Sborník vědeckých prací Vysoké školy báňské - Technické univerzity Ostrava
číslo 2, rok 2006, ročník LII, řada strojní
článek č. 1548

Tomáš SALOKY*, Ondrej LÍŠKA, Vladislav MAXIM***, Jaroslav ŠEMINSKÝ******

SOME PROBLEMS OF NEURAL INFORMATION PROCESSING

NIEKTORÉ PROBLÉMY NEURÓNOVÉHO SPRACOVANIA INFORMÁCIE

Abstract

There are two important ways how to implement intelligence from the computational point of view. One is based on symbolism, and the other, based on connectionism. The former approach (symbolic) models intelligence using symbols, while the latter using connections and associated weights (subsymbolic approach). Evolving by different routes, they both have achieved many successes in practical applications. The paper deals with some problems of artificial intelligence (AI) implementation within subsymbolic approach.

Abstrakt

V súčasnosti existujú (z výpočtového hľadiska) dva zásadné prístupy k implementovaniu inteligencie. Jeden je založený na symbolizme, druhý na konekcionizme. Prvý prístup (symbolický) modeluje inteligenciu použitím symbolov, zatiaľ čo druhý využíva spojenia a asociované váhy (subsymbolický prístup). Hoci sa vyvíjali rôznymi cestami, oba dosiahli významné úspechy v praktických aplikáciách. Príspevok sa zaoberá niektorými problémami implementovania umelej inteligencie v rámci subsymbolického prístupu.

1 INTRODUCTION

Our notions of ourselves as living, conscious beings are intimately linked to our notions of brains and thought. Can thought, feelings, and emotions be represented by a set of rules that can be reproduced in a machine? Is the brain nothing more than an incredibly sophisticated computer?

The mechanistic view is the belief that the workings of the mind can be described in terms of the electro-chemical functioning of the brain. Containing about 100 billion cells with complex interrelations that are still only dimly understood, the brain would have to be considered an extraordinarily intricate machine and certainly an unusual one in that it is composed of living material.

2 SUBSYMBOLIC APPROACH

In contrast to the symbolic approach, the neural network approach adopts the brain metaphor, which suggests that intelligence emerges through a large number of processing elements (neurons) connected together, each performing simple computation. The long-term knowledge of a neural

* Prof., Ing., CSc., Department of Automation and Control, Faculty of Mechanical Engineering, Technical University of Košice, Park Komenského 9, Košice, tel. (+421) 55 602 2342, e-mail tomas.saloky@tuke.sk

** Doc. Ing., CSc., Department of Automation and Control, Faculty of Mechanical Engineering, Technical University of Košice, Park Komenského 9, Košice, e-mail ondrej.liska@tuke.sk

*** Ing., CSc., Department of Automation and Control, Faculty of Mechanical Engineering, Technical University of Košice, Park Komenského 9, Košice, e-mail vladislav.maxim@tuke.sk

**** Doc. Ing., PhD ., Department of Automation and Control, Faculty of Mechanical Engineering, Technical University of Košice, Park Komenského 9, Košice, e-mail jaroslav.seminsky@tuke.sk

network is encoded as a set of weights on connections between units. For this reason, the neural network architecture has also been dubbed the connectionist (Feldman and Ballard 1982).

The progress of neurobiology has allowed researchers to build mathematical models of neurons to simulate neural behavior. This idea dates back to the early 1940s when one of the first abstract models of a neuron was introduced by McCulloch and Pitts (1943).

Hebb (1949) proposed a learning law that explained how a network of neurons learned. Other researchers pursued this notion through the next two decades, such as Minsky (1954) and Rosenblatt (1958). Rosenblatt is credited with the perceptron learning algorithm. At about the same time, Widrow and Hoff developed an important variation of perceptron learning, known as the Widrow-Hoff rule.

Later, Minsky and Papert (1969) pointed out theoretical limitations of single-layer neural network models in their landmark book Perceptrons. Due to this pessimistic projection, research on artificial neural networks lapsed into an eclipse for nearly two decades. Despite the negative atmosphere, some researchers still continued their research and produced meaningful results. For example, Anderson (1977) and Grossberg (1980) did important work on psychological models. Kohonen (1977) developed associative memory models.

In the early 1980s, the neural network approach was resurrected. Hopfield (1982) introduced the idea of energy minimization in physics into neural networks. His influential paper endowed this technology with renewed momentum. Feldman and Ballard (1982) made the term "connectionist" popular. Sometimes, connectionism is also referred to as subsymbolic processes, which have become the study of cognitive and AI systems inspired by neural networks (Smolensky 1988). Unlike symbolic AI, connectionism emphasizes the capability of learning and discovering representations. Insidiously, connectionism has become a common ground between traditional AI and neural network research.

In the middle 1980s, the book Parallel Distributed Processing by Rumelhart and McClelland (1986) generated great impacts on computer, cognitive, and biological sciences. Notably, the backpropagation learning algorithm developed by Rumelhart, Hinton, and Williams (1986) offers a powerful solution to training a multilayer neural network and shattered the curse imposed on perceptrons. A spectacular success of this approach is demonstrated by the NETtalk system developed by Sejnowski and Rosenberg (1987), a system that converts English text into highly intelligible speech. It is interesting to note, however, that the idea of backpropagation had been developed by Werbos (1974) and Parker (1982) independently.

Although the neural network approach rejects the notion of separating-knowledge from the inference mechanism, it does not reject the importance of knowledge in many tasks that require intelligence. It just uses a different way to store and manipulate knowledge.

The information processing principles of biological neural networks have been applied to building a computer system for solving difficult problems whose solutions normally require human intelligence. The neural network approach has attracted wide attention and found a growing number of applications especially in the last two decades.

It should be noted, however, that a neural network can be investigated from a number of distinct perspectives, such as computer science and engineering, artificial intelligence, cognitive science, psychology, and neurobiology.

3 NEURAL INFORMATION PROCESSING

Biological neurons transmit electrochemical signals over neural pathways. Each neuron receives signals from other neurons through special junctions called synapses. Some inputs tend to excite the neuron; others tend to inhibit it. When the cumulative effect exceeds a threshold, the neuron fires and sends a signal down to other neurons.

A simple schematic of a neuron consists of a cell body that has a number of branched protrusions, called dendrites, and a single branch called the axon. Dendrites receive signals from other neurons. When these combined impulses exceed a certain threshold, the neuron fires and an impulse, or "spike," passes down the axon. Branches at the end of the axon form synapses with the dendrites of other neurons. The synapse is the point of contact between neurons. Synapses may be either excitatory or inhibitory.

This description of a neuron is excessively simple, but it captures those features that are relevant to neural models of computation. In particular, each computational unit computes some function of its inputs and passes the result along to connected units in the network. Instead of using explicit symbols and operations, the knowledge of the system emerges out of the entire network of neural connections and threshold values.

Neural architectures are appealing as mechanisms for implementing intelligence for a number of reasons. Traditional AI programs tend to be brittle and overly sensitive to noise: rather than degrading gracefully, such programs tend to either be right or fail completely. Human intelligence is much more flexible; we are good at interpreting noisy input, such as recognizing a face in a darkened room from an odd angle or following a single conversation in a noisy party. Even where a human may not be able to solve some problem, we generally can make a reasonable guess as to its solution. Neural architectures, because they capture knowledge in a large number of fine-grained units, seem to have more potential for partially matching noisy and incomplete data.

Neural architectures are also more robust because knowledge is distributed somewhat uniformly around the network.

Neural architectures also provide a natural model for parallelism, because each neuron is an independent unit. Finally, something is intrinsically appealing about approaching the problems of intelligence from a neural point of view. After all, the brain achieves intelligence and it does so using a neural architecture.

An artificial neuron models these simple biological characteristics. Each artificial neuron receives a set of inputs. Each input is multiplied by a weight analogous to a synaptic strength. The sum of all weighted inputs determines the degree of firing called the activation level. Notationally, each input X_i is modulated by a weight W_i and the total input is expressed as

$$\sum_i X_i W_i$$

or in vector form, **X** . **W** where $\mathbf{X} = [X_1, X_2, ..., X_n]$ and $\mathbf{W} = [W_1, W_2, ..., W_n]$. The input signal is further processed by an activation function to produce the output signal which, if not zero, is transmitted along. The activation function can be a threshold function or a smooth function like a sigmoid or a hyperbolic tangent function.

A neural network is represented by a set of nodes and arrows, which is a fundamental concept in graph theory. A node corresponds to a neuron, and an arrow corresponds to a connection along with the direction of signal flow between neurons. As illustrated in Figure 1, some nodes are connected to the system input and others are connected to the system output for information processing.

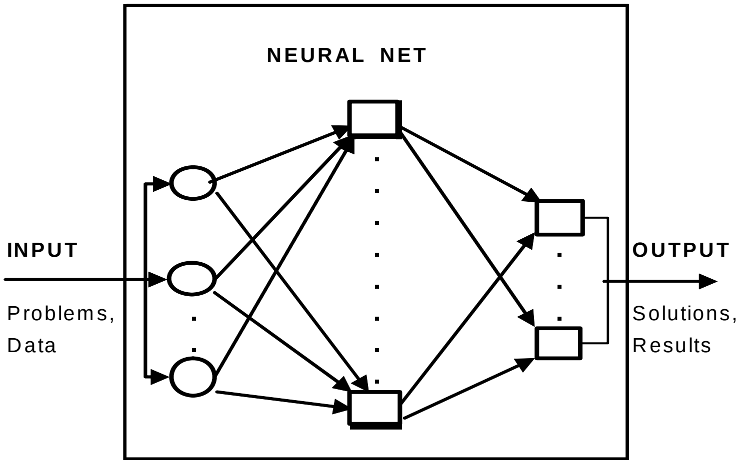

Fig. 1 Neural Information Processing

The dynamic behavior of the neural network is described by either differential equations or difference equations. The former representation assumes continuous time and can be used to simulate the network on an analog computer, whereas the latter uses discrete time and is usually taken to simulate the network on a digital computer.

A neural network has a parallel-distributed architecture with a large number of nodes and connections. Neural networks solve problems by self-learning and self-organization. They derive their intelligence from the collective behavior of simple computational mechanisms at individual neurons. Computational advantages offered by neural networks include:

- Knowledge acquisition under noise and uncertainty: Neural networks can perform generalization, abstraction, and extraction of statistical properties from the data.
- Flexible knowledge representation: Neural networks can create their own representation by self-organization.
- Efficient knowledge processing: Neural nets can carry out computation in parallel. It is known as parallel-distributed processing, or PDP. Special hardware devices have been manufactured which exploit this advantage. Thus, real-time operation is feasible. Notice that training a neural network may be time-consuming, but once it is trained, it can operate very fast.
- Fault tolerance: Through distributed knowledge representation and redundant information encoding, the system performance degrades gracefully in response to faults (errors).

Neural networks can recognize, classify, convert, and learn patterns. (A pattern is a qualitative or quantitative description of an object or concept or event).

Construction of a neural network

Construction of a neural network involves the following tasks:

- Determine the network properties: the network topology (connectivity), the types of connections, the order of connections, and the weight range.
- Determine the node properties: the activation range and the activation (transfer) function.

- Determine the system dynamics: the weight initialization scheme, the activation-calculating formula, and the learning rule.

Inference and learning algorithms

Building an AI system based on the neural network approach will generally involve the following steps:

1. Select a suitable neural network model based on the nature of the problem.
2. Construct a neural network according to the characteristics of the application domain.
3. Train the neural network with the learning procedure of the selected model.
4. Use the trained network for making inference or solving problems. If the performance is not satisfactory, then go to one of the previous steps.

Familiarity with existing applications will help determine the appropriate network architecture and select the best-suited computational model for learning, and inference.

Despite some specific variations with different models, general learning and inference procedures can be extracted as follows.

The Neural Network Learning Algorithm (A general view)

Given *n* training instances,

1. Initialize the network weights. Set i= 1.
2. Present the *i*th instance to the network on the input layer.
3. Obtain the activation levels of the output units using the inference algorithm (described next). If the network performance meets the predefined standard (or the stopping criterion), then exit.
4. Update the weights by the learning rule of the network.
5. If $i = n$, then reset $i = 1$. Otherwise, increment i by 1. Go to step 2.

The Neural Network Inference Algorithm (A general view)

Given a training instance,

1. Present the instance to the network on the input layer.
2. Calculate the activation levels of nodes across the network.
3. For a feedforward network, if the activation levels of all output units are calculated, then exit. For a recurrent network, if the activation levels of all output units become (near) constant, then exit; else go to step 2. However, if the network is found unstable, then exit and fail.

Neural computational models

Neural computational models can be categorized in terms of their applications: classification models, association models, optimization models, self-organization models.

3 CONCLUSIONS

Integration of symbolic AI and neural networks results in a so-called hybrid intelligent system. Under this approach, the fundamental assumptions on intelligence are as follows: Neither the physical symbol system nor the neural network is a necessary means for general intelligent action. The symbolic level and the connectionist level represent two different levels of abstraction for intelligent processes. Knowledge is power. Every intelligent being should have knowledge in one form or another.

Hybrid intelligence is a biologically plausible notion. Recall that humans store knowledge in certain complex molecules such as genes and proteins which determine what we are and how we behave, and at the same time, we have nerve systems to coordinate our behavior.

The presented paper is the result of the scientific grant projects solution KEGA No. 3/120103 and VEGA No. 1/2216/05.

REFERENCES

[1] ANDERSON, J. A., ROSENFELD, E. (eds.) (1998): *Neurocomputing*. MIT Press, Cambridge. MA, 1998.

[2] FELDMAN, J.A, and BALLARD, D.H. (1982). *Connectionist models and their, properties.* Cognitive Science, 6(3), pp. 205-254.

[3] GROSSBERG, S. (1980): *How does a brain build a cognitive code?* Psychological Review,87, pp. 1-51.

[4] HEBB, D.O.(1949): *The Organization oj Behavior.* Wiley, New York.

[5] HOPFIELD, J.J. (1982): *Neural networks and physical systems with emergent: collective computational abi1ities.* In Proceedings oj the National Academyoj/ Science, 79, pp. 2554-2558.

[6] KOHONEN, T. (1977): *Associative Memory: A System-Theoretical Approach,* Springer- Verlag, New York.

[7] MINSKY, M. (1954): *Neural Nets and the Brain-Model Problem*. Ph.D. thesis, Princeton University.

[8] MINSKY, M., and PAPERT, S. (1969): *Perceptrons.* MlT Press, Cambridge, MA

[9] PARKER, D.B., (1982): *Learning Logic.* Invention Report S81-64, File 1, Office of Technology Licensing, Stanford University.

[10] ROSENBLATT, F. (1958): *The Perceptron: A probabilistic model for information storage and organization in the brain.* Psychological Review,65, pp. 386-407.

[11] RUMELHART, D. E., Mc CLELLAND, J. L., and the PDP Research Group (eds.): *Parallel Distributed Processing.* Vol. 1 and Vol. 2. MIT Press, Cambridge. MA, 1986.

[12] RUMELHART, D.E., HINTON, G.E., and WILLIAMS, R.J. (1986): *Learning internal representation by error propagation.* In Parallel Distributed Processing: Explorations in the Microstructures oj Cognition, Vol. 1. MIT Press, Cambridge, MA.

[13] RUMELHART, D.E., Mc CLELLAND, J.L., and the PDP Research Group (eds.). (1986): *Parallel Distributed Processing: Explorations in the Microstructures oj Cognition,* Vol. 1 and Vol. 2. MIT Press, Cambridge, MA.

[14] SALOKY, T. (1999): *Applications of Machine Learning Techniques*. ELFA Košice.

[15] SALOKY, T.(2005): *Some Problems of AI Implementation*. (165-194 pp.) In: Taufer, I., Hrubina, K., Eds.: "*Optimal Control of Processes Based on the Use of Informatics Methods*". Informatech Košice, Pardubice 2005. ISBN 80-88941-30-X (pp. 287)

[16] SEJNOWSKI, T. and ROSENBERG, C.R. (1987): *Parallel networks that learn to pronounce English text.* Complex Systems, I, pp. 145-168.

[17] SMOLENSKY, P. (1988): *On the proper treatment of connectionism*. Behavioral and Brain Sciences, 11, pp. 1-23.

Reviewer: doc. Ing. Lenka Landryová, CSc., VŠB – Technical University of Ostrava