Learning 4DVAR inversion directly from observations

Arthur Filoche[1], Julien Brajard[2], Anastase Charantonis[3], and Dominique Béréziat[1]

[1] Sorbonne Université, CNRS, LIP6, France
[2] NERSC, Norway
[3] ENSIIE, CNRS, LAMME, France

Abstract. Variational data assimilation and deep learning share many algorithmic aspects. While the former focuses on system state estimation, the latter provides great inductive biases to learn complex relationships. We here design a hybrid architecture learning the assimilation task directly from partial and noisy observations, using the mechanistic constraint of the 4DVAR algorithm. Finally, we show in an experiment that the proposed method was able to learn the desired inversion with interesting regularizing properties and that it also has computational interests.

Keywords: Data Assimilation · Unsupervised Inversion · Differentiable Physics

1 Introduction

Data Assimilation [5] is a set of statistical methods solving particular inverse problems, involving a dynamical model and imperfect data obtained through an observation process, with the objective to estimate a considered system state. It produces state-of-the-art results in various numerical weather prediction tasks and is mostly used in operational meteorological centers.

Although they are not initially designed for the same purpose, variational data assimilation [11] and deep learning share many algorithmic aspects [1]. It has already been argued that both methods can benefit from each other [17, 10]. Data assimilation provides a proper Bayesian framework to combine sparse and noisy data with physics-based knowledge while deep learning can leverage a collection of data extracting complex relationships from it. Hybrid methods have already been developed either to correct model error [8, 6], to jointly estimate parameters and system state [4, 14] or to fasten the assimilation process [20]. Most of these algorithms rely on iterative optimization schemes alternating data assimilation and machine learning steps.

In this work we design a hybrid architecture bridging a neural network and a mechanistic model to directly learn system state estimation from a collection of partial and noisy observations. We optimize it in only one step still using the variational assimilation loss function. Finally, We show in an experiment

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_32

using the chaotic Lorenz96 dynamical system, that the proposed method is able to learn the variational data assimilation with desirable regularizing properties, then providing a computationally efficient inversion operator.

2 Related work

Hybridizing data assimilation with machine learning. While deep learning has proven to be extremely useful for a variety of inverse problems where the ground truth is available, unsupervised inversion is still being investigated [15]. For instance, when data are highly-sparse, neural architectures may be hard to train. On the other hand, data assimilation can provide dense data. From this statement, approaches have naturally emerged in the data assimilation community, iterating data assimilation steps and machine learning steps for simultaneous state and parameters estimation [4, 14]. But end-to-end learning approaches are also investigated, in [7] the architecture is constrained to internally behave like a 4DVAR pushing the hybridization further.

Mechanistically constrained neural networks. Variational data assimilation has a pioneering expertise in PDE-constrained optimization [11], making use of automatic differentiation to retro-propagate gradients through the dynamical system. In [3, 13] the output of a neural network is used as input in a dynamical model, and architectures are trained with such gradients, in a supervised and adversarial manner, respectively. Similar methods have been used to learn accurate numerical simulations still using differentiable mechanistic models [18, 19]. Also, Physically-consistent architectures are developed to enforce the conservation of desired quantity by neural architectures [2].

3 Data assimilation and learning framework

3.1 State-space system

A system state $\mathbf{X}_t$ evolves over time according to a considered perfectly known dynamics $\mathbb{M}_t$ and observations $\mathbf{Y}_t$ are obtained through an observation operator $\mathbb{H}$ up to an additive noise ε_{R_t}, as described in Eqs. 1 and 2,

$$\text{Dynamics:} \qquad \mathbf{X}_{t+1} = \mathbb{M}_t(\mathbf{X}_t) \tag{1}$$

$$\text{Observation:} \qquad \mathbf{Y}_t = \mathbb{H}_t(\mathbf{X}_t) + \varepsilon_{R_t} \tag{2}$$

We denote the trajectory $\mathbf{X} = [\mathbf{X}_0, \ldots, \mathbf{X}_T]$, a sequence of state vectors over a temporal window, and $\mathbf{Y}$ the associated observations. The objective of data assimilation is to provide an estimation of the posterior probability $p(\mathbf{X} \mid \mathbf{Y})$ leveraging the information about the mechanistic model $\mathbb{M}$. The estimation can later be used to produce a forecast.

3.2 The initial value inverse problem

When considering the dynamics perfect, the whole trajectory only depends on the initial state $\mathbf{X}_0$, the assimilation is then said with strong-constraint. The whole process to be inverted is summed up in the simple Eq. 3, where $\mathcal{F}$ is the forward model, combining $\mathbb{M}_t$ and $\mathbb{H}_t$. More precisely, by denoting multiple model integrations between two times $\mathbb{M}_{t_1 \to t_2}$, we can rewrite the observation equation as in Eq. 4.

$$\mathbf{Y} = \mathcal{F}(\mathbf{X}_0) + \varepsilon_R \tag{3}$$

$$\mathbf{Y}_t = \mathbb{H}_t \circ \mathbb{M}_{0 \to t}(\mathbf{X}_0) + \varepsilon_{R_t} \tag{4}$$

The desired Bayesian estimation now requires a likelihood model $p(\mathbf{X} \mid \mathbf{Y})$ and a prior model $p(\mathbf{X}) = p(\mathbf{X}_0)$. We assume the observation errors uncorrelated in time so that $p(\mathbf{X} \mid \mathbf{Y}) = \prod_t p(\varepsilon_{R_t})$ and we here make no particular assumption on $\mathbf{X}_0$ corresponding to a uniform prior.

3.3 Variational assimilation with 4DVAR

The solve this problem in a variational manner, it is convenient to also assume white and Gaussian observational errors ε_{R_t}, of known covariance matrices $\mathbf{R}_t$, leading to the least-squares formulation given in Eqs. 5, where $\|\varepsilon_{R_t}\|^2_{R_t}$ stands for the Mahalanobis distance associated with the matrix $\mathbf{R}_t$. The associated loss function is denoted $\mathcal{J}_{4DV}$, Eq. 6, and minimizing it corresponds to a maximum a posteriori estimation, here equivalent to a maximum likelihood estimation.

$$-\log p(\mathbf{X} \mid \mathbf{Y}) = \frac{1}{2} \sum_{t=0}^{T} \|\varepsilon_{R_t}\|^2_{\mathbf{R}_t} - \log K \quad \text{s.t.} \quad \mathbb{M}(\mathbf{X}_t) = \mathbf{X}_{t+1} \tag{5}$$

$$\mathcal{J}_{4DV}(\mathbf{X}_0) = \frac{1}{2} \sum_{t=0}^{T} \|\mathbb{H}_t \circ \mathbb{M}_{0 \to t}(\mathbf{X}_0) - \mathbf{Y}_t\|^2_{\mathbf{R}_t} \tag{6}$$

This optimization is an optimal control problem where the initial state $\mathbf{X}_0$ plays the role of control parameters. Using the adjoint state method, we can derive an analytical expression of $\nabla_{\mathbf{X}_0} \mathcal{J}_{4DV}$ as in Eq. 7. It is worth noting that the mechanism at stake here is equivalent to the back-propagation algorithm used to train neural networks. The algorithm associated with this optimization is named 4DVAR.

$$\nabla_{\mathbf{X}_0} \mathcal{J}_{4DV}(\mathbf{X}_0) = \sum_{t=0}^{T} \left[\frac{\partial (\mathbb{H}_t \circ \mathbb{M}_{0 \to t})}{\partial \mathbf{X}_0} \right]^\top {\mathbf{R}_t}^{-1} \varepsilon_{R_t} \tag{7}$$

3.4 Learning inversion directly from observations

We now consider independent and identically distributed trajectories denoted and the dataset of observations $\mathcal{D} = \{\mathbf{Y}^{(i)}, \mathbf{R}^{-1(i)}\}_{i=1}^{N}$. The associated ground

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_32

truth $\mathcal{T} = \{\mathbf{X}^{(i)}\}_{i=1}^{N}$ is not available so the supervised setting is not an option. The posteriors for each trajectory are then also independent as developed in this equation: $\log p(\mathcal{T} \mid \mathcal{D}) = \sum_{i=0}^{N} \log p(\mathbf{X}^{(i)} \mid \mathbf{Y}^{(i)})$.

Our objective is to learn a parameterized pseudo-inverse $\mathcal{F}^{\star}_{\boldsymbol{\theta}} : (\mathbf{Y}, \mathbf{R}^{-1}) \mapsto \mathbf{X}_0$ that should output initial condition from observations and associated errors covariance, which is exactly the task solved by 4DVAR. Such modeling choice corresponds to the prior $p(\mathbf{X}_0) = \delta(\mathbf{X}_0 - \mathcal{F}^{\star}_{\boldsymbol{\theta}}(\mathbf{Y}, \mathbf{R}^{-1}))$, as we do not use additional regularization, where δ is the Dirac measure.

To learn the new control parameters $\boldsymbol{\theta}$, we leverage the knowledge of the dynamical model $\mathbb{M}$ as in 4DVAR. After outputting the initial condition $\mathbf{X}_0$ we forward it with the dynamical model and then calculate the observational loss. A schematic view of the performed integration is drawn in Fig. 1.

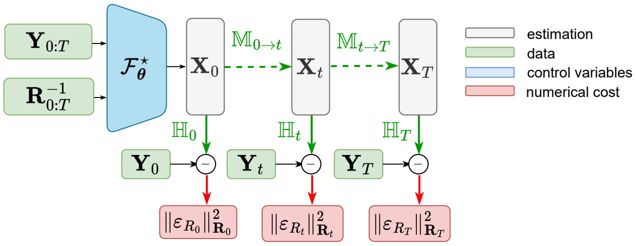

Fig. 1. Schematic view of the hybrid architecture learning the 4DVAR inversion

Then the cost function associated with the MAP estimation can be developed as in Eq. 8. A simple way of thinking it is to run multiple 4DVAR in parallel to optimize a common set of control parameters $\boldsymbol{\theta}$.

$$\mathcal{J}(\boldsymbol{\theta}) = \sum_{\mathcal{D}} \mathcal{J}_{4DV}(\mathbf{X}_0^{(i)}) \quad \text{s.t. } \mathcal{F}_{\boldsymbol{\theta}}(\mathbf{Y}^{(i)}, \mathbf{R}^{-1(i)}) = \mathbf{X}_0^{(i)} \tag{8}$$

To calculate $\nabla_{\boldsymbol{\theta}}\mathcal{J}$ we simply use the linearity of the gradient then the chain rule (Eq. 9) and finally we can re-use $\nabla_{\mathbf{X}_0}\mathcal{J}_{4DV}$ calculated before (Eq. 7). Gradients are back-propagated through the dynamical model first and then through the parameterized pseudo-inverse. Calculating the gradient on the whole dataset at each iteration may be computationally too expensive so one could instead use mini-batch gradient descent.

$$\nabla_{\boldsymbol{\theta}}\mathcal{J} = \sum_{\mathcal{D}} \nabla_{\boldsymbol{\theta}}\mathcal{J}_{4DV} = \sum_{\mathcal{D}} \nabla_{\mathbf{X}_0}\mathcal{J}_{4DV}\nabla_{\boldsymbol{\theta}}\mathbf{X}_0 = \sum_{\mathcal{D}} \nabla_{\mathbf{X}_0}\mathcal{J}_{4DV}\nabla_{\boldsymbol{\theta}}\mathcal{F}^{\star}_{\boldsymbol{\theta}} \tag{9}$$

4 Experiments and Results

4.1 Lorenz96 dynamics and observations

We use the Lorenz96 dynamics [12] as an evolution model Lorenz96, $\frac{d\mathbf{X}_{t,n}}{dt} = (\mathbf{X}_{t,n+1} - \mathbf{X}_{t,n-2})\mathbf{X}_{t,n-1} - \mathbf{X}_{t,n} + F$, numerically integrated with a fourth-order

Runge Kutta scheme. Here n indexes a one-dimensional space. On the right-hand side, the first term corresponds to an advection, the second term represents damping and F is an external forcing. We use the parameters $dt = 0.1$ and $F = 8$ corresponding to a chaotic regime [9]. Starting from white noise and after integrating during a spin-up period to reach a stationary state, we generate ground truth trajectories.

To create associated observations, we use a randomized linear projector as an observation operator, making the observation sparse to finally add a white noise. Noises at each point in time and space can have different variances, $\varepsilon_{R_{n,t}} \sim \mathcal{N}(0, \sigma_{n,t})$, and we use the associated diagonal variance matrix defined by $\mathbf{R}^{-1}_{n,t} = \frac{1}{\sigma^2_{n,t}}$. Figure 2 displays an example of simulated observations. Variances are sampled uniformly such that $\sigma_{n,t} \sim \mathcal{U}(0.25, 1)$. When a point in the grid is not observed we fix "$\mathbf{R}^{-1}_{n,t} = 0$", which corresponds to an infinite variance meaning a lack of information. From a numerical optimization view, no cost means no gradient back-propagated which is the desired behavior.

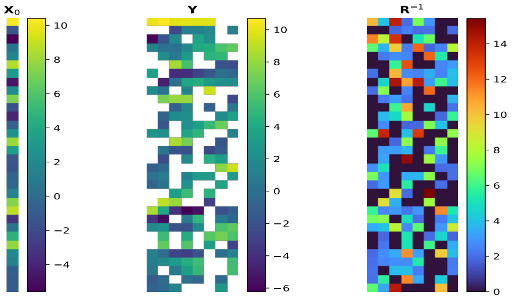

Fig. 2. Observation generated with the Lorenz96 model, a randomized linear projector as observation operator, and a white noise.

4.2 Algorithm benchmarks

We evaluate our method (NN-4DVAR-e2e) on the assimilation task which is estimating $\mathbf{X}_0$. We compare it with a 4DVAR, a 4DVAR with additional $\mathcal{L}_2$ regularization (4DVAR-B), a neural network trained on the output of both 4DVAR estimations (NN-4DVAR-iter and NN-4DVAR-B-iter), and a neural network trained with the ground truth (NN-perfect). The latter should represent the best-case scenario for the chosen architecture while NN-4DVAR-iter plays the role of the iterative method. The same neural architecture is used for all the methods involving learning. Its design is fairly simple, being composed of 5 convolutional layers using 3×3 kernels, ReLu activation, no down-scaling, and a last layer flattening the two-dimensional maps into the shape of $\mathbf{X}_0$. We use 250, 50, and 250 samples for training, validation, and testing, respectively. When learning is involved, the Adam optimizer is used while 4DVAR is optimized with the L-BFGS solver. We notice here that once learned, both NN-4DVAR-iter and NN-4DVAR-e2e provide a computationally cheap inversion operator. For their learning, the computationally intensive step was the forward integration of the dynamical model. Denoting n_iter the number of iterations done in 4DVAR and n_epoch the number of epochs in our learning process, NN-4DVAR-iter, and 4DVAR-e2e cost $N \times n\_iter$ and $N \times n\_epoch$ dynamics integration, respectively. Depending on these parameters, one approach or the other will be less computationally intensive. In our case, we used $n\_iter < 150$ and $n\_epoch = 50$.

4.3 Results

The accuracy of the $\mathbf{X}_0$ estimation on the test set is quantified using the RMSE and the average bias (see Figs 3). We notice first that when 4DVAR is not regularized, some samples induce bad estimations which disturb 4DVAR-NN-iter learning over them. The others methods involving produce RMSE scores on par with 4DVAR-B, the best estimator. However, our 4DVAR-NN-e2e is the less biased algorithm. It is to be noted that 4DVAR-NN-e2e has no additional regularization and still stays robust regarding difficult samples, highlighting desirable properties from the neural architecture.

5 Conclusion

We proposed a hybrid architecture inspired by the 4DVAR algorithm allowing to use of the data assimilation Bayesian framework while leveraging a dataset to learn an inversion operator. We showed in an assimilation experiment that the algorithm was able to desired function while having a stable behavior.

The designed algorithm fixes the maximum temporal size of the assimilation window. For smaller windows, it can still be used filling the masking variance with zeros accordingly but for larger ones, the only possibility is to use sliding windows, then raising to question of the coherence in time. Typically, the method in that form can not fit quasi-static strategies [16] employed in variational assimilation. Also, We made the convenient hypothesis that observational errors are uncorrelated in space, so that $\mathbf{R}^{-1}$ can be reshaped in the observation format, which may not be the case depending on the sensors. However, the method has a computational interest. Once the parameterized inversion operator learned, the inversion task becomes computationally cheap. But this also stands for the iterative approaches. As discussed before, learning the inversion directly with our method may be less computationally costly, in terms of dynamics integration, depending on the number of epochs when learning our architecture, the number of samples in the dataset, and the number of iterations used in 4DVAR.

One of the motivations for the designed architecture was to circumvent algorithms iterating data assimilation and machine learning steps, because of their difficulty of implementation but also their potential bias as exhibited in the experiment. However, we made the debatable, simplifying, perfect model hypothesis. Usually, the forward operator is only partially known and we ambition

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36027-5_32

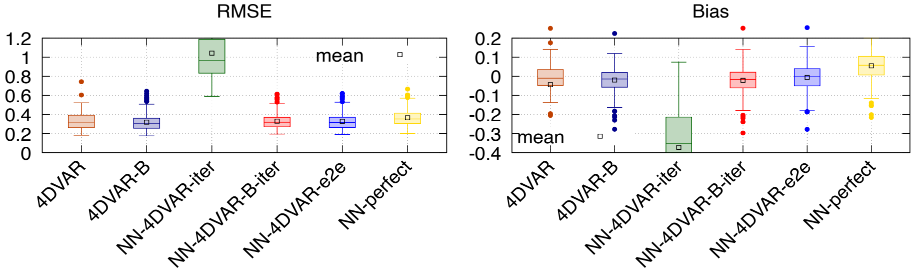

Fig. 3. Boxplot of assimilation accuracy of each algorithm, RMSE and Bias scores, on the 250 samples of the test set

to develop the proposed framework further to relax such a hypothesis. In Fig. 4, we performed an accuracy sensitivity experiment regarding noise and sparsity levels. Particularly, we tested noise levels out of the dataset distribution. We see that learning-based approaches are more sensitive to noise increases while 4DVAR is more concerned by sparsity. Also, we notice that our NN-4DVAR-e2e methods generalize better than NN-4DVAR-B-iter to unseen levels of noise.

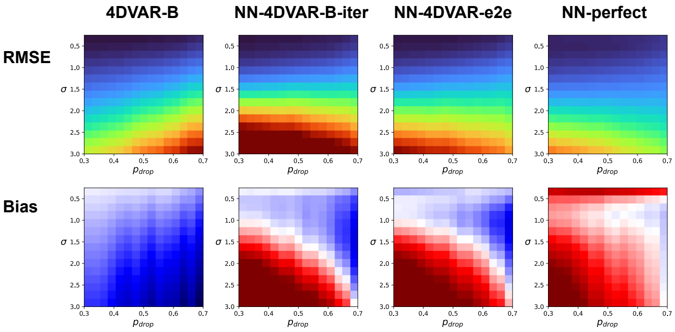

Fig. 4. Sensitivity of the assimilation regarding noise and sparsity levels (σ, p_{drop}). At each pixel, levels are constant and scores are averaged on 25 samples. $\sigma > 1$ not seen in training.

References

1. Abarbanel, H., Rozdeba, P., Shirman, S.: Machine learning: Deepest learning as statistical data assimilation problems. Neural Computation **30**(8), 2025–2055 (2018)
2. Beucler, T., Pritchard, M., Rasp, S., Ott, J., Baldi, P., Gentine, P.: Enforcing analytic constraints in neural networks emulating physical systems. Physical Review Letters **126**(9), 1079–7114 (Mar 2021)

3. de Bézenac, E., Pajot, A., Gallinari, P.: Deep learning for physical processes: Incorporating prior scientific knowledge. Journal of Statistical Mechanics: Theory and Experiment **2019**(12), 124009 (2019)
4. Bocquet, M., Brajard, J., Carrassi, A., Bertino, L.: Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization. Foundations of Data Science **2**(1), 55–80 (2020)
5. Carrassi, A., Bocquet, M., Bertino, L., Evensen, G.: Data assimilation in the geosciences: An overview of methods, issues, and perspectives. Wiley Interdisciplinary Reviews: Climate Change **9**(5), e535 (2018)
6. Düben, P., Modigliani, U., Geer, A., Siemen, S., Pappenberger, F., Bauer, P., Brown, A., Palkovic, M., Raoult, B., Wedi, N., et al.: Machine learning at ECMWF: A roadmap for the next 10 years. ECMWF Technical Memoranda **878** (2021)
7. Fablet, R., Amar, M., Febvre, Q., Beauchamp, M., Chapron, B.: End-to-end physics-informed representation learning for satellite ocean remote sensing data: Applications to satellite altimetry and sea surface currents. ISPRS (2021)
8. Farchi, A., Laloyaux, P., Bonavita, M., Bocquet, M.: Using machine learning to correct model error in data assimilation and forecast applications. Quarterly Journal of the Royal Meteorological Society **147**(739), 3067–3084 (2021)
9. Fertig, E., Harlim, J., Ramon, H.: A comparative study of 4D-VAR and a 4D Ensemble Kalman Filter: Perfect model simulations with Lorenz-96. Tellus (2007)
10. Geer, A.: Learning earth system models from observations: machine learning or data assimilation? Phil. Trans. of the Royal Society A **379** (Feb 2021)
11. Le Dimet, F.X., Talagrand, O.: Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. Tellus A **38**(10), 97 (1986)
12. Lorenz, E.: Predictability: a problem partly solved. In: Seminar on Predictability. vol. 1, pp. 1–18. ECMWF (Sep 1995)
13. Mosser, L., Dubrule, O., Blunt, M.: Stochastic seismic waveform inversion using generative adversarial networks as a geological prior. Mathematical Geoscience pp. 53–79 (2018)
14. Nguyen, D., Ouala, S., Drumetz, L., Fablet, R.: Assimilation-based Learning of Chaotic Dynamical Systems from Noisy and Partial Data. In: ICASSP (2020)
15. Ongie, G., Jalal, A., Metzler, C., Baraniuk, R., Dimakis, A., Willett, R.: Deep learning techniques for inverse problems in imaging. IEEE Journal on Selected Areas in Information Theory **1**(1), 39–56 (2020)
16. Pires, C., Vautard, R., Talagrand, O.: On extending the limits of variational assimilation in nonlinear chaotic systems. Tellus A **48**, 96–121 (1996)
17. Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., Prabhat: Deep learning and process understanding for data-driven earth system science. Nature **566**(7743), 195–204 (2019)
18. Tompson, J., Schlachter, K., Sprechmann, P., Perlin, K.: Accelerating eulerian fluid simulation with convolutional networks. In: ICML. pp. 5258–5267 (2017)
19. Um, K., Brand, R., Fei, Y., Holl, P., Thuerey, N.: Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers. Advances in Neural Information Processing Systems (2020)
20. Wu, P., Chang, X., Yuan, W., Sun, J., Zhang, W., Arcucci, R., Guo, Y.: Fast data assimilation (FDA): Data assimilation by machine learning for faster optimize model state. Journal of Computational Science **51**, 101323 (2021)