A Neural Network Model of Visual Word Recognition

Akathian Santhakumar

University of Toronto Scarborough

PSYC90

Dr. Blair Armstrong

April 8, 2022

Abstract

There has been much debate over how semantic memory is encoded. Most research has revolved around this topic using semantic priming, with little to no research conducted without priming factors. We provide computational evidence for a single mechanism distributed network account by investigating effects of lexical factors such as frequency, reading ability, category dominance, and semantic richness on two measures of reaction time (RT). We choose settling time as the first RT, and the time taken for the network to reach a certain semantic stress value, as the second. We demonstrate a three-way interaction effect between frequency, reading ability and semantic richness on the stress RT, and two two-way interactions, clamping by frequency and clamping by richness, on the settling time RT. These are mostly consistent with previous behavioral research involving priming factors, with the exception of the absence of the three-way interaction for the settling time RT. We also demonstrate the network's ability to differentiate between words and nonwords using these RTs. Taken together, these results provide support for the single mechanism distributed network account for semantic memory.

Introduction

Over the years, the method of studying the effects of factors that can influence reaction time in different tasks has been used to make conclusions about underlying psychological and neurological processes (Borowsky & Besner, 2006). The lexical decision task enables researchers to evaluate the abilities of participants to make use of their mental lexicon. One such task is distinguishing whether a given visual stimulus is a word (Borowsky & Besner, 2006).

There has been much debate over methods of processing models that simulate human lexical decision performances. Most research has been modeling semantic priming effects, the phenomenon where it is faster and more accurate to process words when they are preceded by a related word (Plaut & Booth, 2000). Models simulating different theories, such as spread-of-activation, compound-cue and distributed network theories have been proposed.

Spread-of-activation, or spreading-activation theories propose that semantic memory is constructed as a network of nodes where activations spread across it (Anderson, 1988). The strength between the connections of the nodes are thought to increase with practice. The spread of this activation is believed to be responsible for retrieval of information in the network. Compound-cue theories suggest that compound-cues are used to access memory. A compound-cue is one that is made of the word that is presenting, and the context in which it appears (Ratcliff & McKoon, 1988). With related words appear more frequently together in context, semantic memory retrieval is faster. Finally, distributed networks are made of distributed patterns of activity in an interconnected network of units (Plaut, 1995). Words that are related are given similar patterns. As such, when presented a pattern following presentation of a prime, the network activations will start from their states produced by the prime. These activations are more similar to what is expected to be produced from a related word, compared to an unrelated one. As a result, faster reaction times occur.

These theories have been challenged with the showing that experimental factors, such as target frequency, category dominance and relatedness proportion influence priming effects (Plaut & Booth, 2000). With each challenge, researchers looked to add complexity to the given theory. Borowsky and Besner (1993) have proposed a model with multiple stages of processing, which explained such effects. Conversely, Plaut and Booth (2000) demonstrated a single mechanism model which also explained these effects on semantic priming. Furthermore, Borowsky and Besner (1993) predict only additive effects of these factors with no interactions, which per previous research, suggest that these factors affect multiple stages of processing. On the other hand, Plaut and Booth (2000) predict interactive effects between this factors.

There has been much criticism for Plaut and Booth (2000)'s model, notably by

Borowsky and Besner (2006). They identify three main problems with Plaut and Booth (2000)'s model: it cannot differentiate words and nonwords that are orthographically related while still producing the observed effects on RT, it is inaccurate with how individuals with damaged semantic systems, and it does not simulate the effect of these factors seen in behavioral data. In (Plaut & Booth, 2006), they address these issues with the same model run with different experiments, in turn showing that they were not problematic for their theory.

As such, we use Plaut and Booth (2000)'s model to investigate the effects of the aforementioned lexical factors on reaction time (RT) and accuracy of a model without the use of priming, as there has been very little research on this topic. We operationalize RT as the time at which differences in semantic activations fall below a threshold (settling time [ST]) as the first RT, and the time at which semantic stresses reach another threshold, as the second. Semantic stress aims to measure the level of familiarity of the resulting semantic pattern in the model following presentation of a word or nonword (Plaut, 1997).

When looking at both measures of RT, we expect the model to have faster RTs for words, and be slower for nonwords. We also expect the aforementioned lexical factors to have effects on these RTs. Specifically, we expect slower RTs for: low-frequency targets compared to high-frequency targets, low-richness (richness, i.e. number of features that are on in the semantic pattern) targets compared to high-richness targets, and lower clamping strengths compared to higher ones (Cheyette & Plaut, 2012). We also expect accuracy to be proportional to RTs, where a we expect worse accuracy with lower RTs.

Methods

Network architecture, orthographic representations and semantic representations were based off of Plaut and Booth (2000).

Orthographies

Orthographic representations were generated from a selection of 10 consonants (B, D, K, L, M, N, P, R, S, T) and 5 vowels (A, E, I, O, U). Each letter was represented with a 6 unit binary pattern, where only 2 of the 6 units were active. 128 words from the 500 possible consonant-vowel-consonant (CVC) strings were chosen, and 128 nonwords were chosen from the 250 possible vowel-consonant-vowel (VCV) strings. There were on average 7.18 features shared among words, and 6.75 shared among nonwords.

Semantics

Semantic representations of words were generated from first creating 8 different 100 unit binary patterns, where each feature had a 10% probability of being active (Plaut, 1995). Each of these patterns were used as prototypes for 8 semantic categories, where 16 patterns were generated for each category. These patterns were created by changing features from the corresponding prototypes (Chauvin, 1988). Eight patterns were created starting from the prototype, where each feature had a 20% chance of being re-sampled. The features to be re-sampled would subsequently have a 10% chance of being activated. Patterns generated this way were considered to be high-dominance. Low-dominance patterns were generated similarly, but with a 40% chance that a given feature will be re-sampled. Each generated pattern across all categories were checked such that they differed by at least 4 features. The semantic patterns ranged from 5 to 17 features turned on, with an average of around 10. Each semantic pattern was then randomly assigned to a word. From this, 4 high-dominance words and 4 low-dominance words from each category were randomly chosen to be high-frequency. High-frequency would be presented four times as much during network training as the remaining, low-frequency words. Semantic richness was determined following the generation of words by finding the median number of features that are turned on. Semantic patterns that fell below the median were considered to be low-richness, and those falling on or above the median were considered to be high-richness.

Network Architecture

The network consisted of 18 orthographic (input) units to code for three-letter inputs (6 units per letter). All of these units were connected to 100 hidden units, with each hidden unit being bidirectionally connected to 100 semantic (output) units. Each semantic unit was also connected to every other semantic unit. Hidden and semantic units were connected to a bias unit with activity of 1.0. A random value between -0.25 and +0.25 was set on all weights of connections. The network was constructed such that units would change continuously over time as it received input from other units. Time was discretized into ticks t of duration τ, giving the activation of unit j as: $a_j^{[t]} = \tau\sigma(\Sigma_i w_{ij} a_i^{[t-\tau]}) + (1-\tau)a_j^{[t-\tau]}$ where w_{ij} is the weight from unit i to j, and $\sigma(x)$ is the sigmoid function.

Training

The network was trained by inputting to each orthographic unit a value calculated by: $0.2 + \text{clampStrength}(\text{inputUnit} - 0.2)$ [1] where the inputUnit is the given unit from the orthographic representation. Clamp strength is used to reflect different levels of visual degradation. A clamp strength of 0.8 was used in training, and values of 0.5, 0.6, 0.7 and 0.8 were used during testing. Following the presentation of a word, all units were updated with Equation 1 with $\tau = 0.04$ over 5 units of time. Error between the activations of the semantic units and the target semantic pattern was calculated with cross-entropy: $C = \tau\Sigma_{3<t\leq 4}\Sigma_j s_j \log(a_j^{[t]}) + (1-s_j)\log(1-a_j^{[t]})$ (Hinton, 1989). Continuous backpropagation of error through time was used to calculate its partial derivative with respect to weights: $\Delta w_{ij}(p) = \epsilon\frac{\delta C}{\delta w_{ij}} + \alpha\Delta w_{ij}(p-1)$ with the learning rate $\sigma = 0.005$ and momentum $\alpha = 0.8$ (Pearlmutter, 1989). The network reached training criterion of 0.3

[1] Plaut and Booth (2000) specified a calculation as follows: "an input strength of 0.8 specifies an external input of 0.575 for present features and -3.18 for absent features, because $0.8(1.0-0.2) = 0.64 = \sigma(0.575)$ and $0.2 - 0.8(0.2-0.0) = 0.04 = \sigma(-3.18)$". We identify this as an error as the standard calculation for clamping is as we describe.

after 405 epochs of training.

Testing

We test the network by presenting the orthographic patterns of words and let it process up to 125 ticks ($\tau = 0.04$ with 5 intervals, giving 25 ticks per interval). We allocate the network a grace time of 3 intervals, such that it only starts receiving error at the beginning of the 4th interval (75th tick). Once all words and nonwords have been processed, we calculate the average of changes in activation of semantic units between ticks. We find an optimal threshold such that it segregates words and nonwords as accurately as possible. We expect words to be below this threshold, while nonwords should be above it. Once the threshold has been decided, we record the settling time (ST), the tick at which each word reached the threshold. We record average STs for words based on frequency, dominance, clamp strength and semantic richness. We also record average STs for nonwords based on clamp strength. As in Plaut and Booth (2000), we calculate semantic stress values for each word and nonword at each tick: $S_j = a_j \log_2(a_j) + (1 - a_j)\log_2(1 - a_j) - \log_2(0.5)$. We find the stresses at the calculated STs, and perform a similar thresholding process as with determining STs. However for semantic stress, we expect the values for words to be higher than the threshold, and the values for nonwords to be lower. We record the stress reaction time (stress RT), the tick at which words and nonwords reach the stress threshold. As we did with STs, we record average RTs for words based on frequency, dominance, clamp strength and semantic richness, and average RTs for nonwords based on clamp strength.

Results

Table 1

Fixed Effects between groupings with RT as ST, at epoch 405

Interaction	Estimate	Std. Error	df	t value	Pr(>|t|)	Signif. codes
(Intercept)	-452.366	210.592	228.525	-2.148	0.0328	*
clamp	669.189	283.55	217.537	2.36	0.0192	*
freq	125.334	64.122	235.546	1.955	0.0518	.
rich	46.938	20.669	228.324	2.271	0.0241	*
domlow	146.872	292.167	227.818	0.503	0.6157	
clamp:freq	-178.518	87.135	219.326	-2.049	0.0417	*
clamp:rich	-60.958	27.84	217.253	-2.19	0.0296	*
freq:rich	-9.45	6.272	234.656	-1.507	0.1332	
clamp:domlow	-158.269	393.438	216.915	-0.402	0.6879	
freq:domlow	-19.282	85.197	232.474	-0.226	0.8212	
rich:domlow	-14.64	27.727	228.088	-0.528	0.598	
clamp:freq:rich	13.739	8.525	218.617	1.612	0.1085	
clamp:freq:domlow	30.415	115.626	217.564	0.263	0.7928	
clamp:rich:domlow	16.07	37.375	216.841	0.43	0.6677	
freq:rich:domlow	1.973	8.111	232.223	0.243	0.808	
clamp:freq:rich:domlow	-3.083	11.016	217.217	-0.28	0.7798	
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1						

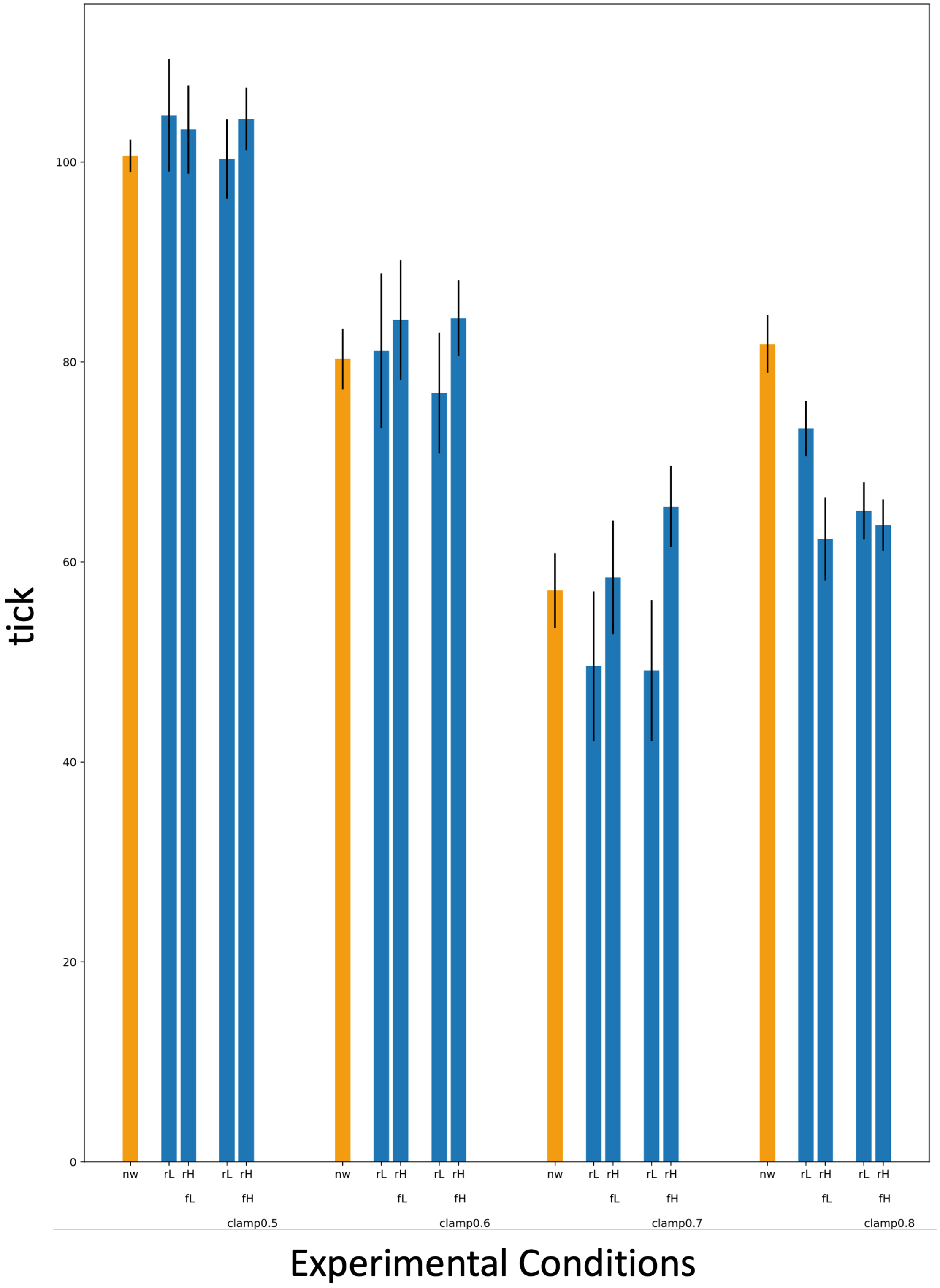
Table 2

Fixed Effects between groupings with RT as stress RT, at epoch 400

Interaction	Estimate	Std. Error	df	t value	Pr(>|t|)	Signif. codes
(Intercept)	353.963	71.926	259.677	4.921	1.53E-06	***
clamp	-345.521	98.208	244.898	-3.518	0.000518	***
freq	-59.699	23.255	263.476	-2.567	0.010806	*
rich	-16.395	7.011	259.246	-2.339	0.020121	*
domlow	-119.653	101.477	259.733	-1.179	0.23943	
clamp:freq	66.969	32.035	244.476	2.09	0.03761	*
clamp:rich	19.705	9.583	244.202	2.056	0.040816	*
freq:rich	5.168	2.258	262.637	2.289	0.022861	*
clamp:domlow	164.467	138.359	245.434	1.189	0.235705	
freq:domlow	37.089	30.793	261.888	1.204	0.229491	
rich:domlow	12.138	9.7	259.52	1.251	0.211964	
clamp:freq:rich	-6.208	3.113	243.595	-1.994	0.047217	*
clamp:freq:domlow	-45.799	42.322	244.131	-1.082	0.28025	
clamp:rich:domlow	-16.242	13.235	245.023	-1.227	0.220931	
freq:rich:domlow	-3.503	2.942	261.222	-1.191	0.234821	
clamp:freq:rich:domlow	4.392	4.045	243.508	1.086	0.278665	
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1						

Figure 1

Bar graph with standard error plotting ST as RTs across groups



Note. We abbreviate as follows: nonwords (nw), low-richness words (rL), high-richness words (rH), low-frequency words (fL), high-frequency words (fH)

Figure 2

Bar graph with standard error plotting accuracy of groups at their average RTs, where ST is RT

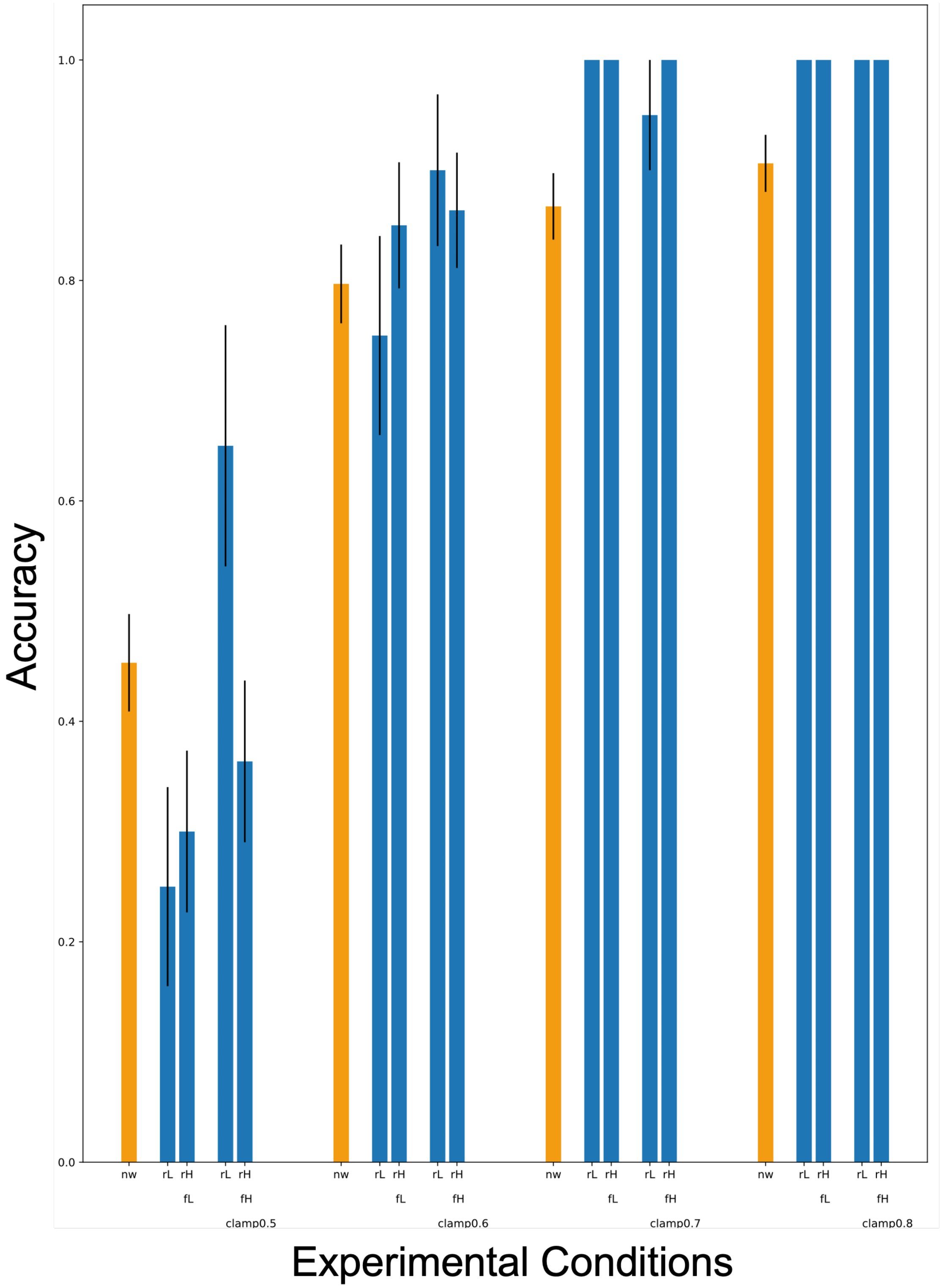


Figure 3

Bar graph with standard error plotting stress RTs across groups

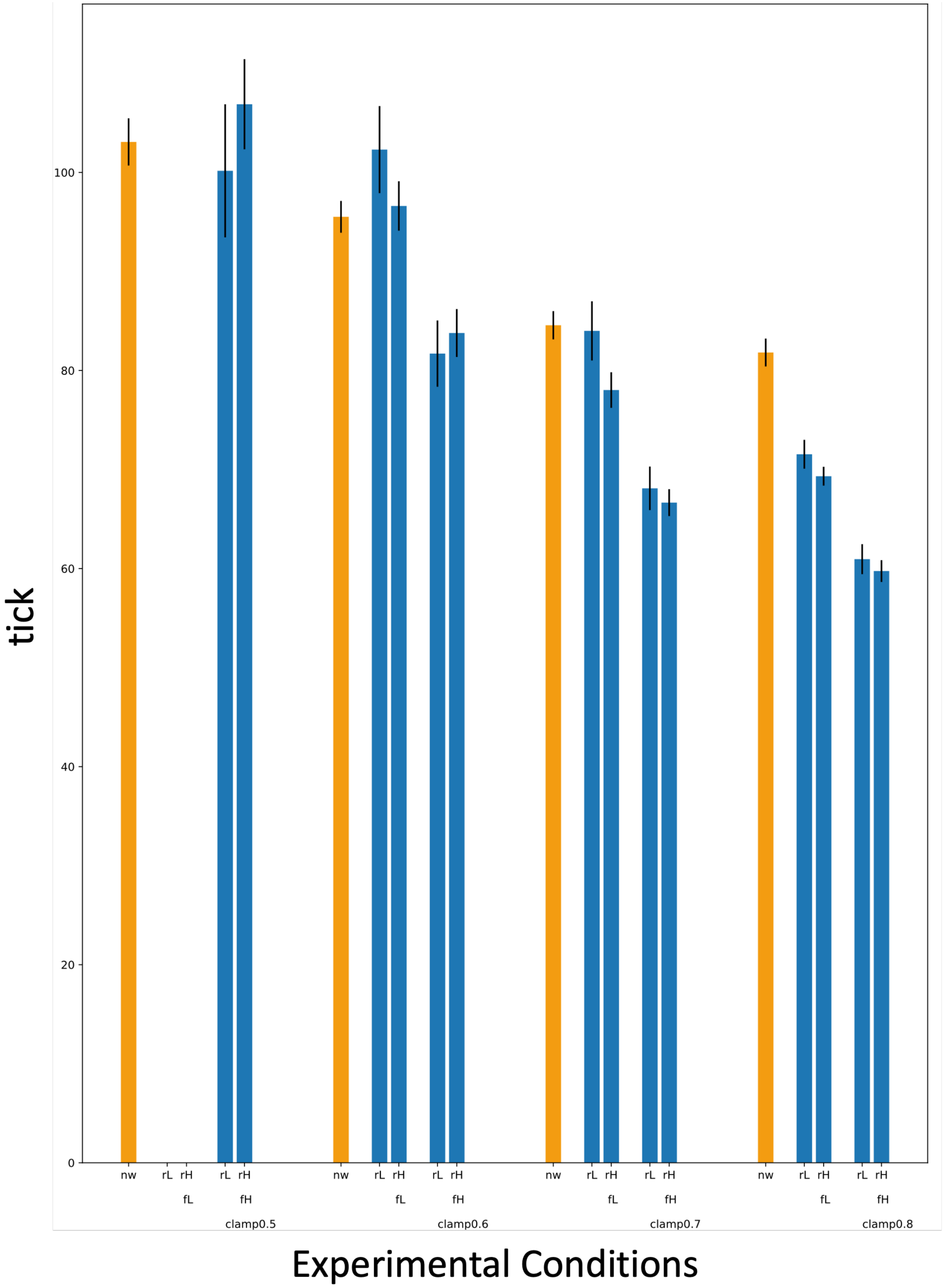


Figure 4

Bar graph with standard error plotting accuracy of groups at their average stress RTs

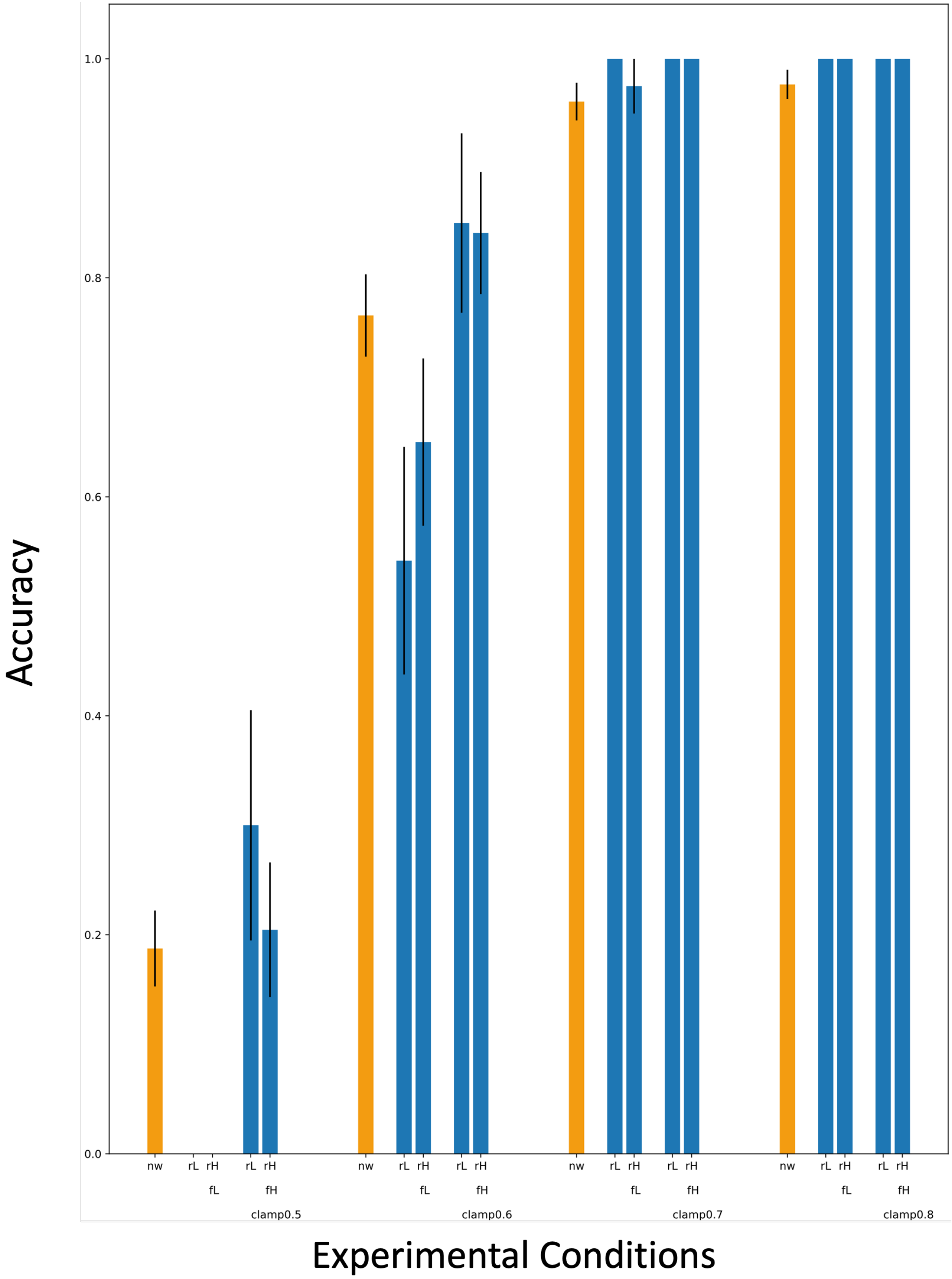
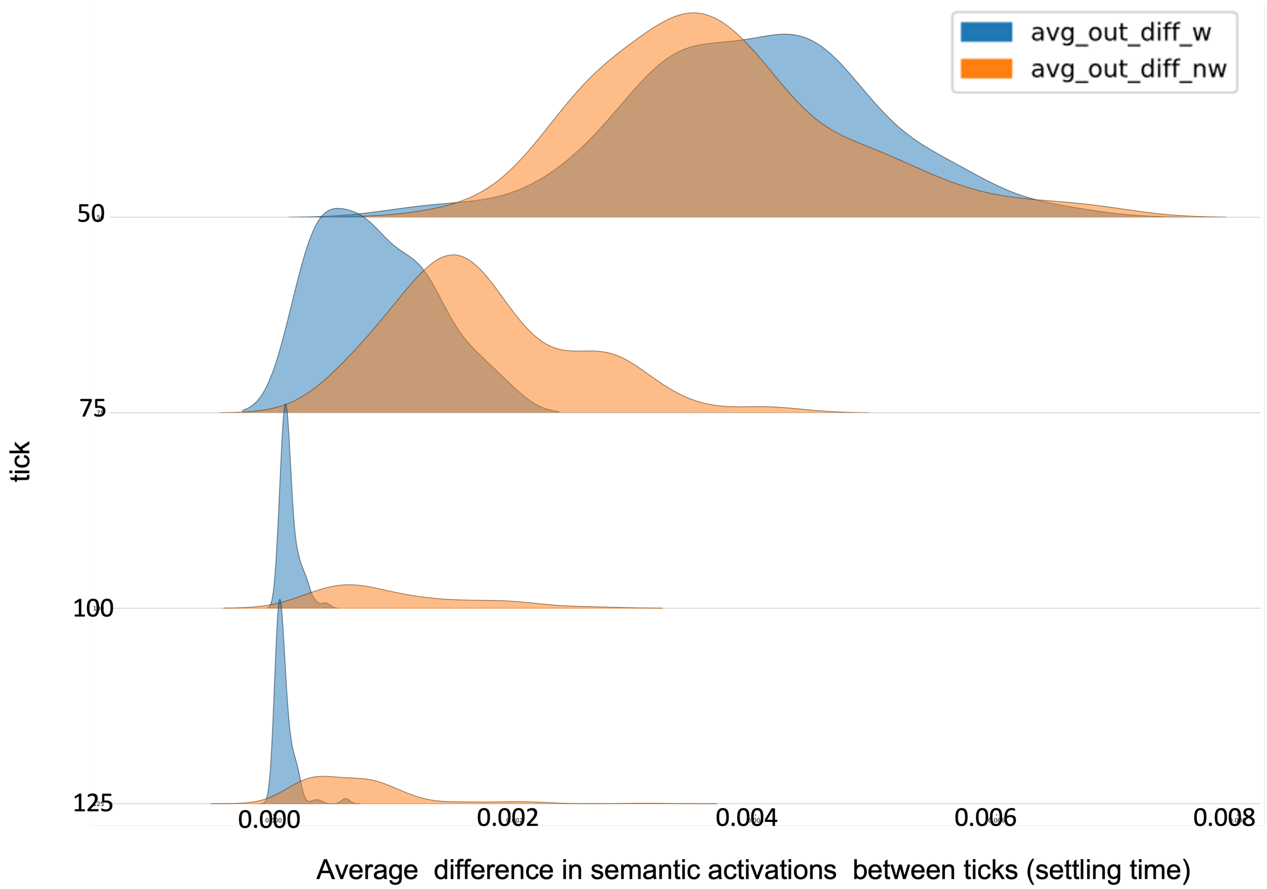


Figure 5

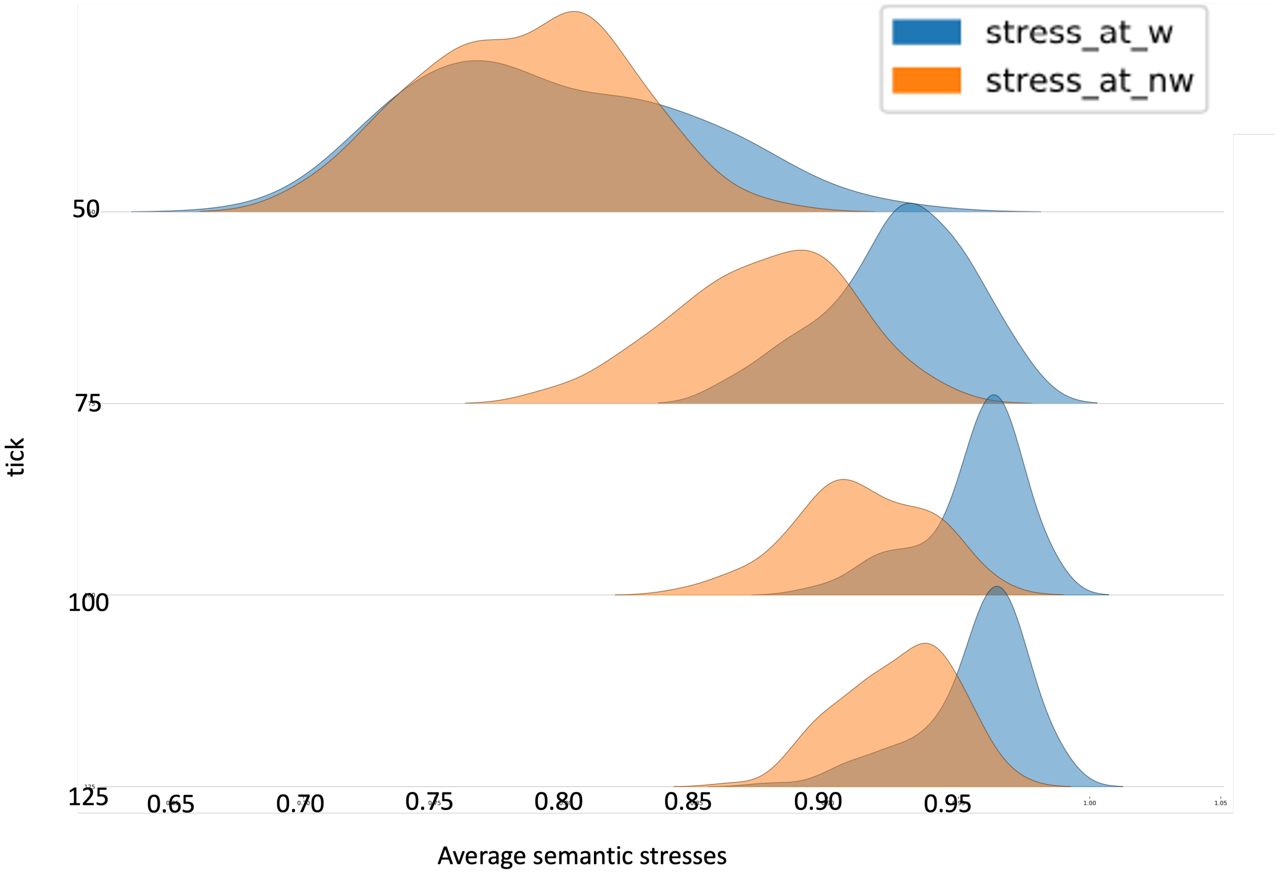
Joy plot showing distribution of settling times in words and nonwords every 25 ticks starting from tick 50 at clamp 0.8



Note. We abbreviate as follows: nonwords (nw), words (w)

Figure 6

Joy plot showing distribution of semantic stresses in words and nonwords every 25 ticks starting from tick 50 at clamp 0.8



The model was trained until criterion was reached, which for our random initialization, was at epoch 405. We record semantic activations for all inputs at every 50 epochs of training, and at the final epoch of training.

We create a linear mixed model fit by REML and t-tests using Satterthwaite's method in R with formula rt ~ clamp $*$ frequency $*$ richness $*$ dominance + (1|word). We perform this analysis on both Rts. When ST is used as an RT at epoch 405 of training, we observe main additive effects of clamping, frequency, richness and interaction effects of clamping by frequency and clamping by richness (see Table 1). When looking at stress RTs at epoch 400 we observe strong main additive effects of clamping, frequency, richness, and interaction effects of clamping by frequency, clamping by richness, frequency by richness, and clamping by frequency by richness (see Table 2). We see no other effects from the tables. Particularly, we observe no categorical dominance effects, nor other interaction effects.

We plot experimental groups on the x-axis, and ST as RT on the y-axis on Figure 1, 2, 3 & 4. These figures are produced following testing of the network. Observing Figure 1 & 2, we see a slower RT and lower accuracy for nonwords at clamp 0.8 than words. We also see slower RTs for low-richness words compared to high-richness words. At clamp 0.7, we observe nonwords having a comparable RT to words, but with a lower accuracy. We notice a reversed effect from 0.8, where low-richness words have faster RTs than high-richness words at the 0.7 clamp. In Figure 3 & 4, at clamp 0.8, we observe a slower and less accurate RT for nonwords than words, slower RTs for low-richness than high-richness words and slower overall RTs for low-frequency than high-frequency words. At the 0.7 clamp, we see the same patterns, but with nonwords having a similar RT to low-richness, low-frequency words. However, the nonwords are less accurate. We see no interpretable patterns for clamps 0.5 and 0.6 across both RT measures.

In Figure 5, we plot the distribution of average differences in semantic activations between ticks. We observe that in the beginning 75 ticks, the words and nonwords cannot be easily separated. Following tick 75, we see sharp differentiation between words and nonwords, where most words are centered around a point, while nonwords are spread out over a much larger distribution. In Figure 6, we can observe similar effects. It is difficult to distinguish between words and nonwords prior to tick 75. Following tick 75 however, the words become more focused around a point than nonwords. Nonwords are also spread over a wider distribution.

Discussion

We have observed the predicted effects at the 0.8 clamp strength, for both RTs. We have also observed the expected effects at the 0.7 clamp for stress RT. Although nonwords and low-richness, low frequency words have similar RTs for this clamp strength, we can see that the nonwords are less accurate. For ST as RT, we observe a speed-accuracy trade-off (SAT) at the 0.7 clamp. As with the stress RT observations, nonwords are marginally faster on average to words but they are less accurate. We also see no discernible patterns at clamp value of 0.6 and lower. This is due to the network only being trained at clamp at 0.8, and has less generalizability with increasing distance from training conditions.

For the stress RT, the main effects on clamping, frequency and richness are characterized by a three-way interaction between them (see Table 2). In the work of Plaut and Booth (2000), they find similar effects but with priming effects as an added factor in interaction effects, instead of richness. Plaut and Booth (2000)'s main findings is that frequency affects semantic priming, depending on perceptual ability. In this current work, we find the same result, with the three-way interaction between factors affecting RTs. They argue that the reason high-frequency words have stronger semantic stress (thus, higher semantic familiarity) than low-frequency words, is due to the difference in training frequency. The network is able to react faster to the word following presentation as a result. Cheyette and Plaut (2012) explain that semantic richness provides some concreteness to the words, resulting in stronger activations in semantic units, and thus, faster RTs. Furthermore, with clamping being an operationalization of reading ability, we observe, as in Plaut and Booth (2000), an effect of clamping on RTs. Taken together, it is expected that these three factors have an interaction effect between them.

Similarly, the same effect is observed in settling time RT, where we have interaction effects of clamping and frequency, and clamping and semantic richness (see Table 1). The two interactions can be explained as with the stress RT. However, we do not see the three-way interaction we expect. Notably, there is no interaction between frequency and richness. This could be due to frequency not having as great of an effect ($\Pr(> |t|) < 0.1^{\cdot}$) on this RT. This can also be attributed to the SAT that we observe at the 0.7 clamp.

The effect of SATs on RTs have been well documented phenomenons over the years in lexical decision tasks (Rinkenauer, Osman, Ulrich, Müller-Gethmann, & Mattes, 2004). It has been observed as an individual's strategy for emphasizing speed over accuracy, and known to be a conscious decision. It can be argued that this is a result of random processes in the methods, such as data generation or network weight initialization. Therefore, even though we do not see an SAT in the original Plaut and Booth (2000) paper, we can accept this effect as still modeling behavioral data.

The stark differentiation between words and nonwords seen in the joy plots of average differences in semantic activations between ticks are also expected (Figure 5). Since we only start injecting error at the 75th tick, we should only expect to see differentiation and model learning starting at this tick. Indeed, following tick 75, we see strong discrimination between words and nonwords when looking at this measure. We observe similar differentiations when observing the joy plots for semantic stress (Figure 6). We can see differentiation between words and nonwords starting to occur following tick 75. Plaut and Booth (2000) produce a similar graph. However, the current result is not consistent with their findings. Their semantic stresses showed much greater differentiation between words and nonwords, with very little overlap between the two. The distribution of their words are also much more focused around a point, with their nonwords also being wider than the current findings. We attribute this finding to the lack of priming factors in this experiment. We theorize that, without priming factors, the network does not associate certain words with others (Plaut & Booth, 2000). As such, the semantic stress (semantic familiarity) for any given words would be lower, resulting in a lesser degree of differentiation between words and nonwords when looking at semantic stress.

We see no dominance effects in either of the RTs. Loftus (1973) notice effects of category and instance dominance of RTs in priming experiments. Category dominance effects were observed when category was presented after the word target, and instance dominance was observed when category presentation preceded the word target. With this result, we can safely accept the absence of dominance effects, as it seems that they only occur in presence of priming factors.

In conclusion, we present additional computational support for single-mechanism distributed network account of lexical factors. Further, we establish the ability of this type of network being capable of accounting for lexical factors, while excluding priming factors. We find some differences in results between the current work and Plaut and Booth (2000)'s work, but we are able to explain these discrepancies, as they are likely due to the absence of priming factors. Future directions for this study would include behavioral trials without priming factors, and seeing whether this model accurately simulates the behavioral data.

References

Anderson, J. R. (1988). A spreading activation theory of memory. *Readings in Cognitive Science*, 137–154. doi: 10.1016/b978-1-4832-1446-7.50016-9

Borowsky, R., & Besner, D. (1993). Visual word recognition: A multistage activation model. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *19*(4), 813–840. doi: 10.1037/0278-7393.19.4.813

Borowsky, R., & Besner, D. (2006). Parallel distributed processing and lexical–semantic effects in visual word recognition: Are a few stages necessary? *Psychological Review*, *113*(1), 181–195. doi: 10.1037/0033-295X.113.1.181

Chauvin, Y. (1988). Symbol acquisition in humans and neural(pdp) networks. *PhD thesis, University of California, San Diego.*.

Cheyette, S. J., & Plaut, D. C. (2012). Modeling the n400 erp component as transient semantic over-activation within a neural network model of word comprehension. *Cognition*, *162*, 153–166. doi: 10.1016/j.cognition.2016.10.016

Hinton, G. E. (1989). Connectionist learning procedures. *Artificial Intelligence*, *40*(1–3), 185–234. doi: 10.1016/0004-3702(89)90049-0

Loftus, E. F. (1973). Category dominance, instance dominance, and categorization time. *Journal of Experimental Psychology*, *97*(1), 70-74. doi: 10.1037/h0033782

Pearlmutter, B. A. (1989). Learning state spacetrajectoriesin recurrent neuralnetwork. *International Joint Conference on Neural Networks*. doi: 10.1016/0004-3702(89)90049-0

Plaut, D. C. (1995). Semantic and associative priming in a distributed attractor network. *Proceedings of the 17th Annual Conference of the Cognitive Science Society*, 37–42.

Plaut, D. C. (1997). Structure and function in the lexical system: Insights from distributed models of word reading and lexical decision. *Language and Cognitive Processes*, *12*(5–6), 765–806. doi: 10.1080/016909697386682

Plaut, D. C., & Booth, J. R. (2000). Individual and developmental differences in semantic priming: empirical and computational support for a single-mechanism account of lexical processing. *Psychological Review*, *107*(4), 786–823. doi: 10.1037/0033-295x.107.4.786

Plaut, D. C., & Booth, J. R. (2006). More modeling but still no stages: Reply to borowsky and besner. *Psychological Review*, *113*(1), 196–200. doi: 10.1037/0033-295X.113.1.196

Ratcliff, R., & McKoon, G. (1988). A retrieval theory of priming in memory. *Psychological Review*, *95*(3), 385-408. doi: 10.1037/0033-295x.95.3.385

Rinkenauer, G., Osman, A., Ulrich, R., Müller-Gethmann, H., & Mattes, S. (2004). On the locus of speed-accuracy trade-off in reaction time: Inferences from the lateralized readiness potential. *Journal of Experimental Psychology: General*, *133*(2), 261–282. doi: 10.1037/0096-3445.133.2.261

Appendix

Supervised Study Passports

Student Name: Akathian Santhakumar
Supervisor: Dr. Blair Armstrong
These were sent through email over the course of the semester, signed digitally by Dr. Armstrong

- Journal for September 23:
  - Worked on getting the model to work in CLens.
  - Modified script to save inputs & targets to required .ex format for CLens
  - Working on building the model's architecture in CLens
  - How do I connect the semantic layer to itself? I am getting an error (attatched)
- Journal for September 30:
  - Made modifications to hyperparameters as discussed in our meeting
  - Got 96% accuracy when testing on training set
  - Start to investigate and work through why the model isn't entirely perfect
  - Need to check whether the semantics were generated the same as in the paper
  - Need to check which words the network is failing for
- Journal for October 7:
  - Evaluated whether the outputs (semantics) are sufficiently different so that they can be linearly separable
  - Figured out that a few pairs of semantics are not different by at least 4 features
  - Modified script to retry semantic generation until it is different by at least 4 features
  - Script takes very long to run (freezes at about semantic #15)
  - I might be understanding the paper wrong, we can discuss the generation of my semantics during tomorrow's meeting
- Journal for October 15:
  - Translated semantics generation c code to python
  - Set
    nFeatures = 100 # number of features per pattern */
    nCategories = 8 # number of clusters (prototypes) */
    nMembers = 16 # number of exemplars per cluster */
    minProbOn = 0 # maximum sparcity of prototype */
    maxProbOn = 1 # minimum sparcity of prototype */

```
minDiff = 4 # minimum bit-wise difference among exemplars */
minProbDistort = 0.2 # min prob that feature is regenerated */
maxProbDistort = 0.4 # max prob that feature is regenerated */
sparse = 1 # generate output in "sparse" (unit numbers) format */
minOn = 1 # Min Number of units to be on in the exemplar */
maxOn = 100 # Max number of units to be on in the exemplar */
maxWCatDiff = 4
```

  - This translated code still hangs? I added the maxAttempts = 1000 parameter as well and it consistently reaches that point. I'm assuming there is something wrong with the parameters I have set, if not the actual code

- Journal for October 21:
  - Entirely translated the c code for generating semantics.
  - Semantics generate properly
  - Network achieves 100% accuracy with 0.00 error
  - How do I start to write code to test settling times? What output do I need from the model for this?

- Journal for October 28:
  - Read & adapted ET_ideal.in for my .in file.
  - Generates outputs thanks to the ET_ideal.in code
  - Need to figure out how to create directories in .in files if the directory does not already exist, since the script throws an error otherwise. I currently need to create the directory manually
  - I am attaching the outputs of the model
  - Need to understand the outputs, so that I can start writing code to measure settling time

- Journal for November 4:
  - Wrote script to get RT's (in ticks) for the last epoch for each word.
  - Did this by using the rounding method. Rounded the outputs to 0 or 1 and took the difference between the target & output. Got the tick # at which there is no more differences.
  - Need to generate the NW .ex file next
  - Perhaps also implement different measures in the script (accuracy for sure, not sure about others)

- Journal for November 11:
  - Modified script such that it outputs to a csv file.

  - CSV file includes word, rt, st and the averages of the differences between tick n and tick n-1, for n=1...20
  - Working on plotting these averages so that we can determine a threshold to set st

- Journal for November 18:
  - Added high/low dominance information to .ex file of words (Also added the prototype#/category# that each exemplar originated from, not sure if useful).
  - Found errors with data generation, when semantics would not differ enough from each other - this was causing training issues. A few orthographies were also the exact same for some reason. I believe we got lucky with the one we have been using
  - I resolved the issues above, and now, all my generated files can be trained to 100% accuracy on the train data (I tried with different seeds)
  - Fixed column names of the analysis files as we discussed (to be more descriptive)
  - Generated .ex file for non-words and made the model try to predict its targets.
  - Got output activations by adding code to the saveTest proc in the .in file
  - Ran same analysis that we have been running on these activations
  - Generated similar graphs to what we made last week in excel, but in python for convenience (with std dev error bars, attached)
  - 4 graphs total. For each word ("train") /non-word ("test) analysis file:
  - graphed the average differences between outputs between tick n and tick n-1, for n=1..20
  - graphed the averages of output averages at each tick
  - I may have misinterpreted some of my notes from our meetings, let me know if I misunderstood something or forgot to add something we have discussed

- Journal for November 25:
  - Fixed small issue in pattern generation that would result in too many semantic features being on (in the 50s range instead of 4-18 range as in the paper)
  - Added the stress, frequency, seeds, net params, word type (word/nonword), number of on features, columns to the csv's I generate
  - Appropriately renamed rt/acc pairs to reflect what it measures better
  - Generated analysis and graphs for the network before training
  - Fixed last week's graphs to use stderr (SEM) instead of stddev
  - Plotted both words and nonwords on the same graph
  - Performed regression in R

- Journal for December 2:

  - Figured out error with the reference paper
  - Added soft clamp and in integr to orthographic units
  - Added clampStrength/degradation
  - Installed R in the docker container, working towards streamlining analyses (almost done)

- Journal for December 9:
  - Split up training to saveTest every 50 epochs
  - Saved analysis of all saveTest results in two dfs (word, nonword)

- Journal for January 13:
  - Generated 5-95 percentile error bar graphs
  - Fixed errors in code and code cleanup for readability

- Journal for January 20:
  - Wrote function to get best thresholds for rt measures
  - Created joyplots for stress
  - Reproduced results for 100 ticks (dt 25) - same as 400 ticks, takes less time to run
  - Added two new measures: sum of sqrts of units and the sum of units greater than 0.5
  - Ran the network with 0.3 traingroupcrit
  - Created automated pdf reports that combine the graphs generated and regressions (attached)

- Journal for January 27:
  - Fixed joyplot to have plot only every 10 ticks
  - Made bar plots for every epoch for every measure (for interaction analysis)
  - Fixed sum gt 0.5 measure
  - Added joyplots to summary for all measures
  - Adjusted the shading of the joy plots

- Journal for Feb 3:
  - Made list of differences between my code and Plaut paper for semantic generation. This is to investigate the bimodality we have been seeing in the stress joyplots.
  - Only 1 difference found. We are creating categories to have exactly 10 on features rather than turning on features with probability of 0.1

  - Dont know that this will cause an issue, since turning on features with 0.1 prob will average out to having 10 units that are on
  - Commented data_gen file for review
  - Tried to generate all 500 CVC and 250 VCV words
  - Fixed mistake in orthography generation. Letters would occasionally have the same orthographic representation. This was due to an error when randomly choosing the indexes for which to turn on, I would consider the order to matter. For ex,
  - Turn on [1, 2] would be considered different from turn on [2, 1], even though they produce the same vectors
  - Fixed code to find threshold to measure accuracy based on num_incorrect = num words incorrectly below cutoff + num nonwords incorrectly above cutoff

- Journal for Feb 10:
  - Increased time intervals (-i) to 5 from 4 (to figure out if we are inducing an artificial ceiling effect for stress)
  - Have not ran analyses for this yet, but will likely have it ready for tomorrow morning
  - With the above changes, also ran the models for clamps from 0.5 to 0.8 in increments of 0.1
  - Again, have not ran analyses for this yet, but will likely have it ready for tomorrow morning
  - Added both word and nonword data for both high and low clamping on RT graphs (attached)
  - Added error bars (stderr and percentile - 90/10) for interactions plot.
  - Two different pdfs -> name_stderr.pdf and name_percentile.pdf
  - Cleaned code for drawing graphs
  - Added function to dynamically change thresholds for stress at settling times per epoch and clamp (attached)
  - Added the horizontal lines for the thresholds as well
  - Changed sum of sqrts to be sum of squares measure

- Journal for Feb 17:
  - Progress from last meeting to this Sunday (before we met):
  - Made different line types for each clamp values for visibility
  - Separated joyplots into clamps and epochs, instead of just epochs. This was to understand the stress "bimodality" better
  - Changed model to use new network values

  - Progress since Sunday to now:
  - Plotted only correct trials for bar graphs (still needs some work - need to fix error bars and labels.)
  - Started threshold search at 25 ticks instead of 0
  - Recorded accuracy for both rt's
  - Still need to address things we talked about on Sunday (Why clamp 0.5 doesn't have thresholds)

- Journal for March 3:
  - Figured that my accuracy measure & rt calculations were a bit off, so we met on Sunday to discuss
  - Implemented the Plaut method
  - 1. Figure out when RTs are generated, based on settling time falling below some threshold
    2. Get the stress values from the tick when the RT fell below some threshold.
    3. Figure out if the response at that RT was correct or incorrect based on finding the optimal stress value that separates words from nonwords
    4. you now have RTs and whether the response was accurate for every word/nonword. Plot accuracy for all data, and plot RTs only for correct responses
  - Accuracy for the above method was low for words. Also observed lower than expected rts. Needs to be at least 75 since that is when we start injecting error
  - Thus, I changed the fixed st threshold to a sweeping st threshold starting at lower values. This improved word accuracy, and delayed rt as expected
  - Need to create bar plots from this data (but not sure if I will be done by 8pm)

- Journal for March 10:
  - Plotted bar graphs for Plaut method
  - Not sure where to go from here - if there is something wrong with my code or anything since Im not exactly seeing what we are expecting
  - Started writing methods:
  - For network architecture & data generation, I forget what you said - do I need to paraphrase the entire methods of Plaut, or can I just leave it as "done as in Plaut...". I'm assuming the former

- Journal for March 24:
  - Plotted accuracy for nonwords based on "Plaut" method
  - We see the expected speed accuracy tradeoff for the 0.7 clamp
  - Continued writing methods section for project paper

- Journal for April 1:
  - Worked on writing methods for the paper
  - Questions are on the doc

Student Signature: Akathian Santhakumar
Supervisor Signature: