Chair of Scientific Computing in Computer Science
School of Computation, Information and Technology
Technical University of Munich

Systems identification with random feature neural networks

Rahul Manavalan, Daniel Lehmberg, and Felix Dietrich

Random feature neural networks.

If $x \in \Omega \subset \mathbb{R}^d$, a generic function $g : \Omega \mapsto \mathbb{R}^p$ can be approximated using a single hidden layer neural network.

$$\hat{g}_x \approx \sum_{l=1}^{L} a_l \phi_l(x) := \sum_{l=1}^{L} a_l \, \sigma(W_l^T x - b_l) \tag{1}$$

Here $a_l \in \mathbb{R}^{p\times 1}, W_l \in \mathbb{R}^{1\times d}, b_l \in \mathbb{R}$ and σ is the activation function (we use $tanh$ here). Compare this with a kernel regression model.

$$\tilde{g}_x \approx K_{xy}\left[K_{yy} + \Lambda\right]^{-1} g_y \tag{2}$$
$$K_{xy}^{ij} = \kappa(x_i - y_j) \tag{3}$$

Here κ is a suitably chosen covariance kernel.
It follows that W_l, b_l are functions of $z \in \Omega\times\Omega$. Based on the work from Bolager et. al[1], one can then evaluate the parameters that depend on randomly sampled $z_r \sim \mathcal{U}_{\Omega\times\Omega}$ using

$$W_l = \alpha\frac{z_l^1 - z_l^2}{\|z_l^1 - z_l^2\|^2}; \quad b_l = \beta + \langle W_l, z_l^1\rangle \tag{4}$$

α, β depend on the choice of σ. a_l is obtained by solving a least squares problem.
In this inference procedure, $\hat{g}$ constitutes a random feature neural network (RFNN).

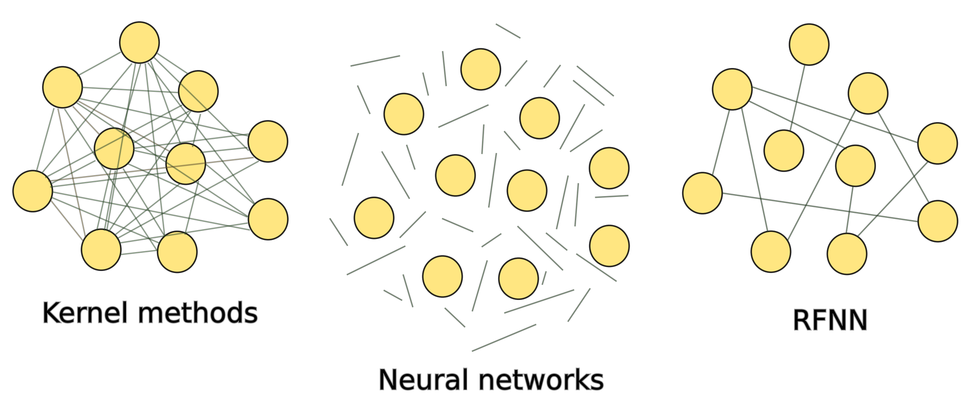

Figure 1 $\mathbb{R}^2$ domain supports for kernel methods with $\kappa(x-y)$, neural networks, random feature neural networks.

Method.

Consider a dynamical system,

$$\frac{d}{dt}x(t) = f(x) \tag{5}$$
$$x(0) = x_0 \in \mathbb{R}^d \tag{6}$$

Given a time-series of observables $[x_0, x_1, ..., x_T]$, infer $\hat{f}(x) \approx f(x)$ using a random feature neural network.

Algorithm 1: SI-RFNN – Systems identification with random feature neural networks.

Require: Reference data points, $M = [x_0, x_1, ..., x_T], t = [t_0, t_1, ..., t_T]$
Query points $Q = [y_1, y_2, ..., y_P]$

Approximate $\frac{dx(t)}{dt}$ with $Df = \text{FD}(M, t)$ – a suitable finite difference scheme.
Sample the weights and biases – $W_{1:L}, b_{1:L}$ of $\hat{f}(x; W, b)$ using subroutine 1.
Assemble $\Phi := \phi_{1:L}$ using eq. (1).
Solve for minimizer $a_* = \arg\min_{a\in\mathcal{A}} \|Df - a^T\Phi\|_2$ with suitable regularization.
Assemble Φ_y, using eq. (1).
return $\hat{f}_y := a_*^T \Phi_y$

Subroutine 1: Sample parameters

Require: Reference datapoints M, its finite difference approximation Df.

Evaluate sampling density $\rho(z)$ using subroutine 2.
Sample $z_{1:L} \sim P_{\rho(z)}$
Evaluate $W_{1:L}, b_{1:L}$ using eq. (4).
return $W_{1:L}, b_{1:L}$

Subroutine 2: Evaluate sampling density

Require: Reference datapoints M, its finite difference approximation Df, sampling frequency q, sampling heuristic function $H : x^1 \times x^2 \times Dx^1 \times Dx^2 \mapsto \mathbb{R}$.

Draw q samples $(x_{1:q}^1, x_{1:q}^2) \sim M \times M$ uniformly.
Lookup the corresponding finite difference approximations $DF_{x_{1:q}^1}, DF_{x_{1:q}^2}$
Evalute $h_l := H(x_l^1, x_l^2, DF_{x_{1:q}^1}, DF_{x_{1:q}^2})$ for $l \in 1 : q$.
Assign $\rho := [h_1, .., h_l, ...h_q]$.
return ρ

Numerical experiments.

Learning flow directly

Temperature estimation – residential heating system $T_{out} : \mathbb{R}^{12} \mapsto \mathbb{R}^3$

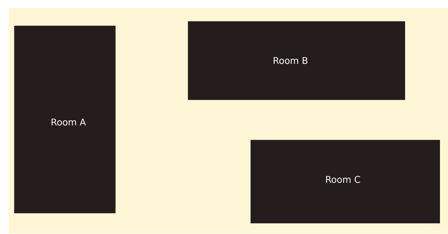

Figure 2 Layout

Let T_{in}, Q_c, Q_h, I be the input temperature, cooling rate, heating rate and occupancy of the three compartments. The target functions is:

$$T_{out}(t) = g\left(T_{in}(t), Q_c(t), Q_h(t), I(t)\right) \tag{7}$$

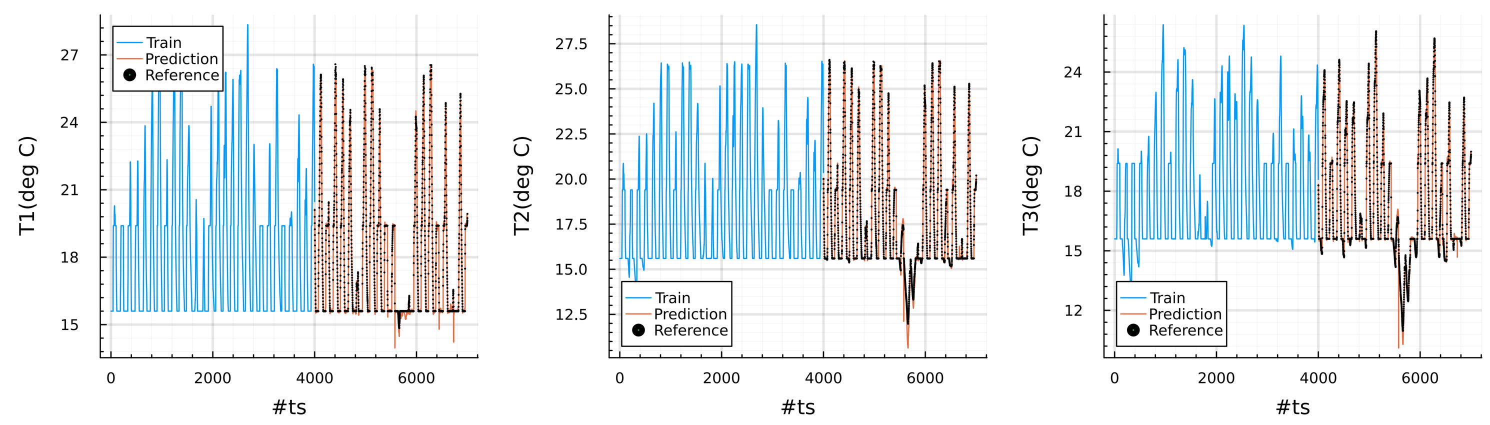

Figure 3 Evolution of temperature over time in three compartments of a residential building.

Learning vector field

Lotka Volterra model

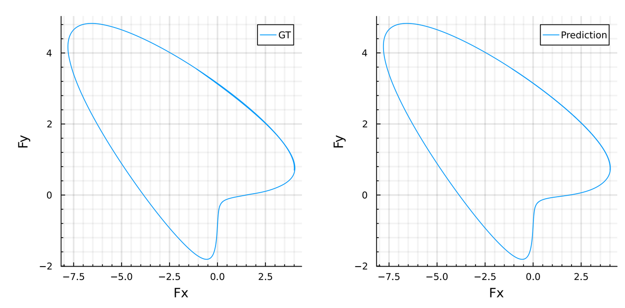

Figure 4 Actual and learned vector field coefficients of a Lotka Volterra model.

Lorenz attractor

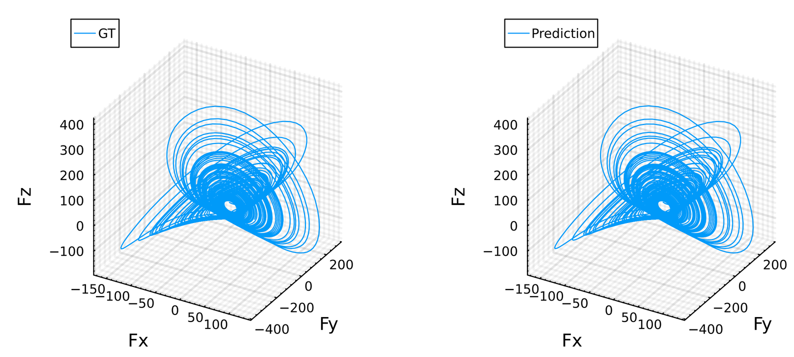

Figure 5 Actual and learned vector field coefficients of the Lorenz attractor.

Vortex shedding - a mean field model[2]

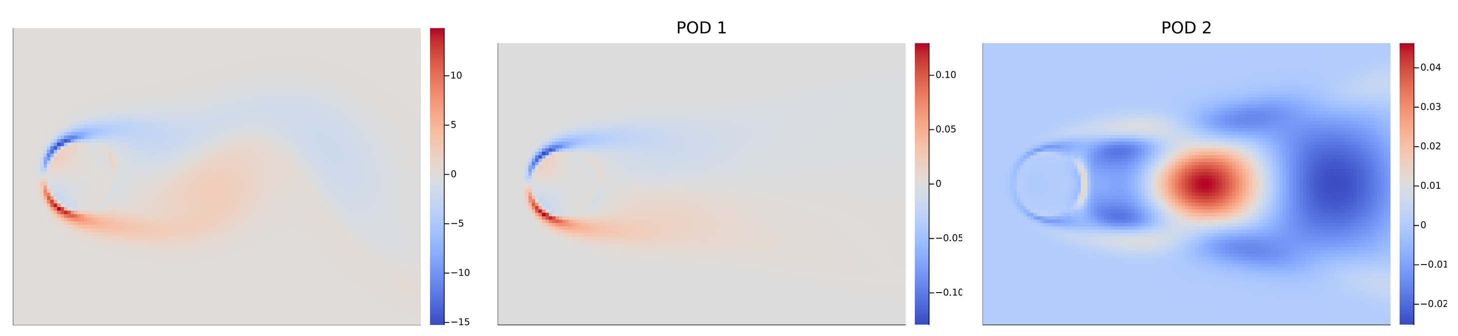

Figure 6 a) Snapshot of vortex shedding simulation b) Principal orthogonal directions

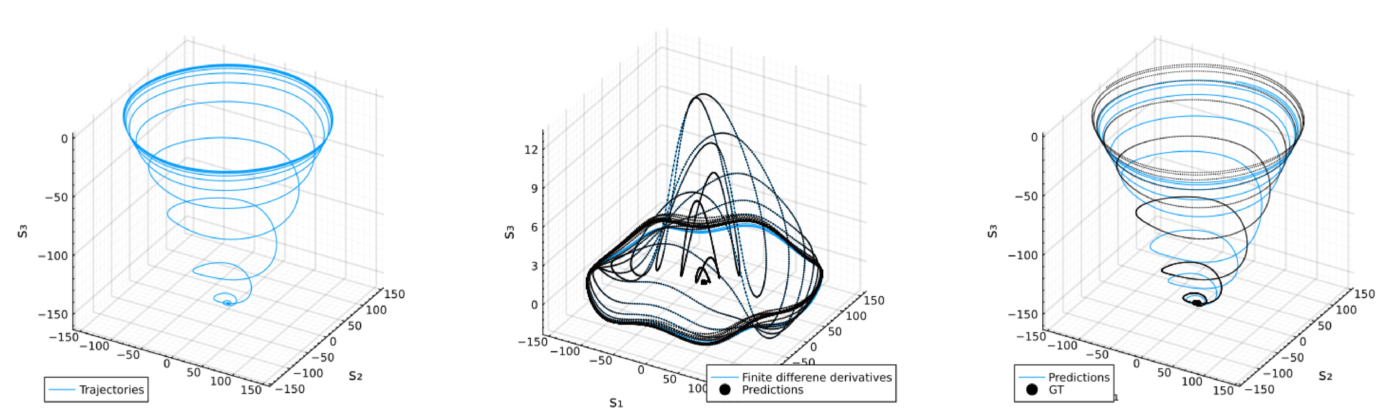

Figure 7 a) Trajectory b) Vector field coefficients c) Trajectory from the learned vector field.

1. Erik Lien Bolager, Iryna Burak, Chinmay Datar, Qing Sun, Felix Dietrich. (2023). Sampling weights of deep neural networks. Pre-print: https://arxiv.org/abs/2306.16830. Accepted at NeurIPS, 2023.

2. Noack, B., Afanasiev, K., Morzynski, M., Tadmor, G., Thiele, F. (2003). A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. Journal of Fluid Mechanics, 497, 335-363. doi:10.1017/S0022112003006694