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 strengthening 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.


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.


Rough Neurons

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

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

\[ \Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n \]

Decision distribution synthesized from the probabilistic neuron
Probabilistic neural network

\[ s_j[k] = \sum_{i=0}^{n} v_{ij} x_i[k] \]
\[ t[k] = \sum_{j=1}^{m} w_j y_j[k] \]

\[ x_0 = \langle x_0[1], \ldots, x_0[r] \rangle \]

\[ x_n = \langle x_n[1], \ldots, x_n[r] \rangle \]

\[ x_0[k] = \ln(\Pr(d = k)) \]
\[ x_i[k] = \ln(\Pr(a_i = v_i / d = k)) \]
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.
Our informal definition of a concept space can be referred to the notion of an *information granule* system \( S = (G, R, \text{Sem}) \):

- \( G \) is a set of parameterized formulas called *information granules*.
- \( R \) is a parameterized relation structure.
- \( \text{Sem} \) is the semantics of \( G \) in \( R \).
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 ($V = R$)
Concept Networks
Concept hierarchy **RULS-WDEC**

![Diagram of concept hierarchy](image)
Neural concept scheme

- $\mathbf{C} = \{ C_1, \ldots, C_n, C \}$ is a collection of concept spaces (C is the target space)
- $\mathbf{MAP} = \{ map_i : C_i \rightarrow C_{i+1} \}$ is a collection of concept mappings, which are the functions linking the consecutive concept spaces
- $\mathbf{LIN} = \{ lin_i : P(C_i \times W_i) \rightarrow C_{i+1} \}$ is a collection of generalized linear combinations with respect to $W_i$
- $\mathbf{ACT} = \{ act_i : C_i \rightarrow C_i \}$ 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 \textit{generalized} (weighted) \textit{concept mapping}.
Concept hierarchy **RULS-WDEC**

- **Derivation of applicability degrees for the first RULS layer**
- **RULS-WDEC mappings**
- **WDEC-DEC mappings**

**Object to be classified**

- **RULS layers**
- **WDEC layers**

**DEC for the object**
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 \( \Phi(s) \)
Concept hierarchy **RULS-WDEC**

- Derivation of applicability degrees for the first RULS layer
- RULS-WDEC mappings
- WDEC-DEC mappings
- Object to be classified
- RULS layers
- WDEC layers
- DEC for the object
Network error – example

- Distance between distributions $h$ and $d$

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

is maximally equal to 1

- It equals 1 only if $h$ and $d$ correspond to different simplex vertices
Derivative error in backpropagation

\[
\frac{\partial E(w_1, \ldots, w_m)}{\partial w_j} = (h - d) \odot D\phi(t) \odot y_j^T
\]

\[
\frac{\partial E(v_{11}, \ldots, v_{nm})}{\partial v_{ij}} = (h - d) \odot D\phi(t) \odot w_j D\phi(s_j) \odot x_i^T
\]
Derivatives

\[
\phi_\alpha(s)[k] = \frac{e^{\alpha s[k]}}{\sum_{l=1}^{r} e^{\alpha s[l]}} \implies D\phi_\alpha(s) = \\
\alpha \cdot \begin{bmatrix}
\phi_\alpha(s)[1] \cdot (1 - \phi_\alpha(s)[1]) & \cdots & -\phi_\alpha(s)[1] \cdot \phi_\alpha(s)[r] \\
\vdots & \ddots & \vdots \\
-\phi_\alpha(s)[r] \cdot \phi_\alpha(s)[1] & \cdots & \phi_\alpha(s)[r] \cdot (1 - \phi_\alpha(s)[r])
\end{bmatrix}
\]
References

Ślężak, D.; Szczuka, M.; Wróblewski, J.
*Feedforward concept networks*

Ślężak, D.; Szczuka, M.; Wróblewski, J.
*Harnessing classifier networks - towards hierarchical concept construction*,

Szczuka, M.; Ślężak, D.; Wróblewski, J.
*Constructing extensions of bayesian classifiers with use of normalizing neural networks,*

Ślężak, D.; Wróblewski, J.; Szczuka, M.
*Neural network architecture for synthesis of the probabilistic rule based classifiers,*
Thank you!

************