Informative Neural Ensemble Kalman Learning

Margaret Trautner    Gabriel Margolis    Sai Ravela

Earth Signals and Systems Group
Earth, Atmospheric and Planetary Sciences
Massachusetts Institute of Technology, Cambridge, MA
http://essg.mit.edu
Outline

• The DDDAS Paradigm

• DDDAS for Informative Learning

• Informative Learning to discover dynamics from data

• Jointly optimizing neural structure and parameters
Dynamic Data Driven Learning Systems

- Predict
- Quantify Uncertainty
- Maximize Information Gain
- Information Fusion
- Model

Data

Informative Approach of DDDAS

Applied to the Dynamics of Learning

Enables Efficacious Learning of Dynamics

Also see: arXiv:2008.09915
Stability, Tractability – motivation for dynamic data driven approach

\[
\begin{align*}
\dot{x} &= f_L(x, u_L) \\
\dot{y} &= f_h(y, u_h) \\
\dot{z} &= f_L(z, u_L) + f_{nn}(z, u_{nn})
\end{align*}
\]
Learning Dynamics from Data

Initial Model

\[
\begin{align*}
\dot{x}_1 &= 0 \\
\dot{x}_2 &= 0 \\
\dot{x}_3 &= 0
\end{align*}
\]

First Selection

\[
\begin{align*}
\dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + a_{14}x_1x_2 \\
\dot{x}_2 &= a_{21}x_1 + a_{24}x_1x_2 + a_{25}x_1x_3 \\
\dot{x}_3 &= a_{33}x_3 + a_{34}x_1x_2
\end{align*}
\]

Second Selection

\[
\begin{align*}
\dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + a_{15}x_1x_3 + a_{17}x_1^2 \\
\dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + a_{24}x_1x_2 + a_{25}x_1x_3 + a_{27}x_1^2 + a_{28}x_2^2 \\
\dot{x}_3 &= a_{33}x_3 + a_{34}x_1x_2 + a_{36}x_2x_3 + a_{39}x_3^2
\end{align*}
\]

Final Solution, Converged

\[
\begin{align*}
\dot{x}_1 &= a_{11}x_1 + a_{12}x_2 \\
\dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + a_{25}x_1x_3 \\
\dot{x}_3 &= a_{33}x_3 + a_{34}x_1x_2
\end{align*}
\]
Neural Networks as Dynamical Systems

The Network
\[ x_N = F_{NN}(x_0; \alpha) \]  \hspace{1cm} (1)

Multistage Process
\[ x_{l+1} = F_l(x_l, u_l; \alpha_l), 0 \leq l < N \]  \hspace{1cm} (2)
\[ y_N = x_N + \nu_N, \ \nu_N \sim \mathcal{N}(0, R) \]  \hspace{1cm} (3)

Neural Dynamical System
\[ x_{n+1} = F(x_n, u_n; \alpha), \]  \hspace{1cm} (4)
\[ y_n = h(x_n) + \nu_n, \ \nu_n \sim \mathcal{N}(0, R_n) \]  \hspace{1cm} (5)
The Dynamics of Learning

Objective

\[ J(\cdot; [x_0, y_N]_s) := \frac{1}{2} (y_N - x_N)^T R_N^{-1} (y_N - x_N) + \sum_{l=1}^{N} \gamma_l^T \{ x_l - F_{l-1}(x_{l-1}; \alpha_{l-1}) \} \]  

Forward-Backward

\[ x_0 := [x_0]_s \]  
\[ x_l = F_{l-1}(x_{l-1}; \alpha_{l-1}), \ 0 < l \leq N \]  
\[ \gamma_N = R_N^{-1} ([y_N]_s - x_N) \]  
\[ \gamma_k = \left( \nabla_{x_k} F_k \right)^T \gamma_{k+1} \]  
\[ \frac{\partial J}{\partial \alpha_k} = \left( \nabla_{\alpha_k} F_k \right)^T \gamma_{k+1} \]  

Gradient Descent

\[ \Delta \alpha_l = \frac{1}{S} \sum_{s=1}^{S} \frac{\partial}{\partial \alpha_k} J(\alpha_l; [x_0, y_N]_s) \]  

To further examine the stochastic dynamics of learning, please see: arXiv:2008.09915
An Ensemble Approach to Deep Learning

Parameter Estimation

\( A_i \) : parameter ensemble at iteration \( i > 0 \)
\( X_{i,s} \) : ensemble of Network Predictions using training input \( x_{1,s} \)
\( Y_s \) : ensemble of targets from training output \( y_{N,s} \)
\( B_i \) : minibatch at iteration \( i \) of size \( S_i \)
\( \tilde{X} \) : ensemble of deviations from mean vector

\[
A_{i+1} = A_i \frac{1}{S_i} \sum_{s \in B_i} \tilde{X}_{i,s}^T (\tilde{X}_{i,s} \tilde{X}_{i,s}^T + R_{N,i})^{-1} (Y_s - X_{i,s}) \\
= A_i \frac{1}{S_i} \sum_{s \in B_i} M_{\alpha,i,s} \quad (14)
\]

The interpretation of “observational noise” \( R_{N,i} \) is the tolerance with which the label or output must be learned.

We use an adaptive version where the \( R_{N,i} \) is reduced over iterations.

- Simple update rule, many variants possible
- Parallelizable – parameter updates do not require backpropagation
- Quantifies Uncertainty
- Enables Information Gain Assessment

Also see: arXiv:2008.09915 for application to smoothers and control
Boston Housing Example

![Boston Housing Test Loss](image1)

- **BOSTON HOUSING TEST LOSS**
- Backprop
- ENKF

![Boston Housing Uncertainty](image2)

- **BOSTON HOUSING UNCERTAINTY**
- Prediction Standard Deviation

Ensemble Filter
MNIST Example

**MNIST TEST ACCURACY**

- Classification Accuracy
- # Weight Updates
- Backprop
- ENKF

**MNIST UNCERTAINTY**

- Prediction Standard Deviation
- # Weight Updates
Information Gain for Variable Selection

Quantify Pairwise Mutual Information

\[ I(A : \mathcal{E}) := [ \Psi_{i,j} ]_{m \times n} (A : \mathcal{E}), \]
\[ \Psi_{i,j}(A : \mathcal{E}) := -\frac{1}{2} \ln \left( 1 - \rho^2(A_i, \mathcal{E}_j) \right). \]

Greedy $\ell_0$ often outperforms $\ell_1$
Proposed approach is exceedingly fast

Forward Variable selection better than
  • Monte Carlo or naïve MCMC
  • Iterating and eliminating small weights e.g., using $\ell_1$

Informative approach: accelerate optimization (e.g. $\ell_1$) with information gain

Sort pairwise mutual information in decreasing order

\[ \Psi^*_l \geq \Psi^*_{l+1}, \ 0 < l < mn, \]
\[ \Psi^*_k := \Psi_{i_k,j_k}, \ 1 \leq i_k \leq m, 1 \leq j_k \leq n. \]

Greedy $\ell_0$ Optimization for Variable Selection

\[ k^* = \arg \min_k \sum_{l=1}^k \left[ 1 - \frac{1}{\Psi^*_l} \sum_{i=1}^m \Psi_{i,\mathcal{E}} \right] + C(k). \]
Structure Learning

• Learn the optimal structure of the Neural Network from Data

• Many Challenges

  • How do we verify the structure is optimal?
  • Is this Neural Network generalizable
  • Does it extrapolate?
  • How to interpret the neural network

• Possible Solution
  • Neural Networks for Polynomial Dynamics
Polynomial Dynamical Systems Have Exact Neural Circuits

RNN for Runge Kutta

Lorenz 63 Equations

\[
\begin{align*}
\dot{x}_1 &= \sigma (x_2 - x_1), \\
\dot{x}_2 &= \rho x_1 - x_2 - x_1 x_3, \\
\dot{x}_3 &= -\beta x_3 + x_1 x_2.
\end{align*}
\]

Neural Network for L63 (trivial)

Exact Solution up to numerical accuracy

Use of the PolyNet construct to validate structure learning methodology – here on L63

Also see: \texttt{arXiv:2008.09915} and \texttt{arXiv:1911.10309}
Informative Ensemble Kalman Learner

Ensemble Kalman Learner Example

\[ X = (x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3, x_1^2, x_2^2, x_3^2) \].

Ensemble Kalman Learner recovers the equations after 80 iterations, initialized with degree 2 polynomial. Convergence slows as the initial set grows!

Informative Ensemble Kalman Learner Recovers equations faster, \emph{ab initio} – 60%
Survival Chart Lorenz-63
Convergence

- Convergence of Correct-term Coefficients
- Selected-term Coefficient Variance Reduction
Summary

• Dynamical Systems are associated with Learning

• Learning is a two-point boundary value problem

• The dynamics of learning is stochastic

• Stochastic Dynamics allows for Information Gain to be quantified

• Ensemble Methods approximate the Fokker Planck

• Informative Ensemble Kalman Learning Efficaciously Recovers Structure

• Poly Nets exactly model discrete time polynomial dynamics and thus correctness is verified.
The Stochastic Dynamics of Learning

Mini-batch

\[ \nabla J(\alpha_i) = \frac{1}{|B_i|} \sum_{s \in B_i} \frac{\partial J(\alpha_i; [x_1, y_N]_s)}{\partial \alpha_i}, \]

Ito Equation

\[ d\alpha_t = \mu_t(\alpha_t) \, dt + \sigma_t(\alpha_t) \, \eta_t. \]

Perturbation Expansion

\[ -\tau \nabla J(\alpha_i) = \mu(\alpha_i) + w(\alpha_i), \]

Fokker Planck

\[ \frac{\partial p_\alpha}{\partial t} = -\sum_{j=1}^{n} \frac{\partial}{\partial \alpha_j} [\mu_{t,j} \, p_\alpha] + \sum_{k=1}^{n} \sum_{l=1}^{n} \frac{\partial^2}{\partial \alpha_k \partial \alpha_l} [D_{t,kl} \, p_\alpha] \]

Stochastic Model

\[ w(\alpha_i) = \sigma(\alpha_i) \, \eta_i, \]