Ensemble Kalman Filter for Neural Network Based One-Shot Inversion

Sassan Mokhtar

July 6, 2020

Outline

Table of Contents

Introduction

The goal is to solve the following inverse problem:
$\rightarrow$ Recover the unknown parameter $u \in \mathcal{X}$ in an abstract model

$$M(u, p) = 0$$

from a finite number of observation of the state $p \in \mathcal{V}$ given by

$$O(p) = y \in \mathbb{R}^{n_y}$$

where

- The operator $M : \mathcal{X} \times \mathcal{V} \rightarrow \mathcal{W}$ describes the underlying forward model, typically a PDE or ODE.
- The variable p denotes the state of the model and is defined on a domain $D \in \mathbb{R}^d$.
- The operator $O : \mathcal{V} \rightarrow \mathbb{R}^{n_y}$ is the observation operator, mapping the state variables p to the observations.

$\rightarrow$ The idea is to solve the underlying model equation simultaneously with the optimality conditions.

Optimization Approach to the Inverse Problem

Notation: For a symmetric positive definite matrix A,

$$\|.\|_A = \|A^{-1/2}.\| \quad \text{and} \quad \langle \cdot, \cdot \rangle_A = \langle \cdot, A^{-1} \rangle$$

Assumption: $\mathcal{X}, \mathcal{W},$ and $\mathcal{V}$ are finite-dimensional.

$\rightarrow$ The optimization approach lead to the following problem,

$$\min_{u,p} \|O(p) - y\|^2_{\Gamma_{obs}}$$
$$\text{s.t. } M(u, p) = 0$$

where $\Gamma_{obs} \in \mathbb{R}^{n_y \times n_y}$.

- ☐ Due to ill-conditioning of the inverse problem, a regularization term on the unknown parameters is often introduced to stabalize the optimization.

Optimization Approach to the Inverse Problem

$$\min_{u,p} \|O(p) - y\|^2_{\Gamma_{obs}} + \alpha_1 \mathcal{R}(u)$$
$$\text{s.t. } M(u,p) = 0$$

where $\mathcal{R} : \mathcal{X} \to \mathbb{R}$ is the regularization and the scalar α_1 is chosen according to prior knowledge on u.

Reduced Problem: The forward problem $M(u,p) = 0$ is typically a well-posed problem, i.e. for each parameter $u \in \mathcal{X}$, there exist a unique state $p \in \mathcal{V}$ such that $M(u,p) = 0$ in $\mathcal{W}$.
$\to$ Introduce the solution operator $S : \mathcal{X} \to \mathcal{V}$ s.t. $M(u, S(u)) = 0$. Then, the optimization problem can be reformulated as following unconstrained optimization problem,

$$\min_{u \in \mathcal{X}} \|O(S(u)) - y\|^2_{\Gamma_{obs}} + \alpha_1 \mathcal{R}(u)$$

Bayesian Approach to the Inverse Problem

Next, the Bayesian approach is adopted to inverse problems.

$\rightarrow$ The unknown parameters u as an $\mathcal{X}$-valued random variable with prior distribution μ_0.

Assumption 1*:* The noise in the observation is described by a random variable $\eta \sim \mathcal{N}(0, \Gamma_{obs})$, i.e.

$$y = O(S(u)) + \eta$$

Assumption 2*:* The noise η is independent of u.
$\rightarrow$ Then, the posterior distribution is given by,

$$\mu^*(du) \propto \exp\Big(-\frac{1}{2}\|O(S(u)) - y\|^2_{\Gamma_{obs}}\Big)\mu_0(du)$$

Connection Between the Optimization and the Bayesian Approach

While the optimization approach leads to a point estimate of the unknown parameters, the Bayesian approach computes the conditional distribution of the unknown parameters given the data.

$\rightarrow$ Use point estimates, such as the maximum a posteriori (MAP) estimate, the most likely point of the unknown parameters.
The MAP estimate is defined as,

$$\arg\max_{u \in \mathcal{X}} \exp\Big(-\frac{1}{2}\|O(S(u)) - y\|^2_{\Gamma_{obs}}\Big)\mu_0(u)$$

Based on the assumption that $\mu_0 \sim \mathcal{N}(u_0, C)$, the MAP estimate is given by the solution of the following minimization problem,

$$\min_{u} \ \frac{1}{2}\|O(S(u)) - y\|^2_{\Gamma_{obs}} + \frac{1}{2}\|u - u_0\|^2_{C}$$

One-Shot Formulation for inverse Problems

The goal is to simultaneously solve the forward and optimization problem.

There are various names for the simultaneous solution of the design and state equation: one-shot method, all-at-once method, piggy-back iterations etc.

$\rightarrow$ Following the one-shot idea, the goal is to solve the problem,

$$F(u,p) = \begin{pmatrix} M(u,p) \\ O(p) \end{pmatrix} = \begin{pmatrix} 0 \\ y \end{pmatrix} =: \tilde{y}$$

Due to noise in the observations and error in the model, we consider

$$y = O(p) + \eta_{obs} \quad \text{and} \quad 0 = M(u,p) + \eta_{model}$$

where $\eta_{obs} \sim \mathcal{N}(0, \Gamma_{obs}),\ \Gamma_{obs} \in \mathbb{R}^{n_y \times n_y}$ and $\eta_{model} \sim \mathcal{N}(0, \Gamma_{model}),\ \Gamma_{model} \in \mathbb{R}^{n_w \times n_w}$.

One-Shot Formulation for inverse Problems

Thus, the problem becomes,

$$\tilde{y} = F(u, p) + \begin{pmatrix} \eta_{model} \\ \eta_{obs} \end{pmatrix}$$

The MAP estimate is then computed by the solution of the following minimization problem,

$$\min_{u,p} \ \frac{1}{2}\|F(u, p) - \tilde{y}\|_{\Gamma}^2 + \alpha_1 \mathcal{R}_1(u) + \alpha_2 \mathcal{R}_2(p)$$

where $\mathcal{R}_1 : \mathcal{X} \to \mathbb{R}$ and $\mathcal{R}_2 : \mathcal{V} \to \mathbb{R}$ are regularization of the parameter $u \in \mathcal{X}$ and the state $p \in \mathcal{V}$, $\alpha_1, \alpha_2 > 0$ and $\Gamma = \begin{pmatrix} \Gamma_{model} & 0 \\ 0 & \Gamma_{obs} \end{pmatrix}$

One-Shot Formulation for inverse Problems

$\rightarrow$ This setting is starting point of incorporating the neural networks into the problem.
Idea: Instead of minimizing with respect to the state p, we will approximate the solution of the forward problem p by a neural network p_θ. Also, θ is the parameters of the neural network to be learn with this framework.

Thus, we obtain the minimization problem,

$$\min_{u, p_\theta} \frac{1}{2}\|F(u, p_\theta) - \tilde{y}\|_\Gamma^2 + \alpha_1 \mathcal{R}_1(u) + \alpha_2 \mathcal{R}_2(p_\theta, \theta)$$

where p_θ denotes the state approximated by the neural network.

Table of Contents

Neural Networks

A neural network p_θ is a mapping

$$\begin{aligned} p_\theta : \mathbb{R}^d &\to \mathbb{R}^{N_L} \\ x &\mapsto p_\theta(x) := x_L \end{aligned}$$

defined by the recursion

$$\begin{aligned} x_0 &:= x, \\ x_l &:= \sigma(W_l x_{l-1} + b_l), \quad \text{for } l = 1, ..., L-1 \\ x_L &:= \sigma(W_L x_{L-1} + b_L) \end{aligned}$$

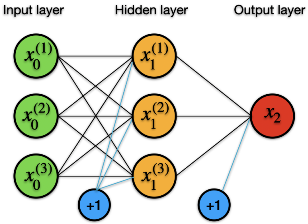
where,

- L is the number of layers in the network and d is the input dimension.
- $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function, applied componentwise
- The neural network parameters θ is the sequence of matrix-vector tuples

$$\theta = \Big((W_l, b_l)\Big)_{l=1}^{L} = \Big((W_1, b_1), ..., (W_L, b_L)\Big), \; W_l \in \mathbb{R}^{N_l \times N_{l-1}}, b_l \in \mathbb{R}^{N_l}$$

Feed-Forward Pass Example

To demonstrate how to calculate the output from the input in neural networks (feed-forward pass), consider the following simple example,



$$x_1^{(1)} = \sigma\left(w_1^{(1,1)} x_0^{(1)} + w_1^{(1,2)} x_0^{(2)} + w_1^{(1,3)} x_0^{(3)} + b_1^{(1)}\right)$$
$$x_1^{(2)} = \sigma\left(w_1^{(2,1)} x_0^{(1)} + w_1^{(2,2)} x_0^{(2)} + w_1^{(2,3)} x_0^{(3)} + b_1^{(2)}\right)$$
$$x_1^{(3)} = \sigma\left(w_1^{(3,1)} x_0^{(1)} + w_1^{(3,2)} x_0^{(2)} + w_1^{(3,3)} x_0^{(3)} + b_1^{(3)}\right)$$
$$x_2 = \sigma\left(w_2^{(1,1)} x_1^{(1)} + w_2^{(1,2)} x_1^{(2)} + w_2^{(1,3)} x_1^{(3)} + b_2^{(1)}\right)$$

Back-Propagation

- The next step is to train the weights and biases (neural network parameters θ) such that the error between the desired output and output of the neural network is minimized.
- Typically, the quasi-Newton methods are applied to train the neural networks
- In this work the Ensemble Kalman Inversion (EKI) is proposed to train the neural network
- EKI can be regarded as an optimization method without requiring derivatives with respect to the weights and biases.

$\rightarrow$ Note that in this work, the input of the neural network is a point in the domain of the state $x \in D$ and the output of the neural network is the state at this point $p_\theta(x) \in \mathbb{R}$, i.e. $N_L = 1$.

Table of Contents

Ensemble Kalman Inversion

$\rightarrow$ The ensemble Kalman inversion (EKI) generalizes the well-known ensemble Kalman filter (EnKF).

Recall the posterior distribution μ^* given by

$$\mu^*(dv) \propto \exp\Big(-\frac{1}{2}\|G(v) - y\|_\Gamma^2\Big)\mu_0(dv)$$

for an inverse problem

$$y = G(v) + \eta$$

where G is the mapping from the unknown $v \in \mathbb{R}^{n_v}$ to the observations $y \in \mathbb{R}^{n_y}$ with $\eta \sim \mathcal{N}(0, \Gamma)$.

Ensemble Kalman Inversion

Then, by scaling the likelihood (data misfit) by the step size $h = N^{-1}, \; N \in \mathbb{N}$, define the intermediate measures

$$\mu_n(dv) \propto \exp\Big(-\frac{1}{2} nh \| G(v) - y \|_\Gamma^2 \Big) \mu_0(dv) \quad n = 0, \ldots, N$$

Idea: To evolve the prior distribution μ_0 into the posterior distribution $\mu_N = \mu^*$ by this sequence of intermediate measures and to apply the EnKF to the resulting artificial time dynamic system.

The EKI then uses an ensemble of J particles $\{v_0^{(j)}\}_{j=1}^J$ with $J \in \mathbb{N}$ to approximate the intermediate measures μ_n by $\mu_n \simeq \frac{1}{J}\sum_{j=1}^J \delta_{v_n^{(j)}}$ where δ_v denotes the delta-Dirac mass located at $v_n^{(j)}$.

- **Motivation:** The sequence of intermediate measures and the resulting artificial time allows to derive the continuous time limit of the iteration by taking the parameter h to zero.

Ensemble Kalman Inversion

$\rightarrow$ The particles are transformed in each iteration by the application of the Kalman update formulas to the empirical mean $\bar{v}_n = \frac{1}{J}\sum_{j=1}^{J} v_n^{(j)}$ and covariance $C(v_n) = \frac{1}{J-1}\sum_{j=1}^{J}(v_n^{(j)} - \bar{v}_n) \otimes (v_n^{(j)} - \bar{v}_n)$ in the form

$$\bar{v}_{n+1} = \bar{v}_n + K_n(y - G(\bar{v}_n)), \quad C(v_{n+1}) = C(v_n) - K_n C^{y,v}(v_n)$$

where $K_n = C^{v,y}(C^{y,y} + \frac{1}{h}\Gamma)^{-1}$ denotes the Kalman gain. Also, for $v = \{v^{(j)}\}_{j=1}^{J}$, the operators are given by (with $\bar{G} = \frac{1}{J}\sum_{j=1}^{J} G(v^{(j)})$),

$$C^{y,y}(v) = \frac{1}{J}\sum_{j=1}^{J}\left(G(v^{(j)}) - \bar{G}\right) \otimes \left(G(v^{(j)}) - \bar{G}\right)$$

$$C^{v,y}(v) = \frac{1}{J}\sum_{j=1}^{J}\left(v^{(j)} - \bar{v}\right) \otimes \left(G(v^{(j)}) - \bar{G}\right)$$

$$C^{y,v}(v) = \frac{1}{J}\sum_{j=1}^{J}\left(G(v^{(j)}) - \bar{G}\right) \otimes \left(v^{(j)} - \bar{v}\right)$$

Ensemble Kalman Inversion

$\rightarrow$ Since this update does not uniquely define the transformation of each particle $v_n^{(j)}$ to the next iteration $v_n^{(j+1)}$, the specific choice of transformation leads to different variants of the EKI.
We use the generalization of the EnKF as introduced by (Iglesias et. al. 2013) resulting in a mapping of the particles of the form,

$$v_{n+1}^{(j)} = v_n^{(j)} + C^{v,y}(v_n)\Big(C^{y,y}(v_n) + \frac{1}{h}\Gamma\Big)^{-1}\Big(y_{n+1}^{(j)} - G(v_n^{(j)})\Big), \; j = 1, \ldots, J$$

where

$$y_{n+1}^{(j)} = y + \xi_{n+1}^{(j)}$$

with $\xi_{n+1}^{(j)}$ are i.i.d. random variables distributed according to $\mathcal{N}(0, \frac{1}{h}\Sigma)$ with $\Sigma = \Gamma$ for the case of perturbed observations and $\Sigma = 0$ to the unperturbed observations.

EKI for Neural Network Based One-Shot Formulation

By approximating the state of the underlying model by a neural network, we would like to optimize u (unknown parameter) and θ (parameters of the neural network).

Idea: To define the function $H(v) := H(u,\theta) = F(u, p_\theta)$ and $v = (u,\theta)^\top$. This yields the empirical summary statistics,

$$\overline{(u,\theta)}_n = \frac{1}{J}\sum_{j=1}^{J}(u_n^{(j)}, \theta_n^{(j)}), \quad \bar{H}_n = \frac{1}{J}\sum_{j=1}^{J} H(u_n^{(j)}, \theta_n^{(j)}),$$

$$C_n^{u\theta,y} = \frac{1}{J}\sum_{j=1}^{J}\left((u_n^{(j)}, \theta_n^{(j)})^\top - \overline{(u,\theta)}_n^\top\right) \otimes \left(H(u_n^{(j)}, \theta_n^{(j)}) - \bar{H}_n\right)$$

$$C_n^{y,y} = \frac{1}{J}\sum_{j=1}^{J}\left(H(u_n^{(j)}, \theta_n^{(j)}) - \bar{H}_n\right) \otimes \left(H(u_n^{(j)}, \theta_n^{(j)}) - \bar{H}_n\right)$$

EKI for Neural Network Based One-Shot Formulation

And the following EKI update

$$(u_{n+1}^{(j)}, \theta_{n+1}^{(j)})^\top = (u_n^{(j)}, \theta_n^{(j)})^\top + C_n^{u\theta,y}\left(C_n^{y,y} + \frac{1}{h}\Gamma\right)^{-1}\left(\tilde{y}_{n+1}^{(j)} - H(u_n^{(j)}, \theta_n^{(j)})\right)$$

where the perturbed observations are computed as before

$$\tilde{y}_{n+1}^{(j)} = \tilde{y} + \xi_{n+1}^{(j)},\ \xi_{n+1}^{(j)} \sim \mathcal{N}(0, \frac{1}{h}\Sigma)$$

with

$$\tilde{y} = \begin{pmatrix} 0 \\ y \end{pmatrix},\ \Gamma := \begin{pmatrix} \Gamma_{model} & 0 \\ 0 & \Gamma_{obs} \end{pmatrix}$$

EKI for Neural Network Based One-Shot Formulation

$\rightarrow$ Description of the EKI applied to solve the neural network based one-shot formulation:

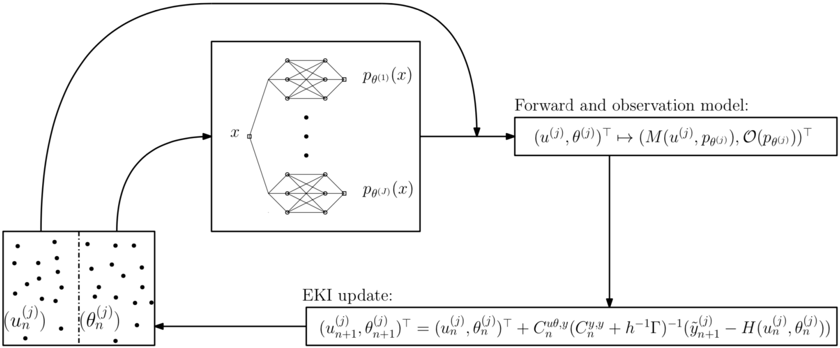


Table of Contents

Conclusion

- We formulate the inverse problem in a one-shot fashion and establish the connection to Bayesian setting.
- We applied neural networks to estimate the solution of the underlying problem.
- Finally, we illustrate how EKI can be used to solve the resulting optimization problem

References

1. Guth, Philipp A and Schillings, Claudia and Weissmann, Simon (2020)
Ensemble Kalman filter for neural network based one-shot inversion

2. Iglesias, Marco A and Law, Kody JH and Stuart, Andrew M (2013)
Ensemble Kalman methods for inverse problems

3. Thomas, Andy (2020)
Neural networks tutorial - A pathway to deep learning