EXAMINING COGNITIVELY PLAUSIBLE RNN MODELS FOR LINGUISTIC TASKS

Hritik Bansal, Gantavya Bhatt, Sumeet Agarwal [*]
Department of Electrical Engineering
IIT Delhi
{ee1160071,ee1160694,sumeet}@ee.iitd.ac.in

ABSTRACT

In recent times, there have been efforts to access the ability of recurrent structures to capture the syntax-level dependencies in language processing. LSTM and its variants, a group of highly sophisticated models with little biological grounding, have shown the ability to encapsulate long-distance syntactic distance through little(Grammaticality) to no supervision(Language Modeling). However, simple Recurrent Neural Network's(RNNs) connection are supposed to be analogous to biological synapses. We infer that there is a need to develop a model which better encodes the neurological processes, and evaluate it against existing recurrent schemes. In this paper, we propose a new architecture - Decay RNN, which has a prior induction of neurological phenomenon - Dale's principle as well as decay in Post Synaptic potentials. We tested our architecture on various language processing tasks and compared our performance with other existing models. As an extension to work, we have organized the models along the spectrum of language tasks(restricted to general) to get some insight into the effect of various design choices.

1 INTRODUCTION

The initial development of the Neural Networks can be dated back to the late 1940s and early 1950s when the first biologically inspired algorithm was proposed as the Hebbian Learning Rule. At the same time, another model of the simple linear classifier was created called the perceptron first introduced in 1958 by Rosenblatt (1958). The earlier development of these neural networks were the models to understand the esoteric design of the human brain. In the 1980s, the idea of using a multilayer perceptron also known as the deep neural network as an approximator was mathematically proven by Cybenko (1989) by proposing a theorem called the universal approximation theorem, claiming about the power of neural networks to approximate any function to any degree of extent, given that the complete setting satisfies some conditions. However, to accomplish this task, one will need enormous computation power, which limited the training of these neural nets at that time. In the gradual years and with the advent of GPUs and parallelizing the training over multi-core architecture came into the picture, in due course, more research was done to improve the performance of these neural nets. The majority of the developed algorithms were profound mathematical equations rather than having any direct motivation from the biological processes. Thus there is a gap between the nonscientific neural network models, which needs to be bridged.

We can understand the human cognitions from many aspects, be it a cellular level modeling, psycholinguistics, etc.. Linguistic ideas and brains have a close correlation with each other. Modeling of the natural language is always an interesting problem for the linguists. In many animals, there set of sounds can be modeled as the output of a Finite State machine. However, human language is not as simple as a Markov chain. It does have a linear structure, but the inter-dependencies between the lexicon are not straightforward. A tree-like structure is the most current language structure. Construction of this tree is governed by a set of principles given by Noam Chomsky. Thus, it might be possible that to comprehend the language human brain inherently correlate the underlying representation of language according to these set of rules. Many linguistic phenomenons were studied

[*] Supervisor

to explain the studies done on children learning their first language have shown that children make common mistakes while learning their first language. This phenomenon of child language accusation and errors during language comprehension was also discussed concisely in Linzen (2019).
To understand these phenomena, we need to have a good model of the brain. However, this brings us back to the question of how good are we in modeling the human mind? As mentioned earlier about the problem with the current state of the art such as LSTMs, GRUs, BERT, etc. they were developed in the regime of improving the mathematical models and lacked the cognitive explainability. The models like LSTMs and GRUs, have the presence of gates which are dependent on input time stamps. However, they require the phenomena of Excitatory and Inhibitory postsynaptic potential decay(EPSP/IPSP). Therefore, we decided to come up with a new architecture that we call the ”Decay RNN.” Our model incorporates the presence of EPSP and IPSP decay with an exponential moving average with a learnable exponent base at the same time being computationally inexpensive. The following sections will talk about the related work related to Cognition and Languages, followed by the formulation of our model besides the existing SOTA. Then we present our experimentation on the tests as described by Linzen et al. (2016) and Marvin & Linzen (2018) followed by a detailed analysis of our model. Towards the end, we present a spectrum of tasks against the difficulty of the tasks.

2 Related Work

There has been prior work on using RNNs and LSTMs for language modeling tasks. It was shown that recurrent networks could model the context-free grammars (Gers & Schmidhuber, 2001). However, as per the investigations carried out in Kuncoro et al. (2018), it was shown that if the model capacity is not enough, then LSTMs cannot generalize well on long term dependencies. Kuncoro et al. (2018) proposes a model that uses the information of phrase structure trees in their architecture to improve the generalization of long term dependencies. However, this process, as compared to child language acquisition is not unsupervised. Child language acquisition is believed to be unsupervised as no information on syntax structure is provided. Thus, method proposed by Kuncoro et al. (2018) is cognitively not plausible. Shen et al. (2019) introduces ON-LSTMs where additional master input & forget gates along with cumax activation function to induce tree bias and learn the inherent tree structure in an unsupervised fashion. However, this model is way more complex than existing LSTM architecture, and their cognitive interpretation cannot be drawn with the given set of phenomena.

The idea of modeling a bunch of neurons as a linear dynamical system and non-linear Kalman filter was exploited in EIRNN Song et al. (2016). Even though their model is one of the most cognitively plausible architectures developed so far, with the given settings of their model, when we performed tests on their model, it failed badly. More analysis of their model is present in Section 5.

Linzen et al. (2016) and Marvin & Linzen (2018) provide a framework that generates set of targeted experiments for the evaluation of representation power of recurrent architecture. However, in work presented by the author, most of their work was designated to LSTMs and very few towards RNN (or other architectures). Since their framework generates a robust set of experiments, hence we will use the same tasks for our experiments. These tasks are further discussed in Section 5.

Our work can be considered as a refinement to the existing simple RNN. We had incorporated the neurological phenomena of repolarization of the action potential produced by the excitatory postsynaptic potential, which was also discussed in Bugmann (1991), Bugmann (1997). At the same time, our work incorporated Dale’s principle with the standard RNNs may also provide a cognitive perspective to the variational dropout mentioned in Gal & Ghahramani (2015)

3 Recurrent Models

The following are the governing equations of the models which we have considered in our present work. Bold symbols indicate they are vector or matrix quantities. All non-bold quantities are strictly scalar.

3.1 Simple RNN

$$\begin{aligned}
\boldsymbol{a}^{(t)} &= \boldsymbol{b}+\boldsymbol{W}\boldsymbol{h}^{(t-1)}+\boldsymbol{U}\boldsymbol{x}^{(t)} \\
\boldsymbol{h}^{(t)} &= \tanh\left(\boldsymbol{a}^{(t)}\right) \\
\boldsymbol{o}^{(t)} &= \boldsymbol{c}+\boldsymbol{V}\boldsymbol{h}^{(t)} \\
\hat{\boldsymbol{y}}^{(t)} &= \operatorname{softmax}\left(\boldsymbol{o}^{(t)}\right)
\end{aligned}$$

$\mathbf{h}_{t-1} \rightarrow$ output of the previous lstm block (at timestamp $t-1$)
$\mathbf{x}_t \rightarrow$ input at current timestamp

3.2 Long Short Term Memory

We will present two LSTM architecture, prior one being the standard LSTM architecture (Hochreiter & Schmidhuber (1997)), while the later one we call as ablated LSTM having subtle differences from standard LSTM described in the following subsections.

3.2.1 Standard LSTM

$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{w}_i\left[\mathbf{h}_{t-1}, \mathbf{x}_t\right]+\mathbf{b}_i\right) \\
\mathbf{f}_t &= \sigma\left(\mathbf{w}_f\left[\mathbf{h}_{t-1}, \mathbf{x}_t\right]+\mathbf{b}_f\right) \\
\mathbf{o}_t &= \sigma\left(\mathbf{w}_o\left[\mathbf{h}_{t-1}, \mathbf{x}_t\right]+\mathbf{b}_o\right) \\
\tilde{\mathbf{c}}_t &= \tanh\left(\mathbf{w}_c\left[\mathbf{h}_{t-1}, \mathbf{x}_t\right]+\mathbf{b}_c\right) \\
\mathbf{c}_t &= \mathbf{f}_t * \mathbf{c}_{t-1}+\mathbf{i}_t * \tilde{\mathbf{c}}_t \\
\mathbf{h}_t &= \mathbf{o}_t * \tanh\left(\mathbf{c}_t\right)
\end{aligned}$$

$\mathbf{i}_t \rightarrow$ input gate.
$\mathbf{f}_t \rightarrow$ forget gate.
$\mathbf{o}_t \rightarrow$ output gate.
$\mathbf{h}_{t-1} \rightarrow$ output of the previous lstm block (at timestamp $t-1$)
$\mathbf{x}_t \rightarrow$ input at current timestamp
$\mathbf{c}_t \rightarrow$ cell state(memory) at timestamp(t)

* means element wise product.

3.2.2 Ablated LSTMs

We define a sub-model called ablated LSTMs. In LSTMs, gate values are direct functions of inputs at each time stamps and the corresponding weights of affine transformation are learned. However, to investigate the direct influence of input values on gate values, we made the values of the gate independent of the inputs and learnable along with the other weights of the network. Since our gates don't depend on inputs, they are the same across all the timestamps. Rest equations remain the same.

$$\begin{aligned}
\mathbf{o}_t &= \sigma\left(\mathbf{w}_o\left[\mathbf{h}_{t-1}, \mathbf{x}_t\right]+\mathbf{b}_o\right) \\
\tilde{\mathbf{c}}_t &= \tanh\left(\mathbf{w}_c\left[\mathbf{h}_{t-1}, \mathbf{x}_t\right]+\mathbf{b}_c\right) \\
\mathbf{c}_t &= \mathbf{f} * \mathbf{c}_{t-1}+\mathbf{i} * \tilde{\mathbf{c}}_t \\
\mathbf{h}_t &= \mathbf{o}_t * \tanh\left(\mathbf{c}_t\right)
\end{aligned}$$

3.3 GRU

Following are the equations proposed in Cho et al. (2014)

$$\begin{aligned}
\mathbf{z}_t &= \sigma_g\left(\mathbf{W}_z \mathbf{x}_t+\mathbf{U}_z \mathbf{h}_{t-1}+\mathbf{b}_z\right) \\
\mathbf{r}_t &= \sigma_q\left(\mathbf{W}_r \mathbf{x}_t+\mathbf{U}_r \mathbf{h}_{t-1}+\mathbf{b}_r\right) \\
\mathbf{h}_t &= \left(1-\mathbf{z}_t\right) * \mathbf{h}_{t-1}+\mathbf{z}_t * \sigma_h\left(\mathbf{W}_h \mathbf{x}_t+\mathbf{U}_h\left(\mathbf{r}_t * \mathbf{h}_{t-1}\right)+\mathbf{b}_h\right)
\end{aligned}$$

$\mathbf{r}_t \rightarrow$ reset gate.
$\mathbf{z}_t \rightarrow$ update gate.

3.4 EIRNN

EIRNN (Song et al. (2016))is a discretised version of a linear dynamical system modelled by the following equations:-

$$\tau\dot{\mathbf{x}} = -\mathbf{x} + W^{\mathrm{rec}}\mathbf{r} + W^{\mathrm{in}}\mathbf{u} + \sqrt{2\tau\sigma^2_{\mathrm{rec}}}\xi$$

Where τ is the time constant of the dynamics, ξ being the system noise. This continuous dynamics will be discretized to the Euler form, where Δt is the sampling time interval. Notice that α is a fixed scalar parameter.

$$\begin{aligned}
\mathbf{x}_t &= (1-\beta)\mathbf{x}_{t-1} + \beta\left(\mathbf{W}^{\mathrm{rec}}\mathbf{r}_{t-1} + \mathbf{W}^{\mathrm{in}}\mathbf{u}_t\right) + \sqrt{2\beta\sigma^2_{\mathrm{rec}}}\mathbf{N}(0,1) \\
\mathbf{r}_t &= [\mathbf{x}_t]_+ \\
\mathbf{z}_t &= \mathbf{W}^{\mathrm{out}}\mathbf{r}_t \\
\beta &= \Delta t/\tau
\end{aligned}$$

4 DECAY RNN

In this section, we present our architecture called "Decay RNN" (DRNN). The equation of a Decay RNN is similar to the existing recurrent models, however, there are subtle differences. We define the following update rule:

$$\mathbf{h}^{(t)} = \tanh[\alpha * \mathbf{h}^{(t-1)} + (1-\alpha) * ((\boldsymbol{ReLU(W).W_{dale}})\boldsymbol{h}^{(t-1)} + \boldsymbol{Ux}^{(t)})]$$

$$[W_{dale}]_{ij} = \begin{cases} m|m \sim Ber(p), m\epsilon\{1,-1\} & i = j \\ 0 & i \neq j \end{cases}$$

We define α as our decay parameter, incorporating the repolarization of a neuron when it attains its full peak. α is a learnable scalar parameter. An action potential occurs as long as the input can produce net excitatory postsynaptic potentials. However, if the input is removed, then neuron re-polarizes leading to a refractory period following to resting state. Our decay parameter plays the same role in modeling the re-polarization of a neuron. It can be seen that the quantity within the parenthesis of tanh is similar to an exponential moving average. Note that, in contrast to Song et al. (2016), hidden state of our model is guaranteed to be bounded.

Let m be the random variable coming from a Bernoulli distribution with output 1 with probability p and -1 with 1-p. If we assume that p is set such that the first 10% diagonal entries of W_{dale} happens to be -1. Then,

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

To examine the importance of Dale's principle in the learning process, we made a variant of our Decay RNN without Dale's principle. We call it as the *Slacked Decay RNN (SDRNN)*.

5 EXPERIMENTS

We tested various models on tasks that evaluate their ability to capture syntax-sensitive dependencies, as done in Linzen et al. (2016). We assess the performance of Decay RNN with different initializations of our learnable parameter α. Experiments were performed to enforce the dependence of α on the inputs and compared against our simple Decay RNN.

5.1 Verb Number Prediction Task

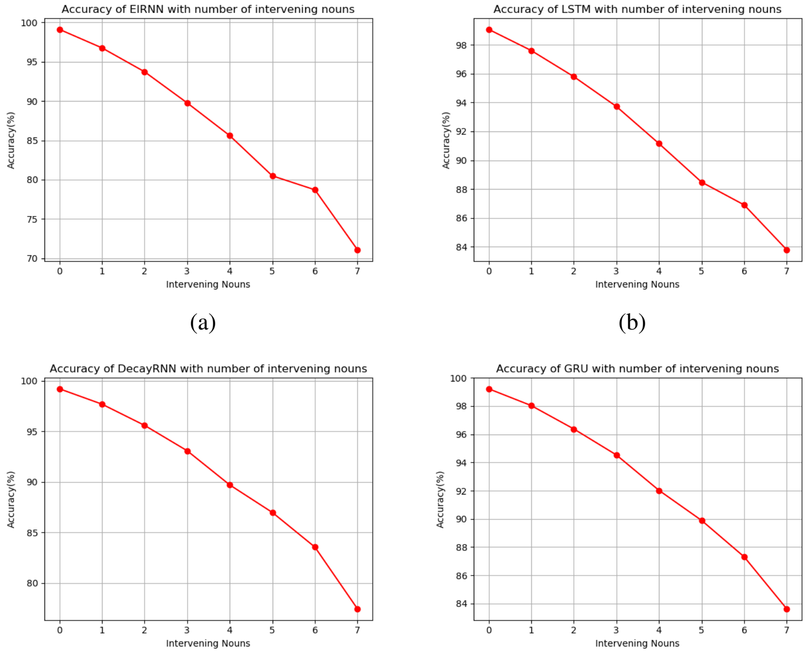
This task was used by Linzen et al. (2016), to predict the number(inflection) of the present verb, given the model has access to the sentence up to that verb. This task is supposed to test the ability of the model to encode the relationship of the verb in question with its syntactic subject and its number.

- The **key** *is* on the table
- The **key** *are* on the table. (*)
- The *keys* to the cabinet .... (training sentence)

The model settings were the same as proposed by Linzen et al. (2016), trained on Wikipedia based 1.35 million examples. We train on 10%, cross validate on 0.5%, and test on the remaining dataset. Example sentence (as per Linzen et al. (2016))
In the above example, the target label will be 1, which corresponds to plural inflection. In this way, the model will be trained.
Table 1 shows the performance of the selected recurrent models. It can be seen that most of the models generalize well on this task. GRUs performed best on this task with the maximum accuracy. Further, one vs one analysis of each model is present in the appendix.

Recurrent Model	**Accuracy(%)**
LSTM	98.59
GRU	**98.81**
EIRNN($\beta = 0.001$)	98.31
Decay RNN	98.66

Table 1: Performance of the language models on Number Prediction task



(a)

(b)

(c)

(d)

Figure 1: (a),(b),(c),(d) : Accuracy scores of EIRNN, LSTM, DecayRNN, and GRU respectively, with increasing number of intervening nouns

We test the models to evaluate the effect of intervening nouns as done in Linzen et al. (2016). Figure 1 indicates the accuracy of LSTM,GRU,EIRNN and DecayRNN with increasing number of noun intervening nouns. The distribution of the number of intervening nouns in the dataset is shown in Figure 6(a).

5.2 Further Targeted Evaluation

5.2.1 Grammaticality of a sentence

This section deals with the grammaticallity evaluation of a sentence. It is weak supervision unlike previous objectives, it doesn't give any syntactic cues to the models. Consider following example :

- The **roses** in the vase by the door *are* red.
- The **roses** in the vase by the door *is* red. (*)

The first sentence is grammatically correct while the other one is incorrect. Given an input sentence, our model predicts whether the output is correct or not.

Recurrent Model	**Accuracy**
LSTM	95.81%
GRU	94.26%
Decay RNN	95.48%
Slacked Decay RNN	**96.83**%
EIRNN($\beta = 0.17$)	91.59%
EIRNN(β = 0.001)	≈50%
EIRNN(FCN)	≈50%
Ablated LSTM	≈50%

Table 2: Performance on Grammaticality judgement; Dataset: Linzen et al. (2016)

5.2.2 Language Modeling

We trained a language model with our architecture and compared it with LSTM based language models. The results were showing that LSTMs were able to learn a language model with less perplexity however our perplexity was nearly 8 fold to that of LSTM's. The LSTM language model was a 2 layered network and was trained for 4 epochs with a batch size of 128, a dropout rate of 0.2 and a learning rate of 20, as mentioned in Shen et al. (2019). To train Decay RNN we set the number of layers to be 2, learning rate to be 0.001, batch size 128 and num epochs to be 2(due to computational constraint).

Metric	**LSTM**	**DecayRNN**
Perplexity	130	928

Table 3: Performance of language models on Marvin & Linzen (2018) task

6 Detailed Analysis of Decay RNN

In this section we will present an analysis of Decay RNN. It is dependent upon various parameters in which most the important one is α. We will discuss about it in the subsequent parts.

6.1 α initialization and Constraints

In this subsection we will present an analysis of decay parameter α.The initial values of α heavily governed the model's performance and learning time. The result is plotted in Figure2. It was experimentally observed that only the initialization in range (-1, 1) led to the proper training of the network. Mathematically, it can be seen that when the value of α is greater than 1, or less than -1, then it is clear that the exponentially weighted average will get unbounded. Hence it will make difficult for the model to learn the parameters so that to keep the system bounded.

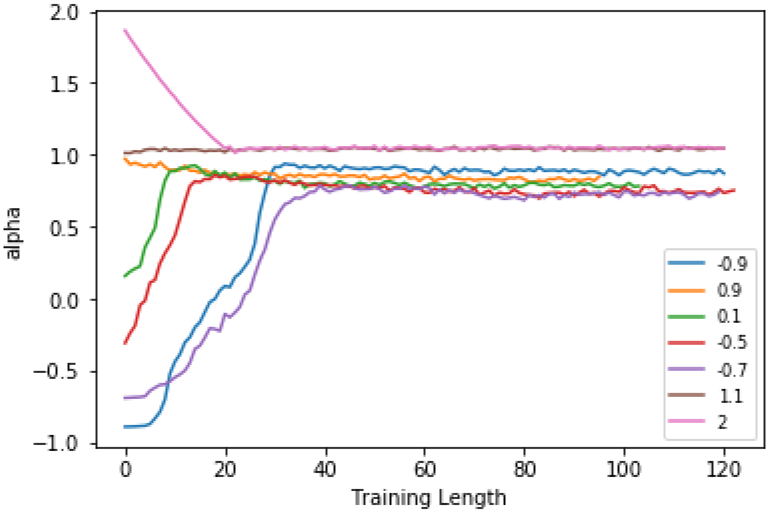


Figure 2: The trajectory of α over training length. Each unit represents 3000 examples. For α greater than 1, its value stagnates at 1; all other trajectories converge at a point close to 0.8. Interestingly all of the well-behaved values of α have an overshoot like an underdamped control system.

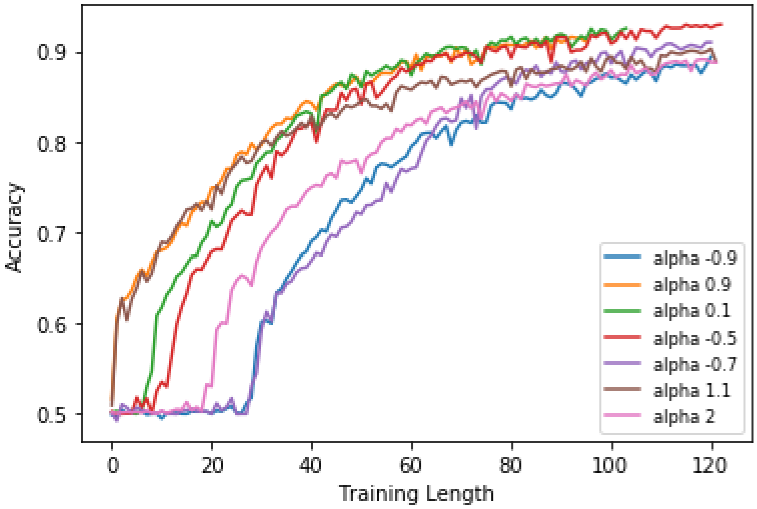


Figure 3: Variation of accuracy over training length. It can be seen that the values of alpha directly affects the test accuracy. Accuracy started to increase only after having attained α value close to 0.6.

Although by the previous experiment it was clear that α tries to be as close as to 0.8 which can be stated as the steady-state value of α, we tried constraining our values of α for better analysis. We imposed two constraints, one by constraining it to be positive, and the other by constraining it to be in between (0,1).
It was observed that in both of the settings, there was no effect on accuracy. The final value of the decay parameter was the same as was in the general case.

6.2 α AS FUNCTION OF INPUTS

The α values as of now are input independent. However, we also tried to make α learnable from the inputs taking the inspiration from Shen et al. (2018). The value of α came out to be the same as that of learned directly. This was because the CNNs weight was highly sparse and therefore the model was just learning a bias. $\alpha = 0.83$ as the final value with final Decay RNN accuracy as 96.62%.

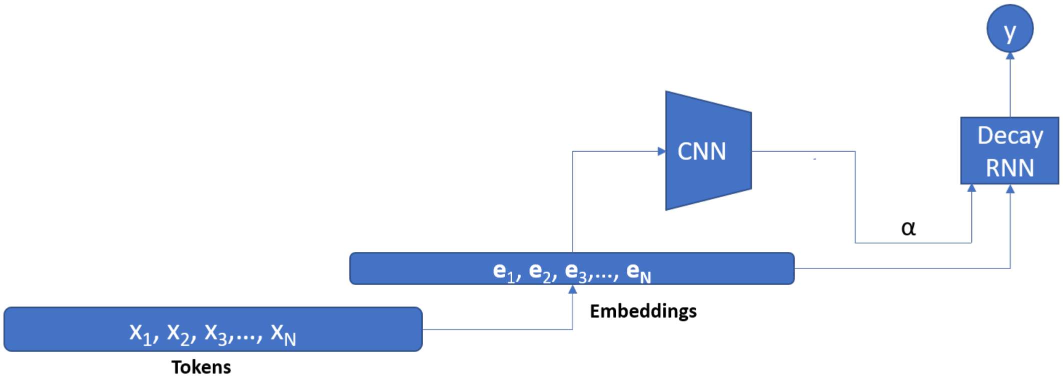


Figure 4: DecayRNN architecture when α is input dependent. Conventionally, Convolutional Neural Nets(CNN) have been widely used to capture local context information in sentences.

We also saw that this method creates a skip connection from the input embedding layer to Sequential network. Therefore, as an indirect implication, we have multiple paths of gradient flow and hence better training of the embedding layer weights.

7 Discussions

We tested our model on a set of Linguistic tasks in an increasing sense of difficulty. Although having similar structure to EiRNN, Song et al. (2016) performs experiments on very low level neurological taks such as Perceptual decision making and motor control, while we perform on high level cognition tasks. Our first test of verb number prediction being the easiest task which is just a binary classifier. On the other hand language modeling being the most difficult task, which is an extreme classification paradigm. Most of the models generalize well on the verb number prediction task, including the naive models such as fully connected EIRNN (Singh R. 2019). We also saw the trend across an an increasing number of intervening nouns which was also expected.

Our second set of task eliminated EIRNN and another set of models - EIRNN(FCN) and ablated LSTM (Singh R. 2019). EIRNN Song et al. (2016) had a very high retention capacity ($\beta = 0.001$) which had detrimental effects in learning the grammmatical cues from the input sentence. Fully connected models (CNN, etc) can be shown to be as good as n-gram models with n being their receptive field, and thus they don't perform to the mark. We saw that having a learnable decay in the hidden state helped to have just sufficient retention of state to learn the grammatical cues.

We saw that our model is susceptible to the initialization and only works when we have α initialized in (-1, 1). We can create a linear dynamical system analogy as presented in Song et al. (2016) for our Decay RNN and can say that having α greater than 1 causes the solutions to differential equations to unbound and hence gradients stagnated for those α models.

We also investigated the importance of Dale's principle and in our Slacked Decay RNN, we saw that having Dale's principle causes a decline toto accuracy. We can claim that Dale's principle encumbers some weight parameters to reach onto their optimal values. Even though we have norm penalties thatthat also encumber weights from their optimal value and have regularizing effects, Dale's way to impose the the penalty is suboptimal and hence is more malignant to model's accuracy.

Finally, we saw in language modeling task, our model collalpsed and LSTMs performed significantly better than Decay RNN. Language modeling being the most difficult task requires Semantic as well as Syntactic knowledge. It is an extreme classification process and is way more difficult than binary classification which was the crux of our first two tasks. and is way more difficult than binary classification which was the crux of our first two tasks.

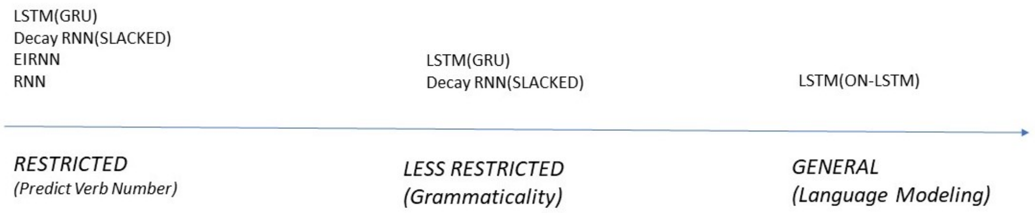


Figure 5: Performance Spectrum: Moving from left to right: the language tasks have decreased in their ability to provide syntactic cues to the recurrent structures. Models above each task indicate the architectures which have performed much better than random on those tasks.

8 Conclusion

In this thesis, we motivated the purpose of a recurrent model which fills the existing gap between simple recurrent schemes and LSTM on a high-level cognitive task-language processing. We have also designed a performance hich presents our findings in a concise form. The performance spectrum suggests that the presence of certain design choices affect the recurrent models' ability to perform on language tasks. The design choices in this case can range from having learnable parameters(α in DecayRNN), dependence on input timestamps(input,forget gates with cell states in LSTM) and nature of vector product(hadamard or scalar multiplication). There is no notion of hierarchical representations in the modern architecture of DecayRNN; however it performs well on less restricted linguistic tasks. Hence, it is essential to analyze the design choices between DecayRNN and other simple variants of RNN, which cause such differential change in the performance scores.

9 Future Work

As a continuation of our thesis, we would like to work on improving the Decay RNN accuracy on the Language Modeling task. We will also need to do the joint analysis of the count of attractors and intervening nouns. In terms of new experiments, we are planning to perform multitask learning with CCG supertags and affirmative question construction to improve the encoding power of the model.

10 Acknowledgement

We thank Supercomputing IIT Delhi for providing us NVIDIA V100 GPUs.We would also thank Rishubh Singh (CSE 2014 entry) for initial foundations of the similar work in his M.Tech thesis.

References

Guido Bugmann. Summation and multiplication: two distinct operation domains of leaky integrate-and-fire neurons. 1991.

Guido Bugmann. Biologically plausible neural computation. *Bio Systems*, 40 1-2:11–9, 1997.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL `http://arxiv.org/abs/1406.1078`.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989. ISSN 1435-568X. doi: 10.1007/BF02551274. URL `https://doi.org/10.1007/BF02551274`.

Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *NIPS*, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL `https://doi.org/10.1162/neco.1997.9.8.1735`.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 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)*, pp. 1426–1436, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1132. URL `https://www.aclweb.org/anthology/P18-1132`.

Tal Linzen. What can linguistics and deep learning contribute to each other? response to pater. *Language*, 95(1):e98–e108, 2019.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *CoRR*, abs/1611.01368, 2016. URL `http://arxiv.org/abs/1611.01368`.

Rebecca Marvin and Tal Linzen. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1151. URL `https://www.aclweb.org/anthology/D18-1151`.

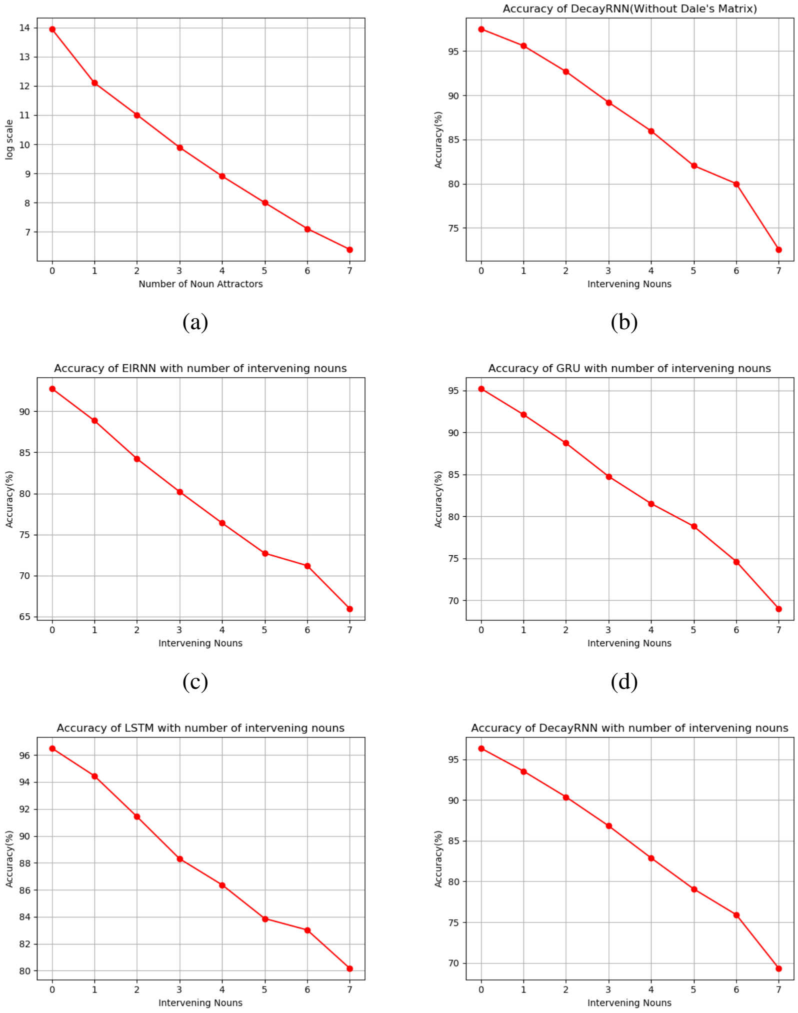
F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pp. 65–386, 1958.

Yikang Shen, Zhouhan Lin, Chin wei Huang, and Aaron Courville. Neural language modeling by jointly learning syntax and lexicon. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rkgOLb-0W`.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=B1l6qiR5F7`.

H. Francis Song, Guangyu R. Yang, and Xiao-Jing Wang. Training excitatory-inhibitory recurrent neural networks for cognitive tasks: A simple and flexible framework. *PLOS Computational Biology*, 12(2):1–30, 02 2016. doi: 10.1371/journal.pcbi.1004792. URL `https://doi.org/10.1371/journal.pcbi.1004792`.

A Appendix



(a) (b)

(c) (d)

(e) (f)

Figure 6: (a):Distribution of Number of Intervening Nouns in Linzen et al. (2016) dataset; (b),(c),(d),(e),(f): Performance scores of Slacked DecayRNN(No dale matrix), EIRNN($\beta = 0.001$), GRU, LSTM, DecayRNN on Grammaticality

Following tables (Table:4 to Table:11 for Grammaticality task; Table:12 to Table:19 for Verb Number prediction task) present an analysis of one on one comparison among the models. Every ij_{th} entry in the table means on how many example only $model_i$ (row model) is correct over $model_j$ (column model). These values are corresponding to testing set of size 1.35 million test examples.

	DRNN	EIRNN	GRU	LSTM	SDRNN
DRNN	-	57470	37227	24032	12876
EIRNN	22657	-	32786	21564	11654
GRU	30230	60602	-	22524	13797
LSTM	31740	64085	37229	-	14459
SDRNN	32244	65835	40162	26119	-

Table 4: No Intervening nouns

	DRNN	EIRNN	GRU	LSTM	SDRNN
DRNN	-	12685	8721	5377	3254
EIRNN	6335	-	7873	4910	3033
GRU	8215	13717	-	5043	3597
LSTM	9058	14941	9230	-	4052
SDRNN	9039	15168	9888	6156	-

Table 5: 1 Intervening nouns

	DRNN	EIRNN	GRU	LSTM	SDRNN
DRNN	-	5544	3783	2514	1736
EIRNN	3055	-	3490	2370	1636
GRU	4001	6197	-	2522	1992
LSTM	4373	6718	4163	-	2221
SDRNN	4358	6747	4396	2984	-

Table 6: 2 Intervening nouns

	DRNN	EIRNN	GRU	LSTM	SDRNN
DRNN	-	5544	3783	2514	1736
EIRNN	3055	-	3490	2370	1636
GRU	4001	6197	-	2522	1992
LSTM	4373	6718	4163	-	2221
SDRNN	4358	6747	4396	2984	-

Table 7: 3 Intervening nouns

	DRNN	EIRNN	GRU	LSTM	SDRNN
DRNN	-	885	677	429	363
EIRNN	614	-	665	430	371
GRU	782	1041	-	413	434
LSTM	890	1162	769	-	517
SDRNN	795	1074	761	488	-

Table 8: 4 Intervening nouns

	DRNN	EIRNN	GRU	LSTM	SDRNN
DRNN	-	401	301	207	186
EIRNN	309	-	276	191	179
GRU	389	456	-	196	225
LSTM	444	520	345	-	286
SDRNN	369	454	320	232	-

Table 9: 5 Intervening nouns

	DRNN	EIRNN	GRU	LSTM	SDRNN
DRNN	-	151	129	72	73
EIRNN	155	-	149	100	90
GRU	175	191	-	80	101
LSTM	221	245	183	-	135
SDRNN	185	198	167	98	-

Table 10: 6 Intervening nouns

	DRNN	EIRNN	GRU	LSTM	SDRNN
DRNN	-	91	70	43	46
EIRNN	93	-	90	57	63
GRU	90	108	-	44	67
LSTM	129	141	110	-	93
SDRNN	87	102	88	48	-

Table 11: 7 Intervening nouns

	DRNN	EIRNN	GRU	LSTM
DRNN	-	6271	5445	6238
EIRNN	5333	-	5621	6480
GRU	5684	6798	-	6604
LSTM	5118	6298	5245	-

Table 12: 0 Intervening nouns

	DRNN	EIRNN	GRU	LSTM
DRNN	-	3624	1965	2431
EIRNN	1982	-	1759	2146
GRU	2618	4054	-	2668
LSTM	2313	3670	1897	-

Table 13: 1 Intervening nouns

	DRNN	EIRNN	GRU	LSTM
DRNN	-	2283	1152	1342
EIRNN	1162	-	1023	1173
GRU	1626	2618	-	1537
LSTM	1468	2420	1189	-

Table 14: 2 Intervening nouns

	DRNN	EIRNN	GRU	LSTM
DRNN	-	1201	550	606
EIRNN	548	-	444	496
GRU	839	1386	-	704
LSTM	736	1279	545	-

Table 15: 3 Intervening nouns

	DRNN	EIRNN	GRU	LSTM
DRNN	-	562	268	306
EIRNN	260	-	224	229
GRU	440	698	-	334
LSTM	414	639	270	-

Table 16: 4 Intervening nouns

	DRNN	EIRNN	GRU	LSTM
DRNN	-	302	135	157
EIRNN	115	-	96	98
GRU	222	370	-	185
LSTM	202	330	143	-

Table 17: 5 Intervening nouns

	DRNN	EIRNN	GRU	LSTM
DRNN	-	119	60	63
EIRNN	60	-	38	49
GRU	106	143	-	77
LSTM	104	149	72	-

Table 18: 6 Intervening nouns

	DRNN	EIRNN	GRU	LSTM
DRNN	-	72	36	34
EIRNN	34	-	18	26
GRU	73	93	-	44
LSTM	72	102	45	-

Table 19: 7 Intervening nouns