# An Insight into Some Aspects of Rough-Neurocomputing

Marcin Szczuka
Warsaw University, Poland
szczuka@mimuw.edu.pl

# Contents

- Neurocomputing – introductory remarks
- Rough Sets and Artificial Neural Networks
  - Coupling RS with ANNs
  - Rough neurons
- Networks of approximations – Rough-Neurocomputing
- Rough-Neurocomputing revisited
  - Concepts, probabilities, decision distributions
  - Feedforward concept networks

# What is Neurocomputing anyway?

- Field of research that deals with behaviour of artificial neurons and artificial neural networks.
- Technique used in approximation and classification tasks.
- Title of journal published by Elsevier.
- ….

  or
- A computational paradigm that makes use of simple processing units bound together by internal connections in order to achieve higher-level results.

# Our idea of neurocomputing

A computing paradigm that:

- Uses simple processing units – neurons.
- Connects processing units to make exchange of information possible.
- Adopts to requirements by strenghtening or weakening connections between processing units.
- Achieves desired goals by adaptation (learning) with use of algorithmically effective procedures.
- Provides robust, noise-tolerant and flexible results.

# Rough Sets
# and
# Artificial Neural Networks

# What do they bring to the table?

**Rough Sets**:

- Reduction
- Approximations (in RS sense)
- Classification – especially decision rules

**Artificial Neural Networks**:

- Learning and adaptability
- Robustness and flexibility, tolerance to noise
- Approximation (in numerical sense)
- Natural approach to continuous data classification (e.g., signals)

# From RS to ANN

Rough set techniques used for reduction, feature selection and preprocessing of training data for ANN. One of first ideas joining RS and ANNs, still in circulation today.

M. Szczuka (1998). *Rough Sets and Artificial Neural Networks.* In: L. Polkowski and A. Skowron (eds.), Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems, Physica-Verlag, Heidelberg, pp. 449-470.

J. F. Peters and M. S. Szczuka. *Rough neurocomputing: A survey of basic models of neurocomputation.* In James J. Alpigini, James F. Peters, Andrzej Skowron, and Ning Zhong, editors, Third International Conference on Rough Sets and Current Trends in Computing RSCTC, volume 2475 of Lecture Notes in Artificial Intelligence, pages 308-315, Malvern, PA, October 14-16 2002. Springer-Verlag.

# ANNs for RS

Using learning/adaptation abilities of a neural network to solve some of RS problems.

Supplementing rule-based RS classifiers with a neural network that solves conflicts between rules i.e., provides voting mechanism.


M. Szczuka, *Refining classifiers with neural networks*, International Journal of Intelligent Systems 16 (2001) pp.39-55.

M. Szczuka, P. Wojdyłło, *Neuro-wavelet classifiers for EEG signals based on rough set methods*, Neurocomputing 36 (2001) pp.103-122.

# Rough Neurons

Instead of processing pure signal the neuron caters upper and lower approximation of the incoming information.

Lingras, P.J. 1998. Comparison of neofuzzy and rough neural networks, Information Sciences: an International Journal, Vol. 110, pp. 207-215.

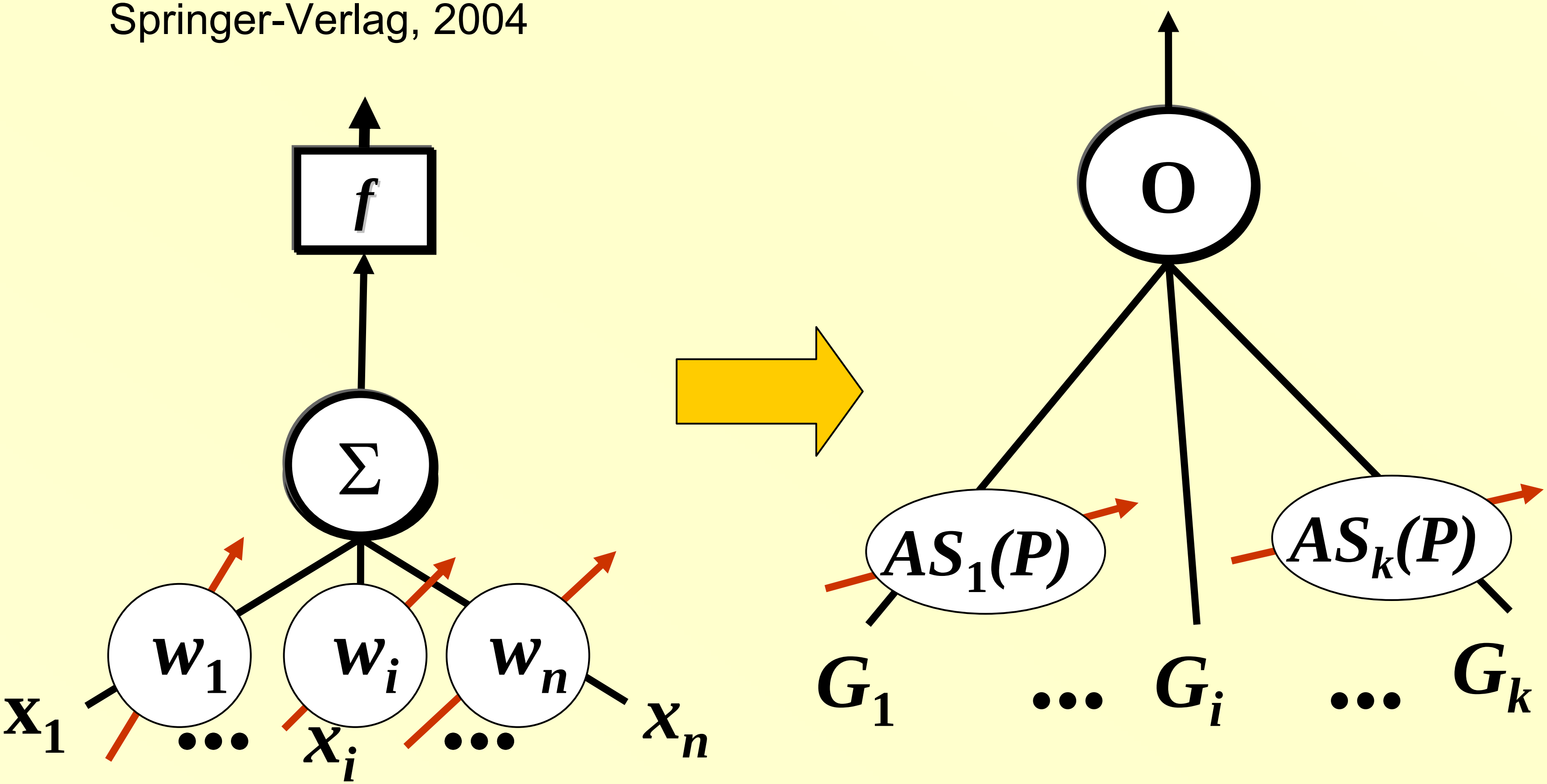
# Rough-Neurocomputing

# Rough-Neural Computing vs. Rough-Neurocomputing

*Rough-Neural Computing: Techniques for Computing with Words*

S.K. Pal, L. Polkowski, A. Skowron (eds.)

Springer-Verlag, 2004



# Rough-Neurocomputing Revisited

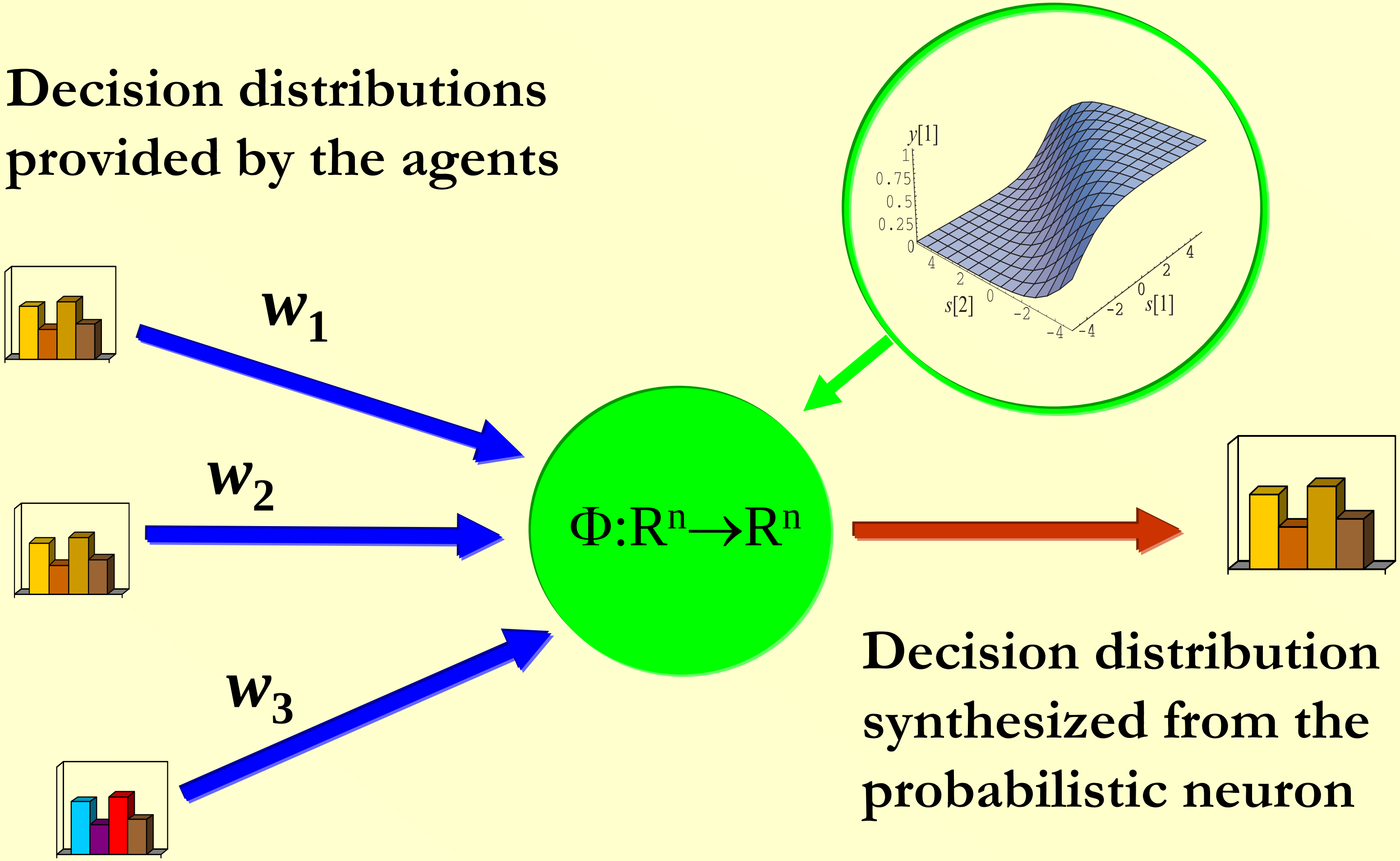
Feedforward Concept Networks

# Classification of complex objects

- Real-world concepts are often compound of parts (sub-concepts)
- Sub-concepts create (unknown) structure
- There may be nontrivial dependencies between sub-concepts
- Sub-concepts can be constructed separately
- Knowledge about a final concept may be distributed among many classifying agents

# The concept synthesis – example

Decision distributions provided by the agents

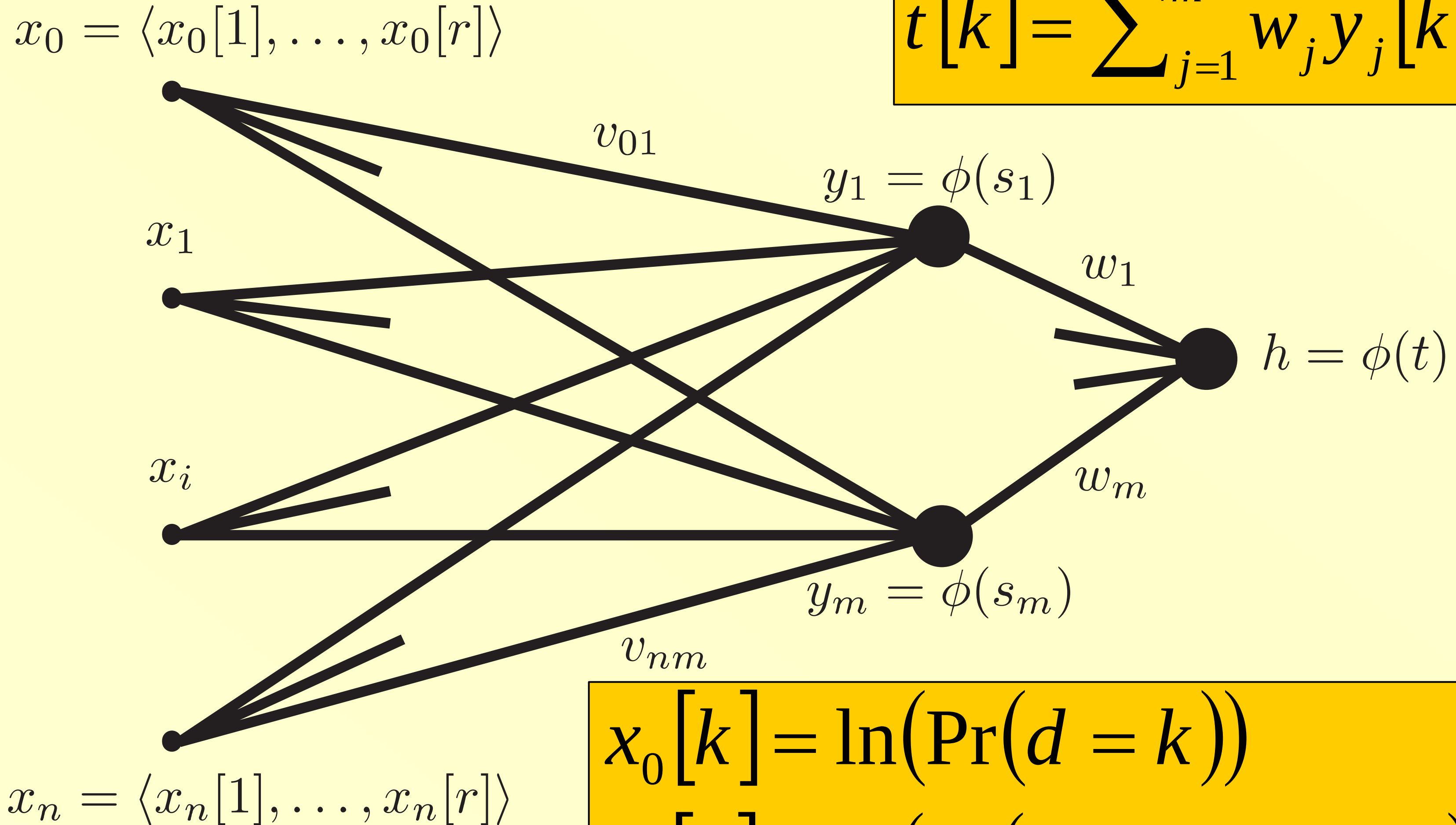



Decision distribution synthesized from the probabilistic neuron

# Probabilistic neural network

$$s_j[k] = \sum_{i=0}^{n} v_{ij} x_i[k]$$

$$t[k] = \sum_{j=1}^{m} w_j y_j[k]$$



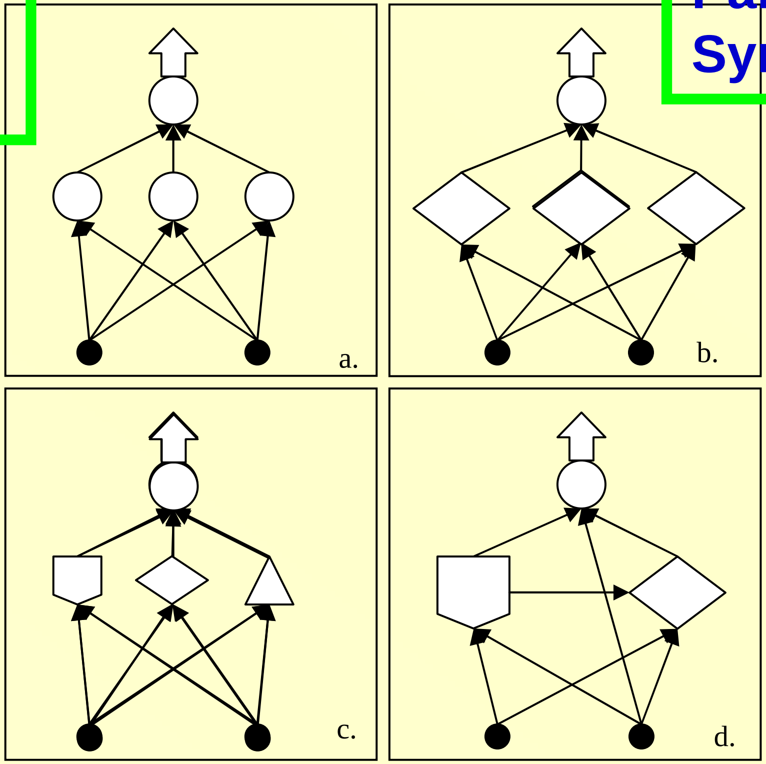

$$x_0[k] = \ln(\Pr(d = k))$$

$$x_i[k] = \ln(\Pr(a_i = v_i / d = k))$$

# Types of structures

Homogenous
Synchronous

Partially heterogenous
Synchronous




Fully heterogenous
Synchronous

Heterogenous
Asynchronous

# Concepts

# Concepts and granules (1)

- A *concept* is an element drawn from a parameterized *concept space*
- By a proper setting of parameters we choose the right concept
- We do not demand that all concepts come from the same space

# Concepts and granules (2)

- Our informal definition of a concept space can be referred to the notion of an *information granule system* $S=(G,R,Sem)$
- $G$ is a set of parameterized formulas called *information granules*
- $R$ is a parameterized relation structure
- $Sem$ is the semantics of $G$ in $R$

# Concepts and granules (3)

- In our approach, we focus on the concept parameterization and, especially, on the ability of parameterized construction of the new concepts from the others
- Our understanding of a concept space can be regarded as equivalent to an information granule system
- The terms „*concept*" and „*granule*" may be used exchangeably

# Weighted compound concepts (1)

- By a *weighted compound concept space C* we mean a space of collections of *sub-concepts* from some *sub-concept space S*, labelled with the *concept parameters* from a given space *V*:

$$C = \bigcup_{X \subseteq S} \{ ( s, v_s ) : s \in X , v_s \in V \}$$

# Weighted compound concepts (2)

- For a given compound concept

$$c = \{ (s, v_s) : s \in X_c, v_s \in V \}$$

  the subset $X_c \subseteq S$ is the *range* of c
- Parameters $v_s \in V$ reflect relative importance of sub-concepts $s \in X_c$ within c

# Example 1

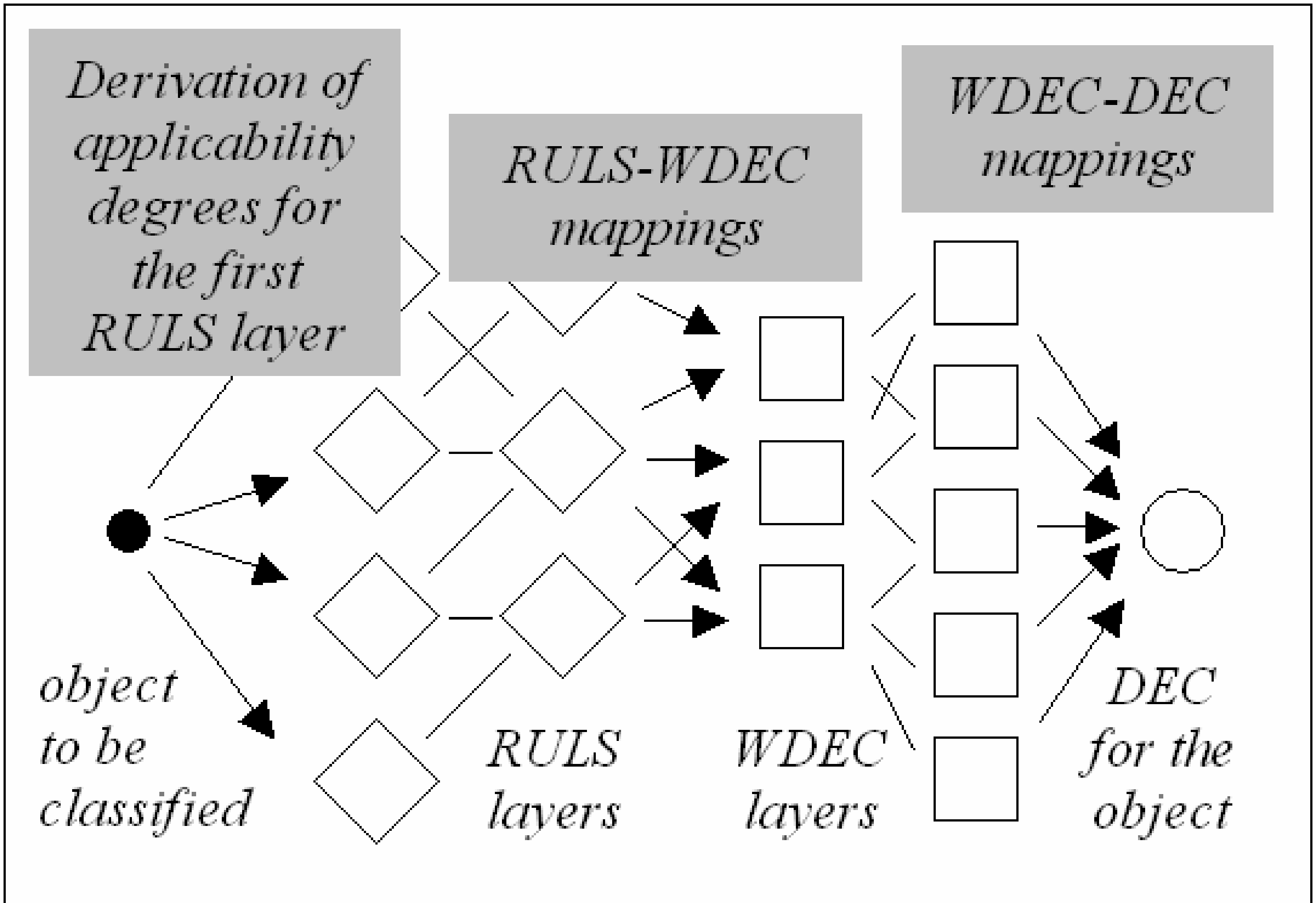
- Let us consider the ensemble of classifiers working on the same data
- Answer of each classifier: the set of decision values and corresponding belief coefficients
- *DEC* – the set of decision values
- *WDEC* – the family of sets containing decision values and belief coefficients

# Example 2

- Let us consider the rule based system
- *DESC* – the family of rule descriptions
- *RULS* – the family of decision rule sets
- Every decision rule is compound of:
  - its description ( in *DESC* )
  - its decision characteristics ( in *WDEC* )
  - its importance ( $V = \mathrm{R}$ )

# Concept Networks

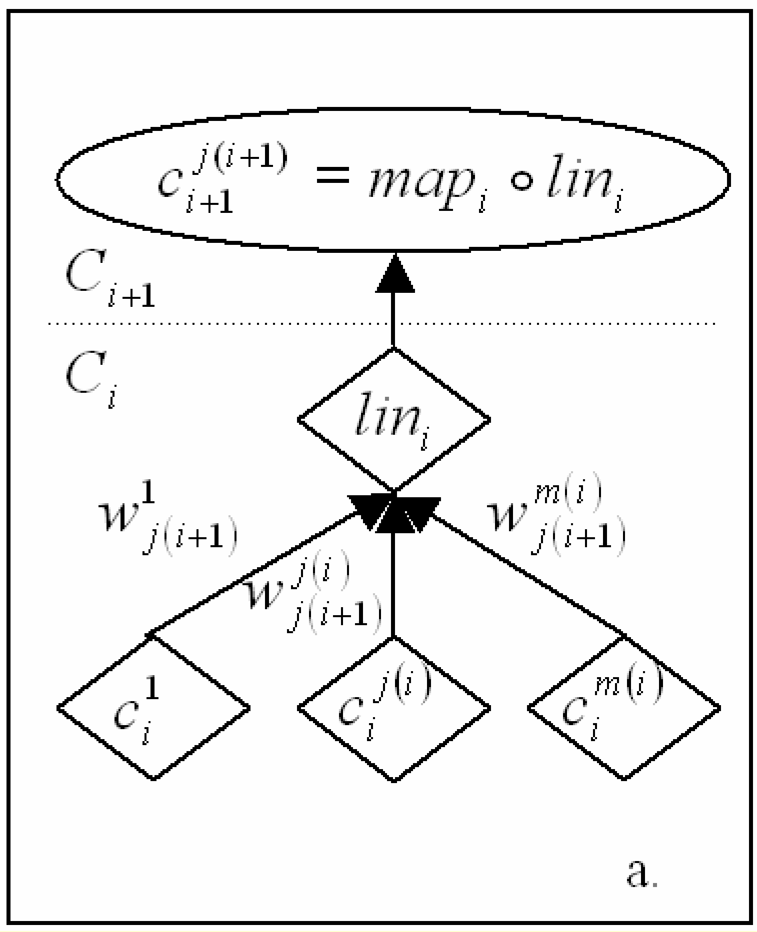
# Concept hierarchy *RULS-WDEC*



# Neural concept scheme

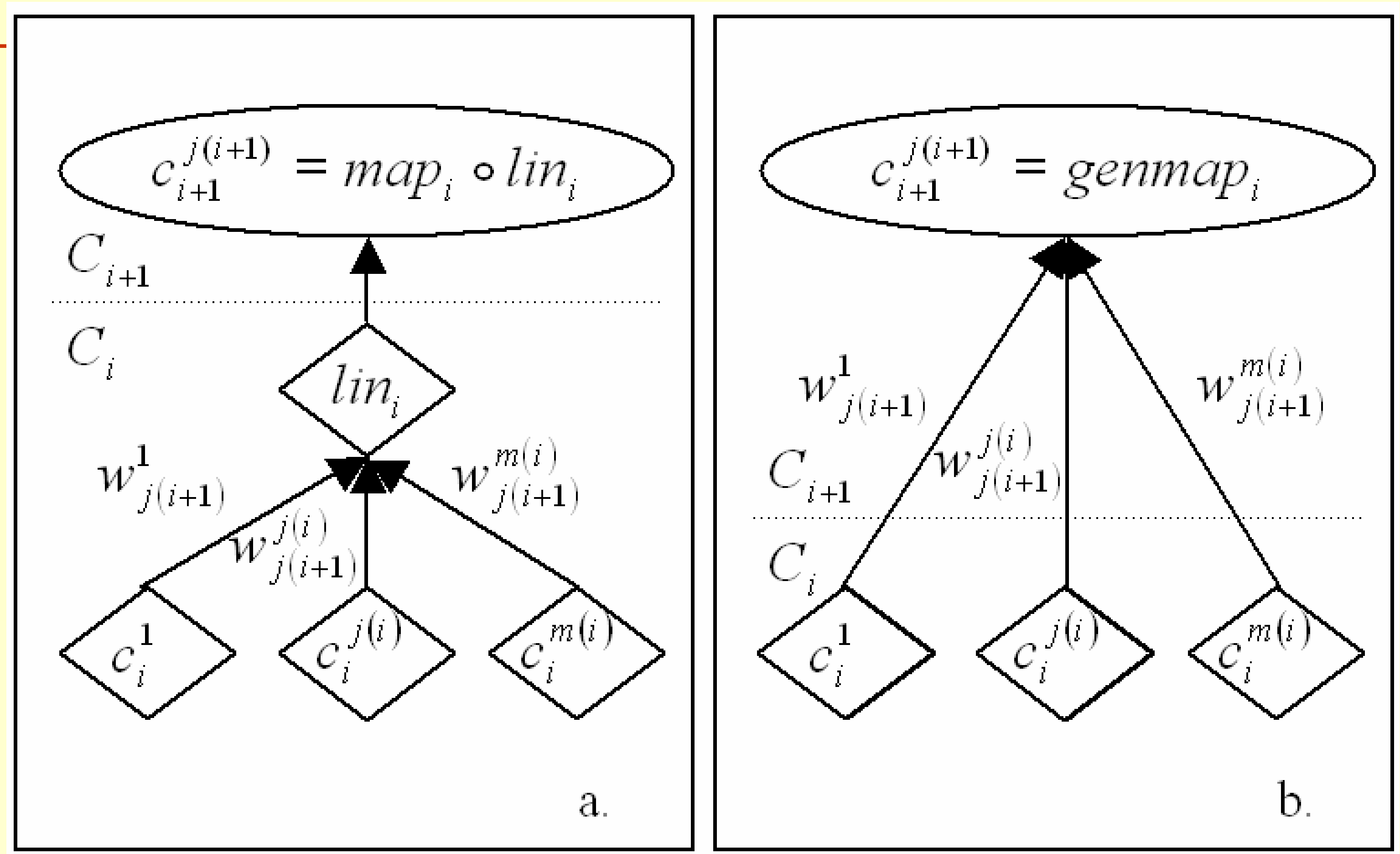
- **C** = { $C_1$ , ... , $C_n$ , $C$ } is a collection of concept spaces ( $C$ is the target space )
- **MAP** = { $map_i : C_i \to C_{i+1}$ } is a collection of *concept mappings*, which are the functions linking the consecutive concept spaces
- **LIN** = { $lin_i : \mathrm{P}(C_i \times W_i) \to C_{i+1}$ } is a collection of *generalized linear combinations* with respect to $W_i$
- **ACT** = { $act_i : C_i \to C_i$ } is a collection of activation functions, which can be used to relate the inputs to the outputs within each i-th layer of a network

# Two ways of concepts' combination



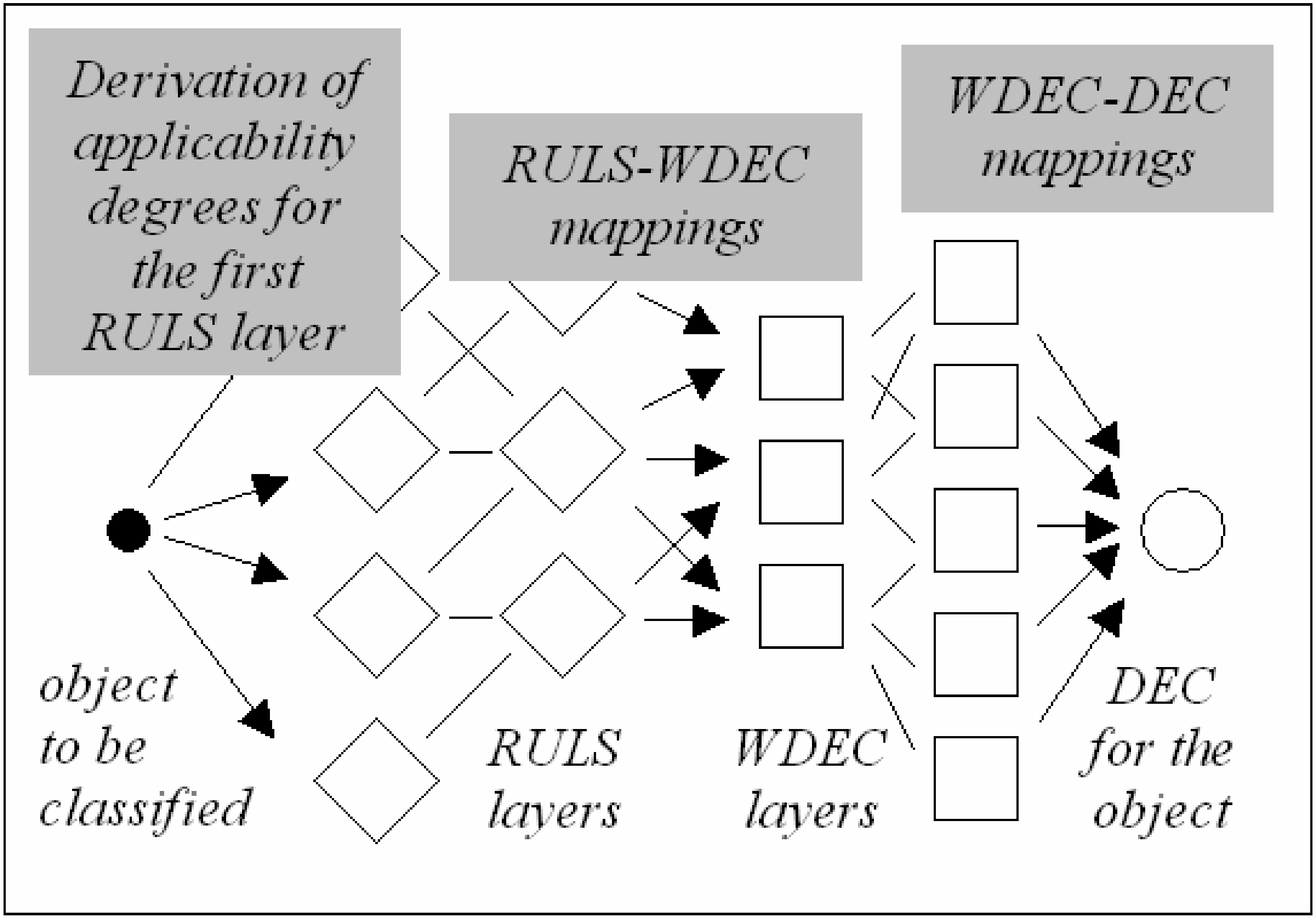

We may either apply generalized linear combination inside space $C_i$ or use **generalized** (weighted) **concept mapping**

# Two ways of concepts' combination



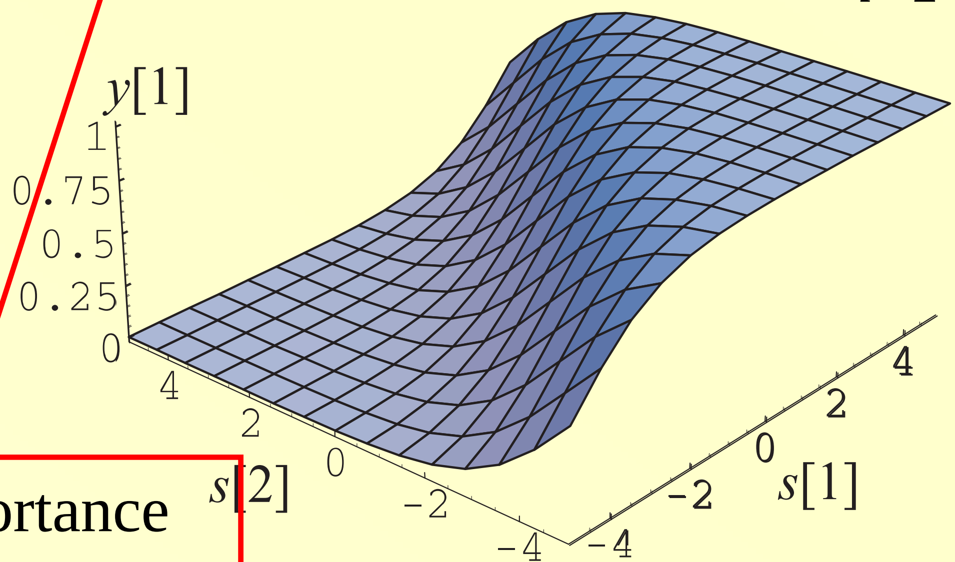

We may either apply generalized linear combination inside space $C_i$ or use **generalized** (weighted) **concept mapping**

# Concept hierarchy *RULS-WDEC*



# Activation functions – example

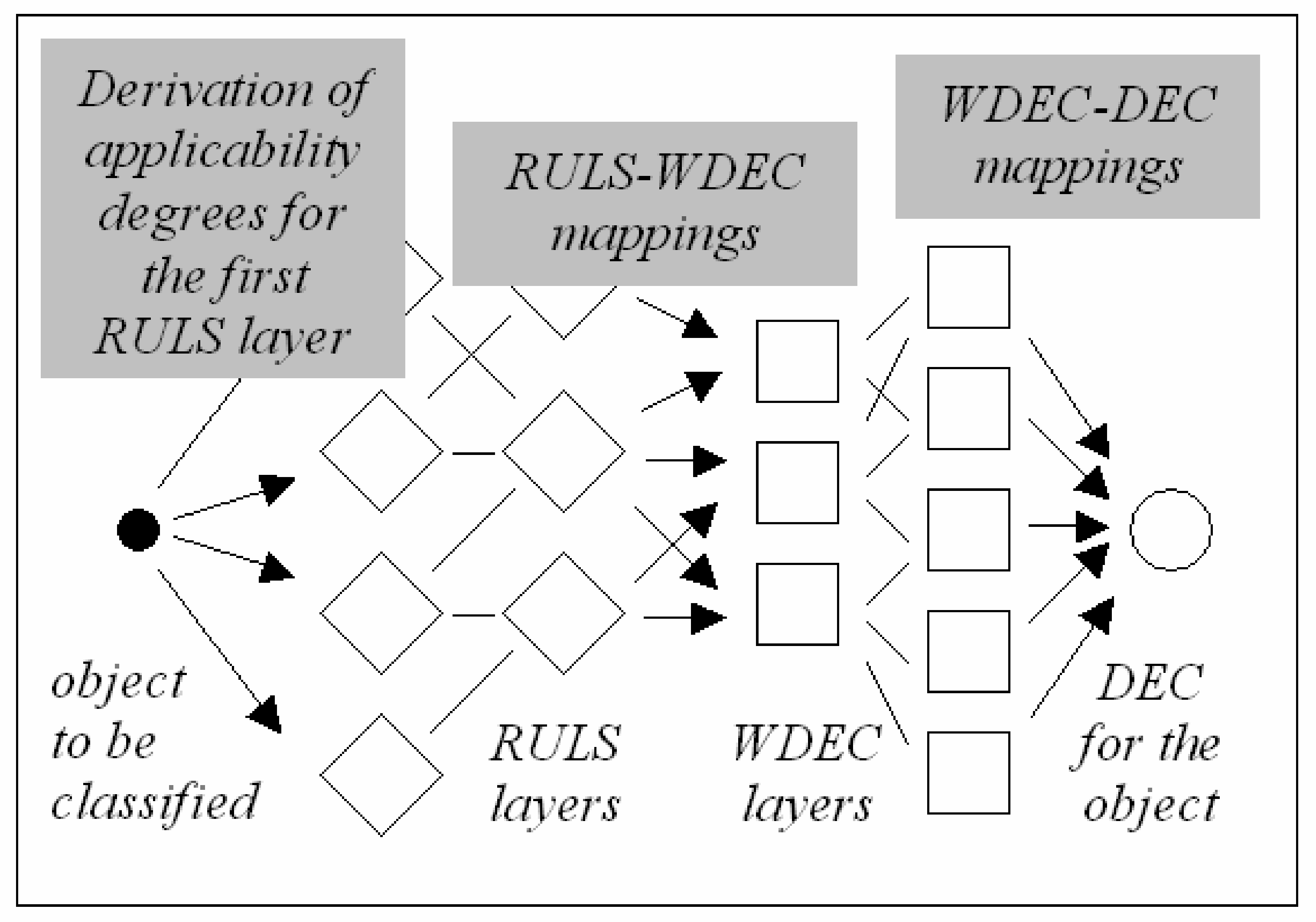
$$y[k] = \phi_\alpha(s)[k] = \frac{e^{\alpha s[k]}}{\sum_{l=1}^{r} e^{\alpha s[l]}}$$




Importance of the k-th part in the concept Φ(s)

Importance of the k-th part in the concept s

# Concept hierarchy *RULS-WDEC*



# Network error – example

- Distance between distributions h and d

$$dist(h,d)=\sqrt{\frac{1}{2}\sum_{k=1}^{r}\left(h[k]-d[k]\right)^2}$$

  is maximally equal to 1
- It equals 1 only if h and d correspond to different simplex vertices

# Derivative error in backpropagation

$$\frac{\partial E\left(w_1,\ldots,w_m\right)}{\partial w_j} = \left(h-d\right) \circ D\phi\left(t\right) \circ y_j^T$$

$$\frac{\partial E\left(v_{11},\ldots,v_{nm}\right)}{\partial v_{ij}} = \left(h-d\right) \circ D\phi\left(t\right) \circ w_j D\phi\left(s_j\right) \circ x_i^T$$

# Derivatives

$$\phi_\alpha(s)[k] = \frac{e^{\alpha\, s[k]}}{\sum_{l=1}^{r} e^{\alpha\, s[l]}} \quad \Rightarrow \quad D\phi_\alpha(s) =$$

$$\alpha \cdot \begin{bmatrix} \phi_\alpha(s)[1]\cdot(1-\phi_\alpha(s)[1]) & \cdots & -\phi_\alpha(s)[1]\cdot\phi_\alpha(s)[r] \\ \vdots & \ddots & \vdots \\ -\phi_\alpha(s)[r]\cdot\phi_\alpha(s)[1] & \cdots & \phi_\alpha(s)[r]\cdot(1-\phi_\alpha(s)[r]) \end{bmatrix}$$

# References

Ślęzak, D.; Szczuka, M.; Wróblewski, J.
*Feedforward concept networks*
In: Monitoring, Security, and Rescue Techniques in Multiagent Systems, Barbara Dunin-Kęplicz, Andrzej Jankowski, Andrzej Skowron, Marcin Szczuka (eds.) Advances in Soft Computing, Springer-Verlag, Berlin, Heidelberg (2005), pp.281-292.

Ślęzak, D.; Szczuka, M.; Wróblewski, J.
*Harnessing classifier networks - towards hierarchical concept construction,*
S. Tsumoto et al. (Eds.): Rough sets current trends in computing, RSCTC 2004, Lecture Notes in Artificial Intelligence **3066** (2004), Springer, pp.554-560.

Szczuka, M.; Ślęzak, D.; Wróblewski, J.
*Constructing extensions of bayesian classifiers with use of normalizing neural networks,*
In: N. Zhong, Z. Raś, S. Ohsuga (Eds.): International Symposium on Methodologies for Intelligent Systems ISMIS, Maebashi, Japan, October 28-31, 2003, Lecture Notes in Artificial Intelligence **2871**, (2003), Springer.

Ślęzak, D.; Wróblewski, J.; Szczuka, M.
*Neural network architecture for synthesis of the probabilistic rule based classfiers,*
In: Proceedings of the Workshop on Rough Sets in Knowledge Discovery and Soft Computing at ETAPS□2003, April 12-13, Warsaw University, 2003, pp.250-262; electronic version in Electronic Notes in Computer Science **82**(4), Elsevier 2003

# Thank you!