An intelligent adaptive model for data forecasting

Iulian Nastac
nastac@ieee.org

Colocviile ETTI - 3/5/2012
Summary

• General issues that concerns an ANN implementation

• Retraining technique

• Forecasting applications
ANN Applications

- Function identification
- Clustering and classification
- Data forecasting
How to design an ANN application

• Analyze the database
• Find the proper ANN architecture
• Use an efficient training algorithm
• Test it on specific conditions
Find a proper ANN architecture

• Use a specific ratio (~ 5/1) between the number of the training samples and the total number of the weights

• Evaluate several architectures
  - Use a multiple loop to search various combinations
Starting point

The ANN weights are initialized to small uniformly distributed values.
Use an efficient training algorithm

• A **fast algorithm** (e.g., Levenberg-Marquardt) uses large computational resources.

• A **slow training algorithm** uses a decent amount of memory and is more robust.

• Usually you don’t need the fastest algorithm since the **validation process** can stop too early the training.
Testing the model

• The ANN model very depend on the training set (that include the validation set).

• Sometimes the test set can come from a changing environment…

• How to deal with non-stationary systems?
General Problem

- computing speed
- parallelism
- reliability
- programming
- generalization
Starting example

Imaging and Learning in Back-Scattered Light by Artificial Neural Networks

I. Nastac - Politehnica Univ. Bucharest (Romania)
R. Matei – Nokia Networks Oy, Helsinki (Finland)
P. Cristea - Politehnica Univ. Bucharest (Romania)
M. Kessler - Friedrich-Alexander Univ. Erlangen-Nuremberg (Germany)
Preliminary assumptions for the retraining procedure:

- **Cases:**
  - *same function and same training set*
  - *same function and different training set*
  - *different function with a new training set*

- **Models:**
  - *BKP* (back-propagation)
  - *momentum*
  - *ALR* (Adaptive Rate Learning)
  - *ALR-momentum* combination

- **Data:**
  - *spectral measurements of oxygenation in tissue and the relative concentration of intracapillarity hemoglobin*
  - Micro-Lightguide Spectrophotometer EMPHO (fast diffuse reflection)
a. one input – one output architecture

b. two inputs - one output architecture
THE WORKING ALGORITHM FOR RETRAINING PROCEDURE ANALYSIS

1) Decide initial function $f$ and training set ($n$ pairs)

2) Choose network architecture

3) Select training procedure

4) Initialize network weights with small uniformly distributed values

5) Train network and hold the training cycle number $V$

6) Assign $L$ values to scaling factor $\gamma$

7) Reduce by parameter $\gamma$ the weights obtained at step 5

8) Repeat learning procedure for each $\gamma$ and memorize $V(\gamma_i)$

9) Repeat step 6 for another training set of function $f$
   or for the case of function $g$ (with a similar graph)
The graphs of functions $f$ and $g$ (spectras measured in different points for the same concentration of 95% hemoglobin) $E_d = 0.025$
Retraining procedure through BKP method (same function $f$ and same points $(x_i, f(x_i))$). It is representing the cycles number for each retraining with $\gamma_i$. 
Retraining procedure through BKP method
(same function $f$ and different points $(x'_i, f(x'_i))$)
Retraining procedure through momentum method:
a. same function $f$ and same points $(x_i, f(x_i))$
b. same function $f$ and different points $(x'_i, f(x'_i))$
c. different function $g$ and different points $(x''_i, g(x''_i))$
Retraining through ALR-momentum for a single layer neural networks with 15 neurons - same function $f$ and same points $(x_i, f(x_i))$ $\gamma = 0.1 \ldots 0.9$ (L = 9)

Retraining through ALR-momentum for a single layer neural networks with 15 neurons - same function $f$ and same points $(x_i, f(x_i))$ $\gamma = 0.02 \ldots 0.9$ (L = 45)
Scanning measurements were performed with a 3D-scanning device and the Spectrophotometer EMPHO II SSK

\[ E_d(f_t, g_t, v) = 230 \]
Retraining procedure through ALR-momentum method:

a. same function $f$ and same points
b. different function $g$ and different points
• **Limit situations:**

- initial training data has been chosen wrongly or incompletely

- inadequate limit error $E_l$

- new training set used for retraining has been incorrectly chosen

- function $g$ differs too much from function $f$

- the parameters of learning algorithms has been inadequately chosen

- some combinations of the previous situations
Outcomes:

• $\gamma_{opt} \in [0.4, 0.6]$

• An increase of the training cycles number for $\gamma \geq 0.7$ in 35% of the analyzed cases

• Generalization of the conclusion regarding retraining procedure for networks with any dimension

• Neural classification system for 2D- and 3D-image recognition in tissue
Effect of hidden layer size on network generalization
Learning error versus test error

Classical approach

Number of Training Cycles
(or Number of Hidden Nodes)

Error

Optimum Network

Validation Error

Training Error
Validation-stop improvement

Training set:
- 85% for training
- 15% for validation
Can validation set act as a kind of test set at the same time?
Pyramidal ANN structure

\[ N_o \leq N_{h2} \leq N_{h1} \leq N_i \]

\[ \sqrt{N_i \times N_{h2}} - 5 \leq N_{h1} \leq \sqrt{N_i \times N_{h2}} + 5 \]

\[ \sqrt{N_{h1} \times N_o} - 5 \leq N_{h2} \leq \sqrt{N_{h1} \times N_o} + 5 \]
THE RETRAINING PROCEDURE OF AN ANN

• Training an Artificial Neural Network in standard way with *validation stop*

• Reduction of the first network weights with a *scaling factor* $\gamma$ ($0 < \gamma < 1$). Usually, $\gamma = 0.1, 0.2, \ldots, 0.9$

• Retraining the network with the new initial weights

• Compare the *validation error* in both cases
Adaptive Retraining Technique
(step by step)

Complex ANN application

Building the input-output function ($f_1$)

ANN structure

Training
$N_{tc}$ (number of training cycles)

Utilization

Modification of the application

Building the new input-output function ($f_2$)

ANN structure remains the same

Retraining
$N_{rc}$ (number of retraining cycles)

- improve the output error

Reutilization
Finding ANN Structure

• Each of the training sessions starts with the weights initialized to small, uniformly distributed values.

• Test several pyramidal ANN architectures, with $Nh_1$ and $Nh_2$ taking values in the vicinity of the geometric mean of the neighboring layers.

\[
N_o \leq Nh_2 \leq Nh_1 \leq N_i
\]

\[
\sqrt{N_i \times Nh_2} - 5 \leq Nh_1 \leq \sqrt{N_i \times Nh_2} + 5
\]

\[
\sqrt{Nh_1 \times N_o} - 5 \leq Nh_2 \leq \sqrt{Nh_1 \times N_o} + 5
\]

• Chose the best model with respect to the smallest error between the desired and the simulated output.
Forecasting issues

Nonstationary Sequences Forecasting

Sequences:  ● temporal sequences (time series)

● spatial sequences (chains of objects, features, DNA, etc.)

Tool: Artificial Intelligence Model

Retraining Neuro-Adaptive Technique
• The data vary dynamically, and large space-delays might occur.

• Solutions are ranked by accuracy of the output forecasting according to the following formula:

$$ERR = \frac{100}{T} \sum_{p=1}^{T} \left| \frac{O_{Rkp} - O_{Fkp}}{O_{Rkp}} \right| \cdot f(p)$$

where $T$ is the number of time steps, $O_{Rkp}$ – the real output $k$ at space/time step $p$, $O_{Fkp}$ – the forecasted output $k$ at space/time step $p$, and

$$f(p) = \frac{T}{T + p}$$
Research Approach

• The sequences under discussion are inherently nonstationary.

• The nonstationary characteristic implies that the distribution of the sequences may changes at different positions.

• Furthermore, some gradual changes in the dependency between the input and output variables may appear.

• The recent data points could provide more important information than the distant data points.

• We propose a new adaptive retraining mechanism to take this characteristic into account.
Forecasting Neural Network
Training Process

\[ y_k(t + 1) = F(X(t + 1 - In\_Del(i)), Y(t - Out\_Del(j))) \]
Adaptive Retraining

Legend:
- training & validation set
- useful predictions

Diagram showing the process of adaptive retraining with steps and initial position.
Extending the forecasts - Iterative Simulations (IS)

Steps:

- **Start**: Construct at least one forecast using real data at the input (selected by In_Del and Out_Del).

- **Treat the previous forecasts** (selected by Out_Del) as observed values in order to produce other successive forecasts (there is a combination of real and simulated data at the input).

Using an iterative process, a forecast can be extended as many time steps as required.
Alternative:
Always Real Inputs (ARI)

• The Always Real Inputs (ARI) approach employs the real previous outputs and not estimated ones.
Output Forecasting
Experiments

• **Cases:** I. Time series: Industrial & Financial Forecasting

  II. Spatial Sequences: DNA Sequence Forecasting

• Scale Conjugate Gradient (SCG) as basic training algorithm

• Different delay vectors for `In_Del` and `Out_Del`:

\[
In\_Del = [i\_d_1, i\_d_2, \ldots, i\_d_n]
\]

\[
Out\_Del = [o\_d_1, o\_d_2, \ldots, o\_d_m]
\]

• ERR
Graphical Analysis

The quality of the predictions can be also analyzed graphically, by enforcing a tube around the real outputs, given by a function like the one below:

\[ f(n) = A + q \cdot n \]

where:

• A is an acceptable prediction error;
• q is an increasing factor;
• n is the number of predicted timesteps.

Then, the predicted output values should lay in the interval \( \text{output}_k(n) \pm f(n) \).

For example:

\[ f(n) = 300 + 0.05 \cdot n \]
The goal is to find a practical mathematical model that describes the relationship between 16 input variables and 4 output variables that model a process in glass manufacturing.

The raw data consist of 16000 rows (timesteps) – one data row every 15 minutes during 6 months.

- The goal is to find a practical mathematical model that describes the relationship between 16 input variables and 4 output variables that model a process in glass manufacturing.
For this industrial example it seems that the outputs have different meanings and different time-delay behaviors. Consequently, the normal way is to split the initial system in four subsystems. Good results were obtained when we started to work under the following assumptions:

- \( \text{In} \_\text{Del} = [0 \ 1 \ 3] \) and \( \text{Out} \_\text{Del} = [0 \ 1] \) for output 1;
- \( \text{In} \_\text{Del} = [0 \ 1 \ 2 \ 3 \ 5 \ 8 \ 12 \ 18 \ 28] \) and \( \text{Out} \_\text{Del} = [0 \ 1 \ 3] \) for output 2;
- \( \text{In} \_\text{Del} = [0 \ 1 \ 2 \ 3 \ 4 \ 6 \ 8 \ 12 \ 20] \) and \( \text{Out} \_\text{Del} = [0 \ 1] \) for outputs 3 and 4;
- \( V = 7500 \) timesteps are enough data for each training / retraining phase;
- \( T = 500 \) timesteps represent the prediction horizon;
- \( \text{Shift} = 500 \) timesteps is the shifting time for the next retraining.

The prediction horizon can be, for example, enlarged to 1500 timesteps. The number of samples for the training (or retraining) interval can be modified according to the experience accumulated.
Unexpected effect of error's decreasing for Iterative Simulations (IS)

\[ f(n) = 3 + 0.003 \cdot n \]
Extension of the forecasting

500 timesteps

1568 timesteps
What's the matter without retraining?

<table>
<thead>
<tr>
<th>Training Interval</th>
<th>Test Interval</th>
<th>ERR (output_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 8000</td>
<td>8001 - 8500</td>
<td>2.8613</td>
</tr>
<tr>
<td>501 - 8500</td>
<td>8501 - 9000</td>
<td>5.4797</td>
</tr>
<tr>
<td>1001 - 9000</td>
<td>9001 - 9500</td>
<td>11.1089</td>
</tr>
<tr>
<td>1501 - 9500</td>
<td>9501 - 10000</td>
<td>1.4871</td>
</tr>
<tr>
<td>2001 - 10000</td>
<td>10001 - 10500</td>
<td>2.0086</td>
</tr>
<tr>
<td>2501 - 10500</td>
<td>10501 - 11000</td>
<td>3.6669</td>
</tr>
<tr>
<td>3001 - 11000</td>
<td>11001 - 11500</td>
<td>2.618</td>
</tr>
<tr>
<td>3501 - 11500</td>
<td>11501 - 12000</td>
<td>2.8078</td>
</tr>
<tr>
<td>4001 - 12000</td>
<td>12001 - 12500</td>
<td>3.672</td>
</tr>
<tr>
<td>4501 - 12500</td>
<td>12501 - 13000</td>
<td>2.9132</td>
</tr>
<tr>
<td>5001 - 13000</td>
<td>13001 - 13500</td>
<td>1.2381</td>
</tr>
<tr>
<td>5501 - 13500</td>
<td>13501 - 14000</td>
<td>1.4595</td>
</tr>
<tr>
<td>6001 - 14000</td>
<td>14001 - 14500</td>
<td>1.8028</td>
</tr>
</tbody>
</table>

---

*without retraining*
Effect of PCA transformation matrix adaptation

without PCA adaptation

with PCA adaptation
Data forecasting of outputs 3 and 4 for a unified model
B. Adaptive Monitoring

Iulian Nastac and Paul Cristea
Politehnica University of Bucharest

- **Goal**: model the **catalyst activity** of a multi-tube reactor, used to oxidize a gaseous feed.

- The primary raw data consist of 5807 rows (timesteps) – one data row every one hour during 8 months.

- Other four sets of 720 rows (one month) each are successively used for test and update de model also.
**Inputs:**
- Measured flow of air (kg/hr);
- Measured flow of combustible gas (kg/hr);
- Measured concentration of combustible component in combustible gas feed in mass fraction;
- Total feed temperature;
- Cooling temperature;
- Seven temperatures (in Celsius) from different points of reactor length;
- Product concentration of oxygen in mass fraction;
- Product concentration of combustible component in mass fraction.

**Output:** *catalyst activity* of a multi-tube reactor
Premise

- \( V = 3407 \) timesteps are enough for the first training phase and then for each retraining phase;

- \( T = 720 \) (or 800 for initially tests) timesteps represent the prediction horizon;

- \( \text{Shift} = 720 \) (or 800 for initially tests) timesteps is the shifting time for the next retraining.
Training and retraining phases

Legend:
- - training & validation set
- - useful predictions

Training
Retraining 1
Retraining 2
Retraining 3
Retraining 4
Retraining 5
Retraining 6

initial position
steps

Test 1
Test 2
Test 3
Test 4
## Characteristics of the models

<table>
<thead>
<tr>
<th></th>
<th><strong>Input delay vector</strong> <em>(In_Del)</em></th>
<th><strong>Output delay vector</strong> <em>(Out_Del)</em></th>
<th>Use time as input</th>
<th><strong>Dimension of PCA trans. matrix</strong></th>
<th><strong>ANN Structure</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>0 1 2 3 4 5 6</td>
<td>0 1 2</td>
<td>Yes</td>
<td>$40 \times 101$</td>
<td>40:19:8:1</td>
</tr>
<tr>
<td>Model 2</td>
<td>0 1 2 3 4 5 6 8</td>
<td>0 1 2</td>
<td>Yes</td>
<td>$40 \times 115$</td>
<td>40:17:3:1</td>
</tr>
<tr>
<td>Model 3</td>
<td>0 1 2 3 4 5 6 8</td>
<td>0 1 2</td>
<td>No</td>
<td>$40 \times 107$</td>
<td>40:10:5:1</td>
</tr>
<tr>
<td>Model 4</td>
<td>0 1 2 3 4 5 6 8 12</td>
<td>0 1 2</td>
<td>Yes</td>
<td>$40 \times 129$</td>
<td>40:18:8:1</td>
</tr>
<tr>
<td>Model 5</td>
<td>0 1 2 4 6 9 12 16 20 24 39</td>
<td>0 1 2 4</td>
<td>Yes</td>
<td>$50 \times 158$</td>
<td>50:9:8:1</td>
</tr>
</tbody>
</table>
Evolution of *ERR* during First Training, Retraining 1 and Retraining 2

<table>
<thead>
<tr>
<th>Model</th>
<th>ERR</th>
<th>First training</th>
<th>Retraining 1</th>
<th>Retraining 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>16.371</td>
<td>2.3826</td>
<td>7.2881</td>
<td></td>
</tr>
<tr>
<td>Model 2</td>
<td>12.922</td>
<td>1.333</td>
<td>10.562</td>
<td></td>
</tr>
<tr>
<td>Model 3</td>
<td>17.879</td>
<td>3.8248</td>
<td>7.5423</td>
<td></td>
</tr>
<tr>
<td>Model 4</td>
<td>21.93</td>
<td>22.307</td>
<td>8.113</td>
<td></td>
</tr>
<tr>
<td>Model 5</td>
<td>14.768</td>
<td>2.656</td>
<td>8.1371</td>
<td></td>
</tr>
</tbody>
</table>
Evolution of $ERR$ during test phases

<table>
<thead>
<tr>
<th>Model</th>
<th>$ERR_h$ Test 1 (Retraining 3)</th>
<th>$ERR_h$ Test 2 (Retraining 4)</th>
<th>$ERR_h$ Test 3 (Retraining 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>53.408</td>
<td>19.726</td>
<td>24.108</td>
</tr>
<tr>
<td>Model 2</td>
<td>38.833</td>
<td>20.87</td>
<td>12.046</td>
</tr>
<tr>
<td>Model 3</td>
<td>46.409</td>
<td>20.49</td>
<td>24.805</td>
</tr>
<tr>
<td>Model 4</td>
<td>42.979</td>
<td>21.485</td>
<td>11.185</td>
</tr>
<tr>
<td>Model 5</td>
<td>44.369</td>
<td>28.099</td>
<td>12.829</td>
</tr>
</tbody>
</table>
Data forecasting of Model 1 for test intervals of first training, retraining 1 and retraining 2
<table>
<thead>
<tr>
<th></th>
<th>Test 1 (Retraining 3)</th>
<th>Test 2 (Retraining 4)</th>
<th>Test 3 (Retraining 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive Model</td>
<td>64.2225</td>
<td>33.7336</td>
<td>40.6924</td>
</tr>
<tr>
<td>ARMA Model</td>
<td>50.915</td>
<td>20.483</td>
<td>41.591</td>
</tr>
<tr>
<td>Neural Adaptive Model</td>
<td>46.409</td>
<td>20.49</td>
<td>24.805</td>
</tr>
</tbody>
</table>
Graphical results of ARMA and Neural-Adaptive models

- **real data**
- **naive simulated output**
- **ARMA simulated output**
- **neural-adaptive simulated output (Model 3)**
C. Exchange Rate Forecasting

Iulian Nastac and Emilian Dobrescu

UPB and Romanian Academy

- The particular system, which resulted by using our approaches, describes the relationship between over 30 variables and one output variable that model the EURO-ROL exchange rate.

- The raw data consist of over 2500 rows (time steps) – one data row every day during 8 years (2000 – 2008).
# Database indicators

<table>
<thead>
<tr>
<th>Indicators</th>
<th>Symbol</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. Statistical information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A. General information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1  1. Real Gross Domestic Product growth</td>
<td>GDP</td>
<td>Quarterly</td>
</tr>
<tr>
<td>2  2. Current Account deficit</td>
<td>CA</td>
<td>Monthly</td>
</tr>
<tr>
<td>3  3. Consolidated general budget deficit as percentage on GDP</td>
<td>CGD</td>
<td>Quarterly</td>
</tr>
<tr>
<td>4  4. Net foreign direct investment</td>
<td>FDI</td>
<td>Monthly</td>
</tr>
<tr>
<td>5  5. Medium and long term external dept</td>
<td>ExD</td>
<td>Monthly</td>
</tr>
<tr>
<td>6  6. NBR Foreign exchange reserve</td>
<td>ER</td>
<td>Monthly</td>
</tr>
<tr>
<td>7  7. Export of good and services</td>
<td>X</td>
<td>Monthly</td>
</tr>
<tr>
<td>8  8. Import of good and services</td>
<td>M</td>
<td>Monthly</td>
</tr>
<tr>
<td>9  9. Net monthly average wage on the economy</td>
<td>Nw</td>
<td>Monthly</td>
</tr>
</tbody>
</table>
### B. Specifics information

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Code</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1. Exchange rate Dollar/ROL</td>
<td>E$</td>
<td>Daily</td>
</tr>
<tr>
<td>11</td>
<td>2. Exchange rate EUR/ROL</td>
<td>Eeur</td>
<td>Daily</td>
</tr>
<tr>
<td>12</td>
<td>3. Consumer goods index</td>
<td>CPIR</td>
<td>Monthly</td>
</tr>
<tr>
<td>13</td>
<td>4. Monetary base M0</td>
<td>M0</td>
<td>Monthly</td>
</tr>
<tr>
<td>14</td>
<td>5. Reference rate of BNR</td>
<td>rd</td>
<td>Monthly</td>
</tr>
<tr>
<td>15</td>
<td>6. Speed between lending and deposit average interest rate of banks for non-government, non-banks clients</td>
<td>Δr</td>
<td>Monthly</td>
</tr>
<tr>
<td>16</td>
<td>7. Total domestic credit</td>
<td>DC</td>
<td>Monthly</td>
</tr>
<tr>
<td>17</td>
<td>8. Portfolio investment, sold</td>
<td>PI</td>
<td>Monthly</td>
</tr>
<tr>
<td>18</td>
<td>9. Current transfers and incomes</td>
<td>CTI</td>
<td>Monthly</td>
</tr>
<tr>
<td>19</td>
<td>10. Turnover</td>
<td>T</td>
<td>Monthly</td>
</tr>
<tr>
<td>20</td>
<td>11. BET Index</td>
<td>BET</td>
<td>Daily</td>
</tr>
</tbody>
</table>

### C. External Information

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Code</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>1. Ratio EUR/Dollar</td>
<td>Ra</td>
<td>Daily</td>
</tr>
<tr>
<td>22</td>
<td>2. Exchange rate Euro/ROL</td>
<td>Reur_{EU}</td>
<td>Daily</td>
</tr>
<tr>
<td>23</td>
<td>3. Refinancing ECB interest rate</td>
<td>R_{ecb}</td>
<td>Monthly</td>
</tr>
<tr>
<td>24</td>
<td>4. Brent oil price</td>
<td>op</td>
<td>Monthly</td>
</tr>
<tr>
<td>25</td>
<td>5. HIPC (EU 27)</td>
<td>HIPC</td>
<td>Monthly</td>
</tr>
</tbody>
</table>
### II: Prospective information

#### A. General information

<table>
<thead>
<tr>
<th>No.</th>
<th>Category</th>
<th>Symbol</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>1. Real GDP growth</td>
<td>$GDP_f$</td>
<td>Annual</td>
</tr>
<tr>
<td>27</td>
<td>2. Export of good and services, FOB, growth rate</td>
<td>$X_f$</td>
<td>Annual</td>
</tr>
<tr>
<td>28</td>
<td>3. Import of good and services, FOB, growth rate</td>
<td>$M_f$</td>
<td>Annual</td>
</tr>
<tr>
<td>29</td>
<td>4. Commercial trade deficit, mill Euro</td>
<td>$C_t$</td>
<td>Annual</td>
</tr>
<tr>
<td>30</td>
<td>5. Growth of consumer price, annual average</td>
<td>$CPI_f$</td>
<td>Annual</td>
</tr>
<tr>
<td>31</td>
<td>6. Growth of consumer price, December/December</td>
<td>$CPI_{d_f}$</td>
<td>Annual</td>
</tr>
</tbody>
</table>

#### B. Specific forecasting information

<table>
<thead>
<tr>
<th>No.</th>
<th>Category</th>
<th>Symbol</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>1. Inflation target</td>
<td>$IT_f$</td>
<td>Annual</td>
</tr>
<tr>
<td>33</td>
<td>2. Future exchange rate Dollar/ROL, 1 month</td>
<td>$FeS$</td>
<td>Daily</td>
</tr>
<tr>
<td>34</td>
<td>3. Future exchange rate Euro/ROL 1 month</td>
<td>$FeEur$</td>
<td>Daily</td>
</tr>
<tr>
<td>35</td>
<td>4. Ratio EUR/Dollar, 1 month</td>
<td>$Fra$</td>
<td>Daily</td>
</tr>
</tbody>
</table>
Input variables:

- 32 (or 35) statistical and prospective financial variables;
- Month indicator L (the days of January are noted by 1, the days of February by 2 and so on).

Output variable:

- Daily exchange rate EUR/ROL
Premise

• We worked under the following assumptions:
  - \( V = 2200 \) is the interval employed for training/validation purpose;
  - \( T = 1 \) (or 3, 7, 15, 30) days represent the prediction horizon (test set);
  - \( Shift = 1 \) is the shifting interval for the next retraining.

• The prediction horizon can be easily enlarged to 60 steps or even more.

• Models:

  Case I: \( \text{In\_Del} = \{1, 2, 3, 4, 5, 6, 8, 12\} \) and
  \( \text{Out\_Del} = \{0, 1, 2, 4\} \).

  Case VII: \( \text{In\_Del} = \{7, 8, 9, 10, 11, 13, 16\} \) and
  \( \text{Out\_Del} = \{0, 1, 2, 4\} \).

• Scale Conjugate Gradient (SCG) is the basic training algorithm.
# Evolution of test error (ERR) for Iterative Simulations

<table>
<thead>
<tr>
<th>Training / retraining interval</th>
<th>Time period for training</th>
<th>ERR (Case I)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>33 inputs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>T=1</td>
</tr>
<tr>
<td>First training</td>
<td>01.01.2000 - 01.08.2006</td>
<td>2.1387</td>
</tr>
<tr>
<td>Retraining 1</td>
<td>01.02.2000 - 01.09.2006</td>
<td>0.25772</td>
</tr>
<tr>
<td>Retraining 2</td>
<td>01.03.2000 - 01.10.2006</td>
<td>0.45772</td>
</tr>
<tr>
<td>Retraining 3</td>
<td>01.04.2000 - 01.11.2006</td>
<td>0.21918</td>
</tr>
<tr>
<td>Retraining 4</td>
<td>01.05.2000 - 01.12.2006</td>
<td>0.80736</td>
</tr>
<tr>
<td>Retraining 5</td>
<td>01.06.2000 - 01.13.2006</td>
<td>0.049294</td>
</tr>
<tr>
<td>Retraining 6</td>
<td>01.07.2000 - 01.14.2006</td>
<td>0.015575</td>
</tr>
<tr>
<td>Retraining 7</td>
<td>01.08.2000 - 01.15.2006</td>
<td>0.11574</td>
</tr>
<tr>
<td>.....</td>
<td>.....</td>
<td>.....</td>
</tr>
<tr>
<td>Retraining 39</td>
<td>02.09.2000 - 02.16.2006</td>
<td>0.063032</td>
</tr>
<tr>
<td>Retraining 40</td>
<td>02.10.2000 - 02.17.2006</td>
<td>0.035246</td>
</tr>
</tbody>
</table>
Evolution of test error (ERR) for Iterative Simulations (IS) and Always Real Inputs (ARI) when T=30

<table>
<thead>
<tr>
<th>Training / retraining interval</th>
<th>Time period for training</th>
<th>ERR (Case I)</th>
<th>33 inputs</th>
<th>36 inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>IS</td>
<td>ARI</td>
<td>IS</td>
</tr>
<tr>
<td>First training</td>
<td>01.01.2000 - 01.08.2006</td>
<td>4.3185</td>
<td>4.2391</td>
<td>6.8036</td>
</tr>
<tr>
<td>Retraining 1</td>
<td>01.02.2000 - 01.09.2006</td>
<td>0.5592</td>
<td>0.54745</td>
<td>1.6479</td>
</tr>
<tr>
<td>Retraining 2</td>
<td>01.03.2000 - 01.10.2006</td>
<td>0.73979</td>
<td>0.7293</td>
<td>1.7582</td>
</tr>
<tr>
<td>Retraining 3</td>
<td>01.04.2000 - 01.11.2006</td>
<td>0.9866</td>
<td>0.95797</td>
<td>3.6997</td>
</tr>
<tr>
<td>Retraining 4</td>
<td>01.05.2000 - 01.12.2006</td>
<td>0.95671</td>
<td>0.92525</td>
<td>0.99947</td>
</tr>
<tr>
<td>Retraining 5</td>
<td>01.06.2000 - 01.13.2006</td>
<td>0.44314</td>
<td>0.4334</td>
<td>0.65897</td>
</tr>
<tr>
<td>Retraining 6</td>
<td>01.07.2000 - 01.14.2006</td>
<td>0.40513</td>
<td>0.39207</td>
<td>0.73477</td>
</tr>
<tr>
<td>Retraining 7</td>
<td>01.08.2000 - 01.15.2006</td>
<td>0.5245</td>
<td>0.49901</td>
<td>0.66889</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Retraining 39</td>
<td>02.09.2000 - 02.16.2006</td>
<td>0.20866</td>
<td>0.20311</td>
<td>0.53497</td>
</tr>
<tr>
<td>Retraining 40</td>
<td>02.10.2000 - 02.17.2006</td>
<td>0.38001</td>
<td>0.35892</td>
<td>0.59025</td>
</tr>
</tbody>
</table>
ERR trend (Case I – 33 inputs) of test sets for the first training and $L = 40$ successive retraining phases
ERR trend (Case I – 36 inputs) of test sets for the first training and $L = 40$ successive retraining phases
Remember

Graphical Analysis

The quality of the predictions can be also analyzed graphically, by enforcing a tube around the real outputs, given by a function like the one below:

\[ f(n) = A + q \cdot n \]

where:

- A is an acceptable prediction error;
- q is an increasing factor;
- n is the number of predicted timesteps.

Then, the predicted output values should lay in the interval \( \text{output}_k(n) \pm f(n) \).

For example:

\[ f(n) = 300 + 0.05 \cdot n \]
Data forecasting for the test interval of retraining 29 (Case I – 33 inputs)

Iterative simulations of output

Always real inputs
Data forecasting for the test interval of retraining 19 (Case I – 33 inputs)
Data forecasting for the test interval of retraining 14 (Case I – 33 inputs)
D. Stock Market Forecasting

• The HEX Forest Industry Index is an important index in Finland economy (HEX - Helsinki Stock Exchange) that illustrates the average global trend in forest industry.

• The goal of was to find a practical mathematical model that describes the relationship between 8 input variables (indices of four biggest Finnish company in forest/ pulp & paper industry; EUR / US Dollar exchange rate; Gold price; NYSE composite index considered to have a noticeable impact; and month indicator), and one output variable that model a forecasting process of this HEX Forest Industry Index.

• The raw data consist of 2960 rows (timesteps) – one data row every working day during 11 years (February 1993 – June 2004).
Premise

• We worked under the following assumptions:
  - $V = 2920$ is the interval employed for training/validation purpose;
  - $T = 20$ steps represent the prediction horizon (test set);
  - $Shift = 1$ is the shifting interval for the next retraining.

• The prediction horizon can be easily enlarged to 50 or 100 working days.

• Model:

  $\text{In\_Del} = [5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 12 \ 16]$
  $\text{Out\_Del} = [0 \ 1 \ 2 \ 4]$
## Evolution of test error (ERR)

<table>
<thead>
<tr>
<th></th>
<th>Training / retraining interval</th>
<th>Test interval</th>
<th>Test Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>First training</td>
<td>1 - 2920</td>
<td>2921 - 2940</td>
<td>4,2686</td>
</tr>
<tr>
<td>Retraining 1</td>
<td>2 - 2921</td>
<td>2922 - 2941</td>
<td>3,6703</td>
</tr>
<tr>
<td>Retraining 2</td>
<td>3 - 2922</td>
<td>2923 - 2942</td>
<td>2,9412</td>
</tr>
<tr>
<td>Retraining 3</td>
<td>4 - 2923</td>
<td>2924 - 2943</td>
<td>2,7541</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Retraining 20</td>
<td>21 - 2940</td>
<td>2941 - 2960</td>
<td>1,8278</td>
</tr>
</tbody>
</table>
Data forecasting for test interval of retraining 20

Iterative simulations of output

Always real inputs

\[ f(n) = 100 + 0.05 \cdot n \]
We were particularly focused on NO2, because its concentration in the air appears to play a negative role in human health.

The model uses 21 inputs that include different kinds of emissions, particles and meteorological parameters.

These parameters were collected every hour during the time period starting from January 2007 to December 2010.

For the training purpose, we used $V=8808$ lines of data that were randomly split in the training set (approx. 85%) and validation set (approx. 15%).
Premise

• For this example we started to work under the following assumptions:
  
  – $In_{Del} = [3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$ and $Out_{Del} = [2]$;
  
  – $V = 8808$ timesteps are enough data for each training / retraining phase since there are data for a bit more than one year;
  
  – $T = 720$ timesteps represent the prediction horizon of one month ($24 \times 30$);
  
  – $Shift = 720$ timesteps is the shifting time for the next retraining.

• The prediction horizon can be, for example, enlarged to 1500 timesteps. The number of samples for the training (or retraining) interval can be modified according to the experience accumulated.
ERR trend of test sets

Iterative Simulation

Always real inputs
Graphical Analysis

The quality of the predictions can be also analyzed graphically, by enforcing a tube around the real outputs, given by a function like the one below:

\[ f(n) = A + q \cdot n \]

where:

- \( A \) is an acceptable prediction error;
- \( q \) is an increasing factor;
- \( n \) is the number of predicted timesteps.

Then, the predicted output values should lay in the interval \( \text{output}_k(n) +/- f(n) \).

For example: \[ f(n) = 15 + 0.005 \cdot n \]
Data forecasting for test interval of retraining 8

Iterative simulations of output

Always real inputs
DNA Sequence Forecasting

Iulian Nastac and Paul Cristea
Biomedical Engineering Centre, UPB

Training Process

The model tries to match the desired value of the output, by properly adjusting the structure in function of the previous values in the DNA sequence.

\[ y(t+1) = F(y(t-Out_Del(j))) \]
Premise

- We worked under the following assumptions:
  - \( V = 5000 \) is the interval employed for \textit{training/validation} purpose;
  - \( T = 500 \) steps represent the prediction horizon (\textit{test set});
  - \( \text{Shift} = 500 \) is the shifting interval for the next retraining.

- The prediction horizon can be easily enlarged to 5000 steps.

- The predicted output values should lay in the interval \( \text{output}(n) +/\!\!- f(n) \), where

\[
f(n) = 1 + 0.05 \cdot n
\]

**Note:** We use the cumulated phases of prokaryote and eukaryote DNA sequences.
Prokaryote DNA sequences forecasting

Evolution of test error (ERR) for Escherichia coli DNA forecasting

<table>
<thead>
<tr>
<th>Training No.</th>
<th>Training/retraining interval</th>
<th>Test interval</th>
<th>Test Error (IS)</th>
<th>Test Error (ARI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>First training</td>
<td>1 - 5000</td>
<td>5001 - 5500</td>
<td>13.051</td>
<td>2.3181</td>
</tr>
<tr>
<td>Retraining 1</td>
<td>501 - 5500</td>
<td>5501 - 6000</td>
<td>22.842</td>
<td>0.5292</td>
</tr>
<tr>
<td>Retraining 2</td>
<td>1001 - 6000</td>
<td>6001 - 6500</td>
<td>24.528</td>
<td>0.8818</td>
</tr>
<tr>
<td>Retraining 3</td>
<td>1501 - 6500</td>
<td>6501 - 7000</td>
<td>3.9406</td>
<td>0.4988</td>
</tr>
<tr>
<td>Retraining 4</td>
<td>2001 - 7000</td>
<td>7001 - 7500</td>
<td>2.463</td>
<td>0.2971</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Retraining 85</td>
<td>42501-47500</td>
<td>47501-48000</td>
<td>0.1772</td>
<td>0.0936</td>
</tr>
<tr>
<td>Retraining 86</td>
<td>43001-48000</td>
<td>48001-48500</td>
<td>0.6931</td>
<td>0.0459</td>
</tr>
<tr>
<td>Retraining 87</td>
<td>43501-48500</td>
<td>48501-49000</td>
<td>0.1590</td>
<td>0.0480</td>
</tr>
</tbody>
</table>

Note: Out_Del=[1 3 5 7 10 14 18 23 28 32 39 47 55 65 80 100 140]
ERR trend of test sets
Data forecasting for test interval of retraining 75 when test set is enlarged to 5000 steps

Note: $\text{ERR}_{\text{ARI}} = 1.4388$
ERR trend of test sets when $T = 5000$
Matrix of the initial PCA block
(the most important eigenvectors)
Matrix of the PCA block
(initial and after 87 retraining steps)
Matrix of the PCA block

(Out_Del=[1 2 3 4 5 ... 35])
Matrix of the PCA block
(initial and after 40 retraining steps)
**Eukaryote DNA sequences forecasting**

Evolution of test error (ERR) for human DNA sequence forecasting (chromosome 22)

<table>
<thead>
<tr>
<th></th>
<th>Training/retraining interval</th>
<th>Test interval</th>
<th>Test Error (IS)</th>
<th>Test Error (ARI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>First training</td>
<td>1 – 5000</td>
<td>5001–5500</td>
<td>27.331</td>
<td>0.5768</td>
</tr>
<tr>
<td>Retraining 1</td>
<td>501–5500</td>
<td>5501–6000</td>
<td>3.5955</td>
<td>1.1674</td>
</tr>
<tr>
<td>Retraining 2</td>
<td>1001–6000</td>
<td>6001–6500</td>
<td>5.5203</td>
<td>1.2491</td>
</tr>
<tr>
<td>Retraining 3</td>
<td>1501–6500</td>
<td>6501–7000</td>
<td>2.3656</td>
<td>0.4446</td>
</tr>
<tr>
<td>Retraining 4</td>
<td>2001-7000</td>
<td>7001–7500</td>
<td>6.9733</td>
<td>0.8124</td>
</tr>
<tr>
<td></td>
<td>....</td>
<td>....</td>
<td>....</td>
<td>....</td>
</tr>
<tr>
<td>Retraining 74</td>
<td>37001-42000</td>
<td>42001-47000</td>
<td>0.6712</td>
<td>0.0429</td>
</tr>
<tr>
<td>Retraining 75</td>
<td>37501-42500</td>
<td>42501-47500</td>
<td>0.0847</td>
<td>0.0223</td>
</tr>
<tr>
<td>Retraining 76</td>
<td>38001-43000</td>
<td>43001-48000</td>
<td>0.2895</td>
<td>0.0244</td>
</tr>
</tbody>
</table>

**Note:** $Out_{Del} = [1, 2, 3, ..., 35]$
ERR trend of test sets (Homo Sapiens case)

Note: $T = 500$ steps and $L = 76$ successive retraining phases
Data forecasting for test interval of retraining 75 (Homo Sapiens case)

Note: $T = 5000$ steps, $ERR_{ARI} = 0.44718$
Matrix of the PCA block
(initial and after 40 retraining steps)
An interesting result

- When training the PCA block on a set of sequences satisfying certain mild statistical regularities, the rows of the resulting matrix are related to the Digital Fourier Transform operator.

- The resemblance with the elements of the Fourier Transformation operator is obvious when the delay vector consists of successive values.

- The evolution of this matrix seems to be very little affected by the changes of the shifted data sets during the retraining steps.

- The first line of the PCA matrix appears to have a slight convex curvature that increases if the genomic signal has a general upward trend (on Escherichia coli sequence), and is going to be more relaxed when genomic signal has an increased trend (on human chromosome 22).
One time series that is not related with DNA

Note: $Out\_Del = [1, 2, 3, \ldots, 35]$
… when we involved many time series

**Note**: $Out_{Del} = [1, 2, 3, ....., 35]$
Open Issues

• Searching for optimum configurations of In_Del and Out_Del to reduce the number of inputs in the recurrent relation:

\[ Y(t + 1) = F(X(t + 1 - In_Del(i)), Y(t - Out_Del(j))) \]

• Managing the length of the data set.
• Preprocessing the data and outliers’ removing.
• Selecting the proper shifting interval.
• Computing the PCA, which depends on:
  - the number of initial inputs and outputs;
  - the number of the elements of In_Del and Out_Del;
  - the number of training pairs;
  - software/hardware limitations.
• Finding the best architecture is time consuming.
A predictive system that can be implemented in a real application
Overall Conclusions

• The adaptive retraining technique can gradually improve, on average, the achieved results.

• The first training always takes a relatively long time, but then the system can be very easily retrained, since there are no changes in the structure.

• The advantage of retraining procedure is that some relevant aspects are preserved ("remembered") not only from the immediate previous training phase, but also from the previous but one phase, and so on.

• The old information accumulated during the older trainings will be slowly forgotten and the learning process will be adapted to the newest evolutions of the process.
Conclusions (cont.)

• Improve the performance of training with validation stop.

• Reduce the number of inputs with PCA.

• It is very easy to change in our tool the SCG algorithm with another one because at the basic level the architecture and the *retraining procedure* are independent of the training algorithm.
Final remarks

• There is a great potential for further development in **data mining**.

• The tool/method can be easily adapted for **other kinds of predictions or classification processes**.
Selected References


