From Image to Word: A Computational Model of Word Recognition in Reading

Gale Martin (GALEM@MCC.COM)
MCC
3500 Balcones Center Drive, Austin, Texas 78759 USA

Abstract

This paper describes a working, computational model of word recognition that combines a letter classification component with a component that segments the string of classified letters into words and uses a dynamic programming method for matching the words against a lexicon of over 2,800 words. The letter classification component is a neural network trained to classify, in parallel, inputs corresponding to 20x188 pixel array images of letter sequences, 14 or more letters long. Consistent with human capabilities, the system can classify all 14 letters at a level above chance, and on average, classifies the first 7 or 8 letters in the sequence correctly. Dictionary lookup improves classification accuracy by 1 character per image. The model is robust, having been trained and tested on the entire text of the book *The Wonderful Wizard of Oz*, printed in multiple fonts and in both mixed and upper-case letters. It provides a computation-level understanding of word recognition capabilities, in which errors are attributable to the theoretically inevitable difficulties associated with learning to classify large input patterns. The model mimics human capabilities for circumventing some of these difficulties by imposing constraints on fixation positions that reduce image variability.

Introduction

Marr (1982) proposed that cognitive processes can be studied at multiple levels. A *computation* level stresses the importance of understanding the nature of the task to be accomplished before proposing algorithms, representations, or physical hardware for accomplishing the task. Understanding the nature of the task involves, among other things, specifying a computational theory that defines the conditions under which it is possible to perform the task. Understanding a cognitive process at this level corresponds to specifying how these conditions are met when people perform the task. At the second level of analysis, the *algorithm* level, an algorithm for accomplishing the task is specified. Understanding a cognitive process in this case, corresponds to relating behaviors people exhibit when performing the task to behaviors exhibited by the more explicitly-specified algorithm. When people and the algorithm exhibit similar errors, or similar skills, in performing the task, it is often concluded that the human errors and skills are caused by characteristics of the algorithm, rather than the computation-level factors involved.

An example of an algorithm-level analysis of word recognition is McClelland & Rumelhart's (1981) Interactive Activation model. The model assumes that word recognition requires both letter classification and dictionary lookup components. The model simulates the dictionary lookup component. It is assumed that when a word is seen, a set of letter detectors is activated in parallel, continuously feeding letter and letter order information to a set of word detectors. Word familiarity is represented as learned associations between letter and word detectors. When the activation pattern arising from a set of letter detectors is consistent with one or more of these associations, activation is amplified, causing word detectors to fire more quickly, and through a backward flow of activation, causing the letters to be identified more quickly as well. Because the model exhibits many of the same phenomena people exhibit when they recognize words, including word frequency effects (Solomon & Postman, 1952); word superiority effects (Reicher, 1969); and pseudo-word superiority effects (Baron & Thurston, 1973) [1], the model can be said to explain these human phenomena in terms of the characteristics of the representations and algorithms of the model.

The present paper adopts a complementary, computation level understanding of word recognition that is based on a computational theory of classification learning. From this perspective, a classifier that converts the image of a text string to an hypothesis about the letter sequence depicted in the image, is characterized as a function that maps some population of inputs onto a corresponding population of outputs. People presumably acquire this function through a classification learning process. The computational theory of classification learning characterizes this learning process as function approximation. The learning system successively ap-

[1] Word frequency effects refer to findings that people identify high frequency words more quickly than low frequency words. Word superiority and pseudo-word superiority effects refer findings that people identify a letter presented in the context of a word or a wordlike letter string more quickly than they identify a single letter in isolation, even when the guessing advantage for words has been eliminated.

proximates the function that underlies the population of input-output pairs by using samples drawn from the population to search through a space of candidate mapping functions, eliminating those functions that are incompatible with the sampled pairs. It can be shown that without any natural constraints on the population, or biases in the function approximation process, the difficulty of classification learning increases exponentially with the sizes of the input and output patterns (Denker, et al, 1987; Geman, Bienenstock & Doursat, 1992). This boundary condition on learning is commonly referred to as the *curse of dimensionality*, and applies to all classification learning systems, human or machine. It arises because the larger the inputs and outputs to the system, the greater will be the potential number of candidate functions in the search space, and hence the longer the search, and the greater the number of input-output pairs that would have to be sampled to sufficiently approximate the function. Because of the exponential relations involved, the curse implies that *tabula rasa* (blank slate) classification learning is impossible for large inputs and outputs.

The curse would not necessarily be a problem relevant to word recognition if people identified words, one letter at a time, such that the input to the classifier was the relatively low-dimensional image of a single letter. However, a variety of evidence suggests that when people read, they classify a relatively long string of characters in parallel. Eye movement studies indicate that people can at least partially classify as many as 14 letters per fixation, and completely identify an average of 7 or 8 letters per fixation (McConkie & Rayner, 1975; Rayner, Well & Pollatsek, 1980; Rayner, Well, Pollatsek & Bertera, 1982). Word superiority and related effects indicate that letter classification occurs in parallel (Baron & Thurston, 1973; Blanchard, McConkie, Sola & Wolverton, 1984; Reicher, 1969). There is also evidence that the parallel nature of the process is not due to words being read on the basis of word shape detectors, since printing words in aLtErNaTiNg cases, which eliminates the familiarity of a word's shape, has relatively small effects on word recognition (McClelland, 1976).

These data indicate that the images input to the letter classification process that underlies human reading are quite large, and that outputs of the process represent a long string of characters, usually corresponding to multiple words. When combined with the curse of dimensionality principle, this implies that people would not be able to learn to classify these images of letter sequences unless constraints exist to limit the variability of the to-be-classified images images, and/or learning is biased to exclude candidate mapping functions from the search space on an *a priori* basis. This suggests that we may understand reading better by understanding the nature of these constraints that make letter classification learning possible. Toward this end, the present paper develops a working computational model of word recognition that uses human-like natural constraints to learn to classify images of long letter sequences.

Previous Work

In a previous paper (Martin, 1996) I supported this computation-level perspective of letter classification by training neural networks to classify images of letter sequences. The goal was to determine the impact on letter classification learning of increasing the width of the input images, and the number of to-be-classified letters, and to determine the corresponding impact of natural constraints on the variability of these images. Note that the point of this work was not to support a claim that people and the networks were necessarily similar at an algorithm level, but rather that both systems were governed by the same computation-level limitations on classification learning, and that both could benefit from the same types of constraints. In other words, the neural networks provide a measure of both the problems associated with high dimensional inputs and outputs, and the potential utility of the constraints in overcoming these problems.

The study produced a number of results. Consistent with the curse, networks that were trained on images of single characters had no difficulty at letter classification learning, but as the image width increased from 20 to 80 pixels and the number of to-be-classified characters increased from 1 to 4 characters, catastrophic effects on learning occurred. One natural constraint that may reduce such difficulties is the regularity in fixation positions that characterizes human reading. People fixate most often at a *preferred viewing location*–slightly to the left of the middle of a word (Rayner, 1979). It is also the case that people identify a word more quickly when the eyes fixate near to this location (O'Regan & Jacobs, 1992). Such regularities may reduce image variability sufficiently to overcome some of the problems associated with the curse. The original nets did not have the benefit of such constraints, as the input images were generated by fixating at each character position within a word. Simulating these regularities resulted in networks that performed as well, or better than the networks trained to classify single-letter images; thus overcoming the curse's negative effects for this size of inputs and outputs.

A third set of simulations addressed the role played by constraints on letter sequences, since words are composed of only a small subset of all possible letter sequences. This role was assessed by examining the extent to which the trained networks exhibited word superiority effects, pseudo-word superiority effects, and word frequency effects similar to those exhibited by people. Such a similarity would indicate that the nets had become specialized for classifying familiar letter sequences at the expense of all possible letter sequences. The nets exhibited

these effects, thus supplying evidence that constraints on letter sequences also facilitate letter classification learning. Note that these results provide an explanation of word frequency effects and word- and pseudo-word superiority effects that differs from the explanation provided by McClelland & Rumelhart (1981) in their Interactive Activation Model, both in the level of the explanation: Computation vs algorithm, and in the source of the effect: Letter classification learning vs dictionary lookup.

Current Work

The current work extends this conception by exploring (1) the extent to which increasing image variability to more realistic levels hinders classification learning, and (2) the extent to which applying post-processing constraints can make up for such deficiencies in classification learning. The model developed previously minimized the variability of to-be-classified images relative to what people face when they learn to read. The to-be-classified images depicted sequences of about 4 letters, as compared to the sequences of 14 letters that people classify, and a highly simplified model for generating eye fixation positions was used. Adding new sources of variability will increase the difficulty of classification learning, and reduce classification accuracy. Such decreases in classification accuracy may not be fatal, however, if post processing mechanisms, such as dictionary lookup, integration across fixations, and syntactic and semantic processing, provide sufficient constraints on classification to correct errors. A more complete computation-level understanding of reading should describe the interplay of these positive and negative influences on letter classification accuracy. The work described here takes a first step in this direction by more accurately modeling the image variability with which the human reading system must contend, and by incorporating a dictionary lookup component.

Image Variability

The variability of to-be-classified images was increased by extending image width from that sufficient to cover letter sequences containing about 4 letters to that sufficient to cover about 14 letters, and by more accurately approximating the fixation position regularities that characterize reading. The original research (Martin, 1996) used a simplified simulation of these regularities, positioning each input image with respect to the center of the 3rd letter in each word containing 3 or more letters. Actual fixation locations are likely to impose greater image variability, and thus greater learning difficulties. Rather than positioning all images with respect to a fixed location in all words, people tend to base fixation locations on word length, at a position slightly to the left of the middle of a word, and fixations are better described as a probability distribution around this location (Rayner, 1979).

Two networks were trained from scratch, one using the previous simplified simulation of fixation regularities that was independent of word length, and the other using the method based on word length [2] The impact of having a probability distribution of fixation positions around a given position was assessed by cloning the network trained with word-length-based fixation points, and then retraining it with images generated as follows. On a randomly-chosen one third of the exposures, the window was shifted to the left or right one character. The result of these endeavors led to the creation of three neural networks, trained on images of increasing variability. The network that was trained with images generated with the constant fixation position on the 3rd letter of a word, presumably encountered the least image variability. The network trained with the noisy, word-length-based fixations presumably encountered the greatest image variability.

Dictionary Lookup Component

In their Interactive Activation model, McClelland & Rumelhart (1981) assumed that the string of letters output from letter detectors corresponded to a single word, so that dictionary lookup simply involved matching this string against the internal lexicon. However, if the output of the letter classification task is an hypothesis about the identities and order of a sequence of 14 letters, then dictionary lookup must also involve segmenting the string into words. The present work integrated dictionary lookup and segmentation to explore the interplay between letter classification and dictionary lookup.

The dictionary lookup component was an extension of one developed in previous work (Martin & Talley, 1995) in which a two-tiered dynamic programming method was used for word segmentation and dictionary lookup to improve the accuracy of a handwriting recognition system. Dynamic programming refers to a general class of efficient search algorithms for use where the elements of the problem have an inviolate order, as is the order of letters within a word, and where it is possible to define a monotonically increasing decomposable objective function that can be minimized over the length of the sequence.

The present approach departs from this earlier work with respect to the method used to identify word boundaries. The output of the letter classification system is a sequence of vectors that can be divided into subsequences corresponding to words. Each vector consists of the activation values of 27 output nodes. The present

[2] It was only possible to train two networks because each net required 6 months to train, running on a relatively fast sparc10 machine. However, previous experience with large nets and large training samples, has indicated little variability across training and generalization. In addition, the initial random states of the two networks were identical.

approach used the activation values of the output nodes corresponding to a between-word "space" to determine a candidate set of possible word boundaries. The intent was to err on the side of proposing too many word boundaries. If the output nodes for "space" had an activation value greater than .1, a possible word segmentation point was recorded. A list of possible sub-sequences corresponding to words was generated by starting at the leftmost character position and ending at each possible word boundary. Each of these sub-sequences was submitted to a dynamic programming function that generated the best match in the dictionary of about 2,800 words from the story. Then, the best match across all of this list was chosen, and the corresponding word replaced the first n letters of the classified string, where n is the number of letters in the word. A space replaced the next letter in the classified string, and the process was repeated again until all 14 character positions in the classified string had been replaced.

Training and Testing Materials

The training and testing materials were the same as those used in the previous study. They were generated from the book *The Wonderful Wizard of Oz* by L. Frank Baum. Text line images were created from 120 pages of text, corresponding to about 160,000 characters, over 30,000 total words and about 2,800 different words. To approach the real-world variability of text images in reading, the text was printed in 3 different type fonts, and in either all upper case letters or the original mix of lower and upper case letters (see Figure 1). The set of text line images were equally divided into the six different font/case conditions, and each of these was equally represented in separate training and test samples. The test samples were sub-divided into two sets, referred to as the *test* and *validation* sets. The first of these was used to monitor generalization performance throughout training; the validation set was only used in testing after training had stopped. The training set contained about 13,600 distinct images of letter sequences comprised of 14 or more letters. The test and validation sets each contained about 1,200 such images.

Dorothy lived in the midst of the great Kansas Prairies.
DOROTHY LIVED IN THE MIDST OF THE GREAT KANSAS PRAIRIES.
Dorothy lived in the midst of the great Kansas Prairies.
DOROTHY LIVED IN THE MIDST OF THE GREAT KANSAS PRAIRIES.
Dorothy lived in the midst of the great Kansas Prairies.
DOROTHY LIVED IN THE MIDST OF THE GREAT KANSAS PRAIRIES.

Figure 1: Samples of type font and case conditions

Neural Network Architecture

The neural network simulations all used a common type of architecture that was an extension of local receptive field, shared weight architectures (see Figure 2) used successfully in a number of Optical Character Recognition (OCR) systems (LeCun, et al, 1990; Martin & Pittman, 1991), and used in the previously mentioned (Martin, 1996) study. In all of these cases, the learning algorithm was backpropagation (Rumelhart, Hinton & Williams, 1986).

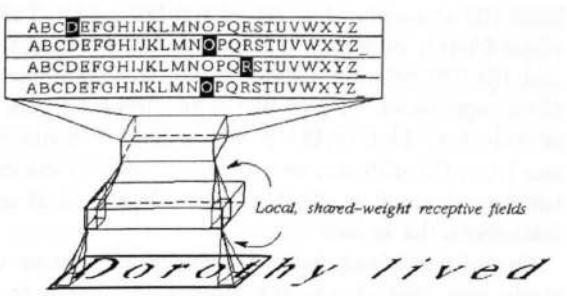

Figure 2: Neural network architecture

The inputs to these earlier versions of the architecture were images of single characters, but in the Martin (1996) study, the input images depicted at least k letters, where $k = 1$, 2, 3, or 4, depending on the specific network, and the outputs corresponded to a vector of the 26 letters (A-Z) and a space, for each of the k possible letter positions. Hidden nodes receive input from a local region (for example, a 6x6 area) in the layer below. Hidden layers are visualized as cubes, made up of separate planes. Hidden nodes within a plane share weights, in the sense that corresponding weights in the nodes' receptive fields are randomly initialized to the same value and updated by the same error, so that different hidden nodes within a plane learn to detect the same feature at different locations. Different feature detectors emerge from hidden nodes within different planes, due to different random initializations of the weights. There are two hidden layers of this type. Output nodes are connected to all nodes in the previous layer, but not to each other. The output vector consists of one set of 27 elements, to represent the letters A-Z and a space, per character position. The same basic architecture can be altered to classify longer sequences by expanding the widths of the inputs and outputs, so that the input window is wide enough to cover the k widest characters ("WWWW") for $k = 4$. The image of a string of narrow characters will therefore depict additional characters to the right, which the net must learn to ignore. Hidden layers are also expanded horizontally, increasing the number of feature detectors, but not necessarily the number of different types of features detected, which would require a vertical expansion of the cubes. As before, networks were trained until training accuracy ceased to improve by at least a tenth of a percent over 5 training epochs, or until generalization performance began to consistently decline

over training epochs (indicating that the net had begun to over-generalize).

The input images, output vectors, and the networks were larger than in the previous study. Whereas previously, the large images containing at least 4 to-be-classified letters consisted of a 20x80 array (1600 pixels); the present study used a 20x188 array (3760 pixels) depicting 14 or more letters. The output vector increased from 108 elements (4 x 27) to 378 (14 x 27). The previous 4-letter nets had 8152 nodes, 581,904 connections and 104,976 different weight values, with 18 unique features represented in each of the two hidden layers. The current nets had 18,481 nodes, 2,399,760 connections, and 1,231,020 different weight values, with 15 unique features represented in the first hidden layer and 24 unique features in the second.

One of the things discovered during the course of this study was that it was not possible to begin training the nets to classify all 14 letters from scratch, because they reached saturation activation levels early in training. Varying network initialization and learning parameters did not eliminate this problem. Training the network to at first only classify the first (leftmost) letter in the image, and then adding training on the other letters successively over time, did eliminate the problem.

Performance of the Model

The results of these efforts demonstrate that it is possible to build a working model of word recognition that incorporates very wide input images, the types of constraints that characterize eye fixations during reading, and integrated dictionary lookup and word segmentation components. Consistent with data on how people read, all three versions of the networks classify letters at all 14 letter positions above the level of chance, and on average classify the first 7 or 8 letters in a sequence.

Figure 3 illustrates the percentage of characters correctly identified, in the generalization test set, as a function of position and type of fixation constraint by the three nets, before any word matching was attempted (lightest bars = constant fixation, black bars = fixation position based on word-length, dark gray bars = word-length + noise). Results were equivalent for the validation test set. All of the networks show a decline in letter classification as one moves to the right, as we might expect, since letter position variability increases with increased distance from the fixation point. The increases in image variability caused by the reduced fixation constraints also take their toll on classification accuracy. Neither of these effects is so catastrophic however, that the nets fail to exhibit performance comparable to that of people when they read.

Figure 4 illustrates the benefits and costs of applying the dynamic programming based word dictionary lookup procedures. It shows that character position accuracy rates for the network trained on images with noisy word

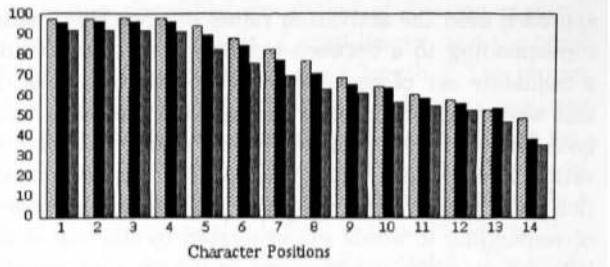

Figure 3: Percent characters correct as a function of position and type of constraints on fixation positions

length fixation positions both before (gray bars) and after dictionary lookup (black bars). Although the overall accuracy rates for this network are lower than for the other two nets, the pattern of performance before and after dictionary lookup is the same. Sometimes the dictionary lookup component helps and sometimes it hurts, on a character by character basis. The effects are not dramatic except for the rightmost characters in the string, where the dictionary lookup tends to hurt rather than help performance. This component can probably be optimized further, though several different approaches were tried, with the results from the best version of the system reported here.

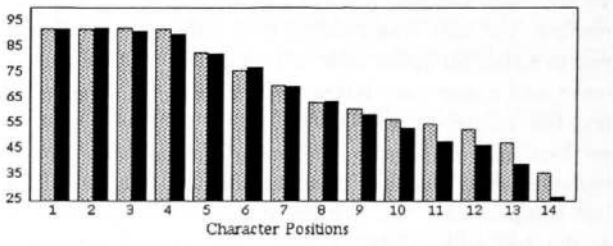

Figure 4: Percent characters correct by position before(gray) and after(black) dictionary lookup.

Table 1 describes the performance of the system from the perspective of the first n letters correctly identified on average. Remember that the human data indicate that people can identify the first 7 or 8 letters per fixation on average. These data show that the dictionary lookup improves the average number of correctly classified consecutive characters by about 1 character. All of these values for the different types of fixation position constraints are comparable to the average number of consecutive characters correctly identified by people, as measured by the average size of forward saccades. Because the network trained on images positioned with respect to word length with noise added, are most reflective of the types of positioning constraints used by people when they read, and this network exhibits comparable levels of performance to that of people when they read.

it corresponds to the best computational model of word recognition.

Table 1. Average number of leftmost characters correctly identified in sequence

Fixation Location Type	Before Dictionary Lookup	After Dictionary Lookup
Constant	8.8	9.9
Word-Length-Based	8.1	9.5
Word-Length-Based+Noise	7.0	8.0

Discussion

This model, and the accompanying computation level understanding of word recognition are significant for at least two reasons. First, the computation-level understanding provides a theoretically-driven basis for proposing one source of reading disabilities and developmental stages. To the extent that poor readers have problems learning to classify letters, they should not be able to identify as many characters per fixation, and they should exhibit irregular fixation patterns. Empirical data support these expectations (Rayner, 1986; Rayner & Pollatsek, 1989). This suggests that we may gain a better understanding of reading disabilities and developmental differences in reading by examining whether or not the source of some reading problems lies with problems in letter classification learning. Second, the model paves the way for building ever more accurate working models of human reading, by incorporating components such as foveal warping of the input images, integration across fixations, and automated generation of saccades (Martin, Rashid & Pittman, 1993). The impact of such additions can be evaluated via their impact on classification learning and accuracy.

References

Baron, J. & Thurston, I. (1973) An analysis of the word superiority effect. *Cog. Psych., 4*, 207-208.

Blanchard, H., McConkie, G., Zola, D., & Wolverton, G. (1984) Time course of visual information utilization during fixations in reading. *Jour. of Exp. Psych.: Human Perc. & Perf., 10*, 75-89.

Denker, J., Schwartz, D., Wittner, B., Solla, S., Howard, R., Jackel, L., & Hopfield, J. (1987) Large automatic learning, rule extraction and generalization, *Complex Systems, 1*, 877-933.

Geman, S., Bienenstock, E., and Doursat, R. (1992) Neural networks and the bias/variance dilemma. *Neural Computation, 4*, 1-58.

Marr, D. (1982) *Vision* San Francisco: W. H. Freeman

Martin, G. L. (1996) The impact of letter classification learning on reading. *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum. 171-176

Martin, G. L. & Pittman, J. A. (1991) Recognizing hand-printed letters and digits using backpropagation learning. *Neural Computation, 3*, 258-267.

Martin, G. L., Rashid, M., & Pittman, J. A. (1993) Integrated segmentation and recognition through exhaustive scans or learned saccadic jumps. In *Advances in Pattern Recognition Systems Using Neural Network Technologies*, I. Guyon and P. S. P. Wang (Eds). World Scientific.

Martin, G. L. & Talley, J. (1995) Recognizing handwritten phrases from U. S. Census Forms by combining neural networks and dynamic programming. *Journal of artificial neural networks, 2*, 167-193.

McClelland, J. L. (1976) Preliminary letter identification in the perception of words and nonwords. *Jour. of Exp. Psych.: Human Perc. & Perf., 2*, 80-91.

McClelland, J. & Rumelhart, D.(1981) An interactive activation model of context effects in letter perception: Pt. 1 *Psych. Rev., 88*, 375

McConkie. G. & Rayner, K. (1975) The span of the effective stimulus in reading. *Perc. & Psychophysics, 17*, 578-586.

O'Regan, J. & Jacobs, A.(1992) Optimal viewing position effect in word recognition. *Jour. of Exp. Psych.: Human Perc.& Perf., 18*, 185-197.

Rayner, K. (1979) Eye guidance in reading. *Perception, 8*, 21-30.

Rayner, K. (1986) Eye movements and the perceptual span in beginning and skilled readers. *Jour. of Exp. Child Psych., 41*, 211-236.

Rayner, K. & Pollatsek, A. (1989) *The Psychology of reading.* Prentice

Rayner, K., Well, A. & Pollatsek, A. (1980) Asymmetry of the effective visual field in reading. *Perc. & Psychophysics, 31*, 537-550.

Reicher, G. (1969) Perceptual recognition as a function of meaningfulness of stimulus material. *Jour. of Exp. Psych., 81*, 274-280.

Rumelhart, D., Hinton, G., & Williams, R. (1986) Learning internal representations by error propagation. In D. Rumelhart and J. McClelland, *Parallel Distributed Processing, 1.* MIT Press.

Solomon, R. & Postman, L. (1952) Frequency of usage as a determinant of recognition thresholds for words. *Jour. of Exp. Psych., 43*, 195-210.