

Intro to Learning in SD -2

Alessio Micheli

E-mail: **micheli@di.unipi.it**

2- Neural Networks for Graphs

Apr 2019

DRAFT, please do not circulate!

www.di.unipi.it/groups/ciml



Dipartimento di Informatica
Università di Pisa



**Computational Intelligence &
Machine Learning Group**

Learning in Structured Domain

Plan in 2 lectures

1. Recurrent and Recursive Neural Networks

Extensions of models for supervised and unsupervised learning in structured domains

- Extensions of models for learning in structured domains
- Motivation and examples (structured data)
- The structured data (recursive)
- Recursive models: RNN and RecNN
- Recursive Cascade Correlation & other recursive approaches

2. Moving to DPAG and graphs: the role of causality

- Recap SD1
- Causality for Recurrent and Recursive models &
- Contextual approaches (BRCC/CRCC and DPAGs)
- Neural Networks for graphs

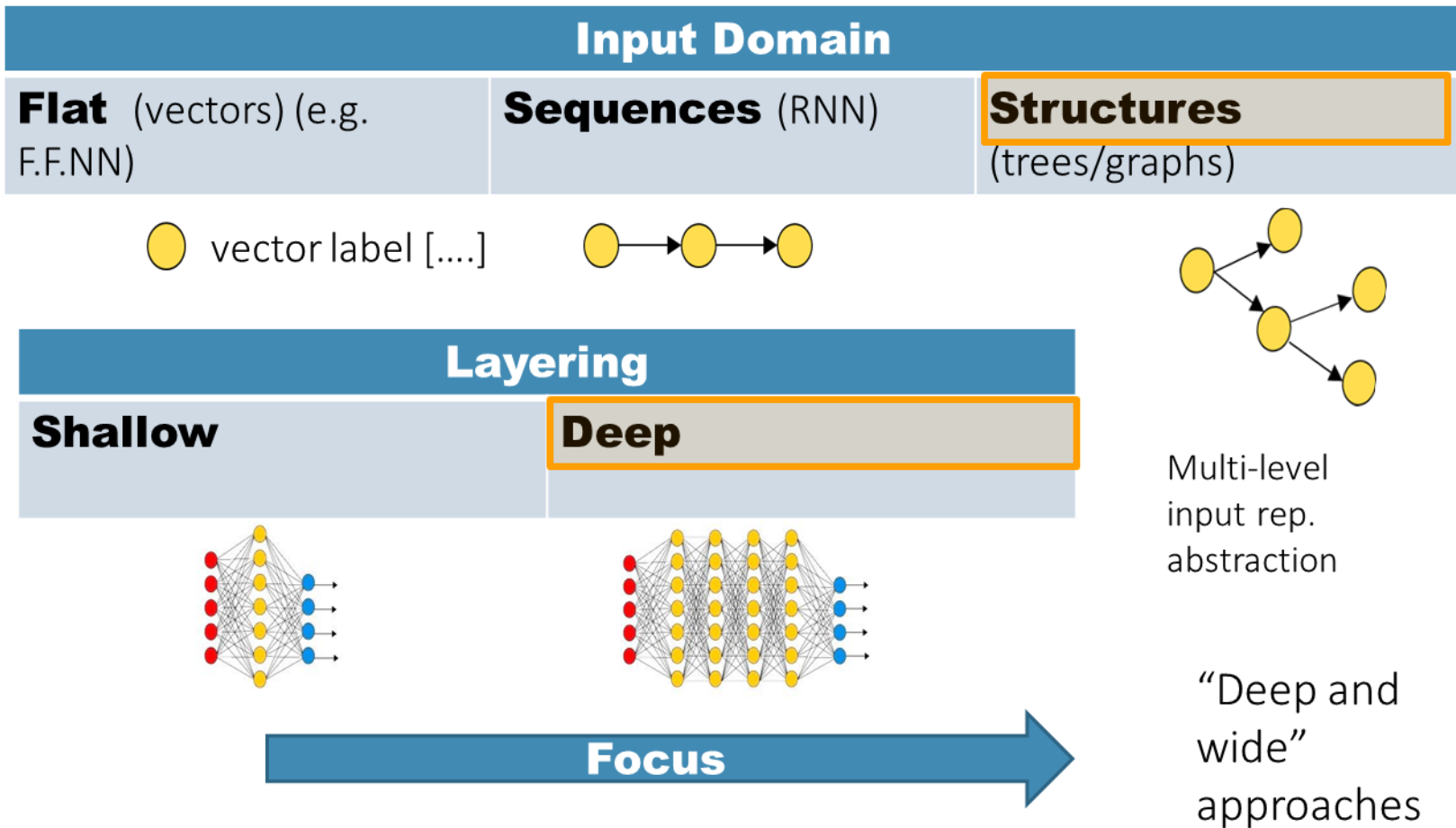
*By a journey through the
causality assumption!*

Recup SD-1:

Adaptive processing of SD

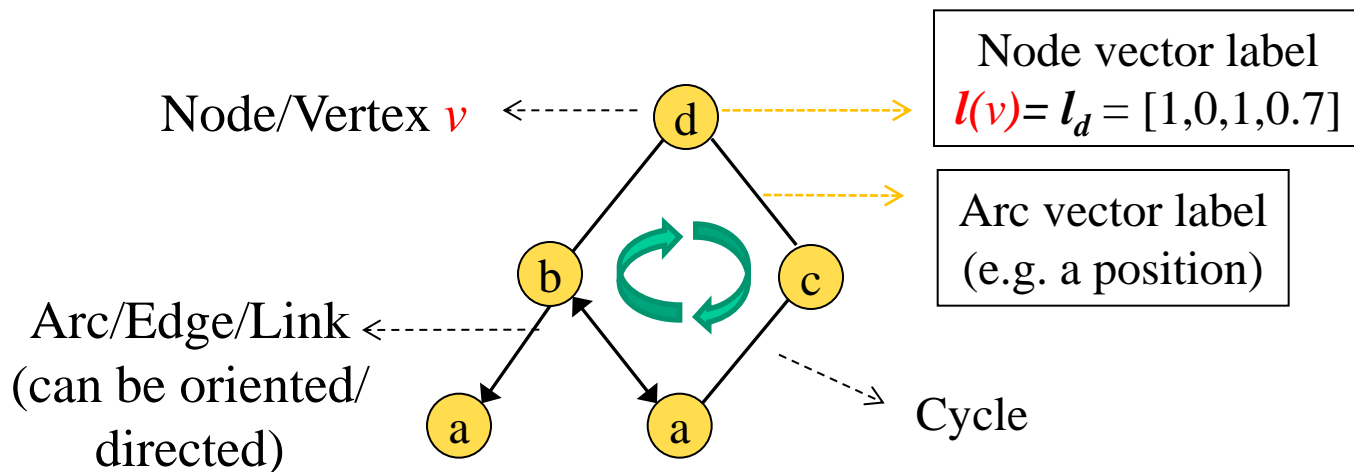
- The problem: there has been no systematic way to extract features or metrics relations between examples for SD
- **Goal:** to learn a mapping between a structured information domain (SD) and a discrete or continuous space (*transduction*).
- **Recursive** and parametric realization of the transduction function
- *Adaptive* by Neural Networks: RecNN
 - Pro: RecNN adapts the model to the hierarchal data
 - Cons: Causality issue: it affects the computational power of RecNN and the class of graphs !

The scenario, terms (and trends)



Our graphs (in the following)

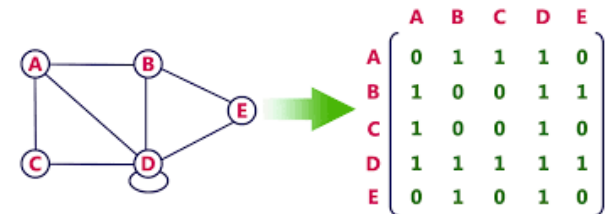
Labeled graphs g



Graph Representations

The problem: there has been no systematic way (of general validity for any task) to extract features or metrics relations between examples for SD

- **Features based** representations are incomplete (or strongly task-dependent, e.g. topological indexes)
- **Adjacent/incident matrix** representations (or other fixed-sizes representations). Issues:
 - Over-dim./incomplete (wasteful by padding/lose inf.)
 - Alignment among different graphs
 - Topological order (make difficult the generalization)
- ML issues for the high proportion between combinatorial number of possible data examples and available data
- “The ability to treat the proper **inherent nature of the input data** is the key feature for a successful application of the machine learning methodologies.”



Learning Models for SD

- Instead of moving *data to model*
(e.g. Graphs into vectors or trees into sequences, with alignment problems, loose of information, etc.)
we move *model to data*
- What we mean for *adaptive* processing of SD:
extraction of the topological information directly from data/ *structure representation learning*
 - \mathcal{H} has to be able to represent (hierarchic) relationships
 - **adaptive** measure of similarity on structures + apt **learning rule**
 - efficient handling of structure variability

Learning in Structured Domain

Plan in 2 lectures

1. Recurrent and Recursive Neural Networks

Extensions of models for supervised and unsupervised learning in structured domains

- Extensions of models for learning in structured domains
- Motivation and examples (structured data)
- The structured data (recursive)
- Recursive models: RNN and RecNN
- Recursive Cascade Correlation & other recursive approaches

2. Moving to DPAG and graphs: the role of causality

- Recap SD1
- Causality for Recurrent and Recursive models &
- Contextual approaches (BRCC/CRCC and DPAGs)
- Neural Networks for graphs

*By a journey through the
causality assumption!*

CRCC: introduction

- Analysis of the **causality** assumption for Recurrent and Recursive neural computing models
- Partial relaxation (or **extension**) of the causality assumption
- First approach to deal with contextual information in SD by Recursive models

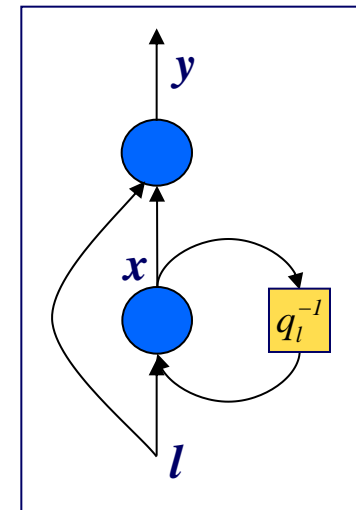
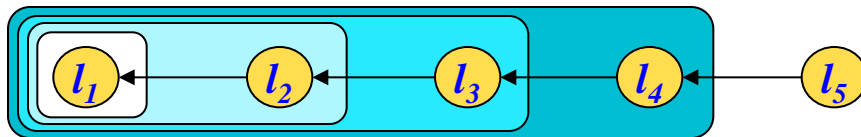
Causal Systems

- Recurrent NN models are based on the **Causality** assumption, i.e. RNN are only able to memorize past information

A system is causal if the output at time t_0 only depends on inputs at time $t \leq t_0$

necessary and
sufficient for
internal state

$$\begin{cases} \mathbf{x}(t) = \tau(\mathbf{x}(t-1), \mathbf{l}(t)) \\ \mathbf{y}(t) = g(\mathbf{x}(t), \mathbf{l}(t)) \end{cases}$$



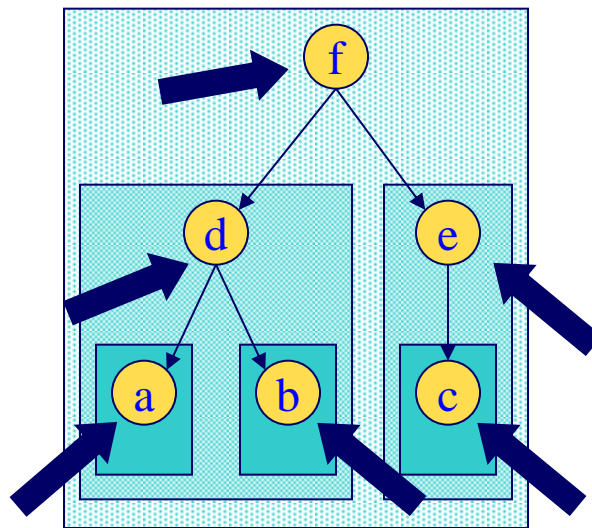
Graphical
model

Shift operator
 q^{-1}

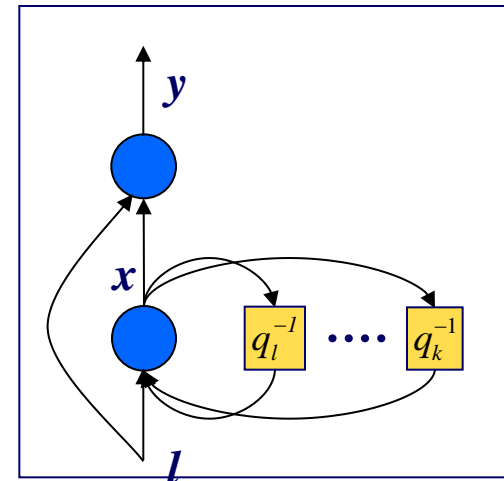
Causal Systems in Structured Domain (RecNN)

- The causality concept can be generalized to structured data transductions as follows

A system is causal if its output for a node v only depends on v and its descendants



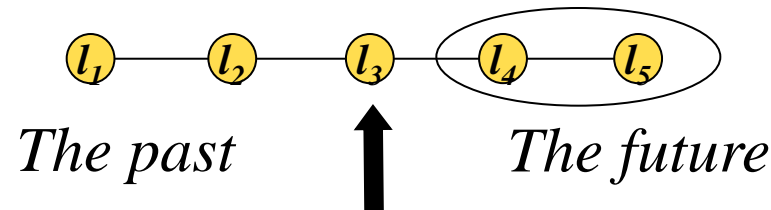
Unfolding the encoding
process through structure



$$\begin{cases} x(v) = \tau(x(\text{ch}[v]), l(v)) \\ y(v) = g(x(v), l(v)) \end{cases}$$

Drawbacks of Causal Systems for sequence domain

- Several prediction tasks involving **sequences** require past and “future” information (*on known sequences*)
 - DNA and Protein analysis / Language understanding / ...



Causality hampers to consider the right part

- Contextual information for **structured** domains: whenever the meaning of a sub-structure depends on the context in which it is found
 - some classes of transductions cannot be computed by causal models (also some causal transduction)
 - extension of the class of graphs
 - *Properties in flat domains cannot be trivially “exported” in SD!*

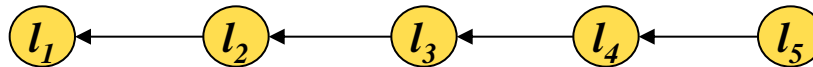
Overcome the Causality Assumption



Dip. Informatica
University of Pisa

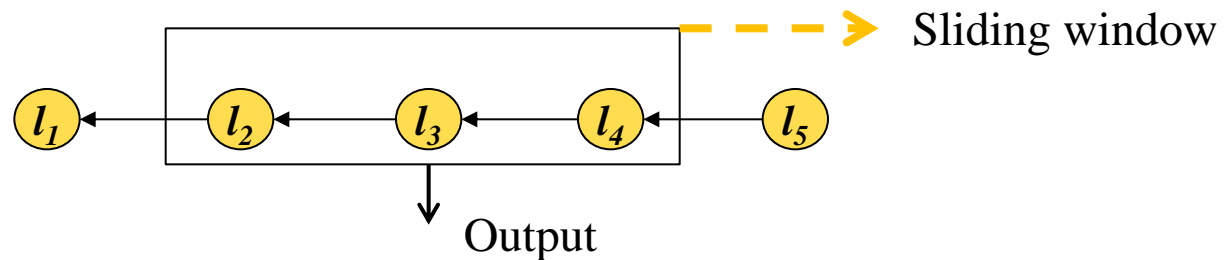
The Sequence Domain

- Standard Approaches
- BRCC



Standard Approaches: Sequences !!!!

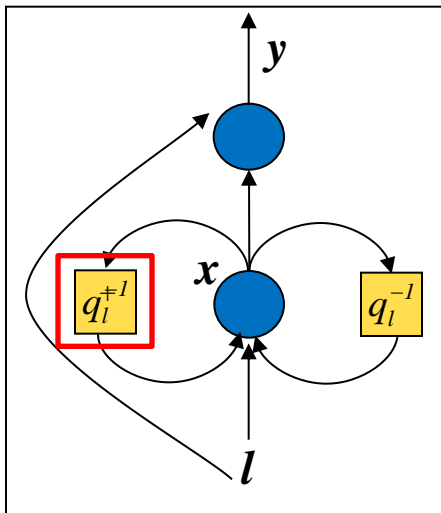
- Typical approaches involve feed-forward networks with a fixed input window
- An extension is given by the Time Delay Nets (for which however a fixed size windows must be specified)



- **And for structured domains ????**

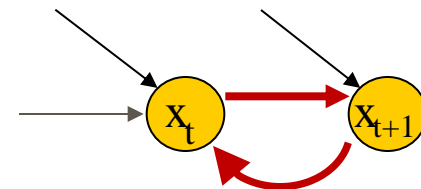
Bi-causal System

- A possible bi-causal model can be



$$\begin{cases} x(t) = \tau(x(t-1), x(t+1), l(t)) \\ y(t) = g(x(t), l(t)) \end{cases}$$

Unfolding
with cycles



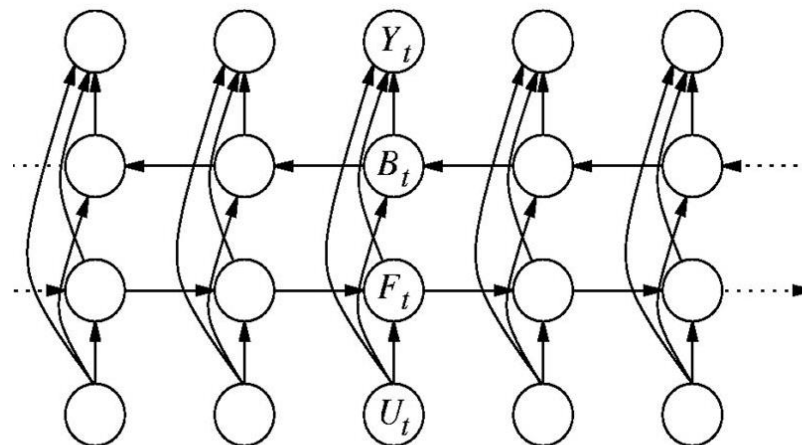
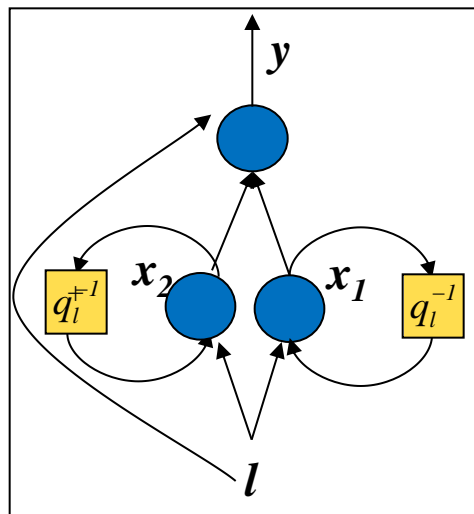
- However this is not easily implementable
 - Cycles: State equations and enc. net. become dynamical systems due to mutual dependencies
 - Different solutions are available (e.g. bidirectional approaches for RNN using a different state for left-to-right or right-to-left encoding)

Bi-directional Approaches

- A bi-directional approach has been proposed e.g. by Baldi et al. (1999) for Bioinformatics applications and nowadays popular in NLP etc., factorizing the internal state as:

$$\mathbf{x}(t) \equiv \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix} = \begin{bmatrix} \tau_1(\mathbf{x}_1(t-1), l(t)) \\ \tau_2(\mathbf{x}_2(t+1), l(t)) \end{bmatrix}$$

Typically
 $\mathbf{x}_1(t) = B_t$
 $\mathbf{x}_2(t) = F_t$



Output layer

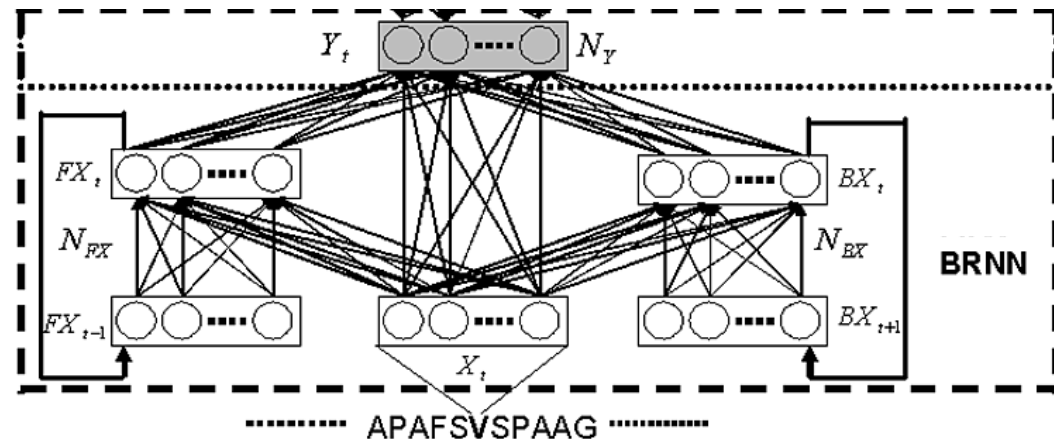
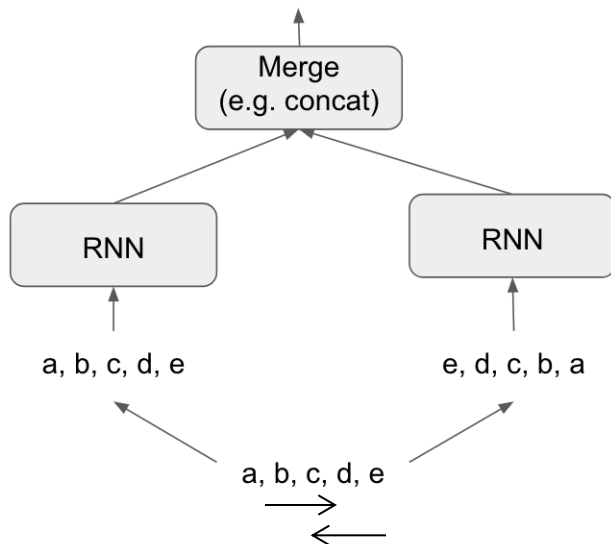
B_t for Backward

F_t for Forward

Input layer

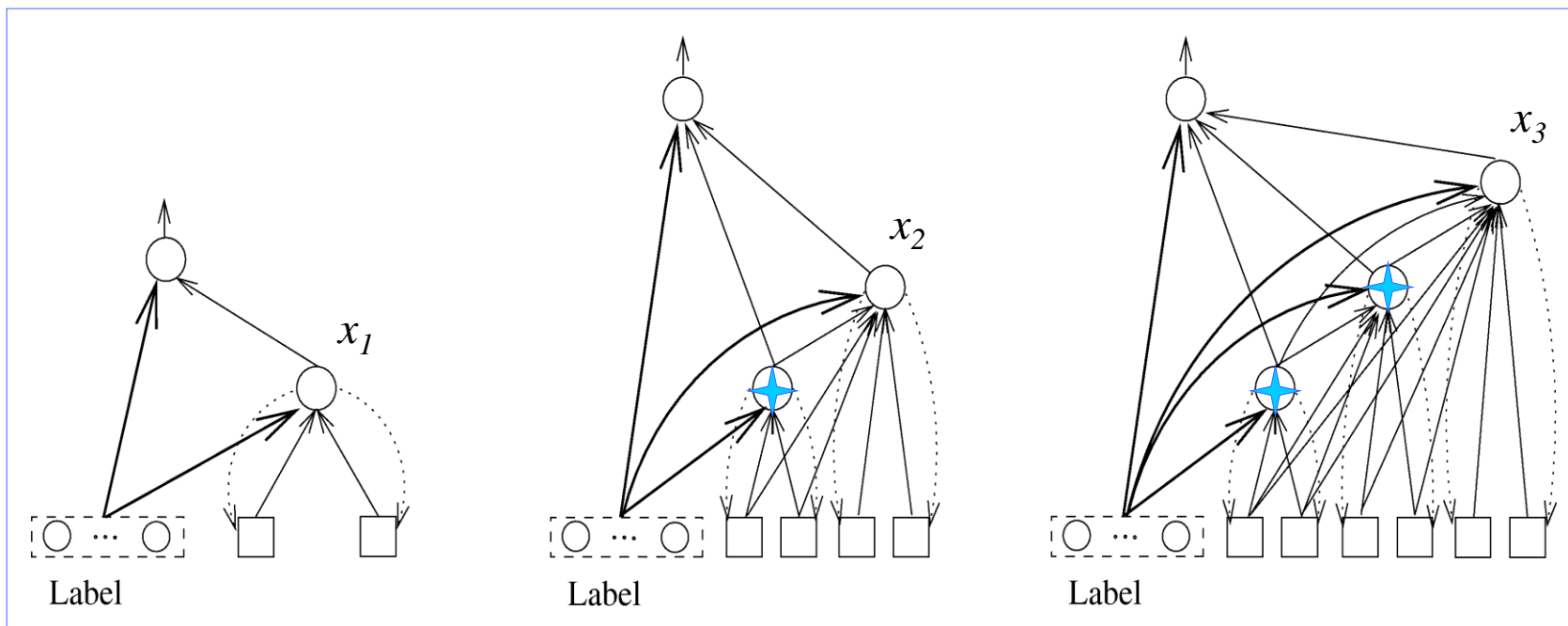
Bidirectional Recurrent NN (BRNN)

- Bi-directional Recurrent NN composed by a committee of three sub-networks, see these examples:
 - With the network size to be decided in advance
 - Not easily extendible to structures



A different idea by RCC Architecture

By a Recursive Cascade Correlation we can realize the recurrent/recursive network by a **constructive approach**:
The hidden units are added to the network, and frozen, during the training



BRCC/CRCC Approach

- We proposed an instance of Bi-Causality suitable for implementation with Recurrent Cascade Correlation
- Each time a unit is frozen, the portion of its (memorized) state encodes knowledge of the **whole** sequence

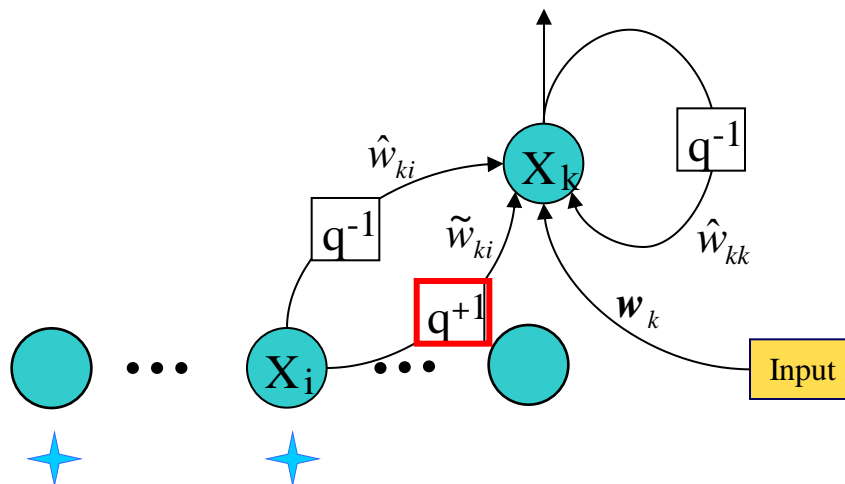
$$\begin{aligned}
 x_1(t) &= \tau_1(x_1(t-1), l(t)) \\
 x_2(t) &= \tau_2(x_2(t-1), x_1(t-1), x_1(t+1), l(t)) \\
 &\vdots \\
 x_m(t) &= \tau_m(x_i(t-1), x_{m-1}(t-1), \boxed{x_{m-1}(t+1)}, \dots, x_1(t-1), \boxed{x_1(t+1)}, l(t))
 \end{aligned}$$

unit m
frozen unit m-1
.....
frozen unit 1

Bi-Causal Recurrent Cascade Correlation

- Assuming stationary transitions the output of the k -th hidden unit of a BRCC can be computed as:

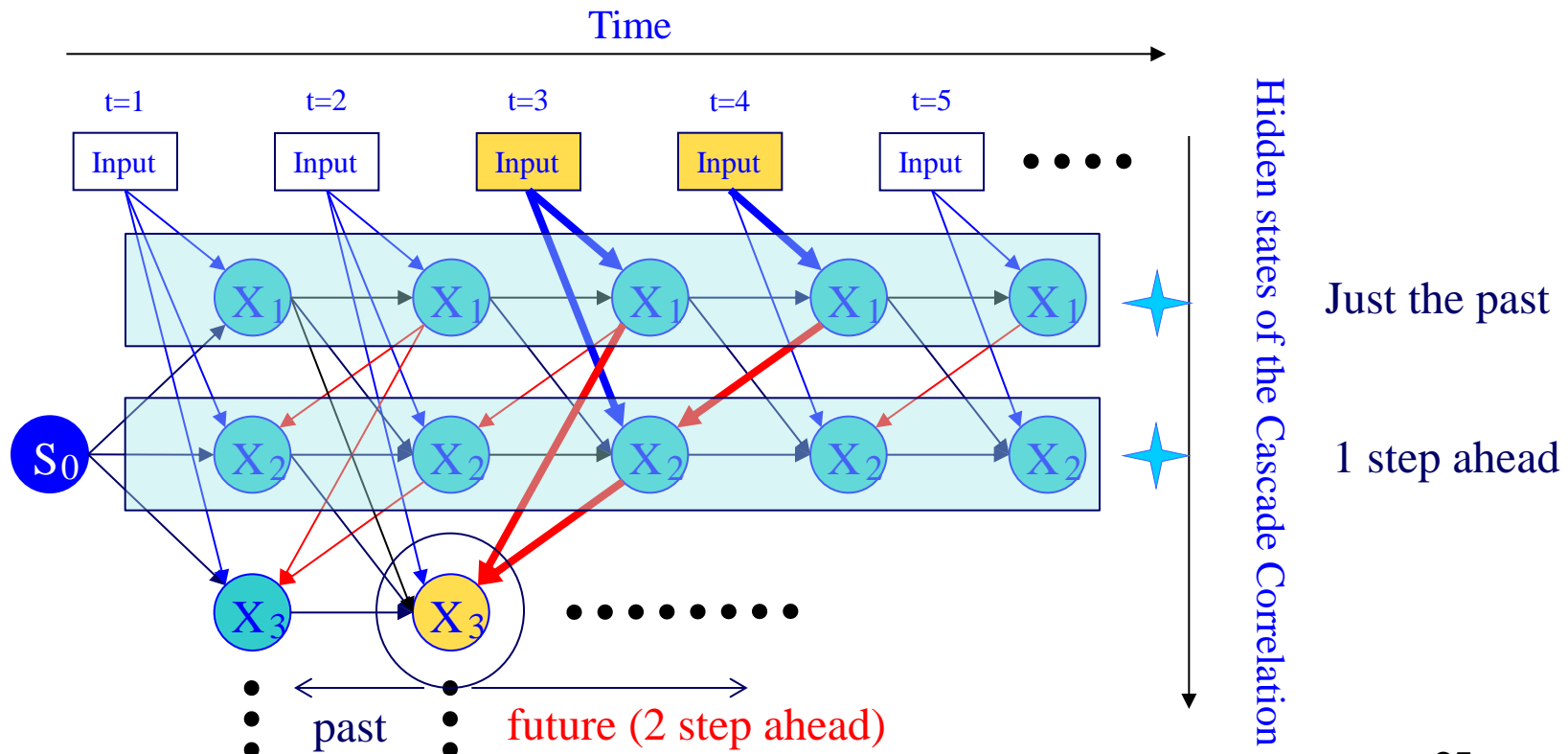
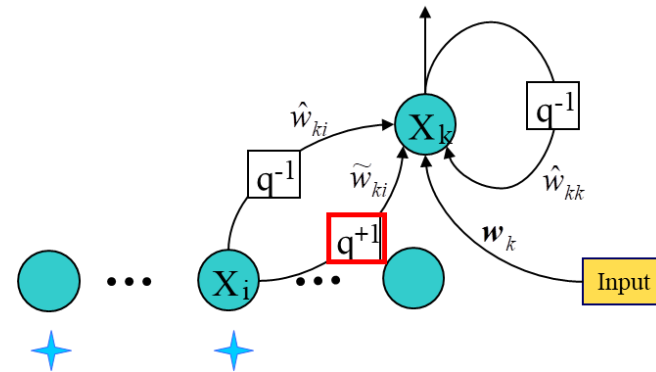
$$x_k(t) = f \left(\sum_{i=1}^n w_{ki} l_i(t) + \sum_{i=1}^k \hat{w}_{ki} x_i(t-1) + \sum_{i=1}^{k-1} \tilde{w}_{ki} x_i(t+1) \right)$$



Graphical Model

Example: CRCC on a sequence

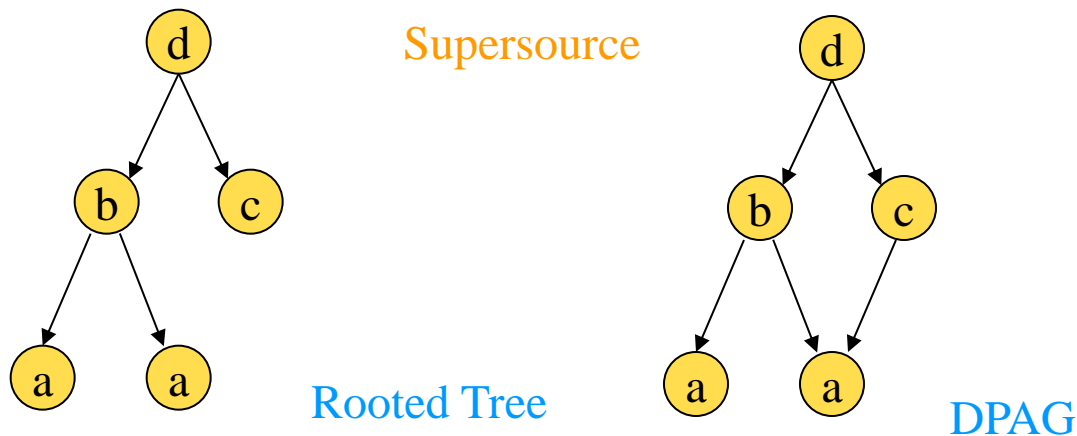
We can gain information on the "future" proportionally to the number of hidden units



Overcome the Causality Assumption for SD: CRCC

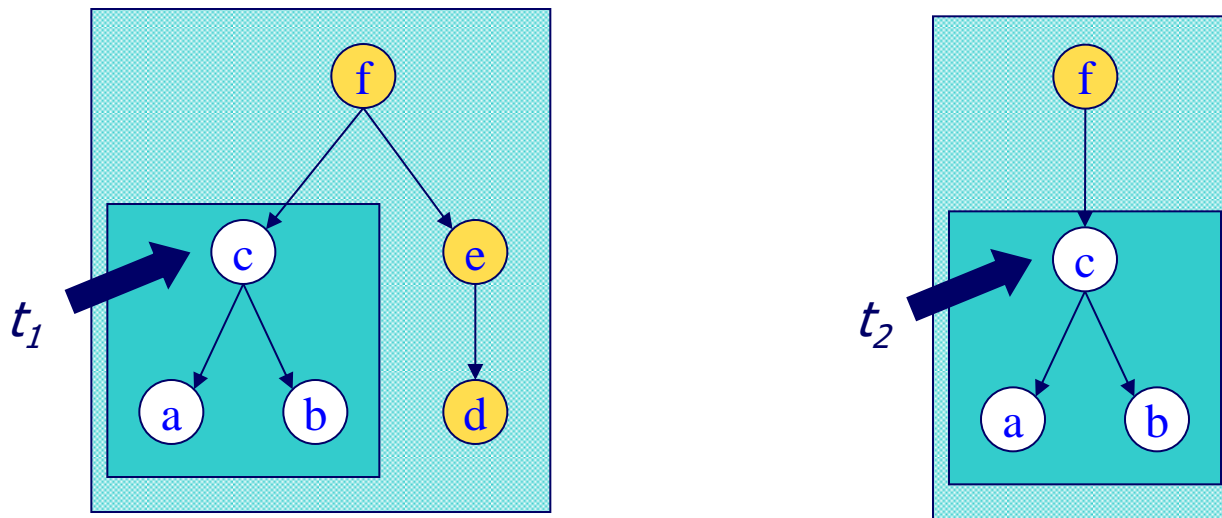
The Structure Domain

- CRCC: Cascade **Recursive** Cascade Correlation: Moving to trees and DPAGs
- Examples of Results



Contextual Target Functions

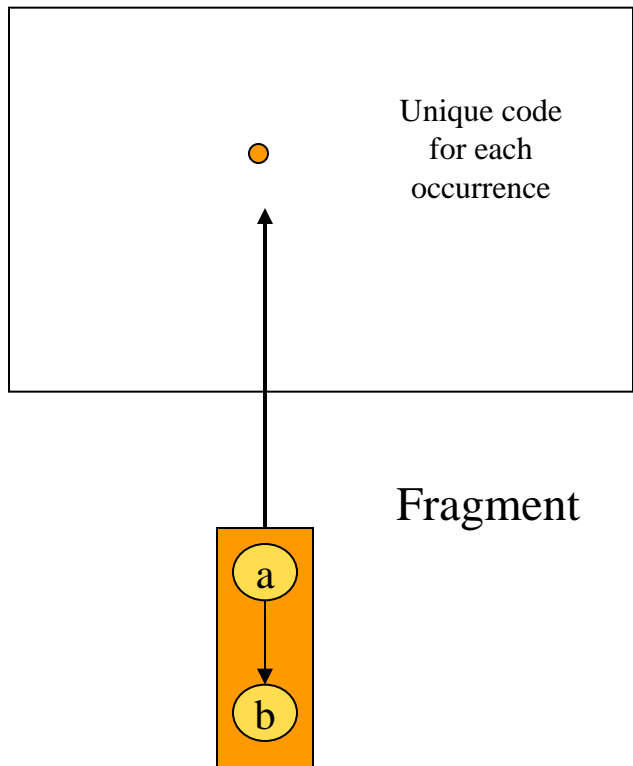
◆ Relevance of contextual processing (I)
contextual IO-isomorphic transductions



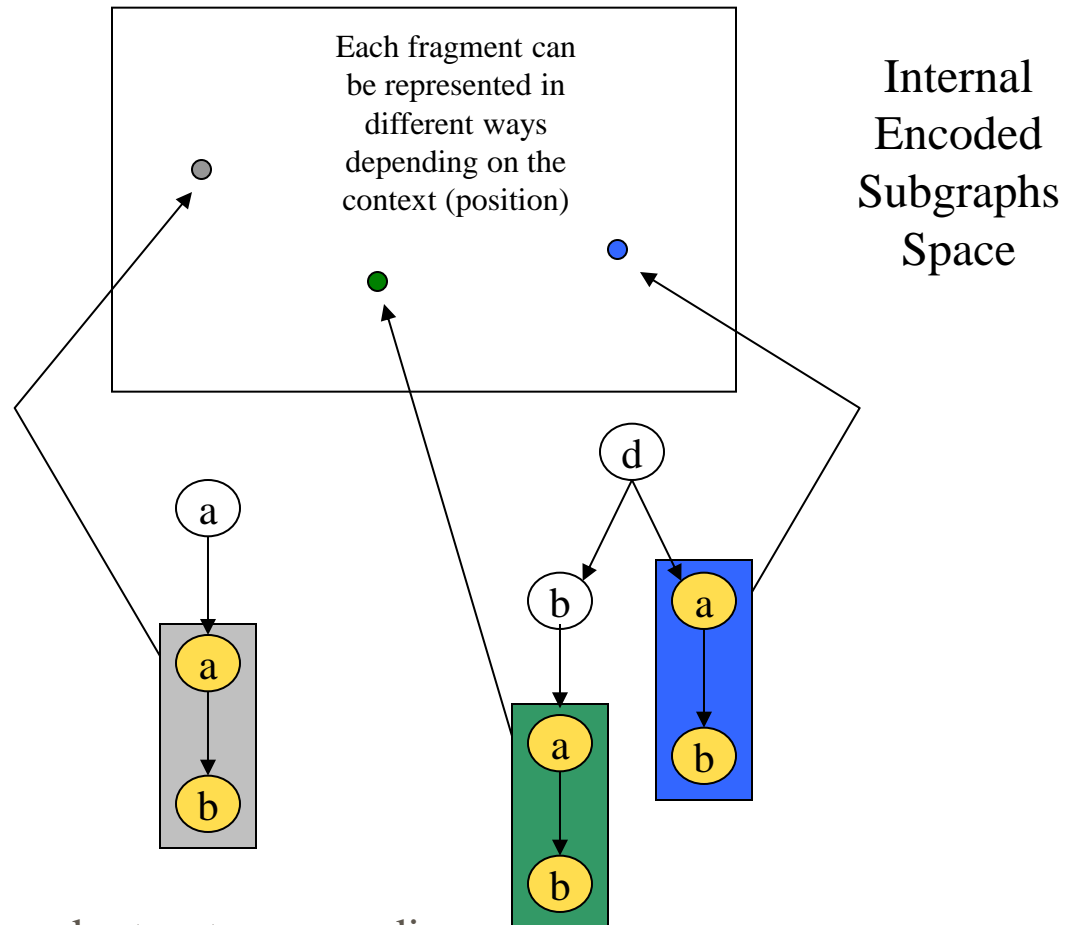
$$\begin{aligned} \text{Target}(t_1) &\neq \text{Target}(t_2) \\ \text{out}_{\text{RCC}}(t_1) &= \text{out}_{\text{RCC}}(t_2) \\ \mathbf{C}(x_k(c_1)) &\neq \mathbf{C}(x_k(c_2)) \end{aligned}$$

Example on the PCA Code Plot

Causal mapping



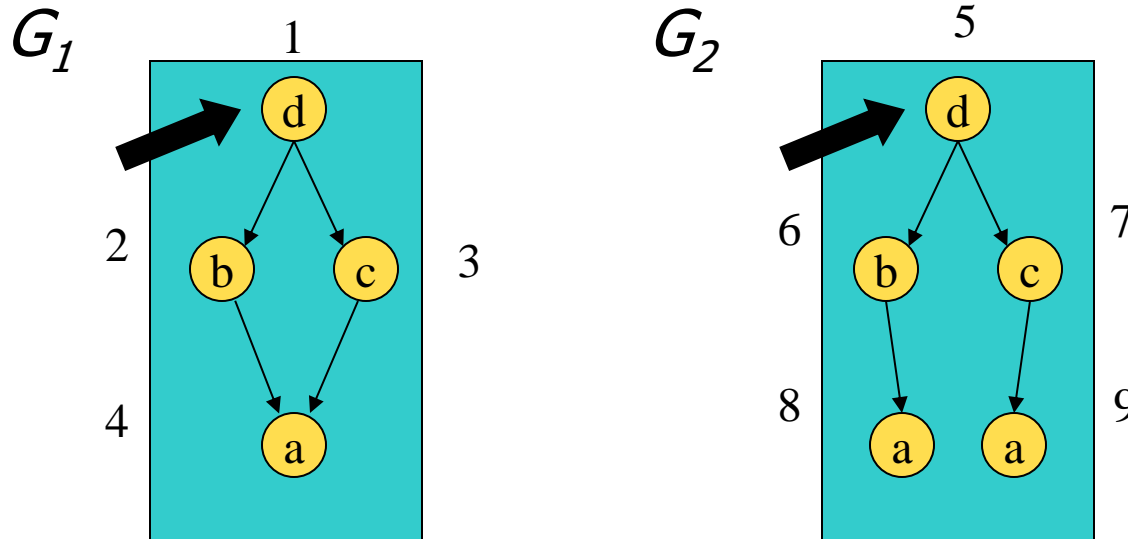
Contextual mapping



DPAG representation: a counter-example

◆ Relevance of contextual processing (II)

- Two different DPAG necessarily mapped into the same output by RNN/RCC (*supersource* causal transductions)

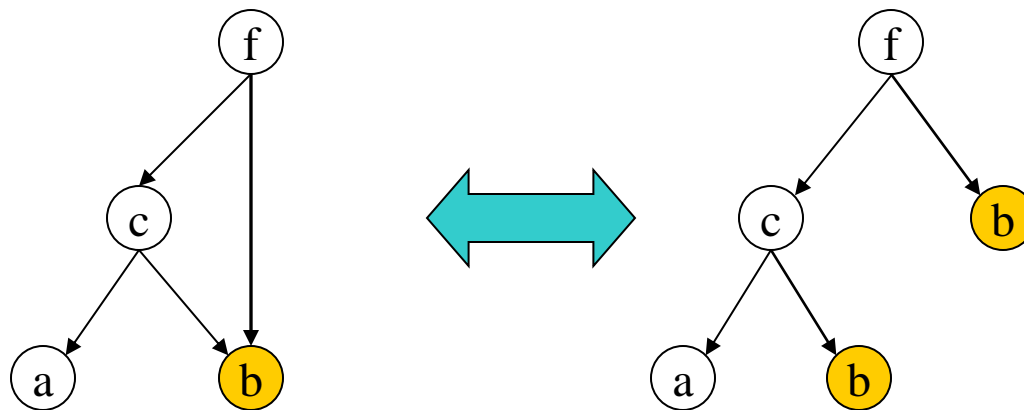


- CRCC **can** distinguish G_1 / G_2 (context for node "a" is different), RecNN **cannot** (*b and c see the same state values*)

DPAGs are not trees !

◆ Relevance of contextual processing (III)

- Causal models allow to rewrite a DPAG as an equivalent tree



- CRCC distinguish them !
- *We extended the domain from trees to DPAGs !*

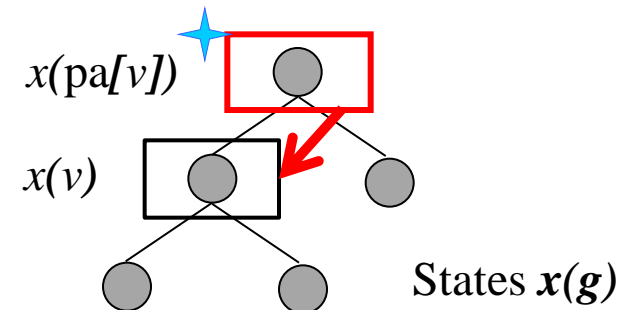
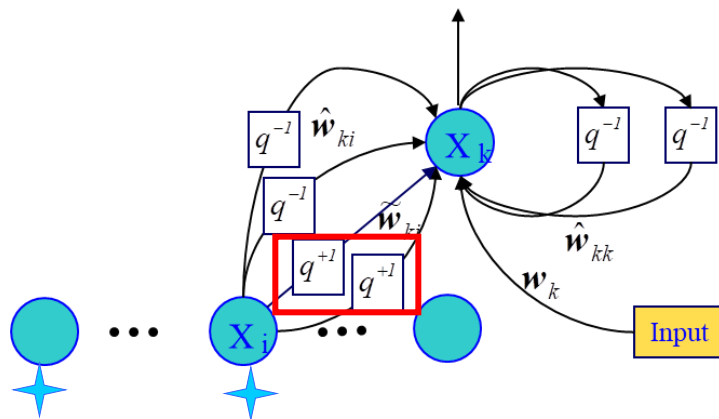
The CRCC Contextual Approach

- Each time a unit is frozen, the portion of its (memorized) state encodes knowledge of "the whole" structure

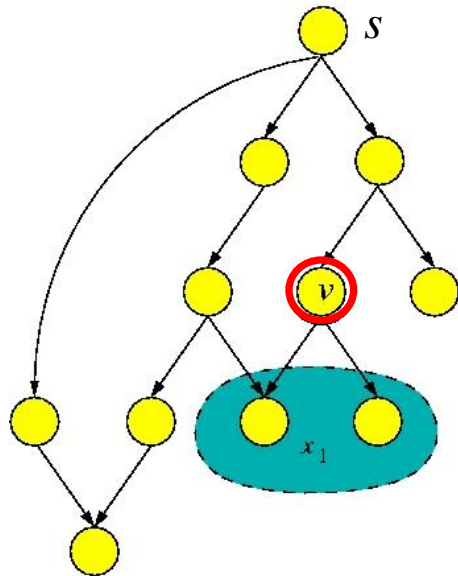
$$\begin{aligned}
 x_1(v) &= \tau_1(x_1(\text{ch}[v]), l(v)) \\
 x_2(v) &= \tau_2(x_2(\text{ch}[v]), x_1(\text{ch}[v]), x_1(\text{pa}[v]), l(v)) \\
 &\vdots \\
 x_m(v) &= \tau_m(x_m(\text{ch}[v]), x_{m-1}(\text{ch}[v]), \boxed{x_{m-1}(\text{pa}[v])}, \dots, x_1(\text{ch}[v]), \boxed{x_1(\text{pa}[v])}, l(v))
 \end{aligned}$$

unit m
frozen unit m-1
.....
frozen unit 1

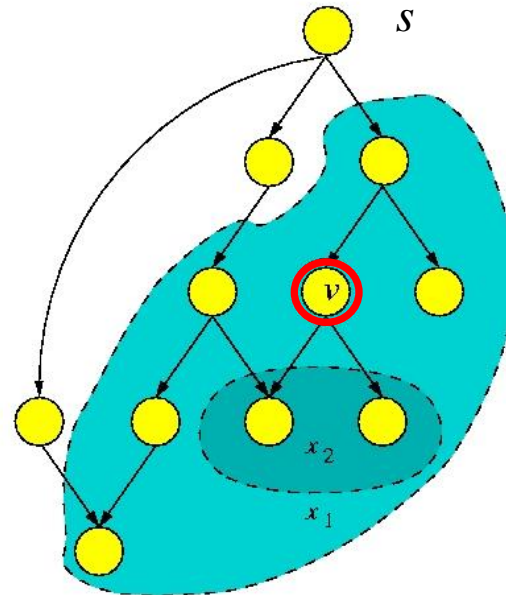
$\text{pa}[v]$: set of parents of v



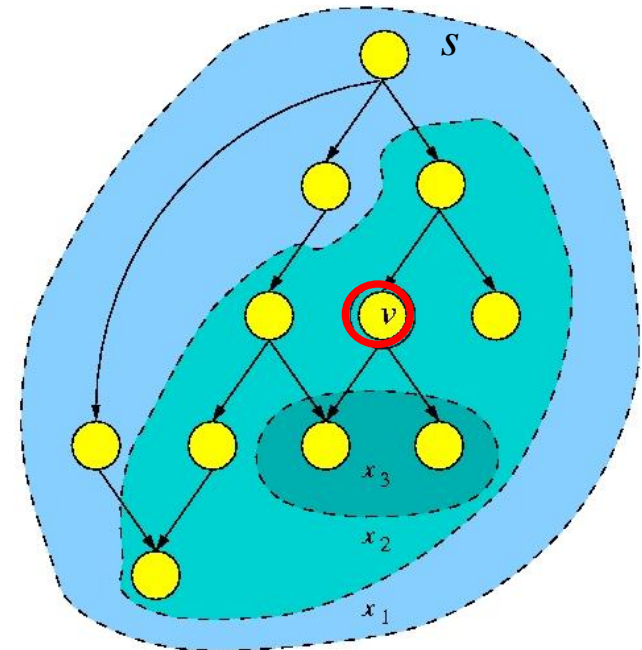
Example: $C(\bullet)$ for DPAGs



$C(x_1(v))$



$C(x_2(v))$



$C(x_3(v))$

The context grows (via `in_set`) including all sub-DPAG met along the (inverse) path $v \rightarrow s$ and $\downarrow v \rightarrow s$

Theory



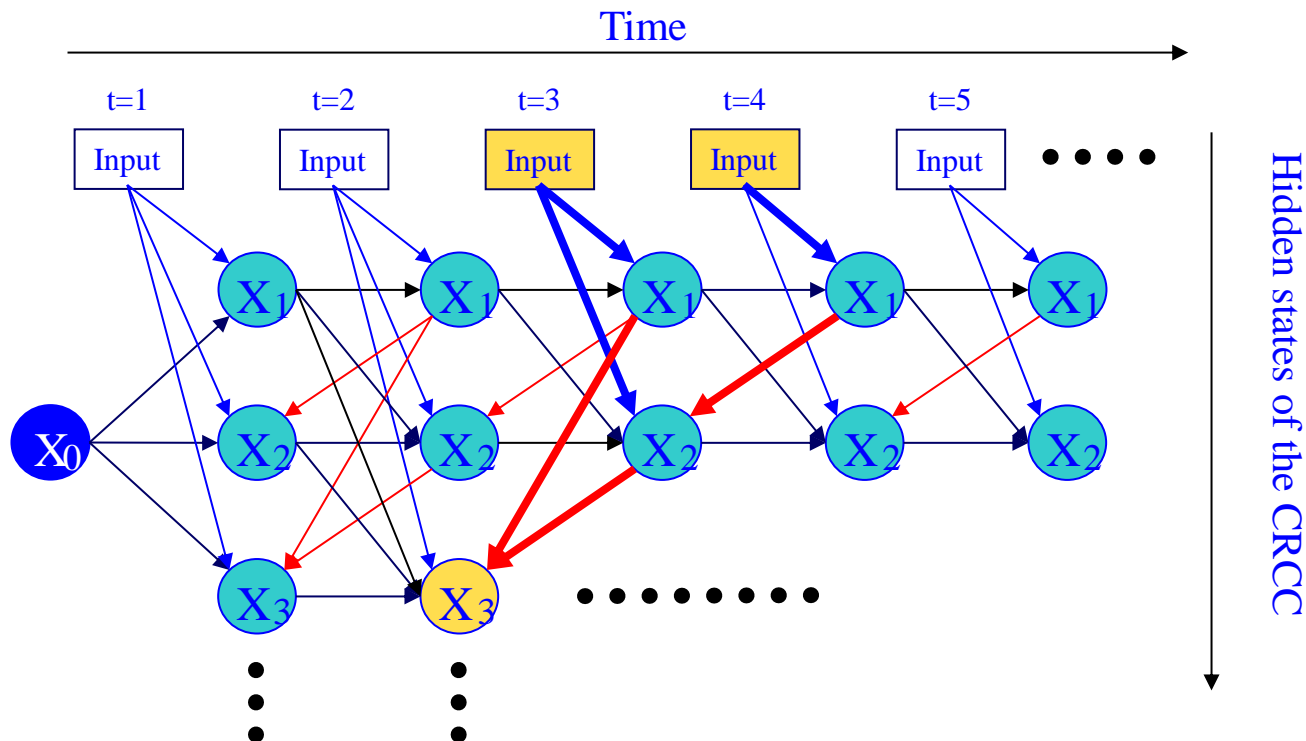
Dip. Informatica
University of Pisa

- Theoretical results have introduced to characterize the *computational power* of CRCC (class of computable functions/transductions vs causal models)
- Solving the examples before:
 - extension to *contextual* IO-isomorphic transductions,
 - e.g. $Target(v)=f(\text{whole structure})$: future dependencies.
 - extension to the class of supersource (causal) transductions involving DPAGs that cannot be computed by causal models
 - while supporting all the function computable by RCC
 - And also:
- Formal compact expression of the “context window”
- Proof of computational power of CRCC (**abstracting from neural realization**)

Example: $C(\bullet)$ for Sequences

$$C(x_k(v)) = \bigcup_{i=1}^{k-1} x_i \cdot \downarrow v_{t+k-i} \mathbf{U} x_k \cdot \downarrow v_{t-1}$$

It is possible to formalize the **context** giving formal expression of state functional dependencies
Example here for sequences.



Context Scope:

Properties relating h and C



- **Proposition 1.** Given a DPAG G with supersource s , for any vertex v such that $\text{dist}(s, v) = d$, the contexts $C(x_h(v))$ with $h > d$ involve all the vertices of G .
- **Proposition 2.** Given a DPAG G with supersource s , there exists a finite number h such that for each vertex v the context $C(x_h(v))$ involves all the vertices of the graph. In particular, any

$$h > \max_v \text{dist}(s, v)$$

satisfies the proposition.



SEE LATER

Universal Approximation



- B. Hammer, A. Micheli , A. Sperduti. **Universal Approximation Capability of Cascade Correlation for Structures** *Neural Computation* **17**, 1109–1159 (2005)
- RecCC can approximate every measurable functions form sequences and trees to real values (in spite of their restricted recurrent architecture) for finite sets.
- **CRCC: Universal approximation capability extended to classes of labeled DPAGs**

f approximated up to any desired degree of accuracy
(up to inputs of arbitrary small probability)

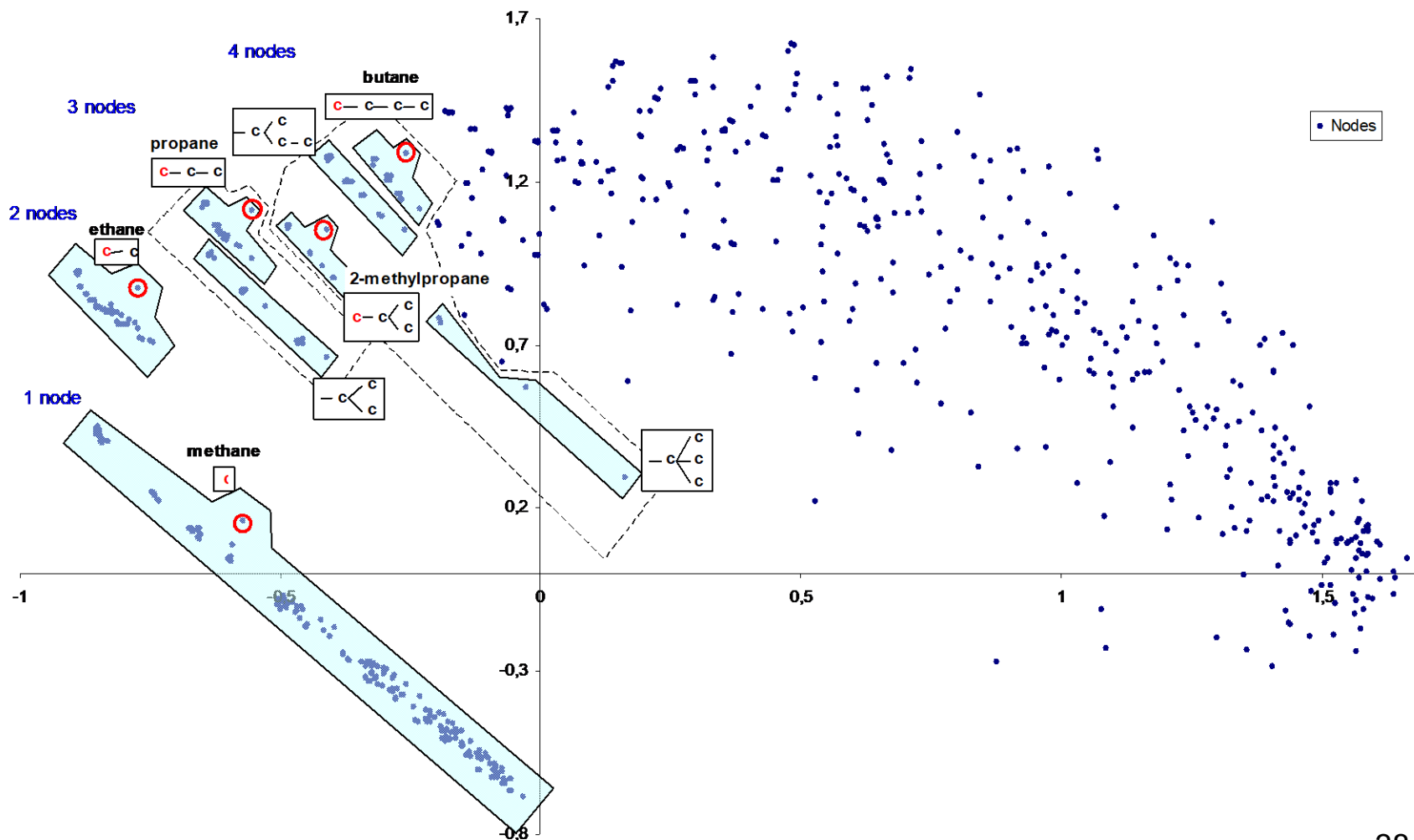
$$P(x \in DPAG : |f(x) - CRCC(x)| > \delta) < \varepsilon$$

Context in a CRCC Application



University of Pisa

PCA of the representation of the sub-structures developed by CRCC
for a chemical regression task



CRCC Conclusions

- Show advantages of including “context” (including parents)
 - Extension of the computation capability
 - Extension of the classes of data to DPAGs
 - Expressive encoding of substructures
 - Performance where causality assumption is unknown
- However, CRCC still requires topological order and supersource, still recursive dynamics: DPAGs/DAGs
- ... New approaches : by retaining and extending context, removing **causality**/recursion ?
 - Yes, ***Move to graphs!***

Learning in Structured Domain

Plan in 2 lectures

1. Recurrent and Recursive Neural Networks

Extensions of models for supervised and unsupervised learning in structured domains

- Extensions of models for learning in structured domains
- Motivation and examples (structured data)
- The structured data (recursive)
- Recursive models: RNN and RecNN
- Recursive Cascade Correlation & other recursive approaches

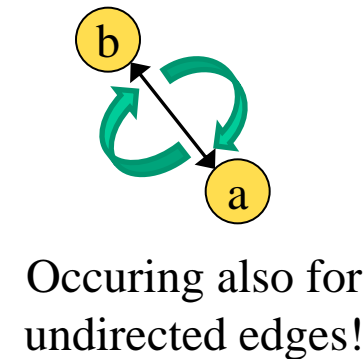
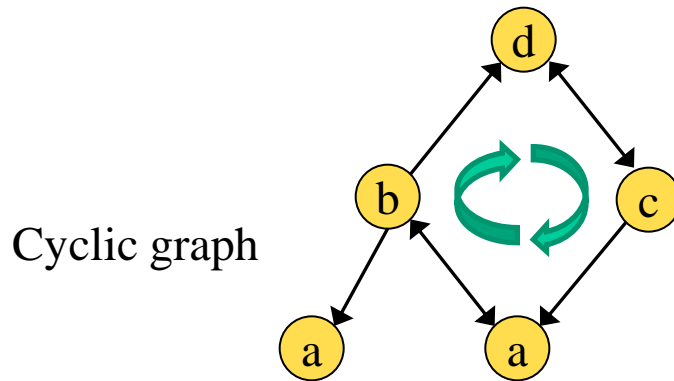
2. Moving to DPAG and graphs: the role of causality

- Recap SD1
- Causality for Recurrent and Recursive models &
- Contextual approaches (BRCC/CRCC and DPAGs)
- Neural Networks for graphs

*By a journey through the
causality assumption!*

Graphs by NN: Cycles

Causality assumption in RecNN introduce issues in processing **cycles** (due to the mutual dependencies among state values)



How to deal with cycles
and causality?

Main approaches for graphs by NN

Different classes of approaches:

1. Rewriting the graph:

- Atomic representation of cycles: e.g. functional groups in chemistry
- To trees/DAGs (e.g. SMILES representation in chemistry)

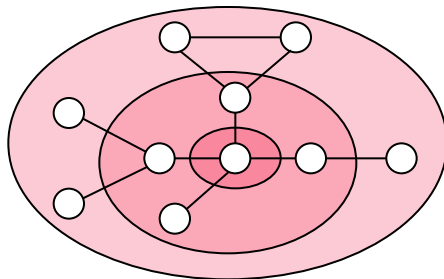
2. RecNN by explicitly treating the cyclic dynamics by contractive constraints (GNN, GraphESN) [1,2]

3. Layering: contextual non-recursive approaches (NN4G [3] /Conv. NN for graphs [4]) → Deep NN

1. Scarselli, Gori, Tsoi, Hagenbuchner, Monfardini. IEEE TNN, 2009.
2. Gallicchio, Micheli. IJCNN, 2010.
3. Micheli. IEEE TNN, 2009.

GNN/GraphESN (2009-2010)

- In GraphESN and GNN the equation are similar to RecNN
- Cycles are allowed (in state computation), the state is computed iterating the state transition function until **convergence**
- Stability of the recursive encoding process is guaranteed by resorting to **contractive** state dynamics (**Banach theorem for fixed point**)
 - In *GNN* imposing constrains in the loss function (alternating learning and convergence)
 - In *GraphESN* the condition is inherited by contractivity of the reservoir dynamics (see ESP conditions): *very efficient!*



GraphESN state transitions

$$\mathbf{x}(v) = \tanh(\mathbf{W}_{in}\mathbf{u}(v) + \sum_{v' \in \mathcal{N}(v)} \hat{\mathbf{W}}\mathbf{x}(v'))$$

Context evolution, with **the iteration,
of the state for the vertex in the center**

(by *diffusion on graph*)

- Scarselli, et al. IEEE TNN, 2009
- Gallicchio, Micheli. IJCNN, 2010.

Also extended to GRU (Li et al. 2015)

2. GNN/GraphESN

Pro/Cons:

- + Extend the domain of RecNN to general graphs
- + Theoretical approximation capability and VC dimension have been proved
- [GNN] elongate training time with the convergence (double mutual iteration)
- Constraints of the weight values \rightarrow bias to contractive transduction
- + GraphESN dose not require training time of the recursive part \rightarrow efficient!

3. Layering

Contextual Multi-Layered approaches for graphs



Dip. Informatica
University of Pisa

Layering basic idea:

- the mutual dependencies are managed (architecturally) through different layers
 - Instead of iterating at the same layer, each vertex can take the context of the other vertices computed in the previous layers, accessing progressively to the entire graph/network
 - And each vertex take information from all the other, including the mutual influences: **Collective inferencing**
- NN4G since 2005-2009 : a pioneer approach following the RecNN/ CRCC line (completely relaxing the recursive causality assumption)
 - In the following
- CNN for graphs since 2015: moving the idea for 2D processing (images) to graph processing through many layer

NN4G: Motivations

- Is it possible to find more general and simpler solutions removing causality without introducing cycles dependencies in the states definition ?

NN4G : Neural Network for Graphs

- Two main ingredients:
 - 1) constructive (feedforward) neural network approach
 - 2) Local and contextual information of each vertex of a graph

But recursive causality is removed

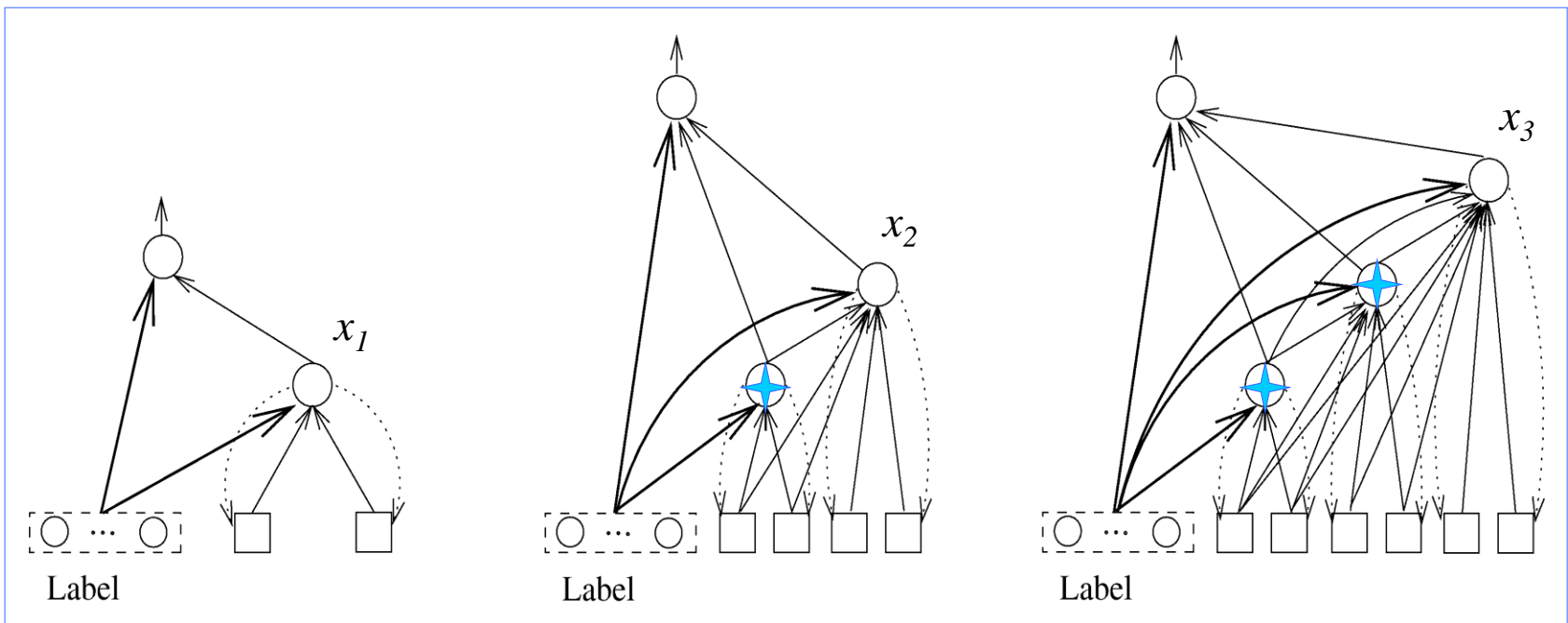
- Micheli, Sestito. WIRN 2005
- Micheli. IEEE TNN, 2009.



1) Constructive Approach

Cascade Correlation (*RecCC* in the picture):

The hidden units are progressively added to the network during training, and **frozen** after insertion

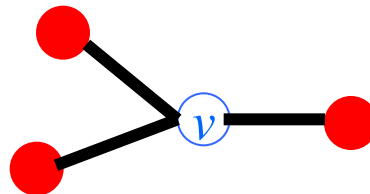


2) Local Context and Structured Domain

- We assume a fairly general class of labeled graphs $g \in \mathcal{G}$
- $Vert(g)$: set of vertexes of g ; $l(v)$: label of v
- $edg(v)$: set of edges incident on v
- **Neighbors of v** :

$$N(v) = \{u \in Vert(g) \mid (u, v) \vee (v, u) \in edg(v)\} \quad \text{Directed}$$

$$N(v) = \{u \in Vert(g) \mid (u, v) \in edg(v)\} \quad \text{Undirected}$$



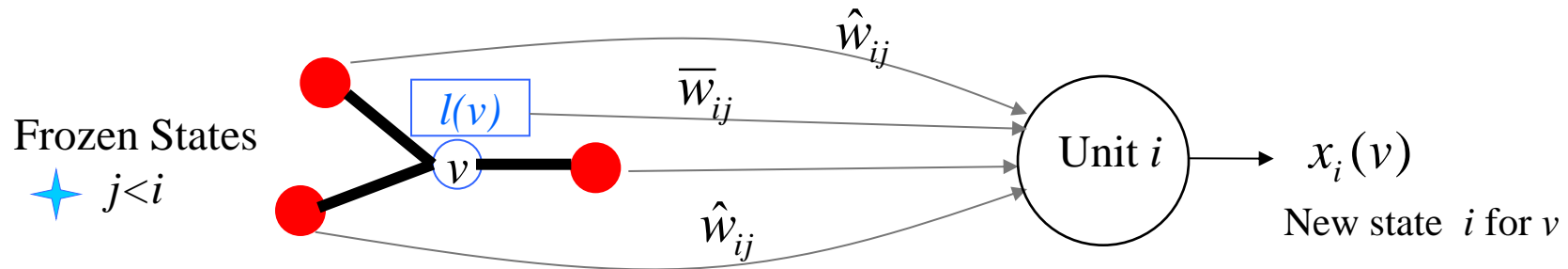
- Context of v is the set of vertexes with a path to/from v affecting the output of v .

NN4G: Hidden Units

- NN4G compute a state variable for each vertex

$$x(v) = \begin{cases} x_1(v) = f\left(\sum_{j=0}^{L^v} \bar{w}_{1j} l_j(v)\right) \\ x_i(v) = f\left(\sum_{j=0}^{L^v} \bar{w}_{ij} l_j(v) + \sum_{j=1}^{i-1} \hat{w}_{ij} \sum_{u \in \mathbf{N}(v)} x_j(u)\right) \end{cases} \quad i = 2, \dots, N$$

Current Label
Context



- Note: **Not Recursive** (no feedbacks): $x_i(v)$ depends only on frozen values ($j < i$)
 - No cyclic dependencies are introduced in the definition of the state transition system
- No topological order to follow: $x_i(v)$ can be computed in parallel for vertexes of g

NN4G: Hidden Units Generalization



- NN4G define a very general computational framework, e.g.

$$x(v) = \begin{cases} x_1(v) = f\left(\sum_{j=0}^{L^v} \bar{w}_{1j} l_j(v)\right) \\ x_i(v) = f\left(\sum_{j=0}^{L^v} \bar{w}_{ij} l_j(v) + \sum_{j=1}^{i-1} \sum_{u \in \mathbf{N}(v)} \hat{w}_{ij}^{(v,u)} x_j(u)\right) \end{cases} \quad i = 2, \dots, N$$

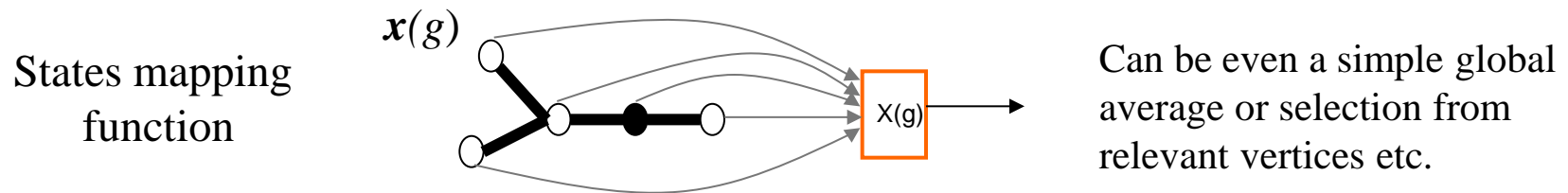
W for edge (v,u)

- (v,u) is unordered (undirected graphs)
- **Stationarity** (weight sharing) strategy: association between weights and edges
 - Entering/leaving edges for directed graphs
 - Position for positional/ordered graphs
 - **Label of the arc** more in general: $W^{(u,v)} = W^{(t,v)}$ if $L(u,v)=L(t,v)$
- First trials: full stationarity: 1 weights for each edges:
 - unordered and undirectd graphs and
 - strong parameters reduction

NN4G: Output unit

1. From states to the output layer

- IO- isomorphic transduction (an output for each vertex) or
- A scalar value for a whole graph can be emitted, using an operator X , e.g.:



2. Output layer: e.g. A single standard neural unit

$$y(g) = f\left(\sum_{j=0}^N w_j X_j(g)\right)$$

- Learning: as in (feedforward) Cascade Correlation: adding hidden units and interleaving min. of error at the output layer and max. of the correlation score for each hidden unit.

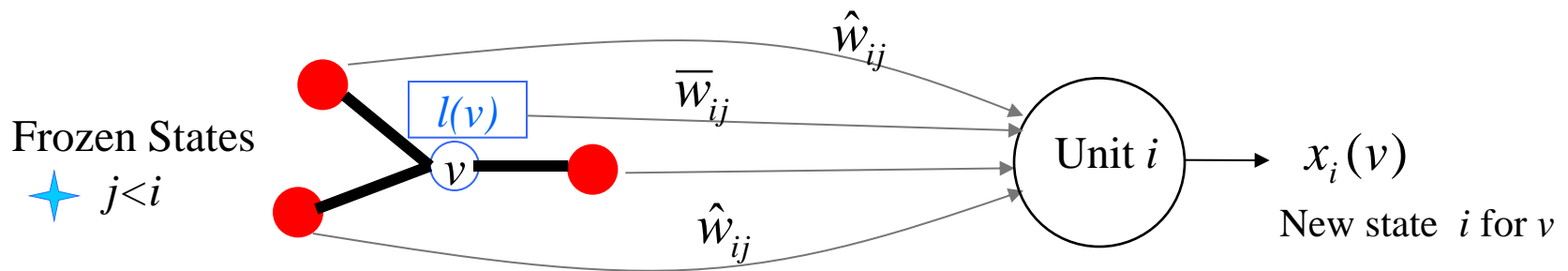


Algorithm

1. For $i=1$ to N
2. For all g in G
3. For all v in $Vert(g)$
4. Compute $x_i(v)$ (*even in parallel*)
5. Compute $X_i(g)$
6. For all g in G
7. Compute $y(g)$

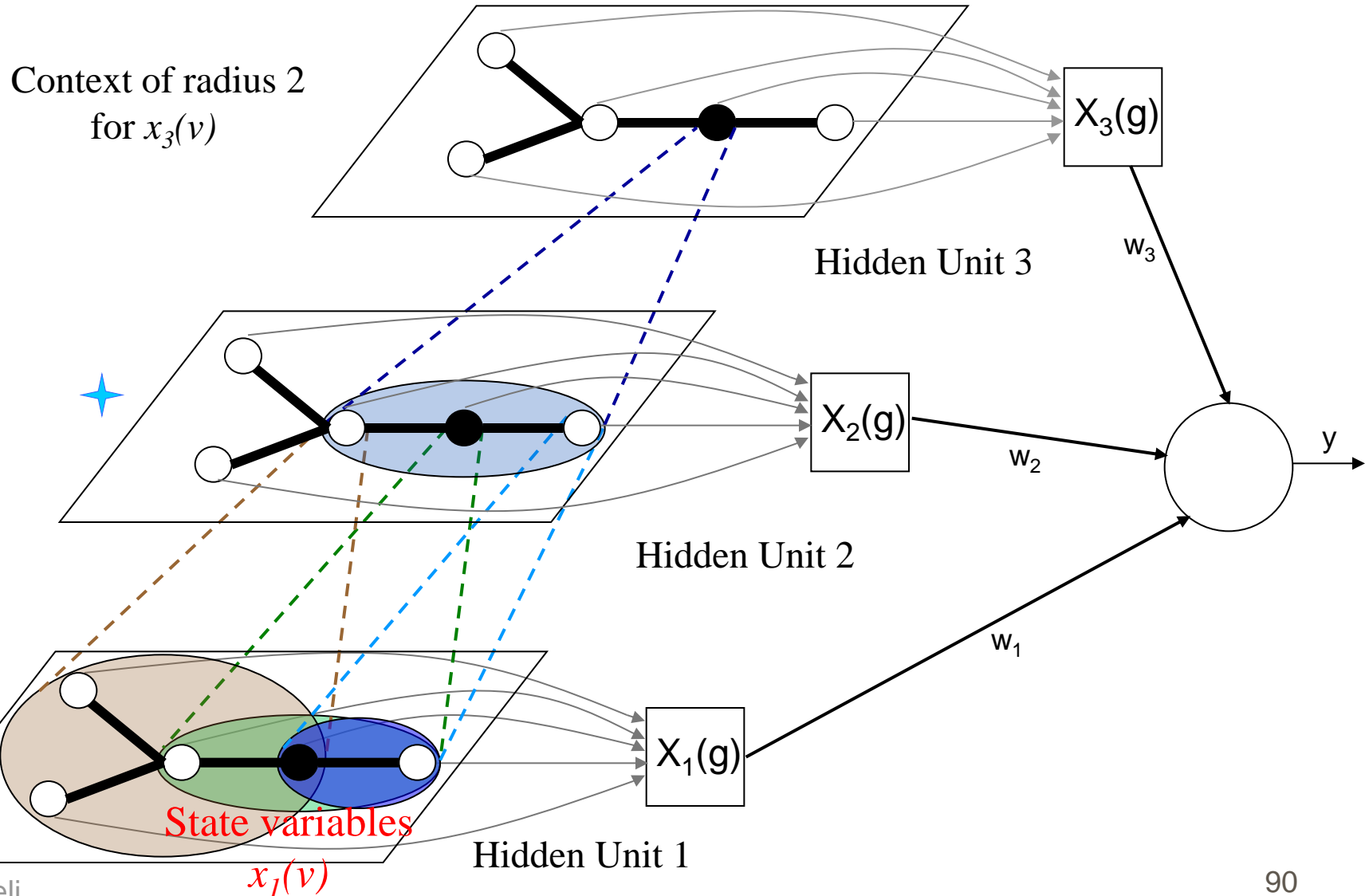
NN4G: 2) Hidden Units and Context

- Is NN4G just a relational approach taking only a local neighborhood (for each hidden unit) ?



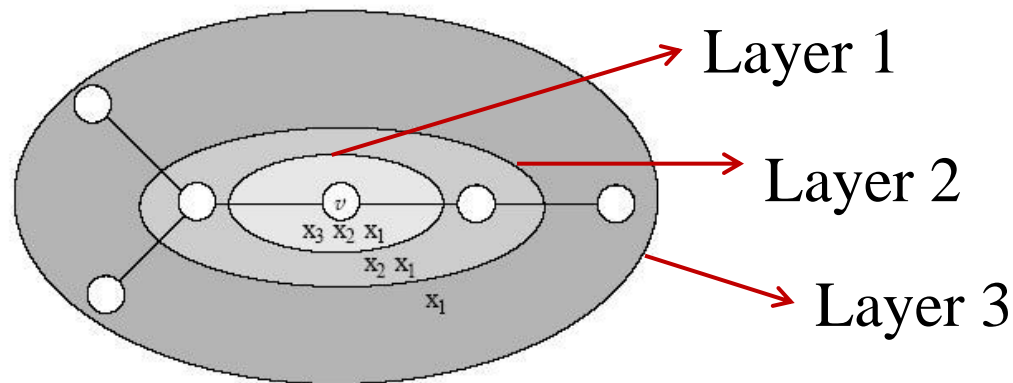
- No**, because through layering NN4G extend the context of each vertex to all the vertices in graph
- Because progressively, **by composition**, the model extends the context of influence to other vertices through the context developed in the previous frozen hidden units (layers) → *see the next slide*

Evolution of the Context (Compositional)



NN4G: Context Growth

- The growth of the context is symmetric in each direction starting from each vertex, and grow with layers



- In such a way, the size of the context window can grow and we do not need to fix it prior to learning.
- The **depth** of networks is functional to context development*

Context Scope:

Formal Properties relating h and C



Dip. Informatica
University of Pisa

- It has been **formally proved** that that the context $C(x_h(v))$ grows one step ahead, for each added unit (layer h), as $N^h(v)$:
 - the dimension of the context is proportional to the number of units,
 - and the structure of the composition is given by the topology of the input graph
- 1. And that $C(x_h(v))$ can involve all the vertices of the graph:

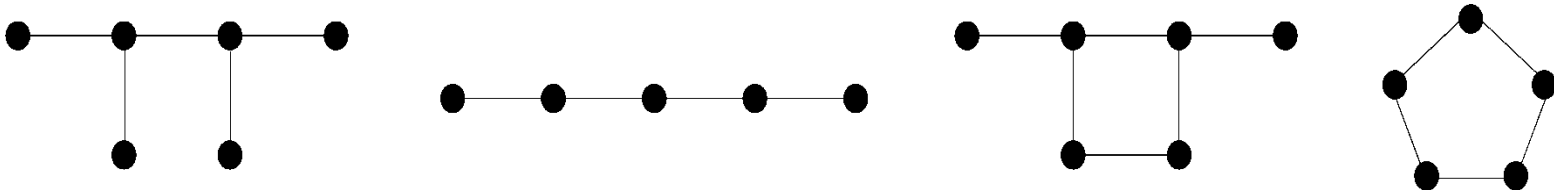
Theorem [NN4G]: Given a finite size graph G , there exists a finite number h of state variables (hidden layers) such that for each v in G the context of v involves all the vertices of G .

- In particular, $h > \text{"diameter"}$ of the graph satisfy the proposition.

Micheli. IEEE TNN, 2009.

Example of experimental assesment: Cyclic versus Acyclic Undirected Graphs

- Artificial task: test NN4G capability to learn a relevant topological feature i.e the **occurrence of cycles** in the input undirected graphs, which cannot be directly treated by RNN.
- **Input domain:** 150 cyclic graphs and 150 acyclic graphs with 3 up to 10 vertexes (2670 Vertexes)



- **100% test classification accuracy** over all the folds of 10-fold cross-validation with 5 trials for each fold.
- Just **2 hidden units**.
- In fact, the second unit is able to distinguish the ratio between the number of edges and vertexes in the graph, which is a sufficient feature to discriminate the input graphs on the basis of the occurrence of cycles in its topology.

NN4G Recap

- NN4G: *A deep model for graphs*
- Characteristics:
 - Direct/undirected cyclic/acyclic labeled graphs
 - W.r.t. RecNN does not assume **causality** over directed structure;
in particular, no assumption on the topological order is needed;
 - ☆ — Incremental, layer by layer learning & Automatic model design
 - **Depth** functional to contextual encoding: Dimension of context grows with layers (*formally proved*)
 - ☆ — **Efficient**: no cyclic def. of state var., divide et impera on the task
 - Scaling: Current model (full stationarity): $O(|G|Vh^2 \text{ epochs})$: Linear in the number of vertices
 - ☆ — Generality: No constraints on weights values
 - Pool strategy (Cascade corr. Training): local minima avoidance, supervised architecture optimization .

A first comparison

NN4G / Conv.NN for Graphs

Concepts in common:

- Visiting (the nodes of) input graphs through units with weight sharing (stationarity)
 - This correspond for CNN to the convolution over (the nodes of) input graphs,
 - i.e. constrained to graph topology instead of 2D matrix
- Layering and hence *moving to deep architecture* (functional to contextual processing)
- Composition for the (no causal) context learning, parsimony, and adaptivity are achieved and extend to *any kind of graphs*
- Node-centric learning can exploit the Collective inference

A first comparison

NN4G / Conv.NN for Graphs

Main differences are more related to the *training*:

- **CNN-Gs** typically use CNN architecture/training approaches,
 - Fixed architecture (few hidden layers)
 - Top-down back-prop (end-to-end): can be quite computational demanding using many layers
- **NN4G**: Incremental, layer by layer learning & automatic model design
 - Advantages: No gradient vanish issue, *divide et impera*, automatic number of layers, etc.

Learning in Structured Domain

Plan in 2 lectures

1. Recurrent and Recursive Neural Networks

Extensions of models for supervised and unsupervised learning in structured domains

- Extensions of models for learning in structured domains
- Motivation and examples (structured data)
- The structured data (recursive)
- Recursive models: RNN and RecNN
- Recursive Cascade Correlation & other recursive approaches

2. Moving to DPAG and graphs: the role of causality

- Recap SD1
- Causality for Recurrent and Recursive models &
- Contextual approaches (BRCC/CRCC and DPAGs)
- Neural Networks for graphs
- **Other models and looking ahead**

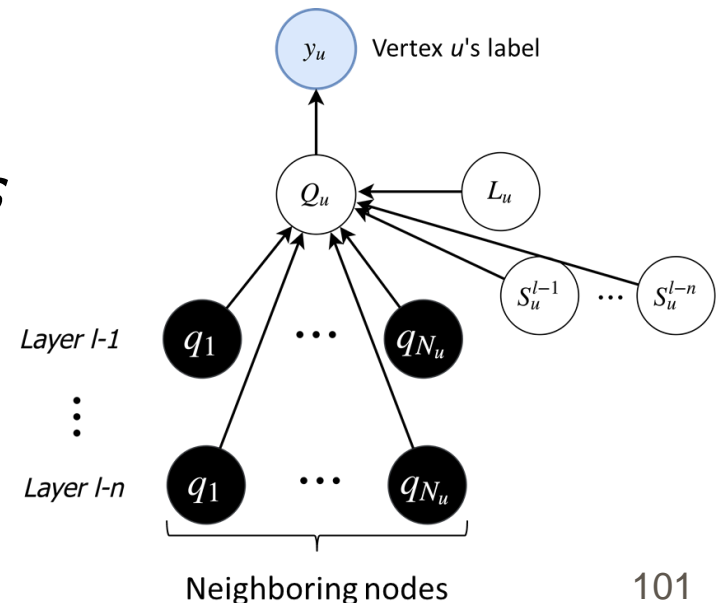
*By a journey through the
causality assumption!*

NN4G+ HTMM

- We can extend such contextual ideas also to RC and HTMM approaches making them deep and for graphs
- E.g. ICML 2018 A NN4G realized by a *generative* approach
- trained by a mix of unsupervised (Markov models for hidden layer) and supervised (output layer) approaches
- Also for unsupervised / semi-supervised probabilistic learning

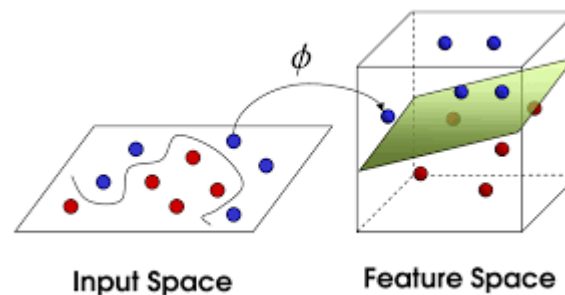
- *Contextual Graph Markov Models*

Bacciu , Errica, Micheli, ICML 2018.



And Kernels?

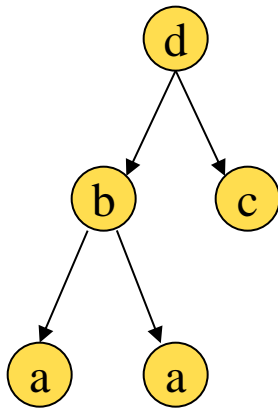
- Kernels for SD from RC [1] and HTMM [2] for SD, e.g. (by CIML):
- Kernels for SD from RC
 1. D. Bacciu, C. Gallicchio, A. Micheli, *A reservoir activation kernel for trees*. ESANN 2016
- Kernels for SD from HTMM (adaptive kernels + generative & discriminative)
 2. D. Bacciu, A. Micheli, A. Sperduti. *Generative Kernels for Tree-Structured Data*, IEEE TNNLS (Transactions on Neural Networks and Learning Systems), 2018



Summarizing the MODELS panorama for SD (examples)

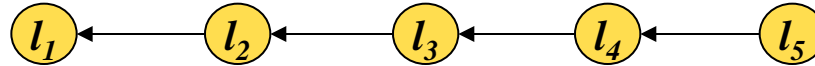


Standard ML models for
flat data

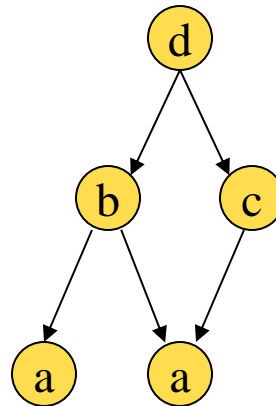


Tree:

- Recursive NN
- Tree ESN
- HTMM
- Tree Kernels
- ...

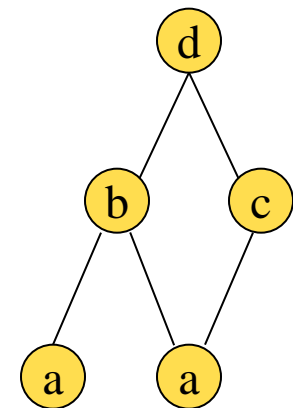


- Recurrent NN/ESN
 - HMM
- Kernel for strings ...



DPAG:

- CRCC



- GNN/GraphESN
 - NN4G
- Graph Kernels
 - SRL
 - ...

See references for models in the bibliography slides (later)

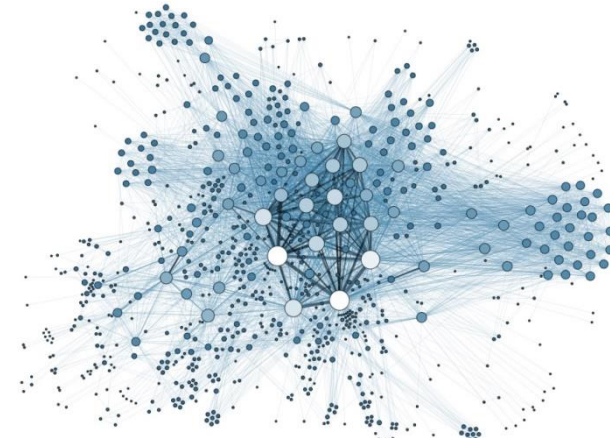
Future – just ahead



Future applications

Under construction! Ready to apply for tree/graph data on:

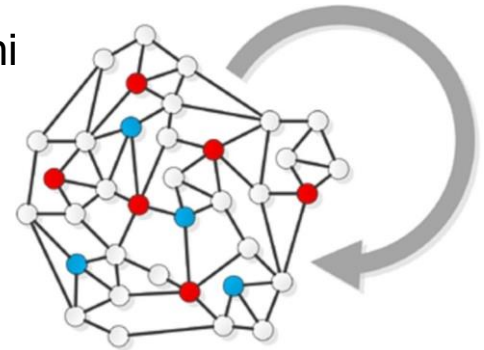
- Network data → next slide
- NLP (e.g. sentiment analysis, Q.C.)
- Images (as trees or graphs on pixel neighbors)
- Parallel programming: Skeleton application description → estimation of execution time and energy
- System biology (graphs/networks)



Processing/Learning Aims

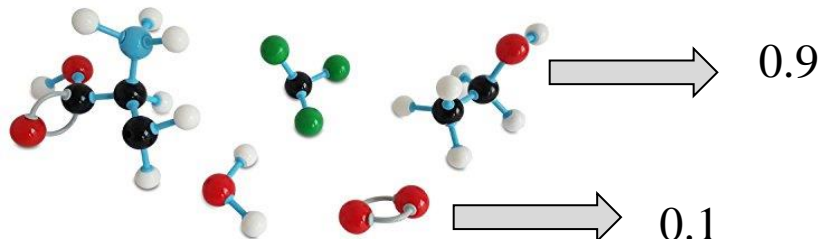
Also task changes according to the data which can be:

- A single graph (typically a network): in-graph learning
 - For instance to classify the nodes of a partially labeled network (as a social network or a graph in semi supervised learning problems)
 - Belong to input-output isomorphic transductions



- A collection of variable size graphs: between-graphs
 - For instance, classify different graphs starting from a training set of know couples as in the molecules example

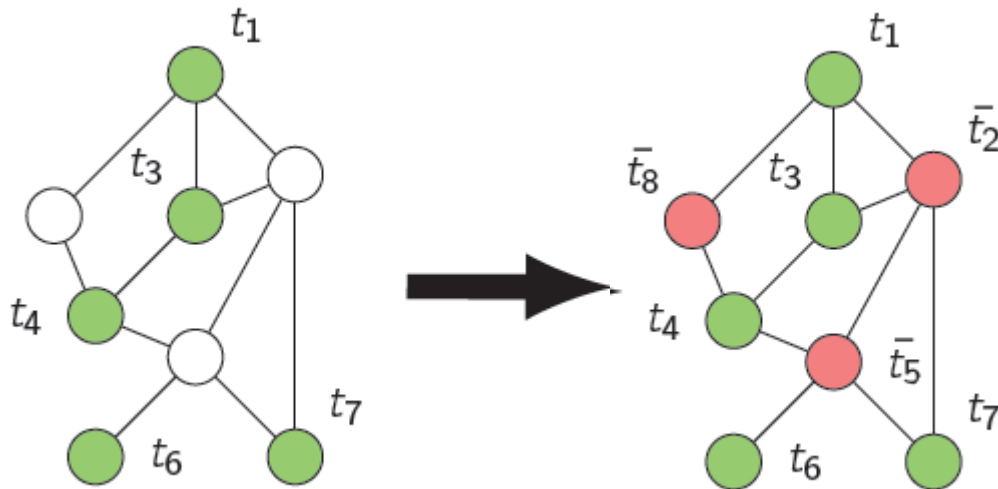
Given a set of examples $(graph_i, target_i)$
Learn an hypothesis mapping $T(graph)$



Networks, an example: *within or in-graph* learning

Network data: entities are interconnected

Within-network learning: training entities are connected to entities whose classifications are to be estimated



An instance of
IO-isomorphic
transduction for graphs!

- Example: web spam (host) classification
- ~10K vertices (hosts) and 500K links (web pages links)
- Classify each vertex as spam/not spam given the target is known for some of them
(*in the same unique graph!*)

A more general aim (CIML -Pisa)

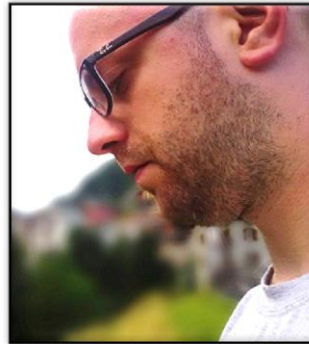
- Adaptive processing of SD
 - without prefixed metrics for SD comparison or encoding
- **A theoretical and practical framework for the automatic design of efficient models** for sequence, trees and graphs (both generative and discriminative) exploiting DL approaches
 - Able to answer the main issue of DL frameworks: how many layers? How many units? Which hyper.? Etc.
 - Open to *semi-supervised* learning and different graph and network tasks
 - *Efficient* by incremental NN and RC approaches



Key People



Alessio Micheli



Luca Oneto



Davide Bacciu



Claudio Gallicchio



Luca Pedrelli

**+ 5 PhD students
on 2018/19**



**Computational Intelligence &
Machine Learning Group**

Bibliography: aims

Different parts in the following:

- Basic/Fundamentals
- * Possible topic for seminars
- May be useful also for future studies
 - Many topics can be subject of study and development
 - Many many works in literature (arrive continuously)!
 - Many possible topics for demand and possible thesis
 - **More bibliography on demand:** micheli@di.unipi.it

Bibliografia (Basic, origins of RecNN)



Dip. Informatica
University of Pisa

RecNN

- A. Sperduti, A. Starita. *Supervised Neural Networks for the Classification of Structures*, IEEE Transactions on Neural Networks. Vol. 8, n. 3, pp. 714-735, 1997.
- P. Frasconi, M. Gori, and A. Sperduti, *A General Framework for Adaptive Processing of Data Structures*, IEEE Transactions on Neural Networks. Vol. 9, No. 5, pp. 768-786, 1998.
- A.M. Bianucci, A. Micheli, A. Sperduti, A. Starita. *Application of Cascade Correlation Networks for Structures to Chemistry*, Applied Intelligence Journal (Kluwer Academic Publishers), Special Issue on "Neural Networks and Structured Knowledge" Vol. 12 (1/2): 117-146, 2000.
- A. Micheli, A. Sperduti, A. Starita, A.M. Bianucci. *A Novel Approach to QSPR/QSAR Based on Neural Networks for Structures*, Chapter in Book : "Soft Computing Approaches in Chemistry", pp. 265-296, H. Cartwright, L. M. Sztandera, Eds., Springer-Verlag, Heidelberg, March 2003.

Bibliography: RecNN approaches-2

* UNSUPERVISED RecursiveNN

- B. Hammer, A. Micheli, M. Strickert, A. Sperduti.
A General Framework for Unsupervised Processing of Structured Data, Neurocomputing (Elsevier Science) Volume 57, Pages 3-35, March 2004.
- B. Hammer, A. Micheli, A. Sperduti, M. Strickert.
Recursive Self-organizing Network Models. Neural Networks, Elsevier Science. Volume 17, Issues 8-9, Pages 1061-1085, October-November 2004.

* TreeESN: efficient RecNN

- C. Gallicchio, A. Micheli.
Tree Echo State Networks, Neurocomputing, volume 101, pag. 319-337, 2013.
- C. Gallicchio, A. Micheli.
Deep Reservoir Neural Networks for Trees. Information Sciences 480, 174-193, 2019.

* HTMM: further developments (generative)

- D. Bacciu, A. Micheli and A. Sperduti.
Compositional Generative Mapping for Tree-Structured Data - Part I: Bottom-Up Probabilistic Modeling of Trees, IEEE Transactions on Neural Networks and Learning Systems, vol. 23, no. 12, pp. 1987-2002, 2012

Bibliography: RecNN applications (example)

* NLP applications (that you can extend with recent instances, and relate them to the general RecNN framework present in this lecture and the basic RecNN bibliography references)

- R. Socher, C.C. Lin, C. Manning, A.Y. Ng,
Parsing natural scenes and natural language with recursive neural networks,
Proceedings of the 28th international conference on machine learning (ICML-11)
- R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C.P. Potts,
Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank
Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1631–1642, Seattle, Washington, USA, 18-21 October 2013

Bibliography: This lecture (I)

Main references



Dip. Informatica
University of Pisa

■ Bidirectional RNN

- M. Schuster, K. Paliwal. "Bidirectional recurrent neural networks." *Signal Processing, IEEE Transactions on* 45(11) (1997): p.p. 2673-2681, 1997
- P. Baldi, et al. "Exploiting the past and the future in protein secondary structure prediction." *Bioinformatics* 15 (11) (1999):p.p. 937-946, 1999
- A. Micheli, D. Sona, A. Sperduti. Bi-causal Recurrent Cascade Correlation, *IJCNN'2000 - Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks* (IEEE Computer Society Press), Volume: 3, 2000, pp. 3-8 , 2000

* RecNN for DPAGs : how to extend the domain (I)

- A. Micheli, D. Sona, A. Sperduti.
Contextual Processing of Structured Data by Recursive Cascade Correlation. *IEEE Transactions on Neural Networks.* Vol. 15, n. 6, Pages 1396- 1410, November 2004.
- Hammer, A. Micheli, and A. Sperduti.
Universal Approximation Capability of Cascade Correlation for Structures. *Neural Computation.* Vol. 17, No. 5, Pages 1109-1159, (C) 2005 MIT press.

Bibliography: This lecture (II)

Main references

* NN for GRAPH DATA: how to extend the domain (II)

- * A. Micheli. *Neural network for graphs: a contextual constructive approach*, IEEE Transactions on Neural Networks, volume 20 (3), pag. 498-511, doi: 10.1109/TNN.2008.2010350, 2009.
- C. Gallicchio, A. Micheli. *Graph Echo State Networks*, Proceedings of the International Joint Conference on Neural Networks (IJCNN), pages 1–8, 2010.
- F. Scarselli, M. Gori, A.C.Tsoi, M. Hagenbuchner, G. Monfardini. *The graph neural network model*, IEEE Transactions on Neural Networks, 20(1), pag. 61–80, 2009.
- F. Scarselli, M. Gori, A.C.Tsoi, M. Hagenbuchner. *The Vapnik–Chervonenkis dimension of graph and recursive neural Networks*. Neural Networks, Vol. 108, 248-259, 2018.

Bibliography: This lecture (III)

Other References

Convolutional Neural Networks for Graphs

- Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pp. 2224–2232, 2015.
- Henaff, M.; Bruna, J.; LeCun, Y. Deep Convolutional Networks on Graph-Structured Data. ArXiv e-prints, 2015.
- Atwood, J. and Towsley, D., 2016. Diffusion-convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1993-2001). <https://arxiv.org/abs/1511.02136>
- Niepert, M., Ahmed, M., and Kutzkov, K. Learning convolutional neural networks for graphs. In *International Conference on Machine Learning*, pp. 2014–2023, 2016.
- Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, pp. 3844–3852, USA, 2016. Curran Associates Inc., 2016.
- Kipf, T. N.; Welling, M. Semi-supervised classification with graph convolutional networks. *Proceedings of ICLR 2017*, 2017.
- Hamilton, W.L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017.
- Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P., 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* 34 (4), 18–42, 2017.
- Bacciu, D., Errica, F., Micheli, A. "Contextual Graph Markov Model: A Deep and Generative Approach to Graph Processing" *ICML 2018*, Vol. 80. 294-303. <https://arxiv.org/abs/1805.10636>
- ..., ..., ... **continuously coming!**

DRAFT, please do not circulate!

For information

Alessio Micheli

micheli@di.unipi.it

www.di.unipi.it/groups/ciml



Dipartimento di Informatica
Università di Pisa - Italy



**Computational Intelligence &
Machine Learning Group**