

# Including Control Architecture in Attribute Grammar Specifications of Feedforward Neural Networks

---

Talib S. Hussain and Roger A. Browse  
Queen's University, Kingston, Ontario

# Main Points

---

- ◆ Need for good representations of complex neural networks
- ◆ Generic neural interpreter provides greatest representational flexibility
- ◆ Explicit representation of control architecture permits generic interpreter
- ◆ Attribute grammars generate classes of neural network architectures
- ◆ Hierarchical grammar design permits useful genetic representation of complex neural networks

# Complex Neural Network?

---

- ◆ Heterogeneous
  - multiple types of neural processing and learning behaviors
- ◆ Highly variable topology
  - few limitations upon topological possibilities
- ◆ Scales well
  - network works efficiently for large problems. May require a disposition towards functionally distinct sub-structures

# Representation vs. Interpreter

---

- ◆ Simple representation requires complex interpreter with many assumptions
  - Inherent in direct or parametric encoding
  - Typical of current grammar encodings
- ◆ Complex representation permits manipulation of many network properties
- ◆ Simple interpreter for arbitrary network design is not provided by current neural network research

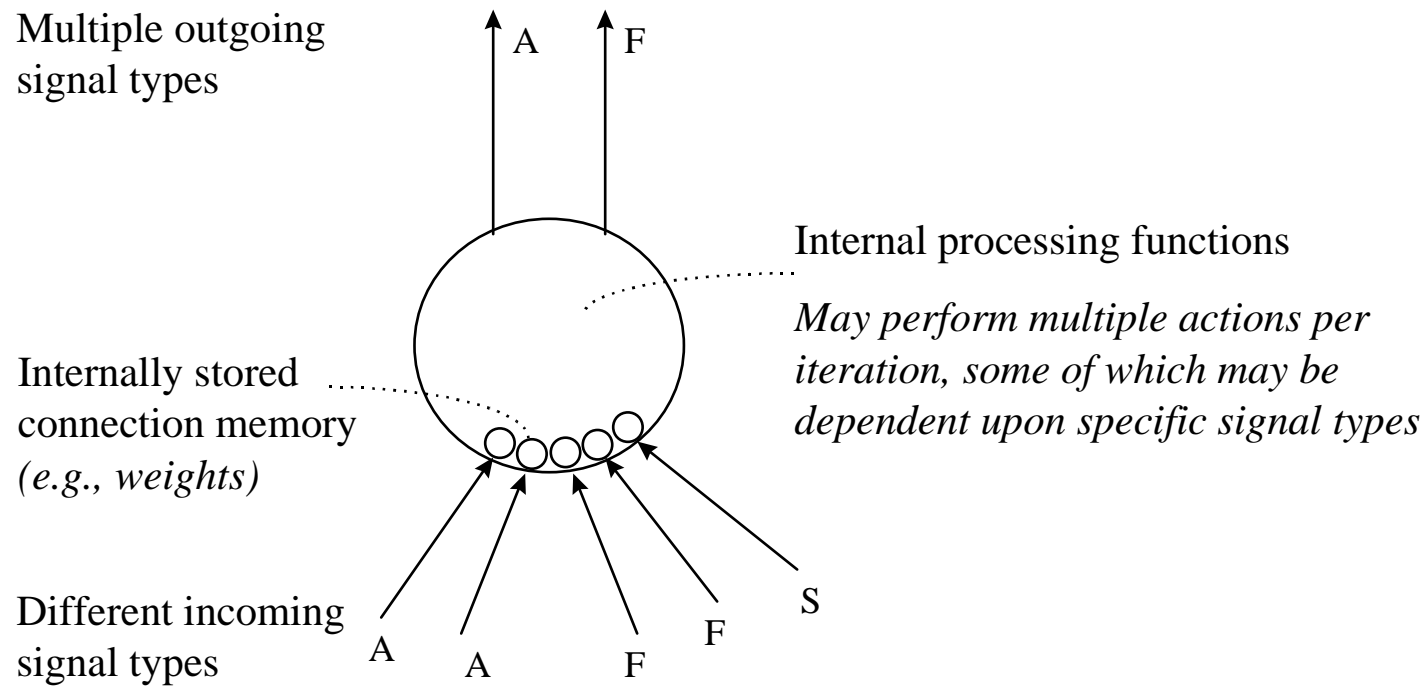
# Network Modeling Approach

---

- ◆ Extended definition of a neuron to include multiple signal types
- ◆ Different signal types permit nodes to implement network control behavior
- ◆ Basic neural interpreter which models only the passing and receipt of signals

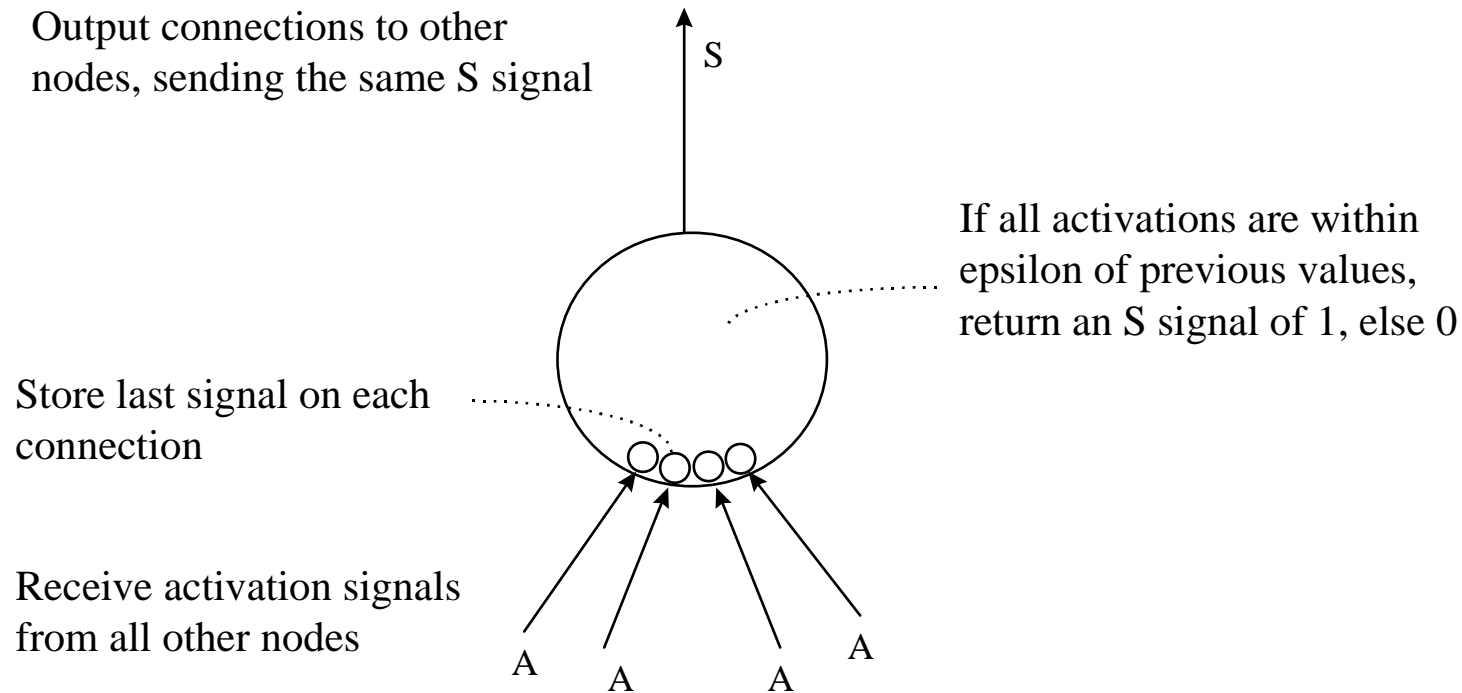
# Basic Neuron Model

---



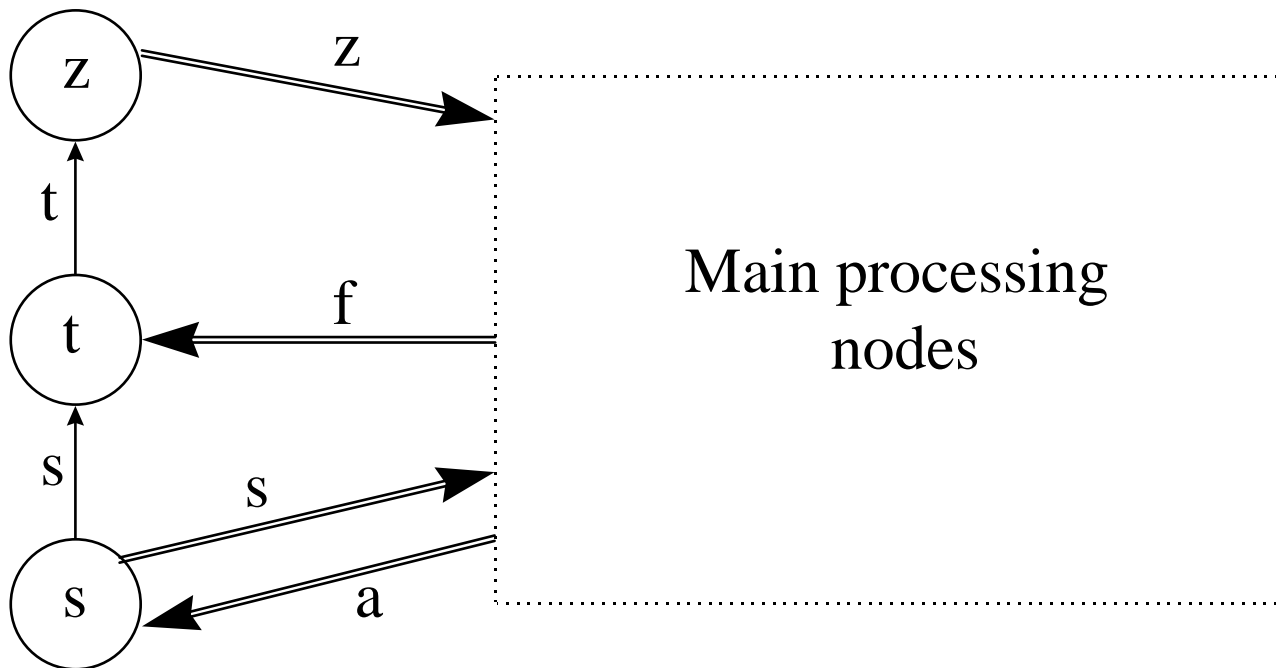
# Control Node: Stable Activation Detector

---



# Back-propagation control

---



# Illustrative CFG Productions

---

Start:  $S \rightarrow N$

Main:  $N \rightarrow C M$

Topology:  $M \rightarrow \text{seq}(M,M) \mid \text{par}(M,M) \mid \text{node}$

Control:  $C \rightarrow s C \mid t C \mid z$

s: stable activation detector

t: stable feedback detector

z: initiate learning node

node: activation+feedback processing node

# Attribute Grammars

---

- ◆ More powerful than context free grammars
- ◆ Still produce nice simple parse trees
- ◆ Hierarchical design permits distinction between functionally different components (e.g., control nodes vs. processing nodes)
- ◆ A complete network representation may be formed in the attributes of the root symbol

# Attribute Grammar Production

---

*Production*

*Context-Free Production*

*Name*

SEQ:  $M_1 \rightarrow M_2 M_3$

*(synthesized)*

**In\_Nodes of  $M_1 :=$  In\_Nodes of  $M_2$ ;**

**Out\_Nodes of  $M_1 :=$  Out\_Nodes of  $M_3$ ;**

**All\_Nodes of  $M_1 :=$  [All\_Nodes of  $M_2 \cup$  All\_Nodes of  $M_3$ ];**

**Connections of  $M_1 :=$  [Connections of  $M_2 \cup$  Connections of  $M_3$**

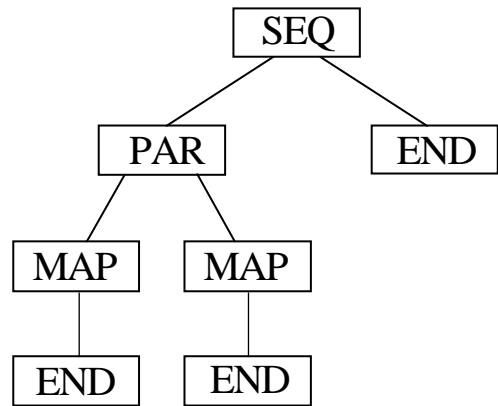
**$\cup$  full\_connect(Out\_Nodes of  $M_2$ , In\_Nodes of  $M_3$ );**

*Attribute*

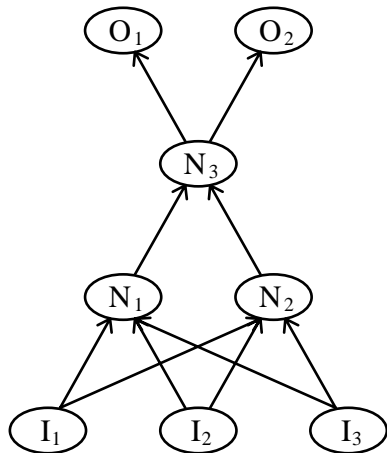
*Evaluation*

*Functions*

# Network Generation Process



*Derivation Tree*

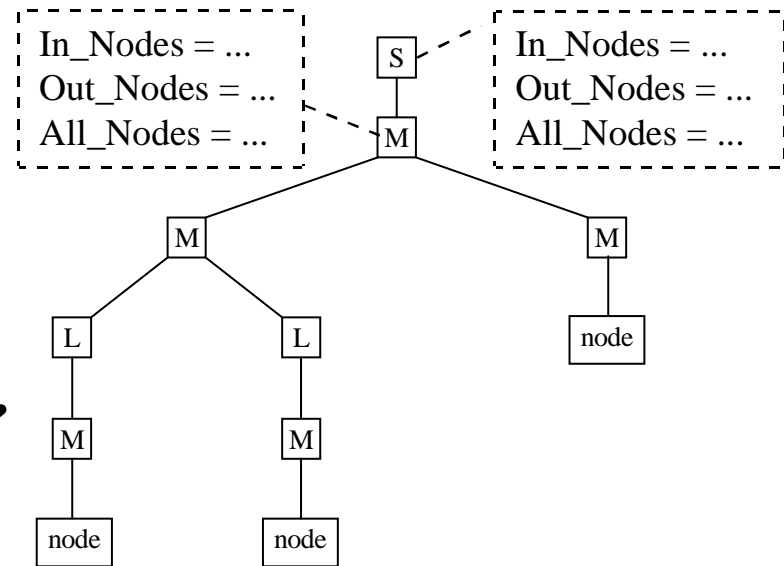


*Neural Network Structure*

→  
Evaluate  
Attributes

↙  
Extract  
Attributes

→  
Interpret  
Structure



*Attributed Parse Tree*

*Executable Neural Network*

# Attribute Grammar Derivation Trees as Genetic Encoding

---

- ◆ Subtrees exhibit strong semantic identity
- ◆ Trees are compact compared to phenotype
- ◆ Grammar design promotes modularity of the phenotype (i.e., modular topology)
- ◆ New productions may enhance compactness
- ◆ Coarse phenotype structure may be represented near root of tree and fine structure near leaves of tree

# “Strongly-Typed” Design

---

- ◆ Each non-terminal symbol in the grammar may be considered as a distinct ‘type’
- ◆ Genetic operators may be designed to preserve symbols, thereby preserving type
- ◆ Ensures legality of complex genotypes
- ◆ Allows logical manipulations of phenotype
- ◆ Permits invariance of certain phenotypic characteristics

# Conclusions

---

- ◆ New modeling approach which permits a simple neural interpreter
- ◆ Attribute grammars may define a class of networks based on this modeling approach
- ◆ Explicit inclusion of control architecture permits manipulation and hybridization of network control behaviour by evolutionary processes