All line attractors are unstable, but some are more unstable than others

May 14, 2023

Abstract

Attractor networks play an essential role in computational neuroscience as a model for (perceptual) decision-making, memory and neural computation more generally. It is well known that some of these networks, for example continuous attractor networks which can represent continuous variables, are structurally unstable. This is a problem in biological neural networks because they are constantly undergoing perturbations caused by noise. We apply an existing result from dynamical systems theory to show that bounded attractors are stable in the following sense. All perturbations of bounded attractors result in systems where the attractor (and its topology) is maintained. This has as consequence that if there is a restorative learning signal there are no exploding gradients for any time length (for backpropagation through time). We contrast this to unbounded attractors that devolve into divergent systems under some perturbations which can lead to exploding gradients. We work out a simple example and show that all perturbations preserve the attractor and demonstrate the principle numerically in some systems relevant to neuroscience, namely the finite, ring and plane bump attractors.

+homeostasis

1 Introduction

Exploding gradients occur when the gradients in a neural network become very large during training. This can cause the weights to be updated with very large values, leading to numerical instability and poor performance. In recurrent neural networks (RNNs), this problem is particularly acute because the gradients can be multiplied many times as they are passed through the recurrent connections.

The issue with exploding gradients is that the weight updates become too large to be useful. When the gradients become very large, the weight updates can become so big that they overshoot the optimal weights, causing the network to diverge and perform poorly. In extreme cases, the weight updates can be so large that they cause the weights to overflow, resulting in numerical errors. Even worse, in case the gradients arise from some noise, the gradient descent algorithm can drift far away from the optimal weights throught the explosion in the gradients.

To prevent exploding gradients in RNNs, a common approach is to use gradient clipping, where the gradients are scaled down when they exceed a certain threshold. This ensures that the weight updates remain within a reasonable range and prevents numerical instability. Additionally, techniques like weight initialization, regularization, and adaptive learning rate methods can also be used to improve the stability of the training process and prevent the occurrence of exploding gradients.

1.1 ODEs

We consider

$$\dot{x} = f(x) \tag{1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}^n$ is a smooth vector field.

1.2 RNNs

[12]

[1]

Continuous attractor stores information on a continuous manifold such that the neural state can be maintained to represent an arbitrary continuous value for an arbitrary temporal scale [6, 9].

1.3 Gradients

The gradient signal is the partial derivative of the loss with respect to each parameter:

$$\frac{\partial L}{\partial w_k} = \lim_{\Delta w \to 0} \frac{L(w_k + \Delta w) - L(w_k)}{\Delta w}.$$

As the definition indicates, positive $\frac{\partial L}{\partial w_k}$ implies infinitesimal decrease in the parameter value w_k is followed by the incresae in the goodness of the output

measured by the loss L. Given the gradient of all adjustable parameters, we can adjust all of them proportional to the negative of the gradients to decrease the loss L in the steepest direction. For example, we can use *gradient flow* that defines a continuous dynamics of the weights:

$$\frac{\mathrm{d}w_k}{\mathrm{d}t} = -\frac{1}{\tau}\frac{\partial L}{\partial w_k}$$

simultaneously for all i parameters. The constant τ is the time constant for learning, and its reciprocal $\frac{1}{\tau}$ is called the learning rate. While this particular method of optimization uses the principle of gradient descent, the theory we develop does not depend on how the gradient descent is implemented or whether or not it is a supervised learning problem, since it describes the quality of the learning signal itself.

Back-Propagation Through Time (BPTT): gradients tend to either (1) blow up or (2) vanish the temporal evolution of the backpropagated error exponentially depends on the size of the weights

Exploding gradients, Case (1), may lead to oscillating weights.

1.3.1 Exploding

Continuous attractor dynamics, a previously known solutions to the EVGP, often suffer from the *fine-tuning problem*—small change in the parameters can drastically reduce the effectiveness.

An exact line attractor can be easily designed and implemented, for example, LDS with null eigenvalues and LSTM with no forgetting.

the realization of line attractors are fragile to parameter changes [9**?**].

Similar phenomenon has also been observed in GRU networks which simplifies the LSTM by removing the linear cell structure [**?**]. A notable exception is the ring attractor which can be implemented using bump attractor network architecture, also seen in the biological neural system. However, stable oscillations are much more robust to parameter perturbations.

1.3.2 Activation functions

ELU [**?**]

1.4 Continuous attractor supports persistent memory and sensitivity

Continuous attractors are characterized by a manifold such as a line, a ring, or a plane, where there is no flow, i.e., such that small perturbations away from the manifold asymptotically returns to the manifold. In the context of recurrent networks, the manifold of a continuous attractor is low-dimensional and embedded in a higher dimensional neural activity space. Since there is a continuum of stable equilibria where the neural activity does not change over time,

the observed autonomous dynamics of the system is similar to a point attractor system, i.e., it generally decays to a fixed state and maintains a constant activity over time. However, unlike point attractors, all states on the manifold exhibit similar behavior—perturbations on the manifold do not return to the initial state and hence are not forgotten.

As a conceptual tool in theoretical neuroscience, they are widely used when working memory of continuous values is needed [5]. In combination with input, continuous attractors are also called integrators that are hypothesized to be the underlying computation for the maintenance of eye positions, heading direction, self-location, target location, sensory evidence, working memory, decision variables, to name a few [10**?**]. A key signature of a continuous attractor is persistent neural activity that can be maintained at various levels during a memory or delay period [**?**]. In neuroscience, a typical implementation of a continuous attractor are bump attractor network models [7**? ?** , 8].

1.5 Bifurcations

In dynamical systems, when infinitely small change in the parameters cause a qualitative change in the dynamics due to changes in the topology of the dynamics, it is said that the system undergoes a bifurcation.

The asymptotic behavior of a recurrent neural network changes qualitatively at certain points in the parameter space, which are known as *bifurcation points*.

At bifurcation points, the output of a network can change discontinuously with the continuous change of parameters and therefore convergence of gradient descent algorithms is not guaranteed in such cases.

Furthermore, learning equations used for error gradient estimation can be unstable

If we randomly pick one point in the parameter space, it is very unlikely that it is a bifurcation point. However, it we change the parameters continuously by learning, there is a non-negligible chance that we encounter codimension-one bifurcations.

The parameter space of a recurrent network is divided into many regions with different qualitative structure of the state space, for example, the regions in which the state space has only one attractor point, one limit cycle, two attractor points, one attractor point and one limit cycle, and so on. Those qualitatively different regions are bordered by bifurcation points. Therefore, some types of bifurcations are inevitable steps in constructing a network with an interesting dynamical behavior. However, when the network goes through a bifurcation, either purposefully or by a mishap, gradient descent algorithms can have problems as shown below.

If a gradient descent algorithm with a fi

xed learning rate is used, a very large error gradient causes a long jump in the parameter space. It can even lead to a numerical overfow. Even if those are not the case, gradient descent does not work effciently when the size of gradient varies so much in different directions.

1.5.1 Codimension-one

In general, bifurcations occur when the vector
eld F satis
es some constraints.

The above cases can be seen on a k-1 dimensional surface in k dimensional parameter space and therefore called codimension-one bifurcations

1.6 Bifurcations of Recurrent Neural Networks

[**?**]

2 Stability concepts

[13]

2.1 Structural stability

In addition to the S-type noise, biological neural systems have constantly fluctuating synaptic weights [11]. In other words, there is noise in the recurrent network dynamics, which we call the *D-type noise*.

An important consequence of the presence of D-type noise is that the neural computation implemented by recurrent dynamics is constantly fluctuating. Therefore, the desirable properties of the dynamical system that require precise weight combinations are not stably achievable due to their unreliability.

The topological structures that are robust under D-type noise are called *structurally stable* – for example, a stable fixed point is structurally stable [**?**].

Unfortunately, continuous attractors are not structurally stable – small changes in the dynamical system can destroy continuous attractors, and as a consequence, the corresponding Lyapunov exponent(s) move away from zero. For example, in machine learning, vanilla RNNs are sometimes initialized at the continuous attractor regime with all zeros such that which avoids the asymptotic EVGP initially, but very quickly loses the continuous attractor after one gradient step. This is a well known problem in neuroscience, often referred to as the "fine tuning problem" of the continuous attractor [7, 8, 10]. There have been remedies to lesson the degradation, often focusing on keeping the short-term behavior close to the continuous attractor case [4**?** **?** **?**].

2.2 Persistence

Fenichel 1971 Theorem 1:

Theorem 1. *Let X be a C^r vector field on $\mathbb{R}^n$ with $r \geq 1$. Let $\overline{M} = M \cup \partial M$ be a C^r compact, connected manifold with boundary, properly embedded in $\mathbb{R}^n$ and overflowing invariant under X. Suppose $\nu(m) < 1$ and $\sigma(m) < \frac{1}{r}$ for all $m \in M$. Then for any C^r vector field Y in some C^1 neighborhood of X there is a manifold $\overline{M_Y}$ overflowing invariant under Y and C^r diffeomorphic to $\overline{M}$.*

Fenichel 1971 Theorem 1:

Proof sketch: Goal: construct invariant manifold as a zero section of a k-dimensional vector bundle N' transversal to $M_1 \coloneqq F^1(M)$

Construct a local coordinate system Introduce: $U_i = \sigma^{-1} D^i$ for disks with radius $i = 1, \ldots, 6$ T big number Take $u \in S$ in space of sections (of $N_\varepsilon | \cup_{i=1}^s U_i^3$) such that the graph of $u|_{\bar{M}}$ is an overflowing invariant manifold under Y Define $G : S_\delta \to S_\delta = \{u \in S | \operatorname{Lip} u \leq \delta\}$ s.t. $Gu = u$ iff graph $u \subset F_Y^T(graphu)$ and u is unique fixed point of G and $\forall t > 0$ graph $u \subset F_Y^t(\text{graph } u)$ Construct G through F_Y^T

Proposition 4 ... Prop 4 implies that F_Y^T induces a map $G : S_\delta \to S$ Properties of G Prop 5 $G : S_\delta \to S_\delta$ Prop 6 G is a contraction on S_δ Corollary $u \in S_\delta$ is unique

2.3 Initialization

When we have an a priori knowledge about the required qualitative structure of the state space, for example, the number of attractors, it is possible to pre-program it by the initial connection weights. For example, if we want to train a network to have multiple limit cycle attractors, it is helpful to set the initial connection weights to have multiple attractor points.

2.4 Task

Copy memory The copy-memory task was first introduced in [42]. It requires a model to retain an input sequence and then reproduce it as the output following a k-step delay. The input consists of a sequence of 10 tokens drawn at random from an alphabet of size 8, followed by k repetitions of a 'blank' and a single 'start' token, and 9 'blank' tokens. The target output is a sequence of k + 10 'blank' tokens followed by the original 10 element-long sequence presented at the beginning of the input.

2.5 Neural computation

2.5.1 Head direction

[14]
[**?**]
[2]

3 Perfect integrator

The MSE on the whole trial is

$$\frac{1}{T} \int_0^T \left(h(t) - \int_0^t x(\tau) d\tau \right)^2 dt,$$

with

$$x(t) = \sum_{i=1}^{\#R} \delta(t - R(i)) - \sum_{i=1}^{\#L} \delta(t - L(i))$$

the input at time t where $\#R$ and $\#L$ are the number of right and left clicks up to time t, respectively and $R(i)$ and $L(i)$ are the timings of the ith right and left clicks respectively.

Assuming the parameters of the perfect one-dimensional integrator (with $\theta = 1$)

$$\frac{dh}{dt} = -\lambda h(t) + x(t) \tag{2}$$

the MSE on the whole trial is zero.

The solution to (2) is

$$h(t) = \sum_{i=1}^{\#R} e^{\lambda(t-R(i))} - \sum_{i=1}^{\#L} e^{\lambda(t-L(i))}, \tag{3}$$

with $\lambda = -(\theta - 1)$.

The integral of the inputs is

$$\int_0^t x(\tau)d\tau = h(t) = \sum_{i=1}^{\#R} 1 - \sum_{i=1}^{\#L} 1$$

So then the MSE is

$$\frac{1}{T}\int_0^T \left(\sum_{i=1}^{\#R} (e^{\lambda(t-R(i))} - 1) - \sum_{i=1}^{\#L} (e^{\lambda(t-L(i))} - 1) \right)^2 dt$$

$$\frac{1}{T}\int_0^T \left(\sum_{i=1}^{\#R+\#L} (-1)^{p(i)} (e^{\lambda(t-R(i))} - 1) \right)^2 dt$$

$$\frac{1}{T}\int_0^T \left(\sum_{i=1}^{\#R+\#L} (-1)^{p(i)} (e^{\lambda(t-R(i))}) - \sum_{i=1}^{\#R+\#L} (-1)^{p(i)} \right)^2 dt$$

3.1 Single click trial

On a trial with a single click with the click time $t = C(1)$ with $C \in \{L, R\}$ we have

$$\frac{1}{T - C(1)} \int_{C(1)}^{T} \left(e^{\lambda(t-C(1))} - 1 \right)^2 dt$$
$$= \frac{3 - 4e^{\lambda(T-C(1))} + e^{2\lambda(T-C(1))} + 2\lambda(T - C(1))}{2\lambda(T - C(1))}$$

In that case

$$\frac{\partial}{\partial\lambda}\frac{3-4e^{\lambda(T-C(1))}+e^{2\lambda(T-C(1))}+2\lambda(T-C(1))}{2\lambda(T-C(1))}$$
$$=\frac{4(T-C(1))e^{\lambda(T-C(1))}+2(T-C(1))e^{2\lambda(T-C(1))}}{2\lambda(T-C(1))}-\frac{3+4e^{\lambda(T-C(1))}+e^{2\lambda(T-C(1))}}{\lambda^2}$$

Which causes exploding gradients.

3.2 Two click trial

The integral from first click at $t = C(1)$ to the second click at $t = C(2)$ with $C \in \{L, R\}$ is

$$\frac{1}{C(2)-C(1)}\int_{C(1)}^{C(2)}\left(e^{\lambda(t-C(1))}-1\right)^2 dt$$
$$=\frac{1}{C(2)-C(1)}\int_{0}^{C(2)-C(1)}\left(e^{\lambda t}-1\right)^2 dt$$
$$=\frac{3-4e^{\lambda(C(2)-C(1))}+e^{2\lambda(C(2)-C(1))}+2\lambda(C(2)-C(1))}{2\lambda(C(2)-C(1))}$$

If the direction of the second click is the same as the first, the integral from the second up to the third click is

$$\frac{1}{C(3)-C(2)}\int_{0}^{C(3)-C(2)}\left(e^{\lambda(t-C(1)+C(2))}+e^{\lambda t}-2\right)^2 dt$$

If the direction of the second click is the opposite as the first, the integral from the second up to the third click is

$$\frac{1}{C(3)-C(2)}\int_{0}^{C(3)-C(2)}\left(e^{\lambda(t-C(1)+C(2))}-e^{\lambda t}\right)^2 dt$$
$$=\frac{1}{C(3)-C(2)}\left[\frac{e^{-2\lambda}(-1+e^{2S\lambda})}{2\lambda}-\frac{2e^{-\lambda}(-1+e^{S(1+\lambda)})}{1+\lambda}+e^S\sinh(S)\right]$$

with $S = C(3) - C(2)$

A lower bound can be found by evaluating the case that at each consecutive click is opposite and we take the limit $C(i+1)-C(i) \to 0$, in which case MSE=0.

4 Noisy gradient descent

We now look at a perturbation of the parameter of the linear integrator around $\theta = 1$ or equivalently $\lambda = 0$.

$$\frac{\partial}{\partial\lambda}\frac{1}{T}\int_0^T\left(\sum_{i=1}^{\#R}(e^{\lambda(t-R(i))}-1)-\sum_{i=1}^{\#L}(e^{\lambda(t-L(i))}-1)\right)^2 dt$$
$$=\frac{2}{T}\int_0^T\left[\sum_{i=1}^{\#R}(t-R(i))e^{\lambda(t-R(i))}-\sum_{i=1}^{\#L}(t-L(i))e^{\lambda(t-L(i))}\right]\left[\sum_{i=1}^{\#R}(e^{\lambda(t-R(i))}-1)-\sum_{i=1}^{\#L}(e^{\lambda(t-L(i))}-1)\right]dt$$

If $\lambda > 0$, i.e. if $\theta > 1$, if the last time point $t = R(i)$ of the stimulus has a right click (and no left click) we get a contribution of

$$\frac{2}{T}\int_{R(i)}^{T}(t-R(i))e^{\lambda(t-R(i))}(e^{\lambda(t-R(i))}-1)dt$$
$$=\frac{2}{T}\int_0^{T-R(i)}te^{\lambda t}(e^{\lambda t}-1)dt$$

5 Another implementation of a perfect integrator

5.1 Parameters of the integrator

5.1.1 Input

Parameter that determines step size along line attractor $\delta << 1$. The size determines the maximum number of clicks as the difference between the two channels.

This pushes the input along the line "attractor" in two opposite directions, see below.

5.1.2 Recurrent dynamics

$$\dot{x} = \mathrm{ReLU}(Wx + b) - x \tag{4}$$

Recurrent

$$W = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} \tag{5}$$

Bias:

$$b = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{6}$$

5.1.3 Output

5.1.4 Workings

Line attractor

$$\{(x, 1-x)|x \in [0,1]\}$$

5.1.5 Parameters of the full network part

5.1.6 Learning from this initialization

5.2 Analysis background

5.2.1 Lyapunov

The sensitivity can be expressed through the linearization of the ordinary differential equation using the Jacobian matrix of the dynamics with respect to the neural state vector:

6 Accumulator systems

6.1 Line attractor

We consider perturbations of the form

$$W' = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + \epsilon \tag{7}$$

with

$$\epsilon = \begin{pmatrix} \epsilon_{11} & \epsilon_{12} \\ \epsilon_{21} & \epsilon_{22} \end{pmatrix} \tag{8}$$

The eigenvalues are computed as

$$\begin{aligned} \det[W' - (1+\lambda)\mathbb{I}] &= (\epsilon_{11} - 1 - \lambda)(\epsilon_{22} - 1 - \lambda) - (\epsilon_{12} + 1)(\epsilon_{21} + 1) \\ &= \lambda^2 - (2 + \epsilon_{11} + \epsilon_{22})\lambda - \epsilon_{11} - \epsilon_{22} + \epsilon_{11}\epsilon_{22} - \epsilon_{12} - \epsilon_{21} - \epsilon_{12}\epsilon_{21} \end{aligned}$$

Let $u = -(2 + \epsilon_{11} + \epsilon_{22})$ and $v = -\epsilon_{11} - \epsilon_{22} + \epsilon_{11}\epsilon_{22} - \epsilon_{12} - \epsilon_{21} - \epsilon_{12}\epsilon_{21}$

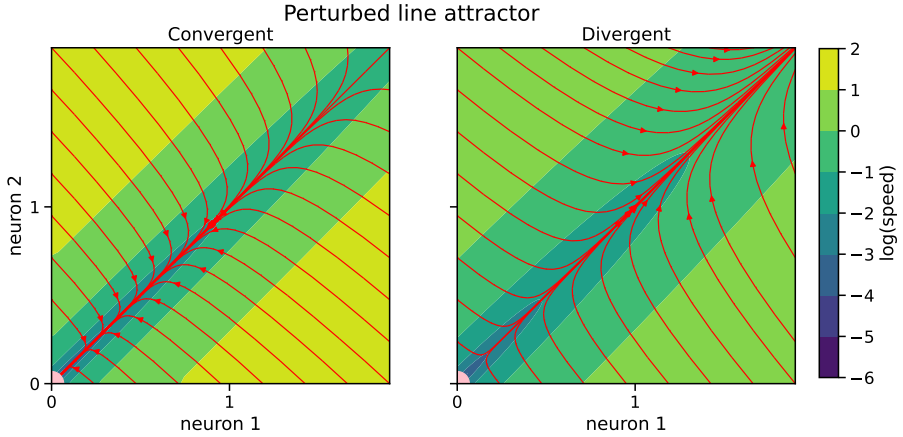

Figure 1: All types of perturbations for the bounded line attractor.

There are only two types of invariant set for the perturbations of the line attractor. Both have as invariant set a fixed point at the origin. What distinguishes them is that one type of perturbations lead to this fixed point being stable while the other one makes it unstable, see Figure 1.

6.2 Bounded line attractor

6.2.1 Definition

7 Lyapunov spectrum

We investigate how perturbations to the bounded line affect the Lyapunov spectrum. The Jacobian

$$J = -\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \tag{9}$$

We apply the perturbation

$$W' = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} + \epsilon \tag{10}$$

with

$$\epsilon = \begin{pmatrix} \epsilon_{11} & \epsilon_{12} \\ \epsilon_{21} & \epsilon_{22} \end{pmatrix} \tag{11}$$

The eigenvalues are computed as

$$\begin{aligned} \det[W' - (1+\lambda)\mathbb{I}] &= (\epsilon_{11} - 1 - \lambda)(\epsilon_{22} - 1 - \lambda) - (\epsilon_{12} - 1)(\epsilon_{21} - 1) \\ &= \lambda^2 - (2 + \epsilon_{11} + \epsilon_{22})\lambda - \epsilon_{11} - \epsilon_{22} + \epsilon_{11}\epsilon_{22} + \epsilon_{12} + \epsilon_{21} - \epsilon_{12}\epsilon_{21} \end{aligned}$$

Let $u = -(2 + \epsilon_{11} + \epsilon_{22})$ and $v = -\epsilon_{11} - \epsilon_{22} + \epsilon_{11}\epsilon_{22} + \epsilon_{12} + \epsilon_{21} - \epsilon_{12}\epsilon_{21}$

$$\lambda = \frac{-u \pm \sqrt{u^2 - 4v}}{2} \tag{12}$$

Case 1: $\mathrm{Re}(\sqrt{u^2 - 4v}) < u$, then $\lambda_{1,2} < 0$
Case 2: $\mathrm{Re}(\sqrt{u^2 - 4v}) > u$, then $\lambda_1 < 0$ and $\lambda_2 > 0$
Case 3: $v = 0$, then $\lambda = \frac{1}{2}(-u \pm u)$, i.e., $\lambda_1 = 0$ and $\lambda_2 = -u$

$$\epsilon_{11} = -\epsilon_{22} + \epsilon_{11}\epsilon_{22} + \epsilon_{12} + \epsilon_{21} - \epsilon_{12}\epsilon_{21} \tag{13}$$

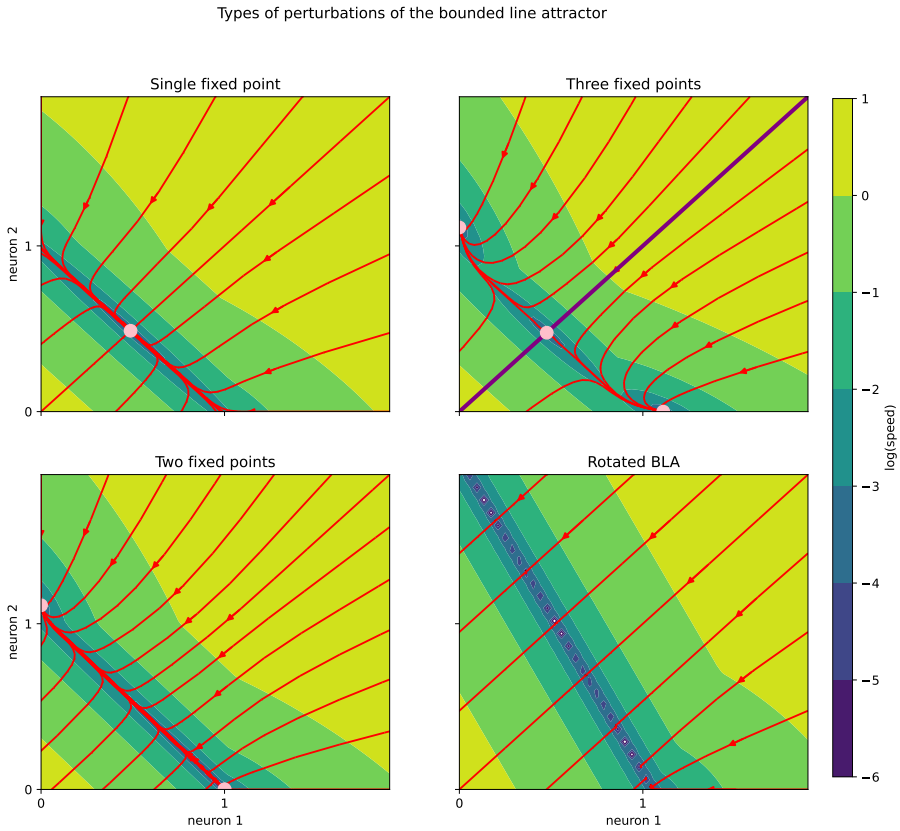

Figure 2: All types of perturbations for the line attractor.

We give some examples of the different types of perturbations to the bounded line attractor. The first type is when the invariant set is composed of a single fixed point, for example for the perturbation:

$$\epsilon = \frac{1}{10}\begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix} \tag{14}$$

See Figure 2, left upper.

The second type is when the invariant set is composed of three fixed points:

$$\epsilon = \frac{1}{10}\begin{pmatrix} 1 & -2 \\ -2 & 1 \end{pmatrix} \tag{15}$$

The third type is when the invariant set is composed of two fixed points, both with partial support.

$$b' = \frac{1}{10}\begin{pmatrix} 1 & -1 \end{pmatrix} \tag{16}$$

The fourth and final type is when the line attractor is maintained but rotated:

$$\epsilon = \frac{1}{20}\begin{pmatrix} 1 & 10 \\ 10 & 1 \end{pmatrix} \tag{17}$$

Theorem 2. *All perturbations of the bounded line attractor are of the types as listed above.*

Proof. We enumerate all possibilities for the dynamics of a ReLU activation network with two units. First of all, note that there can be no limit cycle or chaotic orbits.

Now, we look at the different possible systems with fixed points. There can be at most three fixed points [**?** , Corollary 5.3]. There has to be at least one fixed point, because the bias is non-zero.

General form (example):

$$\epsilon = \frac{1}{10}\begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix} \tag{18}$$

One fixed point with full support:

In this case we can assume W to be full rank.

$$\dot{x} = \mathrm{ReLU}\left[\begin{pmatrix} \epsilon_{11} & \epsilon_{12} \\ \epsilon_{21} & \epsilon_{22} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right] - \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0$$

Note that $x > 0$ iff $z_1 \coloneqq \epsilon_{11}x_1 + (\epsilon_{12} - 1)x_2 - 1 > 0$. Similarly for $x_2 > 0$.

So for a fixed point with full support, we have

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = A^{-1}\begin{pmatrix} -1 \\ -1 \end{pmatrix} \tag{19}$$

with

$$A \coloneqq \begin{pmatrix} \epsilon_{11} - 1 & \epsilon_{12} - 1 \\ \epsilon_{21} - 1 & \epsilon_{22} - 1 \end{pmatrix}.$$

Note that it is not possible that $x_1 = 0 = x_2$.

Now define

$$B \coloneqq A^{-1} = \frac{1}{\det A}\begin{pmatrix} \epsilon_{22} - 1 & 1 - \epsilon_{12} \\ 1 - \epsilon_{21} & \epsilon_{11} - 1 \end{pmatrix}$$

with

$$\det A = \epsilon_{11}\epsilon_{22} - \epsilon_{11} - \epsilon_{22} - \epsilon_{12}\epsilon_{21} + \epsilon_{12} + \epsilon_{21}.$$

Hence, we have that $x_1, x_2 > 0$ if $B_{11} + B_{12} > 0$, $B_{21} + B_{22} > 0$ and $\det A > 0$ and $B_{11} + B_{12} < 0$, $B_{21} + B_{22} < 0$ and $\det A < 0$.

This can be satisfied in two ways, If $\det A > 0$, this is satisfied if $\epsilon_{22} > \epsilon_{12}$ and $\epsilon_{11} > \epsilon_{21}$, while if $\det A > 0$, this is satisfied if $\epsilon_{22} < \epsilon_{12}$ and $\epsilon_{11} < \epsilon_{21}$. This gives condition 1.

Finally, we investigate the condition that specify that there are fixed points with partial support. If $x_1 = 0$ then $(\epsilon_{22} - 1)x_2 + 1 = 0$ and $z_1 < 0$. From the

equality, we get that $x_2 = \frac{1}{1-\epsilon_{22}}$. From the inequality, we get $(\epsilon_{12}-1)x_2+1 \geq 0$, i.e. $\frac{1}{1-\epsilon_{12}} \geq x_2$. Hence,

$$\frac{1}{1-\epsilon_{12}} \geq \frac{1}{1-\epsilon_{22}}$$

and thus

$$\epsilon_{22} \leq \epsilon_{12}. \tag{20}$$

Similiarly to have a fixed point x^* such that $x_2^* = 0$, we must have that

$$\epsilon_{11} \leq \epsilon_{21}. \tag{21}$$

Equation 20 and 21 together form condition 2.

If condition 1 is violated, but condition 2 is satisfied, there are two fixed points on the boundary of the admissible quadrant.

If condition 1 is violated, and only one of the subconditions of condition 2 is satisfied, there is a single fixed point on one of the axes.

If condition 2 is violated, there are three fixed points.

We now look at the possibility of the line attractor being preserved. This is the case if $v = 0$. It is not possible to have a line attractor with a fixed point off it for as there cannot be disjoint fixed points that are linearly dependent [**?** , Lemma 5.2]. □

7.1 Structure of the parameter space

We check what proportion of the bifurcation parameter space is constituted with bifurcations of the type that result in three fixed points.

The conditions are

$$\begin{aligned} 0 &< \epsilon_{11}\epsilon_{22} - \epsilon_{11} - \epsilon_{22} - \epsilon_{12}\epsilon_{21} - \epsilon_{12} - \epsilon_{21}, \\ \epsilon_{22} &\leq \epsilon_{12}, \\ \epsilon_{11} &\leq \epsilon_{21}. \end{aligned}$$

We show that if

$$\begin{aligned} \epsilon_{22} &\leq \epsilon_{12}, \\ \epsilon_{11} &\leq \epsilon_{21}. \end{aligned}$$

then always

$$0 < \epsilon_{11}\epsilon_{22} - \epsilon_{11} - \epsilon_{22} - \epsilon_{12}\epsilon_{21} - \epsilon_{12} - \epsilon_{21}.$$

The only nontrivial cases are

8 Bounded plane attractor

$$W = \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} \tag{22}$$

and

$$b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{23}$$

$$\epsilon = \begin{pmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{pmatrix} \tag{24}$$

8.1 Limit cycle

[**?** , Theorem 2.4] G be an oriented graph with no sinks If $\epsilon < \frac{\delta}{1+\delta}$, then the network has bounded activity and no stable fixed points.

8.1.1 With one fixed point

$$\epsilon = \begin{pmatrix} 0 & -\delta_1 & \delta_2 \\ \delta_2 & 0 & -\delta_1 \\ -\delta_1 & \delta_2 & 0 \end{pmatrix} \tag{25}$$

with $\delta_1 = \frac{1}{100}$ and $\delta_2 = \frac{1}{1000}$.

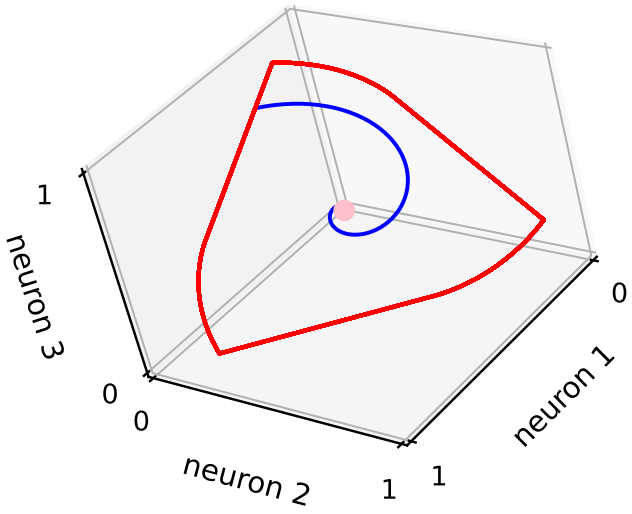

Figure 3: Limit cycle emerging from a perturbation of the bounded plane attractor.

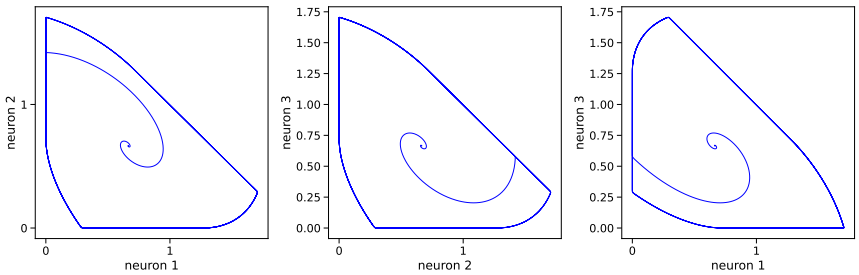

Figure 4: Limit cycle emerging from a perturbation of the bounded plane attractor projected.

8.1.2 Existence of limit cycle

Remark 1. Difficult to prove explicitly, for a proof see [3].

The set $[0,1]^3$ is a globally attracting set [**?** , Lemma 2.1].
The only fixed point is unstable. Conditions: Existence

Uniqueness
Stability

8.1.3 With four fixed points

$$\epsilon = \begin{pmatrix} 0 & -\delta_1 & 0 \\ 0 & 0 & -\delta_1 \\ -\delta_1 & 0 & 0 \end{pmatrix} \tag{26}$$

with $\delta_1 = \frac{1}{100}$.

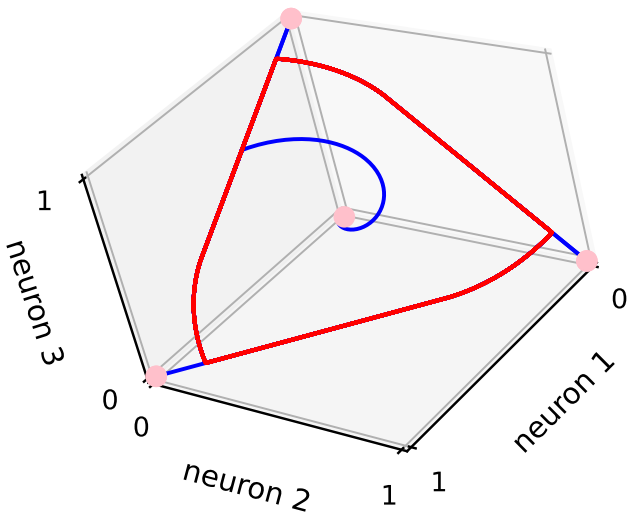

Figure 5: Limit cycle emerging from a perturbation of the bounded plane attractor.

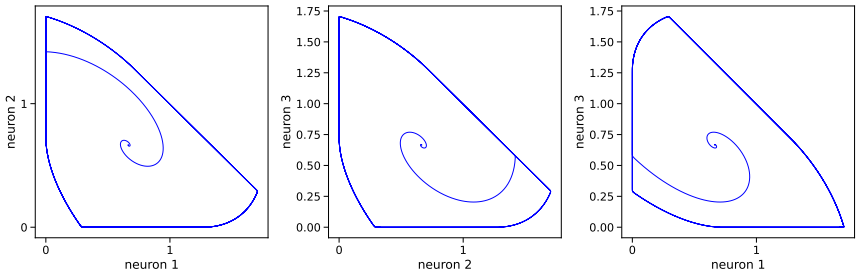

Figure 6: Limit cycle emerging from a perturbation of the bounded plane attractor projected.

8.2 Harmonic oscillator

$$\epsilon = \begin{pmatrix} 0 & -\delta_1 & \delta_1 \\ \delta_1 & 0 & -\delta_1 \\ -\delta_1 & \delta_1 & 0 \end{pmatrix} \tag{27}$$

8.3 Bounded line attractor

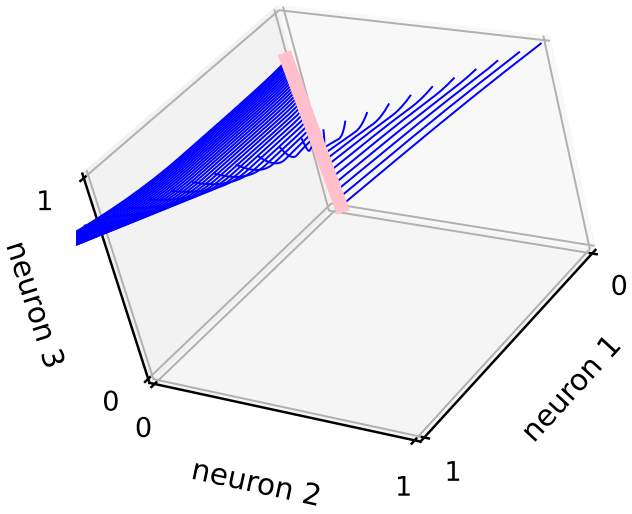

Figure 7: Bounded line attractor emerging from a perturbation of the bounded plane attractor.

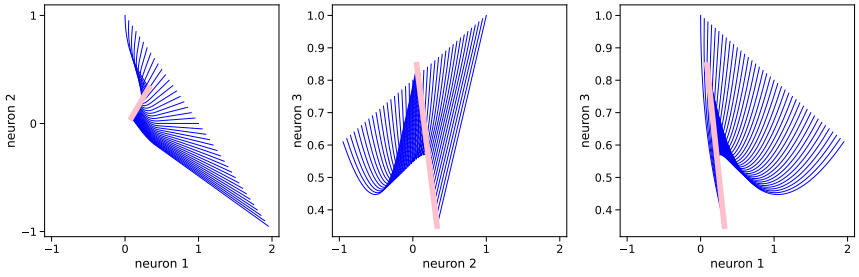

Figure 8: Bounded line attractor emerging from a perturbation of the bounded plane attractor projected.

References

[1] *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm*

Dependencies. IEEE.

[2] Zaki Ajabi, Alexandra T. Keinath, Xue-Xin Wei, and Mark P. Brandon. Population dynamics of head-direction neurons during drift and reorientation.

[3] Andrea Bel, Romina Cobiaga, Walter Reartes, and Horacio G Rotstein. Periodic solutions in threshold-linear networks and their entrainment. *SIAM Journal on Applied Dynamical Systems*, 20(3):1177–1208, 2021.

[4] Martin Boerlin, Christian K Machens, and Sophie Denève. Predictive coding of dynamical variables in balanced spiking networks. *PLoS computational biology*, 9(11):e1003258, 2013.

[5] Peter Dayan and Laurence F Abbott. Theoretical neuroscience: Computational and mathematical modeling of neural systems. 2001.

[6] Christian K Machens, Ranulfo Romo, and Carlos D Brody. Flexible control of mutual inhibition: a neural model of two-interval discrimination. *Science*, 307(5712):1121–1124, 2005.

[7] Marcella Noorman, Brad K Hulse, Vivek Jayaraman, Sandro Romani, and Ann M Hermundstad. Accurate angular integration with only a handful of neurons.

[8] Alfonso Renart, Pengcheng Song, and Xiao-Jing Wang. Robust Spatial Working Memory through Homeostatic Synaptic Scaling in Heterogeneous Cortical Networks. 38(3):473–485.

[9] H. Sebastian Seung. Continuous attractors and oculomotor control. 11(7-8):1253–1258.

[10] H. S. Seung. How the brain keeps the eyes still. 93(23):13339–13344.

[11] Genki Shimizu, Kensuke Yoshida, Haruo Kasai, and Taro Toyoizumi. Computational roles of intrinsic synaptic dynamics. 70:34–42.

[12] Ryan Vogt, Maximilian Puelma Touzel, Eli Shlizerman, and Guillaume Lajoie. On Lyapunov Exponents for RNNs: Understanding Information Propagation Using Dynamical Systems Tools. 8:818799.

[13] Huaguang Zhang, Zhanshan Wang, and Derong Liu. A Comprehensive Review of Stability Analysis of Continuous-Time Recurrent Neural Networks. 25(7):1229–1262.

[14] K Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: A theory. 16(6):2112–2126.