Recovering the parameters underlying the Lorenz-96 chaotic dynamics

Soukayna Mouatadid[1], Pierre Gentine[2], Wei Yu[1]

[1]Department of computer science, University of Toronto, ON, Canada
[2]Department of Earth and Environmental Engineering, Earth Institute, and Data Science Institute, Columbia University, New York, NY, USA

Past and current work

Downscaling	**Seasonal forecasting**
Problem: Understand local climate change	**Problem:** Seasonal and sub-seasonal forecasts predict weather anomalies at monthly and weekly intervals.
ML challenge: Sources of heterogeneity Paucity of labelled data Noise, missing values	**ML challenge:** Long-range dependencies Non-stationarity of extremes under climate change
ML opportunities: Non-uniform LSTMs (Mouatadid et al., 2017) Super-resolution CNNs (Vandal et al., 2017)	**ML opportunities:** Wavelet-LSTM for seasonal forecasts of temperature and drought indices. (Mouatadid et al., 2018, 2019a) Multitask KNN for the sub-seasonal climate forecast rodeo challenge (Hwang et al., 2018)

I will be working on this with Lester this summer

Current work - Parameterization

Problem:

- Climate models need to model relevant physics at fine scales, that are not currently resolved by GCMs.
  Exp: cloud formation, ocean turbulence, land surface heterogeneity, etc.
- Climate scientists use physical intuition + calibration data to come up with approximations of the bulk effect: parameterization schemes.

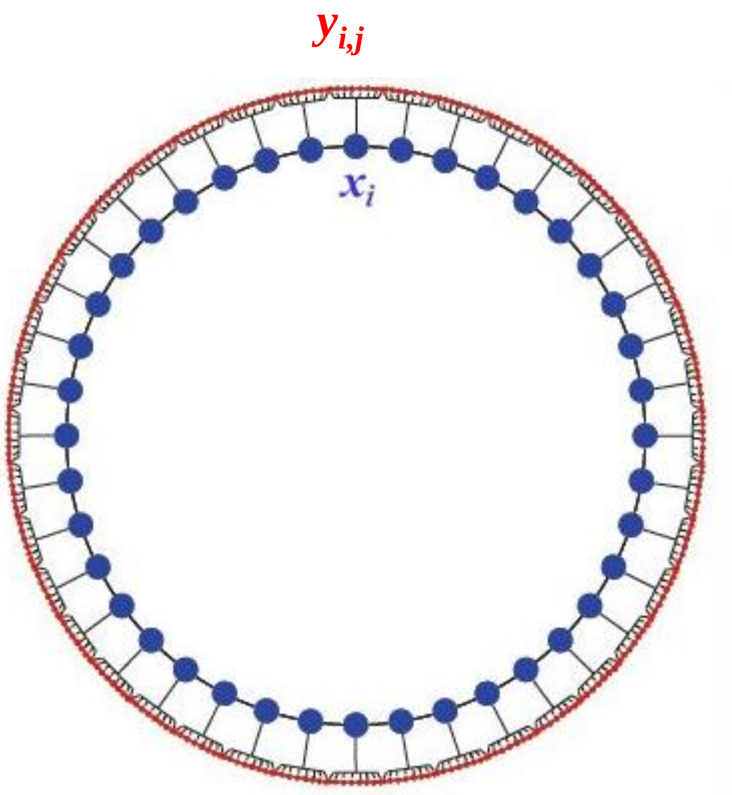
ML challenge:

- Can ML define and automate new parameterization schemes?

ML opportunities:

- NNs to learn from existing schemes (Gentine et al., 2018; Rasp et al., 2018)
- NNs to estimate the underlying parameters of a chaotic system (Mouatadid et al., 2019b)

The Lorenz-96 model



Slow large-scale variables x_i ($i=1,2,\ldots, I$):

$$\frac{dX_i}{dt} = -X_{i-1}(X_{i-2} - X_{i+1}) - X_i + F - \boldsymbol{hc}\bar{Y}_i$$

$$\bar{Y}_i = \frac{1}{J}\sum_{i=1}^{I} Y_{i,j}$$

Fast small-scale variables $y_{i,j}$ ($i=1,2,\ldots,I$; $j=1,2,\ldots,J$):

$$\frac{1}{c}\frac{dY_{i,j}}{dt} = -\boldsymbol{b}Y_{i+1,j}\left(Y_{i+2,j} - Y_{i-1,j}\right) - Y_{i,j} + \frac{h}{j}X_j$$

Lorenz-96 configuration

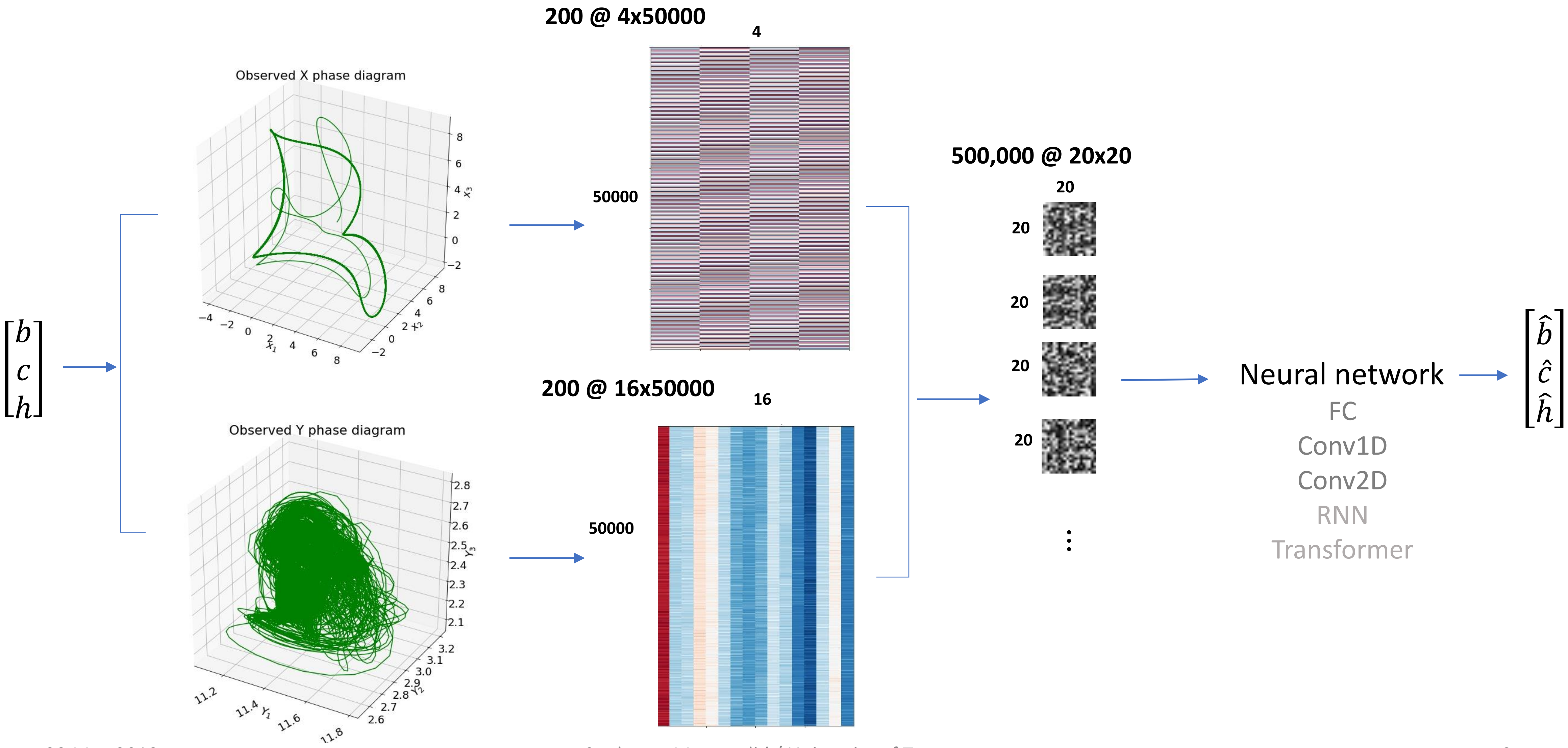
```
def generate_L96():
    import numpy as np
    import os

    K = 4
    J = 4
    F= 8
    b = get_truncated_normal(mean=11, sd=5, low=0, upp=22)
    c = get_truncated_normal(mean=11, sd=5, low=0, upp=22)
    h = get_truncated_normal(mean=1, sd=0.1, low=0, upp=2)
    theta_b, theta_c, theta_h = b.rvs(num_images), c.rvs(num_images), h.rvs(num_images)

    time_step = 0.01
    num_steps = 50000
    burn_in=50

    X = np.zeros(K)
    Y = np.zeros(J * K)
    X[0] = 1
    Y[0] = 1
```

Recovering b, c and h



Loss function and shared aspects

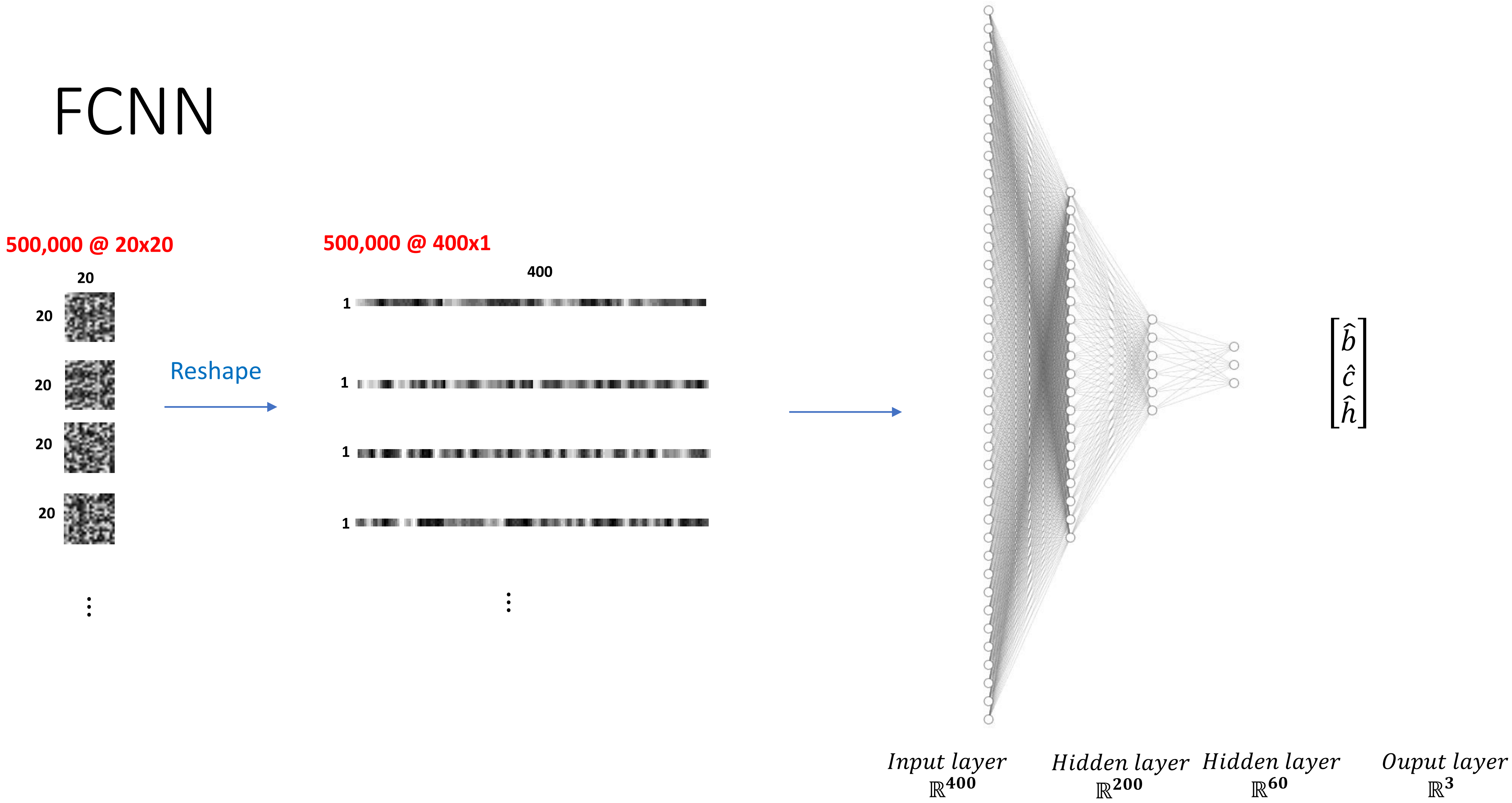
- Loss function:

$$WMSE = \frac{1}{n}\left[\frac{1}{\sigma_b}\sum_{i=1}^{n}(b_i - \hat{b}_i)^2 + \frac{1}{\sigma_c}\sum_{i=1}^{n}(c_i - \hat{c}_i)^2 + \frac{1}{\sigma_h}\sum_{i=1}^{n}(h_i - \hat{h}_i)^2\right]$$

- Non linear activation: LeakyRelu with $\alpha = 0.001$

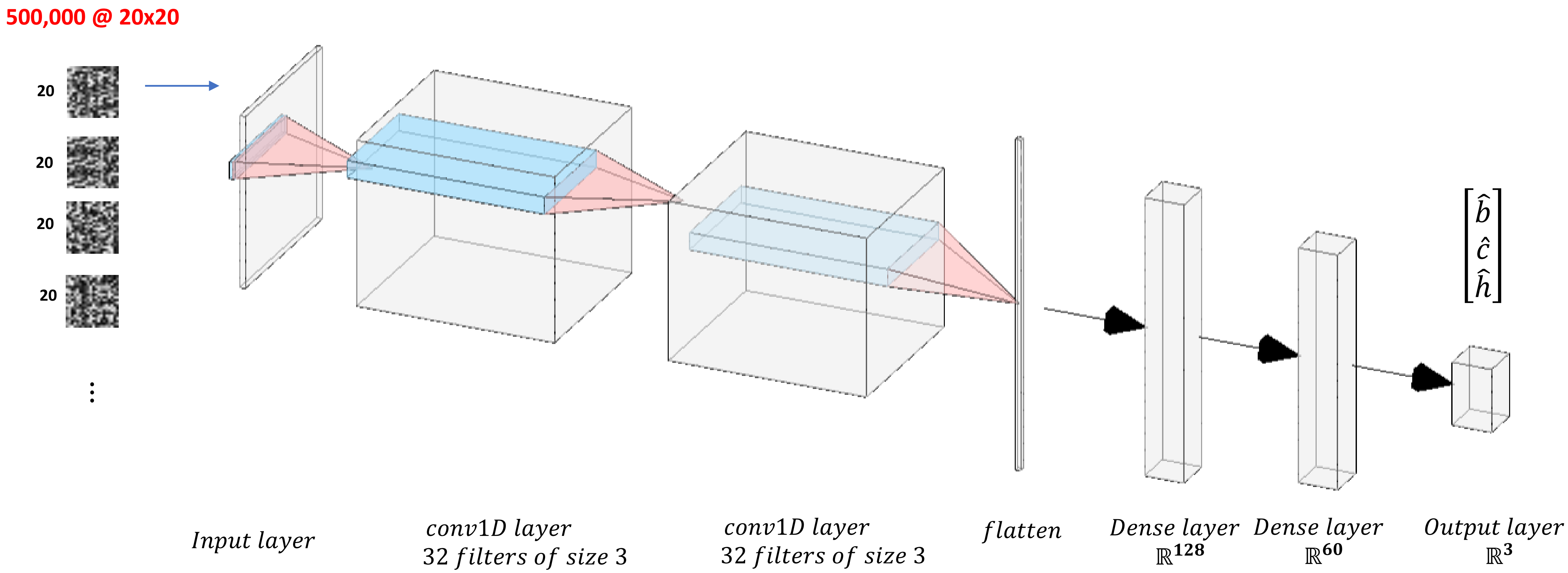
- Optimizer: Adam

FCNN



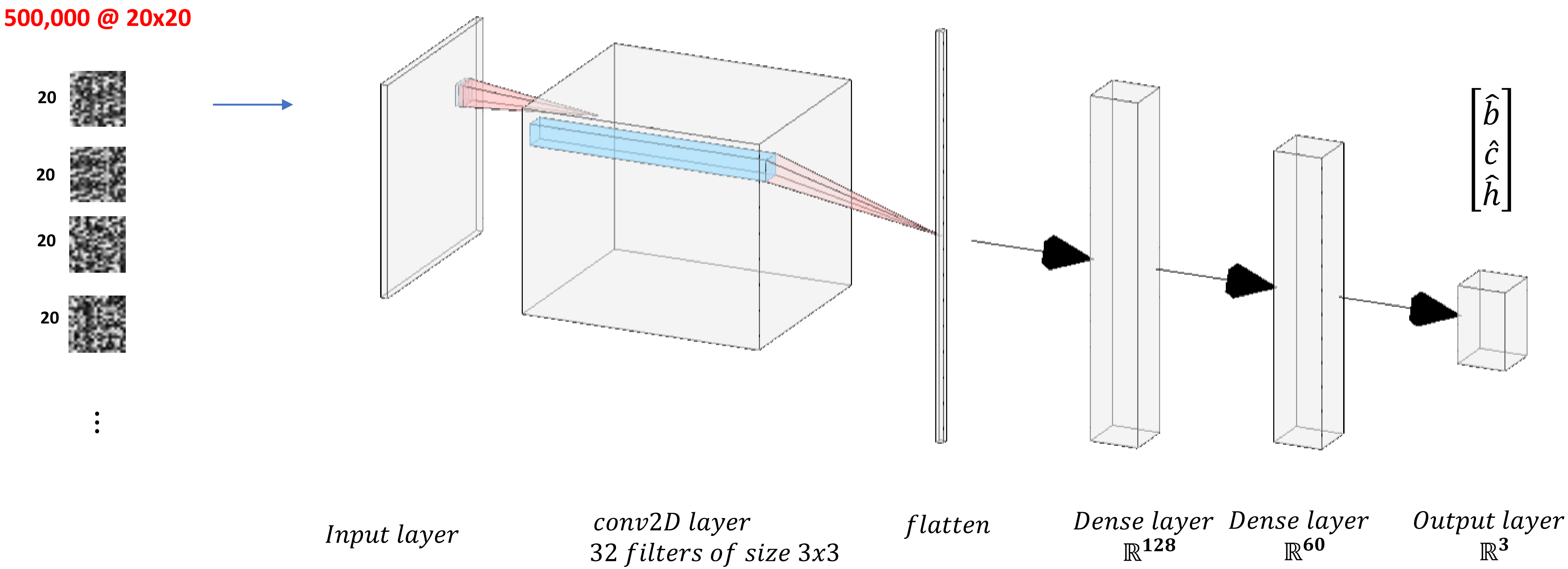
500,000 @ 20x20
20
20
20
20
20
Reshape
500,000 @ 400x1
400
1
1
1
1
Input layer
$\mathbb{R}^{400}$
Hidden layer
$\mathbb{R}^{200}$
Hidden layer
$\mathbb{R}^{60}$
Ouput layer
$\mathbb{R}^{3}$
$\begin{bmatrix}\hat{b}\\ \hat{c}\\ \hat{h}\end{bmatrix}$

CONV1D



500,000 @ 20x20
20
20
20
20
Input layer
conv1D layer
32 filters of size 3
conv1D layer
32 filters of size 3
flatten
Dense layer
$\mathbb{R}^{128}$
Dense layer
$\mathbb{R}^{60}$
Output layer
$\mathbb{R}^{3}$
$\begin{bmatrix} \hat{b} \\ \hat{c} \\ \hat{h} \end{bmatrix}$

CONV2D



500,000 @ 20x20
20
20
20
20
Input layer
conv2D layer
32 filters of size 3x3
flatten
Dense layer
$\mathbb{R}^{128}$
Dense layer
$\mathbb{R}^{60}$
Output layer
$\mathbb{R}^{3}$
$\begin{bmatrix} \hat{b} \\ \hat{c} \\ \hat{h} \end{bmatrix}$

Learning tasks

- Learning from both slow and fast variables
- Learning from fast variables only

- Setting $test\_mode$ to $False$
- Setting $test\_mode$ to $True$

Results

TEST MODE	MODEL	TRAIN LOSS	TEST LOSS	TRAIN r^2	TEST r^2
		LEARNING FROM X AND Y			
	FC	0.6583	0.6714	0.9094	**0.9074**
	CONV1D	0.6682	0.6812	0.9079	**0.9060**
	CONV2D	0.6502	0.6861	0.9105	**0.9054**
False		LEARNING FROM Y ONLY			
	FC	0.6647	0.6808	0.9084	**0.9061**
	CONV1D	0.6968	0.7073	0.9041	**0.9024**
	CONV2D	0.6744	0.7063	0.9071	**0.9026**

TEST MODE	MODEL	TRAIN LOSS	TEST LOSS	TRAIN r^2	TEST r^2
		LEARNING FROM X AND Y			
	FC	0.7064	1.3262	0.9028	**0.8212**
	CONV1D	0.7029	1.2822	0.9031	**0.8263**
	CONV2D	0.6577	1.3260	0.9070	**0.8125**
True		LEARNING FROM Y ONLY			
	FC	0.6805	1.3197	0.9063	**0.8220**
	CONV1D	0.6898	1.2726	0.9050	**0.8276**
	CONV2D	0.6577	1.3260	0.9094	**0.8210**

Results

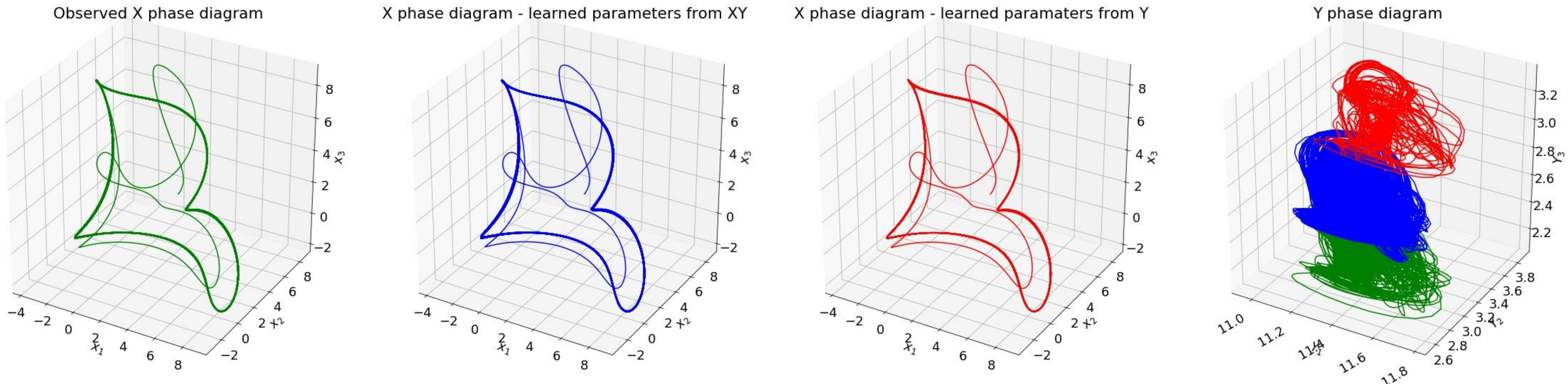


Figure 1. Lorenz-96 phase diagram of the first three slow (X) and fast (Y) variables using observed parameters (green), learned parameters from the X and Y variables (blue) and learned parameters from the Y variables only (red) . The learning algorithm is a fully connected network with *test_mode* set to *False*.

Source: Mouatadid et al., 2019b

Results

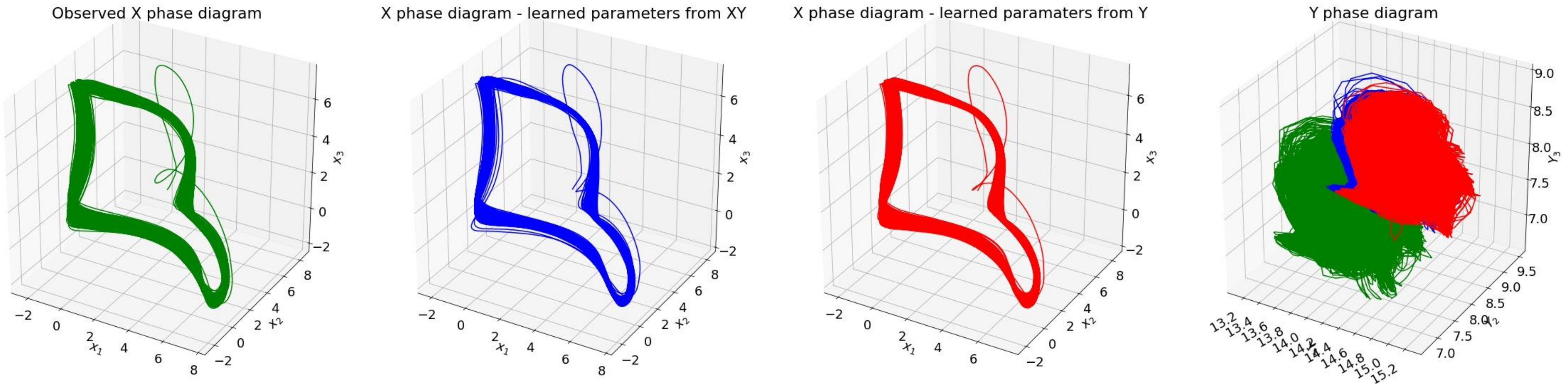


Figure 2. Lorenz-96 phase diagram of the first three slow (X) and fast (Y) variables using observed parameters (green), learned parameters from the X and Y variables (blue) and learned parameters from the Y variables only (red). The learning algorithm is a 1D convolutional model with *test_mode* set to *True.*

Results

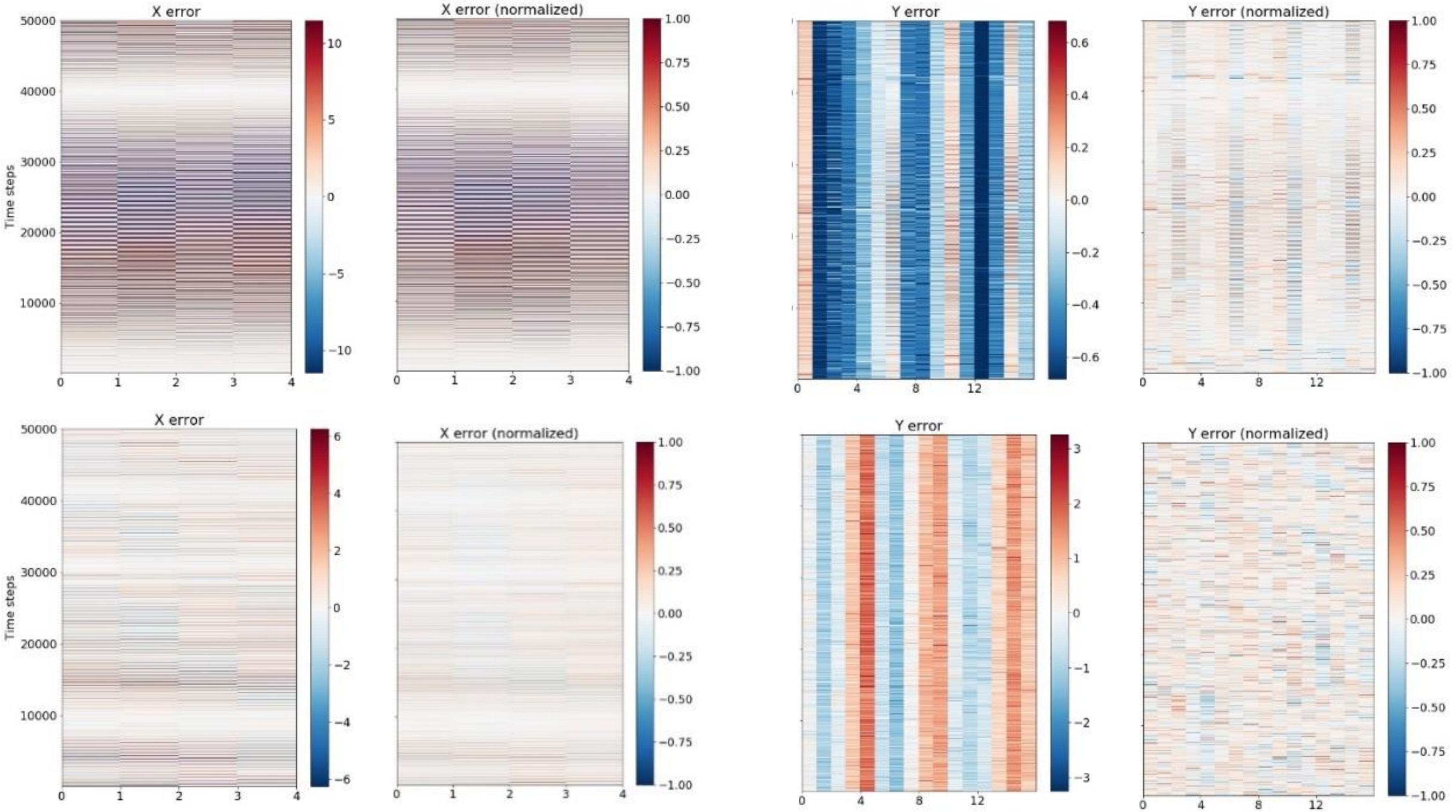


Figure 3.

Errors between the Lorenz-96 slow (X) and fast (Y) variables generated using the observed parameters and the inferred parameters using the FC model trained on the Y variables with *test_mode* set to *False* (top row) and using the Conv1D model trained on Y variables with *test_mode* set to *True* (bottom row).

Discussion/What’s next?

- Why do FCs outperform CNNs?
- (20, 20) image shapes

- Spherical CNNs
- Flexible CNN filters
- Assigning weights to different channels
- Making the network invertible

References

- Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., & Yacalis, G. (2018). Could machine learning break the convection parameterization deadlock?. *Geophysical Research Letters*, *45*(11), 5742-5751.
- Hwang, J., Orenstein, P., Pfeiffer, K., Cohen, J., & Mackey, L. (2018). Improving subseasonal forecasting in the western us with machine learning. *arXiv preprint arXiv:1809.07394*.
- Mouatadid, S., Easterbrook, S., & Erler, A. R. (2017). A machine learning approach to non-uniform spatial downscaling of climate variables. In *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on* (pp. 332-341). IEEE.
- Mouatadid, S., Raj, N., Deo, R. C., & Adamowski, J. F. (2018). Input selection and data-driven model performance optimization to predict the Standardized Precipitation and Evaporation Index in a drought-prone region. *Atmospheric research*, *212*, 130-149.
- Mouatadid, S., Adamowski, J. F., Tiwari, M. K., & Quilty, J. M. (2019a). Coupling the maximum overlap discrete wavelet transform and long short-term memory networks for irrigation flow forecasting. *Agricultural Water Management*, *219*, 72-85.
- Mouatadid, S., Gentine, P., & Yu, W. (2019b). Recovering the parameters underlying the Lorenz-96 chaotic dynamics. ICML climate change workshop.
- Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, *115*(39), 9684-9689.
- Vandal, T., Kodra, E., Ganguly, S., Michaelis, A., Nemani, R., & Ganguly, A. R. (2017, August). Deepsd: Generating high resolution climate change projections through single image super-resolution. In *KDD 2017* (pp. 1663-1672). ACM.