Data Assimilation Project
Lorenz Models State Reconstructions

Simon Driscoll, Charlotte Durand, Anastasia Gorbunova, Oscar Jacquot

December 7

Objective

Goal : from a partially, noisy observed state : find the real state using neural networks

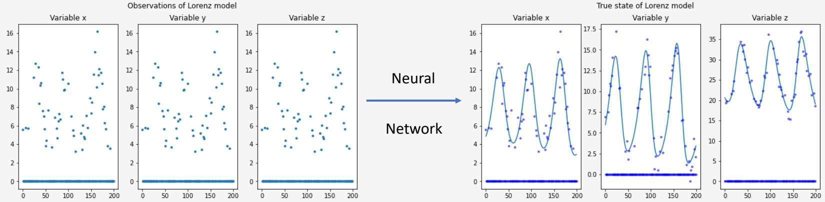


Figure: Reconstruction Objective

Physical model : Lorenz 63

3 ODEs :

$$\frac{dx}{dt} = \sigma(y - x)$$
$$\frac{dy}{dt} = x(\rho - z) - y$$
$$\frac{dz}{dt} = xy - \beta z$$

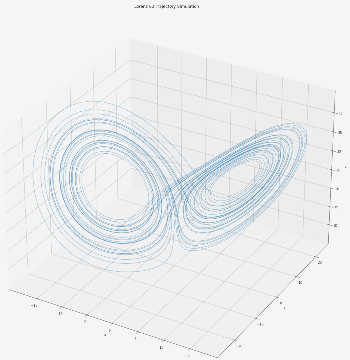


Figure: Simulation of Lorenz 63 model

Physical model : Lorenz 96

40 Variables : for $i = 1 \ldots N$

$$\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F$$

With F = 8, it causes chaotic behaviour

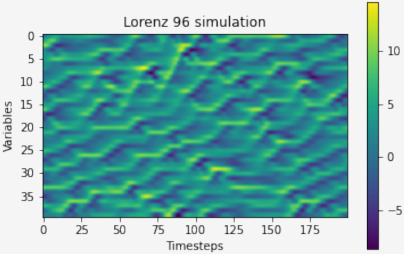
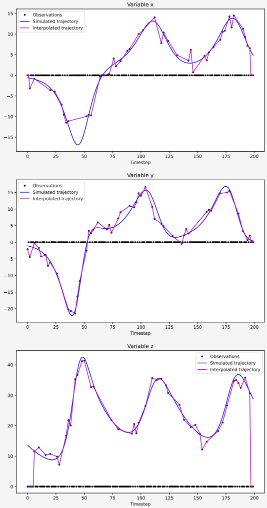


Figure: Simulation of Lorenz 96 model

Framework - Datasets



- 12800 examples for training dataset, 2560 examples for validation and test datasets
- Example of 200 time steps
- Observation frequence of $\frac{1}{2}$
- Sparse observation coefficient between 0 (no observation) and 1 (full observation)
- Noisy observations : $\mathcal{N}(0, \sqrt{2})$
- Possibility to mask complete variables

Framework

3 different neural network to reconstruct

- 'Classical' Data Assimilation scheme : 4D-Var
- Convolutional Neural Network
- 4DVarNet Neural Network[1]

[1] R. Fablet et al. "Learning Variational Data Assimilation Models and Solvers". In: *Journal of Advances in Modeling Earth Systems* 13.10 (2021). e2021MS002572 2021MS002572. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2021MS002572.

4DVar

Cost function minimisation : let's find $\hat{x}$ that minimize J :

$$J = \alpha(\hat{x} - x_{pred})^2 + (1 - \alpha)(\hat{x} - x_{obs})^2$$
$$\alpha = 0.99$$

$\hat{x}$ is found with a gradient descent : use of Pytorch automatic differentiation tools.

4DVar

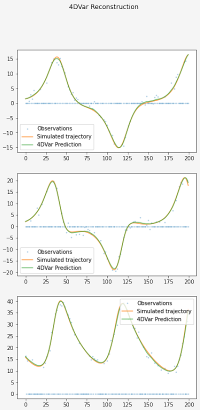


Figure: 4DVar L63

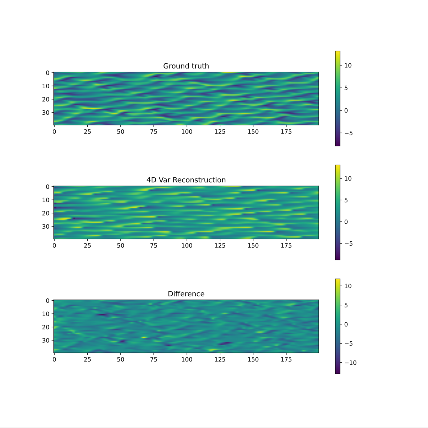
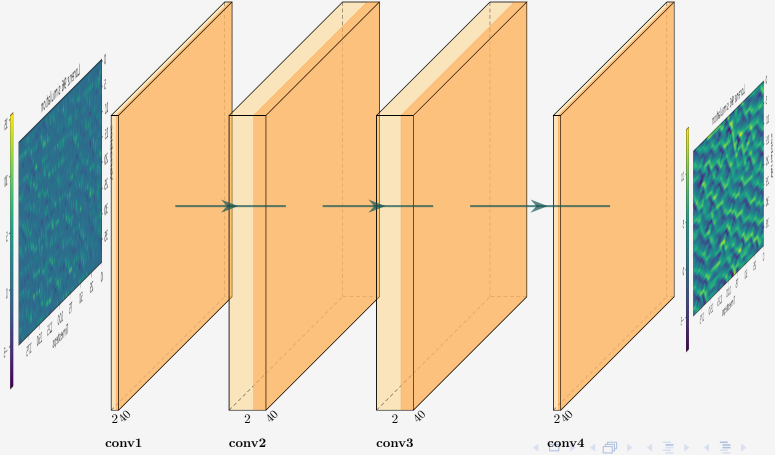


Figure: 4DVar L96

CNN - Convolutional Neural Network

Reconstruction model is learned with a fully convolutional neural network. We can play on the number of layers of the neural network and the kernel of the convolution.



CNN - Convolutional Neural Network

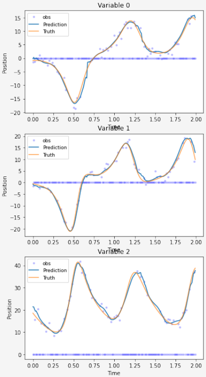


Figure: CNN L63

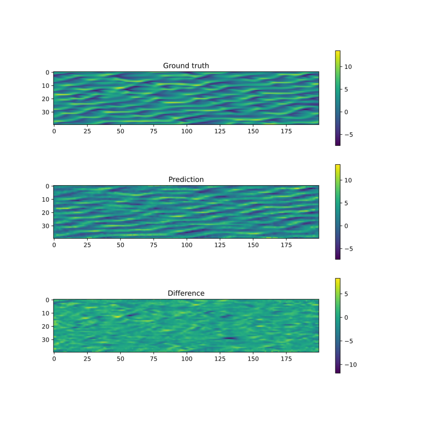


Figure: CNN L96

4DVarNet

To make it simple, we want to minimize the variational cost $U_\Phi(x, y, \Omega)$ such that :

$$U_\Phi(x, y, \Omega) = \lambda_1 \|x - y\|_\Omega^2 + \lambda_2 \|x - \Phi(x)\|^2 \tag{1}$$

With x the observations over the masks Ω and y the ground-truth over the time-window.

Φ is formulated as a Constrained convolutional Neural Network based on an 'U-net' structure (down-sampling and up-sampling)

End-to-end architecture : we also learn the gradient descent to find 'x' as a NN

4DVarNet

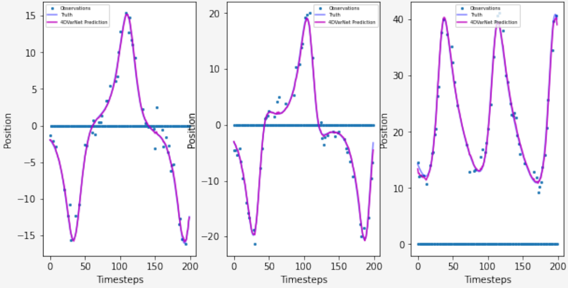


Figure: 4DVarNet L63

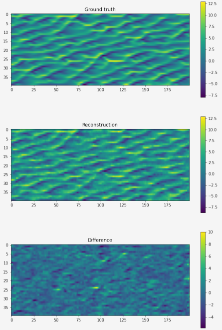


Figure: 4DVarNet L96

Evaluation metrics

To compare our models, we use a Mean Square Error between the ground-truth and the prediction of the neural network : it's a reconstruction score. Example :

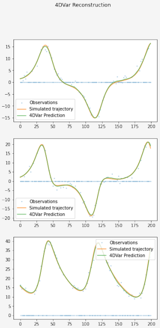


Figure: 4DVarNet L63

Gives a reconstruction score of R-score = 0.327

Comparison

For Lorenz 63

Model	R-Score
4D Var	0.388
4D Var Net	0.327
CNN	1.03

For Lorenz 96

Model	R-Score
4D Var	2.812
4D Var Net	1.486
CNN	1.663

[2]

[2] Note : Best CNN from grid-search : 6 number of layers and 8 kernel size

CNN Grid search

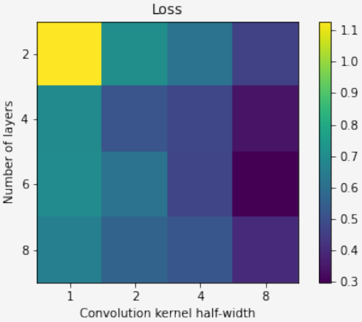
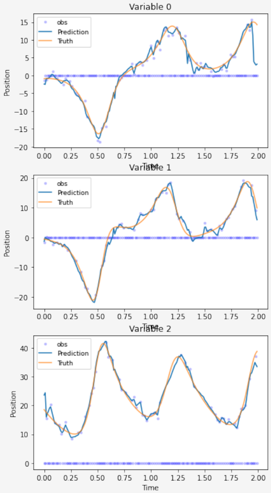


Figure: Grid search for CNN optimisation

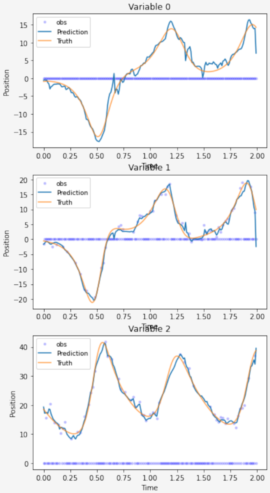
Grid search for CNN optimisation :

- Number of layers :
  2, 4, 6, 8
- Convolution Kernel size :
  1, 2, 4, 8
- Learning rate :
  4 values per power of ten, ranging from 10^{-2} to 10^{-6}

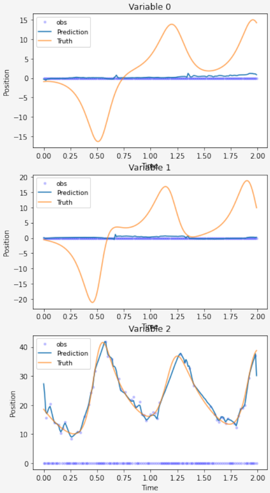
Can we learn the trajectory of some variables without any observations ?



(a) 3 Variables



(b) 2 Variables



(c) 1 Variables

References

[1] R. Fablet et al. "Learning Variational Data Assimilation Models and Solvers". In: *Journal of Advances in Modeling Earth Systems* 13.10 (2021). e2021MS002572 2021MS002572. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2021MS002572.