



Informative Ensemble Kalman Learning for Neural Structure

Margaret Trautner^{1,2}, Gabriel Margolis^{1,3}, and Sai Ravela¹(✉)

¹ Earth Signals and Systems Group, Massachusetts Institute of Technology,
Cambridge, USA
ravela@mit.edu

² Department of Mathematics, Massachusetts Institute of Technology,
Cambridge, USA

³ Department of Aeronautics and Astronautics,
Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract. We characterize the stochastic dynamics of Neural Learning to develop Informative Ensemble Kalman Learning. Specifically, an adaptive Ensemble Kalman Filter replaces backpropagation to quantify uncertainty and maximize information gain during Learning. Demonstrating competitive performance on standard datasets, we show Structure Learning using the Informative Ensemble Kalman Learner quickly recovers the dynamical equations of the Lorenz-63 system *ab initio* from data. Results indicate that extending DDDAS key informative paradigm to optimize Learning Systems is promising.

Keywords: Deep learning · Ensemble Kalman Filter · Informative learning · Systems dynamics and optimization

1 Introduction

The use of data to dynamically control an executing model and, conversely, using the model to control the instrumentation process is a central tenet of Dynamic Data Driven Applications Systems. Applications such as Cooperative Autonomous Observing Systems (CAOS) embody this paradigm [1] in a stochastic system's dynamics and optimization (SDO) loop, maximizing information gain from model predictions to plan observations or select sensors for improved model estimation or reduction. Various called informative-planning, -estimation, or -sensing, this approach improves nonlinear, high-dimensional stochastic process modeling and prediction, including systems with epistemic uncertainties.

Learning machines are in demand as surrogate or hybrid models for SDO, but SDO is rarely applied to Machine Learning. In particular, Neural Learning dynamics are also stochastic, nonlinear, and high-dimensional. Could DDDAS'

Support from ONR grant N00014-19-1-2273 and VTSIX INC are gratefully acknowledged.

informative approach help machines learn better or faster? In addition to improving CAOS, we posit doing so would broadly impact Machine Learning.

From a systems perspective, training a neural network is parameter estimation [14], and backpropagation [10] restates the variational solution to multistage two-point boundary value problems [2] (2BVP). In practice, Learning implies stochastic dynamics (e.g., due to mini-batches) and the associated Fokker-Planck equations describe the evolution of parameter uncertainty. Much like in nonlinear dynamics, an ensemble approximation to the Fokker-Planck, e.g., the Ensemble Kalman Filter [3], can train a neural network. The benefits include adjoint-free Learning, parallelism, and quantified uncertainty.

The uncertainty quantification benefit implies that one could optimize Learning by maximizing information gain between the training error distribution and key variables. Doing so unifies several paradigms, including parameter selection (Feature/Model Selection), input selection (Relevance and Active Learning), and term/variable selection (Structure Learning). Thus, by analogy to informative approaches in DDDAS, *Informative Learning is the coupling of stochastic learning dynamics with maximization of information gain.*

Although this paradigm is not new from a DDDAS perspective, we cannot ignore the unifying formalism or practical benefits from a Machine Learning perspective. Due to space limitations, we refer the reader elsewhere [14] for details of the stochastic methodology and learning paradigms. Here, we focus on a few critical numerical examples.

First, we show that an adaptive version of the Ensemble Kalman Learner is competitive with backpropagation [10] on standard datasets. Second, we use Informative Ensemble Kalman Learning to learn the structure and parameters of a neural dynamical system [15]. We use numerical simulations of the Lorenz system [6] as training data to demonstrate successful generalization, extrapolation, and interpretation. We can do this because numerical solutions of dynamical systems with polynomial nonlinearities have exact Neural Networks [15]. Thus, neural structure learning reveals the dynamical equations (terms and coefficients). We show that the Informative Learning approach rapidly learns the Lorenz-63 equations to numerical accuracy, *ab initio*, from simulated data. It is both uncertainty aware and sparsity promoting but without iterative burdens, which is an exciting result.

The rest of this paper is as follows. Section 2 describes related work. Section 3 describes Ensemble Kalman Learning, and Sect. 4 uses it for Informative Structure Learning.

2 Related Work

There are connections between Informative Learning and Active Learning [13]. However, the latter does not embody a stochastic dynamical perspective. Informative Learning is related to applications in planning and estimation [1]; Learning has scarcely seen such methodology applied. The application of the Ensemble Kalman Filter [3] to Learning has received scant interest so far. However, adaptive Ensemble Kalman Learning proposed here offers competitive performance,

and Informative Ensemble Kalman Learning is new. Ensemble Kalman Learning is itself related to Bayesian Deep Learning [16], but it remains distinct from the extant methodology that typically emphasizes variational Bayesian approaches or Bayesian Active Learning [13].

The stochastic dynamics of Learning naturally form a Markov chain [11]. Stochastic gradient descent using Kalman-SGD [7] and Langevin dynamics [17] have both been developed, but these are unrelated to our work. Finally, learning physics from data [8] is receiving some attention, but our proposed approach is new. Neural structure optimization has also received some attention [18], but our approach still appears to be novel. Note that the presented Informative Ensemble Kalman Learning paradigm is applicable wherever backpropagation is. In fact, it can be broadly applied to other learning systems as well.

3 Ensemble Kalman Learning

In this section, we describe a framework for Informative Learning briefly, referring the reader to details elsewhere [14]. Let us define a standard Neural Network as a N -stage process [2]:

$$x_{l+1} = F_{l+1}(x_l, u_l; \alpha_l) \quad y_N = x_N + \nu_N \quad 0 \leq l < N \quad (1)$$

where $x_l \in \mathbb{R}^{n_l, x}$ are the layer l nodes, F_l is the function, $\alpha_l \in \mathbb{R}^{n_l, \alpha}$ are the weights and biases, $u_l \in \mathbb{R}^{n_l, u}$ represents feed-forward (e.g. ResNet) or feed-back terms (e.g. recurrent network). The vector $y_N \in \mathbb{R}^{n_y}$ refers to (imperfect) training outputs with additive noise $\nu_l \in \mathbb{R}^{n_l, \nu}$. All subscripted variables n are positive integers. We may refer to the network as a single function embedding all layers:

$$x_N = F_{NN}(x_1, u; \alpha) \quad (2)$$

Here, α is the collection of network weights and biases, and a training sample s is $([x_1, y_N]_s)$. We are also interested in neural dynamical systems which are dynamical systems described at least in part by neural networks [15]. A special case is a discrete-time autonomous system:

$$x_{i+1} = F(x_i, u_i; \alpha) \quad y_i = h(x_i) + \nu_i \quad (3)$$

Where x_i is the network input at time step i , h is the observation operator and ν_i is additive observational noise. These equations also have standard systems interpretations in terms of state, parameter, and control input and measurement/output vectors. Please note that the network types defined here are necessary to apply DDDAS concepts to Learning in this paper, but the definitions are incomplete (e.g., missing stochastic neural dynamical system).

Training a neural network using backpropagation, although not commonly described as such¹, is equivalent to solving a two-point boundary value problem adjoined with the dynamics as mentioned above in Eqs. 1–3. However, it is challenging to be uncertainty aware or informative this way.

¹ See course at <http://essg.mit.edu/ml>.

An alternative to 2BVP is the ensemble approach [3, 9] to Learning, which is akin to its use in parameter estimation for nonlinear dynamics and admits both stochastic dynamical and Bayesian estimation perspectives. Specifically, consider the Ensemble Kalman Filter (EnKF) [3, 9], which fundamentally leverages a sample approximation to gradients for inference.

For the purpose of this discussion, we interpret Eq. 3 as a standard discrete dynamical system with Gaussian observational noise $\nu_i \sim \mathcal{N}(0, R = r^2 I)$. Defining $X_i = [x_{i,1} \dots x_{i,E}]$ to be a matrix of E state (column) vector samples obtained by solving F model equations from an initial condition ensemble at the previous time step, and define an observational projection $Z_i = [h(x_{i,1}) \dots h(x_{i,E})]$ and $Y = [y_{i,1} \dots y_{i,E}]$ as an ensemble of observations². We adopt the notation that \tilde{Q} is a deviation matrix obtained by removing the mean column vector of Q from its columns. Then, the filter state estimate X_i^+ may be written as

$$X_i^+ = X_i + \tilde{X}_i \tilde{Z}_i^T [\tilde{Z}_i \tilde{Z}_i^T + R]^{-1} (Y_i - Z_i) = X_i M_{x,i} \tag{4}$$

The EnKF in Eq. 4 is a weakly nonlinear update, and it also enables building Lagged filters and smoothers equivalent to 2BVP [9]. For parameter estimation, let $A_i = [\alpha_1 \dots \alpha_E]_i$ be the matrix of parameter samples at time-step i . The parameters are assumed to persist from one time step to the next in the absence of observation. Thus, the update is:

$$A_i^+ = A_i M_{\alpha,i+k} \quad k > 0 \quad (\text{Parameter Estimation}) \tag{5}$$

Here, an initial ensemble of parameters at initial condition and fixed control input sequence (*wlog*) is used for a k - *step* ensemble simulation to derive a parameter update. The matrix R is just the observational covariance.

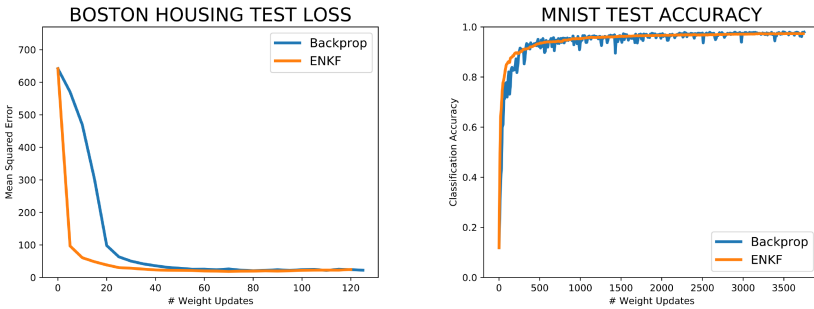


Fig. 1. The Ensemble Kalman Learner offers competitive performance on Boston Housing and MNIST datasets relative to backpropagation.

Parameter estimation immediately provides the basis for neural learning. In particular, consider A_1 to be the initial parameter ensemble (generated with a

² Perturbed observations are used here for simplicity. This is not strictly necessary.

first-guess Gaussian distribution) and B_i to be the minibatch of size S at iteration i . Then,

$$A_{i+1} = A_i \frac{1}{S} \sum_{s \in B_i} M_{\alpha, i, s} \quad i > 0 \quad (\text{Learning}) \quad (6)$$

In this formulation, $M_{\alpha, i, s}$ is the update produced at iteration i from the ensemble simulation of neural networks for each parameter ensemble member A_i and each training sample $[x_1, y_N]_{s \in B_i}$. Thus $S \times E$ parallel simulations are performed. The ‘‘noise model’’ (R) is used to specify a tolerance or performance index in achieving training outputs. The resulting A_{i+1} are then the parameters at iteration $i + 1$. The same approach are applicable to recurrent systems (with rollout) and neural dynamical systems. In contrast to 2BVP it is applicable to stochastic neural dynamical systems and learning systems in general.

Ensemble Kalman Learning has several interesting properties. Network linearization and analytical parameter gradients are unnecessary; loss functions are not limited. The directly-obtained uncertainty estimates further allow us to quantify information gain. In contrast to 2BVP, all layer weights update in parallel. Parallel simulations further reduce computational expense. For small parameter ensembles, the update is compact.

We conducted examples with the Boston Housing [4] and MNIST [5] datasets³. For Boston Housing, we use a neural network with two 32-neuron hidden layers, ReLU activations, least-squares loss function, minibatch of size 16, and 100-member parameter ensemble. IID zero-mean Gaussian with a standard deviation of 0.01 generates the initial parameter ensemble, and the target tolerance is $r = 0.01$. SGD with a learning rate of 0.1 in backpropagation. The results (see Fig. 1) show that the Ensemble Kalman Learner achieves a converged error similar to tuned backpropagation within five epochs.

The MNIST dataset [5] neural network architecture consists of two batch-normalized convolutional layers, max-pooling, and ReLU activations, followed by a single ReLU-activated linear layer of width 10, finally followed by a softmax-activated categorical output layer. We use a least-squares loss function, minibatch size 16, parameter ensemble size 1000, and a target error tolerance of 0.015 to match the observed performance of a highly-performing backpropagation-trained network. Furthermore, the target error tolerance adapts as ensemble variance reduces, up to a lower bound of 0.0015. SGD, with a learning rate of 1.0, is used for backpropagation. The Ensemble Kalman Learner achieves a final test accuracy of 97.1%, competing well with backpropagation at 97.9%. It does this while maintaining better stability at a high learning rate.

4 DDDAS: Informative Structure Learning

In this section, Informative Ensemble Kalman Learning is developed and applied to Neural Structure Learning, a difficult problem. For example, just learning $y = x^2$ with a \tanh activation node is hard (\tanh has no even Taylor expansion

³ Obtain code from <https://github.com/sairavela/EnsembleTrain.git>.

terms). In general, a poor structural basis, poor interpretability, generalization, and extrapolation are all confounding factors. However, in the restricted setting of learning the structure of neural dynamical systems [15] trained from the non-trivial and large class of polynomial dynamics, these issues are overcome. Neural networks with multiplicative gates [15] represent dynamical systems with polynomial nonlinearities exactly. Therefore, learning neural structure from data generated by polynomial dynamics is exactly equal to recovering the polynomial equations (terms and coefficients). Here, consider the problem of learning neural structure and parameters from data generated by numerical solutions to the chaotic Lorenz-63 [6] system⁴, which is defined as:

$$\dot{x}_1 = \sigma(x_2 - x_1), \quad \dot{x}_2 = \rho x_1 - x_2 - x_1 x_3, \quad \dot{x}_3 = -\beta x_3 + x_1 x_2. \quad (7)$$

Suppose the starting model is a second-degree polynomial with nine terms per equation $\mathbf{X} = (x_1, x_2, x_3, x_1 x_2, x_1 x_3, x_2 x_3, x_1^2, x_2^2, x_3^2)$. There are thus 27 unknown parameters $\{a_{ij}\}$, where i indexes $x_{1\dots 3}$ and j indexes \mathbf{X} . The “true” Lorenz equations are simulated from an arbitrary initial condition $\mathbf{x}_0 = (-1.1, 2.2, -2.7)$ with parameters $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$, and time step $dt = 0.01$. The model equations are also simulated using a parameter ensemble of size 100, each initialized i.i.d. from Gaussian with mean 0 and variance 100. If the parameter matrix for the k^{th} ensemble member is denoted $A_k := [a_{ij}^k]$, then $A_k \mathbf{X}_t$ are the predictions. The parameters are then updated using Eq. 4.

When the Ensemble Kalman Learner estimates all 27 possible parameters, the model system converges in approximately 85 iterations when we use an initial parameter mean 0, variance 100, and high-precision/small target variance of 1×10^{-10} . Actual parameters are recovered to within 3×10^{-4} with a posterior variance of 3.2×10^{-7} with the “wrong” term coefficients going to zero. Convergence was repeatable and, remarkably, required no additional sparsity constraints.

However, there are clear limitations. The initial model is arguably quite close to the true model because all true terms are given as options; in other words, the true model lies within the space of candidate models. In general, this is not the case, and the dimensionality of the starting model may be quite high.

The following Informative Ensemble Kalman Learning approach is a better. Instead of automatically updating all terms using Ensemble Kalman Learning, we automatically select a small initial subset of terms as candidates, then alternate between parameter estimation and term selection until achieving sufficient prediction accuracy. To select terms, we first quantify the pairwise mutual information between each of the structure terms and each of the current model’s three training error variables. Pairwise conditional mutual information assumes Gaussian ensembles, but other approaches are feasible [12]. After that, greedy term selection maximizes the cumulative sorted pairwise mutual information while minimizing the number of terms selected. This is sparsity promoting but noniterative and akin to information selection criteria, and thus faster than classical

⁴ Code may be found at <http://github.com/sairavela/LorenzStructureLearn>.

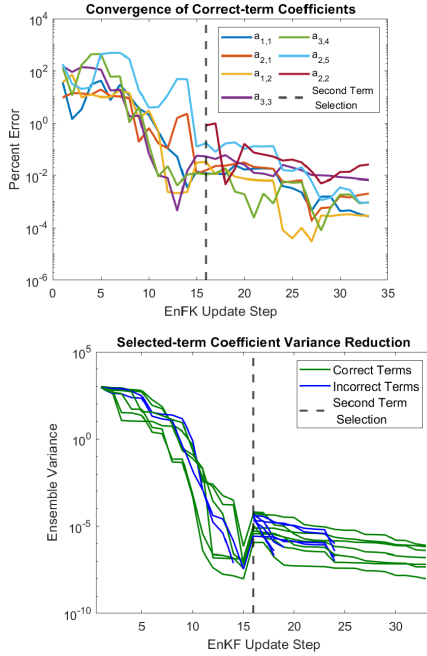


Fig. 2. Convergence of the true parameters and variance reduction of chosen terms for learning the structure of the Lorenz-63 system.

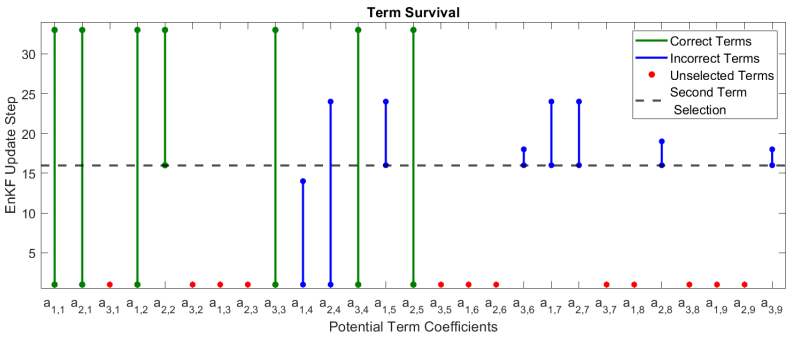


Fig. 3. Survival of terms over iterations. Green are the correct terms, red are unselected, and blue are terms selected and later rejected. (Color figure online)

sparse optimization. The chosen terms augment the system equations. Ensemble Kalman Learning proceeds for a specified variance reduction, at the end of which terms with parameter values approaching zero leave the system equations. The selection cycle repeats. Variances are then rescaled and balanced in the new parameter ensemble, and Ensemble Learning proceeds.

By alternating the maximization of information gain with Ensemble Learning, we recover the Lorenz system equations from the initial model $\hat{\mathbf{x}}_{1\dots 3} = 0$ within approximately 35 iterations requiring three or fewer selection steps. Not only is this more efficient but the incremental selection-rejection (prediction-correction) is automatic and overcomes the dimensionality concern. As shown in Fig. 2, the true equations were recovered with parameter estimates within 1%. The final system structure learned is structurally exact: $\dot{x}_1 = a_{11}x_1 + a_{12}x_2$, $\dot{x}_2 = a_{21}x_1 + a_{22}x_2 + a_{25}x_1x_3$, and $\dot{x}_3 = a_{33}x_3 + a_{34}x_1x_2$. The progression of term presence in the equations throughout the term selection process can be seen in Fig. 3.

5 Conclusions

DDDAS's informative optimization paradigm applies to neural Learning. We develop adaptive Ensemble Kalman Learning, and results on two standard datasets were comparable to stochastic gradient descent. Informative Learning promotes sparsity while maximizing information gain. We applied information gain to learn the Lorenz system equations *ab initio* quickly and incrementally without iterative optimization for selecting terms. In future work, discovering equations of natural hazards and other tractable inference models for Learning and Information Gain are of interest.

References

1. Blasch, E., Ravela, S., Aved, A. (eds.): Handbook of Dynamic Data Driven Applications Systems. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-95504-9>
2. Bryson, A., Ho, Y.C.: Applied Optimal Control. Hemisphere Publishing Corporation (1975)
3. Evensen, G.: The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dyn.* **53**, 343–367 (2003)
4. Harrison Jr., D., Rubinfeld, D.L.: Hedonic housing prices and the demand for clean air. *J. Environ. Econ. Manage.* **5**(1), 81–102 (1978)
5. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
6. Lorenz, E.N.: Deterministic nonperiodic flow. *J. Atmospheric Sci.* **20**(2), 130–141 (1963)
7. Patel, V.: Kalman-based stochastic gradient method with stop condition and insensitivity to conditioning (2015). <https://doi.org/10.1137/15M1048239>
8. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. arXiv preprint [arXiv:1711.10561](https://arxiv.org/abs/1711.10561) (2017)
9. Ravela, S., McLaughlin, D.: Fast ensemble smoothing. *Ocean Dyn.* **57**(2), 123–134 (2007)
10. Rumelhart, D., Hinton, G., Williams, R.: Learning representations by back-propagating errors. *Nature* **323**(6088), 533–536 (1986)

11. Shwartz-Ziv, R., Tishby, N.: Opening the black box of deep neural networks via information (2017)
12. Tagade, P., Ravela, S.: A quadratic information measure for data assimilation. In: IEEE American Control Conference, pp. 598–603, Portland, USA (2014). <https://doi.org/10.1109/ACC.2014.6859127>
13. Tran, T., Do, T.T., Reid, I., Carneiro, G.: Bayesian generative active deep learning (2019)
14. Trautner, M., Margolis, G., Ravela, S.: Informative Neural Ensemble Kalman Learning (2020). [arXiv:2008.09915](https://arxiv.org/abs/2008.09915), <http://arxiv.org/abs/2008.09915>
15. Trautner, M., Ravela, S.: Neural integration of continuous dynamics (2019). [arXiv:1911.10309](https://arxiv.org/abs/1911.10309), <http://arxiv.org/abs/1911.10309>
16. Wang, H., Yeung, D.Y.: Towards bayesian deep learning: a survey (2016). [arxiv:1604.01662](https://arxiv.org/abs/1604.01662), <http://arxiv.org/abs/1604.01662>
17. Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient langevin dynamics. In: ICML (2011)
18. Zhang, Q., Li, B., Wu, Y.: Evolutionary structure optimization of convolutional neural networks for deployment on resource limited systems. In: Huang, D.-S., Jo, K.-H., Zhang, X.-L. (eds.) ICIC 2018. LNCS, vol. 10955, pp. 742–753. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95933-7_82