Soft Computing in SHRUTI: — A neurally plausible model of reflexive reasoning and relational information processing

Lokendra Shastri* **and Carter Wendelken**
International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704, USA

Key Words: Structured Knowledge; Inference; Evidence Combination; Neural Networks; Connectionist Models

Abstract

Understanding language is the quintessential soft-computing problem. In order to understand language, a hearer must integrate a wide array of fuzzy, incomplete, and common sense knowledge. Yet we understand language effortlessly, spontaneously, and with remarkable efficiency. This remarkable human ability poses a challenge for computational neuroscience: How can a system of slow neuron-like elements represent a large body of knowledge and perform a wide range of inferences with such speed? SHRUTI attempts to address this challenge by demonstrating how a neural network can encode a large body of (i) specific facts, (ii) rules involving variables, negation, quantification, multiple antecedents and consequents, and (iii) knowledge about entities and types, and yet perform a wide range of inferences within a few hundred milliseconds. This paper provides a brief overview of SHRUTI with an emphasis on the encoding of soft (evidential) rules and facts, and the manner in which SHRUTI combines evidence during inference.

1 Introduction

Understanding language is the quintessential soft-computing problem[10]. In order to understand language, a hearer must integrate a wide array of fuzzy, incomplete, and common sense knowledge, and draw a number of inferences to establish referential and causal coherence. Yet we understand language effortlessly, spontaneously, and with remarkable efficiency. This remarkable human ability poses a challenge for computational theories of intelligence and computational neuroscience: How can a system of slow neuron-like elements represent a large body of knowledge and perform a wide range of inferences with such speed?

SHRUTI[7] is a neurally plausible (connectionist) model that attempts to address the above challenge and demonstrates how a network of neuron-like elements could encode a large body of structured knowledge consisting of (i) specific facts, (ii) rules involving variables, negation, quantification, multiple antecedents and consequents, and (iii) entities and types, and yet perform a wide range of predictive and explanatory inferences within a few hundred milliseconds[6][5].

SHRUTI suggests that the encoding of relational information (frames, predicates, etc.) is mediated by neural circuits composed of *focal clusters* and that the dynamic representation and communication of relational *instances* involves the transient propagation of *rhythmic* activity across these clusters. A role-entity binding is represented in this rhythmic activity by the *synchronous* firing of appropriate cells. Rules are encoded by links that enable the propagation of rhythmic activity across focal clusters, and a fact in long-term memory is a temporal pattern matching circuit.

SHRUTI identifies a number of constraints on the rapid processing of relational knowledge and predicts the capacity of the active (working) memory underlying *reflexive reasoning*[1]. First, it predicts that a large number of facts (relational instances) can be active simultaneously and a large number of rules can fire in parallel during an episode of reflexive reasoning. However, the number of distinct entities participating as role-fillers in these active facts and rules must remain very small (≈ 7). Second, systematic reasoning involving variable bindings can only be sustained over a shallow depth. As the depth of inference increases, inference reduces to a mere spreading of activation and all binding information is lost. Third, only a small number of instances of any given relation can be active simultaneously (this also limits the depth of recursion).

SHRUTI has been applied to develop models of tactical decision making under stress in collaboration with B. Thompson and M. Cohen. An implementation of SHRUTI on the

---

*Corresponding author. Email `shastri@icsi.berkeley.edu`; FAX +1 510.643.7684.

[1] Since the reasoning underlying language understanding happens spontaneously and effortlessly, almost as if it were a reflex response of the cognitive apparatus, it has been described as reflexive reasoning[6].

CM-5 [2] can encode over 500,000 rules and facts, and yet respond to a range of queries requiring derivations of depth five in under 250 milliseconds.

2 An overview of SHRUTI

The following simple example illustrates the basic representational machinery of SHRUTI. It shows how causal knowledge is encoded, and how observations can automatically lead to predictions and a search for an explanation. The example and the illustration are deliberately kept simple. In general, the model can encode positive as well as negated information; deal with inconsistent beliefs; represent rules with multiple antecedents and multiple consequents, exhibit priming effects, support context-sensitive unification of entities, and tune network weights and rule-strengths via supervised learning. Figure 1 illustrates a partial encoding of the following knowledge consisting of rules, facts, and type relationships:

1. ∀*(x:agent y:agent z:thing) give(x,y,z) ⇒ own(y,z) [800,800];*
2. ∀*(x:agent y:thing) buy(x,y) ⇒ own(x,y) [900,980];*
3. *give(John, Mary, Book-17) [1000];*
4. *buy(John, Car-23) [1000];*
5. *is-a(John, Human);*
6. *is-a(Mary, Human);*
7. *is-a(Human, Agent);* and
8. *is-a(Book-17, Book);*

Note that rule (1) also states selectional restrictions for its arguments. Thus it states that when an entity of *type agent* gives something to another entity of type *agent*, then the latter comes to own it. In Section 3 we will discuss in more detail how SHRUTI deals with evidence combination, for now it suffices to note that the pair of weights [a,b] associated with a rule has the following interpretation: a indicates the degree of evidential support for the antecedent being the probable cause (or explanation) of the consequent, and b indicates the degree of evidential support for the consequent being a probable effect of the antecedent. Weights in SHRUTI lie in the interval [0,1000].

Representation of relations using focal clusters: Each generic relation is represented by a focal cluster depicted by a dotted ellipse in Figure 1. Consider the focal cluster for the relation *give* (top left of Figure 1). This cluster includes an enabler node labeled *?:give*, two collector nodes labeled *+:give* and *-:give*, and three role nodes labeled *giver*, *recipient* and *give-object* for each of its three roles.[2] The positive and negative collectors are mutually inhibitory (inhibitory links are depicted by filled blobs). In general, the focal cluster for an n-place generic relation contains n role nodes.

Semantic import of enabler and collector nodes: Assume that the roles of a relation *P* have been dynamically bound to some fillers and thereby represent an active instance of *P* (we will see how this is done, shortly). The activation of the enabler *?:P* means that the system is seeking an explanation for the active instance of *P*. In contrast, the activation of the collector *+:P* means that the system is affirming the active instance of *P*. Similarly, the activation of the collector *-:P* means that the system is affirming the negation of the active instance of *P*. The activation *level* of *?:P* signifies the strength with which information about *P* is being sought. Similarly, the activation *level* of *+:P* (*-:P*) signifies the degree of belief in the truth (falsity) of the active instance of *P*.

Degrees of belief and contradiction: The levels of activation of *+:P* and *-:P* collectors are the result of the activation incident on them from the rest of the network and their mutual inhibition. The activation levels of the two collectors can encode a graded belief ranging continuously from *no* on the one extreme (only *-:P* is active), to *yes* on the other extreme (only *+:P* is active), and *don't know* in between (neither collector is very active). If both the collectors receive comparable and strong activation then both collectors can be in a high state of activity, in spite of the mutual inhibition between them. This signals a contradiction.

Significance of intra-cluster collector to enabler connections: The link from the collector nodes to the enabler node of a generic relation converts a dynamic assertion of a relational instance into a query about the assertion. This means that the system continually seeks support (or an explanation) for active assertions. For a given relation *P*, the weight on the link from *+:P* (*-:P*) to ?:P is proportional to the system's overall propensity for seeking explanations and inversely proportional to the probability of occurrence of a positive (negative) instance of P.

Encoding of types and instances: This is illustrated at the bottom right of Figure 1. The focal cluster of each entity, *A* consists of a *?:A* and a *+:A* node. In contrast, the focal cluster of each type, *T* consists of a pair of *?* (*?e:T* and *?v:T*) and a pair of + nodes (*+e:T* and *+v:T*). While the nodes *+v:T* and *?v:T* participate in expression of knowledge (facts and attributes) involving the whole type *T*, the nodes *+e:T* and *?e:T* participate in the encoding of knowledge involving particular instances of type *T*. Thus the pair of *v* nodes and the pair of *e* nodes signify universal and existential quan-

---

[2] Each *label* corresponds to a node. Nodes are computational abstractions and correspond to *small ensembles of cells*. Similarly, a connection from a node A to a node B corresponds to several connections from cells in the A ensemble to cells in the B ensemble.

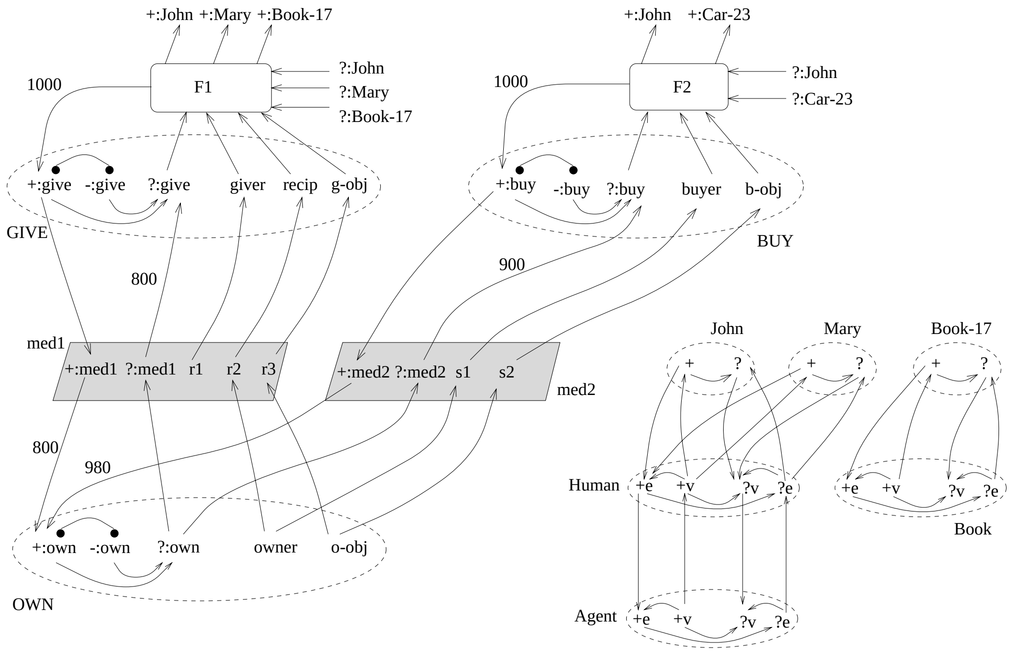

Figure 1: An example SHRUTI network. Links between mediator and type structures have been omitted.

tification, respectively. The interconnection among nodes within and across focal clusters of instances and types lead to a wide range of type inferences. The *levels* of activation of *?:A, ?v:T, and ?e:T* nodes signify the strength with which information about entity *A*, type *T*, and an instance of type *T*, respectively, is being sought. Similarly, the *levels* of activation of *+:A*, *+v:T*, and *+e:T* signify the degree of belief that the entity *A*, the type *T*, and an instance of type *T*, respectively, play appropriate roles in the current situation.

Dynamic bindings: The *dynamic* encoding of a relational instance corresponds to a *rhythmic* pattern of activity wherein bindings between roles and entities are represented by the *synchronous* firing of appropriate role and entity nodes. With reference to Figure 1, the dynamic representation of the relational instance *(give: ⟨giver=John⟩, ⟨recipient=Mary⟩, ⟨give-object=a Book⟩)* (i.e., “John gave Mary a book”). will involve the synchronous firing of *+:John* and *giver*, the synchronous firing of *+:Mary* and *recip*, and the synchronous firing of *+e:Book* and *g-obj*. The entities *+:John*, *+:Mary* and *+e:Book* will fire in distinct phases.

Mutual exclusion and collapsing of phases: Instances in the type hierarchy can be part of a *phase-level* mutual exclusion clusters (ρ-mex clusters). Members of a ρ-mex cluster inhibit one another, and hence, only the member with the highest activation fires in a given phase. A similar ρ-mex cluster can be formed by mutually exclusive types. If the + node of an entity *I* fires in multiple phases i and j, and if the + node of no other entity is firing in phase j, then phase j gets collapsed into phase i.

E-facts and T-facts: SHRUTI encodes two types of relational instances (i.e., facts) in its long-term memory: episodic facts (E-Facts) and taxon facts (T-facts). These facts provide closure between the enabler node and the collector nodes. While an E-fact corresponds to a specific instance of a generic relation, a T-fact corresponds to a distillation or statistical summary of various instances of a generic relation and can be viewed as coding *prior probabilities*. In general, T-facts can be conditioned on the type of role-fillers (e.g., the T-fact *buy(a-person,a-Car)* which encodes how likely it is that a person would buy a car, will presumably have a higher weight than the T-fact *buy(a-person,an-island)*). While E-facts are sensitive to matches and mismatches, T-facts are sensitive only to matches, and respond like attractors in an associative memory.

Encoding of rules: A rule is encoded via a mediator focal cluster (shaded region) that mediates the flow of activity between the antecedent and the consequent clusters. The mediator consists of a collector and an enabler node and as many role-instantiation nodes as there are distinct variables in the rule. The encoding of a rule establishes links between nodes in the antecedent, consequent, and mediator clusters as follows: (i) The roles of the consequent relations are linked to the roles of the antecedent relations via appropriate role-instantiation nodes in the mediator. This linking reflects the correspondence between antecedent and consequent roles specified by the rule. (ii) The enablers of the consequent relations are connected to the enablers of the antecedent relations via the enabler of the mediator. (iii) The appropriate (+/–) collectors of the antecedent relations are linked to the appropriate (+/–) collectors of the consequent relations via the collector of the mediator. A collector to collector link originates at the + (–) collector of an antecedent relation if the relation appears in its positive (negated) form in the antecedent. The link terminates at the + (–) collector of a consequent relation if the relation appears in a positive (negated) form in the consequent. If a role-instantiation node receives activation from the mediator enabler and one or more consequent role nodes, it simply propagates the activity onward to the connected antecedent role nodes. If on the other hand, the role-instantiation node receives activity only from the mediator enabler, it sends activity to the *?:e* node of the type specified in the rule as the type restriction for this role. This causes the *?:e* node of this type to become active in an unoccupied phase. The *?:e* node of the type conveys activity in this phase to the role-instantiation node which in turn propagates this activity to connected antecedent role nodes. This interaction between the mediator and the type representation, in effect, creates activity corresponding to "Does there exist some role filler of the specified type?"

An example of inference: Figure 2 depicts a schematized response of the SHRUTI network shown in Figure 1 to the query "Does Mary own a book?" (*exists x:Book own(Mary, x)?*). This query is posed by activating *?:Mary* and *?e:book* nodes, the role nodes *owner* and *o-obj*, and the enabler *?:own*, as shown in Figure 2. We will refer to the phases of activation of *?:Mary* and *?e:book* as ρ_1 and ρ_2, respectively. Activation from the focal cluster for *own* reaches the mediator structure of rules (1) and (2). Consequently, nodes *r2* and *r3* in the mediator for rule (1) become active in phases ρ_1 and ρ_2, respectively. Similarly, nodes *s1* and *s2* in the mediator of rule (2) become active in phases ρ_1 and ρ_2, respectively. At the same time, the activation from *?:own* activates the enablers *?:med1* and *?:med2* in the mediators of rules (1) and (2). Since *r1* does not receive any activation from any role in its consequent's focal cluster (i.e., from *own*), it activates the node *?e:agent* which becomes active in a free phase (say $\rho 3$) and, in turn, activates *r1* in this phase. The activation from nodes *r1*, *r2* and *r3* reach the roles *giver*, *recip* and *g-obj* in the *give* focal cluster, respectively. Similarly, activation from nodes *s1* and *s2* reach the roles *buyer* and *b-obj* in the *buy* focal cluster, respectively. In essence, the system has created new bindings for the *give* and *buy* relations. These bindings together with the activation of the enabler nodes *?:give* and *?:own* encode two new queries: "Did *some agent* give Mary a book?", and "Did Mary buy a book?". At the same time, activation travels in the type hierarchy and maps the query to a large number of queries such as "Did a human give Mary a book?", "Did John give Mary Book-17?", "Did some institution give all humans Book-10?", "Did Mary buy all books" etc. The fact *give(John, Mary, Book-17)* now becomes active as a result of matching the query *give(John, Mary, Book-17)?* and causes *+:give* to become active. This in turn causes *c:med1*, to become active and transmit activity to *+:own*. This results in an affirmative answer to the query "Does Mary own a book?" and creates a reverberant loop of activity involving the clusters *own*, *med1*, *give*, the fact F1, and the entities *John*, *Mary*, and *Book-17*.

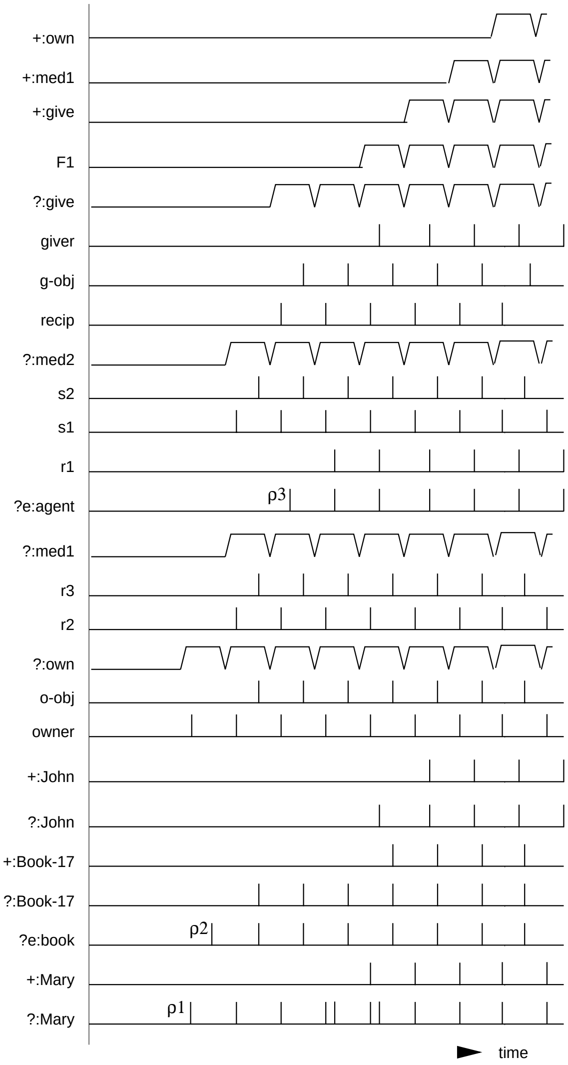

Figure 2: A schematized activation trace of selected nodes for the query *own(Mary,Book-17)?*.

Parallel inference: The encoding of rules by the explicit encoding of the inferential dependency between predicates and predicate roles, in conjunction with the use of temporal synchrony, provides an efficient mechanism for propagating dynamic bindings and performing reasoning. Conceptually, the proposed encoding of rules creates a directed *inferential dependency graph* and the evolution of the system's state of activity corresponds to a *parallel* breadth-first traversal of this inferential dependency graph. Consequently, the time taken to perform an inference is simply proportional to the depth of its derivation and is otherwise independent of the number of items in the knowledge base.

3 Evidence combination

There are many places in SHRUTI where activity converging on a node from different sources must be combined to determine an output value for the node. Such evidence combination occurs at the collector node of a rule mediator where activity arrives from *multiple* antecedents of the rule, and at the collector node of a relation where activity arrives from collectors of distinct rule mediators. Similarly, evidence combination occurs at the enabler node of a rule mediator where activity arrives from *multiple* consequents of the rule, and at the enabler node of a relation where activity arrives from the enabler nodes of various rule mediators. Moreover, evidence combination also occurs when evidence from different facts impinges on collector nodes, and during the propagation of activity through the type hierarchy.

An analysis of the evidence combination requirements suggests that a flexible set of evidence combination functions is required for the proper integration of evidence. In view of this, SHRUTI employs a broad range of evidence combination functions (ECFs). A knowledge engineer wishing to model a domain needs only to select the appropriate ECFs and the approximate values of link weights. The system subsequently fine-tunes these link weights through supervised learning. In selecting a range of ECFs, the goal was to have a set large enough to adequately model real-world data, but small enough to enable a simple choice of appropriate ECFs for a given situation. It was also desired that these ECFs be computationally simple. We discuss these ECFs in brief below. A more detailed discussion appears in [8].

An obvious source of inspiration in developing a family of ECFs was fuzzy logic where numerous functions have been developed to combine fuzzy membership values [10]. In particular, binary operators known as T-norms and S-norms were highly relevant. These represent, respectively, general forms of fuzzy set intersection and union.

In neural networks, representational adequacy is achieved by adding additional nodes and links, while keeping the ECF very simple. Generally, a single ECF is used throughout the network, the most common being the *sigmoid-sum*. While other ECFs can be used in place of the sigmoid-sum, it does not make sense to pick and choose different ECFs for different nodes in a typical neural network, since the nodes do not have any *prior* meaning. In contrast, the nodes in a structured connectionist system like SHRUTI have meaningful interpretations, and hence, it is possible to choose appropriate ECFs for different nodes.

Belief nets combine evidence using conditional probability tables (CPT) associated with each node. The use of a full CPT allows considerable flexibility but at an exorbitant storage and computational cost. The noisy-OR function [4] reduces these demands, but is overly restrictive if used universally. Other means of reducing complexity, such as encoding the CPT with a tree structure, offer a different approach to evidence combination than that envisioned here [3]. It would be interesting to compare the representational flexibility afforded by a range of ECFs with that allowed by structured CPTs.

The ECFs developed for SHRUTI form a continuum, with *and* at one end and *or* at the other. In between these extremes are four basic categories of functions: *soft-and* (with values up to min), *soft-min* (ranging from min to average), *soft-max* (ranging from average to max), and *soft-or* (with values beyond max). Although specific functions have been chosen to represent each category, many of the functions developed for fuzzy logic could also be used here.

We believe that most meaningful evidence combination situations can be characterized as belonging to one of the four basic categories listed above on the basis of the *necessity*, *sufficiency*, and the *degree of correlation* of the inputs. In general, correlated antecedents are expressed as the antecedents of a single multiple-antecedent rule. Similarly, correlated consequents are expressed as the consequents of a single multiple-consequent rule. In contrast, uncorrelated factors reside in separate rules. The family of ECFs used in SHRUTI allows the expression of rules involving both correlated as well as independent factors.

Link weights can play an important and context-dependent role in many ECFs. The standard use of link weights is to multiply them with the values of input nodes prior to doing evidence combination. All the proposed ECFs support this "standard" use of weights. However, in addition to simply affecting values before they are combined, weights can be used as additional function parameters, with different interpretations for different functions. The use of link weights in this manner, elaborated below, provides a high degree of flexibility in the kinds of relations that can be represented.

At one end of the spectrum of ECFs are the *soft-and* functions. *Soft-and* functions corresponding to *T*-norms of fuzzy logic are appropriate for combining causes which are deemed necessary. The primary representative of this category is the weighted *and*, calculated as $\prod_i (1-(1-X_i)W_i)$. The use of weights as parameters imparts some interesting properties to the *and* function. Since the main characteristic of the *and* function is that the inputs are regarded as necessary, an obvious interpretation of the weights is that they reflect the degree of necessity. In probabilistic terms, this would be the probability that the consequent is false given that the antecedent is false. This means that lower weights should result in higher output values. While this may seem counterintuitive, the assignment of degrees of necessity is appropriate from a knowledge engineering standpoint, and makes the *and* function remarkably flexible.

A graph of a weighted *and* function with two antecedents

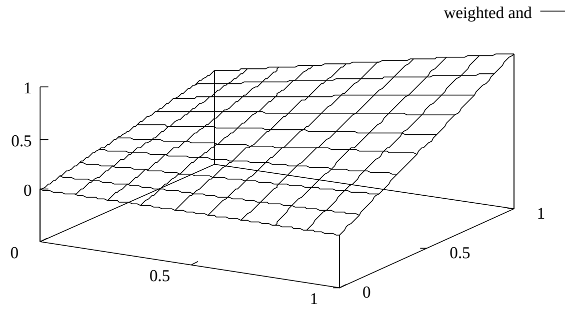

Figure 3: Graph of a weighted *and* function with antecedent weights of 1000 and 600.

of weights 1000 and 600, respectively, is shown in Figure 3. The relative importance of the weights is reflected in the gradual decrease in the function value when traveling from 1 to 0 along the near axis (which corresponds to the 600 weight), versus a rapid decrease when traveling from 1 to 0 along the far axis (corresponding to the 1000 weight).

Use of weights in the above manner represents a softening of the necessity requirement of the unweighted *and* function (i.e., the product of weighted values). Another sort of softening involves a relaxation of the independence assumption (e.g., assuming a positive correlation among inputs). This has been achieved with a *soft-and* function similar to the Hamacher product *T*-norm $H(x,y) = xy/(w + (1-w)(x+y-xy))$ (where $w \in [0,1]$) generalized to n variables [1].

At the other end of the spectrum are the *soft-or* functions, with the primary representative being the *weighted or*, given as $(1 - \prod_i (1 - X_i * W_i))$. This is perhaps the simplest of the functions in that its weighted and unweighted versions are the same (differing only in the time at which the weights are multiplied in). *Or*-like functions, which can be thought of as those having output values at least equal to the maximum input, are used when any individual antecedent is sufficient to affect the consequent. These correspond roughly to the S-norms of fuzzy logic, and many of these fuzzy operators might be adapted to the task. *Or* is the most commonly used function for combining activity from different rules that converge on a particular concept. The natural interpretation of weights for *or*-like functions is that they represent the degree of sufficiency of the source concept, i.e., the probability of the consequent given the particular antecedent.

As with the *and* function, a further softening of the *or* function to reflect positive correlations among inputs is possible and can be achieved with a *soft-or* function defined in terms of the *soft-and* function referred to above.

Covering the range between *min* and *max* are the weighted averages. Weighted averages are appropriate when individual antecedents are neither necessary nor sufficient. For all of these functions the link weights represent degrees of influence, giving the relative effect of an antecedent value on the output. There are three main functions in this class: the simple weighted *average* $\sum_i (X_i W_i) / \sum_i (W_i)$, the *soft-min* function $((\sum_i X_i^k W_i)/(\sum_i W_i))^{1/k}$ for $k \in (0,1)$, and the *soft-max* function with $k \in (1, \infty)$. It should be noted that *min* and *max* are the limits of the given *soft-min* and *soft-max* functions, as $k \rightarrow 0$ and $k \rightarrow \infty$, respectively, and also that the weighted *average* is the same form as the others with $k = 1$. So this whole range from *min* to *max* is really only one functional form with a varying parameter. *Soft-min* is used when it is necessary that most of the evidence for the antecedents be available in order to conclude the consequent, but unlike *and*, no single piece is required. Combining evidence about the symptoms associated with a particular syndrome is a situation where *soft-min* can be appropriate. In this case, a lack of evidence for one of the particular symptoms should weigh heavily against a positive conclusion. But it should still be possible to conclude that a syndrome is present even if evidence for one of its symptoms is absent. Consequently, any *and*-like function would not be appropriate, and *soft-min* is the function of choice. With *soft-max*, only a fraction of the potential evidence is sufficient to lead to strong activity in the consequent, but unlike *or*, no single piece is sufficient on its own.

The simple weighted *average* is probably the most basic of all the ECFs and is appropriate when antecedents are not known to be either causes or prerequisites; all that is known is that greater activity among the antecedents should lead to greater activity in the consequents. An example of a rule that uses the weighted average function is:

∀ x:Person [average(chilled(x) 400, lostSleep(x) 600, exposed(x) 1000) ⇒ commonCold(x)]

This rule says that being chilled, losing sleep, and being exposed to someone with a cold are risk factors for developing a common cold. Moreover, these risk factors have different degrees of influence, with exposure being the most important. No risk factor is alone necessary or sufficient.

4 Learning

SHRUTI performs supervised learning by means of a gradient descent algorithm based on backpropagation[9]. A number of features of the system require that the learning algorithm be somewhat different from standard neural network backpropagation. Because of its structure, the network has numerous cycles and only a subset of link weights are adjustable. These are primarily those links that feed into the various ECFs. As all of the ECFs are differentiable in terms of both the source values and link weights, error assignment and adjustment of these weights is straightforward.

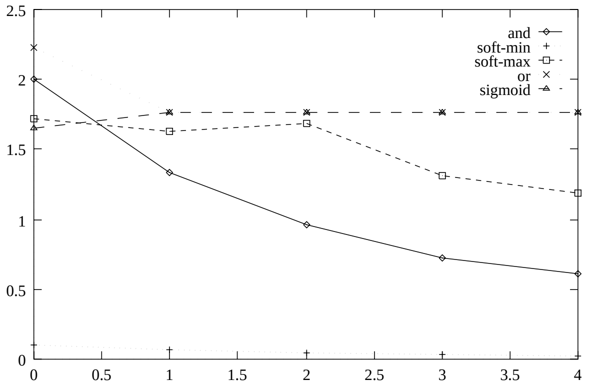

Figure 4: Learning curves for each ECF trained on a synthetic data-set. Scenarios were generated by assigning random values to antecedents and choosing a target value of 1000 or 0 for the consequent using probabilities determined by the *soft-min* combination of the antecedents.

As a demonstration of learning and the importance of having a choice of ECFs, a simple dataset was created using the *commonCold* rule from above. Scenarios were created by assigning varying values to each of the three antecedents. The target value of the *commonCold* consequent for each scenario was then determined based on the *soft-min* value with weights of 400, 600, and 1000 as given above. The rule was encoded with each of the ECFs and initial weights uniformly set to 800. For each function, five batches of training data were presented. The sum-of-squared-errors for *commonCold* over all of the scenarios in a batch was recorded for each batch to produce a learning curve. Figure 4 shows the learning curves for *soft-min*, *and*, *soft-max*, *or*, and *sigmoid-sum*. Note that the dataset in question is best represented by *soft-min*. The *and* function, which is most similar to *soft-min*, shows the second best results. With *or* and *sigmoid-sum*, the error remains high, indicating that these functions are inadequate for representing the data. This demonstrates a definite weakness in systems which rely on only a single function like *sigmoid* or *noisy-OR*.

5 Conclusion

We have provided a brief overview of SHRUTI and described how it encodes evidential knowledge and how it combines evidential support during inference. The representational machinery of SHRUTI includes weighted links, a repertoire of ECFs, and the ability to encode complex rules involving selectional restrictions, multiple antecedents, multiple consequents, and exceptions. We believe that these mechanisms allow the encoding and processing of a rich variety of knowledge involving fuzzy as well as probabilistic aspects.

The initial focus of work on SHRUTI had been the development of neurally plausible representational mechanisms for encoding and processing structured knowledge. More recently, our focus has shifted to issues of evidence combination and soft-computing. The work summarized in this paper points to where we are headed and should help explicate existing as well as potential linkages between SHRUTI and the fields of fuzzy logic and probabilistic inference.

Acknowledgment

This work was partially funded by NSF grant SBR-9720398 and ONR grant N00014-93-1-1149, and subcontracts from Cognitive Technologies Inc. related to ONR contract N00014-95-C-0182 and ARI contract DASW01-97-C-0038.

References

[1] D. Driankov, H. Helendoorn, & M. Reinfrank *An Introduction to Fuzzy Control*. Springer-Verlag: Berlin. 1996

[2] D.R. Mani. “The Design and Implementation of Massively Parallel Knowledge Representation and Reasoning Systems: A Connectionist Approach”, Ph.D. Dissertation, Department of Computer and Information Science, University of Pennsylvania, 1995.

[3] N. Friedman & M. Goldszmidt. Learning Bayesian Networks with Local Structure. In *Proc. of the Twelfth Conference on Uncertainty in Artificial Intelligence*, 252–262, Morgan Kaufmann, San Francisco, 1996

[4] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[5] L. Shastri. Advances in SHRUTI — A neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony *Applied Intelligence* (in press).

[6] L. Shastri & V. Ajjanagadde. From simple associations to systematic reasoning. *Behavioral and Brain Sciences* 16 (3) 417–494, 1993.

[7] “www.icsi.berkeley.edu/∼shastri/shruti/index.html”

[8] C. Wendelken & L. Shastri. Evidence combination in a Structured Connectionist Network, Submitted.

[9] P. Werbos. *The roots of backpropagation — from ordered derivatives to neural networks and political forecasting*, Wiley and Sons, New York. 1994.

[10] L. Zadeh. *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A Zadeh*, G. Klir & B. Yuan (eds), World Scientific: Singapore, 1996.