Journées Systèmes de Référence Spatio-Temporels
Scientific developments from highly accurate space-time reference systems
16 - 18 September 2013 - Paris

Prediction of UT1 using Artificial Neural Networks

Pedro Lopes[1], Jean-Yves Richard[1], Christophe Barache[1], Daniel Gambis[1] , Christian Bizouard[1]
(1) Observatoire de Paris SYRTE / GRGS, Paris, France (Pedro.Lopes@obspm.fr)

Session 4: Earth Rotation - Theory, modelling and observations

ABSTRACT

The monitoring of the Earth Orientation Parameters (EOP) variations is the main task of the IERS Earth Orientation Centre. EOP have applications on navigation, precise orbit determination or leap seconds announcements, meaning short that and long term predictions are required.
Currently, the method applied for predictions is based on deterministic processes, namely Least Square fitting and autoregressive filtering. Here, we present Artificial Neural Networks (ANN) as an alternative. These have successfully been applied for pattern recognition, and have been tested by various authors for EOP predictions. Though, so far no real improvement was shown when compared with the current methods.
However, recent mechanisms allow reconsidering the use of ANN for EOP predictions. Hence, a series of simulations for short and long term predictions; as well as statistics and comparisons with the current methods are presented.

INTRODUCTION

The object of the project is to study the use of Artificial Neural Networks, namely Multi-Layer Perceptron (MLP), to perform predictions of one of Earth's rotation parameters, UT1 and to estimate the possibility to use those networks to predict the introduction of Leap Second.

Artificial Neural Networks – ANNs

Like the human brain, the ANN learns by training and can perform tasks such as function approximation, pattern recognition or prediction of future events. Fig. 1 shows a typical example.

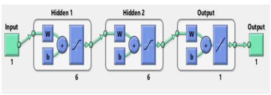

Fig. 1 - A representation of a feed forward ANN with connections between layers: 2 hidden layers with 6 neurons each and an output layer.

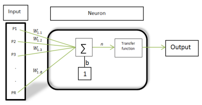

Fig. 2 - Simple Neuron Diagram: input vector is sent to the neuron and processed by a transfer function, resulting in the neural network output

Single Neuron

A single neuron follows a simple mathematical equation, where the output is the result of the product between inputs and weights (Fig.2 and equation 1).

$$output = f(\boldsymbol{Wp} + \boldsymbol{b}) \quad (1)$$

The input vector, **p** of dimensions R, is multiplied by the weight matrix, **W**. This matrix is randomly generated by the toolbox every time a new network is created and trained. The result of this dot product is called weighted input that is added to the vector **b**, bias, to form the net input, *n*.

$$n = W_{1,1}p_1 + W_{1,2}p_2 + \cdots W_{1,R}p_R + b \quad (2)$$

This net input is then passed through the transfer function, *f*, which will in turn produce the network's output. The most common and the one used for this project is the tangent-sigmoid function.

Training the Neural Network

During training, the values of weights and bias are changed to increase the network's performance, which is measured by the network performance function - mean squared error (MSE) between output and targets.

Long Term Prediction on UT1

Presentation:
This section aims to study the performance of a feed forward network when used with real data with a time step of 5 days, trying to approximate and predict the UT1-TAI from 1962 to 2013, in order to estimate the maximum prediction horizon possible with ANN. Two kinds of networks have been simulated and summarized in table 1.

Network 1 *with validation checks (NN1)*	Network 2 *without validation checks (NN2)*
2 hidden layers with 6 neurons each Network parameters: • Multi-Layer Perceptron networks (MLP) • Neural Network Structure: Feed Forward • Tangent-Sigmoid transfer function • Training algorithms Levenberg-Marquardt (LM) back propagation • Random Data Division: division splits the input data into three sub datasets: 70% for Training, 15% for Validation and 15% for Testing • **W**eights and **b**ias initialization: random values	2 hidden layers with 12 neurons each Network parameters: • Multi-Layer Perceptron networks (MLP) • Neural Network Structure: Feed Forward • Tangent-Sigmoid transfer function • Training Algorithm: Bayesian Regularization (BR) • Random Data Division disabled: all the training sample will be used for training and there will be no validation stop. This means that each trial will take all 10000 iterations to finish or training will be stopped when one of the other parameters converge • **W**eights and **b**ias initialization: random values

Table 1: ANN ***NN1*** & ***NN2*** simulated for long term predictions

Training data is used only for training, adjusting weights and bias
Validation set is used for training and to stop it if the errors increase 6 times in a row (validation stop)
Test data is only used to compare models and represents the answer of the network having no effect in training
Performance evaluation: using Mean Square Error & Regression Analysis

Procedure:
- Each training sample goes through N trials, defined by the user (set to 10 in this experimentation), where the sample is used to train a new network and generate new predictions N times. All network parameters and results are saved in an *.mat file* after each trial. The best network is chosen based on the longest prediction horizon with en error < 0,9s.
- Training sample is increased with one year of data and the procedure repeats this behavior until the training sample size reaches its maximum.

Results
Figure 3 shows the long term prediction limit where |UT1-UTC| reaches 0,9s using Neural Network NN1 & NN2 averaged and best network among the 10 trials each and the Least Squares Method ([7] Gambis Model).

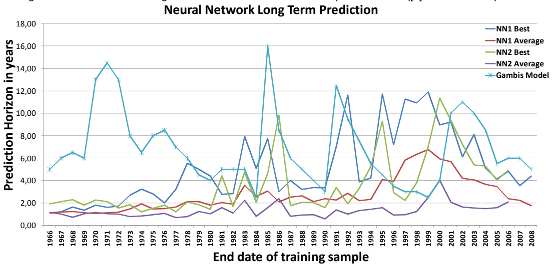

Fig.3 - Long Term Prediction using ANN compared to the Least Squares Method (Gambis Model)

Fig. 4 shows the prediction of the network 2 at cycle 21 (MJD=47519), and although it manages to begin to predict a small sinusoidal signal, it fails to actually approximate the real data and diverges quickly soon after. The occasional good and long prediction showed by these networks seem to be more a matter of chance than a good result where the prediction turns out to be a curve, without changes and tangent to the target data (see Fig.5 for cycle 27 of Network 1).

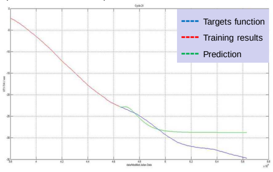

Fig.4 – Network 2 Prediction at cycle 21

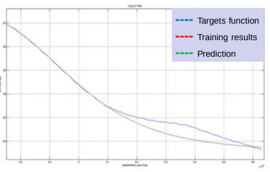

Fig.5 – Network 1 Prediction at cycle 27: Tangent Prediction

Bibliography

[1] *SYRTE*. (n.d.). Retrieved from http://hpiers.obspm.fr/eop-pc/index.php?index=leapsecond&lang=en
[2] Mark Hudson Beale, M. T. (2013). *Neural Network Toolbox™ User's Guide.* Retrieved from http://www.mathworks.co.uk/help/nnet/index.html
[3] *Levenberg*. (n.d.). Retrieved from Mathworks: http://www.mathworks.co.uk/help/nnet/ref/trainlm.html
[4] *Bayesian*. (n.d.). Retrieved from Mathworks: http://www.mathworks.co.uk/help/nnet/ref/trainbr.html
[5] *Improved Generalization*. (n.d.). Retrieved from Mathworks: http://www.mathworks.co.uk/help/nnet/ug/improving-generalization.html
[6] Machine Learning Methods in the Environnemental Sciences, Neural Networks and Kernels, William W. Hsieh, Cambridge University Press
[7] Gambis,D and Luzum B., 2011, Earth rotation monitoring, UT1 determination and prediction, Metrologia 48, S165-S170.
[8] Prediction of Earth orientation parameters by artificial neural networks, 2002, Schuh H., M. Ulrich, D. Egger, J.Muller, W. Schwegmann, Journal of Geodesy 76: 247–258

Short Term Prediction

Presentation:
The prediction horizon is estimated from 5 to 25 days using different kinds of neural networks and 1 to 10 days.
The goal is to minimize Root Mean Squared Error (RMSE) by testing several critical parameters of the networks, such as training sample size and number of neurons.

Network 1 with Increasing sample size and proportional number of neurons	*Network 2 with Increasing sample size and constant number of neurons*
2 hidden layers with increasing number of neurons from 1 to 12 (the number of training months) Network Parameters: • Multi-Layer Perceptron networks (MLP) • Neural Network Structure: Feed Forward • Tangent-Sigmoid transfer function • Training algorithms Levenberg-Marquardt (LM) back propagation • Random Data Division: removes data division, all data is used for training • **W**eights and **b**ias initialization: random values	2 hidden layers with 4 neurons each Network Parameters: • Multi-Layer Perceptron networks (MLP) • Neural Network Structure: Feed Forward • Tangent-Sigmoid transfer function • Training algorithms Levenberg-Marquardt (LM) back propagation • Random Data Division: removes data division, all data is used for training • **W**eights and **b**ias initialization: random values

Table 2: ANN 1 & 2 simulated for short term predictions

Procedure:
- Iteration of training data size from 1 month to 12 months with 1 month steps
- For each training sample 10 trials are performed and the best is chosen in terms of MSE for each prediction horizon (from 5 to 25 days by 5 days stepping)
- the training sample window is shifted by 1 month N times in order to collect the RMS for each horizon for different samples of constant size.

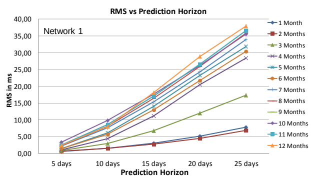

Fig. 6 - RMS for an increasing sample and number of neurons

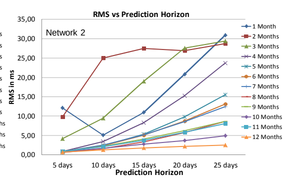

Fig. 7 - RMS for increasing sample and constant number of neurons

Network 3: daily training, 2 hidden layers of 2 neurons
Using the daily data of UT1-TAI for training – Prediction horizon until 10 days Method: training with different sizes from 4, 10, 20 and 365 days • These training samples of constant size are trained over 100 steps and for each step 10 trials are performed • From the 10 trials, the best is chosen, based on the lowest RMS, which represents the trial with the least error • This trial will then represent the prediction error of that step. The same is done for each of the next steps • Based on the prediction error of the 10 steps we calculate the RMS for that sample (table 3)

Sample size	RMS in milliseconds for a prediction horizon of:									
Days	1 days	2 days	3 days	4 days	5 days	6 days	7 days	8 days	9 days	10 days
4 (Series 1)	0,913575	1,69958	2,953057	4,30566	5,6858689	7,088269	8,505233	9,92678	11,35163	12,78
10 (Series 2)	0,336091	0,706539	1,041066	1,37853	1,8503255	2,51942	3,357175	4,32092	5,387449	6,5438
20 (Series 3)	0,185875	0,322345	0,491731	0,67442	0,8713904	1,101273	1,38747	1,74448	2,174952	2,6722
365 (Series 4)	0,651144	0,790716	0,954395	1,14036	1,3516377	1,586719	1,845221	2,12814	2,433499	2,7617

Table 3 : Daily predictions for different training samples sizes

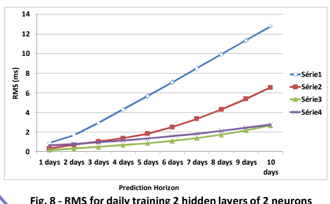

Fig. 8 - RMS for daily training 2 hidden layers of 2 neurons

Results:
Network 1: The evolution of the RMS for the smaller samples vs. larger samples suggests that the dependency between sample size and neurons is not linear and as such there might be too many neurons for the largest samples.
Network 2: Based on Network 1 results the RMS for the larger samples was improved reducing the number of neurons. The increase in RMS for the smaller samples supports the previous conclusion.
Network 3: Results show again the necessity of finding proper balance between sample size and number of neurons. With the 20 days sample returning very similar results to the "3 Months" sample (18 points) of Network 1 at 5 and 10 days horizon.

CONCLUSION

- For Long Term Predictions standard MLP appear to have a maximum prediction horizon of 1 year with 4 years of training and 2 years with 16 years of training before the predictions diverge too much from the real function (NN1). This and the long computation time that is taken to run such large training samples makes them very limited option for our applications.
- Short Term Predictions, however, can be applied and are generally faster to compute when using small samples (20 days for training). In order for them to be accurate, it requires a proper balance between sample size and number of neurons in order to minimize the RMS.
- Although the standard MLP fails to return reliable long term predictions, Matlab Neural Network Toolbox has more advanced networks available that could be of interest, such as dynamic networks called NARX (Nonlinear Auto-Regressive with EXternal Input) that use their own output to predict the next values of the series. Their predictive potential is seen when used to predict a Sinus function, being able to predict the function over 6 periods whereas the standard MLP only predicts up to a quarter of a period, similar to the limit of long term predictions seen above.