Regularities in a Random Mapping from Orthography to Semantics

Daniel S. Clouse and **Garrison W. Cottrell**
Computer Science & Engineering 0114
University of California, San Diego
La Jolla, CA 92093
{dclouse,gary}@cs.ucsd.edu

Abstract

In this paper we investigate representational and methodological issues in a attractor network model of the mapping from orthography to semantics based on [Plaut, 1995]. We find that, contrary to psycholinguistic studies, the response time to concrete words (represented by more 1 bits in the output pattern) is slower than for abstract words. This model also predicts that response times to words in a dense semantic neighborhood will be faster than words which have few semantically similar neighbors in the language. This is conceptually consistent with the neighborhood effect seen in the mapping from orthography to phonology [Seidenberg & McClelland, 1989, Plaut et al., 1996] in that patterns with many neighbors are faster in both pathways, but since there is no regularity in the random mapping used here, it is clear that the cause of this effect is different than that of previous experiments. We also report a rather distressing finding. Reaction time in this model is measured by the time it takes the network to settle after being presented with a new input. When the criterion used to determine when the network is "settled" is changed to include testing of the hidden units, each of the results reported above change the direction of effect – abstract words are now slower, as are words in dense semantic neighborhoods. Since there are independent reasons to exclude hidden units from the stopping criterion, and this is what is done in common practice, we believe this phenomenon to be of interest mostly to neural network practitioners. However, it does provide some insight into the interaction between the hidden and output units during settling.

Introduction

The publication of the Seidenberg and McClelland (1989) model of naming set into motion an extensive debate on the nature of the processes used in the recognition and pronunciation of English words. A number of existing models relied on two separate mechanisms: a rule-following mechanism for the pronunciation of regular words and novel nonwords, and a look-up mechanism for the pronunciation of irregular words. Seidenberg and McClelland claimed that a single mechanism in the form of a neural network could perform both rule-like and exception mappings, and account for regularity effects seen in reaction time studies using the lexical decision and naming tasks. They showed that their parallel distributed processing model demonstrates a so-called neighborhood effect in which the reaction-time to a word is decreased if it has a number of neighbors with similar orthographic to phonological mappings.

The principal attribute of a neural network which makes it a good model of regularity effects in reading is that of neighborhood training. When the input and output representations are chosen appropriately, training a single input/output pattern improves the performance not only of that pattern, but of similar patterns also. Thus, a rule-like behavior is induced which causes novel inputs to be pronounced in a manner similar to human subjects, and also accounts for the neighborhood effect.

Since that time, the focus in modeling lexical access effects has shifted from feed-forward to attractor networks, recurrent networks which are trained to settle to a stable output. Using attractor network models, A number of experiments have been reported which lend credence to the idea that many of the effects seen during lexical access with human subjects are naturally modeled using attractor networks.

Plaut et al. (1996) showed that the regularity effect demonstrated by [Seidenberg & McClelland, 1989] holds for attractor networks as well as for feed-forward networks. Plaut (1995) demonstrated semantic and associative priming in an attractor network model which implemented a random mapping whose intent was to simulate the mapping from orthography to semantics. Plaut and Shallice (1993) demonstrated an attractor network model which, when damaged showed a number of symptoms of deep dyslexia.

The purpose of our research is to look for regularities in the behavior of attractor networks implementing a mapping from orthography to semantics. The principle aspect of the mapping from orthography to phonology, that of regularity, is lacking in the mapping from orthography to semantics. Even though two words which are spelled similarly have a good chance of having similar pronunciations, there is little reason to believe that they will have similar meanings. In spite of this lack of regularity in the mapping, we show that regularities still persist in the behavior of attractor network models of this mapping.

We present a comprehensive set of network simulations in which we independently vary several factors:

- the frequency with which an input/output pattern is seen during training
- the semantic neighborhood density, the proximity and number of output patterns which are similar to a pattern of interest,
- the number of bits which are on in an output pattern.

The first factor, training frequency, we equate with the frequency of occurrence of a word in speech or text for human subjects. Since word frequency is perhaps the most reliable determiner of reaction time in reading, any study which left out this factor would be incomplete.

The second factor, semantic neighborhood density, is one possible source of a neighborhood effect in networks which are trained without a regular mapping. Evidence for such an effect without a regular mapping would have interesting implications for our understanding of the regularity effect in reading. It would also make a unique prediction of attractor networks. We are aware of only one paper that looks for an effect of semantic neighborhood [Buchanan et al., 1996]. This paper reports the results of testing three deep dyslexic patients in a naming task. Reading errors of one of the patients showed significant correlation with the density of the semantic neighborhood of the stimulus words. Though this study counts errors rather than looking at reaction times as does our model, by showing some effect of a semantic neighborhood, it does lend credence to the concept.

The third factor, output pattern bit density, has been used to encode concreteness of word meaning in a number of connectionist models [Cottrell & Plunkett, 1995, Plaut & Shallice, 1993]. In this research, patterns which have more bits on are taken as having more features instantiated, thus representing more concrete concepts.

A number of psycholinguistic studies show an effect of concreteness on reaction time. A significant facilitation in reaction time for either concrete or highly imageable words has been reported both for the lexical decision task [James, 1975, Whaley, 1978, Kroll & Merves, 1986, de Groot, 1989], and the naming task [de Groot, 1989]. In general, the effect seems to be stronger in the lexical decision task than in the naming task.

All of these lexical decision studies which found the main effect and which included statistics on interactions[1] reported a significant interaction between frequency and concreteness or imageability such that the concreteness effect was stronger for low frequency words. Indeed, two of the studies [James, 1975] [Kroll & Merves, 1986] report no concreteness effect for high frequency words while reporting a significant effect for low frequency.

De Groot (1989) does not report a significant interaction between frequency and imageability, but analysis of [Strain et al., 1995] shows a nonsignificant trend in the same direction as the lexical decision studies. In their first experiment, which is the only experiment which includes high frequency words, the facilitation for high imageable words over low imageable words is 24.5 msec. for low frequency words, but only 6 msec. for high frequency words. Most of this difference comes from words with exceptional pronunciations. Their second experiment shows a significant interaction between the regularity of pronunciations and imageability such that facilitation for highly imageable words is much stronger for exception words.

Together these results suggest that semantics plays a role in lexical decision and naming, and also in the pronunciation of exception words, as has been suggested by [Plaut & Shallice, 1993, Plaut et al., 1996].

[1] One study [Whaley, 1978] reported multiple regression statistics and therefore did not report statistics on any interactions.

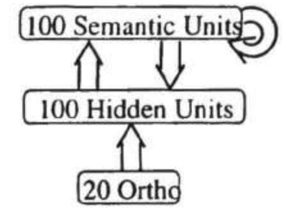

Figure 1: Network Architecture

Network Simulations

To come to a better understanding of the factors that affect settling time in an attractor network model which maps from orthography to semantics, we ran a number of simulations. In this section, we explain our simulation methods and present results. These experiments represent a parametric study of the effects of certain kinds of semantic structure on lexical access.

Network Architecture

Figure 1 shows the network architecture used in our simulations. There were 20 inputs, 100 hidden units, and 100 outputs units with recurrent connections within the output layer and back to the hidden layer. This architecture and much of the training method are taken directly from [Plaut, 1995]. Much of the technique is similar to that used in experiment 3 of [Plaut et al., 1996].

In order to approximate continuous time, continuous network units are used. The following formula describes the activation behavior of these units. The activation of unit j at time t, $s_j^{[t]}$, is simply the squashed net input of unit j at that time. The net input $x_j^{[t]}$ is a time varying average:

$$x_j^{[t]} = \tau \sum_i s_i^{[t-\tau]} w_{ij} + (1 - \tau) x_j^{[t-\tau]}$$

The τ parameter regulates the granularity of discrete time used to approximate continuous time by specifying the smallest tick of time the network operates on. We will refer to these smallest units of time as ticks, and the actual units of time we are approximating, t, as simulation time units. In the graphs below, settling time is in terms of simulation time units.

The recurrence and the use of continuous units may seem overly complex, since the problem of mapping between our representations of orthography and semantics can be learned by a feed-forward network. We chose this methodology because we are interested in an accurate simulation of the time it takes the network to settle to the correct output pattern after the input is changed, as a measure of reaction time to the stimulus.

Training Patterns

The network was trained to map from "orthography" to "semantics." We include quotation marks around these two labels because we do not attempt to set up a correspondence between a word representation and any word of English or any other natural language. Instead, following (Plaut 1995), we use random 0/1 bit patterns to represent both the orthography and semantics of each word. The input "orthographic" patterns were uniformly distributed, sparse bit patterns. The probability of any bit being on is 0.1. We imposed a more interesting structure on the output "semantic" representation to capture the concepts of imageability, and semantic neighborhood.

Highly imageable or concrete words are often characterized in the literature as having a richer representation [Breedin et al., 1994]. This may take the form of more semantic features for concrete words, or more sensory connections for highly imageable words. In the context of our simulations, we capture this concept by varying a single parameter, $P(ONE)$, the probability of a bit being a one in the output representation. Representations with more features, or 1 bits, we call concrete. Those with fewer bits, we call abstract. The $P(ONE)$ parameter is set to 0.5 for concrete words, and 0.1 for abstract words. Note that similar representations are assumed elsewhere in the literature [Cottrell & Plunkett, 1995, Plaut & Shallice, 1993].

To capture the concept of a semantic neighborhood, in which a number of words have similar meanings, we want to generate output patterns such that two patterns from the same semantic neighborhood have more bits in common than two patterns from different neighborhoods. We capture this concept using the $P(FLIP)$ parameter. This parameter is the probability that any single bit in the semantic representation will change, or "flip" between two neighbors. This controls how tightly packed the patterns in a neighborhood will be. We use the terms "dense" and "sparse" to distinguish low and high $P(FLIP)$ values respectively.

All patterns in a neighborhood are generated from a single prototype pattern which is not included in the final set of training patterns. The process we use to produce these patterns is designed so that:

- the probability that a bit in any pattern is a 1 is independent of all other bits, and is given by $P(ONE)$,
- the probability that two corresponding bits between a pair of patterns in a neighborhood are different is independent of all other bits in the pattern and is given by $P(FLIP)$.

First a prototype pattern is generated where each bit is randomly set to 1 with probability $P(ONE)$. To generate a single exemplar pattern from the prototype, we randomly and independently decide whether or not to flip each bit. This decision is made in two stages. First, with probability p, the bit is designated to be resampled. Next, each bit so designated is set to 1 with probability $P(ONE)$ similar to the process for the original prototype. The value of any bit which is not designated to be resampled is left unchanged.

The value of the probability p is chosen so that the probability of corresponding bits being different between two patterns in the neighborhood will be $P(FLIP)$. The value of p necessary to achieve a given $P(FLIP)$ varies depending upon the $P(ONE)$ parameter. That is, with one setting of p for two different settings of $P(ONE)$, the expected distance between patterns in a neighborhood would change. Hence we had to compute the proper value of p for each value of $P(FLIP)$ so that the expected distance between two patterns in a neighborhood is independent of our choice of $P(ONE)$.

A third parameter, $NNBORS$, is the total number of patterns in the neighborhood. Note that $NNBORS$ is independent of $P(FLIP)$. For instance, it is possible to have a tightly packed neighborhood containing only a few patterns (small $P(FLIP)$, small $NNBORS$), or a loosely packed neighborhood containing many patterns (large $P(FLIP)$, large $NNBORS$). Despite the two parameters being independently controlled, we do expect these two parameters to have similar effects on the simulation, so we will call a neighborhood containing many patterns "dense" and one with few "sparse".

So far we have mentioned three parameters used to generate our random semantic representations: The $P(ONE)$ parameter determines the concreteness or imageability of a word, while $NNBORS$ and $P(FLIP)$ determine how many semantic neighbors a word has and how close together they are. There is a fourth and final parameter attached to each word, the $FREQ$ parameter. This determines the relative frequency of the word, or how often it is seen during training.

Table 1 records the various levels used for each of the four parameters. Our base simulation was a $2 \times 2 \times 2 \times 2$ experiment with 6 patterns in each cell for each network resulting in 96 word patterns for each network.

Table 1: Summary of Simulation Parameters.

$P(ONE)$	Probablity that a bit in the pattern is a one	0.1 = abstract 0.5 = concrete
$P(FLIP)$	Prob. that bit flips between 2 patterns in n'hood	0.05 = dense 0.15 = sparse
$NNBORS$	Number of patterns in the neighborhood	6 = dense 2 = sparse
$FREQ$	Relative number of presentations in training	1 = low 4 = high

Training Method

We trained and tested ten separate networks, each with different initial weight values, and different sets of patterns generated using the parameters described above. Our networks were trained using Pearlmutter's method for continuous, recurrent back-propagation through time [Pearlmutter, 1989].

During training, before the presentation of a new input pattern, the activation of each unit ($s_j^{[t]}$) was set to 0.3, and the time-averaged net input ($x_j^{[t]}$) was set to -0.8473, which is the inverse-sigmoid of the initial activation of 0.3. This initial activation value was chosen as it is the average value of all output pattern bits in the training set.

On each presentation, mean 0 Gaussian noise with standard deviation 0.05 was used to corrupt the input portion of the training pattern. The activation values of the input layer were set at time $t = 0$ to this corrupted pattern, and activation was propagated forward using the formula of the previous page, until simulation time $t = 4$. During this time the input layer activations were clamped to their noise-corrupted values so they did not change between time 0 and 4.

During the back propagation phase, cross entropy error [Hinton, 1989] between the activation of the output layer and the output training pattern was injected between time $t = 2$ and 4. The error accumulated during this period was back-propagated to time 0, but no farther.

Gradient descent with momentum was used to calculate new weights, and the weight changes were applied before the next pattern was presented. The learning rate was 0.005, with momentum 0.8. Each network was trained for 1000 epochs at $\tau = 0.2$. This was followed by "annealing training" consisting of 60 epochs at $\tau = 0.05$, and 40 more epochs at $\tau = 0.01$.

Simulation 1 - Results & Discussion

After training, we assessed the settling times for each of the 96 patterns for each of the ten networks. Each network unit was initialized to its starting activation of 0.3, and allowed to settle. In this simulation, the network was considered settled when no output unit changed by more than 0.001 in a single time tick. This is the method used in [Plaut, 1995], and is similar to that of [Plaut et al., 1996].

At this point, for every training pattern, the sum squared error on the final activation of the network is within $\sqrt{0.5}$ of the training pattern, which is adequate to assure us that no bit is off by more than 0.5.

Figure 2 plots the main effects on settling time for the four parameters. In these graphs, the dependent axis plots the average amount of simulation time the network took to settle to a stable semantic output. The error bars plot one standard error on each side of the mean. Mean settling times are plotted after various amounts of training ranging from 200 to 1000 epochs. The weights used in these graphs were derived by doing annealing training starting from the weights produced on the same set of 10 networks after 200, 400, 600, 800, and 1000 epochs of training.

We performed $2 \times 2 \times 2 \times 2$ repeated measures ANOVAs on the settling times at 1000 epochs. Each of the 10 networks was treated as a separate subject, and all 4 factors were treated as within subject sources of variance. The results show that all the main effects, except for $NNBORS$, are significant. Frequent words settle faster than infrequent ($F_{FREQ}(1,9)$

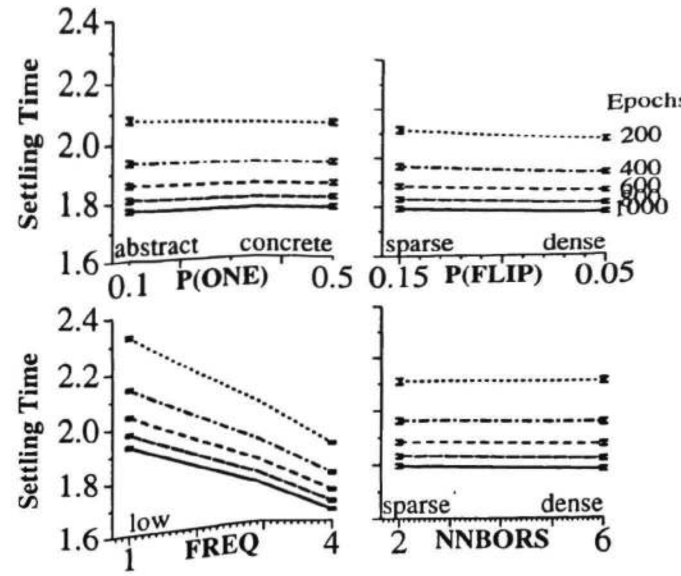

Figure 2: Main Effect for Simulation 1

$= 897.7$, $p < 0.001$). Abstract words are faster than concrete ($F_{P(ONE)}(1,9) = 12.1$, $p = 0.007$). The influence of the $P(FLIP)$ parameter is to make words in dense semantic neighborhoods faster ($F_{P(FLIP)}(1,9) = 6.90$, $p = 0.027$). The direction of influence for $NNBORS$ is consistent with $P(FLIP)$, but the result is not significant ($F_{NNBORS}(1,9) = 4.92$, $p = 0.054$).

The same pattern of significance is obtained when the analysis is performed at 200, 400, 600, and 800 epochs. We have also performed preliminary trials using a number of other manipulations: changing the magnitude of the stopping criterion, the range of the $P(ONE)$ parameter, and the range of $FREQ$. With the exception of the $FREQ$ experiment, none of these manipulations had any noticeable effect on the result. In the $FREQ$ manipulation, frequencies of 1 and 8 were used. All significant effects were the same as reported above except for $P(FLIP)$ which showed a non-significant trend consistent with our reported results. Informal analysis suggests that much of the effect of $P(FLIP)$ is lost at high frequencies.

The fact that abstract words are faster than concrete words is inconsistent with the findings of a number of psycholinguistic studies which report that reaction times to abstract words in the naming and lexical decision tasks are slower than to concrete words. These findings were summarized in the introduction to this paper. This is a problem for those who believe that attractor networks make good models of lexical access phenomena, while at the same time hold that the primary distinction between abstract and concrete words is that concrete words have denser semantic bit representations. Our finding suggests that a difference in the density of the bit representation is inadequate to explain reaction time differences of human subjects.

The simulation shows that being in a dense semantic neighborhood results in decreased settling times. This is consistent with the regularity effect in reading. There, according to [Seidenberg & McClelland, 1989] and [Plaut et al., 1996] the effect is a result of neighborhood training. Training on one pattern has the effect of reducing reaction times on similar patterns. In our simulation, the mapping between inputs and outputs is random. Knowing the correct output for some input can give no information about what is the correct output for any other input. Therefore, the result reported in this paper cannot be caused by the same kind of neighborhood training. We suggest rather that training on a pattern may have the effect of decreasing settling time of patterns which are nearby in output space. This *within-domain* neighborhood effect is distinct from the neighborhood training observed by [Seidenberg & McClelland, 1989] which we will call a *between-domain* neighborhood effect, although it may have significant influence on settling times even in problems which feature a regular mapping. Further study to determine the relative size of these two effects is required.

For problems without a regular mapping, this within-domain neighborhood effect may serve as a distinctive prediction of attractor network models. As such, it is a candidate for distinguishing between such models and alternative hypotheses.

Simulation 2 - Results & Discussion

We reran the testing phase of the previous section, using the same set of ten networks, and the same final weights. The only difference between this run and the earlier one is that in this second run hidden units as well as output units were included in the test for settling. Under this new stopping criterion, each of the main effects is significant at $p < 0.05$. Unfortunately, the direction of all of the effects except frequency are reversed. Frequent words are still facilitated ($F_{FREQ}(1,9) = 247.1$, $p < 0.001$). However, now concrete words are faster than abstract ($F_{P(ONE)}(1,9) = 208.5$, $p < 0.001$). Also, each of the factors controlling the density of a semantic neighborhood now interferes with settling ($F_{P(FLIP)}(1,9) = 52.5$, $p < 0.001$; $F_{NNBORS}(1,9) = 13.1$, $p = 0.006$). Figure 3 plots the results.

We have recently successfully reproduced the results of [Plaut, 1995], which demonstrates possible mechanisms behind associative and semantic priming in an attractor network model. In [Plaut, 1995], associative priming is shown to have a much stronger influence on reaction times than does semantic priming. We have found that in our replication including hidden units in the stopping criterion has the effect of reversing this result. With hidden units in the stopping criterion, semantic priming has a stronger influence than does associative priming. For details see [Clouse, 1998].

Note that in both sets of simulations, the only change in methodology was the introduction of hidden units into the stopping criterion. How could such a small change have such a large impact on the results?

Given some input pattern and some set of weights, the trajectory followed through activation space for the two tests must necessarily be identical up to the point where the output-only criterion is met. After this point, the output units change very little, but the hidden units continue to change until the

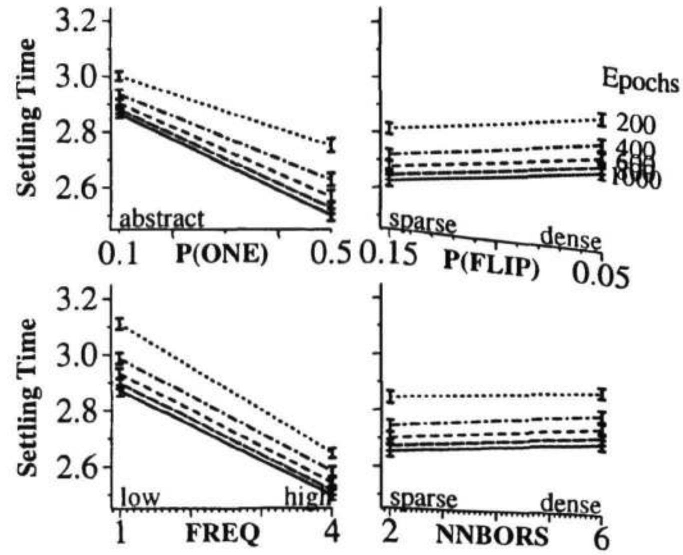

Figure 3: Main Effect for Simulation 2

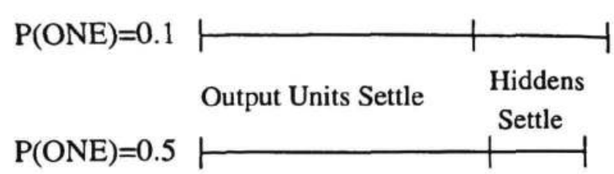

Figure 4: Settling times for two words

settling criterion is met. The time during which the hidden units change by themselves is approximately half the length of time required for the output units to reach criterion. This is long enough to reverse much if not all of the difference in settling times of the output units. See figure 4.

For the case of the $P(ONE)$ effect, we have some understanding of why the settling time reverses when hidden units are included in the stopping criterion. The settling behavior can be characterized by a number of stages. In the earliest stage little influence from the input pattern has yet "leaked through" to the output layer. The observed behavior, which we believe is based mainly on the bias weights of the output units, is to turn all outputs off. This is followed by a second stage in which selected bits corresponding to ones in the target pattern are turned on in the output. The distance traveled from all zeros to the correct number of output ones is much greater for patterns with $P(ONE) = 0.5$ than for those with $P(ONE) = 0.1$. This may account for the longer settling times for concrete patterns when only the output units are tested for settling. In the final stage, after the output units have settled, the output units change very little, therefore the hidden units are settling under constant input. The formulas governing activation under these conditions dictate that units with more extreme inputs will tend to reach settling criterion faster than those with inputs near zero. We have observed that the inputs to hidden units are generally more extreme for patterns with $P(ONE) = 0.5$, which accounts for the faster settling of hidden units for these patterns. That the inputs to $P(ONE) = 0.5$ patterns are more extreme is reasonable considering that these inputs consist of more non-zero terms, due to the greater number of ones in the output. We have yet to work out a satisfying explanation for the semantic neighborhood effect.

Since changing the stopping criterion has such a profound influence on the simulation results, a principled choice of stopping criterion is very important. Common practice is to exclude hidden units from the stopping criterion. This, we believe, is based on the reasonable assumption that the activation of the hidden units is, by definition, not visible to other networks, and therefore has no influence their behavior, including any network whose task requires deciding if this network has settled.

Conclusion

In this paper we have reported the results of a simulation of the effects of a number of different variables on an existing attractor network model of the mapping from orthography to semantics. We find significant effects of frequency, the number of ones in the output pattern, and two measures of neighborhood density.

We find that more ones in the output pattern tends to slow settling. One possible representation of concrete words used by connectionists includes more one bits. Since the psycholinguistic literature suggests that reaction times to concrete words are faster than to abstract, our result suggests that this representation causes a conflict with human behavior. A more radical conclusion, not shared by both authors, is that concrete words should be represented by patterns with few bits on, suggesting a more "focal" pattern of activity. We also find a semantic neighborhood effect in which words which have many semantically similar words in the language tend to settle faster.

In a second set of simulations, we show that both of these effects reverse when hidden units are included in the stopping criterion. This we interpret as an effect which is of interest principally to neural network researchers, rather than as a prediction of human behavior.

Acknowledgments

We would like to express our thanks to Paul Munro, Cathy Harris, David Plaut and the GURU research group at UCSD for helpful discussions. Daniel Clouse was supported in part by UC Senate Grant number RW252G/B to GWC.

References

[Breedin et al., 1994] Breedin, S. D., Saffran, E. M., and Coslett, H. B. (1994). Reversal of the concreteness effect in a patient with semantic dementia. *Cognitive Neuropsychology*, 11(6):617–660.

[Brown & Watson, 1987] Brown, G. D. A. & Watson, F. L. (1987). First in, first out: Word learning age and spoken word frequency as predictors of word familiarity and word naming latency. *Memory & Cognition*, 15(3):208–216.

[Buchanan et al., 1996] Buchanan, L., Burgess, C. & Lund, K. (1996). Overcrowding in semantic neighborhoods: Modeling deep dyslexia. *Brain & Cognition*, 32:111–114.

[Clouse, 1998] Clouse, D. S. (1998). PhD thesis, University of California, San Diego. in preparation.

[Cottrell & Plunkett, 1995] Cottrell, G. W. & Plunkett, K. (1995). Acquiring the mapping from meaning to sounds. *Connection Science*, 6(4):379–412.

[de Groot, 1989] de Groot, A. M. B. (1989). Representational aspects of word imageability and word frequency as assessed through word association. *J Exper Psych: Learning, Memory & Cognition*, 15(5):824–845.

[Hinton, 1989] Hinton, G. E. (1989). Connectionist learning procedures. *Artificial Intelligence*, 40:185–234.

[James, 1975] James, C. T. (1975). The role of semantic information in lexical decisions. *J Exper Psych: Human Perception & Performance*, 104(2):130–136.

[Kroll & Merves, 1986] Kroll, J. F. & Merves, J. S. (1986). Lexical access for concrete and abstract words. *J Exper Psych: Learning, Memory & Cognition*, 12(1):92–107.

[Pearlmutter, 1989] Pearlmutter, B. A. (1989). Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1(2):263–269.

[Plaut, 1995] Plaut, D. C. (1995). Semantic and associative priming in a distributed attractor network. In Moore, J. D. & Lehman, J. F., editors, *Proceedings of the 17th Annual Conference of the Cognitive Science Society*. Erlbaum.

[Plaut et al., 1996] Plaut, D. C., McClelland, J. L., Seidenberg, M. S. & Patterson, K. E. (1996). Understanding normal and impaired word reading: Computational principles in quasi-regular domains. *Psychological Review*, 103(1):56–115.

[Plaut & Shallice, 1993] Plaut, D. C. & Shallice, T. (1993). Deep dyslexia: A case study of connectionist neuropsychology. *Cognitive Neuropsychology*, 10(5).

[Seidenberg & McClelland, 1989] Seidenberg, M. S. & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, 96(4):523–568.

[Strain et al., 1995] Strain, E., Patterson, K. & Seidenberg, M. S. (1995). Semantic effects in single-word naming. *J Exper Psych: Learning, Memory & Cognition*, 21(5):1140–1154.

[Whaley, 1978] Whaley, C. P. (1978). Word-nonword classification time. *J Verbal Learning and Verbal Behavior*, 17:143–154.