First-order Deduction in Neural Networks

Ekaterina Komendantskaya

Department of Mathematics, University College Cork, Ireland

International Conference on Language and Automata Theory and Applications,
29 June – 4 April, 2007

Outline

1. Motivation
   - Neuro-Symbolic Integration
   - Connectionist Neural Networks and Logic Programs

Outline

1. Motivation
   - Neuro-Symbolic Integration
   - Connectionist Neural Networks and Logic Programs
2. SLD-resolution

Outline

1. Motivation
   - Neuro-Symbolic Integration
   - Connectionist Neural Networks and Logic Programs
2. SLD-resolution
3. First-Order Deduction in Neural networks

Outline

Motivation

Symbolic Logic as Deductive System

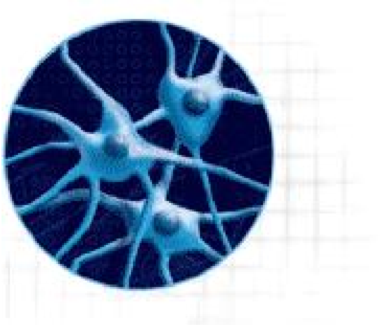
1. Axioms: $(A \supset (B \supset A))$; $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$; $(((\neg B) \supset (\neg A)) \supset (((\neg B) \supset A) \supset B))$; $((\forall x A) \supset S^x_t A)$; $\forall x (A \supset B)) \supset (A \supset \forall x B)))$;
2. Rules: $\dfrac{A \supset B, \; A}{B}$; $\dfrac{A}{\forall x A}$.

Motivation

Symbolic Logic as Deductive System

1. Axioms: $(A \supset (B \supset A))$; $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$; $(((\neg B) \supset (\neg A)) \supset (((\neg B) \supset A) \supset B))$; $((\forall x A) \supset S^x_t A)$; $\forall x(A \supset B)) \supset (A \supset \forall x B)))$;
2. Rules: $\dfrac{A \supset B,\ A}{B}$; $\dfrac{A}{\forall x A}$.

Neural Networks



- spontaneous behavior;
- learning and adaptation

Motivation

Logic Programs

- $A \leftarrow B_1, \ldots, B_n$

Motivation

Logic Programs

- $A \leftarrow B_1, \ldots, B_n$
- $T_P(I) = \{A \in B_P :$
  $A \leftarrow B_1, \ldots, B_n$
  is a ground instance of a clause in P and
  $\{B_1, \ldots, B_n\} \subseteq I\}$

Motivation

Logic Programs

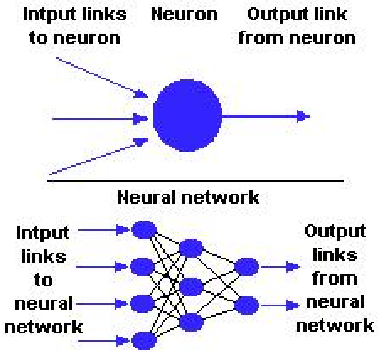
- $A \leftarrow B_1, \ldots, B_n$
- $T_P(I) = \{A \in B_P :$ $A \leftarrow B_1, \ldots, B_n$ is a ground instance of a clause in P and $\{B_1, \ldots, B_n\} \subseteq I\}$
- $\mathtt{lfp}(T_P \uparrow \omega) =$ the least Herbrand model of P.

Motivation

Logic Programs

- $A \leftarrow B_1, \ldots, B_n$
- $T_P(I) = \{A \in B_P :$
  $A \leftarrow B_1, \ldots, B_n$
  is a ground instance of a clause in P and
  $\{B_1, \ldots, B_n\} \subseteq I\}$
- $\mathtt{lfp}(T_P \uparrow \omega) =$ the least Herbrand model of P.

Artificial Neural Networks



An Important Result, [Kalinke, Hölldobler, 94]

Theorem

For each propositional program P, there exists a 3-layer feedforward neural network which computes T_P.

- No learning or adaptation;
- Require infinitely long layers in the first-order case.

A Simple Example

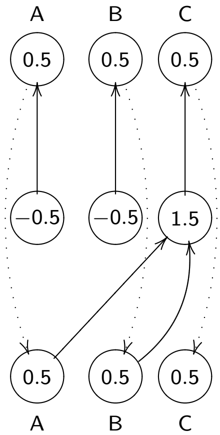
$B \leftarrow$
$A \leftarrow$
$C \leftarrow A, B$

$T_P \uparrow 0 = \{B, A\}$
$lfp(T_P) = T_P \uparrow 1 = \{B, A, C\}$

A Simple Example

$B \leftarrow$
$A \leftarrow$
$C \leftarrow A, B$

$T_P \uparrow 0 = \{B, A\}$
$lfp(T_P) = T_P \uparrow 1 = \{B, A, C\}$



A Simple Example

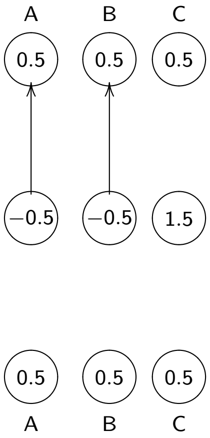
$B \leftarrow$

$A \leftarrow$

$C \leftarrow A, B$

$T_P \uparrow 0 = \{B, A\}$

$lfp(T_P) = T_P \uparrow 1 = \{B, A, C\}$



A Simple Example

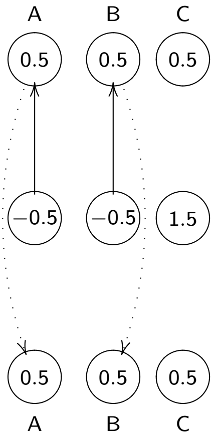
$B \leftarrow$

$A \leftarrow$

$C \leftarrow A, B$

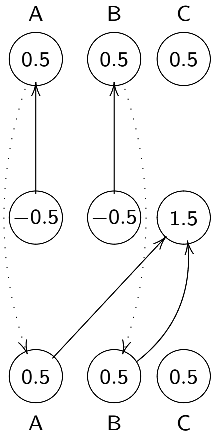
$T_P \uparrow 0 = \{B, A\}$

$lfp(T_P) = T_P \uparrow 1 = \{B, A, C\}$



A Simple Example

$B \leftarrow$
$A \leftarrow$
$C \leftarrow A, B$
$T_P \uparrow 0 = \{B, A\}$
$lfp(T_P) = T_P \uparrow 1 = \{B, A, C\}$



A Simple Example

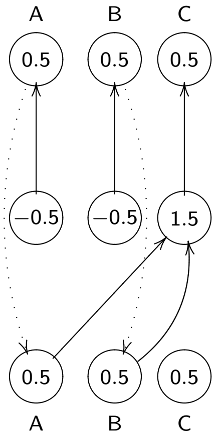
$B \leftarrow$

$A \leftarrow$

$C \leftarrow A, B$

$T_P \uparrow 0 = \{B, A\}$

$lfp(T_P) = T_P \uparrow 1 = \{B, A, C\}$



Most General Unifier

MGU

Let S be a finite set of atoms. A substitution θ is called a unifier for S if S is a singleton. A unifier θ for S is called a *most general unifier* (mgu) for S if, for each unifier σ of S, there exists a substitution γ such that $\sigma = \theta\gamma$.

Example: If $S = (Q(f(x_1, x_2)), Q(f(a_1, a_2)))$, then $\theta = \{x_1/a_1;\ x_2/a_2\}$ is the mgu.

Disagreement set

Disagreement set

To find the *disagreement set* D_S of S locate the leftmost symbol position at which not all atoms in S have the same symbol and extract from each atom in S the term beginning at that symbol position. The set of all such terms is the disagreement set.

Example: For $S = (Q(f(x_1, x_2)), Q(f(a_1, a_2)))$ we have $D_S = \{x_1,\ a_1\}$.

Unification algorithm

1. Put $k = 0$ and $\sigma_0 = \varepsilon$.
2. If $S\sigma_k$ is a singleton, then stop; σ_k is an mgu of S. Otherwise, find the disagreement set D_k of $S\sigma_k$.
3. If there exist a variable v and a term t in D_k such that v does not occur in t, then put $\theta_{k+1} = \theta_k\{v/t\}$, increment k and go to 2. Otherwise, stop; S is not unifiable.

Unification algorithm

1. Put $k = 0$ and $\sigma_0 = \varepsilon$.
2. If $S\sigma_k$ is a singleton, then stop; σ_k is an mgu of S. Otherwise, find the disagreement set D_k of $S\sigma_k$.
3. If there exist a variable v and a term t in D_k such that v does not occur in t, then put $\theta_{k+1} = \theta_k\{v/t\}$, increment k and go to 2. Otherwise, stop; S is not unifiable.

Unification theorem.

SLD-resolution - Example

$$
\begin{aligned}
Q_1(f(x_1, x_2)) &\leftarrow Q_2(x_1), Q_3(x_2)\\
Q_1(f(x_1, x_2)) &\leftarrow Q_4(x_1)\\
Q_2(a_1) &\leftarrow\\
Q_3(a_2) &\leftarrow
\end{aligned}
$$

SLD-resolution - Example

$$
\begin{aligned}
{\color{red}Q_1(f(x_1,x_2))} &\leftarrow Q_2(x_1), Q_3(x_2)\\
Q_1(f(x_1,x_2)) &\leftarrow Q_4(x_1)\\
Q_2(a_1) &\leftarrow\\
Q_3(a_2) &\leftarrow
\end{aligned}
$$

- $G_0 = \leftarrow Q_1(f_1(a_1, a_2))$.

SLD-resolution - Example

$$
\begin{aligned}
Q_1(f(x_1, x_2)) &\leftarrow Q_2(x_1), Q_3(x_2) \\
Q_1(f(x_1, x_2)) &\leftarrow Q_4(x_1) \\
Q_2(a_1) &\leftarrow \\
Q_3(a_2) &\leftarrow
\end{aligned}
$$

- $G_0 = \leftarrow Q_1(f_1(a_1, a_2))$. $S = \{Q_1(f_1(a_1, a_2)), Q_1(f_1(x_1, x_2))\}$.

SLD-resolution - Example

$$
\begin{aligned}
Q_1(f(x_1, x_2)) &\leftarrow Q_2(x_1), Q_3(x_2) \\
Q_1(f(x_1, x_2)) &\leftarrow Q_4(x_1) \\
Q_2(a_1) &\leftarrow \\
Q_3(a_2) &\leftarrow
\end{aligned}
$$

- $G_0 = \leftarrow Q_1(f_1(a_1, a_2))$. $S = \{Q_1(f_1(a_1, a_2)), Q_1(f_1(x_1, x_2))\}$. $D_S = \{x_1, a_1\}$. Put $\theta_1 = x_1/a_1$.

SLD-resolution - Example

$$
\begin{aligned}
Q_1(f(x_1,x_2)) &\leftarrow Q_2(x_1), Q_3(x_2)\\
Q_1(f(x_1,x_2)) &\leftarrow Q_4(x_1)\\
Q_2(a_1) &\leftarrow\\
Q_3(a_2) &\leftarrow
\end{aligned}
$$

- $G_0 = \leftarrow Q_1(f_1(a_1,a_2))$. $S = \{Q_1(f_1(a_1,a_2)), Q_1(f_1(x_1,x_2))\}$.
  $D_S = \{x_1, a_1\}$. Put $\theta_1 = x_1/a_1$.
  $S\theta_1 = \{Q_1(f_1(a_1,a_2)), Q_1(f_1(a_1,x_2))\}$.

SLD-resolution - Example

$$\begin{aligned} Q_1(f(x_1,x_2)) &\leftarrow Q_2(x_1), Q_3(x_2) \\ Q_1(f(x_1,x_2)) &\leftarrow Q_4(x_1) \\ Q_2(a_1) &\leftarrow \\ Q_3(a_2) &\leftarrow \end{aligned}$$

- $G_0 = \leftarrow Q_1(f_1(a_1,a_2))$. $S = \{Q_1(f_1(a_1,a_2)), Q_1(f_1(x_1,x_2))\}$. $D_S = \{x_1, a_1\}$. Put $\theta_1 = x_1/a_1$. $S\theta_1 = \{Q_1(f_1(a_1,a_2)), Q_1(f_1(a_1,x_2))\}$. $D_{S\theta} = \{x_2, a_2\}$ and $\theta_2 = x_2/a_2$. $S\theta_1\theta_2$ is a singleton.

SLD-resolution - Example

$$\begin{aligned} Q_1(f(x_1,x_2)) &\leftarrow Q_2(x_1), Q_3(x_2) \\ Q_1(f(x_1,x_2)) &\leftarrow Q_4(x_1) \\ Q_2(a_1) &\leftarrow \\ Q_3(a_2) &\leftarrow \end{aligned}$$

- $G_0 = \leftarrow Q_1(f_1(a_1,a_2))$. $S = \{Q_1(f_1(a_1,a_2)), Q_1(f_1(x_1,x_2))\}$. $D_S = \{x_1, a_1\}$. Put $\theta_1 = x_1/a_1$. $S\theta_1 = \{Q_1(f_1(a_1,a_2)), Q_1(f_1(a_1,x_2))\}$. $D_{S\theta} = \{x_2, a_2\}$ and $\theta_2 = x_2/a_2$. $S\theta_1\theta_2$ is a singleton.
- $G_1 = \leftarrow Q_2(a_1), Q_3(a_2)$.

SLD-resolution - Example

$$
\begin{aligned}
Q_1(f(x_1, x_2)) &\leftarrow Q_2(x_1), Q_3(x_2)\\
Q_1(f(x_1, x_2)) &\leftarrow Q_4(x_1)\\
Q_2(a_1) &\leftarrow\\
Q_3(a_2) &\leftarrow
\end{aligned}
$$

- $G_0 = \leftarrow Q_1(f_1(a_1, a_2))$. $S = \{Q_1(f_1(a_1, a_2)), Q_1(f_1(x_1, x_2))\}$. $D_S = \{x_1, a_1\}$. Put $\theta_1 = x_1/a_1$. $S\theta_1 = \{Q_1(f_1(a_1, a_2)), Q_1(f_1(a_1, x_2))\}$. $D_{S\theta} = \{x_2, a_2\}$ and $\theta_2 = x_2/a_2$. $S\theta_1\theta_2$ is a singleton.
- $G_1 = \leftarrow Q_2(a_1), Q_3(a_2)$.

SLD-resolution - Example

$$
\begin{aligned}
Q_1(f(x_1,x_2)) &\leftarrow Q_2(x_1), Q_3(x_2)\\
Q_1(f(x_1,x_2)) &\leftarrow Q_4(x_1)\\
Q_2(a_1) &\leftarrow\\
Q_3(a_2) &\leftarrow
\end{aligned}
$$

- $G_0 = \leftarrow Q_1(f_1(a_1,a_2))$. $S = \{Q_1(f_1(a_1,a_2)), Q_1(f_1(x_1,x_2))\}$. $D_S = \{x_1, a_1\}$. Put $\theta_1 = x_1/a_1$. $S\theta_1 = \{Q_1(f_1(a_1,a_2)), Q_1(f_1(a_1,x_2))\}$. $D_{S\theta} = \{x_2, a_2\}$ and $\theta_2 = x_2/a_2$. $S\theta_1\theta_2$ is a singleton.
- $G_1 = \leftarrow Q_2(a_1), Q_3(a_2)$.
- $G_2 = \leftarrow Q_3(a_2)$

SLD-resolution - Example

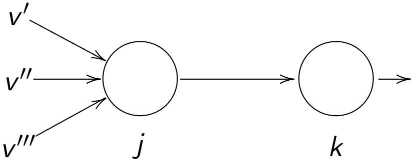
$$\begin{aligned} Q_1(f(x_1,x_2)) &\leftarrow Q_2(x_1), Q_3(x_2) \\ Q_1(f(x_1,x_2)) &\leftarrow Q_4(x_1) \\ Q_2(a_1) &\leftarrow \\ Q_3(a_2) &\leftarrow \end{aligned}$$

- $G_0 = \leftarrow Q_1(f_1(a_1,a_2))$. $S = \{Q_1(f_1(a_1,a_2)), Q_1(f_1(x_1,x_2))\}$. $D_S = \{x_1, a_1\}$. Put $\theta_1 = x_1/a_1$. $S\theta_1 = \{Q_1(f_1(a_1,a_2)), Q_1(f_1(a_1,x_2))\}$. $D_{S\theta} = \{x_2, a_2\}$ and $\theta_2 = x_2/a_2$. $S\theta_1\theta_2$ is a singleton.
- $G_1 = \leftarrow Q_2(a_1), Q_3(a_2)$.
- $G_2 = \leftarrow Q_3(a_2)$
- $G_3 = \square$.

Connectionist Neural Networks

$$p_k(t) = \left(\sum_{j=1}^{n_k} w_{kj} v_j(t)\right) - \Theta_k$$

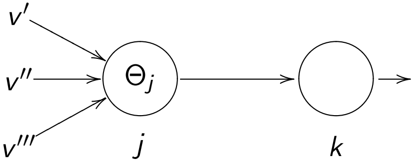
$$v_k(t + \Delta t) = \psi(p_k(t)) = \begin{cases} 1 & \texttt{if } p_k(t) > 0 \\ 0 & \texttt{otherwise.} \end{cases}$$



Connectionist Neural Networks

$$p_k(t) = \left(\sum_{j=1}^{n_k} w_{kj} v_j(t)\right) - \Theta_k$$

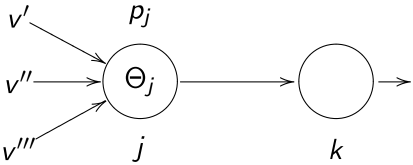
$$v_k(t + \Delta t) = \psi(p_k(t)) = \begin{cases} 1 & \texttt{if } p_k(t) > 0 \\ 0 & \texttt{otherwise.} \end{cases}$$



Connectionist Neural Networks

$$p_k(t) = \left(\sum_{j=1}^{n_k} w_{kj} v_j(t)\right) - \Theta_k$$

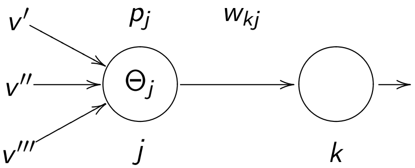
$$v_k(t+\Delta t) = \psi(p_k(t)) = \begin{cases} 1 & \texttt{if } p_k(t) > 0 \\ 0 & \texttt{otherwise.} \end{cases}$$



Connectionist Neural Networks

$$p_k(t) = \left(\sum_{j=1}^{n_k} w_{kj} v_j(t)\right) - \Theta_k$$

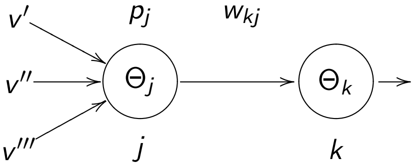
$$v_k(t + \Delta t) = \psi(p_k(t)) = \begin{cases} 1 & \texttt{if } p_k(t) > 0 \\ 0 & \texttt{otherwise.} \end{cases}$$



Connectionist Neural Networks

$$p_k(t) = \left(\sum_{j=1}^{n_k} w_{kj} v_j(t)\right) - \Theta_k$$

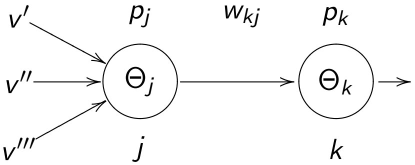
$$v_k(t + \Delta t) = \psi(p_k(t)) = \begin{cases} 1 & \texttt{if } \; p_k(t) > 0 \\ 0 & \texttt{otherwise.} \end{cases}$$



Connectionist Neural Networks

$$p_k(t) = \left(\sum_{j=1}^{n_k} w_{kj} v_j(t)\right) - \Theta_k$$

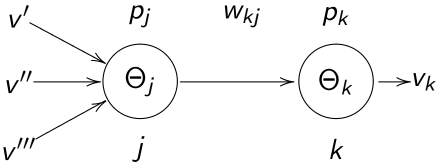
$$v_k(t + \Delta t) = \psi(p_k(t)) = \begin{cases} 1 & \texttt{if } p_k(t) > 0 \\ 0 & \texttt{otherwise.} \end{cases}$$



Connectionist Neural Networks

$$p_k(t) = \left(\sum_{j=1}^{n_k} w_{kj} v_j(t)\right) - \Theta_k$$

$$v_k(t + \Delta t) = \psi(p_k(t)) = \begin{cases} 1 & \texttt{if } p_k(t) > 0 \\ 0 & \texttt{otherwise.} \end{cases}$$



Gödel Numbers of Formulae

Each symbol of the first-order language receives a **Gödel number** as follows:

- variables $x_1, x_2, x_3, \ldots$ receive numbers $(01), (011), (0111), \ldots$;
- constants $a_1, a_2, a_3, \ldots$ receive numbers $(21), (211), (2111), \ldots$;
- function symbols $f_1, f_2, f_3, \ldots$ receive numbers $(31), (311), (3111), \ldots$;
- predicate symbols $Q_1, Q_2, Q_3, \ldots$ receive numbers $(41), (411), (4111), \ldots$;
- symbols (,) and , receive numbers 5, 6 and 7 respectively.

Operations on Gödel Numbers

- **Disagreement set**: $g_1 \ominus g_2$;

Operations on Gödel Numbers

- **Disagreement set**: $g_1 \ominus g_2$;
- **Concatenation**: $g_1 \oplus g_2 = g_1 \mathbf{8} g_2$;

Operations on Gödel Numbers

- **Disagreement set**: $g_1 \ominus g_2$;
- **Concatenation**: $g_1 \oplus g_2 = g_1 \mathbf{8} g_2$;
- **Gödel number of substitution**: $s = g_1 \mathbf{9} g_2$;

Operations on Gödel Numbers

- **Disagreement set**: $g_1 \ominus g_2$;
- **Concatenation**: $g_1 \oplus g_2 = g_1 \mathbf{8} g_2$;
- **Gödel number of substitution**: $s = g_1 \mathbf{9} g_2$;
- **Substitution**: $g \odot s$;

Operations on Gödel Numbers

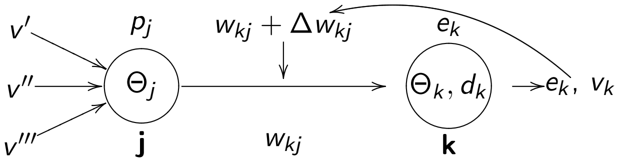
- **Disagreement set**: $g_1 \ominus g_2$;
- **Concatenation**: $g_1 \oplus g_2 = g_1 \mathbf{8} g_2$;
- **Gödel number of substitution**: $s = g_1 \mathbf{9} g_2$;
- **Substitution**: $g \odot s$;
- **Algorithm of unification**.

Unification in Neural Networks

Claim 1

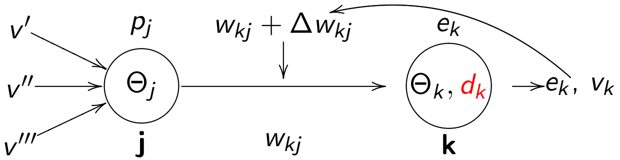
Unification Algorithm can be performed in finite (and very small) neural networks with error-correction learning.

Error-Correction (Supervised) Learning



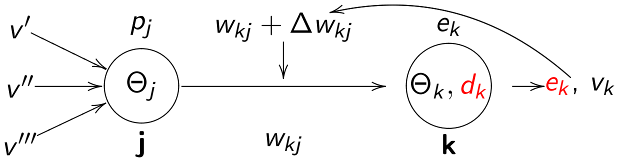
Error-Correction (Supervised) Learning

We embed a new parameter, **desired response** d_k into neurons;



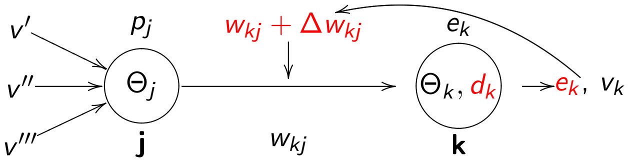
Error-Correction (Supervised) Learning

We embed a new parameter, **desired response** d_k into neurons;
Error-signal: $e_k(t) = d_k(t) - v_k(t)$;



Error-Correction (Supervised) Learning

We embed a new parameter, **desired response** d_k into neurons;
Error-signal: $e_k(t) = d_k(t) - v_k(t)$;
Error-correction learning rule: $\Delta w_{kj}(t) = \eta e_k(t) v_j(t)$.

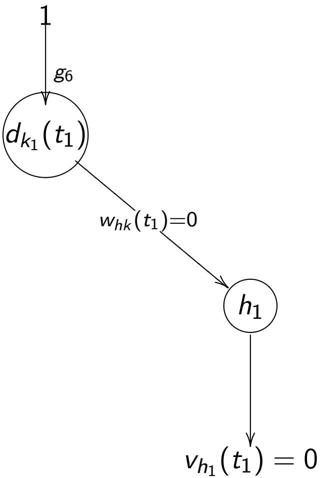


Main Lemma

Lemma

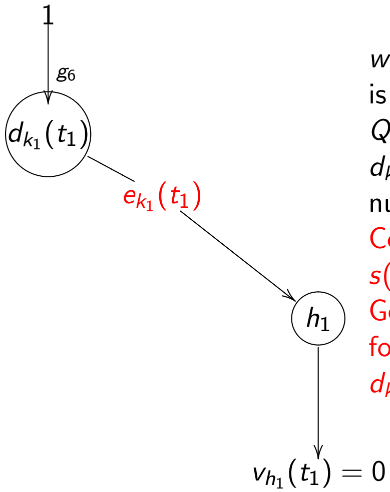
Given two first-order atoms A and B, there exists a two-neuron learning neural network that performs the algorithm of unification for A and B.

Example of Unification in Neural Networks: time $= t_1$.



$w_{ik}(t_1) = v_i(t_1) = g_6$ is the Gödel number of $Q_1(f(a_1, a_2))$;
$d_k(t_1) = g_1$ is the Gödel number of $Q_1(f(x_1, x_2))$.

Example of Unification in Neural Networks: time $= t_1$.

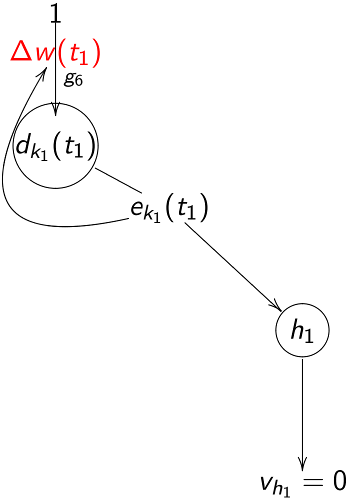


$w_{ki}(t_1) = v_i(t_1) = g_6$ is the Gödel number of $Q_1(f(a_1, a_2))$;

$d_k(t_1) = g_1$ is the Gödel number of $Q_1(f(x_1, x_2))$;

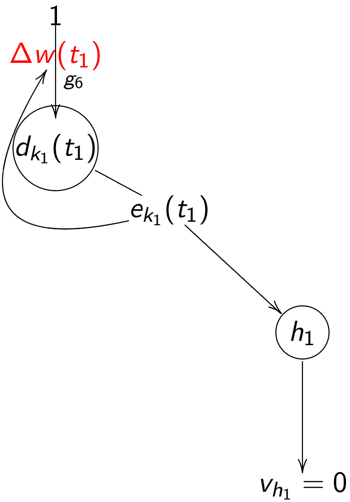
Compute $e_k(t_1) = s(d_k(t_1) \ominus v_k(t_1))$ - the Gödel number of substitution for the disagreement set $d_k(t_1) \ominus v_k(t_1)$.

Example of Unification in Neural Networks: time $= t_1$.



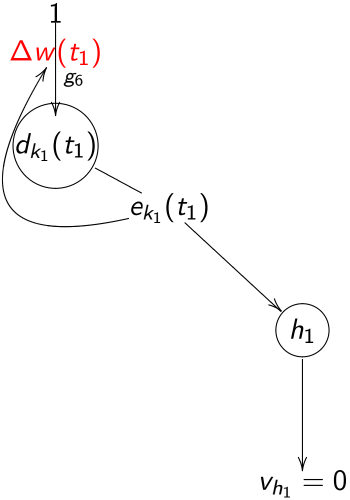
$w_{ki}(t_1) = v_k(t_1) = g_6$ is the Gödel number of $Q_1(f(a_1, a_2))$;
$d_k(t_1) = g_1$ is the Gödel number of $Q_1(f(x_1, x_2))$;
$e_k(t_1) = s(d_k(t_1) \ominus v_k(t_1))$ - the Gödel number of substitution for the disagreement set $d_k(t_1) \ominus v_k(t_1)$;
$\Delta w(t_1) = v_i(t_1)e_k(t_1) = e_k(t_1)$.

Example of Unification in Neural Networks: time $= t_1$.



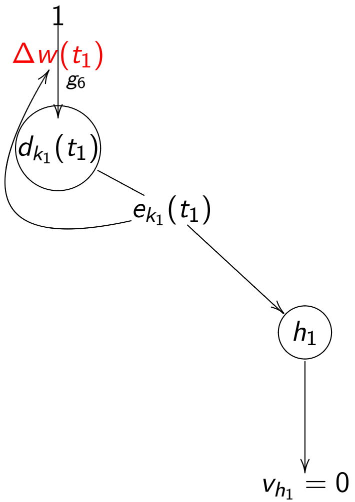
$w_{ki}(t_1) = v_k(t_1) = g_6$ is the Gödel number of $Q_1(f(a_1, a_2))$;
$d_k(t_1) = g_1$ is the Gödel number of $Q_1(f(x_1, x_2))$;
$e_k(t_1) = s(d_k(t_1) \ominus v_k(t_1))$ - the Gödel number of substitution x_1/a_1;
$\Delta w(t_1) = v_i(t_1)e_k(t_1)$;
$w_{ki}(t_2) = w_{ki}(t_1) \odot \Delta w_{ki}(t_1)$ and $d_k(t_2) = d_k(t_1) \odot \Delta w_{ki}(t_1)$ applies substitutions.

Example of Unification in Neural Networks: time $= t_{1-2}$.



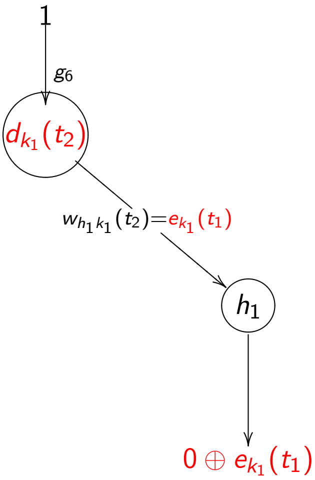
$w_{ki}(t_1) = v_k(t_1) = g_6$ is the Gödel number of $Q_1(f(a_1, a_2))$;
$d_k(t_1) = g_1$ is the Gödel number of $Q_1(f(x_1, x_2))$;
$e_k(t_1) = s(d_k(t_1) \ominus v_k(t_1))$ - the Gödel number of substitution x_1/a_1;
$\Delta w(t_1) = v_i(t_1)e_k(t_1)$;
$w_{ki}(t_2) = w_{ki}(t_1) \odot \Delta w_{ki}(t_1)$ and $d_k(t_2) = d_k(t_1) \odot \Delta w_{ki}(t_1)$ applies substitutions.

Example of Unification in Neural Networks: time $= t_{1-2}$.



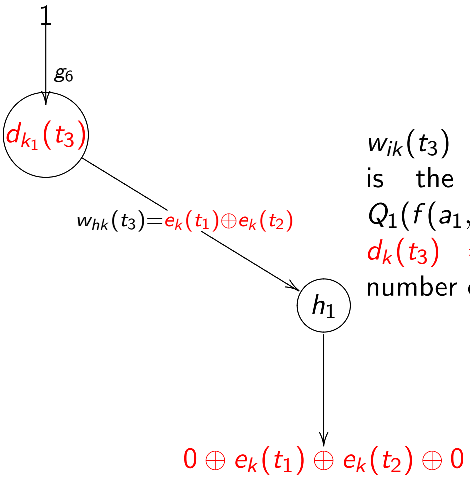
$w_{ki}(t_1) = v_k(t_1) = g_6$ is the Gödel number of $Q_1(f(a_1, a_2))$;
$d_k(t_1) = g_1$ is the Gödel number of $Q_1(f(x_1, x_2))$;
$e_k(t_1) = s(d_k(t_1) \ominus v_k(t_1))$ - the Gödel number of substitution x_1/a_1;
$\Delta w(t_1) = v_i(t_1) e_k(t_1)$;
$w_{ki}(t_2) = w_{ki}(t_1) \odot \Delta w_{ki}(t_1)$ and $d_k(t_2) = d_k(t_1) \odot \Delta w_{ki}(t_1)$ applies substitutions. $w_{h_1 k}(t_2) = w_{h_1 k}(t_1) \oplus \Delta w_{h_1 k}(t_1)$.

Example of Unification in Neural Networks: time = t_{1-2}.



$w_{ik_1}(t_2) = v_i(t_2) = g_6$ is the Gödel number of $Q_1(f(a_1, a_2))$;
$d_k(t_2) = g_7$ is the Gödel number of $Q_1(f(a_1, x_2))$.

Example of Unification in Neural Networks: time $= t_{2-3}$.



$w_{ik}(t_3) = v_i(t_3) = g_6$ is the Gödel number of $Q_1(f(a_1, a_2))$;
$d_k(t_3) = g_6$ is the Gödel number of $Q_1(f(a_1, a_2))$.

Some conclusions

Properties of these neural networks

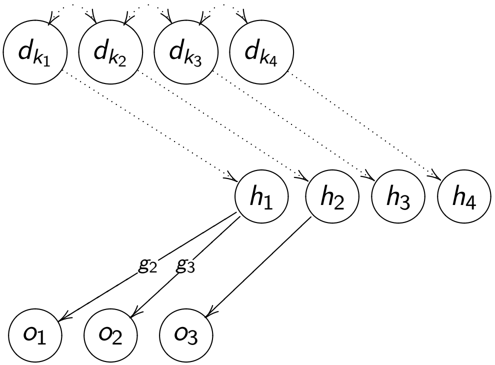
- First-order atoms are embedded directly into a neural network via Gödel numbers.
- Neural networks are finite and give deterministic results, comparing with infinite layers needed to perform substitutions in [HK94].
- Unification algorithm is performed as an adaptive process, which corrects one piece of data relatively to the other piece of data.

Main theorem

Theorem

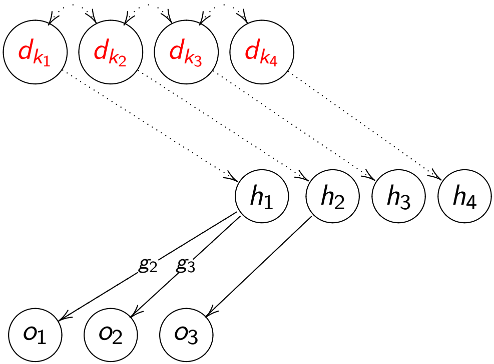
Let P be a definite logic program and G be a definite goal. Then there exists a 3-layer recurrent neural network which computes the Gödel number s of substitution θ if and only if SLD-refutation derives θ as an answer for $P \cup \{G\}$. (We will call these neural networks SLD neural networks).

Example. Time $= t_1$.



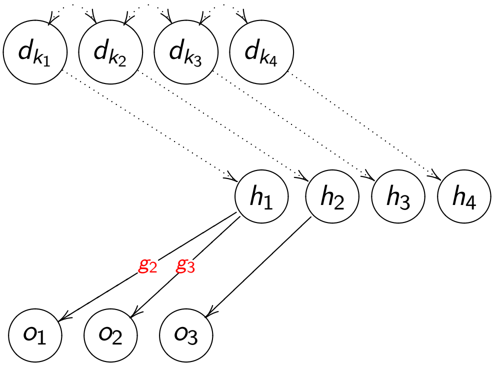
$Q_1(f(x_1, x_2)) \leftarrow$
$Q_2(x_1), Q_3(x_2);$
$Q_1(f(x_1, x_2)) \leftarrow Q_4(x_1);$
$Q_2(a_1) \leftarrow;$
$Q_3(a_2) \leftarrow.$

Example. Time $= t_1$.



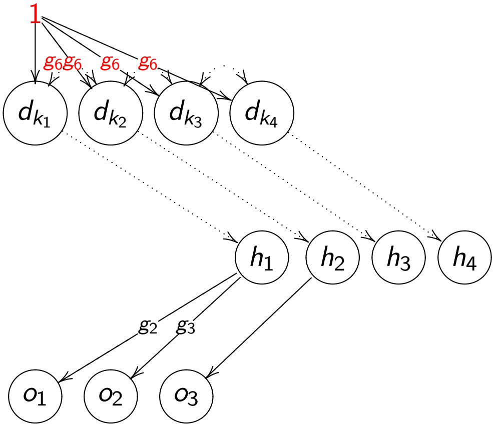
$Q_1(f(x_1, x_2)) \leftarrow$
$Q_2(x_1), Q_3(x_2);$
$Q_1(f(x_1, x_2)) \leftarrow Q_4(x_1);$
$Q_2(a_1) \leftarrow;$
$Q_3(a_2) \leftarrow.$

Example. Time $= t_1$.



$Q_1(f(x_1, x_2)) \leftarrow$
$Q_2(x_1), Q_3(x_2);$
$Q_1(f(x_1, x_2)) \leftarrow Q_4(x_1);$
$Q_2(a_1) \leftarrow;$
$Q_3(a_2) \leftarrow.$

Example. Time = t_1.



$g_6 = Q_1(f(a_1, a_2))$.

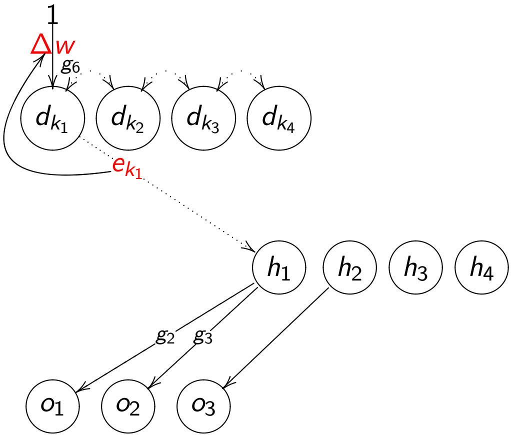
$Q_1(f(x_1, x_2)) \leftarrow$
$Q_2(x_1), Q_3(x_2)$;

$Q_1(f(x_1, x_2)) \leftarrow Q_4(x_1)$;

$Q_2(a_1) \leftarrow$;

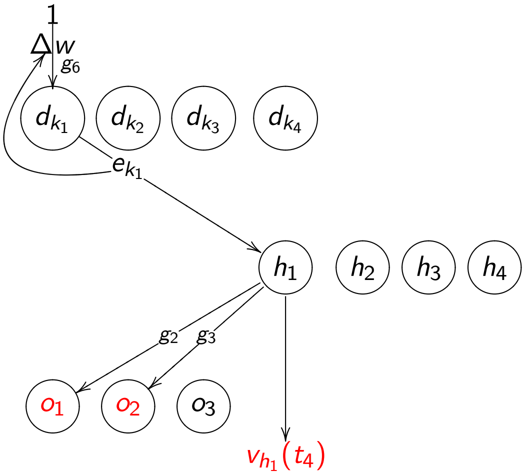
$Q_3(a_2) \leftarrow$.

Example. Time t_1: signals are filtered and unification initialized.



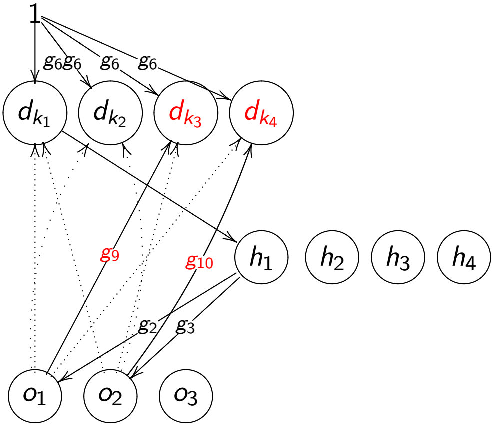
$g_6 = Q_1(f(a_1, a_2))$.
$Q_1(f(x_1, x_2)) \leftarrow$
$Q_2(x_1), Q_3(x_2)$;
$Q_1(f(x_1, x_2)) \leftarrow Q_4(x_1)$;
$Q_2(a_1) \leftarrow$;
$Q_3(a_2) \leftarrow$

Example. Time $t_2 - t_4$: unification.



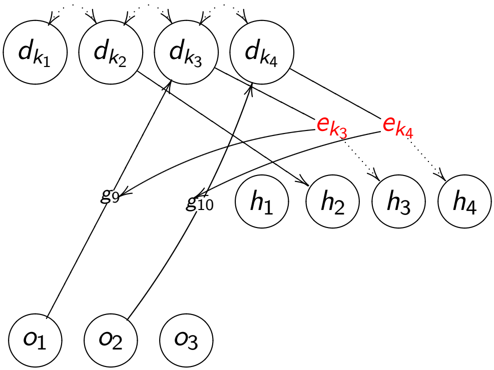
$g_6 = Q_1(f(a_1, a_2))$.
$Q_1(f(x_1, x_2)) \leftarrow$
$Q_2(x_1), Q_3(x_2)$;
$Q_1(f(x_1, x_2)) \leftarrow Q_4(x_1)$;
$Q_2(a_1) \leftarrow$;
$Q_3(a_2) \leftarrow$

Example. Time = t_5: values at layer o are computed:



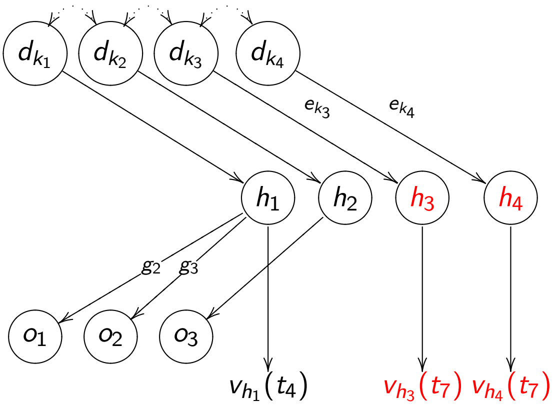
$g_6 = Q_1(f(a_1, a_2))$.
$Q_1(f(x_1, x_2)) \leftarrow$
$Q_2(x_1), Q_3(x_2)$;
$Q_1(f(x_1, x_2)) \leftarrow Q_4(x_1)$;
$Q_2(a_1) \leftarrow$;
$Q_3(a_2) \leftarrow$

Example. Time $= t_6$: new iterations starts, excessive signals are filtered, and unification initialized:



$g_6 = Q_1(f(a_1, a_2))$.
$Q_1(f(x_1, x_2)) \leftarrow$
$Q_2(x_1), Q_3(x_2)$;
$Q_1(f(x_1, x_2)) \leftarrow Q_4(x_1)$;
$Q_2(a_1) \leftarrow$;
$Q_3(a_2) \leftarrow$

Example. Time $= t_7$: unification is performed, answers are sent as an output:



$g_6 = Q_1(f(a_1, a_2))$.
$Q_1(f(x_1, x_2)) \leftarrow$
$Q_2(x_1), Q_3(x_2)$;
$Q_1(f(x_1, x_2)) \leftarrow$
$Q_4(x_1)$;
$Q_2(a_1) \leftarrow$;
$Q_3(a_2) \leftarrow$

Conclusions

- SLD neural networks have finite architecture, but their effectiveness is due to several learning functions.
- Unification is performed as adaptive process.
- Atoms and substitutions are represented in SLD neural networks directly, via Gödel numbers, and hence allow easier machine implementations.

Future Work

- Practical implementations of SLD neural networks.

Future Work

- Practical implementations of SLD neural networks.
- Theoretical development:
  - SLD neural networks allow higher-order generalizations.
  - ...can therefore be extended to higher-order Horn logics, hereditary Harrop logics...
  - ...can be extended to non-classical logic programs: linear, many-valued, etc...

Thank you!