

Representations in neural network learning of phonology

Max Nelson, Joe Pater and Brandon Prickett

University of Massachusetts Amherst

UCLA Linguistics Colloquium
October 9, 2020



Introduction

What representations are needed for learning of phonological generalizations in neural networks (NNs)?

- This was a central issue in the applications of NNs to learning of the English past tense in Rumelhart and McClelland (1986) and in following work of that era
- The question can be addressed anew given subsequent developments in NN technology

In this talk we will cover recent research at UMass Amherst:

- Are variables needed for phonological assimilation and dissimilation? (JP based on work by Amanda Doucette)
- Are variables needed to model learning experiments involving reduplication (e.g. Marcus et al. 1999)? (BP)
- What kind of architecture is necessary for the full range of natural language reduplication? (MN)

Our general conclusion will be that standard variable-free* NN architectures handle the cases we have examined (cf. Berent et al. 2012: LI on Hayes and Wilson's MaxEnt model)

Neural net basics

We'll start with an intro to NNs from Pater (2019)

- This net maps from 2 Input features to an Output node that is either On or Off (Perceptron = NN in Rosenblatt 1957 *et seq.*)

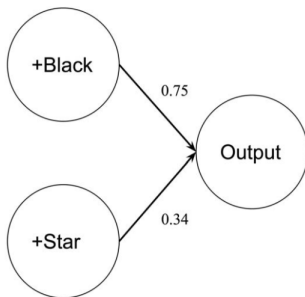


FIGURE 1. A simple perceptron for object classification.

Neural net basics

This table shows how the net responds to 4 objects, activating the Output node for just the black ones

- That is, when the weighted sum of Input feature activations exceeds the 0.5 threshold for activation of the Output node

INPUT	+Black 0.75	+Star 0.34	WEIGHTED SUM	ACTIVATION (> 0.5 input)
★	1	1	1.09	1
☆	0	1	0.34	0
◆	1	0	0.75	1
◇	0	0	0	0

TABLE 1. A perceptron classifying the set of black objects.

Neural net basics

Famously, this simple type of net cannot do 'exclusive or' XOR classification (Minsky and Papert, 1969)

- E.g. no set of weights will lead to activation of the Output node for only the objects that are +Black and +Star, but not those that have both features (black stars)

Neural net basics

The addition of a 'hidden' layer between Input and Output nodes increases the net's expressive power, allowing it to do XOR classification

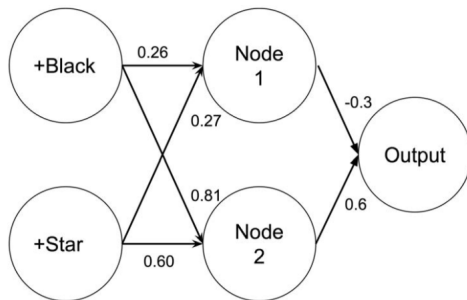


FIGURE 2. A multilayer perceptron that performs XOR classification.

Neural net basics

First, the hidden layer (Node 1 in bold):

- Node 1 is activated only for the black star
- Node 2 is activated for the black objects and stars

INPUT	+Black 0.26 0.81	+Star 0.27 0.60	WEIGHTED SUM	ACTIVATION (> 0.5)
★	1	1	0.53 1.41	1 1
☆	0	1	0.27 0.60	0 1
◆	1	0	0.26 0.81	0 1
◇	0	0	0 0	0 0

TABLE 2. Multilayer perceptron part 1: input to hidden layer. Node 1 values are in boldface.

Neural net basics

The output:

- A negative weight on Node 1 lowers the weighted sum for the black star beneath the 0.5 activation threshold
- Only the white star and black triangle activate the Output node

INPUT	NODE 1	NODE 2	WEIGHTED	ACTIVATION
	-0.3	0.6	SUM	(> 0.5)
★	1	1	0.3	0
☆	0	1	0.6	1
◆	0	1	0.6	1
◇	0	0	0	0

TABLE 3. Multilayer perceptron part 2: hidden layer to output.

Representations in RM 1986

- Rumelhart and McClelland (1986) used a single layer net; the concept of a hidden layer will be important to understand what follows (RNNs)
- They used fairly standard phonological features in their Input and Output representations (though this is usually not recognized in linguistic critiques)
- The difficulty was how to deal with ordering of features through time
- The triphone 'Wickelfeature' representations they adopted were a central focus of Pinker and Prince's (1988) critique

Recurrent neural nets (RNNs)

The now standard approach to ordering through time is the use of a recurrent neural net (RNN; Elman 1990, 1991)

At each step in a sequence, the net includes a copy of the previous step's hidden layer ('context' below) in the generation of the output

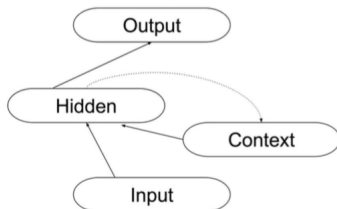


FIGURE 4. The structure of a simple recurrent neural network (adapted from Lewis & Elman 2001).

RNNs and alpha variables

Doucette (2017) shows that the adoption of an RNN has consequences for phonological representation

- With an RNN, alpha variables (Halle, 1962) may not be necessary to make assimilation and dissimilation ‘special’

Moreton (2012):

- Single feature phonotactic generalizations across segments are easier to learn than those involving multiple features

E.g. ‘if C1 is [+Voice] C2 is [+Voice]’ is easier than ‘if C1 is [+Voice] C2 is [+Coronal]’

- This can be modeled by using alpha variable representations in a feedforward net

RNNs and alpha variables

Doucette's training data, like the human experiments in Moreton (2012), had a space of 256 CVCV items, with 128 In and 128 Out (but the network saw all the data, labeled).

Pattern	Features in Pattern
1	C1+voi and V1+back C1-voi and V1-back
2	C1+voi and V2+back C1-voi and V2-back
3 ->	C1+voi and C2+voi C1-voi and C2-voi
4 ->	C1+voi and C2-voi C1-voi and C2+voi
5	C1+voi and C1+cor C1-voi and C1-cor
6 ->	C1+voi and C2+cor C1-voi and C2-cor

Table 1: Feature descriptions of patterns.

RNNs and alpha variables

Training was done using on-line gradient descent with random initial weights; learning time was to greater probability for 'In' than 'Out'.

The assimilation and dissimilation patterns (3 and 4) were learned more quickly than the analagous 2-feature pattern (6).

Pattern	Mean	St. Err.	St. Dev.
1	22.03	0.22	11.78
2	22.74	0.26	14.49
3 ->	19.91	0.29	15.88
4 ->	20.20	0.29	16.08
5	19.65	0.15	8.42
6 ->	22.46	0.23	12.79

Table 2: Number of epochs to learn patterns with full training set.

Reduplication and Neural Networks

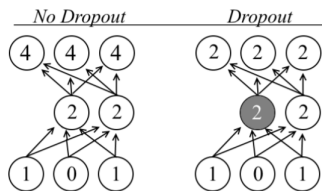
- Another phonological phenomenon that has been linked to variables is *reduplication*, which involves the copying of all or part of a word (Marcus, 2001; Berent, 2013).
- Definitions for what constitutes a variable are hard to nail down, although Marcus (1999) suggests that a variable could be any part of a model's architecture that explicitly compares the similarity of two points in time.
- Linguists have proposed several ways of representing reduplication that fall under this definition (e.g. Marantz, 1982; McCarthy and Prince, 1994).

Infant Learning of Reduplication

- Marcus et al. (1999) ran an experiment to test whether variables are necessary in models of cognition:
 - Infants were exposed to one of two reduplicative patterns: either ABB (e.g. [wofefe]) or ABA (e.g. [wofewo]).
 - They then gave the infants novel words made up of segments absent from training.
 - The infants listened longer to novel stimuli that violated their pattern, suggesting that humans can generalize reduplication to novel words and segments.
- Simple RNNs trained on the same data failed to generalize to words with novel segments.
- Marcus et al. (1999) argued that neural networks' lack of explicit variables were to blame for this lack of generalization (see also Marcus, 2001).

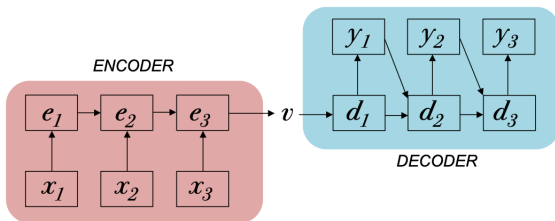
Beyond Simple RNNs

- Neural networks have advanced considerably since Marcus et al. (1999) attempted to simulate their experiment.
- Prickett, Traylor, and Pater (2020; henceforth PTP) wanted to test a more modern neural network to see if it could generalize like the infants did.
- PTP's model differed from simple RNNs in 3 ways:
 - It used a *sequence-to-sequence (Seq2Seq)* architecture (Sutskever, Vinyals, and Le, 2014).
 - It used *LSTM* layers (Hochreiter and Schmidhuber, 1997).
 - It was regularized using *dropout* (Srivastava et al., 2014).



Seq2Seq Networks

- Originally, simple RNNs could only map between strings of equal lengths, but Sutskever, Vinyals, and Le (2014) introduced Seq2Seq networks to overcome this limitation in the domain of machine translation.
- They did this by connecting two separate RNNs (an encoder and decoder) via connections in the networks' hidden layers.
- Seq2Seq networks have been useful for modeling morphological (Cotterell et al., 2016; Kirov and Cotterell, 2018) and phonological mappings (Kirov, 2017), since these often involve insertion and deletion of segments.



Modeling Marcus et al. (1999) without Explicit Variables

- To simulate the Marcus et al. experiment, PTP:
 - Pretrained a Seq2Seq model on 1,000 random words created from the syllables in the experiment.
 - Trained the model on the experiment stimuli (repeated 1,500 times), represented as vectors of phonologically informed, numerical features.
 - Measured the model's mean squared error on the test data, averaged over 50 runs in each condition (ABA and ABB).
- For both of Marcus et al.'s experiments (each of which had slightly different stimuli), the model had significantly more error on novel words that didn't follow the reduplicative pattern the model was trained on.

	Average MSE (SE)				Average listening time (SE)			
	Conf.	Nonconf.	t(99)	p	Conf.	Nonconf.	F(14)	p
Exp. 1	0.49 (0.01)	0.52 (0.01)	-2.8	<.01*	6.3 (0.65)	9.0 (0.54)	25.7	<.01*
Exp. 2	0.67 (0.01)	0.68 (0.01)	-3.3	<.01*	5.6 (0.47)	7.35 (0.68)	25.6	<.01*

Why did the Seq2Seq network succeed?

- The Seq2Seq model seems to have succeeded where past models failed (although, for similar successes, see Alhama and Zuidema, 2018; Beguš, 2020).
- To better understand why, PTP probed the network using Berent's (2013) *scopes of generalization*:
 - Novel words/syllables (predicted by any analysis other than memorization)
 - Novel segments (predicted by variables and feature-based copying)
 - Novel feature values (predicted by variables only)

Novel Syllable

	i	e	o	a
p	pi	pe	po	pa
b	bi	be	bo	ba
t	ti	te	to	ta
d	di	de	do	da

Novel Segment

	i	e	o	a
p	pi	pe	po	pa
b	bi	be	bo	ba
t	ti	te	to	ta
d	di	de	do	da

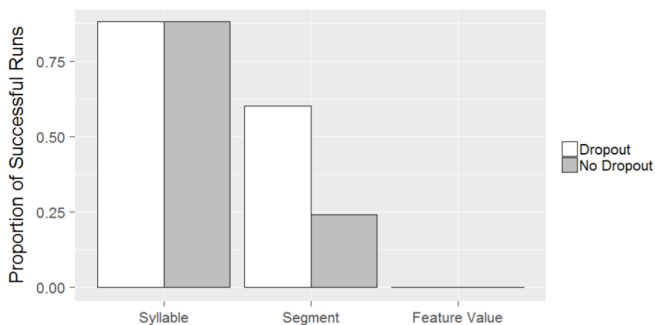
Novel Feat. Value

	i	e	o	a
p	pi	pe	po	pa
b	bi	be	bo	ba
t	ti	te	to	ta
d	di	de	do	da
n	ni	ne	no	na

Finding the Network's Scope of Generalization

- To do this, PTP trained the model on mappings of the form $C_1 V_2 \rightarrow C_1 V_2 C_1 V_2$ and then tested it to see if it could correctly map novel inputs.
- We ran 25 simulations for each of Berent's scopes of generalization, with and without dropout.
- Training data consisted of every possible word, given a randomly produced segment inventory, except the data that were being withheld for testing.
- Features in training were all either -1 or 1 , so mappings in testing were considered successful if every feature value in every segment of the output had the correct sign ($-/+$).

Scope Results



- The results above show that, without dropout, the network can only generalize consistently to novel syllables.
- But with dropout, the model is able to generalize to novel segments a majority of the time (although novel feature values are still impossible).

Dropout + Pretraining = Successful Generalization

- The results from our probe into the model's scope of generalization revealed that with dropout, the network could generalize to novel segments.
- Since most past attempts to model the Marcus et al. experiment did not use pretraining (although, see Seidenberg and Elman, 1999), their models had to generalize to novel segments in testing (and couldn't, likely due to their lack of dropout).
- Crucially, the Marcus et al. (1999) experiment didn't test generalization to novel features values.
 - We're not aware of any experiments that convincingly test humans for this scope of generalization (although, see Berent et al., 2002; Berent, Dupuis, and Brentari, 2014; Berent et al., 2016, for claims to the contrary).

Computing natural language reduplication

- PTP showed a seq2seq model that can generalize reduplication
- Gasser (1993) failed to get RNNs to learn a similar reduplication pattern, and proposed that 'a variable of some sort' might be needed
- Weiss, Goldberg, and Yahav (2018) showed that with finite precision and saturating activations, RNNs are regular
 - Natural language (total) reduplication is not regular, so it should not be computable by an RNN
 - PTP focus on a case with a small alphabet and fixed string and reduplicant lengths - regular reduplication
- Is PTPs success attributable to the fact that they test a simple reduplication pattern?

Computing natural language reduplication

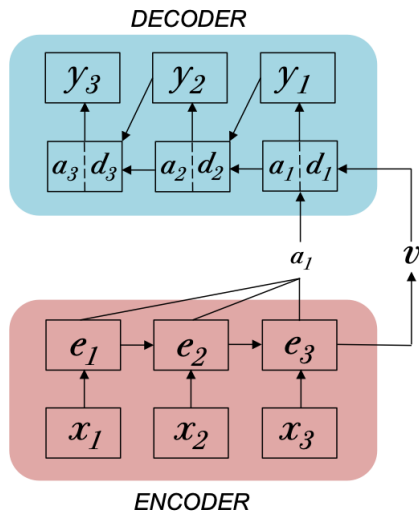
- Merrill et al. (2020) show that adding a decoder to an RNN increases its expressivity
 - Only discusses single non-linear layer, no analysis of RNN decoder
 - Weiss et al. (2018) does not preclude the possibility that seq2seqs can generalize total reduplication
- The component RNNs in a seq2seq are regular, natural language reduplication is not, but the complete seq2seq architecture may be more expressive than its parts.
- So can seq2seq networks without variables learn natural language reduplication?
 - Nelson et al. (2020) scale up seq2seq reduplication to realistic scale and complexity

Pushing seq2seq reduplication

- Unlike PTP, who focus on Marcus et al. (1999)'s experiments and consequently only test fixed-size reduplicants, Nelson et al. (2020) test cases in which reduplicant size is variable
 - Fixed window, initial foot, and total reduplication
- Seq2seq models trained/tested on 7000 + 3000 unique reduplication I-O mappings generated from a library of transducers (Dolatian and Heinz, 2019)
- Manipulate alphabet size and maximum string length
- Not explicitly looking at the use of variables. However we are testing networks with and without **attention**

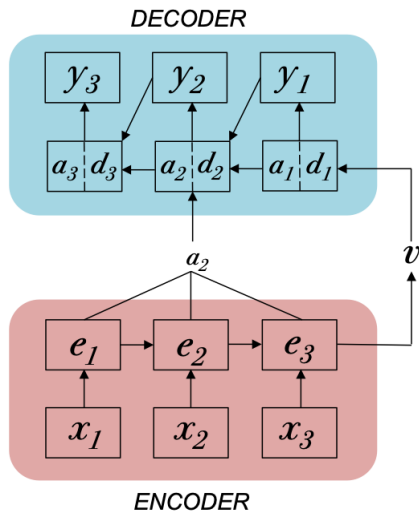
Attention

- Unlike the normal seq2seq architecture, a seq2seq with attention can look back at states in the first RNN
- The network directly compares its states while reading inputs to its current state while predicting an output symbol
- Resulting weights look very much like input-output correspondences



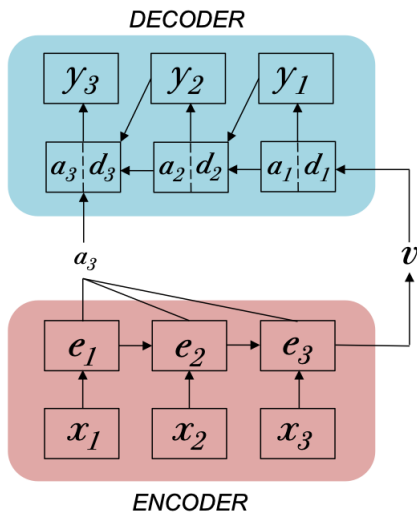
Attention

- Unlike the normal seq2seq architecture, a seq2seq with attention can look back at states in the first RNN
- The network directly compares its states while reading inputs to its current state while predicting an output symbol
- Resulting weights look very much like input-output correspondences



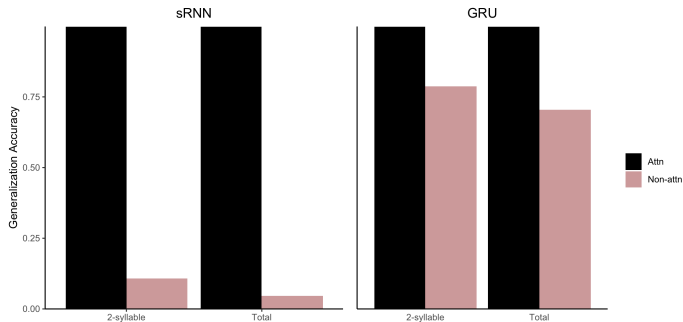
Attention

- Unlike the normal seq2seq architecture, a seq2seq with attention can look back at states in the first RNN
- The network directly compares its states while reading inputs to its current state while predicting an output symbol
- Resulting weights look very much like input-output correspondences



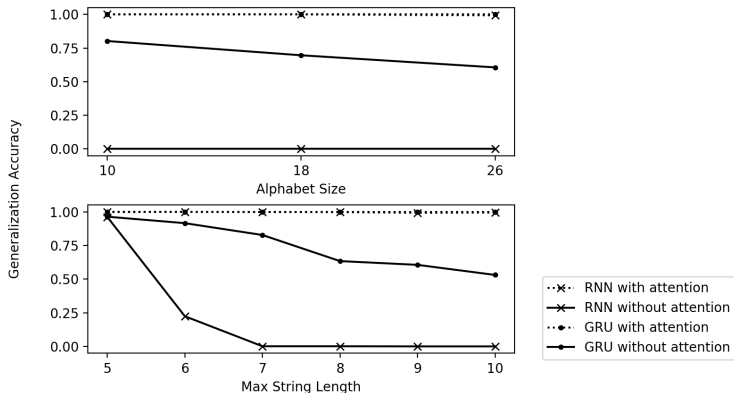
Reduplication with and without attention

- First tested total, and 2-syllable on a small language with a 9 symbol alphabet and maximum string length of 10
- Longer strings than PTP, and variable sized reduplicants, but no novel segments/features
- Models with attention generalize, models without do not



Fixed window reduplication

- Generalizing fixed-size reduplication (as in PTP) but with variable length strings (not in PTP), we see that non-attention models fail with realistic alphabet sizes and string lengths
- Still not generalizing to novel segments/features, in that sense an easier task than PTP



Attention and variables

- One of Marcus et al. (1999)'s original diagnostics for a variable was that it explicitly compares the similarity of two points in time
 - If this is accepted, seq2seq networks with attention are not variable-free
- Shultz and Bale (2001) describe Marcus et al. (1999)'s definition as 'highly idiosyncratic', however they also define variables as having bindings that are preserved and accessible to future computation
 - A seq2seq model with attention saves the intermediate states of the encoder but not their bindings.
- Endress, Dehaene-Lambertz, and Mehler (2007) suggest that Shultz and Bale (2001) use variables, because their model encodes phoneme positions in the inputs.
 - In recurrent models the network has no way of knowing the position of the current input in the sequence
- Attention may be considered a variable, but only under a very specific definition

Discussion

- To summarize, we presented results here that showed:
 - A simple RNN capturing a learning bias previously explained using variables (i.e. *alpha notation*).
 - A variable-free Seq2Seq network capturing experiment results involving the generalization of reduplication.
 - A network with attention being able to handle reduplicative patterns found in natural language.
- The question of whether NN attention is formally equivalent to a variable needs further work
- This question of whether variables are needed to capture phonology has been the subject of a number of recent studies that we'd like to look at with these models in future work:
 - Phonotactic Generalization (Berent et al., 2012; Gallagher, 2013) and learning biases (Gallagher, 2013)
 - Cross-modal generalization (Berent, Dupuis, and Brentari, 2014; Berent et al., 2016)

References I



Alhama, Raquel G. and Willem Zuidema (2018). “Pre-Wiring and Pre-Training: What does a neural network need to learn truly general identity rules?” In: Journal of Artificial Intelligence Research 61, pp. 927–946.



Beguš, Gašper (2020). Identity-Based Patterns in Deep Convolutional Networks: Generative A arXiv: 2009.06110 [cs.CL].



Berent, Iris (2013). “The phonological mind”. In: Trends in cognitive sciences 17.7, pp. 319–327.








Berent, Iris, Amanda Dupuis, and Diane Brentari (2014). “Phonological reduplication in sign language: Rules rule”. In: Frontiers in psychology 5, p. 560.



Berent, Iris et al. (2002). “The scope of linguistic generalizations: Evidence from Hebrew word formation”. In: Cognition 83.2, pp. 113–139.

References II

-  Berent, Iris et al. (2012). “On the role of variables in phonology: Remarks on Hayes and Wilson 2008”. In: Linguistic inquiry 43.1, pp. 97–119.
-  Berent, Iris et al. (2016). “The double identity of linguistic doubling”. In: Proceedings of the National Academy of Sciences 113.48, pp. 13702–13707.
-  Cotterell, Ryan et al. (2016). “The SIGMORPHON 2016 shared task—morphological reinflection”. In: Proceedings of the 14th SIGMORPHON Workshop on Computational Morphology, pp. 10–22.
-  Dolatian, Hossep and Jeffrey Heinz (2019). “Reduplication with finite-state technology”. In: Proceedings of the 53rd Annual Meeting of the Chicago Linguistics Society.
-  Doucette, Amanda (2017). “Inherent Biases of Recurrent Neural Networks for Phonological Assimilation and Dissimilation”. In: arXiv preprint arXiv:1702.07324.

References III



Elman, Jeffrey L. (1990). "Finding structure in time". In: Cognitive science 14.2, pp. 179–211.



Elman, Jeffrey L (1991). "Distributed representations, simple recurrent networks, and grammatical structure". In: Machine learning 7.2-3, pp. 195–225.



Endress, Ansgar D., Ghislaine Dehaene-Lambertz, and Jacques Mehler (2007). "Perceptual constraints and the learnability of simple grammars". In: Cognition 105.3, pp. 577–614.








Gallagher, Gillian (2013). "Learning the identity effect as an artificial language: bias and generalisation". In: Phonology 30.2, pp. 253–295.










Gasser, Michael (1993).

Learning words in time: Towards a modular connectionist account of the
Indiana University, Department of Computer Science.

References IV

-  Halle, Morris (1962). “A descriptive convention for treating assimilation and dissimilation”. In: Quarterly Progress Report 66, pp. 295–296.
-  Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory”. In: Neural computation 9.8, pp. 1735–1780.
-  Kirov, Christo (2017). “Recurrent Neural Networks as a Strong Domain-General Baseline for Morpho-Phonological Learning”. In: Poster presented at the 2017 Meeting of the Linguistic Society of America
-  Kirov, Christo and Ryan Cotterell (2018). “Recurrent Neural Networks in Linguistic Theory: Revisiting Pinker & Prince (1988) and the Past Tense Debate”. In: Transactions of the Association for Computational Linguistics. Vol. 6, pp. 651–665.
-  Marantz, Alec (1982). “Re reduplication”. In: Linguistic inquiry 13.3, pp. 435–482.

References V

-  Marcus, Gary (1999). “Do infants learn grammar with algebra or statistics? Response”. In: Science 284.5413, pp. 436–437.
-  – (2001). The algebraic mind. Cambridge, MA: MIT Press.
-  Marcus, Gary et al. (1999). “Rule learning by seven-month-old infants”. In: Science 283.5398, pp. 77–80.
-  McCarthy, John J. and Alan S. Prince (1994). “The emergence of the unmarked: Optimality in prosodic morphology”. In:
-  Merrill, William et al. (2020). “A Formal Hierarchy of RNN Architectures”. In: arXiv preprint arXiv:2004.08500.
-  Minsky, Marvin and Seymour Papert (1969). Perceptrons: An Introduction to Computational Geometry. Cambridge, MA, USA: MIT Press.
-  Moreton, Elliott (2012). “Inter-and intra-dimensional dependencies in implicit phonotactic learning”. In: Journal of Memory and Language 67.1, pp. 165–183.

References VI



Nelson, Max et al. (2020). “Probing RNN Encoder-Decoder Generalization of Subregular Functions using Reduplication”. In: Proceedings of the Society for Computation in Linguistics 3.1, pp. 31–42.



Pater, Joe (2019). “Generative linguistics and neural networks at 60: foundation, friction, and fusion”. In: Language 95.1, e41–e74. DOI: [10.1353/lan.2019.000](https://doi.org/10.1353/lan.2019.000).



Pinker, Steven and Alan Prince (Mar. 1988). “On language and connectionism: Analysis of a parallel distributed processing model of language acquisition”. In: Cognition 28.1, pp. 73–193. ISSN: 0010-0277. DOI: [10.1016/0010-0277\(88\)90032-7](https://doi.org/10.1016/0010-0277(88)90032-7). URL: <http://www.sciencedirect.com/science/article/pii/0010027788900327> (visited on 01/18/2018).



Rosenblatt, Frank (1957). The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory.

References VII



Rumelhart, DE and JL McClelland (1986). “On learning the past tenses of English verbs”. In: Parallel Distributed Processing: Explorations in the Microstructure of C Ed. by JL McClelland and DE Rumelhart. Vol. 2: Psychological and Biological Models. The MIT Press, pp. 216–271.



Seidenberg, Mark S. and Jeff L. Elman (1999). “Do infants learn grammar with algebra or statistics?” In: Science 284.5413, 433f–433f.



Shultz, Thomas R. and Alan C. Bale (2001). “Neural network simulation of infant familiarization to artificial sentences: Rule-like behavior without explicit rules and variables”. In: Infancy 2.4, pp. 501–536.



Srivastava, Nitish et al. (2014). “Dropout: A simple way to prevent neural networks from overfitting”. In: The Journal of Machine Learning Research 15.1, pp. 1929–1958.

References VIII



Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014). “Sequence to sequence learning with neural networks”. In: Advances in neural information processing systems, pp. 3104–3112.



Weiss, Gail, Yoav Goldberg, and Eran Yahav (2018). “On the practical computational power of finite precision RNNs for language recognition”. In: arXiv preprint arXiv:1805.04908.

Appendix I: Sample Attention Weights

