GMDH Applications in QSAR and QSAR ensembling

Giuseppina Gini

Department of Electronics and Information,
Politecnico di Milano, Italy

What is QSAR?

- A QSAR is a **mathematical relationship** (model) between a biological activity of a molecular system and its geometric and chemical characteristics.
- The problem of QSAR is to find coefficients $C_0, C_1, \ldots C_n$ such that the obtained linear (or non linear) equation:

$$\textbf{\textit{Biological activity} = } C_0 + (C_1 * P_1) + \ldots + (C_n * P_n)$$

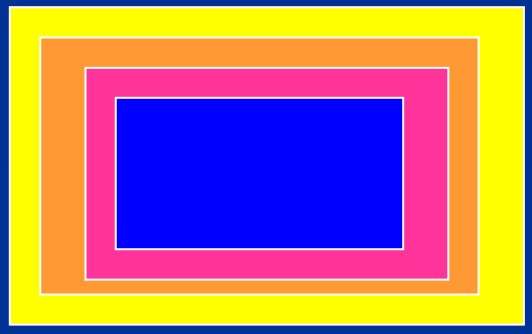
minimizes the prediction error.

Also QSAR attempts to find consistent relationship between biological activity and molecular properties, so that these "rules" can be used to evaluate the activity of new compounds.

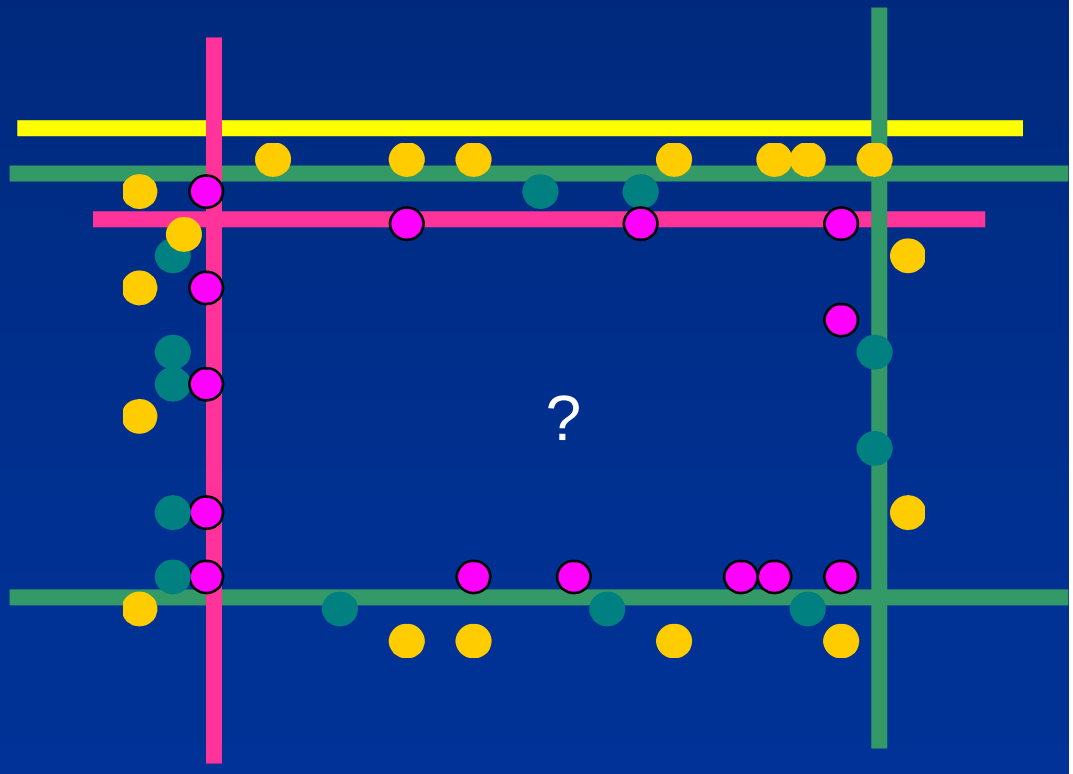
What is a model?

Inducing a model

- Given the illustrated points, what is the model?



?



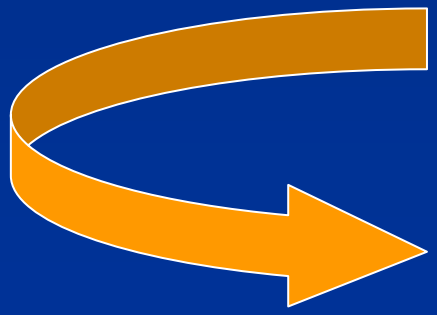
Trends

- from Statistics to Intelligent Data Analysis to Machine Learning
  - move from basic statistics to more complex models
    - Lack of adaptation
    - Noisy data
    - Computational problem Trend from exact to approximate solutions

Exact solution

optimization

NP hard



Approximate solution

“Soft computing”

Modelling: scientific communities

- statistics
- machine learning
- pattern recognition
- neural networks
- computational learning theory
- knowledge discovery and data mining
- …

The equivalence of algorithms

- Artificial neural networks
- Evolutionary Algorithms
- Bayesian networks
- Decision trees
- …

More interpretability?

EQUIVALENCE - if algorithm A outperforms algorithm B on some learning task, then there must exist exactly as many other tasks where B outperforms A.

Techniques in QSAR (ISI 1990-2005)

Query in text	**Number of items**
QSAR or predictive toxicology	4835
QSAR (in text)	4738
QSAR (in title)	1860
Technique used	
QSAR and regression	950
QSAR and NN	237
QSAR and classification	223
QSAR and PCA	104
QSAR and GA	95
QSAR and ML	20
QSAR and ensemble	19
QSAR and SVM	17
QSAR and Radial	14
QSAR and fuzzy	10
QSAR and Bayesian	4

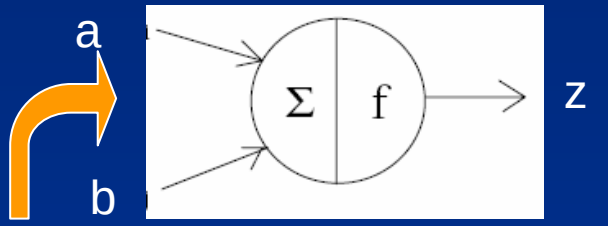
Proposal of the modelling method: GMDH

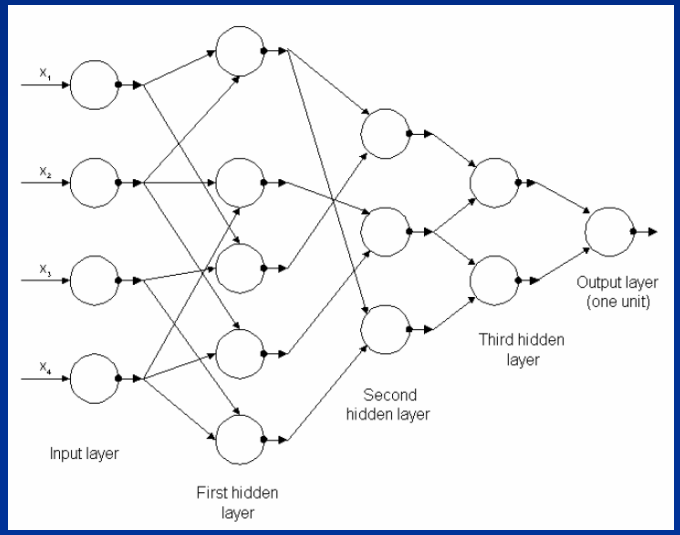
- *GMDH* (*Group Method of Data Handling,* A.G. Ivakhnenko, 1967)
- Proposed both for the QSAR activity and for the integration of different models (ensembling)
- Is auto-organizing
- Uses and external criterion
- Avoids overfitting of data
- Does not require expertise in ANN construction

Description (1)

1. automatically creates the network structure
2. neurons are 2-levels structures; nets are multi-layered
3. learning is contextual to net construction; **w** are found by solving Linear Regression equations with **z** = **y**
4. the structure is designed by gradually increasing complexity
5. it avoids overfitting through a regularity criterion applied to a validation subset of the training set
6. pruning criteria use the results on the validation subset

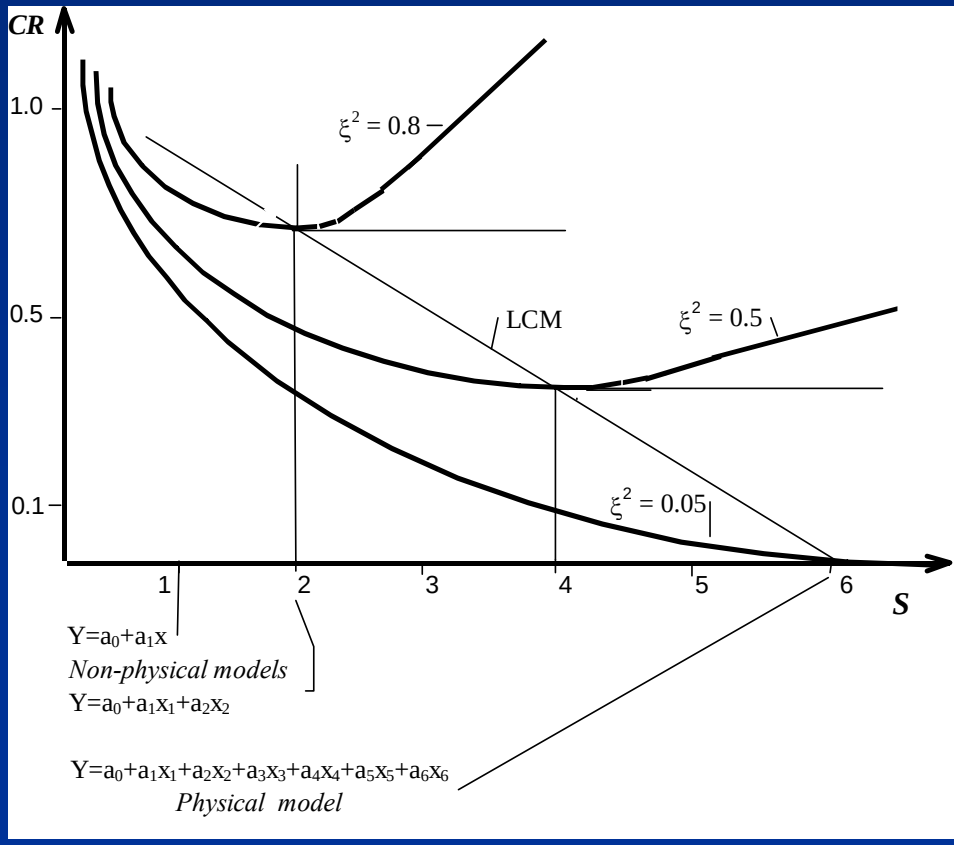
$$z = w_0 + w_1 a + w_2 b + w_3 a^2 + w_4 b^2 + w_5 ab$$



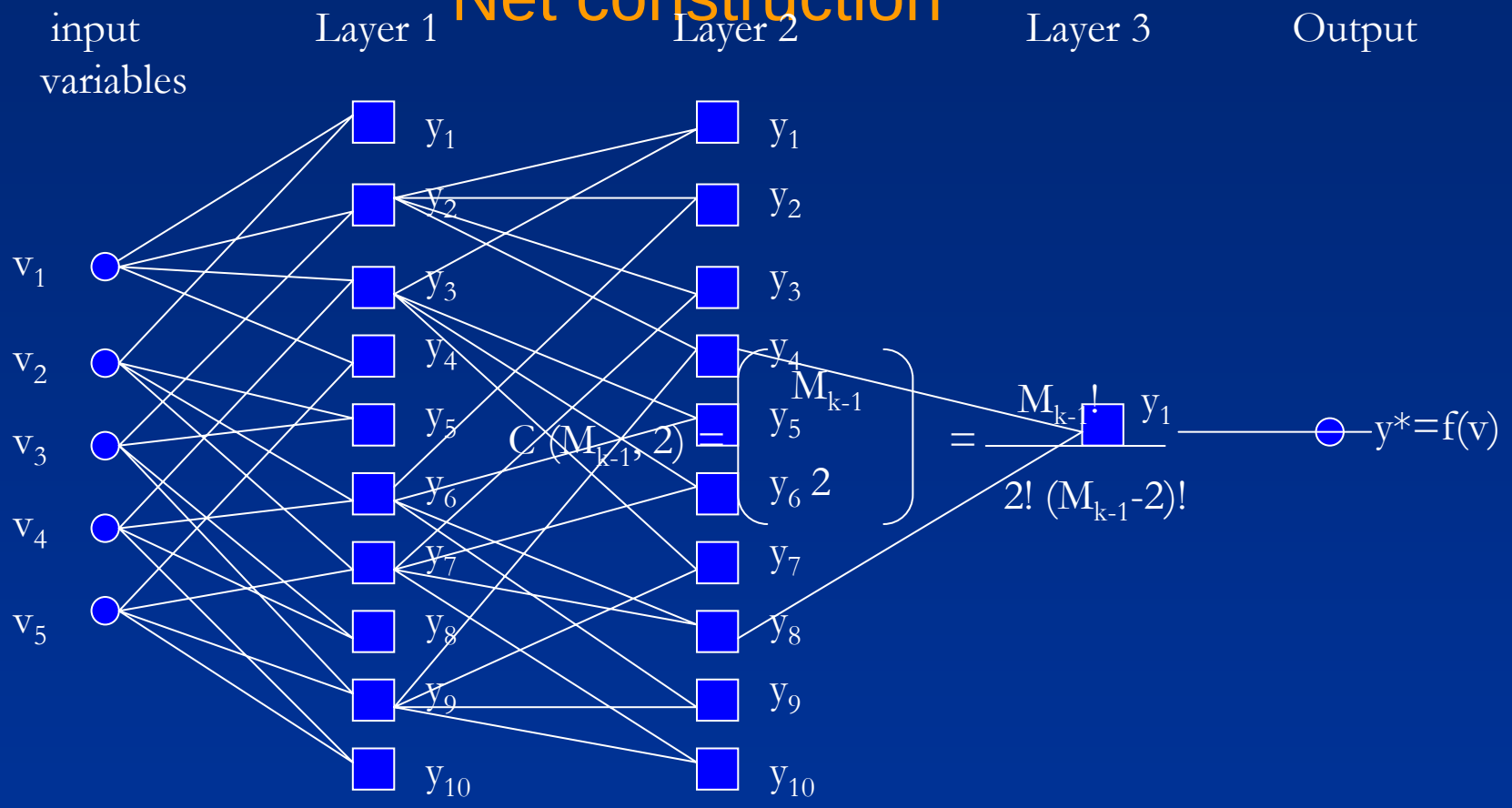


Description (2)

- Inductive algorithms select the optimal non physical model
- We see the mimimum values of the external criterion (LCM) versus the complexity of the structure S of the model for different values of noise variance ξ^2.



Net construction



…our algorithm: poliGMDH

- Multilayer Algorithm
- Ivakhnenko polynomial

$$z^k = a(z_i^{k-1})^2 + b(z_i^{k-1})(z_j^{k-1}) + c(z_j^{k-1})^2 + d(z_i^{k-1}) + e(z_j^{k-1}) + f$$

- In each layer C(n_(k-1),2) neurons
- Selection of good neurons
  1. m.s.e. value on a selection set
  2. Neurons are ordered
  3. The last are eliminated

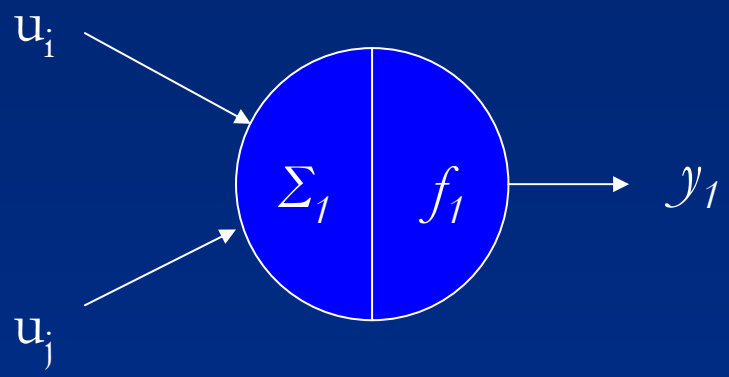
$$C(n\_(k-1),2) = \binom{n_{k-1}}{2}$$

Polynomials

- find the polynomial of optimal complexity
- high order polynoms cause overfitting.
- second order polynoms are preferred

$p_1(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2$
$p_2(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$
$p_3(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2$
$p_4(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_1 x_2 + w_3 x_1^2$
$p_5(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_1 x_2$
$p_6(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2$
$p_7(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_1^2 + w_3 x_2^2$
$p_8(\mathbf{x}) = w_0 + w_1 x_1^2 + w_2 x_2^2$
$p_9(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2$
$p_{10}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2$
$p_{11}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_1 x_2 + w_3 x_1^2 + w_4 x_2^2$
$p_{12}(\mathbf{x}) = w_0 + w_1 x_1 x_2 + w_2 x_1^2 + w_3 x_2^2$
$p_{13}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_1 x_2 + w_3 x_2^2$
$p_{14}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2^2$
$p_{15}(\mathbf{x}) = w_0 + w_1 x_1 x_2$
$p_{16}(\mathbf{x}) = w_0 + w_1 x_1 x_2 + w_2 x_1^2$

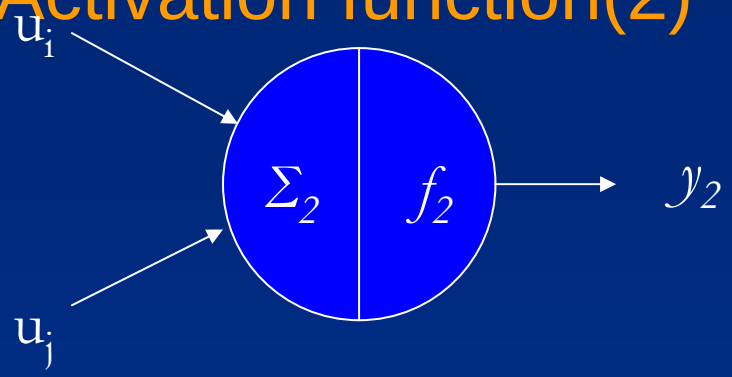
Activation function (1)



- Ivakhnenko:
- Σ_1: *(non-linear)*
- $z_1 = A*x_1*x_2 + B*x_1^2 + C*x_2^2 + D*x_1 + E*x_2 + F$

- f_1: *(linear)*
- $y_1 = z_1$

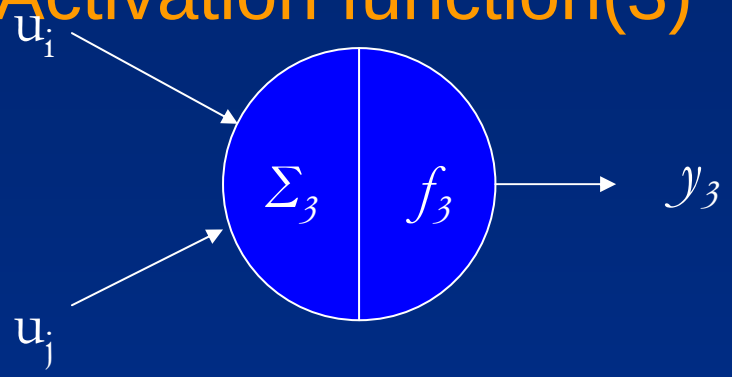
Activation function(2)



- bi-quadratic:
- Σ_2: *(non-linear)*
- $z_2 = A*x_1*x_2 + B*x_1 + C*x_2 + D$

- f_2: *(linear)*
- $y_2 = z_2$

Activation function(3)



- Exponential
- Σ_3: *(non-linear)*
- $z_3 = y_2 = z_2 = A*x_1*x_2 + B*x_1 + C*x_2 + D$

- f_3: *(non-linear)*
- $y_3 = exp(-z_3^2) = exp(-y_2^2)$

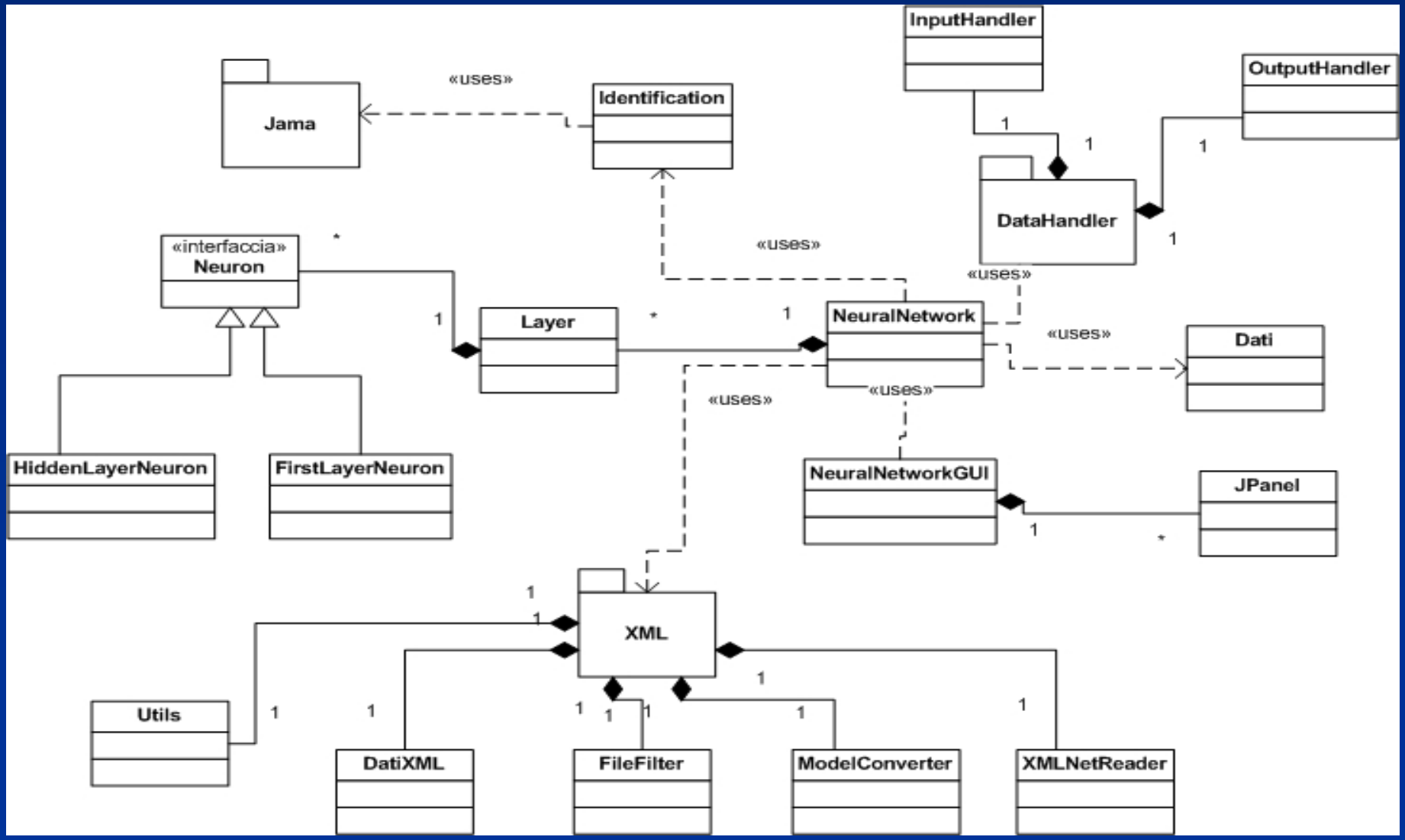
Transfer functions

- 1) Ivakhnenko:
- Σ_1: *(non-linear)*
- $z_1 = A*x_1*x_2 + B*x_1^2 + C*x_2^2 + D*x_1 + E*x_2 + F$
- f_1: *(linear)* $y_1 = z_1$

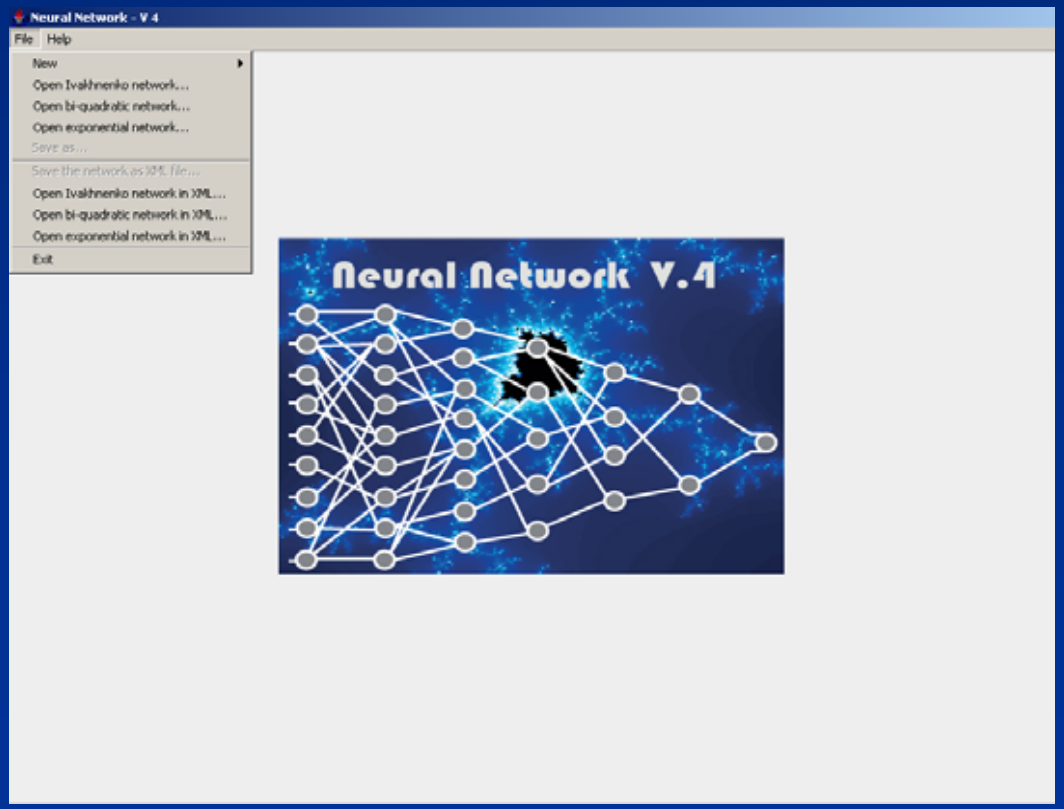
- 2) bi-quadratic:
- Σ_2: *(non-linear)*
- $z_2 = A*x_1*x_2 + B*x_1 + C*x_2 + D$
- f_2: *(linear)* $y_2 = z_2$

- 3) exponential:
- Σ_3: *(non-linear)*
- $z_3 = y_2 = z_2 = A*x_1*x_2 + B*x_1 + C*x_2 + D$
- f_3: *(non-linear)* $y_3 = exp(-z_3^2) = exp(-y_2^2)$

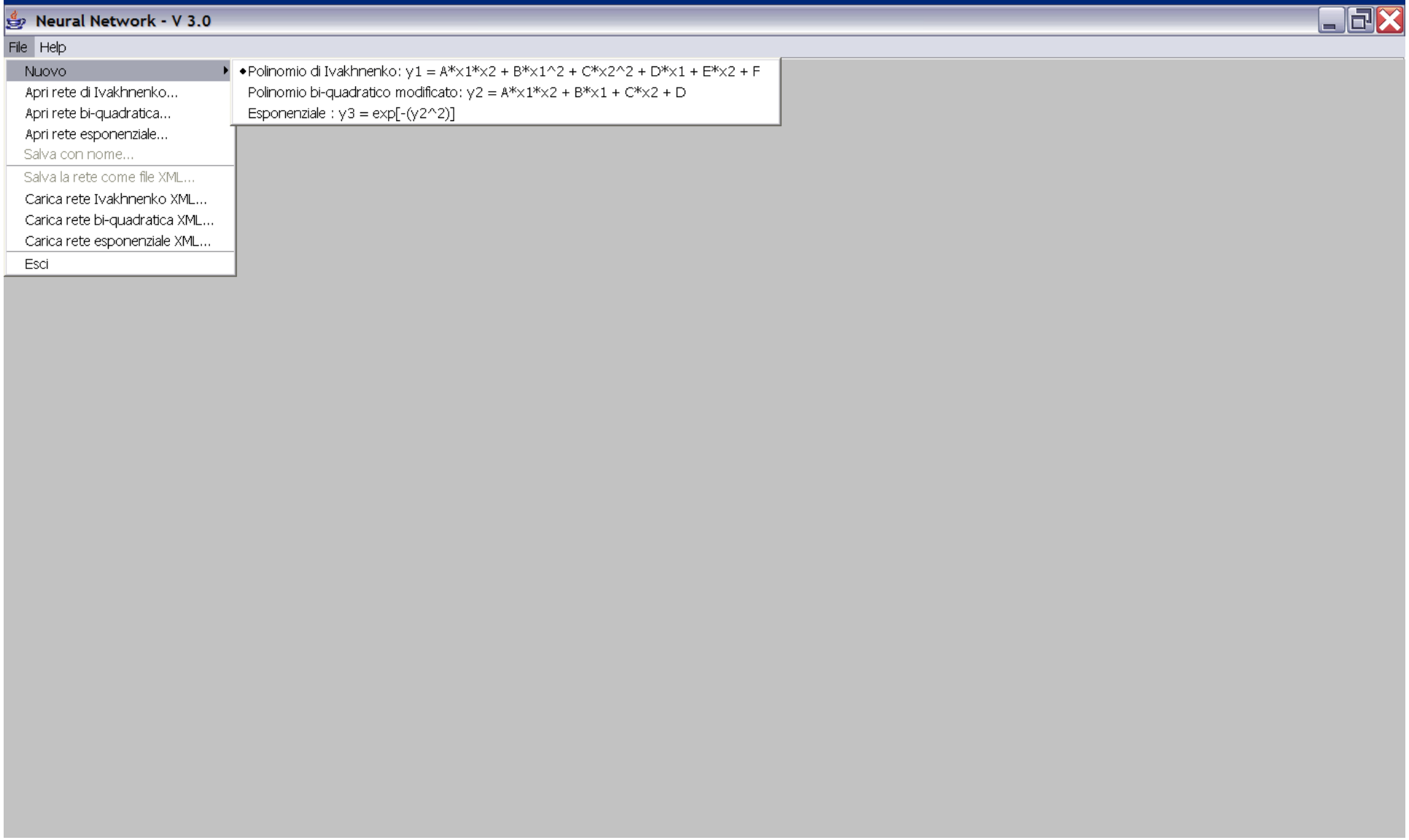
software – Class Diagram



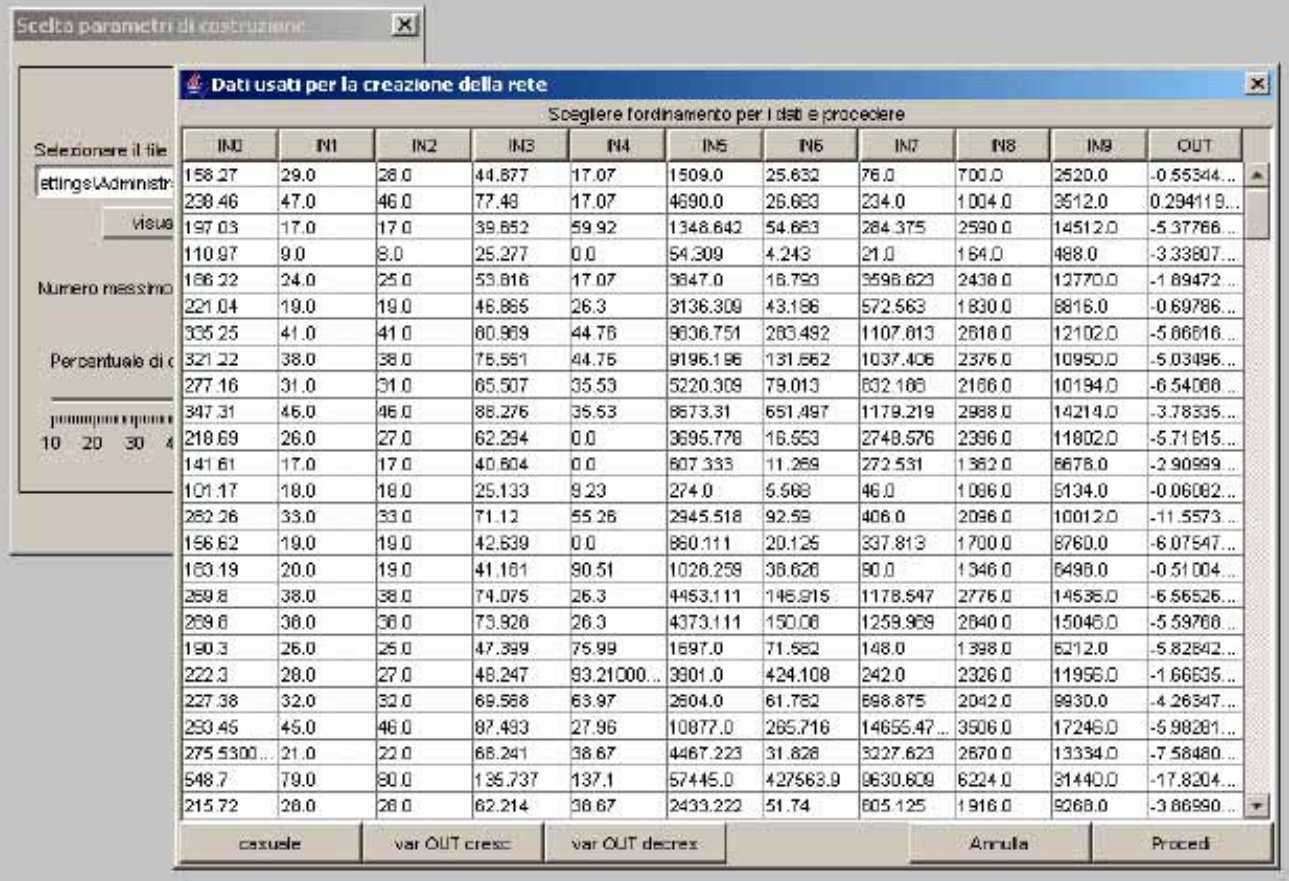
The Neural Network construction in poliGMDH



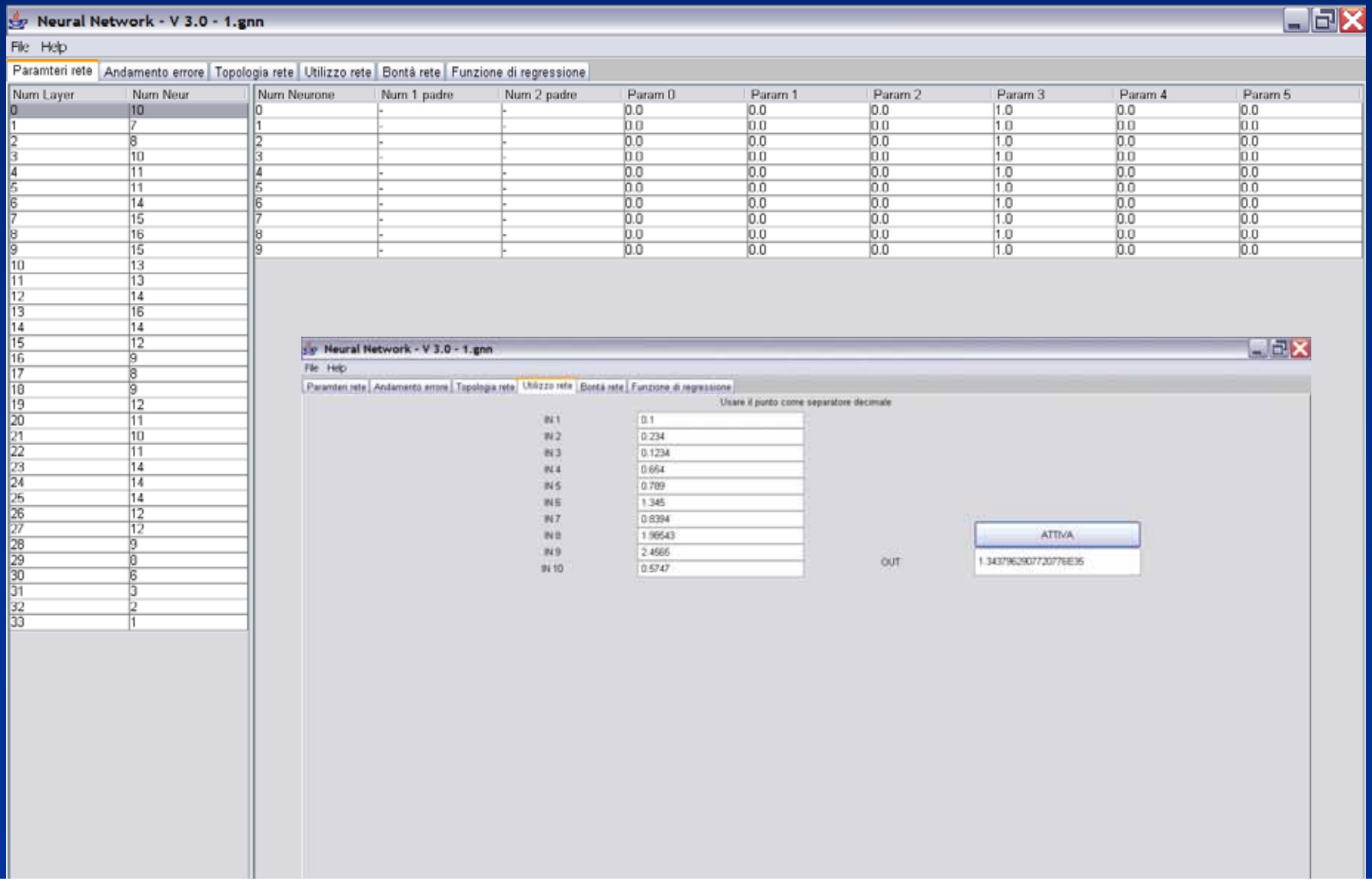
File menu


Neural Network - V 3.0
File
Help
Nuovo
Apri rete di Ivakhnenko...
Apri rete bi-quadratica...
Apri rete esponenziale...
Salva con nome...
Salva la rete come file XML...
Carica rete Ivakhnenko XML...
Carica rete bi-quadratica XML...
Carica rete esponenziale XML...
Esci
Polinomio di Ivakhnenko: y1 = A*x1*x2 + B*x1^2 + C*x2^2 + D*x1 + E*x2 + F
Polinomio bi-quadratico modificato: y2 = A*x1*x2 + B*x1 + C*x2 + D
Esponenziale : y3 = exp[-(y2^2)]

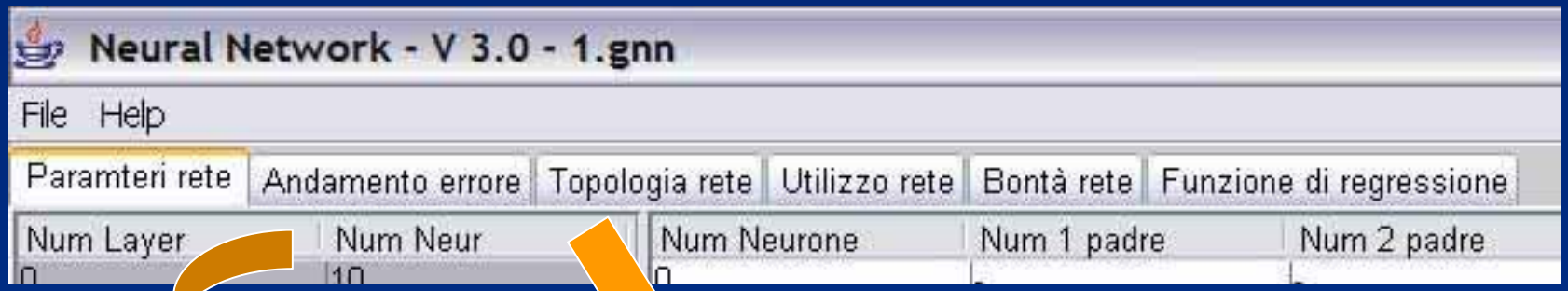
Building the net

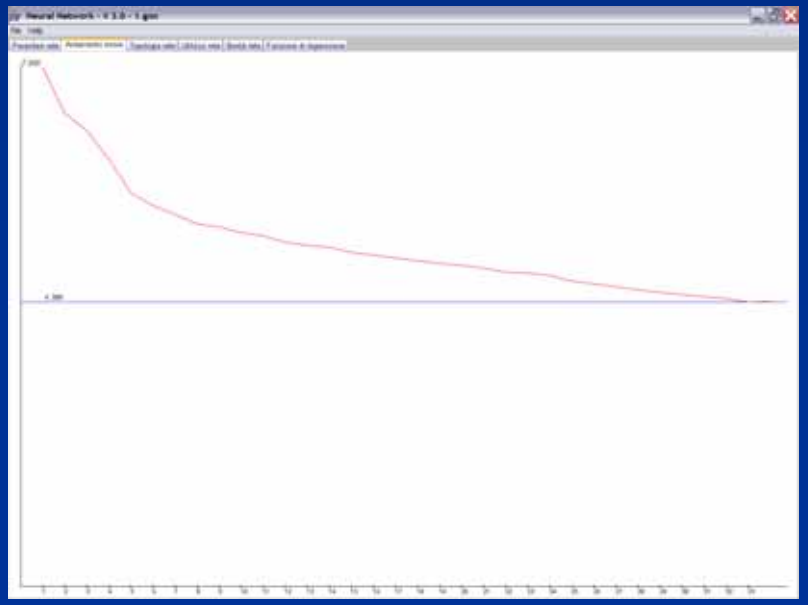


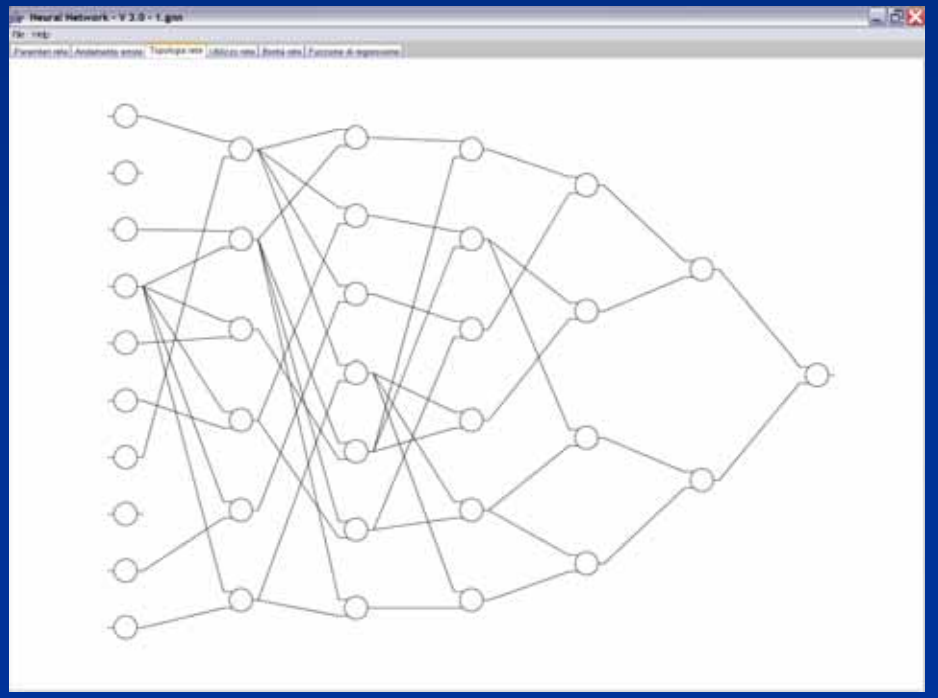
Using the net



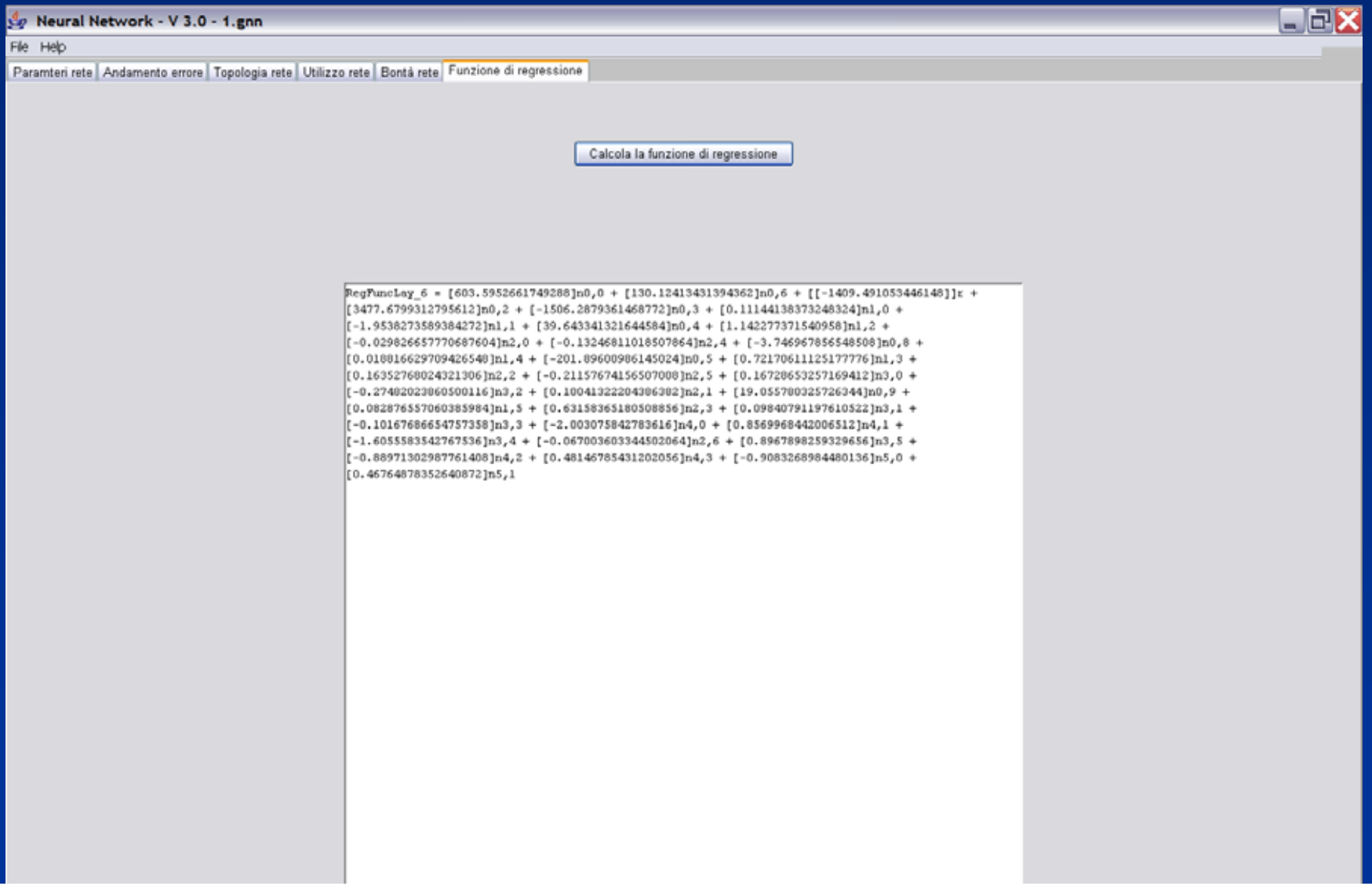
The interface



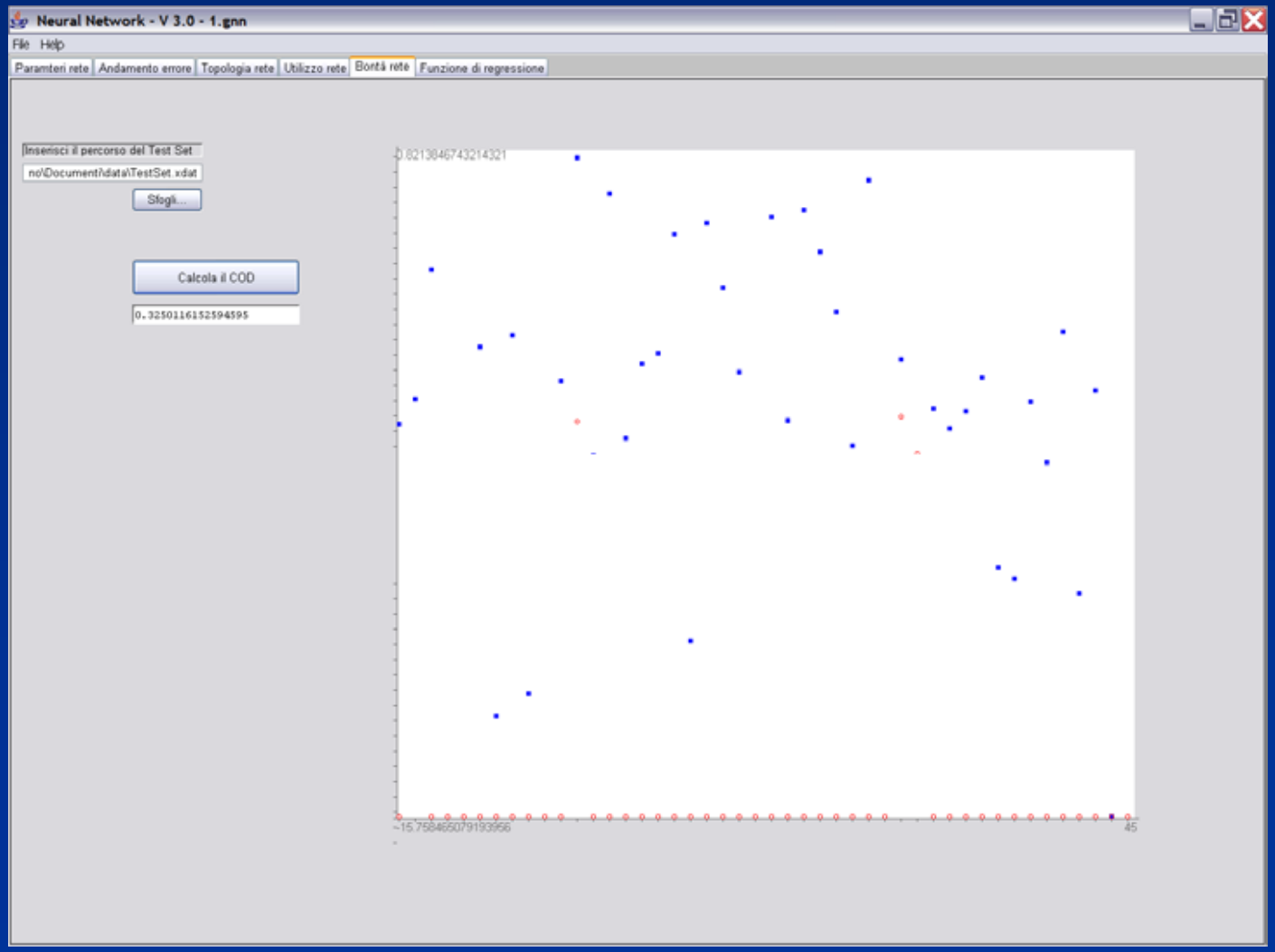




Regression function



Computing evaluation parameters



Model Validation

- Total variance:
- Explained variance:
- Not explained variance:

$$\frac{\sum_{i=1}^{n}(y_i - m_y)^2}{n}$$

$$\frac{\sum_{i=1}^{n}(\hat{y}_i - m_y)^2}{n}$$

$$\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

$$\frac{\sum_{i=1}^{n}(\hat{y}_i - m_y)^2}{\sum_{i=1}^{n}(y_i - m_y)^2}$$

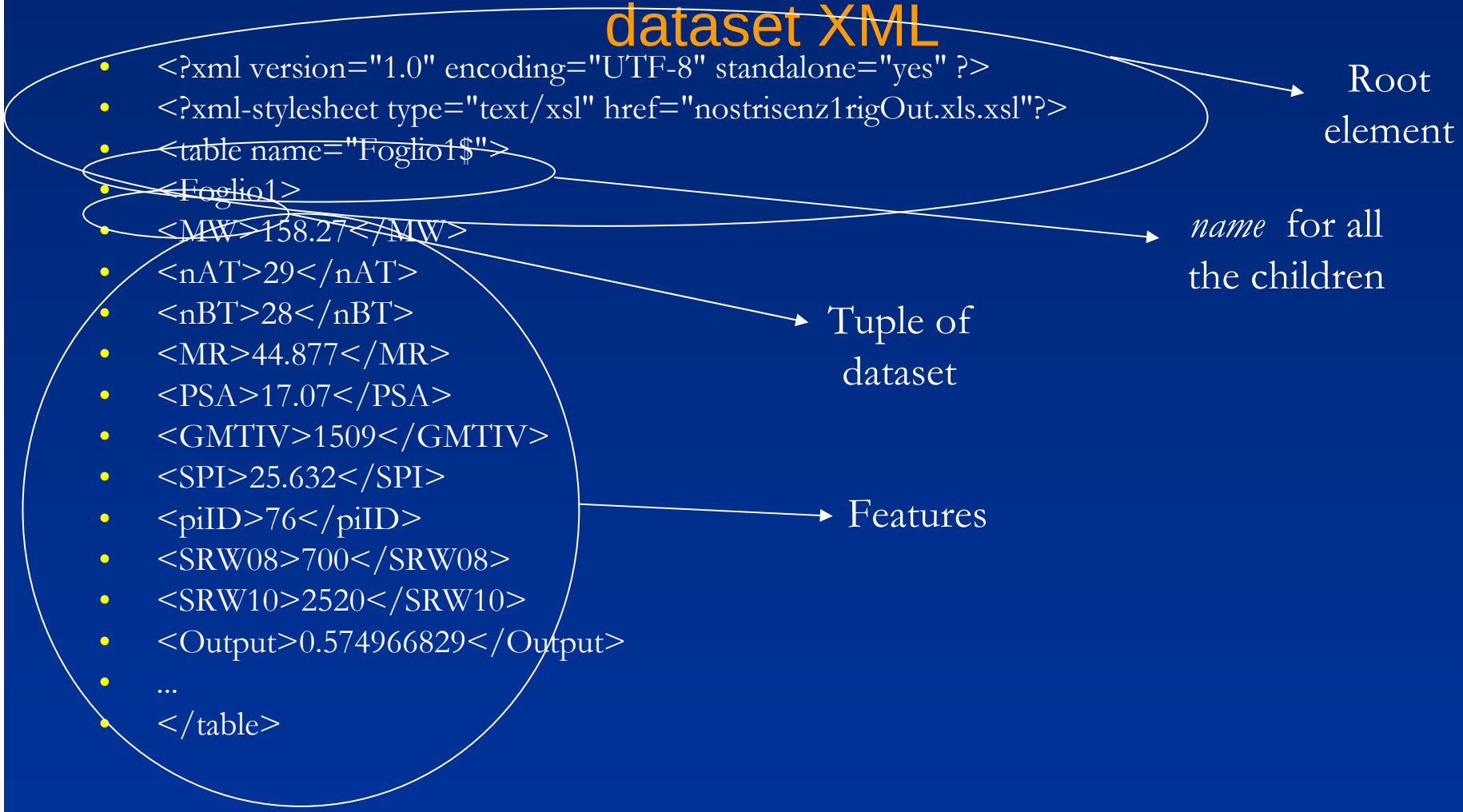
- Determination coefficient COD
- *COD* is $0 \leq r^2 \leq 1$:

Standard

- *XML (Extensible Markup Language).*

- Standard interchange language
- Trained net saved in XML
- Net parsed in the application
- Open to future web services and applications

dataset XML


<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<?xml-stylesheet type="text/xsl" href="nostrisenz1rigOut.xls.xsl"?>
<table name="Foglio1$">
<Foglio1>
<MW>158.27</MW>
<nAT>29</nAT>
<nBT>28</nBT>
<MR>44.877</MR>
<PSA>17.07</PSA>
<GMTIV>1509</GMTIV>
<SPI>25.632</SPI>
<piID>76</piID>
<SRW08>700</SRW08>
<SRW10>2520</SRW10>
<Output>0.574966829</Output>
...
</table>
Root element
name for all the children
Tuple of dataset
Features

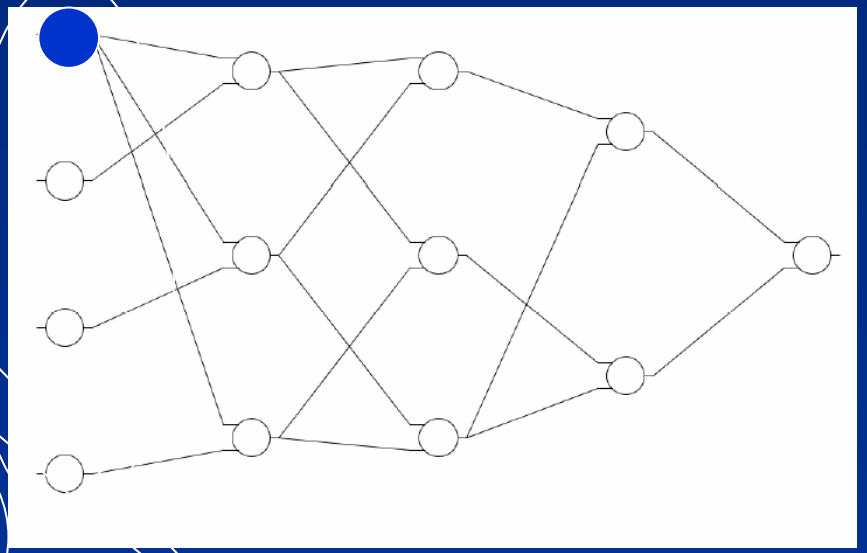
Network in XML

root

Layer

Neuron

```
<?xml version="1.0" encoding="UTF-8" ?>
<NeuralNetwork>
<Layer id="0">
<Neuron id="0">
<CombinationFunction name="FirstLayerNeuron" />
<ActivationFunction name="IvakhnenkoPolynomial"
    num-params="6">
<AFParameter id="1">0.0</AFParameter>
<AFParameter id="2">0.0</AFParameter>
<AFParameter id="3">0.0</AFParameter>
<AFParameter id="4">1.0</AFParameter>
<AFParameter id="5">0.0</AFParameter>
<AFParameter id="6">0.0</AFParameter>
</ActivationFunction>
</Neuron>
```



Sub-element of Neuron

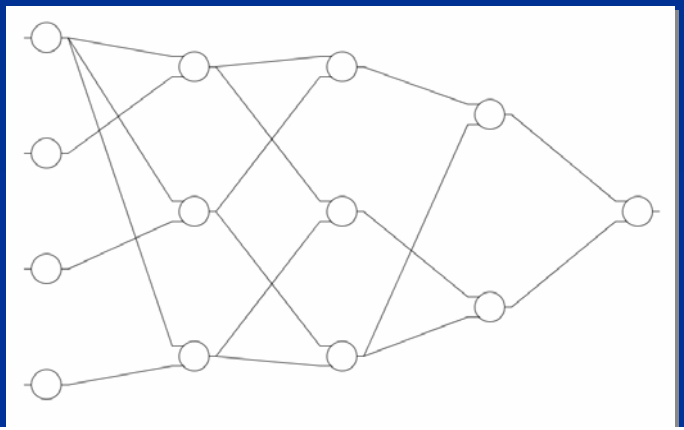
Parametrs of the activation function

Data, equations, models as text (XML)

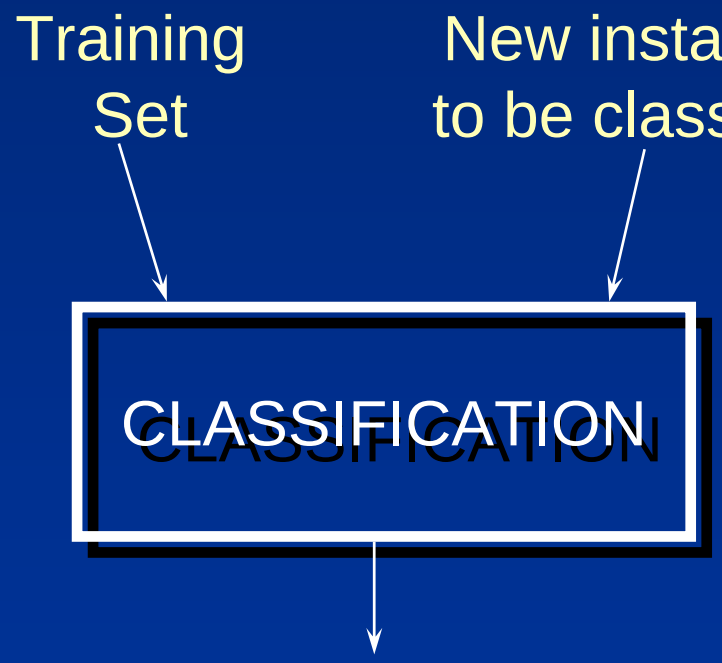
```
<?xml version="1.0" encoding="UTF-8" ?>
<NeuralNetwork>
<Layer id="0">
<Neuron id="0">
<CombinationFunction name="FirstLayerNeuron" />
<ActivationFunction name="IvakhnenkoPolynomial" num-params="6">
<AFParameter id="1">0.0</AFParameter>
<AFParameter id="2">0.0</AFParameter>
<AFParameter id="3">0.0</AFParameter>
<AFParameter id="4">1.0</AFParameter>
<AFParameter id="5">0.0</AFParameter>
<AFParameter id="6">0.0</AFParameter>
</ActivationFunction>
</Neuron>
```

Emphasis on services

```
="2">
tionFunction name="NeuronSelection" num-params="2">
neter id="0">
" type="Nref" lr="0" nrn="0" />
" type="weight">1</Arg>
neter>
neter id="1">
" type="Nref" lr="0" nrn="3" />
" type="weight">1</Arg>
neter>
ationFunction>
nFunction name="IvakhnenkoPolynomial" num-params="6">
neter id="1">0.0895283684330046</AFParameter>
neter id="2">0.27905286981012445</AFParameter>
neter id="3">0.7434228721524345</AFParameter>
<AFParameter id="4">2.610780910353352</AFParameter>
<AFParameter id="5">-1.730970145378251</AFParameter>
<AFParameter id="6">-0.35500543418743646</AFParameter>
</ActivationFunction>
</Neuron>
```



Next step



- one model is built
- More models are similar. Which is the best?
- What if (a few) models are combined?

What is ensemble learning?

Ensemble learning refers to a collection of methods that learn a target function by training a number of individual learners and combining their predictions

Hybrid – a technical definition

- involves the use of two or more intelligent techniques and approaches, (neural networks, knowledge-based methods, fuzzy techniques, genetic algorithms, agent-based techniques, case based reasoning etc).
- The combination can be done in any form, either by a modular integration of two or more intelligent methodologies, or by fusing one methodology into another…

Ensembles: different names

- multiple models
- multiple classifier systems
- combining classifiers (regressors *etc*)
- integration of classifiers
- mixture of experts
- decision committee
- committee of experts
- classifer fusion
- …

ensembles

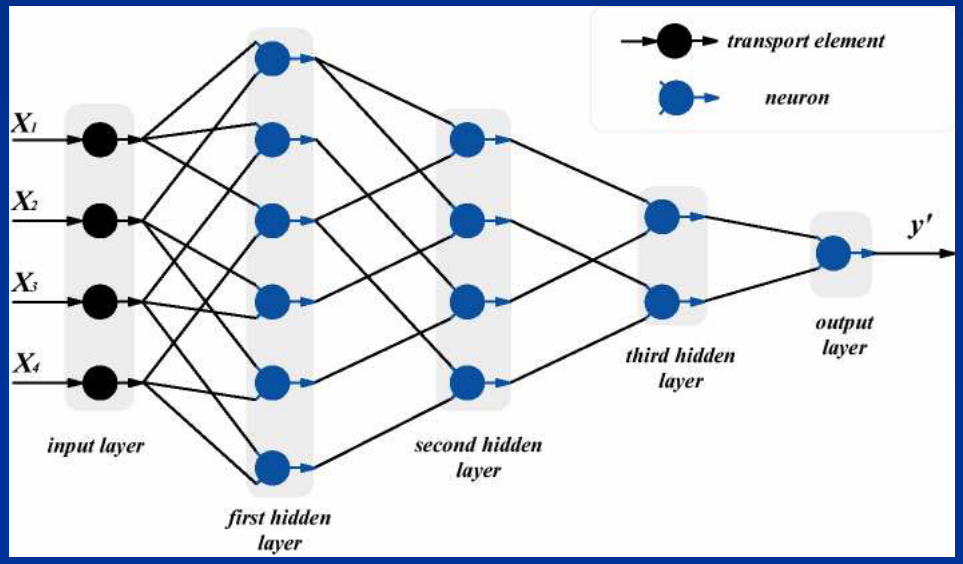
- base classifiers
- component classifiers
- individual classifiers
- members (of a decision committee)
- level 0 experts
- … what else?

classifiers in ensembles

2 views on ensembling

- A method to reduce errors (bias and variance)
- A method to integrate more concepts (hybrid)

if a then b



Why ensemble learning?

- *Accuracy*: a more reliable result by combining the output of multiple ”experts”
- *Efficiency*: a complex problem can be decomposed into multiple sub-problems that are easier to understand and solve

- no single model that works for all pattern recognition problems

•”To solve really hard problems, we’ll have to use several different representations……. It is time to stop arguing over which type of pattern-classification technique is best……. Instead we should work at a higher level of organization and discover how to build managerial systems to exploit the different virtues and evade the different limitations of each of these ways of comparing things.” Minsky, 1991.

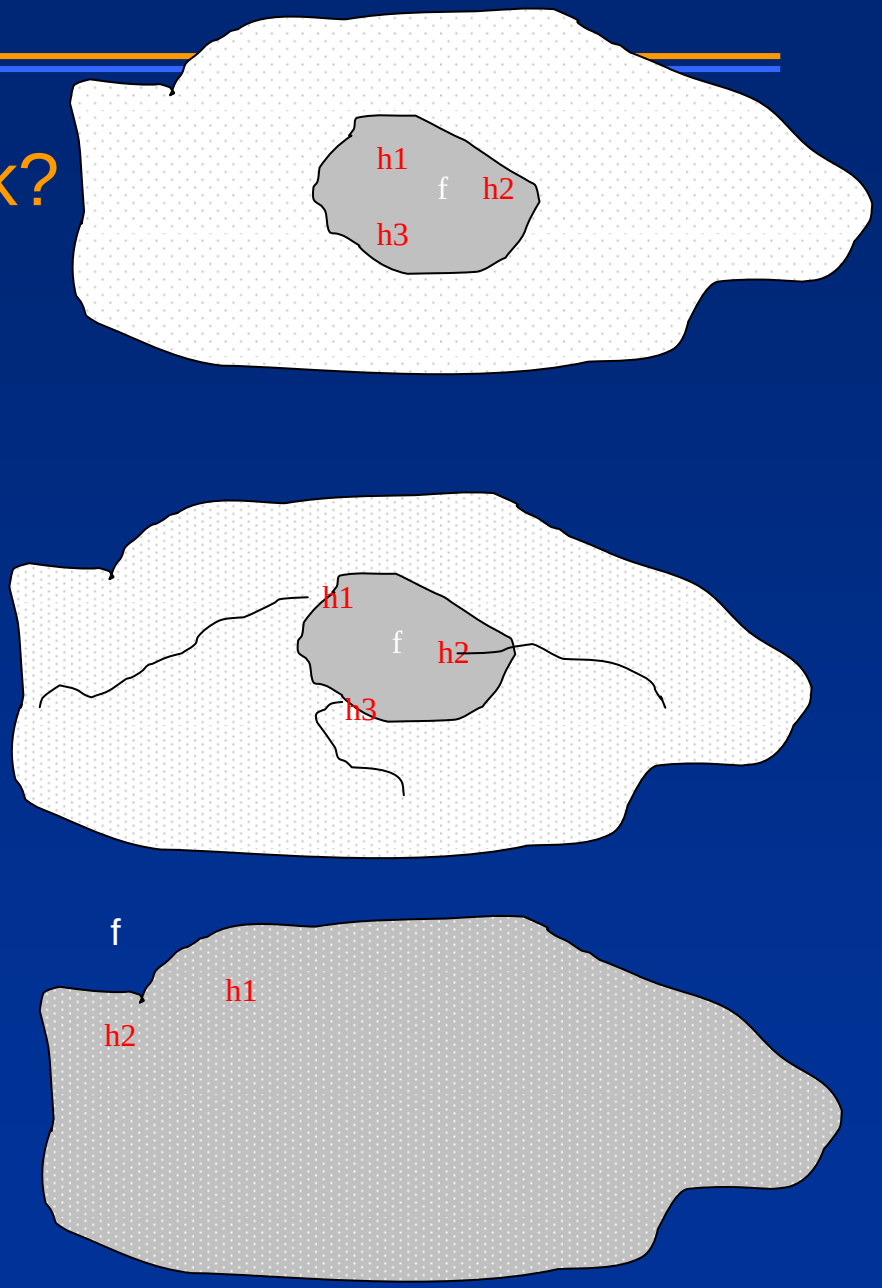
When ensemble learning?

- When you can build base classifiers that are *more accurate than chance*, and
- that are as much as possible *independent* from each other

Why do ensembles work?

ensembles overcome three problems:

- ***Statistical Problem*** : there are many hypotheses with the same accuracy, the learning algorithm chooses one of them – better to mix them.
- ***Computational Problem*****:** the learning algorithm cannot guarantee reaching the best hypothesis.
- ***Representational Problem***: the hypothesis space does not contain any good approximation of the target classes.



Theoretical results

Hansen & Solomon (1990):

If we can assume that classifiers are independent in predictions and their accuracy > 50%, can push accuracy arbitrarily high by combining more classifiers

Key assumption:

classifiers are *independent* in their predictions

How to make an effective ensemble?

Two basic decisions when designing ensembles:

1. How to generate base classifiers? (***generation strategy***)
2. How to integrate them? (***integration strategy***)

Methods for Independently Constructing Ensembles

One way to force a learning algorithm to construct multiple hypotheses is to run the algorithm several times and provide it with somewhat different data in each run.

This idea is used in methods as:

- ***Bagging (instance selection)***
- ***Feature-Selection Ensembles***
- ***Random Forest*** - both sources of randomness

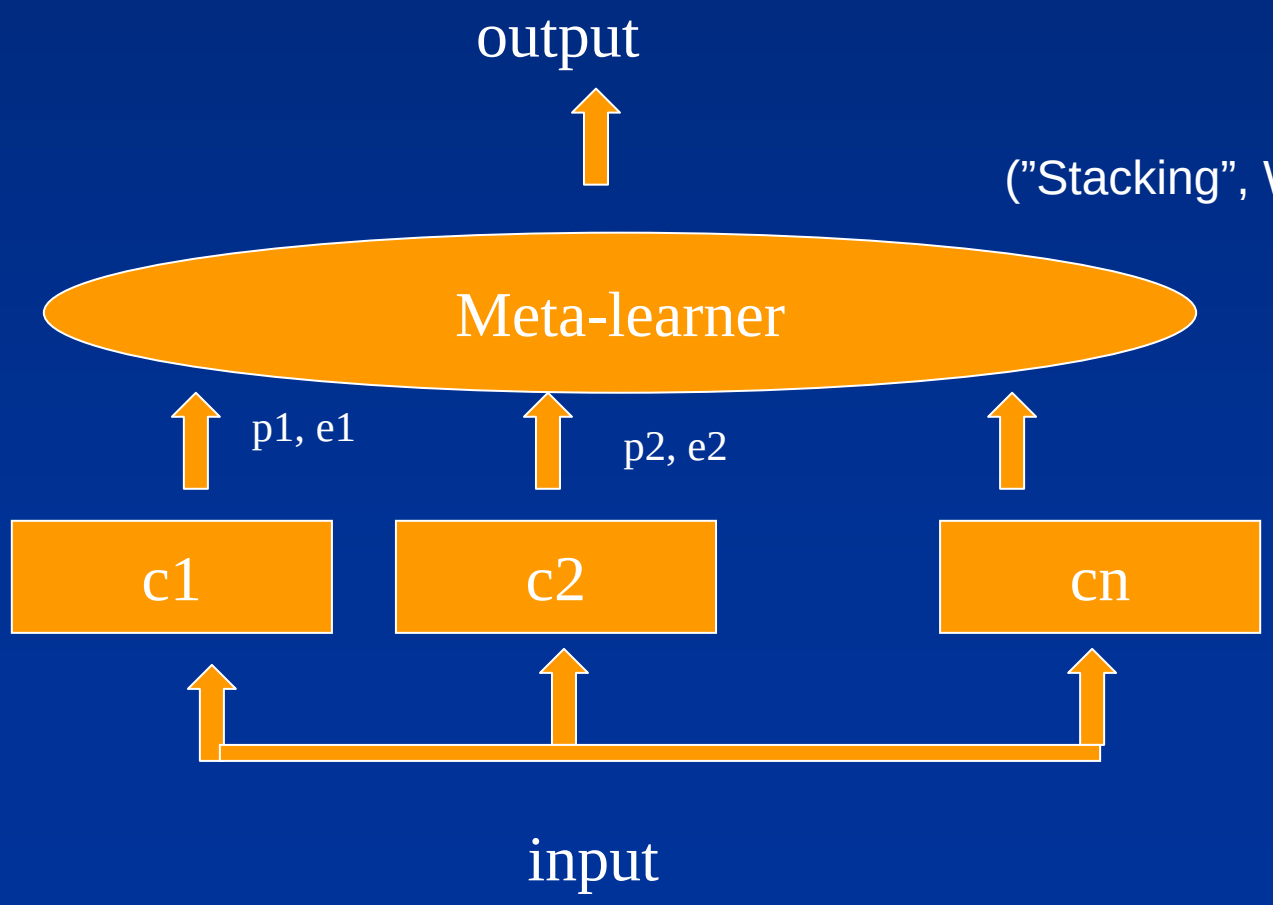
Methods for Coordinated Construction of Ensembles

The key idea is to learn complementary classifiers so that instance classification is realized by taking weighted sum of the classifiers. This idea is used in :

- Boosting - new models are influenced by performance of previously built ones
- Stacking.

Stacking

Uses *meta learner* instead of voting
Predictions of base learners (*level-0 models*) are used as input for meta learner (*level-1 model)*



("Stacking", Wolpert, 1992)

Overfitting in ensembles

- Not that much research has been done to this time
- A surprising finding:
  ensembles of overfit base classifiers
  (DTs, ANNs) are in many cases better than the ensembles of non-overfit base classifiers
  - This is related most probably to the fact that in that case the ensemble diversity is much higher

acknowledgements

- The development of poliGMDH is gratefully acknowledged to different EU projects:
  - Allelochem (2003-5) – the power of GMDH has been proven on a QSAR problem
  - EASYRING (2003-6) – a complete working version of poliGMDH (Neural Network v3) and the development of models for endocrine disruptors
  - Demetra (2003-6)– the poweer of the method and its use in ensembling
  - ION (2005-6) – the role of the method in the process of drug discovery