Inverse Maps Based on Deep Neural Networks for Inverse Problems with Challenging Physical Models

Johann Rudi

Mathematics and Computer Science Division, Argonne National Laboratory

collaborations with
German Villalobos and Andreas Mang (Houston University)
Julie Bessac and Amanda Lenzi (Argonne National Laboratory)

Outline

Introduce the forward and inverse problem

Propose inverse maps with dense and convolutional neural networks

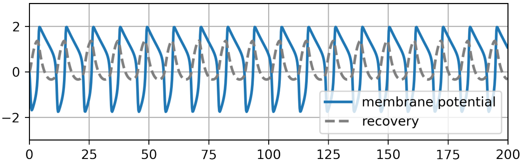
Demonstrate parameter estimation capabilities and sensitivities

Extend inference to more parameters and different models

Forward problem: ODE modeling spiking biological neurons

The FitzHugh–Nagumo[1] model is a nonlinear system of two ODEs

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \gamma\left(u - \frac{u^3}{3} + v + \zeta\right)$$
$$\frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{1}{\gamma}\left(u - \theta_0 + \theta_1 v\right)$$

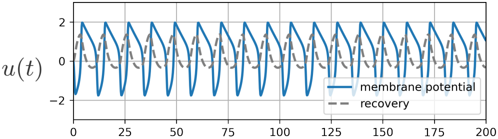


Example solution of FitzHugh–Nagumo model ($u(t)$ in blue color).

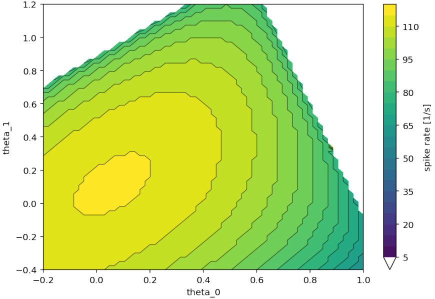
- ODE unknowns: membrane potential $u(t)$, recovery variable $v(t)$
- Known: stimulus $\zeta \equiv$ const., damping $\gamma \equiv$ const.
- Uncertain parameters: $\boldsymbol{m} := (\theta_0, \theta_1)$
- Observed data: membrane potential $u(t)$

[1]FitzHugh 1961; Nagumo, Arimoto, and Yoshizawa 1962.

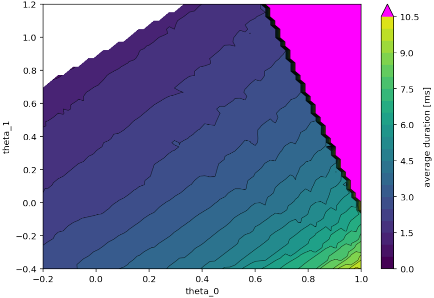
Influence of the chosen parameters on model outputs



Average spike rate of $u(t)$



Average spike duration of $u(t)$



→ Parameters θ_0, θ_1 are chosen because of their fundamental influence on the membrane potential $u(t)$.

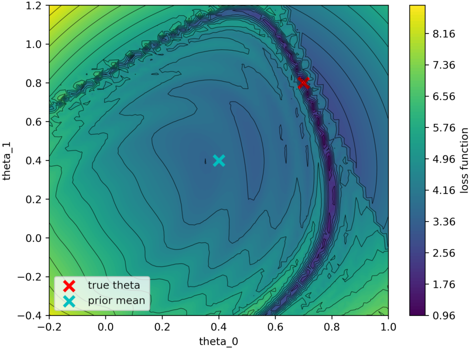
Inverse problem is problematic for gradient-based methods

Consider, e.g., MAP point estimate

$$\min_{\boldsymbol{m}} \frac{1}{2} \left\| (d(t) - u_{\boldsymbol{m}}(t))/\sigma_{\text{noise}} \right\|_{L_2}^2 + \frac{1}{2} \left| \boldsymbol{m} - \bar{\boldsymbol{m}}_{\text{pr}} \right|_{\Sigma_{\text{pr}}^{-2}}^2$$

Data: $d(t) = u_{\boldsymbol{m}}(t) + \eta(t)$, for "true" params $\boldsymbol{m}$

Noise: $\eta(t_i) = \rho\, \eta(t_{i-1}) + \epsilon(t_i)$, $\eta(t) \sim \mathcal{N}\left(0, \sigma^2/\Delta_t^2\right)$



Challenges:

- Highly nonlinear and nonconvex loss (figure)
- Sharp gradients, strong nonlinear dependencies between parameters, multiple local minima
- Weak assumptions on regularization / prior, because little is known about the parameter values in practice

Outline

Introduce the forward and inverse problem

Propose inverse maps with dense and convolutional neural networks

Demonstrate parameter estimation capabilities and sensitivities

Extend inference to more parameters and different models

Idea: Inverse maps based on deep artificial neural networks

Replace optimization by computationally learning an inverse map[2] using deep neural networks (NNs)

$$\tilde{\mathcal{F}}^{-1}_{\text{training data}} : \boldsymbol{d} \mapsto \boldsymbol{m}, \quad \text{where } \tilde{\mathcal{F}}^{-1}_{\text{training data}} \text{ is a NN}$$

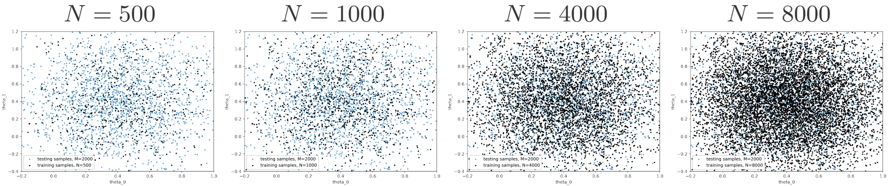
- Observational data $\boldsymbol{d}$ (membrane potential + noise) is input to the NN
- Parameters of ODE $\boldsymbol{m}$ are output of the NN
- NN is learning to directly represent a "pseudoinverse" of the forward operator
- $\tilde{\mathcal{F}}^{-1}_{\text{training data}}$ depends on training data, or the distribution which gave rise to the samples of the training data
- NN layers in $\tilde{\mathcal{F}}^{-1}_{\text{training data}}$ are dense, convolutional, or average pooling

[2] Arridge et al. 2019; Khoo and Ying 2019.

Idea: Use prior distribution to generate training data

Training data

- Sample parameters from the prior distribution (here, a Gaussian)
- Use different quantities of training samples: $N = 500, 1000, 4000, 8000$



Parameters sampled from prior used as training data *(black dots)* versus testing data *(blue dots)* that is fixed to $M = 2000$ samples.

- Simulate ODE for each parameter sample (driver of computational cost)

Questions considered in this talk

- How accurate are the estimates from trained NN?
- What is the influence of the NN architecture?
- What is the sensitivity with respect to size of the training data?
- What is the sensitivity with respect to noise in the training data?
- Does the framework generalize to other parameters and models?

Outline

Introduce the forward and inverse problem

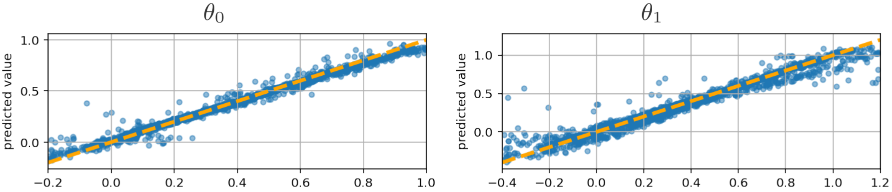
Propose inverse maps with dense and convolutional neural networks

Demonstrate parameter estimation capabilities and sensitivities

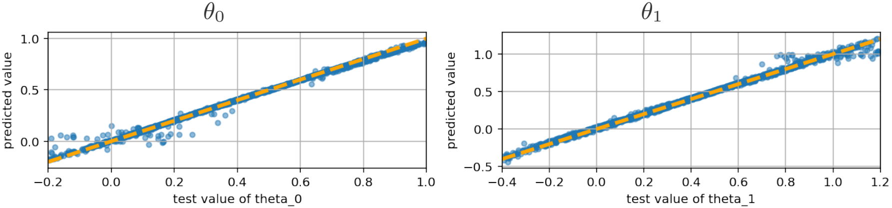
Extend inference to more parameters and different models

Accuracy of ODE parameters estimated with NNs

- Perform a search of NN hyperparameters (#layers, #units, #filters, ...)
- Result with **dense NN**: 4 dense layers with 32 units each



- Result with **CNN**: 3 convolutional layers (8,16,32 filters) and 2 dense layers (32 units)



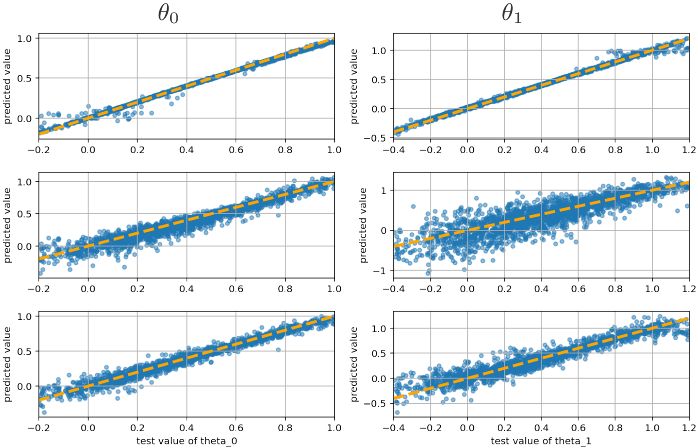
Sensitivity of estimation accuracy w.r.t. data noise

CNN: 3 convolutional layers (8,16,32 filters) and 2 dense layers (32 units)

train w/o noise
test w/o noise

train w/o noise
test w/ noise

train w/ noise
test w/ noise



Simulated ODE output with estimated parameters

Simulate the FitzHugh–Nagumo ODE with predicted parameters from **CNN**.

Each graph shows the median percentile of MSE between testing and simulated time series.

train w/o noise
test w/o noise

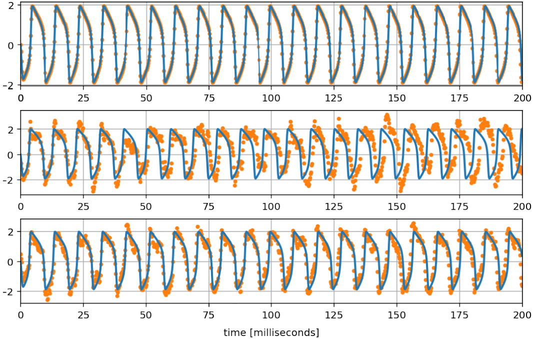
Time series error:
Sq. bias $= 2.8 \times 10^{-6}$
C-MSE $= 0.061$

train w/o noise
test w/ noise

Time series error:
Sq. bias $= 1.8 \times 10^{-3}$
C-MSE $= 2.001$

train w/ noise
test w/ noise

Time series error:
Sq. bias $= 1.2 \times 10^{-3}$
C-MSE $= 0.536$



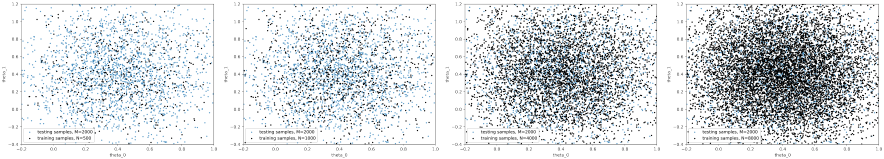
Model outputs of FitzHugh–Nagumo ODE *(blue lines)* using parameters from CNN estimates; corresponding data that gave rise to estimates are shown as *orange dots*.

Sensitivity of CNN predictions to training data sizes

Consider the metrics **Median-APE** (i.e., median relative error) and **R^2** (coefficient of determination in brackets, $R^2 = 1$ is ideal)

N	**train noise-free test noise-free**	**train noise-free test with noise**	**train with noise test with noise**
500	0.023 (0.990)	0.169 (0.788)	0.098 (0.921)
1000	0.014 (0.995)	0.174 (0.763)	0.096 (0.938)
4000	0.014 (0.997)	0.204 (0.710)	0.060 (0.970)
8000	0.014 (0.998)	0.251 (0.617)	0.053 (0.976)

$N = 500$ $N = 1000$ $N = 4000$ $N = 8000$



Parameters sampled from prior used as training data *(black dots)* versus testing data *(blue dots)* that is fixed to $M = 2000$ samples.

Outline

Introduce the forward and inverse problem

Propose inverse maps with dense and convolutional neural networks

Demonstrate parameter estimation capabilities and sensitivities

Extend inference to more parameters and different models

Joint inference of parameters in neuron and noise models

- Joint inference of parameters of (deterministic) physical models and of statistical models is rarely attempted with traditional methods
- Neuron model (ODE): $\frac{\mathrm{d}u}{\mathrm{d}t} = \gamma\left(u - \frac{u^3}{3} + v + \zeta\right), \quad \frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{1}{\gamma}\left(u - \theta_0 + \theta_1 v\right)$
- Noise model (AR): $\eta(t_i) \coloneqq \rho\,\eta(t_{i-1}) + \epsilon(t_i), \quad \eta(t) \sim \mathcal{N}\left(0, \sigma^2/\Delta_t^2\right)$
- Observational data $d(t) = u(t) + \eta(t)$, parameters $\boldsymbol{m} = (\theta_0, \theta_1, \sigma, \rho)$
- Challenge: Orders-of-magnitude difference in time scales in physical models vs. statistical processes

Number of training samples	Data type	ODE parameter		AR parameter	
		θ_0	θ_1	σ	ρ
1000	Time series d	0.914	0.812	−0.62	−0.80
	Fourier spectrum of d	0.460	0.577	0.524	0.603
	Time & Fourier	0.935	0.856	0.589	0.645
8000	Time series d	0.962	0.933	0.627	0.557
	Fourier spectrum of d	0.580	0.797	0.669	0.721
	Time & Fourier	0.968	0.942	0.684	0.722

Results obtained with the **CNN**. Shown is the metric R^2 ($\mathrm{R}^2 = 1$ is ideal).

Estimation of parameter means and covariance matrices

- Estimate local Gaussians $\mathcal{N}(\bar{\boldsymbol{\theta}}, \boldsymbol{C})$, analogously to MAP estimate with Laplace approximation at MAP point
- Neuron model (ODE): $\frac{\mathrm{d}u}{\mathrm{d}t} = \gamma\left(u - \frac{u^3}{3} + v + \zeta\right), \quad \frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{1}{\gamma}\left(u - \theta_0 + \theta_1 v\right)$
- Observational data $d(t) = u(t) + \eta(t)$ with AR noise $\eta(t)$
- Parameters $\boldsymbol{m} = (\bar{\boldsymbol{\theta}}, \boldsymbol{C})$ are mean vector $\bar{\boldsymbol{\theta}} = (\theta_0, \theta_1)$, covariance matrix $\boldsymbol{C}$
- Covariance matrices can be computed by using (incremental) adjoints

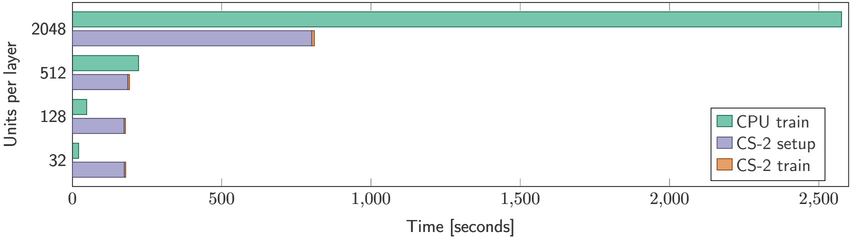
Training & testing data	Median rel. err. $\left\|\bar{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}\right\|_2 / \left\|\bar{\boldsymbol{\theta}}\right\|_2$	Median rel. err. $\left\|\boldsymbol{C} - \hat{\boldsymbol{C}}\right\|_F / \left\|\boldsymbol{C}\right\|_F$
train noise-free, test noise-free	0.032	0.124
train noise-free, test with noise	0.215	0.229
train with noise, test with noise	0.093	0.161

Results obtained with the **CNN**. In $\left\|\bar{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}\right\|_2$, $\bar{\boldsymbol{\theta}}$ is true and $\hat{\boldsymbol{\theta}}$ is predicted.

Training large inverse maps with AI accelerator hardware

Performance comparison between a reference CPU system and Cerebras CS-2 AI accelerator. The runtimes are given for 1,000 epochs (processing of a total of 1,000,000 training samples) and for a **dense NN** with 8 layers (dropout=0.2).

Units per layer		32	128	512	2048
Trainable NN parameters		39,490	243,970	2,352,130	31,428,610
CPU (reference)	Total [sec]	21.0	47.9	221.6	2,576.2
Cerebras CS-2	Total [sec]	178.6	177.6	191.1	810.4
	Setup [sec]	173.7	173.0	185.4	801.5
	Train [sec]	4.9	4.6	5.7	8.9



Thank you

Main reference for this talk

Johann Rudi, Julie Bessac, and Amanda Lenzi (2021). "Parameter estimation with dense and convolutional neural networks applied to the FitzHugh–Nagumo ODE." In: *Proceedings of Mathematical and Scientific Machine Learning (MSML21)*

- **arXiv:** 2012.06691
- **open source code:** `https://github.com/johannrudi/fhn_ode`

References I

Arridge, Simon et al. (2019). "Solving inverse problems using data-driven models." In: *Acta Numerica* 28, pp. 1–174.

Fan, Yuwei, Cindy Orozco Bohorquez, and Lexing Ying (2019). "BCR-Net: A neural network based on the nonstandard wavelet form." In: *Journal of Computational Physics* 384, pp. 1–15. DOI: 10.1016/j.jcp.2019.02.002.

FitzHugh, Richard (1961). "Impulses and physiological states in theoretical models of nerve membrane." In: *Biophysical Journal* 1.6, pp. 445–466.

Khoo, Yuehaw and Lexing Ying (2019). "SwitchNet: A neural network model for forward and inverse scattering problems." In: *SIAM Journal on Scientific Computing* 41.5, A3182–A3201. DOI: 10.1137/18M1222399.

Nagumo, Jinichi, Suguru Arimoto, and Shuji Yoshizawa (1962). "An active pulse transmission line simulating nerve axon." In: *Proceedings of the IRE* 50.10, pp. 2061–2070.

References II

Rudi, Johann, Julie Bessac, and Amanda Lenzi (2021). "Parameter estimation with dense and convolutional neural networks applied to the FitzHugh–Nagumo ODE." In: *Proceedings of Mathematical and Scientific Machine Learning (MSML21)*.