GENERATING OF NEURO-EXPERT SYSTEMS APPLIED IN POWER INDUSTRY

MIHAI POPPER – S.C. Termoelectrica S.A.

Abstract. The concept of expert system can be defined as a structure based on artificial intelligence, on the notion of knowledge in the field, can acquire knowledge from human experts, following that by logical inference to ensure efficient handling of specific issues in the field of expertise. The power of expert systems can be increased by using built-in neural structures, being able to learn from examples when human experts have not sufficient knowledge to solve the problems. Specificity of artificial intelligence consists precisely in identifying of the quantifiable elements of reasoning, using the abstraction process, for use these in computer programs. Thus, the qualitative power of human intelligence transferred on computer, working together with the quantitative power of the computing machines can offer a viable solution for applications in engineering industries.

Keywords: systems, expert, neuronal, intelligence, inference

The first stage of industrial processes informatization consisted by developing software applications in industrial engineering, power sector being one of the first industries that began this process. Applications that units of power systems have implemented in this first round were from the following categories:

- Programs to optimize technological processes
- Programs for equipment diagnosis and maintenance of installations
- Programs to predict energy equipment operation
- Monitoring programs of technical parameters of the system

These applications were developed by research institutes in the field and by computing centers in the major energy companies and have reduced costs for information and decision support by default to reduce energy cost. However, in the early 1980s occurred increasingly obvious shortcomings of this type of computerization realized by:

- High cost of achieving the primary information collected by human operators
- Low performance in managing of an ever more collected and processed data volume.
- Logical model of problem solving is dependent on the program it is incorporated; any change in the logical model requires modification program and new situations do require development of new programs, especially when switching to a new technology industries
- Human user has not access to a logical model of the problem, which it incorporates in the application being made by analysts and programmers, under the guidance of specialists in engineering industries
- Programs offers no explanation on how the final results were obtained
- Not offered opportunities to learn from examples in the absence of clear existence of a general model for solving problems

Based on these disadvantages, there was a new stage of the process of informatization, materialized in the emergence and the widespread use of computers to process, telecommunications networks and database performance.

The automatic collection system parameters using computers to process and transmit their energy to dispeceratele achieved considerable reduction of cost and increase of data collection quality, thereby leading to reduced energy cost implication.

Modern databases provide the processing performance of a large database, providing necessary information to base management decisions. Thus, the first two drawbacks are remedied listed in the first stage of the process of informatization.

The third wave of informatization industrial processes began in the late 80's and early 90s and continues today. The specifics of this phase consist in the wide application of artificial intelligence techniques in industrial engineering.

Artificial intelligence pursues playing or simulation of human reason with computing equipment, which provides quantitative performance superior processing performance in the human brain. In addition, these techniques take some of the qualitative performance of the human brain, far superior performance to any machine. Specificity of artificial intelligence consists precisely in identifying, through the process abstraction, the quantifiable elements of reasoning, for use in computer programs. Thus, the power quality of human intelligence with computer transferred quantity of machines computing power offers a viable solution for applications in engineering industries.

From the artificial intelligence techniques neuro-expert system occupies a special place, with a wide range of applicability in the energy sector. The concept of expert system can be defined as a structure based on artificial intelligence, on the notion of knowledge in the field, can acquire knowledge from human experts, following that by logic inference to ensure efficient handling of specific issues in the field of expertise. The power of expert systems can be increased by using neural structures built in, able to learn from examples when human experts are not sufficient knowledge to solve problems.

In the expert system distinguish these types of reasoning:

- formal reasoning for handling data structures in the purpose of creating new structures based on rules based on inferential rules;
- procedural reasoning, using simulation, enable the formulation of responses to the considered issues;
- abstract reasoning, which is in the process of abstract thinking;
- meta-reasoning based on knowledge processing to obtain new knowledge.

Structure of expert systems foundation is the concept of artificial intelligence that is based on engineering knowledge to solve problems through formal systems. Expert system contains five modules, in addition with knowledge base:

The cognitive module remember knowledge in the knowledge base that it is updated, then search for pieces of knowledge using inferential.

The resolvent module choose control strategy and develops software to solve problems arising, it also check the consistency of steps to resolve the objectives of the matter structure, building control information and reasoning paths traced.

The explanatory module user warrant against the solution obtained for the problem solved by interpreting reasoning ways, analyze the causes which have encouraged or led to failure of execution and highlights the missing pieces of knowledge inferential chain, preventing errors and risks involved in the processing of knowledge.

Communication module connects between system and user using the language of communication and representation of knowledge, making a communication between input and output data, providing data and control devices efectorial.

- Knowledge Base includes concepts for understanding, formulating and solving structural problems factual and procedural knowledge, which may be the true / false and can be deterministic and / or probability. Knowledge Base consists of:
- factual base contains facts form structure of those areas of expertise related to solving the investigated problem, it functions as a working memory by passing communications from the rules for inference mechanism and contains elements of heuristic knowledge on the analyzed. field .
- rules base is the conceptula level of expert system, it is susceptible to periodic adjustments and corrections.

Inference engine comprises a software product that exploits the rules and apply reasoning methodologies premises of the facts in order to elaborate the final conclusions. Inference engine is a reasoning mechanism which has its own knowledge base and, together with the user interface, form a system for obtaining immediate results.

Functional structure inference engine includes:

- a set of specialized algorithms that detect verified rules that can be used;
- selection mechanism of the meta-rules (rules for applying rules) which marked the onset rules lead to expanding the base of facts;

Inference mechanism can use three modes of reasoning:

- forward chained inference (guided by facts);
- back chained inference (goal directed);
- mixt inference.

An inference process starts with chained before a lot of initial facts that are loaded in working memory. At each step it is determined a rule whose premises are satisfied with the current contents of memory. It triggers the action associated with that rule, then the process repeats. This process stops if one is made of the following conditions:

- working memory contents satisfying a certain condition;
- does not exist anymore applyable rules;
- amount of effort has been exhausted a default calculation.

An inference process starts with chained back from a goal (a task) that we decomposed in elementary problems. Therefore it is said that this mode of reasoning is directed in the sense of purpose to achieve its finding of facts. Note that this logic is reversed than deductive reasoning. At each step it selects a rule which has in the conclusions at least one of the objectives do. The premises of this rule are added to the targets to be resolved. This process stops if one is made of the following conditions:

- set of objectives becomes empty;
- does not exist anymore applyable rules;
- amount of effort has been exhausted a default calculation.

Mixed inference tries to make compromises between the two basic modes of reasoning. The general principle of this mode of reasoning is this: it sets a goal and determine the deductible facts; shall first apply reasoning with chained back, which will require the user to specify values to facts known but interrogation and then apply the reasoning of chained before to infer what is possible.

An original method of inference is based on logical disjunction, unlike conventional methods based on implication. This method involves the representation of rules in a disjunctive form, corresponding to several implications.

General algorithm for the inference process by disjunction is:

1. It is formed the inference matrix, as follows
   - matrix columns correspond to the facts names;
   - matrix rows correspond to the rules and current facts values;
   - for a rule matching row every item is is filled with values: 1 - if that coressponding fact is without the negative, with 0 - if that is accompanied by negative or by x - If there is no appears in rule;
   - for facts corresponding rows every item that is filled with values 0 or 1 as appropriate and the remaining elements are completed with x;
2. For every row of the inference matrix:
   2.1 it select rows that can be combined, so there only one column for which the elements of the two

 ISSN 2067 – 5534

lines have opposite values, namely 0 and 1 (we call junction column);

2.2 Combination rows is achieved resulting in a new line after:

- for junction column the will be x;
- for the rest of the columns (not in opposition elements) the result will be x-if both parent rows have the value x to that column, 0 or 1 otherwise; the possible combinations are: (0,0) $\rightarrow$0; (1,1) $\rightarrow$1; (0,x) $\rightarrow$0; (1,x) $\rightarrow$1; (x,x) $\rightarrow$x;
- if all elements of a line are obtained by combining have the value of x, it follows that parent rows contain conflicting information and we stand at a logical nonsense information.
- obtained "child" row is added in the matrix of inference, after the last existing row so that it can be selected to combine with other rows which follow the "father" row ("father" row is the row of the inference matrix for that algorithm selected it for combining available information and the "mother" row is selected by algorithm for this purpose);

2.3. To reduce the time of inference can remove duplicates, are situations in which we want to keep duplicates to the end of the algorithm, because we want to explain all the ways that the information was obtained.

3. move to next "father" row of the inference matrix ;
4. Algorithm ends after attending full inference matrix or when there is no possible combination

Unlike traditional expert systems, fuzzy systems include two additional software components: fuzzyficator and defuzzyficator.

Fuzzyficator is a program that is converting user input into fuzzy values (between 0 and 1) that describe the input in a form recognized and accepted by the inference engine and also preserves complex way of representing data. After the inference process is done, the output obtained will be processed by defuzzyficator which will convert to the format accepted by human users (natural language), which were placed in the same input data format.

Fuzzy function can be: continuous, the interval [0,1] or discrete event that will return a positive value corresponding subunit in a range of values which fit the function argument, that the information specified by user input human.

Achieving an expert system involves scrolling through the following activities:

- development;
- consulting;
- upgrade.

The development is built on knowledge. Knowing it separates factual or declarative knowledge and procedural rules dominated. This stage is performed by inference engine and other modules of expert system (explanatory resolution, communicative, meta-resolvent, etc.). Interfaces for dialogue with users, tests and sample programs are made complex comprising cogniticiens teams, programmers, engineers and specialists in theory of knowledge and areas of investigation.

Development of expert system is a structure that does not include the knowledge base. The consultation phase validated expert system is found in user operation. Users consult and obtain answers to questions raised by inferential engine. Inferential power depends on the expert system knowledge representation methods.

In phase of improvement expert system can specialize in a process of growth parameters for functional structure

An expert system can be created and exploited, either by a specialized program on the field, directed exclusively to it, either by a software product for generating and exploitation of expert systems. These software products enable the development of expert systems to solve problems in different fields, using different knowledge base from case to case, but their operation is provided by the same inference engine.

Expert systems generators working on at least two levels: application and scenario.

The application is the top level that defines the concept to general knowledge about an area of expertise: the classification of facts and rules that reglemented the field. In the application, the facts are defined, without receiving specific values. An application may include one or more scenarios, each of which is a concrete problem to solve in the field of expertise managed by the application.

The scenario is the lower level, which defined specific values of the known facts in a particular situation again. In the scenario can work with the particulars and general rules, defined at the top level. At this level inference engine is moved to reveal facts unknown values, and if this is not entirely possible, expert systems provide possible or probable variant.

For applications with a rich set of facts, which allowed a variety of possible scenarios, you can define classes of scenarios. They represent a level intemediar between application and scenario. A class of scenarios defines specific values of certain facts which are then transmitted to all scenarios in class. These classes can derive a scenario of the other, the heritage values of the facts, and at the elementary level of scenario one starts from the values submitted by its class, which add new value to the specific facts of that scenario.

Similarly, we can define classes of applications among which are defined facts and general rules, then transmitted derived applications.

The main advantages of using expert systems are:

- very effectively with complex problems that require a large amount of knowledge, in the absence of an unique algorithm and clear solution;
- using experience and knowledge of human experts;
- can be faster and more efficient than human experts to solve problems of great complexity;
- ensure total independence of your problem solving and reasoning;
- enables solution explaining and show the resoning details of solving the problem;
- treat uncertainty in an explicit manner, that hope especially of human experts, can be inspected and checked;

- allow easy addition and updating of knowledge;
- offers great flexibility in solving problems by separating the fundamental knowledge in the field of information specific to a problem;
- possesses the capacity for obtaining new knowledge on existing ones;

Thus, were eliminated most inconvenience found in the second wave of industrial informatization, mentioned above. However, in the expert systems manifest limits of which the most important are:

- largely depend on stored knowledge of human expert who express them and update;
- can not act based on intuition or common sense, because they are not represented;
- learning is not automatically, updating of the basic knowledge required human intervention (expert system can only acquire new knowledge on existing ones, but can not learn by analogy, without a theoretical basis);
- nu pot rezolva probleme nedeterministe, din domenii in care cunostintele umane nu sunt suficiente.

To address these shortcomings, expert systems can work with other smart instruments, called neural networks, which mimic to some extent in neuronal structure of the human brain, with opportunities to learn from examples, without a knowledge base page. By combining the two instruments and their methods of reasoning to get a neuro-expert system, which consists of an expert system and neural structure.

In an artificial neural network, output processing element may be the end result or may become another entry in the manufacturing process. Each input x_i is associated with a weight (each entry is weighted, that value is multiplied by the value of the w_i called a share), which is actually stored on own neuron. Weighted inputs are added together. The way the network learns is the repeating re-adjusting of these weightings. Processing elements (neurons) calculates the weighted sum of its inputs being activated only if the amount exceeds a certain threshold, called threshold activation function of the sum of inputs is associated with an artificial neuron relationship: $NET_j = \sum_{i=1}^{N} X_i W_i$

where:

X_i = input

W_i = inputs associated weight

Apply the amount obtained an activation function F, resulting in neuron output value respectively:

Output y_j = F (net j)

Neuron state is updated regularly by the following mechanism: to determine the potential of neural net j, calculating the weighted sum of inputs Xj, which are outputs of other neurons in the network or information from the input neurons; this potential is compared with a threshold tj, neuron activating it (yj = 1),if $net_j \geq t_j$ or become passive (yi = -1) if $net_j < t_j$. This information processing within the neuron corresponds to a step-type transfer functions (heaviside), commonly called activation function. Activation functions can be linear or nonlinear.

Artificial neural networks support more structures, each one having its own network training algorithm. The training or training artificial neural connections we mean determining weights and activation thresholds in order to stabilize the network by minimizing the deviations between the desired output of the network and the registered owner. A trained neural network can process new input data sets, generating close to the desired outputs.

The types of neural networks, we should focus on a structure that allows learning logic functions by researching the history facts. The skills acquired in this way will be imported in the specific form of an expert system production rules. This structure is called neural tree.

The neural tree is a neural network with two layers (for binary systems) or four layers (for fuzzy systems) to assist the expert system in:

- calculating certainty factors of rules;
- learning new rules (or modifying existing ones).

The neural tree structure is:

a) the input layer meaning the facts

b) first output layer is expressed by OR logic rules;

Elements of input nodes connected to this layer through a system of weights and thresholds to simulate the logic OR function as:

- if the item appears with negative rule will have a negative weight, if no negation occurs will have a positive weight, and if it not appears, its weight is 0;
- threshold node-negative weighted sum rule added to the smallest positive ratio.

c) second layer includes a single output node, resulting rule base, rule type nodes it connects to the final node in the logical function AND (all weights are positive and the final node threshold will be the sum of these weights) .

In applying a data set, each node of first layer output will return 1 if okay to be respected or 0, otherwise. Final node will return 1 only if all rules were followed.

For fuzzy systems, between input layer and layer of material is also interposed rules defining two additional layers of elements belonging to the intervals corresponding entry in the rule.

The first of these layers will contain nodes that will compare the input value of items with a value parameterized inside the rule. If the input value must be greater than the parameter of the rule, then the share will be 1 and the threshold will receive the parameter value. If on the contrary, the entry must be less than the parameter, then the share will be -1 and the threshold will be less parameter.

The second layer will complete the definition of belonging to the period specified in rule by executing the logical AND function on the previous layer nodes.

Learning algorithm of logical functions using neural tree can be described as:

1. It starts from the first example of training and initializes the tree is as follows:
   - constructed input layer nodes corresponding values of the facts;
   - each node will construct a node in layer rules equivalent to the amount of training example.
2. For each example of training is done the following:

 ISSN 2067 – 5534

3. To check each rule to the values of the current example;
4. If the value is 1 rule, the rule remains unchanged;
5. If the value is 0 and the rule certainty factor calculated is less than required, to switch to strengthening the rule through several versions of completion. The rule strengthens the values of the current example (extending the function or on them) for each item there is an option of completing;
6. Old format of the rule is removed (to remove the corresponding node in the layer of rules);
7. Check the rules obtained, so that there are no duplicates and does not include each other;
8. Pass the next training example.

For quick training of a neural network can use genetic algorithms. Genetic algorithms are the most developed branch of evolutionary algorithms, the stochastic search methods that simulate natural biological evolution. The idea of such intelligent systems is an evolutionary theory based on principles of mutation processes, re-combination and reproduction.

Genetic algorithms are used to solve complex problems of nonlinear optimization in a large number of engineering disciplines. Like other artificial intelligence techniques like neural networks and fuzzy systems that are run on computers to solve particular problems. The theory of genetic algorithms is using five types of concepts, namely: gene, chromosome, individual, population and genetic algorithms.

The population is all individuals on which the genetic algorithm acts. The individual is a structure that encodes the parameters that must be done.

In biology, chromosome is the structure that determine individual characteristics of organisms. In theory of genetic algorithms, chromosome is a collection of genes that includes structural information about individuals that form the genetic algorithm population.

A gene contains the information contained in a location of a chromosome. It is in fact the low representation of the parameters of the problem can be solved with genetic algorithms.

The combinations of genetic algorithms and neural networks are divided into two categories: support combinations and collaborative combinations.

In support combinations of these technologies (genetic alghoritms or neural netowork) plays the primary way role of solving the problem, while the other plays a support role, supporting the process of finding solutions or analyzing solution once it has been found. They can be used independently in the same process.

Most researchers have found it natural to use genetic algorithms to assist the neural network than vice versa. There are three ways to assist a neural network by genetic algorithms:

- to select the features or characteristics of a space transformation neural classifier;
- to select learning rules or parameters that control learning in a neural network;
- for analyzing a neural network.

Genetic algorithm is used for training data in two ways:

a) characteristics
space transformation

b) selection of subsets of relevant features.

The first method, the method to transform the space characteristics, was applied mainly to the algorithms of the "nearest neighbor", although it applies to other types of neural networks classifiers.

The second direction consists in limiting the crowd characteristics. A lot of features will often improve performance limited type of classifier neural networks and reduce computing effort.

The application of genetic algorithms in neural networks is assisting on feature extraction. This is one of the problems that the bit string representation of the genetic algorithm is very natural-each position indicating that a particular feature is present or not. The main problem is the required time during the calculation for training each network using the specified characteristics of each chromosome.

In collaborative combinations, genetic algorithms and neural network works with to solve the problem. Collaborative studies are divided into the group using a genetic algorithm as a method of learning and the group using genetic algorithm for defining the neural network topology (which in turn uses a local learning method, like back propagation).

Genetic algorithms for learning neural networks involves optimizing neural network weights with a predefined topology. A reliable means of achievement is given to each share of network coding as binary substring. The entire coding binary will be composed of a string concatenated substrings. Therefore, each member of population genetic specify a complete set of weights for neural network..

There are several important arguments that suggest that may be advantageous to apply genetic algorithms in neural network weights optimization problem. It is easy to argue that genetic algorithm has great potential to make a global search on the space weights and thereby elude local minimals.

Genetic learning algorithms cascade correlation algorithm is a novelty in literature that refers to training neural networks using genetic algorithms. Architecture cascade correlation learning by building a proper structure dynamically engage and installation of a neuron at each moment of time until the problem is learned. Two important features of this structure are:

- to train only one layer of weights at any time;
- to train and add a hidden neuron to the network at any time.

A major disadvantage of genetic algorithms seems to be the time scale, meaning they are very slow. Were designed numerous changes to the standard genetic algorithm, precisely in order to improve performance. There are still areas where genetic algorithms can make a decisive contribution to learning neural networks. Particularly, because these algorithms do not need and do not use derivative information, the most appropriate applications are problems in the gradient error is not available or it's getting too expensive.

In conclusion we can say that the neuro-expert system is a highly complex structure whose power intelligence is determined by combining the two models of intelligence:

- ability to learn from examples, based on which

general rules are generated (by the neural structure, possibly aided evolutionary strategies);

- ability to acquire new information on existing ones, applying general rules to obtain concrete information (provided by expert system).

The two main components of the neuro-expert system (expert system and neural network) are put in touch by a specialized module for this purose that "translates" specific language of the two components. It converts the language used weights neural structures in the language used by expert systems logic functions and vice versa, realizing at the same time and exchange of information between the two structures. By combining the two types of reasoning, we obtain a complex system able to handle a wide range of industry issues.

Bibliography

[1] Barahona P., Alferes J.J Progress in Artificial Intelligence Springer Verlag, Berlin, 1999
[2] Benchimol G. Sytemes expert dans l'enterprise Edition Hermes, Paris, 1986
[3] Breton Ph. Histoire d'informatique La Decouverte, Paris, 1987
[4] Franklin S. Artificial Minds The M.I.T. Presse, Cambridge, Massachusetts
[5] Bishop C.M Neuronal Networs for Pattern Recognition, Clarendon Press, Oxford, 1996

 ISSN 2067 – 5534