

# Redes Neurais Artificiais na Melhoria de Desempenho de Métodos de Assimilação de Dados: Filtro de Kalman

R.S. CINTRA<sup>1</sup>, H.F. DE CAMPOS VELHO<sup>2</sup>, Laboratório de Computação e Matemática Aplicada - LAC, Instituto Nacional de Pesquisas Espaciais - INPE, Av. dos Astronautas, 1758, 12.227-010 São José dos Campos, SP, Brasil

R. TODLING<sup>3</sup>, Global Modelling and Data Assimilation Office - GMAO, NASA/Goddard Space Flight Center, 27770 Greenbelt, MD, EUA.

**Resumo.** Assimilação de Dados é um método que combina dados de um modelo matemático e dados de observações, permitindo uma melhoria na previsão do modelo. Métodos seqüenciais ótimos são baseados em teoria de estimativa formal que minimiza os erros dos dados de acordo com a dinâmica do modelo. Métodos de assimilação de dados utilizando Redes Neurais Artificiais (RNA) vêm sendo propostos muito recentemente apresentando resultados consistentes: computacionalmente eficientes e eficazes quanto aplicação. Este trabalho apresenta uma abordagem do método de assimilação por RNA, onde aplica-se uma RNA para substituir o cálculo da inversão de matrizes de erros constante do algoritmo de assimilação baseado em filtro de Kalman. Para exemplo da aplicação desta abordagem, utilizou-se o Sistema de Lorenz e o Filtro de Kalman Estendido para obter parâmetros usados no treinamento da RNA e na comparação dos resultados.

**Palavras-chave.** Assimilação de dados, redes neurais artificiais, sistemas não lineares.

## 1. Introdução

O problema de determinar as melhores condições iniciais para Previsão Numérica de Tempo (PNT) é de grande importância prática, e assunto de muitos estudos por pessoas de cenários diferentes. Em meteorologia e outros ramos de geofísica o processo de aproximar o “verdadeiro” estado de um sistema físico em um determinado momento é chamado análise. Os modelos de PNT são simulações computacionais da atmosfera que tomam a análise como ponto de partida e desenvolvem o estado da atmosfera em um determinado instante de tempo, usando a compreensão de física e dinâmica do fluido atmosférico. A análise meteorológica é produzida pelo processo

---

<sup>1</sup>rosangela.cintra@lac.inpe.br

<sup>2</sup>haroldo@lac.inpe.br

<sup>3</sup>todling@gmao.gsfc.nasa.gov

de assimilação de dados, onde a informação recebida da observação é usada em conjunto com a previsão mais recente de um modelo numérico no instante de tempo que a observação foi feita [11].

A qualidade da Previsão do Tempo é fruto da formulação dos modelos matemáticos e físicos e da quantidade e qualidade dos dados observados. A atmosfera se comporta como um sistema caótico, pois é muito sensível à pequenas variações nas condições iniciais. Por esta razão, existe a busca constante de obter a melhor análise, que é a melhor estimativa do estado atual da atmosfera para iniciar o ciclo de PNT. Para maior compreensão e maiores detalhes em assimilação de dados atmosféricos veja em [4].

Diferentes algoritmos de assimilação podem ser derivados de uma fonte comum, com aproximações próprias para cada algoritmo (Interpolação Ótima(OI), métodos variacionais (3D-VAR, 4D-VAR) e Filtro de Kalman). A Assimilação de dados pode ser descrita como um processo de dois passos:

$$\begin{aligned} \text{Passo de previsão:} \quad x_n^f &= F[x_{n-1}^f] \\ \text{Passo de análise:} \quad x_n^a &= F[x_n^f] + d_n, \end{aligned}$$

onde  $x_n$  representa o vetor de estado do modelo no passo de tempo  $t_n$ ,  $F[\cdot]$  é o modelo matemático de previsão, os sobrescritos  $f$  e  $a$  denotam respectivamente os valores preditos e de análise, finalmente  $d_n$  é o incremento da análise. O vetor incremento de análise é calculado como sendo um produto entre uma matriz de ponderação e uma função que mede a discrepância entre a previsão do modelo e as observações  $y^o$ :

$$d_n = M_n(y_n^o - x_n^f),$$

sendo  $M_n$  a matriz de ponderação ou matriz de ganho. Esta matriz pode ser calculada por um estimador de mínimos quadrados [4]. Para o sistema de análise, sabe-se que existem erros no modelo e nas observações. Os diversos métodos de assimilação existentes procuram uma estratégia que minimize a diferença entre a análise e a “verdade”. Para projetar um algoritmo que faça isto automaticamente, é necessário representar matematicamente os erros dos estados, modelados estatisticamente usando conceitos probabilísticos. Assim, o algoritmo de análise pode ser formulado como um problema de otimização, onde se quer minimizar o erro médio quadrático.

O desafio computacional para a metodologia clássica de assimilação de dados reside na dimensão destas matrizes desenvolvidas em modelos de PNT, atualmente na ordem de um milhão de equações (o que equivale a matrizes cheias da ordem de  $10^{12}$  elementos!). É neste cenário que se insere novas metodologias para assimilação de dados. A metodologia que utiliza Redes Neurais Artificiais(RNA) pode ser uma solução, veja em Nowosad [15] e Härter [8, 7].

Com o enfoque na solução do algoritmo de análise formulado como um problema de otimização aplicou-se a técnica de Redes Neurais Artificiais (RNA) neste trabalho, desenvolvendo uma rede Perceptron de Múltiplas Camadas (PMC) com o algoritmo de treinamento *retro-propagação*. Como exemplo de sistema dinâmico usou-se o Sistema de Lorenz devido a sua não linearidade e natureza caótico e como método de assimilação de dados, o Filtro de Kalman Estendido (EKF). A carga computacional do EKF reside na avaliação da evolução da matriz de covariância

de erros do estado [20] e computação da matriz *ganho*. Neste ponto, a nossa investigação na utilização da RNA foi para diminuição da carga computacional do EKF, uma vez que a avaliação da matriz de erro do estado não é computada. Os resultados do Filtro de Kalman com o Modelo de Lorenz foram as entradas da PMC no treinamento.

As redes neurais apresentam uma melhoria no desempenho computacional dos sistemas de assimilação e são capazes de realizar com sucesso o processo de ingestão de dados de observação sem se desviar da dinâmica do processo.

## 2. Assimilação de Dados

Assimilação de dados é um meio de manter o estado do modelo próximo à natureza pela assimilação de observações. Com base na formulação matemática do problema de análise faz-se a definição do espaço de trabalho. Técnicas clássicas de assimilação para reduzir a resolução ou domínio de análise utilizam o espaço da observação, ou seja, a localização da observação o mesmo espaço escolhido neste trabalho. Admitese que este espaço em modelos mais realistas é composto “somente” por observações convencionais.

Na formulação do problema, a representação do estado atmosférico, é uma matriz chamada vetor de estado  $x$ . A melhor representação da realidade é chamado  $x^t$ , o estado *verdadeiro* no instante da análise. O vetor de estado  $x^b$ , é um estado de *referência*, ou seja, uma previsão do modelo recente para o mesmo instante da análise, que representa um estado hipotético verdadeiro. Finalmente, o vetor análise chamado  $x^a$ , o estado que se quer encontrar. O problema da análise pode ser encontrar uma correção  $\delta x$  de  $x$  tal que  $x^a = x^b + \delta x$ , a análise  $x^a$  deve ser o mais próximo possível de  $x^t$  [19].

Valores observados são reunidos em um vetor observação  $y$ , este vetor deve ser capaz de ser comparado com o vetor estado  $x^b$ . Uma função que ajusta o espaço do modelo para o espaço da observação e a unidade da variável analisada, é chamada de operador observação, de notação  $H$ . As diferenças entre as observações e o vetor de estado, no ponto da observação  $y - H(x)$ , é o vetor de partida e quando calculado com o modelo de referência  $x^b$ , é chamado *vetor inovação*, e calculado com a análise  $x^a$  de *incremento de análise* [4]. A função  $H$  será usada também para que o incremento de análise volte para o espaço do modelo.

A equação fundamental para análise linear em uma forma algébrica geral aqui representada, será a estimação de mínimos quadrados [19]. Definida pelas seguintes equações:

$$x^a = x^b + K(y - H[x^b]) \quad (2.1)$$

$$K = BH^T(HBH^T + R)^{-1}, \quad (2.2)$$

onde o operador linear  $K$  é chamado de ganho, ou matriz peso, da análise.

Para representar as incertezas do modelo, das observações e da análise, assumemse alguns modelos de erros, calculados com uma função de densidade de probabilidade (FDP) para cada tipo de erro. As FDPs dos erros de observação e do modelo são “gaussianas”, então  $x^a$  é também a estimação de variância mínima de  $x^t$ .  $\mathbf{B}$  e  $\mathbf{R}$

são matrizes de covariâncias de erros do modelo e da observação respectivamente, são *pré-determinadas estatisticamente* [1].

## 2.1. Filtro de Kalman

O Filtro de Kalman constitui um processo recursivo *eficiente* de estimação, uma vez que o erro quadrático médio é “minimizado”, ou seja, é uma solução eficiente do método de *mínimos quadrados*. A implementação do filtro de Kalman padrão para um modelo numérico de PNT atual tem uma carga computacional inaceitável. Para obter um filtro computacionalmente eficiente em assimilação de dados, simplificações têm sido introduzidas.

O modelo propriamente dito é definido por duas equações: a equação do processo e a equação da medida. O Filtro de Kalman (KF) [10] na sua versão estendida *Extended Kalman Filter (EKF)* desenvolve o método de mínimos quadrados para uma estrutura não linear, na qual o modelo de referência provém de uma previsão que é iniciada por uma análise prévia. As equações do filtro de Kalman linear são exatamente as equações (2.1) e (2.2), descritas anteriormente, exceto que as Matrizes de covariância de erro do modelo de referência (de previsão) e da análise passam a ser chamadas  $P^f$  e  $P^a$  respectivamente, pois no *KF* de fato calcula-se a Matriz de covariância de erro da análise em tempo de execução. Podemos observar o algoritmo do Filtro de Kalman na (Figura 1).

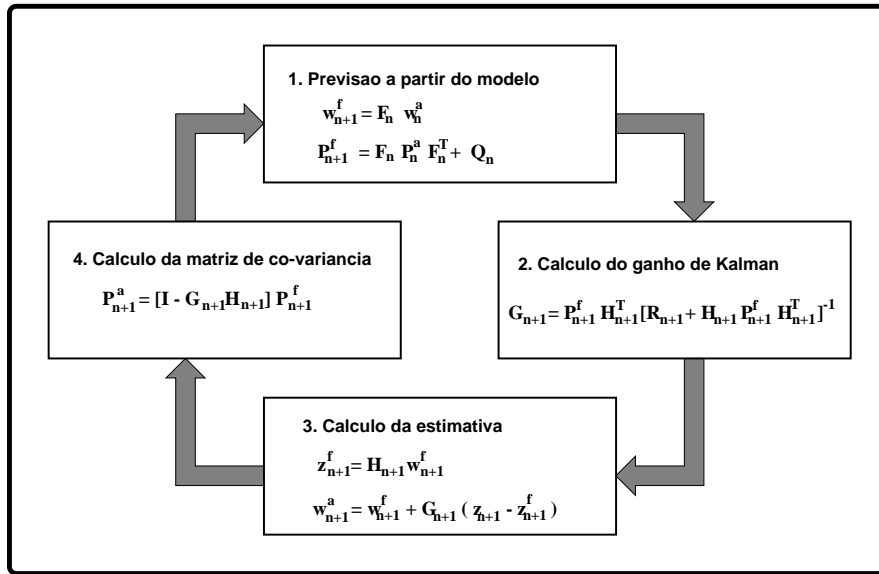


Figura 1: Algoritmo do Filtro de Kalman linear.

## 2.2. Redes Neurais Artificiais

As RNA são sistemas paralelos distribuídos compostos por unidades de processamento simples (nós ou neurônios) que computam determinadas funções matemáticas (normalmente não lineares). Tais unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões. Um modelo de rede neural denominado Perceptron Multicamadas (PMC) [12], possui interconexões das entradas à camada de saída por pelo menos uma camada de neurônios intermediária denominada camada escondida. No modelo PMC as conexões estão associadas a “pesos”, que armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede [5].

As RNA possuem a capacidade de aprender através exemplos. Um algoritmo de aprendizado é um conjunto de procedimentos usados para adaptar os parâmetros da RNA e produzir uma saída esperada. A convergência do processo de aprendizado (supervisionado) por correção de erros é a diferença mínima entre a soma ponderada das entradas pelo pesos e a saída desejada. Este algoritmo é chamado de *retropropagação* do erro e foi o algoritmo utilizado neste trabalho. A forma genérica para alteração dos pesos por correção de erros é definida por  $w_i(t+1) = w_i(t) + \eta e(t)x_i(t)$ , onde  $\eta$  é a taxa de aprendizado e  $x_i(t)$  é a entrada para o neurônio  $i$  no instante de tempo  $t$ . O ajuste dos pesos  $w_i(t)$  deve ser proporcional ao produto do erro entre a saída calculada pela rede e a saída esperada, utilizando o valor de entrada  $x_i(t)$  naquele instante de tempo  $t$ . Cada camada (escondida ou saída) tem um número pré-determinado de neurônios e padrões de conectividade fixos.[13] O número de neurônios na camada escondida, geralmente é definido empiricamente, considerando a dimensão das entradas e a convergência do treinamento na medida suficiente para extrair as características dos dados de entrada para a generalização da rede.

## 2.3. Assimilação de Dados: Filtro de Kalman + Redes Neurais

A formulação matemática da equação da análise utilizando a RNA neste trabalho busca a melhoria de desempenho computacional, comparada à aplicação das equações da análise acima descritas. Esta formulação é equivalente no caso linear como:

$$x^a - x^b = BH^T(HBH^T + R)^{-1}(y - H(x^b)) \quad (2.3)$$

dividida em duas igualdades:

$$\lambda = (HBH^T + R)^{-1}(y - H(x^b)) \quad (2.4)$$

$$x^a - x^b = BH^T\lambda, \quad (2.5)$$

onde  $\lambda$  tem a mesma dimensão que  $y$  e pode ser considerado o *incremento de análise* no espaço de observação, considerando que  $BH^T$  é um termo que mapeia este parâmetro para o espaço do modelo. O objetivo é resolver o problema de análise em termos de  $\lambda$  em lugar da *matriz produto da multiplicação das matrizes de covariâncias do erro*. Para resolver para  $\lambda$  o sistema linear é

$$(HBHT + R)\lambda = (y - H(x^b)) . \quad (2.6)$$

Na implementação da análise utilizando redes neurais artificiais, temos o seguinte algoritmo:

1. Cálculo do vetor de partidas  $y - H(x^b)$ ;
2. Obtenção do vetor  $\lambda$  com a ativação da RNA;
3. Multiplicação do vetor  $\lambda$  estimado pelo  $BH^T$  para obter os incrementos de análise;
4. Adicionar os incrementos ao modelo de referência  $x^b$  e obter a análise  $x^a$  no espaço do modelo.

Neste trabalho, utilizou-se um PMC para determinar o parâmetro  $\lambda$  após o treinamento com os dados de entrada:  $\lambda$  “alvo” e vetor inováção da execução do EKF com Modelo de Lorenz e observações sintéticas. A obtenção do parâmetro  $\lambda$  foi após o cálculo do termo  $(HBHT + R)(y - H(x^b))$ . No final da execução do EKF obteve-se também a Matriz de Covariância  $B$  para utilização na ativação da PMC. O treinamento da RNA foi feito para obter os pesos “ótimos” na sua convergência. Com os pesos adequados, a ativação da PMC determinou o  $\lambda$  para cada entrada do modelo de Lorenz em nova realização. O EKF foi executado também para comparação dos resultados.

## 2.4. Análise de Complexidade

A expressão *complexidade* aqui é no sentido de avaliar o número de operações escalares de pontos flutuante dos sistemas de assimilação deste trabalho. O desempenho de um sistema pode ser medido pela complexidade de seu algoritmo. As medidas numéricas de tempo de execução não serão mostradas aqui, pois a aplicação trabalha com modelos simples, isto é, o processamento é muito rápido.

Para avaliar a complexidade computacional dos métodos de assimilação de dados, será adotada a hipótese de uma rede com  $m_0 = 2m_2$  entradas,  $L$  camadas escondidas de  $m_1$  neurônios e uma camada de saída com  $m_2$  neurônios, *treinada* para emular um Filtro de Kalman. Em cada neurônio, as  $m$  entradas alimentam a função de ativação através do produto interno

$$s = \sum_{i=1}^m \theta_i x_i, \quad (2.7)$$

cuja complexidade é  $O(m)$ . Qualquer que seja a função de ativação, pode-se supor que esta função de ativação seja representada por uma série de Taylor truncada. O valor funcional de um polinômio pode ser calculado pela regra de Horner (ou *multiplicação aninhada*) [2, 17] (páginas 33 e 44, respectivamente):

$$y = \varphi(s) = a_0 + a_1 s + a_2 s^2 + \dots + a_{n-1} s^{n-1} + a_n s^n \quad (2.8)$$

$$= a_0 + s(a_1 + s(a_2 + \dots + s(a_{n-1} + a_n s) \dots)), \quad (2.9)$$

que calcula o polinômio  $\varphi(s)$  com  $n$  multiplicações e  $n$  somas, tendo complexidade, portanto, ordem  $O(n)$ .

A função de assimilação implementada pela rede tem complexidade de ordem

$$m_1 \cdot [O(m_0) + O(n)] + (L - 1) \cdot m_1 [O(m_1) + O(n)] + m_2 [O(m_1) + O(n)].$$

Se  $O(n) < O(m_1) = O(m_2) = O(m)$  e  $L \ll m_2$  então o algoritmo para calcular a saída da função de assimilação de dados por redes neurais  $x_n^a = F_{ANN}(x_n^f; x_n^o)$  terá complexidade  $O(m^2)$  [15].

Por outro lado, o número de operações para o filtro de Kalman é  $O(2m^2n) + O(2mn^2) + (m^3) + O(n^3)$ . Se  $n = m$  então o número de computações é equivalente a  $O(3m^3)$ , ou seja, a complexidade de um Filtro de Kalman discreto com  $m$  variáveis de estado e  $n$  observações [6]. Este desempenho é devido aos produtos matriciais a cada passo do algoritmo KF.

Aqui, a rede neural é usada como um *substituto* para o cálculo do ganho de Kalman (2.3) (ver também Figura 1). O ganho de Kalman é expresso por:

$$B_{t+1} G_{t+1} = P_{t+1}^f H_{t+1}, \quad \text{onde } B_{t+1} = R_{t+1} + H_{t+1} P_{t+1}^f H_{t+1}^T. \quad (2.10)$$

Na equação acima, há 4 produtos de matrizes de dimensões  $m \times m$  e de  $m$  sistemas para o cálculo das colunas da matriz  $G_{t+1}$ . O produto de matrizes tem a mesma complexidade do cálculo das colunas da matriz  $G_{t+1}$  (equivalente a complexidade de inversão de matriz):  $O(m^3)$ . O algoritmo de Strassen tem complexidade  $O(m^{2.807})$  [18]. Em 1990, Coppersmith e Winograd [3] apresentaram um algoritmo de complexidade  $O(m^{2.376})$ . Embora este algoritmo seja uma referência para análises teóricas, ele não é prático (diferentemente do algoritmo de Strassen), pois só existe vantagem para matrizes muito grandes, que não podem ser processadas pelos atuais computadores [16].

## 3. Resultados

### 3.1. Modelo de Lorenz

Edward Lorenz (1963) [14] desenvolveu um modelo matemático do modo como o ar se move na atmosfera, e chegou à conclusão que com pequenas variações nos valores iniciais das variáveis do seu modelo, obteve resultados muito divergentes. Em sistemas dinâmicos complexos, estes resultados *instáveis* dizem respeito à evolução temporal como função de seus parâmetros e variáveis. Lorenz em sua pesquisa de sistemas dinâmicos usou três equações para representar graficamente o comportamento dinâmico através de computadores, descreveu um sistema relativamente simples com um padrão de complexidade infinita, onde se verificou que a partir de estados iniciais ligeiramente diferentes, o sistema de equações diferenciais resultava em soluções completamente diferentes entre si.

O sistema de Lorenz consiste de três equações diferenciais ordinárias de primeira ordem, acopladas:

$$\begin{aligned} \frac{dx}{dt} &= -\sigma x - y \\ \frac{dy}{dt} &= -\rho x - y - xz \end{aligned} \quad (3.1)$$

$$\frac{dz}{dt} = xy - \beta z,$$

onde  $\sigma, \rho, \beta$  são parâmetros do modelo, com esta abordagem caótica, utilizou-se os valores 10, 28 e  $8/3$ , respectivamente. As variáveis  $x, y$  e  $z$  possuem uma interpretação espacial. A consequência da instabilidade dos resultados notada no modelo de Lorenz é que mesmo em sistemas determinísticos existe uma grande sensibilidade a perturbações e erros. O sistema de Lorenz é de uso freqüente para testar técnicas de assimilação de dados, por ser um modelo dinâmico simples com comportamento caótico, sensível às condições iniciais [8].

### 3.2. Assimilação para o Modelo de Lorenz

A rede possui três entradas relativas ao vetor inovação das variáveis  $x, y, z$  e  $\lambda$  das respectivas variáveis como *saída esperada* durante o treinamento. Para a rede perceptron de múltiplas camadas, com treinamento por *back-propagation*, não há uma regra estabelecida para determinação de alguns parâmetros da rede, como número de neurônios ou número de camadas escondidas. Aqui, realizamos vários experimentos numéricos para se determinar o número de neurônios da rede que apresentasse melhor desempenho. No final, seis neurônios foram utilizados na camada escondida e na camada de saída três neurônios referentes ao parâmetro  $\lambda$  de  $x, y$  e  $z$ .

Os treinamentos da RNA foram feitos após 32.000, 16.000 e 2.000 passos de tempo, onde ( $\Delta t = 0,00625$ ) na execução do modelo. As informações de entrada foram inseridas em intervalos de 10, 20 e 40 passos. Aqui foram usados dados de observação sintéticos, ou seja, dados do modelo de Lorenz corrompidos com um ruído branco gaussiano e a estratégia de validação cruzada foi empregada no treinamento. A ativação da RNA foi feita com os pesos fixos obtidos após o treinamento e nova execução do modelo de Lorenz.

Os resultados apresentam as trajetórias das variáveis  $x, y, z$  na execução do modelo hipotético *verdadeiro*, do modelo iniciado com a análise do *EKF*, as *observações* e o modelo iniciado com a análise *RNA*. As figuras abaixo indicam que a RNA acompanhou o modelo até o final.

Verificou-se que a trajetória do estado com a condição inicial gerada pela RNA, acompanha as trajetória do estado verdadeiro e do iniciado com EKF, o método que a rede se propôs a emular. A figura 2 representam a dinâmica do modelo no estado verdadeiro. As figuras 3 e 4 mostram uma comparação dos estados verdadeiro, com assimilação EKF e com assimilação das redes neurais.

## 4. Conclusão

Neste trabalho implementou-se o Modelo de Lorenz caótico, o EKF e uma rede PMC com o algoritmo de treinamento *retropropagação* propondo a melhoria de desempenho em métodos de assimilação. As metodologias aplicadas em centros de previsão, embora os resultados sejam satisfatórios, há dúvidas se os algoritmos são computacionalmente eficientes para realizar a tarefa de assimilação com o aumento exponencial de dados de observação, em tempo de uma previsão operacional. É importante investigar algoritmos que sejam computacionalmente mais eficientes.



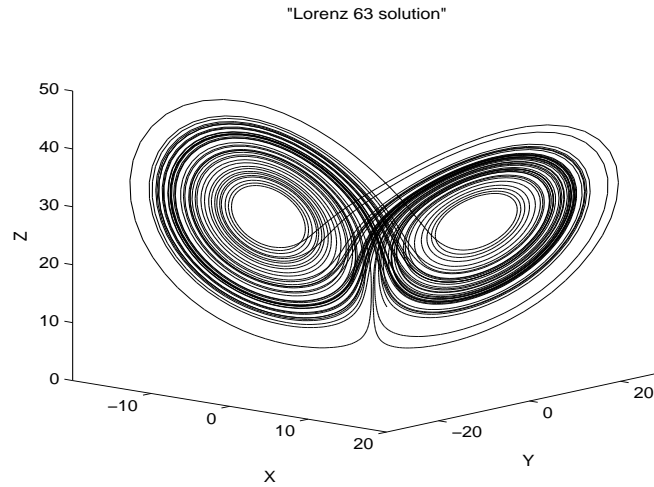


Figura 2: Sistema de Lorenz [14].

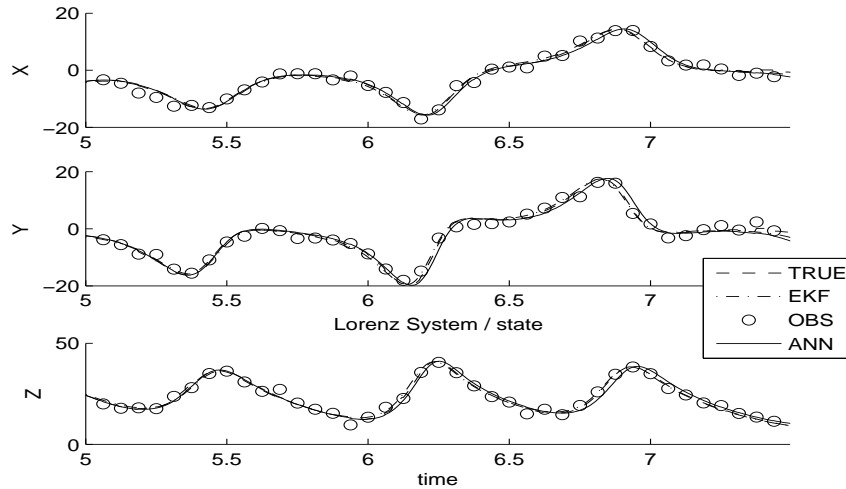


Figura 3: Sistema de Lorenz - observações a cada dez passos (círculos), estado verdadeiro (linha pontilhada), estado com EKF (linha traço-ponto), estado com RNA (linha sólida).

Investigou-se uma abordagem da metodologia de assimilação de dados com Redes Neurais Artificiais em um modelo simples e verificou-se que com a obtenção do *pseudo-ganho* ( $\lambda$ ) pela RNA (já treinada) apresenta melhoria computacional em relação ao FK. Este cálculo de matrizes é realizado com os métodos tradicionais como Interpolação Ótima e FK, para obtenção do campo inicial das variáveis meteorológicas de modelos numéricos de previsão de tempo.

Pode-se concluir que as variáveis de estado de um modelo numérico de previsão atual num processo de assimilação de dados tem ordem de  $10^{7-9}$  e requer matrizes

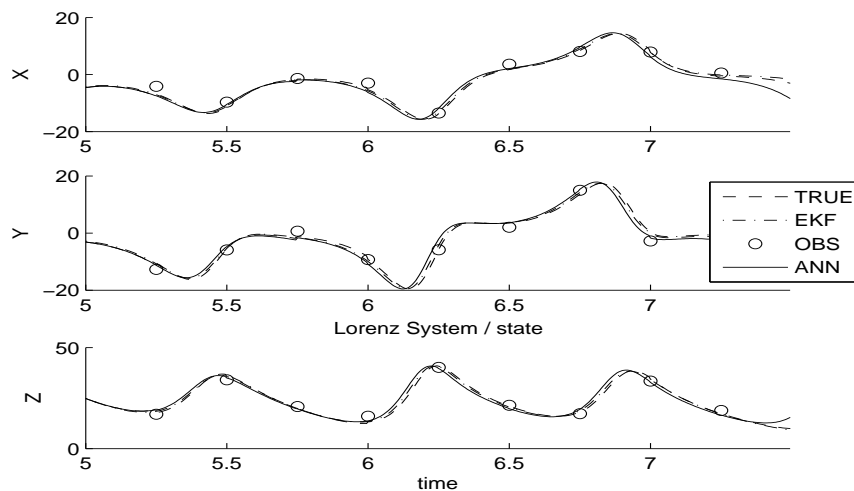


Figura 4: Sistema de Lorenz - observações a cada quarenta passos (círculos), estado verdadeiro (linha pontilhada), estado com EKF (linha traço-ponto), estado com RNA (linha sólida).

cheias não-nulas no FK – complexidade  $O(m^{2.807})$  (ou, de forma teórica,  $O(m^{2.376})$ ), e numa rede neural treinada para emular a assimilação de Kalman (complexidade  $O(m^2)$ ) obterá uma *análise* em tempo de computação significativamente menor.

**Abstract.** Data assimilation is a method for combining data from a mathematical model and observations, producing a better model prediction. Optimal sequential methods are based on formal estimation theory to minimize the errors in the data according to model dynamics. Artificial neural networks (ANN) have been recently applied to data assimilation presenting consistent results: they are computationally efficient and effective. This paper presents one application of ANN to calculate inverse matrix in the Kalman filter algorithm. The scheme is tested using the Lorenz system. The standard extended Kalman filter was used for a comparison and to obtain the data set for training the ANN.

## Referências

- [1] F. Boutier, P. Courtier, Data Assimilation concepts and methods, in “Meteorological Training Course”, Reading, UK, 1998.
- [2] S.D. Conte, C. de Boor, “Elementary Numerical Analysis: an Algorithmic Approach”, McGraw-Hill, 3rd Edition, 1980.
- [3] D. Coppersmith, S. Winograd, Matrix multiplication via arithmetic progressions, *Journal of Symbolic Computation*, **9** (1990), 251–280.
- [4] R. Daley, “Atmospheric Data Analysis”, Cambridge University Press, 1991.

- [5] M.W. Gardner, S.R. Dorling, Artificial neural networks (the multilayer perceptron) – a review of applications in the atmospheric sciences, *Atmospheric Environment*, **32**, No. 14–15 (1998), 2627–2636.
- [6] M.J. Goris, D.A. Gray, I.M.Y. Marcelis, Reducing the computational load of a Kalman filter, *IEE Electronics Letters*, **33**, No. 12 (1997), 1539–1541.
- [7] F.P. Härter, H.F. de Campos Velho, New approach to applying neural network in nonlinear dynamic model, *Applied Mathematical Modelling*, **32** (2008), 2621–2633. doi: 10.1016/j.apm.2007.09.006
- [8] F.P. Härter, “Redes Neurais Recorrentes Aplicadas à Assimilação de Dados em Dinâmica Não Linear”, Tese de Doutorado, Computação Aplicada, CAP-INPE, São José dos Campos, SP, 2004.
- [9] F.P. Härter, H.F. de Campos Velho, Recurrent and feedforward neural networks trained with cross correlation applied to the data assimilation in chaotic dynamic, *Revista Brasileira de Meteorologia*, **20**, No. 3 (2005), 411–420.
- [10] R.E. Kalman, A new approach to linear filtering and prediction problems, *Trans. of the ASME—Journal of Basic Engineering*, **82**, No. D (1960), 35–45.
- [11] E. Kalnay, “Atmospheric Modeling, Data Assimilation and Predictability”, Cambridge University Press, 2003.
- [12] S. Haykin, “Neural Networks: A Comprehensive Foundation”, Pearson Education, 2002.
- [13] S. Haykin, “Adaptive Filter Theory”, Mcmillan, 1994.
- [14] E.N. Lorenz, Deterministic nonperiodic flow, *Journal of the Atmospheric Physics*, **20** (1963), 130–141.
- [15] A.G. Nowosad, “Novas Abordagens para Assimilação de Dados Meteorológicos”, Tese de Doutorado, CAP-INPE, 2001.
- [16] S. Robinson, Toward an optimal algorithm for matrix multiplication, *SIAM News*, **38**, No. 9 (2005).
- [17] J. Stoer, R. Bulirsch, “Introduction to Numerical Analysis”, Springer-Verlag, 3rd Edition, 2000.
- [18] V. Strassen, Gaussian elimination is not optimal, *Numerische Mathematik*, **13** (1969), 354–356.
- [19] O. Talagrand, Assimilation of observations, an introduction, *J. Meteor. Soc. Japan*, **75** (1997), 91–209.
- [20] R. Todling, S.E. Cohn, Suboptimal schemes for atmospheric data assimilation based on the Kalman filter, *Mon. Weat. Rev.*, **122** (1994), 2530–2557.