Artificial Neural Networks
in financial applications

February 2003

Content

- Artificial neural networks
  - Introduction
  - Multi-Layer Perceptron
  - Radial-Basis Function Networks
  - Self-Organizing Maps
- About the choice of parameters
  - # of units or parameters
  - # of inputs
- Two application examples in finance
  - Time series forecasting
  - Classification of investment funds
Content

- Artificial neural networks
  - Introduction
  - Multi-Layer Perceptron
  - Radial-Basis Function Networks
  - Self-Organizing Maps
- About the choice of parameters
  - # of units or parameters
  - # of inputs
- Two application examples in finance
  - Time series forecasting
  - Classification of investment funds

Artificial Neural Networks (ANN)

- ANN are:
  - regression and/or classification models with some nice properties (compared to other models)

  - Universal approximation
  - Learning from examples without assumption about the distributions
  - Easy possible scalability to large dimensions
Learning in ANN

- **Supervised learning**: building an input-output relation known through examples (input-output pairs)
  
  ![Diagram of supervised learning]

  - Observed phenomenon
  - Neural network
  - Desired outputs
  - Output

- **Unsupervised learning**: modeling a property of the input data (for example density)
  
  ![Diagram of unsupervised learning]

  - Observed phenomenon
  - Neural network
  - Output

Content

- **Artificial neural networks**
  - Introduction
  - Multi-Layer Perceptron
  - Radial-Basis Function Networks
  - Self-Organizing Maps
- **About the choice of parameters**
  - \# of units or parameters
  - \# of inputs
- **Two application examples in finance**
  - Time series forecasting
  - Classification of investment funds
Multi-layer perceptron (MLP)

\[ y_k(x) = h\left(\sum_{j=0}^{M} w_{kj}^{(2)} g\left(\sum_{i=0}^{D} w_{ij}^{(1)} x_i\right)\right) \]

\[ y(x) = h\left(w_{0}^{(2)} g(w_{0}^{(1)} x)\right) \]

Convention: 2 layers of weights (in literature: sometimes 3 layers of units or neurons)
\(g\) and \(h\) continuous, bounded, non-linear functions (tanh, sigmoid)
\(h\) can be linear but not \(g\) (otherwise only one layer)

Learning in MLP

Learning =
- definition of an error criterion \(E\)
- evaluation of derivatives of \(E\) w.r.t. parameters \(w\)
- adjustments of parameters \(w\) according to derivatives

Error criterion
(for one output):

\[ E = \sum_{p=1}^{P} \| y^p - d^p \|^2 \]

Training examples
Learning in MLP

✓ “Back-propagation”: \( \frac{\partial E}{\partial w_{ij}} \)

✓ For last layer \( w_{ij}^{(2)} \): easy

✓ For other layers: computed according to derivatives of next layer
  \( \rightarrow \) back-propagation of derivatives

✓ More elaborated learning methods:
  ✓ Conjugate gradients
  ✓ Levenberg-Marquardt
  ✓ ...

Universal approximation property

✓ “A 2-layer MLP can approximate arbitrarily well any (functional) continuous mapping, provided the number \( M \) of hidden units is sufficiently large”

✓ But:
  ✓ what about generalization?
  ✓ what about the number \( M \) of hidden units?
  ✓ What about initialization and local minima?
  ✓ Learning is slow, difficult, needs expertise, etc.
Content

- Artificial neural networks
  - Introduction
  - Multi-Layer Perceptron
  - Radial-Basis Function Networks
  - Self-Organizing Maps
- About the choice of parameters
  - # of units or parameters
  - # of inputs
- Two application examples in finance
  - Time series forecasting
  - Classification of investment funds

Radial-Basis Function Networks (RBFN)

\[ F(x) = \sum_{i=1}^{K} w_i \varphi(||x - c_i||) \]

\[ \varphi(||x - c_i||) = \exp\left( - \frac{||x - c_i||^2}{2\sigma_i^2} \right) \]

- Recommended:

\[ F(x) = \sum_{i=1}^{K} w_i \exp\left( - \frac{||x - c_i||^2}{2\sigma_i^2} \right) + \sum_{i=1}^{D} w'_i x_i + w'_0 \]

- RBFN are a generalization of the
  - interpolation problem
  - regularization problem

because \( K \ll P \)
RBFN: learning strategies

\[ F(x) = \sum_{i=1}^{K} w_i \phi(\|x - c_i\|) \quad \phi(\|x - c_i\|) = \exp\left( -\frac{\|x - c_i\|^2}{2\sigma_i^2} \right) \]

- Parameters to be determined: \( c_i, \sigma_i, w_i \)
- Traditional learning strategy: splitted computation
  1. centers \( c_i \)
  2. widths \( \sigma_i \)
  3. weights \( w_i \)

RBFN: computation of centers

- Idea: centers \( c_i \) must have the (density) properties of learning points \( x_k \)
  \[ \rightarrow \text{vector quantization} \]
  - selected at random (in learning set)
  - competitive learning
  - frequency-sensitive learning
  - Kohonen maps
- This phase only uses the \( x_k \) information, not the \( t_k \)
RBFN: computation of widths

- Universal approximation property: valid with identical widths
- In practice (limited learning set): variable widths \( \sigma_i \)
- Idea: RBFN use *local* clusters

choose \( \sigma_i \) according to standard deviation of clusters

\[ F(x) = \sum_{i=1}^{K} w_i \phi(||x - c_i||) \]

\[ \phi(||x - c_i||) = \exp \left( \frac{-||x - c_i||^2}{2\sigma_i^2} \right) \]

Problem becomes linear!

Solution of least square criterion

\[ E(F) = \frac{1}{2P} \sum_{P=1}^{P} (t^0 - F(x^P))^2 \]

leads to

\[ w = \Phi^T \bar{t} \]

where

\[ \Phi = \phi_{kj} = \phi(||x^k - c_j||) \]

In practice: use SVD!
RBFN: gradient descent

\[ F(x) = \sum_{i=1}^{K} w_i \exp \left( -\frac{|x - c_i|^2}{2\sigma_i^2} \right) \]

3-steps method:

- Supervised
- Unsupervised

Once \( c_i, \sigma_i, w_i \) have been set by the previous method, possibility of gradient descent on all parameters
- Some improvement, but
  - Learning speed
  - Local minima
  - Risk of non-local basis functions
  - Etc.

Content

- Artificial neural networks
  - Introduction
  - Multi-Layer Perceptron
  - Radial-Basis Function Networks
  - Self-Organizing Maps
- About the choice of parameters
  - # of units or parameters
  - # of inputs
- Two application examples in finance
  - Time series forecasting
  - Classification of investment funds
Self-organizing maps (SOM): biological inspiration

Human sensory and motor maps

Kohonen map

- Vector quantization
- and topological ordering

- 2-dimensional grid
- for visualisation!

Also possible:
- 1-dimensional string
  (sometimes used)
- x-dimensional cube
  (rarely used)
Examples after convergence

2-dimensional input spaces!

SOM equations

- Choice of the winner
  \[ y_k = \max_j \{ f(w_j^T x) \} \]

- Adaptation of weights
  \[ w_j(t+1) = \begin{cases} w_j(t) + \alpha(t)(x - w_j(t)) & \text{if } d(k, i) < r(t) \\ w_j(t) & \text{otherwise} \end{cases} \]

- \( r(t) \) is made decreasing with time
SOM used as data analysis tool

- Mapping (projection) of a continuous distribution to a discrete set (the centroids)

- After learning: nearest neighbour rule in the input space

Macroeconomical data (1/2)

- Factors: annual increase (%), infant mortality (%), illiteracy ratio (%), school attendance (%), GIP, annual GIP increase (%)

From “Data analysis: How to compare Kohonen neural networks to other techniques?”, F. Blayo, P. DAMETTINES, in IWANN'91 (Granada, Spain) proceedings, Springer-Verlag Lecture Notes in Computer Sciences 540, pp. 469-476.
Macroeconomical data (2/2)

PCA

Kohonen

Content

- Artificial neural networks
  - Introduction
  - Multi-Layer Perceptron
  - Radial-Basis Function Networks
  - Self-Organizing Maps
- About the choice of parameters
  - # of units or parameters
  - # of inputs
- Two application examples in finance
  - Time series forecasting
  - Classification of investment funds
# of units or parameters in ANN

- # units: illustration of bias-variance dilemma
- # units increases: better learning, overfitting ↑
- # units decreases: poorest learning, overfitting ↓

- A posteriori test with ≠ numbers of units/parameters!

- Aim: optimal generalization error

\[ E_{\text{gen}}(\theta) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} (g(x_t, \theta) - y_t)^2 \]

Estimation of generalization error

- Estimates of generalization error:
  - (k-fold) cross-validation
  - leave-one-out
  - bootstrap

- General principle: use different samples to
  - learn
  - validate (compare and select models)
  - test (assess the performances)
Content

- Artificial neural networks
  - Introduction
  - Multi-Layer Perceptron
  - Radial-Basis Function Networks
  - Self-Organizing Maps
- About the choice of parameters
  - # of units or parameters
  - # of inputs
- Two application examples in finance
  - Time series forecasting
  - Classification of investment funds

# of inputs in ANN

- Learning in high-dimensional spaces:
  
  \[ \text{number of samples} \approx e^{\text{space dimension}} \]

  \( \rightarrow \) necessity to reduce the input space dimension!

- How?
  - Selection of input variables
  - Test and errors (comparison of models)
  - Projection of input variables
    - Principal Component Analysis
    - Curvilinear Component Analysis

- How much?
  - A posteriori measures (idem model selection)
  - Specific methods (ex: time series)
Input variables selection and projection

- Starting with many input variables, then reduce their number
- Two options:
  1. selection of input variables
     - interpretability
     - limited to existing variables
  2. projection of input variables
     - linear: PCA
     - non-linear: CCA, Kohonen, etc.
     - forecasting: based on Taken's theorem

Content

- Artificial neural networks
  - Introduction
  - Multi-Layer Perceptron
  - Radial-Basis Function Networks
  - Self-Organizing Maps
- About the choice of parameters
  - # of units or parameters
  - # of inputs
- Two application examples in finance
  - Time series forecasting
  - Classification of investment funds
Application of dimension reduction to forecasting tasks

Regressor: - past values $x(t-i)$
- exogenous data $in(j)$

Forecasting

$$x(t+1) = f(x(t), x(t-1), \ldots, x(t-k), in(1), in(2), \ldots, in(l))$$

Non-linear forecasting:
- 1. optimise regressor on linear predictor
- 2. use the same regressor with non-linear predictor $f$
- trials and errors (computational load !)
- (non-linear) projection of regressor variables

Forecasting: Taken’s theorem 1/2

Takens’ theorem:

$$q \leq \text{size of regressor} \leq 2q+1$$

(AR model)
Forecasting: Taken’s theorem 2/2

\( q \leq \text{size of regressor} \leq 2q+1 \)

\( p \) in the \( 2q+1 \) space, there exists a \( q \)-surface without intersection points

\( p \) Projection from \( 2q+1 \) to \( q \) possible!

Forecasting: 1st example 1/2

\( p \) Artificial series

\[ x(t+1) = ax(t)^2 + bx(t-2) + \epsilon(t) \]

Two past values!

\( p \) Linear AR model

\[ \text{Sum of errors (on 1000 samples)} \]

\[ \text{AR order} \]
Forecasting: 1st example 2/2

- Non-linear AR model
  - initial regressor: size=6
  - intrinsic dimension: 2
  - CCA from dim=6 to dim=2
  - MLP on 2-dim data

```
<table>
<thead>
<tr>
<th>AR order</th>
<th>Sum of errors (on 1000 samples)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>
```

Forecasting: 2nd example 1/2

- Daily returns of BEL20 index
- 42 indicators from inputs and exogenous variables:
  - returns: $x_t$, $x_{t-1}$, $x_{t-2}$, $x_{t-40}$, ..., $y_t$, $y_{t-1}$, ...
  - differences of returns: $x_t-x_{t-5}$, $x_{t-5}-x_{t-10}$, ..., $y_t-y_{t-5}$
  - oscillators: $K(20)$, $K(40)$, ...
  - moving averages: $MM(10)$, $MM(50)$, ...
  - exponential moving averages: $MME(10)$, $MME(50)$, ...
  - etc
Forecasting: 2nd example 2/2

Method:
- 42 indicators
- PCA → 25 variables
- Grassberger-Proccacia: intrinsic dimension = 9
- CCA → 9 variables
- RBF → forecasting

Result: % of correct approximations of sign (90-days average)

In average: 57.2% on test set

Content

- Artificial neural networks
  - Introduction
  - Multi-Layer Perceptron
  - Radial-Basis Function Networks
  - Self-Organizing Maps
- About the choice of parameters
  - # of units or parameters
  - # of inputs
- Two application examples in finance
  - Time series forecasting
  - Classification of investment funds
Why This Study?

Announced Strategy ≠ Implemented Strategy
Why This Study?

Announced Strategy ≠ Implemented Strategy

Goal of managers:
To be compared to other ones with weaker performances

44 millions of families in USA

Not able to estimate the real risk
Classifications exist
They are not very good

Extraction of Features
Linear Regression

\[ R_i(t) = b_{i1} F_1(t) + b_{i2} F_2(t) + \cdots + b_{in} F_n(t) + e_i(t) \]

Return of a fund  \quad \text{Index Return}

\( b_i \) determined using a Least-Square method

\[ \sum_{j} b_{ij} = 1 \]

\( b_{ij} \) percentages of investment

\( b_{ij} \geq 0, \forall i, j \)

Extraction of Features
Linear Regression under constraints
Extraction of Features
Multi-colinearity problem and PCA

\[ R_i(t) = b_{1i}F_1(t) + b_{2i}F_2(t) + \cdots + b_{ni}F_n(t) + e_i(t) \]

Not equal, but highly correlated
\( b_i \) are difficult to determine

---

Extraction of Features
Multi-colinearity problem and PCA

\[ R_i(t) = b_{1i}F_1(t) + b_{2i}F_2(t) + \cdots + b_{ni}F_n(t) + e_i(t) \]

\[ F_1, F_2, \ldots, F_n \xrightarrow{\text{PCA}} G_1, G_2, \ldots, G_m \]

\[ R_i(t) = c_{1i}G_1(t) + c_{2i}G_2(t) + \cdots + c_{mi}G_m(t) + e_i(t) \]

\[ G_1, G_2, \ldots, G_m \xrightarrow{\text{Inverse of PCA}} F_1, F_2, \ldots, F_n \]

\[ R_i(t) = b'_{1i}F_1(t) + b'_{2i}F_2(t) + \cdots + b'_{ni}F_n(t) + e_i(t) \]
Classification of investment funds

- Extraction of features
  - Linear Regression (under constraints)
  - Multi-collinearity problem and PCA

- Classification
  - Kohonen Maps
  - Ward algorithm

- Application
  - CRSP database (the Chicago University)
  - Comparison with classification from the ICDI and S&P's Fund Services.

CRSP database
(the Chicago University)

<table>
<thead>
<tr>
<th>Indexes</th>
<th></th>
<th>5822 funds</th>
<th>33 indexes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1    Dow Jones 30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2    Lehman Brothers’ US Credit Bond Index</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8    Salomon Brothers’ Non-US Government Bond Index</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9    S&amp;P400 Medium Capitalization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14   FTSE100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17   FTSE Small Capitalization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18   UK Bank Bills 3 month</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21   TOPIX100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24   Japan Benchmark 2 year Government Index</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26   CAC40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29   France Benchmark 2 year Government</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32   Germany Money Market 3 month</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### CRSP database (the Chicago University)

<table>
<thead>
<tr>
<th>Indexes</th>
<th>$b'_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dow Jones 30</td>
<td>0.1156</td>
</tr>
<tr>
<td>Lehman Brothers' US Credit Bond Index</td>
<td>0.0322</td>
</tr>
<tr>
<td>Salomon Brothers' Non-US Government Bond Index</td>
<td>0.0055</td>
</tr>
<tr>
<td>S&amp;P400 Medium Capitalation</td>
<td>0.1042</td>
</tr>
<tr>
<td>FTSE100</td>
<td>0.6090</td>
</tr>
<tr>
<td>FTSE Small Capitalization</td>
<td>0.0044</td>
</tr>
<tr>
<td>UK Bank Bills 3 month</td>
<td>0.0336</td>
</tr>
<tr>
<td>TOPIX100</td>
<td>0.0326</td>
</tr>
<tr>
<td>Japan Benchmark 2 year Government Index</td>
<td>-0.0048</td>
</tr>
<tr>
<td>CAC40</td>
<td>-0.0057</td>
</tr>
<tr>
<td>France Benchmark 2 year Government</td>
<td>-0.0012</td>
</tr>
<tr>
<td>Germany Money Market 3 month</td>
<td>0.0016</td>
</tr>
</tbody>
</table>

---

![Graph of CRSP database](image)
CRSP database
Kohonen

Classification
Ward Algorithm
Intra-class inertia

\[ I_w = \frac{1}{N} \sum_{l=1}^{P} N_l I_l \]

Merge the most similar centroids

CRSP database
Ward: 20 classes
Comparison to classification from the ICDI and S&P’s Fund Services.

<table>
<thead>
<tr>
<th>Strategies</th>
<th>Government Securities</th>
<th>Taxable Money Market</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggressive Growth</td>
<td>Gov Securities</td>
<td></td>
</tr>
<tr>
<td>Balanced</td>
<td>International Equities</td>
<td>High Quality Municipal Bonds</td>
</tr>
<tr>
<td>High Quality Bonds</td>
<td>Income</td>
<td>Option Income</td>
</tr>
<tr>
<td>High Yield Bonds</td>
<td>Long-Term Growth</td>
<td>Precious Metals</td>
</tr>
<tr>
<td>Global Bonds</td>
<td>Tax-Free Money Market</td>
<td>Sector Funds</td>
</tr>
<tr>
<td>Global Equity</td>
<td>Gov Securities Money Market</td>
<td>Special Funds</td>
</tr>
<tr>
<td>Growth &amp; Income</td>
<td>High Quality Municipal Bonds</td>
<td>Total Return</td>
</tr>
<tr>
<td>Ginnie Mae Funds</td>
<td>Single-State Municipal Bonds</td>
<td>Utility Funds</td>
</tr>
</tbody>
</table>

Comparison to classification from the ICDI and S&P’s Fund Services.

<table>
<thead>
<tr>
<th>Strategies</th>
<th>Government Securities</th>
<th>Taxable Money Market</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggressive Growth</td>
<td>Gov Securities</td>
<td></td>
</tr>
<tr>
<td>Balanced</td>
<td>International Equities</td>
<td>High Quality Municipal Bonds</td>
</tr>
<tr>
<td>High Quality Bonds</td>
<td>Income</td>
<td>Option Income</td>
</tr>
<tr>
<td>High Yield Bonds</td>
<td>Long-Term Growth</td>
<td>Precious Metals</td>
</tr>
<tr>
<td>Global Bonds</td>
<td>Tax-Free Money Market</td>
<td>Sector Funds</td>
</tr>
<tr>
<td>Global Equity</td>
<td>Gov Securities Money Market</td>
<td>Special Funds</td>
</tr>
<tr>
<td>Growth &amp; Income</td>
<td>High Quality Municipal Bonds</td>
<td>Total Return</td>
</tr>
<tr>
<td>Ginnie Mae Funds</td>
<td>Single-State Municipal Bonds</td>
<td>Utility Funds</td>
</tr>
</tbody>
</table>
Comparison to classification from the ICDI and S&P’s Fund Services.

SINGLE-STATE MUNICIPAL BOND FUNDS

- 2% (C16)
- 4% (C4)
- 12% (C8)
- 16% (C1)
- 15% (C17)
- 8% (C15)
- 6% (C12)
- 5% (C13)
- 9% (C1)
- 66% (C4)

LONG-TERM GROWTH FUNDS

- 28% (C14)
- 9% (C13)
- 8% (C15)
- 12% (C7)
- 13% (C1)
- 16% (C3)
- 4% (C5)
- 2% (C6)
- 2% (C18)
- Others

Comparison to classification from the ICDI and S&P’s Fund Services.
Analysis

Intra-class inertia for classifications:

- 0.07 for the Kohonen/Ward classification
- 0.13 for the reference classification

Why some differences between the two classifications?

- classification from ICDI based on information given by the managers?
- reference classification not so sophisticated?
- strategy not constant?

Conclusion

ANN are regression/classification tools:

- powerful in theory
- powerful in practice
- but application needs caution and expertise:
  - choosing # of parameters
  - choosing # and which inputs
  - learning procedure (local minima, etc.)
  - validation!
  - comparison with more classical tools
  - etc.