A Neural Network Approach to Learning Steady States and Their Stability of Parametric Dynamical Systems

Bo Li
Department of Mathematics
University of California, San Diego

Joint work with Y. Zhang, A. Cloninger, & X. Tian

Supported by the US National Science Foundation

School of Mathematical Sciences
Fudan University
March 28, 2024

Outline

1. Introduction
2. A Parameter-Solution Neural Network (PSNN)
   - Target functions
   - The architecture of PSNN
   - Trainging the PSNN
   - Locating solutions
3. An Approximation Theory
4. Numerical Results
5. Conclusion

1. Introduction

Dynamical systems and first-order ODEs with parameters

$$\begin{cases} \dfrac{du_1}{dt} = g_1(u_1, \ldots, u_n; \theta_1, \ldots, \theta_m), \\ \qquad \vdots \\ \dfrac{du_n}{dt} = g_n(u_1, \ldots, u_n; \theta_1, \ldots, \theta_m). \end{cases}$$

Compact form

$$\frac{dU}{dt} = G(U, \Theta),$$

$$G(U,\Theta) = \begin{bmatrix} g_1(U,\Theta) \\ \vdots \\ g_n(U,\Theta) \end{bmatrix}, \quad U(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_n(t) \end{bmatrix}, \quad \Theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_m \end{bmatrix}.$$

Given Θ: U is a **steady state** (or steady-state solution) if

$$G(U, \Theta) = 0.$$

A steady state U is (linearly) **stable** if the linearized system at U is asymptotically stable, and is (linearly) **unstable** otherwise.

Characterization of linear stability.

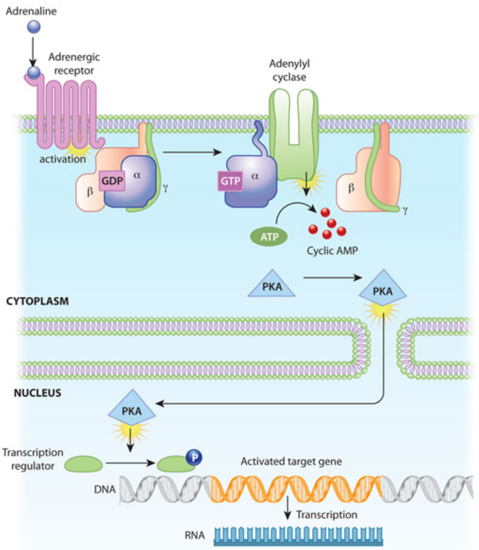
All eigenvalues of the Jacobi matrix have non-positive real part, and the algebraic and geometric multiplicity for any pure imaginary eigenvalue are the same.

Applications

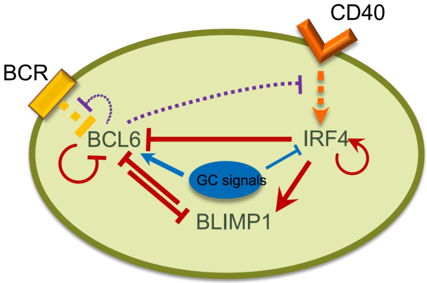
- Chemical reactions
  - Cell signal pathways
  - Metabolic dynamics
  - Pattern formation
  - $\cdots$
- Population dynamcs
- Electric circuits
- $\cdots$

Emerging issues: Large systems with many **parameters**

- Difficult to measure experimentally
- Choice of model parameters
- Sensitive to existence of multiple solutions, bifurcation, solution behavior, etc.



Cell signaling (Nature Education).



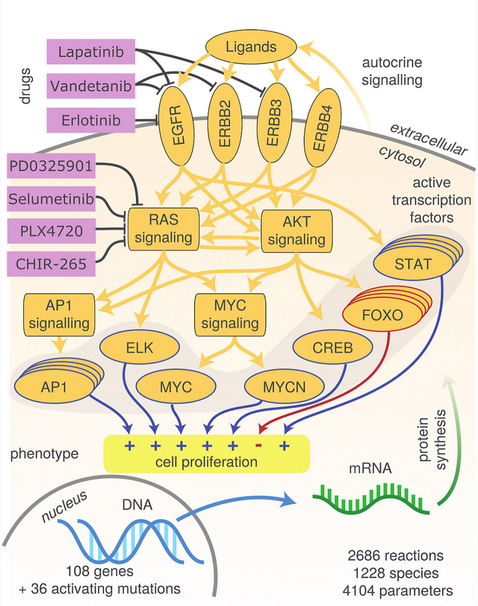
$$\frac{dP}{dt} = \mu_1 + \sigma_1 \frac{R^2}{k_3^2 + R^2} + \sigma_1 \frac{k_2^2}{k_2^2 + B^2} - P$$

$$\frac{dB}{dt} = \mu_2 + \sigma_2 \frac{k_2^2}{k_2^2 + B^2} \frac{k_1^2}{k_1^2 + P^2} \frac{k_3^2}{k_3^2 + R^2} - (1 + b(BCR))\, B$$

$$\frac{dR}{dt} = \mu_3 + \sigma_3 \frac{R^2}{k_3^2 + R^2} + p(CD40) - R$$

$$\mathrm{b(BCR)} = \frac{\mathrm{bcr}}{B^2 + k^2} \qquad \mathrm{p(CD40)} = \frac{\mathrm{cd40}}{B^2 + k^2}$$

Network of terminal differentiation of B cell (Martinez et al. PNAS 2012 & Axenie et al. Symmetry 2021).



Network of some cancer cell (Fröhlich et al. Cell Systems 2021).

TABLE 2

Rate parameters for the Feed-Forward NF-κB non-canonical ODE model

All other parameters pertaining to canonical NF-κB model are reported in Kalita *et al.* (24) after being fitted to the single-cell dynamical data in the study. All the new parameters for the feed-forward model are described in the table below. The comment column indicates the origin of the nominal values for this model.

Symbol	Values	Units	Description	Comments
n_{1a}	5×10^{-7}	s^{-1}	TRAF1-inducible mRNA synthesis	Assumption
n_{1b}	0.5	s^{-1}	TRAF1 translation rate	Fitted
n_{1c}	9.62×10^{-5}	s^{-1}	TRAF1 mRNA degradation	Fitted, Hao and Baltimore (19)
n_{1d}	0.0003	s^{-1}	TRAF1 degradation rate	Fitted
n_{1e}	0.0	μMs^{-1}	TRAF1 constitutive mRNA synthesis	Assumption
n_{2a}	5×10^{-7}	s^{-1}	TRAF2-inducible mRNA synthesis	Assumption
n_{2b}	0.5	s^{-1}	TRAF2 translation rate	Fitted
n_{2c}	0.0004	s^{-1}	TRAF2 mRNA degradation	Fitted
n_{2d}	0.0003	s^{-1}	TRAF2 degradation rate	Fitted
n_{2e}	0.0	μMs^{-1}	TRAF2 constitutive mRNA synthesis	Assumption
n_{3a}	2.5×10^{-9}	s^{-1}	NIK-inducible mRNA synthesis (NF-κB-independent)	Assumption
n_{3b}	0.5	s^{-1}	NIK translation rate	Fitted
n_{3c}	0.0004	s^{-1}	NIK mRNA degradation	Fitted
n_{3d}	0.0	μMs^{-1}	NIK-constitutive mRNA synthesis	Assumption
b_1	1.0	s^{-1}	TRAF2-NIK association	Any large
b_2	6.42×10^{-5}	s^{-1}	NIK degradation from TRAF2-NIK complex	Fitted
b_3	0.5	s^{-1}	TRAF1 association with TRAF2-NIK complex	Assumption
b_4	0.25	s^{-1}	formation of TRAF1-NIK complex by displacing TRAF2 from TRAF2-NIK complex	Fitted
nc_1	2.5×10^{-8}	s^{-1}	P100-inducible mRNA synthesis	Basak *et al.* (11)
nc_2	0.5	s^{-1}	P100 translation rate	Fitted
nc_3	3.2×10^{-5}	s^{-1}	P100 mRNA degradation	Basak *et al.* (11)
nc_4	0.0004	s^{-1}	P100 degradation rate	Fitted
nc_5	0.002	s^{-1}	TRAF1-NIK and p100 association	Fitted
nc_6	0.0	μMs^{-1}	p100-constitutive mRNA synthesis	Assumption
nc_7	7.5×10^{-4}	s^{-1}	p52 nuclear import	Basak *et al.* (11)
nc_8	0.0002	s^{-1}	p52 nuclear export	Basak *et al.* (11)
ncl_{1a}	5×10^{-7}	s^{-1}	Naf1-inducible mRNA synthesis	Assumption
ncl_{1b}	0.5	s^{-1}	Naf1 translation rate	Fitted
ncl_{1c}	0.0004	s^{-1}	Naf1 mRNA degradation	Fitted
ncl_{1d}	0.0003	s^{-1}	Naf1 degradation rate	Fitted
ncl_{1e}	0.0	μMs^{-1}	Naf1-constitutive mRNA synthesis	Assumption

NF-κB pathway (Choudhary et al. J. Bio. Chem 2013).

Questions

- For a given set of parameters, are there any steady states?
- If yes, how to find all the solutions accurately and efficiently?
- What is the stability of each steady state?
- What are solution behaviors?

Goal: Develop an artificial neural network and machine learning approach to address these issues.

- Neural network constructions, numerical algorithms, data processing, etc.
- Mathematical foundation: approximation theory, convergence rates, etc.
- Improvement and generalization.

2. A Parameter-Solution Neural Network

Consider

$$G(U,\Theta)=0 \qquad \forall U\times\Theta\in D\times\Omega.$$

- $D\subset\mathbb{R}^n$: solution space, open and bounded.
- $\Omega\subset\mathbb{R}^m$: parameter space, open and bounded.

Assumptions

A1. Partition of parameter space $\Omega=\cup_{i=0}^{M}\Omega_i$:

- $\Theta\in\Omega_0$: no solution ($\mathcal{N}_0=0$);
- $\Theta\in\Omega_i\,(1\le i\le M)$: N_i solutions, $S^\Theta=\{\hat{U}_1^\Theta,\ldots,\hat{U}_{\mathcal{N}_i}^\Theta\}$.

A2. If $1\le i\le M$ then $\overline{\Omega}_i=\overline{\Omega}_{i,1}\cup\overline{\Omega}_{i,2}$:

- $\Theta\in\Omega_{i,1}$: solution is linearly stable;
- $\Theta\in\Omega_{i,2}$: solution is linearly unstable.

Target functions

A target function for solution $\Phi : D \times \Omega \to \mathbb{R}$:

$$\Phi(U, \Theta) = \sum_{i=1}^{M} \chi_{\Omega_i}(\Theta) \sum_{j=1}^{\mathcal{N}_i} \exp\left(-\frac{|U - \hat{U}_j^{\Theta}|^2}{\delta(\Theta)} \right),$$

$$\delta(\Theta) := \begin{cases} \max\left\{ \frac{1}{4} \min_{\hat{U}_j^{\Theta}, \hat{U}_{j'}^{\Theta} \in S^{\Theta}, j \neq j'} \|\hat{U}_j^{\Theta} - \hat{U}_{j'}^{\Theta}\|_2, \delta_0 \right\} & \text{if } |S^{\Theta}| \geq 2, \\ \delta_1 & \text{if } |S^{\Theta}| = 1, \end{cases}$$

where $\delta_0 > 0$ and $\delta_1 > 0$ are numerical parameters.

- $0 \leq \Phi(U, \Theta) \leq 1$ for all $(U, \Theta) \in D \times \Omega$.
- $\Phi(U, \Theta) = 1$ if and only if $\Theta \in \Omega_i$ for some $i \geq 1$ and $U \in S^{\Theta}$.

A target function for stability $\Phi^{\mathrm{s}} : D \times \Omega \to \mathbb{R}$:

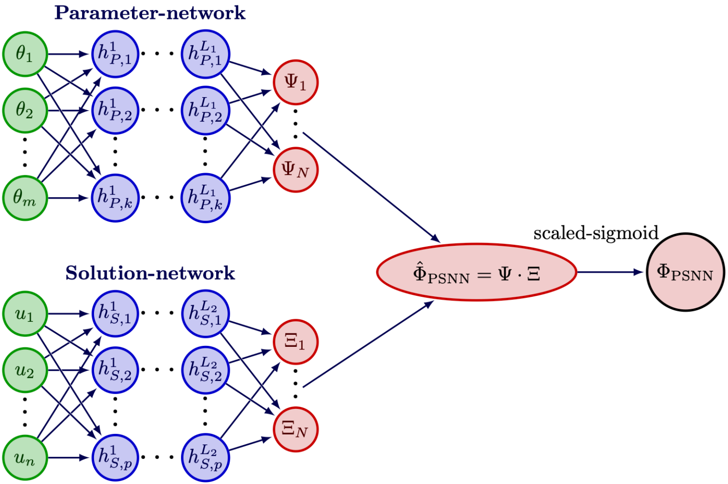
$$\Phi^{\mathrm{s}}(U,\Theta) = \sum_{i=1}^{M} \chi_{\Omega_i}(\Theta) \sum_{j=1}^{\mathcal{N}_i} (-1)^{s_j^{\Theta}} \exp\left(-\frac{|U - \hat{U}_j^{\Theta}|^2}{\delta(\Theta)}\right),$$

where $s_j^{\Theta} = 0$ if $\hat{U}_j^{\Theta} \in S^{\Theta}$ is (linearly) stable and $s_j^{\Theta} = 1$ if $\hat{U}_j^{\Theta} \in S^{\Theta}$ is (linearly) unstable.

If $\Theta \in \Omega_i$ for some $i \in \{1, \ldots, M\}$ and $U \in S^{\Theta}$, then

- U is stable if and only if $\Phi^{\mathrm{s}}(U,\Theta) \approx 1$;
- U is unstable if and only if $\Phi^{\mathrm{s}}(U,\Theta) \approx -1$.

The architecture of PSNN



Fix an integer $N \geq 1$.

Define a *parameter neural network* (PNN) and a *solution neural network* (SNN)

$$\Phi_{\mathsf{PNN}}(\cdot, \omega_{\mathsf{P}}) : \Omega \to \mathbb{R}^N \quad \text{and} \quad \Phi_{\mathsf{SNN}}(\cdot, \omega_{\mathsf{S}}) : D \to \mathbb{R}^N.$$

- In the form $T_{L+1} \circ T_L \circ \cdots \circ T_1$ (L: number of hidden layers). $T_j(x) = a_j((A_j x + b_j))$ $(1 \leq j \leq L)$, $T_{L+1}(x) = A_{L+1}x + b_{L+1}$, each $a_j = \mathsf{ReLU}$, A_j matrix, and b_j vector.
- ω_{P} and ω_{S} : neural network parameters.

Define $\Phi_{\mathsf{PSNN}}(\cdot, \cdot, \omega) : D \times \Omega \to \mathbb{R}$ by

$$\Phi_{\mathsf{PSNN}}(U, \Theta, \omega) = \sigma(\Phi_{\mathsf{PNN}}(\Theta, \omega_{\mathrm{P}}) \cdot \Phi_{\mathsf{SNN}}(U, \omega_{\mathrm{S}})).$$

- $\omega = \omega_{\mathrm{P}} \cup \omega_{\mathrm{S}}$.
- $\sigma : \mathbb{R} \to (-\eta, 1+\eta)$ is a rescaled sigmoid function with $\eta > 0$ and $1 + \eta > \sup_{\Omega \times D} \Phi$.

The neural network $\Phi^{\mathrm{s}}_{\mathrm{PSNN}}$ approximating Φ^{s} is similar.

Training the PSNN

Training data

$$\mathcal{T}_{\mathrm{train}} = \left\{ \left(\Theta_i, \hat{U}_j^{\Theta_i}, \Phi(\hat{U}_j^{\Theta_i}, \Theta_i) \right)_{j=1}^{\mathcal{N}_{m_i}}; (\Theta_i, U_j, \Phi(U_j, \Theta_i))_{j=1}^{N_{\mathrm{random}}} \right\}_{i \in I_{\mathrm{train}}}$$

Loss function

$$\mathcal{L}(\omega) = \frac{1}{|\mathcal{T}_{\mathrm{train}}|} \sum_{i \in I_{\mathrm{train}}} \left[\sum_{j=1}^{\mathcal{N}_{m_i}} \left(\Phi_{\mathrm{PSNN}}(\hat{U}_j^{\Theta_i}, \Theta_i, \omega) - \Phi(\hat{U}_j^{\Theta_i}, \Theta_i) \right)^2 + \sum_{j=1}^{N_{\mathrm{random}}} \left(\Phi_{\mathrm{PSNN}}(U_j, \Theta_i, \omega) - \Phi(U_j, \Theta_i) \right)^2 \right]$$

Minimizing the loss function by ADAM, a stochastic gradient deschent algorithm.

Locating solutions

Given $\Theta \in \Omega$, solutions in S^Θ are peaks of the graph of $U \mapsto \Phi(U, \Theta)$ ($U \in D$).

- Based on the structure of the target function: A sum of Gaussian radial basis functions.
- No peaks means no solutions.

Locate the solutions in three steps:

(1) Choose a set of points $\mathcal{U} \subset D$ uniformly scattered in D and calculate $\Phi_{\mathsf{PSNN}}(U, \Theta, \omega)$ for all $U \in \mathcal{U}$.

(2) Choose a cut value $L_{\mathrm{cut}} \in (0, 1)$ and collect all the points $U \in \mathcal{U}$ such that $\Phi_{\mathrm{PSNN}}(U, \Theta, \omega) \geq L_{\mathrm{cut}}$. Denote by $\mathcal{U}^\Theta_{\mathrm{collected}}$ the set of such points.

(3) Apply the K-means clustering method with pre-chosen maximum cluster size $C_{\max}$ and silhouette score $\mathrm{sil} \in (0, 1)$, on the set of collected points $\mathcal{U}^\Theta_{\mathrm{collected}}$ to locate the centers.

Algorithm

Input: A parameter vector $\Theta \in \Omega$, an optimal threshold value $L_{\text{cut}} \in (0,1)$, a set of points $\mathcal{U} \subset D$, the set of neural network parameters ω of the trained PSNN Φ_{PSNN}, a maximum number of clusters $C_{\max}$, and a silhouette scoring number $\text{sil}_1 \in (0,1)$.

Output: A set of centers $\mathcal{U}^{\Theta}$.

Get $\mathcal{U}^{\Theta}_{\text{collected}} = \{U \in \mathcal{U} : \Phi_{\text{PSNN}}(U, \Theta; \omega) \geq L_{\text{cut}}\}$

Get $\mathcal{U}^{\Theta} =$ `Cluster`$(\mathcal{U}^{\Theta}_{\text{collected}})$

Function `Cluster`$(\mathcal{U})$

- **while** $2 \leq j \leq C_{\max}$ **do**
  - perform K-means with j clusters on $\mathcal{U}$ and get an average score sil_{j}
- $k \leftarrow \underset{2 \leq j \leq C_{\max}}{\arg\max} \{\text{sil}_{\text{j}}\}$
- **if** $\text{sil}_{\text{k}} \geq \text{sil}_1$ **then**
  - perform K-means with k clusters on $\mathcal{U}$ and get the set of centers $\mathcal{U}^{\Theta}$
- **else**
  - take the mean on $\mathcal{U}$ and get the set of centers $\mathcal{U}^{\Theta}$
- **return** $\mathcal{U}_{\Theta}$

3. An Approximation Theory

Main Theorem. Assume $D \subset \mathbb{R}^n$ is a hyperrectangle, each Ω_i ($0 \leq i \leq M$) is smooth, and $\Phi(U, \Theta)$ is smooth in U and piecewise smooth in Θ. Then, for any $\epsilon > 0$, there exists Φ_{PSNN} such that

$$\left\|\Phi - \Phi_{\mathrm{PSNN}}\right\|_{L^2(D\times\Omega)} < \epsilon.$$

Moreover, the required maximum number of layers only depends on the smoothness of Φ, m, and n. The required maximum number of nonzero weights and the dimension N are $O(\epsilon^{-r})$ and $O(\epsilon^{-s})$ with r and s depending on m, n, and the smoothness of Φ.

Key ideas of the proof.

(1) $\Psi_N(y) \cdot \Xi_N(x) \approx \Phi(y, x)$ by kernel decomposition.

(2) Estimte the decay rate for eigenvalues.

(3) Deep ReLU approximation of piecewise smooth functions.

(1) Approximate $\Phi(y,x)$ by $\Psi(y)\cdot\Xi(x)$.

Define $K_\Phi \in L^2(D\times D)$ and $T_\Phi : L^2(D)\to L^2(D)$:

$$K_\Phi(y,z) = \int_\Omega \Phi(y,x)\Phi(z,x)dx \qquad \forall y,z\in D.$$

$$(T_\Phi\phi)(y) = \int_\Omega K_\Phi(y,z)\phi(z)\,dz \qquad \forall y\in D.$$

- Both are well defined.
- The kernel K_Φ is a Mercer's kernel:
  - $K_\Phi \in C(\overline{D\times D})$;
  - symmetric: $K_\Phi(y,z) = K_\Phi(z,y)$; and
  - positive semi-definite: $\langle T_\Phi f, f\rangle \geq 0$ for any $f\in L^2(D)$.
- The operator K_Φ is linear, self-adjoint, nonnegative, and compact.

Classical operator theory:

- $T_\Phi : L^2(D) \to L^2(D)$ possesses countably many eigenvalues, all nonnegative;
- Each positive eigenvalue has finitely many eigenfunctions; and
- All eigenfunctions form a complete orthonormal system of the Hilbert space $L^2(D)$.

Mercer's Theorem. Let λ_j $(j \geq 1)$ be all the positive eigenvalues (counting multiplicity), nondecreasing, and $\{e_j\}_{j=1}^{\infty}$ the corresponding orthonormal eigenfunctions for T_Φ. Then,

- All $e_j \in C(\overline{D})$ $(j \geq 1)$;
- $K_\Phi(y,z) = \sum_{j=1}^{\infty} \lambda_j e_j(y) e_j(z)$ $(y, z \in D)$ with abolute and uniform convergence.

Expand the target function

$$\Phi(x,\cdot) = \sum_{j=1}^{\infty} \varphi_j(x) e_j \quad \text{in } L^2(D) \quad \forall x \in \Omega,$$
$$\varphi_j(x) = \langle \Phi(x,\cdot), e_j \rangle_D.$$

Define

$$\Psi_N = (e_1, \dots, e_N) \in [L^2(D)]^N,$$
$$\Xi_N = (\varphi_1, \dots, \varphi_N) \in [L^2(\Omega)]^N,$$
$$(\Psi \cdot \Xi)_N(y,x) = \Psi_N(y) \cdot \Xi_N(x).$$

Then,

$$\|\Phi - (\Psi \cdot \Xi)_N\|^2_{L^2(D\times\Omega)} = \int_D \sum_{k=N+1}^{\infty} \lambda_k e_k(y)^2 dy = \sum_{k=N+1}^{\infty} \lambda_k.$$

(2) Estimate the decay rate for eigenvalues.

Known: $K_\Phi \in H^p(D \times D)$ then $0 \le \lambda_k \le Ck^{-p/n}$.

$$\sum_{k=N+1}^{\infty} \lambda_k \le C \sum_{k=N+1}^{\infty} Ck^{-p/n}) \le C \int_N^{\infty} t^{-p/N} dt \le CN^{-(p-n)/n}.$$

(3) Deep ReLU approximation of piecewise smooth functions.
There exist ReLU neural networks Φ_{NN} and Ξ_{NN} such that

$$\|\Psi_N - \Psi_{\mathrm{NN}}\|^2_{L^2(D,\mathbb{R}^N)} \le \frac{\epsilon}{P} \quad \text{and} \quad \|\Xi_N - \Xi_{\mathrm{NN}}\|^2_{L^2(\Omega,\mathbb{R}^N)} \le \frac{\epsilon}{P}$$

with $\sqrt{\sum_{j=1}^N \lambda_j}/P + \sqrt{N}/P + 1/P^2 < 1/2$.
(Rescale the sigmoid function if necessary.)

Wrap up: Define $\Phi_{PSNN} = \Psi_{rmNN} \cdot \Xi_{\mathrm{NN}}$ and use the triangle inequality.

4. Numerical Results

The Gray–Scott model

$$\begin{cases} -uv^2 + f(1-u) = 0, \\ uv^2 - (f+k)v = 0. \end{cases}$$

- $U = (u, v)$, $\Theta = (f, k)$, $D = (0, 1)^2$, $\Omega = (0, 0.03) \times (0, 0, 08)$.
- Exclude the trivial solution. Then, $\overline{\Omega} = \overline{\Omega_0} \cup \overline{\Omega_1}$, where

$$\Omega_0 = \{(f, k) \in \Omega : \Delta(f, k) < 0\},$$
$$\Omega_1 = \{(f, k) \in \Omega : \Delta(f, k) > 0\},$$
$$\Delta(f, k) = f^2 - 4f(f+k)^2.$$

Define the solution phase boundary

$$\Gamma_{\mathrm{soln}} = \{(f, k) \in \Omega : \Delta(f, k) = 0\}.$$

$\Theta = (f, k) \in \Omega_0$: no solution.
$\Theta = (f, k) \in \Omega_1$: two distinct solutions $\hat{U}_1^{\Theta}$ and $\hat{U}_2^{\Theta}$.

$$\hat{U}_1^{\ominus} = (\hat{u}_1^{\ominus}, \hat{v}_1^{\ominus}) = \left(\frac{f - \sqrt{\Delta(f,k)}}{2f}, \frac{f + \sqrt{\Delta(f,k)}}{2(f+k)} \right),$$

$$\hat{U}_2^{\ominus} = (\hat{u}_2^{\ominus}, \hat{v}_2^{\ominus}) = \left(\frac{f + \sqrt{\Delta(f,k)}}{2f}, \frac{f - \sqrt{\Delta(f,k)}}{2(f+k)} \right).$$

We have further $\overline{\Omega_1} = \overline{\Omega_{1,1}} \cup \overline{\Omega_{1,2}}$, where

$$\Omega_{1,1} = \{(f,k) \in \Omega_1 : f\sqrt{\Delta(f,k)} + f^2 - 2(f+k)^3 > 0\},$$
$$\Omega_{1,2} = \{(f,k) \in \Omega_1 : f\sqrt{\Delta(f,k)} + f^2 - 2(f+k)^3 < 0\}.$$

$\Theta = (f,k) \in \Omega_{1,1}$: $\hat{U}_1^{\ominus}$ is stable and $\hat{U}_2^{\ominus}$ is unstable.
$\Theta = (f,k) \in \Omega_{1,2}$: both solutions are unstable.

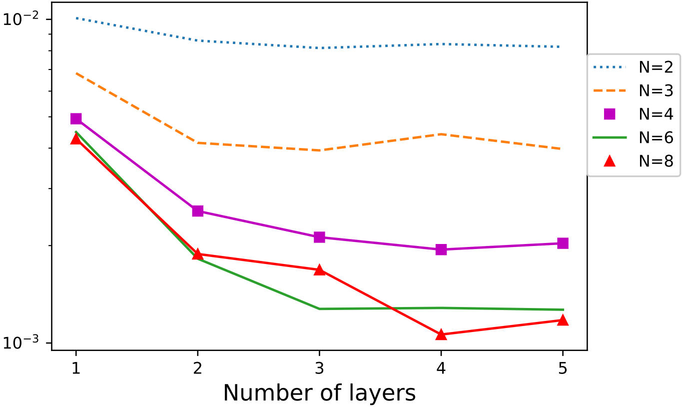
Define the stability phase boundary

$$\Gamma_{\mathrm{stab}} = \{(f,k) \in \Omega_1 : f\sqrt{\Delta(f,k)} + f^2 - 2(f+k)^3 = 0\}.$$

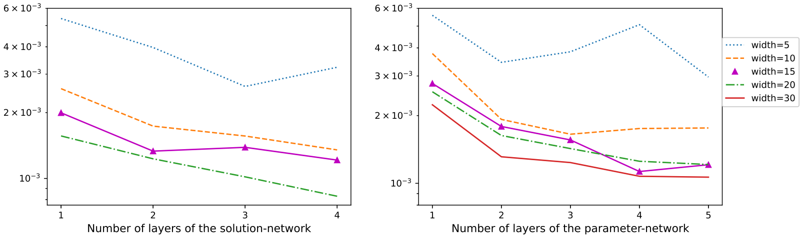
The target functions for solution and stability are

$$
\begin{aligned}
\Phi(U,\Theta) &= \Phi((u,v),(f,k)) \\
&= \chi_{\Omega_1}((f,k)) \sum_{j=1}^{2} \exp\left(-\frac{|u-\hat{u}_j^{\Theta}|^2}{\delta((f,k))} - \frac{|v-\hat{v}_j^{\Theta}|^2}{\delta((f,k))} \right), \\
\Phi^{\mathrm{s}}(U,\Theta) &= \Phi^{\mathrm{s}}((u,v),(f,k)) \\
&= \chi_{\Omega_{1,1}}((f,k)) \Bigg(\exp\left(-\frac{|u-\hat{u}_1^{\Theta}|^2}{\delta((f,k))} - \frac{|v-\hat{v}_1^{\Theta}|^2}{\delta((f,k))} \right) \\
&\qquad - \exp\left(-\frac{|u-\hat{u}_2^{\Theta}|^2}{\delta((f,k))} - \frac{|v-\hat{v}_2^{\Theta}|^2}{\delta((f,k))} \right) \Bigg) \\
&\qquad - \chi_{\Omega_{1,2}}((f,k)) \Bigg(\exp\left(-\frac{|u-\hat{u}_1^{\Theta}|^2}{\delta((f,k))} - \frac{|v-\hat{v}_1^{\Theta}|^2}{\delta((f,k))} \right) \\
&\qquad + \exp\left(-\frac{|u-\hat{u}_2^{\Theta}|^2}{\delta((f,k))} - \frac{|v-\hat{v}_2^{\Theta}|^2}{\delta((f,k))} \right) \Bigg).
\end{aligned}
$$

Convergence test

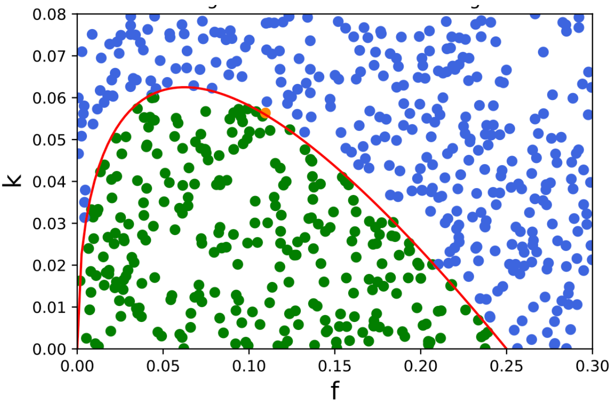


Errors in log scale against the depth (i.e., number of layers) of the two sub-networks for different output dimensions N.

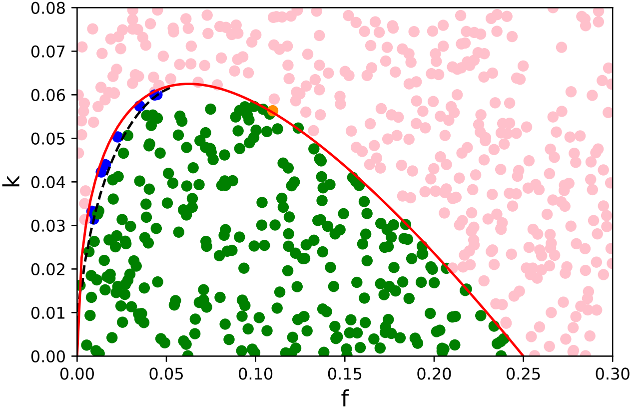


Errors in log scale against the depth with varying width for the PSNN with a fixed dimension $N = 8$ of the output vectors. The x-axis represents the depth of the solution-network or parameter-network. (Left) The parameter-network has the fixed structure $L_1 = 4$ and $W_1 = 30$. (Right) The solution network has the fixed structure $L_2 = 4$, $W_2 = 30$.

Locating solutions and phase boundaries

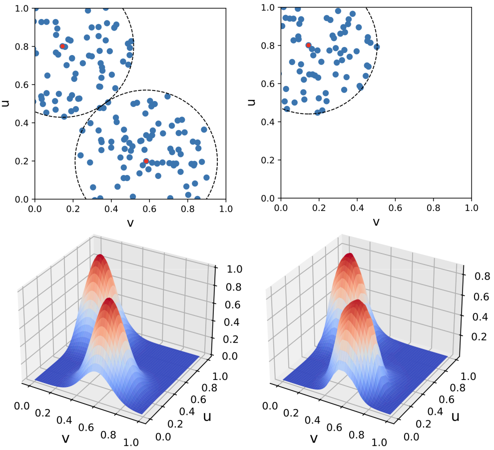


The blue points represent the parameter pairs that the algorithm recognized as *no-solution*, and the brown points, green points and orange points respectively correspond to *1-solution*, *2-solution*, *3-and-more-solution*. The red curve is the solution phase boundary Γ_{soln}.

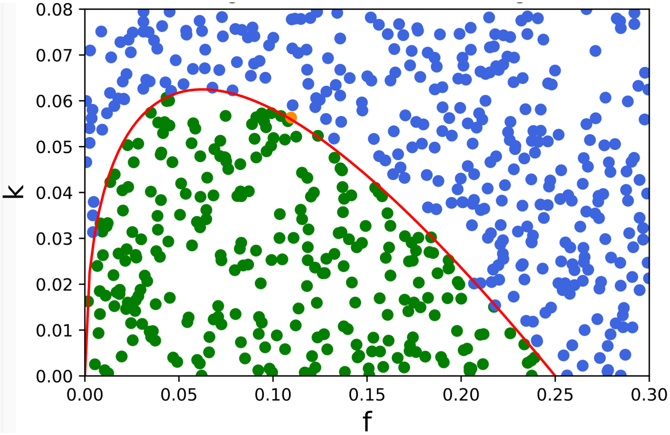


The solution phase diagram with stability information. The blue points, green points, brown points respectively represent *2-unstable-solution*, *1-stable-1-unstable-solution*, *2-stable-solution*, and the black dashed curve is the stability boundary Γ_{stab}.

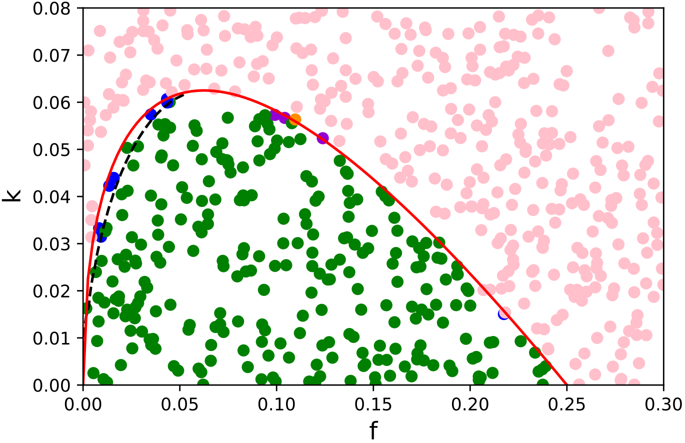
Incomplete data



(Top) With a given $\Theta = (f, k)$ for which two solutions exist, "complete data" (left) and "incomplete" data are generated. (Bottom) Plot of the target function (left) and the trained neural network (right).



The solution phase boundary predicted by the PSNN trained with incomplete data.



The solution and stability phase boundaries predicted by the PSNN trained with incomplete data.

5. Conclusion

Summary

- Construction of PSNN approximating target functions that characterizing parameter-solution pairing and solution stability.
- An approximation theory: kernel decomposition, eigenvalue decay rates, and neural networks approximations.
- Numerical tests: convergence, phase boundaries, and recover of missing information from incomplete data.

Discussions

- Locating solutions with grid points: curse of dimensionality?
- Incomplete data: unsupervised learning?
- Large systems? More efficient algorithms?
- Applications?
- ▶ Extension and improvement.

Thank You!