Bachelor thesis

Artificial Intelligence

Radboud University 

Robustness of rate-based recurrent neural networks to cell death

Author:
Max May
S1004443

First supervisor:
dr. S.W. Keemink
Donders Centre for Cognition
sander.keemink@donders.ru.nl

Second reader:
dr. R.S. van Bergen
Donders Centre for Cognition
ruben.vanbergen@donders.ru.nl

July 7, 2022

Abstract

Rate-based recurrent neural networks are popular and successful in the field of machine learning. However, they are not well-known for being robust to cell silencing, and studies on this subject are scarce. In addition, recurrent neural networks' calculations are still difficult to understand and visualize. Work has been done on spiking neural networks to visualize the system's behaviour, specifically its behaviour after cell silencing. The studies demonstrated the conditions in which spiking networks are robust. However it is not clear whether we can think alike for non spiking networks. In this study, we will utilize ordinary differential equations so that we may compute the continuous trajectories of the network's behaviour, using differential equations. We shall demonstrate the consequences of cell silencing on the behaviour of the network using a revised equation for the state space of the network. Which will allow us to visualize the neuronal contributions as vectors in the phase plots. This approach yields new insights into what happens when neurons are silenced in a network. In particular it shows that silencing a neuron corresponds to the removal of that vector in the phase plots and that recurrent neural networks become more robust as the amount of neurons in the networks grows. This allows for greater insight on what the consequences of cell death are to the robustness of rate-based recurrent neural networks. Through visualization, this will also aid in the comprehension of and prevention of overfitting in recurrent neural networks.

Contents

Chapter 1

Introduction

Robustness is the ability to maintain normal function after withstanding perturbations, e.g., neuronal cell death. Biological systems appear to be robust to cell death [1, 2, 3]. Some artificial neural networks now also include cell death during the training phase, as a means to prevent overfitting [4, 5, 6]. However, it is unknown whether these artificial neural networks are robust. So the question then becomes, under what conditions are artificial neural networks robust to cell death and is there a way to visualize how these networks operate.

An increasing number of current day artificial neural networks include one or more "dropout" layers during the training phase, to decrease overfitting of the network. This dropout layer's function is to intentionally terminate some of the neurons, so that none of the neurons can be depended on and so every neuron has to become more robust and learn more general features [4, 5, 6]. However studies have shown that recurrent neural networks do not work well with dropout layers [7]. Due to the fact that these dropout layers are commonly used during the training phase and not during the actual network function. It is not clear how well this knowledge extents to the explanation of behavioural change due to the neuron silencing in recurrent networks.

However, Calaim et al.(2020) demonstrated a very clear functional robustness and understanding in spiking neural networks. They researched a way to get a better grasp of how spiking neural networks behave to various perturbations (e.g. noise, varying number of neurons, firing thresholds, etc.) [8, 9, 10]. They achieved this by conveying simple geometric figures to illustrate the behaviour of this network, see figure 1.1, and to what extent their network preserved its function under the varying conditions. Their network functionality is described as staying within a geometric object (in figure 1.1 this is the hexagon, surrounding the plus sign). The findings of their study were that the network preserved this functionality as long as the perturbations did not destroy the shape of the geometric object (i.e.,

the hexagon); meaning that as long as the remaining neurons could form an adequate bounding box, that could contain the output inside the geometric object the network would be considered robust, since it retained its functionality [10, 11].

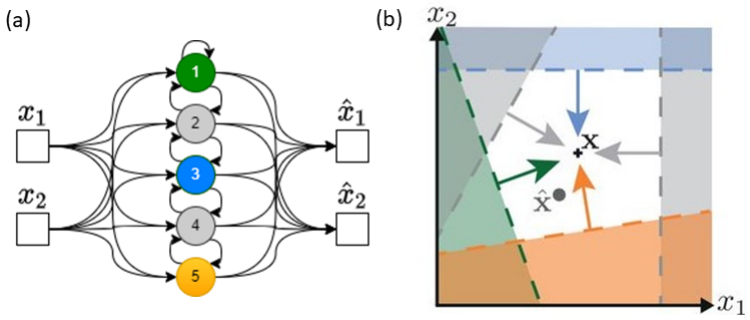


Figure 1.1: Geometric representation of a 'bounding box' and the corresponding network. (a) Network representation of a spiking network, with five neurons. The grey neurons are silenced. The inputs x_1, x_2 and the predicted outputs $\hat{x}_1, \hat{x}_2$ (b) Geometric representation, where every colored dashed line represents a neuron and the grey lines represent neurons that have been terminated. The arrow that is perpendicular to the dashed lines is the direction that the output is forced in, if the output reaches that line; reproduced from Calaim et al. (2020)

The visualization and understanding of recurrent networks is difficult is due to the fact that these networks often have a high number of neurons and outputting their behaviour directly would result in a high-dimensional output. Making it difficult to interpret the figures. Additionally recurrent networks typically operate in discrete time, meaning that if the trajectories would be visualized, they would be coarse. Often when neurons stop working, many cannot explain why the behaviour of the network changes, which results in the so called "black box" models [12].

In order to visualize the robustness of recurrent neural networks we will work on small networks, where it shall be assumed that the recurrent networks are ordinary differential equations (ODEs) (i.e., dynamical systems) and the focus will be on networks with low-rank representations. The small networks will be used to build an intuition, which later can be expanded to bigger networks. Dynamical system theory will be used to understand their workings. In particular phase plots will be used to understand, in detail, the behaviour of recurrent networks. After showing how to interpret the networks as ODEs and visualizing the vector fields, we will show how to interpret individual neuron contributions to the vector field. Allowing us to understand why a network's behaviour changes, when a neuron is silenced.

This visualisation method aims to solve the problem we have, where we cannot comprehend what will happen to a network when a neuron dies. It makes recurrent neural networks easier to understand. And with the help of this intuition, we are able to explain when recurrent networks are robust, but will not risk overfitting.

Chapter 2

Methods

2.1 Networks

Although Calaim et al. (2020) worked on the geometric visualization of spiking neural networks, this work cannot be extended to recurrent neural networks. Consequently, there is still a lack in both the visualization of recurrent neural networks and our capacity to develop an intuition for these networks, regarding what transpires during cell death and how recurrent networks remain robust.

2.1.1 Ordinary Differential Equations

To investigate where robustness in rate-based recurrent networks comes from and to visualize this, we want to simulate a recurrent network for which the trajectory can be calculated so that its inner mechanisms can be comprehended. To simulate the trajectory of a recurrent network, means that the future location of the network must be known for all moments in time. The derivative of a network's behaviour is required, in order to calculate this trajectory. In this case behaviour is the route taken by the network to reach is stabilization point. To see where the network is headed, the derivative of behaviour at specific time points will be utilized. However, recurrent networks often operate in discrete time, resulting in a coarse trajectory (Fig. 2.1 Left). In the study Chen et al, they demonstrated that recurrent neural networks are a rough approximation of an underlying Ordinary Differential Equation (ODE) [13]. This indicates that a recurrent neural network's rough approximation can be replaced by its smoother equivalent, i.e., the ODE. These differential equations enable continuous calculations of the trajectories, which in turn enables a smoother visualization of the network's

trajectory (Fig. 2.1 Right).

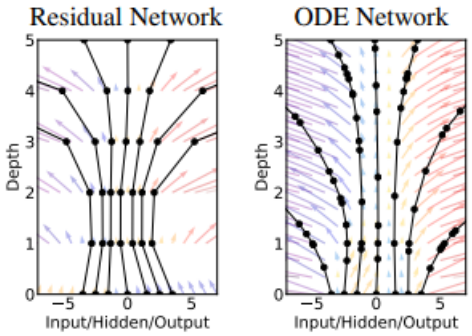


Figure 2.1: Adapted from Chen et al. Left: A Residual network defines a discrete sequence of finite transformations. Right: A ODE network defines a vector field, which continuously transforms the state. Both: Circles represent evaluation locations[13].

2.1.2 General recurrent networks

Building an intuition on how recurrent networks behave under different circumstances requires a network where the trajectories of the network using ODEs can be plotted, as explained in the previous section.

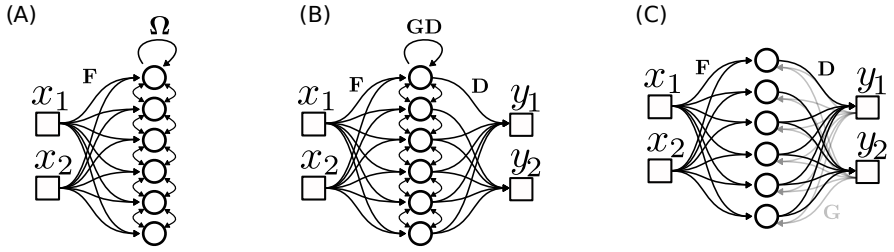


Figure 2.2: Adapted from a figure by S.W. Keemink with permission

To achieve this, the general recurrent network (Fig. 2.2A) can be described by the dynamical equation.

$$\dot{\mathbf{r}}(\mathbf{t}) = -\lambda \mathbf{r}(t) + f(\mathbf{F}\mathbf{x}(t) + \mathbf{\Omega}\mathbf{r}(t)), \tag{2.1}$$

where $\mathbf{r} \in \mathbb{R}^N$ stands for the neurons' output activity of dimensionality N, for N neurons, λ is the leak time-constant, $\mathbf{x}(t) \in \mathbb{R}^K$ are the K-dimensional inputs, $\mathbf{F} \in \mathbb{R}^{N \times K}$ are the forward weights, $\mathbf{\Omega} \in \mathbb{R}^{N \times N}$ are the recurrent weights and $f()$ is the Sigmoid activation function (2.2).

$$f(t) = \frac{1}{1 + e^{-(\mathbf{F}\mathbf{x}(t) + \mathbf{\Omega}\mathbf{r}(t))}}. \tag{2.2}$$

2.1.3 Low-rank networks

When the behaviour of the independent neurons is directly outputted, neural networks get a N-dimensional output. When $N > 3$, the dimensionality is already too great for visualizing the behaviour of the network. This in turn challenges the possibility to build an intuition for the network. To understand the output of the network its behaviour will be visualized in a two-dimensional plot, meaning that the behaviour should be outputted in a two-dimensional output-space. This means that the network should have a two-dimensional output-space $[y_1, y_2]$. To reduce the N dimensional network back to a two-dimensional output and keep the recurrence, a decoding matrix $\mathbf{D}$ was added to the network and an encoding matrix $\mathbf{G}$ (Fig. 2.2B). It is assumed that the recurrent connectivity $\mathbf{\Omega}$ is given by $-\mathbf{GD}$ resulting in a network, where the connectivity is constrained to be low-rank, such that the output is fed back into the neurons (Fig. 2.2C).

$$\dot{\mathbf{r}}(t) = -\lambda \mathbf{r}(t) + f(\mathbf{F}\mathbf{x}(t) - \mathbf{GDr}(t)), \tag{2.3}$$

where recurrent connectivity is now explained by the encoding matrix $\mathbf{G} \in \mathbb{R}^{N \times M}$, and the decoding matrix $\mathbf{D} \in \mathbb{R}^{M \times N}$

We assume $M < N$ because this results in a low-rank connectivity within the network and low-dimensional dynamics[14, 15]. Allowing us to get a grasp on the network behaviour in the output-space $[y_1, y_2]$. However, this equation allows for the interpretation of the neural contributions $\mathbf{r}$, but it does not allow for the interpretation purely in the output-space. So then it becomes useful to define the output of the network as the combination of neuronal behaviour $\mathbf{r}$ and the decoding matrix $\mathbf{D}$, resulting in $\mathbf{y} = \mathbf{Dr}$. To receive the dynamics of this network, the time-derivative $\dot{\mathbf{y}}$ has to be calculated. From this point on the time steps t will be omitted for clarity,

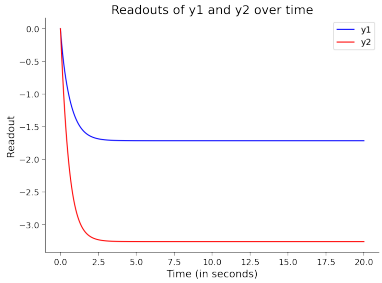
$$\dot{\mathbf{y}} = \mathbf{D}\dot{\mathbf{r}} = -\lambda \mathbf{Dr} + \mathbf{D}f(\mathbf{Fx} - \mathbf{GDr}). \tag{2.4}$$

However, to be able to plot the phase plot of the network, the derivative as a function of $\mathbf{y}$ is needed. So $\mathbf{Dr}$ will be substituted with $\mathbf{y}$ to get the function,

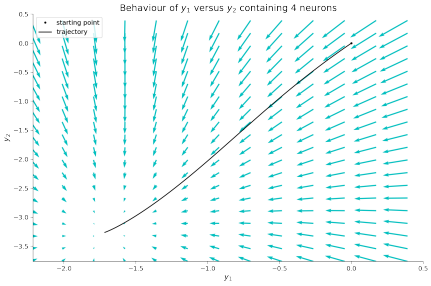
$$\dot{\mathbf{y}} = -\lambda \mathbf{y} + \mathbf{D}f(\mathbf{Fx} - \mathbf{Gy}). \tag{2.5}$$

Which allows for the visualization of two figures, in which the readouts for the output-space $[y_1, y_2]$ can be visualized in a separate manner (Fig. 2.3A) and against each other (Fig. 2.3B). This allows for the visualization of when the network reaches a stable point, so where y_1 and y_2 become a horizontal line (Fig. 2.3A) and where the black curve ends (Fig. 2.3B). It also allows for the description of the full dynamics of the network in

the output-space (vector field showing the trajectory of the network in Fig. 2.3B).



(a) Readouts of the 4 neuron network



(b) Behaviour with the vector field

Figure 2.3: Both figures represent the same simulation of a network containing 4 neurons. (a) The readouts for the outputs y_1 and y_2, stabilizing around 2.5 seconds. (b) The behaviour of the network when you plot the readouts from (a) against each other. The blue arrows represent the vector field, which show where the network is headed for a specific moment in time.

2.2 Neuronal contributions

The purpose of the study is to visualize the behaviour of recurrent networks undergoing cell death, so that an intuition can be build on where the robustness in these systems stems from. Therefore it is important to have the ability to show the neuronal contributions of each neuron in the network, so that the difference in behaviour between a normal network and one where a neuron has ceased working may be observed. To visualize the contribution of each neuron, their contributions need to be extracted from the calculation of the full state space (i.e., the calculation of every trajectory for every possible starting point). To obtain the calculation for the full state space, we need to make a slight modification to eq.(2.5). We choose $\frac{1}{\tau}$ as the leaky time integrator, resulting in:

$$\dot{\mathbf{y}} = -\frac{\mathbf{y}}{\tau} + \mathbf{D}f(\mathbf{F}\mathbf{x} - \mathbf{G}\mathbf{y}). \tag{2.6}$$

$\mathbf{D}f(\mathbf{F}\mathbf{x} - \mathbf{G}\mathbf{y})$ can be extended to include the matrix used in the activation function.

$$\mathbf{D}f(\mathbf{F}\mathbf{x} - \mathbf{G}\mathbf{y}) = \mathbf{D}f(\begin{bmatrix} \mathbf{F}_1^\top \mathbf{x} - \mathbf{G}_1^\top \mathbf{y} \\ \vdots \\ \mathbf{F}_N^\top \mathbf{x} - \mathbf{G}_N^\top \mathbf{y} \end{bmatrix}) \tag{2.7}$$

$\mathbf{D}$ can be written as a row vector and $f()$ can be inserted in the matrix of $\mathbf{Fx} - \mathbf{Gy}$

$$\mathbf{D}f(\begin{bmatrix} \mathbf{F}_1^\top \mathbf{x} - \mathbf{G}_1^\top \mathbf{y} \\ \vdots \\ \mathbf{F}_N^\top \mathbf{x} - \mathbf{G}_N^\top \mathbf{y} \end{bmatrix}) = \begin{bmatrix} \mathbf{D}_1 & \cdots & \mathbf{D}_N \end{bmatrix} \begin{bmatrix} f(\mathbf{F}_1^\top \mathbf{x} - \mathbf{G}_1^\top \mathbf{y}) \\ \vdots \\ f(\mathbf{F}_N^\top \mathbf{x} - \mathbf{G}_N^\top \mathbf{y}) \end{bmatrix} \tag{2.8}$$

The matrix multiplication can also be written as the summation of $\mathbf{D}f(\mathbf{F}^\top \mathbf{x} - \mathbf{G}^\top \mathbf{y})$ over every neuron:

$$\dot{\mathbf{y}} = -\frac{\mathbf{y}}{\tau} + \sum_{i=1}^{N} \mathbf{D}_i f(\mathbf{F}_i^\top \mathbf{x} - \mathbf{G}_i^\top \mathbf{y}) \tag{2.9}$$

With N being the amount of neurons in the network, $\mathbf{y}$ the output at that point in time, $\frac{1}{\tau}$ being the leaky time integrator, $\mathbf{D_i}$ being the decoding vector for neuron $i \in N$, $\mathbf{F_i}$ being the encoding vector for neuron $i \in N$ and $\mathbf{G_i}$ being the encoding vector for neuron $i \in N$, allowing for the recurrent connectivity.

The calculation for a single vector in the vector field now consists of the decay $(-\frac{\mathbf{y}}{\tau})$ and the summation of $(\mathbf{D}_i f(\mathbf{F}_i^\top \mathbf{x} - \mathbf{G}_i^\top \mathbf{y}))$ for neuron $i \in N$. This results in the ability to visualize both parts of eq. 2.9.

2.3 Cell death

Calaim et al. (2020) developed a geometric representation of what happened to spiking neural networks, demonstrating how the behaviour altered as individual neurons were silenced. Due to the difference between recurrent networks and spiking networks, this method cannot be expanded to the visualization of recurrent networks. But the problem will be approached in the same way.

2.3.1 Cell death in neural networks

To develop an intuition for what happens to the behaviour of a recurrent network undergoing cell death, the ability to silence individual neurons is needed. In the previous section 2.1.3, it is explained that equation (2.3) can be seen as the calculation of the neuron's output activity $\mathbf{r}$, where $-\mathbf{GDr}$ expresses the recurrent connectivity. When a neuron has been silenced (and thus theoretically has died), it cannot output its activity anymore and thus $\mathbf{r} = 0$. To visualize the biggest shift in behaviour, the network is allowed to stabilize before silencing a neuron. Based on the time needed for the network to stabilize, a certain termination point (t_s) is selected to simulate

this. From this point on, the activity for an arbitrary neuron $p \in N$ is set to be zero ($\mathbf{r}_p = 0$, for $[t_s,t_{end}]$, with neuron $p \in N$). Combining this with the fact that the output is $\mathbf{y} = \mathbf{Dr}$, results that for $\mathbf{r}_p = 0$, the equation evaluates to 0. So the silenced neuron has no influence on the output and remaining neurons. This allows for the network's new behaviour to be plot visualized.

2.4 Results

Visualization of a network's behaviour experiencing cell death is still lacking. This in turn makes it harder to gain an intuitive understanding of these networks. So to develop this intuitive understanding, the results of the phase plots of networks of different sizes, undergoing cell death will be shown.

2.4.1 Small networks

Firstly small networks ($N = 4$) will be simulated to make sure that it is understood what occurs when a neuron dies and to make sure the difference in behaviour will be significant. All of the networks are simulated for $t = 20$s.

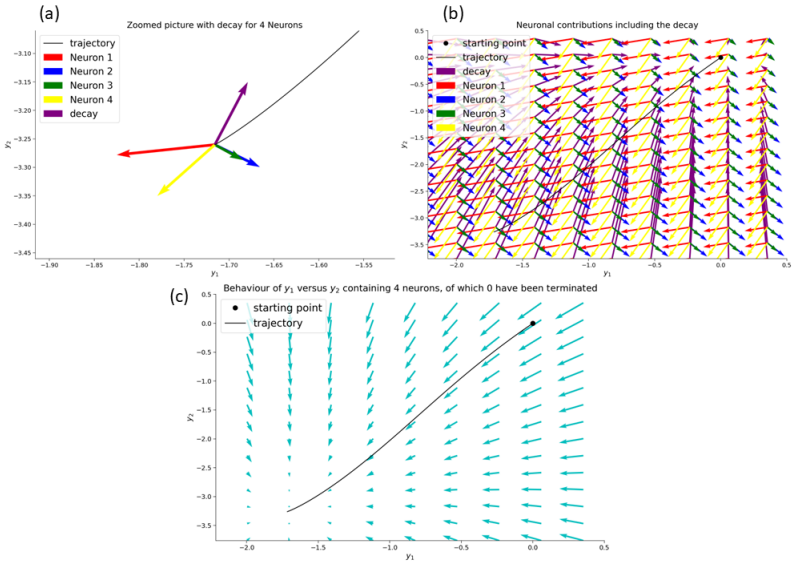


Figure 2.4: Phase plots of normal behaviour of a network, where no neurons have been silenced. (a) Zoomed in figure of the stabilization point, where the black line represents the behaviour of the network, the colored vectors represent a neuron and the purple vector represents the decay. (b) Phase plot of the normal behaviour if the vectors from (a) were drawn for every point. (c) The normal phase plot. With the black dot as starting point and where each blue vector represents the summation of the colored vectors including the decay, as seen in (a).

To understand how cell death affects the robustness of a recurrent neural network, there will be zoomed in on a single point in the phase plot (Fig 2.4a), to make a clear indication of how it would look like before this is extended to the entire phase plot. The vectors of each neuronal contribution, including the decay, will be plotted here in accordance with equation (2.9). What is illustrated is the black line indicating the behaviour of the network towards its stabilisation point and four colored vectors (red, blue, green and yellow), which represent the four neurons respectively. The purple vector indicates the decay, which always points towards the origin. For further plots the decay will be omitted, since this vector will decrease the readability of the phase plots. This will be visible when Fig. 2.4a is extended to include the full phase plot (Fig. 2.4b). The decay vector becomes larger, the further away it is from the origin, covering large parts of the phase plot. It is important to keep in mind that every location, where five vectors are drawn (four neuron contributions and one decay), adds up to one vector (eq. (2.9)), pointing towards the stabilization point as seen in figure 2.4c, where the normal phase plot of the network's behaviour is illustrated.

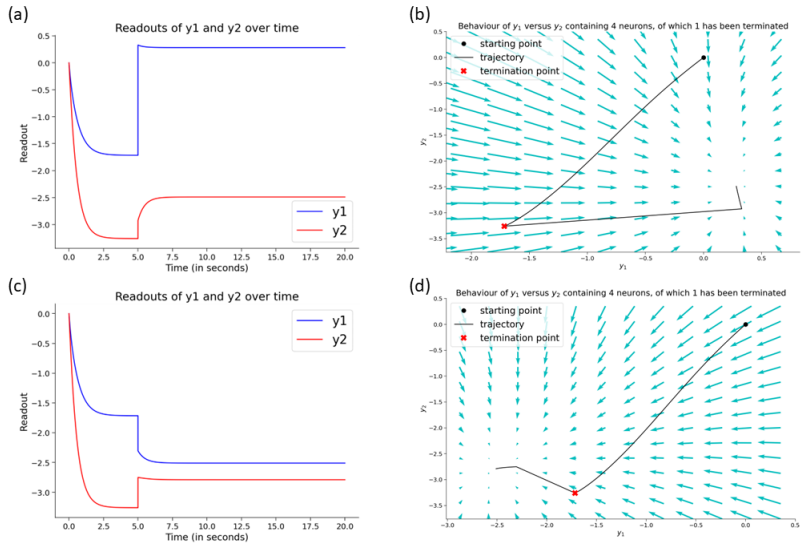


Figure 2.5: Readouts with their corresponding phase plots. (a) Readouts of y_1, y_2 in a network, where neuron one is silenced at $t = 5$s. (b) Phase plot of network, where neuron one is silenced. The black dot indicates the starting point, the black line indicates the behaviour of the network and the red cross indicates the termination point at $t = 5$s. (c) Readouts of y_1, y_2 in a network, where neuron three is silenced at $t = 5$s. (d) Phase plot of network, where neuron three is silenced. The red cross indicates the termination point at $t = 5$s.

Figure 2.3a, shows that the stabilization of the network occurs at about 2.5 seconds, indicated by the horizontal readouts of y_1 and y_2. Given that the stabilization occurs at about 2.5 seconds, the termination point will be set at $t_s = 5$ seconds, ensuring that the network has fully stabilized before a neuron is silenced. To fully comprehend what happens when a neuron dies, a randomly selected neuron will be chosen to silence and the resulting behaviour visualized.

The readouts of the resulting network (Fig. 2.5a,c) reveal that a shift happens around $t = 5$ seconds, after the network achieves its typical stabilization point at around $t = 2.5$ seconds. After this shift the readouts stabilize towards a new stabilization point. The corresponding phase plots (Fig. 2.5b,d) show that the normal stabilization point is reached, indicated by the red cross and that a big shift happens in different directions, after which another line stabilizes. The blue vectors indicate this new stabilization point as the behavior shifts to the right (Fig. 2.5b) or left (Figure 2.5d).

Although the figures (Fig. 2.5) demonstrate that the behavioural shifts

are distinct from one another, they are not sufficiently descriptive, as they do not explain where the difference comes from. To make the figures more descriptive, the previous figures (Fig. 2.5b,d) will be combined with the neuronal contribution (Fig. 2.4a), omitting the decay to make the figures easier to read.

The phase plots and related networks in Fig. (2.6) illustrate which neuron has been silenced. Through this new way of plotting, more information becomes available. In the phase plot of the normal behaviour (Fig. 2.6b), the four colored vectors are all visible. In contrast, it is easy to see that in the following figure (Fig. 2.6d) the red vector is gone, signifying that neuron one has been silenced. In the last figure (Fig. 2.6f) the missing vector is more difficult to notice, but the green vector is missing, signifying that neuron three has been silenced. Equation (2.9) further demonstrates that eliminating a single neuron corresponds to removing a specific direction since in both examples the vector that is missing corresponds to the neuron that has been silenced. Additionally, when both figures are put side by side, the missing vectors can be seen to be pointing in the opposite direction as the behavioral shift in the figure where they were silenced.

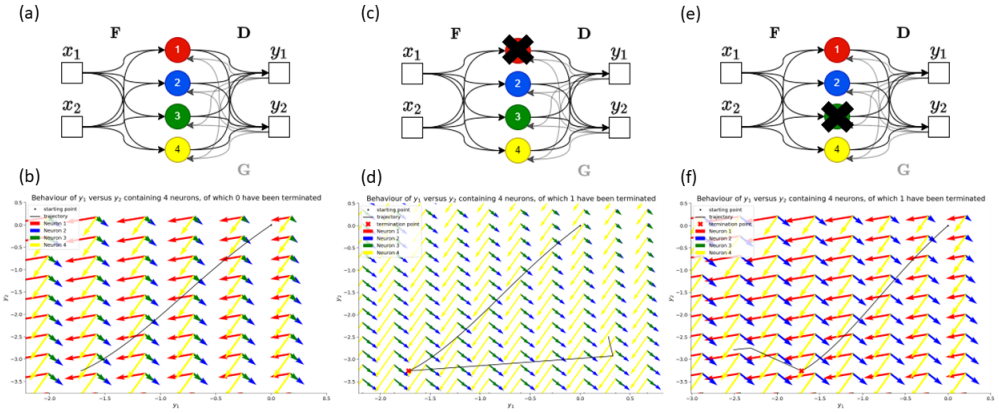


Figure 2.6: Network representations corresponding to their phase plots. (a) Representation of a normal network, with the neurons colored as their vector counterpart in the following phase plot. (b) Phase plot of the normal behaviour, where each colored line represents a neuron, the black dot is the starting point and the black line represents the behaviour of the network. The decay is excluded for visualization purposes. (c) Same network representation as (a), but now neuron one is silenced. (d) Phase plot of representation (c), where the red cross represents the point where neuron one was silenced. (e) Same network representation as (a), but now neuron three has been silenced. (f) Phase plot of representation (e), where the red cross represents the point where neuron three was silenced.

2.4.2 Big networks

The prior findings will be expanded to larger networks ($N = 100$) to deepen our intuitive grasp of the effects of cell death.

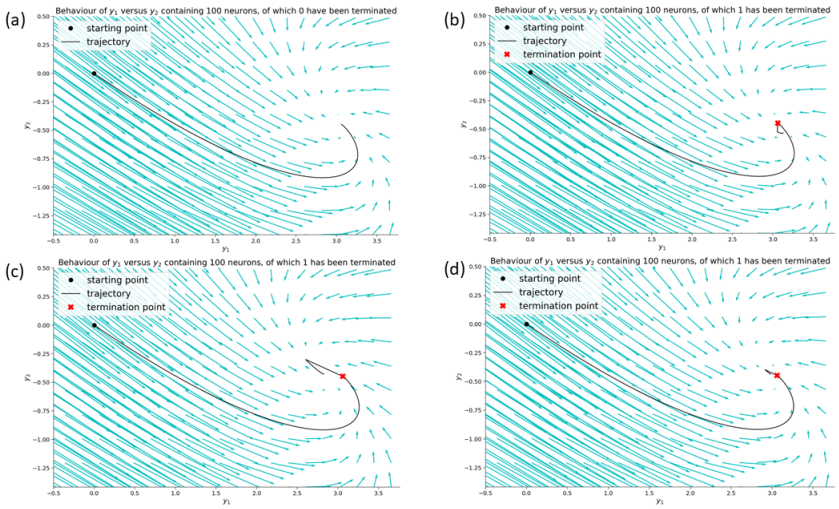


Figure 2.7: Phase plots of the same 100 neuron network, with different neurons silenced. Where the red cross indicates a neuron termination. (a) Phase plot of the network, where no neurons have been silenced. (b) Phase plot of the network, where neuron 1 has been silenced. (c) Phase plot of the network, where neuron 51 has been silenced. (d) Phase plot of the network, where neuron 100 has been silenced.

To visualize how the behavior of a network of 100 neurons is affected by neuron termination, the behaviour of networks, where a neuron has been silenced (Fig. 2.7b,c,d), is contrasted to the normal behaviour (Fig. 2.7a). Only one figure, compared to the others, has a reasonably large shift in behaviour (Fig. 2.7c), despite the fact that all of the networks exhibit a minor change in behaviour. These phase plots are insufficient to explain this disparity in behavioural shifts, as was noted in the previous section about small networks. In an effort to remedy this issue, the neuronal contributions will once more be shown.

Unfortunately, the same plots as before (Fig. 2.6) can no longer be utilized for a network of 100 neurons. Due to the amount of neurons in the network, these figures become too cluttered and difficult to interpret (Fig. 2.8a). Consequently only the stabilization point will be shown, with one set of 100 vectors. Only the vectors that belong to the neurons that have been silenced will be colored differently than the other vectors in order to clearly distinguish them from one another (Fig. 2.8b).

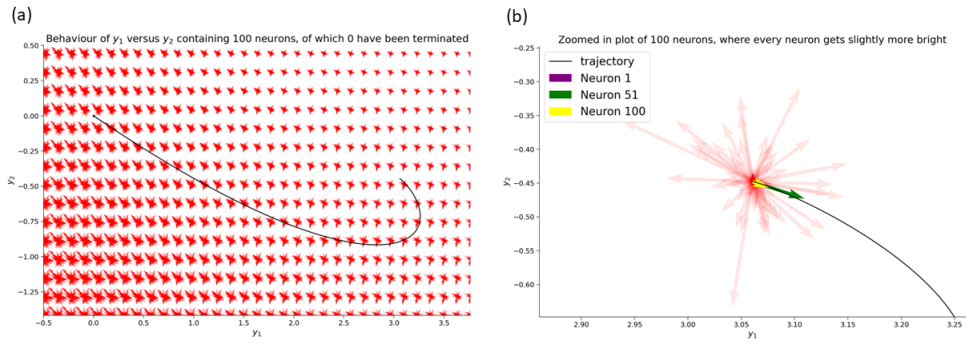


Figure 2.8: Neuronal contributions of a 100 neuron network. (a) Phase plot, where the contributions of every neuron have been plotted. This figure shows the need for a different strategy. (b) A zoomed in figure of the stabilization point of the 100 neuron network, where the silenced neurons (from the previous figure (Fig. 2.7)) have been colored, purple, green and yellow, respectively.

Even though the lengths of the colored vectors vary, it is obvious that the green vector is longer than the others. Using the acquired intuition from the small networks that the vectors point in opposite direction of the shift, the behavioural shifts can be related to the vectors. It then becomes clear that this green vector corresponds to the phase plot (Fig. 2.7c) where neuron 51 has been silenced. Once again supporting equation (2.9)'s assertion that the removal of a neuron results in the elimination of a vector.

Chapter 3

Discussion and conclusions

In summary, we created a new visualization technique to develop our intuitive understanding of where robustness in rate-based recurrent neural networks come from, expanding on the work that was previously done for spiking neural networks [10]. By evaluating our recurrent neural network as an ODE [13], we could use the differential equation in continuous time, allowing for smoother trajectories. By keeping the network low-rank [14], we were able to output directly in the output-space, allowing us to plot the behaviour of the networks in a two-dimensional plot.

We showed that the equation for the full state space can be written to include the specific neuron contributions, allowing us to visualize these contributions. Resulting in the conclusion that the silencing of a neuron can be seen as removing a directional vector. from the phase plots we were also able to draw the conclusion, that as the number of neurons increases, so do the number of vectors that point in various directions, resulting in a neural network that is more robust.

Because having more neurons does not always relate to being more robust, this does set a dangerous precedent. Since having more neurons increases the likelihood that the network would overfit, producing a few large directional vectors. If one of these neurons were to die, the network would experience a significant behavioural shift and might not be able to stabilize at the initial stabilization point .

3.1 Broader impact

This visualization technique can be used to help develop an intuitive understanding of cell death in rate-based recurrent neural networks and what is needed to make these networks more robust. Giving us the tools to show the individual neuron contributions, allows us to understand why a network has overfitted and how to prevent this.

3.2 Future work

Due to time constraints and the scope of the research, the testing of robustness in these networks could have been more extensive. In the small networks, we only tested the results for when we silenced one neuron at a time. And for the large networks we only showed the silencing of three neurons, where we again only silenced one neuron at a time.

But this also gives way to future research, where we focus more on what happens when groups of neurons are silenced at the same time, or groups of neurons that all point towards the same direction. Or where we illustrate the difference between silencing a group of the largest vectors and a group of the smallest vectors. Another important research could be, if a network becomes more robust when we initialize the network in a way that there will be a vector pointing in every direction.

Bibliography

[1] David GT Barrett, Sophie Denève, and Christian K. Machens. Optimal compensation for neuron loss. *eLife*, 12 2016.

[2] J H Morrison and P R Hof. Life and death of neurons in the aging brain. *Science*, 278(5337):412–419, October 1997.

[3] Olga V Volodina and https://pnojournal.wordpress.com/2022/07/01/volodina-3/. Formation of future teachers' worldview culture by means of foreign-language education. *P Sci Edu*, 57(3):126–159, July 2022.

[4] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[5] Joshua Shunk. Neuron-specific dropout: A deterministic regularization technique to prevent neural networks from overfitting & reduce dependence on large training samples, 2022.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. 25, 2012.

[7] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[8] David G Barrett, Sophie Denève, and Christian K Machens. Firing rate predictions in optimal balanced networks. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[9] Martin Boerlin, Christian K. Machens, and Sophie Denève. Predictive coding of dynamical variables in balanced spiking networks. *PLOS Computational Biology*, 9(11):1–16, 11 2013.

[10] Nuno Calaim, Florian Alexander Dehmelt, Pedro J. Gonçalves, and Christian K. Machens. Robust coding with spiking networks: a geometric perspective. *bioRxiv*, 2020.

[11] Allan Mancoo, Sander Keemink, and Christian K Machens. Understanding spiking networks through convex optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8824–8835. Curran Associates, Inc., 2020.

[12] Cynthia Rudin and Joanna Radin. Why are we using black box models in ai when we don't need to? a lesson from an explainable ai competition. *Harvard Data Science Review*, 1(2), 11 2019. https://hdsr.mitpress.mit.edu/pub/f9kuryi8.

[13] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. 31, 2018.

[14] Friedrich Schuessler, Francesca Mastrogiuseppe, Alexis Dubreuil, Srdjan Ostojic, and Omri Barak. The interplay between randomness and structure during learning in rnns. 33:13352–13362, 2020.

[15] Francesca Mastrogiuseppe and Srdjan Ostojic. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, 99(3):609–623.e29, 2018.