

INTEGRATION OF PERCEPTION AND REASONING IN FAST NEURAL MODULES

David G. Fritz
Research Scientist
Institute for Artificial Intelligence
The George Washington University
and
Cognitive Information Systems Co.
Silver Spring, Maryland

Abstract. *Artificial neural systems promise to integrate symbolic and sub-symbolic processing to achieve real time control of physical systems. Two potential alternatives exist. In one, neural nets can be used to front-end expert systems. The expert systems, in turn, are developed with varying degrees of parallelism, including their implementation in neural nets. In the other, rule-based reasoning and sensor data can be integrated within a single hybrid neural system. The hybrid system reacts as a unit to provide decisions (problem solutions) based on the simultaneous evaluation of data and rules.*

This paper discusses a model hybrid system based on the fuzzy cognitive map (FCM). The operation of the model is illustrated with the control of a hypothetical satellite that intelligently alters its attitude in space in response to an intersecting micrometeorite shower.

Concept. Artificial perception, cognition, and learning are increasingly possible by imitating natural information processing mechanisms. Information processing in living systems occurs in two major forms, genetic evolution and chemical/ electric cell-to-cell communication. Both of these natural processes are being exploited for what they can contribute to artificial computation. The field of Computational Genetic Algorithms has borrowed ideas from natural evolutionary theory to develop learning and problem-solving programs. Artificial Neural Systems (ANS) have taken inspiration from natural

nervous systems in order to model the parallel, distributed processing of the brain. These two currents of thought derive their power from the inherent parallelism of natural solutions to information processing.

Small, low cost information processors were first developed by natural living organisms. This tactic served developing life forms well for hundreds of millions of years. Only late in the evolutionary process did bundles of these small localized neural processors coalesce into large brains to which other peripheral knots of neurons could report in turn. This process of natural development can be emulated in artificial systems by initially producing small, intelligent information integrating devices able to categorize and classify local information, and which communicate and coordinate their state with a non-local decision-making center. Such devices would be capable of learning and reasoning, and of operating in continuous real-time.

Given the current investment in expert systems, and the relative immaturity of neural connectionist systems and their learning interfaces, practical applications in intelligent processing enhancement will probably consist initially of expert system/ neural net combinations, rather than of neural nets alone. There are two likely approaches to this near term scenario: symbolic expert systems with neural nets at the data collection points (front ending), and computational systems of mixed representation that allow the close

integration of concepts and rules with low-level data. Both of these approaches have relative strengths and weaknesses, but only the second is close to being "natural" in the sense discussed here.

The first of the two approaches consists of the straightforward combination of an expert system with whatever neural net models are needed to provide input at the data collection points of the expert system's rule graph. In its simplest form, the integrated program combines a normative expert system with selected ANS. Whereas the expert system typically queries a user or a data-base/ knowledge-base for information, the integrated program also queries, or extracts information from, neural nets. In this system, machine learning occurs at the sub-symbolic level in the neural nets. However, neural net input-output patterns can also be extracted as symbolic rules for incorporation into the expert system's rule graph as they are learned.

The mixed representation approach provides for systems that allow the close integration of high-level concepts with low-level data. These systems do both the data collation and a degree of symbolic level processing as a unit. The system thereby behaves as a symbolic/ sub-symbolic hybrid. The hybrid is different than the first approach described above in that a major portion of the data hybrid is implemented entirely under the ANS paradigm. The hybrid is different than a connectionist expert system in that the neural model does not simply replicate the functionality of an expert system, but is aimed at fusing in real-time the information provided by sensors and through conceptual relationships.

On this basis, the hybrid system's strength consists of the capability to provide a fine-grained integration of symbolic concepts with sub-symbolic information. Operations on the two types of knowledge occur at the lowest computational level. Incomplete, inaccurate or contradictory rules are buttressed by the natural fault tolerance/ graceful degradation of the neural

elements, providing for automatic truth-maintenance support. In addition, the automatic translation of sub-symbolic representations into symbolic rules can occur in the same computational neighborhood, such that the high level conceptual portion of the system learns as well and as easily as the neural elements do.

Both of these approaches are useful in their own right. But, whatever approach is chosen to enhance intelligent processing by way of neural models, it is important to address the art and practice of neuralism as well as the science. Science provides testable ideas but does no work. Useful work will result from realized improvements in practice by way of new ideas. Neural models provide two basic beneficial improvements: conceptual and associative learning from sub-symbolic information, and simultaneous processing activity. The first of these will be important over the long run in addressing the knowledge acquisition and maintenance bottleneck. The second is of more immediate relevance.

The standout value of connectionist systems is going to be in harnessing parallel behavior. It is important, therefore, to look to eventual hardware implementations of chosen neural models in order to provide this parallelism. Without a hardware realization, the often-used signal Hebb law is basically another algorithm, of the many excellent ones available. New algorithms may or may not be better than those that came before, depending on the problem to which they are applied. For example, some comparative studies have shown that neural emulation algorithms can be inferior to the methods they were devised to replace (Taber and Deich 1988). However, it is worth remembering that these and nearly all other software programs are currently designed to run on von Neumann computers. The real power of connectionist models will ultimately result from providing an escape from such serial machines.

Of the two practical approaches to intelligent processing enhancement mentioned in this section, the hybrid approach was selected to test the potential of intermingling data with rules in a compact modular fashion. It was chosen because its design can be small and straightforward, because it can integrate data and rules in a fine-grained fashion, and because it has more potential for providing these features in a fast, dedicated, neural device than more complex schemes.

To summarize this section, the concept that drives this investigation is that of integrating rules and data in a fine-grained, modular processing environment that has the potential of being realized in highly parallel "neural" hardware. An approach that will co-locate these integrating elements was chosen over one that would compartmentalize them.

The Science. The model selected for initial investigations of the concept problem is the fuzzy cognitive map (FCM) (Kosko 1988). The FCM is based on well known equations, and features input from fuzzy set theory, providing for an inherent credibility measure on its output. Its expert system capabilities have also been demonstrated (Taber and Siegel 1987).

The FCM is a single layer net from the family of unsupervised learning - feedback recall neural models. It can encode arbitrary patterns

$$A^k = (a_1^k, \dots, a_n^k), k = 1, 2, \dots, m,$$

using either hardwired or differential Hebbian learning (Kosko 1986). The topology is shown in Fig. 1.

Hardwired encoding requires that the connection strengths be initially determined off-line and selectively assigned. This encoding can be used to represent the symbolic concepts and the relationships among them. Signed values are provided in the range $[-1 \dots 1]$ to each lateral (synaptic) connection, where fuzzy positive values represent causal increase, fuzzy negative values represent causal

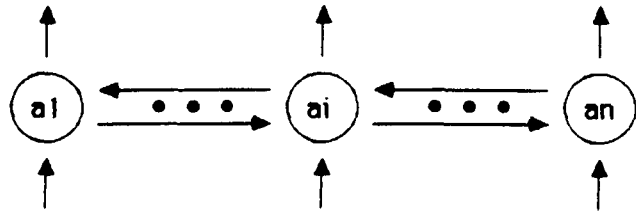


Fig. 1. The topology of the fuzzy cognitive map, (FCM). Input, output, and lateral connections are shown. The single layer Fa processing element array consists of neurons a_1 through a_n .

decrease, and zero represents no causal connection. In the basic model, once the causal values are assigned, they are not expected to change without further off-line intervention.

Adaptive encoding requires that the neural map automatically infer the connection values between patterns (data), and between patterns and concepts, using differential Hebbian learning. The learning algorithm correlates changes in processing element (neural) activations, such that only changes in activity in the same direction (either both increasing or both decreasing) will affect the connection (synaptic) weights. This activity is called concomitant variation. The encoding equation is

$$\dot{w}_{ij} = -d w_{ij} + \dot{S}(a_i^k) \dot{S}(a_j^k)$$

where w_{ij} is the connection strength from the i th to the j th neuron, a_i^k and a_j^k are the i th and j th components of the k th inference vector A^k or alternatively the activation levels of the i th and j th neurons, and $dS()/dt$ is the time derivative of a sigmoid function. The first term in the equation is passive decay and the second term is the differential Hebbian correlation term.

For recall, the additive STM (Short Term Memory) recall equation was chosen over

the more general additive recall equation for simplicity's sake. It is of the form

$$\dot{a}_i = -ba_i + \sum_{j=1}^n S(a_j)w_{ji} + I_i$$

where I_i is the i th component of the initial inference state or the i th data value. The first term in the equation is passive decay, the second term is lateral feedback, and the last term is external input.

Except for the fact that the FCM utilizes the differential rather than the signal Hebbian, its topology and function are nearly identical to the Additive Grossberg (AG) model. The AG is the simplest of a family of neural models culminating in the ART2 system. Given that ART2 is basically an evolved AG, this suggests a migration path for the FCM.

When encoded with concepts that are highly inter-related, the FCM does not exhibit stable point behavior, but exhibits oscillatory or limit cycle behavior instead. Limit cycles consist of two or more unique sets of neurons being repeatedly activated. The dynamics of the FCM are amenable to limit cycle stability analysis given a derivation of the Lyapunov energy function (Simpson 1988). In practice, with no further external input the activation cycle soon decays as the network becomes de-energized.

The Art. No working neural network emulation program can be produced solely and directly from the formulas given above. As a working FCM model was developed by the author based on these equations, it is helpful to see how this was done.

The first issue is the choice of sigmoid function. The following function was found to produce satisfactory results

$$S_i = 1 / (1 + e^{-a_i})$$

where a_i is the activation of the i th neuron. When incorporated into the encoding equation, the following algorithm results

$$w_{ij} = w_{ij} + (((S_i * (1 - S_i)) * \Delta w_{ij}) * ((S_j * (1 - S_j)) * \Delta w_{ji}) - (w_{ij} * d))$$

where w_{ij} is the synaptic strength from the i th to the j th neuron, Δw_{ij} is the change in w_{ij} over the last unit time period, and d is a decay term.

Another issue is the training method. As, under training, encoded patterns that are not continuously reinforced tend to decay, it is preferable to present the patterns in an interleaved fashion rather than in batch mode. If patterns a , b , and c are in the training set, they are input for encoding as $a, b, c, a, b, c, a, b, c$ rather than as $a, a, a, b, b, b, c, c, c$. Furthermore, unless the synaptic connections affecting a , b , and c are clamped after training, to train on a further pattern d would require resurrecting these earlier patterns for training as well.

Decay terms must also be set. Too great a rate of decay and the neural net never develops enough energy to activate categorizing neurons or to fire rules. Too low a rate of decay and the neural net becomes overheated, activating and firing any number of neurons that bear only a weak relationship to the knowledge being recalled. In practice it was found that decay factors in the range of 0.1 to 0.2 were most suitable.

As training is mediated by a sigmoid function, synaptic weights eventually approach asymptotic values. It is often the practice to stop training when the rate of change of a weight falls below an arbitrary value, ϵ . The results reported later in this paper were achieved with an ϵ of 0.001.

To simulate simultaneous updating of synaptic and activation values (simulated parallel behavior), the new values and new delta values are found for the entire neural network before the updating of any neurons or synaptic connections occurs. This keeps the neurons from affecting one another until the entire network is ready to move. In contrast to "instantaneous" updating, spreading activation would

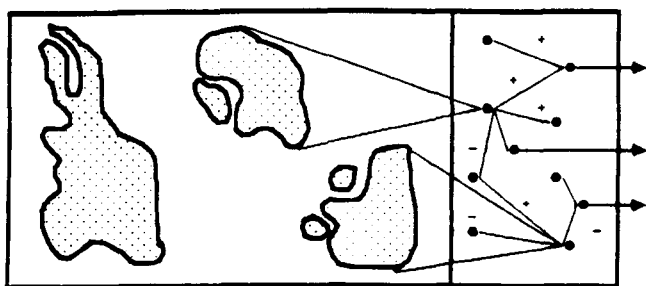


Fig. 2. The outline of a model neural net in computer memory. Dispersed data patterns (left) are mapped to concept neurons (right). Fuzzy rules are represented by the inter-relation of concepts.

produce different and less predictable activity in the network. Although spreading activation is a valid approach and is sometimes used, it was not implemented in this study.

Finally, using a discrete, serial algorithm to simulate the passage of time is no more than declaring each iteration of the network (a single training or activation pass) to be a time "unit". The change in synaptic weight or in neural activation is represented by a delta value, as found by the equations in the previous section. The time derivative value is this delta value, with respect to a single iterative time unit.

Operation. The operation of the above system is fairly straightforward. A network of processing elements or neurons is mapped out and cast in computer memory. A certain predetermined portion is then set aside for the data driven elements. Data elements can be visualized as occupying this space from top to bottom. The remainder of the network is reserved for concepts and rules (Fig. 2).

All neurons are initialized to a base or resting activation state of zero. A vigilance parameter is set to detect neurons with significant activity, the level of significance being given by the vigilance value.

Concepts are elicited, and the relationships between them set. For example, if the concept audible snarl (a_i) is thought to be twice as important as the concept fangs (a_j) in implying the concept wave big stick (a_h), then w_{ih} becomes $+0.66$ and w_{jh} is $+0.33$. The available range of values $[-1...1]$ allows for fuzzy adjustments to this rule as long as the ratio of w_{ih} to w_{jh} remains approximately 2:1 or whatever it is judged to be.

Once set, these weights are clamped while the net as a whole is trained on data sets. Training is achieved by presenting an "analog" input pattern to the net while simultaneously turning one or more of the concepts cells on. The "analog" input consists of a dispersion of data points that take on real values in $[-1...1]$ such that a pattern is created. This pattern is then mapped (input) to the corresponding neurons in the net. Activated pattern neurons reinforce their relationship to various degrees with the firing concept neurons, until the rate of change falls below ϵ . This operation completes the linking of concept neurons to data neurons.

After training, many of the concept neurons may be thickly connected to dispersed areas of the data portion of the neural map. However, some of the concept neurons may only be connected to other such neurons and not at all to data cells. These neurons can be activated directly by conceptual input if they are input cells, but only indirectly by data acting through other concept neurons if they are not.

During recall, input occurs to various neural elements in the network. Typically, a continuous and changing (time variant) "analog" data pattern is read into the net, while certain concepts may be simultaneously turned on or off. Settling of the network occurs continuously as input is read in. This activity is represented in Fig. 3, where both types of input are shown.

Reporting neurons can be of various types, depending on how the neural net was trained. Some models, such as the AG,

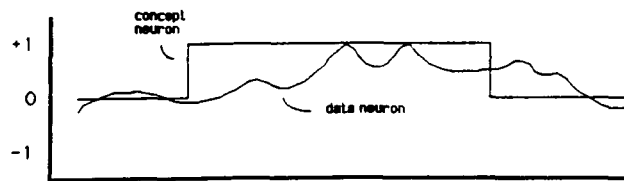


Fig. 3. Recall input to a model neural net. Concept neurons are either switched on or off, the activity of a data neuron fluctuates as real input data arrives at the system.

utilize a winner-take-all approach (a max function) to select the one cell whose output will be recognized. The system described here currently uses a vigilance threshold to detect all firing neurons over that threshold.

Test programs have been run on a Sun-3 workstation with enough memory to load a medium-sized universally interconnected net, but with no floating-point hardware. Speeds of approximately 12K IPS (interconnects per second) were achieved on an unloaded station.

Results. The system just described has been exercised via a test program based on a hypothetical space vehicle problem. The devised task is to orient an object in space such that optimal mission-sensitive behaviors can be maintained. To simplify things, the problem space was reduced to the object's purported vulnerability to particle bombardment, without regard to type or source. The problem space consists of a few identifiable surface structures on the object, a few internal operational characteristics, and a surface mapping of the object's "skin", with sufficient sensors to detect arriving particles such as micrometeorites. It should be possible under this scenario to train a neural module of the type discussed here to "recognize" vulnerable portions of its host's surface, to rank these areas in order of importance, and to offer suggestions as to appropriate orientational responses for any given micro-impact situation. If we

recall the underlying thesis of this study, that of hardware realizability, the above response should realistically occur in nanosecond time frames.

The test set for which the results are presented below was based on a small rule set and the following concepts: power (available, unavailable), mission critical communications (occurring, not occurring), and the sensitive surface structures camera lens, receiving antenna, and solar cell array. These concepts and the relationships among them were knowledge engineered into the system (as described in the section Operation). Following this, the system was trained to recognize the surface structures by turning on concepts while the "sensor" - neural net component was "bombarded" with time and space variant real-valued impact patterns. In such an artificial situation care was taken to ensure that the bombardment was not random, but massed on certain areas of the sensor (data) component of the neural system. In this way, learning could occur.

Once the system was trained, test recall could be performed. Fig. 4 presents the results of one such test. An initial single spike was delivered to the system at time $t=0$ centered on, but not coincident with,

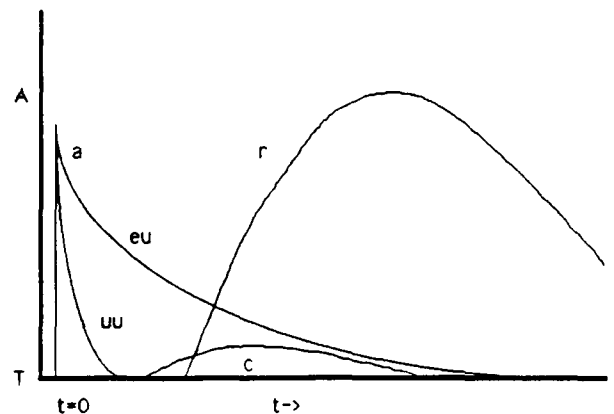


Fig. 4. The results of activating a model neural net. An activation spike a energizes cells that have been trained to a concept c (eu) and cells that have not (uu). The response cell r is activated as a result of rule firing within the system. T represents the activation (A) threshold, t is time. The figure is to scale, units are not shown.

one of the surface structures, camera. The activated neurons that represent this area of the net begin to immediately decay, the cells trained on the camera falling off more slowly than those that are not. Soon, the concept neuron representing the target concept, camera, is activated, and crests just slightly over the threshold. Almost as quickly the attached rule fires, provoking the response neuron *r* to activate. Within a 100 or so time iterations, most activity has subsided below threshold, and the system returns to its normal resting state.

In another test, input was provided over a period of time rather than as a single event. The results were similar to those above except that as more energy was being input to the system, the vigilance threshold had to be raised to mask the activity of neurons fired by weak intermediate connections (rules) in the system.

The above tests were performed with input concepts switched off. Tests run with an input neuron switched on also produced expected results— the appropriate rules fired and the correct response (asserted) neurons reported output. However, the time when the concept input neuron was switched was important. If switched on too early or too long, the neural system became energized around this neuron, such that the system became overly sensitive to rules that had this neuron as a component. If it was switched on too late or not long enough, the relevant rule would not fire. One way to adjust for this observed behavior is to reconfigure the rule(s) or the decay rate to conform to the desired level of sensitivity.

In summary, the results show that it is possible to integrate data and rules in a single neural net and achieve expected outputs. Given the small scale of these tests, however, the real problem of scaling the system to operational sizes remains unanswered. There are two issues that have a bearing on the question of scalability, and they are addressed in the next section.

Conclusions and Forecast. The evidence, albeit preliminary, seems to suggest that the implementation of the problem concept put forward in this paper is feasible. Pattern recognition and some other forms of sub-symbolic data processing are well known strengths of neural networks. And, there is enough experience with heuristic processing to feel comfortable with small sets of rules that can be easily understood by one person. Bringing them together in some fashion should draw from the strengths of both without incurring any of the major problems of either.

The issue of scalability is a factor, however. If necessary, scaling up may be done uniformly, by merely extending the neural model to ever greater numbers of neurons and IV's (interconnect values). Or, growth can be achieved through a system of inter-related, interconnected subcomponents, which need not be similar to one another.

For the moment, neural processors can be kept tractable and the major problems just alluded to kept to a minimum by targeting projects of moderate scale. If a target device is task oriented, data mapping will be of a predetermined size and kind that reduces the eventual complexity and hence the size of the neural architectural model.

Two issues that bear on ultimate size are both derived from the expert system experience. Because of the well known cost of knowledge acquisition and management (getting it, maintaining it, and ensuring that it continues to work right), rules are best kept to a small stable set, particularly in environments that are not highly human being interactive. And, if fast, compact, easily understood, task-oriented modular systems are desirable, it does not make sense to engineer a huge number of rules. Large knowledge based systems are not fast, and they are certainly not compact.

By limiting the system to a small size, however, something must be bought in exchange. What this may be is the possibility of focusing the selected model

on a simple, well defined task, such that the system's overall operation can be optimized.

Eventually it may be possible to build large neural systems and to create bushy architectures of symbols and their relationships through natural, sub-symbolic learning. Until then, there is some promise that these capabilities can be provided in a small, focused manner in fast, compact, task-driven neural modules.

References.

Kosko, B. 1986. Differential Hebbian learning. AIP Conference Proceedings: Neural Networks for Computing, J.S. Denker [ed.], p.277-282, Snowbird, Utah, April, 1986.

Kosko, B. 1988. Hidden patterns in combined and adaptive knowledge networks. *International Journal of Approximate Reasoning* 2(1988):377-393.

Simpson, P. K. 1988. A review of artificial neural systems II: paradigms, applications, and implementations. May be requested from the author at General Dynamics, San Diego.

Taber, W. R., and R. O. Deich. 1988. A comparison of feedforward neural networks: fuzzy operators and acoustic ship signatures. In review, *International Journal of Intelligent Systems*, June 1988. May be requested from the authors at General Dynamics, San Diego.

Taber, W. R., and M. A. Siegel. 1987. Estimation of expert weights using fuzzy cognitive maps. *IEEE International Conference on Neural Networks II*(1987):319-329.