Substance Use & Misuse, 33(2), 439–461, 1998

MetaNet*: The Theory of Independent Judges

Massimo Buscema, Dr.

Semeion Research Center of Sciences of Communication, Viale di Val Fiorita 88, Rome, Italy

THE THEORY OF INDEPENDENT JUDGES

It is known that different ANNs provided with different learning equations and qualitatively and quantitatively different internal architectures can compute similar functions (Kosko, 1992). This property can have important consequences in instances of solution of *classification* problems. Not all ANNs codify all *k* classes of the problem with the same efficacy, nor are their performances' hierarchies always linear. Whereas ANN 1 can codify class *A* in a more efficacious way, ANN 2 can better codify class *B* and so on. Under such condition it is appropriate to select a MiniMax type of analysis: choosing the ANN with *less* costly errors and whose successes produce *more*. But this is a "monarchical" logic and the ANN, selected as the "less worse", is an imperfect "monarch".

We propose an opposing logic inspired by *democratic* consideration. It consists, as it were, of *democratic committees* constituted by laws and statute by simple, imperfect subjects, but inclined in their plurality and globalness to compensate for the weaknesses of their single participants.

We have defined the scientific consequences of this hypothesis: *Theory of Independent Judges* (TIJ) (Buscema, 1994).

In the TIJ each ANN is an expert judge of the problem which he has had to face. Its *credibility* is defined by its performances in the *Testing* and/or *Validation* phase.

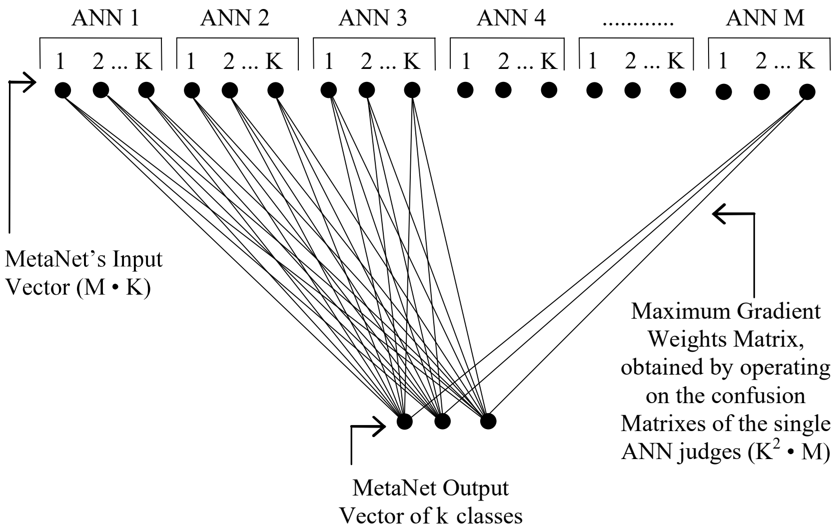
* This Network was created by Massimo Buscema in 1995.

In cases of classification problems (1 of N), each judge will have a *different credibility* of the different aspects of the same problem, where each aspect is 1 of N codification classes of the Input vectors.

The TIJ believes that the specific credibility of each judge-ANN of a classification problem is implicitly contained in the *Confusion Matrix* (CM) that defines its performances regarding that problem during the *Testing* phase.

STRUCTURE OF METANET

Then, the TIJ posits that given *M judges-ANN* and *M Confusion Matrixes* which define their performances in an ordered way, it is possible to create a *MetaNet* which has: as *Input* the Output sum of all *M* judges; as *Output* the *N* classes of the problem; as *Weights Matrix* the application solution of a specific algorithm of all judges confusion matrixes; and finally as an *equation of the signal propagation* a generic cooperative and competitive algorithm of the Feed Forward type.



The aim of MetaNet's Weights Matrix is to establish the *local credibility value* of each judge about the classification problem. Then, each ANN in order to be a part of MetaNet judges Pool must give a history (curriculum) of its previous performances about the same problem.

In this sense the *Confusion Matrix* between Correct Classification and Hypothesized Classification that each ANN has created during the Testing phase is a useful curriculum.

Each judge-ANN's Confusion Matrix will be read in the following way (Table 1):

Table 1. Confusion Matrix of each judge-ANN.

		Output $_j$						
	ANN 1	A	B	C	......	Z	*Null*	*Tot. Row*
	A	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	......	$X_{1,Z}$	$X_{1,Z+1}$	$\dot{R}_1$
	B	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	......	$X_{2,Z}$	$X_{2,Z+1}$	$\dot{R}_2$
Target $_i$	C	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	......	$X_{3,Z}$	$X_{3,Z+1}$	$\dot{R}_3$
	....	......	......	......	......	......	......	....
	Z	$X_{Z,1}$	$X_{Z,2}$	$X_{Z,3}$	......	$X_{Z,Z}$	$X_{Z,Z+1}$	$\dot{R}_Z$
	Tot. Column	$\dot{C}_1$	$\dot{C}_2$	$\dot{C}_3$	......	$\dot{C}_Z$	$\dot{C}_{Z+1}$	

from which:

(1) $$\mathit{Successes}\ (S) = \frac{X_{ii}}{\dot{R}_i}$$ (X_{ii} mean $i = j$)

(2) $$\mathit{Failed\ Blows}\ (B) = \frac{\sum_{i \neq j}^{Z+1} X_{ij} - X_{ii}}{\dot{R}_i}$$

(3) $$\mathit{False\ Attributions}\ (F) = \frac{\sum_{j \neq i}^{Z} X_{ij} - X_{jj}}{\dot{C}_j}$$

(4) $$Correct\ Eliminations\ (E) = \frac{X_{jj}}{\overset{\bullet}{C}_j}$$

Starting from the *CM* of each judge-ANN, it is possible to calculate MetaNet's Weights Matrix in different ways. We are going to propose some of them.

METHODS FOR THE CALCULATION OF METANET WEIGHTS

Confusion Matrixes and Obsession Indexes: Bayes's Equation

The first way for calculating the MetaNet weights utilizes the doubly conditioned probability which we express in the following form:

(5) $$w_{ij} = -\lg_n \frac{P\left(x_i = 1 \cap x_j = 0\right) \cdot P\left(x_i = 0 \cap x_j = 1\right)}{P\left(x_i = 1 \cap x_j = 1\right) \cdot P\left(x_i = 0 \cap x_j = 0\right)}$$

where w_{ij} = MetaNet's weight value between an ANN Input j and the MetaNet Output i.

In order to calculate the *Self-Categorical* (Output of a judge-ANN connected as an Input to the same Output type of MetaNet) weights between a judge-ANN and the MetaNet Output, equation (5) will take the following form:

(6) $$w_{ij} = -\lg_n \frac{B \cdot F}{S \cdot E}$$

In order to calculate the *Hetero-Categorical* weights value between a judge-ANN and the MetaNet Output, equation (6) will remain unvaried, while it will be necessary to "*de-idealize*" equations (1)–(4).

In fact, they suppose each judge-ANN's successes are only readable on the principal diagonal of the *CM*. It will be sufficient to create a *relative* point of view which allows each *CM* component to count up "its successes" in order to obtain a series of *Obsessions Indexes* of the ANN.

In fact, if we assume that K is the relative point of view which defines the subjective exactness pertinent to each component of the CM line vector, then we have that:

$$\text{(1a)} \quad \textit{Obsession Rate } (O) = \frac{X_{ik}}{\dot{R}_i}$$

$$\text{(2a)} \quad \textit{Removal Rate } (R) = \frac{\sum_{j \neq k}^{Z+1} X_{ij} - X_{ik}}{\dot{R}_i}$$

$$\text{(3a)} \quad \textit{Perversion Rate } (P) = \frac{\sum_{i \neq k}^{Z} X_{ji} - X_{ik}}{\dot{C}_j}$$

$$\text{(4a)} \quad \textit{Simplification Rate } (T) = \frac{X_{jk}}{\dot{C}_j}$$

The *Obsessions* (O) represents how much an ANN persists in classifying those Input vectors as belonging to a *specific class* (obsessive class) that which "objectively" belongs to *another specific class* (removed class).

The *Removals* (R) represents how much an ANN refuses to consider those Input vectors as belonging to one of its specific class those which objectively belong to another specific class and which it usually assigns to its obsessive class.

The *Perversions* (P) represents how much an ANN also assigns to one of its "obsessive classes" the Input vectors which don't belong to its "removed class".

The *Simplifications* (T) represents how much an ANN tends to *globalize* (to bring together) in its obsessive classes the whole universe of the Input vectors. We could speak of its of degree of *monodependence*.

Collecting the scores in a table which are at the basis of the Self-Categorical and Hetero-Categorical weights matrix of the MetaNet, we will have:

Weights Matrix of MetaNet

Self-Categorical Weights		**Hetero-Categorical Weights**
Successes (S)	$\longleftrightarrow$	Obsessions (O)
Failed Blows (B)	$\longleftrightarrow$	Removals (R)
False Attributions (F)	$\longleftrightarrow$	Perversions (P)
Correct Eliminations (E)	$\longleftrightarrow$	Simplifications (T)
$\downarrow$		$\downarrow$
(6) $w_{ij} = -\lg_n \dfrac{B \cdot F}{S \cdot E}$		(6a) $w_{ij} = -\lg_n \dfrac{R \cdot P}{O \cdot T}$

It is possible in this way to find the values of M weights' matrix between each judge-ANN and MetaNet. Furthermore, each weights' matrix will identify the *local credibility* with which each judge-ANN will be able to offer a useful or misleading contribution, when assigning a specific belonging class to each Input vector.

The Normalized SoftMax Equation

This method consists of calculating all MetaNet weights starting from the Confusion Matrixes of each judge-ANN according to the following equation, already known as SoftMax (Bridle, 1989):

$$(7) \qquad w_{ij} = \frac{e^{Row_{ij}}}{\sum_k^M e^{Row_{ik}}}$$

where: $Row_{ij} = \dfrac{Freq_{ij}}{FreqTotRow_i}$; M = Output Number of MetaNet

According to equation (7) all MetaNet connection values between 0 and 1 are included.

If we want to transform the heterocategorical connections of Metanet, we must first normalize them:

$$w_{ii} = \sum_{i \neq j}^{N} w_{ij}$$

where N = Column number of the Confusion Matrix of each judge-ANN;

and afterwards *the sign is changed* to the new normalized other heterocategorical weights' value.

In practice:

(8a) $$Sum_i = \sum_{i \neq j}^{N} w_{ij} \,;$$

(8b) $$Ratio_i = \frac{w_{ii}}{Sum_i};$$

(8c) $$w'_{ij} = w_{ij} \cdot Ratio_i \qquad \text{per } i \neq j \,;$$

and finally:

(8d) $$w''_{ij} = -w'_{ij} \qquad \text{per } i \neq j \,.$$

This method's advantages are dual. Firstly, it consists in *forcing* MetaNet to avoid the most common *classification errors* in many of the judge-ANNs. Secondly it does *not inhibit* too forcefully the *correct classification* possibility present only in some judge-ANNs.

The AutoAssociative Back Propagation Method

A second way to define the Weights Matrix between each judge-ANN and MetaNet consists of utilizing an AutoAssociative ANN of a Back Propagation type.

In this second case, the *CM* lines of each judge-ANN constitute the Patterns to be learned, and the columns constitute each Pattern's Nodes value.

Once the learning has occured (with a very reduced RMSE – Root Means Square Error), the weights' matrix matured from the AutoAssociative ANN will be the weights matrix which connects each judge-ANN Output to each MetaNet Output.

The theoretical advantage of this second procedure compared to the first is due to the fact that the Successes, Failed Blows, False Attributions, and Correct Eliminations on one side, and the Obsessions, Removals, Predictions and Simplifications of each judge-ANN on the other side are computed in parallel, and not through computation of variable couples. The disadvantage lies in the fact that this procedure takes a great amount of computation time.

The Self-Reflexive Method

The Self-Reflexive ANN (SR) have been demonstrated to be very good energy minimizators of the function $f(x) = 0$ (Buscema, 1995).

The SR can be used in order to *optimize* the parallel connections among all variables of the Confusion Matrix for each single judge-ANN.

The method is rather simple: the CM Rows are the Models that the SR must learn, and the Columns represent the Input vector. For the learning a Monodedicated SR is used (compare Buscema, 1995) by initializing all the weights with the same value. At the end of the learning, with a very small RMSE, the weights matrix that connects the Hidden layer with that of Output is extracted from the SR, and this matrix is used as the matrix which connects the Input of a judge-ANN with the MetaNet Output.

Remember that in an SR, the Hidden-Output weights matrix is the projection in a dimension space N^2 of the Input vector dimension space N.

This method's advantages are remarkable: parallel calculation of the correlations between each variable and any other; a faster convergence than BP; and precision in the definition of the trained weights value, because all are initialized with the same value.

THE ANSWERS' COMPUTATION

At this point the MetaNet Weights Matrix is complete. A feed forward algorithm is needed, capable of transfering the signal from each judge-ANN Output node to each MetaNet Output node.

For this purpose, we have chosen an algorithm of a hybrid nature, known as SoftMax, which synthetizes characteristics of *cooperative* and *competitive* types among the Output nodes.

MetaNet's forward cycle takes place in two steps:

1. Calculation of the *Net Input* at each Output Node:

(9a) $$Net_i = \sum_J^N u_j \cdot w_{ij}$$

where N is the number of MetaNet's Input and it is equal at M • K, where M is the number of judge-ANNs and K is the number of the Output classes of each judge.

2. Calculation of the activation value of each Output Node:

(9b) $$u_i = \frac{e^{Net_i}}{\sum_z^K e^{Net_z}}$$

where K is the number of MetaNet's Output, or of each Input assigning classes.

At this point it will be sufficient to choose as MetaNet Output the Node with the highest value assigned by MetaNet to equation (9b).

METANET'S POSSIBLE RESULTS

MetaNet's performances can theoretically be of four types:

1. MetaNet's answers result percentwise close to the mean answers of the ANN which are part of the judges' pool or are quite lower than they. In such an instance the present article should make no sense.
2. MetaNet's answers are close to the best judge-ANN's answers of the pool. In this case it is necessary to consider if in each class, MetaNet will be able to select better judge-ANNs among the best performances or at least avoid the poorest answers which the judge-ANNs present with certain classes.
3. MetaNet's answers are results globally better than those of the best of the judge-ANNs; but each of its performances, in each class, does not exceed the best performance in that same class of the other judge-ANNs.
4. MetaNet's answers, in addition to being better than those of the best judge-ANN, improve the best performances in one or more classes in those classes of the other judge-ANNs. In this case MetaNet demonstrates having an *inferential* and *generalization capacity* which goes beyond the simple *optimization* capacity.

The experiment documents that according to the judges' pool, which operates as MetaNet's knowledge base, MetaNet always offers results which can be noted above in the last two cases.

PSYCHOTIC AND SENSIBLE NETWORKS

Besides the experimental advantages that MetaNet offers and that are documented further in this article, MetaNet also allows for interesting theoretical cues.

The pool of the judge-ANN on which the MetaNet works is not and must not be constructed by ANNs which have shown themselves to be the best in classification tasks, nor to be the most balanced in predicting the belonging of Input vectors to the different classes.

Let's imagine the performances of three judge-ANNs in assigning to three Output classes' Input vectors:

	K= 1	**K= 2**	**K= 3**	**Mean**
ANN 1	90%	90%	70%	83.3%
ANN 2	85%	95%	65%	81.6%
ANN 3	75%	65%	80%	73.3%

Let's suppose that we need to choose only two ANNs to put into the judges' pool (in reality this is not necessary). From a näive perspective it would seem that the first and the second ANNs are better than the third.

In fact, if the third ANN improves the recognition performances of class n. 3 by 10-15 percentage points it makes the recognition of class n. 1 worse by as many as 10-15 percentage points and furthermore, it makes the recognition of class n. 2 worse by 25-30 percentage points. Therefore, according to a certain point of view, choosing the second ANN is worse than choosing the first, and has a *minimum advantage* to a *maximum cost*.

From MetaNet's point of view the third ANN is an indispensable judge for its pool. This is so because MetaNet's weights matrix is such that it *rewards* the good results and *punishes* the bad results of each ANN. As a result MetaNet would positively feel the good performances of the first two ANNs with classes n. 1 and n. 2, and it would not feel the bad results of the third ANN with the same classes without feeling the mediocre performances the same class of the first two ANNs.

Of course, all this is theoretically true and has also been true in the experiments; but every time, MetaNet's result depends on the number and quality of the examples on which the judge-ANNs are confirmed and on the distribution of each of their answers.

Nevertheless, it can be asserted that MetaNet shows that many *unbalanced judges*, being somewhat obsessive and paranoic, work much better on the whole than many *balanced judges* having the same competencies.

It could be said that as "human health" is the compensated product of the *local lack of balance* of the different organic and psychologic actors of a same man, so too is MetaNet's efficacy the result of the different "psychoses" of the judges' pool which constitutes its local actors.

In short: many local lacks of balances which compensate themselves produce a global dynamic balance, while many local balances bring the system to emphasize their own vulnerabilities.

Then: different is good and works better. Health is a collection of micro-pathologies which "talk" to themselves. Disease is the autarchy of local well-beings.

In this sense, MetaNet is also a *multiethnic ANN*: it has ANNs of different "races" (Back Propagation, Learning Vectorial Quantizators, Resonance Systems, etc.) and it improves the performances of every single one, on condition that these give as their contribution *different cultures* of data with which they have had experience.

In addition, MetaNet, as any correct connector ANN, "knows" nothing of its judges' history and architecture. In fact, MetaNet doesn't "know" if one of its judges is a LVQ ANN or a Radial Basis Function. Nor does it "know" whether some of its judges are ANNs or Statistical Models or Expert Systems. Its knowledge of the judges' pool which constitutes its Consultative Committee is *absolutely local*.

In fact, what MetaNet "knows" of each of its judges is only the *Output* which it has generated and the respective *Target* obtained by each judge during the validation phase. Consequently, *classification systems* of a MetaNet's judges pool can and should be of a *completely different type* and, therefore, not only ANNs.

This allows one to utilize MetaNet as a *system integrator* and not simply as one example of a different neural system among many others.

In our opinion this is the principal line of necessary future research. This is also what concerns the simulation of the human brain. MetaNet's philosophy can perhaps successfully face the management and integration of solutions which contribute to problems of the different neural networks of the brain without recourse to symbolic structures of the type "if ..... then".

In this sense MetaNet is the demonstration that human intelligence is a complex composition of *diversity* and *unbalanced systems* more than it is a *Darwinian optimizator*.

MetaNet is a Parasite Neural Network or, if preferred, a democratic politician who organizes the diversity of the resources with regard to the need's diversity.

Our future research will concern itself with understanding how to generalize MetaNet's use with different problems than those of simple classification.

METANET APPLICATIONS

The MetaNet system has been experimented with a large number of multinomial problems. This article describes some of these applications and the results obtained.

The following descriptive layout has been adopted: description of the problem (Input variables and Output classes), presentation of the classification results obtained through Validation using various types of ANNs, and lastly, the results obtained using a MetaNet system selecting as *judges* all the ANNs which were used in relation to the problem in question.

Classification of Six Types of Surface Defects in Stainless Steel Plates

The aim was to correctly classify the type of surface defects in stainless steel plates, with six types of possible defects.

The Input vector was made up of 27 indicators that approximately "narrated" the geometric shape of the defect and its outline.

Semeion was commissioned by the *Centro Sviluppo Materiali* (Italy) for this task. It is therefore not possible to provide details on the nature of the 27 indicators used as Input vectors or the types of the 6 classes of defects.

Training of the various ANNs was carried out using a Data Base (DB) with 634 defects that were representative of the frequency of the six types of defects that could arise during the production cycle of the plates. A second DB of 634 equally representative defects was used for the validation of the various ANNs and MetaNet.

Training was carried out with 7 different ANNs:

a. **Fuzzy Art Map** (Carpenter, 1991). It has used the version implemented on NeuralWorks Professional II Plus platform (version 5.04, Patent NW2E04-62517, NeuralWare, Inc., Pittsburgh, PA). The most effective classification architecture was made up of 50 categories at the F2 level (**FAM** in short).

b. **Learning Vector Quantization** (Kohonen, 1995). The versions LVQ1 and LVQ2 implemented on NeuralWorks Professional II Plus platform (version 5.20, Patent NW2E20-60857, NeuralWare, Inc., Pittsburgh, PA) were used. The most effective architecture included 126 code-books or prototypes (21 for each of the six classes) (**LVQ** in short).
c. **Back Propagation with Self Momentum** (Rumelhart, 1986: Buscema, 1994). The version used is the one implemented on the SQUASH software package (Buscema, 1996a). The most effective architecture was made up of 2 layers of Hidden Units, with 16 units each (**SQ** in short).
d. **Functional Link Network** (Liu, 1992a, 1992b). The version used is the one implemented on the SQUASH software package (Buscema, 1996a). The preprocessing typical of this architecture transformed the original Input vector into a vector of 162 units, encoded by 2 layers of Hidden Units of 16 units each (**FL** in short).
e. **Logicon Projection Network** (Wilenky, 1992). The version implemented on NeuralWorks Professional II Plus platform (version 5.04, Patent NW2E04-62517, NeuralWare, Inc., Pittsburgh, PA) was used. The architecture involved 16 hidden units preceded by 28 preprocessing units (27 + 1) (**LP** in short).
f. **Back Propagation with Momentum and SoftMax** (Rumelhart, 1986; Bridle, 1989). The version implemented on NeuralWorks Professional II Plus platform (version 5.20, Patent NW2E20-60857, NeuralWare, Inc., Pittsburgh, PA) was used. The most effective architecture involved 2 layers of Hidden Units of 16 units each (**BP** in short).
g. **Self-Reflexive Network** (Buscema, 1995). The version used is the one implemented on the MAPS software package (Buscema, 1996b). The supervised ANN with a cascade connection to the Self-Reflexive Network had a layer of 54 Input Units and 2 Hidden Units of 16 units each (**SR** in short).

The results obtained during the Testing phase of each of the 7 ANNs as well as the results of the 4 MetaNet systems using all the 7 trained ANNs are shown in Table 2, and graphically shown in Figures 1–7.

Table 2. Results of the 7 ANNs and of the 4 MetaNet systems.

								MetaNet systems*			
% Right Recognition	**FAM**	**LVQ**	**SQ**	**FL**	**LP**	**BP**	**SR**	**MN 1**	**MN 2**	**MN 3**	**MN 4**
Class "A"	77.78	70.00	64.44	75.56	72.22	71.22	71.11	77.78	77.78	76.67	77.78
Class "B"	94.44	91.67	91.67	86.11	88.89	94.44	94.44	91.67	91.67	91.67	91.67

Class "C"	79.17	95.83	83.33	95.83	79.17	95.83	83.33	95.83	95.83	95.83	95.83
Class "D"	79.02	71.22	83.90	80.49	72.68	80.00	81.95	85.85	85.37	85.85	84.88
Class "E"	76.47	86.27	90.20	85.29	76.47	83.33	87.25	87.25	86.27	87.25	87.25
Class "F"	94.92	96.66	98.31	95.48	96.61	99.44	98.87	98.87	98.27	98.87	98.87
Total Right	83.75	81.55	86.59	85.65	81.07	86.12	86.75	89.27	88.96	89.12	88.96

* MetaNet 1 is based on Self-Reflexive algorithm; MetaNet 2 is based on Back Propagation algorithm; MetaNet 3 is based on Bayes algorithm; MetaNet 4 is based on SoftMax algorithm.

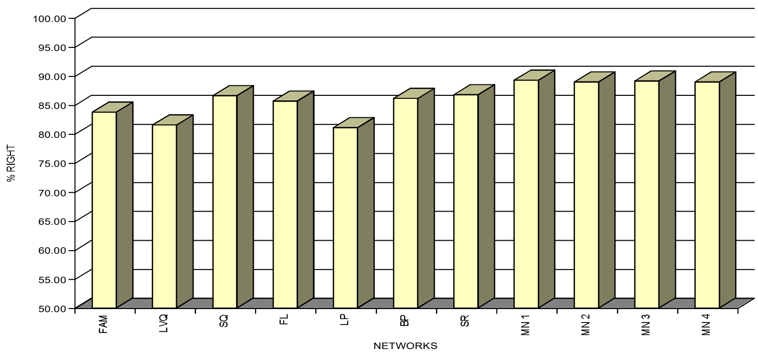


Figure 1. Results of the 7 ANNs and of the 4 MetaNet systems in all the 6 classes.

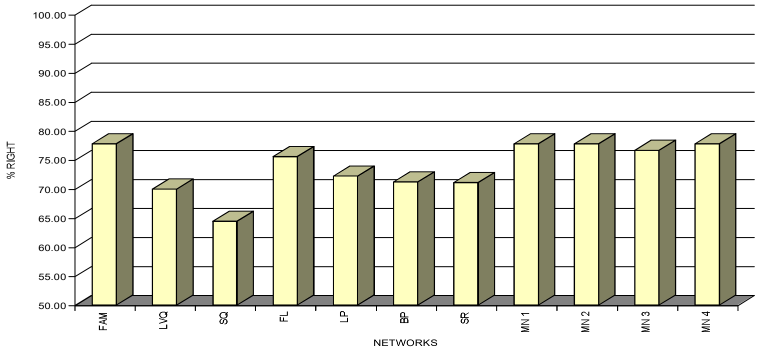


Figure 2. Results of the 7 ANNs and of the 4 MetaNet systems in the class A.

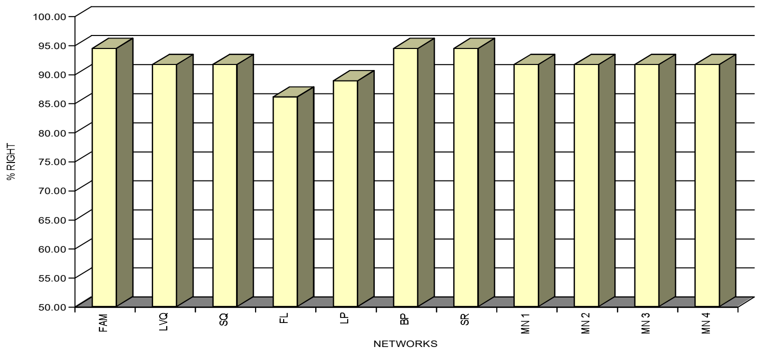


Figure 3. Results of the 7 ANNs and of the 4 MetaNet systems in the class B.

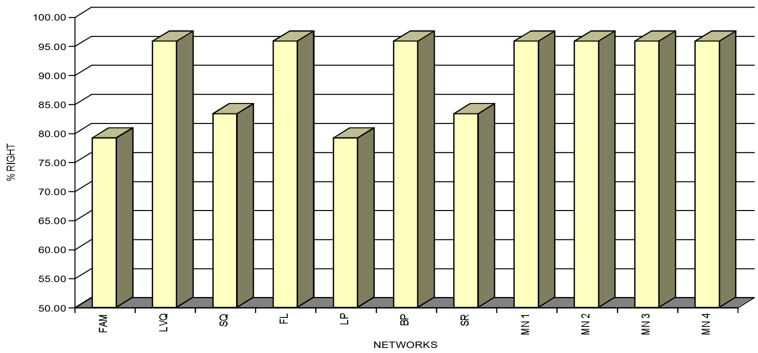


Figure 4. Results of the 7 ANNs and of the 4 MetaNet systems in the class C.

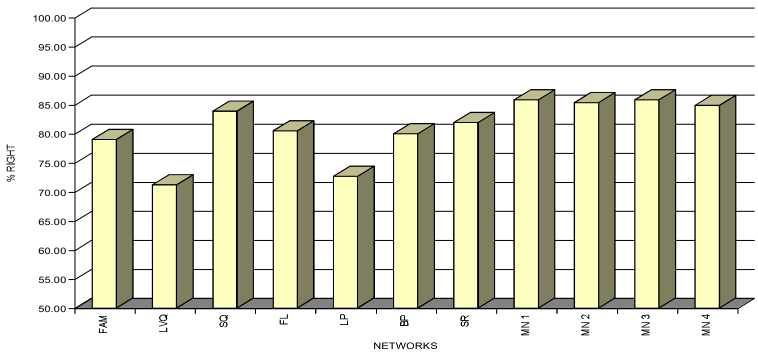


Figure 5. Results of the 7 ANNs and of the 4 MetaNet systems in the class D.

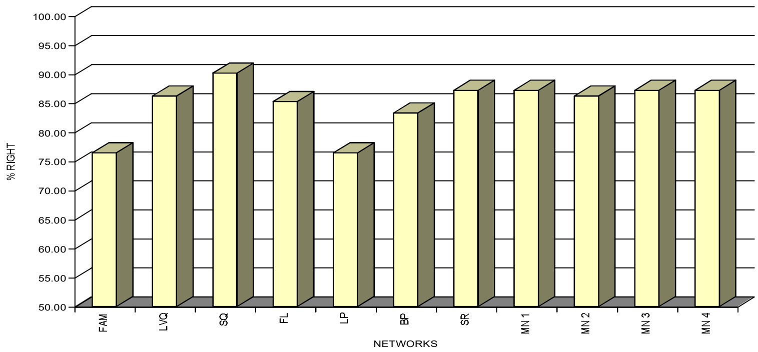


Figure 6. Results of the 7 ANNs and of the 4 MetaNet systems in the class E.

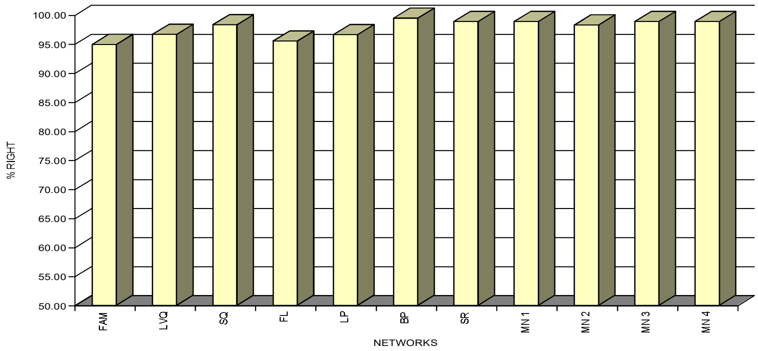


Figure 7. Results of the 7 ANNs and of the 4 MetaNet systems in the class F.

Recognition of Numbers Written by Hand

The aim was to correctly classify the figures from 0 to 9 which were handwritten by several individuals in various situations.

This experiment was carried out by Semeion in collaboration with *Tattile S.r.l.* (Italy) that was in charge of the entire procedure for images.

Input for the ANNs was reduced to a 16×16 square (256 units) of the 0/1 type (to render recognition more difficult).

Four types of ANNs were used:

- Logicon Projection Network (**LP** in short)
- Back Propagation with Momentum and SoftMax (**BP** in short)
- Back Propagation with Self Momentum and SoftMax (**SQ** in short)
- Learning Vector Quantization (**LVQ** in short)

797 figures were used for learning and an equal amount for Testing.

The two BP ANNs were provided with 2 layers of Hidden Units of 40 units each. The Logicon Projection Network had only one Hidden Unit with 40 units. The Learning Vector Quantization had 32 prototypes for each class of Output (320 code-books).

The MetaNet system used was based on the Back Propagation algorithm. The value of the results obtained must be considered in the light of the differences between the results of the 4 ANNs and those of the MetaNet system.

Table 3. Results obtained during the Testing phase of the 4 ANNs and of the MetaNet systems.

Numbers	LP	BP	LVQ	SQ	METANET
Zero	90.00%	96.25%	96.25%	92.50%	96.25%
One	85.19%	88.89%	95.06%	86.42%	95.06%
Two	92.50%	88.75%	91.25%	86.25%	95.00%
Three	87.34%	88.61%	89.87%	83.54%	91.14%
Four	74.07%	82.72%	91.36%	83.95%	88.89%
Five	93.67%	89.87%	96.20%	92.41%	97.47%
Six	88.89%	95.06%	95.06%	96.30%	95.06%
Seven	86.08%	79.75%	91.14%	83.54%	91.14%
Eight	93.51%	83.12%	88.31%	85.71%	93.51%
Nine	79.75%	87.34%	78.48%	87.34%	87.34%
Total Right %	87.06%	88.07%	91.33%	87.81%	93.09%

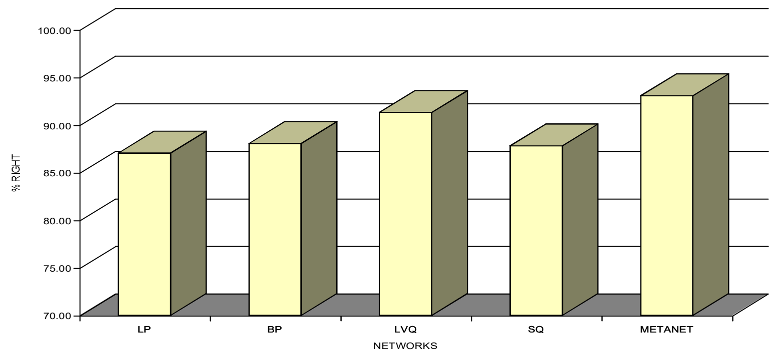


Figure 8. Results of the 4 ANNs and of the MetaNet systems in all the numbers (Total Right).

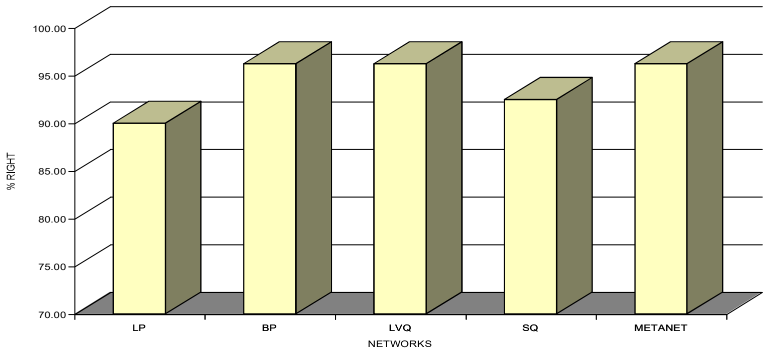


Figure 9. Results of the 4 ANNs and of the MetaNet systems in the number ZERO.

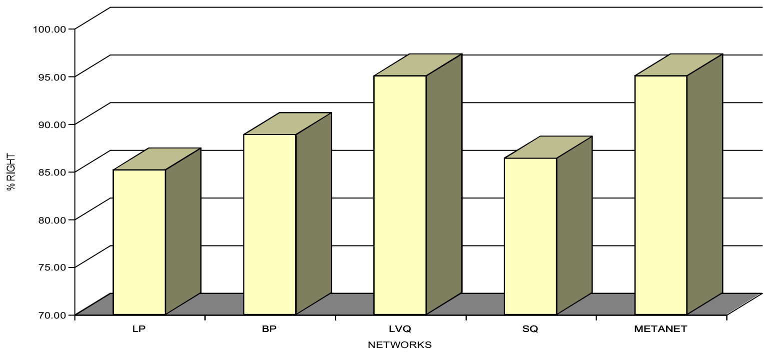


Figure 10. Results of the 4 ANNs and of the MetaNet systems in the number ONE.

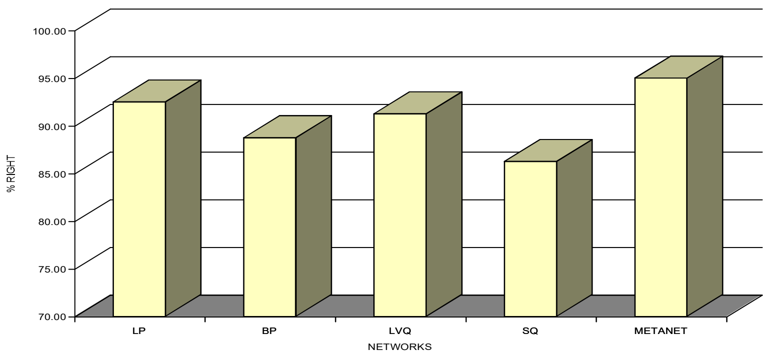


Figure 11. Results of the 4 ANNs and of the MetaNet systems in the number TWO.

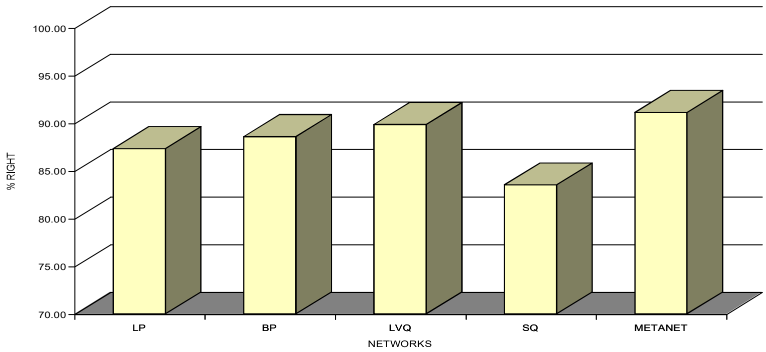


Figure 12. Results of the 4 ANNs and of the MetaNet systems in the number THREE.

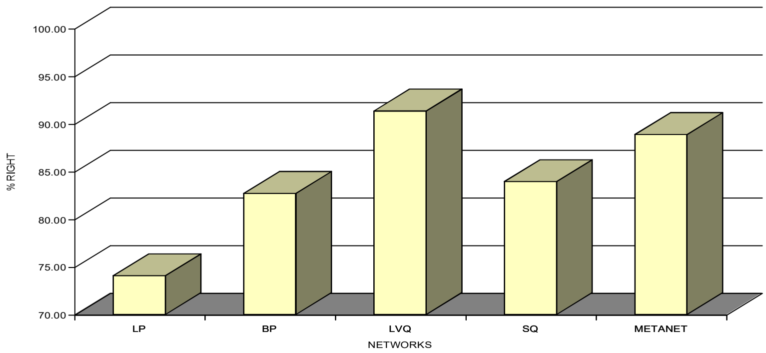


Figure 13. Results of the 4 ANNs and of the MetaNet systems in the number FOUR.

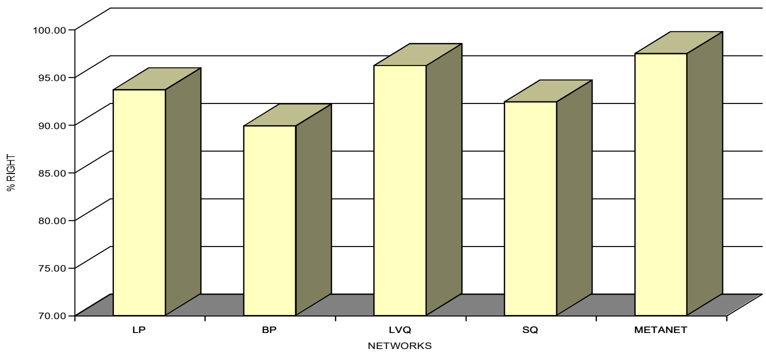


Figure 14. Results of the 4 ANNs and of the MetaNet systems in the number FIVE.

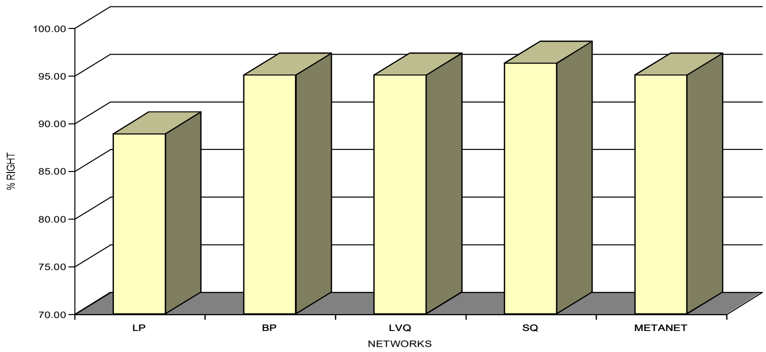


Figure 15. Results of the 4 ANNs and of the MetaNet systems in the number SIX.

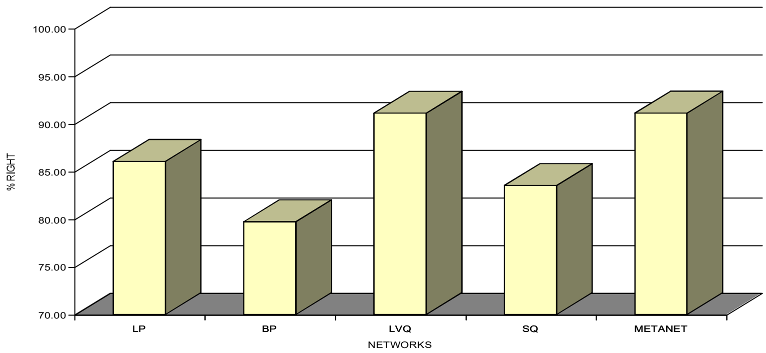


Figure 16. Results of the 4 ANNs and of the MetaNet systems in the number SEVEN.

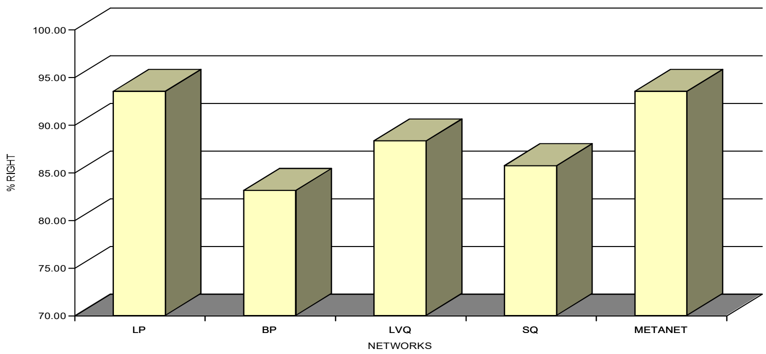


Figure 17. Results of the 4 ANNs and of the MetaNet systems in the number EIGHT.

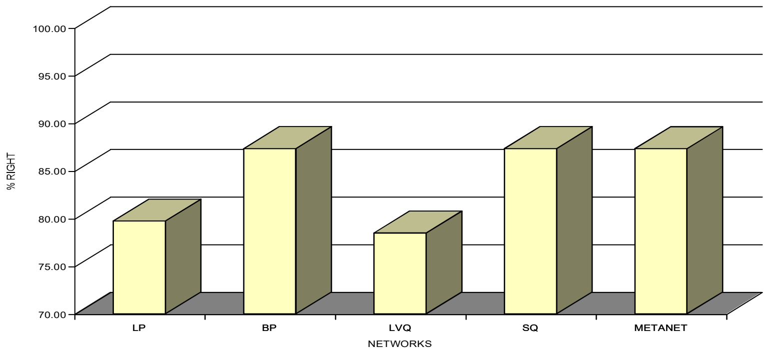


Figure 18. Results of the 4 ANNs and of the MetaNet systems in the number NINE.

REFERENCES

BRIDLE, 1989: J. S. Bridle, *Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition*, in F. Fogelman-Soulié and J. Hérault (eds.), *Neuro-computing: Algorithms, Architectures*, Springer-Verlag, New York, 1989.

BUSCEMA, 1994: M. Buscema, *Squashing Theory. Modello a Reti Neurali per la Previsione dei Sistemi Complessi*, Collana Semeion, Armando Editore, Rome [Squashing Theory: A Neural Network Model for Prediction of Complex Systems, Semeion Collection by Armando Publisher].

BUSCEMA, 1995: M. Buscema, *Self-Reflexive Networks. Theory, Topology, Applications*, in Quality & Quantity, Kluver Academic Publishers, Dordrecht, The Netherland, Vol. 29, n. 4, pp. 339–403, November 1995.

BUSCEMA, 1996a: M. Buscema, *SQUASH. Shell for Program Feed Forward Neural Networks*, Semeion Software n. 5, Rome (1992–1996).

BUSCEMA, 1996b: M. Buscema, *MAPS. Shell for Program Self-Reflexive Neural Networks*, Semeion Software n. 7, Rome (1992–1996).

CARPENTER, 1991: G. A. Carpenter and S. Grossberg, *Pattern Recognition by Self-Organizing Neural Network*, The MIT Press, Cambridge, MA, 1991.

JACOBS, 1991: R. A. Jacobs, M. I. Jordan, S. P. Nowlan, and G. E. Hinton, *Adaptive Mixtures of Local Experts*, Neural Computation, 3, pp. 79-87.

JORDAN, 1992: M. I. Jordan and R. A. Jacobs, *Hierarchies of Adaptive Experts*, Advances in Neural Information Processing Systems, 4, pp. 985–992.

KOHONEN, 1995: T. Kohonen, *Self-Organizing Maps*, Springer Verlag, Berlin, Heidelberg, 1995.

KOSKO, 1992: B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prectice Hall, Englewood Cliffs, N.J., 1992.

LIU, 1992a: Q. Liu, S. Hirono, and I. Moriguchi, *Application of Functional-Link Net in QSAR. 1. QSAR for Activity Data Given by Continuous Variate*, in Quant. Struct. -Act. Relat. //, 135–141, School of Pharmaceutical Sciences, Kitasato University, Shirokane, Minato-ku, Tokyo 108, Japan.

LIU, 1992b: Q. Liu, S. Hirono, and I. Moriguchi, *Application of Functional-Link Net in QSAR. 2. QSAR for Activity Data Given by Continuous Variate*, in Quant. Struct. -Act. Relat. //, 318–324, School of Pharmaceutical Sciences, Kitasato University, Shirokane, Minato-ku, Tokyo 108, Japan.

RIPLEY, 1996: B. D. Ripley, *Pattern Recognition and Neural Networks*, University of Oxford, Cambridge University Press, Cambridge, MA, 1996.

RUMELHART, 1986: D. E. Rumelhart and J. L. McClelland (eds.), *Parallel Distributed Processing*. *Explorations in the Microstructure of Cognition*, Vol.1: *Foundation*, Vol. 2: *Psychological and Logical Models*, The MIT Press, Cambridge, MA, 1986.

THE AUTHOR

Massimo Buscema, Dr., computer scientist, expert in artificial neural networks and adaptive systems. He is the founder and Director of the Semeion Research Center of Sciences of Communication, Rome, Italy. He is formerly Professor of Science of Communication, University of Charleston, Charleston, West Virginia, USA, and Professor of Computer Science and Linguistics at the State University of Perugia, Perugia, Italy. He is a member of the Editorial Board of *Substance Use & Misuse*, a faculty member of the *Middle Eastern Summer Institute on Drug Use*, co-editor of the Monograph Series *Uncertainty* and co-creator and co-director of *The Mediterranean Institute*. He is consultant of *Scuola Tributaria Vanoni* (Ministry of Finance), *Ufficio Italiano Cambi* (Bank of Italy), *ENEA* (Public Oil Company), *Sopin Group* (Computer Science Corporation) and many Italian Regions. He has published books and articles; among them: *Prevention and Dissuasion*, EGA, Turin, 1986; *Expert Systems and Complex Systems*, Semeion, Rome, 1987; *The Brain within the Brain*, Semeion, Rome, 1989; *The Sonda Project: Prevention from self and heterodestructive behaviors*, Semeion, Rome, 1992; *Gesturing Test: A Model of Qualitative Ergonomics* ATA, Bologna, 1992; *The MQ Model: Neural Networks and Interpersonal Perception*, Armando, Rome, 1993; *Squashing Theory: A Neural Networks Model for Prediction of Complex Systems*, Armando, Rome, 1994; *Self-Reflexive Networks: Theory, Topology, Application*, Quality & Quantity, 29, Kluver Academic Publishers, Dordrecht, Holland; *Idee da Buttare*, Edizioni Sonda, Turin, 1994; *Artificial Neural Networks and Finance*, Armando, Rome, 1997; *A General Presentation of Artificial Neural Networks*, in *Substance Use & Misuse*, 32(1), Marcel Dekker, New York, 1997; *The Sonda Project: Prevention, Prediction and Psychological Disorder*, in *Substance Use & Misuse*, 32(9), Marcel Dekker, New York, 1997.