Comparing Computational-level and Algorithmic-level models

Comparing Computational-level and Algorithmic-level models

Computational-level model

COHERENCE (formalization 3)

Input: A Hopfield network $G = (V, E)$, with a weight $-1 < w_{ij} < +1$ for each connection $(v_i, v_j) \in E = V \times V$.

Output: An activation pattern $A : V \rightarrow \{-1, 1\}$ such that harmony $H(A) = \sum_i \sum_j w_{ij} A(v_i) A(v_j)$ is maximized.

Computational-level model

DISCRIMINATING COHERENCE

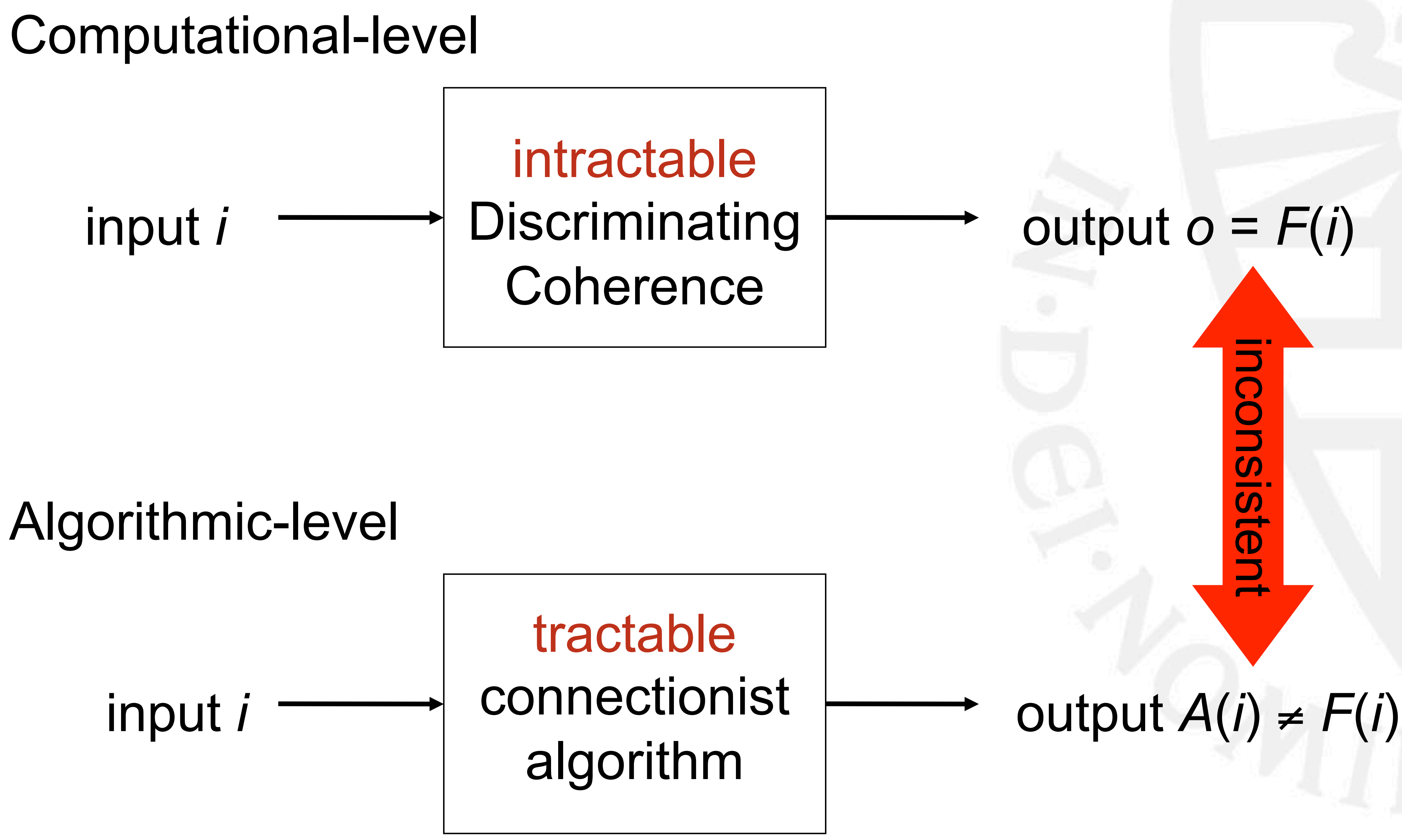
Input: A graph $G = (V, E)$, with a weight $w_{ij} > 0$ for each edge $(v_i, v_j) \in E = C^+ \cup C^- \subseteq V \times V$ (where $C^+ \cap C^- = \emptyset$) and a weight w_d for each $d \in D \subseteq V$.

Output: A truth assignment $T : V \rightarrow$ {true, false} such that Coh(T) =

$$\sum_{(v_i, v_j) \in C^+ \text{and } T(v_i) = T(v_j)} w_{ij} + \sum_{(v_i, v_j) \in C^- \text{and } T(v_i) \neq T(v_j)} w_{ij}$$

$$+ \sum_{d \in D, T(d) = true} w_d$$ is maximized.

Comparing Computational-level and Algorithmic-level models



Comparing Computational-level and Algorithm-level models

Practical assignment 2

Implement an exact (exhaustive search) algorithm Exh_{Coh} and the connectionist coherence algorithm N_{Coh}.

Generate (small!) instances of the D-Coherence problem on which to test the N_{Coh} algorithm.

Try to characterize as best as possible the performance of N_{Coh} by comparing its output to that of Exh_{Coh} (e.g., Is N_{Coh} doing better than chance? How does its performance scale?)

What are the implications of your findings?

How do you think the inconsistency between Exh_{Coh} and N_{Coh} can be resolved? (Suggestions can pertain to changes at both the computational- and the algorithmic level theory of coherence)

Connectionist Coherence Algorithm

Connectionist Coherence Algorithm

DISCRIMINATING COHERENCE	**Connectionist Algorithm**
Belief network $G = (V, C^+ \cup C^-)$	Neural network $G = (V, E)$
Elements V	Nodes V
Positive constraint $(v_i, v_j) \in C^+$, with $w_{ij} > 0$	Excitatory connection $(v_i, v_j) \in E$, $w_{ij} > 0$
Negative constraint $(v_i, v_j) \in C^-$, with $w_{ij} > 0$	Inhibitory connection $(v_i, v_j) \in E$, $w_{ij} < 0$
Element accepted, $T(v_i) = 1$	Node activated, $A(v_i) > 0$
Element rejected, $T(v_i) = 0$	Node deactivated, $A(v_i) < 0$

Connectionist Coherence Algorithm

DISCRIMINATING COHERENCE	**Connectionist Algorithm**
—	Continuous output is an activation pattern $A_{\text{continuous}} : V \rightarrow [-1, 1]$
Output is a truth assignment $T : V \rightarrow \{0, 1\}$	Discretized output is an activation pattern $A_{discrete} : V \rightarrow \{-1, 1\}$
Data element $d \in D \subseteq V$, with a weight w_d	Two nodes $d, s \in V$, with $w_{sd} = w_d$, and $A(s)$ fixed at +1
Coherence $Coh(T)$	Harmony $H(A) = \sum_i \sum_j w_{ij} A(v_i) A(v_j)$
Output T is such that Coh(T) is global maximum	Output $H(A)$ is local maximum (where network 'settles')

Updating Rule in the Connectionist Coherence Algorithm

$$A(v_j, t+1) = A(v_j, t)(1-d) + \begin{cases} net(v_j)(\max - A(v_j, t)) \text{ if } net(v_j) > 0 \\ net(v_j)(A(v_j, t) - \min) \text{ otherwise} \end{cases}$$

where $$net(v_j) = \sum_i w_{ij} A(v_i, t)$$

See also footnote 6 in Thagard & Verbeurgt (1998).

Note: The authors claim that using this updating rule, activations will not grow beyond +1 and –1 but this is not true in general. You may want to impose strict upper and lower bounds in your computer implementation.

Settling Rule in the Connectionist Coherence Algorithm

The network is said to have ‘settled’ if

(1) the activation values are stable; i.e., the activation of no node changes more than some minimal amount (parameter: *Min change*),

OR

(2) some maximum number of cycles has been reached (parameter: *Max time*).

Parameters of the Connectionist Coherence Algorithm

Parameter	Default value
Initial state	$A(v) = 1$for special element $s \in V$, and $A(v) = 0.1$ (or 0.01) for all other $v \in V$
Min, Max	1, –1
Decay d	0.05
Min change	0.01 (or 0.001)
Max time	200 cycles
Excitatory weights	All set to +0.4 (except, $w_{ds} = +0.5$ for all d connected to special element s)
Inhibitory weights	All set to –0.6

Comparing Computational-level and Algorithmic-level models

Practical assignment 2

Implement an exact (exhaustive search) algorithm Exh_{Coh} and the connectionist coherence algorithm N_{Coh}.

Generate (small!) instances of the D-Coherence problem on which to test the N_{Coh} algorithm.

Try to characterize as best as possible the performance of N_{Coh} by comparing its output to that of Exh_{Coh} (e.g., Is N_{Coh} doing better than chance? How does its performance scale?)

What are the implications of your findings?

How do you think the inconsistency between Exh_{Coh} and N_{Coh} can be resolved? (Suggestions can pertain to changes at both the computational- and the algorithmic level theory of coherence)