2023

【海報論文】

□一般組 □研究生組

□大學組

11th 台灣TAIWAN 工業與應用數學會年會

TWSIAM Annual Meeting

Prediction of Chaotic Systems with Noisy Observations using Data Assimilation and Deep Learning

Chung-Yu Shih (施重宇), ***Department of Mathematics, National Central University, Taiwan*** (davidavidshih11@gmail.com)

Advisor: Prof. Feng-Nan Hwang(黃楓南) , ***Department of Mathematics, National Central University, Taiwan***

Abstract: Prediction problems, such as numerical weather prediction and computational fluid dynamics, often require rich physical laws and background knowledge to establish mathematical models. These physics-driven approaches have been developed over time and have become reliable. However, for some problems, such as space weather, which are more chaotic and involve unfamiliar phenomena, it can be challenging to build a functional mathematical model due to the lack of necessary knowledge. In such cases, observation data, which is often noisy, is the only available information. To avoid the problem of finding physical laws, we built a purely data-driven model using deep learning called long short-term memory (LSTM) **[1]**. Based on the neural network we trained, we developed an ensemble data assimilation system **[2]** to denoise the observations and improve the accuracy of the initial conditions. With these more precise initial conditions, we can provide more accurate predictions. The experiments in this study follow the observing system simulation experiments (OSSEs) **[3]** and are based on data generated by the Lorenz 63 model **[4]**.

Problem Description

What if the only information we trust is tons of noisy observations?

Traditional approach

Traditionally, to provide a numerical forecast using background knowledge and real-world observations, we follow these four steps:

Step 1: Build a mathematical model

The first step is to use physical laws to establish the governing equation with respect to time, such as the Lorenz 63 model **[4]**:

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

Step 2: Find the constant

Next, we must determine the constants in the system. For this experiment, we have set the parameters as

$$\sigma = 10, \quad \rho = 28, \quad \beta = 8/3.$$

Step 3: Find a model solver

We need to discretize and solve the prediction model. For an ordinary differential equation (ODE) system like Lorenz 63, we can use the fourth-order Runge-Kutta method (RK4).

Step 4: Give initial condition

We also require an initial condition to make a forecast. Chaotic systems like weather phenomena are highly sensitive to initial data **[4]**, and therefore meteorologists use data assimilation to reduce observation errors. One example of data assimilation is the extended Kalman filter (EKF), shown in **Figure 1**, which combines the background (previous prediction) X_p with the observation X_o using the error covariance matrix (ECM) to find the analysis state X_a, which is believed to be more reliable and accurate. By using X_a as the initial condition, the forecast will be improved. However, most data assimilation methods face problems such as the requirement for a tangent linear model L, which is a predictor for error, and the issue of filter divergence **[2]** etc.

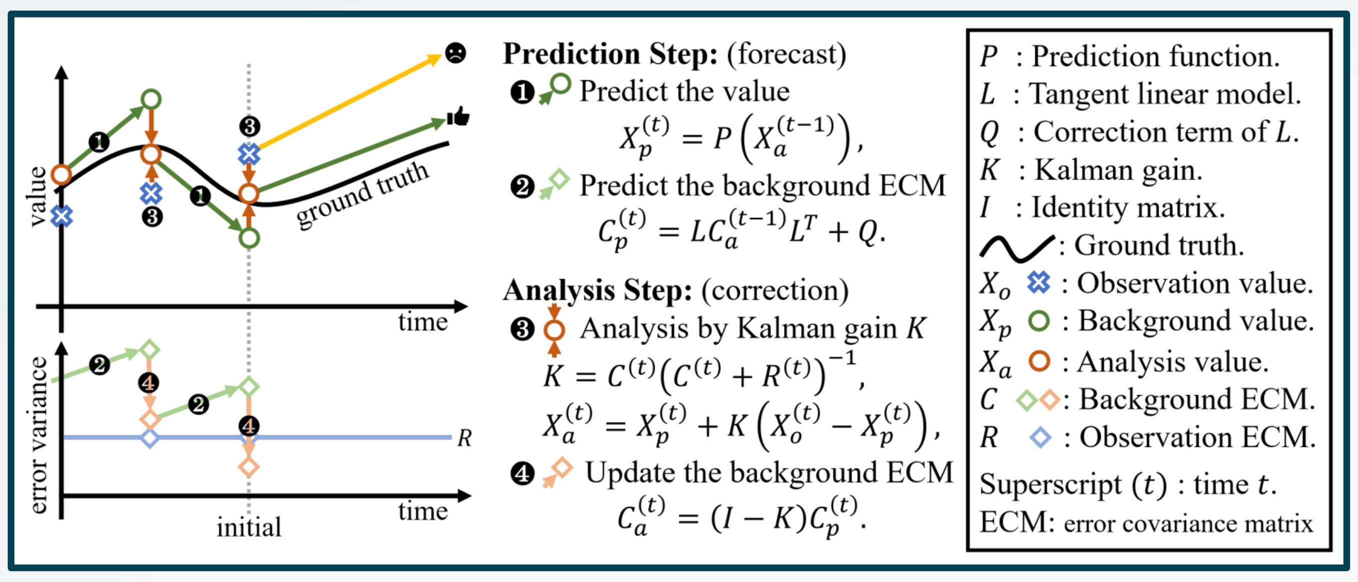

Figure 1. The process of the extended Kalman filter, which can improve the accuracy of the initial data and lead to a better prediction.

The problems of traditional approach

If we attempt to provide a precise and long-term forecast for a chaotic and unfamiliar phenomenon using only observation data, we cannot simply follow the traditional approach. We will encounter the following challenges:

- In ***Step 1***, constructing a state-of-the-art physics-driven model for a complex phenomenon is challenging and often requires collaboration among multiple scientists and mathematicians.
- In ***Step 2***, model errors can make finding the right parameters tricky. This can lead to an underfitting model that lacks detail, some cases requiring additional measurements from real-world experiments.
- In ***Step 3***, some models can be very difficult to solve, and the limitations of numerical solvers can sometimes be unrealistic, such as requiring boundary conditions. Additionally, computational cost and numerical errors can still be problematic.
- In ***Step 4***, data assimilation relies on the prediction model, and some systems require the additional predictor L to estimate the ECM. If both P and L is not sufficiently accurate, the data assimilation process will be ineffective and may result in filter divergence.
- The errors in each step will accumulate and become amplified when we need a long-term forecast.

Methodology

LSTMEnKF

To avoid all the problems we have mentioned, we have set up a purely data-driven model using the long short-term memory (LSTM) neural network **[1]**. We have then developed a variant of the stochastic ensemble Kalman filter (EnKF) **[2]** as the data assimilation system. By taking advantage of ensemble forecasting, we can easily combine these two methods, and call it the LSTMEnKF method. To compare it with the traditional approach, we will follow these four steps:

Step 1: Build LSTM model

We present the LSTM neural network architecture in **Table 1**.

Step 2: Trian the LSTM model

We train the unknown parameters in each layer of the neural network according to the settings provided in **Table 1**.

NN Structure	Sequence Input Layer (3 unit) LSTM Layer (30 unit) LSTM Layer (30 unit) Fully Connected Layer (3 unit) Regression Layer
Epoch	60 Epoch
Train info.	18 sec. (CPU) with 3600 data
Optimizer	Adaptive Moment Estimation (Adam)

Table 1. LSTM structure and training information, with noisy Lorenz 63 data.

Step 3: LSTM don't need a solver

As a prediction model, LSTM is an easily computable model that does not consume too many computer resources.

Step 4: Use LSTMEnKF to build analysis data as initial

With ensemble forecasting, the EnKF uses the distribution of the ensemble members to estimate the background ECM. We have

$$C = \frac{1}{m}\left(X_p - \overline{X_p}\right)\left(X_p - \overline{X_p}\right)^T,$$

where m is the number of members, X_p is state and $\overline{X_p}$ is mean

$$X_p = \begin{bmatrix} x_1 & x_2 & \dots & x_m \\ y_1 & y_2 & \dots & y_m \\ z_1 & z_2 & \dots & z_m \end{bmatrix},\ \overline{X_p} = \begin{bmatrix} \bar{x} & \bar{x} & \dots & \bar{x} \\ \bar{y} & \bar{y} & \dots & \bar{y} \\ \bar{z} & \bar{z} & \dots & \bar{z} \end{bmatrix}.$$

Compared to EKF, EnKF does not require ❷ and ❹. we can avoid the building of tangent linear model L, and give more nonlinearity to the background ECM.

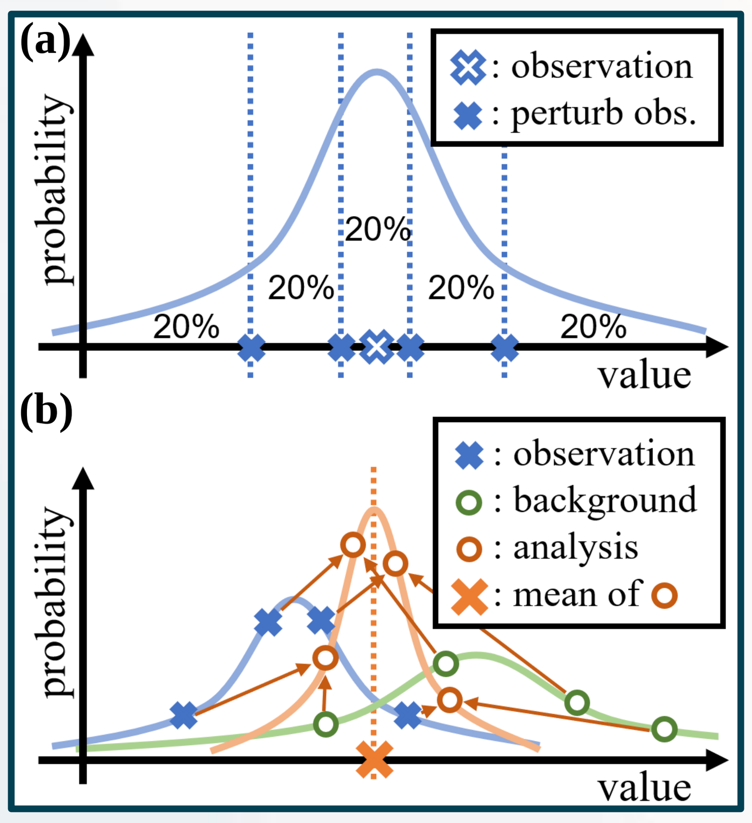

Figure 2. An example of 4 ensemble members, **(a)** is perturb disterbution, **(b)** is the pairing of analysis.

For prediction steps ❶, we use the LSTM model as predictor P. For analysis steps ❸, unlike stochastic EnKF, which uses Gaussian random noise to perturb the observations, in **Figure 2 (a),** we use the normal inverse cumulative distribution function to ensure that the perturbed observations stay in a fixed and perfect distribution. Additionally, in **Figure 2 (b)** the analysis pairing is based on the value. Both of these changes are made to avoid filter divergence.

Experiment and Results

Observing System Simulation Experiments (OSSEs)

The experiment will follow the OSSEs **[3]** process, which is a standard procedure for testing the data assimilation systems. We will solve the Lorenz 63 model and use it as the ground truth. Next, we will perturb it with Gaussian noise to simulate observation data which is shown in **Figure 3**. Due to space limitations, only the data of x will be presented.

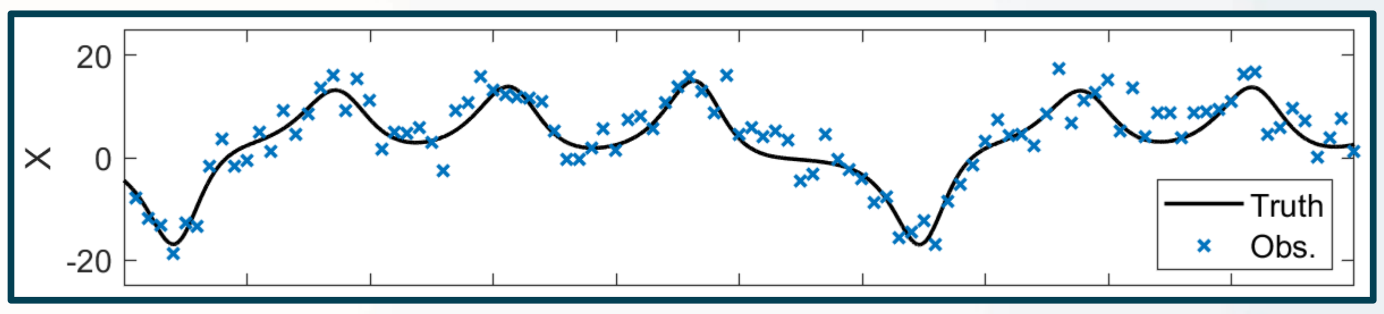

Figure 3. Simulated observation data and ground turth for Lorenz 63.

We will split the noisy observation data into 3600 (90%) training data, and 400 (10%) testing data. We trained the LSTM model (using the setting in **Table 1**) only with the noisy training data. We then used the LSTMEnKF to filter the testing data and use the analysis state to make a forecast. As shown in **Figure 4**, using the analysis data as initial state, the LSTMEnKF provides better long-term predictions.

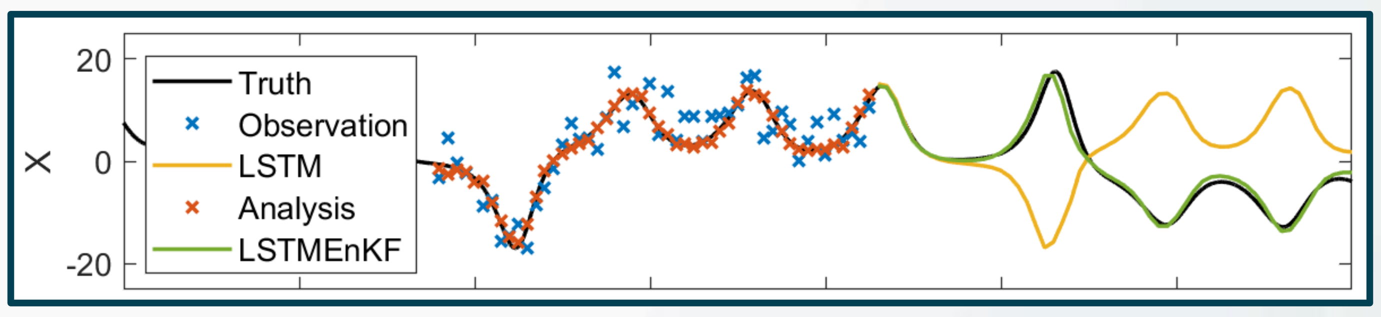

Figure 4. Comparison of LSTM and LSTMEnKF

Another advantage is that the LSTMEnKF can also denoise the noisy observation data, as seen in **Table 2** where it is effective in denoising both training and testing data.

	x	y	z
Obs.	3.07 / 3.12	2.98 / 2.90	3.01 / 3.07
Ana.	**0.90 / 0.87**	**1.35 / 1.27**	**1.39 / 1.17**

Table 2. RMSE of observation data and LSTMEnKF analysis data. (train / test)

Conclusions: The LSTMEnKF is a successful approach that combines deep learning models and data assimilation to provide better predictions under noisy observations. The modular approach allows us to replace the neural network architecture or use it as a filter. However, the success of neural networks in this task does not necessarily mean that traditional mathematical models will be replaced. Traditional mathematical models have been developed over a long period of time and offer the advantage of describing chaotic phenomena and simulating them at different scales. This is one of the main reasons why we aim to establish a neural network that can compete with differential equations. However, the simple architecture of LSTM is more suitable for finding periodicity or over-smoothing the model to reduce loss. In the future, we hope to combine mathematical simulations, data assimilation, and deep learning to obtain more accurate results from real-world data and experiments such as space weather forecast.

References

[1] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[2] Evensen, Geir. *Data assimilation: the ensemble Kalman filter*. Vol. 2. Berlin: springer, 2009.

[3] Arnold Jr, Charles P., and Clifford H. Dey. "Observing-systems simulation experiments: Past, present, and future." *Bulletin of the American Meteorological Society* 67.6 (1986): 687-695.

[4] Lorenz, Edward N. "Deterministic nonperiodic flow." *Journal of atmospheric sciences* 20.2 (1963): 130-141.