Using Deep Learning to increase the ensemble size in an EnKF with the recentering technique : experiments with Lorenz 1996 model

Vikram Khade

EnKF workshop, Balestrand, Norway

30 May 2022

Motivation

- Localization is required in an EnKF because the ensemble size is far less than the model dimensionality. (N <<<< d)
- Ensemble size is small because it is computationally expensive to run model forecasts.
- Deep Learning (DL) model can be trained to simulate the forecast model and then it can be used to run multiple forecasts.
- Typically, for real models (like ECCC's GEM) training a DL model is very expensive – but running a trained model is relatively cheap (Weyn et. al, 2020).
- Increasing the ensemble size using DL simulated model could allow for a longer localization radius. This will improve balance and reduce RMSE since each observation will update variables out to longer distances. In an EnKF the background estimate of covariance improves as the ensemble size increases.
- Today I will present some results regarding these ideas using the Lorenz-96 model (L96).

Demonstrates deep learning of 2d fields

JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2020MS002109

Key Points:

- A convolutional neural net (CNN) is developed for global weather forecasts on the cubed sphere
- Our CNN produces skillful global forecasts of key atmospheric variables at lead times up to 7 days
- Our CNN computes stable 1-year simulations of realistic atmospheric

Improving Data-Driven Global Weather Prediction Using Deep Convolutional Neural Networks on a Cubed Sphere

Jonathan A. Weyn[1], Dale R. Durran[1], and Rich Caruana[2]

[1]Department of Atmospheric Sciences, University of Washington, Seattle, WA, USA, [2]Microsoft Research, Redmond, WA, USA

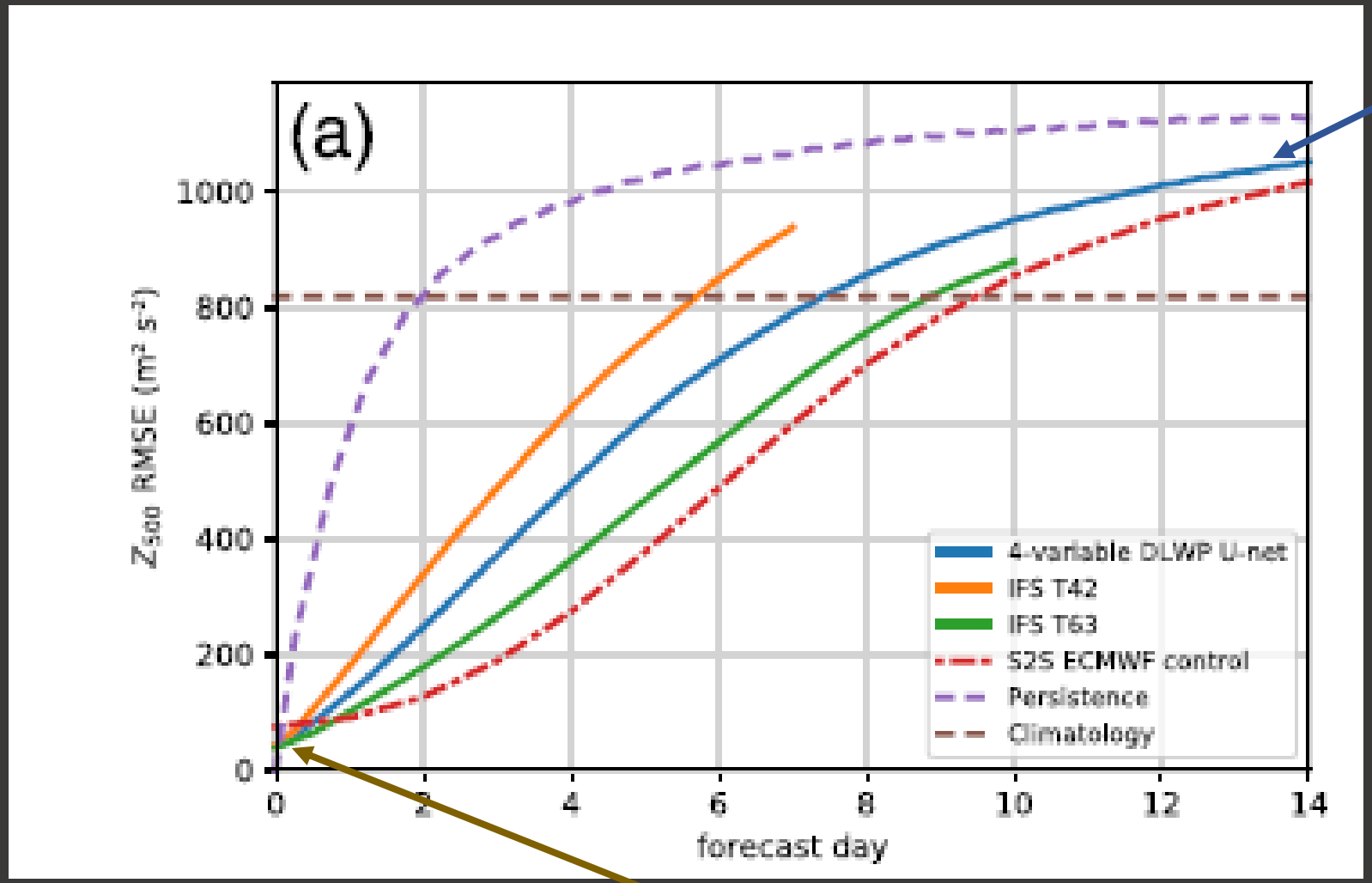
Weyn et. al.

Training data used : ERA5, regridded to 2 degree (1979-2018)

Inputs : Geopotential height at 500 hPa.
Geopotential height at 1000 hPa.
300-700 hPa geopotential thickness.
2m temperature.
ToA insolation.
Land-sea mask.
Topographic height.

Outputs the fields mentioned in Inputs.

Deep learning model outperforms T42.



DL simulated model

IFS T42 : ECMWF IFS model, 62 levels, 2.8 deg.

IFS T63 : ECMWF IFS model, 137 levels, 1.9 deg.

S2S : ECMWF IFS subseasonal to seasonal model
16 km – 31 km.
Fully coupled to ocean and sea ice models.

I am interested in 6-12 hours !

Computational speed

Although our DLWP model lags the performance of a high-resolution operational NWP model by about 2–3 days of forecast lead time relative to climatology, it does have one significant advantage: computational speed. After a one-time computational cost of 2–3 days for training on a single NVidia Tesla V100 GPU, our DLWP model can produce a global 4-week forecast in less than two tenths of a second. At this speed, one could generate a 1,000-member ensemble of 1-month forecasts in about 3 min. In contrast, the full dynamical IFS model at approximately equivalent T63 horizontal resolution, run albeit somewhat inefficiently on a 36-core computing node, requires nearly 24 min to produce a single 4-week forecast, or about 16 days for the same 1,000-member ensemble forecast. Operationally, ECMWF, despite vast supercomputing resources,

Single Nvidia Tesla V100 GPU

Training : 2-3 days

Forecasting : 1000, 1-month forecasts in 3 minutes !

40 dimensional Lorenz 1996 model

$$dx_i/dt = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F$$

The x_i are the components or variables.

i = 1…40 (Dimensionality = d = 40)

F = 8 (Forcing)

$x_0 = x_N$

0.05 time units ~ 6 hours in atmospheric model.

This model has been used by many researchers as a test bed to try out new data assimilation schemes, improvements to DA techniques etc.

Weyn et. al. : Takeaway message

- Though this paper is not about 3d fields, the computational speed versus accuracy trade off is shown to be quite good.
- It appears that it might be possible to train a DL network to forecast the 3d state of the atmospheric model over a 6-12 hour period (using the archived data of the model).
- The computational efficiency might be enough to produce 100s (if not 1000s) of ensemble members to augment the ensemble size in the EnKF.

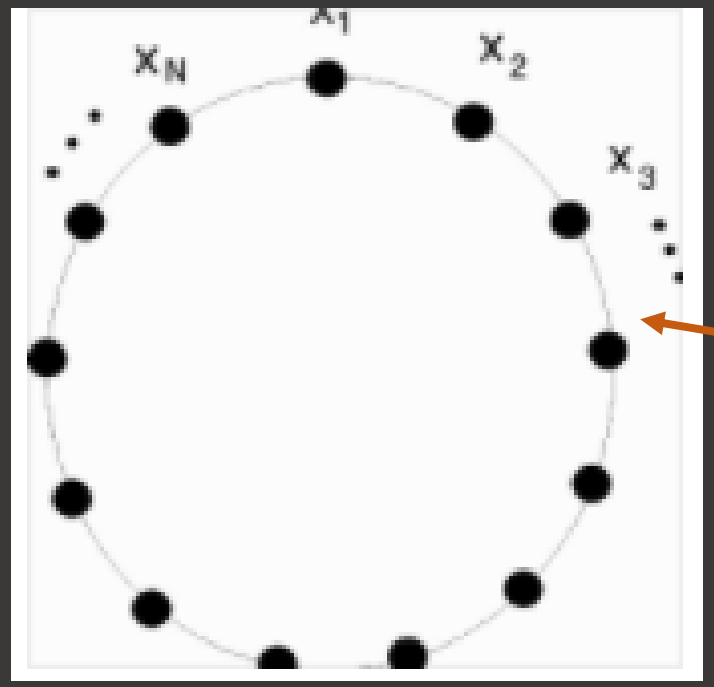
40 dimensional Lorenz 1996 model

$$dx_i/dt = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F$$

Advection

Diffusion

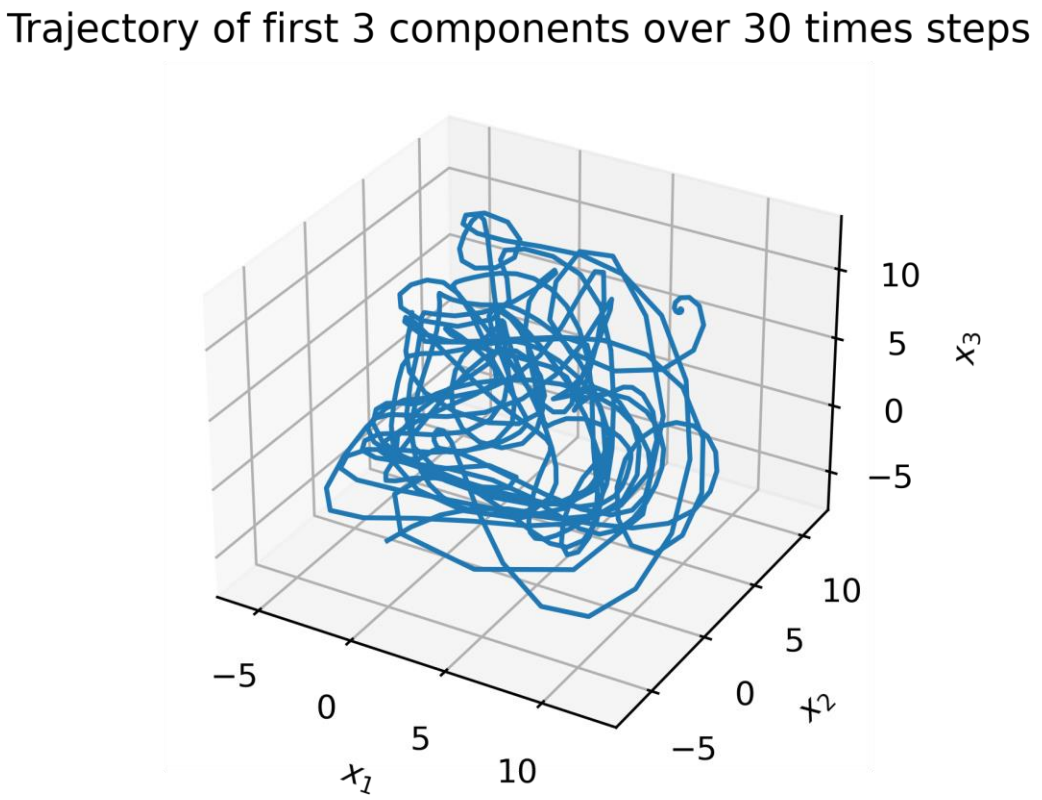
Forcing

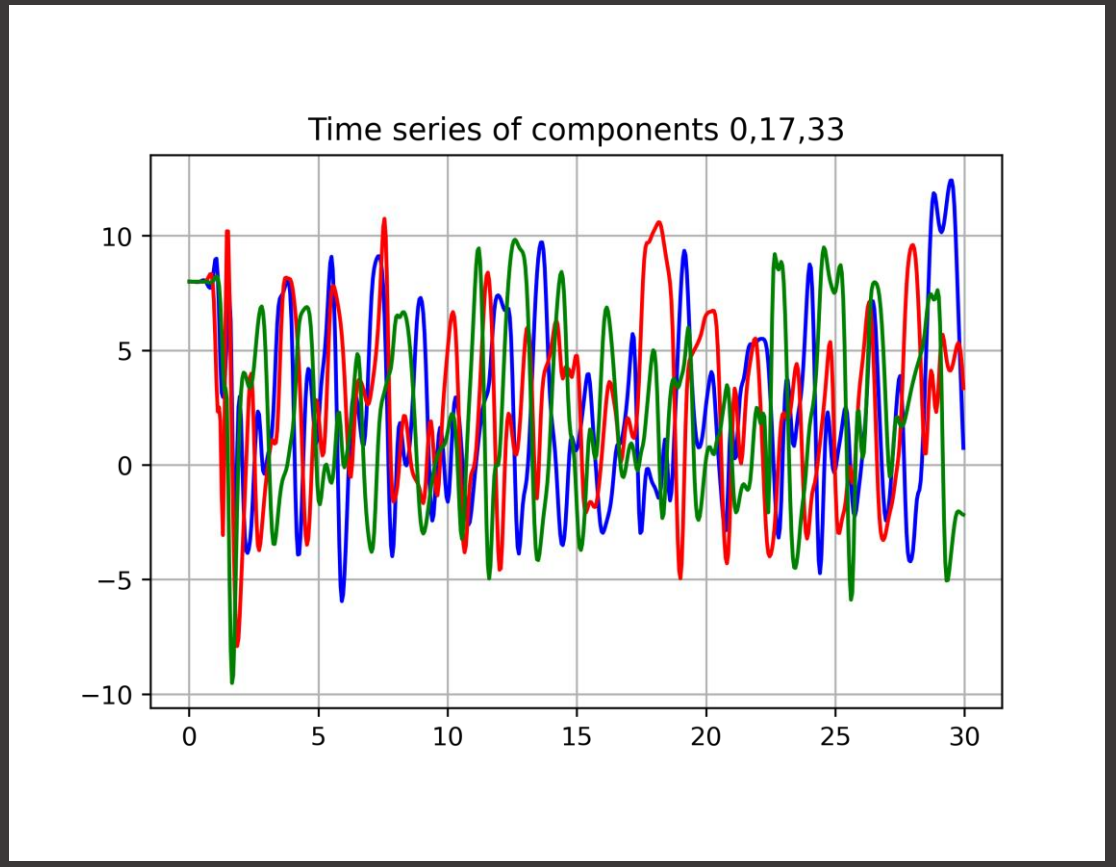


The components can be thought of as dynamical variables around a latitude circle.

Reference :
Seung-Jong Baek et. al, 2005: Localized error bursts in estimating the state of spatiotemporal chaos.

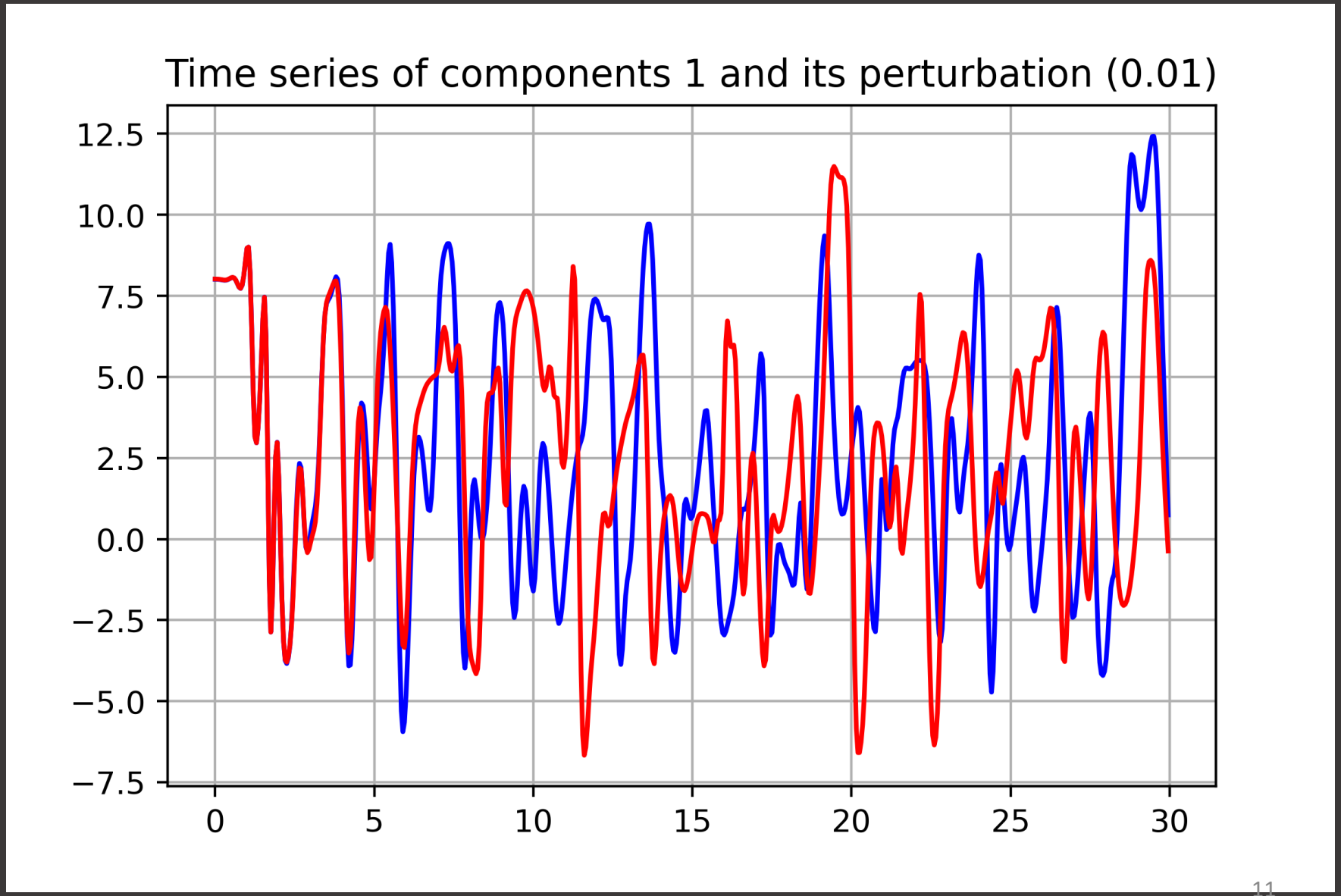
40 dimensional Lorenz 1996 model





Attractor size = 3.78 (Average standard deviation)

Lorenz 1996 model is chaotic !



Computational resources

- I have used Google Collaboratory Pro which is a public resource.
- I have used Keras for DL. Keras is a high level software library which is an interface to TensorFlow DL library.

A GPU consists of thousands of processor cores running in parallel. These are very efficient at some types of scientific calculations (which are used in ML/DL).

CPU consists of a few cores. Each core has a broader instruction set than a core in a GPU. An individual CPU core is much more versatile than a GPU core. However, thousands of cores in a GPU are good at fast execution of specific tasks such as image/video processing, Deep learning etc. All practical problems using DL are carried out using GPUs.

The current work is done using CPUs (not GPUs).

Computational resources : Google Colab Pro

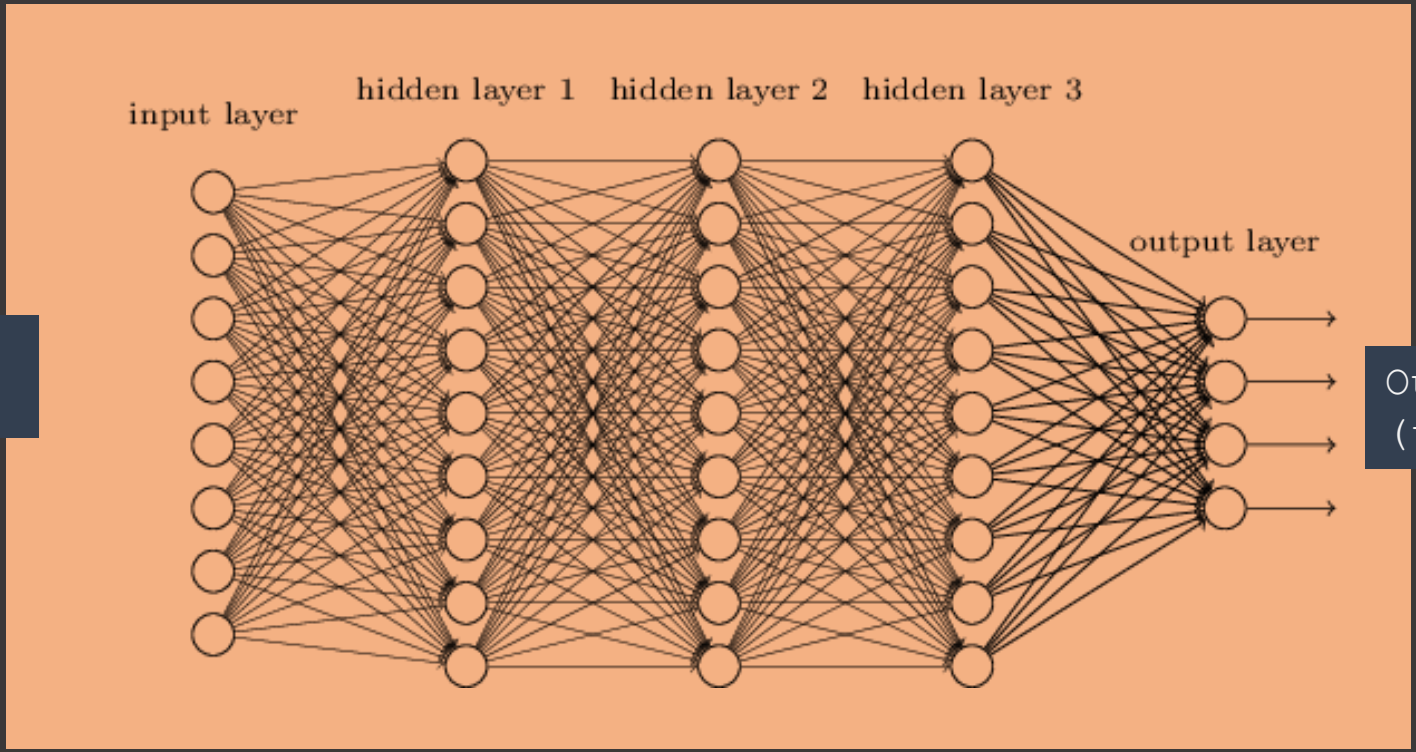
	Colab Free	Colab Pro	Colab Pro +
Guarantee of resources	Low	High	Even Higher
GPU	K80	K80, T4 and P100	K80, T4 and P100
RAM	16 GB	32 GB	52 GB
Runtime	12 hours	24 hours	24 hours
Background execution	No	No	Yes
Costs	Free	9.99$ per month	49.99$ per month
Target group	Casual user	Regular user	Heavy user

Reference :
B. Droste : Google Colab Pro+ : Is it worth $49.99 ?
https://towardsdatascience.com/google-colab-pro-is-it-worth-49-99-c542770b8e56

How does Deep Learning (DL) work ?

- Each connection has a weight associated with it.
- The weights are initialized with random values.
- The predictors of samples are inputted and the DL calculates the output.
- The error between this output and sample target is calculated.
- This error is back propagated to update the weights.
- This process is continued till all the samples are used.

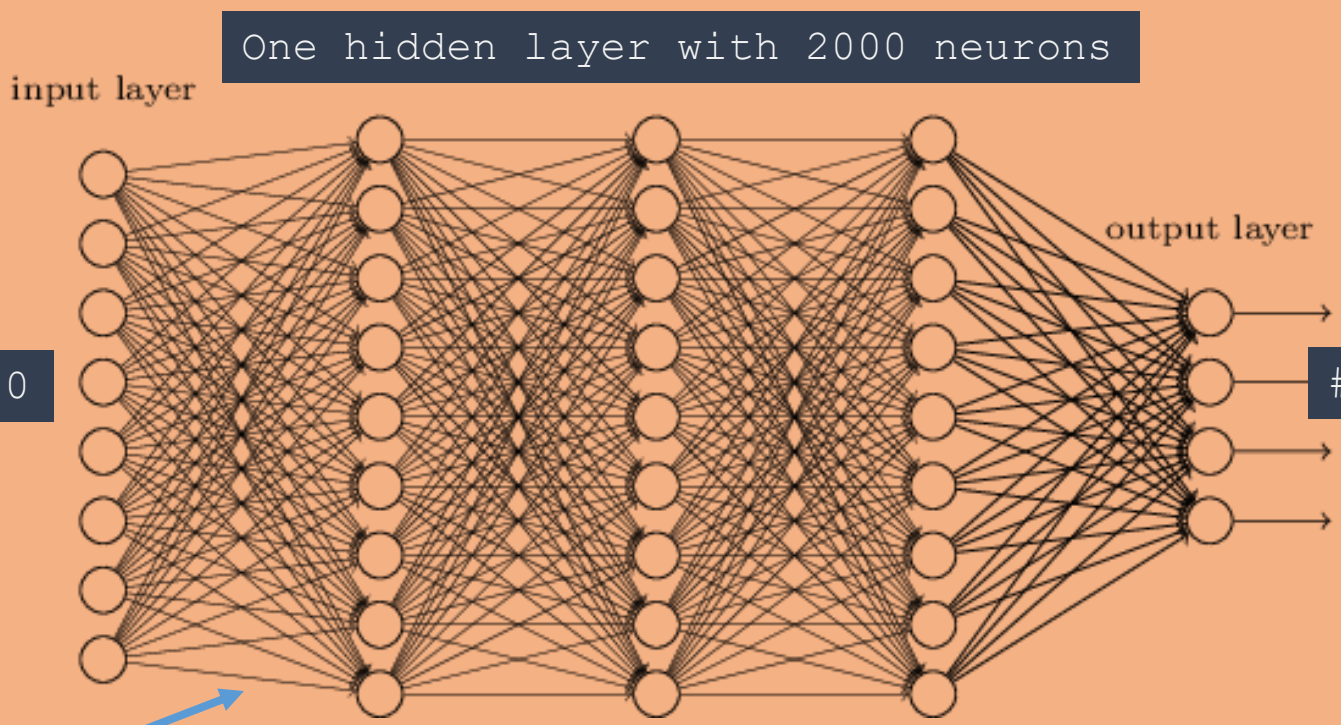
Input predictors



Output predictands (targets)

Deep Learning network architecture for L96

One hidden layer with 2000 neurons



Input neurons = 40

Output neurons = 40

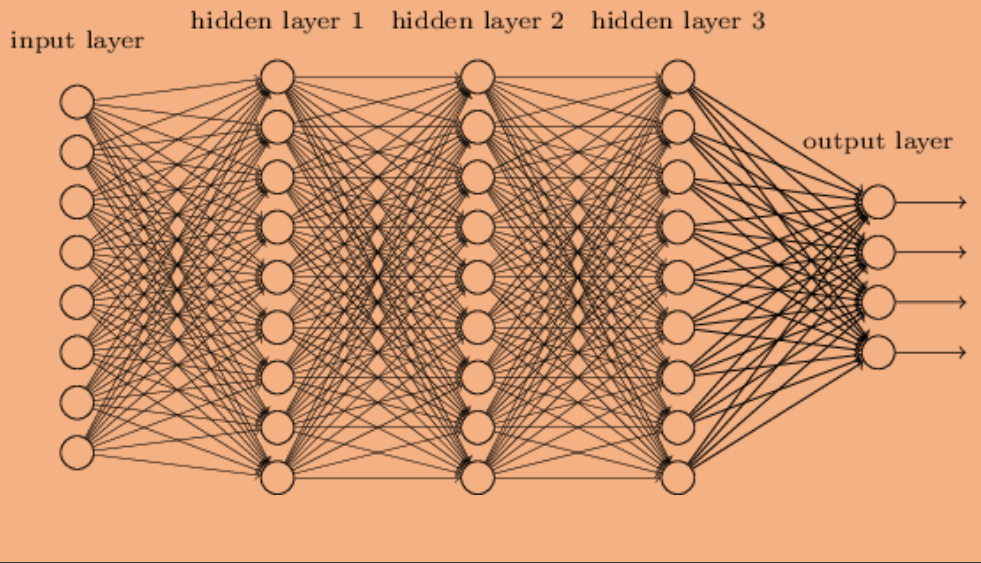
Sigmoid activation

There is a weight associated with each of these connections.

weights = 2 x 40 x 2000

Image copyright : http://neuralnetworksanddeeplearning.com/chap5.html

Deep Learning has many moving parts !



Training a DL involves some trial and error to tune these hyperparameters.

One epoch uses all the data for training.

One epoch is divided into batches of 1000 each (The gradient is calculated over one batch).

How many hidden layers ?	1 layer
How many neurons in hidden layer ?	2000
Which activation function ?	Sigmoid
Batch size ?	1000
Learning rate ?	0.01
Gradient descent optimization ?	RMSprop

More advanced DL techniques like

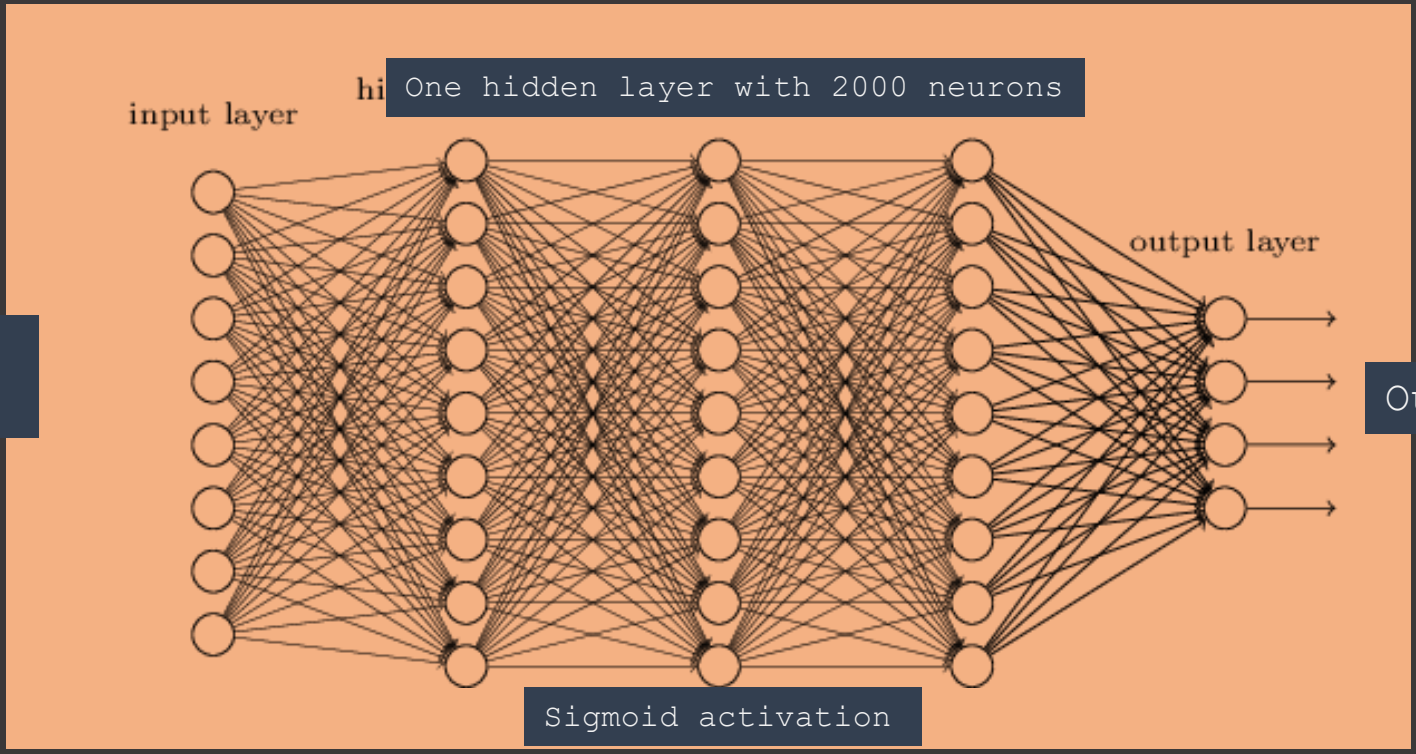
CNN (Convolutional neural network)
RNN (Recurrent neural network)

have many more tunable hyperparameters.

DL training and validation for L96

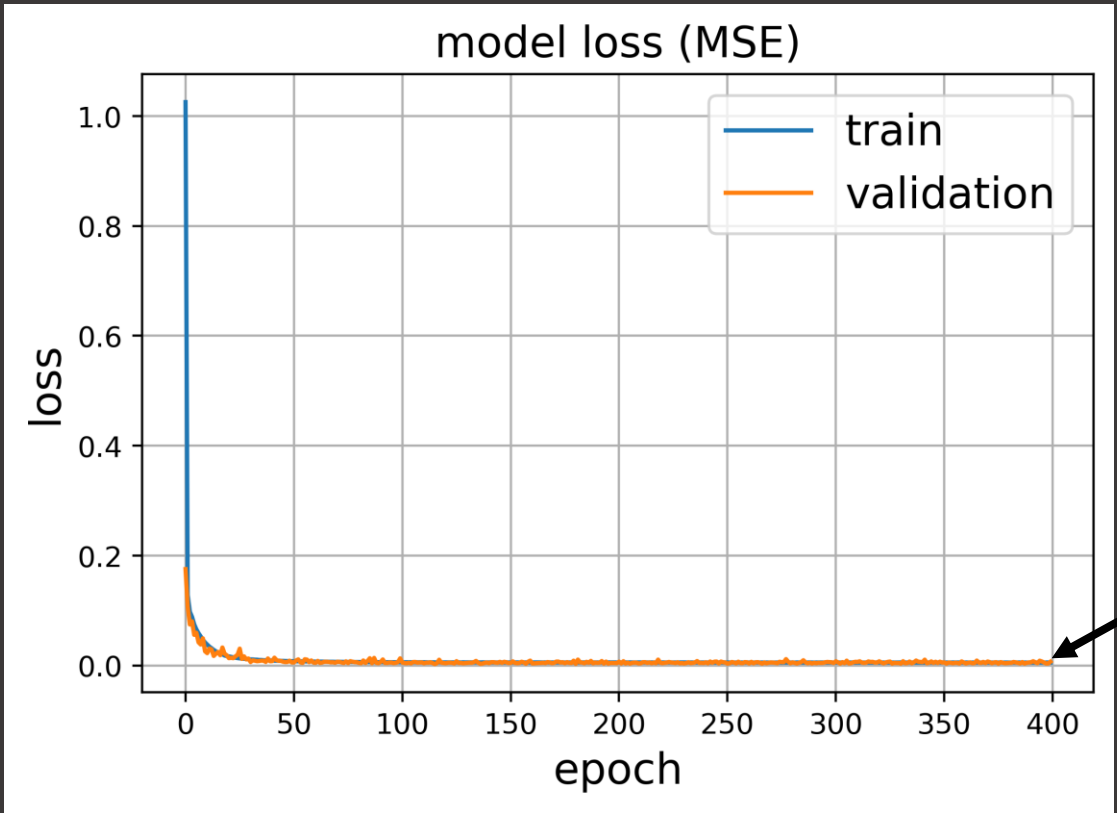
- I generated 10^6, 6 hours lead time forecasts using L96 model.
- These million initial conditions are located in various parts of the phase space of the L96 model. These data (predictors, predictand) = (initial condition, forecast) are known as *samples*.
- These data are then used to train the DL network.
- 70% of the data are used for training. The remaining 30% are used for validation.

Input initial conditions



Output DL forecast.

DL training



Small non-zero simulation error.

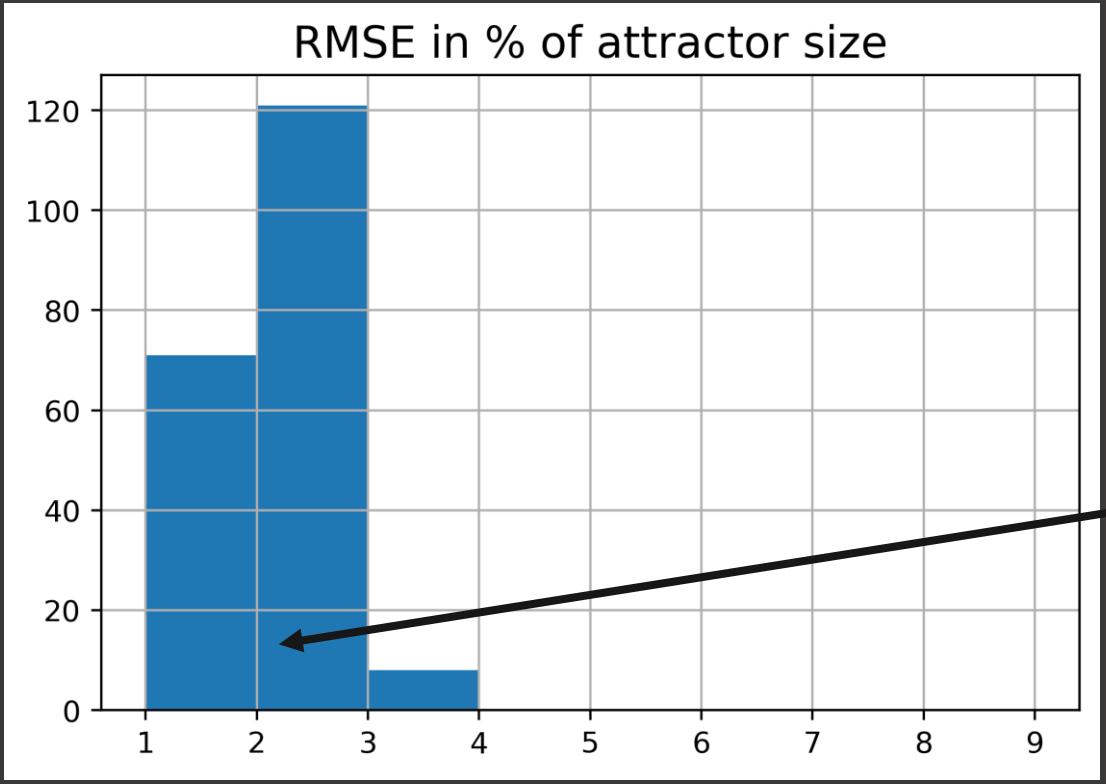
Training and validation errors are comparable – no overfitting !

How small/large is this error ?

After the training is completed the model is saved. The saved model basically contains all the optimized weights.

How big is the simulation error ?

- Launch forecasts starting from 200 different locations in phase space.
- Use DL simulated model to launch forecasts from the same locations.
- Calculate error between the forecasts pairs.

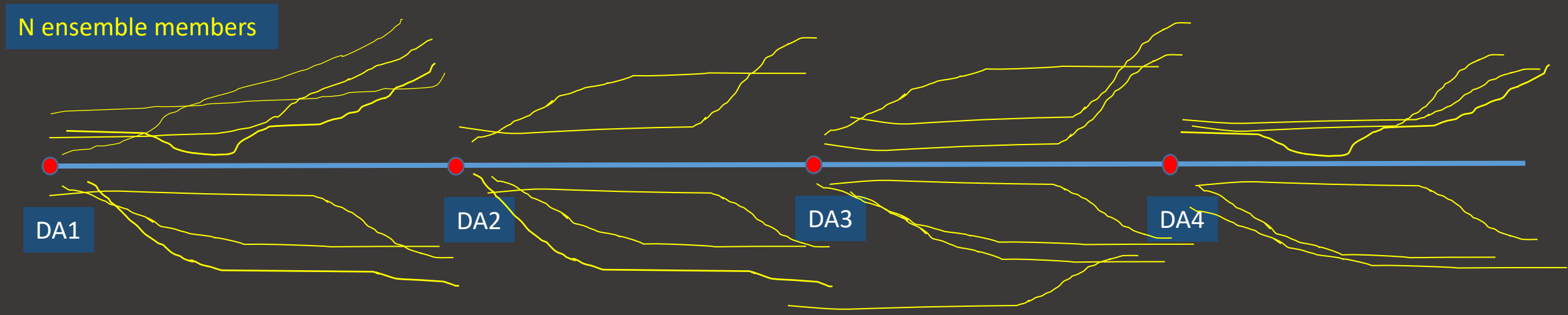


DL simulation RMSE is about 2 % of attractor size.

This error better be smaller than the observation error used in DA.

Computational speed

- The L96 model is quite low dimensional compared to real atmospheric models. The forecast using numerical integration of L96 equations is fast.
- Even so, the DL model is faster by a factor of 3 in executing the forecasts compared to the numerical integration of L96 equations. As shown by Weyn et. al., this factor could be orders of magnitude for real models.
- The aim of this presentation is to demonstrate the principle of using DL to increase the size of the ensemble in an EnKF.



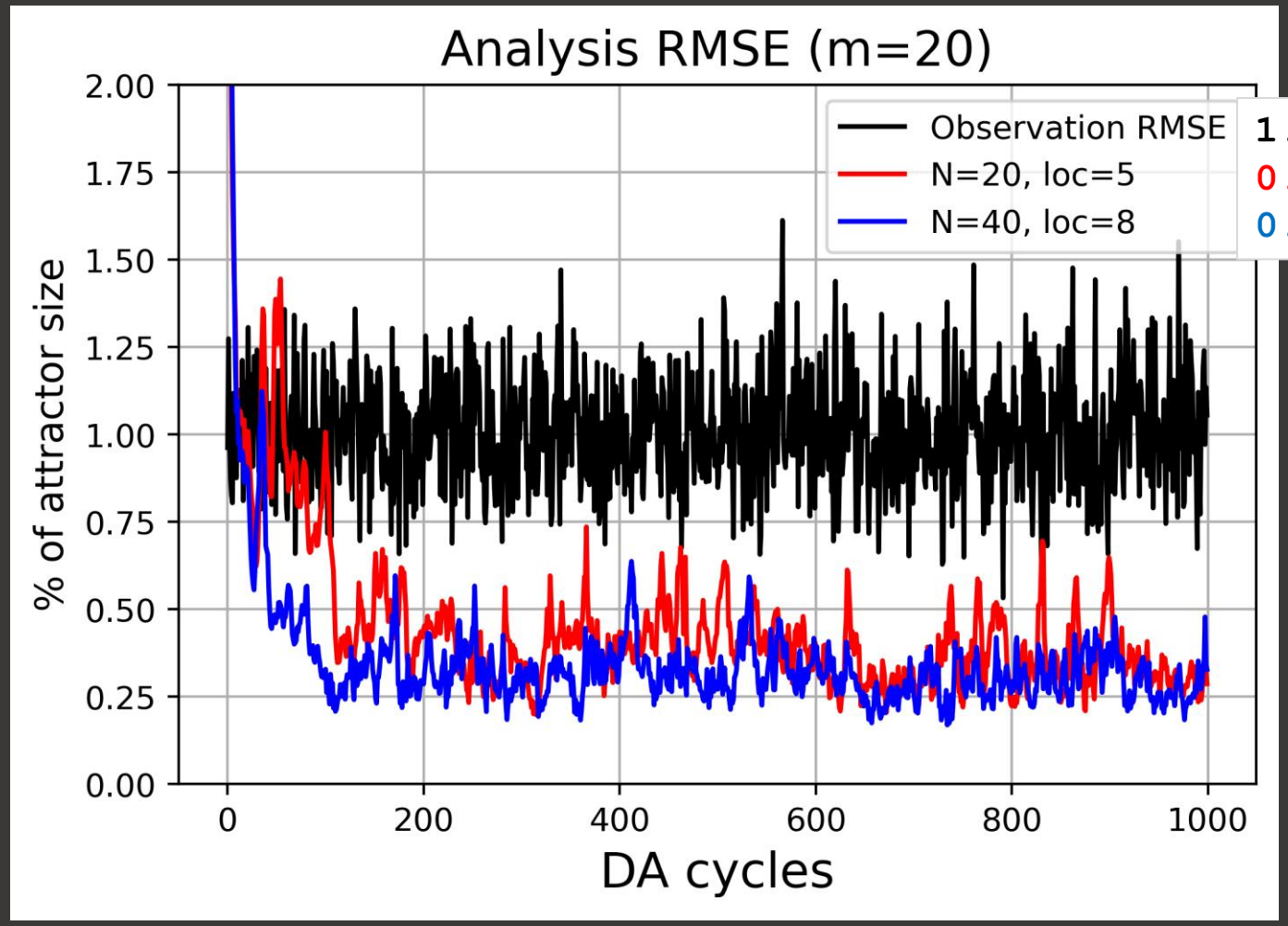
- OSSE setup is used. Observations are drawn from a truth trajectory.
- Perturbed obs EnKF is used.
- Every other component is observed (m=20).
- H is identity.
- R = 1% of attractor size = Observation error. (attractor size is average standard deviation)
- 1000 DA cycles with assimilation every 6 hours.
- Box localization is used.

DA experiments with model ensemble

- With N=40, I determined the optimal localization radius to be 8 in the units of components.

- With N=20, I determined the optimal localization radius to be 5 in the units of components.

DA experiments with model ensemble



DA experiment with DL ensemble

- With N=40, the RMSE diverges.
- With N=20, the RMSE diverges.

The DL simulation error is too large !

Next … mix DL and model ensemble members.

Mixed ensemble experiment

- The N=40 mixed ensemble consists of Nmd=20 and Ndl=20 sub-ensembles.
- However, the RMSE of this ensemble diverges as the DA experiment proceeds.
- This is because of the DL simulation error. Though Nmd=20 ensemble members are perfect, each of the Ndl=20 members have a simulation error.

Mixed ensemble experiment (Recentering)

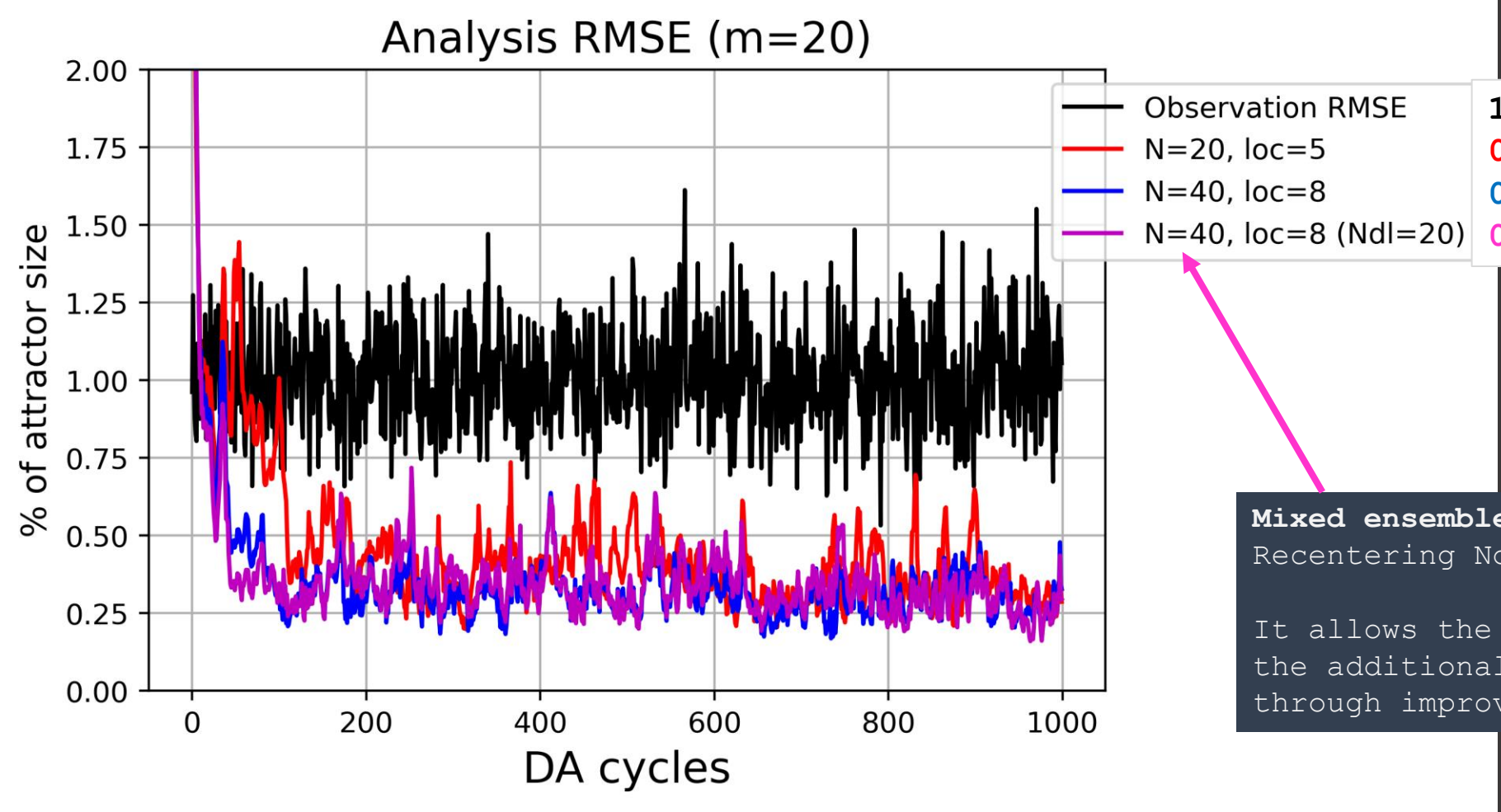
I solved this problem by ***recentering***….

- The mean of the trial model sub-ensemble is calculated. Then the DL members are recentered on this mean.

- This recentering has the effect of correcting a part of the simulation error.

Recentering prevents the RMSE from diverging and one obtains a stable result.

DA experiment with mixed ensemble

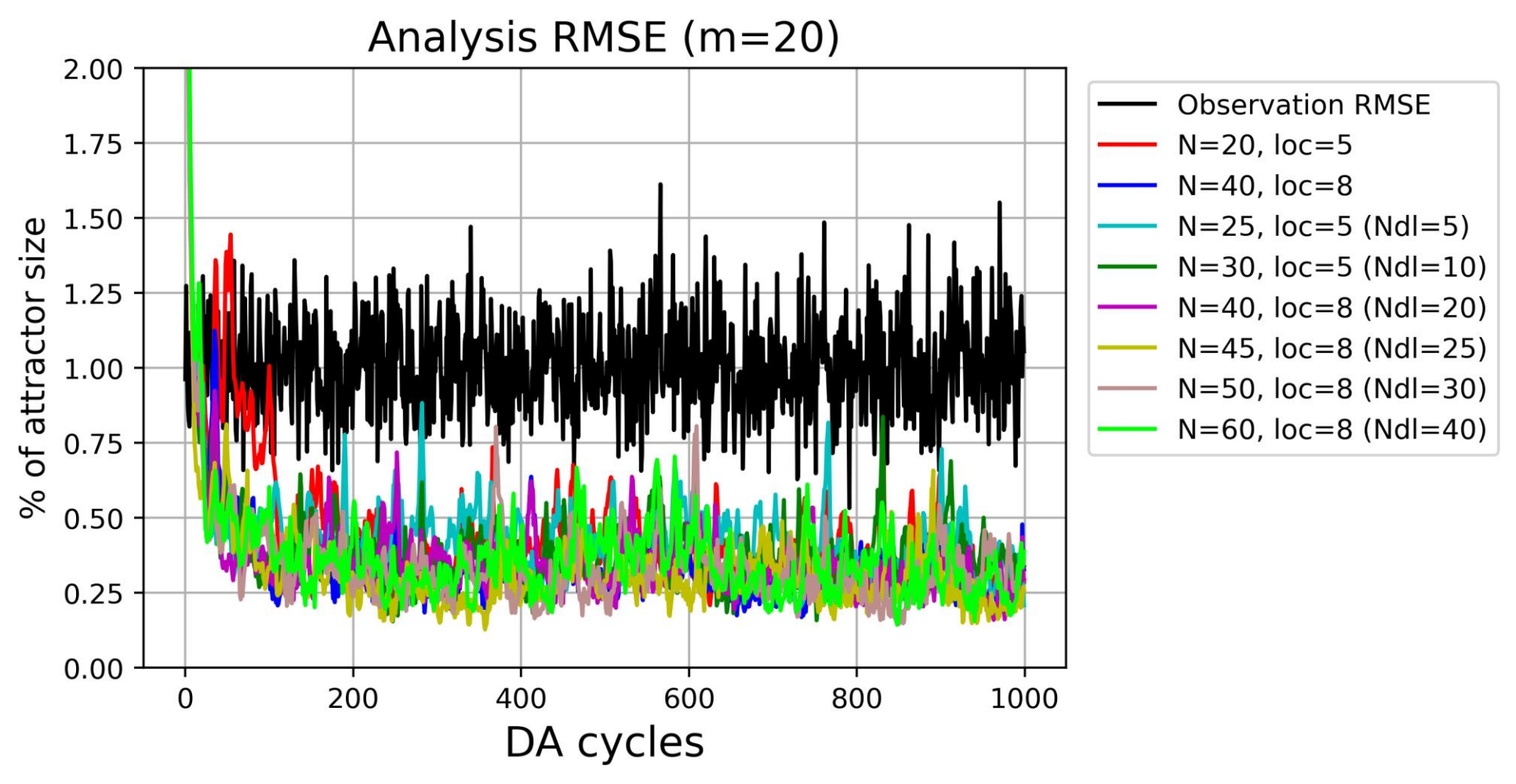


Mixed ensemble approximates the RMSE of a pure model ensemble of same size.

Mixed ensemble :
Recentering Ndl corrects the mean error.

It allows the EnKF to take advantage of the additional 20 ensemble members through improved covariance estimates.

Sensitivity to DL ensemble size



N=45
performs the best !

As Ndl increases the mean correction becomes less effective. Hence RMSE starts increasing.

Recentering

- Recentering results in ensemble mean being estimated only using Nmd=20 ensemble members.
- The improvement in results is due to the improved covariance estimate due to the additional ensemble members from DL. The RMSE decreases from 0.37 (N=20) to 0.28 (N=45, Ndl=25) due to improved covariance estimate.
- If the DL simulation error is made very small, it is possible that recentering might not be necessary. In such a scenario the DL ensemble members will improve the mean estimate of the ensemble apart from the covariance estimate.

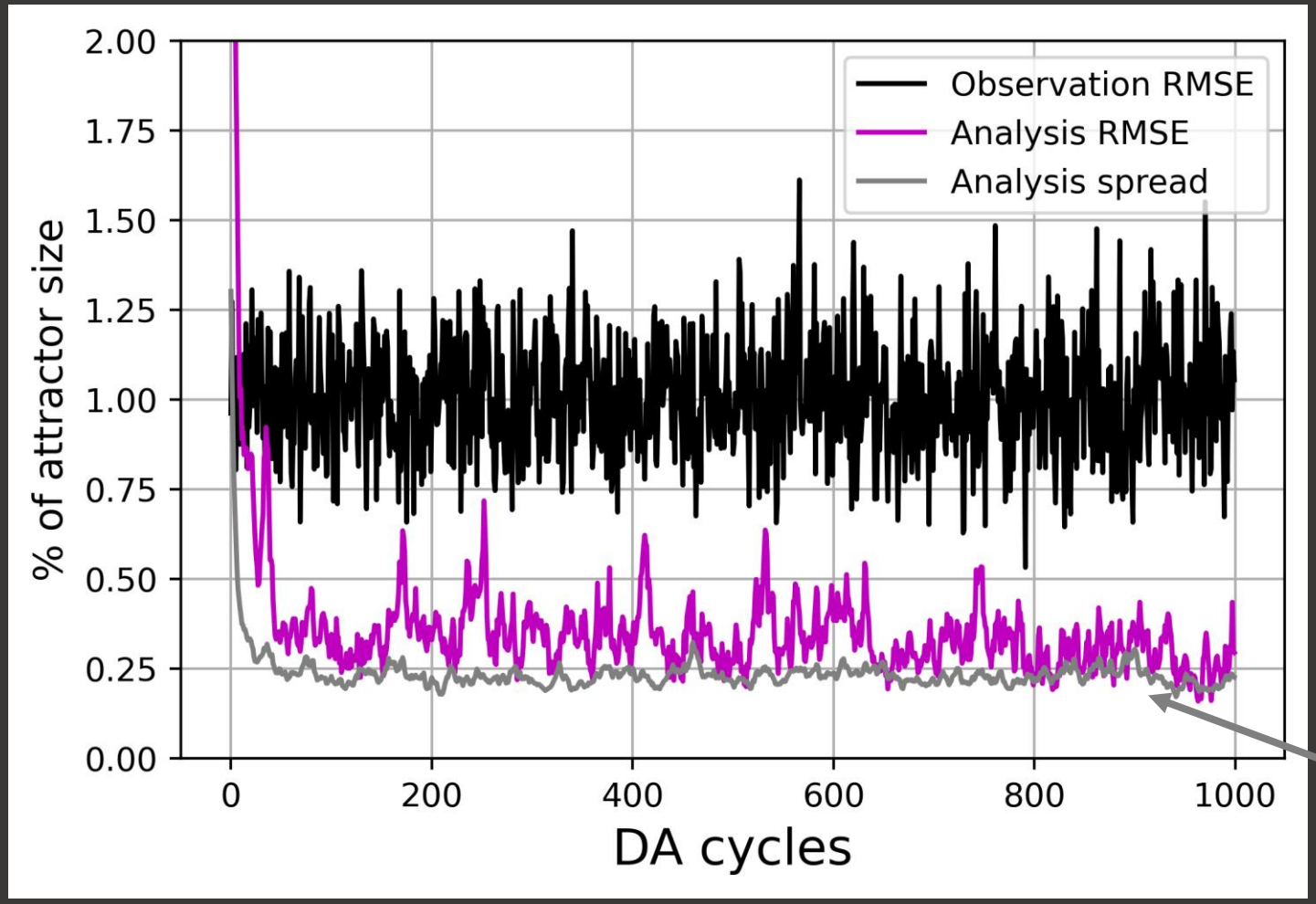
Analysis RMSE for different Obs errors

Obs RMSE in attractor size	0.5	1	2	4
N=40	0.155	0.31	0.62	1.25
N=40 (Ndl=Nmd=20)	0.163	0.33	0.65	1.33
N=45 (Ndl=25,Nmd=20)	0.142	0.28	0.57	1.12

For any Obs error magnitude, the minimum RMSE is realized for the mixed ensemble with N=45 (Ndl=25).

RMSE and Spread

N=45 (Nmd=20, Ndl=25) experiment



Spread is under dispersive.

Inflation might improve results.

Conclusion

- A chaotic dynamical model can be simulated using a DL network given enough data.
- The mixed ensemble with correction applied to the DL subensemble approximates the analysis RMSE of the full model ensemble.
- In spite of the simulation error in each ensemble member the improved covariance estimate allows the mixed ensemble to perform well.
- As the size of the DL subensemble increases this advantage decreases and the RMSE increases. The minimum RMSE is realized for N=45 with Ndl=25.

Further work with L96

These results raise several questions which can be answered using L96 model :

- Sensitivity of the results to localization radius as Ndl is increased beyond 20.
- Use Gaspari-Cohn instead of box localization.
- What is the sensitivity of RMSE to Ndl/Nmd ratio for various values of Nmd ?
- What is the minimum Nmd required to obtain an acceptable RMSE ? Can one use only Nmd=5 and Ndl=35 ? After all, the Nmd ensemble is required to correct the Ndl members using recentering. How about Nmd=1 ?
- Look at the spreads (apart from the RMSE). Does inflation improve results ?
- Sensitivity to the DL simulation error – train another DL network with more training data resulting in lower simulation error. Universal approximation theorem suggests this is possible. How do the DA results change with simulation error ?
- The DL network can be optimized by making it deeper (and slimmer).
- Sensitivity of these results to observing fewer components (m).

Ultimate objective …

I am using L96 model to test out these ideas. The ultimate objective is to use this technique in the operational LETKF/EnVAR system at ECCC.

- Currently the LETKF uses 256 members.
- The goal is to increase this ensemble size using the DL technique as discussed in this presentation.
- ECCC's operational weather model is GEM. The archived forecasts of this model over the past 20 years are available. The first step is to train a DL model to simulate GEM's 6 hour forecast using this archived data.
- I will be using a CNN/RNN DL architecture to simulate GEM forecast.
- The archived GEM data corresponds to different GEM versions. The model parameters can be made part of predictors.

Any suggestions/questions/criticisms are welcome.
Thanks.

References

https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/

Michael Nielsen
http://neuralnetworksanddeeplearning.com/chap4.html

https://www.omnisci.com/technical-glossary/cpu-vs-gpu

Lorenz, Edward (1996) : Predictability - A problem partly solved, *Seminar on Predictability, Vol. I, ECMWF*.

Côté, J., Gravel, S., Methot, A., Patoine, A., Roch, M., and Staniforth, A. (1998): The operational CMC-MRB Global Environmental Multiscale (GEM) model. Part I: Design considerations and formulation, Mon. Weather Rev., 126, 1373-1395, https://doi.org/10.1175/1520-0493(1998)126<1373:TOCMGE>2.0.CO;2,