How much complexity does an RNN architecture need to learn syntax-sensitive dependencies?

Gantavya Bhatt*[1], Hritik Bansal*[1], Rishubh Singh*[1], Sumeet Agarwal[1]
[1] Indian Institute of Technology Delhi
{gantavya.iitd, hbansal10n, rishubhsingh135}@gmail.com
sumeet@iitd.ac.in

Abstract

Long short-term memory (LSTM) networks and their variants are capable of encapsulating long-range dependencies, which is evident from their performance on a variety of linguistic tasks. On the other hand, simple recurrent networks (SRNs), which appear more biologically grounded in terms of synaptic connections, have generally been less successful at capturing long-range dependencies as well as the loci of grammatical errors in an unsupervised setting. In this paper, we seek to develop models that bridge the gap between biological plausibility and linguistic competence. We propose a new architecture, the *Decay RNN*, which incorporates the decaying nature of neuronal activations and models the excitatory and inhibitory connections in a population of neurons. Besides its biological inspiration, our model also shows competitive performance relative to LSTMs on subject-verb agreement, sentence grammaticality, and language modeling tasks. These results provide some pointers towards probing the nature of the inductive biases required for RNN architectures to model linguistic phenomena successfully.

1 Introduction

For the last couple of decades, neural networks have been approached primarily from an engineering perspective, with the key motivation being efficiency, consequently moving further away from biological plausibility. Recent developments (Song et al., 2016; Gao and Ganguli, 2015; Sussillo and Barak, 2013) have however incorporated explicit constraints in neural networks to model specific parts of the brain and have found a correlation between the learned activation maps and actual neural activity recordings. Thus, these trained networks can perhaps act as a proxy for a theoretical investigation into biological circuits.

Recurrent Neural Networks (RNNs) have been used to analyze the principles and dynamics of neural population responses by performing the same tasks as animals (Mante et al., 2013). However, these networks violate Dale's law (Dale, 1935; Strata and Harvey, 1999), which states that the neurons have either a purely excitatory or inhibitory effect on other neurons in the mammalian brain. The decaying nature of the potential in the neuron membrane after receiving signals (excitatory or inhibitory) from the surrounding neurons is also well-studied (Gluss, 1967). The goal of our work is to incorporate these biological features into the RNN structure, which gives rise to a neuro-inspired and computationally inexpensive recurrent network for language modeling, which we call a *Decay RNN* (Section 4). We perform learning using the backpropagation algorithm. Despite its differences with the way learning is believed to happen in the brain, it has been argued that the brain can implement its core principles (Hinton, 2007; Lillicrap et al., 2020). We assess our model's ability to capture syntax-sensitive dependencies via multiple linguistic tasks (Section 6): number prediction, grammaticality judgement (Linzen et al., 2016) which entails subject-verb agreement, and a more complex language modeling task (Marvin and Linzen, 2018).

Subject-verb agreement, where the *main noun* and the *associated verb* must agree in number, is considered as evidence of hierarchical structure in English. This is exemplified using a sentence taken from the dataset made available by Linzen et al. (2016):

1. *All **trips** on the expressway **requires** a toll.
2. All **trips** on the expressway **require** a toll.

The effect of agreement attractors (nouns having number opposite to the main noun; *expressway*

*Equal Contribution

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 244–254

in the above example[1]) between the main noun and main verb of a sentence has been well-studied (Linzen et al., 2016; Kuncoro et al., 2018). Our work also highlights the influence of non-attractor intervening nouns. For example,

- A **chair** created by a hobbyist as a gift to someone **is** not a commodity.[2]

In the number prediction task, if a model correctly predicts the grammatical number of the verb (singular in case of 'is'), it might be due to the (helpful) interference of non-attractor intervening nouns ('hobbyist', 'gift', 'someone') rather than necessarily capturing its dependence the main noun ('chair'). From our investigation in Section 6.2, we find that the linear recurrent models take cues present in the vicinity of the main verb to predict its number, apart from the agreement with the main noun.

In the subsequent sections, we investigate the performance of the Decay RNN and other recurrent networks, showing that no single sequential model generalizes well on all (grammatical) phenomena, which include subject-verb agreements, reflexive anaphora, and negative polarity items as described in Marvin and Linzen (2018). Our major outcomes are:

1. Designing a relatively simple and bio-inspired recurrent model: the Decay RNN, which performs on-par with LSTMs for linguistic tasks such as subject-verb agreement and grammaticality judgement.

2. Pointing to some limitations of analyzing the intervening attractor nouns alone for the subject-verb agreement task and attempting joint analysis of non-attractor intervening nouns and attractor nouns in the sentence.

3. Showing that there is no linear recurrent scheme which generalizes well on a variety of sentence types and motivating research in better understanding of the nature of biases induced by varied RNN structures.

2 Related Work

There has been prior work on using LSTMs (Hochreiter and Schmidhuber, 1997) for language modeling tasks. The work of Gers and Schmidhuber (2001) has shown that LSTMs can learn simple context-free and context-sensitive languages. However, as per the investigations carried out in Kuncoro et al. (2018), it was observed that if the model capacity is not enough, then LSTMs may not generalize the long-range dependencies. Recently many architectures have explicitly incorporated the knowledge of phrase structure trees (Kuncoro et al., 2018; Alvarez-Melis and Jaakkola, 2017; Tai et al., 2015) which have shown improvement in generalizing over long-range dependencies. At the same time, Shen et al. (2019) proposed ON-LSTMs, a modification to LSTMs that provides an inductive tree bias to the structure. However, Dyer et al. (2019) have shown that the success of ON-LSTMs was due to their proposed metric to analyze the model, not necessarily due to their architecture.

From the biological point of view, Capano et al. (2015) used a hard reset of the membrane potential in contrast to a soft decay observed in a neuronal membrane. At the same time, their learning paradigm is similar to the Hebbian learning scheme (Hebb, 1949), which does not involve error backpropagation (Rumelhart et al., 1986). Our work is closely related to the idea of modeling the population of neurons as a dynamical system (EIRNN) proposed by Song et al. (2016). However, their time constant parameter was based on the concepts described in Wang (2002) while the sampling rate was arbitrarily chosen. Given that the chosen values only considered a certain class of neurons (Yang et al., 2019), we believe that it is not necessary to have the same values of the parameters for each cognitive task. Thus, we build on their formulation by making the sampling rate and time constant learnable as manifested by our decay parameter, described in the next section.

3 Biological Preliminaries

According to Dale's principle, a neuron is either excitatory or inhibitory (Eccles, 1976). If a neuron output produces a negative (positive) change in the membrane potential of all the connected neurons via its synapse, then it is said to be an inhibitory (excitatory) neuron. In a set of N neurons, if $\mathbf{W}$ is the synaptic connection matrix, then the connection from the neuron j to neuron i is 'excitatory' if $W_{ij} > 0$, and 'inhibitory' if $W_{ij} \leq 0$. Capano et al. (2015) have argued that a balance between structural and response variability (entropy),

[1] Main noun and verb are highlighted in bold. Intervening nouns are underlined. Asterisks mark unacceptable sentences.

[2] Sentence taken from the dataset made available by Linzen et al. (2016).

and excitability (synaptic strength) of a network maximizes the overall learning. This balance is governed by the ratio of inhibitory and excitatory neurons. They have further shown that this balance also maximizes the overall performance in multitask learning. Catsigeras (2013) mathematically prove that Dale's principle is necessary for an optimal[3] neuronal network's dynamics.

In the postsynaptic neuron, the integration of synaptic potentials is realized by the addition of excitatory (+ve) and inhibitory (-ve) postsynaptic potentials (PSPs). PSPs are electronic voltages, that decay as a function of time due to spontaneous reclosure of the synaptic channels. The decay of the PSPs is controlled by the membrane constant τ, i.e., the time required by the PSP to decay to 37% of its peak value (Wallisch et al., 2009).

4 Decay RNN

Here we present our proposed architecture, which we call the *Decay RNN* (DRNN). Our architecture aims to model the decaying nature of the voltage in a neuron membrane after receiving impulses from the surrounding neurons. At the same time, we incorporate Dale's principle in our architecture. Thus, our model captures both the microscopic and macroscopic properties of a group of neurons. Adhering to the stated phenomena, we define our model with the following update equations for given input $\mathbf{x}^{(t)}$ at time t:

$$\mathbf{c}^{(t)} = (ReLU(\mathbf{W})\mathbf{W}_{dale})\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} + \mathbf{b}$$
$$\mathbf{h}^{(t)} = f(\alpha\mathbf{h}^{(t-1)} + (1-\alpha)\mathbf{c}^{(t)})$$

Here f is a nonlinear activation function, $\mathbf{W}$ and $\mathbf{U}$ are weight matrices, $\mathbf{b}$ is the bias and $\mathbf{h}^{(t)}$ represents the hidden state (analogous to voltage). We define $\alpha \in (0,1)$ as a learnable parameter to incorporate a decay effect in the hidden state (analogous to the decay in the membrane potential). Here α acts as a balancing factor between the hidden state $\mathbf{h}^{(t-1)}$ and $\mathbf{c}^{(t)}$.[4] $\mathbf{W}_{dale}$ is a diagonal matrix, and based on the empirical results on the mammalian brain (Hendry and Jones, 1981), we set the last 20% of entries to -1, representing the inhibitory connections, and the rest to 1 (See Appendix A.3).[5] Unlike Song et al. (2016), we keep self-connections in the network. Besides biological inspiration, our model also has the following salient features.

First, the presence of α acts as a coupled gating mechanism to the flow of information (Figure 1), at the same time maintaining an exponential moving average of the hidden state. Thus, α values close to 1 correspond to memories of the distant past. It is worth mentioning that Oliva et al. (2017) have considered the exponential moving average in the context of RNNs. However, their approach manually selected a set of scaling parameters, whereas we have a systematic way of arriving at the values of those parameters by making them learnable for the task at hand.

Second, our model also has an intrinsic skip connection deriving out of its formulation.Yue et al. (2018) has shown that the architectures with skip connections provide an alternate path for the flow of gradients during the error backpropagation. At the same time presence of coupled gates slows down the vanishing of gradient (Bengio et al., 2013). Thus, despite of its simple un-gated structure, the features discussed above provide safeguards against vanishing gradient.

To examine the importance of Dale's principle in the learning process, we made a variant of our Decay RNN without Dale's principle, which we call the *Slacked Decay RNN* (SDRNN), with updates to $\mathbf{c}^{(t)}$ made as follows:

$$\mathbf{c}^{(t)} = \mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} + \mathbf{b}$$

To understand the role of the correlation between the hidden states in the Decay RNN formulation, we devised an ablated version of our architecture, which we refer to as the *Ab-DRNN*. With the following update equation, we remove the mathematical factor ($\mathbf{W}\mathbf{h}^{(t-1)}$) that gives rise to a correlation between hidden states:

$$\mathbf{h}^{(t)} = f(\alpha\mathbf{h}^{(t-1)} + (1-\alpha)(\mathbf{U}\mathbf{x}^{(t)} + \mathbf{b}))$$

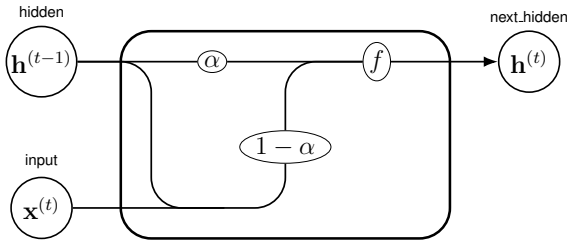

Figure 1: Decay RNN cell, comprising of a skip connection and coupled scalar gates.

5 Datasets

For the number prediction (Section 6.1) and grammaticality judgment (Section 6.3) tasks, we used a

[3] In the sense of showing the most diverse set of responses.

[4] It was kept bounded using a sigmoid function. Our results did not change when we used a linear function instead.

[5] Our results did not change when we chose a different set of -1 entries instead of the last 20%.

corpus of 1.57 million sentences from Wikipedia (Linzen et al., 2016), of which 10% were used for training, 0.4% for validation, and the remaining were reserved for testing. On the other hand, for the language modeling task (Section 6.4), the model was trained on a 90 million word subset of Wikipedia comprising of 3 million training and 0.3 million validation sentences (Gulordava et al., 2018).

Despite having a large number of training points, these datasets have certain drawbacks, including the lack of a sufficient number of syntactically challenging examples leading to poor generalization over the sentences out of the training data distribution. Therefore, we construct a generalization set as described in Marvin and Linzen (2018), where we generate the sentences out of templates that can be described using a non-recursive context-free grammar. The use of the generalization set allows us to test on a much broader range of linguistic phenomena. We will use this dataset for the targeted syntactic evaluation of our trained models.

6 Experiments

Here we will describe our experiments[6] to assess the models' ability to capture syntax-sensitive dependencies. Details regarding the training settings are available in Appendix A.4.

6.1 Number Prediction Task

The number prediction task was proposed by Linzen et al. (2016). In this task, the model is required to predict the grammatical number of the verb when provided a sentence up to the verb.

1. The **path** to success **is** not straight forward.
2. The path to success _____

The model will take the second sentence as input and has to predict the number of the verb (here, singular). Table 1 shows the results on the number prediction task. All the models including SRNs performed well on this task. Thus, this indicates that even vanilla RNNs can identify singular and plural words and can associate the main subject with the upcoming verb.

6.2 Joint Analysis of Intervening Nouns

So far in the literature, when looking at intervening material in agreement tasks, the research has tended to focus on agreement attractors, the intervening nouns with the opposite number to the main noun (Kuncoro et al., 2018). However, we posit that the role of non-attractor intervening nouns may also be important when understanding a model's decisions. For long-range dependencies in agreement tasks, a model may be influenced by the presence of non-attractor intervening nouns instead of purely capturing the verb's relationship with the main subject. Hence an analysis done solely based on the number of agreement attractors may be misleading. Table 2 shows an improvement in the verb number prediction accuracy with an increasing number of non-attractors (n), even as the subject-verb distance and the attractor count are kept fixed. This indicates that the models are also using cues present in the vicinity of the main verb to predict its number, apart from agreement with the main noun.

Model	No. Prediction	Grammaticality
SRN	97.70	50.12
LSTM	98.59	95.81
GRU	**98.81**	94.26
EIRNN	94.68	84.51
DRNN	98.66	95.48
SDRNN	98.65	**96.83**
Ab-DRNN	97.37	85.98

Table 1: % Accuracy of models when tested on $\sim$ 1.4 million sentences for the number prediction and grammaticality judgement tasks.

Model	n=0	n=1	n=2
DRNN	90.65	95.56	96.06
LSTM	90.4	95.56	95.63

Table 2: Number prediction % accuracy with an increasing number of non-attractor intervening nouns (n). The distance between the main subject and the corresponding verb is held constant at 7 and the attractor count at 1.

6.3 Grammaticality Judgement

The previous objective was predicting the grammatical number of the verb after providing the model an input sentence only up to the verb. However, this way of training may give the model a cue to the syntactic clause boundaries. In this section, we describe the grammaticality judgment task. Given an input sentence, the model has to predict whether it is grammatical or not. To perform well on this task, the model would presumably need to allocate more resources to determine the locus of ungrammaticality. For example, consider the following

[6] Our code is available at https://github.com/bhattg/Decay-RNN-ACL-SRW2020

pair of sentences[2] :

1. The **roses** in the vase by the door **are** red.
2. *The **roses** in the vase by the door **is** red.

The model has to decide, for input sentences such as the above, whether each one is grammatically correct or not. Table 1 shows the performance of different recurrent architectures on this task. It can be seen that SRNs, which were comparable to LSTMs and GRUs on the prediction experiment described in Section 6.1, are no better than random on the grammaticality judgment task. On the other hand, the Ab-DRNN performed better than the SRN. This highlights the importance of a balance between the uncorrelated hidden states ($\mathbf{h}^{(t)}$), and the connected hidden states ($\mathbf{W}\mathbf{h}^{(t)}$), which is modeled by the Decay RNN. Due to its architectural similarity with the Independent RNN (Li et al., 2018), which has independent connections among neurons in a layer, Ab-DRNN did not suffer from the vanishing gradient problem.

Importance of the generalization set

Capano et al. (2015) had argued that the inclusion of Dale's principle improved generalization abilities for multitask learning. For our models trained on a single task, we use the generalization set to determine the number prediction confidence profile over the sentences. Figure 2 describes the average number prediction confidence at each part of speech for all prepositional phrases with inanimate subjects. We note the anomalously low confidence of the SDRNN at plural inanimate subjects (like 'movies', 'books'), unlike the DRNN.

Task	DRNN	SDRNN
Across object RC (no that) anim	**0.45**	0.28
Reflexive Sentential Comp.	**0.65**	0.6
Long VP Coordination	**0.53**	0.43

Table 3: Accuracy comparison of DRNN and SDRNN when tested on the generalization set for the grammaticality judgement task; 'anim' refers to an animated noun.

In Table 3,[7] we present the result of the models trained for the grammaticality judgment task and tested on the synthetic generalization set. From the results, we can see that despite having nearly the same accuracy on the original testing data (Table 1), there is a substantial difference in the generalization accuracies of the DRNN and SDRNN. The DRNN shows better generalization than the SDRNN in the experiments mentioned in Table 3 and Figure 2. This might be due to regularising effects induced by Dale's constraint. This is an interesting observation that merits further investigation.

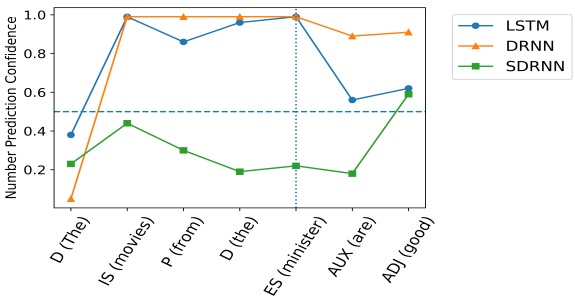

Figure 2: Number prediction confidence (for the correct verb number) averaged over the generalization set (540 sentences) for prepositional phrases with plural inanimate subjects (IS). An example word for each position is indicated in parentheses. Values at ES indicate the confidence for the following verb/auxiliary. For the example sentence, confidence < 0.5 implies singular verb number prediction, and confidence > 0.5 plural.

6.4 Language Modeling

Word-level language modeling is a task that helps in the evaluation of the model's capacity to capture the general properties of language beyond what is tested in specialized tasks focused on, e.g., subject-verb agreement. We use perplexity to compare our model's performance against standard sequential recurrent architectures. Table 4 shows the validation perplexity of different language models along with the number of learnable parameters for the task. From the Table 4, we observe that incorporating the components of the Ab-DRNN and the SRN in a coupled way might have led to the improved performance of the Decay RNN.

6.5 Targeted Syntactic Evaluation

Targeted syntactic evaluation (Marvin and Linzen, 2018) is a way to evaluate the language model across different classes of structure-sensitive phenomena. This includes subject-verb agreement, reflexive anaphora, and negative polarity items (NPI).[8] Table 4 shows that even with a simple architecture, the Decay RNN class of models performs

[7] Here, we present three tests from the targeted syntactic evaluation framework. Others test results can be found in Appendix A.2.

[8] The definitions of these linguistic terms are provided in the supplementary material of Marvin and Linzen (2018).

	SRN	GRU	LSTM	**DRNN**	**SDRNN**	**Ab-DRNN**	ON-LSTM
Validation Perplexity	114.74	53.78	52.73	76.67	76.88	86.42	-
Parameters	1.4M	4.2M	5.6M	1.4M	1.4M	0.55M	-
Short-Range Dependency							
SV Agreement:							
Simple	0.88	0.95	0.92	0.95	0.97	0.90	**0.99**
Sentential Complement	0.84	0.86	0.93	0.89	0.92	0.85	**0.95**
Short VP Coord	0.5	0.87	0.85	0.73	0.77	0.69	**0.89**
In an object RC	0.59	0.75	**0.87**	0.77	0.74	0.63	0.84
In an object RC (no that)	0.57	0.67	0.75	0.74	0.71	0.62	**0.78**
Reflexive Anaphora:							
Simple	0.51	0.85	0.85	0.75	0.73	0.63	**0.89**
Sentential Complement	0.56	0.78	0.83	0.68	0.65	0.62	**0.86**
Negative Polarity Items :							
Simple (grammatical vs. intrusive)	0.01	0.51	**0.56**	0.25	0.01	0.29	0.18
Simple (intrusive vs. ungrammatical)	**0.7**	0.66	0.48	0.54	0.5	0.51	0.5
Simple (grammatical vs. ungrammatical)	0.11	**0.67**	0.55	0.45	0.38	0.31	0.07
Long-Range Dependency							
SV Agreement:							
Long VP coordination	0.51	**0.8**	**0.8**	0.55	0.62	0.51	0.74
Across a PP	0.51	**0.75**	0.6	0.56	0.54	0.53	0.67
Across a subject RC	0.52	**0.67**	**0.67**	0.53	0.55	0.52	0.66
Across an object RC	0.51	0.51	0.55	**0.64**	0.58	0.57	0.57
Across an object RC (no that)	0.50	0.50	0.51	**0.65**	0.60	0.59	0.54
Reflexive Anaphora :							
Across a RC	0.51	0.58	0.57	0.62	**0.66**	0.58	0.57
Negative Polarity Items:							
Across a RC (grammatical vs. intrusive)	**0.87**	0.55	0.55	0.32	0.48	0.57	0.59
Across a RC (intrusive vs. ungrammatical)	0.02	0.29	0.22	**0.5**	0.37	0.36	0.20
Across a RC (grammatical vs. ungrammatical)	0.1	0.2	0.03	0.1	**0.3**	0.11	0.11
Mean Arithmetic Rank	5.94	3	3.31	3.52	3.68	4.73	**2.94**

Table 4: Accuracy of models on targeted syntactic evaluation. RC: Relative Clause, PP: Prepositional Phrase, VP : Verb Phrase. Closeness in the mean arithmetic rank of models (other than SRNs) across tasks suggests that within the current space of sequential recurrent models, none dominates the others.

fairly similarly to LSTMs and much better than SRNs for many tests.[9] In the case of long-range dependencies and NPI involving relative-object clauses, our models perform substantially better than LSTMs. High variability in the performance of the models in the case of NPIs might be due to non-syntactic cues as pointed out by Marvin and Linzen (2018). Based on the mean ranks observed in Table 4, we conjecture that there is no sequential recurrent structure at present which outperforms the others across the board. However, SRNs alone are not sufficient for most purposes.

7 Conclusion

In this paper, we proposed the Decay RNN, a bio-inspired recurrent network that emulates the decaying nature of neuronal activations after receiving excitatory and inhibitory impulses from upstream neurons. We have found that the balance between the free term ($\mathbf{h}^{(t)}$) and the coupled term ($\mathbf{W}\mathbf{h}^{(t)}$) enabled the model to capture syntax-level dependencies. As shown by McCoy et al. (2020); Kuncoro et al. (2018), explicitly modeling hierarchical structure helps to discover non-local structural dependencies. The contrast in the performance of the language models encourages us to look at the inductive biases, which might have led to better syntactic generalization in certain cases. Recently, Maheswaranathan and Sussillo (2020) showed the existence of a line attractor in the dynamics of the hidden states for sentiment classification. Thus, similar dynamical-system-based analysis can be extended to our settings to further understand the working of the Decay RNN.

From the cognitive neuroscience perspective, it would be interesting to investigate if the proposed Decay RNN can capture some aspects of actual neuronal behaviour and language cognition. Our results here do at least indicate that the complex gating mechanisms of LSTMs (whose cognitive plausibility has not been established) may not be essential to their performance on many linguistic tasks, and that simpler and perhaps more cognitively plausible RNN architectures are worth exploring further as psycholinguistic models.

Acknowledgements

We wish to thank the anonymous reviewers, and Jakob Prange and ACL SRW for the post-acceptance mentorship program; Pankaj Malhotra for valuable comments on earlier versions of this paper; and Tal Linzen for helpful discussion.

[9] Results for the ON-LSTM are directly quoted from Shen et al. (2019).

References

David Alvarez-Melis and Tommi S. Jaakkola. 2017. Tree-structured decoding with doubly-recurrent neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*. OpenReview.net.

Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. 2013. Advances in optimizing recurrent networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8624–8628. IEEE.

Vittorio Capano, Hans J Herrmann, and Lucilla De Arcangelis. 2015. Optimal percentage of inhibitory synapses in multi-task learning. *Scientific Reports*, 5:9895.

Eleonora Catsigeras. 2013. Dale's principle is necessary for an optimal neuronal network's dynamics. *Applied Mathematics*, 4(10B):15–29.

Henry Dale. 1935. Pharmacology and nerve-endings. *Proceedings of the Royal Society of Medicine*, 28(3):319–332.

Chris Dyer, Gábor Melis, and Phil Blunsom. 2019. A critical analysis of biased parsers in unsupervised parsing. *arXiv preprint*, arXiv:1909.09428.

John Carew Eccles. 1976. From electrical to chemical transmission in the central nervous system: the closing address of the sir henry dale centennial symposium cambridge, 19 september 1975. *Notes and records of the Royal Society of London*, 30(2):219–230.

Peiran Gao and Surya Ganguli. 2015. On simplicity and complexity in the brave new world of large-scale neuroscience. *Current Opinion in Neurobiology*, 32:148–155.

Felix A Gers and E Schmidhuber. 2001. LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340.

Brian Gluss. 1967. A model for neuron firing with exponential decay of potential resulting in diffusion equations for probability density. *The Bulletin of Mathematical Biophysics*, 29(2):233–243.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Donald O Hebb. 1949. *The organization of behavior: A neuropsychological theory*. New York: John Wiley & Sons.

Stewart H Hendry and EG Jones. 1981. Sizes and distributions of intrinsic neurons incorporating tritiated gaba in monkey sensory-motor cortex. *Journal of Neuroscience*, 1(4):390–408.

Geoffrey Hinton. 2007. How to do backpropagation in a brain. Invited talk at the NIPS'2007 Deep Learning Workshop.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. 2018. Independently recurrent neural network (IndRNN): Building a longer and deeper RNN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5457–5466.

Timothy P. Lillicrap, Adam Santoro, Luke Marris, Colin J. Akerman, and Geoffrey Hinton. 2020. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21:335–346.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Niru Maheswaranathan and David Sussillo. 2020. How recurrent networks implement contextual processing in sentiment analysis. *arXiv preprint*, arXiv:2004.08013.

Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. 2013. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2020. Does syntax need to grow on trees? Sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140.

Junier B. Oliva, Barnabás Póczos, and Jeff Schneider. 2017. The statistical recurrent unit. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2671–2680, International Convention Centre, Sydney, Australia. PMLR.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron C. Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019*. OpenReview.net.

H Francis Song, Guangyu R Yang, and Xiao-Jing Wang. 2016. Training excitatory-inhibitory recurrent neural networks for cognitive tasks: a simple and flexible framework. *PLoS Computational Biology*, 12(2):e1004792.

Piergiorgio Strata and Robin Harvey. 1999. Dale's principle. *Brain Research Bulletin*, 50(5-6):349–350.

David Sussillo and Omri Barak. 2013. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Computation*, 25(3):626–649.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Pascal Wallisch, Michael Lusignan, Marc Benayoun, Tanya I. Baker, Adam S. Dickey, and Nicholas G. Hatsopoulos. 2009. Synaptic transmission. In *Matlab for Neuroscientists*, pages 299–306. Elsevier.

Xiao-Jing Wang. 2002. Probabilistic decision making by slow reverberation in cortical circuits. *Neuron*, 36(5):955–968.

Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. 2019. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22(2):297–306.

Boxuan Yue, Junwei Fu, and Jun Liang. 2018. Residual recurrent neural networks for learning sequential representations. *Information*, 9(3):56.

A Appendix

A.1 Effect of agreement attractors

In this section, we present the trends in the testing performance of the LSTM and the Decay RNN (DRNN) for the grammaticality judgment task. Figure 3 shows the performance of the models when we fix the number of intervening nouns and vary the count of attractors between the main subject and the corresponding verb. The decreasing performance of the models with the introduction of more attractors indicates that they cause the models to get more confused about the upcoming verb number.

A.2 Comparison between DRNN and SDRNN

In Section 6.3, we saw that in terms of testing accuracy for grammaticality judgment, the Slacked Decay RNN (SDRNN) outperformed the Decay RNN (DRNN). For a robust investigation of this behaviour, we tested our models on the generalization set and mentioned a subset of our results on grammaticality judgment in Table 3. Here we present a bar graph (Figure 4) depicting the model performance when tested on the generalization set for the grammaticality judgment task. A substantial difference in the performance of the SDRNN and the DRNN reinforces the possibility of the regularizing effects of Dale's principle.

A.3 Implementation of Dale's constraint

$$\forall w_{i,j} \in ReLU(\mathbf{W}), w_{i,j} \geq 0$$

$$\mathbf{ReLU(W)W}_{dale} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,n} \\ w_{2,1} & w_{2,2} & \dots & w_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n,1} & w_{n,2} & \dots & w_{n,n} \end{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & -1 \end{bmatrix} = \begin{bmatrix} + & + & \dots & - \\ + & + & \dots & - \\ \vdots & \vdots & \vdots & \\ + & + & \dots & - \end{bmatrix}$$

A.4 Training settings

For the number prediction task and the grammaticality judgment task the network is trained as a binary classifier. The network is single-layered, with ReLU activation and trained with embedding and hidden layer dimension being 50, and a batch size of 1. We have reported the average accuracies after 3 separate runs in Table 1. For targeted syntactic evaluation, we have trained a language model to predict the grammaticality of a sentence. In our language model, we used a 2-layered network with $tanh$ activation, a dropout rate of 0.2 with embedding dimension 200, hidden dimension 650, and a batch size of 128. All models are trained with a learning rate of 0.001 using the Adam optimizer (Kingma and Ba, 2015).

A.5 Decay parameter (α) learning

In the main text, we describe the balancing effect of α in the Decay RNN model. We present the trend in the learned value of α throughout training for the grammaticality task for various initializations in Figure 5. We observe that for all α initializations in the range (0,1), the learned value converges to around 0.8. Hence, we initialize our α to 0.8 at the start of the training process.

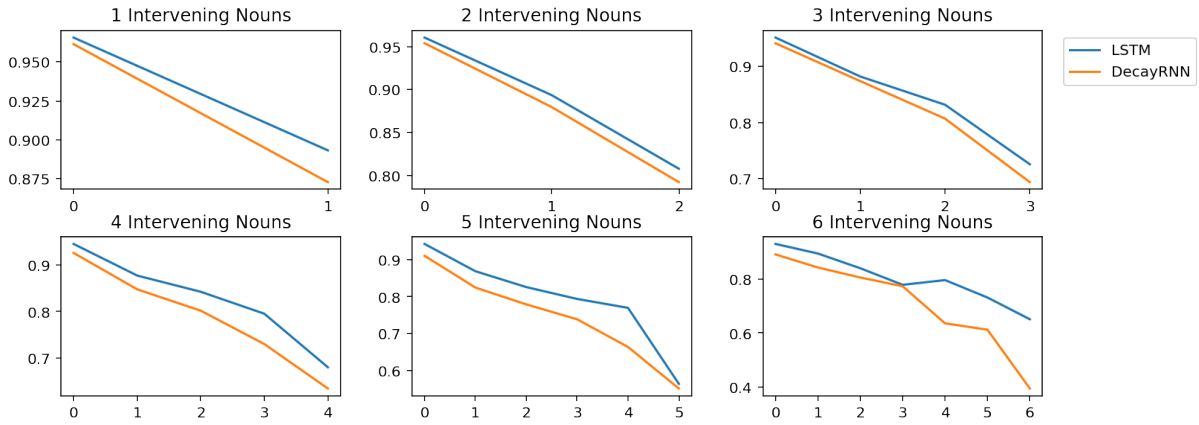

Figure 3: Trends in the performance of the LSTM (blue) and DRNN (orange) models with increasing numbers of intervening nouns. For each subplot corresponding to a fixed intervening noun number, the number of agreement attractors increases as we move from left to right on the x-axis.

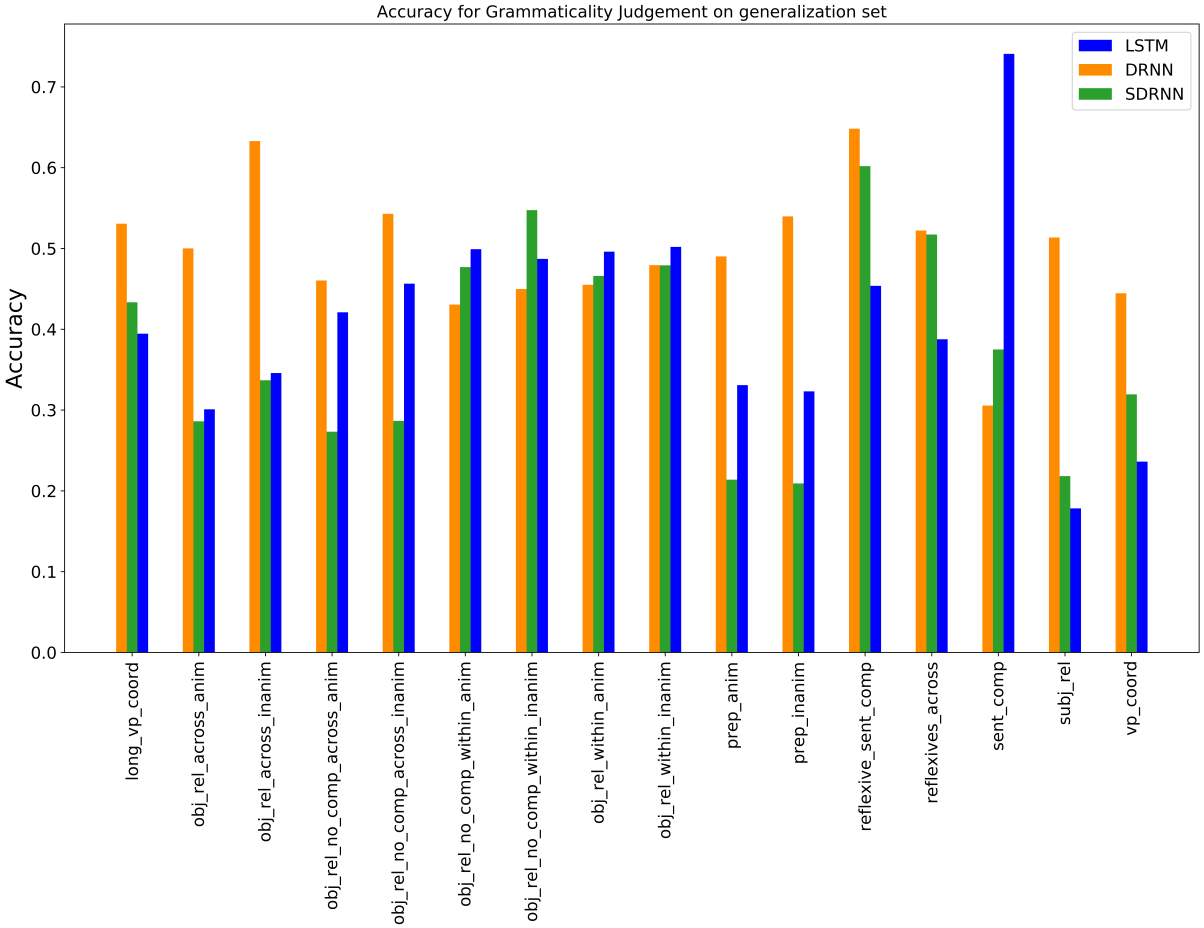

Figure 4: Performance of the LSTM (blue), DRNN (orange), and SDRNN (green) models for the different types of sentences in the generalization set, when trained for the grammaticality judgment task. There were at least 200 test sentences for each of these types.

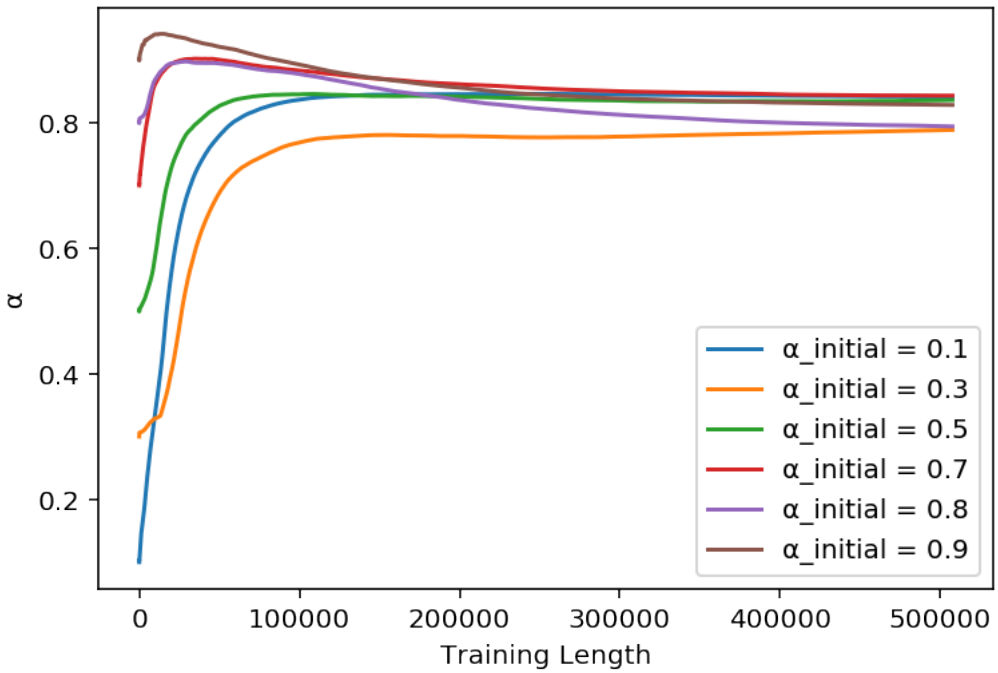

Figure 5: Moving average of α over the course of training for different initializations. 1 unit of training length is 1 forward pass.