

Quo vadis, computational intelligence?

Włodzisław Duch¹ and Jacek Mańdziuk²

¹ Department of Informatics, Nicholas Copernicus University,
ul. Grudziądzka 5, 87-100 Toruń, Poland

² Faculty of Mathematics and Information Science, Warsaw University of Technology,
Plac Politechniki 1, 00-661 Warsaw, Poland.

Abstract. What are the most important problems of computational intelligence? A sketch of the road to intelligent systems is presented. Several experts have made interesting comments on the most challenging problems.

1 Introduction.

In the introduction to the “MIT Encyclopedia of Cognitive Sciences” M. Jordan and S. Russell [33] used the term “computational intelligence” to cover two views of artificial intelligence (AI): engineering and empirical science. Traditional AI started as an engineering discipline concerned with the creation of intelligent machines. Computational modeling of human intelligence is an empirical science. Both are based on computations.

Artificial Intelligence (AI) has established its identity quite early, during the Dartmouth conference in 1956 [6]. It had clearly defined goals, exemplified by great early projects, such as the General Problem Solver of Simon and Newell. There are many definitions of AI [48,64], for example: “... the science of making machines do things that would require intelligence if done by humans” (Marvin Minsky), “The study of how to make computers do things at which, at the moment, people are better” [48]. In essence AI tries to solve problems for which effective algorithms do not exist, using knowledge-based methods.

In the 1970-ties AI has contributed to the development of cognitive science and to the goal of creating “unified theories of cognition”, as Allen Newell called it. Ambitious theories of high cognitive functions were formalized by John Anderson in his Act* theory [3], and by Newell and his collaborators in the Soar theory [43]. Both were very successful and led to many practical (and commercial) applications. In the last decade intelligent agents become the focus of AI, entities that can perceive and act in a rational goal directed way to achieve some objectives.

Machine learning has been important from the beginning in AI. Samuel’s checker-playing system (1959) learned to play far superior checkers than its creator. Although initially research on perceptrons has developed as a part of AI in the late 1950-ties machine learning became preoccupied with inductive, rule based knowledge [50]. AI development has always been predominately concerned with high-level cognition, where symbolic models are appropriate.

In 1973 the book of Duda and Hart on pattern recognition appeared [18]. The authors wrote that “pattern recognition might appear to be a rather specialized topic”. It is obviously a very broad topic now, including good part of neural networks research [5]. In 1982 Hopfield network [26], in 1986 the backpropagation algorithm [49], and a year later the PDP books [7] brought the neural network field to the center of attention. Since that time the field of neural computing has been growing rapidly in many directions and became very popular in the early 90-ties. Computational neuroscience, connectionist systems in psychology and neural networks for data analysis are very different branches with rather different goals. The last of these branches gained solid foundations in statistics and Bayesian theory of learning [5,25].

Soft computing conferences started to draw people from neural, fuzzy sets and evolutionary algorithms communities. Applications of these methods overlap with those dealt with by pattern recognition, AI and optimization communities.

Computational Intelligence (CI) is used as a name to cover many existing branches of science. This name is used sometimes to replace artificial intelligence, both by book authors [47] and some journals (for example, “Computational Intelligence. An International Journal”, by Blackwell Publishers). There are several computational intelligence journals dedicated to the theory and applications of artificial neural networks, fuzzy systems, evolutionary computation and hybrid systems. In our opinion it should be used to cover all branches of science and engineering that are concerned with understanding and implementing functions for which effective algorithms do not exist. From this point of view some areas of AI and a good part of pattern recognition, image analysis and computational neuroscience are subfields of CI.

What is the ultimate goal of computational intelligence and what are the short-term and the long term challenges to the field? What is it trying to achieve? Without setting up clear goals and yardsticks to measure progress on the way, without having a clear sense of direction many efforts will end up nowhere, going in circles and repeating the same problems.

We hope that this paper will start a discussion about the future of CI that should clarify some of the issues involved. First we shall make some speculations about the goals of computational intelligence, think how to reach them, and raise some questions worth answering. Then we will write about some challenges. We have asked several experts what they consider to be the greatest challenges in their field. Finally some conclusions will be given.

2 The ultimate goals of CI.

From the perspective of cognitive sciences artificial intelligence is concerned with high level cognition, dealing with such problems as understanding of language, problem solving, reasoning, planning and knowledge engineering at the symbolic level. Knowledge has complex structure, the main problems are combinatorial in nature and their solution requires heuristic search techniques. Learning

is used to gain knowledge that expert systems may employ for reasoning, and is frequently based on logic.

Other branches of computational intelligence are concerned with lower level problems, are more on the pattern recognition side, closer to perception, are at the subsymbolic level. Complex knowledge structures do not play important role, most methods work in fixed dimensional feature spaces. Combinatorial character of problems and knowledge-based heuristic search are not an issue. Numerical methods are used more often than discrete mathematics.

The ultimate goal of AI is to create a program that would pass the Turing test, that is would understand human language and be able to think in a similar way to humans. The ultimate AI project is perhaps CYC, a super-expert system with over a million of logical assertions describing all aspects of the world.

The ultimate goal of other CI branches may be to build an artificial rat (this was the conclusion of a discussion panel on the challenges to CI in the XXI century, at the World Congress on Computational Intelligence in Anchorage, Alaska, in 1998). Problems involved in building an artificial animal that may survive in a hostile environment are rather different than problems related to the Turing test. Instead of symbolic knowledge problems related to perception, direction of attention, orientation, motor control and motor learning have to be solved. Behavioral intelligence that is embodied in the Cog project is perhaps the most ambitious project of this kind [1].

Each branch of CI has its natural areas of application and the overlap between them is sometimes small. For example, with very few exceptions AI experts are separated from communities belonging to other CI branches, and vice versa. Even neural networks and pattern recognition communities, despite a considerable overlap in applications, tend to be separated.

Is there a common ground where the two fields could meet? The ultimate challenge for CI seems to be a robot that combines high behavioral competence with human-level higher cognitive competencies. Building creative systems of such kind all branches of CI will be required, including symbolic AI and lower level pattern recognition methods. At the one end of the spectrum we have neurons, at the other brains and societies.

3 A roadmap to creative systems.

The brain is not a homogenous, huge neural network, but has quite specific modular and hierarchical structure. Neural network models are inspired by processes at a low level of this hierarchy, while symbolic AI works at the highest level. Little work has been devoted to the description and understanding of intermediate levels, although investigation of connections between them can be quite fruitful [9].

Below we have sketched a roadmap from the level of single neurons to the highest level of creative societies of brains, presenting some speculations and research directions that seem unexplored. Cooperation of individual elements that have some local knowledge leads to emergence of a higher-order unit that should

be regarded at its own footing. The same principles may operate at different scales of complexity. A major challenge for CI is to create models and learn how to scale up systems to reach higher level.

3.1 Threshold neurons and perceptrons

Neurons in simple perceptrons have only one parameter, the threshold for their activity, and the synaptic weights that determine their interactions. Combined together perceptrons create the popular multi-layer perceptron (MLP) networks that are quite powerful, able to learn any multidimensional mapping starting from examples of required input/output relations. Usually the network aspect is stressed while learning processes are discussed: the whole, with interacting elements, is bigger than its parts. Real biological networks involve a huge number of neurons with thousands of connections each. Instead of looking at the fixed architecture of neural network it may be better to imagine that synaptic connections define interactions between subsets of individual elements. Clusters of activity, or forming sub-networks, has been observed in networks of spiking neurons [24]. Similar effects have not been investigated in MLP networks.

Perceptron neural networks may be regarded as collections of primitive processing elements (PEs). Individual element do not understand the task the whole collection is faced with, but are able to adjust to the flow of information, performing local transformations of the incoming data and being criticized or corrected by other members of the team (i.e. network). Hebb principle provides reinforcement for PE, regulating the level of their activity in solving different cooperative problems. Backpropagation procedure provides another kind of critique of the activity of single neurons.

Some parameters are internal to the neural units (thresholds), while other parameters are shared, allowing for interactions between units during the learning procedure. Neural networks use many units (neurons) that cooperate to solve problems that are beyond the capabilities of a single unit. Interactions and local knowledge of a simple PEs determine the type of problems that networks of such elements may solve.

Networks are able to generalize what has been learned, creating a model of states of local environment they are embedded in. Generalization is not yet creativity, but is a step in the right direction.

3.2 Increasing complexity of internal PE states

Next step beyond the single parameter (threshold) describing internal state of a neuron is to add more internal parameters, allowing each PE to realize a bit more than a hyperplane discrimination. Perceptrons are not able to solve the famous connectedness and other problems posed by Minsky and Papert [39] as a challenge for neural networks. Adding more network layers does not help (see the second edition of [39]), the problem scales exponentially with the growing size of images. This problem may be solved with neural oscillator networks in a biologically

plausible way [61], but rather complex networks are required. Adding one additional internal parameter (phase) is sufficient to solve this problem [32]. What is the complexity class of problems that may be solved this way? Can all problems of finding topological invariants be solved? What can be gained by adding more parameters? The answers are not clear yet.

Computational neuroscience investigates models of cortical columns or Hebbian cell assemblies. Modular neural networks may be regarded as networks with super PEs that adapt to requirements of the complex environment. Instead of simple units with little internal knowledge and fixed relations (fixed architecture of MLP networks), more powerful PEs dynamically forming various configurations (virtual networks) should be used. More complex internal knowledge and interaction patterns of PEs are worth investigation.

The simplest extension of network processing elements that adds more internal parameters requires abandoning the sigmoidal neurons and using a more complex transfer functions. A Gaussian node in a Radial Basis Function network [5] has at least N internal parameters, defining position of the center of the function in N -dimensional space. Weights define the inverse of dispersions for each dimension, determining interaction with other network nodes through adaptation of parameters to the data flow. Although research efforts have been primarily devoted to improvement of neural training algorithms and architectures there are good reasons to think that transfer functions may significantly influence the rate of convergence, complexity of the network and the quality of solution it provides [16].

What do these more complex PEs represent? If their inputs are values of some features they model areas of the feature space that may be associated with some objects, frequently appearing input patterns. Recurrent neural networks, including networks of spiking neurons, are used as autoassociative memories that store prototype memories as attractors of network dynamics [2]. Basins of these attractors define areas of the feature space associated with each attractor. A single complex PE, or a combination of a few PEs, represent such areas directly, replacing a sub-network of simpler neurons.

This was the original motivation for the development of the Feature Space Mapping (FSM) networks [13,9]. Nodes of FSM networks use separable transfer functions $G(\mathbf{X})=\prod_i G_i(x_i)$, instead of radial functions (the only separable radial function is Gaussian). Their outputs model the probability of recognizing a particular combination of input features as some object. Each PE may be treated as a fuzzy prototype of an object, while each component $G_i(x_i)$ may be treated as a membership function for feature x_i . Thus FSM is a neurofuzzy system that allows for control of the decision borders around each prototype by modifying the internal parameters of the PEs (transfer functions). Precise control of basins of attractors in dynamical networks is usually impossible.

In contrast to MLP neural networks many types of functions with different internal parameterizations are used. First steps towards neural networks with heterogonous PEs were made [11,17,29]. Theoretically they should allow for discovery of an inductive bias in the data, selecting or adapting transfer functions to the data using minimal number of parameters. Creation of efficient algorithms for networks with heterogonous PEs is quite challenging task.

Each complex PE represents a module that adapts to the data flow adjusting its basin of influence in the feature space. Is this approximation sufficient to replace dynamical networks with spiking neurons or recurrent networks? What are the limitations? Many questions are still to be answered.

3.3 Increasing complexity of PE interactions

Rigorous transition from attractor networks to equivalent FSM networks may be based on fuzzy version of symbolic dynamics [4] or on the cell mapping method [28]. It should be possible to characterize not only the asymptotic properties of dynamical models, but also to provide simplified trajectories preserving sufficient information about basins of attractors and transition probabilities. This level of description is more detailed than the finite state automata, since each state is an object represented in the feature space.

Such models are a step from neural networks to networks representing low-level cognitive processes. They are tools to model processes taking place in feature spaces. FSM networks use clusterization-based procedures to create initial model of the input data and then learn by adaptation of parameters. Adding knowledge to the feature space is easy by creating, deleting and merging nodes of the network. FSM may work as associative memory, unsupervised learning, pattern completion system or a fuzzy inference system. Constraints on variables, such as arithmetic relations, or laws $Y=F(X_1, \dots, X_N)$ may be directly represented in feature spaces.

Although using complex PEs in networks adds internal degrees of freedom interactions between the nodes are still fixed by the network architecture. Even if nodes are added and deleted the initial feature space is fixed. An animal has a very large number of receptors and is able to pay attention to different combinations of sensory stimuli. Attractor networks are combinatorially productive, activating many combinations of neural modules. Feedforward networks, even with complex PEs, have fixed path of data flow. Although internal representations of PEs go beyond logical predicates they are not dynamic. Thus they are not able to model networks of modules that interact in a flexible way depending on the internal states of their modules.

One reason for changes in the internal states of cortical brain modules is due to the recent history (priming effects), another is due to changes in neuromodulation controlled by a rough recognition and emotional responses in the limbic areas. A simplified model of interacting modules should include the fact that all internal parameters should depend either directly on inputs $P(\mathbf{X})$, or indirectly on hidden parameters $P(H(\mathbf{X}))$ characterizing internal states of other modules. Each module should estimate how competent it is in a given situation and add its contribution to the interaction with other modules only if its competence is sufficient.

Recently this idea has been applied to create committees of competent classifiers [15]. A committee is a network of networks, or a network where each element has been replaced by a very complex PE, made from individual network. Outputs $O(\mathbf{X}; M_i)$ from all network modules (classifiers) M_i are combined together with weights W_i in the perceptron-like architecture. The weights of these combinations

are modulated (multiplied) by factors $F(\mathbf{X};M_i)$ that are small in the feature space areas where the model M_i makes many errors and large where it works well. Thus the effective weights depend on the current state of the network, $W_i(\mathbf{X}) = W_i F(\mathbf{X};M_i)$. This method may be used to create virtual subnetworks, with different effective path of information flow.

Modulation of the activity of modules is effective only if the information about the current state is distributed to all modules simultaneously. In the brain this role may be played by the working memory (cf. Newman and Baars [42]). The step from associations to sequential processing is usually modeled by recurrent networks. Here we have networks of modules adjusting their internal states (local knowledge that each module has learned) and their interactions (modulations of weights) to the requirements of the information flow through this system.

At this level systematic search processes may operate. In [13] we have shown that a complex problem requiring combinatorial approach may be solved quite easily by search processes that activate simple FSM modules. The electrical circuit example from the PDP book has been used [7] to demonstrate it. Each FSM module has learned qualitatively to analyze relations between the 3 variables, such as the Ohm's law $U=IR$ etc. The amazing result is [9] that almost any relation $\Delta A=f(\Delta B, \Delta C)$ representing changes of variables leads to the same objects in the feature space model. In the electric circuit example there are 7 variables and 5 laws that may be applied to this circuit. If values of some variables are fixed activity of the 5 FSM modules (each corresponding to a 3-term relation, and each identical) that are competent to add something new to the solution is sufficient to specify the behavior of the remaining variables.

Thus modular networks, such as the FSM model, may be used as powerful heuristics to solve problems requiring reasoning. The solution is found by systematic search, as in the reasoning systems, but each logical (search) step is supported by the intuitive knowledge of the whole system (level of activity of the competent modules). Such systems may be used for simple symbolic processing, but creating flexible modular networks of this type that could compete with experts systems is still a challenge.

3.4 Beyond the vector space concept

Feature space representation lies at the foundation of pattern recognition [18], but it is doubtful that it plays such an important role in the brain. Even at the level of visual perception similarity and discrimination may be sufficient to provide the information needed for visual exploration of the world [44]. At the higher cognitive levels, in the abstract reasoning or sentence analysis processes, vector spaces with fixed number of features are of little use. In such applications complex knowledge structures are created and manipulated by knowledge-based AI expert systems.

Although a general framework for processing of structural data, based on recurrent neural networks and hidden Markov models, has been introduced [20], it is rather difficult to implement and use. Perhaps a simpler approach could be sufficient. The two most common knowledge representation schemes in AI are based

on the state or the problem description [48,64]. The initial state and the goal state are also represented in the same way, the goal being usually a desired state, or a simple problem that has known solution. A set of operators is defined, transforming the initial object (state, problem), into the final object (goal). Each operator has some costs associated with its use. Solutions are represented by paths in the search graph. The best solution has lowest costs of transforming the initial object into the final object. An efficient algorithm to compute such distances may be based on dynamical programming [36].

Lowest costs of transformation that connect two complex objects are a measure of similarity of these objects. Mental operations behind evaluations of similarity are rather complex and are not modeled directly at this level. Similarity is sufficient for categorization and once it has been evaluated original features are not needed any more. At the level of perception sensory information is of course feature-based, but different types of higher-level features are created for different objects from the raw sensory impressions. At the higher cognitive level “intuitive thinking” is probably based on similarity evaluation that cannot be analyzed by logical rules. Crisp or fuzzy rules have limited expressive powers [12], prototype-based rules that evaluate similarity are more powerful alternative [14]. General framework for similarity-based systems includes most types of neural networks as special cases [10]. Pattern recognition methods that are based on similarity or dissimilarity matrices and do not require vector spaces based on features have been published (cf. [45]).

Another research direction may be inspired by Lev Goldfarb's criticism of the vector space as a foundation for inductive class generalization [23]. His system of evolving transformations tries to synthesize new operators for object transformation and similarity functions, allowing for evaluation of similarities between two objects that have quite different structure. This is necessary for example in chemistry, comparing molecules that have different structure although they belong to the same class (have the same activity or other high-level properties). In other words some kind of a measure of functional isomorphism or similarity (not necessarily corresponding to the structural one) is required in such applications.

3.5 Flexible incremental approaches

One of the fundamental impediments in building large, scalable learning systems based on neural networks is the problem of catastrophic forgetting. In order to alleviate this problem several ideas concerning both the network structures and the training algorithms have been introduced. The main approaches reported in the literature include modular networks [60,52,41], constructive approaches [19,21].

In modular networks the problem to be learned is divided into subproblems, each of which is learned independently by a separate module and then the solution for the whole problem is obtained as a proper combination of subproblem solutions. In constructive approaches the network starts off with a small number of nodes and its final structure is being built systematically by adding nodes and links whenever necessary. Both types of methods are well known in the community so their advantages and weak points will not be discussed here.

Other examples of flexible incremental approaches are the lifelong learning methods [57,58] in which learning new tasks becomes relatively easier when the number of already learned tasks increases.

One possible approach of that type is to start training procedure based on very simple, “atomic” problems. Structures developed while solving these atomic problems are frozen and consequently will not be obliterated in subsequent learning – only fine tuning would be permitted. These small atomic networks will serve as building blocks for solving more complicated problems – say level 1 problems. Larger structures (networks) obtained in the course of training for solving level 1 problems will serve as blocks of building even larger structures capable of solving more complex problems (level 2 ones), etc. Once in a while the whole system is tested based on the previously learned (or similar to them) atomic, level 1, level 2, etc. problems.

The above scheme can be viewed as an example of constructive approach, however - unlike in typical constructive approaches – it is postulated that the network starts off with the number of nodes and links a few times exceeding the number of actually required ones (i.e. “enough” spare nodes and links is available in the system). Hence the potential informational capacity of the system is available right from the beginning of the learning process (similarly to biological brains). After completion of the training process the nodes and links not involved in the problem representation are pruned unless the system is going to be exposed to another training task in future.

We have used similar to the above scheme to solving supervised classification problem. The training scheme called Incremental Class Learning (ICL) was successfully applied to unconstrained Handwritten Digit Recognition problem [34,35]. The system was trained digit by digit (class by class) and atomic features developed in the course of learning were frozen, and available in subsequent learning. These frozen features were shared among several classes. The ICL approach not only takes advantage of existing knowledge when learning a new problem, it also offers a large degree of immunity from the catastrophic interference problem. The ICL idea can possibly be extended to the case of multimodal systems performing several learning tasks where different tasks are characterized by different features. This would require adaptation of the above scheme to the case of multimodal feature spaces.

The above mentioned incremental learning methods are suitable for supervised, off-line classification tasks in which multi-pass procedure is acceptable. Alternative approaches – probably based on unsupervised training – must be used in problem domains requiring real-time learning ability. Ideally, large, scalable network structures should be suited to immediate, one pass incremental learning schemes. An examples – to some extent – of such fast trainable networks are Probabilistic Neural Networks [53] and General Regression Neural Networks [54] often applied to financial prediction problems [51]. However the cost of fast training ability is a tremendous increase of memory requirements since all training patterns must be memorized in the network. The other disadvantage is relatively slow response of the system in the testing phase. The search for efficient, fast incremental training

algorithms and suitable network architectures is regarded as one of the challenges in computational intelligence.

3.6 Evolution of networks

Another important research direction is changing from static (deterministic) networks into evolving (context dependent) ones. Possible approaches here include networks composed of nodes with local memory that process information step-wise, depending on the previous state(s). Evolving networks should be capable of adding and pruning nodes and links along with the learning process. Moreover, the internal knowledge representation should be based on redundant features sets as opposed to highly distributed representations. Non-determinism and context dependence can, for example, be achieved by using nodes equipped with simple fuzzy rules (stored in their local memories) that would allow for intelligent, non-deterministic information processing. These fuzzy rules should take into account both local parameters (e.g. the number of active incoming links, the degree of local weights density, etc.) as well as global ones (e.g. the average level of global activation – “temperature of the system”, global level of wiring of the system, etc.).

Knowledge representation should allow for off-line learning, which will be performed by separate parts of the systems – not involved in the very fast, on-line learning. The off-line learning should allow for fine tuning of the knowledge representation and also would be responsible for implementation of appropriate re-learning schemes. One of the possible approaches are the ECOS (Evolving COnectionist Systems) introduced by Kasabov [31], which implement off-line re-training schemes based on internal representation of the training examples in the system. Similar idea was also introduced in our paper [35] where the network was trained layer by layer and the upper layer was trained based on the feature representation developed in the lower layer.

Another claim concerning flexible learning algorithms and network structures is that structures of network modules as well as training methods should have some degree of fuzziness or randomness. Ideally, several network modules starting with exactly the same structure and internal parameters after some training period should diverge from one another though still stay functionally isomorphic. Some amount of randomness in the training procedure would allow for better generalization capabilities and higher flexibility of these modules.

Flexibility and hierarchy of information (knowledge) can be partly realized by the use of multidimensional links. Very simple associations will be represented by classical one dimensional links (form one neuron to another neuron). More complex facts will be represented by groups of links joint together and governed by sophisticated fuzzy rules taking into account context information. In other words multidimensional link will be a much more complex and powerful structure than the simple sum of all one dimensional links being their parts. A dimension of the link will be proportional to the degree of complexity of information it represents.

This idea has its roots in the design of associative memories where depending on the nature and complexity of stored associations suitable type of memory can be used (autoassociative, bidirectional or multidirectional).

3.7 Transition to symbolic processing

AI has concentrated on symbolic level of information. Problems related to perception, analysis of visual or auditory scenes, analysis of olfactory stimuli are solved by real brains working on spatiotemporal patterns. There are many challenges facing the computational cognitive neuroscience field that deals with modeling such brain functions. Spiking networks may have some advantages in such applications [61]. Several journals specialize in such topics and a book with subtitle “Towards Neuroscience-inspired computing” appeared recently [63], discussing modular organization, timing and synchronization, learning and memory models inspired by understanding of the brain.

We are interested here only in identification of promising routes to simplified models that may be used for processing of dynamic spatiotemporal patterns, going from low to high-level cognition. One mechanism proposed by Hopfield and Brody [27] is based on recognition of the spatiotemporal pattern via transient synchrony of the action potentials of a group of neurons. The recognition is in their model invariant to uniform time warp and uniform intensity change of the input events. Although modeling of recognition in feature spaces is rather straightforward invariance is rather difficult to obtain.

Recognition or categorization of spatiotemporal patterns allows for their symbolic labeling, although such labeling may sometimes be a crude approximation. Transition from recurrent neural networks (RNNs) to finite state automata rules and symbols may be done in several ways: extracting transition rules from dynamics of RNNs, learning finite state behavior by RNNs, or encoding finite-state automata in neural networks [63,22]. Although a lot of effort has been devoted to this subject most papers assume only two internal states (active or not) for automata and for network PEs, severely restricting their possibilities. Relations between more complex PEs and automata with complex internal states are very interesting but not much is known about them. Sequential processes in modular networks, composed of subnetworks with some local memory, should roughly correspond to the information processing by neocortex. These processes could be approximated by probabilistic multi-state fuzzy automata. Complex network processing elements with local memory may process information step-wise, depending on their history. Modules, or subnetworks, should specialize in solving fragments of the problem. Such approach may be necessary to achieve the level of non-trivial grammar that should emerge from analysis of transitions allowed in finite state automata corresponding to networks.

3.8 Up to the brains and the societies of brains

The same scheme may be used at higher levels: modular networks described above are used to process information in a way that roughly corresponds to functions of various brain areas, and these networks become modules that are used to build next-level “supernetworks”, functional equivalents of whole brains. The principles at each level are similar: networks of interacting modules adjust to the flow of information changing their internal knowledge and their interactions with other modules. Only at quite low level, with very simple interaction and local knowledge of PEs, efficient algorithms for learning are known. The process of learning leads to emergence of novel, complex behaviors and competencies. Maximization of system information capacity may be one guiding principle in building such systems: if the supernetwork is not able to model all relations in the environment then it should recruit additional members that will specialize in learning facts, relations or behaviors that have been missing.

At present all systems that reach the level of higher cognitive functions and are used for commonsense reasoning and natural language understanding are based on artificial intelligence expert system technology. The CYC system (www.cyc.com) with over one million assertions and tens of thousands of concepts does not use any neural technology or cognitive inspirations. It is a brute-force symbolic approach. Other successful AI models, such as the Soar [43] or Act [3] systems, have developed also quite far remaining at the level of purely symbolic processing. Can such technology be improved using subsymbolic computational intelligence ideas? Belief networks may be integrated in such systems in relatively easily, but it is still a big challenge for neural systems to scale up to such applications. DISCERN was the only really ambitious project that used neural lexicon for natural language processing [38], but it did not go too far and has been abandoned.

Very complex supernetworks, such as the individual brains, may be further treated as units that co-operate to create higher-level structures, such as groups of experts, institutions, think-tanks or universities, commanding huge amounts of knowledge that is required to solve the problems facing the whole society. Brainstorming is an example of interaction that may bring ideas up that are further evaluated and analyzed in a logical way by groups of experts. The difficult part is to create ideas. Creativity requires novel combination, generalization of knowledge that each unit has, applying it in novel ways. This process may not fundamentally differ from generalization in neural networks, although it takes place at much higher level of complexity. The difficult part is to create a system that has sufficiently rich, dense representation of useful knowledge to be able to solve the problem by combining or adding new concepts/elements.

4 Problems pointed out by experts

Certainly, the statements presented in the previous sections, reflecting authors' point of view on the subject, do not pretend to be a complete and comprehensive answer to the question “Quo vadis, computational intelligence?”. The field of CI is

very broad and still expanding, so – in a sense – even listing all of its branches or sub-fields may be considered a challenge itself.

Having that in mind we had an idea that a good way to make the *real* and *efficient* search for the challenging problems is to post this question to a group of the well known experts in several branches of CI. Therefore, we asked a few leading scientists working in the field of computational intelligence (understood in a very broad sense) what - according to them – would be the most challenging problems for the next 5-10 years in their area of expertise, and what solutions (if known) are at the horizon.

CI disciplines represented by the experts included neural networks, genetic algorithms, evolutionary computing, swarm optimization, artificial life, Bayesian methods, brain sciences, neuroinformatics, robotics, computational biology, fuzzy systems, rough sets, mean field methods, control theory, and related disciplines. Both theoretical as well as applicative challenges were asked for.

Our first idea was to collect the answers and then try to identify some number of common problems that may be of general interest for computational intelligence community. However, after collecting the responses we decided that presentation of individual experts' opinions with some comments from us will be more advantageous to potential readers. In order to precisely express views and opinions provided by the experts we have decided to present several citations from their responses. For the sake of clarity of the presentation in the following text all citations of experts' opinions will be distinguished by *italic font*.

Problems posted by the experts can be divided into two main categories:

- general CI problems related to human-type intelligence,
- specific problems within various CI disciplines.

4.1 General CI problems related to human-like intelligence

Among general problems envisioned by the experts two were proposed by Lee Giles. The first one concerns *bringing robotics into the mainstream world*. Probably the effective way of bringing robotics (and CI in general) into the mainstream world will require the development of CI-based user-friendly everyday applications able to convince people of the usefulness and practical value of CI research. Several devices of that kind already exist, e.g. intelligent adaptive fuzzy controllers installed in public lifts or various household machines. These bottom-level, practical successes of CI are however not well advertised and therefore not well known (or actually not at all known) to general public. Paradoxically, events which seem to be much more “abstract” achievements of AI (at least for non-professionals) became recently very influential signs of AI successes. These include the defeat of Kasparov by Deep Blue supercomputer or design of artificial dogs – Aibo and Poo-Chi.

The other challenge pointed out by Giles is *integrating the separate successes of AI - vision, speech, etc - into an intelligent SYSTEM*. In fact building of intelligent agents has been of primary concern for AI experts for about a decade now.

Development of new robotic toys, such as the Aibo dogs, requires integration of many branches of CI. What capabilities will such toy robots show in 20 years? Perhaps similar progress as in the case of personal computer hardware (for example graphics) and software (from DOS to Windows XP) should be expected here. Several advanced robotics projects are being currently developed in the industrial labs. The most ambitious one concerning humanoid robots - developed from and around the Cog project at MIT - demanded integration of several perceptual and motor systems [1]. Social interaction with humans demands much more: identification and analysis of emotional cues during interactions with humans, speech prosody, shared attention and visual search, learning through imitation, and building theory of mind.

Similar challenge concerning the design of *the advanced user interfaces using natural language, speech, and visualizations* is listed by Erkki Oja. According to Oja realization of such integrated, human friendly interfaces *requires very advanced and robust pattern recognition methods*. As a possible approach to tackle these tasks Oja proposes application of *some kind of machine learning, as well as probabilistic modeling* methods aimed at finding – in unsupervised manner – *a suitable compressed representation of the data*. The underlying idea is that *when models are learned from the actual data, they are able to explain the data, and meaningful inferences and decisions can be based on the compressed models*. These issues are also connected with the questions about functioning of the learning algorithms in human brains. *If we can really find out the learning algorithms that the brain is using, this will have an enormous impact on both neuroscience and on the artificial neural systems* (Oja).

A related challenge from the domain of intelligent human-like interfaces is also stated by John Taylor: *how is human language understanding achieved? This is needed to be answered to enable language systems to improve and to allow human-machine interaction*. Before answering this questions two other major challenges in the area of building intelligent autonomous systems need to be considered. The first one is concerned with the problem of *how is attention-controlled processing achieved to create (by learning) internal goals for an autonomous system? This requires a reward learning structure, but more specifically a way of constructing (prefrontal-like) representations of goals of action/object character at the basis of schema development*. Current research on reinforcement learning draws little inspiration from brain research. Perhaps the subject is not understood well enough. On the other hand considerable progress has been achieved by Ai Enterprises in building a “child machine”, trained by reinforcement learning to respond to symbols like an infant [39]. The transcripts from the program have been evaluated by a developmental psychologist as a healthy babbling of 18-month old baby. This is still only babbling and it will be fascinating to see how far can one go in this way.

The other challenging problem is answering the question of *how is automatization of response achieved by learning? Initial controlled response needs to be 'put on automatic' in order to enable an autonomous system to concentrate on other tasks. This may be solved by further understanding of the processes occurring in the frontal lobes in their interaction with the basal ganglia. At the same time episodic and working memory are clearly crucially involved* (Taylor). In other words,

how is the task initially requiring conscious decisions taken by the brain at the highest level, such as learning to drive, becomes quite automatic? What is the role of working memory here? Perhaps it is needed only to provide reinforcement by observing and evaluating the actions that the brain has planned and executed? Is this the main role of consciousness? Relating one's own performance to memorized episodes of performance observed previously requires evaluation and comparison followed by emotional reactions that provide reinforcement and increase neuromodulation, facilitating rapid learning. Working memory is essential to perform such complex task, and after the skill is learned there is no need for reinforcement and it becomes automatic (subconscious). Unfortunately working memory models are not well developed.

Similarly to Oja, the need for appropriate data (state) representation is also stressed by Christoph von der Malsburg: *In the classical field of AI, this question is left entirely open, a myriad of different applications being dealt with by a myriad of different data formats. In the field of Artificial Neural Networks, there is a generic data format – neurons acting as elementary symbols – but this data format is too impoverished, having no provision for representing hierarchical structures, and having no provision of the equivalent of the pointers of AI.*

A related challenging problem pointed out by von der Malsburg is design of autonomous self-organization processes in the (hierarchical) state organization – *the state of an intelligent system must be built up under the control of actual inputs and of short-term and long-term stored information. The algorithmic approach to state construction ... must be overcome and be replaced by autonomous organization.* State organization must be conformed to a general underlying idea of the ability of intelligent systems to generalize based on the current state: *Intelligent systems relate particular situations to more general patterns. This is the basis for generalization, the hall-mark of intelligence. Each situation we meet is new in detail. It is important to recognize general patterns in them so that known tools and reactions can be applied. To recognize a specific situation as an instance of a general pattern, the system must find correspondences between sub-patterns, and must be able to represent the result by expressing these correspondences as a set of links. Finding such sets of links is an exercise in network self-organization* (von der Malsburg).

On the other hand self-organization alone seems to be not powerful enough in order to create intelligent systems (behaviors) in limited time and with limited resources. Therefore some kind of learning with a teacher seems to be indispensable. *An important sub-category of learning is guided by teaching. Essential instruments of teaching are: showing of examples, pointing, naming and explanation. To provide the necessary instruments that underlay these activities constitutes an considerable array of sub-problems. Human intelligence is a social phenomenon and is based on teaching. The alternative is evolution, but we will hardly have the patience to create the intelligence if only of a mouse or a frog by purely evolutionary mechanisms* (von der Malsburg).

Another problem stressed by von der Malsburg is the ability of intelligent autonomous systems to learn from natural environments: *Intelligent systems must be able to pick up significant structure from their environment. Machine learning in AI is limited to pre-coded application fields. Artificial neural networks promise*

learning from input, finding significant structure on the basis of input statistics, but that concept fails beyond a few hundred bits of information per input pattern, requiring astronomical learning times. Animals and humans demonstrate learning from one or a few examples. To emulate this, mechanisms must be found for identifying significant structure in single scenes. Similarly to Oja, von der Malsburg underlines the role of interaction with natural environment - intelligent systems must be able to autonomously interact with their environment, by interpreting the signals they receive and closing the loop from action to perception.

The next step on the way of building intelligent systems is the stage of hierarchical integration of separate modules or operational paradigms into one, coherent organizational structure. Two major challenges concerning this issue were put forward by von der Malsburg. The first problem is the *subsystem integration*. *An intelligent system is to be composed of (a hierarchy of) individual modules, each representing an independent source of information or computational process, and problems are to be solved by coupling these modules in a coherent way. This process may be likened to a negotiation process, in which the different players try to reach agreement with each other by adjusting internal parameters. If there is sufficient redundancy in the system, a globally coherent state of the system arises by self-organization. This process is the basis for the creativity of intelligent systems. The problem is to find the general terms and laws which make subsystem integration possible.*

The other – closely related – challenge is *structuring of general goal hierarchies*. *Whatever intelligent systems do, they are pursuing goals, which they themselves set out with or recognize as important for a given scene or application area. To organize goal-oriented behavior, a system starts with rather generally defined goals (survive, don't get hurt, feed yourself,..) and must be able to autonomously break those goals down to specific settings, and to self-organize consistent goal hierarchies.*

The key issues in development of computational intelligence field according to Harold Szu lie in the area of unsupervised learning. The CI science is now *in the cross road of taking the advantage of the exponential growth of information sciences modeling and linear growth of neurosciences experiments. The key is to find the proper representation of the complex neuroscience experiment data that can couple the two together.* The idea of learning without a teacher has a “natural” support in biological world, since we – people *have pairs of eyes, ears, etc. Therefore, the proper representation is a vector time series whose components are input of a pair of eyes, ears, etc. - smart sensor pairs.* Szu believes that the unsupervised learning Hebb rule results from the thermodynamics Helmholtz free energy (see [56] for mathematical formulation).

According to Szu one of the intermediate problems that need to be solved on the way is developing of appropriate and efficient procedures for sampling information from the environment. One of the key sub-issues are the redundancy problem and the problem of dimensionality reduction.

Another central challenge is stated by Paul Werbos in his recent paper [62]¹: *Artificial neural networks offer both a challenge to control theory and some ways to help meet that challenge. We need new efforts/proposals from control theorists and others to make progress towards the key long-term challenge: to design generic families of intelligent controllers such that one system (like the mammal brain) has a general-purpose ability to adapt to a wide variety of large nonlinear stochastic environments, and learn a strategy of action to maximize some measure of utility across time. New results in nonlinear function approximation and approximate dynamic programming put this goal in sight, but many parallel efforts will be needed to get there. Concepts from optimal control, adaptive control and robust control need to be unified more effectively.*

The above citation presents the general statement concerning the need for new ideas/proposals that might influence research in the intelligent control area. Going further Werbos states several goals and suggests possible approaches to achieve them. In fact the paper [62] was written with the similar intention as our work and we encourage anybody interested in the subject to read it. Since we are unable to present all ideas from this paper we have chosen only two problems that appear to us to be very important. One of them addresses the problem of appropriate balance between problem independent approach to learning in intelligent systems versus methods taking advantage of problem specific knowledge. *In the most challenging applications, the ideal strategy may be to look for a learning system as powerful as possible, a system able to converge to the optimal strategy without any prior knowledge at all – and then initialize that system to an initial strategy and model as close as possible to the most extensive prior knowledge we can find.*

Another suggestion is to regard artificial intelligent systems in the rational framework which means defining our goals and expectations towards them in the realistic way. *We cannot expect the brain or any other physical device to guarantee an exact optimal strategy of action in the general case. That is too hard for any physically realizable system. We will probably never be able to build a device to play a perfect game of chess or a perfect game of Go. ... We look for the best possible approximations, trying to be as exact as we can, but not giving up on the true nonlinear problems of real interest.* We would definitely agree with that. In any real situation when non-trivial goals are to be achieved the optimal strategy cannot be “calculated” in a reasonable time. We believe that one of the main obstacles on the way of developing intelligent autonomous systems were – right from the beginning – too high expectations regarding their abilities and the lack of properly defined, achievable, realistic goals. The brains are not all-powerful devices, but have been prepared by millions of years of evolution to make reasonable decisions in situations that are natural from the environmental point of view. In many unnatural situations humans suffer from “cognitive illusions” [46].

1 Submitted to IEEE CDC02 conference by invitation.

4.2 General problems within certain CI disciplines

Several problems stated by the experts concerned particular disciplines that constitute computational intelligence. In the context of neural networks two of the proposed problems were connected with the reduction of data dimension in both theoretical as well as applicative aspects.

One of them known as the *curse of dimensionality* is pointed out by Vera Kurkova: *for some tasks implementation of theoretically optimal approximation procedures becomes unfeasible because of unmanageably large number of parameters. In particular, high-dimensional tasks are limited by the “curse of dimensionality”, i.e., an exponentially fast scaling of the number of parameters with the number of variables. One of the challenges of mathematical theory of neuro-computing is to get some understanding what properties make high-dimensional connectionist models efficient, what attributes of multivariable mappings guarantee that their approximation by certain types of neural networks guarantee does not exhibit the “curse of dimensionality”.*

Similar challenging problem concerned with unmanageable data dimensionality is put forward by Lipo Wang: *which features are relevant and which of them are important for a task at hand?* In several application domains high-dimensional data, except for being computationally infeasible, is also difficult to be properly interpreted. In other words *when data dimensionality is high, information can be obscured, because of the presence of irrelevant features. This is important for many data mining tasks, such as classification, clustering, and rule extraction.* In most practical problems estimation of relative importance of particular data properties comes out from experts' knowledge or experience. Quite rarely it becomes available as a result of theoretical analysis. In neural networks domain some general methods supporting that kind of analysis have been already developed. The most popular examples include the Principal Component Analysis – allowing reduction of data dimensionality based on its orthogonalisation and defining the most relevant dimensions. The other well known method is Independent Component Analysis – allowing for blind source separation in case of multi-source and noisy data. Both methods perform well in many cases however their applicability is not unconditional. For example application of PCA method in case of highly interrelated data (e.g. when sampled from multidimensional chaotic systems) may lead to degradation of performance compared to using data that was not pre-processed by PCA [30].

The need for reliable identification of relevant features in multidimensional data is especially important within popular, fast growing disciplines where the increase of the amount of available data is enormous. Indeed, in bioinformatics – for example - tens or even hundreds of thousands of features are defined for some problems, and the selection of information becomes a central issue. One possible approach is to use feature aggregation instead of feature selection. Such hierarchical processing of information allows for integration of very rich input data into simpler, but more informative, higher-level structures. Integration of distributions of time-dependent input (sensory) signals creates distributions at the higher levels. Although interval arithmetic is known, relevant mathematics for computing with

arbitrary distributions has not yet been formulated. The other promising idea – similar to the way in which attention facilitates control - is selecting subsets of relevant features. This results in a dynamical process, serving the short and long-term goals of the system.

Another challenging issue connected with data processing is design and construction of intelligent systems capable of providing a focused search through the huge amount of available data (e.g. published over the Internet). According to Oja *in short and medium term, we will have a great demand for fast and reliable computer systems to manage and analyze the vast and ever increasing data masses (text, images, measurements, digital sound and video, etc.) available in databases and the Web. How to extract information and knowledge, to be used by humans, from this kind of scattered data storages?* The problem is well known and various solutions are being proposed. One group of solutions is based on visualization techniques, for example using the Web-SOM variant of Kohonen's networks. Many other clusterization methods and visualization techniques are certainly worth using. Data mining techniques for modeling the user interest and extracting knowledge from data are in the experimental stage. Latent Semantic Indexing [8] is based on the analysis of the terms-document frequency matrix using singular Value Decomposition to find Principal Components that are treated as "concepts". Unfortunately these concepts are vector coefficients and are only useful for estimation of similarity of documents - but they are not understandable concepts interesting to humans. The Interspace Research Project is aimed at semantic indexing of multimedia information and facilitating communication between different communities of experts that use similar concepts. Only a few CI methods have been applied to this field so far.

Certainly, the problem of explosive growing of the amount of accessible data has a great impact on artificial systems' (and also humans') ability to preprocess and analyze this data and consequently make optimal (or at least efficient) decisions. Gathering a *compact set* of relevant and complete information concerning a given task requires much more efficient search engines than those available now. One of the underlying features of these "future" search engines must be ability to analyze the data *contextually*. Ultimately understanding of texts requires sophisticated natural language processing techniques. At this stage the best programs for natural language understanding (NLU) are based on huge ontologies, human created hierarchical description of concepts and their interrelations. The release of the OpenCyc tools by CycCorp in 2001 made such applications easier, but hybrid systems, combining NLU techniques developed by AI experts with the data mining techniques developed by CI experts, have not yet been created.

One of the possible directions on the way to design intelligent decision support systems capable of extracting useful information and knowledge form large data repositories is distributive multi-agent approach in which a set of agents automatically searches various databases and professional services (e.g. the Internet ones) in real time in order to provide an up-to-date, relevant information. This would also require the soft mechanisms for checking information reliability. Furthermore, efficient mechanisms of automated reasoning based on CI techniques need to be applied in such systems.

Another challenge was identified in the area of combinatorial optimization (Lipo Wang): *evolutionary computation and neural networks are effective approaches to solving optimization problems. How can we make them more powerful?* This question is really hard to answer. In the framework of neural networks two main approaches to solving constraint optimization problems exist: evolving template matching and Hopfield-type networks. Both of them suffer from serious intrinsic limitations. Template matching methods require that the problem to be solved has appropriate geometrical representation. Hopfield-type approaches suffer from gradient minimization scheme and the lack of general recipes for defining constraints coefficients in the energy function. Despite enormous number of papers devoted to the above two types of approaches and despite the development of various modifications to their original formulations it seems that the efficacy of neural network-based optimization methods – although significantly increased compared to the initial approaches - cannot be proven for the problems exceeding a certain level of complexity. Similar situation exists in evolutionary computation domain where, for example, no general rules were yet developed concerning the efficient coding scheme or choosing *a priori* a suitable form of crossover operation or appropriate mutation probability. In most cases the above very basic choices are still being decided by trial and error methods. Consequently, in complex problem domains the time required to achieve reasonably good solutions is prohibitive.

Similar problems concerning scalability of evolutionary computation algorithms are pointed out by Xin Yao: *There have been all kinds of evolutionary techniques, methods and algorithms published in the literature which appear to work very well for certain classes of problems with a relatively small size. However, few can deal with large and complex real world problems. It is well known that divide-and-conquer is an effective and often the only strategy that can be used to tackle a large problem. It is unclear, though, how to divide a large problem and how to put the individual solutions back together in a knowledge-lean domain. Automated approaches to divide-and-conquer will be a challenge, as well as an opportunity to tackle the scalability issue, for evolutionary computation researchers.*

An issue closely related to scalability problem is the lack of theoretical estimations of computational complexity of evolutionary methods: *We still know very little about the computational time complexity of evolutionary algorithms on various problems. It is interesting to observe that a key concern in the analysis of algorithms in the mainstream computer science is computational time complexity, while very few complexity results have been established for evolutionary algorithms. It is still unclear where the real power, if any, of evolutionary algorithms is (Yao).*

Another problem emphasized by Yao is the need for suitable mechanisms that allow promotion of a "team work" rather than the best (highest scored) individuals: *Evolutionary computation emphasizes populations. While one can use a very large population size, it is often the best individual that we are after. This is in sharp contrast to our own (human) experience in problem solving, where we tend*

to use a group of people to solve a large and complex problem. Clearly, we need to rethink our endeavour in finding the best individual. Instead, we need the best team to solve a large and complex problem. This challenges us to think about questions such as how to evolve/design the best team and how to scale up the team to deal with increasingly large and complex problems.

5 Summary

In this short article only a few challenges facing various branches of computational intelligence may obviously be identified. According to several suggestions the underlying issues are related to – generally speaking – emulation of human-type intelligent behavior. Within this research area several specific goals and challenges are identified, e.g.

- flexible data (state) representations and suitable, context (state) dependent training methods,
- training methods involving both supervised and unsupervised paradigms allowing to combine learning from examples with self-organizing evolutionary development of the system,
- further investigation of the working mechanisms of biological brains,
- integration of solutions achieved for partial (individual) problems into more complex, efficiently working systems,
- theoretical investigations on the complexity, potential applicability and limitations of explored ideas.

We have tried to show some promising directions that should allow to model certain brain-like functions, going beyond the current applications of neural networks in pattern recognition. UCI repository of data for machine learning methods [37] has played a very important role in providing the pattern recognition problems to be solved. Collection of more ambitious problems for testing new approaches going beyond classification and approximation is urgently needed.

We hope that the issues pointed out by professionals and by ourselves will serve as useful pointers – especially for young and less experienced researchers looking for interesting problems – in developing computational intelligence in promising directions.

Acknowledgments

We would like to thank our expert colleagues who supported this project by sending descriptions of problems that according to them are the most challenging issues in the field of computational intelligence. We gratefully acknowledge the helpful comments from C. Lee Giles (The Pennsylvania State University), Vera

Kurkova (Academy of Science of the Czech Republic), Christoph von der Malsburg (Ruhr-Universität Bochum), Erkki Oja (Helsinki University of Technology), Harold Szu (Naval Research Laboratory), John G. Taylor (King's College London), Lipo Wang (Nanyang Technical University), Xin Yao (University of Birmingham) and Paul Werbos (National Science Foundation). W.D. would like to thank the Polish State Committee for Scientific Research for support, grant no. 8 T11C 006 19.

6 References

1. Adams, B., Breazeal C., Brooks, R., Scassellati, B.: Humanoid Robots: A New Kind of Tool, *IEEE Intelligent Systems* **15** (2000) 25-31
2. Amit D.J.: The Hebbian paradigm reintegrated: local reverberations as internal representations. *Brain and Behavioral Science* **18** (1995) 617-657
3. Anderson, J.R.: *Rules of the Mind*. Erlbaum, Hillsdale, N.J. (1993)
4. Bedford, T., Keane M., Series, C.: *Ergodic theory, symbolic dynamics and hyperbolic spaces*. Oxford University Press, Oxford, UK (1991)
5. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press (1995)
6. Crevier, D.: *AI: The Tumultuous History of the Search for Artificial Intelligence*. Basic Books, New York (1993)
7. McClelland, J.L., Rumelhart D.E. and the PDP research group.: *Parallel distributed processing*. The MIT Press, Cambridge, MA (1987)
8. Deerwester, Dumais, Landauer, Furnas, Harshman (1990) Indexing by latent semantic analysis, *Journal of the American Society for Information Science* **41**(6): 391-407
9. Duch, W.: Platonic model of mind as an approximation to neurodynamics. In: *Brain-like computing and intelligent information systems*, ed. S. Amari, N. Kasabov. Springer, Singapore (1997) 491-512
10. Duch, W.: Similarity-Based Methods. *Control and Cybernetics* **4** (2000) 937-968
11. Duch, W., Adamczak, R., Dierksen, G.H.F.: Constructive density estimation network based on several different separable transfer functions. *9th European Symposium on Artificial Neural Networks (ESANN)*, Brugge. De-facto publications (2001) 107-112
12. Duch, W., Adamczak, R., Grabczewski, K.: Methodology of extraction, optimization and application of crisp and fuzzy logical rules. *IEEE Transactions on Neural Networks* **12** (2001) 277-306
13. Duch, W., Dierksen, G.H.F.: Feature Space Mapping as a universal adaptive system. *Computer Physics Communications* **87** (1995) 341-371
14. Duch, W., Grudziński, K., Prototype based rules - new way to understand the data. *Int. Joint Conference on Neural Networks*, Washington D.C., July 2001, 1858-1863
15. Duch, W., Itert, L., Grudziński, K.: Competent undemocratic committees. *Int. Conf. on Neural Networks and Soft Computing*, Zakopane, Poland (in print, 2002)
16. Duch W., Jankowski, N.: Survey of neural transfer functions. *Neural Computing Surveys* **2** (1999) 163-213
17. Duch, W., Jankowski, N.: Transfer functions: hidden possibilities for better neural networks. *9th European Symposium on Artificial Neural Networks (ESANN)*, Brugge. De-facto publications (2001) 81-94
18. Duda, R.O, Hart, P.E, Stork, D.G.: *Pattern Classification*, 2nd Ed, John Wiley & Sons, New York (2001)

19. Fahlman, S.E., Lebiere, C.: (1990) The cascade-correlation learning architecture, In D. Touretzky (Ed.), *Advances in Neural Information Processing Systems*, 2, Morgan Kaufmann, 524-532
20. Frasconi, P., Gori, M., Sperduti, A.: A General Framework for Adaptive Processing of Data Structures. *IEEE Transactions on Neural Networks* 9 (1998) 768-786
21. Frean, M.: (1990) The upstart algorithm: a method for constructing and training feed-forward neural networks, *Neural Computation* 2: 198-209
22. Giles, L.C., Gori, M. (Eds): *Adaptive processing of sequences and data structures*. Springer, Berlin (1998)
23. Goldfarb, L. Nigam, S.: The unified learning paradigm: A foundation for AI. In: V.Honovar, L.Uhr, Eds. *Artificial Intelligence and Neural Networks: Steps Toward Principled Integration*. Academic Press, Boston (1994)
24. Golomb, D., Hansel, D., Shraiman, B, Sompolinsky, H.: Clustering in globally coupled phase oscillators. *Phys Rev. A* 45 (1992) 3516-3530
25. Hasti, T, Tibshirani, R, Friedman J.: *The Elements of Statistical Learning*. Springer Series in Statistics, New York (2001)
26. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities, *Proc. National Academy of Science USA*, 79 (1982) 2554-2558
27. Hopfield, J.J., Brody, C.D.: What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration. *PNAS* 98 (2001) 1282-1287
28. Hsu C.S.: Global analysis by cell mapping, *J. of Bifurcation and Chaos* 2 (1994) 727-771
29. Jankowski, N., Duch W.: Optimal transfer function neural networks. 9th European Symposium on Artificial Neural Networks (ESANN), Brugge. De-facto publications (2001) 101-106
30. Jaruszewicz, M., Mańdziuk, J. (2002) Short-term weater forecasting with neural nets, *International Conference on Neural Networks and Soft Computing*, Zakopane. Poland, (submitted)
31. Kasabov, N. (1988) ECOS - A framework for evolving connectionist systems and the 'eco' training method, *Proc. of ICONIP'98 - The Fifth International Conference on Neural Information Processing*, Kitakyushu, Japan, 3: 1232-1235
32. Kunstman, N., Hillermeier C., Rabus, B., Tavan P.: An associative memory that can form hypotheses: a phase-coded neural network. *Biological Cybernetics* 72 (1994) 119-132
33. *MIT Encyclopedia of Cognitive Sciences*. Ed. M.A. Wilson, F.C. Keil, MIT Press, Cambridge, MA (1999)
34. Mańdziuk, J., Shastri, L. (1999) Incremental Class Learning – an approach to longlife and scalable learning, *Proc. International Joint Conference on Neural Networks (IJCNN'99)*, Washington D.C., USA, (6 pages on CD-ROM)
35. Mańdziuk, J., Shastri, L. (2002) Incremental Class Learning approach and its application to Handwritten Digit Recognition problem, *Information Sciences* (in press)
36. Marczak, M., Duch, W., Grudziński, K., Naud, A.: (2002) Transformation Distances, Strings and Identification of DNA Promoters. *Int. Conf. on Neural Networks and Soft Computing*, Zakopane, Poland (in print, 2002)
37. Mertz, C.J., Murphy, P.M.: UCI repository of machine learning databases, <http://www.ics.uci.edu/pub/machine-learningdatabases>
38. Miikkulainen, R. *Subsymbolic natural language processing: an integrated model of scripts, lexicon and memory*. MIT Press, Cambridge, MA (1993)
39. Minsky M., Papert S.: *Perceptrons*. MIT Press, Cambridge, MA (1969), 2nd ed. (1988)
40. Mitchell T.: *Machine learning*. McGraw Hill (1997)
41. Mitra, P., Mitra, S., Pal, S.K.: (2000) Staging of cervical cancer using Soft Computing, *IEEE Transactions on Biomedical Engineering*, 47(7): 934-940

42. Newman J, Baars B.J.: Neural Global Workspace Model. *Concepts in Neuroscience* 4 (1993) 255-290
43. Newell, A.: *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press (1990)
44. O'Regan, J.K., Noë, A.: A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences* 24(5) (2001, in print)
45. Pękalska, E., Pacilik, P., Duin, R.P.W.: A generalized kernel approach to dissimilarity-based classification. *J. Machine Learning Research* 2 (2001) 175-211
46. Piattelli-Palmarini, M.: *Inevitable Illusions: How Mistakes of Reason Rule Our Minds*. John Wiley & Sons (1996)
47. Poole D., Mackworth, A., Goebel, R.: *Computational Intelligence. A Logical Approach*. Oxford University Press, New York (1998)
48. Rich E., Knight K.: *Artificial Intelligence*. McGraw Hill Inc, Int'l Edition (1991)
49. Rumelhart, D.E., Hinton, G.E., Williams R.J.: Learning representations by back-propagating errors, *Nature* **323** (1986) 533-536
50. Russell, S. J., Norvig, P.: *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, N.J. (1995)
51. Saad, E.W., Prokhorov, D.V., Wunsch II, D.C.: (1998) Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks, *IEEE Transactions on Neural Networks*, 9(6): 1456-1470
52. Shastri, L., Fontaine, T.: (1995) Recognizing handwritten digit strings using modular spatio-temporal connectionist networks, *Connection Science* 7(3): 211-235
53. Specht, D.: (1990) Probabilistic neural networks, *Neural Networks* 3: 109-118
54. Specht, D.: (1991) A general regression neural network, *IEEE Transactions on neural Networks* 2: 568-576
55. Sun, R., Giles, L. (Eds): *Sequence learning*. Springer Verlag, Berlin (2001)
56. Szu, H., Kopriva, I (2002) Constrained equal a priori entropy for unsupervised remote sensing, *IEEE Transactions on Geoscience Remote Sensing*
57. Thrun, S., *Explanation based neural network learning. A lifelong learning approach*. Kluwer Academic Publishers, Boston / Dordrecht / London, 1996
58. Thrun, S., Mitchell, T.M.: (1994) Learning one more thing, Technical Report: CMU-CS-94-184
59. Treister-Goren, A., Hutchens, J.L.: *Creating AI: A unique interplay between the development of learning algorithms and their education*. Technical Report, AI Enterprises, Tel-Aviv 2000. Available from <http://www.a-i.com>
60. Waibel, A.: (1989) Consonant recognition by modular construction of large phonemic time-delay neural networks, In D. Touretzky (Ed.), *Advances in Neural Information Processing Systems*, 1, Morgan Kaufmann, 215-223
61. Wang, D.: On Connectedness: A Solution Based on Oscillatory Correlation. *Neural Computation* 12 (2000) 131-139
62. Werbos, P., *Neural Networks for Control: Research Opportunities and Recent Developments*, IEEE CDC'02 Conference, (submitted, 2002)
63. Wermter, S., Austin, J., Willshaw, D., (Eds.): *Emergent neural computational architectures based on neuroscience. Towards Neuroscience-inspired computing*. Springer, Berlin (2001)
64. Winston P.: *Artificial Intelligence*. 3rd ed, Addison Wesley (1992)