Initial explorations on chaotic behaviors of Recurrent Neural Networks

Bagdat Myrzakhmetov[1,2], Zhenisbek Assylbekov[1], and Rustem Takhanov[1]

[1] School of Science and Technology, Nazarbayev University, Astana, Kazakhstan
{bagdat.myrzakhmetov, zhassylbekov, rustem.takhanov}@nu.edu.kz
[2] National Laboratory Astana, Nazarbayev University, Astana, Kazakhstan

Abstract. In this paper we analyzed the dynamics of Recurrent Neural Network architectures. We explored the chaotic nature of state-of-the-art Recurrent Neural Networks: Vanilla Recurrent Network and Recurrent Highway Networks. Our experiments showed that they exhibit chaotic behavior in the absence of input data. We also proposed a way of removing chaos from Recurrent Neural Networks. Our findings show that initialization of the weight matrices during the training plays an important role, as initialization with the matrices whose norm is smaller than one will lead to the non-chaotic behavior of the Recurrent Neural Networks. The advantage of the non-chaotic cells is stable dynamics. At the end, we tested our chaos-free version of the Recurrent Highway Networks (RHN) in a real-world application. In the language modeling task, chaos-free versions of RHN perform on par with the original version.

Keywords: Chaos Theory · Recurrent Neural Networks · Recurrent Highway Networks · Language Modeling.

1 Introduction

The dynamics of the Neural Networks has been studied in recent papers ([2, 4]). Laurent and Brecht ([7]) proposed to design architecture of a Recurrent Neural Network (RNN) cell in such a way that it is not chaotic. The concept of chaos ([6, 8]) comes from the theory of nonlinear dynamical systems and essentially means that wide divergence in outcomes of a system is due to small differences in initial conditions (such as those due to rounding errors in numerical computation). So, Laurent and Brecht show that the widely-used RNN cells, LSTM ([5]) and GRU ([3]), are chaotic. Depending on the initialization of the weights, LSTM and GRU might show a chaotic behavior. The proposed Chaos Free Network (CFN) architecture is devoid of chaos and is not inferior to LSTM. Recently, there were two main advancements over ubiquitous LSTM architecture: 1) Zoph and Le [20] used LSTM to generate a new RNN cell, which they refer to as a 'Neural Architecture Search' (NAS) cell; 2) Zilly et al. [1] extended the success of Highway networks ([12]) to recurrent networks and suggested a new RNN cell, which they refer to as a 'Recurrent Highway Network' (RHN) cell. Both, NAS and RHN cells significantly outperform the LSTM cell in language modeling

tasks when evaluated on a traditional PTB dataset ([11]). Therefore the following questions arise: Are these new state of the art architectures chaotic? If so, then according to Laurent and Brecht [7] there should be non-chaotic alternatives that do not underperform significantly. And if there are no such analogs, can chaos be necessary after all? We will try to answer these questions in this paper.

We explored the state of the art Recurrent Highway Networks (RHN, [1]) and vanilla RNN ([16]) for chaotic behavior. Our experiments showed that both RHN and vanilla RNN, depending on the initialization of the weights, might show a chaotic behavior. This chaotic behavior may lead to a high degree of sensitivity to the initial state in the long-term behavior of forward orbits. We found out that the initialization of the weight matrices heavily affects chaoticity of Neural Networks.

2 CHAOTIC NATURE OF SIMPLE VANILLA RNN

2.1 Vanilla RNN in 1D case

In this section we consider the dynamics of the Vanilla Recurrent Neural Network. Before analyzing the complex neural network architectures of Recurrent Highway Network (RHN) and Neural Architecture Search (NAS), we started by analyzing the simple Recurrent Neural Network (RNN), proposed by Elman in 1990 [16], for chaoticity. So, for the Simple RNN architecture we want to discuss the nonlinear map $f(x) = \tanh(Wh + Ux + b)$, where W and U are the weight matrices. We assume that there is no input data is provided, and the bias term is zero, so our system will become: $f(x) = \tanh(Wh)$.

In this subsection, we consider the simple RNN architecture in 1D case, we assume that our values $h, W \in \mathbb{R}$, i.e. our state and weight are scalars.

Claim 1 *A dynamical system induced by Simple RNN:*

$$h_{t+1} = \tanh(Wh_t), \quad h_t, W \in \mathbb{R} \tag{1}$$

is non-chaotic when $W \in (-1, 1)$.

Fixed point and Bifurcation Analysis One of the main goals of bifurcation theory [13] is to find the fixed points and the periodic points of maps and then look for the region of their stability. The fixed points of the mappings are calculated by solving the equation $f(x) = x$. For our case, $h = \tanh(Wh)$, for $W \leq 1$ there is only one solution: $h = 0$, for other values, there are 3 solutions.

The implicit plot of $h = \tanh(Wh)$ is given in Figure 1. Plots of h and $\tanh(Wh)$ for $W = 1, 2$ are provided in Figure 2a and 2b.

To discuss the stability of the above fixed points, we can use the stability criterion which say that if $|f'(x)|_{x=x^*} < 1$ then the fixed point $x = x^*$ is stable, otherwise it is unstable [13].

In our case, we have fixed points $h = 0$, $h_1(W)$ and $h_2(W)$. For the first fixed point $h = 0$: $|f'(h)|_{h=0} = W(1 - \tanh^2(Wh)) = W(1 - \tanh^2(0)) = W(1 - 0) =$

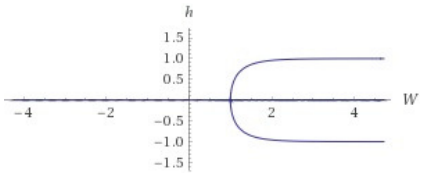
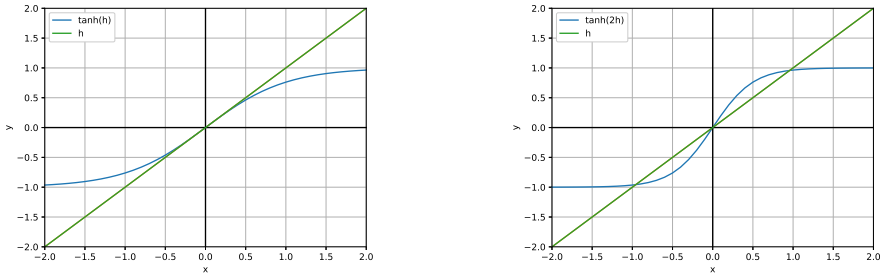


Fig. 1: Implicit plot of $h = \tanh(Wh)$.



(a) One solution of $h = \tanh(h)$. (b) Three solutions of $h = \tanh(2h)$.

Fig. 2: Solutions of $h = \tanh(Wh)$ for $W = 1$ and $W = 2$

W. So, by using the above notation, fixed point $h = 0$ is stable when $|W| < 1$, otherwise it is unstable. Hence $h = 0$ remains as a stable fixed point when $-1 < W < 1$.

As W crosses the value 1, the stable fixed point $h = 0$ becomes an unstable one. Thus, a qualitative change in the behavior of the fixed point occurs at $W = 1$ of the parameter value. So we consider $W = 1$ as the first bifurcation point. Also at $W = -1$, there also occurs a bifurcation.

To analyze the fixed points $h_1(W)$ and $h_2(W)$, we have to solve the inequality: $|W(1 - \tanh^2(Wh))| < 1$, where $h_1 \in (0, 1)$ and $h_2 \in (-1, 0)$. If this inequality holds, then the fixed points are stable. The solutions of this inequality will be $|W| \operatorname{sech}^2(Wh) < 1$. Most of the values of h_1 and h_2 are close to 1 and -1. If we put these values of h into the inequality $|W| \operatorname{sech}^2(Wh) < 1$, then this inequality holds for all W, therefore we do not consider these fixed points.

The bifurcation diagram of the $1D$ RNN is given in Figure 3.

Since our main aim is to study the long term behavior of the map, so, after understanding the behavior of the fixed points of $f = \tanh(Wh)$, we now consider the periodic points of period 2 and higher and look into their stability property. The period 2 points are fixed points of the second order iteration of the map. So, let us consider the iterated map $f^2(h)$.

If we draw the graph of $f^2(h) = \tanh(\tanh(h))$ for $W = 1$ with the line $x = y$, there will be only one point of intersection, which is $h = 0$, which is already our first order fixed point of f. This graph is shown in Figure 4.

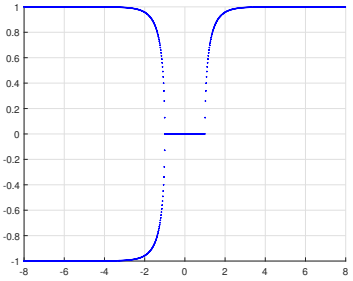


Fig. 3: Bifurcation diagram of the l1D RNN $h_{n+1} = \tanh(Wh_n)$.

The fixed points of f are also fixed points of f^2 as $f(h) = h \Rightarrow f(f(h)) = f(h) \Rightarrow f^2(h) = h$.

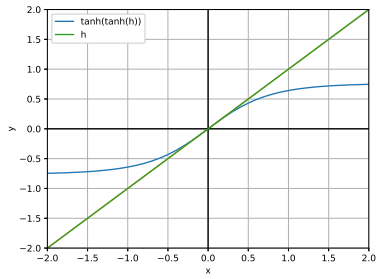


Fig. 4: Solutions of $h = \tanh(\tanh(h))$.

The period 2 points of the map are given by the solution of the equation: $f^2(h) = f(f(h)) = \tanh(W(\tanh(Wh))) = h$. We find the solutions $h \approx -1$, $h = 0$ and $h \approx 1$ for other values of W (except $W = 1$).

Let us see how the derivatives of the second iterate function change at the bifurcation value.

$$\frac{\partial}{\partial h}(\tanh(W\tanh(Wh))) = W^2\operatorname{sech}^2(Wh)\operatorname{sech}^2(W\tanh(Wh)).$$

Now let's analyze the bifurcation points:
$|f'(h)| = W^2\operatorname{sech}^2(Wh)\operatorname{sech}^2(W\tanh(Wh)) \Rightarrow$
$|f'(h)|_{h=1} = W^2\operatorname{sech}^2(W)\operatorname{sech}^2(W\tanh(W))$.

The value of the above equation is always less than the absolute values of $|1|$ for any values of W. So all points of W are stable on the second order periods.

So for all values of W, $-1 < W^2\operatorname{sech}^2(W)\operatorname{sech}^2(W\tanh(W)) < 1$ will be between -1 and 1.

Also for the second fixed point, we have

$$|f'(h)|_{h=-1} = W^2 \operatorname{sech}^2(-W) \operatorname{sech}^2(W \tanh(-W)).$$

Here also $-1 < W^2 \operatorname{sech}^2(-W) \operatorname{sech}^2(W \tanh(-W)) < 1$ is for any values of W. This means that these two fixed points of f^2 are stable fixed points for all values of W and they will not become unstable. Periodic points of period 2 will not occur.

Lyapunov Exponent analysis As said before, a chaotic system is sensitive to initial conditions. Lyapunov exponent is the rate at which nearby trajectories diverge from each other with time ([15]) and a measure for identifying the chaoticity of the system ([14]). Now let's consider two iterations of our map starting from two values of x which are very close to each other, i.e. with the very small difference δ: x_0 and $x_0 + \delta x_0$. Under the rule of the map let these points be shifted to $x_1, ..., x_n$ and $x_1 + \delta x_1, ..., x_n + \delta x_n$. If we expand $f(x)$ about x_n we have $\delta x_n = f'(x_{n-1})\delta x_{n-1}$ assuming that δx_n is sufficiently small. Hence the divergence of the two trajectories after n steps, δx_n, is related to their initial separation, δx_0, by $\left|\frac{\delta x_n}{\delta x_0}\right| = \prod_{i=0}^{n-1} \left|f'(x_i)\right|$.

We expect that this will vary exponentially at large n, $\left|\frac{\delta x_n}{\delta x_0}\right| = e^{\lambda n}$. So the Lyapunov exponent is defined by $\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} ln\left|f'(x_i)\right|$.

Obviously, if $\lambda > 0$, neighboring trajectories separate from each other at large n, which corresponds to chaos. However, if trajectories converge to a fixed point or a limit cycle they will get closer together, which corresponds to $\lambda < 0$. Hence, we can determine whether or not the system is chaotic by the sign (+ or -) of the Lyapunov exponent. Below, we have given the calculated values of the Lyapunov exponent for some values of the parameter W in case of the Simple RNN map. We have considered iteration size of 100000 to get the values.

In our case, $f(h) = \tanh(Wh)$ and $f'(h) = W(1 - \tanh^2(Wh))$.

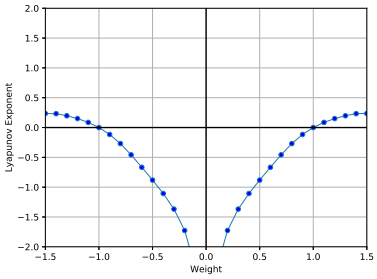


Fig. 5: Lyapunov coefficient versus W value.

In Figure 5 we have shown the values of Lyapunov Exponent versus the weight value W. From this Figure we can see that the values of -1.1 and 1.1 of

W the Lyapunov Exponents become positive, showing the beginning of a chaotic region. Also this Figure 5 further supports the first two bifurcation points as -1.0 and 1.0 where the Lyapunov Exponent is almost zero. Interestingly, after attaining the chaotic region at $W = 1.0$ and $W = -1.0$, we see the negative Lyapunov exponent values. They signify that within the chaotic region also, at certain values of the parameter, there are regular behaviors. This is supported also by the bifurcation diagram which we have drawn in Figure 3 in the previous section. (After some values of W, it will stabilize).

2.2 Multidimensional case for Vanilla RNN

Now, consider the high dimension cases. For vanilla RNN:

$$h_{(t+1)} = \tanh(Wh_t),\ h_t \in R^d \tag{2}$$

Claim 2 *If $||W|| < 1$, then for (2) we have the following statement: for any h_0 we have $lim_{(n\to\inf)} h_t = 0$.*

Proof. $||h_{t+1}|| = ||\tanh(Wh_t)|| \leq ||Wh_t|| \leq ||W||||h_t||$. Therefore, $||h_t|| \leq ||W||^t||h_0|| \to 0$, $t \to \inf$.

Claim 3 *There exists W with $||W|| > 1$, such that induced dynamical system (2) is chaotic (means that there should be at least 1 nontrivial attractor, i.e. attractor which is not a point).*

When do we have these Claims 2 and 3? Let $||h||$ be a norm on $\mathbb{R}^d$ such that $||\tanh(h)|| \leq ||h||$. Examples of such norm are:

a) $||h||_p = (\sum_{i=1}^{d} h_i^p)^{1/p}$ is a l_p norm. For any such norm let us define corresponding matrix norm as $||W||_p = max_{h:||h||_p=1}||Wh||_p$.

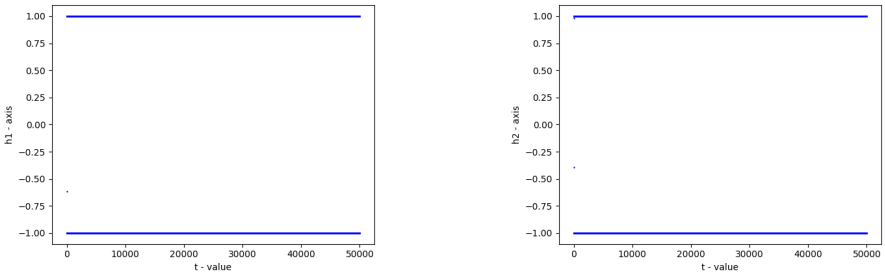
Now, we tested the weight matrix, the norm of which is greater than 1. Lets consider the weight matrix $W = \begin{bmatrix} -1 & -4 \\ -3 & -2 \end{bmatrix}$. If we plot the graph of $h^{(1)}$ vs. t and $h^{(2)}$ vs. t, then we get the graphs shown in Figures 6a and 6b.

In the above example, all norms are larger than one (Frobenius norm, nuclear norm (trace norm), max norm, l_1 norm, l_2 norm).

3 CHAOTIC BEHAVIOR OF RHN

After exploring the vanilla RNN, we considered the dynamics of the state of the art RNN architectures. Recurrent Highway Network (RHN) was proposed by Zilly et al. [1] and introduced a new theoretical analysis based on the Geršgorins circle theorem [17]. This theorem helps to clarify many optimization issues and modeling. Their approach allows transition depths to be larger than one.

The main idea behind increasing the depth of the step-to-step recurrent state transition is to allow the RNN tick for several time steps per step of the sequence



(a) $h^{(1)}$ vs t (b) $h^{(2)}$ vs t

Fig. 6: State vs. time graphs for $2D$ case when the norm is larger than one

([18, 19]). By using this technique we can adapt the recurrence depth to the problem.

RHN architecture is given in the following form: the Highway layer computation is defined as:

$$y = h \odot t + x \odot c \tag{3}$$

where

$$t := \sigma(W_t x + R_t s + b_t); \tag{4}$$

$$h := \tanh(W_h x + R_h s + b_h); \tag{5}$$

And then the RHN layer is defined as:

$$s_{n+1} = t \odot (h - s_n) + s_n. \tag{6}$$

$\odot$ denotes Hadamard product.

3.1 RHN chaoticity in 1D

In this subsection we analyze the dynamics of the Recurrent Highway Network (RHN) in 1D case. First, we can start with the analysis of fixed points and check the region of their stability.

If we assume that no input is provided, then the induced form of the RHN will become:

$$t := \sigma(R_t s); \tag{7}$$

$$h := tanh(R_h s); \tag{8}$$

$$s_{n+1} = t \odot (h - s_n) + s_n. \tag{9}$$

If we put everything together, we will get the following equation:

$$s_{n+1} = \sigma(R_t s_n) \odot (tanh(R_h s_n) - s_n) + s_n. \tag{10}$$

For the Recurrent Highway Networks we have the following claim.

Claim 4 *A dynamical system induced by RHN in Equation 10 shows non-chaotic behavior when $W \in (-1, 1)$, as thus follows from the properties of Vanilla RNN.*

Proof. To find the fixed points, we have to solve the equation: $x = f(x)$, so for the Equation 10 we will have:

$$\begin{aligned} s &= \sigma(R_t s) \odot (tanh(R_h s) - s) + s \Rightarrow \\ 0 &= \sigma(R_t s) \odot (tanh(R_h s) - s) \Rightarrow \\ 0 &= \sigma(R_t s) \text{ and } 0 = (tanh(R_h s) - s) \end{aligned}$$

$0 = \sigma(R_t s) \Rightarrow$ no solutions exists, as the values of sigmoid function lies between 0 and 1. So we will consider only the second part.

$$\begin{aligned} 0 &= tanh(R_h s) - s \Rightarrow \\ s &= tanh(R_h s) \end{aligned}$$

This equation $s = tanh(R_h s)$ is the case for the simple RNN. We already considered the fixed point analysis of the simple RNN in section 2.1. So the fixed points of the Recurrent Highway Networks are the same as the vanilla RNN and the fixed point analysis of the Vanilla RNN can be applied for the RHN. This proves our Claim 4.

3.2 RHN chaoticity in 2D

Claim 5 *There exists W: $||W|| > 1$ such that a dynamical system induced by RHN in Equation 10 is chaotic.*

We performed experiments to check the chaotic behavior of the RHN in 2D. We show that in the absence of the input data RHN can lead to dynamical systems $s_{n+1} = \Phi(s_n)$ that are chaotic ([6]). Again, we assume that there is no input data is provided. Then the dynamical system induced by a two-dimensional RHN with weight matrices: $R_t = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $R_h = \begin{bmatrix} -5 & -8 \\ 8 & 5 \end{bmatrix}$ and zero bias for the model. s can be initialized with any values. If we assume that no input data is provided and all bias terms are zero, then the induced RHN architecture will become as in Equations 7, 8 and 9.

Now we plot the RHN state values $s_n^{(1)}$ vs. $s_n^{(2)}$ for $n = 100000$ iterations. The resulting plot is shown in Figure 7. Most trajectories converge toward the depicted attractor. We can get above pictures for any initial values of s (we can initialize with zeros or any values) and for any number of highway layers (we tried 1, 5, 10 highway layers). This picture shows the strange attractor as in LSTM and GRU given in Laurent and Brecht [7].

Now we studied time series analysis of this system. If we plot s^1 vs. n we can notice that the values of s^1 will jump from one place to another in the chaotic manner. There is no convergence. This is given in Figure 8a. This is also true for s^2 vs. n (given in Figure 8b). Then, if we plot the graph s^1 vs. s^2, we can get the strange attractor as shown in Figure 7.

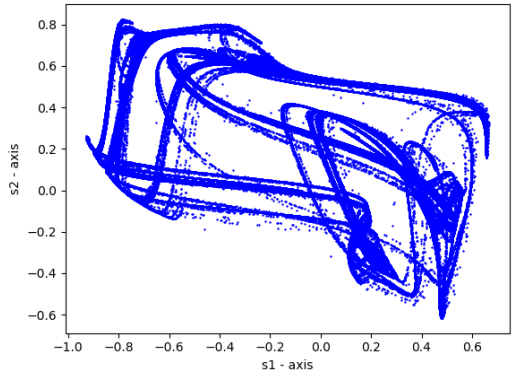
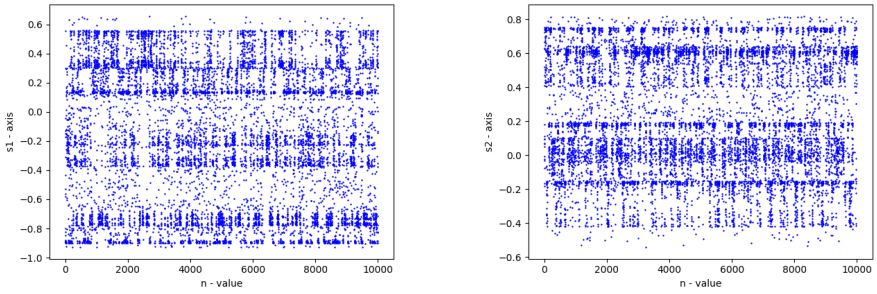


Fig. 7: Strange attractor of chaotic behavior of RHN for the weight matrices: R_t=[[0,1],[1,0]] and R_h=[[-5,-8],[8,5]]



(a) $s^{(1)}$ vs n (b) $s^{(2)}$ vs n

Fig. 8: State vs. time graphs for 2D case RHN

Next we tested chaoticity of the RHN by using the Lyapunov instability of Bernoulli shift [8] as in section 2.1. We consider the two points which are initially very close to each other, with δx_0 "infinitesimally small" differences: $\delta x_0 := |x_0' - x_0|$. Then we iterate these two points through our induced RHN map $s_{n+1} = \Phi(s_n)$, in Equation 9, 100 times and calculated the Euclidean distance between $|\hat{s}_n - s_n|$ these points. The graph is given in Figure 9. From this graph we can see that after some iteration, two trajectories diverge exponentially despite the fact that initially these two points are highly localized, with the distance no more than 10^{-7}.

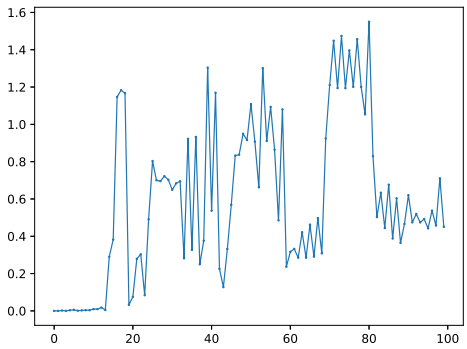


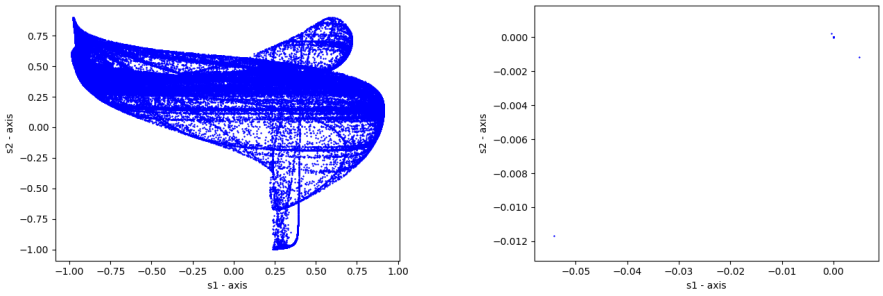
Fig. 9: $|\hat{s}_n - s_n|$ for 2 trajectories: x_0 and $x_0 + \delta x_0$

Also we tested the weight matrices $R_t = \begin{bmatrix} -2 & 6 \\ 0 & -6 \end{bmatrix}$ and $R_h = \begin{bmatrix} -5 & -8 \\ 8 & 5 \end{bmatrix}$. The norm of these matrices are larger than one. If we plot the graph of s^1 vs. s^2 with $n = 100000$, then we again explore the strange attractor as shown in Figure 10a. Here again, for any initial value of s and for any number of highway layers we will get this picture.

After exploring the chaotic behavior in RHN, we now tried to build chaos-free Neural Networks. For RHN, we again use our Claim 2 which was applied in vanilla RNN. Here, if we initialize the weights with matrices whose norm is smaller than one, then again we can have the non-chaotic behavior in RHN.

In the above cases, the norm of the weight matrices are larger than one and we explored the chaotic behavior. Now, let's analyze the case when the norm of the matrices are smaller than 1. For example, we can test these weight matrices: $R_t = \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix}$ and $R_h = \begin{bmatrix} -0.5 & -0.8 \\ 0.8 & 0.5 \end{bmatrix}$.

The norm of these two matrices are smaller than one. If we explore the values of s^1 and s^2 for $n \to \text{inf}$, then, both values of s will go to zero. We also plotted the graph of s^1 vs. s^2. The plot is given in Figure 10b. From this, we can see



(a) Strange attractor of chaotic behavior of RHN for the weight matrices: R_t = [[-2,6], [0,-6]] and R_h = [[-5,-8], [8,5]]

(b) Attractor for weight matrices: R_t = [[0, 0.5], [0.5, 0]] and R_h=[[-0.5, -0.8], [0.8, 0.5]]

Fig. 10: Strange and regular attractor of chaotic and non-chaotic RHN for 2D case

that we can get non-chaotic RHN when we initialize the weight matrices with the values whose norm is smaller than one.

4 EXPERIMENTS

In this section, we tested our non-chaotic neural cells in real-world applications. Our aim is to identify, how non-chaotic version will affect on the performance. Is a non-chaotic behaviour good in a real-world application? Do we need a chaoticity? Or is it good to have a chaotic systems? To answer these questions, we performed experiments.

We examined Recurrent Highway Networks on the language modeling task. We use Penn Tree Bank (PTB) [11] corpus, which was pre-processed by Mikolov et al. [10]. First we reproduced the initial results from Zilly et al. [1] without weighting (WT) of input and output mappings and got the 68.355 perplexity on the validation set and 65.506 perplexity on the test set. These results are similar to the results in the paper (In the paper it was 67.9 and 65.4).

Then we tested our chaos-free version. We initialized the weight matrix in a way, such that their Frobenius norm do not exceed 1. We use TensorFlow ([9]) to perform our experiments. We first created a matrix whose norm is smaller than one and feed it during the initialization. We used the same hyper-parameters as in Zilly et al. [1] during the training. On PTB dataset, our non-chaotic neural cells showed 68.715 perplexity on the validation set and 66.290 perplexity on the test set. Full results and results of Chaos Free Network (CFN) [7] are given in Table 1. From this, we can see that the chaos free version of RHN showed similar results as the chaotic version and that chaos-free initialization will not lead to a decrease in performance.

Table 1: Perplexity on the PTB set.

Model	Validation Perplexity	Test Perplexity
Variational RHN + WT [1]	68.355	65.506
Non-chaotically initialized RHN	**68.715**	**66.290**
CFN (2 layers)+dropout [7]	79.7	74.9

5 Conclusion and future work

In this paper we analyzed the dynamics of the Recurrent Neural Networks. Our analyses showed that the vanilla RNN and the most recent RHN architecture exhibit a chaotic behavior in the absence of input data. We found out that, depending on the initialization of the weight matrices, we can have non-chaotic systems. Our experiments showed that the initialization of the weights with the matrices whose norm is less than one can lead to non-chaotic behavior. The advantage of non-chaotic cells is stable dynamics. We also performed experiments with non-chaotic RHN cells. Our experiments on language modeling with the PTB dataset showed similar results as an RHN cell with chaos by using the same hyper-parameters. In the future, we are going to test non-chaotic RHN cells for other tasks: speech processing, image processing. Also for NAS architecture, at this moment, generating the architecture is an expensive process for us, as there are not enough resources. We will test our chaos-free initialization for NAS architectures again.

Acknowledgement

This work has been funded by the Committee of Science of the Ministry of Education and Science of the Republic of Kazakhstan, contract #346/018-2018/33-28, IRN AP05133700. The work of Bagdat Myrzakhmetov partially has been funded by the Committee of Science of the Ministry of Education and Science of the Republic of Kazakhstan under the research grant AP05134272. The authors would like to thank Professor Anastasios Bountis for his valuable feedback.

References

1. J. G. Zilly, R. K. Srivastava, J. Koutník, and J. Schmidhuber. Recurrent Highway Networks. In *International Conference on Machine Learning*, 4189–4198 (2017).
2. D. Sussillo and O. Barak. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural computation*, **25**(3):626–649 (2013).
3. K. Cho, B. Van Merrinboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
4. R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, 1310–1318 (2013).

5. S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, **9**(8):1735–1780 (1997).
6. S. H. Strogatz. Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering. Westview press (2014).
7. T. Laurent and J. von Brecht. A recurrent neural network without chaos. *arXiv preprint arXiv:1612.06212* (2017).
8. E. Ott. Chaos in dynamical systems. Cambridge university press (2002).
9. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard and M. Kudlur. Tensorflow: a system for large-scale machine learning. In *OSDI* Vol. 16, pp. 265–283 (2016, November).
10. T. Mikolov, and M. Karafiát, L. Burget, J. Černockỳ and S. Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association* (2010).
11. M. P. Marcus, M. A. Marcinkiewicz and B. Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, **19(2)**, pp.313–330 (1993).
12. R. K. Srivastava, K. Greff and J. Schmidhuber. Training very deep networks. In *Advances in neural information processing systems (NIPS)*, pp. 2377–2385 (2015).
13. Y. A. Kuznetsov. Elements of applied bifurcation theory. (Vol. 112). Springer Science & Business Media (2013).
14. A. Wolf, J. B. Swift, H. L. Swinney and J. A. Vastano. Determining Lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena*, **16(3)**, pp.285–317 (1985).
15. A. M. Lyapunov. The general problem of the stability of motion. *International journal of control*, 55(3), pp.531–534 (1992).
16. J. L. Elman. Finding structure in time. *Cognitive science, 14(2)*, pp.179-211 (1990).
17. S. Geršgorin (S. Gerschgorin). Über die Abgrenzung der Eigenwerte einer Matrix *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, no. 6, 749–754 (1932).
18. R. K. Srivastava, B. R. Steunebrink and J. Schmidhuber. First experiments with POWERPLAY. *Neural networks: the official journal of the International Neural Network Society*, 41, pp.130-136 (2013).
19. A. Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983* (2016).
20. B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578* (2016).