Neural Networks for Structured Knowledge

International Joint Conference on Artificial Intelligence
IJCAI '97

Franz J. Kurfess

Computer and Information Sciences Department
New Jersey Institute of Technology, Newark, U.S.A

and

Alessandro Sperduti

Computer Science Department,
University of Pisa, Italy

Course Overview

Introduction

Knowledge Representation

Knowledge Processing

Neural Networks and Structured Knowledge

Generalized Recursive Neuron

Computational and Complexity Issues

Conclusions

Terminology

Knowledge, Data and Information

Knowledge:

definition:[1] Information and understanding about a subject which a person has in his or her mind or which is shared by all human beings

similar terms:[2] *learning, lore, scholarship, wisdom, instruction, book-learning, enlightenment, expertise, intelligence, light, theory, science, principles, philosophy, awareness, insight, education, substance, store of learning, know-how*

important aspects: possibly complex structure (relations between items)

[1] [Sinclair, 1987]
[2] [Laird, 1982]

Data:

definition:[1] Information, usually in the form of facts or statistics that you can analyse, or that you use to do further calculations

similar terms:[2] *evidence, reports, details, results, notes, documents, abstracts, testimony, matters of direct observation, facts, raw materials, memorandums, statistics, figures, measurements, conclusions, information, circumstances, experiments*

important aspects: rigid, simple structure (tables)

[1] [Sinclair, 1987]
[2] [Laird, 1982]

Information:

definition:[1]

1. knowledge acquired through experience or study
2. knowledge of specific and timely events or situations; news
3. the act of informing or the condition of being informed
4. an office, agency, etc. providing information
5. a charge or complaint made before justice of the peace
6. the results derived from the processing of data according to programmed instructions
7. another word for data

[1] [Hanks, 1979]

similar terms:[1] *derived knowledge, acquired facts, evidence, knowledge, reports, details, results, notes, documents, testimony, facts, figures, statistics, measurements, conclusions, deductions, plans, field or laboratory notes, learning, erudition; news, report, notice, message*

important aspects:

rather vague usage in common language, precise definition in information theory

[1] [Laird, 1982]

Knowledge Hierarchy

Structured Knowledge

information items and their relationships

information items

objects, concepts, features, attributes

relationships

hierarchical, membership, component, similarity, location, ...

Knowledge Representation

formalisms to describe information items and their relationships

adequate
: are essential aspects captured?

comprehensible
: is the represented knowledge understandable?

transferable
: can the knowledge be communicated?

uniform
: is identical information consolidated?

composite
: can components be grouped into ensembles?

reliable
- belief / truth
- consistency

verifiable
- objective / subjective
- facts
- derived knowledge
- basic assumptions

efficient
- usage of space
- execution time for basic operations

Requirements

on a knowledge model

unambituity

no confusion between objects
no mixing of properties (cross talk)
orthogonality

content-oriented access

addressing of objects via features
similarity

arbitrary complexity

depth
dimensionality

context

application of operations according to the current context

structure sensitivity

processing is guided by the internal structure of an object

graceful degradation

no complete breakdown under stress
e.g. memory limitation, execution errors

parallel processing

simple elementary operations

statistical relevance

acquisition and preparation of knowledge according to statistical properties of the real environment

statistical inference

conclusions based on statistical relevance

Basic Functionalities

important constructs for dealing with knowledge

new concepts

extension and organization of knowledge
natural / real or formal / nominal causes

identification of individuals

assignment of properties to individuals (concrete, uniquely identifiable objects)

classification

arraignment of individual into classes
comparison and abstraction of features

definition

specification of necessary and sufficient constraints for an individual to belong to a certain class

relations

relationships between objects must be representable and interpretable

systems

structured grouping of individuals
e.g. aggregation of objects with a certain relation

roles

extraction of an element from a relation

events

systems with temporal aspects as an important property (state)

actions

subject-oriented sequence of events

Knowledge Representation

with Neural Networks

representation methods

local / distributed

symbol-oriented / subsymbolic

knowledge types

Which types of NNs are best suited for which types of knowledge?

Representation Methods

important principles independent of NN types

local

each unit represents one particular concept or object

distributed

the representation of a single concept or object is carried by several units

symbol-oriented

concepts and their properties are only accessible via symbols

subsymbolic

access via features of a concepts

Symbol

sign or token used to represent an object or concept

- unambiguous identification
- loclized
- arbitrary shape
- needs explanation or mapping

Symbol & NN

arbitrary mapping from symbol to object

associative memory
learning of the relation between object and symbol

Icon

also: *Simile*, simplified picture

- inherent similarity to the object to be represented
- not necessarily unambiguous
- localized
- obvious mapping

Icon & NN

retains essential properties of the object to be represented

associative memory
learning of the relation between object and representation

only feasible if the icon is already defined

Index

indicator

- elicits an important feature of the object
- usually unambiguous
- localized
- inherent mapping

Index & NN

direct representation of one essential aspect of the object

simple network
computation of a function from input to output

Object & Properties

mapping of an object and its properties

hetero-associative memory

learning of the relation between an object and its properties

Structured Objects

objects are composed from components (compositionality)

recursive auto-associative memory

a reduced representation of the input layer in the hidden layer
this reduced representation is recursively used as part of the input layer

Symbol Grounding

establishment of a mapping between a symbol and the object to be represented

- emergence of the mapping
- causal correlation?
- basis for a good model

Binding-Problem

simultaneous unambiguous mapping for several objects in one network

tensor-product
: matrix with entries where object and properties meet
problem: quadratic space requirements

synchronicity
: associated elements are identified by simultaneous activity
problem: limited by available phases

signatures
: different objects use different signals problem: signatures are essentially symbols

Symbols and Formal Systems

semantics for a formal system based on concrete experiences with a real environment, not only in reference to other symbol structures

- meaning often assigned by the programmer / designer
- "parasitic" mapping (not intrinsic, but interpreted)
- possibility of errors due to interpretation of symbols ("concept" vs. string)
- often not plausible as a model for cognitive phenomena

Extreme Positions

about knowledge and its representation

formalistic assumption
: knowledge can be represented by finite structures composed of discrete atomic symbols in accordance with a finite number of syntactic relations [1]

relativistic assumption
: knowledge can only be described in a meaningful way with respect to a framework incorporating non-quantitative aspects like experience, belief, expectation, feelings, ...
as a consequence, knowledge cannot be described in absolute terms [2]

---

[1] [MacLennan, 1994]
[2] Joseph Weizenbaum

Physical Symbol Systems Hypothesis

A *Physical Symbol System* consists of symbols and structures that have to be realized physically; it has the necessary and sufficient conditions for an intelligent system [1]

[1] [Newell and Simon, 1976]

Knowledge Processing Methods

- knowledge recall
- identification of structures or patterns
- content-based access
- similarity-based access
- search
- generalization
- (logical) inference
- probabilistic, evidential, fuzzy, analogical, ... reasoning
- metaphors
- planning
- ...

Neural Network Models

used for knowledge processing

- simulated annealing
- distributed production rules
- syllogistic reasoning
- reflexive reasoning
- spreading activation
- evidential reasoning

Logic and Simulated Annealing

satisfiability of a logical formula is checked via energy minimization

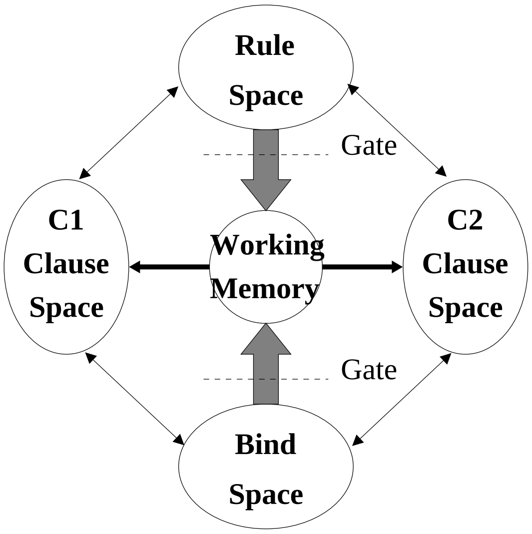
- transformationof the formula into an equivalent network
  äquivalentes Netzwerk
- energy minimization via simulated annealing

Distributed Production Rules

Distributed Connectionist Production System

- many simple, neuron-like units
- rule interpreter for production rules
- distributed representation

DCPS Architecture



Syllogistic Reasoning

modeling of higher cognitive functions through mental models

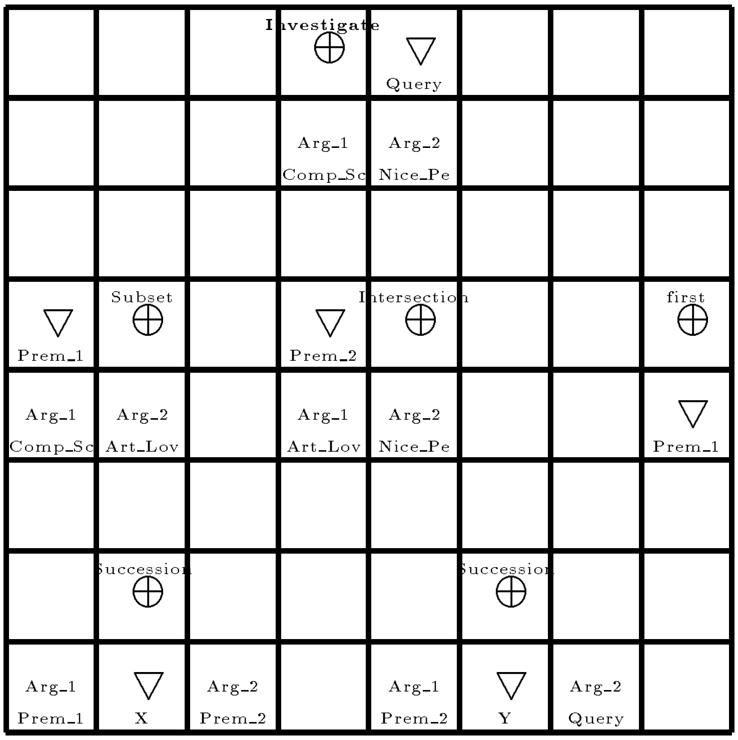
- abstract example situations
- generation and evaluation of hypotheses
- short-term memory for several co-existing objects and hypotheses
- similarity-based access
- similar concepts are represented in spatial proximity
- similarly structured concepts are processed in a similar way
- encoding via relative positions in the network
- association between common activity patterns in relevant substructures

Conposit: *An Example*

`(Query:)`	`Computer scientists, nice people?`
`(Premise 1:)`	`Some computer scientists are art lovers.`
`(Premise 2:)`	`All art lovers are nice people.`
`(Conclusion:)`	`Some computer scientists are nice.`

Notation:

bold typeface	action / relation	e.g. **Investigate**
italics typeface	type / role	e.g. Arg_1
SMALLCAPS TYPEFACE	name / object	e.g. ART_LOV
$\bigoplus$	class	
$\bigtriangledown$	instance	



Advantages and Problems

of CONPOSIT

+ cognitively plausible reasoning patterns
+ similarity-based processing techniques
- essentially sequential method

Reflexive Reasoning

modeling of reasoning techniques through structured networks

- many simple, neuron-like units
- fast
- large number of rules
- limited expressiveness

Spreading Activations

reasoning process is modeled through spreading activation in a network

- based on semantic networks
- local representation
- variable bindings through signatures
- disambiguation
- access to similar events

CONSYDERR

CONnectionist SYstem with Dual representation for Evidential Robust Reasoning

- two-level architecture:
  local (concepts)
  distributed (features)
- based on vague, evidential reasoning (fuzzy evidential logic)
- able to deal with
  partial information
  similarity-based unification
  inheritance of properties
  Inconsistencies

CONSYDERR Architecture

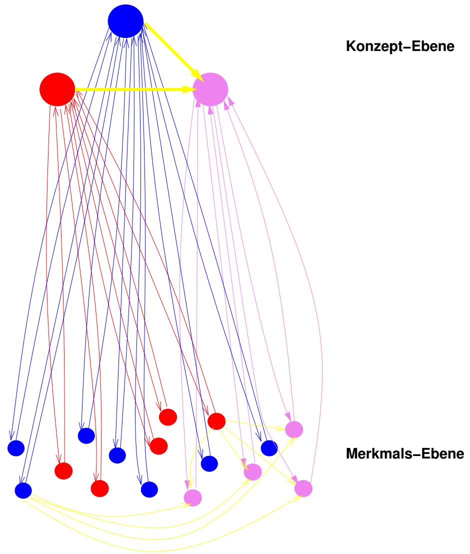
local layer

reasoning over *concepts*
rules are represented via directed edges from premise nodes to conclusion nodes
specific mechanism for variable bindings (transmission and representation of signatures)

distributed layer

reasoning over *features*
similarity appears as overlap in features

CONSYDERR Layers



Evaluation Model

- links between nodes of the local and the distributed layer
- concept nodes in the local layer have connections to those nodes in the distributed layer that represent the same concept
- rule connections from the local layer are duplicated diffusely in the distributed layer
- interaction between the two layers according to a fixed cycle:

  Phase I links from local to distributed are active (top down)

  Phase II links within both layers are active

  Phase III links from distributed to local are active (bottom up)

Advantages and Problems

of CONSYDERR

+ different types of reasoning
+ connectionist evaluation model
- complex, biologically implausible neuron model
- rigid interaction scheme
- limited similarity notion

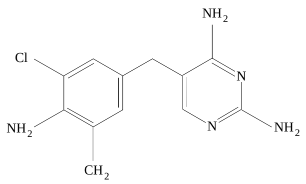
PART II

Neural Networks for Structured Domains

MOTIVATION

Extension of neural networks to structured domains

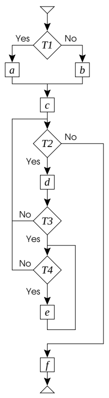
Example 1: Chemistry



- Quantitative Structure-Activity Relationship (QSAR)
- Quantitative Structure-Property Relationship (QSPR)
- DNA Analysis

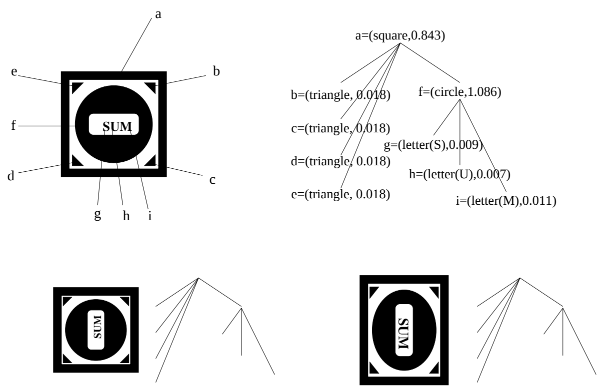
Example 2: Software Engineering

```
program  name (list);
var
 . . .
begin
if  T1 then
        a
else
        b;
c;
while T2 do
        begin
        d;
        if  T3 then
        while T4 do
                e
        end;
f
end.
```



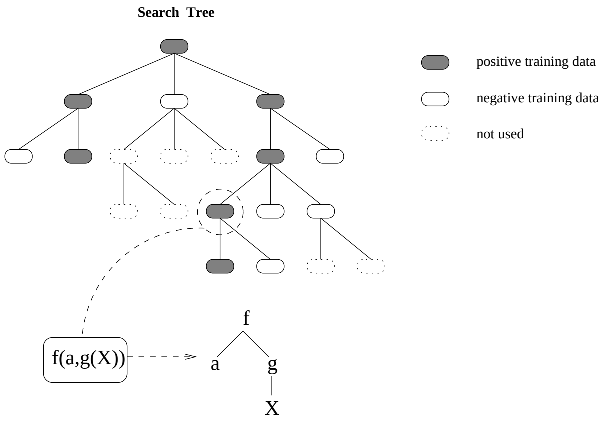
- Software Quality: metric for quality, reusability, maintainability, etc.

Example 3: Pattern Recognition



- Recognition of structured patterns

Example 4: Problem Solving



- Learning Heuristics

WHY NEURAL NETWORKS ?

Because

- Universal Approximators
- Automatic Inference (Learning)
- Nice Classification Capability
- Robust to Noise
- Can Deal with Real Valued Information

PROBLEMS

- PROBLEM 1
  how can we represent a structure **in** a neural network ?
- PROBLEM 2
  how can we classify a structure **by** a neural network ?
- PROBLEM 3
  how can we store **in** and recall a structure **by** a neural network ?

Representation of Structures in NN

- **PROBLEM 1**: Representing Structures in NN
- Reduced Descriptions
- RAAM Family Models

Representing Structures in NN

Structures can be represented in NN by using:

- **Local Representation**
  - spatial encoding
  - time encoding
- **Distributed Representation**

Local Representation

In a local representation

- each neuron represents a component of the structure
- each connection represents a relation between components
- trivial and not efficient

Examples

- MPP (Waltz and Pollack, 1985), PPN (Selman and Hirst, 1985)
- Parsing Network (Charniak and Santos, 1987)
- ACME (Holyoak and Thagard, 1989)
- μKLONE (Derthick, 1990)
- SHRUTI (Ajjanagadde and Shastri, 1991)

Distributed Representation

- each component of the structure is represented by a set of units
- each unit contribute to the representation of a set of components

Examples

- DCPS (Touretzky and Hinton, 1988)
- Tensor Products (Smolensky, 1990)
- RAAM (Pollack, 1990), LRAAM (Sperduti, 1994)
- Convolution-based models:
  TODAM (Murdock, 1983), CHARM (Metcalfe, 1991), TODAM2 (Murdock, 1993), **Holographic Reduced Representations** (Plate, 1991)

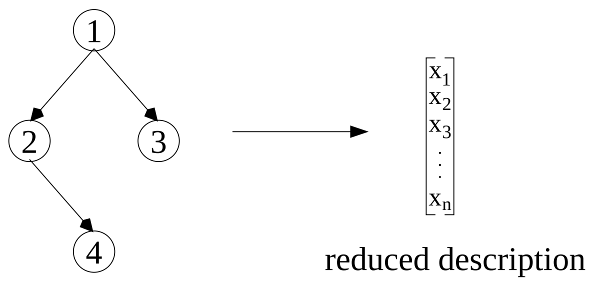
Distributed Representations (cont.)

Reduced Descriptions (Hinton, 1987; 1990)

- Encoding of complex conceptual structure in distributed representations
- Four desiderata
  1. **Representational adequacy:** reconstruct the full representation from the reduced description
  2. **Reduction:** fewer units
  3. **Systematicy:** related in a systematic way to the full representation
  4. **Informativeness:** tell us something about the concept that it refers to, without it being necessary to reconstruct the full representation

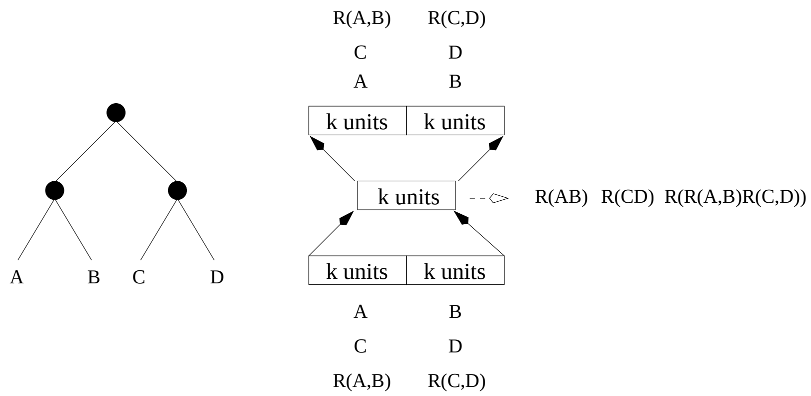
Distributed Representations (cont.)

Both **RAAM** and **Holographic Reduced Representations** implement reduced descriptions

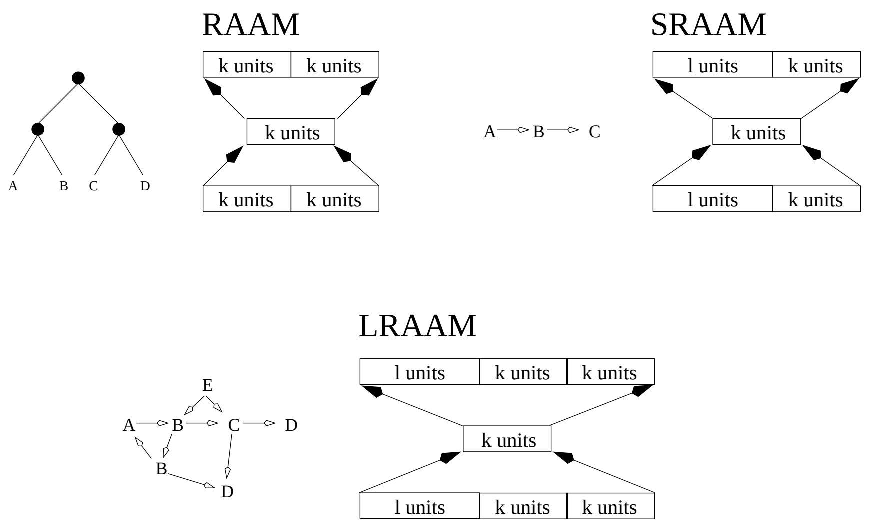


Recursive Auto-Associative Memory

- Tree with labeled leaves → Numeric vector
- Small sets of fixed-width trees encoded in patterns all of the same size



RAAM Family

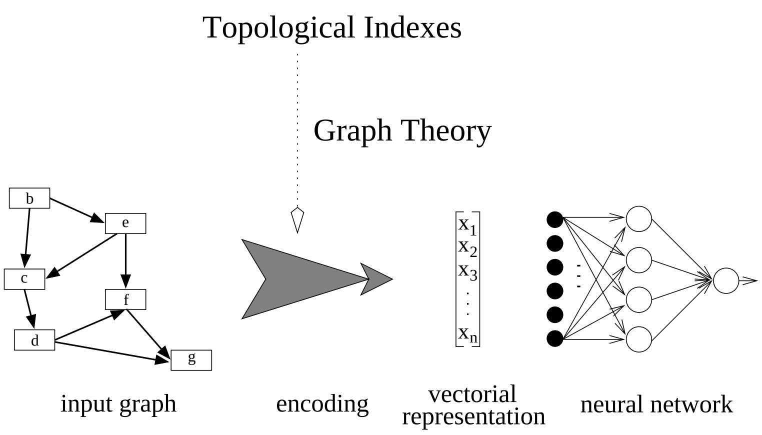


Classification of Structures by NN

Outline

- **PROBLEM 2**: Classification
- Generalized Recursive Neuron
- Models of Recursive Neural Networks
- Computing the Gradient

Classification with Standard NN



PROBLEM:

a priori definition of the encoding process

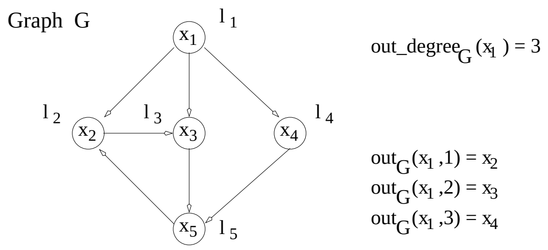
- the relevance of different features of the graphs may change dramatically for different classification tasks

- the encoding process has to work for any classification task ($\rightarrow$ representations difficult to classify)

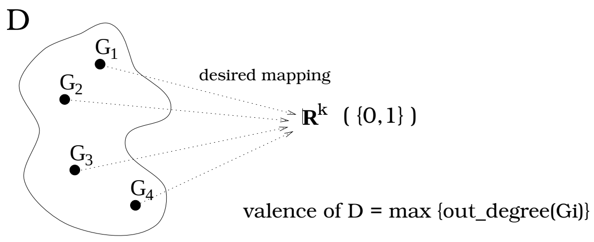
SOLUTION:

encoding performed by a NN

PRELIMINARIES

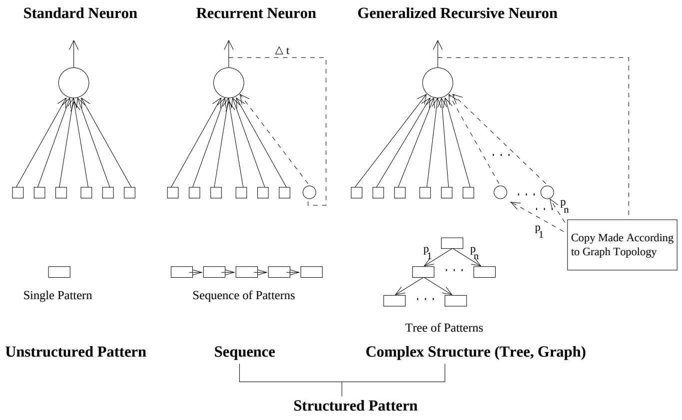


STRUCTURED DOMAIN



Generalized Recursive Neuron

- standard and recurrent neurons are not suited to dealing with labeled structures
- to solve this inadequacy we introduce the *generalized recursive neuron*



Generalized Recursive Neuron (def.)

The output $o^{(g)}(x)$ of a generalized recursive neuron

$$o^{(g)}(x) = f(\sum_{i=1}^{N_L} w_i l_i + \sum_{j=1}^{out\_degree_X(x)} \hat{w}_j \, o^{(g)}(out_X(x,j)))$$

where

x is a vertex of graph X

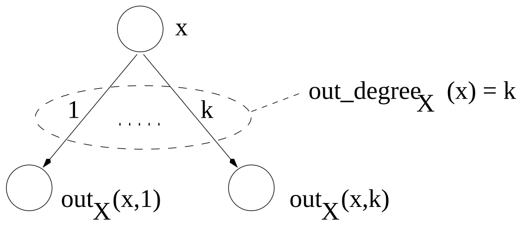
f is a sigmoidal function

N_L is the number of units encoding the label $\mathbf{l} = \phi_X(x)$ attached to the current input x

$\hat{w}_j$ are the weights on the recursive connections

Generalized Recursive Neuron (cont.)

- the output is computed recursively on the offsprings

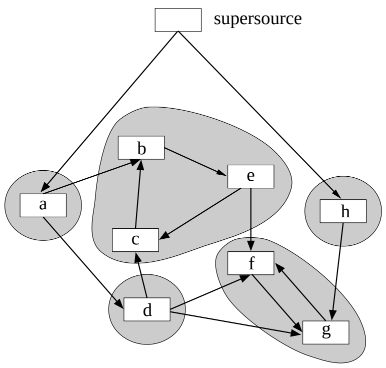


- the following two conditions must hold:
  - Number of Connections: as many recursive connections as the domain valence;
  - Supersource: graph X must have a reference *supersource*.

Supersource

graph without a supersource $\rightarrow$ add an extra vertex s

- with a minimal number of outgoing edges
- such that s is a supersource for the new graph



EXAMPLES (Generalized Recursive Neuron)

Sequence: $x_k \to x_{k-1} \to \ldots \to x_1$

- Generalized recursive neuron

$$o(x_1) = f(\textstyle\sum_{j=1}^{N_L} w_j l_j^{(1)})$$

$$o(x_i) = f(\textstyle\sum_{j=1}^{N_L} w_j l_j^{(i)} + \hat{w}_1 o(x_{i-1})) \quad i = 2, \ldots, k$$

- Standard recurrent neuron (time t of processing explicit)

$$o(t=1) = f(\textstyle\sum_{j=1}^{N_L} w_j l_j^{(1)})$$

$$o(t) = f(\textstyle\sum_{j=1}^{N_L} w_j l_j^{(t)} + \hat{w}_1 o(t-1)) \quad t = 2, \ldots, k$$

EXAMPLES (Generalized Recursive Neuron)

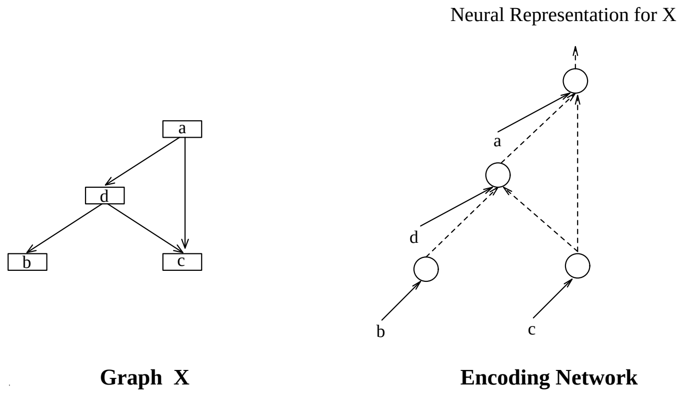
Graph with cycle

$$\begin{array}{ccccc} x_6 & \rightarrow & x_5 & \leftarrow & x_4 \\ & & \downarrow & & \uparrow \\ & & x_3 & \rightarrow & x_2 & \rightarrow & x_1 \end{array}$$

$$\begin{aligned}
o(x_1) &= f(\textstyle\sum_{j=1}^{N_L} w_j l_j^{(1)}) \\
o(x_2) &= f(\textstyle\sum_{j=1}^{N_L} w_j l_j^{(2)} + \hat{w}_1 o(x_1) + \hat{w}_2 o(x_4)) \\
o(x_3) &= f(\textstyle\sum_{j=1}^{N_L} w_j l_j^{(3)} + \hat{w}_1 o(x_2)) \\
o(x_4) &= f(\textstyle\sum_{j=1}^{N_L} w_j l_j^{(4)} + \hat{w}_1 o(x_5)) \\
o(x_5) &= f(\textstyle\sum_{j=1}^{N_L} w_j l_j^{(5)} + \hat{w}_1 o(x_3)) \\
o(x_6) &= f(\textstyle\sum_{j=1}^{N_L} w_j l_j^{(6)} + \hat{w}_1 o(x_5))
\end{aligned}$$

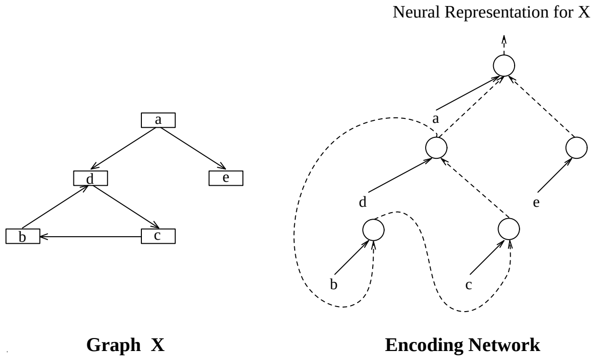
ENCODING NETWORK

since each graph X has a supersource s, then we can see $o(s)$ as computed by an **Encoding Network** obtained by replicating the (set of) generalized recursive neuron(s) according to the graph topology



ENCODING NETWORK (cont.)

if the graph X contains a cycle, then the **Encoding Network** is recurrent

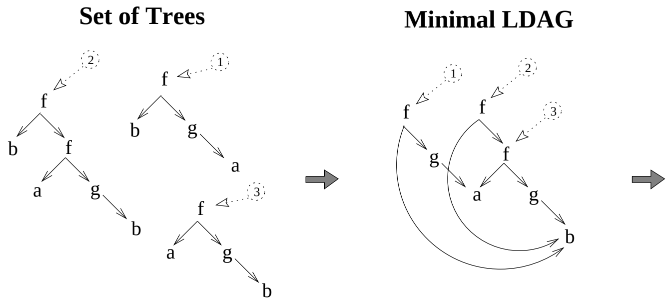


WARNING: there is no guarantee that the Encoding Network is convergent

OPTIMIZATION TRAINING SET

if we only have DAGs, then the training set can be optimized:

1. all the DAGs in the training set are merged into a single minimal DAG, i.e., a DAG with a minimal number of vertices
2. a topological sort on the vertices of the minimal DAG is performed to determine the updating order on the vertices for the network



Sorted Training Set

#	label	left	right
0	b	nil	nil
1	a	nil	nil
2	g	nil	0
3	g	nil	1
4	f	1	2
5	f	0	3
6	f	0	4

NEURAL NETWORKS FOR STRUCTURES

- Back-Propagation Networks
- Simple Recurrent Networks (Elman-Style)
- Cascade-Correlation
- Neural Trees
- Real-Time Recurrent Networks

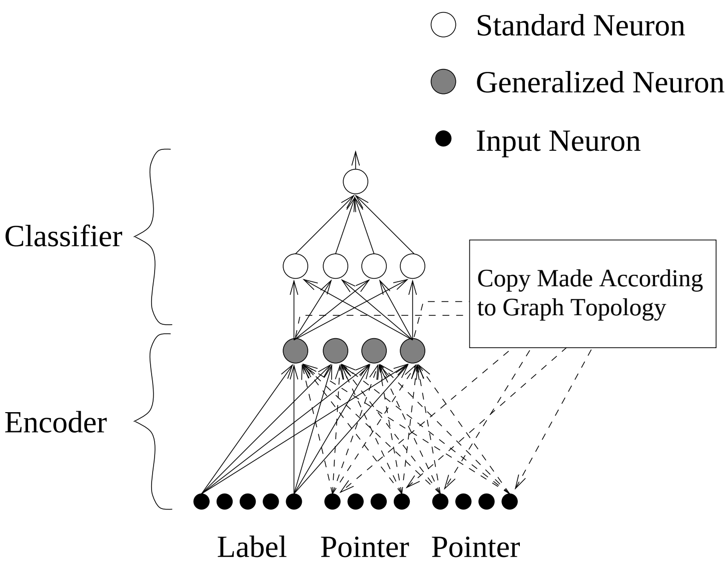
$+$

- Generalized Recursive Neurons

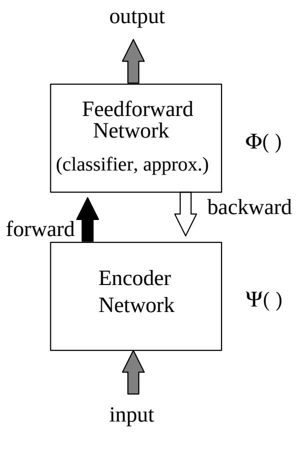
$\Downarrow$

- **NEURAL NETWORKS FOR STRUCTURES**

Back-Propagation Networks

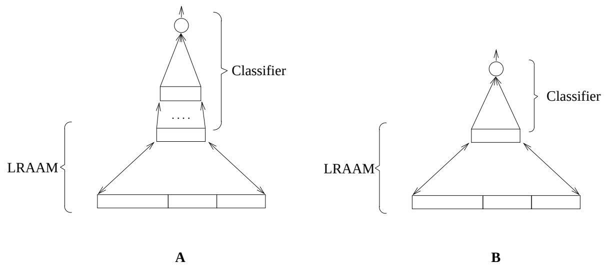


Back-Propagation Networks (cont.)



- DAGs → Back-Propagation Through Structure (see Goller)
- Cyclic Graphs → Extension of Recurrent Back-Propagation (Pineda)

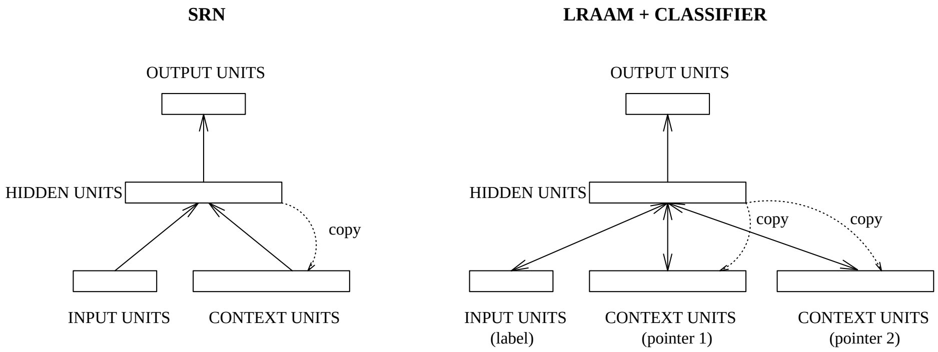
LRAAM-based Networks



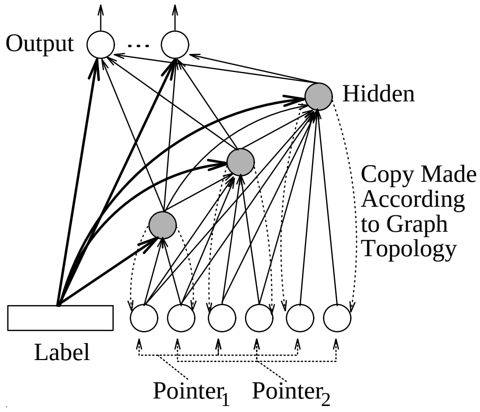
Several options for training networks **A** and **B** are available:

- different learning rates for the classifier and the LRAAM
- classifier training is not started until **x** percent of the training set has been correctly encoded and subsequently decoded by the LRAAM
- the error from the classifier is backpropagated across **y** levels of the structures encoded by the LRAAM

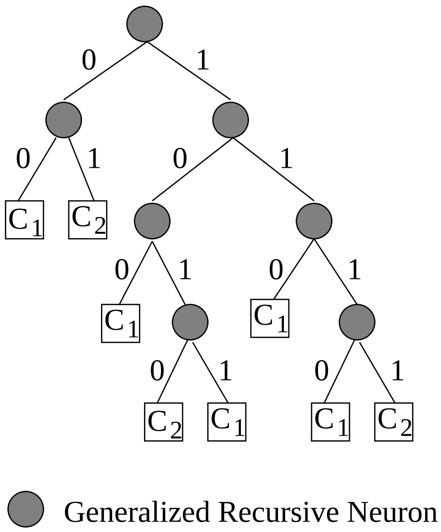
Elman-Style Networks



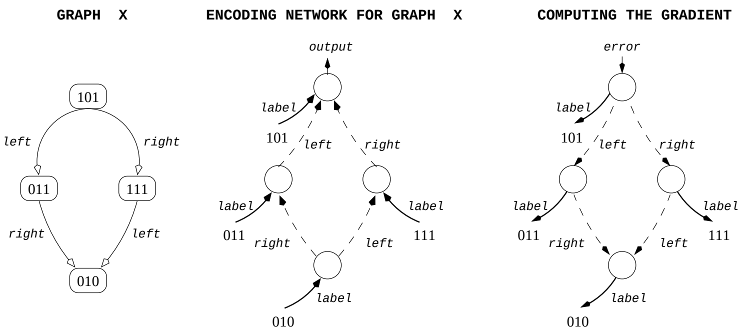
Cascade-Correlation



Neural Tree

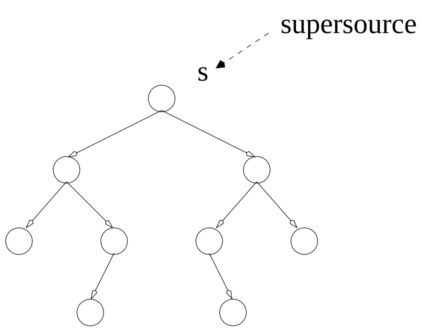


Computing the Gradient



Computing the Real-Time Gradient

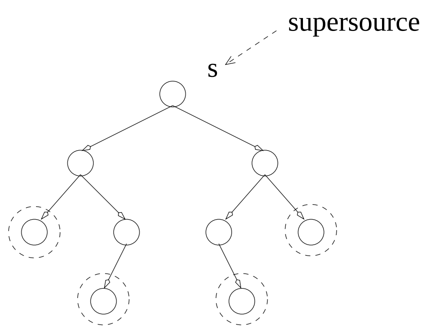
$$o(s) = f(W^t L(s) + \sum_{j=1}^{out\_degree(s)} \hat{w}_j o(out(s,j)))$$



Computing the Real-Time Gradient (cont.)

Consider v such that $out\_degree(v) = 0$, then

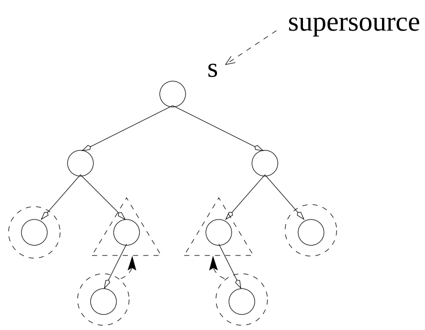
$$\frac{\partial o(v)}{\partial w_i} = f' L(v)_i \quad \frac{\partial o(v)}{\partial \hat{w}_j} = 0$$



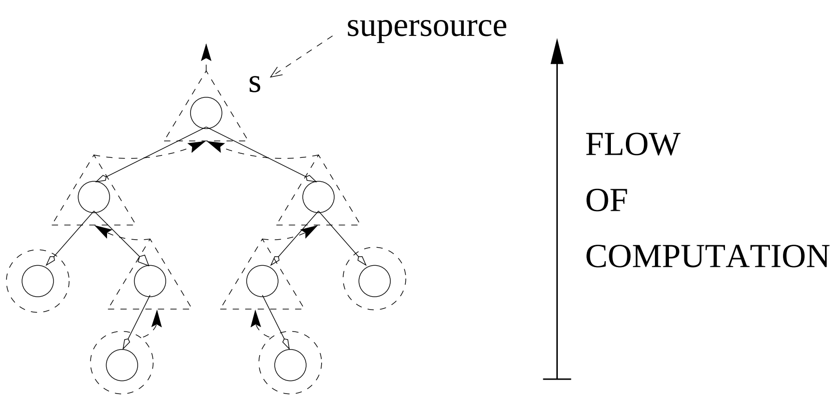
Computing the Real-Time Gradient (cont.)

$$\frac{\partial o(v)}{\partial w_i} = f'(L(v)_i + \sum_{k=1}^{out\_degree(v)} \hat{w}_k \frac{\partial o(out(v,k))}{\partial w_i})$$

$$\frac{\partial o(v)}{\partial \hat{w}_j} = f'(o(out(v,j)) + \sum_{k=1}^{out\_degree(v)} \hat{w}_k \frac{\partial o(out(v,k))}{\partial \hat{w}_j})$$



Computing the Real-Time Gradient (cont.)



Computational and Complexity Issues

- Tree Grammars and Tree Automata
- Computational Power of Some Models
- Complexity Issues
- Approximation

COMPUTATIONAL POWER

Recurrent Networks $\Longleftrightarrow$ *Finite State Automata*

Networks for Structures $\Longleftrightarrow$ Tree Automata

Tree Grammars in Expansive Form

$\Updownarrow$

Frontier-to-Root Tree Automata

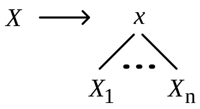
TREE GRAMMARS

TREE GRAMMAR $G_t = (V, r, P, S)$

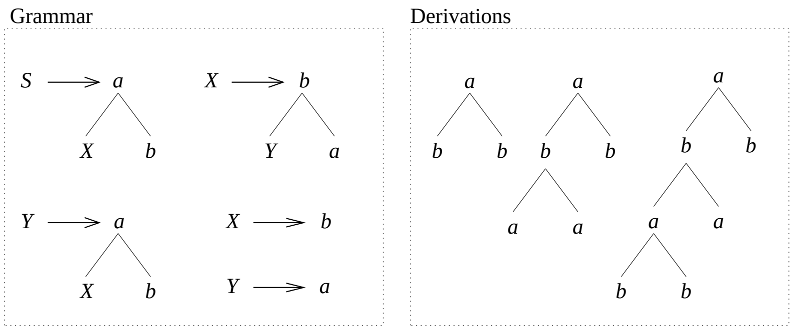
- $V = N \cup \Sigma$ is the grammar alphabet (nonterminals and terminals)
- (V, r) a ranked alphabet
- productions in P are of the form $T_i \rightarrow T_j$, where T_i and T_j are trees
- S in T_V is a finite set of starting trees
- T_V denotes the set of trees with nodes labeled by elements in V

TREE GRAMMARS (cont.)

TREE GRAMMARS in EXPANSIVE FORM

- all productions are of the form 

- example



TREE AUTOMATA (frontier-to-root)

TREE AUTOMATON $\mathcal{A}_t = (Q, F, \{f_a | a \text{ in } \Sigma\})$ (deterministic)

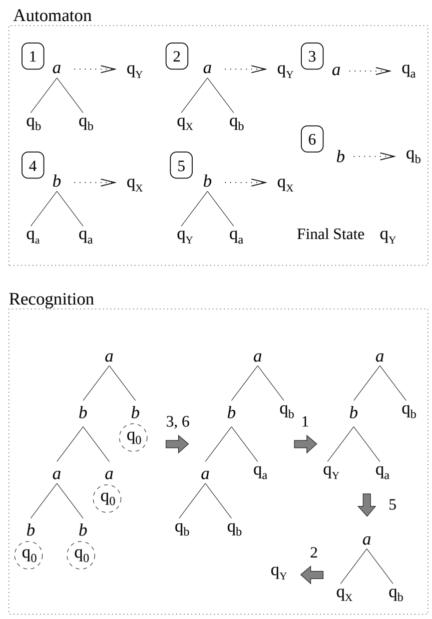
- Σ is a ranked alphabet
- Q is a finite set of states
- F is a set of final states, a subset of Q
- $\{f_a | a \text{ in } \Sigma\}$ is a set of transition functions $f_a : Q^m \to Q$ such that m is a rank of the symbol $a \in \Sigma$

TREE AUTOMATA (cont.)

RECOGNITION PROCESS (informal)

1. The frontier of the state tree is labeled q_0 (the status associated to nodes with $m = 0$, i.e, the leaves)
2. For any node in the input labeled l with rank m, the corresponding node of the state tree is labeled $f_l(q_1, \ldots, q_m)$, where $q_1, \ldots, q_m$ are the states labeling the offsprings of that state tree node
3. an input tree is accepted by $\mathcal{A}_t$ if the automaton can enter a final state upon encountering the root

EXAMPLE OF TREE AUTOMATON (FRA)



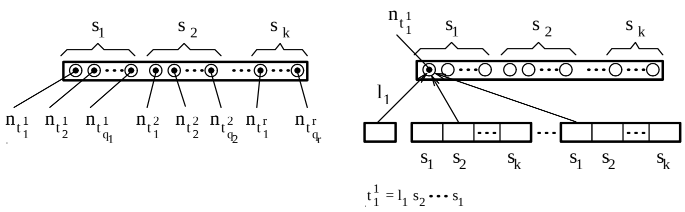
COMPUTATIONAL RESULTS

Theorem An Elman-style network can simulate any FRA.

Proof Schema

For *each* tuple $t_i^j \in V_{s_j}$, $i \in [1, \ldots, \#V_{s_j}]$, and for *each* state s_j, we introduce a unit $n_{t_i^j}$ at the hidden level

The units $n_{t_i^j}$ will represent the state s_j at the hidden level of the network



COMPUTATIONAL RESULTS (cont.)

Theorem A Cascade-Correlation network for structures (CCS) cannot simulate any FRA.

Proof Schema

CCS with $N_v = 1$ is equivalent to Recurrent Cascade - Correlation (RCC)

RCC cannot simulate any Finite State Machine (FSM) (Giles et al., 1995; Kremer, 1995; Kremer, 1996)

FSM is equivalent to FRA with relations having rank 1

Theorem A Neural Tree for structures cannot simulate any FRA.

COMPUTATIONAL RESULTS (cont.)

Theorem Any Neural Tree for structures can be implemented by a CCS.

Proof Schema

- A Neural Tree (NT) can be restructured as a layered network by following the rules below:

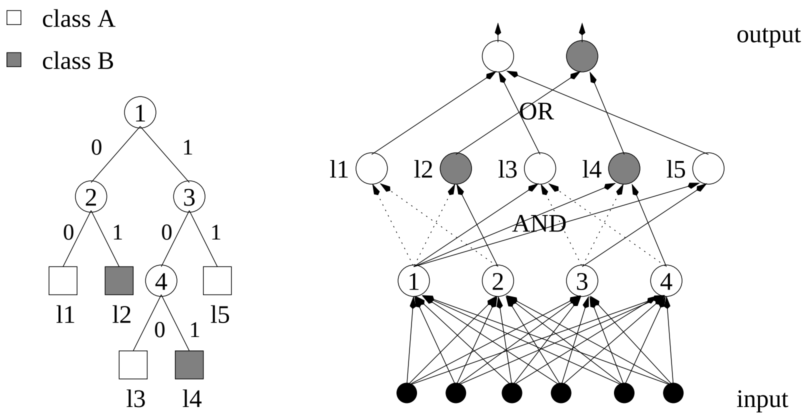
(R1) same units as the internal nodes of the NT

(R2) **and** of nodes which belong to the same path

(R3) **or** of paths

- Layered network implemented by a Cascade-Correlation network.

COMPUTATIONAL RESULTS (example)

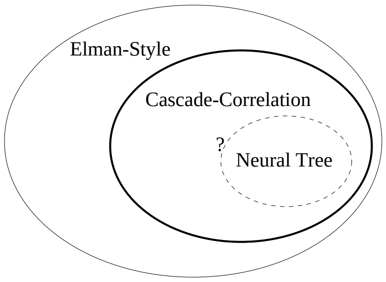


COMPUTATIONAL RESULTS (cont.)

Corollary A Neural Tree for **sequences** cannot simulate any Finite State Machine.

Corollary Any Neural Tree for **sequences** can be simulated by a Recurrent Cascade-Correlation network.

SUMMARY



COMPLEXITY

Question:

how many nodes are needed to implement an m-state FRA ?

- l input-output labels (alphabet) encoded in $\lceil \log l \rceil$ bits
- maximum rank N_v
- m different states encoded in $\lceil \log m \rceil$ bits
- transition (and output) function as a boolean logic function:

$$\Phi : \{0,1\}^{\log l + N_v \log m} \rightarrow \{0,1\}^{\log l + \log m}$$

COMPLEXITY RESULTS: four layers

Theorem[Four Layers Insertion]

A four layers recurrent neural network can implement FRAs with m states, l input-output labels, and maximum rank N_v, with a node complexity of $O(\sqrt{\frac{(\log l+\log m)lm^{N_v}}{\log l+N_v \log m}})$.

Proof

Follows from the lemma:

Lemma (Lupanov, 1972) Arbitrary boolean logic functions of the form $f:\{0,1\}^x \rightarrow \{0,1\}^y$ can be implemented in a four layer network of perceptrons with a node complexity of $O(\sqrt{y2^x/(x-\log y)})$.

COMPLEXITY RESULTS: three layers

Theorem[Three Layers Insertion]

A three layers recurrent neural network can implement FRAs with m states, l input-output labels, and maximum rank N_v, with:

- a node complexity of $O((\log l+\log m)\sqrt{lm^{N_v}})$
- a number of connections of $O((\log l+\log m)lm^{N_v})$.

Proof

Follows from a construction based on the telescopic lemma (Siu et. al. 1995)

Approximation

- S set of Σ-labeled trees
- $\bar{f} : S \rightarrow \Re^q$
- $\mathcal{F}$ set of functions implementable by a feed-forward NN for structures

Theorem (Hammer, 1996) For any $\epsilon > 0$ there exists a function $f^\epsilon \in \mathcal{F}$ such that $\forall s \in S,\ P(|f^\epsilon(s) - \bar{f}(s)| > \epsilon) < \epsilon$

NN Memory Models for Structures

Outline

- **PROBLEM 3**: Storing and Recalling Structures
- From LRAAM to the Content Access Network
- Stability Results for the Content Access Network
- One Step Beyond: Generalized Hopfield Model

LRAAM network

$$\mathbf{F}_E(\vec{x}) = \mathbf{F}(\mathbf{E}\vec{x} + \vec{\theta}_H) \equiv \vec{h},$$
$$\mathbf{F}_D(\vec{h}) = \mathbf{F}(\mathbf{D}\vec{h} + \vec{\theta}_O) \equiv \vec{o},$$

Encoding Matrix:

$$\mathbf{E} = [\mathbf{E}^{(l)}, \mathbf{E}^{(p_1)}, \ldots, \mathbf{E}^{(p_n)}]$$

Decoding Matrix:

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}^{(l)} \\ \mathbf{D}^{(p_1)} \\ \vdots \\ \mathbf{D}^{(p_n)} \end{bmatrix}$$

CA Network

The Content Access (CA) network:

$$\textbf{LRAAM network} \; + \; \vec{o} \equiv \vec{x}.$$

- discrete time asymmetric Hopfield network
- pattern in the LRAAM training set $\rightarrow$ fixed point of the CA network
- fixed points asymptotically stable $\rightarrow$ memories
- the structure of each pattern exploited to constrain the relaxation of the CA network

Constraints

- constraint c is defined as a couple $(id, \vec{k}^{(id)})$, where:
  - $id \in \{l, p_1, \ldots, p_n\}$ is the name of a field in the network
  - $\vec{k}^{(id)}$ is the *access key*
- during the relaxation of the CA network, the output of the field id is fixed to $\vec{k}^{(id)}$
- access procedure defined as a set C of constraints

Network Dynamics

given a set C of constraints

- $\mathbf{E}^C$ is the matrix obtained by removing from $\mathbf{E}$ the submatrices associated with fields whose names appear in C (the same for $\mathbf{D}^C$)
- $\vec{o}^{\,C}$ is the vector obtained by removing from $\vec{o}$ the subvectors associated with fields whose names appear in C (the same for $\vec{\theta}_O$)

Dynamics:

$$\mathbf{F}_{E^C}(\vec{o}^{\,C}(t)) = \mathbf{F}(\mathbf{E}^C \vec{o}^{\,C}(t) + \vec{\theta}_H + \vec{\theta}^{\,C}_{key}) \equiv \vec{h}(t)$$

$$\mathbf{F}_{D^C}(\vec{h}(t)) = \mathbf{F}(D^C \vec{h}(t) + \vec{\theta}^{\,C}_O) \equiv \vec{o}^{\,C}(t+1)$$

$$\vec{\theta}^{\,C}_{key} = \sum_{(id, \vec{k}^{(id)}) \in C} \mathbf{E}^{(id)} \vec{k}^{(id)}$$

Access by Content

Access procedure on the CA network:

1. choosing one or more fields in the input layer according to the access key(s)
2. clamping the output of these units to the information known, i.e., the access key(s)
3. randomly setting the output of the remaining units in the network
4. letting the unclamped units in the network to relax into a stable state.

Validation of a Fixed Point

A validation test of the reached stable state can be performed:

1. by unfreezing the clamped units in the input layer
2. if the stable state is no longer stable the result of the procedure is considered to be **wrong** and another run is performed
3. otherwise the stable state is considered a **success**

Stability

pointer transformations:

$$\mathbf{F}^{(p_j)}(\vec{d}\,) = \mathbf{F}(\mathbf{D}^{(p_j)}\vec{d} + \vec{\theta}_O^{(p_j)}), \quad j = 1, \ldots, n,$$

label decoding function:

$$\mathbf{F}^{(l)}(\vec{d}\,) = \mathbf{F}(\mathbf{D}^{(l)}\vec{d} + \vec{\theta}_O^{(l)})$$

- $\mathbf{J}^{(p_j)}(\vec{d}\,) = \mathbf{\Lambda}(\vec{net}^{(p_j)})\mathbf{D}^{(p_j)}$ is the Jacobian matrix of $\mathbf{F}^{(p_j)}(\vec{d}\,)$
- $\vec{net}^{(p_j)} = \mathbf{D}^{(p_j)}\vec{d} + \theta_O^{(p_j)}$
- $\mathbf{\Lambda}(\vec{net}^{(p_j)}) \in \Re^{N_H \times N_H}$ is the diagonal matrix whose ith diagonal element is $f'(\vec{net}_i^{(p_j)})$
- $(\vec{d}_H, \vec{d}_O^{\,C})$ fixed point of the **CA** network under the set of constraints C

Stability: Sufficient Conditions

Theorem A fixed point is asymptotically stable if at least one of the following two conditions is verified:

Condition at layer H

$$\sum_{k=1}^{N_H} |b_{ik}| < 1, \qquad i = 1, \ldots, N_H$$

where b_{ik} is the element (i,k) of $\mathbf{B}^{(H,C)} = \mathbf{J}^{(E^C)}(\vec{d}_O^{\,C})\mathbf{J}^{(D^C)}(\vec{d}_H)$;

Condition at layer O

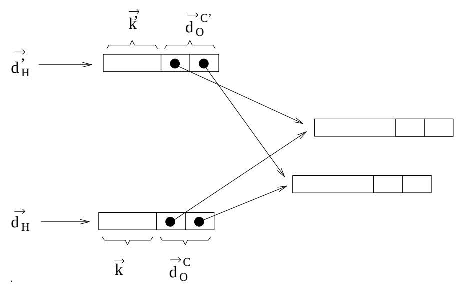
$$\sum_{k=1}^{N_O^C} |b_{ik}| < 1, \qquad i = 1, \ldots, N_O^C$$

where b_{ik} is the element (i,k) of $\mathbf{B}^{(O,C)} = \mathbf{J}^{(D^C)}(\vec{d}_H)\mathbf{J}^{(E^C)}(\vec{d}_O^{\,C})$.

Stability: Sufficient Conditions (cont.)

There are other theorems defining sufficient conditions for:

- Preservation of asymptotical stability of a fixed point when adding a constraint to C
- Extension of asymptotical stability to related components, e.g.:

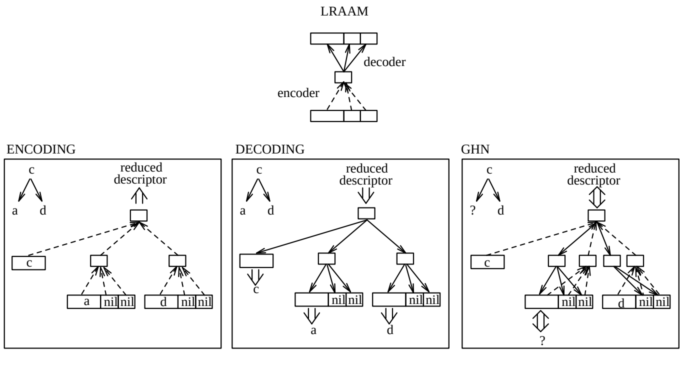


- Other cases

ACCESS by PARTIAL STRUCTURE

Question: how can we modify the model to cope with **access by partial structure** ?

Answer: by combining the encoding and decoding procedures

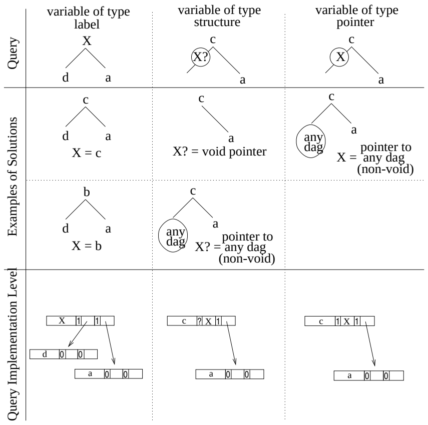


Generalized Hopfield Network

Query (def.)

QUERY: Tree with

- constants and variables of type: label, pointer, and structure
- partially defined labels, and pointers



Network Generation

Question: how do we construct the GHN from the query ?

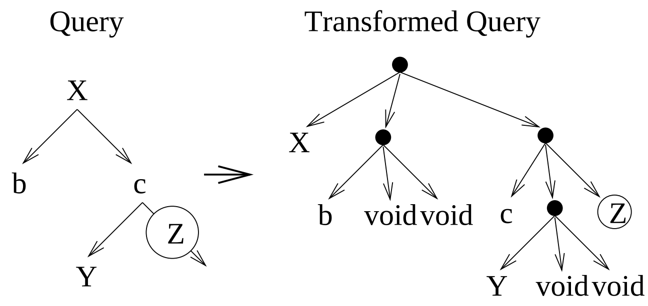
(t1) Query → Transformed Query

(t2) Transformed Query → Query Graph

(t3) Query Graph → Query Network

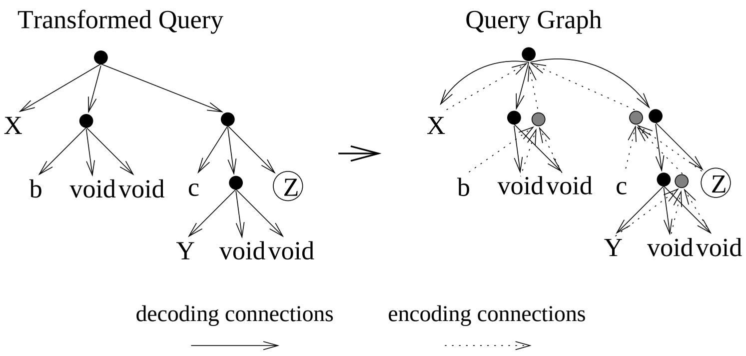
Query → Transformed Query

- the Query tree is transformed in a tree with:
  - unlabeled internal nodes
  - labeled leaves which correspond to nodes of the original query
- variable and void pointers are explicitely represented by leaves



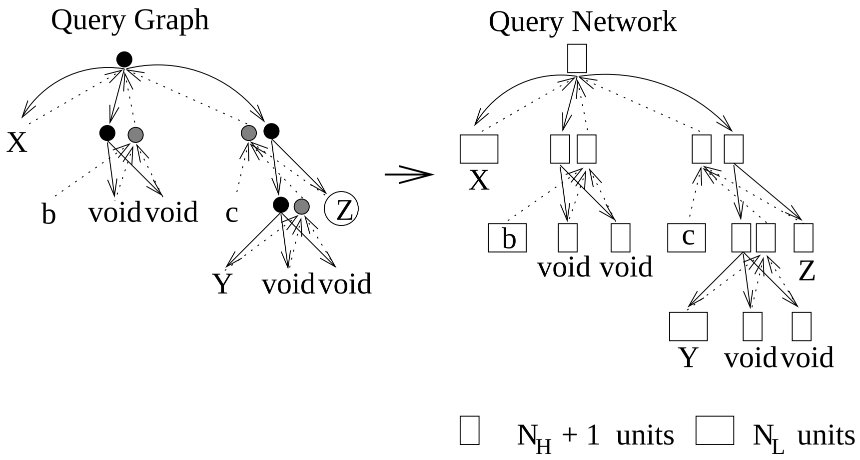
Transformed Query → Query Graph

- the Query Graph defines the general structure of the network implementing the query
- the Query Graph is obtained from the Transformed Query by duplicating internal nodes



Query Graph → Query Network

- the Query Network is build from the Query Graph using the weights of the LRAAM as bulding blocks.



Multiple Solutions

- Uniform Random Search

  uniformly distributed random pointers for the top layer

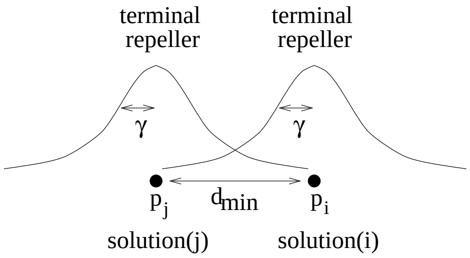
- Terminal Repellers

  introduction of Terminal Repellers (Zak, 89) at the top layer

$$\dot{x}_i^{(top)} + x_i^{(top)} = \sum_{j=1}^{N_O} E_{ij} f(x_j^{IN(top)}) +$$

$$\alpha \sum_{k=0}^{M} (x_i^{(top)} - p_i^{(s^k)})^{1/3} e^{-\gamma(\sum_{i=1}^{N_H} (x_i^{(top)} - p_i^{(s^k)})^2)}$$

How to Set γ ?



$$\gamma = -\frac{log(\epsilon)}{d_{min}^2}$$

$$d_{min} = \min_{ij} \quad \{d_{ij} | d_{ij} = \|p^{(i)} - p^{(j)}\|, \text{ and } p^{(i)},\ p^{(j)} \text{ are in the training set}\}$$

How to Set α ?

Problem:

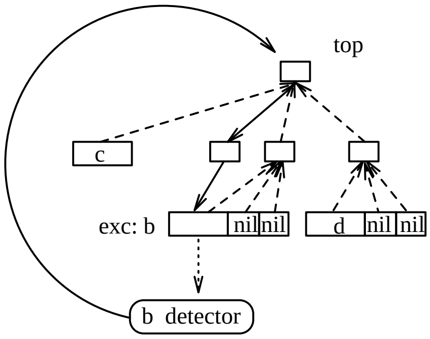
Terminal repellers can introduce cycles

Solution:

Decay equation for α

$$\dot{\alpha} = -\frac{1}{\tau}\alpha$$

Exceptions



Repeller

$$\alpha(x_i^{(top)} - x*_i^{(top)})^{1/3} e^{-\gamma(\sum_{i=1}^{N_H}(x_i^{(top)} - x*_i^{(top)})^2)}$$

Research Directions

Soft Computing BISC [1]

integration of fuzzy logic, neural networks, and probabilistic reasoning

Neurocontrol [2]

dynamic optimization of complex systems based on neural networks

Intelligent Agents

largely independent systems performing actions on behalf of the user

Autonomous Systems

self-contained systems acting in a real environment

[1] *B*erkeley *I*nitiative in *S*oft *C*omputing, [Zadeh, 1990]

[2] [Werbos, 1992]

Knowledge Fusion

integration of knowledge and information available in different modes (linguistic, visual, aural, etc.)

Artificial Life

development of complex systems based on evolutionary concepts

Data Mining

extraction of useful knowledge from huge amounts of data

Adaptive User Interfaces

systems learns preferences of the user

Conclusions

neural networks and structured knowledge

fundamental research

formal models to deal with validation and verification of complex systems
methods for the integration of different computation paradigms
representation methods for various types of knowledge
connections to cognitive phenomena

tools

development tools and environment for the efficient implementation of hybrid systems

applications

real-world applications to gain experience

Outlook

neural networks and structured knowledge

capabilities

more powerful systems through combinations of techniques

challenges

new, exciting research problems

promising applications

increasing importance

as successful applications and experience build up

visibility

might increase (see above)

might decrease because hybrid becomes the norm

References

[A. Sperduti, 1995] A. Sperduti, A. S. (1995). Modular labeling raam. In *International Conference on Neural Networks and Genetic Algorithms*, pages 368–371.

[Ajjanagadde and Shastri, 1991a] Ajjanagadde, V. and Shastri, L. (1991a). Rules and Variables in Neural Nets. *Neural Computation*, 3:121–134.

[Ajjanagadde and Shastri, 1991b] Ajjanagadde, V. and Shastri, L. (1991b). Rules and variables in neural nets. *Neural Computation*, 3(1):121–134.

[Ajjanagadde and Shastri, 1991c] Ajjanagadde, V. and Shastri, L. (1991c). Rules and variables in neural nets. *Neural Computation*, 3:121–134.

[Ajjanagadde and Shastri, 1995] Ajjanagadde, V. and Shastri, L. (1995). *Reasoning with Rules and Variables in Neural Networks*, chapter 11. In [Goonatilake and Khebbal, 1995].

[Anderson, 1990] Anderson, J. (1990). Data Representation, Neural Networks, and Hybrid Computation. In *Neural Networks for Knowledge Representation and Inference.*

[Barnden, 1988a] Barnden, J. (1988a). Simulations of Conposit, a Supra-Connectionist Architecture for Commonsense Reasoning. In *2nd Symposium on the Frontiers of Massively Parallel Computation, Fairfax, VA.*, Las Cruces.

[Barnden, 1988b] Barnden, J. (August 1988b). The Right of Free Assosiacion: Relative-Position Encoding for Connectionist Data Structures. *In Procs. of the Tenth Annual Conference of the Cognitive Science Society.*

[Barnden and Holyoak, 1992] Barnden, J. and Holyoak, K. (1992). *Advances in Connectinoist and Neural Computation Theory*, volume II: Analogical Connections. Ablex, Norwood, NJ.

[Barnden, 1989] Barnden, J. A. (1989). Neural-net implementation of complex symbol-processing in a mental model approach to syllogistic reasoning. In *Proc. of the 11 th IJCAI*, pages 568–573, Detroit, MI.

[Barnden, 1990] Barnden, J. A. (1990). Syllogistic Mental Models: Exercising Some Connectionist Representation and Control Models. Memoranda in Computer and Cognitive Science MCCS-90-204, Computing Research Laboratory, New Mexico State University, Las Cruces, NM.

[Barnden, 1991] Barnden, J. A. (1991). Encoding complex symbolic data structures with some unusual connectionist techniques. In [Barnden and Pollack, 1991].

[Barnden, 1994] Barnden, J. A. (1994). Complex symbol-processing in conposit, a transiently localist connectionist architecture. In [Sun and Bookman, 1994], chapter 2.

[Barnden and Pollack, 1991] Barnden, J. A. and Pollack, J. B., editors (1991). *Advances in Connectionist and Neural Computation Theory: High-level connectionist models*, volume 1. Ablex Publishing Corporation.

[Barnden and Srinivas, 1990] Barnden, J. A. and Srinivas, K. (1990). Overcoming Rule-Based Rigidity and Connectionist Limitations through Massively-Parallel Case-Based Reasoning. Memoranda in Computer and Cognitive Science MCCS-90-187, Computing Research Laboratory, New Mexico State University, Las Cruces, NM.

[Benaim, 1995] Benaim, M. (1995). Convergence theorems for hybrid learning rules. *Neural Computation*, 7(1):19–24.

[Bengio, 1996] Bengio, Y. (1996). *Neural Networks for Speech and Sequence Recognition.* International Thomson Computer Press.

[Bengio et al., 1995] Bengio, Y., LeCun, Y., Nohl, C., and Burges, C. (1995). LeRec: A NN/HMM hybrid for on-line handwriting recognition. *Neural Computation*, 7(6):1289–1303.

[Bennani, 1995] Bennani, Y. (1995). A modular and hybrid connectionist system for speaker identification. *Neural Computation*, 7(4):791–798.

[Bickhard and Terveen, 1995] Bickhard, M. H. and Terveen, L. (1995). *Foundational Issues in Artificial Intelligence and Cognitive Science: Impasse and Solution.* Elsevier Science.

[Charniak and Santos, 1987] Charniak, E. and Santos, E. (1987). A connectionist context-free parser which is not context-free but then it is not really connectionist either. In *Proceedings of 9th Annual Conference of the Cognitive Science Society.*

[Derthick, 1990] Derthick, M. (1990). Mundane reasoning by settling on a plausible model. *Artificial Intelligence*, 46(1-2):107–158.

[Diederich and Kurfeß, 1993] Diederich, J. and Kurfeß, F. J. (1993). Learning unification with neural networks. Technical report, Universität Ulm, Abteilung Neuroinformatik, D-89069 Ulm. (also in Proc. Australian Neural Networks Conference 1994).

[Dorffner, 1991] Dorffner, G. (1991). *Konnektionismus.* Teubner.

[D.R.Mani and Shastri, 1994] D.R.Mani and Shastri, L. (1994). Reflexive reasoning with multiple instantiation in a connectionist reasoning system with a type hierarchy. *Connection Science.* special issue on "Integrating Neural and Symbolic Processes".

[Duwe et al., 1994] Duwe, I., Kurfeß, F. J., Paass, G., and Vogel, S., editors (1994). *Konnektionismus und neuronale Netze – Beiträge zur Herbstschule HeKoNN 94*, number 242 in GMD-Studien, D-53754 Sankt Augustin. Gesellschaft für Mathematik und Datenverarbeitung (GMD).

[Elman, 1990] Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14:179–211.

[Fahlman, 1991] Fahlman, S. (1991). The recurrent cascade-correlation architecture. In Lippmann, R., Moody, J., and Touretzky, D., editors, *Advances in Neural Information Processing Systems 3*, pages 190–196, San Mateo, CA. Morgan Kaufmann Publishers.

[Fahlman and Lebiere, 1990] Fahlman, S. E. and Lebiere, C. (1990). The cascade-correlation learning architecture. In Touretzky, D. S., editor, *Advances in Neural Information Processing Systems 2*, pages 524–532. San Mateo, CA: Morgan Kaufmann.

[Fu, 1992] Fu, L. (1992). A parallel distributed approach for knowledge-based inference and learning. In [Kandel and Langholz, 1992].

[Giacometti, 1992] Giacometti, A. (1992). *Hybrid Models for Expertise.* French doctoral thesis, Ecole Nationale Supérieure des Télécommunications, Paris.

[Giacometti et al., 1992] Giacometti, A., Amy, B., and Grumbach, A. (1992). Theory and experiments in connectionist ai: A tightly coupled hybrid system. In *International Conference on Artificial Neural Networks (ICANN '92)*, Amsterdam. Elsevier.

[Goller and Küchler, 1996] Goller, C. and Küchler, A. (1996). Learning task-dependent distributed structure-representations by backpropagation through structure. In *IEEE International Conference on Neural Networks*, pages 347–352.

[Goller et al., 1994] Goller, C., Sperduti, A., and Starita, A. (1994). Learning distributed representations for the classification of terms. Technical Report AR-94-05, Institut für Informatik, Technische Universität München.

[Goonatilake and Khebbal, 1995] Goonatilake, S. and Khebbal, S. (1995). *Intelligent Hybrid Systems.* John Wiley & Sons.

[Hammer, 1996] Hammer, B. (1996). Universal Approximation of Mappings on Structured Objects using the Folding Architecture. Osnabrücker Schriften zur Mathematik Reihe P, Heft 183, Fachbereich Mathematik/Informatik, Universität Osnabrück, 49069 Osnabrück.

[Hammer and Sperschneider, 1997] Hammer, B. and Sperschneider, V. (1997). Neural networks can approximate mappings on structured objects.

In *ICCIN '97, 2nd International Conference on Computational Intelligence and Neuroscience.*

[Hanks, 1979] Hanks, P., editor (1979). *Collins Dictionary of the English Language.* William Collins Sons & Co. Ltd., London, Glasgow.

[Herrmann,] Herrmann, C. S. A hybrid fuzzy-neural expert system for diagnosis. In *Proc. of the 14 th IJCAI*, Montreal, Canada.

[Hinton, 1990] Hinton, G. E. (1990). Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, 46:47–75.

[Hofstadter and Dennett, 1981] Hofstadter, D. R. and Dennett, D. C. (1981). *The Mind's I – Fantasies and Reflections on Self and Soul.* Penguin.

[Hölldobler and Kurfeß, 1991] Hölldobler, S. and Kurfeß, F. J. (1991). CHCL – A Connectionist Inference System. In Fronhöfer, B. and Wrightson, G., editors, *Parallelization in Inference Systems*, Lecture Notes in Artificial Intelligence, pages 318–342. Springer.

[Holyoak and Thagard, 1989] Holyoak, K. and Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive science*, 13:295–355.

[Kanal and Raghavan, 1992] Kanal, L. and Raghavan, S. (1992). Hybrid Systems- A Key to Intelligent Pattern Recognition. In *IJCNN-92, Int. Joint. Conf on Neural Networks,.*

[Kandel and Langholz, 1992] Kandel, A. and Langholz, G., editors (1992). *Hybrid Architectures for Intelligent Systems.* CRC Press.

[Khebbal and Sharpington, 1996] Khebbal, S. and Sharpington, C., editors (1996). *Rapid Application Generation of Business and Finance Software.* Kluwer.

[Kokinov, 1992] Kokinov, B. N. (1992). A hybrid model of reasoning by analogy. In [Barnden and Holyoak, 1992].

[Kozato,] Kozato, F. A hybrid rule-based system – how variables are involved in connectionist rule-based systems. In *Proc. of the 14 th IJCAI*, Montreal, Canada. reviewed for IJCAI '95.

[Kremer, 1995] Kremer, S. (1995). On the computational power of elman-style recurrent networks. *IEEE Transactions on Neural Networks*, 6(4):1000–1004.

[Kremer, 1996] Kremer, S. (1996). Finite state automata that recurrent cascade-correlation cannot represent. In Touretzky, D., Mozer, M., and Hasselno, M., editors, *Advances in Neural Information Processing Systems 7.* MIT Press. 612-618.

[Kuncicky et al., 1992] Kuncicky, D. C., Hruska, S., and Lacher, R. C. (1992). Hybrid systems: The equivalence of expert system and neural network inference. *International Journal of Expert Systems.*

[Kuncicky et al., 1991] Kuncicky, D. C., Hruska, S. I., and Lacher, R. C. (1991). Hybrid Systems: The Equivalence of Rule-Based Expert System and Artificial Neural Network Inference. *International Journal of Expert Systems.*

[Kurfeß, 1993] Kurfeß, F. J. (1993). WINA – Wissensverarbeitung mit Symbolischen und Sub-Symbolischen Mechanismen. In Paaß, G. and Kurfeß, F. J., editors, *Wissensverarbeitung mit neuronalen Netzen*, number 221 in GMD-Studien, pages 45–49. Gesellschaft für Mathematik und Datenverarbeitung (GMD), Schloß Birlinghoven, 53757 Sankt Augustin, Germany. KI '93 Workshop on "Wissensverarbeitung mit neuronalen Netzen (Knowledge Processing with Neural Networks)".

[Kurfeß, 1994] Kurfeß, F. J. (1994). Why neural networks for logic and reasoning. Technical report, University of Ulm, Department of Neural Information Processing, D-89069 Ulm. presented at the Workshop "Logic and Reasoning with Neural Networks" during the International Conference on Logic Programming (ICLP-94).

[Kurfeß, 1995] Kurfeß, F. J. (1995). Knowledge Processing with Symbolic and Sub-Symbolic Mechanisms in the Hybrid System WINA. Technical report, New Jersey Institute of Technology, Computer and Information Science Department, Newark, NJ 07102. accepted for publication, SOCO '95 Workshop, the International ICSC Symposium on Soft Computing, Fuzzy Logic, Artificial Neural Networks and Genetic Algorithms in Rochester, New York, USA (Oct 24-27, 1995).

[Kurfeß and Hölldobler, 1993] Kurfeß, F. J. and Hölldobler, S. (1993). Logic and reasoning in connectionist systems. Tutorial T14, IJCAI 93.

[Kurfeß and Paaß, 1994] Kurfeß, F. J. and Paaß, G. (1994). Integration Neuronaler und Wissensbasierter Ansätze. In Stoyan, H., editor, *Conference

Proceedings KI-94, volume Workshop-Extended Abstracts, Saarbrücken, Germany. GI, Springer.

[Lacher, 1992] Lacher, R. C. (1992). Node error assignment in expert networks. In [Kandel and Langholz, 1992].

[Laird, 1982] Laird, C. (1982). *Webster's New World Thesaurus*. Warner Books, New York. Paperback Edition.

[Lallement et al., 1993] Lallement, Y., Cornu, T., and Vialle, S. (1993). An abstract machine for implementing connectionist and hybrid systems on multi-processor architectures. Technical report, Nancy University, Swiss Federal Institute of Technology, Supelec Metz. preprint from: Parallel Processing for Artificial Intelligence 2; H. Kitano (ed.); Elsevier.

[Lange, 1992] Lange, T. E. (1992). Hybrid Connectionist Models. In Dinsmore, J., editor, *Closing the Gap: Symbolism vs. Connectionism*, chapter 10, pages 237–289. Lawrence Erlbaum Associates.

[Lange and Dyer, 1989] Lange, T. E. and Dyer, M. G. (1989). High-level inferencing in a connectionist network. *Connection Science*, 1:181–217.

[Lange and Wharton, 1992] Lange, T. E. and Wharton, C. M. (1992). REMIND: Retrieval from episodic memory by inferencing and disambiguation. In [Barnden and Holyoak, 1992].

[Lau et al., 1995] Lau, H., Szeto, K., Wong, K., and Yeung, D. (1995). A hybrid expert system for error message classification. In *IWANNT*95*.

[Lonardi et al., 1994] Lonardi, S., Sperduti, A., and Starita, A. (1994). Encoding pyramids by labeling raam. In *IEEE Workshop on Neural Networks for Signal Processing*, pages pp. 651–660.

[MacLennan, 1994] MacLennan, B. (1991 (revised 1994)). Characteristics of Connectionist Knowledge Representation. Technical Report CS-91-147, Computer Science Department, University of Tennessee, Knoxville.

[Medsker, 1994] Medsker, L. (1994). *Hybrid Neural Network and Expert Systems.* Kluwer, Boston.

[Metcalfe, 1991] Metcalfe, E. (1991). Recognition failure and charm. *Psychological Review*, 98(4):529–553.

[Minsky, 1974] Minsky, M. (1974). A Framework for Representing Knowledge. *Artificial Intelligence*, pages 339–419.

[Minsky, 1986] Minsky, M. (1986). *The Society of Mind.* Simon & Schuster, New York.

[Mozer et al., 1994] Mozer, M., Smolensky, P., Touretzky, D., Elman, J., and Weigend, A., editors (1994). *1993 Connectionist Models Summer School*, Hillsdale, NJ. Erlbaum Associates.

[Murdock, 1983] Murdock, B. (1983). A distributed memory model for serial-order information. *Psychological Review*, 90(4):316–338.

[Murdock, 1993] Murdock, B. (1993). Todam2: A model for the storage and retrieval of item, associative, and serial-order information. *Psychological Review*, 100(2):183–203.

[Neal, 1993] Neal, R. M. (1993). Probabilistic inference using markov chain monte carlo methods. Technical report, Department of Computer Science, University of Toronto.

[Newell, 1990] Newell, A. (1990). *Unified Theories of Cognition.* Harvard University Press.

[Newell and Simon, 1976] Newell, A. and Simon, H. (1976). Computer science as empirical enquiry: Symbols and search. *Communications of the ACM*, 19(3):113–126.

[Newquist, 1995] Newquist, H. P. (1995). The dark side of hybrid systems. *AI Expert*, 10(3):41–42.

[Paaß and Kurfeß, 1993] Paaß, G. and Kurfeß, F. J. (1993). Wissensverarbeitung mit neuronalen Netzen. *KI (Künstliche Intelligenz)*, pages 51–54. Sonderheft zur 17. Fachtagung für Künstliche Intelligenz (KI '93).

[Palm et al., 1991] Palm, G., Rückert, U., and Ultsch, A. (1991). Wissensverarbeitung in neuronaler Architektur. In Brauer, W. and Hernández, D., editors, *Verteilte Künstliche Intelligenz und kooperatives Arbeiten*, Informatik-Fachberichte. Springer Verlag, Berlin, Heidelberg, New York.

[Perrone and Cooper, 1993] Perrone, M. P. and Cooper, L. N. (1993). When Networks Disagree: Ensemble Method for Neural Networks. In Mammone, R. J., editor, *Neural Networks for Speech and Image processing.* Chapman-Hall.

[Pineda, 1988] Pineda, F. J. (1988). Dynamics and architecture for neural computation. *Journal of Complexity*, 4:216–245.

[Pinkas, 1990] Pinkas, G. (1990). Connectionist Energy Minimization and Logic Satisfiability. Technical report, Center for Intelligent Computing Systems, Department of Computer Science, Washington University.

[Pinkas, 1991a] Pinkas, G. (1991a). Constructing Proofs in Symmetric Networks. In *NIPS '91*.

[Pinkas, 1991b] Pinkas, G. (1991b). Converting Binary Threshold Networks into Equivalent Symmetric Networks. Technical Report TR 91-31, Department of Computer Science, Washington University, St. Louis, MO 63130.

[Pinkas, 1991c] Pinkas, G. (1991c). Expressing First Order Logic in Symmetric Connectionist Networks. In *IJCAI 91 Workshop on Parallel Processing in AI*, Sydney, Australia.

[Pinkas, 1991d] Pinkas, G. (1991d). Propositional Non-Monotonic Reasoning and Inconsistency in Symmetric Neural Networks. In *International Joint Conference on Artificial Intelligence (IJCAI-91*.

[Pinkas, 1991e] Pinkas, G. (1991e). Symmetric Neural Networks and Propositional Logic Satisfiability. *Neural Computation*, 3(2):282–291.

[Pinkas, 1991f] Pinkas, G. (1991f). Symmetric neural networks and propositional logic satisfiability. *Neural Computation*, 3(2):282–291.

[Pinkas, 1992] Pinkas, G. (1992). Representation and Learning of Propositional Knowledge in Symmetric Connectionist Networks. Technical report, Department of Computer Science, Washington University, St. Louis, MO 63130.

[Pinkas, 1994] Pinkas, G. (1994). *Propositional Logic, Nonmonotonic Reasoning, and Symetric Networks — On Bridging the Gap between Symbolic and Connectionist Knowledge Representation*, chapter 7, pages 175–204. Lawrence Erlbaum Associates, Hillsdale, NJ.

[Pinkas and Dechter, 1992] Pinkas, G. and Dechter, R. (1992). An Improved Connectionist Activation Function for Energy Minimization. Technical report, Department of Computer Science, Washington University, St. Louis, MO 63130.

[Pinkas and Loui, 1992] Pinkas, G. and Loui, R. P. (1992). Reasoning from Conflicting Beliefs: A Taxonomy of Cautious-Bold Inferences. Technical report, Department of Computer Science, Washington University, St. Louis, MO 63130.

[Plate, 1995] Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641.

[Pollack, 1990] Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence*, 46(1-2):77–106.

[Popper and Eccles, 1977] Popper, K. R. and Eccles, J. C. (1977). *Das Ich und sein Gehirn*. Piper Verlag.

[Rohwer, 1993] Rohwer, R. (1993). A Representation of Representation Applied to a Discussion of Variable Binding. Technical report.

[Rose and Belew, 1989] Rose, D. E. and Belew, R. K. (1989). Legal Information Retrieval: A Hybrid Approach. In *International Conference on AI and Law*, pages 138–146, Vancouver, B.C. ACM.

[Russell, 1948] Russell, B. (1948). *Human Knowledge – Its Scope and Limits.* Simon and Schuster.

[Schwefel, 1995] Schwefel, H.-P. (1995). *Evolution and Optimum Seeking.* Wiley.

[Selman and hirst, 1985] Selman, B. and hirst, G. (1985). A rule-based connectionist parsing system. In *Proceedings of 7th Annual Conference of the Cognitive Science Society*, pages 212–221.

[Shastri, 1988] Shastri, L. (1988). A Connectionist Approach to Knowledge Representation and Limited Inference. *Cognitive Science*, 12:331–392.

[Shastri and Ajjanagadde, 1989] Shastri, L. and Ajjanagadde, V. (1989). A Connectionist System for Rule Based Reasoning with Multi-Place Predicates and Variables. Technical report, University of Pennsylvania, Computer and Information Science Department, Philadelphia.

[Shastri and Ajjanagadde, 1990] Shastri, L. and Ajjanagadde, V. (1990). From Simple Associations to Systematic Reasoning: A Connectionist Representation of Rules, Variables and Dynamic Bindings. Technical Report MS-CIS-90-05, Computer And Information Science Department, University of Pennsylvania, Philadelphia, PA 19104.

[Shastri and Ajjanagadde, 1993] Shastri, L. and Ajjanagadde, V. (1993). From Simple Associations to Systematic Reasoning: A Connectionist representation of rules, variables, and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, 16:417–494.

[Shastri and Feldman, 1986] Shastri, L. and Feldman, J. (1986). Neural Nets, Routines, and Semantic Networks. In Sharkey, N., editor, *Advances in Cognitive Science 1*, pages 158–203.

[Shastri and Feldman, 1985] Shastri, L. and Feldman, J. A. (1985). Evidential Reasoning in Semantic Networks: A Formal Theory. In *IJCAI '85*, pages 465–474.

[Shastri and Grannes, 1995] Shastri, L. and Grannes, D. J. (1995). Dealing with negated knowledge and inconsistency in a neurally motivated model of memory and reflexive reasoning. Technical Report TR-95-041, International Computer Science Institute, Berkeley, CA 94704-1105.

[Sims and Bell, 1992] Sims, P. and Bell, D. (1992). Hybrid systems - neural networks in problem-solving environments. In *Second Irish Neural Networks Conference*, Dept. of Information Systems, University of Ulster at Jordanstown.

[Sinclair, 1987] Sinclair, J., editor (1987). *Collins Cobuild English Language Dictionary.* Collins, London, Glasgow.

[Sirat and Nadal, 1990] Sirat, J. A. and Nadal, J.-P. (1990). Neural trees: a new tool for classification. *Network*, 1:423–438.

[Siu et al., 1995] Siu, K.-Y., Roychowdhury, V., and Kailath, T. (1995). *Discrete Neural Computation.* Englewood Cliffs, New Jersey: Prentice Hall.

[Smolensky, 1988] Smolensky, P. (1988). On the Proper Treatment of Connectionism. *Behavioral and Brain Sciences*, 11:1–74.

[Smolensky, 1990] Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46:159–216.

[Soucek and the IRIS Group, 1991] Soucek, B. and the IRIS Group, editors (1991). *Neural and Intelligent Systems Integration.* Wiley.

[Sperduti, 1994a] Sperduti, A. (1994a). Encoding of Labeled Graphs by Labeling RAAM. In Cowan, J. D., Tesauro, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems 6*, pages 1125–1132. San Mateo, CA: Morgan Kaufmann.

[Sperduti, 1994b] Sperduti, A. (1994b). Labeling RAAM. *Connection Science*, 6(4):429–459.

[Sperduti, 1995] Sperduti, A. (1995). Stability properties of labeling recursive auto-associative memory. *IEEE Transactions on Neural Networks*, 6(6):1452–1460.

[Sperduti, 1997] Sperduti, A. (1997). On the computational power of recurrent neural networks for structures. *Neural Networks*, 10(3).

[Sperduti et al., 1996] Sperduti, A., Majidi, D., and Starita, A. (1996). Extended cascade-correlation for syntactic and structural pattern recognition. In Perner, P., Wang, P., and Rosenfeld, A., editors, *Advances in Structural and Syntactical Pattern Recognition*, Lecture notes in Computer Science, pages 90–99. Springer-Verlag, Berlin.

[Sperduti and Starita, 1993] Sperduti, A. and Starita, A. (1993). An example of neural code: Neural trees implemented by LRAAMs. In *International Conference on Neural Networks and Genetic Algorithms*, pages 33–39.

[Sperduti and Starita, 1996] Sperduti, A. and Starita, A. (1996). A general learning framework for the raam family. In Marinaro, M. and Tagliaferri, R., editors, *Neural Nets WIRN Vietri-95*, pages 136–141. World Scientific, Singapore.

[Sperduti et al., 1995] Sperduti, A., Starita, A., and Goller, C. (1995). Learning distributed representations for the classification of terms. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 509–515.

[Sun, 1994] Sun, R. (1994). A neural network model of causality. *IEEE Transaction on Neural Networks*, 5(4):604–611.

[Sun, 1995a] Sun, R. (1995a). A new approach towards modeling causality in commonsense reasoning. *International Journal of Intelligent Systems*, 10(3).

[Sun, 1995b] Sun, R. (1995b). On schemas, logics, and neural assemblies. *Applied Intelligence*, 5(2):83–102.

[Sun, 1996] Sun, R. (1996). Commonsense reasoning with rules, cases, and connectionist models: a paradigmatic comparison. *Fuzzy Sets and Systems.*

[Sun and Bookman, 1994] Sun, R. and Bookman, L., editors (1994). *Computational Architectures Integrating Neural and Symbolic Processes: A Perspective on the State of the Art.* Kluwer.

[Sun et al., 1996] Sun, R., Peterson, T., and Merrill, E. (1996). Bottom-up skill learning in reactive sequential decision tasks. In *18th Cognitive Science Society Conference*, pages 684–690.

[Thatcher, 1973] Thatcher, J. W. (1973). Tree automata: An informal survey. In Aho, A. V., editor, *Currents in the Theory of Computing*. Prentice-Hall, Englewood Cliffs, NJ.

[Touretzky, 1989] Touretzky (1989). Rules and maps in connectionist symbol processing. CMU-CS 89-158, Carnegie Mellon.

[Touretzky,] Touretzky, D. BoltzCONS: dynamic symbol structures in a connectionist network. In Hinton, G. E., editor, *Connectionist Symbol Processing*, Bradford Book, pages 5–46. MIT Press, Cambridge, MA, 1 edition.

[Touretzky and Hinton, 1985] Touretzky, D. and Hinton, G. (1985). Symbols Among the Neurons: Details of a Connectionist Inference Architecture. In *IJCAI '85*, pages 238–243, Pit.

[Touretzky, 1990] Touretzky, D. S. (1990). Boltzcons: Dynamic symbol structures in a connectionist network. *Artificial Intellicence*, 46:5–46.

[Touretzky and Hinton, 1988a] Touretzky, D. S. and Hinton, G. (1988a). A distributed connectionist production system. *Cognitive science*, 12(3):423–466.

[Touretzky and Hinton, 1988b] Touretzky, D. S. and Hinton, G. E. (1988b). A Distributed Connectionist Production System. *Cognitive Science*, 12:423–466.

[Touretzky and Wheeler, 1991] Touretzky, D. S. and Wheeler, D. W. (1991). Sequence manipulation using parallel mapping networks. *Neural Computation*, 3(1):98–109.

[Waltz and Pollack, 1985] Waltz, D. and Pollack, J. (1985). Massively parallel parsing: a strongly interactive model of natural language interpretation. *Cognitive science*, 9:51–74.

[Watrous and Shastri, 1987] Watrous, R. L. and Shastri, L. (1987). Learning Phonetic Features using Connectionist Networks: An Experiment in Speech Recognition. In *IJCNN*, volume 4, pages 381–388, San Diego, CA.

[Welch et al., 1996] Welch, L. R., Yu, G., Ravindran, B., Kurfeß, F., Henriques, J., and Wilson, M. (1996). Reverse engineering of complex navy systems. *International Journal of Software Engineering and Knowledge Engineering*, 6(4).

[Werbos, 1992] Werbos, P. J. (1992). Neurocontrol: Where it is going and why it is crucial. *Artificial Neural Networks*, 2:61–68.

[Williams and Zipser, 1989] Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–280.

[Xiru Zhang and Waltz, 1992] Xiru Zhang, J. P. M. and Waltz, D. L. (1992). Hybrid System for Protein Secondary Structure Prediction. *Journal of Molecular Biology*, 225.

[Zadeh, 1990] Zadeh, L. A. (1990). Fuzzy sets and systems. *International Journal of General Systems*, 17(2-3):129–138.

[Zurada et al., 1994] Zurada, J. M., Marks, R. J., and Robinson, C. J., editors (1994). *Computational Intelligence: Imitating Life*. IEEE Press.