IFT 6085 - Guest Lecture
(Dynamical Systems and Stability Theory)

This version of the notes has not yet been thoroughly checked. Please report any bugs to the scribes or instructor.

Scribes
Winter 2019: [Anirudh Goyal, Alex Lamb]

Instructor: Ioannis Mitliagkas

1 Summary

Dynamical systems theory is concerned with the analysis of systems that change and evolve over time. A major theme in this line of research is the study of "stability" and "chaos". Intuitively, we can think about whether a system has the tendency to evolve in a consistent way when slightly perturbed, or whether small perturbations will lead to large variations (as an example, consider an inverted pendulum). In this lecture we discuss the following subjects:

- The basics of dynamical systems
- The basics of stability theory
- Stability and eigenvalues of jacobian
- Lyapunov stability and Lyapunov exponentials
- How can stability be used to analyze RNNs?

2 Background on Dynamical Systems

2.1 Flows

A dynamical system can be written as a triplet (X, T, ϕ). In this, X refers to a state-space, T refers to a time space, and ϕ refers to the flow itself. A flow is a mapping from one state to another state which is time dependent $(x, t) \rightarrow \phi_t(x)$, where $x \in X$ and $t \in T$.

These flows can be defined in either continuous or discrete time. A continuous time flow is a function over x which also depends on the time. A discrete time flow is a function just over x: $\phi(x)$ which is iterated for a certain number of time intervals. An example of a continuous time flow might be the flow of water or electricity through a physical system. An example of a discrete flow might be a computer's CPU state (assuming it's self contained), as the clock fires on discrete intervals.

2.2 Trajectories

A trajectory of a flow ϕ is a set defined by the states reachable from a given initial condition $x_0 \in X$. We can write a continuous flow as $O(x_0) = \{\phi_t(x_0) \mid t \in \mathbb{R}\}$. If the flow is irreversible, we restrict ourselves to only positive numbers of steps: $t \in \mathbb{R}^+$. For a discrete flow, we can write the flow as the discrete step iterated $t \in \mathbb{N}$ times: $O(x_0) = \{\phi^t(x_0) \mid t \in R\}$. Note that these notations seem similar, but in the discrete case ϕ^t refers to function iteration but in the continuous case ϕ_t, t is an input to the function ϕ. To express a reversible discrete flow, we would need to define an inverse of the flow ϕ.

2.3 Flows in Practice

We have briefly characterized the notation for flows. However, where can we find or understand flows in practice? One potential technique is to first describe how the system changes on each step, for example with ordinary differential equations. We could then solve these to produce either a numerical or symbolic understanding of the system's flow.
If we have access to a vector field f, describing how the system changes with each step, then we can write:

Theorem 1.

$$\frac{d}{dt}\phi_t(x) = f(\phi_t(x)) \tag{1}$$

Given such a vector field f, the existence of a flow is guaranteed under a differentiability condition. In a simple case, we can think of the vector field f as the derivative describing how the system evolves, and the flow as the integral over that vector field.
One simple example of this is Newton's cooling law, where T is the ambient temperature, x is the temperature of the object under consideration, and k is a cooling constant.

Theorem 2 (Newton's Cooling Law)**.**

$$\frac{dx}{dt} = -k(x - T) \tag{2}$$

By solving this differential equation, we can produce a continuous flow describing the cooling of a system at any particular point in time.
In general it is not possible to find an exact expression characterizing the flow. In general (non-linear dynamics) it is difficult or impossible to characterize the flow exactly. This is a major motivation for dynamical systems theory, which aims to gain an understanding of the characteristics of these flows without necessarily being able to solve them.

3 Stability Theory Basics

The goal of stability theory is to understand how a dynamical system will tend to evolve in the long-run, without necessarily understanding its exact properties. For example, if we're rolling a ball down a hill, its exact trajectory could be very noisy and complicated - but we can still say that eventually it will get to the bottom of the hill and stop. This point where the system stops changing is called a "fixed point", and knowing that a system has these fixed points gives us some understanding of how the system works without necessarily understanding all of its details.

3.1 Stability and Fixed Points

We can write that $x*$ is a fixed point for a dynamical system F if $x^* = F(x^*)$. For linear systems, an eigenvector with an eigenvalue of 1.0 would be such a fixed point.
We define stability around two sets U and V, with U being a strict subset of V. We define stability as $x_0 \in U, t \geq 0 \rightarrow x_t \in V$. Thus we state that if we start with an inital condition anywhere in U, if we run for any number of timesteps, we will always stay within the set V.
Asymptotic stability refers to a stronger claim that as the number of iterations increases, the system will approach a fixed point if it starts in a set U. This set U is referred to as a basin of attraction for the fixed point.

Theorem 3 (Basin of Attraction of x^*)**.**

$$\forall x_0 \in U \tag{3}$$

$$\lim_{t \to \infty} \|x_t - x^*\| = 0 \tag{4}$$

3.2 Determining a Fixed Point's Stability

We assume that the initial condition is close to the optimal state but off by a small amount y_0. We refer to the error on step T as y_T.

Theorem 4 (Stability and Jacobian Bound)**.**

$$\begin{aligned} x_0 &= x^* + y_0 \\ x_{t+1} &= F(x_t) \\ x_{t+1} &= F(x^*) + J(x^*)(x_t - x^*) + O(\|x_t - x^*\|) \\ x^* + y_{t+1} &= x^* + J(x^*)y_t + O(\|y_t\|) \\ y_{t+1} &= J(x^*)y_t + O(\|y_t\|) \\ y_{t+1} &\approx J(x*)y_t \\ y_t &\approx tJ(x*)y_0 \end{aligned}$$

Thus up to a first-order approximation on F, the amount of error is approximately product of the jacobian of the flow J at the fixed point multiplied by the number of time steps.

4 Linear Stability and Eigenvalues of Jacobian

We can see that the error term decays asymptotically if the eigenvalues of the jacobian all have magnitude smaller than 1.0. A saddle point occurs when some eigenvalues are smaller and some eigenvalues are larger. It indicates that some directions shrink asymptotically while others "explode".

5 Lyapunov Stability

How can we analyze stability of a trajectory, as opposed to just measuring how close we will stay to a fixed point?

Theorem 5 (Lyapunov Stability)**.**

$$\begin{aligned} x_t = F^t(x_0) = F_0(F_0(...(F_0(x_0)))) \\ \epsilon > 0 \\ \|v\| = 1 \\ \left\|F^t(x_0 + v\epsilon) - F^t(x_0)\right\| \approx \epsilon e^{\lambda t} \\ \lim_{\epsilon \longrightarrow 0} \rightarrow \lambda \approx \frac{1}{t} log(\left\|J^t(x_0)v\right\|) \end{aligned}$$

These λ are referred to as lyapunov exponents, and they behave similarly to the eigenvalues from the fixed point analysis, in that all $\lambda < 0$ indicates stability and any $\lambda > 0$ indicates instability.
Note that this result is very similar to what was achieved for stability around a fixed point: the amount of error from a trajectory is closely related to the product of the jacobian across the steps. And likewise, all of the lyapunov exponentials being negative indicates stable dynamics, but if any are greater than zero, it indicates chaotic behavior, including strange attractors and sensitivity to the initial conditions.

6 Recurrent Neural Network Stability Analysis

Recurrent Neural Networks are a class of neural network architecture which are potentially very powerful yet are known to have challenges with stability. While this stability issue is well known empirically, an intriguing question is whether stability theory can give insights into RNNs. An RNN (sometimes called a vanilla RNN in the literature) consists of a state h_{t+1} which a function of the previous state and the current input. In the simplest case, this function consists of multiplying by a parameterized weight matrix and applying a non-linear "activation" function ϕ_h. Then

the predicted output for each step is a function of the state, which consists of multiplying by a learned weight matrix and applying another non-linear activation function ϕ_y.

Theorem 6 (Recurrent Neural Networks)**.**

$$h_{t+1} = \phi_h(Wh_t + W_i y_t$$
$$y_{t+1} = \phi_y(W_o h_{t+1})$$

An investigation into RNN behavior on the 3-bit flop task [1] found that a simple recurrent neural network learns basins of attractions around the fixed points which correspond to different types of contents to store in memory. Changes between these fixed points are input driven.

6.1 Vanishing and Exploding gradients

One challenge with stability in RNNs is related to the gradient. For example, it is common for vanilla RNNs trained with many time steps to have gradients which are either very small, or which or large enough to lead to overflow. We can analyze this through the lens of stability theory by viewing the gradient across the steps (going backwards in time, as in backpropagation through time) as a dynamical system.

Theorem 7 (RNN Vanishing and Exploding Gradients)**.**

$$\nabla_{h_t} L = W^T \nabla_{o_t} L + J_{t+1}^T W^T \nabla_{o_{t+1}} L + ... + J_{t+1}^T J_t^T J_{t-1}^T ... J_\tau^T W^T \nabla_{o_\tau} L$$
$$\nabla_{h_t} L = \sum_{s=t+1}^{\tau} ({}_{r=t+1}^{s} J_r^T) W^T \nabla_{o_s} L$$

Thus we can see that one of the terms in the gradient is based on the product of the jacobian across the time steps. If all of the eigenvalues of this jacobian are less than 1, then the gradient vanishes with more steps. This can also be seen in terms of the lyapunov exponents, where for any i, $\lambda_i > 0$ indicates some instability which manifests itself as "exploding" gradients.
An interesting are for future work might be to study LSTMs or self-attention RNNs from a stability point of view.

References

[1] D. Sussillo and O. Barak. Opening the black box: Low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Comput.*, 25(3):626–649, Mar. 2013. ISSN 0899-7667. doi: 10.1162/NECO_a_00409. URL http://dx.doi.org/10.1162/NECO_a_00409.