# RNNs as Cognitive Models for Learning Syntax-Sensitive Dependencies

**Author 1 (Author 1 Email)**
Author 1 Affiliation

**Author 2 (Author 2 Email)**
Author 2 Affiliation

## Abstract

The goal of our research is to create cognitive and biologically plausible recurrent neural networks for learning syntax-sensitive dependencies. Long short term memory models (LSTMs) are able to capture long-range statistical regularities, but it is questionable how plausible they are as models of brain architecture. We begin addressing this problem by using biologically plausible neural networks like the Excitatory-Inhibitory Recurrent Neural Network (EIRNN) (Song, Yang & Wang, 2016) to model number agreement learning tasks and comparing it to LSTMs. We probe the competence of multiple recurrent neural network architectures over a variety of closely related tasks, which show that vanilla RNNs are worse at remembering/finding the locus of grammatical errors than capturing long range dependencies when compared to LSTMs. We further find relations between performance on similar tasks that indicate expectation vs. locality effects of providing more information. We also study ablation effects on LSTMs which show that the input and forget gates are the most important for LSTMs to learn grammaticality. Finally, we propose a new model termed a 'Decay RNN' that may be more biologically plausible and appears to capture grammaticality as well as LSTMs without using gates.

**Keywords:** RNN; LSTM; Recurrent Neural Networks; Syntax Sensitive Dependencies; Biologically Plausible Models

## Introduction

Early work on neural networks in the 1940s–60s mainly described biologically inspired learning: Hebbian Learning (Hebb, 1949) and the Perceptron (Rosenblatt, 2958) being the most notable. The second wave of Connectionism came in the 1960s–80s, with the invention of backpropagation, which is still the backbone for training most artificial neural networks (ANNs) today. But for the last 4–5 decades, neural networks have been approached primarily from an engineering perspective with the key motivation being efficiency, consequently moving further away from biological inspiration.

Recent developments in the past decade have used ANNs with explicit constraints to model specific parts of the brain and found correlation in activations in the ANN and actual neural activity recordings (Song et al., 2016; Gao, & Ganguli, 2015; Mante, Sussillo, Shenoy, & Newsome, 2013; Sussillo, 2014). In contrast, ANNs have very recently also been used to model human performance on specific tasks. Deep learning has revolutionized the field of Natural Language Processing (NLP) and LSTMs are the main building blocks of most state-of-the-art-deep learning models for NLP. Linzen, Dupoux and Goldberg (2016) show that LSTMs can capture complex dependencies in natural language and their performance on multiple aspects is close to that of humans.

In this paper, we explore recurrent neural networks as biologically plausible models for learning dependencies in natural language. We assess the ability and plausibility of known recurrent networks like the RNN, EIRNN, LSTM, and its ablated versions. We show that models other than LSTMs are ineffective at capturing grammaticality and propose a new recurrent network model termed a 'Decay RNN', which shows similar performance to LSTMs but may be more biologically plausible.

## Syntax-Sensitive Dependencies

The form of an English third-person present tense verb depends on whether the head of the syntactic subject (in what follows, we'll refer to it as "the subject"), is plural or singular:

- The **pan is** on the stove.

The noun and verb need not always be adjacent :

- * The **pan** from the <u>cupboard</u> **are** on the stove.[1]

Given a syntactic parse of the sentence, identifying the head of the subject corresponding to the verb and using that information to determine the number of the verb is simple. Although models that are insensitive to structure can run into difficulties capturing this dependency, one of the major issues being that there's no limit of the complexity of the subject noun phrase, and any number of words can appear between the noun and the verb. Thus fixed n-gram models

[1] The subject and the corresponding verb are marked in bold and asterisks mark unacceptable sentences. Agreement attractors (intervening nouns with the opposite number from the subject head) are underlined and intervening nouns with the same number as the subject is italicized.

cannot capture this dependency while recurrent networks with non-finite length constraint have the ability to do so.

The potential presence of agreement attractor entails that the model must identify the head of the syntactic subject that corresponds to a given verb in order to choose the correct inflected form of that verb. We initially test all recurrent models on the following tasks designed by Linzen et al. (2016).

**Number Prediction Task :** The basic task to understand the extent to which sequence models can learn to be sensitive to the hierarchical structure of natural structure is Number Prediction. In this task, the model sees the sentence up to but not including a present-tense verb, eg :

- The pan from the cupboard _______

The model then needs to predict the number of the following verb as Singular or Plural.

**Inflection Task :** The only difference between this task and number prediction is that the network receives the singular form of the upcoming verb apart from the words leading up to the verb. Having access to the semantics of the verb can help the network identify the noun that serves as its subject without using the syntactic subjecthood criteria.

**Grammaticality :** The previous objectives explicitly indicate the location in the sentence in which a verb can appear, giving the network a cue to syntactic clause boundaries. They also explicitly direct the network's attention to the number of the verb. The grammaticality judgement objective thus uses a form of weaker supervision. In this scenario, the network is given a complete sentence, and is asked to judge whether or not it is grammatical. The corpus is created by randomly flipping the number of the verb of half of the samples.

- * The **keys** to the cabinet **is** on the table.

## Methods

The words are encoded as one-hot vectors which are embedded in a 50-dimensional vector space. The recurrent networks read those words in a sequence; the final state of the network is fed into a logistic regression classifier for which we have used a fully connected single layer neural network with two outputs[2]. The model is then trained in an end-to-end fashion, including the word embeddings. Infrequent words were replaced with POS tags.

### Dale's Principle

For biological plausibility, we consider the constraints and principles provided in the EIRNN paper, the most important of them being Dale's Principle, which states that the neurons in the mammalian cortex have purely excitatory or inhibitory effects on other neurons. The analogous constraint that all connection weights from a given unit must have the same sign can have a profound effect on the types of dynamics, such as non-normality, that operate in the circuit. Connections from excitatory and inhibitory neurons also exhibit different levels of sparseness and specificity, with non-random features in the distribution of connection patterns among neurons. Notably, long-range projections between areas are primarily excitatory. Therefore, such details must be included in a satisfactory model of local and large-scale cortical computation.

**EIRNN :** The architecture of EIRNN is the same as a vanilla RNN, except that the input is the same across all recurrent units, i.e. the model can essentially be represented as a deep neural network with weights shared across all layers having the same number of units and the number of layers equal to the number of recurrences. The recurrence can be represented by :

$$h_t = [\alpha \,.\, \{Dale(w_{rec})\, h_{t-1} + ReLU(w_{in})\, x\} \,] + [\, (1-\alpha) \,.\, (h_{t-1}) \,]$$

$$o_t = ReLU(w_{out}) \,.\, h_t$$

EIRNN also has explicit constraints on the architecture enforcing Dale's principle and others as described above using :

$$Dale(a) = ReLU(a) \,.\, Diag(80\%)$$

where Diag($\beta$) is a diagonal matrix of the same size as '$a$' with the first $\beta$ entries 1 and the rest -1. We have used $\beta$ = 80% as used in the original paper (Song et al., 2016).

**RNN Dale[3] :** We add the same biological constraints in EIRNN to the vanilla RNN with no additional architectural changes :

$$h_t = tanh(w_{ih}\, x_t + b_{ih} + Dale(w_{hh})\, h_{t-1} + b_{hh})$$

**Ablated LSTM :** To observe the effect and importance of different gates in LSTM, we ablate different gates one by one. We'll refer to Ablated LSTM or AbLSTM as the one with no input and forget (*i and f*) gates as we observe significant differences on removing these gates. One can also observe that these gates are incorporated in the architecture using Hadamard products, the presence of which is not possible in a biologically plausible model.

$$g_t = tanh(w_{ig}\, x_t + b_{ig} + w_{hg}\, h_{t-1} + b_{hg})$$

$$o_t = sigm(w_{io}\, x_t + b_{io} + w_{ho}\, h_{t-1} + b_{ho})$$

$$c_t = f \,.\, c_t + i \,.\, g_t$$

$$h_t = o_t \,.\, tanh(c_t)$$

where $i$ and $f$ are learnable vectors independent of $h$ and $x$.

[2] Each giving the confidence of the prediction being 0 and 1 respectively. Cross-entropy loss is used to train the model.

[3] All results on vanilla RNN were similar to RNN Dale with RNN Dale outperforming sometimes by ~ 0.1-0.2%.

## Results

All models except EIRNN perform similarly on the Number Prediction (NPred), Inflection (Infl) and Grammaticality (Gram) tasks as shown in Table 1.

We observe that the EIRNN still has a comparatively high performance on these tasks despite the fact that it does not process the sentence explicitly sequentially. On analysis, we observe that the dataset is skewed, having no intervening nouns (80.8%), which makes the task easier given the syntax sensitivity does not play a major role here if the model can interpret POS tags well from the sentence.

Table 1: Accuracies on the number prediction, inflection, and grammaticality tasks (Accuracy in %).

| Model (Hidden Units) | NPred | Infl | Gram |
|---|---|---|---|
| EIRNN (3) | 92.8 | 92.5 | ~50 |
| EIRNN (15) | 94.1 | 94.2 | ~50 |
| RNN Dale (50) | 97.8 | 98.0 | ~50 |
| AbLSTM (50) | 98.0 | 98.1 | ~50 |
| LSTM (50) | 98.7 | 98.9 | 95.5 |

The results on the Grammaticality task are the most surprising. No other model except the LSTM (RNN, EIRNN, or AbLSTM) is able to model grammaticality, and their accuracy is just 50% (same as random choice).

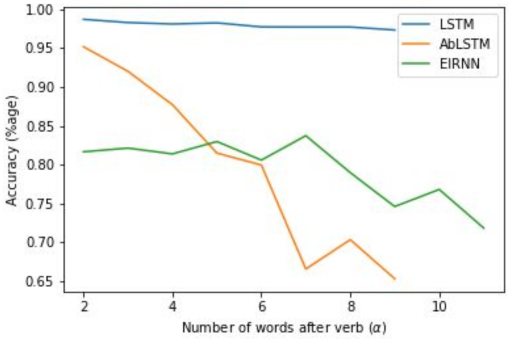

Figure 1: LSTM, AbLSTM, and EIRNN on Plus-*n*.

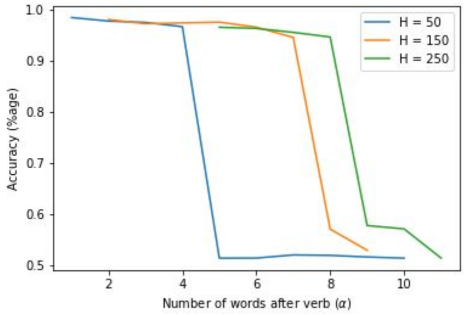

Figure 2: RNN Dale (H = hidden units) on Plus-*n*.

**Plus Grammaticality :** To understand if the bad performance is due to the inherent difficulty of the task or because of absence of a cue to the position of the error, we designed the 'Plus Grammaticality'[4] task, in which we provide the input sentence with at most *n*[5] words following the verb in consideration in that sample. Performance of various models on can be seen from Figure 1, 2.

### Gradient Norm Analysis

To analyze the performance of models other than LSTMs on Grammaticality and Plus-*n* tasks, we plotted the gradient norm for model parameters during training (Figure 3).

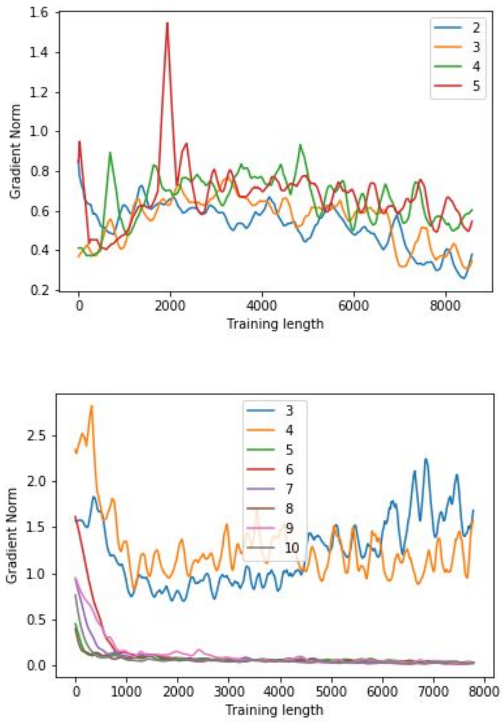

Figure 3: Gradient Norm plots for recurrent matrix ($W_{rec}$) LSTM (Top) and RNN (H=50) (Bottom).

### Decaying RNN

Taking inspiration from the decaying nature of the voltage of a neuron membrane after receiving activation from preceding neurons (Gluss (1967); Bugmann (1997)), we add a learnable constant decay factor to the current state to compute the updated next state :

$$h_t = tanh( (\alpha . h_{t-1}) + (1-\alpha) (w_{ih} x_t + b_{ih} + w_{hh} h_{t-1} + b_{hh}))$$

To the best of our knowledge, we are the first to create such a recurrent neural network architecture that performs at par

[4] The task is mentioned in the rest of the paper in the abbreviation as 'Plus-*n* Gram' or 'Plus-*n*'.

[5] It is also referred to as $\alpha$ in figures. It is not the same as the $\alpha$ used in recurrences ($n \geq 0$).

with LSTMs and GRUs on these tasks without gating or using Hadamard products. A summary of the performance of the Decay RNN is provided in Table 2. The performance of the Decay RNN being comparable to LSTMs is suggestive of the fact that the particular notion of 'memory' encoded in the latter is just a modelling feature and may not be essential for language processing in the brain.

Table 2: Decay RNN Performance.

| Task | Accuracy (%) |
|---|---|
| Number Prediction | 98.1 |
| Inflection | 98.2 |
| Grammaticality | 94.8 |
| Plus-10 Gram | 96.9 |

## Discussion and Future Work

Neural network architectures have reached high accuracies in tasks based on natural language, LSTMs being the core of most state-of-the-art models. A lot of previous research has debated the cognitive plausibility of these models and tried creating models that are close in learning and performance to humans. We aimed to explore the architectural plausibility of ANN models in relation to the brain which has mostly neglected so far.

The poor performance of all models except the LSTM and the Decay RNN on the Grammaticality task was both surprising and cognitively interesting. Although the gradient norm plots point towards the vanishing gradient problem, it is not clear that it is the only factor. The sudden drop in accuracy of the RNN (from over 90% to random) on adding a single word to the training and test data in the Plus-*n* tasks cannot be explained using just vanishing gradients.

We observed a low correlation in errors made by a model on tasks that differ by the number of maximum words that follow the locus of the grammatical mistake (Plus-*n* tasks), e.g., the LSTM model predicted 40% of the samples correctly in Plus-*4* that it was wrong in predicting in Plus-*3*. This is indicative of the fact that extra information can help in finding the noun-verb pairs in the sentence by inherently helping the model create a better representation of the sentence. This may relate to expectation and locality effects studied in psycholinguistics to understand when and if more information hurts or helps (Husain, Vasishth & Srinivasan, 2014). Further linguistic analysis on the corpus and the error sets of the models would provide more concrete evidence.

Decay RNN's performance can be attributed to solving the vanishing gradient problem but we intend to explore other possibilities. As a connection to biology, the decay term in the recurrence can be seen as analogous to voltage decay of membrane potential which brings the model closer to structural plausibility. Exploring more biologically plausible training methods like the recently proposed HSIC bottleneck method (Ma, Lewis & Kleijn, 2019) would be an interesting direction for future work.

The bigger picture question is whether syntactic parsing is necessary for learning number agreement and other more complicated linguistic tasks or are sequences enough. Moreover, are syntax trees just for the study of natural language? This question for modelling these tasks using neural networks when we talk of cognitive plausibility translates into: by modelling language using sequential models like LSTMs/RNNs, are we showing that explicit syntactic information is unnecessary? Or does the neural network implicitly end up creating hidden representations like the syntax trees while learning such tasks? Are we modelling the brain, or some aspect of the way the brain processes linguistic information, using these recurrent neural networks? This of course remains an open question, but we hope our findings here can provide some starting points for further investigation.

## References

Song, H. F., Yang, G. R., & Wang, X-J. (2016). Training excitatory-inhibitory recurrent neural networks for cognitive tasks : A simple and flexible framework. *PLoS Comput Biol* (12(2): e1004792).

Hebb, D. (1949). The Organization of Behaviour. *New York: John Wiley & Sons, Inc*.

Rosenblatt, F. (1958). The Perceptron : A probabilisitc model for information organization in the brain. *Psychological Review 65.6*.

Gao, P., & Ganguli, S. (2015). On simplicity and complexity in the brave new world of large-scale neuroscience. *Curr Opin Neurobiol*;32:148–155

Mante, V., Sussillo, D., Shenoy, K. V., & Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*;503:78-84

Sussillo, D. (2014). Neural circuits as computational dynamical systems. *Curr Opin Neurobiol*;25:156-163

Linzen, T., Dupoux, E., & Goldberg, Y. (2016). Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics* (vol. 4, 521-535).

Husain, S., Vasishth, S., & Srinivasan, N. (2014). Strong expectations cancel locality effects : evidence from Hindi. *PLoS ONE* (9(7): e100986).

Gluss, B. (1967). A model for neuron firing with exponential decay of potential resulting in diffusion equations for probability density. *Bulletin of Mathematical Biophysics* (Vol 29, Issue 2, 233-243).

Bugmann, G. (1997). Biologically plausible neural computation. *Biosystems* (Vol 40, Issue 1-2, 11-19).

Ma, W-D. K., Lewis, J. P., & Kleijn, W. B. (2019). The HSIC bottleneck: Deep learning without back-propagation. *arXiv preprint* (arXiv:1908.01580v1).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation* (9(8): 1735– 1780).