MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Data Assimilation by EnKF and Neural Networks

for Geophysical Models

Haroldo F. de Campos Velho

INPE: National Institute for Space Research - BRAZIL

E-mail: haroldo.camposvelho@inpe.br

Presentation outline

- Motivation: machine learning applications
  - Climate precipitation prediction on South America
- Neural Networks for Data Assimilation
  - Global atmospheric model: SPEED model
  - Global atmospheric model: COAPS-FSU model
  - COAPS-FSU model: ensemble prediction
  - Meso-scale atmosferic model: WRF-NCAR model
  - Ocean circulation model: Shallow water – FPGA
- Final remarks

Research team

- Haroldo F. de Campos Velho
  INPE: National Institute for Space Research – Brazil
  E-mail: haroldo.camposvelho@inpe.br

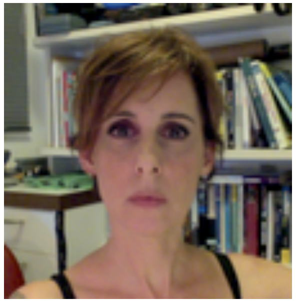
- Rosangela S. C. Cintra
  INPE: National Institute for Space Research – Brazil
  E-mail: rocintra@gmail.com

- Steven Cocke
  FSU: Florida State University – USA
  E-mail: scocke@fsu.edu

Research team

- Juliana A. Anochi
  INPE: National Institute for Space Research – Brazil
  E-mail: juliana.anochi@inpe.br

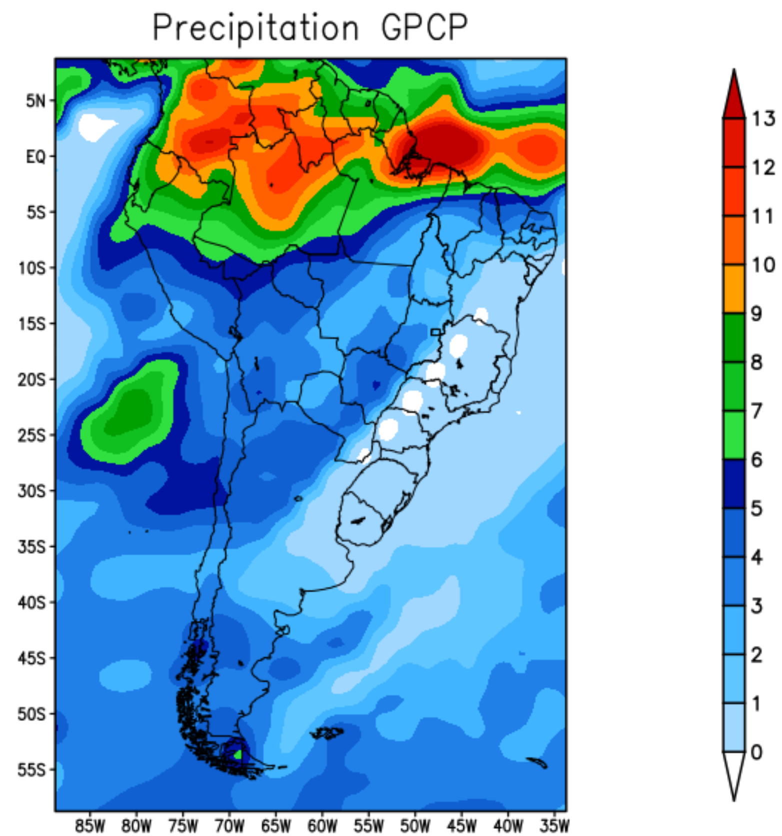
- Vinicius A. Almeida
  UFRJ: Federal University of Rio de Janeiro – Brazil
  E-mail: vinicius@lma.ufrj.br



- Sabrina B. M. Sambatti
  Independent researcher (MPCA developer) – Brazil
  E-mail:

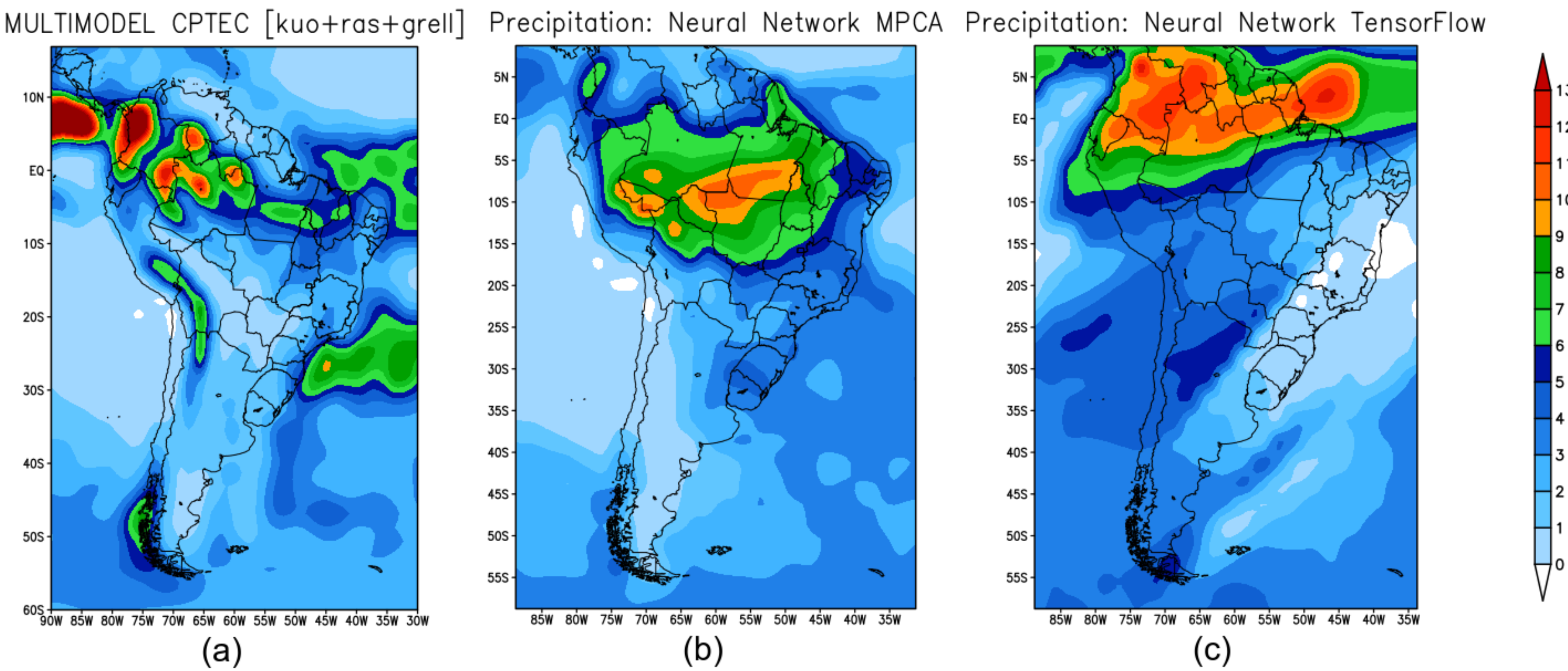
Machine learning for meteorology: motivation

- Seasonal climate precipitation prediction over South America
  - Observation: Fall 2019 – GPCP/NOAA



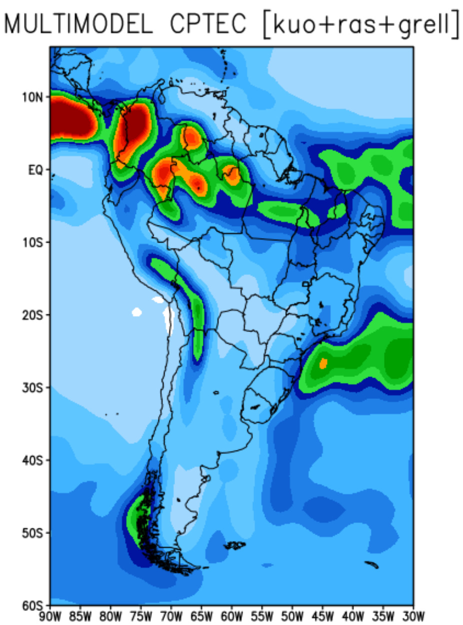
Machine learning for meteorology: motivation

- Seasonal climate precipitation prediction on South America
  - Observation: Fall 2019 – GPCP/NOAA
  - Prediction by BAM-model, NN-MPCA, NN-TensorFlow

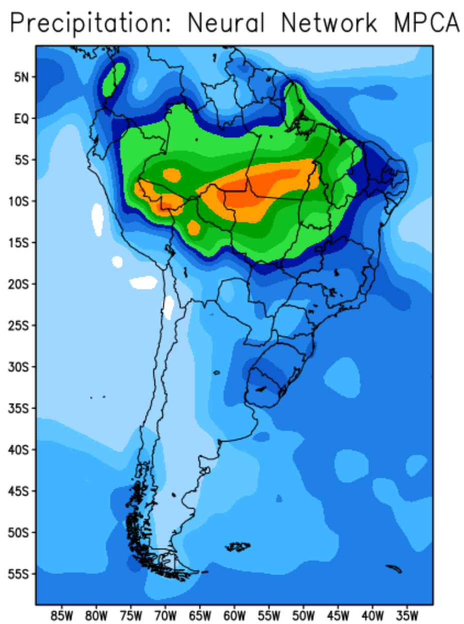


Machine learning for meteorology: motivation

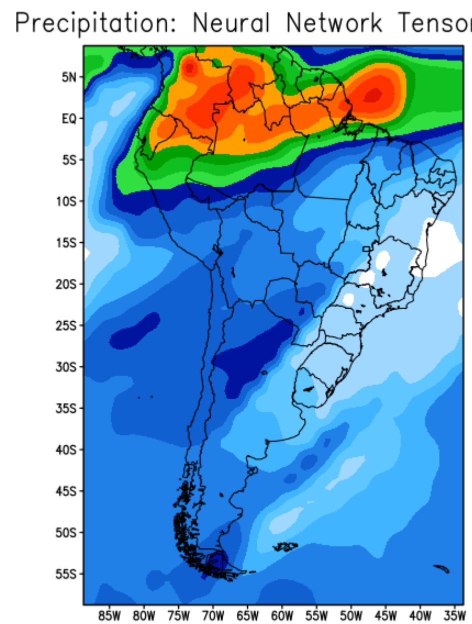
- Seasonal climate precipitation on South America
  - Observation: Fall 2019 – GPCP/NOAA
  - Prediction by BAM-model, NN-MPCA, NN-TensorFlow



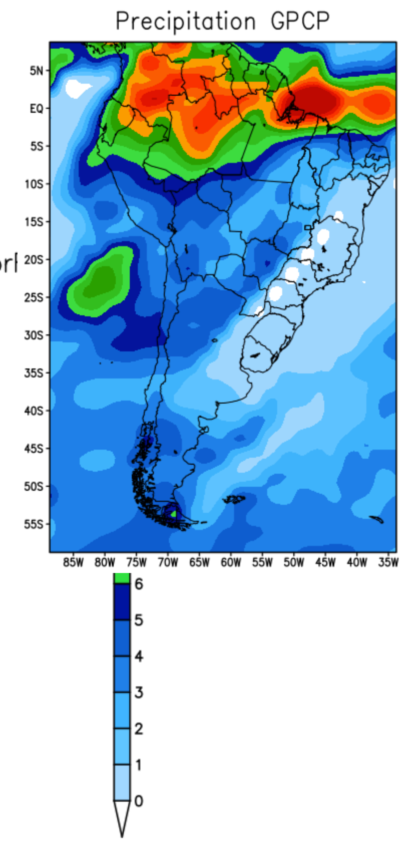
(a)



(b)

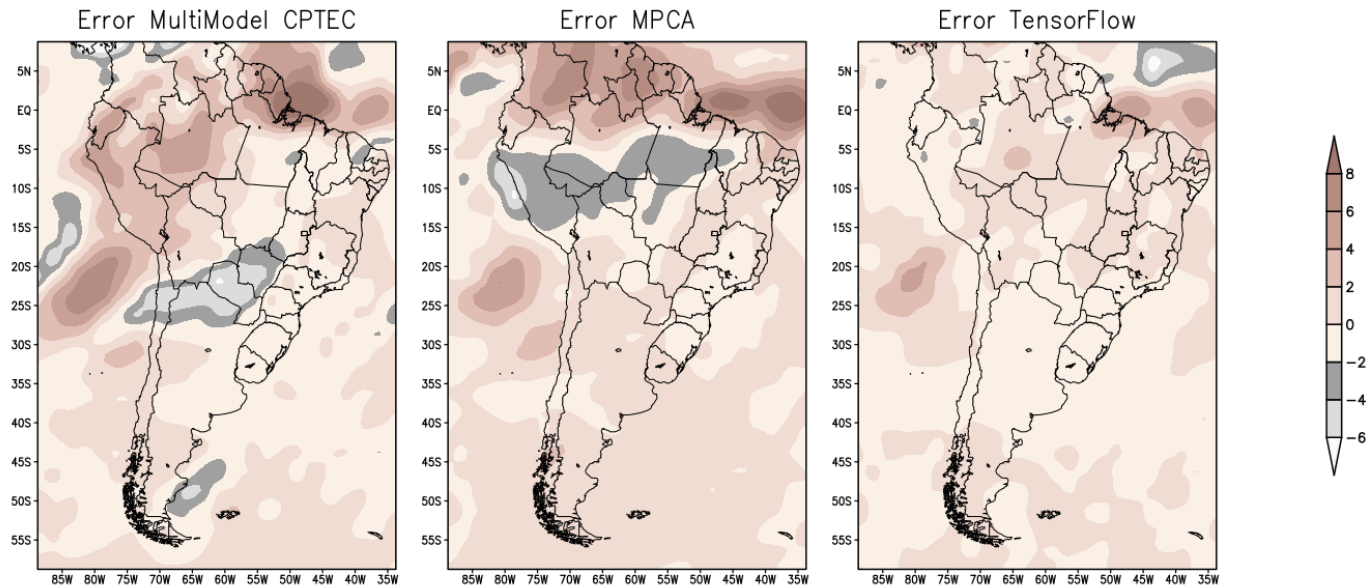


(c)

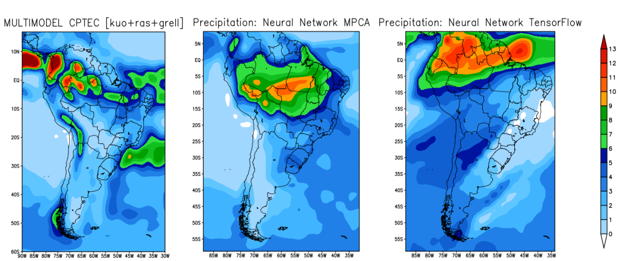


Machine learning for meteorology: motivation

- Seasonal climate precipitation prediction on South America
  - Observation: Fall 2019 – GPCP/NOAA



(a) (b) (c)



(a) (b) (c)

Machine learning for meteorology: motivation

- Seasonal climate precipitation prediction on South America
  - Observation: Fall 2019 – GPCP/NOAA
  - **Performance: RMSE**

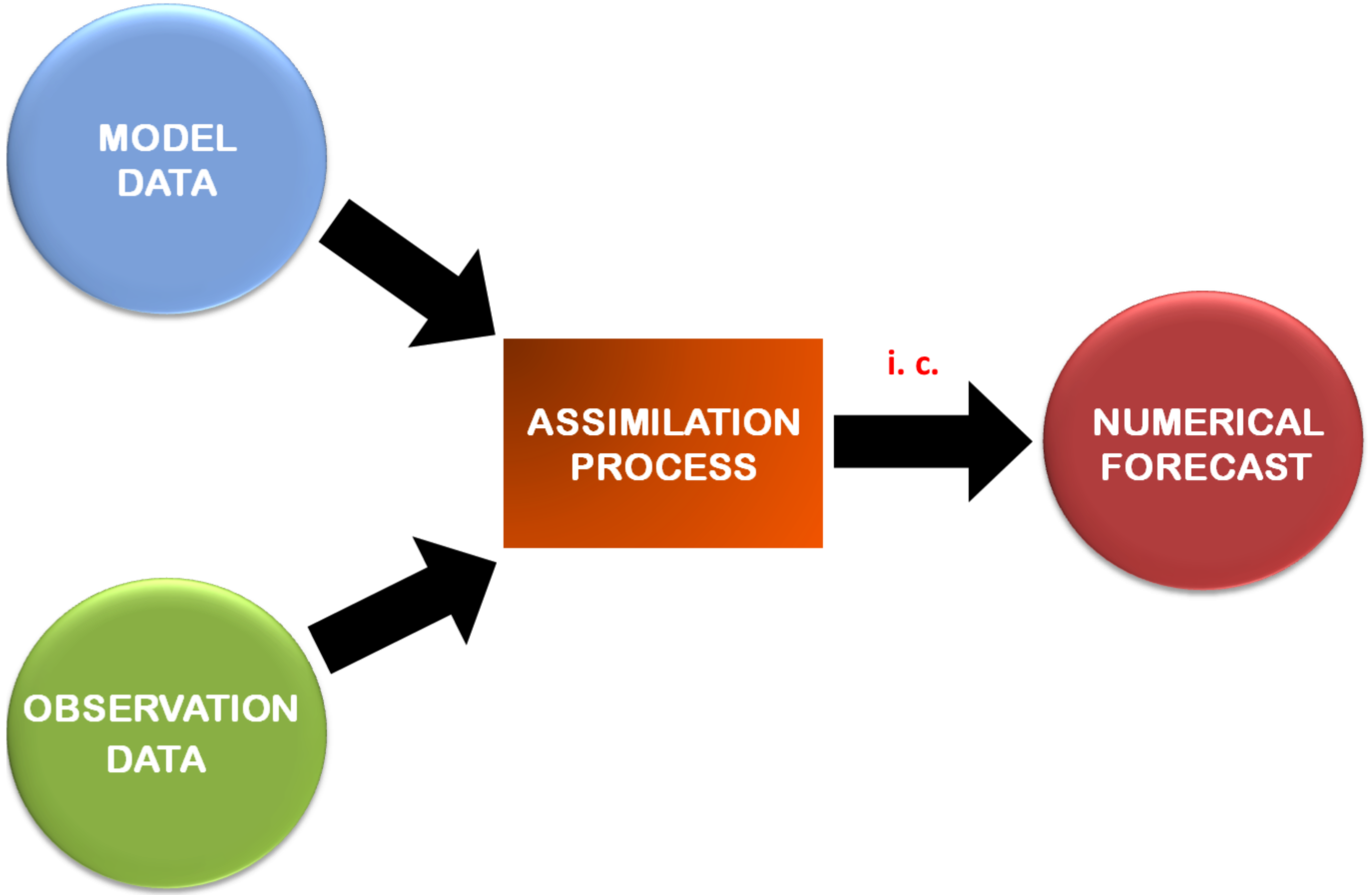
MODELS:	BAM-model	NN-MPCA	NN-TensorFlow
RMSE:	6.30	5.06	0.86

  - **Performance: CPU-time**

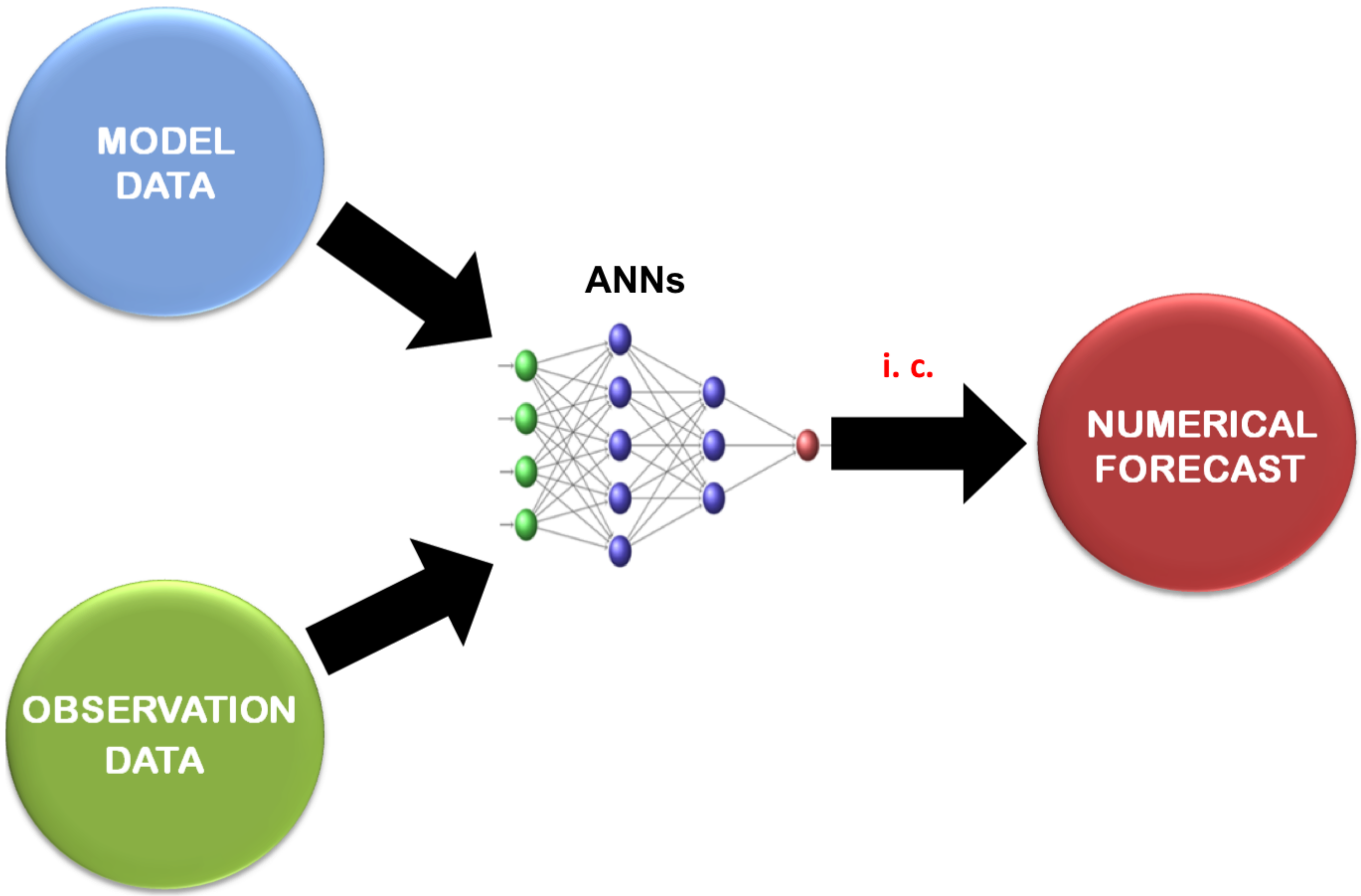
MODELS Hardware	BAM-model Cray X50 120-cores	NN-MPCA Laptop Intel 1-core	NN-TensorFlow Colab Intel 1-core
CPU-time	9.60×10^3 sec	20.19 sec	0.15 sec

Data assimilation – concept

Data assimilation – concept



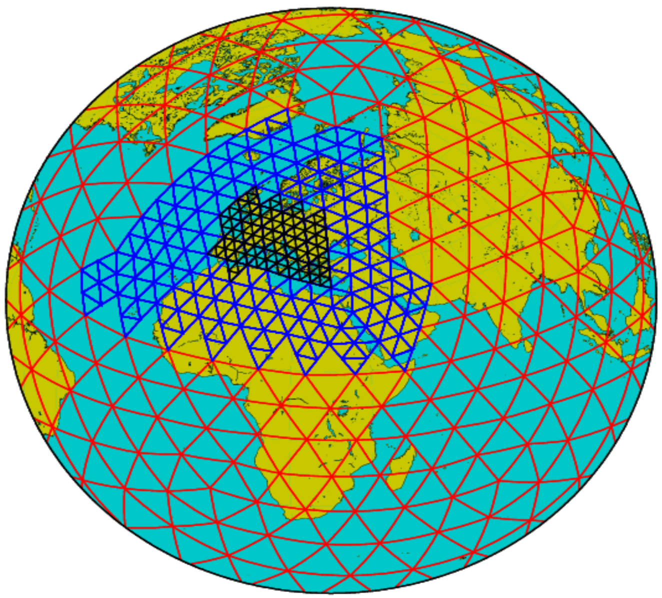
Data assimilation – concept



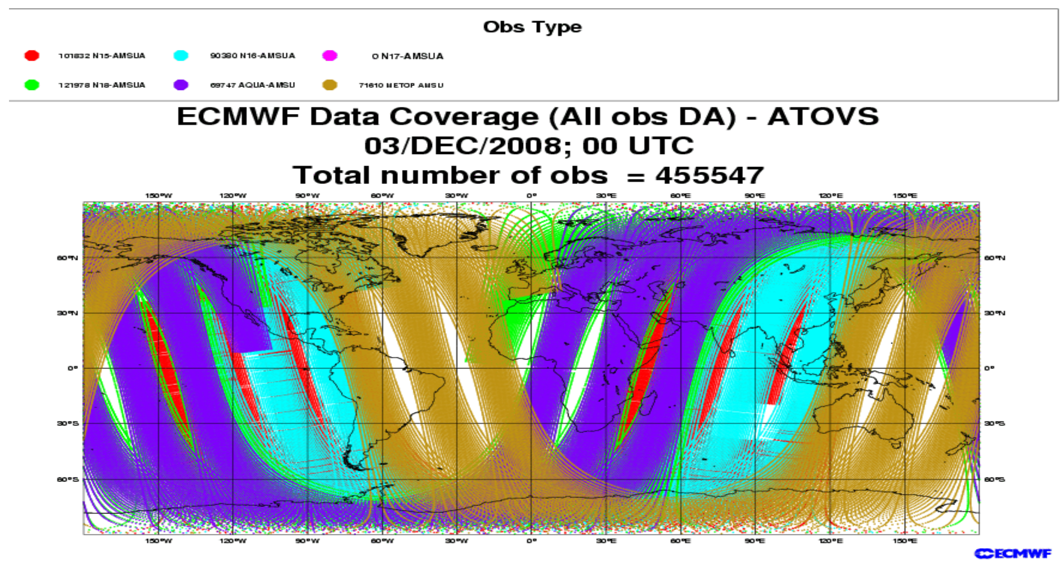
Data assimilation (DA) – methods

- Newtonian relaxation (nudging)
- Statistical (“optimal”) interpolation
- Kalman filter
- Variational method: 3D and 4D
- New methods for data assimilation:
    - Ensemble Kalman filter
    - Hybrid method: variational + EnKF
    - Particle filter
    - **Artificial neural networks**

Why? Exponential growth for the available data



Numerical models with very high resolution



Number of observation are increasing: different satellites with thousands of bands, sensor cost decreasing.

DA: neural networks – our methodology

- We are using supervised neural networks
- We use NN for emulating another technique
- Why to emulate another technique?
  For saving processing time – at least!
- Database: a set of predictions, observations, analysis
- Domain decomposition
  - Each subdomain with different NN
  - Assimilation for each model grid point
  - Automatic configuration for all neural networks

Finding an OPTIMAL neural network

- Design of supervised neural network:

  Optimization problem – cost function:

$$E_{trein} = \frac{1}{N}\sum_{k=1}^{N}(d_k - s_k)^2$$

$$E_{gen} = \frac{1}{(M-N+1)}\sum_{k=N+1}^{M}(d_k - s_k)^2$$

$$F_{obj} = penalty * \frac{[\rho_1 * E_{trein} + \rho_2 * E_{gen}]}{\rho_1 + \rho_2}$$

$$penalty = \underbrace{\left(c_1 * \left(e^{\#neuron}\right)^2\right)}_{\text{complexity factor-1}} \times \underbrace{\left(c_2 * (\#epoch)\right)}_{\text{complexity factor-2}} + 1$$

MPCA: Multi-Particle Collision Algorithm

Available for download:
www.epacis.net/jcis/PDF_JCIS/JCIS11-art.01.pdf

PAN-AMERICAN ASSOCIATION OF COMPUTATIONAL INTERDISCIPLINARY SCIENCES

Journal of Computational Interdisciplinary Sciences (2008) 1(1): 3-10

ISSN 1983-8409
http://epacis.org

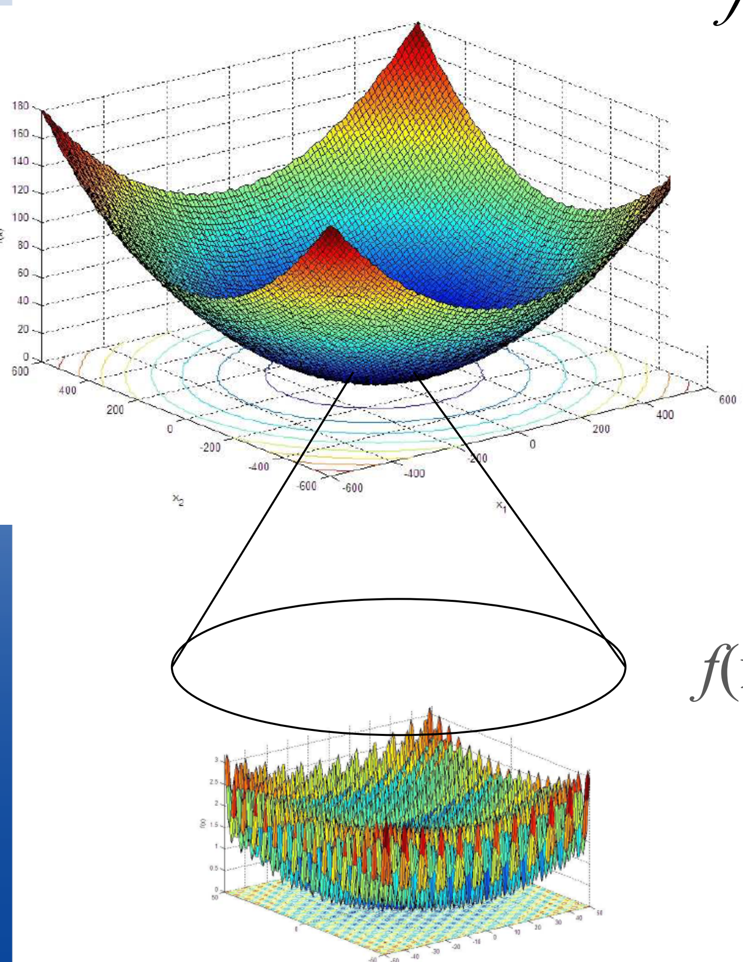
A new multi-particle collision algorithm for optimization in a high performance environment

Eduardo Fávero Pacheco da Luz, José Carlos Becceneri and Haroldo Fraga de Campos Velho

Manuscript received on July 31, 2008 / accepted on October 5, 2008

PCA vs MPCA (2)

Griewank function



$$f(x_1,\ldots,x_n) = 1 + \sum_{j=1}^{n} \frac{x_1^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

$$\left\|(x_1,\ldots,x_2)\right\|_2^2 \leq 600$$

$$\min : (0,\cdots,0), \quad f(0,) = 0$$

PCA	MPCA
(-3.14, 4.43)	$(-1.8\text{x}10^{-8}, -3.3\text{x}10^{-8})$
$f(x_1,x_2) = 7.4 \times 10^{-3}$	$f(x_1,x_2) = 3.3 \times 10^{-16}$

Finding an OPTIMAL neural network

- Supervised neural network: Multi-Layer Perceptron (MLP)

MPCA solution

# hidden layers	# neurons layer-1	# neurons layer-2	# neurons layer-3	Activation function	Momentum ratio	Learning ratio

Parameters	Vallue
Number of hidden layers	|1| |2| |3|
Number of neurons for each layer	|1| ... |32|
Learning ratio	|0| ... |1|
Momentum	|0| ... |0.9|
Activation function	|Tanh| |Log| |Gauss|

Data assimilation – first application

- Kalman filter

$$\boldsymbol{x}_{n+1} = F\left[\boldsymbol{x}_n, t_n\right] \approx \boldsymbol{F}_n + \left.\frac{\partial F}{\partial \boldsymbol{x}}\right|_{t=t_n} \boldsymbol{x}_n + O(\Delta t^2) \approx \boldsymbol{E}_n \boldsymbol{x}_n$$

1. Advance in time:

$$\mathbf{x}^f_{n+1} = \mathbf{F}_n \mathbf{x}_n$$

$$\mathbf{P}^f_{n+1} = \mathbf{F}_n \mathbf{P}^a_n \mathbf{F}^T_n + \mathbf{W}^b_n$$

???

2. Kalman gain

$$\mathbf{G}_{n+1} = \mathbf{P}^f_{n+1} \mathbf{H}^T_{n+1} \left[\mathbf{W}^o_n + \mathbf{H}_{n+1} \mathbf{P}^f_n \mathbf{H}^T_n\right]^{-1}$$

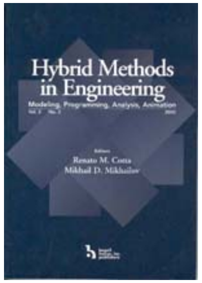
3. Update estimation

$$\mathbf{x}^a_{n+1} = \mathbf{x}^f_{n+1} + \mathbf{G}_{n+1}\left[\mathbf{x}^o_{n+1} - \mathbf{H}_{n+1}\left(\mathbf{x}^f_{n+1}\right)\right]$$

4. Update error covariance

$$\mathbf{P}^a_{n+1} = \left[\mathbf{I} - \mathbf{G}_{n+1} \mathbf{H}^T_n\right] \mathbf{P}^f_{n+1}$$

Bayesian filters



- **Kalman filter**
- Estimating the error modeling co-variance matrix
  - Estimating $\boldsymbol{W}^b$ by parameterization
  - Estimating $\boldsymbol{W}^b$ by Fokker-Planck equation
  - Estimating $\boldsymbol{W}^b$ by ensemble strategy

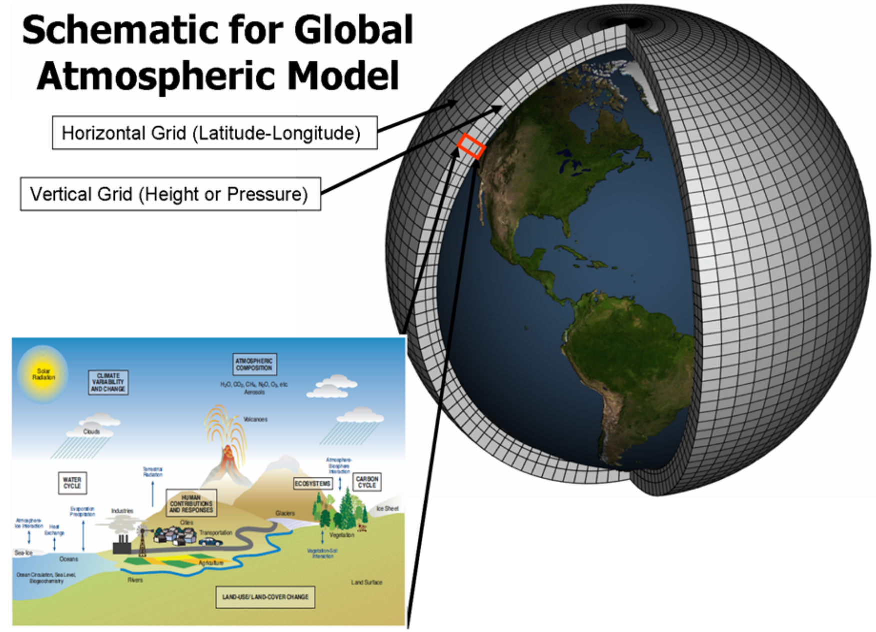
$$\mathbf{W}^b \approx \frac{1}{N_k - m} \sum_{k \neq 1}^{N_k} \left(\mathbf{x}_k^f - \bar{\mathbf{x}}^f\right)\left(\mathbf{x}_k^f - \bar{\mathbf{x}}^f\right)^{\mathrm{T}} \qquad \begin{cases} \bar{\mathbf{x}}: \text{ ensemble average} \\ N_k: \text{ number of members} \\ m = 1 \text{ or } 2 \end{cases}$$

Bayesian filters

- **Ensemble Kalman filter**
- Estimating the error modeling co-variance matrix
  - Estimating $\boldsymbol{W}^b$ by parameterization
  - Estimating $\boldsymbol{W}^b$ by Fokker-Planck equation
  - Estimating $\boldsymbol{W}^b$ by ensemble strategy

$$\mathbf{W}^b \approx \frac{1}{N_k - m} \sum_{k \neq 1}^{N_k} \left(\mathbf{x}_k^f - \overline{\mathbf{x}}^f \right) \left(\mathbf{x}_k^f - \overline{\mathbf{x}}^f \right)^{\mathrm{T}} \quad \begin{cases} \overline{\mathbf{x}} : \text{ ensemble average} \\ N_k : \text{ number of members} \\ m = 1 \text{ or } 2 \end{cases}$$

SPEED model



Forward model (x^f):

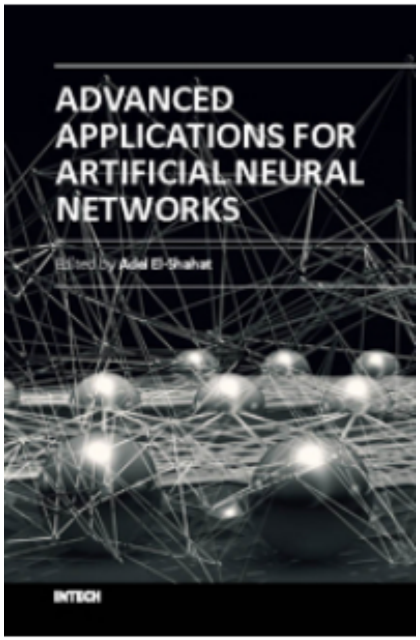
SPEED model

- Atmospheric general circulation model
- 3D spectral model
- simplified parameterization

Vertical coordinates: $\sigma = p_s/p$.
Horizontal coordinates: (lat , long) on a Gaussian grid

The spectral model: T30 horizontal resolution and 7 vertical levels
Observations: 12035 (00 and 12 UTC) = 415 x 4 x 7 + 415
Observations: 2075 (00 and 12 UTC) = 415 x 5 (only surface)

SPEED model



Chapter 14

Data Assimilation by Artificial Neural Networks for an Atmospheric General Circulation Model

Rosangela Saher Cintra and
Haroldo F. de Campos Velho

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.70791

SPEED: atm. general circulation model

Spectral 3D model, with simplified parameterization

$$\frac{\partial \zeta}{\partial t} = -\nabla \cdot (\zeta + f)\mathbf{U} - \mathbf{k} \cdot \nabla \times \left(RT' \nabla lp + \dot{\sigma} \frac{\partial \mathbf{U}}{\partial \sigma} + \mathbf{F} \right)$$

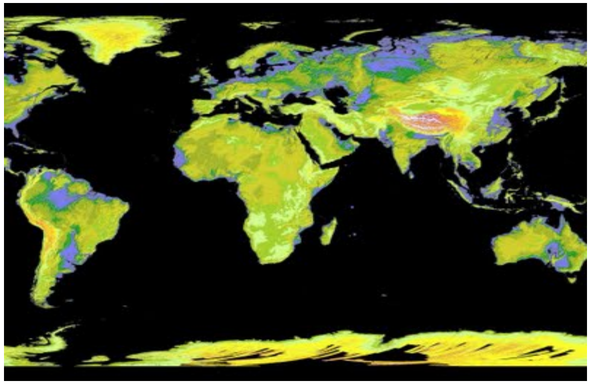
$$\frac{\partial D}{\partial t} = \mathbf{k} \cdot \nabla \times (\zeta + f)\mathbf{U} - \nabla \cdot \left(RT' \nabla lp + \dot{\sigma} \frac{\partial \mathbf{U}}{\partial \sigma} + \mathbf{F} \right) - \nabla^2 (\Phi' + RT_0 \, lp + \frac{1}{2} \mathbf{U} \cdot \mathbf{U})$$

$$\frac{\partial T}{\partial t} = -\nabla \cdot \mathbf{U}T' + T'D + \dot{\sigma}\gamma - \frac{RT}{c_p} \left(D + \frac{\partial \dot{\sigma}}{\partial \sigma} \right)$$

{with: $\phi = gh$; and: $\sigma = p/p_0$}

$$\frac{\partial q}{\partial t} = -D - \frac{\partial \dot{\sigma}}{\partial \sigma} - \mathbf{U} \cdot \nabla \, lp$$

{with: $q = \log(p_0)$}

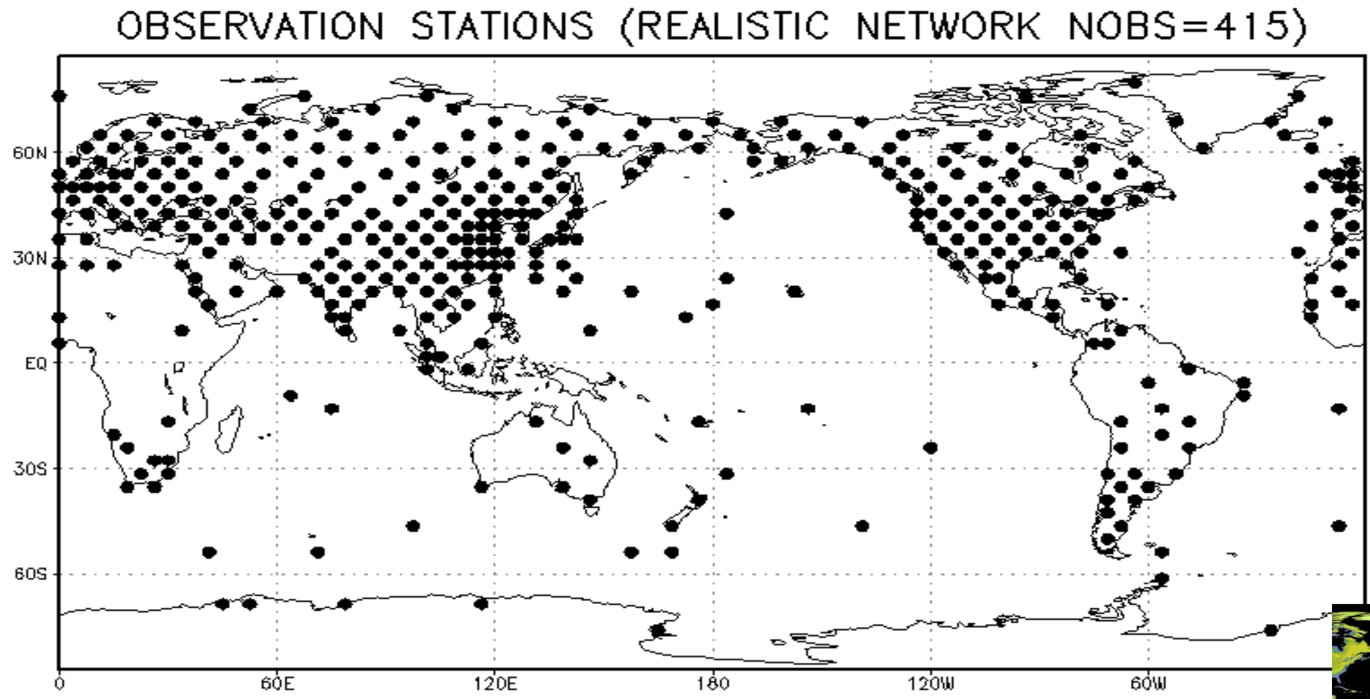


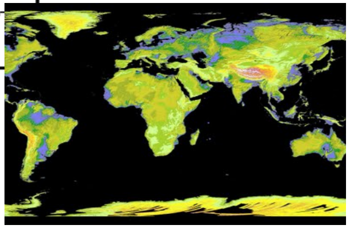
(a) ζ: vorticity
(b) D: divergence
(c) T: temperature
(d) q: moisture

SPEED: atmospheric general circulation model

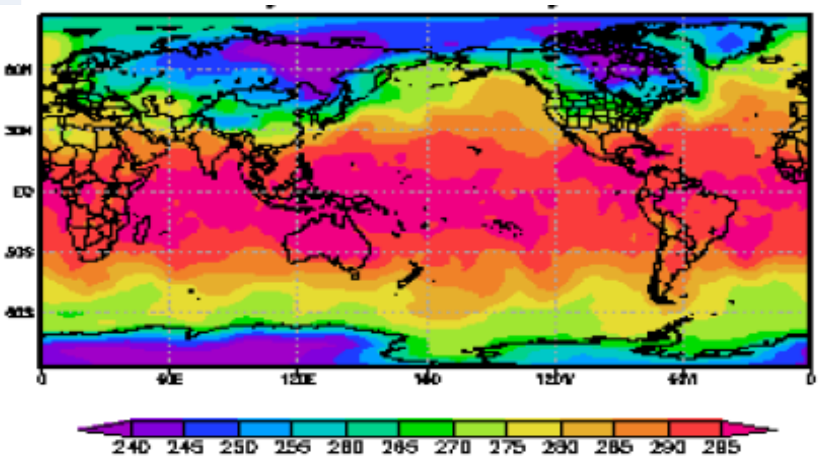
Spectral 3D model, with simplified parameterization

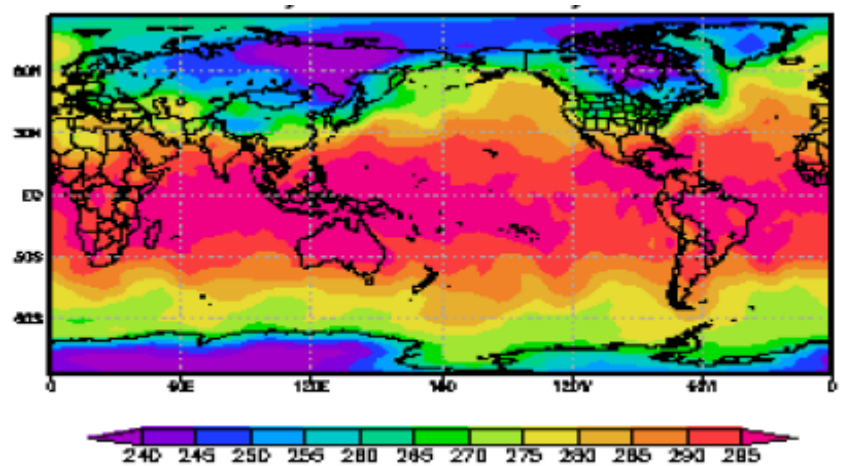
Observation grid: NN emulating LEnTKF

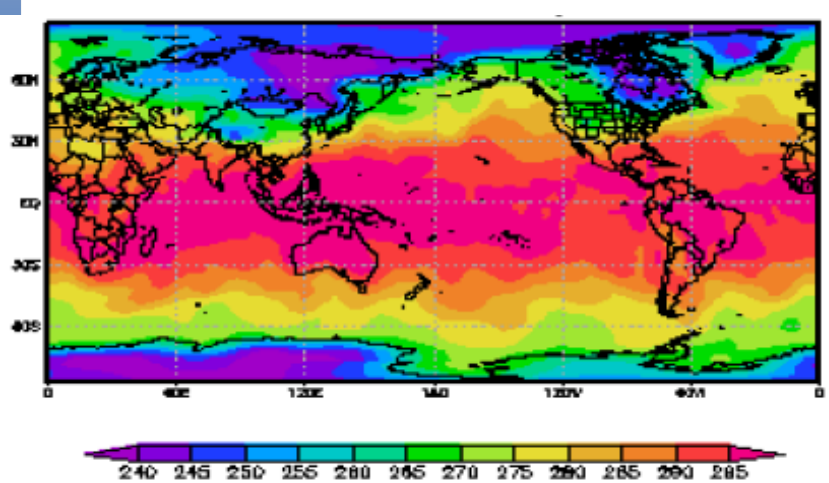




Temperature: assimilation experiment







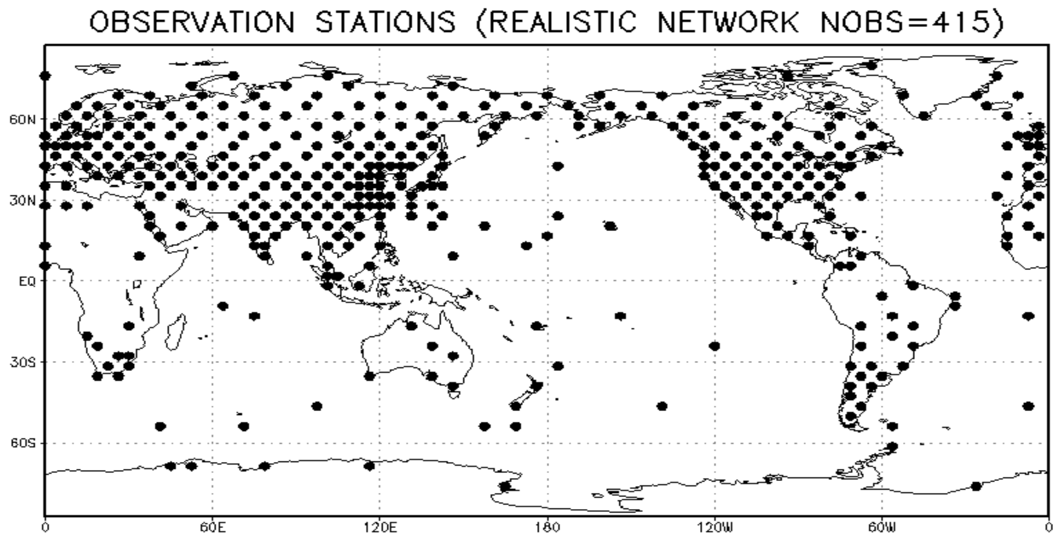
LETKF

neural network

True

Results from Rosangela Cintra PhD thesis (2011)

Experiment: LETKF and neural network



Execution time	
LETKF method	**ANN method**
04:20:39	**00:02:53**
hours : minutes : seconds	

General atmospheric Circulation Modelo 3D (spectral model):

SPEEDY (Simplified Parameterizations primitivE Equation DYnamics)

Gaussian grid: 96 x 48 (horizontal) x 7 lvels (vertical) = T30L7

Total grid points: 32.256 Total de variáveis: 133.632

Observations: (00, 06, 12, 18 UTC) – radiosonders "OMM stations"

Observations: 12035 (00 e 12 UTC) = 415 x 4 x 7 + 415

Observations: 2075 (00 e 12 UTC) = 415 x 5 (only surface)

Results from the Rosangela Cintra's PhD thesis (2011)

Global model for NWP

- **FSU-COAPS global model:** equations

$$
\begin{aligned}
\frac{\partial \zeta}{\partial t} &= -\nabla \cdot (\zeta + f)\vec{v}_H - \vec{k} \cdot \nabla \times \left(RT\nabla q + \dot{\sigma}\frac{\partial \vec{c}_H}{\partial \sigma} - \vec{f} \right) \\
\frac{\partial D}{\partial t} &= \vec{k} \cdot \nabla \times (\zeta + f)\vec{v}_H - \nabla \cdot \left(RT\nabla q + \dot{\sigma}\frac{\partial \vec{c}_H}{\partial \sigma} - \vec{f} \right) - \nabla^2 \left(\phi + \frac{\vec{v}_H \cdot \vec{v}_h}{2} \right) \\
\frac{\partial T}{\partial t} &= -\nabla \cdot (T\vec{v}_H) + TD + \dot{\sigma}\gamma - \frac{RT}{c_p}\left(D + \frac{\partial \dot{\sigma}}{\partial \sigma} + H_T \right) \\
\frac{\partial q}{\partial t} &= -\vec{v}_H \cdot \nabla q - D - \frac{\partial \dot{\sigma}}{\partial \sigma} \\
\sigma \frac{\partial \phi}{\partial \sigma} &= -RT \\
\frac{\partial r}{\partial t} &= -\nabla \cdot (r\vec{v}_H) + rD - \dot{\sigma}\frac{\partial r}{\partial \sigma} + M
\end{aligned}
$$

{with: q = log(p_0)}

{with: $\phi = gh$; and: $\sigma = p/p_0$}

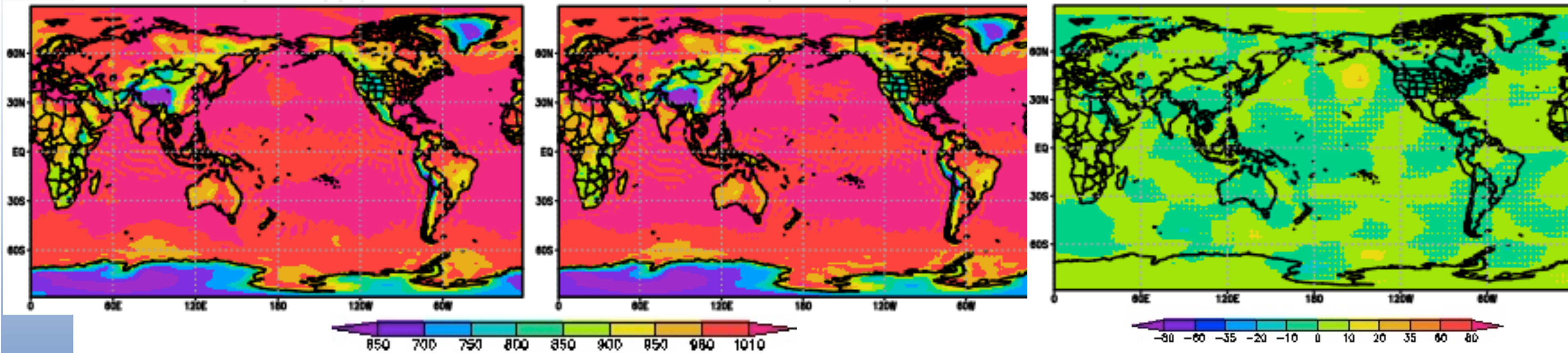
{with: r moisture and: M source/sink}

Data assimilation: LETKF x ANN (FSU model)

- LETKF with 40 members
- Model resolution T63L27: 63 spherical harmonic components for horizontal resolution (~ 1.875), and 27 unevenly spaced vertical levels.
- Number of grid points: 96 x 192 x 27
- MLP-NNs: **96** (4 horiz x 6 vert x 4 variables)
- Cray XE6 CPTEC: 24 nodes - 2 Opteron 12-cores

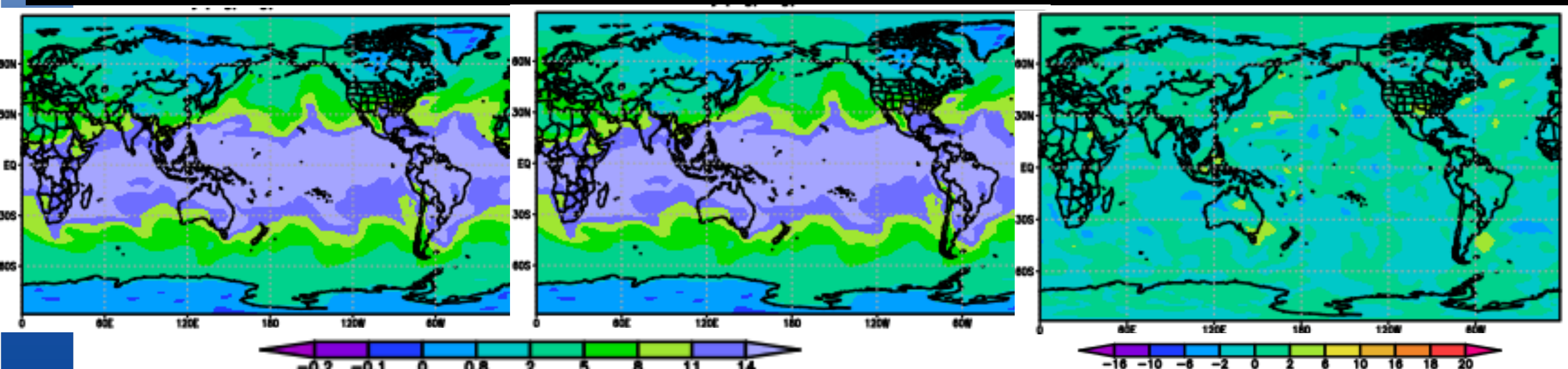
Data assimilation: LETKF x ANN (FSU model)

Surface Pressure(Kg/Kg) generalization 04/Jan/2005 – 12 UTC



LETKF analysis NN_MLP analysis Differences analysis

Specific Humidity (Kg/Kg) generalization 04/Jan/2005 – 12 UTC



LETKF analysis NN_MLP analysis Differences analysis

Data assimilation: LETKF x ANN (FSU model)

Execution of 124 cycles	MLP-DA (hour:min:sec)	LETKF (hour:min:sec)
Analysis time	00:02:29	11:01:20
Ensemble time	00:00:00	15:50:40
Parallel model time	00:27:20	00:00:00
Total Time	00:29:49	26:52:00

266 times faster

55 times faster

The LETKF analysis runs on 40 nodes at Cray XT/16 (1280 nodes, each node with 2 Opteron 12 cores, total of 30720 cores) (http://www.cptec.inpe.br/supercomputador)).

MLP-DA computed analyses for the FSUGSM model:

- **Analyses with similar LETKF quality**
- **Analysis with better computer performance.**

Predictability

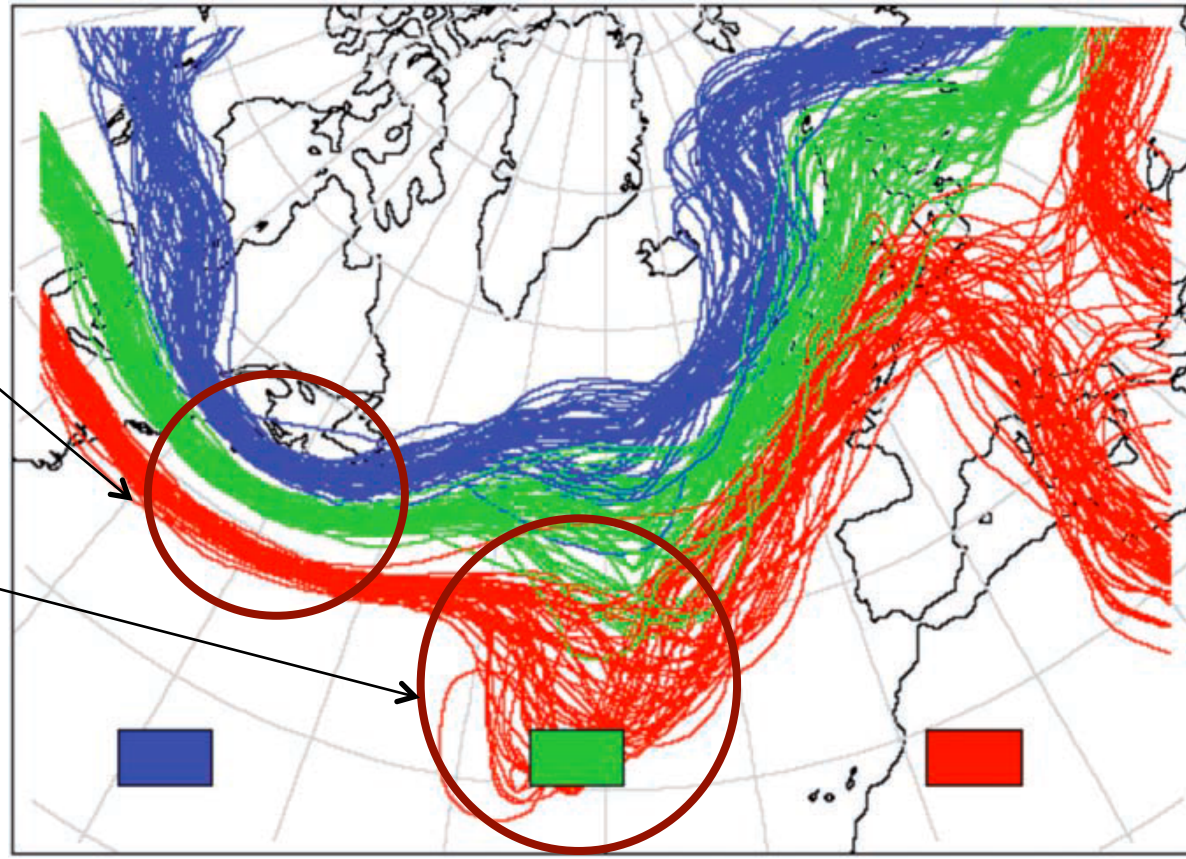
How good is the prediction?

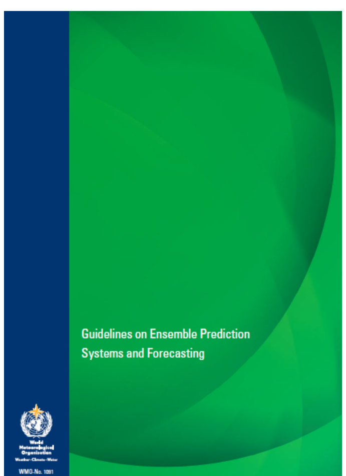
- Ensemble prediction
  - Data for statistical properties
  - Statistical tendencies
- Confidence interval
  - Large confidence interval: low predictability
  - Short confidence interval: high predictability

Predictability by ensemble prediction

High predictability
(ensemble convergence)

Low predictability
(ensemble dispersion)

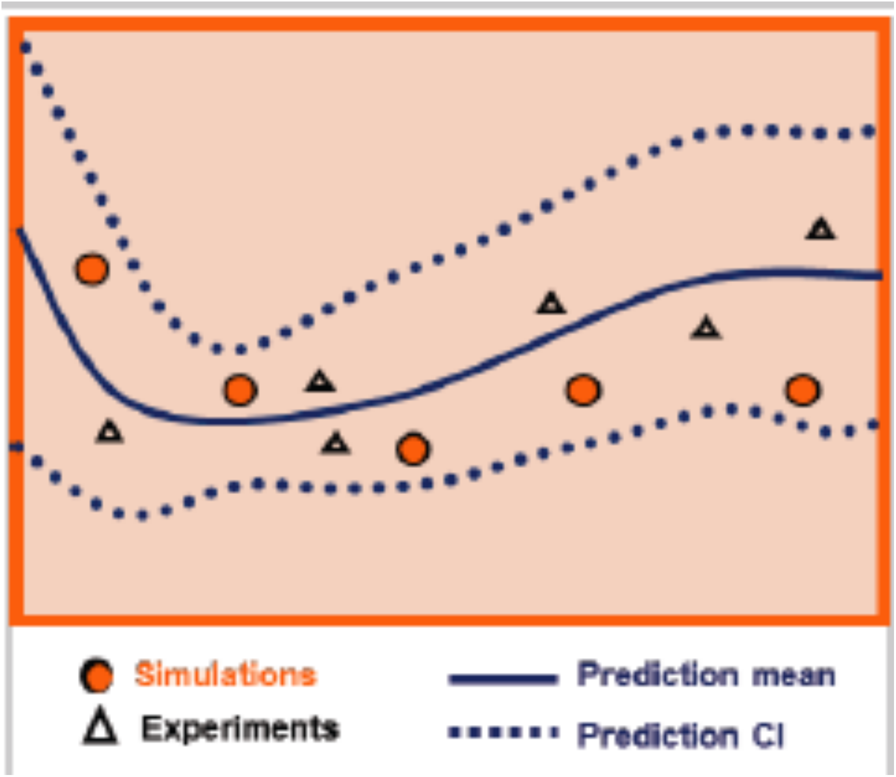




WMO's report describing/suggesting ensemble prediction

Predictability by ensemble prediction

- Ensemble prediction and confidence interval



Predictability by ensemble prediction

- Ensemble prediction and confidence interval

Proceedings of the joint ICVRAM ISUMA UNCERTAINTIES conference
Florianópolis, SC, Brazil, April 8-11, 2018

2018
ICVRAM-ISUMA
UNCERTAINTIES

Data assimilation by neural networks with ensemble prediction

Cintra, Rosangela S.[1,2]; Cocke, Steven[2] and Campos Velho, Haroldo F.[1]

[1] National Institute for Space Research (INPE), São José dos Campos (SP), Brazil.
[2] Florida State University, Tallahassee (FL), USA.

Predictability by ensemble prediction

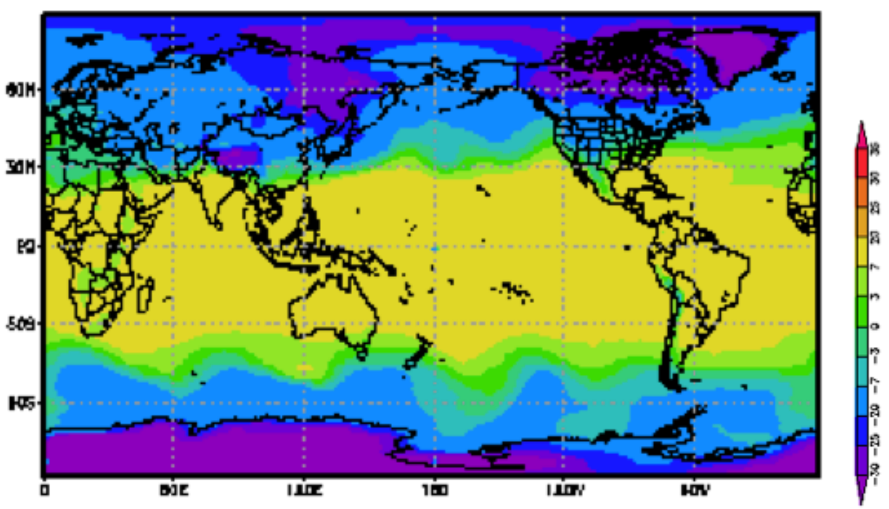
- Model execution by ensemble with 40 members
- Model resolution T63L27: 63 spherical harmonic components for horizontal resolution (~ 1.875), and 27 unevenly spaced vertical levels.
- Number of grid points: 96 x 192 x 27
- Data assimilation with **96 MLP-NNs**
- Data assimilation cycle: **each 6 hours**
- Cray XE6 CPTEC: 24 nodes - 2 Opteron 12-cores

Ensemble prediction

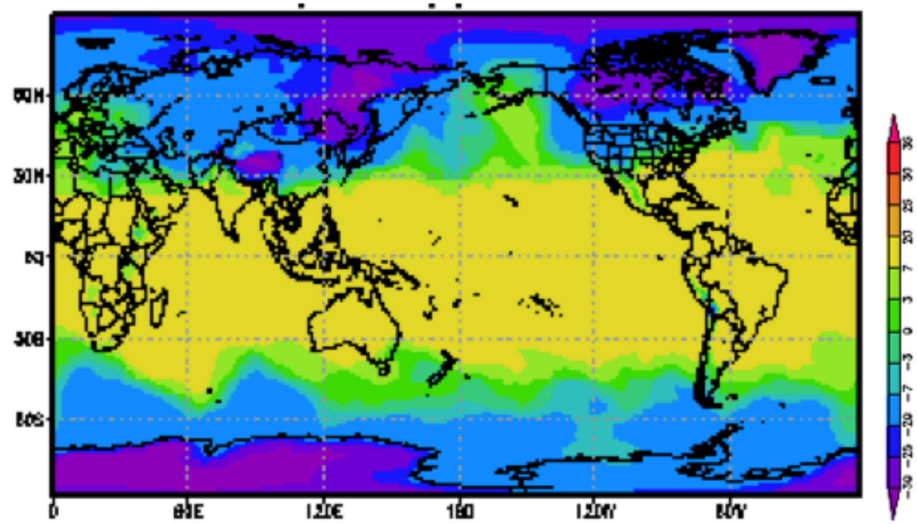
- **FSU global model: January 2005**

Temperature 500 hPa at 08/Jan/2005

LETFK



Control

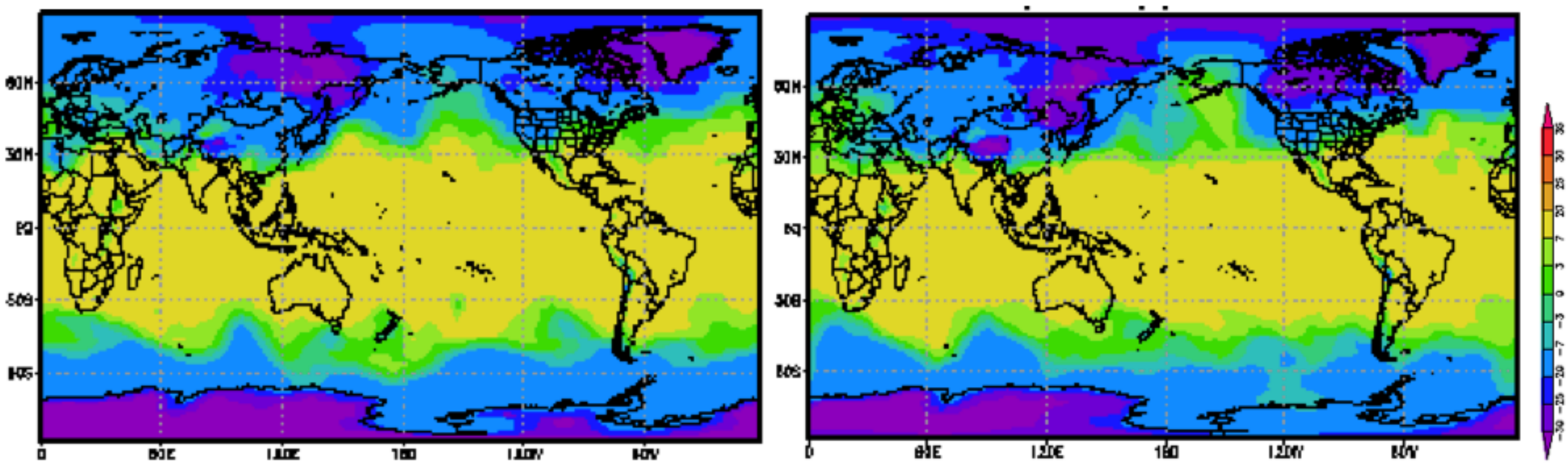


Ensemble prediction

- **FSU global model: January 2005**

Temperature 500 hPa at 08/Jan/2005

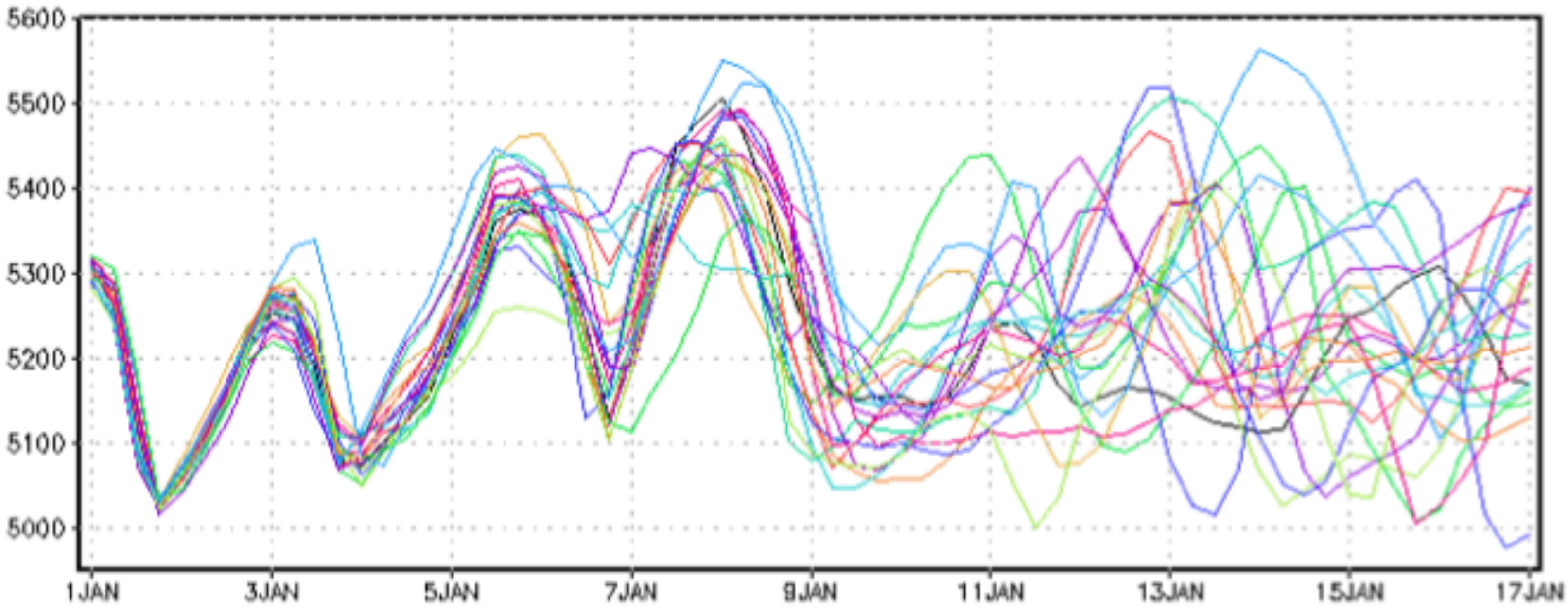
NN-MLP Control



Ensemble prediction

- **FSU global model: January 2005**

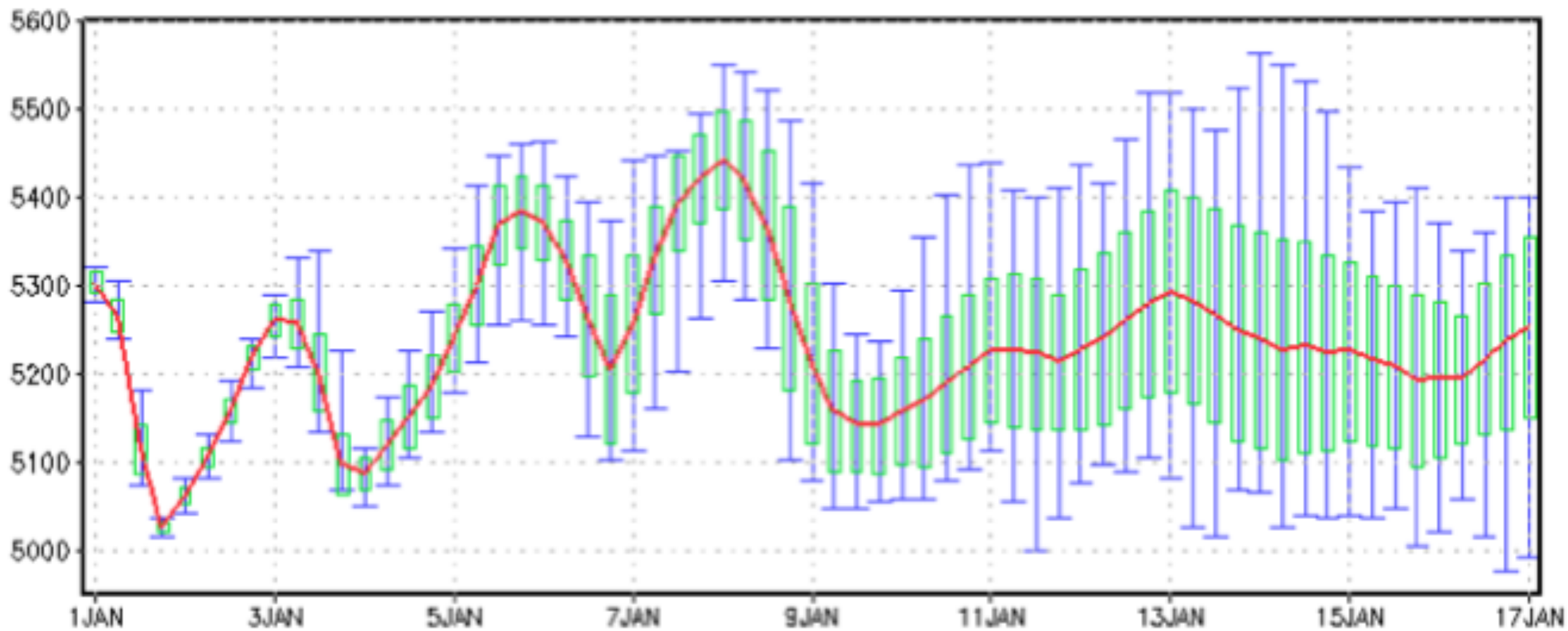
  Spaguetti plots



Ensemble prediction

- **FSU global model: January 2005**

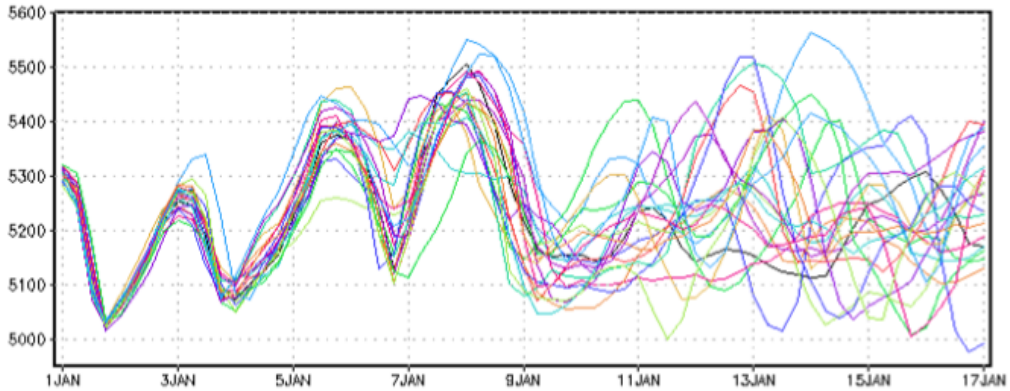
  Confidence intervals

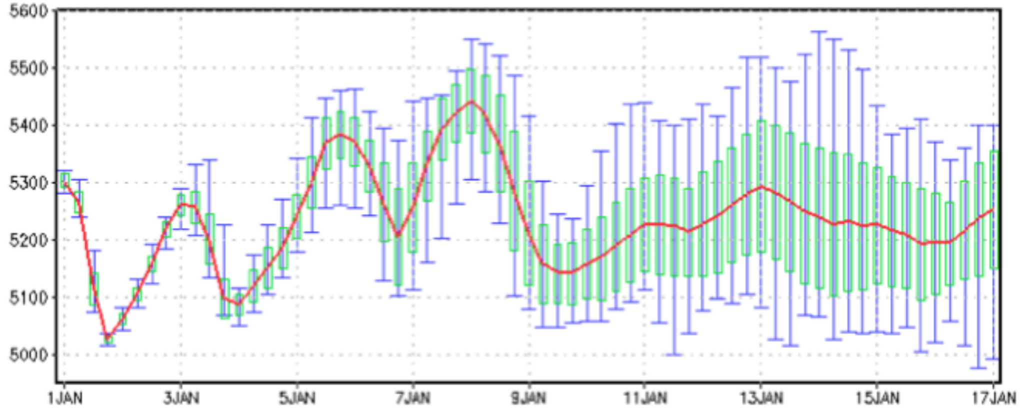


Ensemble prediction

- **FSU global model: January 2005**

  Spaguetti plots and confidence intervals

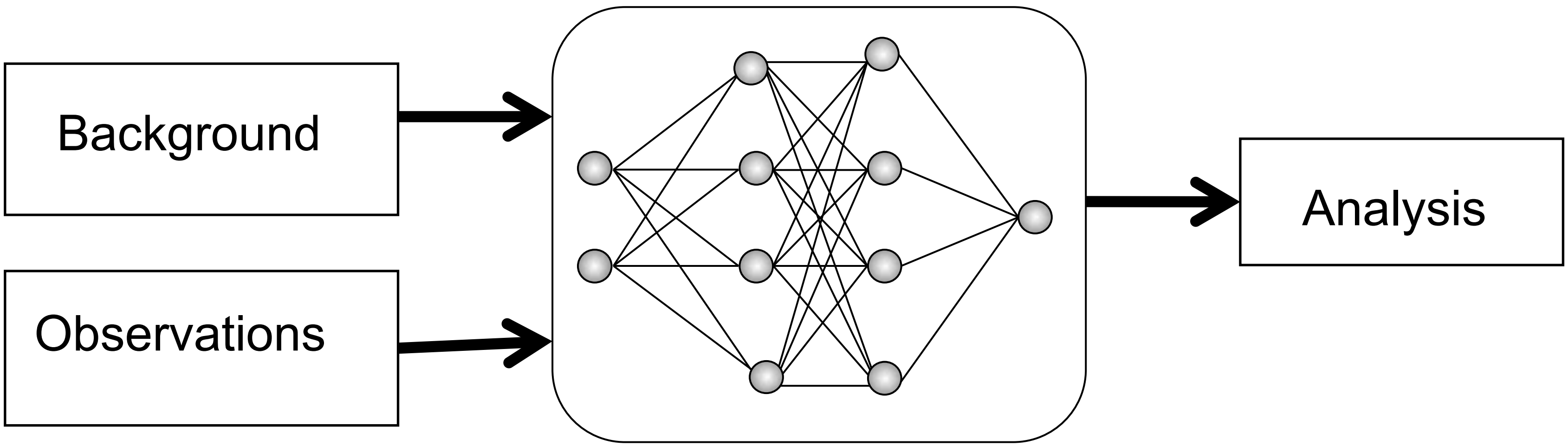




Uncertainty quantification by NN

Data assimilation: analysis by NN

Step-1: Data assimilation by NN



DA: Neural operator 1

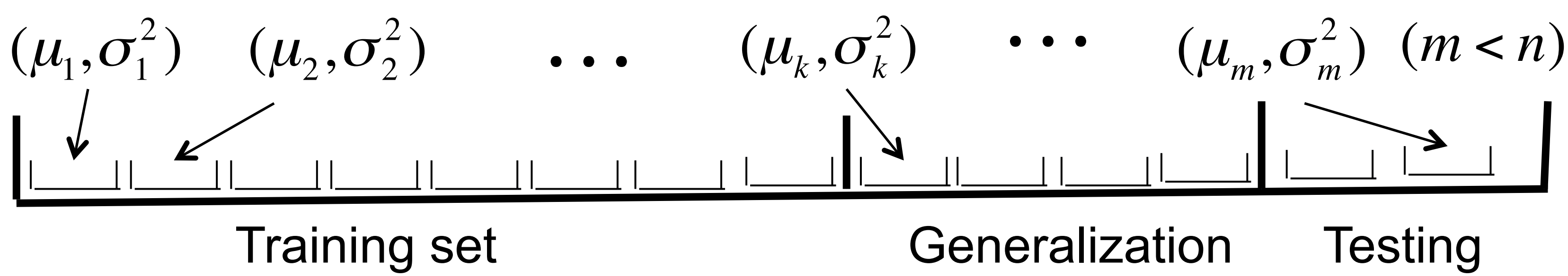
Uncertainty quantification by NN

Prediction: uncertainty quantification

Step-2: redesign the NN

A partition on the data-set used to define the neural fuser.

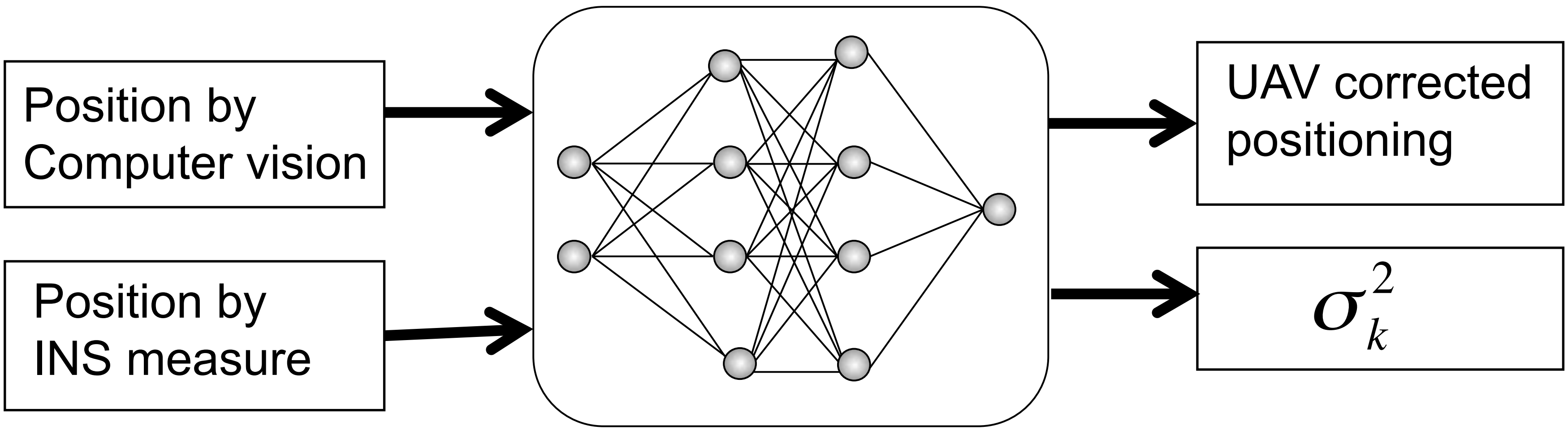
From the partition, with time series $\left\{\mu_k, \sigma_k^2\right\}_{k=1}^{m}$ **new NN.**



Drone positioning algorithm

Positioning by NN: uncertainty quantification

Step-2: New neural fuser self-configured by MPCA

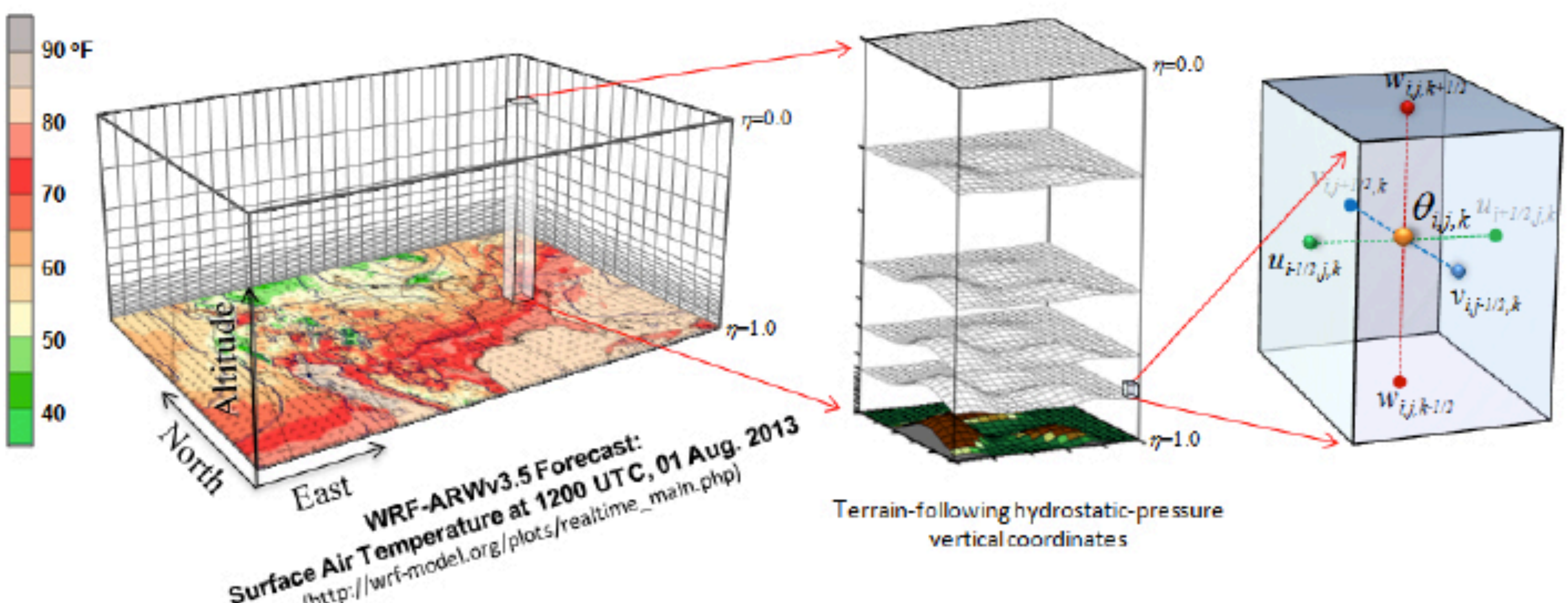


Analysis + Uncertainty
Neural operator 2

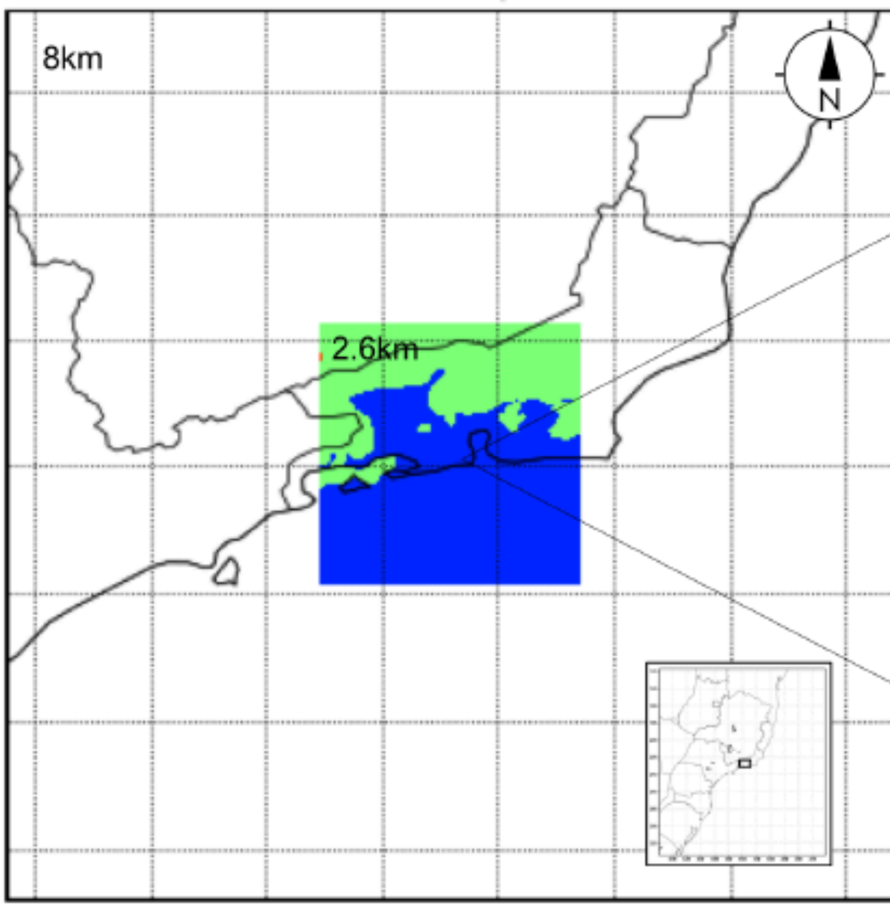
WRF: data assimilation by NN

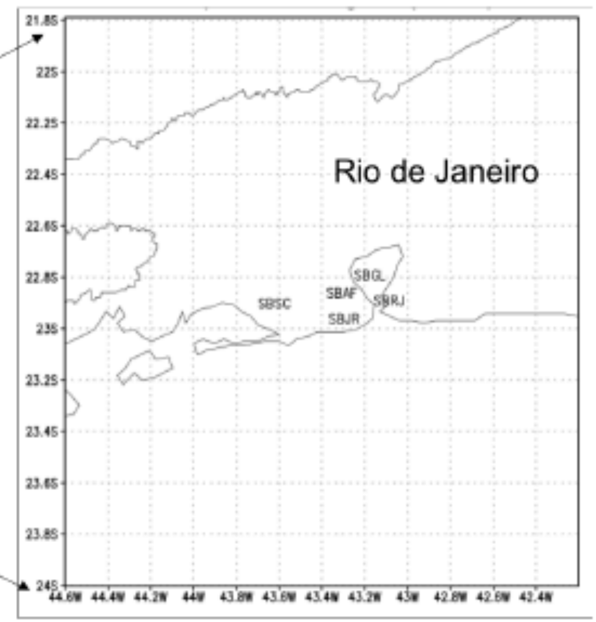
- Cooperation:
  - CODPT-INPE (BR)
  - Universities (BR): UFPel + IFI-Bagé + UFOPA + UFRJ
  - LNCC (BR)

WRF 3D Grid Cell Representation



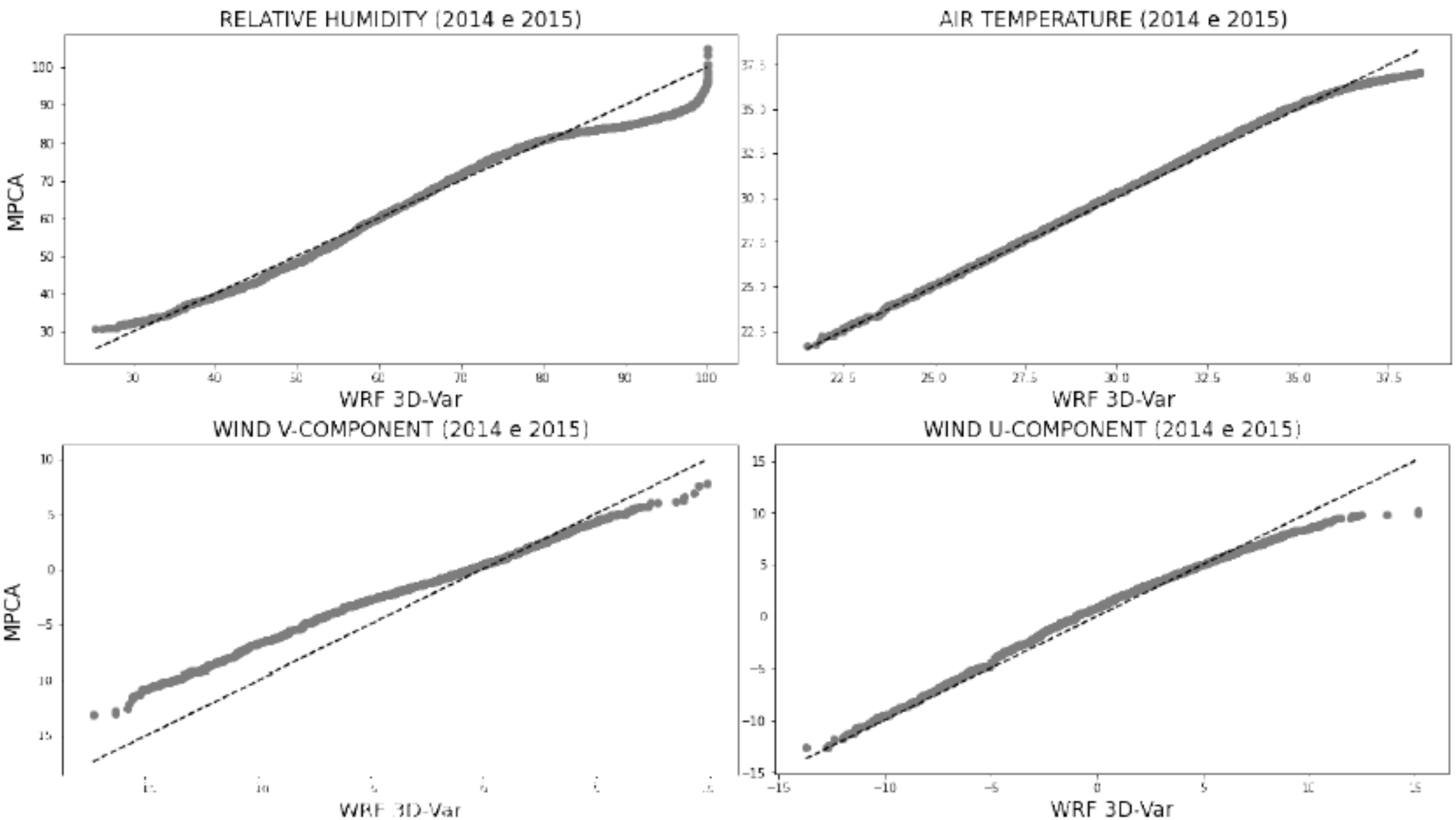
WRF-NCAR model





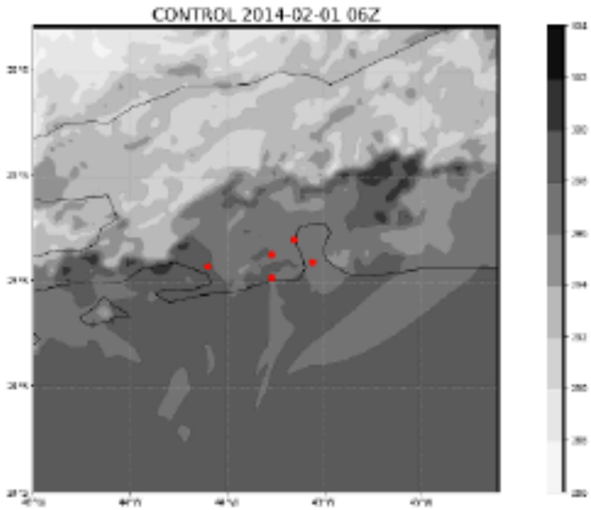
SBSC, SBAF, SBGL, SBRJ, SBJR are airports within the study area

WRF-NCAR model

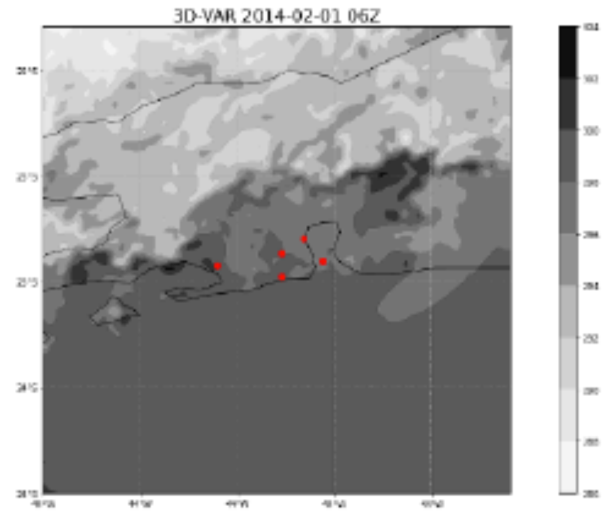


WRF-NCAR model

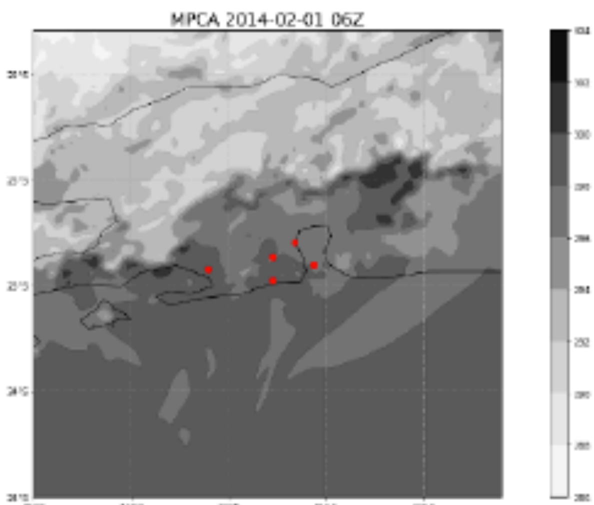
Control



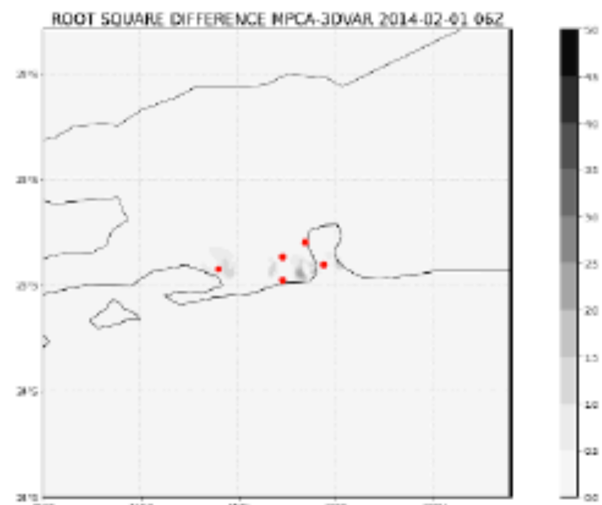
3dvar



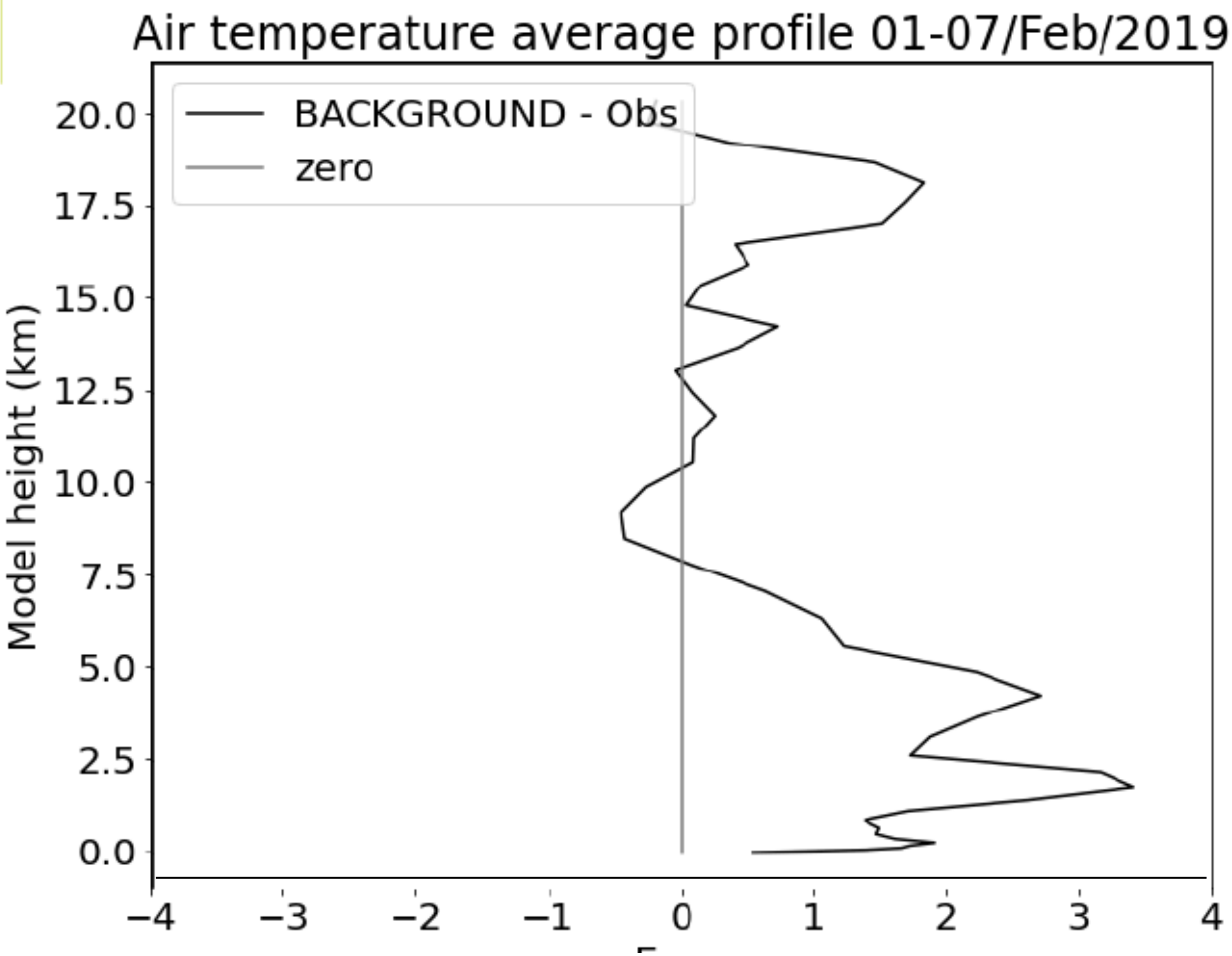
MPCA



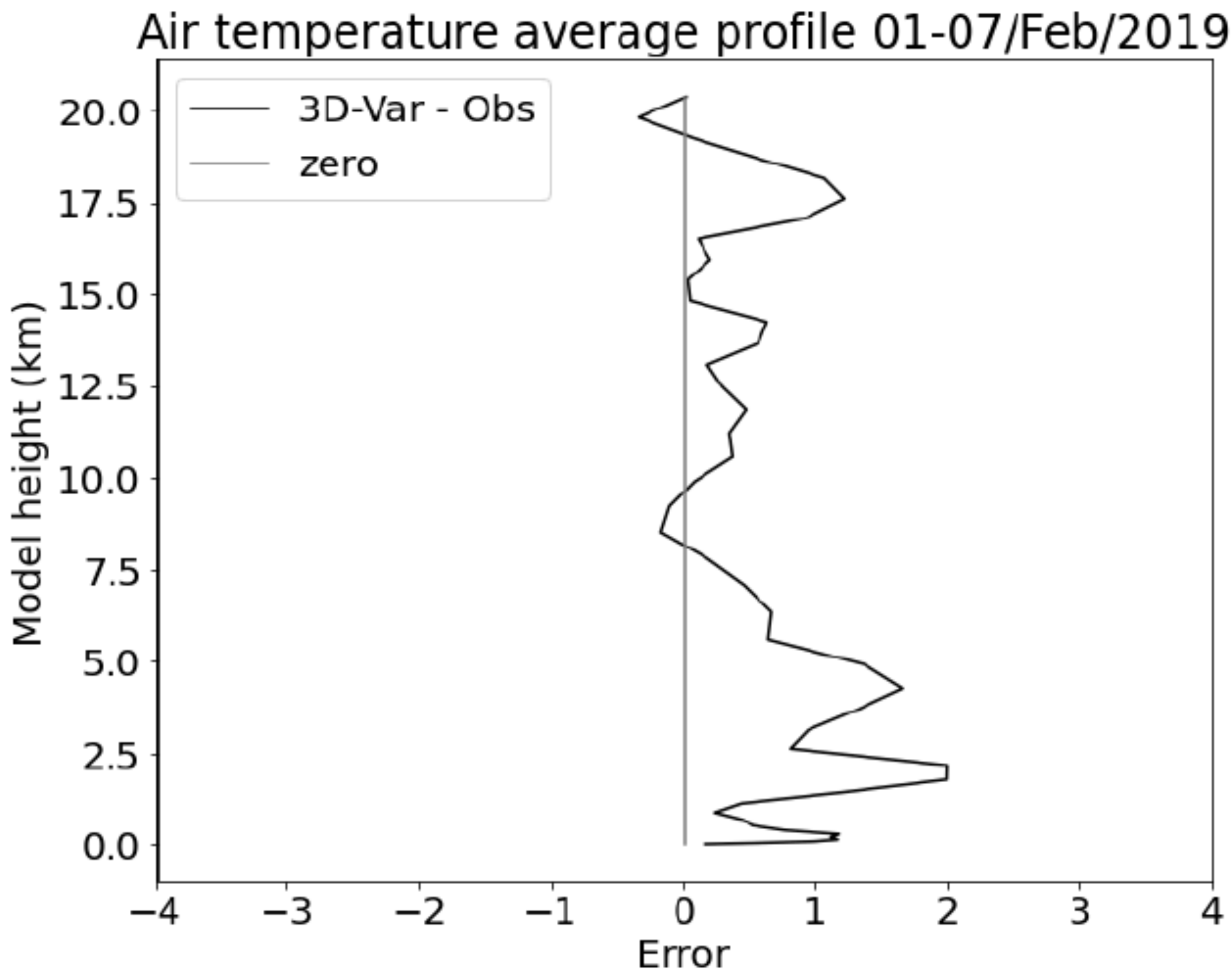
MPCA-3dvar



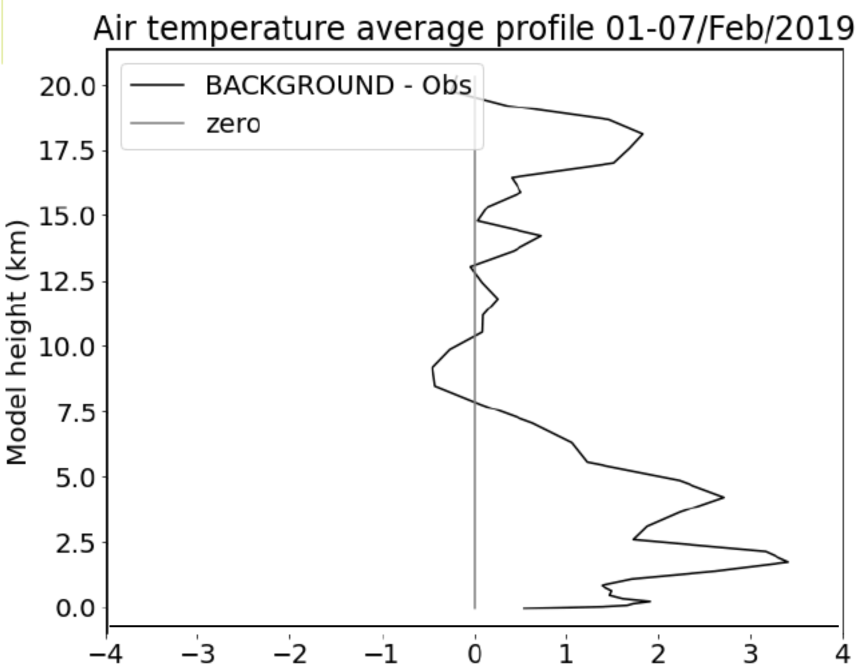
WRF-NCAR model

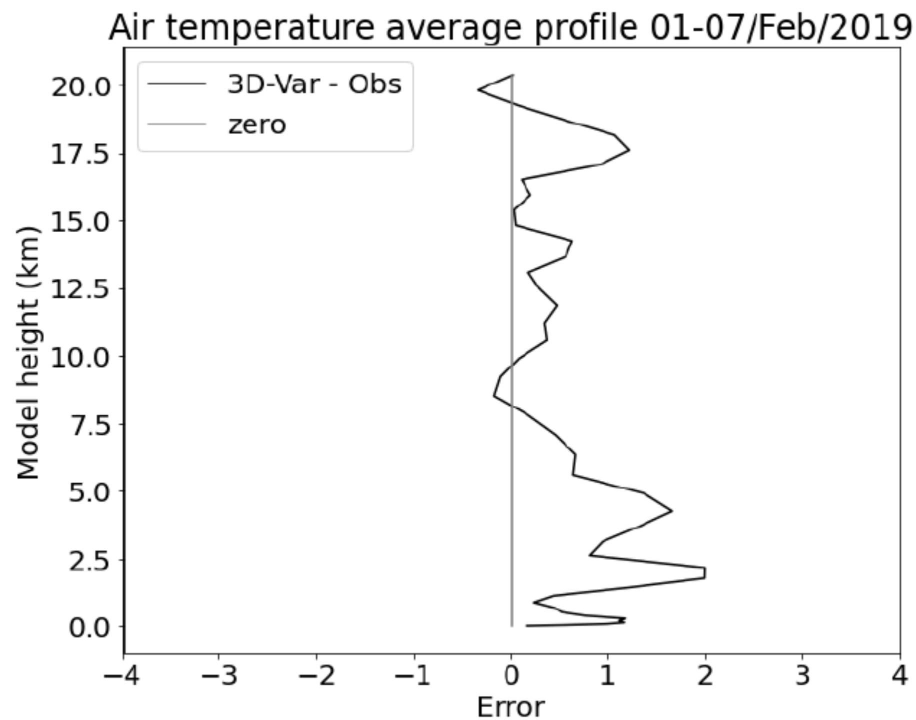


WRF-NCAR model

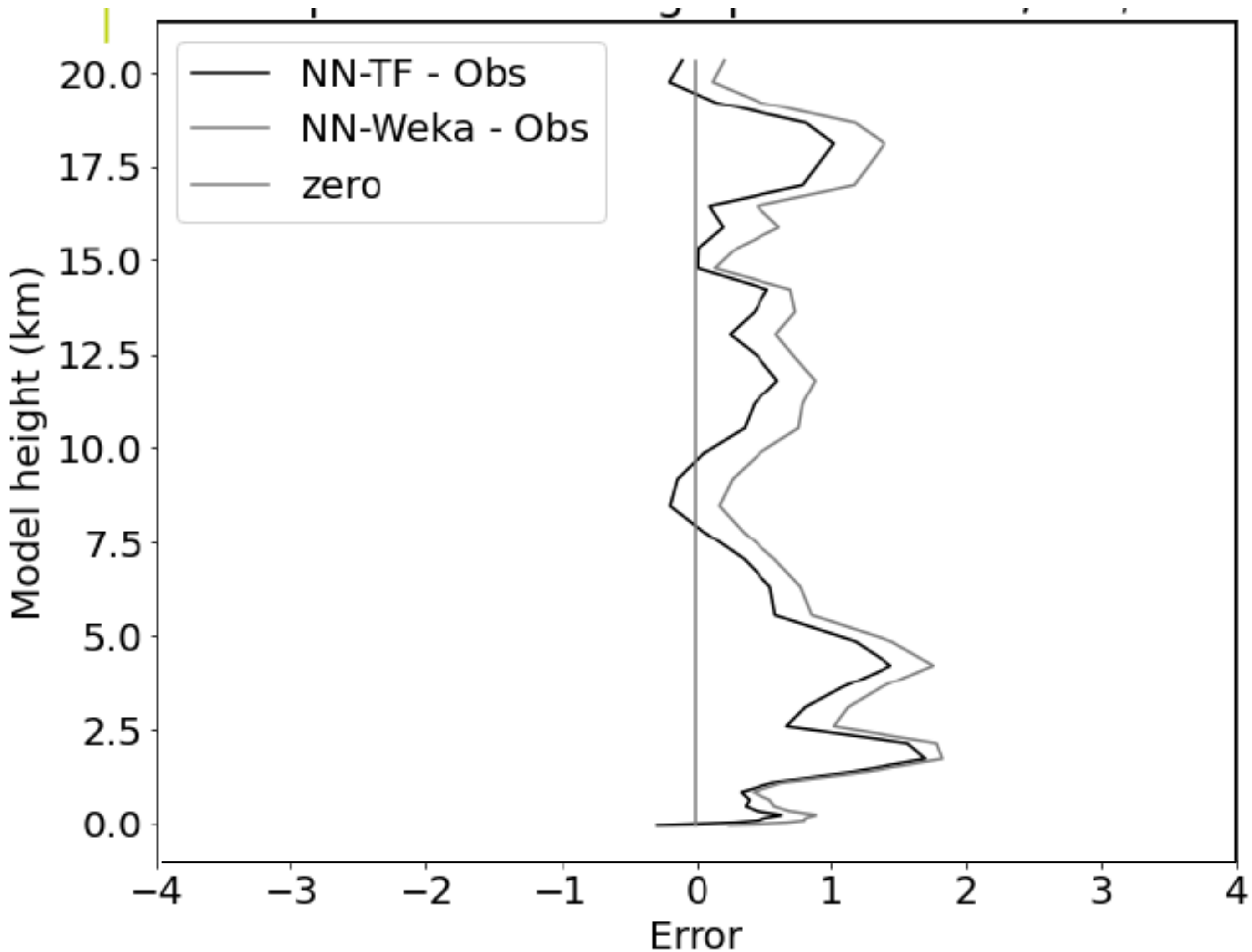


WRF-NCAR model

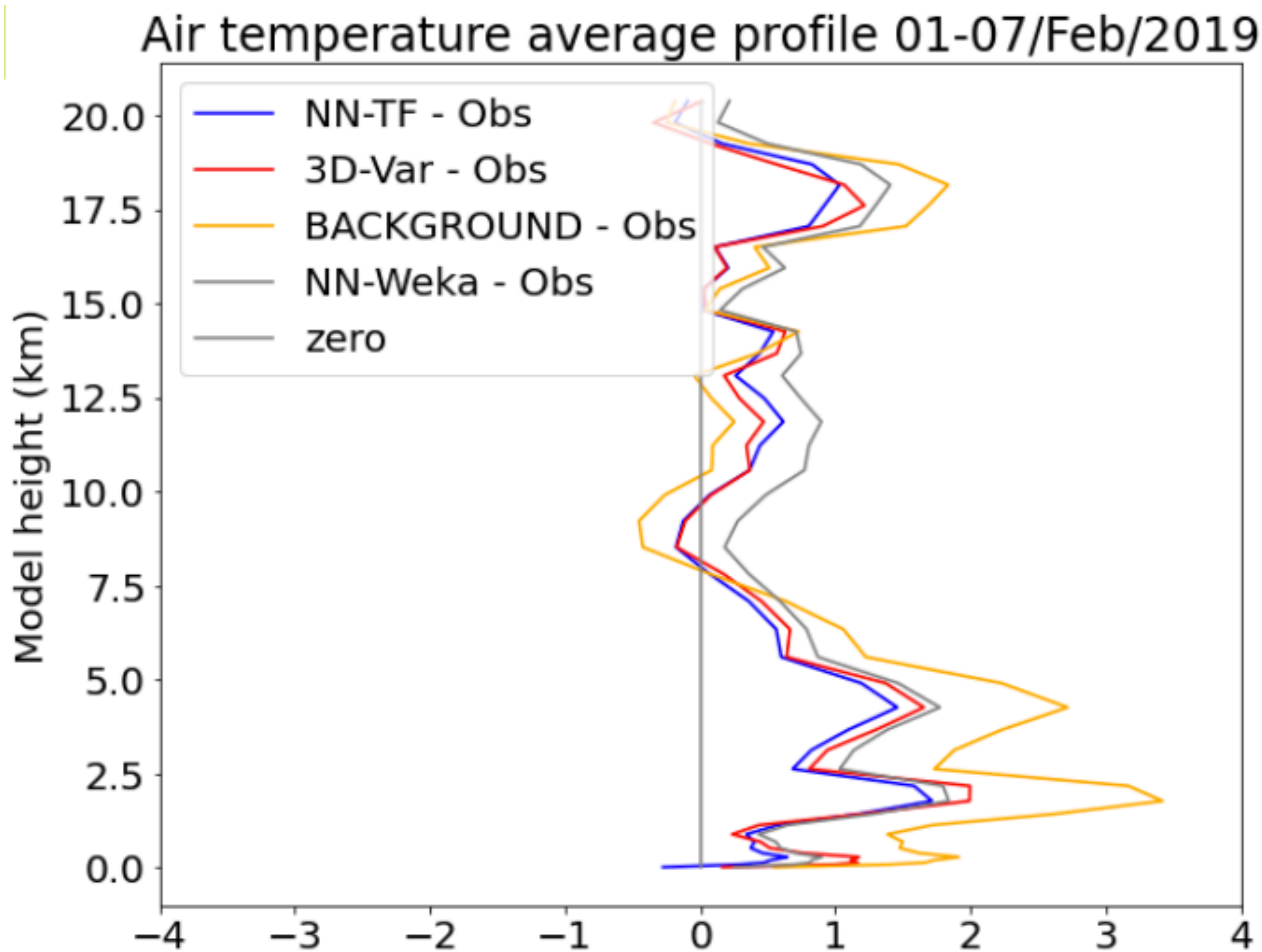




WRF-NCAR model



WRF-NCAR model



WRF-NCAR model

- **CPU-time**

	Time/cycle	Total
3D-Var	00:01:11	00:33:08
NN-MPCA	00:00:01	00:00:28

71 times faster

Data assimilation: NN vs. “standard” methods

- **CPU-time**

MODEL	SPEED (G)	COAPS-FSU (G)	WRF (R)
Method	**LEnTKF | NN**	**LEnTKF | NN**	**3D-Var | NN**
CPU-time	04:20:39 | 00:02:53	26:52:00 | 00:29:49	00:33:08 | 00:00:28

95 times faster

55 times faster

71 times faster

Data assimilation – NN emulating KF

- NN emulating Kalman filter: Linear wave 1D

Springer Link

Springer Nature is making Coronavirus research free. View research | View latest news | Sign up for updates

Pure and Applied Geophysics
pp 1–21 | Cite as

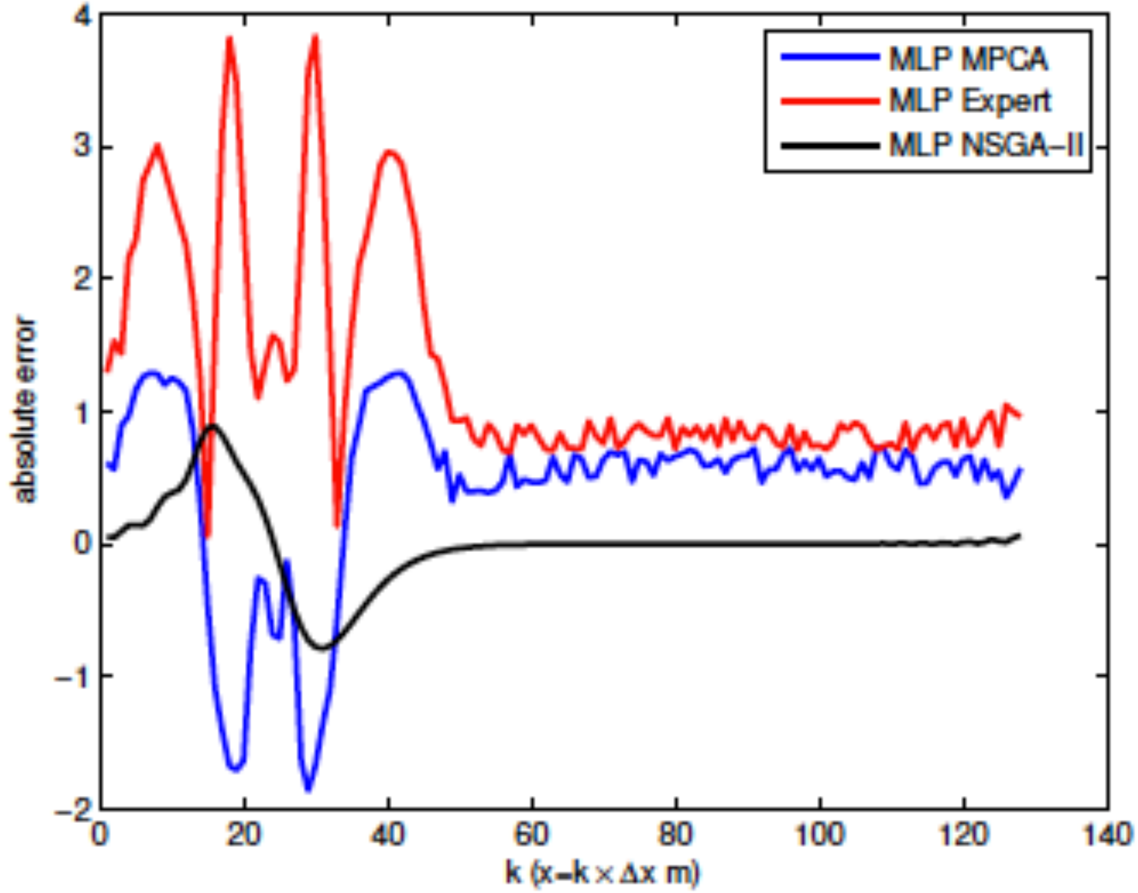
Two Geoscience Applications by Optimal Neural Network Architecture

Authors   Authors and affiliations

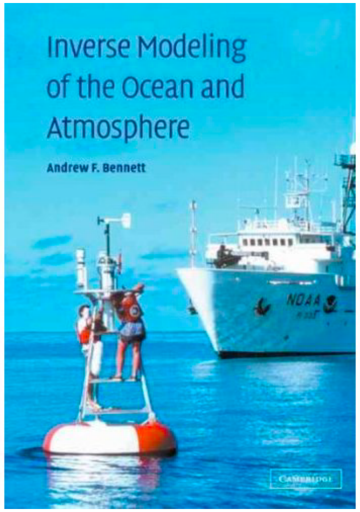
Juliana Aparecida Anochi, Reynier Hernández Torres, Haroldo Fraga de Campos Velho

Data assimilation – NN emulating KF

- NN emulating Kalman filter: Linear wave 1D



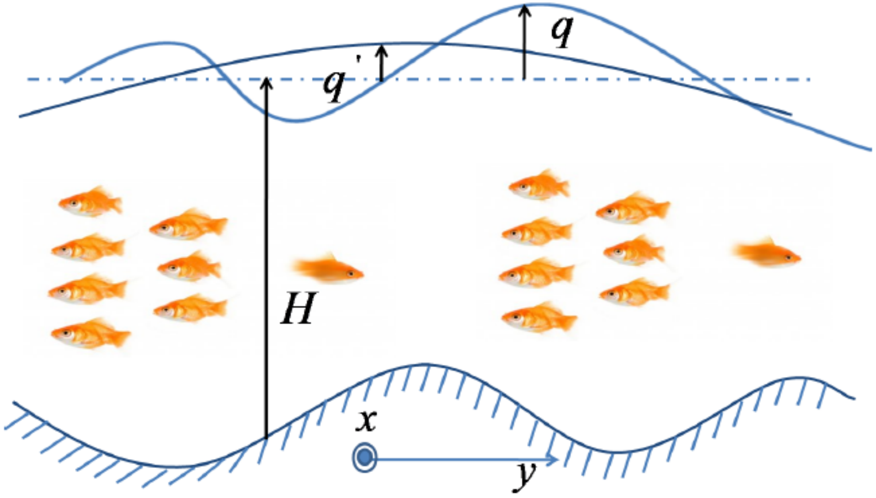
Shallow water 2D for ocean circulation



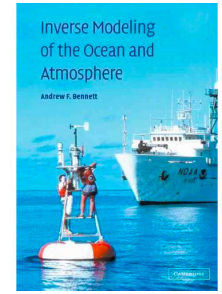
$$\frac{\partial u}{\partial t} - fv + g\frac{\partial q}{\partial x} + r_u u = F_u$$

$$\frac{\partial v}{\partial t} + fu + g\frac{\partial q}{\partial y} + r_v v = F_v$$

$$\frac{\partial q}{\partial t} + H\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + r_q q = 0$$

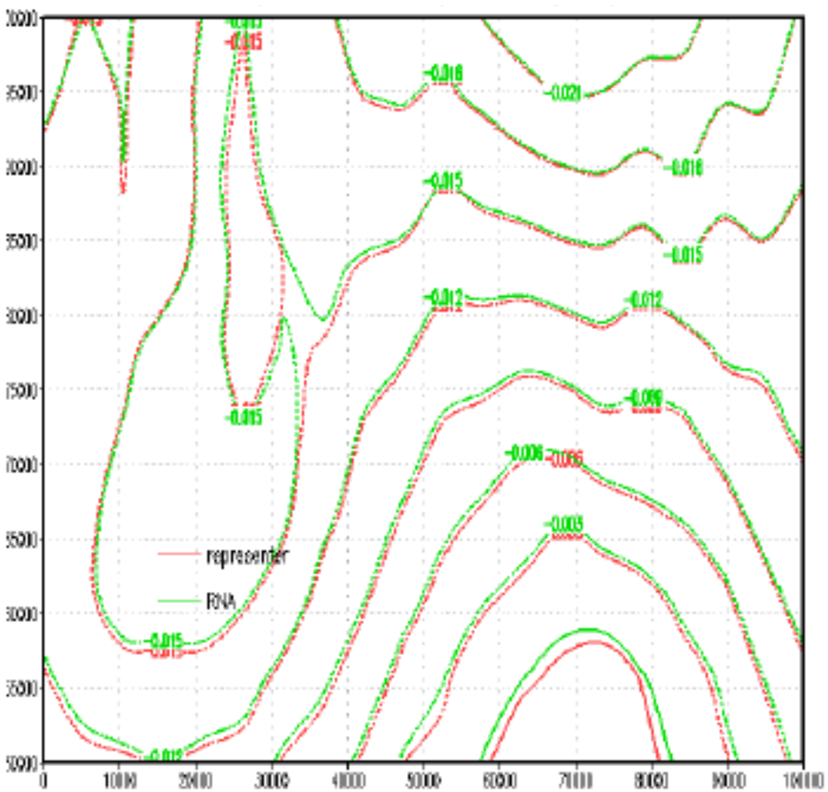


Shallow water 2D: representer (variational) vs neural network

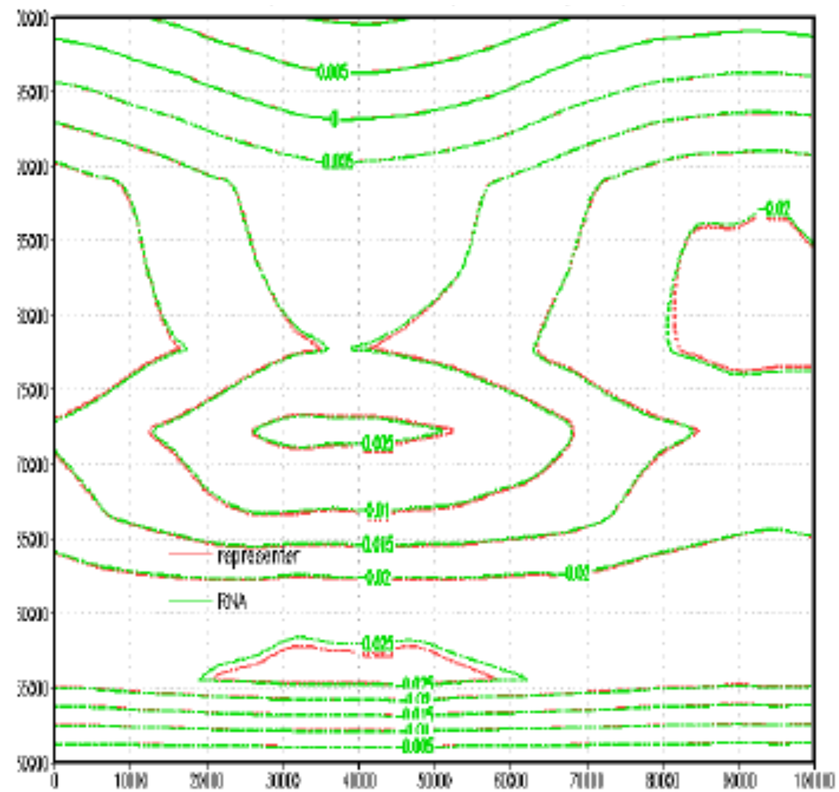


Green = representer / Red = neural network

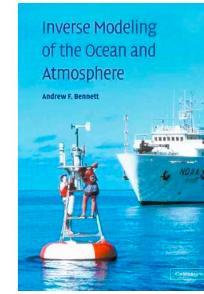
Zonal wind (u)



Meridionalwind (v)

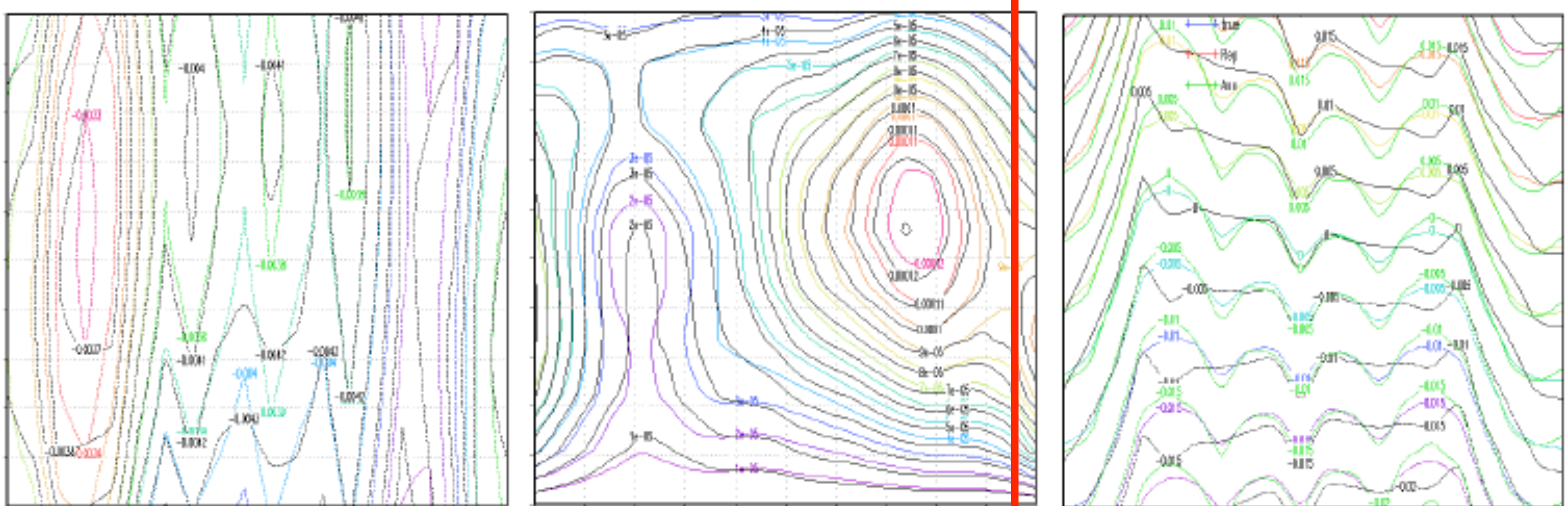


Shallow water 2D: representer (variational) vs neural network



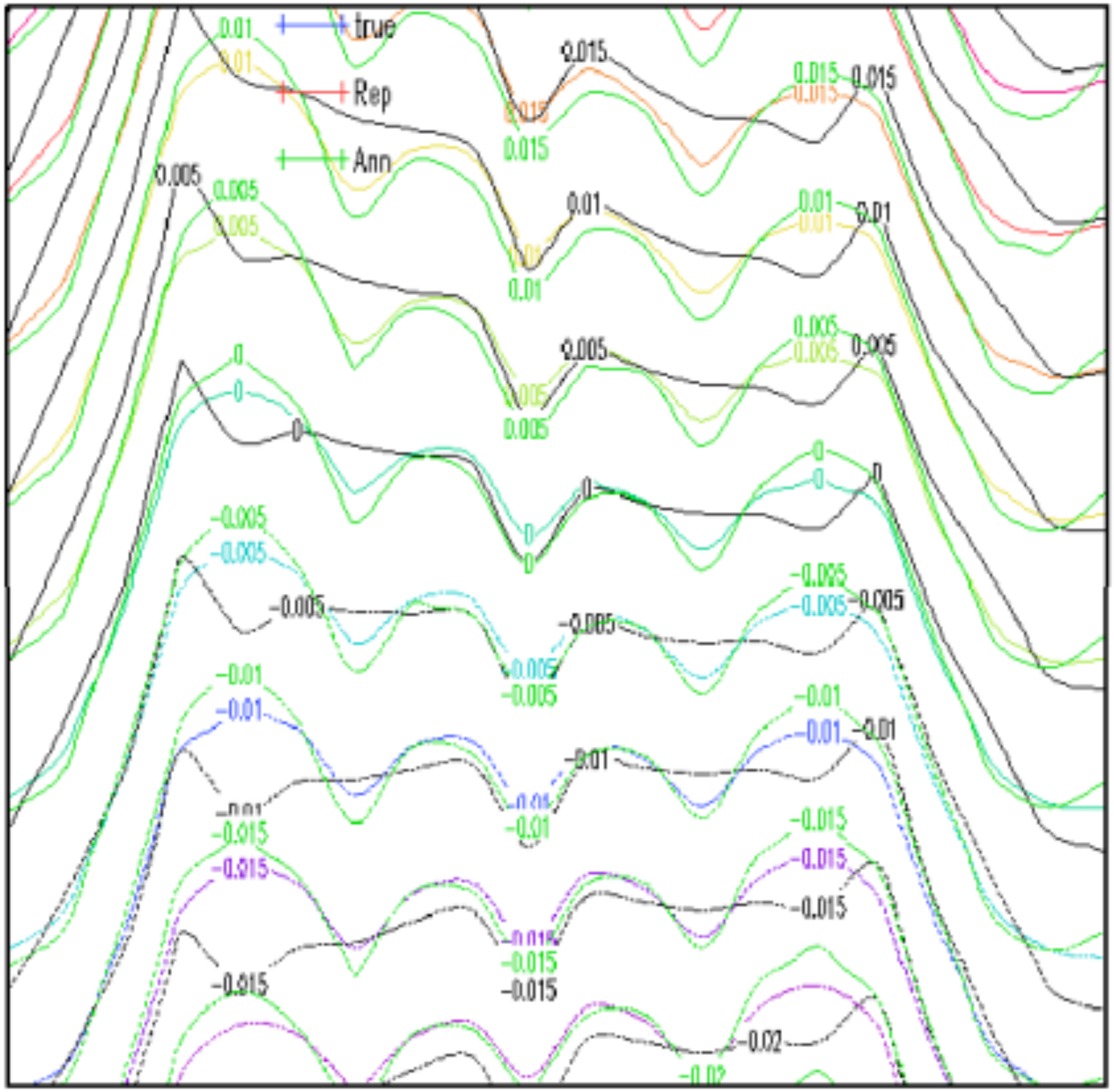
Data assimilation: Variational representer and neural network

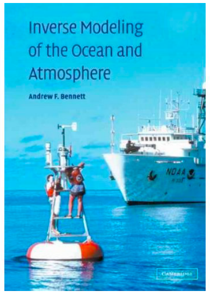
Zonal wind (u) Meridionalwind (v) height (q)



Shallow water 2D: representer (variational) vs neural network

Representer vs neural network: zoom for $q(x,y)$

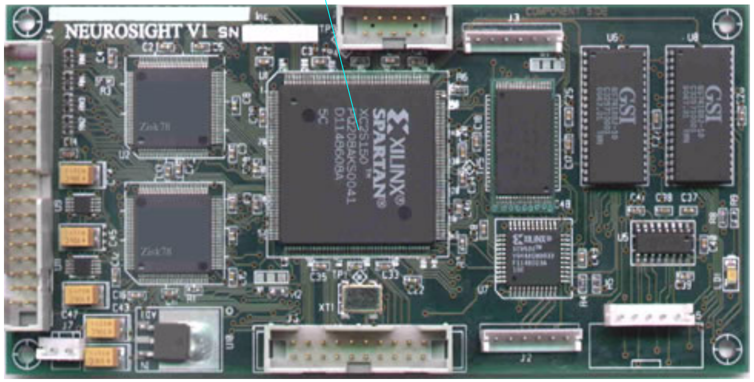




Data assimilation by NN: hardware components

The Cray XD1 -
Reconfigurable Computing

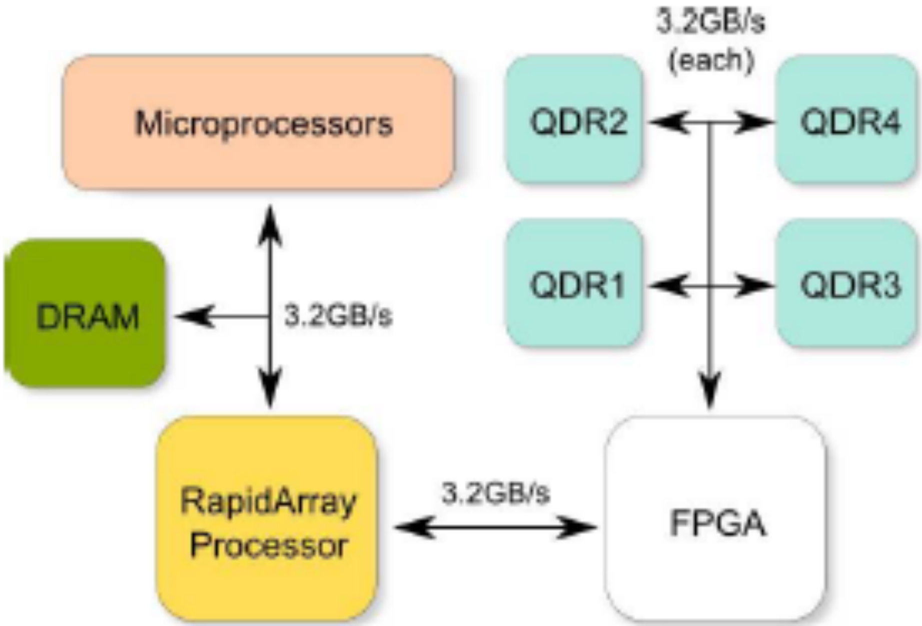
FPGA



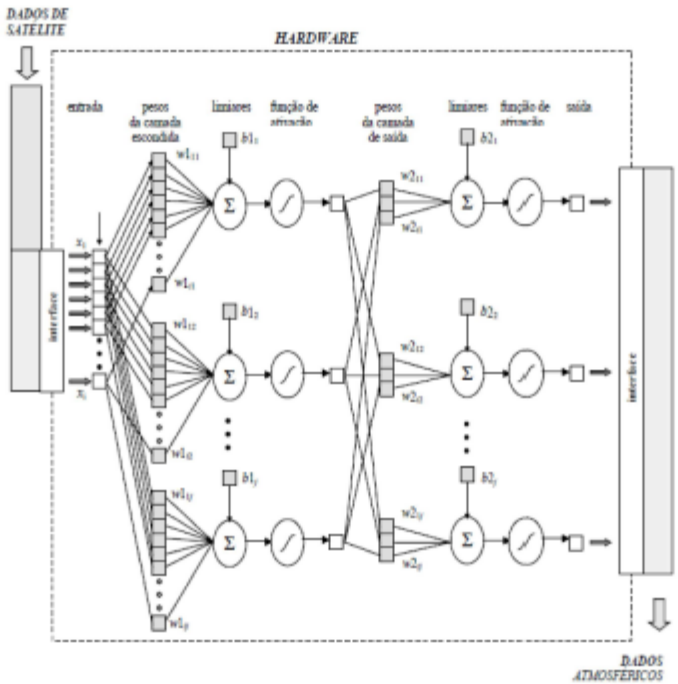
Hybrid computing with FPGA

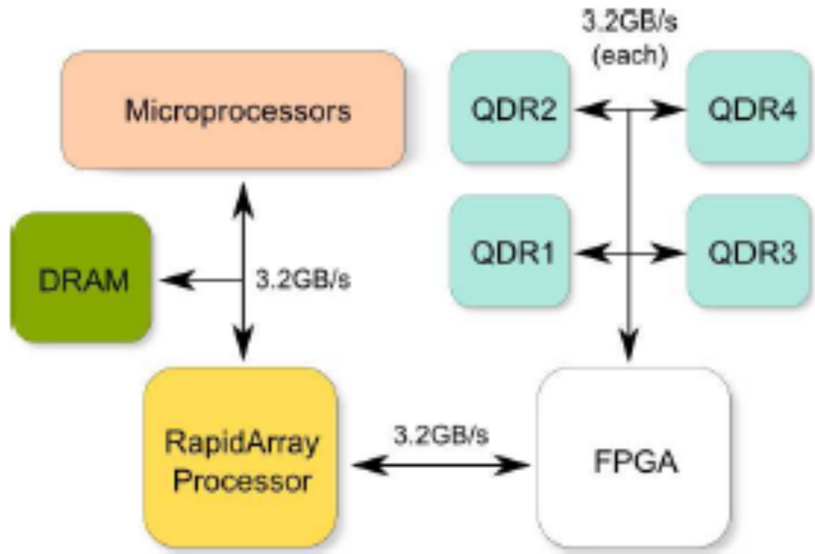
Blade
2 AMD Opteron 64bits 2.4GHz
1 FPGA Xilinx Virtex II Pro

Cray XD1

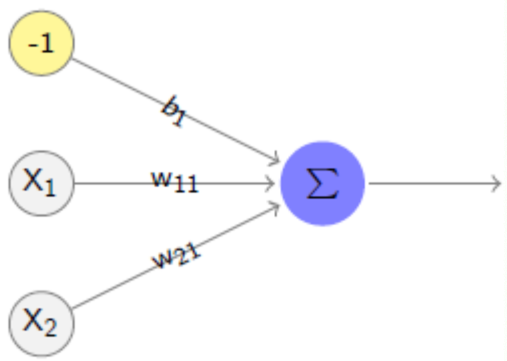


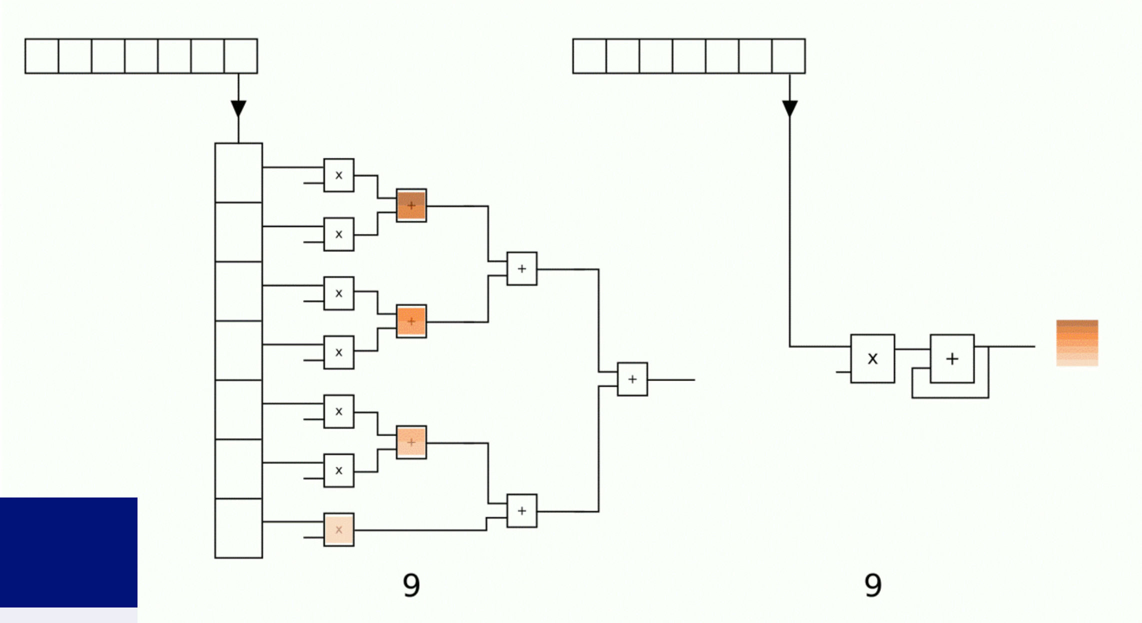
Perceptron-NN for the Cray XD1





Perceptron-NN for the Cray XD1





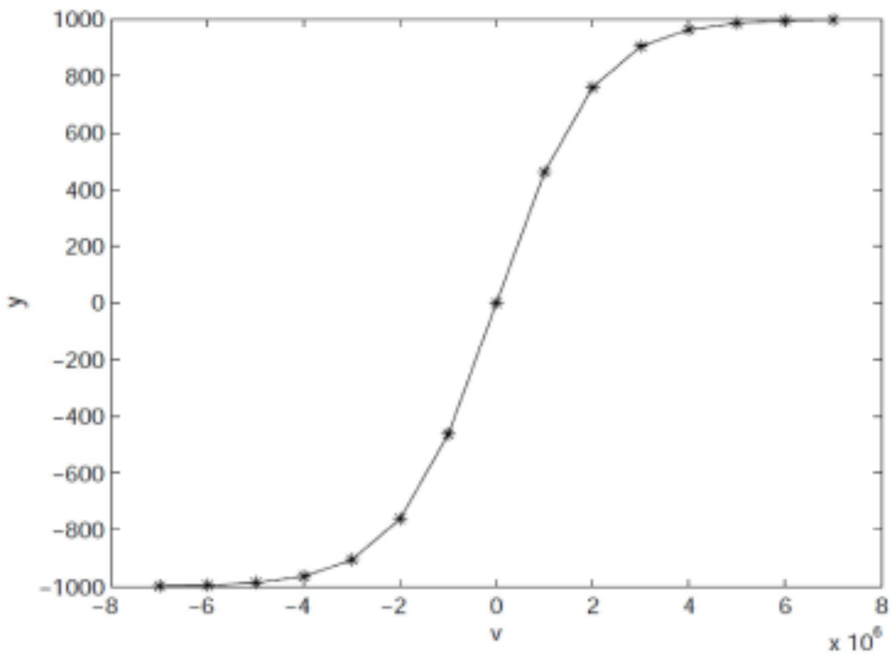
Cost:

- Multipliers: 7
- Summation: 1
- Cycles: 14 = 7 + 1 + 6

Perceptron-NN for the Cray XD1

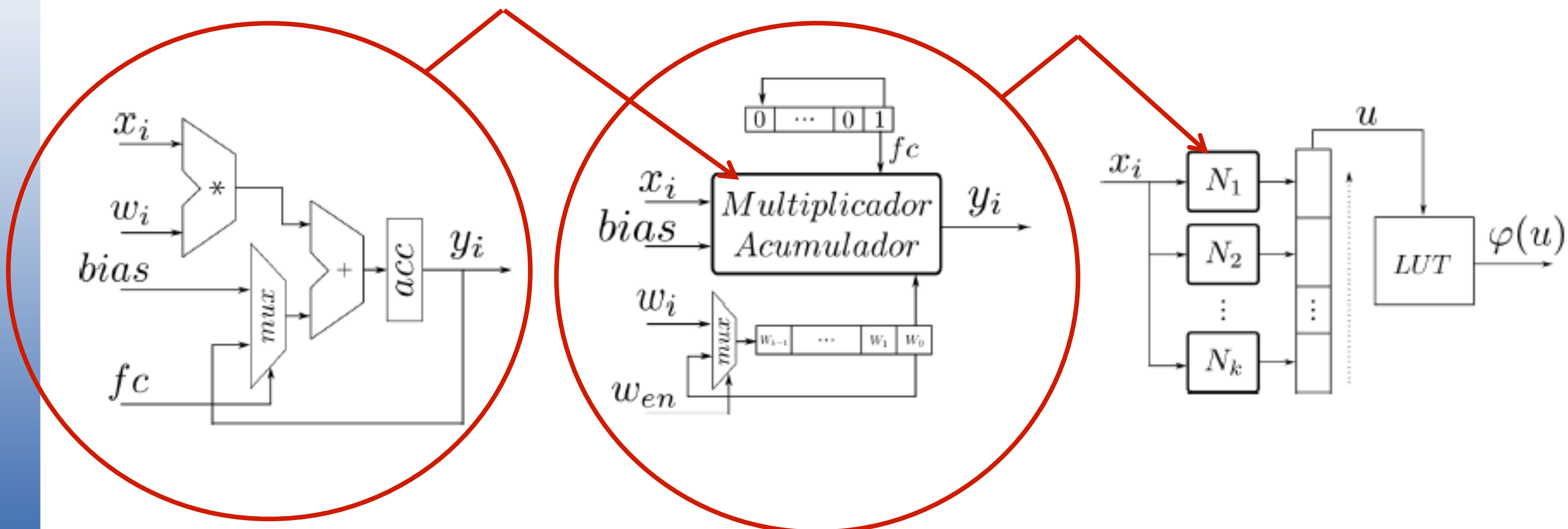
Activation function

- tanh(x)
- Lookup Table (LUT)
- QDR: 1 M



Sigmoid function: tanh(x)

Perceptron-NN for the Cray XD1

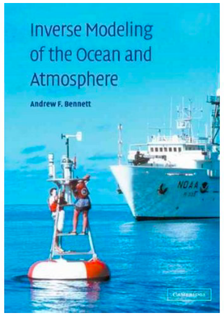


MAC:
Multiplier /accumulator
Input x weight
Storaged on ACC or bias
(depending on fc signal

Neuron
Uses of MAC

MLP-NN:
Combining neurons
Inputs connected by one bus
Ready to receive new data
Results to Lookup Table (LUT):
the pipeline

Shallow water 2D for ocean circulation



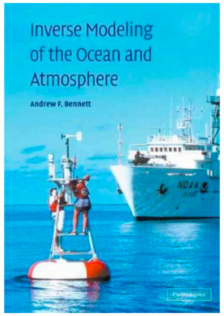
Process	Time (µs)
Software (CPU)	121709
Hardware (FPGA)	209187

$$\frac{\partial u}{\partial t} - fv + g\frac{\partial q}{\partial x} + r_u u = F_u$$

$$\frac{\partial v}{\partial t} + fu + g\frac{\partial q}{\partial y} + r_v v = F_v$$

$$\frac{\partial q}{\partial t} + H\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + r_q q = 0$$

Shallow water 2D for ocean circulation



Process	Time (µs)
Software (CPU)	121709
CPU to FPGA	181365
FPGA	2
FPGA to CPU	9455
FPGA (Total)	209187

$$\frac{\partial u}{\partial t} - fv + g\frac{\partial q}{\partial x} + r_u u = F_u$$

$$\frac{\partial v}{\partial t} + fu + g\frac{\partial q}{\partial y} + r_v v = F_v$$

$$\frac{\partial q}{\partial t} + H\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + r_q q = 0$$

Special thanks to Dr. Xiaodong Luo Luo

EnKF Workshop 2021 organizer

Thanks for our sponsors:

Brazilian agencies for resarch support:

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

CAPES

FAPESP

FINEP

Thank you!

Data assimilation – first application

Data Assimilation Using an Adaptative Kalman Filter and Laplace Transform

A.G. Nowosad[a] (DCM)   A. Rios Neto[b]   H.F. de Campos Velho[a] (LAC)

[a] Instituto Nacional de Pesquisas Espaciais (INPE)
Caixa Postal 515
12201-970 – São José dos Campos (SP), BRAZIL,
E-mail: alex@met.inpe.br - haroldo@lac.inpe.br

[b] Instituto de Pesquisa e Desenvolvimento (IP&D)
Universidade do Vale do Paraíba (UNIVAP)
Av. Shishima Hifumi, 2.911 - Urbanova
12245-720 - São José dos Campos (SP), BRAZIL
E-mail: atair@univap.br

Hybrid Methods in Engineering: (2000) 2(3): 291-310

Data assimilation – NN emulating KF

- NN emulating Kalman filter: **Lorenz's system**

Revista Brasileira de Meteorologia, **v.20**, **n.3**, 411-420, 2005

REDES NEURAIS RECORRENTES TREINADAS COM CORRELAÇÃO CRUZADA APLICADAS A ASSIMILAÇÃO DE DADOS EM DINÂMICA NÃO-LINEAR

FABRÍCIO PEREIRA HÄRTER e HAROLDO FRAGA DE CAMPOS VELHO

Data assimilation – NN emulating KF

- NN emulating Kalman filter: **Lorenz's system**

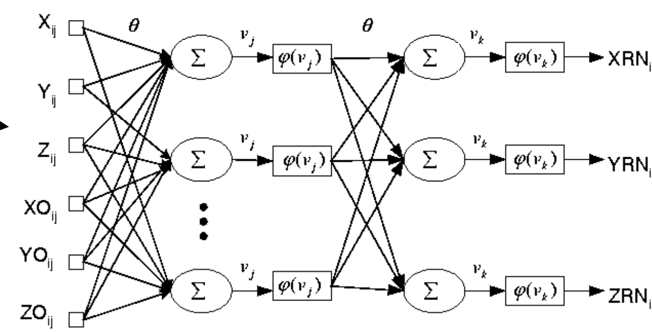
$$
\begin{aligned}
dX/dt &= -\sigma(X - Y) \\
dY/dt &= RX - Y - XZ \\
dZ/dt &= XY - bZ
\end{aligned}
$$

$w_0 \equiv [X_0 \; Y_0 \; Z_0]^T = [1.508870 \quad -1.5312 \quad 25.46091]^T$

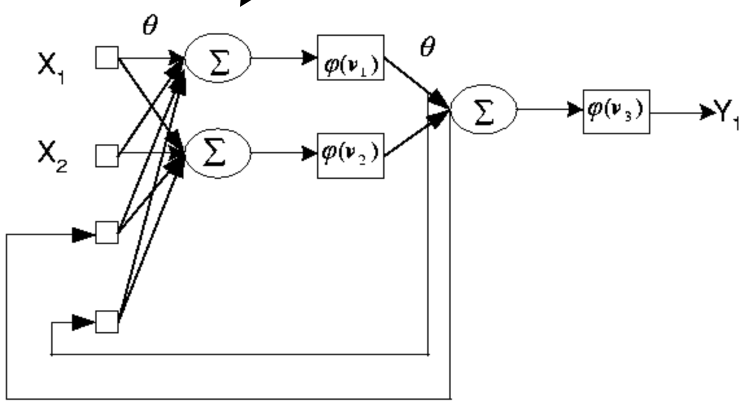
Data assimilation – NN emulating KF

- NN emulating Kalman filter: **Lorenz's system**

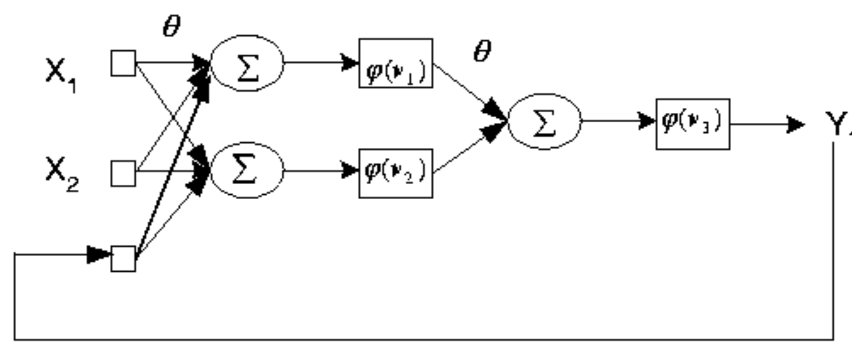
Standard NN



Recurrent NNs:



Elman-NN

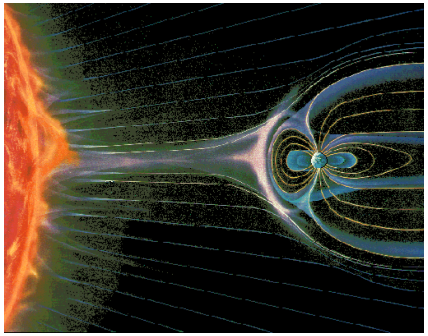


Jordan-NN

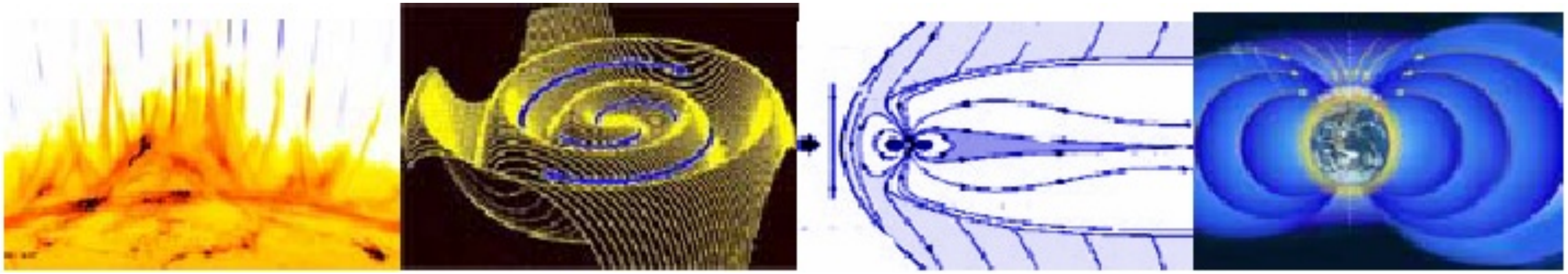
Data assimilation – NN emulating KF

- NN emulating Kalman filter: **Space Weather**

Interaction: Sun-Earth

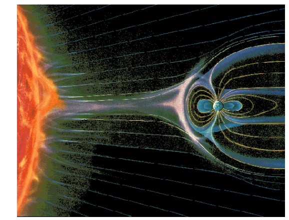


Solar Activity — Propagation — Impact on magnetosphere — ionosphere



Data assimilation – NN emulating KF

- NN emulating Kalman filter: **Space Weather**



Interaction: Sun-Earth

Equations: three-waves coupled

$$dA_L / d\tau = \nu_L A_L + A_W A_A$$

$$dA_W / d\tau = \nu_W A_W - A_L A_A^*$$

$$dA_A / d\tau = (i\delta + \nu_A) A_A - A_L A_W^*$$

$$\nu_L = 1$$

$$\nu_L = \nu_L = -\nu$$

$$\delta = 2$$

$$\tau \equiv \kappa(z - \nu t)$$

Data assimilation – NN emulating KF

- NN emulating Kalman filter: **Space Weather**

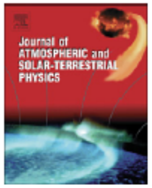
Journal of Atmospheric and Solar-Terrestrial Physics 70 (2008) 1243–1250

ELSEVIER

Contents lists available at ScienceDirect

Journal of
Atmospheric and Solar-Terrestrial Physics

journal homepage: www.elsevier.com/locate/jastp



Review article

Neural networks in auroral data assimilation

Fabrício P. Härter [a,b,c,*], Haroldo F. de Campos Velho [a,b,c],
Erico L. Rempel [a,b,c], Abraham C.-L. Chian [a,b,c]

[a] *University of Waterloo (UofW), Waterloo, ON, Canada 2NL 3G1*
[b] *National Institute for Space Research (INPE), São José dos Campos, SP, 12227-010, Brazil*
[c] *Instituto Tecnológico de Aeronáutica, Praça Marechal Eduardo Gomes, 50, CEP 12228-900, São José dos Campos, Brazil*

A R T I C L E I N F O

Article history:
Received 14 November 2006
Received in revised form
14 February 2008
Accepted 23 March 2008
Available online 18 April 2008

Keywords:
Auroral radio emissions
Nonlinear dynamics
Chaos
Data assimilation
Kalman filter
Neural networks

A B S T R A C T

Data assimilation is an essential step for improving space weather forecasting by means of a weighted combination between observational data and data from a mathematical model. In the present work data assimilation methods based on Kalman filter (KF) and artificial neural networks are applied to a three-wave model of auroral radio emissions. A novel data assimilation method is presented, whereby a multilayer perceptron neural network is trained to emulate a KF for data assimilation by using cross-validation. The results obtained render support for the use of neural networks as an assimilation technique for space weather prediction.

Data assimilation – NN emulating KF

- NN emulating Kalman filter: **Shallow Water 1D**

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Applied Mathematical Modelling 32 (2008) 2621–2633

APPLIED MATHEMATICAL MODELLING

www.elsevier.com/locate/apm

New approach to applying neural network in nonlinear dynamic model

Fabrício P. Härter *, Haroldo Fraga de Campos Velho

Instituto Nacional de Pesquisas Espaciais, Laboratório Associado de Computação e Matemática Aplicada, São José dos Campos, SP, Brazil

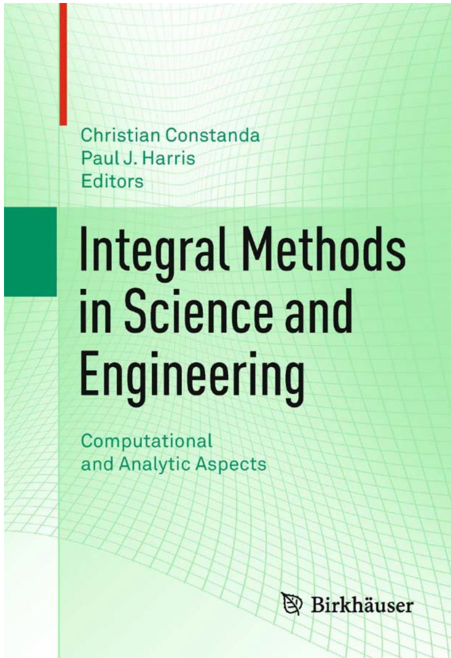
Received 2 January 2007; received in revised form 31 July 2007; accepted 17 September 2007
Available online 30 October 2007

Data assimilation – NN emulating PF

- NN emulating Particle filter: **Lorenz's system**

Adaptive Particle Filter for Stable Distribution

H.F. de Campos Velho and H.C. Morais Furtado



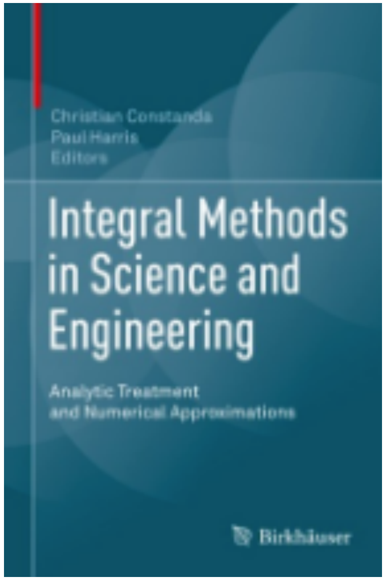
Data assimilation – NN emulating PF

- NN emulating Particle filter: **Lorenz's system**

Integral Methods in Science and Engineering pp 25-35 | Cite as

q-Calculus Formalism for Non-extensive Particle Filter

Amarisio S. Araújo, Helaine C. M. Furtado, Haroldo F. de Campos Velho



Data assimilation – NN emulating PF

- NN emulating Particle filter: **Lorenz's system**

IOPscience iopscience.iop.org

Dynamic Days South America 2010 IOP Publishing
Journal of Physics: Conference Series **285** (2011) 012036 doi:10.1088/1742-6596/285/1/012036

Neural networks for emulation variational method for data assimilation in nonlinear dynamics

Helaine C. Morais Furtado

Haroldo F. de Campos Velho

Elbert E. Macau

Data assimilation – NN emulating PF

- NN emulating Particle filter: **Lorenz's system**

IOPscience   iopscience.iop.org

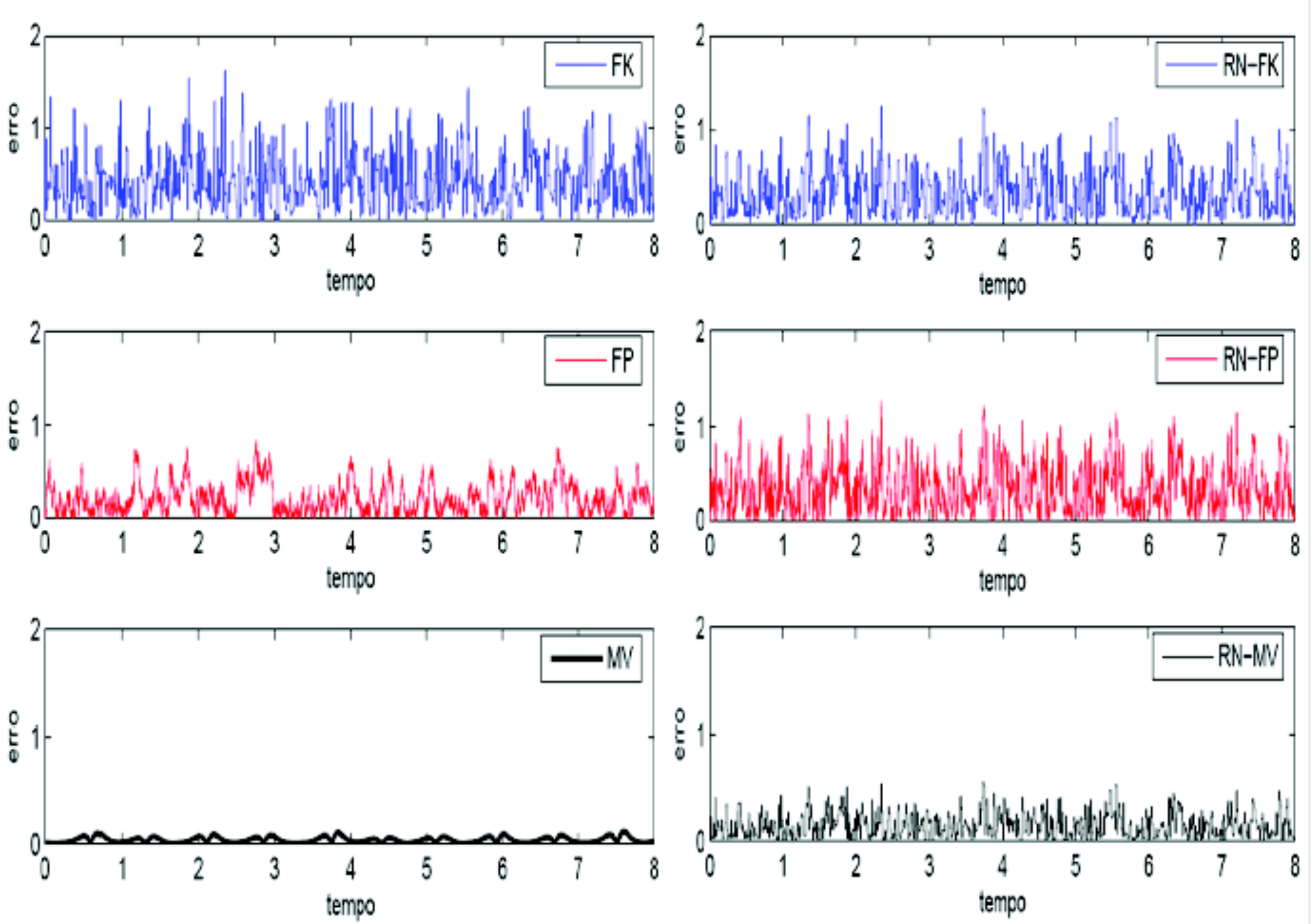
6th International Conference on Inverse Problems in Engineering: Theory and Practice   IOP Publishing
Journal of Physics: Conference Series **135** (2008) 012073   doi:10.1088/1742-6596/135/1/012073

Data assimilation: particle filter and artificial neural networks

Helaine Cristina Morais Furtado,
Haroldo Fraga de Campos Velho,
Elbert Einstein Nehrer Macau

Data assimilation – comparison

- NN emulating: Kalman filter, particle filter, 4D-Var (ERROR)



Data assimilation – NN applications

Neural network emulating (error evolution):

1. Kalman Filter[1]
2. Particle Filter[2]
3. Variational method (4D-Var[3] and Representer[4])
4. LETKF[5] (Local Ensemble Transform Kalman Filter)

Models:

a) Low dimensional model: Lorenz63[1,2,3], shallow water 1D[1]
b) Solar dynamics[1]
c) Oceanic circulation[1,4] (shallow water 2D)
d) AGCM: SPEEDY[5] and FSUGSM[5] (global spectral models)