40
Introduction

(1986)
Terrence J. Sejnowski and Charles R. Rosenberg

NETtalk: a parallel network that learns to read aloud
The Johns Hopkins University Electrical Engineering and Computer Science Technical Report JHU/EECS-86/01, 32 pp.

English spelling is notoriously inconsistent. However, it does follow rules, although weak rules with exceptions and qualifications. This structure makes it well suited to a neural network, since networks are good at picking up statistical regularities, and are not bothered by occasional inconsistencies, as would be a more rigid rule based system.

As an example, consider the sentence "This is a test." The two letters "is" occur in both the first and second words, each time followed by a space, so the immediate context of 's' is identical. However, in the word "this" the 's' is unvoiced, that is, pronounced /s/. In the word "is" the 's' is voiced, that is, pronounced /z/.

In spite of its inconveniences, some have defended inconsistent spelling as making English related to both phonetic and an ideographic representations. Since there are so many dialects in English, it would be impossible to come up with a universal phonetic spelling scheme that would fit them all. Each dialect having its own phonetic spelling might signal the start of the fragmentation of English into separate languages. One of the editors of this collection of papers still recalls with dismay and amusement an article on rationalized English spelling read a number of years ago in a British publication. The author of the article proposed eliminating the terminal 'r' in words like "butter" or "corner" since it was not pronounced in modern English, ignoring, of course, the fact that most English speaking inhabitants of North America pronounce it. One man's phonetic spelling is another man's arbitrary representation.

NETtalk is a neural network that has learned how to read: specifically, it takes strings of characters forming English text and converts them into strings of phonemes that can serve as input to a speech synthesizer. The Digital Equipment Corporation sells a commercial product named "DECtalk" that also turns text into speech—hence the similarity of names. DECtalk is a complex rule based expert system developed over a number of years. Since NETtalk was a learning system, it potentially could learn to 'read' quickly and mechanically.

Sejnowski and Rosenberg used a three-layer feedforward network, with an input and output layer of units, as well as a layer of hidden units between the input and output layers. They have used both the Boltzmann machine learning algorithm (paper 38) and back propagation (papers 41 and 42) on this problem, with comparable results, though back propagation was faster to learn.

The input layer looked at a seven-character window of text. The network generated an output corresponding to the center character in the window. A window of this size seems to be enough to pronounce all except a few unusually difficult characters. Some pronounciation decisions require detailed knowledge of English, for example, to differentiate between the verb 'to lead' and the metal 'lead.'

The representation of characters at the input layer is localized, with one active unit

representing a character, space, or punctuation, for a total of 29 units, times 7 character positions, for a total of 203 input units. The number of hidden units varied from simulation to simulation, but for continuous speech 80 hidden units were used. There were 26 features (23 articulatory features, 3 for stress) represented in the output layer. Phonemes were represented by multiple simultaneously active output units. Simulations used about 300 units total with around 20,000 connections. The authors also did some experiments with distributed input representations, with results ultimately comparable to the local representation. In one simulation they used the system to develop its own preferred input representation.

Sejnowski and Rosenberg taught the system both isolated words and continuous speech. Error correction was used and the network was trained to give the correct output phoneme when presented with input character strings. The authors were able to find samples of transcribed speech in the literature on children's language and used this as the training set.

After roughly 12 CPU hours of training on a DEC VAX, NETtalk was producing phonemes from the training set correctly 95% of the time. New samples of similar text showed a drop in accuracy, but still at around the 80% level. Sejnowski has played a "developmental" tape of NETtalk's output at a number of meetings. It passes through a babbling phase, then starts approximating speech more and more closely. By the time learning stops, it is quite understandable, though it still makes occasional errors. The errors are usually close enough to the correct output so that the speech is still comprehensible.

Sejnowski and Rosenberg were interested in what was going on in the hidden layers, because of the theoretical reasons for believing that the hidden layer units are developing efficient representations. Units in the hidden layers each seem to respond to several input characters, and have no obvious interpretations. Later work has pursued this point. Some units seem to respond more strongly to some classes of characters (for example, vowels) than others, but nowhere is there strict localization. Representations in the hidden layers are distributed to some degree.

One of the reasons for the resurgence of interest in neural networks is their potential for practical applications. NETtalk is significant, because it seems to be close to a real application. It is a good example of a network being applied to a problem it can handle well: a system with regularities, but with exceptions to the regularities. In short, it is a cognitive problem of the messy kind that humans work with all the time and are good at coping with.

An expanded version of this technical report was recently published in the journal *Complex Systems* (Sejnowski and Rosenberg, 1987).

Reference

T. J. Sejnowski and C. R. Rosenberg (1987), "Parallel networks that learn to pronounce English text," *Complex Systems* 1:145–168.