Localized Self-Contained Adaptive Networks for Hybrid-Symbolic Reasoning

Yousuf C. H. Ma and Ronald F. DeMara
Dept. of Electrical and Computer Engineering
University of Central Florida
P. O. Box 162450
Orlando, FL 32816-2450
E-mail: rfd@ece.engr.ucf.edu

Abstract

Hybrid-Symbolic processing has been gaining interest over the past decade. This is due to the problems of symbolic representations which are ambiguous, brittle, lack of learning capabilities, and have low availability of parallelism. Sub-symbolic representations have problems of lacking variable binding, symbolic composition and decomposition, and structured representations. Integration of these two representations can mitigate each other's shortcomings. The proposed paradigm: Localized Self-Contained Adaptive Networks (LSCAN) is a localist network using *AND* and *OR* evaluators to represent relations between knowledge entities. For optimization of each sub-network, the LSCAN provides learning capabilities for both of feed-forward and lateral relations between network nodes.

1 Introduction

Symbolic processing directly emulates high-level human cognitive processes. Inputs to symbolic processing systems are typically representations in the form of character strings. Hence, real world information at high cognitive levels is readily represented and stored. On the other hand, connectionist, or sub-symbolic processing, replaces fixed symbols with dynamic numerical values, and process those numerical patterns among connected simple processing nodes. Each node constructs an output based on some contribution of its total input strengths. Connections between nodes are associated with numerical weights which can be adjusted through a systematic learning algorithm. Connectionist models provide advantages of learning, handling incomplete information, and parallel processing. However, inputs to sub-symbolic processing systems consist of numerical data which are low-level representations not directly discernible by human reasoning processes.

The combinations of symbolic and sub-symbolic paradigms gain advantages which can not be obtained by each paradigm alone. Since a decade ago, researchers have been exploring hybrid-symbolic systems and obtaining some fruitful results such as CONSYDERR[10] and SHRUTI[9]. The CONSYDERR was designed using two-leveled networks. The upper level is a localist network, each network node represents a concept. This level is called as a conceptual level. The lower level contains nodes which are the features of the nodes in the conceptual level. This lower level is called as a subconceptual level. The CONSYDERR paradigm provided some good features which are vital in artificial intelligence processing. The most promising features are similarity reasoning, top-down and bottom-up inheritance. The SHRUTI is a network to process predicate logic based systems. Each network node represents one predicate clause. The facts are connected to the related network nodes to initiate node firing. The SHRUTI resolves the decision-brittleness and sequential processing of predicate logic. However, more research remains to integrate various learning schemes.

The proposed paradigm: *Localized Self-Contained Adaptive Networks* (LSCAN) provides a more generic view of knowledge representation and reasoning. The LSCAN addresses the problems of ambiguity, brittleness, lack of learning capabilities, and low availability of parallelism. It also provides feed-forward and lateral learning schemes to optimize the relations between network nodes.

2 LSCAN Configuration

Knowledge entities are represented by interconnected network nodes, one node for each knowledge entity or concept. The LSCAN system propagates numerical values among the network nodes while maintaining high-level symbolic structures. Each node in the LSCAN network is sufficient to make its own decision from its own local information without global information. In addition to flexibilities of decision making possessed by localist connectionist networks, LSCAN systems provide learning capabilities. At the node level, each node associates a function to derive the outputs from the input data.

2.1 Knowledge Representation

In LSCAN, knowledge is represented as groups of knowledge entities and entity relations. Each entity is either a conjunction of other knowledge elements, which represents the AND relation, or each entity is a disjunction of an alternative of several knowledge elements, which represents the OR relation. In this definition, a knowledge entity can be decomposed into fine elements; or many fine elements together can be abstracted into a higher level knowledge entity. Let **n** represent the number of entities in a specific knowledge domain. If an entity vector $E(e_1, e_2, e_3, \ldots, e_n)$ includes all required entities in a knowledge domain then the knowledge in a specific knowledge domain can be represented in the following forms:

1. $e_i \Longrightarrow e_j$.
2. e_1 `AND` e_2 `AND` e_3 `AND` $\ldots \Longrightarrow e_j$
3. e_1 `OR` e_2 `OR` e_3 `OR` $\ldots \Longrightarrow e_j$
4. (e_1 `AND` e_2 `AND` $\ldots$) `OR` (e_2 `AND` e_9 `AND` $\ldots$) `OR` $\ldots \Longrightarrow e_j$
5. (e_1 `OR` e_2 `OR` $\ldots$) `AND` (e_7 `OR` e_{10} `OR` $\ldots$) `AND` $\ldots \Longrightarrow e_j$
6. $e_i \Longrightarrow e_j, e_k, e_l, \ldots$
7. e_1 `AND` e_2 `AND` e_3 `AND` $\ldots \Longrightarrow e_j, e_k, e_l, \ldots$
8. e_1 `OR` e_2 `OR` e_3 `OR` $\ldots \Longrightarrow e_j, e_k, e_l, \ldots$

where $1 \leq i, j \leq n$, and $\Longrightarrow$ means implication. Each of the above forms is called an *implication rule*, The word implication is used to indicate that the assertion of an entity is depending on the strengths of the input entities. However, to simplify the discussion, the word, *rule*, is used to represent a LSCAN knowledge representation form. The rule #1 uses a single entity. The rule #2 uses more than two entities combined by the operator AND. The rule #3 uses more than two entities combined by the operator OR. The rules #4 and #5 use mixed combinations of the rules #3 and #4. Thus, representations of the rules #4 and #5 can be resolved by the representations of the rules #2 and #3. The rule #6 represents the one-to-many mapping. The rule #7 represents the many-to-many mapping using AND operator. The rule #8 represents the many-to-many mapping using OR operator. Therefore, the LSCAN knowledge representations can be represented in the forms of the rules #1, #2, #3, #6, #7, and #8.

Each entity carries two kinds of information. One is the physical values of an entity. The other information is the belief factor about an entity. There are three types of physical values: a binary number, a real number, and a character string. The belief factor is a real number between 0 and 1.

2.2 Reasoning Mechanisms

The LSCAN reasoning mechanism manipulates belief factors explicitly between knowledge entities. The assertion of the LSCAN reasoning mechanism is based on the values of belief factors among a set of entities. The largest belief factor wins the assertion of an information query.

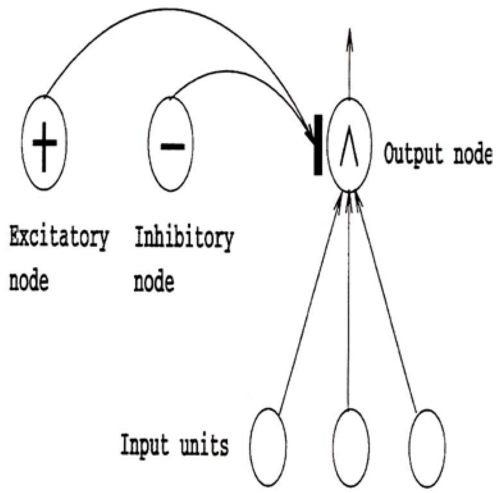

Figure 1: A Complete AND E-node Structure

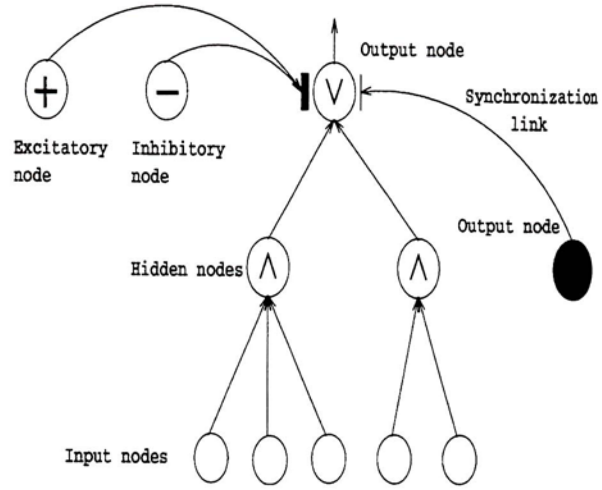

Figure 2: A Complete OR E-node Structure

2.3 Structures of Evaluator Nodes

The AND operator obtains the summation of the attribute entity strengths. Let ϵ represent the summation for the entity node e_i. Then, the mathematics expression for the AND operator is:

$$\epsilon_i = \sum_{j=1..n} \xi_j w_{ji} \tag{1}$$

where $\xi_j, j = 1..n$, are n belief factors of the active entities connected to the entity e_i and w_{ji} are the connection strengths between the entities e_j and the entity e_i. A complete ANDed evaluating node is illustrated in Figure 1.

The OR relation indicates sufficient conditions which are not always necessary. If an entity has OR relations with sets of ANDed entities then the entity picks up the strongest set as its belief factor. The OR operator is analogous to a local *winner-get-all* [8] operation. The locality of competition is among a small set of ORed neural units. As illustrated in Figure 2, each hidden unit gets the summation of its input strengths by applying Equation 1. Then, the output unit picks up the largest strength ϵ_{ik} from the hidden unit e_{ik}. The mathematics expression for the OR operator is:

$$\epsilon_i = MAX_{k=1..m}\,(\epsilon_{ik}) \tag{2}$$

The NOT operator inverts the belief factor of an entity. The mathematics expression for the NOT operator is:

$$\overline{\xi}_j = 1 - \xi_j \tag{3}$$

where ξ_j is the belief strength of the entity e_j and $\overline{\xi}_j$ is the inverted belief strength of the entity e_j.

Each entity is embeded in an evaluator node (E-node). Let ξ_i be the output strength of the entity e_i, and ξ_i is expressed as:

$$\xi_i = \left(\frac{1}{1+e^{-(\varepsilon_i - \delta_{sh})}}\right)\left(\frac{\varepsilon_i}{\delta_{sf}}\right) \tag{4}$$

where ε_i is a normalized equation from Equation 1 based on the ratio of number of input connections d_{in} and the constant δ_{sf}; in mathematical expression:

$$\varepsilon_i = \epsilon_i \left(\frac{\delta_{sf}}{d_{in}}\right) \tag{5}$$

The constant δ_{sf} is a pre-selected constant number in order to scale the first factor in Equation 4 in the range of [0, 1]. An appropriate default value is $\delta_{sf} = 28.4$ which is near saturation of the function:

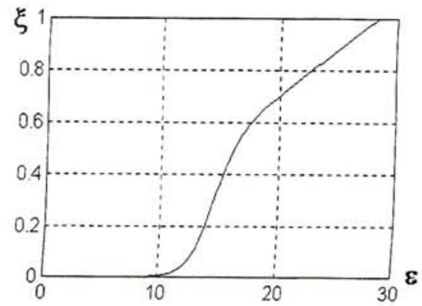

Figure 3: *Scaled-Shift-Right Sigmoid Function*

$$\frac{1}{1+e^{-(\varepsilon-\delta_{sh})}} \tag{6}$$

where δ_{sh} is equal to $\frac{\delta_{sf}}{2}$. Equation 4 is called a *Scaled-Shift-Right Sigmoid* function which maps inputs to an output value in the range of [0, 1]. The plot for Equation 4 is illustrated in the Figure 3. Equation 4 is for an ANDed E-node which is illustrated in the Figure 1. For an ORed E-node, the mapping function or activation function is similar to Equation 4 and is expressed as:

$$\xi_i = \left(\frac{1}{1+e^{-(\varepsilon_{ik}-\delta_{sh})}}\right)\left(\frac{\varepsilon_{ik}}{\delta_{sf}}\right) \tag{7}$$

There are also three lateral relations between the LSCAN E-nodes: excitatory, inhibitory, and synchronization interactions. The excitatory interactions permit an E-node to excite other E-nodes. On the other hand, the inhibitory interactions discourage other E-nodes. As illustrated in the Figure 1 and Figure 2, both of excitatory and inhibitory interactions can coexist resulting in the summation of the excitatory and inhibitory strengths. If the E-node e_i has p excitatory interactions and q inhibitory interactions, then the total influence of the other E-nodes on the E-node e_i is:

$$\rho_i = \sum_{j=1\ldots p} \epsilon_j \omega_{ji} - \sum_{k=1\ldots q} \epsilon_k \omega_{ki} \tag{8}$$

where ω_{ji} and ω_{ki} are lateral connection strengths between the E-node e_i and other E-nodes e_j and e_k.

If ρ_i is positive then ρ_i is mapped into the range [1, 2] with the mapping factor η_i. Then, η_i is used to magnify the output strength of the E-node e_i by multiplying the mapping factor η_i. The mapping factor η_i is given as:

$$\eta_i = \left(\frac{-1}{1+e^{-(\delta_{lh}-\rho_i)}}\right)\left(1-\frac{\rho_i}{\delta_{lf}}\right)+2 \tag{9}$$

where δ_{lf} is a constant value which makes η_i just about saturation. A good practical value for δ_{lf} is 10 so that $\delta_{lh} = \frac{\delta_{lf}}{2}$.

If ρ_i is negative, then ρ_i is mapped into the domain [0, 1] which is represented by the mapping factor η_i. The mapping factor η_i is multiplied to the output strength of the E-node e_i. Therefore, the E-node e_i is discouraged. In this case, η_i is given as:

$$\eta_i = \left(\frac{1}{1+e^{-(\delta_{lh}-\rho_i)}}\right)\left(1-\frac{\rho_i}{\delta_{lf}}\right) \tag{10}$$

Equation 9 and Equation 10 are mirror functions of each other. The result of η_i is applied to Equation 4 and Equation 7. The output ξ_i of an E-node e_i becomes:

$$\xi_i = \xi_i \eta_i \tag{11}$$

The lateral synchronization connection is always two-valued: a connection is either active or inactive. If an E-node e_i is active while a lateral synchronization connection is asserted from the other active E-node e_j, then the E-node e_i is inactivated regardless the activation strength of the E-node e_i. The lateral synchronization connection is required when an ORed subnetwork has different depths in its branches.

3 Learning Mechanisms

The LSCAN systems support learning from the input units of an E-node and lateral connections. The inputs contribute to learning by allowing partial sets

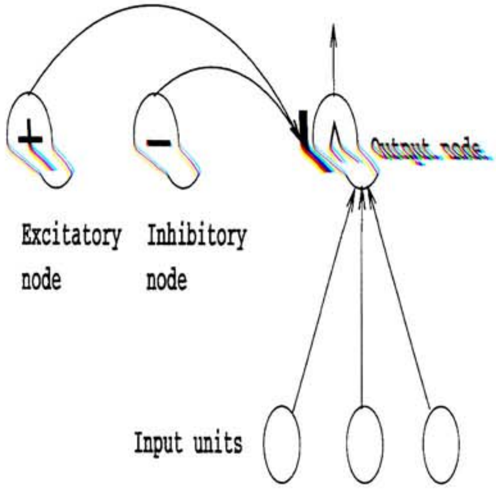

Figure 1: A Complete AND E-node Structure

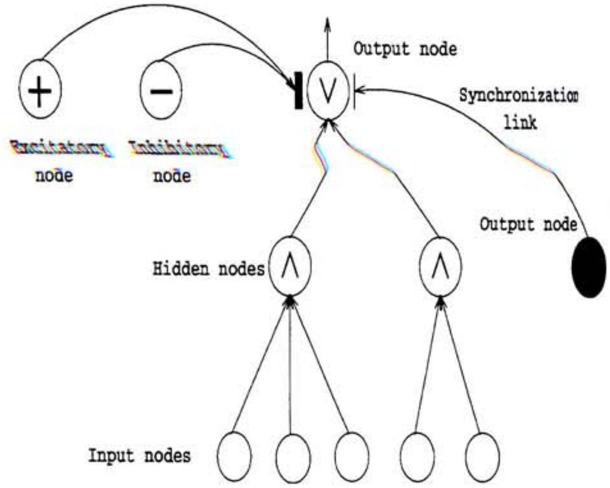

Figure 2: A Complete OR E-node Structure

2.3 Structures of Evaluator Nodes

The AND operator obtains the summation of the attribute entity strengths. Let ϵ represent the summation for the entity node e_i. Then, the mathematics expression for the AND operator is:

$$\epsilon_i = \sum_{j=1..n} \xi_j w_{ji} \tag{1}$$

where $\xi_j, j = 1..n$, are n belief factors of the active entities connected to the entity e_i and w_{ji} are the connection strengths between the entities e_j and the entity e_i. A complete ANDed evaluating node is illustrated in Figure 1.

The OR relation indicates sufficient conditions which are not always necessary. If an entity has OR relations with sets of ANDed entities then the entity picks up the strongest set as its belief factor. The OR operator is analogous to a local *winner-get-all* [8] operation. The locality of competition is among a small set of ORed neural units. As illustrated in Figure 2, each hidden unit gets the summation of its input strengths by applying Equation 1. Then, the output unit picks up the largest strength ϵ_{ik} from the hidden unit e_{ik}. The mathematics expression for the OR operator is:

$$\epsilon_i = MAX_{k=1..m}(\epsilon_{ik}) \tag{2}$$

The NOT operator inverts the belief factor of an entity. The mathematics expression for the NOT operator is:

$$\overline{\xi}_j = 1 - \xi_j \tag{3}$$

where ξ_j is the belief strength of the entity e_j and $\overline{\xi}_j$ is the inverted belief strength of the entity e_j.

Each entity is embeded in an evaluator node (E-node). Let ξ_i be the output strength of the entity e_i, and ξ_i is expressed as:

$$\xi_i = \left(\frac{1}{1+e^{-(\varepsilon_i - \delta_{sh})}}\right)\left(\frac{\varepsilon_i}{\delta_{sf}}\right) \tag{4}$$

where ε_i is a normalized equation from Equation 1 based on the ratio of number of input connections d_{in} and the constant δ_{sf}; in mathematical expression:

$$\varepsilon_i = \epsilon_i \left(\frac{\delta_{sf}}{d_{in}}\right) \tag{5}$$

The constant δ_{sf} is a pre-selected constant number in order to scale the first factor in Equation 4 in the range of [0, 1]. An appropriate default value is $\delta_{sf} = 28.4$ which is near saturation of the function:

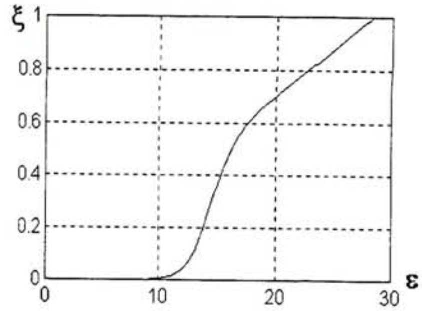

Figure 3: Scaled-Shift-Right Sigmoid Function

$$\frac{1}{1+e^{-(\varepsilon-\delta_{sh})}} \tag{6}$$

where δ_{sh} is equal to $\frac{\delta_{sf}}{2}$. Equation 4 is called a *Scaled-Shift-Right Sigmoid* function which maps inputs to an output value in the range of [0, 1]. The plot for Equation 4 is illustrated in the Figure 3. Equation 4 is for an ANDed E-node which is illustrated in the Figure 1. For an ORed E-node, the mapping function or activation function is similar to Equation 4 and is expressed as:

$$\xi_i = \left(\frac{1}{1+e^{-(\varepsilon_{ik}-\delta_{sh})}}\right)\left(\frac{\varepsilon_{ik}}{\delta_{sf}}\right) \tag{7}$$

There are also three lateral relations between the LSCAN E-nodes: excitatory, inhibitory, and synchronization interactions. The excitatory interactions permit an E-node to excite other E-nodes. On the other hand, the inhibitory interactions discourage other E-nodes. As illustrated in the Figure 1 and Figure 2, both of excitatory and inhibitory interactions can coexist resulting in the summation of the excitatory and inhibitory strengths. If the E-node e_i has p excitatory interactions and q inhibitory interactions, then the total influence of the other E-nodes on the E-node e_i is:

$$\rho_i = \sum_{j=1\ldots p} \epsilon_j \omega_{ji} - \sum_{k=1\ldots q} \epsilon_k \omega_{ki} \tag{8}$$

where ω_{ji} and ω_{ki} are lateral connection strengths between the E-node e_i and other E-nodes e_j and e_k.

If ρ_i is positive then ρ_i is mapped into the range [1, 2] with the mapping factor η_i. Then, η_i is used to magnify the output strength of the E-node e_i by multiplying the mapping factor η_i. The mapping factor η_i is given as:

$$\eta_i = \left(\frac{-1}{1+e^{-(\delta_{lh}-\rho_i)}}\right)\left(1-\frac{\rho_i}{\delta_{lf}}\right)+2 \tag{9}$$

where δ_{lf} is a constant value which makes η_i just about saturation. A good practical value for δ_{lf} is 10 so that $\delta_{lh} = \frac{\delta_{lf}}{2}$.

If ρ_i is negative, then ρ_i is mapped into the domain [0, 1] which is represented by the mapping factor η_i. The mapping factor η_i is multiplied to the output strength of the E-node e_i. Therefore, the E-node e_i is discouraged. In this case, η_i is given as:

$$\eta_i = \left(\frac{1}{1+e^{-(\delta_{lh}-\rho_i)}}\right)\left(1-\frac{\rho_i}{\delta_{lf}}\right) \tag{10}$$

Equation 9 and Equation 10 are mirror functions of each other. The result of η_i is applied to Equation 4 and Equation 7. The output ξ_i of an E-node e_i becomes:

$$\xi_i = \xi_i \eta_i \tag{11}$$

The lateral synchronization connection is always two-valued: a connection is either active or inactive. If an E-node e_i is active while a lateral synchronization connection is asserted from the other active E-node e_j, then the E-node e_i is inactivated regardless the activation strength of the E-node e_i. The lateral synchronization connection is required when an ORed subnetwork has different depths in its branches.

3 Learning Mechanisms

The LSCAN systems support learning from the input units of an E-node and lateral connections. The inputs contribute to learning by allowing partial sets of inputs to result in the firing of an E-node, or learning from incomplete information. Supervised learning and unsupervised learning schemes are both provided within the LSCAN paradigm. Lateral learning occurs when two or more conflicting E-nodes resolve the conflict by allowing one or more than one E-node to fire. Learning schemes adjust the weights of the links between E-nodes, as discussed below.

3.1 Supervised Learning

Each firing E-node adjusts its own input connection weights based on the difference between the actual and the expected output strengths and the number of active inputs. If the inputs of an E-node are not all active, then this E-node is learning how to deal with incomplete information. Let an ANDed E-node, e_i, have n input units with the input connection weights, ω_{ji}. Also, let μ_i be the expected output strength and the actual output strength be ξ_i. At the time of firing the E-node e_i, the number of active input units is m and m is less than n. In order to adjust each weight of ω_{ji}, ν_i has to be computed from μ_i using Equation 4; ξ_i is replaced by μ_i and ε_i is replaced by ν_i. However, Equation 4 is not reversible, so the Newton-Raphson [3] method is used to find an approximate ν_i from μ_i. Let the difference between ν_i and ε_i be $\Delta\varepsilon_i = \nu_i - \varepsilon_i$. The adjustment of weights $\omega_{ji(new-active)}$ is proportional to the radios of $\frac{\nu_1}{\delta_{sf}}$ and $\frac{\Delta\varepsilon_1}{m\xi_{ave}}\frac{n}{\delta_{sf}}$. The final weight adjustment for active input units is:

$$\omega_{ji(new-active)} = \omega_{ji(old-active)} + \frac{n\nu_i}{\delta_{sf}^2 m}\frac{\Delta\varepsilon_i\lambda}{\xi_{ave}} \quad (12)$$

where ξ_{ave} is the average input strength and λ is a learning rate which limits the learning step size. The value of λ is less than 1 and greater than 0. In the case of $\lambda = 1$ only one learning step is required. A smaller λ creates a longer training period. On the other hand, weights of inactive input units are adjusted based on the total amount weight change for active input units. Then, for inactive input units, their weights are adjusted as:

$$\omega_{ji(new-inactive)} = \omega_{ji(old-inactive)} - \frac{1}{n-m} \times \left(\sum_{j=1\ldots m} \left(\omega_{ji(new-active)} - \omega_{ji(old-active)}\right) \right) \quad (13)$$

In the case when $n = m$ which implies all input units of an E-node are active. Equation 13 is not necessary. The weight adjustment, can be based on each input strength; stronger input units gain more weight; on the other hand, weaker input units lose weights. Let ξ_{ave} be the average input strength for the E-node e_i. Then, the weight adjustment is given as:

$$\omega_{ji(new)} = \omega_{ji(old)} + \left(\frac{\xi_j}{\xi_{ave}} - 1\right)\frac{\nu_i}{\delta_{sf}^2}\frac{\Delta\varepsilon_i\lambda}{\xi_{ave}} \quad (14)$$

The training process proceeds until all supervised E-nodes reach the expected outputs. For an ORed E-node, learning occurs on the winning hidden unit.

3.2 Unsupervised Learning

During unsupervised learning, expected values are not assigned. However, the systems allow β E-nodes to fire in each firing cycle. Let γ be the number of E-nodes fired in one firing cycle. If γ is greater than or equal to β, then no any E-node needs to be trained. Otherwise, $(\beta - \gamma)$ E-nodes with the highest output strengths are trained under the condition of ς. If ς equals 1 then learning is exercised. For the E-node e_i, the condition ς_i is given as:

$$\varsigma_i = \begin{cases} 1 & \text{if } \xi_i > \alpha\theta_i \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

where ξ_i is the output strength of e_i and α is a factor to control whether an E-node has to be trained. The value of α is between 0 and 1. Under an unsupervised learning scheme, an E-node is trained only when the actual output strength ξ_i is greater than a fraction of the threshold θ_i; otherwise, an E-node is not trained. Equations 12, 13, and 14 are used to change the input weights of e_i. However, the expected output strength μ_i is replaced by $\theta_i + \Delta_i$, where Δ_i is a small given value for the purpose of the training of e_i.

3.3 Competitive Learning

Competitive learning occurs at the lateral level. At each firing stage, multiple E-nodes are fired simultaneously. Conflicting E-nodes establish inhibitory connections from the preferred E-nodes. For the supervised competitive learning, LSCAN selects one or more preferred E-nodes which can be continued to trigger other E-nodes while the other firing E-nodes are inhibited. The inhibited E-nodes are degraded under their own thresholds in order to not fire. Therefore, the inhibitory connections have weights regulated according to the total inhibitory strength. Let ω_{ik} be the inhibitory connection weights from e_i to e_k and $i = 1 \ldots \rho$. Then, the weights ω_{ik} are adjusted by:

$$\omega_{ik(new)} = \omega_{ik(old)} + \frac{\xi_i}{\sum_{j=1\ldots\rho} \xi_j} \frac{\varepsilon_{\mu k}}{\delta_{lf} \xi_{i(ave)}} \qquad (16)$$

where $\varepsilon_{\mu k}$ is derived from θ_k / ξ_k by the Newton-Raphson [3] method and $\xi_{i(ave)}$ is the average inhibitory strength from all E-nodes $e_{i=1\ldots\rho}$, to e_k. Under the supervised competitive learning, at one firing stage, the LSCAN systems can inhibit some E-nodes from different non-inhibitory E-nodes.

Under unsupervised competitive learning, the maximum number of E-nodes that can be fired at a firing stage is specified as β. The first β E-nodes with greater output strengths are used to inhibit the others. Equation 16 is used to adjust the weights ω_{ik}. Under the unsupervised competitive learning, all E-nodes are inhibited by the same group of the preferred E-nodes.

References

[1] Douglas S. Blank, Lisa A. Meeden and James B. Marshall, "Exploring the Symbolic/Sub-symbolic Continuum: A Case Study of RAAM", In John Dinsmore (Ed.), The Symbolic and Connectionist Paradigms: Closing the Gap, Lawrence Erlbaum Associates, Publishers, pp. 113-148, 1992.

[2] Ronald F. DeMara and Dan Moldovan, "The SNAP-1 Parallel AI Prototype", IEEE Transactions on Parallel and Distributed Systems, Vol. 4, No. 8, August 1993.

[3] Burden Faires, Numerical Analysis, 5th Ed. PWS-KENT.

[4] Stan C. Kwasny and Kanaan A. Faisal, "Symbolic Parsing Via Subsymbolic Rules", in John Dinsmore (Ed.), The Symbolic and Connectionist Paradigms: Closing the Gap, Lawrence Erlbaum Associates, Publishers, pp. 209-235, 1992.

[5] Trent E. Lange, "Hybrid Connectionist Models: Temporary Bridges Over the Gap Between the Symbolic and the Subsymbolic", in John Dinsmore (Ed.), The Symbolic and Connectionist Paradigms: Closing the Gap, Lawrence Erlbaum Associates, Publishers, pp. 237-289, 1992.

[6] Yousuf C. H. Ma, "A Survey of The Affect of Noise on Several Selected Artificial Neural Networks", Master Thesis, University of Central Florida, 1991.

[7] Steve G. Romaniuk and Lawrence O. Hall, "Injecting Symbol Processing Into a Connectionist Model", in Branko Soucek (Ed.), Neural and Intelligent Systems Integration for Fifth and Sixth Generation Integrated Reasoning Information Systems. John Wiley & Sons, Inc., 1991.

[8] David E. Rumelhart and D. Zipser, "Feature Discovery by Competitive Learning", in David E. Rumelhart and James L. McClelland, (Eds.) Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 1, The MIT Press, 1989.

[9] Lokendra Shastri, "Temporal Synchrony, Dynamic Bindings, and SHRUTI - a representational but non-classical model of reflexive reasoning", Behavioral and Brain Sciences Vol. 19, No.2, 331–337, 1996.

[10] Ron Sun, Integrating Rules and Connectionism for Robust Commonsense Reasoning, John Wiley & Sons, Inc., 1994.

[11] David S. Touretzky, "BoltzCONS: Dynamic Symbol Structures in a Connectionist Network", Artificial Intelligence, 46, pp. 5-46, 1990.

This document is an author-formatted work. The definitive version for citation appears as:

Y. Ma and R. F. DeMara, "Localized Knowledge Representation Scheme for Parallel Processing," in *Proceedings of the Fourth Joint Conference on Information Sciences (JCIS'98)*, pp. 81 – 86, Research Triangle Park, North Carolina, U.S.A., October 24 – 28, 1998.