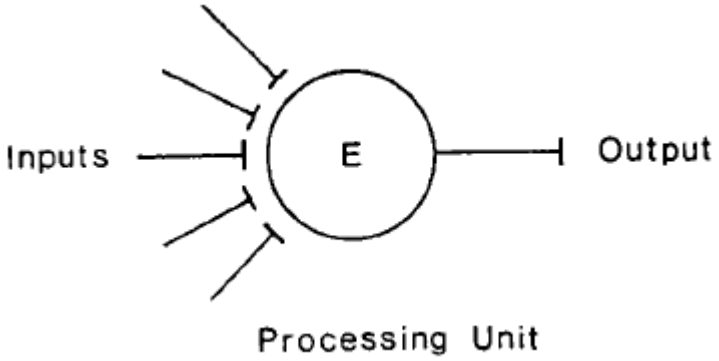
Step 3. motoneurons control muscles to produce sounds

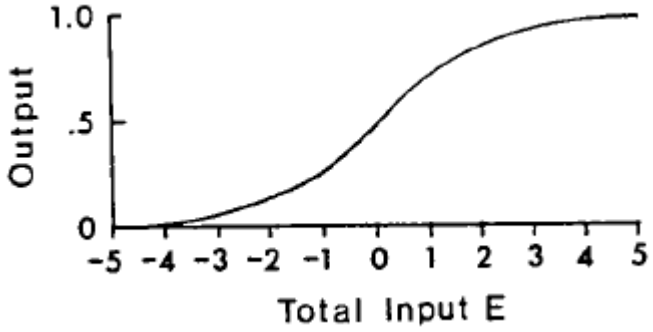


- NETtalk network architechture



- Processing units





Activation function: $$s_i = P(E_i) = \frac{1}{1+e^{-E_i}}$$

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

- Learning algorithm: back-propagation
- Goal: minimize average squared error

$$Error = \sum_{i=1}^{26} (s_i^* - s_i^{(N)})^2$$

where N represents the output layer, s_i^* is the correct output provided by a teacher

- Learning algorithm: back-propagation
- Calculating gradients layer by layer

first compute the error gradient on the output layer:

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

then propagating it backwards through the network:

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

- Learning algorithm: back-propagation
- Adjusting weights

Compute weight gradients with an exponentially decaying filter (u is the no. of input patterns):

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

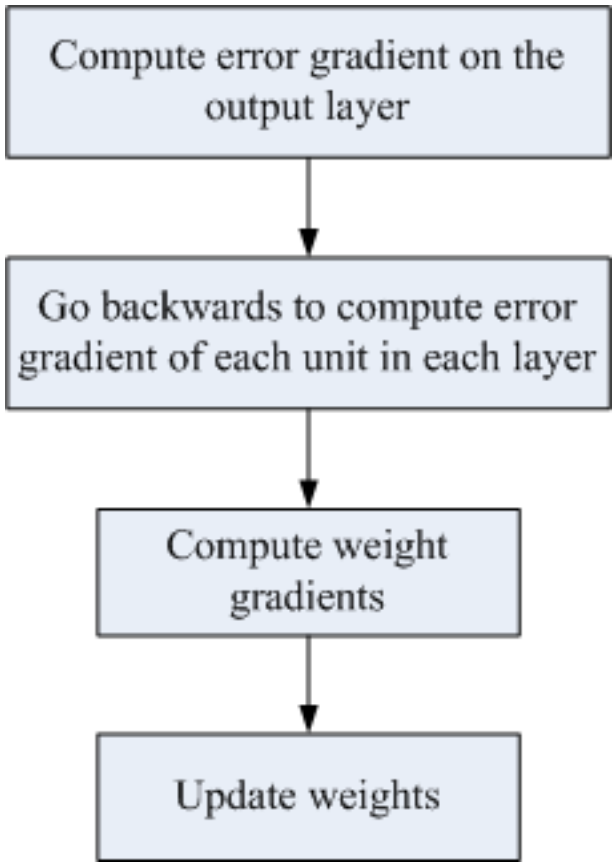
we get the general term:

$$\Delta w_{ij}^{(n)}(u) = \alpha^u \Delta w_{ij}^{(n)}(0) + (1-\alpha^u)\delta_i^{(n+1)} s_j^{(n)}$$

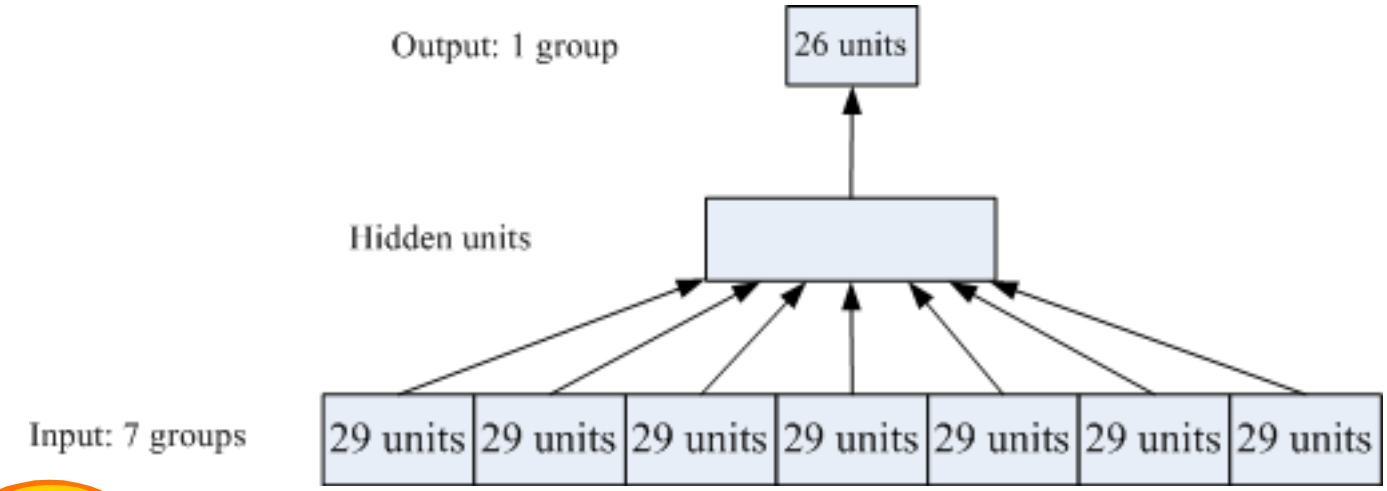
then we can use $\Delta w_{ij}^{(n)}(u)$ to update the weights:

$$w_{ij}^{(n)}(t+1) = w_{ij}^{(n)}(t) + \varepsilon \Delta w_{ij}^{(n)}$$

- Whole procedure



- Converting English text to speech
  - 7-letter-window



Why 7?

Correct pronunciation of a letter is contributed by the nearby letters;
Resource limited

- Converting English text to speech
- Representation of letters

  In the input side, each group contains 29 units:

  26 letters + comma + period + word boundary marker

- Converting English text to speech
- Representation of phonemes

  In the output side, there are 26 units in total:

  21 articulatory feature + right syllable boundary + left syllable boundary + primary stress + Secondary Stress + Tertiary Stress

  (Refer to the Appendix)

- Converting English text to speech
- Translate output to phoneme:

  Suppose v : output vector

  w_i : vector of each phoneme

  both v and w_i are binary vectors

  EX.

$$v = (\underbrace{1, 0, 0, \cdots, 1, 0, 1}_{\text{26 components}})$$

Compute inner product

$$\langle v | w_i \rangle$$

Choose w_j which made the smallest angle with v (closest to v) as the "best guess"

- Converting English text to speech
- Training materials:

1. Phonetic transcriptions from informal, continuous speech of a child

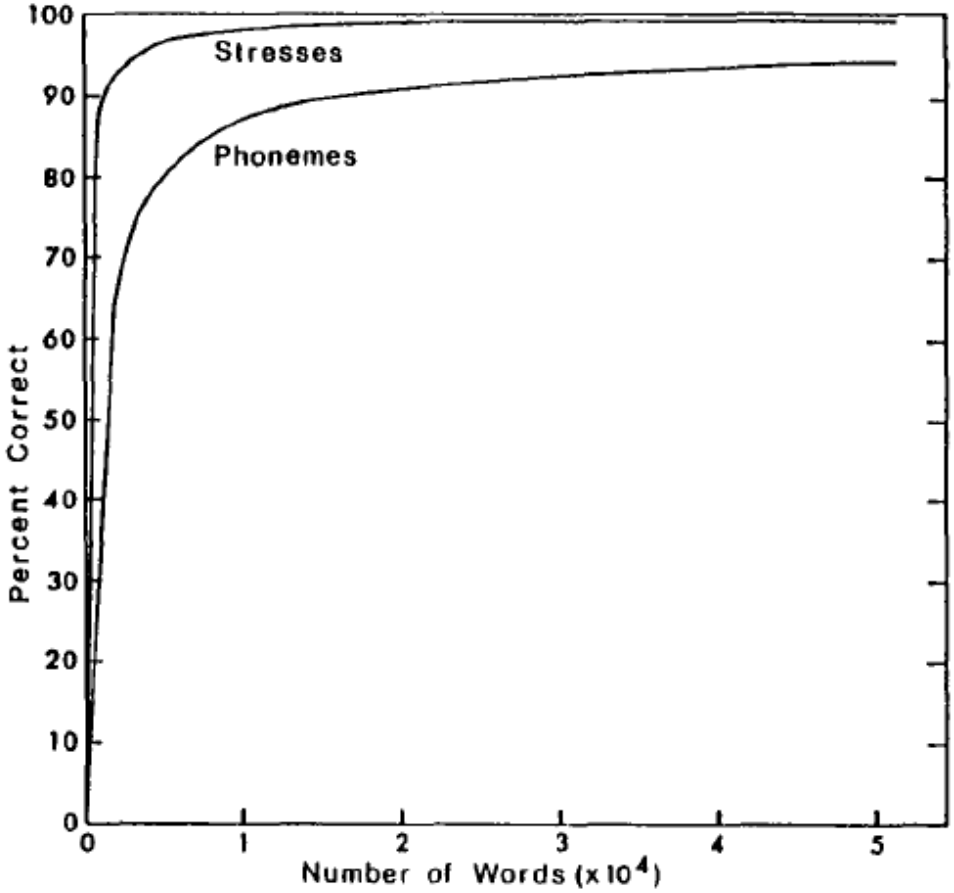
(text moved through input window with boundary symbols between words)

2. Miriam Webster's Pocket Dictionary

(words moved through input widow individually)

Performance

- Continuous informal speech (1024 word corpus)



- Analysis

  When plotted on double logarithmic scale, learning curves were approximately straight lines, so that the learning follows a power law, which is a characteristic of human skill learning.

- Audio sample

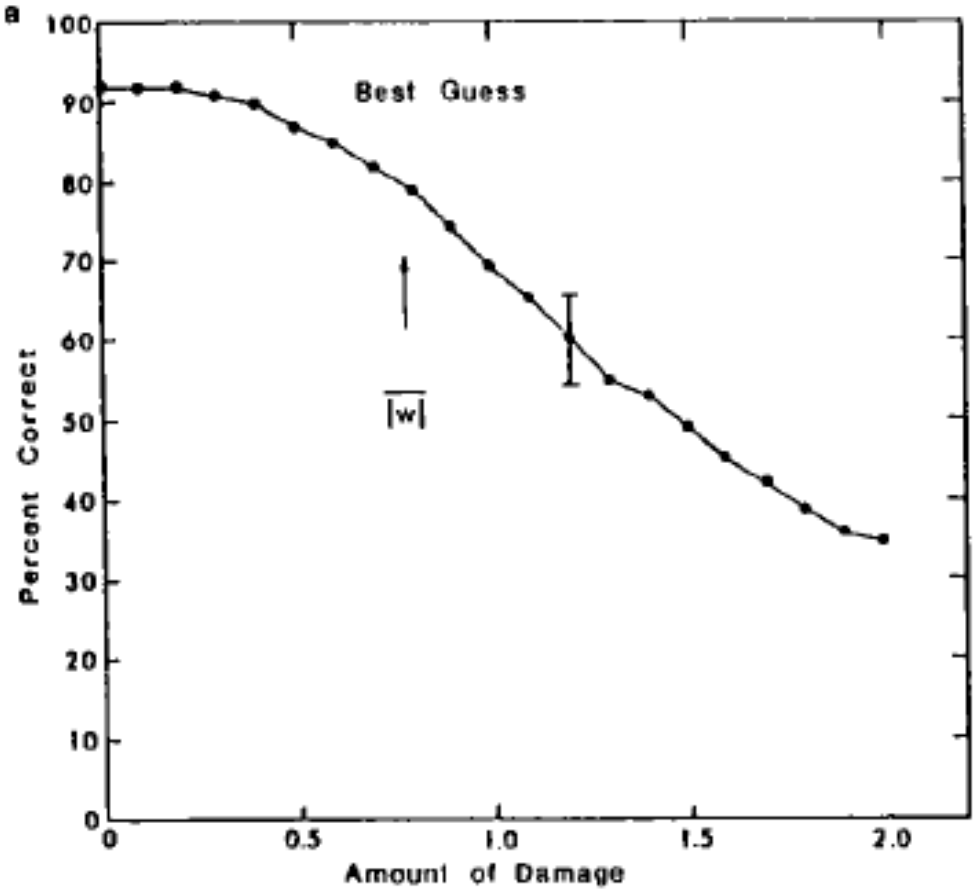
  nettalk.mp3

  Learning corpus taken from *Informal Speech*

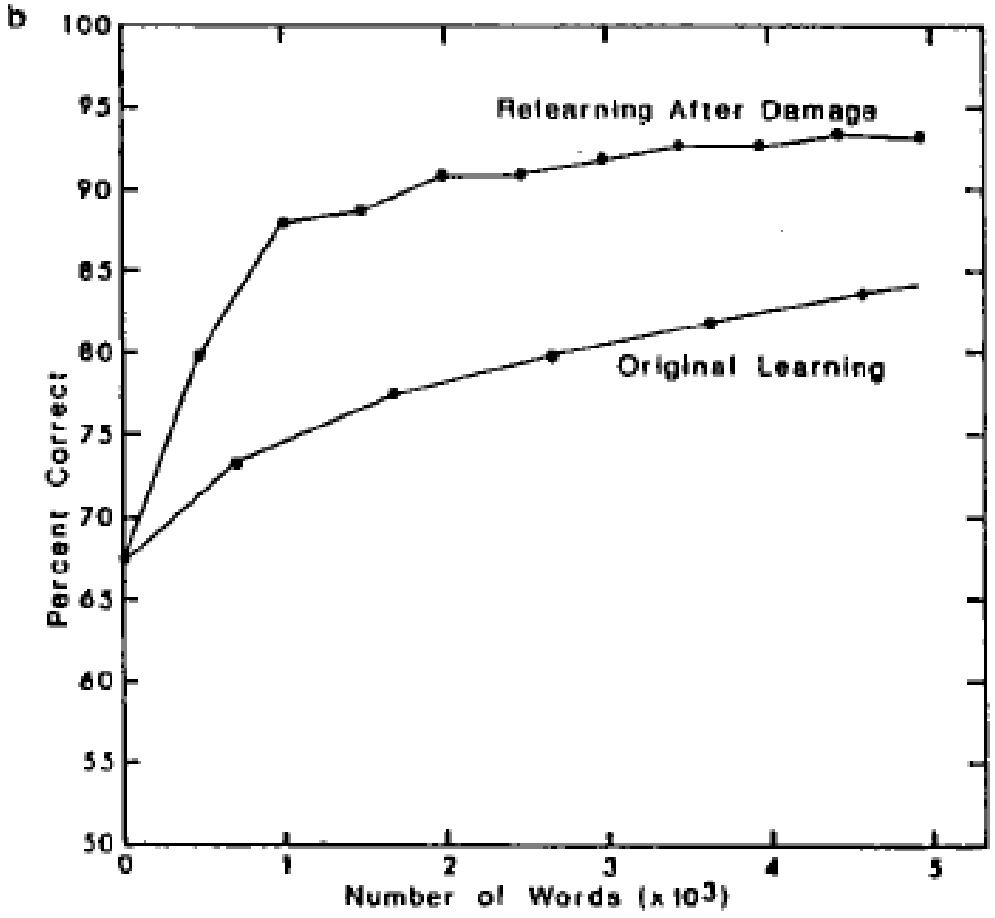
  There are three recordings:

  1. Network studying from zero weights
  2. After 20 passes
  3. Generalized to fresh text

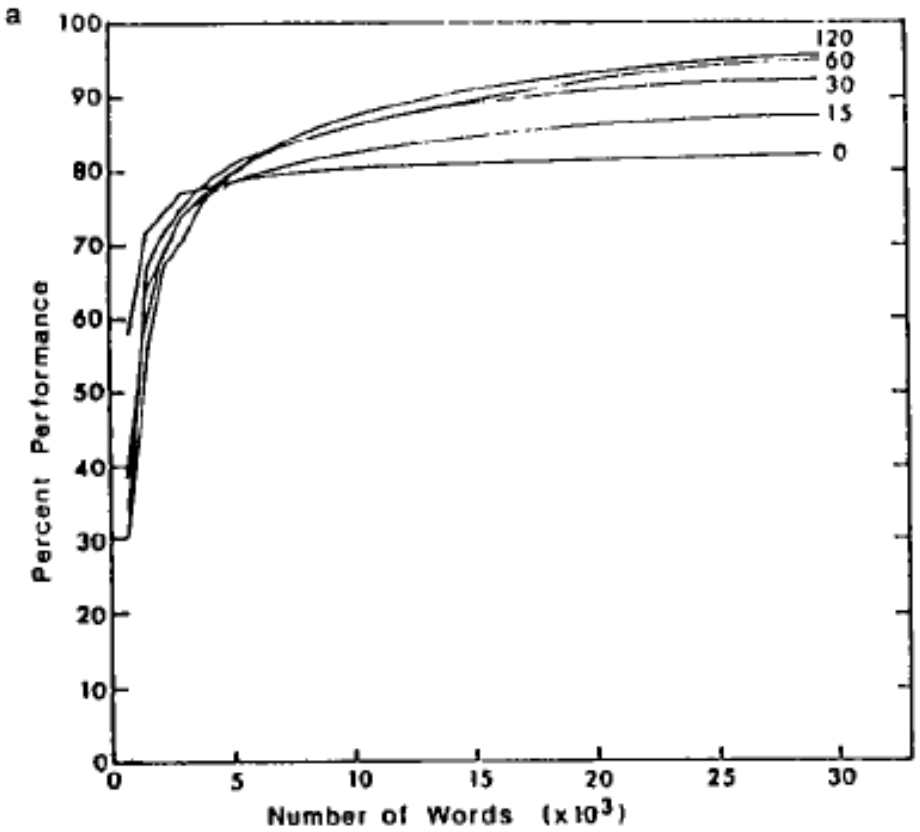
- Resistance to damage



- Recovery from damage



- Dictionary (1000 commonly used words)
  - Performance related to number of hidden units



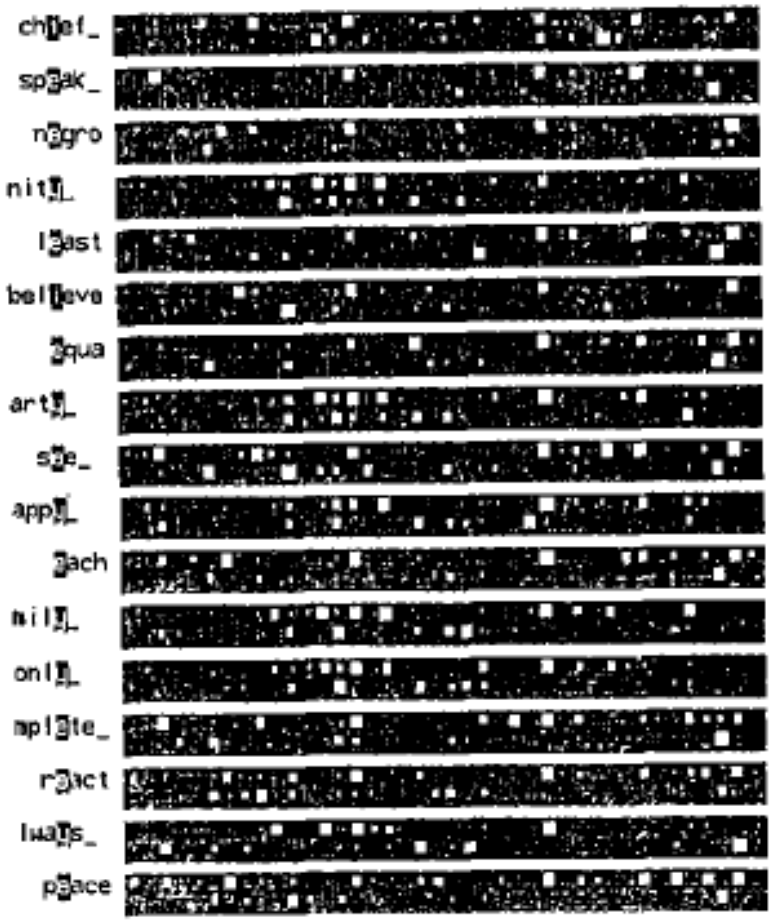
- Generalization ability:

  Average 77% on a 20,012 words dictionary; If continue to learn, reach 85% after first pass, 90% after five passes
- Performance related to size of input window:

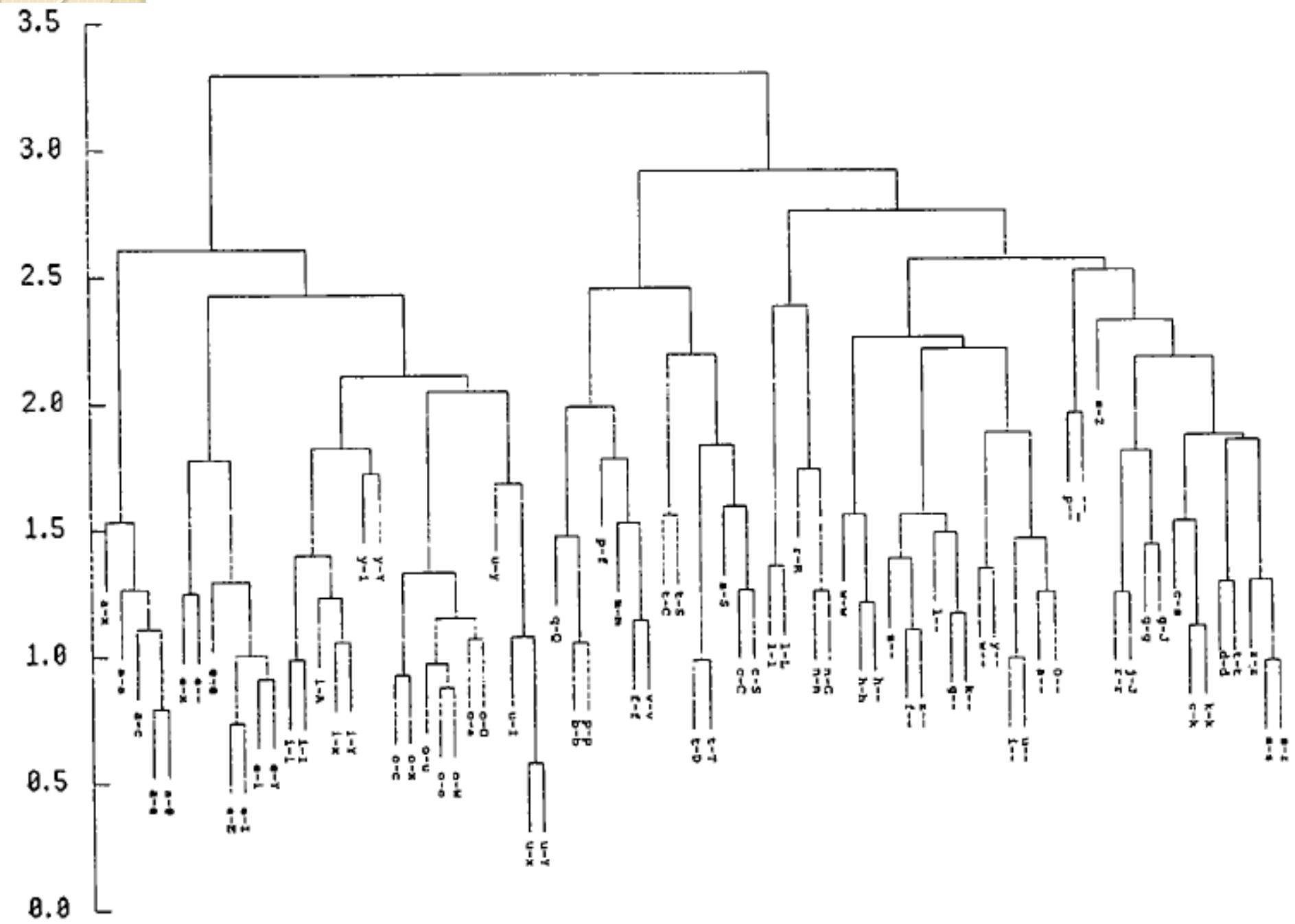
  Network with 11 input groups performs a little better than that with 7 input groups

Analysis of the Hidden Units

- The network had discovered some coding methods: (test case:7 input groups and 80 hidden units)



- Hierarchical clustering of hidden units for letter-to-sound correspondences:
  - Vowels and consonants were completely separated
  - Vowels were clustered mostly according to letter
  - Consonants were clustered mostly according to similarity of their sounds


3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0

Conclusion

- Could be trained on any language with the same set of letters and phonemes
- Learn from practice, following a power law
- Resistant to damage, due to distributed information
- Recovery from damage is quick