Logic Programs and Connectionist Networks

Steffen Hölldobler
International Center for Computational Logic
Technische Universität Dresden, Germany

- History & Motivation
- The Very Idea
- The Core Method
- Propositional Core Methods
- First-Order Core Methods
- Discussion

Some Historical Remarks

- **McCulloch, Pitts 1943:**
  ***A logical calculus and the ideas immanent in the nervous activity*.**
  - **Networks of binary threshold units are finite automata and vice versa.**
  - **Bader, H., Scalzitti 2004: Weighted automata are semiring artificial neural networks.**
- **Ballard 1986: *Parallel logic inference and energy minimization*.**
  - **Restricted unit resolution and symmetric networks.**
- **Pinkas 1991: *Symmetric neural networks and logic satisfiability*.**
  - **Propositional logic and symmetric networks.**
  - **Strohmaier 1997: Multi-flip networks.**

Historical Remarks – Structured Connectionist Networks

- Shastri, Ajjanagadde 1993: *From Associations to Systematic Reasoning: A Connectionist Representation of Rules, Variables and Dynamic Bindings using Temporal Synchrony*.
  - A limited inference system for reflexive reasoning.
  - Beringer, H. 1993: Reflexive reasoning is reasoning by reduction.
- Stolcke 1989: *Unification as constraint satisfaction in structured connectionist networks*.
  - Unification of feature structures without occurs check.
  - H. 1990: A connectionist unification and matching algorithm.
- H., Kurfess 1992: *CHCL – A connectionist inference system*.
  - Horn clause logic with limited resources based on the connection method.

Motivation

- **Smolensky 1987:**
  Can we find ways of naturally instantiating the power of symbolic computation within fully connectionist systems?
- **McCarthy 1988:**
  Propositional fixation of current connectionist systems.
- **Fodor, Pylyshin 1988:**
  Reasoning is based on structured objects and structure-sensitive processes.

Our Goal: To develop connectionist models for first-order reasoning.

The Very Idea

- Various semantics for logic programs coincide with fixed points of associated immediate consequence operators (e.g., Apt, vanEmden 1982).
- Banach Contraction Mapping Theorem
  - A contraction mapping f defined on a complete metric space (X, d) has a unique fixed point.
  - The sequence $y,\ f(y),\ f(f(y)),\ \ldots$ converges to this fixed point for any $y \in X$.

  Fitting 1994: Consider logic programs, whose immediate consequence operator is a contraction.

- Funahashi 1989: Every continuous function on the reals can be uniformly approximated by feedforward connectionist networks.

  H., Kalinke, Störr 1999: Consider logic programs, whose immediate consequence operator is continuous on the reals.

The Core Method

- Let $\mathcal{L}$ be a logic language.
- Given a logic program $\mathcal{P}$ together with immediate consequence operator $T_{\mathcal{P}}$.
- Let $\mathcal{I}$ be the set of interpretations for $\mathcal{P}$.
- Find a mapping $R : \mathcal{I} \longrightarrow \mathbb{R}^n$.
- Construct a feed-forward network computing $f_{\mathcal{P}} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, called the core, such that the following holds:
  - If $T_{\mathcal{P}}(I) = J$ then $f_{\mathcal{P}}(R(I)) = R(J)$, where $I, J \in \mathcal{I}$.
  - If $f_{\mathcal{P}}(\vec{s}) = \vec{t}$ then $T_{\mathcal{P}}(R^{-1}(\vec{s})) = R^{-1}(\vec{t})$, where $\vec{s}, \vec{t} \in \mathbb{R}^n$.
- Connect the units in the output layer recursively to the units in the input layer.
- Show that the following holds
  - $I = \mathit{lfp}\,(T_{\mathcal{P}})$ iff the recurrent network converges to or approximates $R(I)$.

Connectionist model generation using recurrent networks with feed forward core.

Propositional Core Method using Binary Threshold Units

- Let $\mathcal{L}$ be the language of propositional logic over a set $\mathcal{V}$ of variables.
- Let $\mathcal{P}$ be a propositional logic program, e.g.,

$$\mathcal{P} = \{A,\ C \leftarrow A \wedge \neg B,\ C \leftarrow \neg A \wedge B\}.$$

- $\mathcal{I} = 2^{\mathcal{V}}$ is the set of interpretations for $\mathcal{P}$.
- $T_{\mathcal{P}}(I) = \{A \mid A \leftarrow L_1 \wedge \ldots \wedge L_m \in \mathcal{P}$ such that $I \models L_1 \wedge \ldots \wedge L_m\}$.
- Let $n = |\mathcal{V}|$ and identify $\mathcal{V}$ with $\{1, \ldots, n\}$.
- Define

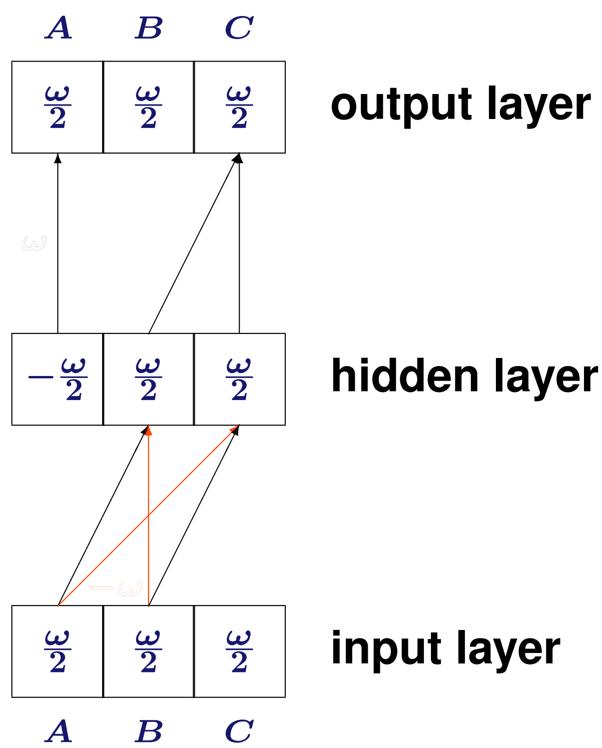
$$R(I)[j] = \begin{cases} 1 & \text{if } j \in I, \\ 0 & \text{if } j \not\in I. \end{cases}$$

  E.g., if $\mathcal{V} = \{A, B, C\} = \{1, 2, 3\}$ and $I = \{A, C\}$ then $R(I) = (1, 0, 1)$.
- Other encodings are possible, e.g.,

$$R(I)[j] = \begin{cases} 1 & \text{if } j \in I, \\ -1 & \text{if } j \not\in I. \end{cases}$$

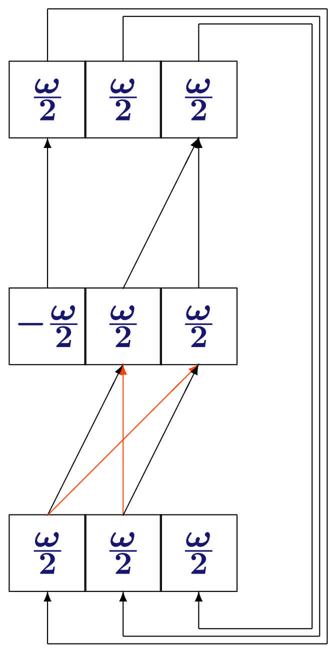
Propositional Core Method – Computing the Core

- Consider again $\mathcal{P} = \{A,\ C \leftarrow A \wedge \neg B,\ C \leftarrow \neg A \wedge B\}$.
- A translation algorithm translates $\mathcal{P}$ into a core of binary threshold units:



Propositional Core Method – Some Results

- **Proposition** 2-layer networks cannot compute $T_{\mathcal{P}}$ for definite $\mathcal{P}$.
- **Theorem** For each program $\mathcal{P}$, there exists a core computing $T_{\mathcal{P}}$.
- Adding recurrent connections:



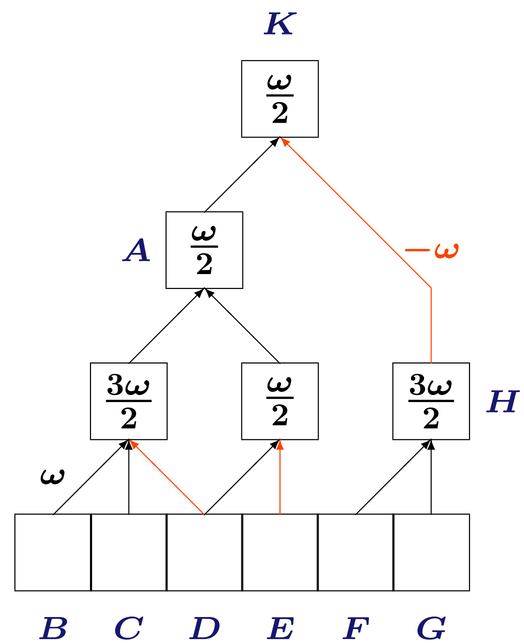
Propositional Core Method – More Results

- A logic programs $\mathcal{P}$ is said to be strongly determined if there exists a metric d on the set of all Herbrand interpretations for $\mathcal{P}$ such that $T_{\mathcal{P}}$ is a contraction wrt d.
- Corollary Let $\mathcal{P}$ be a strongly determined program. Then there exists a core with recurrent connections such that the computation with an arbitrary initial input converges and yields the unique fixed point of $T_{\mathcal{P}}$.
- Let n be the number of clauses and m be the number of propositional variables occurring in $\mathcal{P}$.
  - $2m + n$ units, $2mn$ connections in the core.
  - $T_{\mathcal{P}}(I)$ is computed in 2 steps.
  - The parallel computational model to compute $T_{\mathcal{P}}(I)$ is optimal.
  - The recurrent network settles down in $3n$ steps in the worst case.
- See H., Kalinke 1994 or Hitzler, H., Seda 2004 for details.

Knowledge Based Artificial Neural Networks

- **Towell, Shavlik 1994: Can we do better than empirical learning?**
- **Sets of hierarchical logic programs, e.g.,**

$$\mathcal{P} = \{A \leftarrow B \wedge C \wedge \neg D,\ A \leftarrow D \wedge \neg E,\ H \leftarrow F \wedge G,\ K \leftarrow A, \neg H\}.$$



Propositional Core Method using Sigmoidal Units

- d'Avila Garcez, Zaverucha, Carvalho 1997:
  Can we combine the ideas in Towell, Shavlik 1994 and H., Kalinke 1994?
- Consider propositional logic language.
- Let I be an interpretation and $a \in [0, 1]$.

$$R(I)[j] = \begin{cases} v \in [a, 1] & \text{if } j \in I, \\ w \in [-1, -a] & \text{if } j \notin I. \end{cases}$$

- Translate $\mathcal{P}$ into a core of bipolar sigmoidal units.
- Relation to logic programs is preserved.
- The core is trainable by backprobagation.
- Many interesting applications.
- For more details see d'Avila Garcez, Broda, Gabbay 2002.

Propositional Core Method – Three-Valued Logic

- **Kalinke 1994:** Consider truth values $\top,\ \bot,\ u$.
- Interpretations are pairs $I = \langle I^+, I^- \rangle$.
- Immediate consequence operator $\Phi_{\mathcal{P}}(I) = \langle J^+, J^- \rangle$, where

$$\begin{array}{rcl} J^+ & = & \{A \mid A \leftarrow L_1 \wedge \ldots \wedge L_m \in \mathcal{P} \text{ and } I(L_1 \wedge \ldots \wedge L_m) = \top\}, \\ J^- & = & \{A \mid \text{for all } A \leftarrow L_1 \wedge \ldots \wedge L_m \in \mathcal{P} : \ I(L_1 \wedge \ldots \wedge L_m) = \bot\}. \end{array}$$

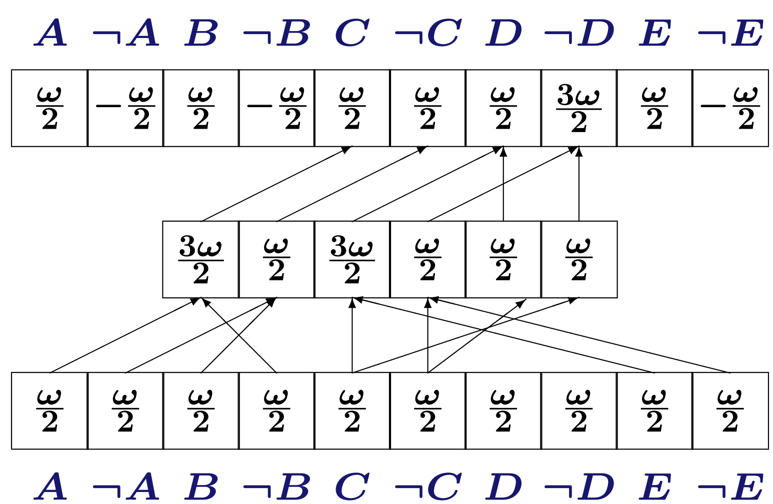
- Let $n = |\mathcal{V}|$ and identify $\mathcal{V}$ with $\{1, \ldots, n\}$.
- Define $R : \mathcal{I} \longrightarrow \mathbb{R}^{2n}$ as follows:

$$R(I)[2j-1] = \left\{ \begin{array}{ll} 1 & \text{if } j \in I^+ \\ 0 & \text{if } j \not\in I^+ \end{array} \right\} \quad \text{and} \quad R(I)[2j] = \left\{ \begin{array}{ll} 1 & \text{if } j \in I^- \\ 0 & \text{if } j \not\in I^- \end{array} \right\}$$

Propositional Core Method – Multi-Valued Logics

- For each program $\mathcal{P}$, there exists a core computing $\Phi_{\mathcal{P}}$, e.g.,

$$\mathcal{P} = \{C \leftarrow A \wedge \neg B,\ D \leftarrow C \wedge E,\ D \leftarrow \neg C\}.$$



- **Lane, Seda 2004:** Extension to finitely determined sets of truth values.

Propositional Core Method – Modal Logic

- Garcez, Lamb, Gabbay 2002.
- Let $\mathcal{L}$ be a propositional logic language plus
  - the modalities $\Box$ and $\Diamond$ and
  - relations between worlds.
- Modal logic programs $\mathcal{P}$.
- Corresponding semantic operator $T_{\mathcal{P}}$.
- Translation algorithm such that $T_{\mathcal{P}}$ is again computed by a core.
- For each world, turn the core into a recurrent network.
- Connect cores with respect to the given set of relations.

First Order Logic Programs

- Given a logic program $\mathcal{P}$ over a first order language $\mathcal{L}$.
- Let *ground* $(\mathcal{P})$ be the set of all ground instances of clauses in $\mathcal{P}$.
- Let $B_{\mathcal{L}}$ be the corresponding Herbrand base.
- $2^{B_{\mathcal{L}}}$ is the set of Herbrand interpretations.
- $T_{\mathcal{P}} : 2^{B_{\mathcal{L}}} \longrightarrow 2^{B_{\mathcal{L}}}$ is defined as

$$T_{\mathcal{P}}(I) = \{A \mid A \leftarrow L_1 \wedge \ldots \wedge L_m \in \textit{ground}(\mathcal{P}) : \; I \models L_1 \wedge \ldots \wedge L_m\}.$$

- $B_{\mathcal{L}}$ is countably infinite.
- The propositional core method is not applicable.

How can the gap between the discrete, symbolic setting of logic, and the continuous, real valued setting of connectionist networks be closed?

First Order Core Method – The Goal

- **Find $R : 2^{B_\mathcal{L}} \longrightarrow \mathbb{R}$ and core computing $f_\mathcal{P} : \mathbb{R} \longrightarrow \mathbb{R}$ such that the following conditions hold.**
  - **If $T_\mathcal{P}(I) = J$ then $f_\mathcal{P}(R(I)) = R(J)$ for all $I, J \in 2^{B_\mathcal{L}}$.**
    If $f_\mathcal{P}(s) = t$ then $T_\mathcal{P}(R^{-1}(s)) = R^{-1}(t)$ for all $s, t \in \mathbb{R}$.
    - $\rightsquigarrow$ **$f_\mathcal{P}$ is a sound and complete encoding of $T_\mathcal{P}$.**
  - **$T_\mathcal{P}$ is a contraction on $2^{B_\mathcal{L}}$ iff $f_\mathcal{P}$ is a contraction on $\mathbb{R}$.**
    - $\rightsquigarrow$ **The contraction property and fixed points are preserved.**
  - **$f_\mathcal{P}$ is continuous on $\mathbb{R}$.**
    - $\rightsquigarrow$ **A connectionst network approximating $f_\mathcal{P}$ is known to exist.**

Acyclic Logic Programs

- Let $\mathcal{P}$ be a program over a first order language $\mathcal{L}$.
- A level mapping for $\mathcal{P}$ is a function $l : B_{\mathcal{L}} \longrightarrow \mathbb{N}$.
  - We define $l(\neg A) = l(A)$.
- We can associate a metric $d_{\mathcal{L}}$ with $\mathcal{L}$ and l. Let $I, J \in 2^{B_{\mathcal{L}}}$:

$$d_{\mathcal{L}}(I, J) = \begin{cases} 0 & \text{if } I = J \\ 2^{-n} & \text{if } n \text{ is the smallest level on which } I \text{ and } J \text{ differ.} \end{cases}$$

- Proposition $(2^{B_{\mathcal{L}}}, d_{\mathcal{L}})$ is a complete metric space Fitting 1994.
- $\mathcal{P}$ is said to be acyclic wrt a level mapping l, if for every $A \leftarrow L_1 \wedge \ldots \wedge L_n \in$ *ground*$(\mathcal{P})$ we find $l(A) > l(L_i)$ for all i.
- Proposition Let $\mathcal{P}$ be an acyclic logic program wrt l and $d_{\mathcal{L}}$ the metric associated with $\mathcal{L}$ and l, then $T_{\mathcal{P}}$ is a contraction on $(2^{B_{\mathcal{L}}}, d_{\mathcal{L}})$.

Mapping Interpretations to Real Numbers

- Let $\mathcal{D} = \{r \in \mathbb{R} \mid r = \sum_{i=1}^{\infty} a_i 4^{-i},$ where $a_i \in \{0, 1\}$ for all $i\}$.
- Let l be a bijective level mapping.
- $\{\top, \bot\}$ can be identified with $\{0, 1\}$.
- The set of all mappings $I : B_{\mathcal{L}} \longrightarrow \{\top, \bot\}$ can be identified with the set of all mappings $f : \mathbb{N} \longrightarrow \{0, 1\}$.
- Let $I_{\mathcal{L}}$ be the set of all mappings from $B_{\mathcal{L}}$ to $\{0, 1\}$.
- Let $R : I_{\mathcal{L}} \longrightarrow \mathcal{D}$ be defined as

$$R(I) = \sum_{i=1}^{\infty} I(l^{-1}(i))4^{-i}.$$

- **Proposition** R is a bijection.

We have a sound and complete encoding of interpretations.

Mapping Immediate Consequence Operators to Functions on the Reals

- We define $f_{\mathcal{P}} : \mathcal{D} \to \mathcal{D} : r \mapsto R(T_{\mathcal{P}}(R^{-1}(r)))$.

$$\begin{tikzcd} I \arrow[r, "T_{\mathcal{P}}"] \arrow[d, "R"'] & I' \arrow[d, "R"] \\ r \arrow[r, "f_{\mathcal{P}}"'] & r' \end{tikzcd}$$

We have a sound and complete encoding of $T_{\mathcal{P}}$.

- **Proposition** Let $\mathcal{P}$ be an acylic program wrt a bijective level mapping. $f_{\mathcal{P}}$ is a contraction on $\mathcal{D}$.

Contraction property and fixed points are preserved.

Approximating Continuous Functions

- **Corollary** $f_{\mathcal{P}}$ is continuous.
- **Theorem Funahashi 1989** Suppose that $\phi : \mathbb{R} \longrightarrow \mathbb{R}$ is non-constant, bounded, monotone increasing and continuous. Let $K \subseteq \mathbb{R}^n$ be compact, let $f : K \longrightarrow \mathbb{R}$ be continuous, and let $\varepsilon > 0$. Then there exists a 3-layer feed forward network with sigmoidal function ϕ for the hidden layer and linear activation function for the input and output layer whose input-output mapping $\overline{f} : K \longrightarrow \mathbb{R}$ satisfies

$$\max_{x \in K} |f(x) - \overline{f}(x)| < \varepsilon.$$

  - Every continuous function $f : K \longrightarrow \mathbb{R}$ can be uniformly approximated by input-output functions of 3-layer feed forward networks.

- **Theorem** $f_{\mathcal{P}}$ can be uniformly approximated by input-output functions of 3-layer feed forward networks.
  - $T_{\mathcal{P}}$ can be approximated as well by applying R^{-1}.

A connectionist network approximating immediate consequence operator exists.

An Example

- Consider $\mathcal{P} = \{q(0),\ q(s(X)) \leftarrow q(X)\}$ and let $l(q(s^n(0))) = n + 1$.
  - $\mathcal{P}$ is acyclic wrt l, l is bijective, $R(B_{\mathcal{L}}) = \frac{1}{3}$.
  - $$\begin{aligned} f_{\mathcal{P}}(R(I)) &= 4^{-l(q(0))} + \sum_{q(X)\in I} 4^{-l(q(s(X)))} \\ &= 4^{-l(q(0))} + \sum_{q(X)\in I} 4^{-(l(q(X)))+1)} = \frac{1+R(I)}{4}. \end{aligned}$$
- Approximation of $f_{\mathcal{P}}$ to accuracy ε yields

$$\tilde{f}(x) \in \left[\frac{1+x}{4} - \varepsilon, \frac{1+x}{4} + \varepsilon\right].$$

- Starting with some x and iterating $\tilde{f}$ yields in the limit a value

$$r \in \left[\frac{1-4\varepsilon}{3}, \frac{1+4\varepsilon}{3}\right].$$

- Applying R^{-1} to r we find

$$q(s^n(0)) \in R^{-1}(r) \text{ if } n < -\log_4 \varepsilon - 1.$$

Approximation of Interpretations

- Let $\mathcal{P}$ be a logic program over a first order language $\mathcal{L}$ and l a level mapping.
- An interpretation I approximates an interpretation J to a degree $n \in \mathbb{N}$ if for all atoms $A \in B_{\mathcal{L}}$ with $l(A) < n$ we find $I(A) = \top$ iff $J(A) = \top$.
  - I approximates J to a degree n iff $d_{\mathcal{L}}(I, J) \leq 2^{-n}$.

Approximation of Supported Models

- Given an acyclic logic program $\mathcal{P}$ with bijective level mapping.
- Let $T_{\mathcal{P}}$ be the immediate consequence operator associated with $\mathcal{P}$ and $M_{\mathcal{P}}$ the least supported model of $\mathcal{P}$.
- We can approximate $T_{\mathcal{P}}$ by a 3-layer feed forward network.
- We can turn this network into a recurrent one.

Does the recurrent network approximate the supported model of P?

- **Theorem** For an arbitrary $m \in \mathbb{N}$ there exists a recursive network with sigmoidal activation function for the hidden layer units and linear activation functions for the input and output layer units computing a function $\tilde{f}_P$ such that there exists an $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$ and for all $x \in [-1, 1]$ we find

$$d_{\mathcal{L}}(R^{-1}(\tilde{f}_P^n(x)), M_{\mathcal{P}}) \leq 2^{-m}.$$

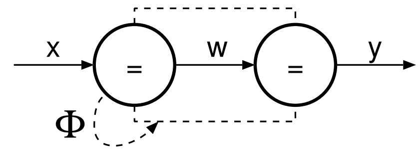
- For more details see H., Kalinke, Störr 1999.

First Order Core Method – Extensions

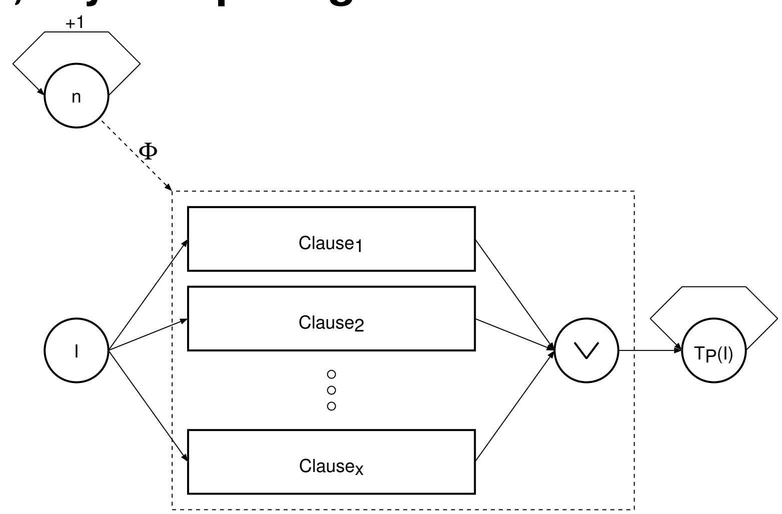
- Detailed study in (topological) continuity of semantic operators Hitzler, Seda 2003 and Hitzler, H., Seda 2004:
  - many-valued logics,
  - larger class of logic programs,
  - other approximation theorems.
- A core method for reflexive reasoning H., Kalinke, Wunderlich 2000.
- The graph of $f_{\mathcal{P}}$ is an attractor of some iterated function system Bader 2003 and Bader, Hitzler 2004:
  - representation theorems,
  - fractal interpolation,
  - core with units computing radial basis functions.
- Finitely determined sets of truth values Lane, Seda 2004.

Constructive Approaches: Fibring Artificial Neural Networks

- **Fibring function** Φ **associated with neuron** i **maps some weights** w **of a network to new values depending on** w **and the input** x **of** i **Garcez, Gabbay 2004.**



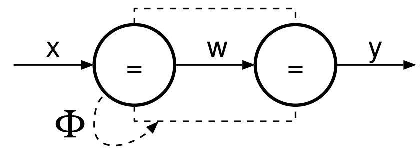
- **Idea approximate** $f_{\mathcal{P}}$ **by computing values of atoms with level** $n = 1, 2, \ldots$**.**



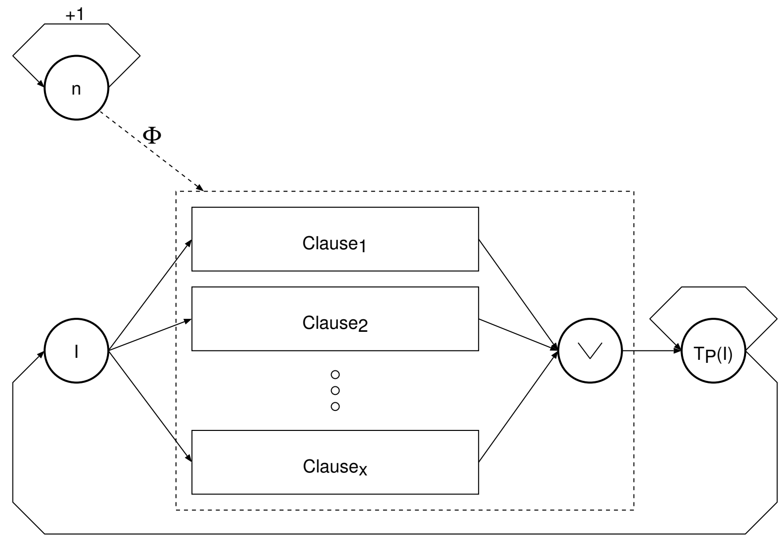
Works well for acyclic logic programs with bijective level mapping Bader, Garcez, Hitzler 2004.

Constructive Approaches: Fibring Artificial Neural Networks

- **Fibring function** Φ **associated with neuron** i **maps some weights** w **of a network to new values depending on** w **and the input** x **of** i **Garcez, Gabbay:2004.**



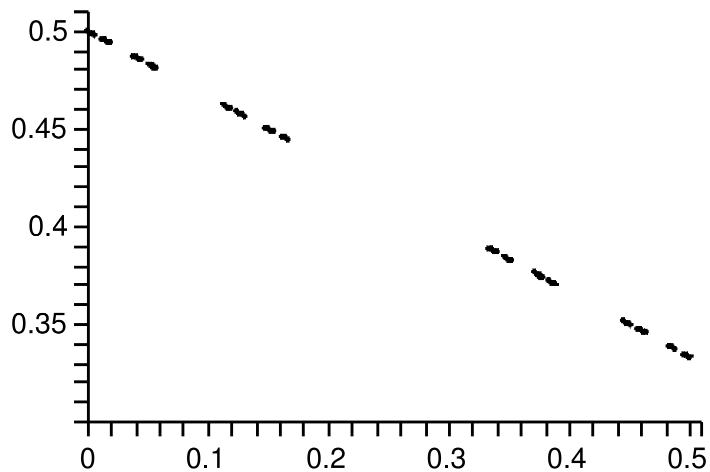
- **Idea approximate** $f_{\mathcal{P}}$ **by computing values of atoms with level** $n = 1, 2, \ldots$**.**



- **Works well for acyclic logic programs with bijective level mapping Bader, Garcez, Hitzler 2004.**

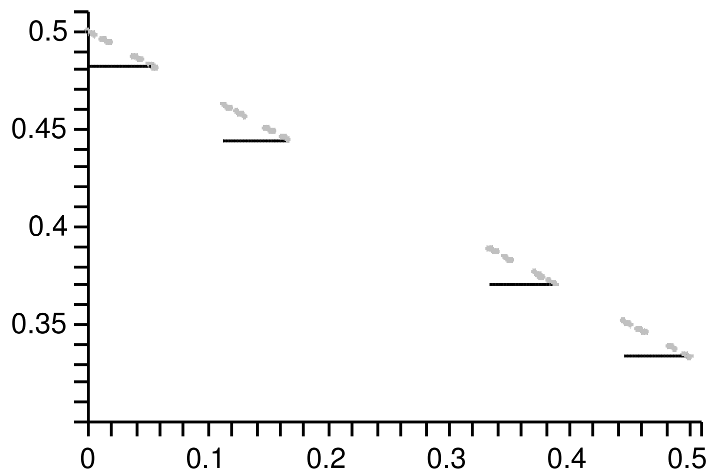
Constructive Approaches: Approximating Piecewise Constant Functions

- **Consider graph of $f_{\mathcal{P}}$.**
- Approximate $f_{\mathcal{P}}$ up to a given level l.
- Construct network computing piecewise constant function.
  - Step activation functions.
  - Sigmoidal activation functions.
  - Radial basis functions.



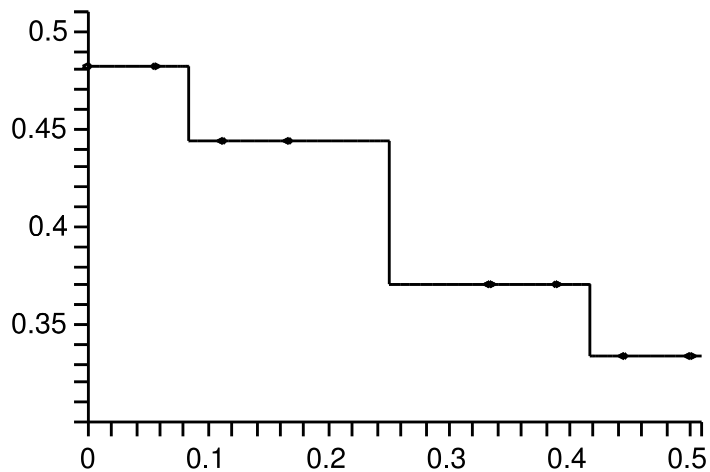
Constructive Approaches: Approximating Piecewise Constant Functions

- **Consider graph of $f_{\mathcal{P}}$.**
- **Approximate $f_{\mathcal{P}}$ up to a given level l.**
- Construct core computing piecewise constant function.
  - Step activation functions.
  - Sigmoidal activation functions.
  - Radial basis functions.



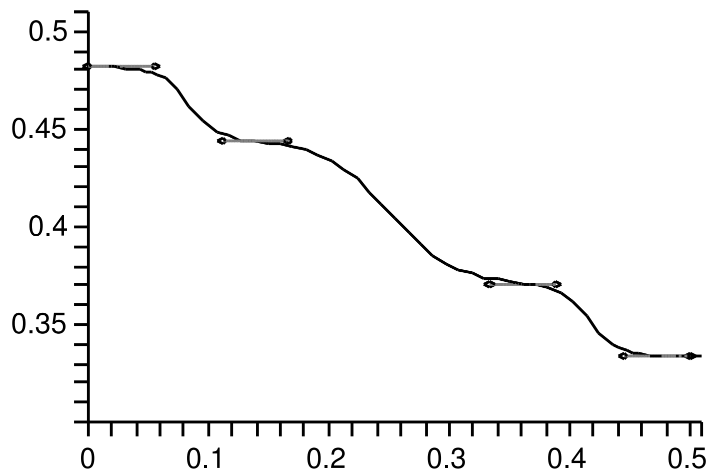
Constructive Approaches: Approximating Piecewise Constant Functions

- Consider graph of $f_{\mathcal{P}}$.
- Approximate $f_{\mathcal{P}}$ up to a given level l.
- Construct core computing piecewise constant function.
  - Step activation functions.
  - Sigmoidal activation functions.
  - Radial basis functions.



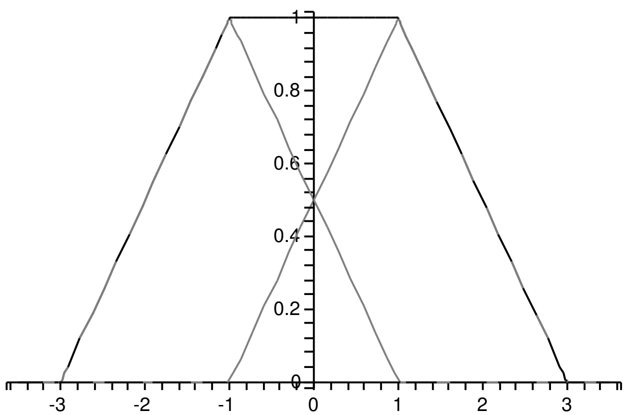
Constructive Approaches: Approximating Piecewise Constant Functions

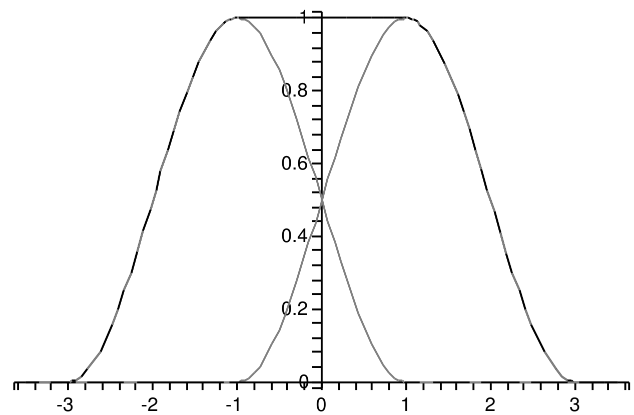
- Consider graph of $f_{\mathcal{P}}$.
- Approximate $f_{\mathcal{P}}$ up to a given level l.
- Construct core computing piecewise constant function.
  - Step activation functions.
  - Sigmoidal activation functions.
  - Radial basis functions.



Constructive Approaches: Approximating Piecewise Constant Functions

- Consider graph of $f_{\mathcal{P}}$.
- Approximate $f_{\mathcal{P}}$ up to a given level l.
- Construct core computing piecewise constant function.
  - Step activation functions.
  - Sigmoidal activation functions.
  - Radial basis functions.





- Bader, Hitzler, Witzel 2005.

Open Problems

- How can first order terms be represented and manipulated in a connectionist system? Pollack 1990, H. 1990, Plate 1994.
- Can the mapping R be learned? Gust, Kühnberger 2004.
- How can first order rules be extracted from a connectionist system?
- How can multiple instances of first order rules be represented in a connectionist system? Shastri 1990.
- What does a theory for the integration of logic and connectionist systems look like?
- Can such a theory be applied in real domains outperforming conventional approaches? Witzel 2005.
- How does the core method relate to model-based reasoning approaches in cognitive science (e.g. Barnden 1989, Johnson-Laird, Byrne 1993)?

Many thanks to Yvonne McIntyre (Kalinke), Hans-Peter Störr,

Jörg Wunderlich, Pascal Hitzler, Sebastian Bader

Antje Strohmaier (Beringer) and Andreas Witzel.