Neuro-fuzzy tools for toxicity prediction

Dr. Dan NEAGU
Contents...

- Artificial Neural Networks.
- Fuzzy processing.
- Neuro-fuzzy processing and hybrid intelligent systems.
- A neuro-fuzzy network tool applied in toxicity modelling and prediction: NIKE.
What is a neural network?

- “... a neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes."
  - DARPA Neural Net Study, AFCEA Int’l Press, p. 60, 1988

- “Artificial neural systems, or neural networks, are physical cellular systems which can acquire, store, and utilize experiential knowledge.”
McCulloch-Pitts model

McCulloch-Pitts model of an artificial neuron

\[ y = f \left( w_1 p_1 + \ldots + w_j p_j + \ldots + w_R p_R + b \right) \]

\[ y = f \left( W \cdot p + b \right) \]

- \( p = (p_1, \ldots, p_R)^T \) is the input column-vector
- \( W = (w_1, \ldots, w_R) \) is the weight row-vector

*) The bias \( b \) can be treated as a weight whose input is always 1.
ANN topology

- **Feedforward ANN:**
  - the connections between units do not form cycles.
  - usually produces a response to an input quickly.
  - can be trained using a wide variety of efficient conventional numerical methods.

- **Feedback (recurrent) ANN:**
  - there are cycles in the connections.
  - for each presented input, the ANN iterates for a potentially long time before produces a response.
  - are usually more difficult to train than feedforward ANNs.
Data

- **Categorical variables**
  - take only a finite number of possible values.
  - may have symbolic values (e.g., "red", "green", "blue") that must be encoded into numbers before being given to ANN.
  - both supervised learning with categorical target values and unsupervised learning with categorical outputs are called **classification**.

- **Quantitative variables**
  - are numerical measurements of some attribute, such as length in meters.
  - supervised learning with quantitative target values is called **regression**.
Vocabulary

- **Pattern**: a vector of values presented at one time to all the input units of ANN (also called "case", "example", "sample").
- **Input variable**: a vector of values presented at different times to a single input unit.
- **Data set** is the matrix of patterns (usually, patterns are rows of the matrix, while variables are columns).
- **Training set**: a set of examples used for learning, that is to fit the parameters [i.e., weights] of the classifier.
- **Validation set**: a set of examples used to tune the parameters [i.e., architecture, not weights] of a classifier, for example to choose the number of hidden units in a neural network.
- **Test set**: a set of examples used only to assess the performance [generalization] of a fully-specified classifier.
Learning algorithms for ANN

- **Supervised learning:**
  - the correct results (target values, desired outputs) are known and given to the ANN during training to adjust its weights.
  - after training, the ANN is tested by giving it input values, and seeing how close it performs the correct target values.

- **Unsupervised learning:**
  - the NN is not provided with the correct results during training.
  - unsupervised ANNs usually perform some kind of data compression, such as dimensionality reduction or clustering.
Vocabulary

- **Combination functions:**
  - each non-input unit in a neural network combines values that are fed into it via synaptic connections from other units, producing a single value called the "net input".
  - is a vector-to scalar function.

- **Activation (transfer) functions**
  - units in neural networks transform their net input by using a scalar-to-scalar function called an "activation function", yielding a value called the unit's "activation".
  - Activation functions for the hidden units are needed to introduce nonlinearity into the network. With sigmoid units, a very small change in the weights will usually produce a change in the outputs, which makes it possible to tell whether that change in the weights is good or bad.
What is backprop?

- "backprop" is short for "backpropagation of error".
- Standard backprop is the generalized delta rule (Rumelhart): the most widely used supervised training method for neural nets.
- It refers to a training method that uses backpropagation to compute the gradient.
  - An iterative steepest descent algorithm, in which the performance index is the mean square error $E$ between the desired response and network’s actual response.
- A backprop network is a feedforward network trained by backpropagation.
Backprop algorithm

**Phase I**: The input vector is propagated forward (fed-forward) through the consecutive layers of the ANN.

**Phase II**: The errors and recursively back-propagated through the layers and appropriate weight changes are made. Because the output error is an indirect function of the weights in the hidden layers, we have to use the “chain rule” of calculus when calculating the derivatives with respect to the weights and biases in the hidden layers. These derivatives of the squared error are computed first at the last (output) layer and then propagated backward from layer to layer using the “chain rule.”
Most ANNs that can learn to generalize effectively from noisy data are similar or identical to statistical methods:

- **Feedforward nets** with no hidden layer are basically generalized linear models.
- **Feedforward nets** with one hidden layer are closely related to projection pursuit regression (a subset of the class of nonlinear regression and discrimination models).
- **Probabilistic nets** are identical to kernel discriminant analysis.
- **Kohonen nets** for adaptive vector quantization are very similar to k-means cluster analysis.
- **Kohonen self-organizing maps** are discrete approximations to principal curves and surfaces.
- **Hebbian learning** is closely related to principal component analysis.

A classification of ANNs.
Acute toxicity 96 hours (LC\textsubscript{50}), for fathead minnow (*Pimephales promelas*): 560 compounds.

Descriptors - Code

- Total Energy (kcal/mol): QM1
- Heat of Formation (kcal/mol): QM3
- LUMO (eV): QM6
- Relative number of N atoms: C9
- Relative number of single bonds: C24
- Molecular weight: C35
- Kier&Hall index (order 0): T6
- Average Information content (order 1): T22
- Moment of inertia B: G2
- Molecular volume: G10
- Molecular surface area: G12
- TMSA Total molecular surface area: E13
- FPSA-2 Fractional PPSA (PPSA-2/TMSA): E24
- PPSA-3 Atomic charge weighted PPSA: E28
- FPSA-3 Fractional PPSA (PPSA-3/TMSA): E31
- logD: pH9
- logP: logP
Let’s train IKM-ANN through NIKE

- **Required values:**
  - **Path:** C:\IMAGETOX\DuluthMols\work (inserted in the text files C:\IMAGETOX\DuluthMols\work\ProjectPath.dan, and C:\NIKE\work\ProjectPath.dan)
  - All the other parameters are listed in the files of the folder Data of the project (C:\IMAGETOX\DuluthMols\work\Data)
  - **ParametersCNN.dan:**
    - transfer functions for I, H, O layers (logsig, logsig, purelin);
    - Training algorithm (traingdx) combines adaptive learning rate with momentum training, the goal to stop training (0.001);
    - Number of epochs (100) to show, the number of training epochs if not succeed (5000);
    - The momentum term (0.95): a relatively high learning rate ensures rapid finding of the error function minimum, and a high momentum term prevents too many oscillations of the error function;
    - Three parameters about bias connection to I, H, O (0, 0, 0).
NIKE (Neural explicit & Implicit Knowledge inference system)

- NIKE is a hybrid intelligent system based on modular neural/neuro-fuzzy networks, which is a shell supporting different strategies to build assemblies of neural, neuro-fuzzy, and fuzzy inference systems implemented in Matlab R12 © Mathworks.

- The implicit knowledge (NIKE) is the knowledge represented by neural/neuro-fuzzy networks, created and adapted by a learning algorithm.

- The explicit knowledge (NIKE) is a collection of neural networks, which are computationally identical to the I/O relations set, and are created by mapping the given fuzzy rules into hybrid neural networks.
Let’s train IKM-ANN through NIKE (2)

- **ProjectVars.dan:**
  - number of input variables (17)
  - number of training patterns (401);
  - number of test patterns (568);
  - number of hidden neurons (100);

- **TrainI.dan, TrainO.dan:** training set (70%x568);
- **TestI.dan, TestO.dan:** test set (568);
- **PredictI.dan, PredictO.dan:** prediction test pattern (1);
- **VarNames.dan:** the names of I/O variables of the project.

- Run the project: NIKE and go to the section IKM – crisp values.

- Choose the current number of the hidden neurons:
  - for an ANN, to be able to generate closed decision regions, the minimum number of hidden units must be greater than the number of input units.
  - the maximum number of hidden units in ANN, needed to represent any function of \( n \) variables, is less than twice the number of inputs \( 2 \times n_{\text{input}} + 1 \).

- IKM-CNN could be re-trained!
Let’s train IKM-ANN through NIKE (3)
Results of IKM-ANN through NIKE

The predicted value: 0.8417 | the real value: 0.8436
Check file: ComputedOutputCNN23H.dat for test values

Number of Hidden Neurons NH: 23

IKM-MLP(CNN): crisp values
Fuzzy processing

- Fuzzy Sets
- Membership functions
- Operations on Fuzzy Sets
- Linguistic variables: descriptors
- Fuzzy Rules and Inferences
What is a Fuzzy Set?

- A fuzzy set is a set without a crisp, clearly defined boundary.
- It can contain elements with only a partial degree of membership, i.e. the set of young people graphically represented by its characteristic function.
Membership Functions

- A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1.
  - Let $X$ be a nonempty set. A fuzzy set $A$ in $X$ is characterized by its membership function $\mu_A : X \rightarrow [0,1]$.
  - $\mu_A(x)$ is interpreted as the degree of membership of element $x$ in fuzzy set $A$ for each $x$ from $X$.

- Membership functions: Bell, Gaussian, Pi, S, Z, Triangular, Trapezoidal, and Sigmoidal.
Operations on Fuzzy Sets

- Let $A$ be a fuzzy interval between 5 and 8 and $B$ be a fuzzy number about 4:
Linguistic variables

- A numerical variable takes numerical values: \( \text{LUMO}=0.5572 \)
- A linguistic variable takes linguistic values: \( QM6 \) is Medium
- A linguistic value is a fuzzy set.
- The collection of all the linguistic values is a term set: \( QM6=\{\text{Low,Medium,High}\} \)
Fuzzy shapes for descriptors
Fuzzy IF-THEN Rules

- Mamdani:
  - IF D₁ is Low AND D₂ is High THEN Tox is Medium

- zero-order Sugeno fuzzy rule:
  - IF D₁ is Low AND D₂ is High THEN Tox=k

- first order Sugeno fuzzy rule:
  - IF D₁ is Low AND D₂ is High THEN Tox=0.72x D₁+0.12xD₂-0.11
Fuzzy Inference System

1. Fuzzify inputs.

2. Apply fuzzy operation (OR = max).

3. Apply implication method (min).

4. Apply aggregation method (max).

Result of a fuzzy rule

- In general, the input to an if-then rule is the current value for the input variable and the output is an entire fuzzy set.
- Interpreting an if-then rule involves distinct parts:
  - first evaluating the antecedent (which involves fuzzifying the input and applying any necessary fuzzy operators)
  - second applying that result to the consequent (known as implication).
- The result set will later be defuzzified, assigning one numerical value to the output.
Defuzzify

- The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single output value from the set.

- There are five built-in methods:
  - som: smallest of maximum method
  - lom: largest of maximum method
  - bisector: bisector of area method
  - centroid: center of area (under the shape of the output) method
  - mom: mean of maximum method (the average of the maximum value of the output set)
The neuro-fuzzy approach

- Artificial neural networks are good at recognizing patterns, they are not so good at explaining how they reach their decisions.

- Fuzzy logic systems, which can reason with imprecise information, are good at explaining their decisions, but they cannot automatically acquire the rules they use to make those decisions.

- Fuzzy logic could be considered the software implementation of human reasoning, and neural networks its hardware counterpart.
The MAPI neuron (Rocha 1992)

- A generalized neuron with fuzzy processing capabilities
- The process of fuzzy reasoning based on GMP (Zadeh) is performed in four steps:
  - **Matching** (the compatibility at the input between input and pattern);
  - **Aggregation** (based on triangular norm);
  - **Projection**: is a function measuring the compatibility of: (Y is B') with (Y is B);
  - **Inverse-Matching and Defuzzification** (performed at the axonic terminals of MAPI neuron).
Neuro-Fuzzy Systems NFS

- **Concurrent NFS**: modifies the output or produce the input of a FS
- **Cooperative NFS**: the ANN determines off line membership functions (Nomura), fuzzy rules (Pedrycz) or fuzzy weights (Kosko)
- **Hybrid NFS**: an architecture that can be interpreted as ANN with fuzzy inputs/weights/outputs/activation functions
Hybrid Fuzzy Neural Networks

- HNN: a neural network with T-norm/ T-conorm aggregation and activation functions of units processing real signals and weights.
- FNN: a neural network with fuzzy inputs and/or weights:
  - FNN1: real inputs, fuzzy weights;
  - FNN2: fuzzy inputs, real weights;
  - FNN3: fuzzy inputs, fuzzy weights.
Let’s develop IKM-FNN2 through NIKE
Let’s develop IKM-FNN2 through NIKE (2)

- **Required values:**
  - **Path:** C:\IMAGETOX\DuluthMols\work (inserted in the text file C:\IMAGETOX\DuluthMols\work\ProjectPath.dan, and C:\NIKE\work\ProjectPath.dan)
  - All the other parameters are listed in the files of the folder Data of the project (C:\IMAGETOX\DuluthMols\work\Data)
  - **ParametersFNN.dan:**
    - transfer functions for I, H, O layers (logsig, logsig, purelin);
    - Training algorithm (traingdx) combines adaptive learning rate with momentum training, the goal to stop training (0.001);
    - Number of epochs (100) to show, the number of training epochs if not succeed (5000);
    - The momentum term (0.95): a relatively high learning rate ensures rapid finding of the error function minimum, and a high momentum term prevents too many oscillations of the error function;
    - Three parameters about bias connection to I, H, O (0, 0, 0).
Let’s develop IKM-FNN2 through NIKE (3)

- **ProjectVars.dan:**
  - number of input variables (17)
  - number of training patterns (401);
  - number of test patterns (568);
  - number of hidden neurons (100);

- **TrainI.dan, TrainO.dan:** training set (70%x568);
- **TestI.dan, TestO.dan:** test set (568);
- **PredictI.dan, PredictO.dan:** prediction test pattern (1);
- **VarNames.dan:** the names of I/O variables of the project.

- **FuzzyIO.fis** - the information about the fuzzy system:
  - generated in a neutral working form through Fuzzify! button (NIKE);
  - it could be modified: AndMethod='min', OrMethod='max', ImpMethod='min', AggMethod='max', DefuzzMethod='centroid';
  - The fuzzy shape of any input or the output (see FuzzyShapes.pdf): NumMFs=3, MF1='Low':'trapmf',[0 0 0.2 0.4], MF2='Med':'trapmf',[0.2 0.4 0.6 0.8], MF3='High':'trapmf',[0.6 0.8 1 1];
  - The rules part is subject for EKM in Mamdani form!!!
Let’s develop IKM-FNN2 through NIKE (4)

- Run the project: NIKE… eventually fuzzify! and go to the section IKM-fuzzy values.
- Choose the current number of the hidden neurons of FNN.
- IKM-FNN could be re-trained!
Results of IKM-FNN through NIKE

Prediction accuracy for the fuzzy output of FNN25H

The predicted values of log10(C50) vs. the real values of log10(C50)

The predicted value: 0.72325 | the real value: 0.8436

Number of Hidden Neurons NH: 25

Check file: ComputedOutputFNN25H.dat for test values

NIKEX Project

Defuzzification Methods for the fuzzy output of FNN25H

NIKEX Project

Predict

Click the [Predict] button to predict the output from trained FNN module using PREDICT.dat data.

Use the slider to choose the number of neurons in the hidden layer.

Number of Hidden Neurons NH: 25
Hybrid Intelligent Systems

Hybrid Neuro-Symbolic Systems

Two of the major approaches in machine learning are: symbolic and neural (sub-symbolic).

Until the last decade both approaches progressed independently. In the last decade along with independent progress in both areas, researchers started investigating ways of integrating both approaches.

There has been a great progress in the last decade in both areas, but is still a lot to be done.
Major approaches in Hybrid Neuro-Symbolic Systems

- Integrating neural nets with expert systems.
- Connectionist expert systems.
- Domain knowledge integration in neural networks.
- Converting neural networks to decision trees or converting decision trees to neural networks.
Types of integration in Hybrid Neuro-Symbolic Systems

- Chainprocessing
- Subprocessing
- Metaprocessing
- Coprocessing
Advantages of symbolic and subsymbolic approaches

- Reasoning with noise, imprecise, incomplete data
- Experience based inferences
- Intuitive cognitive processing
- Modularization
- Learning by examples
- Neuro-physiology applications

CONNECTIONIST AI

SYMBOLIC AI

- Natural language processing
- Explanation
- Step-by-step reasoning
- High-level models
Combining is the key

- One of the weakest part of expert systems is knowledge acquisition. Artificial Neural Networks gather knowledge from available examples.
- The weakest side of neural networks is the lack of explanation capability. Explanation capability is one of the strongest aspects of the expert systems.
- A hybrid neural-expert system is supposed to solve both of these problems. The system is supposed to be able to learn from examples and give explanations.
MPNN equivalent with a fuzzy rule

MPNN equivalent with a rule with two premises.
Fuzzy rules extracting from trained ANNs

- The method of knowledge acquisition based on fuzzy connectives programming.
- The rules extraction techniques based on the effects of input neurons to the outputs of the network.
- The number of rules equal to the number of hidden neurons.
Fuzzy Rules extracted from IKM-CNN through NIKE

IF (QM1 is greater than approximately 21.59)
iOR(QM3 is greater than approximately 6.795)
iOR(QM6 is greater than approximately 2.75)
iOR(C9 is greater than approximately 2.757)
iOR(C24 is greater than approximately 4.056)
iOR(C35 is not greater than approximately 4.389)
iOR(T6 is not greater than approximately 21.57)
iOR(T22 is not greater than approximately 3.199)
iOR(G2 is greater than approximately 4.532)
iOR(G10 is greater than approximately 2.769)
iOR(G12 is greater than approximately 8.424)
iOR(E13 is greater than approximately 20)
iOR(E24 is greater than approximately 6.056)
iOR(E28 is greater than approximately 9.155)
iOR(E31 is greater than approximately 4.816)
iOR(pH9 is not greater than approximately 6.802)
iOR(logP is greater than approximately 2.476)

THEN log1/LC50=-0.0545
Weighted fuzzy rules extraction from a trained neuro-fuzzy network

- The identification of fuzzy rules is based on an analysis of the strengths of the connections between the input neurons through the hidden layer to the output neurons.
- Two general approaches seem possible:
  - the number of connections to be considered is reduced by selecting only those that meet some threshold value.
  - the overall effect of each input neuron on each output through a process of weight vector multiplication is estimated.
- Both methods supply monotonic rules sets, are limited to single hidden layer neuro-fuzzy networks and require inputs and outputs fuzzification before training.
Fuzzy Rules extracted from IKM-FNN through NIKE

Effect Measure Method:

IF QM1 is:High THEN log1/LC50 is:Low (100.00%)
IF QM3 is:Low THEN log1/LC50 is:Low (97.15%)
IF QM3 is:Med THEN log1/LC50 is:Low (72.73%)
IF QM6 is:Med THEN log1/LC50 is:Medium (41.82%)
IF QM6 is:High THEN log1/LC50 is:Low (81.41%)
IF C9 is:Low THEN log1/LC50 is:VeryLow (42.38%)
IF C9 is:Low THEN log1/LC50 is:Medium (64.36%)
IF C24 is:Low THEN log1/LC50 is:Medium (60.12%)
IF C24 is:Med THEN log1/LC50 is:Low (57.83%)
IF C35 is:Low THEN log1/LC50 is:Low (51.86%)
IF C35 is:High THEN log1/LC50 is:Low (96.70%)
IF C35 is:High THEN log1/LC50 is:High (32.69%)
IF T6 is:Med THEN log1/LC50 is:Medium (79.06%)
IF T22 is:Low THEN log1/LC50 is:Medium (69.79%)
IF T22 is:Low THEN log1/LC50 is:High (51.32%)
IF T22 is:High THEN log1/LC50 is:Low (48.24%)
IF G2 is:Low THEN log1/LC50 is:Medium (60.02%)
IF G2 is:Med THEN log1/LC50 is:High (33.84%)
IF G2 is:High THEN log1/LC50 is:Medium (49.07%)
Combining neural modules

- Aspects of modularization in neural networks systems.
- Modular neural networks.
- Knowledge insertion in modular neural structures.
- Explicit and Implicit Knowledge-based System.
Advantages of connectionist modular architectures

- **Learning speed**
  - Modular structures can take advantage of function decomposition.
  - Modular architectures can be design to minimize conflicting training data (spatial crosstalk and temporal crosstalk) that tends to retard learning.

- **Generalization**
  - Decomposing a complex function into a set of simpler functions learned by small neural networks.
  - Modular architectures realize a summum of local generalizations.

- **Representation**
  - Modular structures enforce easy understanding symbolic representation design of neural modules.
  - Modular representation focuses the efforts to solve complex problems.
  - Easy-to-debug modules.
  - Simplifies data learning efforts.
  - Proposes incremental growing-up design of neural systems.

- **Hardware constraints**
  - Suitably designed modular architectures can reduce the number of units and the length of connections.
Modular networks

\[ Y = \sum_{k=1}^{K} g_k Y_k \]
Implicit knowledge neural networks pre-programming through explicit knowledge insertion

Methods trying to incorporate a priori knowledge into neural networks without immediate access to the internal structure are so called Concept Support Methods (CSM-Prem et al., 1993).

The basic idea of CSM is to pre-train the neural network on the basis of symbolic knowledge, so as to obtain a weights matrix on which further learning with training samples is based.

Several variation of CSM can be distinguished:

- explicit knowledge insertion based on rules describing a subset of cases of the desired input-output mapping;
- pre-training on the basis of relevant concepts (CSM based on a part of the hidden structure of the network).
E&IKM integration

- The global architecture combines the explicit sub-modules EKM and implicit sub-modules IKM, using a gating network, which mediates the competition of all involved expert networks.
END - the conclusion: HIS = Soup pot!