Deep Learning for Graphs

University of Pisa

Alessio Micheli

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

Davide Bacciu

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

ECML-PKDD 2018 Tutorial

14th September 2018

Dipartimento di Informatica
Università di Pisa - Italy

Computational Intelligence & Machine Learning Group

Ourself: CIML and Learning in SD

University of Pisa

- Learning in Structured Domains (SD)
  - Pioneering since the 90's the development of Recursive NN for **trees** and **graphs**
  - New models by RC, Kernels and generative approaches
  - Deep learning for SD
- Interdisciplinary applications (med., bio., chem., eng., robotics, …)

Alessio Micheli

Davide Bacciu

Claudio Gallicchio

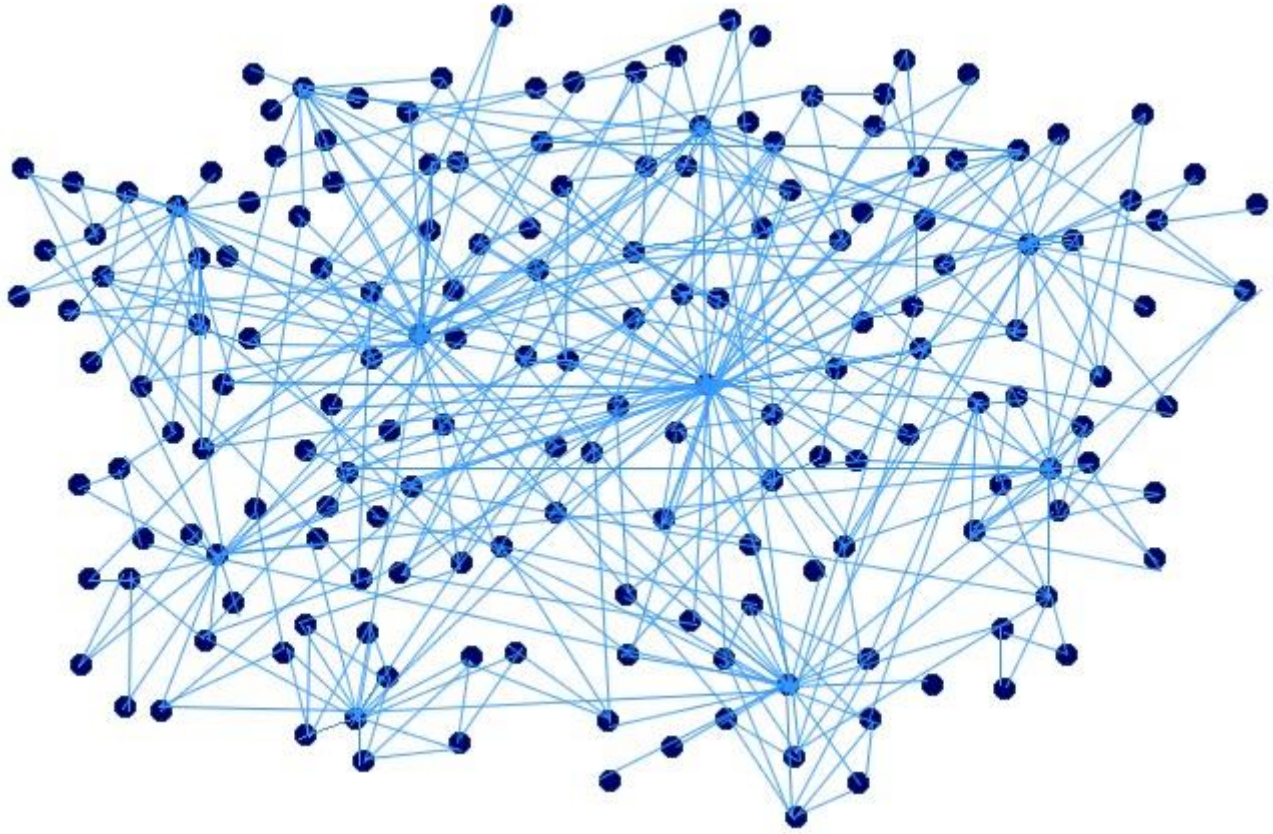
Computational Intelligence & Machine Learning Group

www.di.unipi.it/groups/ciml

Motivations

University of Pisa

Why structured data ?



Because data have relationships

Motivations: tutorial aim

University of Pisa

A top-down view considering both:

1. foundational works on structured data developed in the last 20 years, whose knowledge and understanding is currently not widely diffused in the community
   - highlighting concepts that can help the current research

2. (integrating them with) the latest current edge results of deep learning research

DLSD@ECML-2018:
https://sites.google.com/view/dl4sd

Deep Learning for Graphs
Plan in 2 lectures

1. **Foundational models:**
   Intro to Learning in Structured Domains
   - Extensions of original flat models to supervised and unsupervised learning in structured domains: from vectors to graphs

2. **Deep learning models**
   - Current view of deep learning for graph and network data

DLSD@ECML-2018:
https://sites.google.com/view/dl4sd

Learning in Structured Domain

1. Foundational models

- Introduction to structured data processing
- Recursive models:
  - From sequences to trees
  - Moving to DPAG the role of causality
- Learning variable-size graphs with cycles
  - Contractive encodings approaches for graphs
  - Contextual Multi-Layered approaches for graphs

By a journey through the causality assumption!

Introduction: ML for SD

University of Pisa

- Most of known ML methods are limited to the use of flat and fixed form of the data (*fixed-length attribute-value vectors*)

- Central: *data representation*
- Graph: very useful abstraction for real data

- **Labeled graphs** = vector patterns + relationships
  - **natural**: for structured domain
  - **richness**
  - **efficiency**: repetitive nature inherent the data

- **SD + ML =** adaptive processing of structured information
  - Structured domain learning/ Learning in Structured Domain
  - Relational Learning
  - Structure/Graph Mining
    - Molecule Mining
  - ***… Deep Learning for Graphs***

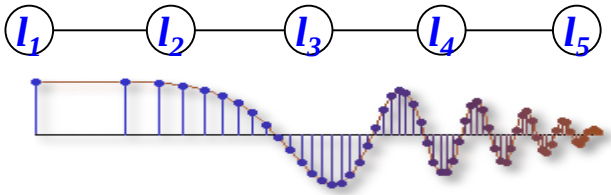
From flat to structured data

University of Pisa

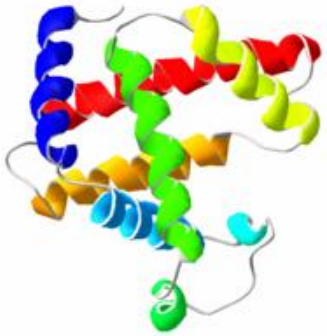
- **Flat:** standard vectors of features
- **Structured:** Sequences, trees, graphs, multi-relational data

stringa_in_italiano

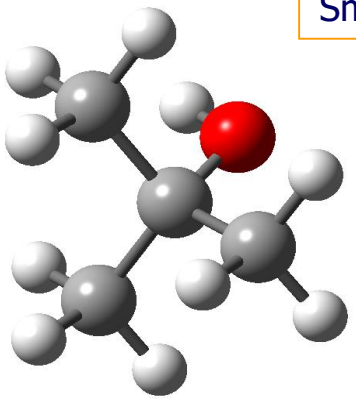
Strings



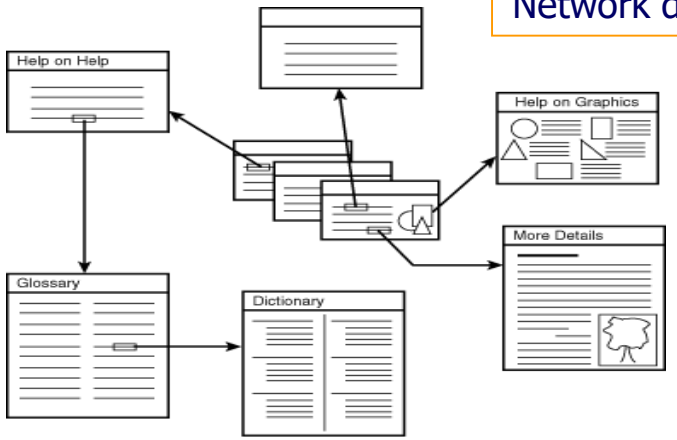
Series/
temporal stream



Proteins



Small molecules

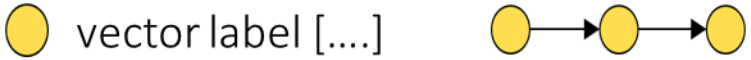


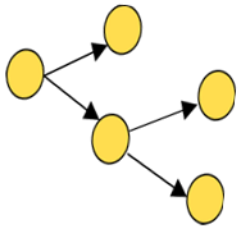
Network data

The scenario, terms (and trends)

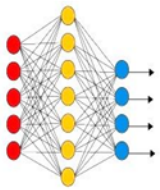
University of Pisa

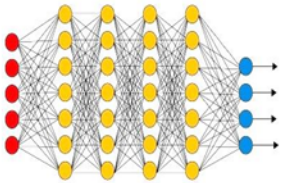
Input Domain		
Flat (vectors) (e.g. F.F.NN)	**Sequences** (RNN)	**Structures** (trees/graphs)





Layering	
Shallow	**Deep**





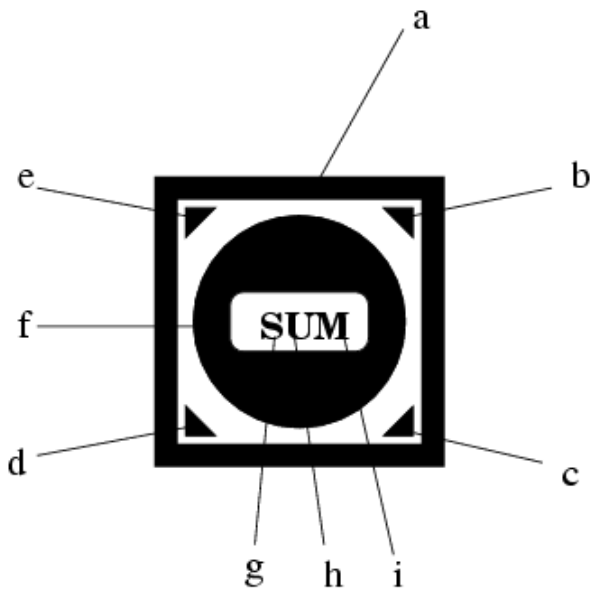
Multi-level input rep. abstraction

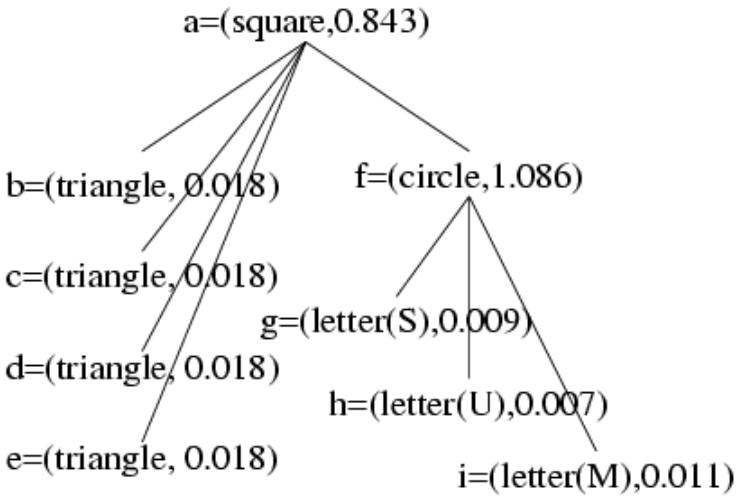
Focus

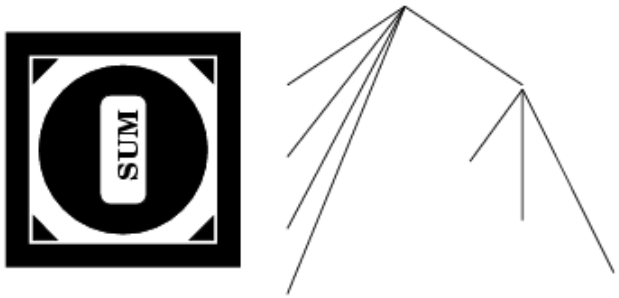
"Deep and wide" approaches

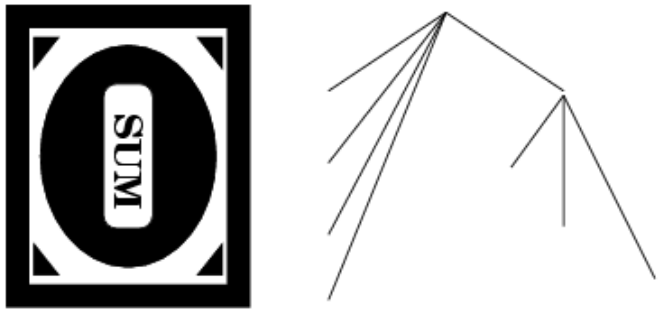
Example: logo recognition

University of Pisa



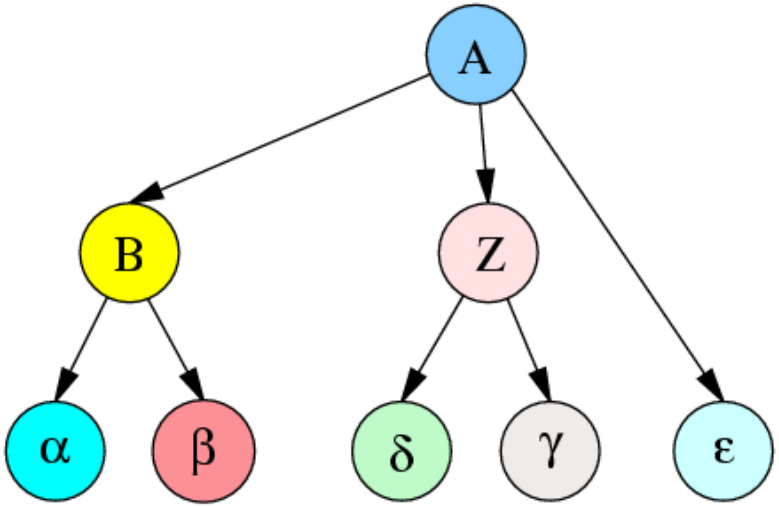




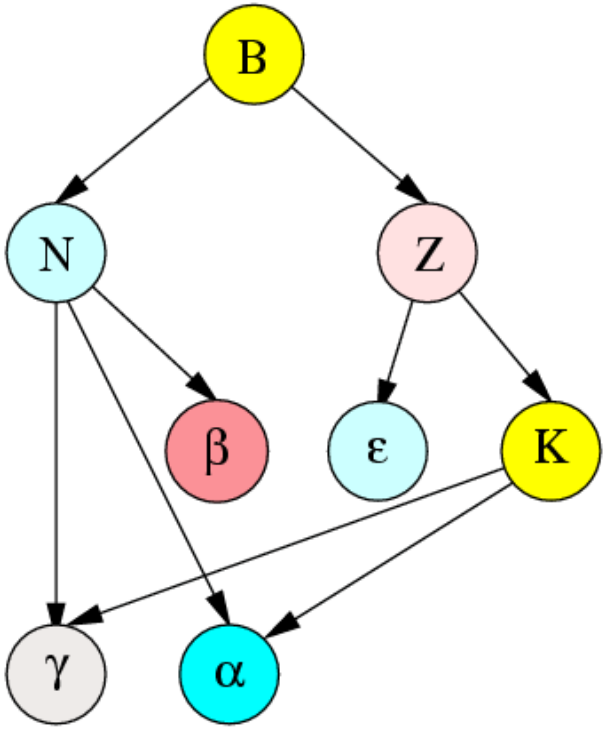


Example: Terms in 1st order logic

University of Pisa



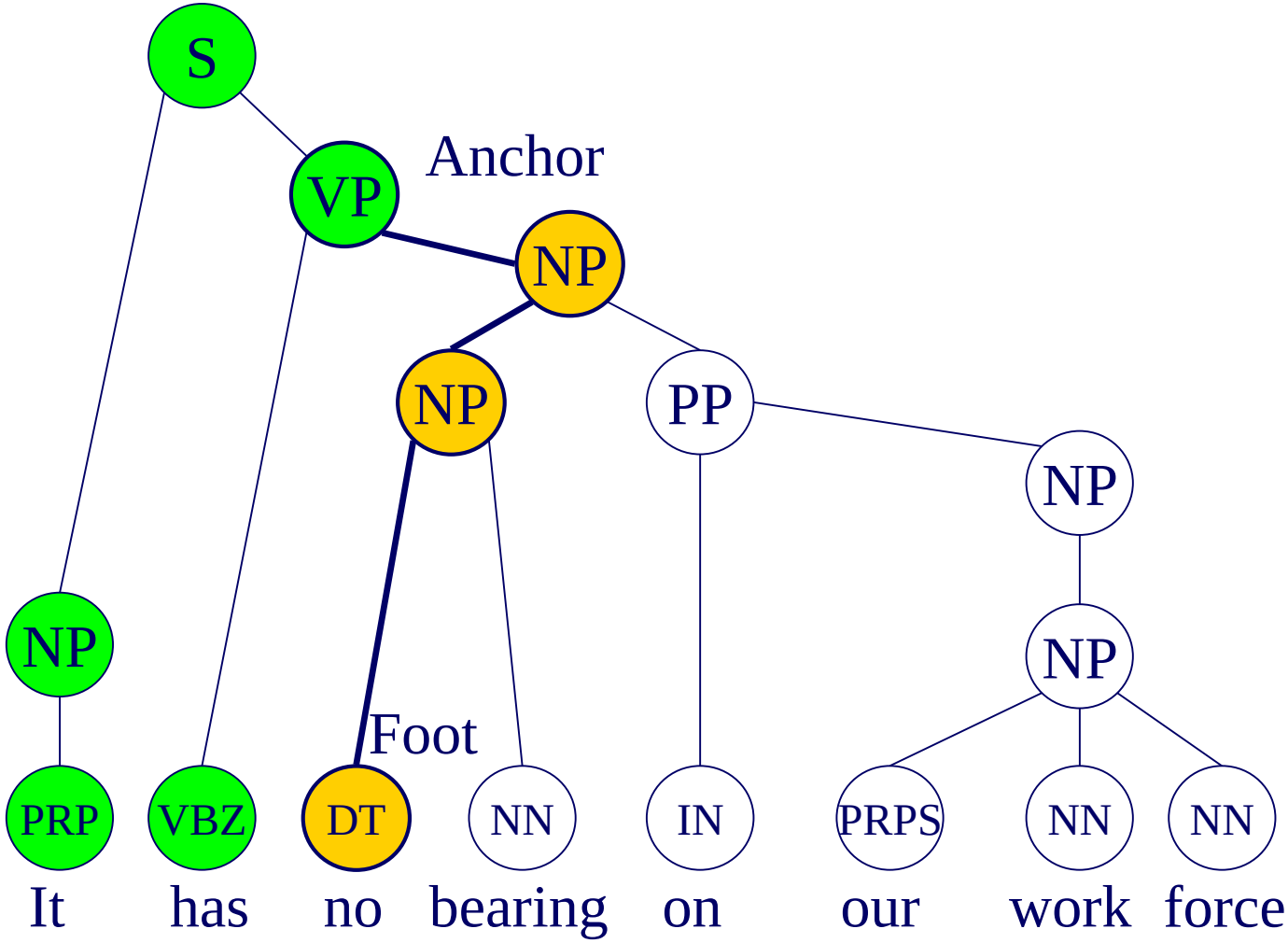
A(B(α,β),Z(γ,δ),ε)



B(N(γ,α,β),Z(ε,K(γ,α)))

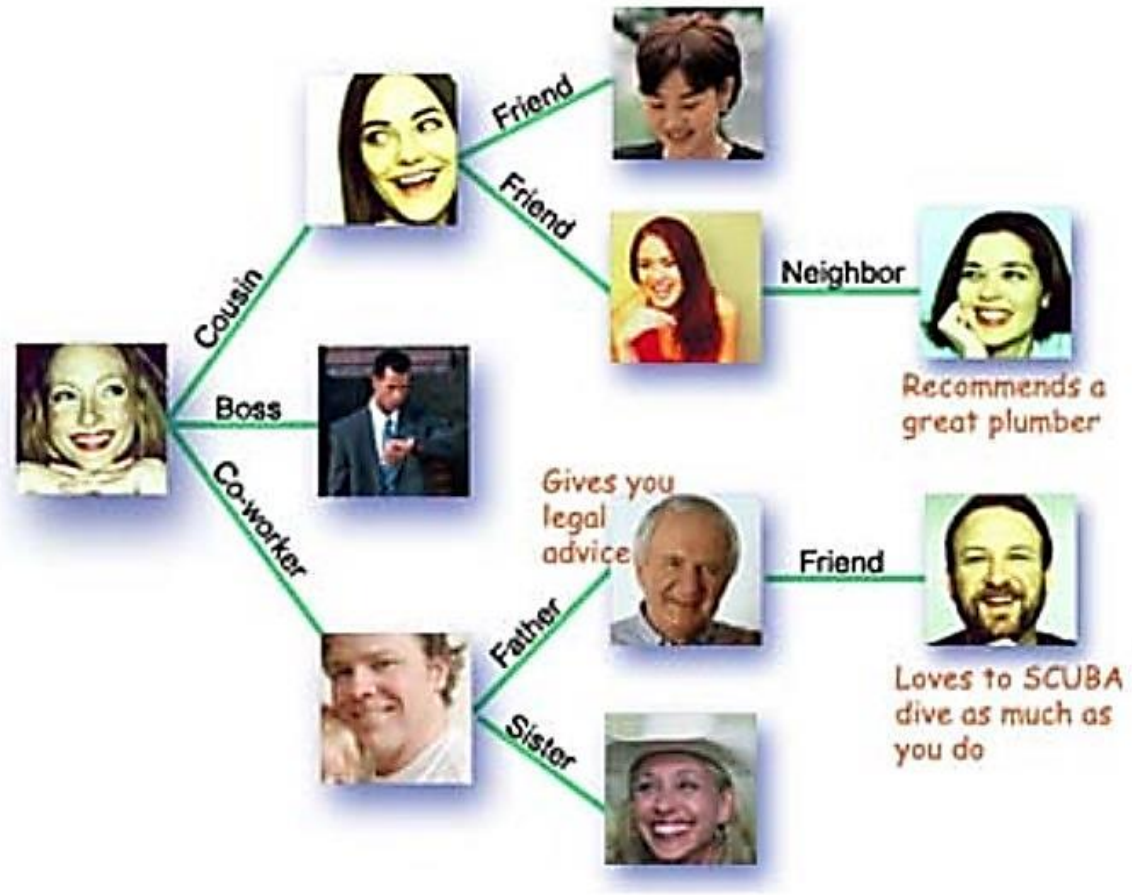
Example: language parsing

University of Pisa



Example: Social Networks

University of Pisa

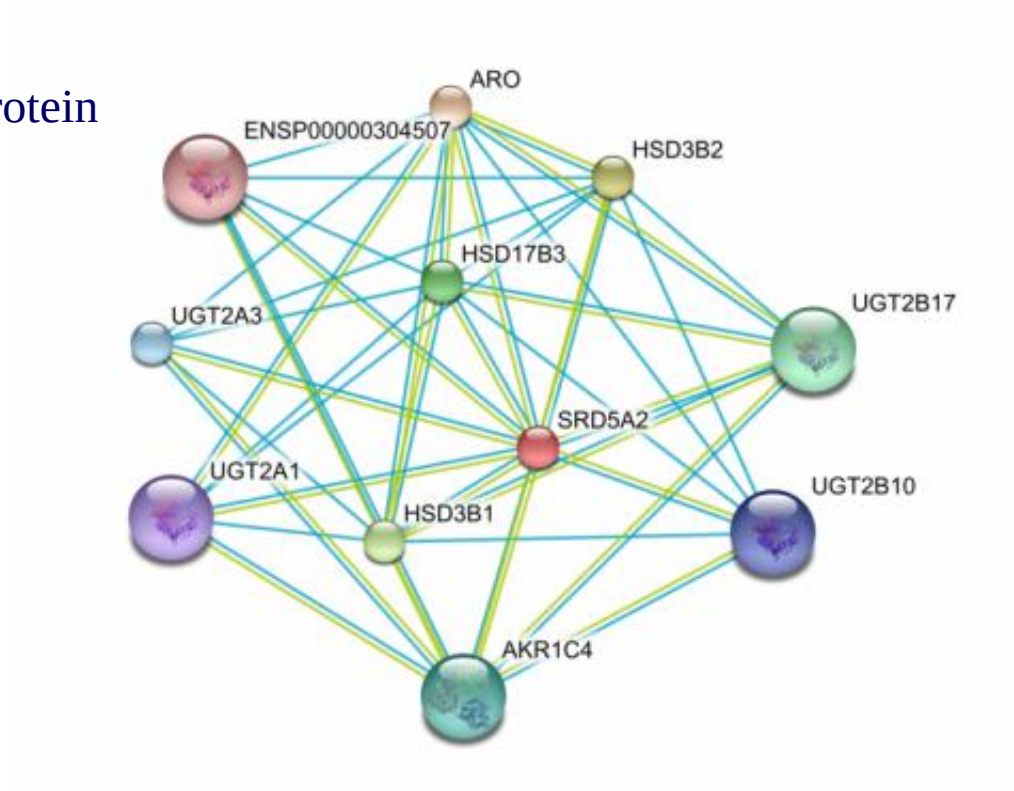


Example: Biological Networks

University of Pisa

- Node for protein.
- Link for **interaction** or similarity

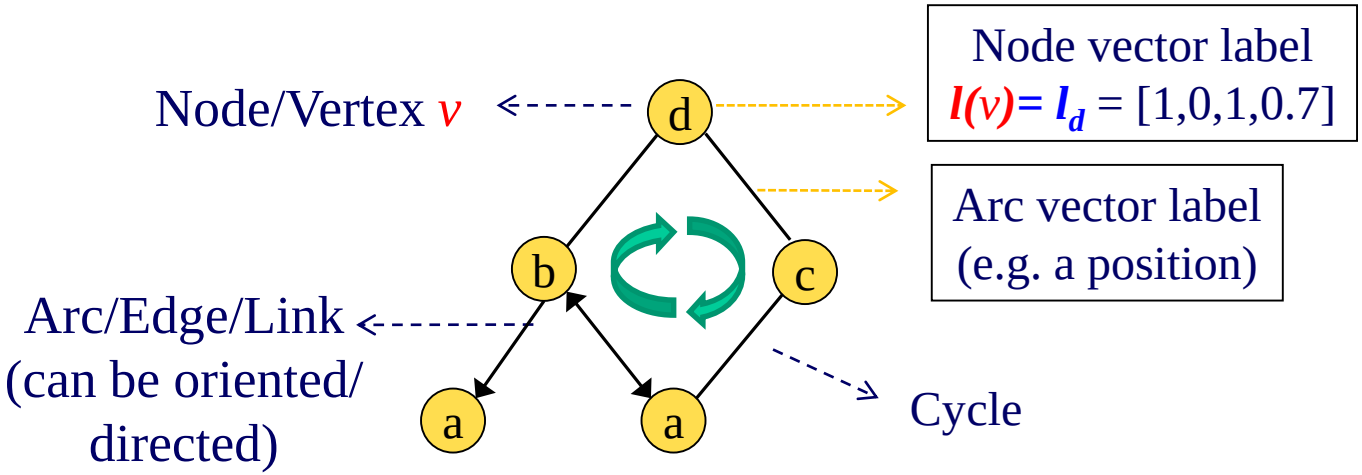
Protein



Our graphs (in the following)

University of Pisa

Labeled graphs *g*

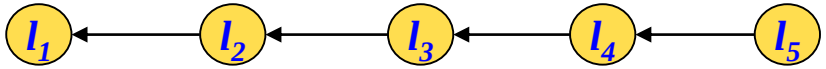


Structured Data: Classes

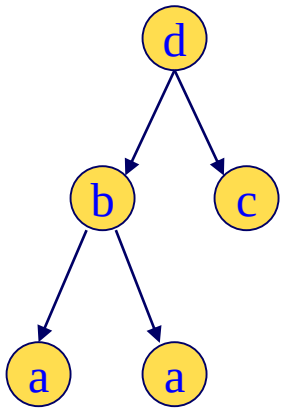
University of Pisa

- Labeled Sequences, Trees, DOAGs- DPAGs, graphs

Single labeled vertex

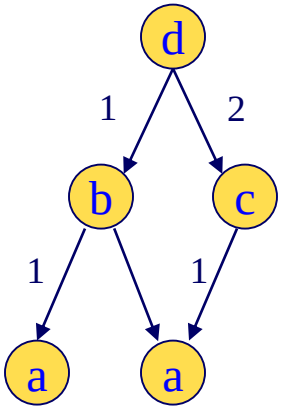


Sequence

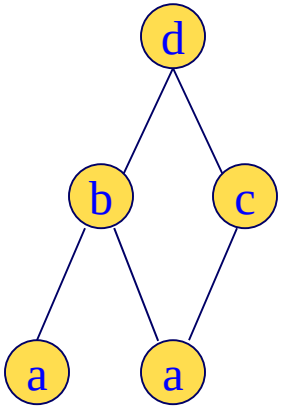


Rooted Tree

Supersource



DPAG



Graph (undirected)

K-ary tree: root and bounded *out-degree* (*k*) for each node (# of chlidren).

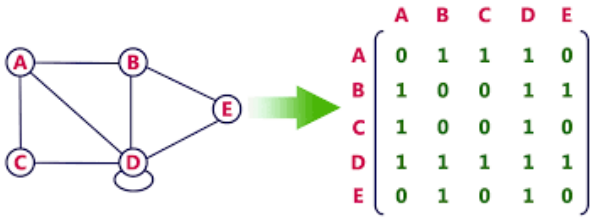
DPAG: labeled direct positional acyclic graphs with *supersource*, bounded *in-degree* and *out-degree* (*k*).

Graph Representations

University of Pisa

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)



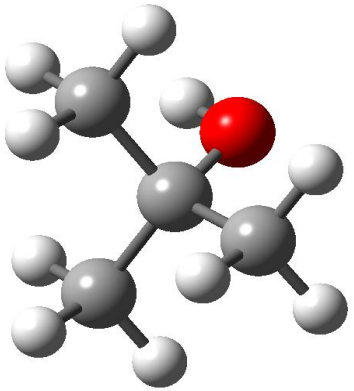
- ML issues for the high proportion between combinatorial number of possible data examples and available data

- "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."

Processing/Learning Aims

University of Pisa

- 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*).



*Property/Activity = **T**(Structure)*

T

Property Value (regression)
Toxic (yes/no) (classification)

QSPR: Correlate chemical structure of molecules with their properties

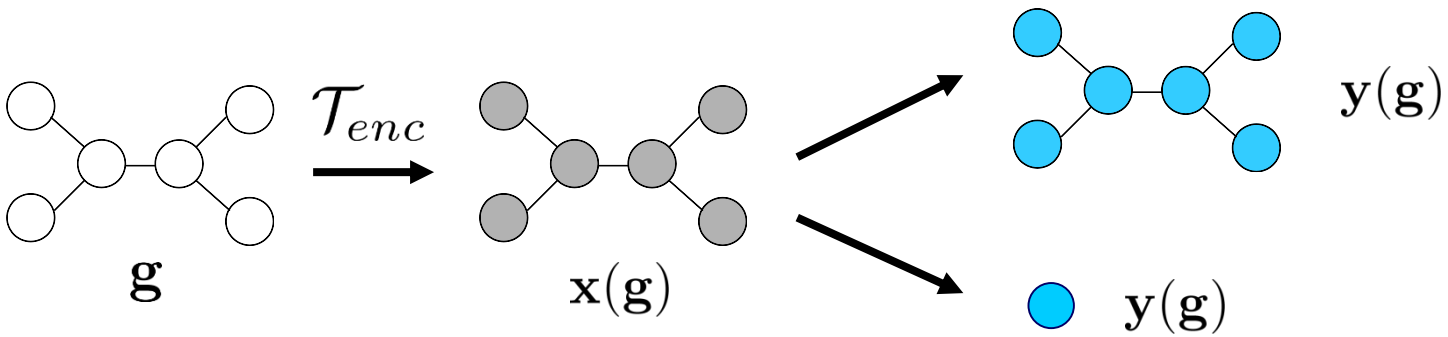
Molecules can be more naturally represented by varying size ***structures***

Can we predict directly from structures ?

Processing/Learning Aims

University of Pisa

- A more general **trasduction** can be either



Structure-to-Structure
(input-output isomorphic)

Input graph

Possible internal representation/encoding

Node/Graph embedding

Structure-to-Scalar/Element
(regression/classification)

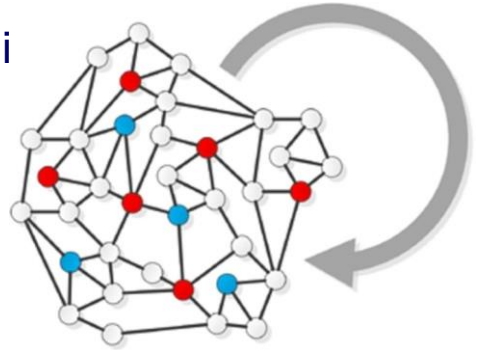
Or (in general) it can also be **non-isomorphic**

Processing/Learning Aims

University of Pisa

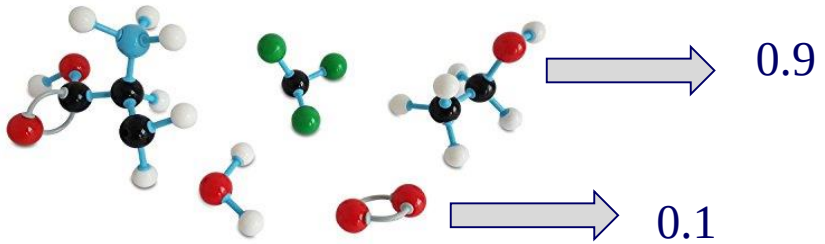
Also task changes according to the data which can be:

- 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 hyphotesis mapping $T(graph)$



Learning Models for SD

University of Pisa

- 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

Guidelines Concepts

University of Pisa

- We will follow both an historical perspective
- And we will instantiate for transductions concepts of:

- **Compositionality (structured encoding):** processing of structures and context of vertices by the hierarchical *composition* of its sub-structures representation (to deal with variable/arbitrary size structures)
- **Parsimony:** fix the number of free parameters independently from structure's size (assuming some *stationary* conditions)
- **Adaptivity:** the transduction can be tailored to the data and task at hand

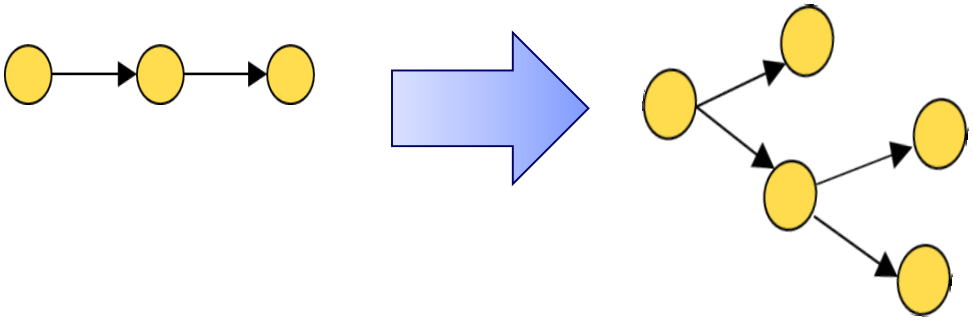
Learning in Structured Domain

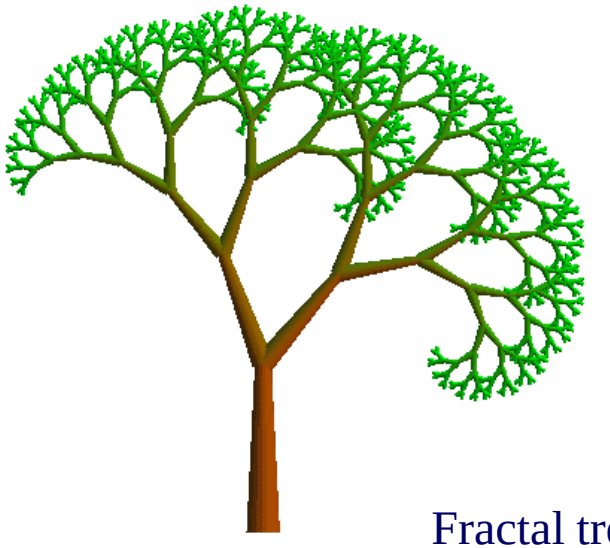
1. Foundational models

- Introduction to structured data processing
- Recursive models:
  - From sequences to trees
  - Moving to DPAG the role of causality
- Learning variable-size graphs with cycles
  - Contractive encodings approaches for graphs
  - Contextual Multi-Layered approaches for graphs

By a journey through the causality assumption!

From Sequences to Trees in Neural Networks





Recurrent / Recursive NN:

- ***Recursive*** and parametric realization of the **transduction** function (node embedding by neural state machines)
- *Concept of stationarity and casualty*
- *Adaptivity* by Neural Networks

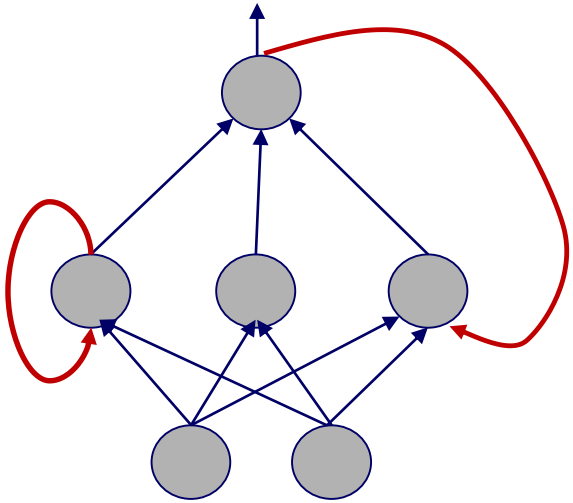
Fractal tree: a recursive structure

Feedforward versus Recurrent

University of Pisa

- **Feedforward NN**: direction: input→ output
- **Recurrent** neural networks: A different category of architecture, based on the addition of *feedback loops* connections in the network topology
  - The presence of **self-loop** connections provides the network with dynamical properties, letting a memory (**state**) of the past computations in the model.
  - This allows us to extend the representation capability of the model to the processing of sequences (and structured data).

Recurrent Neural Networks (RNN):



Recurrent Neural Networks

University of Pisa

- RNN: a state-transitions system with free-parameters

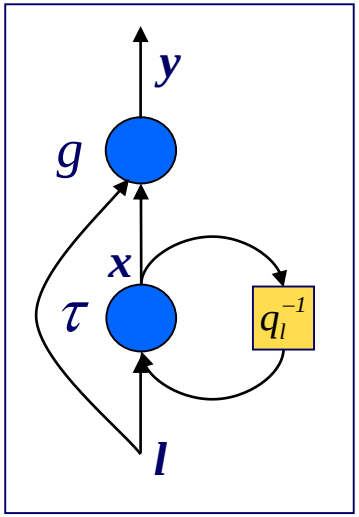
Given x(0) =0

internal state

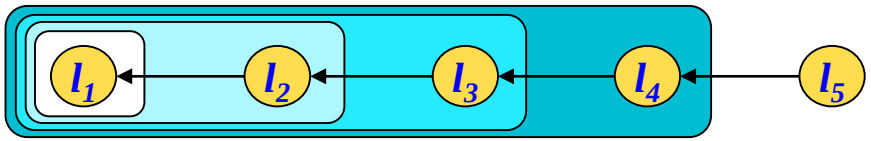
$$\begin{cases} \boldsymbol{x}(t) = \tau(\boldsymbol{x}(t-1), \boldsymbol{l}(t)) \\ \boldsymbol{y}(t) = g(\boldsymbol{x}(t), \boldsymbol{l}(t)) \end{cases}$$

The **state** summarize the past information (*context or node/sub-sequences embedding*)

τ is the state transition (next-state) function realized by a NN using *weights* and sigmoidal functions



self-loop/ feedbacks connection



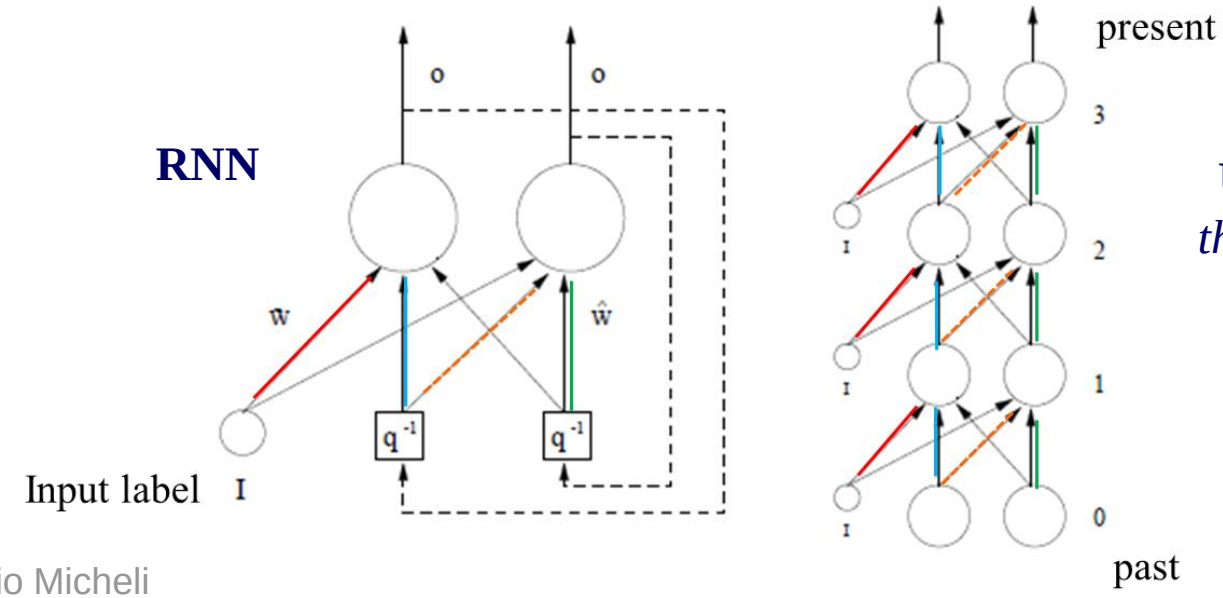
State formation over time

Graphical model (**RNN**, **HMM**, ...)

RNN properties (assumptions)

University of Pisa

- Causality (for **compositionality**): A system is causal if the output at time t_0 (or vertex v) only depends on inputs at time $t<t_0$ (depends only on v and its descendants)
  - necessary and sufficient for *internal state*
- Stationarity (for **parsimony**): time invariance, state transition function τ is independent on node v (the same in any time) → *weight sharing* in the unfolding version
- Adaptivity: transition functions are realized by NN (with free weight parameters), hence they are **learnt** from data
- Universal approximation capibility and Turing equivalence



Unfolding
through time

Training: BPTT
(Back-Prop
Through Time)

Deep?

RNN nowadays

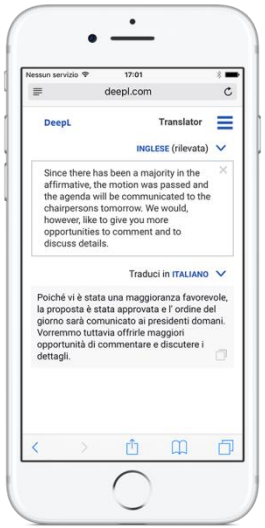
University of Pisa

Many architectural variants.

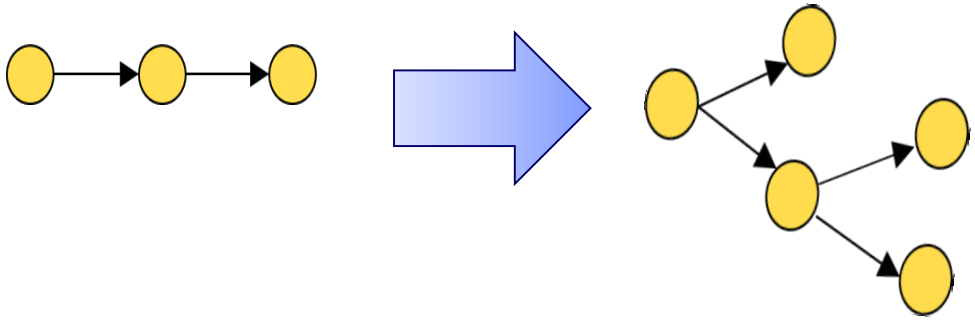
- NARX, LSTM, GRU, ESN (RC), ...
- Deep RNN and DeepESN

It is reference approach for sequence processing, especially for (state-of-the-art results):

- Speech recognition/processing
- Text processing
- Machine translation etc.…
- and also Generating text/speech/music …

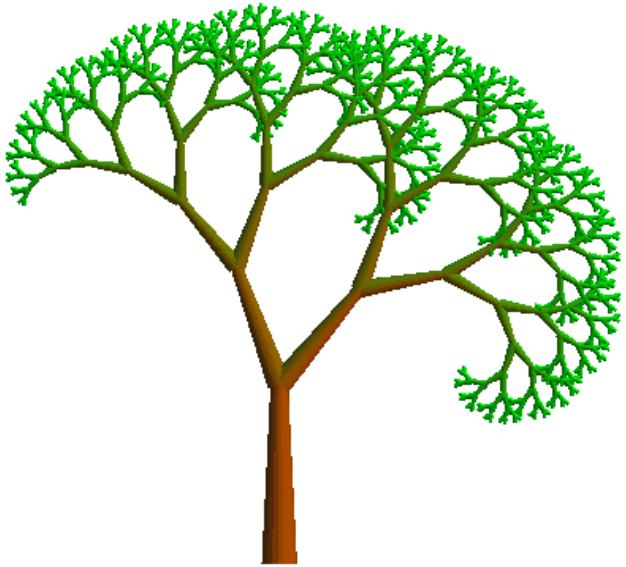


From Sequences to Trees in Neural Networks



Trees

- Representing hierarchical information in many real-world domains
- Many examples seen above:
  - Molecular Biology,
  - Document (XML) Processing,
  - Natural Language Processing
  - …

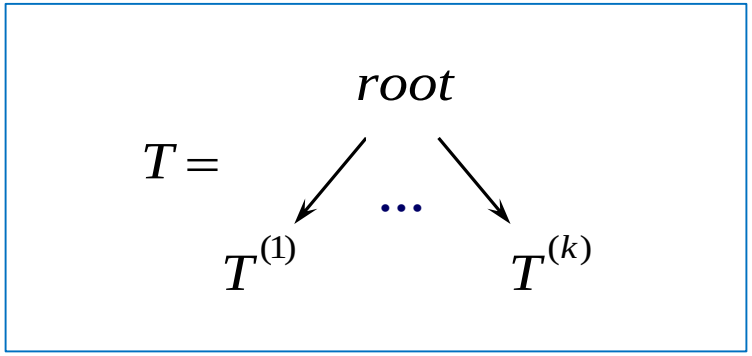


Fractal tree: a recursive structure

K-ary Trees

University of Pisa

- ***k-ary trees*** (*trees* in the following) are rooted positional trees with finite out-degree k.
- Given a node v in the tree $T \in \boldsymbol{G}$:
  - The children of v are the node successors of v, each with a position $j=1,\ldots k$;
  - k is the maximum out-degree over G, i.e. the maximum number of children for each node;
  - $L(v)$ in is the input label associated with v.
  - The *subtree* $T^{(j)}$ is a tree rooted at the j-th children of v.



RecNN

University of Pisa

- The RecNN approach has a long history, which has its roots in the first proposals of models and applications (about 20 years ago) [1, 2, 3, 4]
- Up to composing a *framework* that includes different paradigms of learning and applicative areas.
- The common idea underlying all these variants is to deal with hierarchical structured data extending RNN by a recursive computation implemented by (embedding nodes by states of) a state-transitions system with free-parameters.

1. Sperduti, Starita. IEEE TNN, 1997.
2. Frasconi et al. IEEE TNN, 1998
3. Bianucci et al. Applied Intelligence, 2000
4. More at wikipedia (**Recursive neural network**)

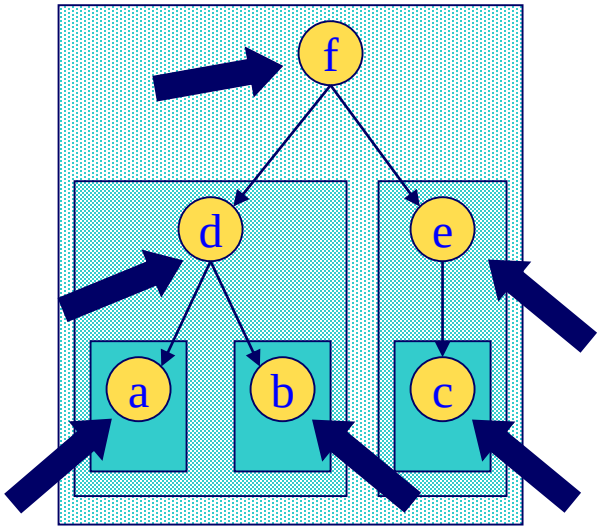
RecNN – State Trans. System

University of Pisa

Given x(leaf) =0

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

State transition system (tree automata) for node (*sub-trees*) embedding



Unfolding the encoding process through structure

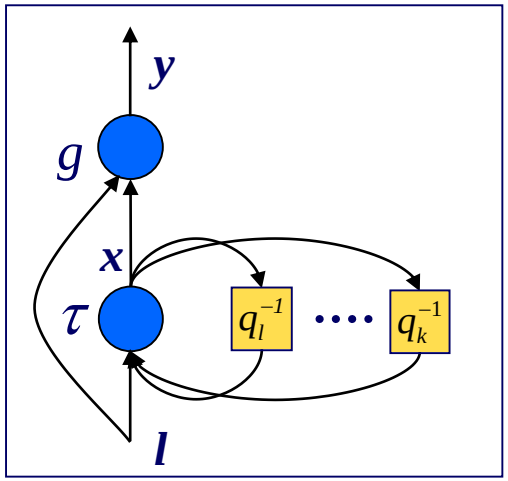
Node/Graph embedding

$$\boldsymbol{x}(\mathrm{ch}[v]) = \boldsymbol{x}(\mathrm{ch}_1[v]), \ldots, \boldsymbol{x}(\mathrm{ch}_k[v])$$

State for children

The context is for sub-trees

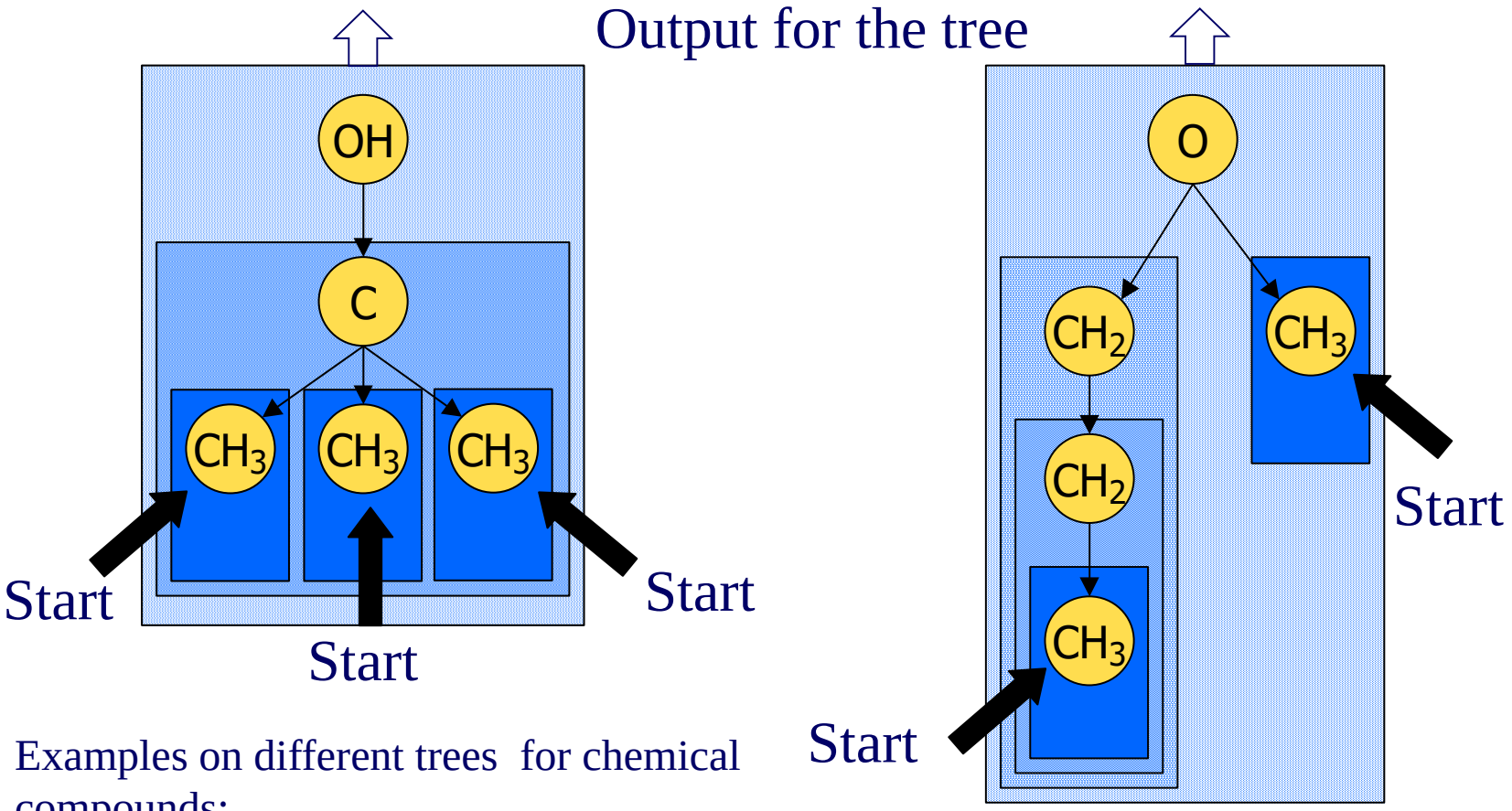
Graphical model for trees



 # feedbacks = # children

RecNN over different structures

University of Pisa



Examples on different trees for chemical compounds:
Unfolding through different structures.

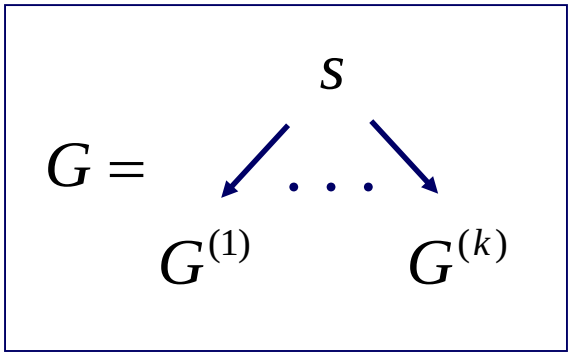
Recursive Transduction View

University of Pisa

Recursive definition of τ_E (encoding function)

Node/Graph embedding

$\boldsymbol{x}(root) = \tau_E(G)$



s can be either a root for a tree or a super-source for a DPAG

$$\tau_E(G) = \begin{cases} \boldsymbol{0} & \text{if } G \text{ is empty} \\ \tau_{NN}(L_{root}, \tau_E(G^{(1)}), \ldots, \tau_E(G^{(k)})) \end{cases}$$

τ_E : systematic visit of $G \rightarrow$ it guides the application of τ_{NN} to each node of tree (bottom-up). *Causality* and *stationary* assumption.

RecNN Properties (I)

University of Pisa

Extending the RNN properties:

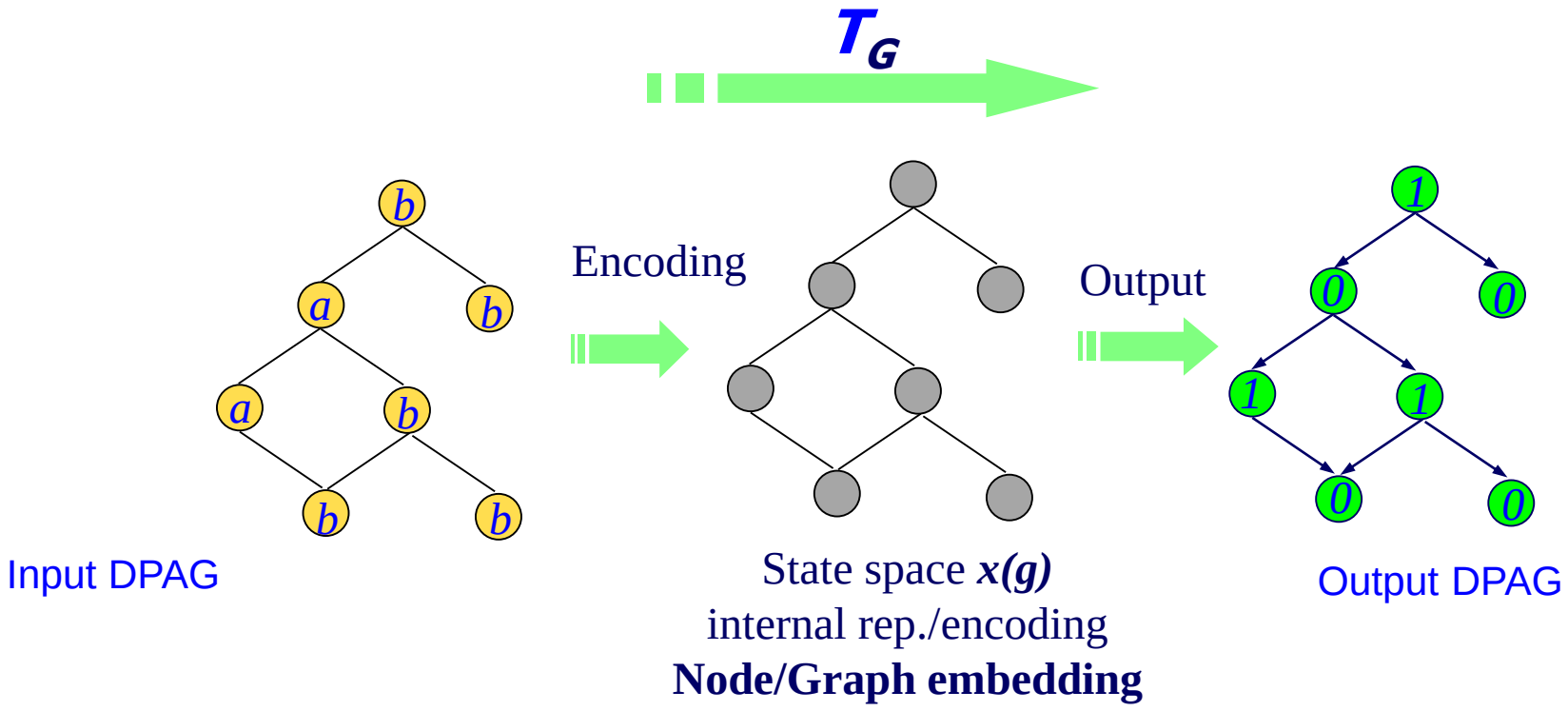
- **Causality (for Compositionality):** the output for a vertex v only depends on v and its descendants (*induced sub-graphs*)
- **Stationary (for Parsimony)**: state transition function τ (τ_{NN}) is independent on vertex v
- **Adaptivity** : NN. learn. Alg.
  + Universal approximation over the tree domain [Hammer 2005-2007]

Frasconi et al. IEEE TNN, 1998
Hammer. Springer, 2007

Properties (II): graphical view

University of Pisa

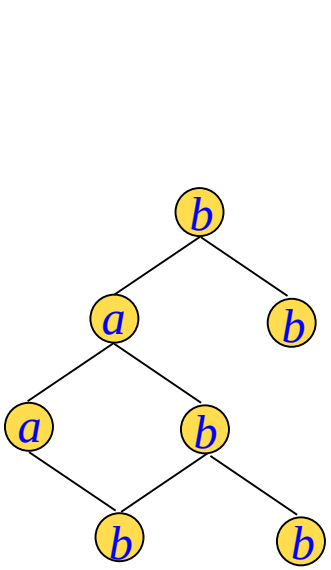
- T_G is *IO-isomorph* if G and $T_G(G)$ have the same skeleton (graph after removing labels)



Properties (II): graphical view

University of Pisa

- T_G *Supersource* transductions

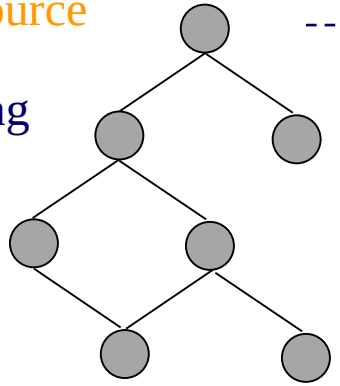


Input DPAG

T_G

Supersource

Encoding



State space ***x(g)***
internal rep./encoding
Node/Graph embedding

Output

y(s)

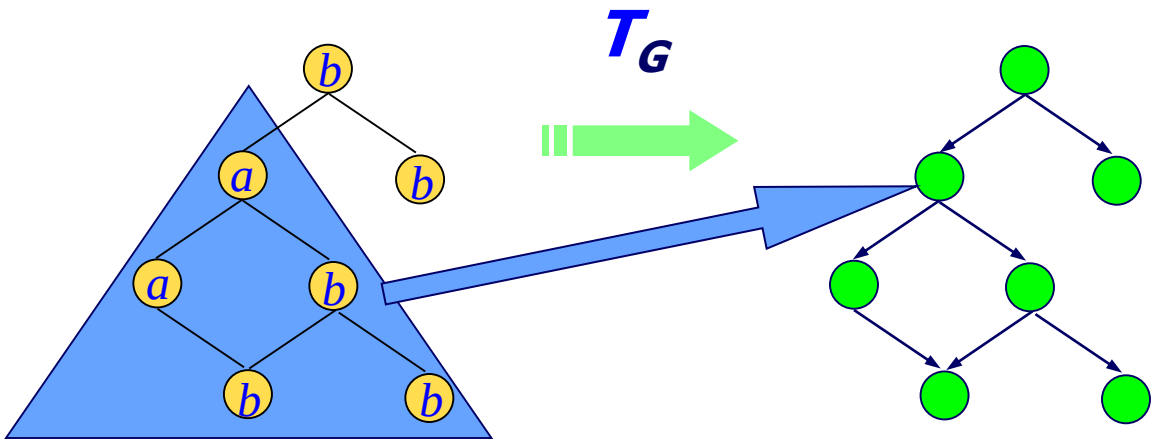
Scalar Value

Also knows as
Structure-to-Element
Supersource (*s*)
transductions

Properties (III)

University of Pisa

- IO-isomorph **causal** transduction



Only the sub-structure is considered

RecNN Training

University of Pisa

Extension of the RNN training algorithms, over the RecNN unfolding networks:

- Back-Propagation Through Structures
- RTRL
- …

- Efficiency (it scales with number of vertices)

- See main references (at the end)

RecNN Recent Applications:

University of Pisa

- Currently successful applications in NLP
  (e.g. by the Stanford NLP group)

- Shown the effectives of RecNN applied to tree representation of language (and images) data and tasks.
  Started in 2011-13

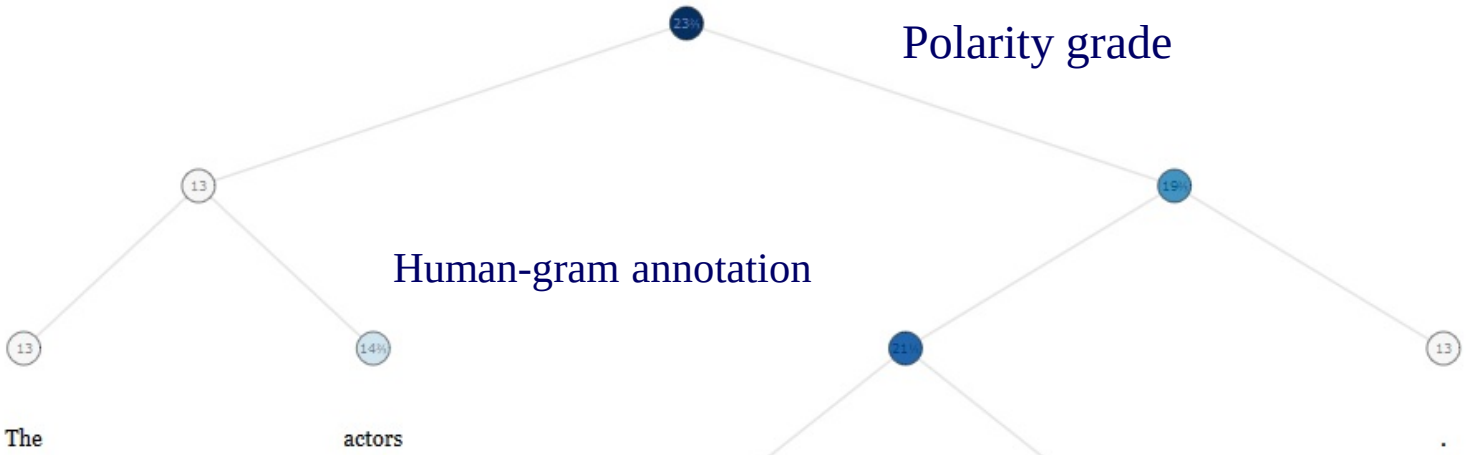
- E.g. Sentiment Treebank
  - Sentiment labels (movies reviews) for 215,154 phrases in the **parse trees** of **11,855** sentences
  - Recursive NN pushes the state of the art in single sentence positive/negative classification from 80% up to 85.4%.

Socher et al. ICML, 2011
Socher et al. EMNLP, 2013

Examples
Sentiment Treebank Parse Trees

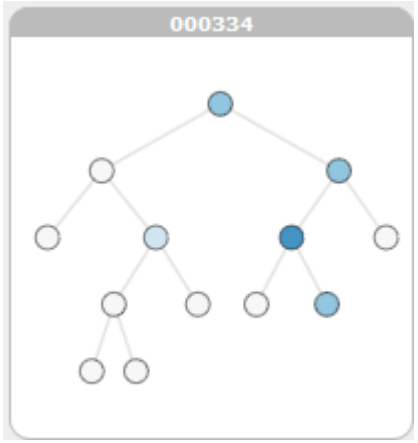
University of Pisa

000027



Polarity grade

Human-gram annotation



...

Other instances in the dataset

The Recursive Approach Family

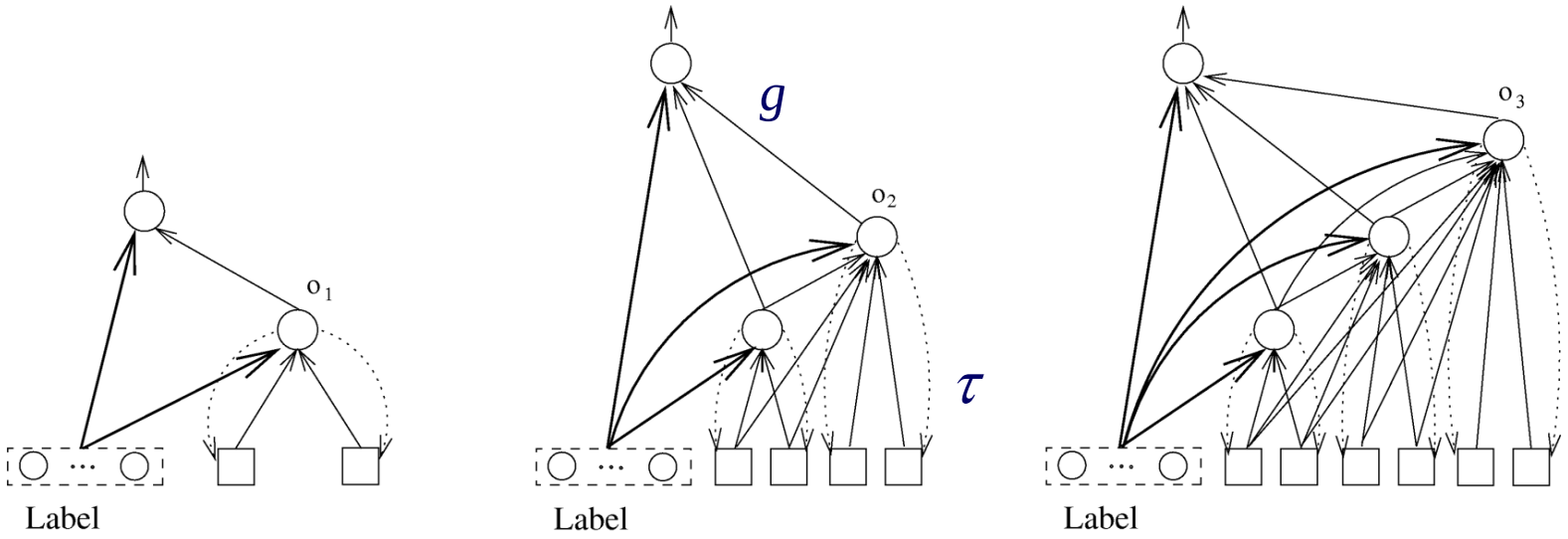
- A framework sharing the idea of a recursive transduction over hierarchical data
- A very rapid overview of many models! (details stay in the slide and in the reference papers)

Costructive: RecCC (1997/2000)

University of Pisa

Recursive Cascade Correlation (RecCC)

- RecNN by RecCC : constructive approach →
  - It adds a new layer for each training step (interleaving output and hidden units training)
  - The number of layers is automatically computed by the training algorithm.
- Nowadays view: Build a *Deep RecNN Model* by constructions

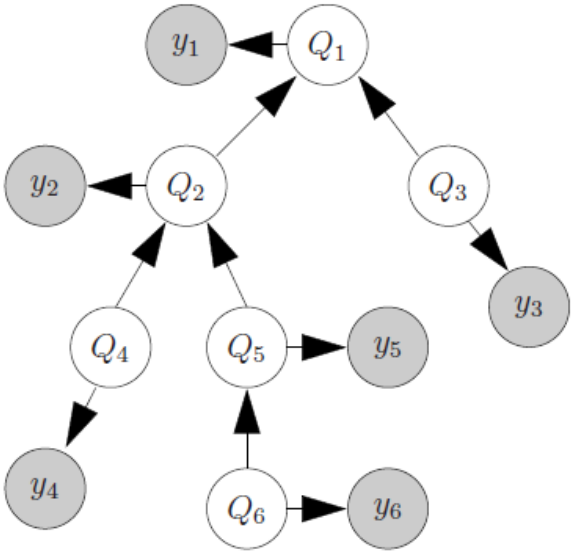


Sperduti, Starita. IEEE TNN, 1997.
Bianucci et al. Applied Intelligence, 2000

Generative: HTMM (2012-2018)

University of Pisa

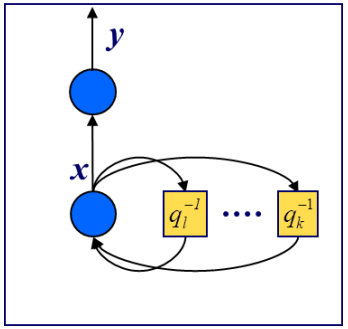
- E.g Bottom-up Hidden Tree Markov Models extend HMM to trees exploiting the recursive approach



Bayesian network unfolding graphical model over the input trees; y: observed elements
Q: hidden states variables with discrete values

- Generative process from the leaves to the root
- Markov assumption (conditional dependence)
  $Qch1(u), \ldots, Qch_K(u) \rightarrow Qu$

Children to parent hidden state transition
$P(Qu \mid Qch1(u), \ldots, Qch_K(u))$



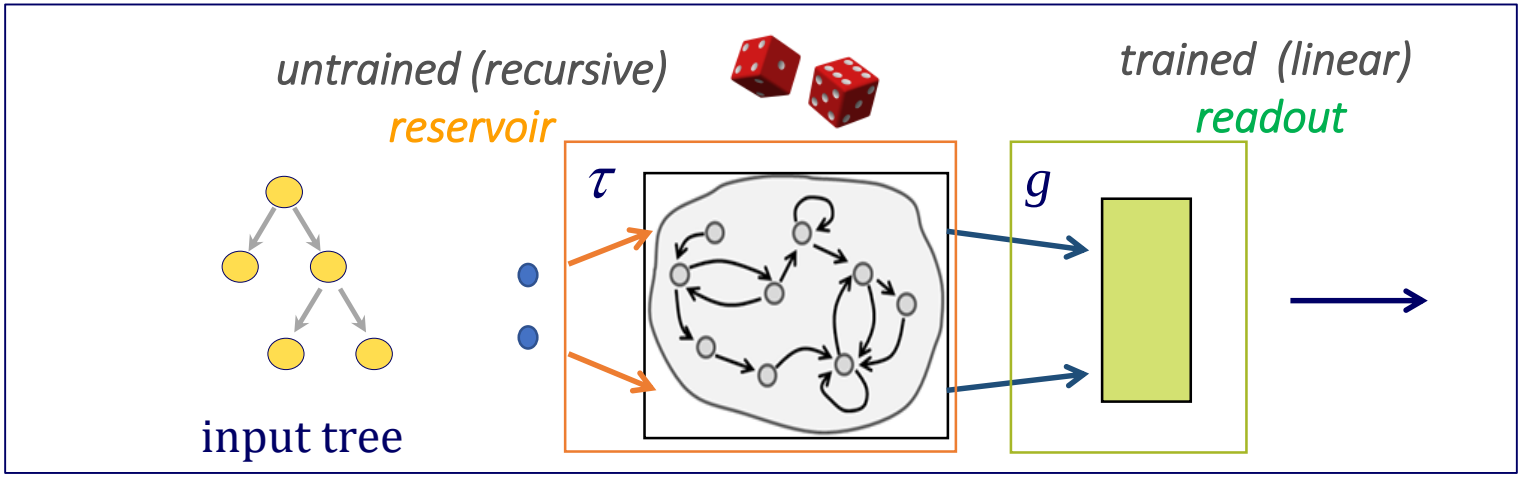
Issue: how decompose this joint state transition? (see ref.).

Bacciu, Micheli, Sperduti. IEEE TNNLS, 2012

Efficient: TreeESN (2010-13)

University of Pisa

- Combine Reservoir Computing (un-trained layer of recurrent units with linear redout) and recursive modeling
  - Extend the applicability of the RC/ESN approach to tree structured data
  - Extremely efficient way of modeling RecNNs (randomized approaches)
  - Architectural and experimental performance baseline for trained RecNN models with often comptetive results.



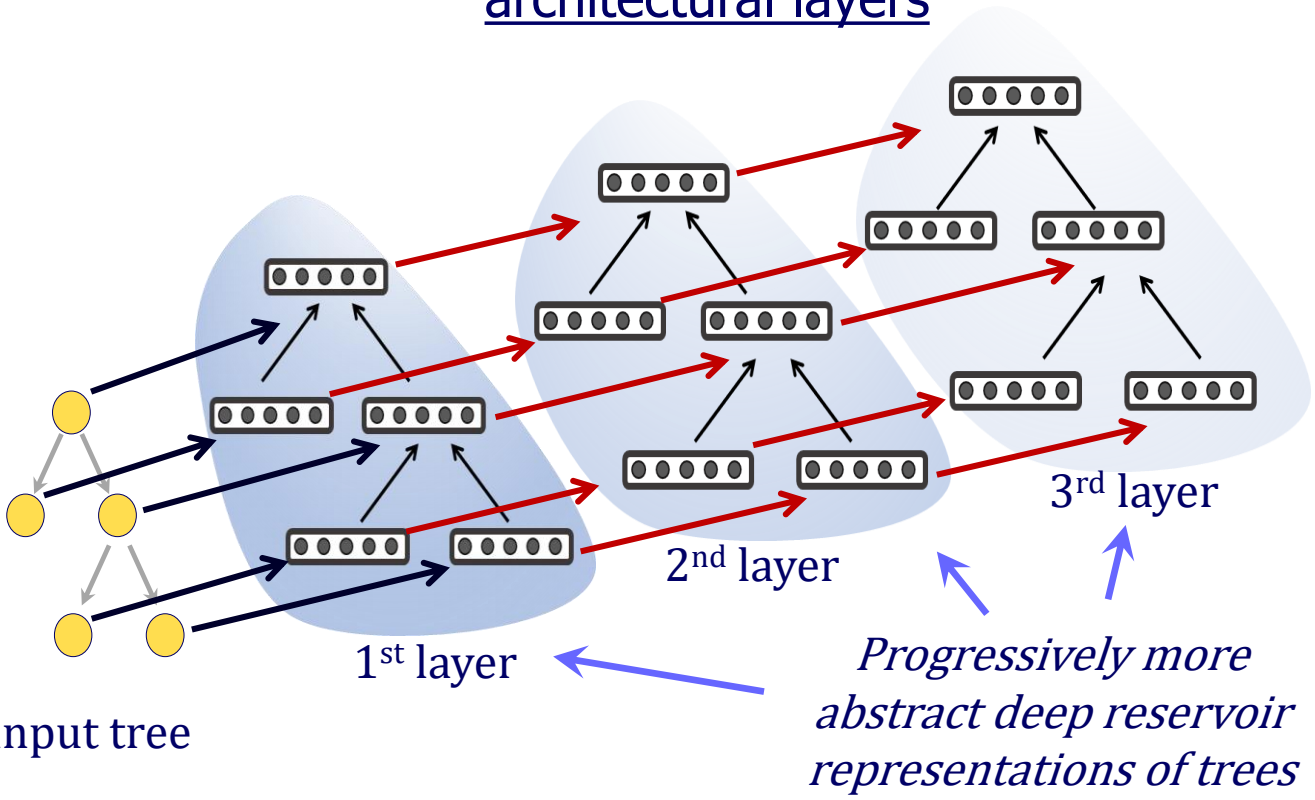
The recursive process of RecNN made by efficient RC approaches

C. Gallicchio, A. Micheli. Neurocomputing, 2013.

Deep: Deep Tree ESN (2018)

University of Pisa

Hierarchical abstraction **both** through the input structure and architectural layers



C. Gallicchio, A. Micheli IEEE IJCNN 2018

- Improve efficiency (giving same #units)
- Improve results

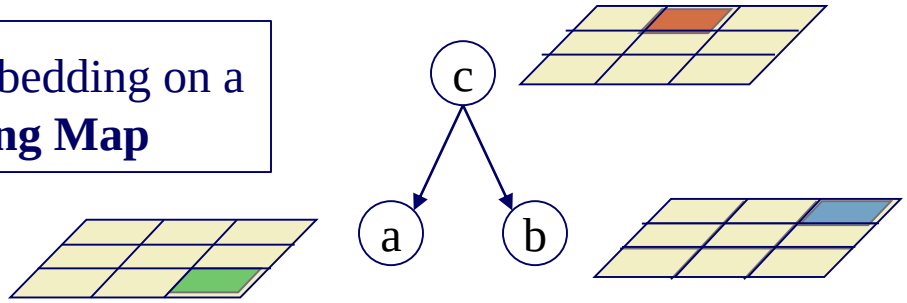
Unsupervised recursive models (2003-2005)

University of Pisa

- Transfer *recursive* idea to unsupervised learning
- No prior metric/pre-processing (but still bias!)
- Evolution of the similarity measure through *recursive comparison* of sub-structures

- Iteratively compared via bottom-up encoding process

Recursive nodes embedding on a **Self-Organizing Map**



It uses e.g. the SOM coordinates for node embedding

M. Hagenbuchner et al. IEEE TNN, 2003
B. Hammer et al. Neural Networks, 2005

RecNN Analysis: Assumptions and open problems

University of Pisa

Inherited from time sequence processing:

- Allow adaptive representation of SD
  - handling of variability by causality and stationarity
- **Stationarity**:
  - efficacy solution to parsimony without reducing expressive power
- **Causality:** affects the computational power !
  - RNN are only able to memorize past information
  - RecNN outputs depend only on sub-structures
  - The domain is restricted to sequences and trees due to causality
- Toward partial relaxation (or **extension**) of the causality assumption

Learning in Structured Domain

1. Foundational models

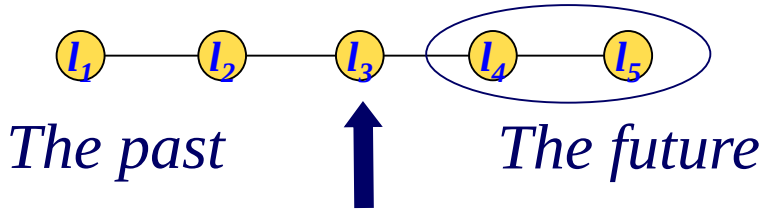
- Introduction to structured data processing
- Recursive models:
  - From sequences to trees
  - Moving to DPAG the role of causality
- Learning variable-size graphs with cycles
  - Contractive encodings approaches for graphs
  - Contextual Multi-Layered approaches for graphs

By a journey through the causality assumption!

Drawbacks of Causal Systems

University of Pisa

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



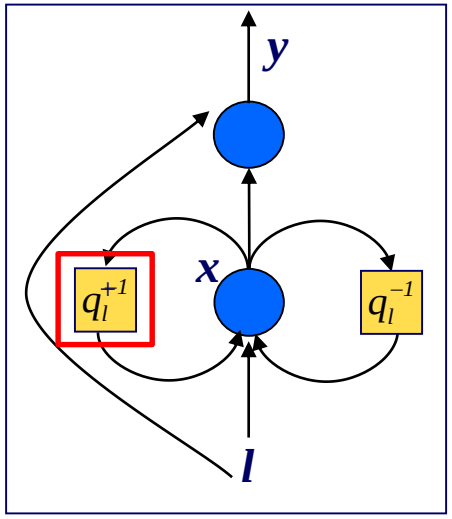
Causility hampers to consider the right part

- 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" to SD!*

Bi-causal System

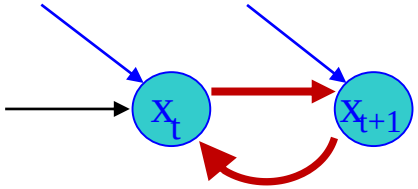
University of Pisa

- A possible bi-causal model can be



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

Unfolding
with cylces

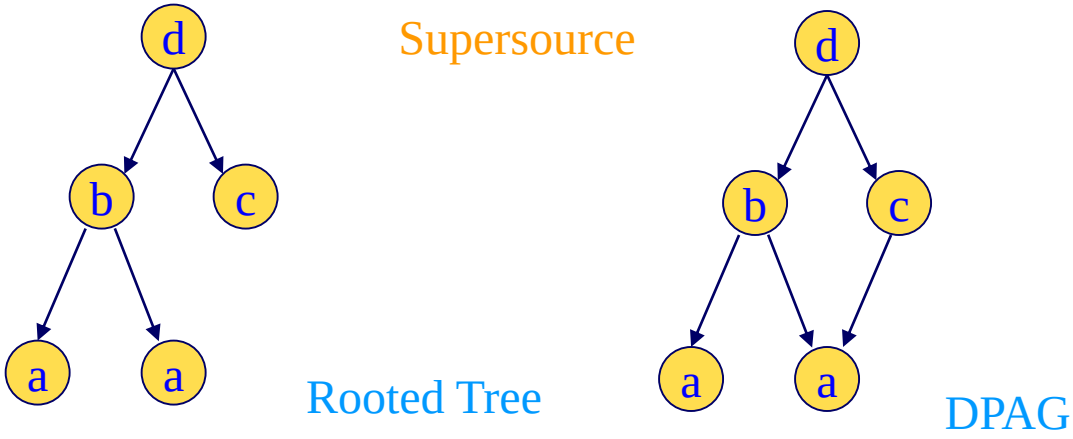


- However this is not easily implementable
  - Cycles: State equations and enc. net. become dynamical systems due to mutual dependencies
  - Different solutions are available (e.g. bidirectional approaches for RNN using a different state for left-to-right o right-to-left encoding)

Overcome the Causality Assumption for SD

University of Pisa

- For structures, let us start with the meaning of contextual transduction
- The CRCC: **Contextual** Recursive Cascade Correlation: Moving from trees to DPAGs

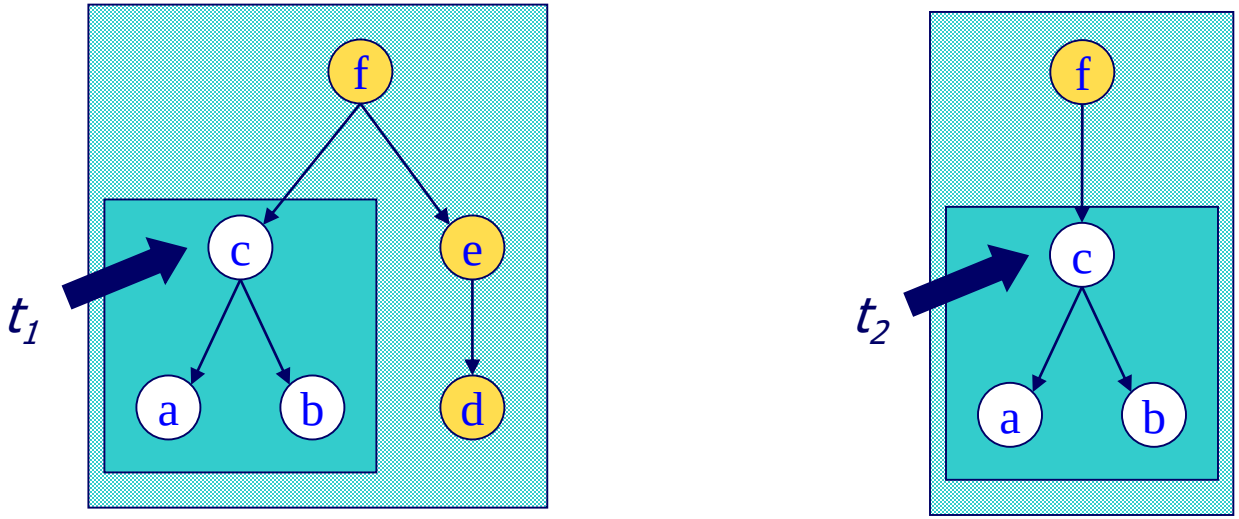


Contextual Target Functions

University of Pisa

- Relevance of contextual processing (I)

contextual IO-isomorphic transductions



$$\text{Target}(t_1) \neq \text{Target}(t_2)$$

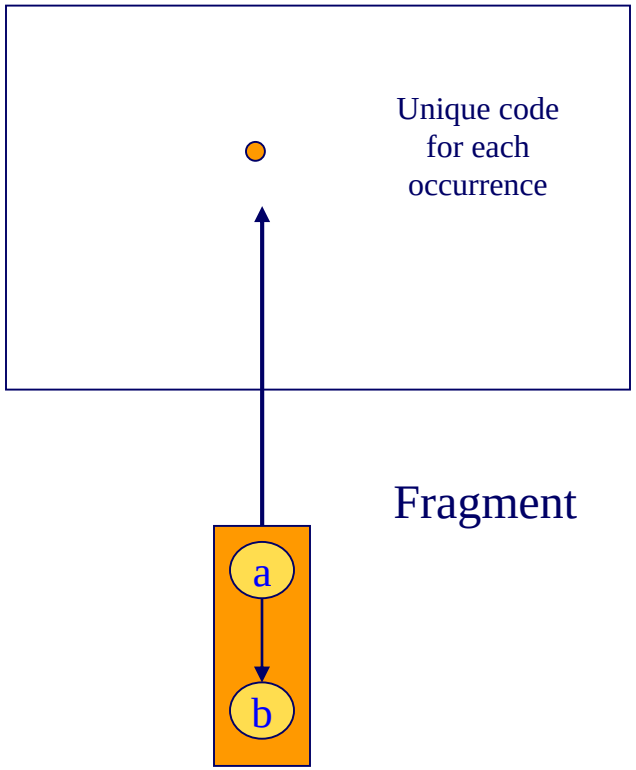
$$\text{out}_{\text{RecNN}}(t_1) = \text{out}_{\text{RecNN}}(t_2)$$

$$\boldsymbol{C}(x_k(c_1)) \neq \boldsymbol{C}(x_k(c_2))$$

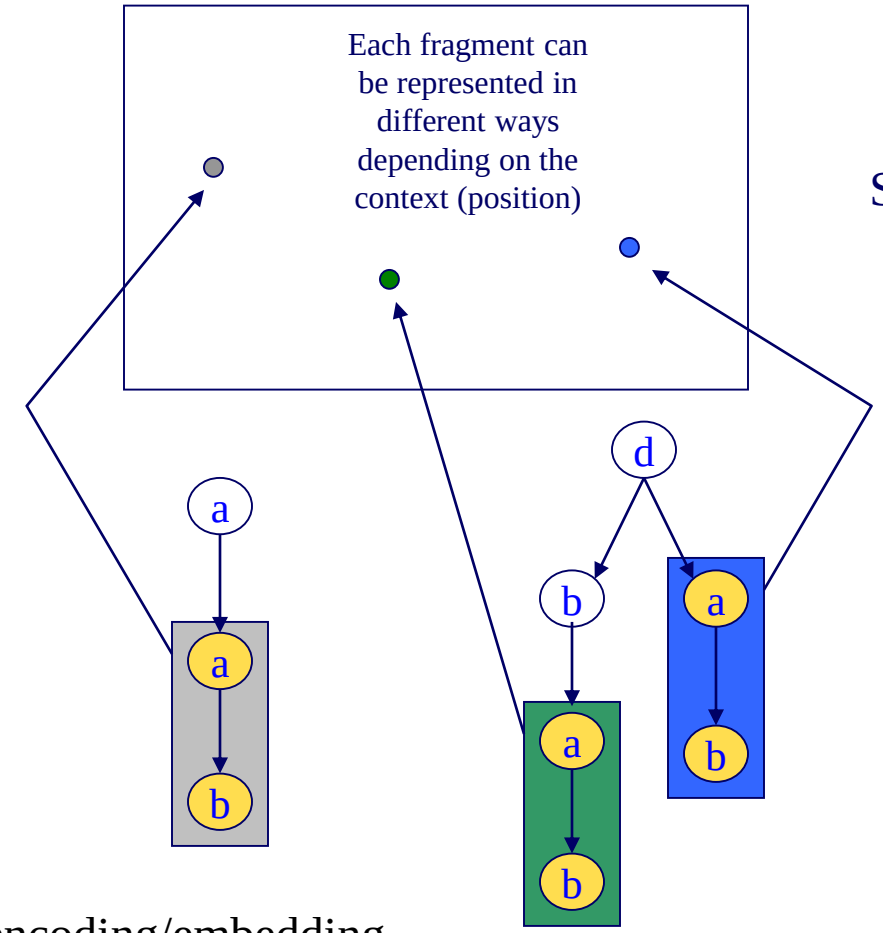
Example on a State Space

University of Pisa

Causal mapping



Contextual mapping



Internal Encoded Subgraphs Space

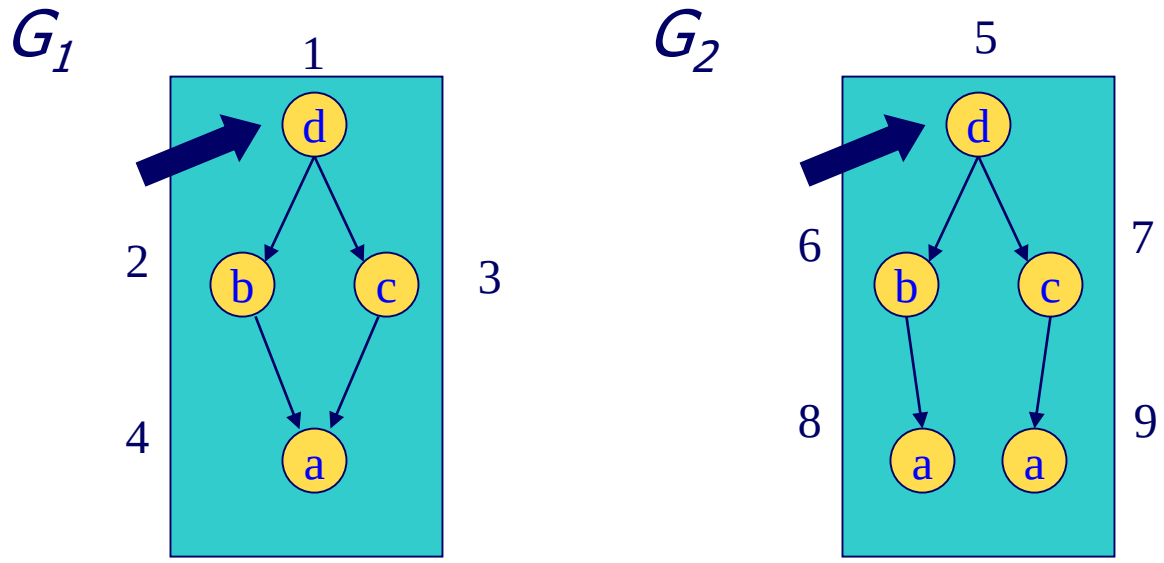
More expressive sub-structures encoding/embedding

DPAG representation: a counter-example

University of Pisa

- Relevance of contextual processing (II)

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



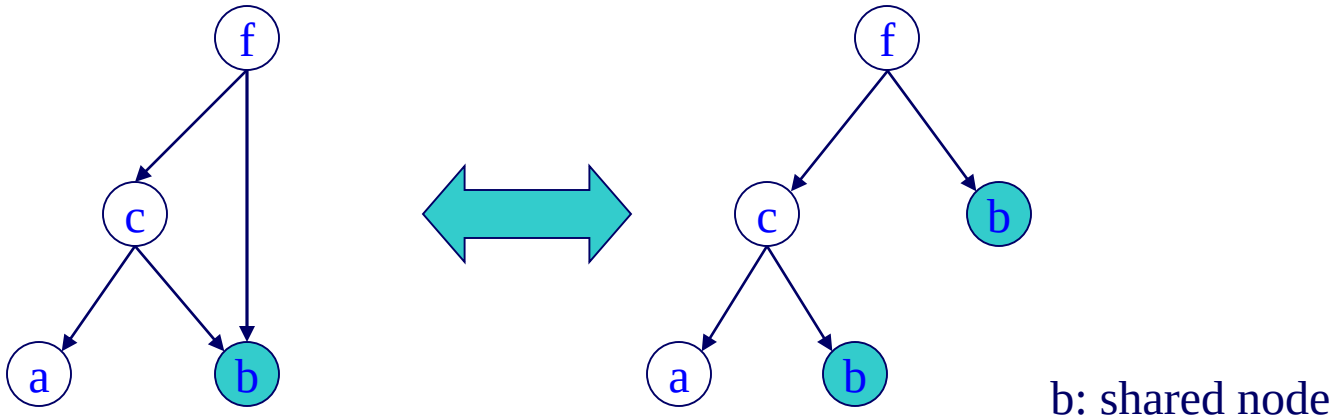
- CRCC **can** distinguish G_1 / G_2 (*context for node "a" is different*), RecNN **cannot** (*b and c see the same state values*)

DPAGs are not trees !

University of Pisa

- Relevance of contextual processing (III)

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



b: shared node

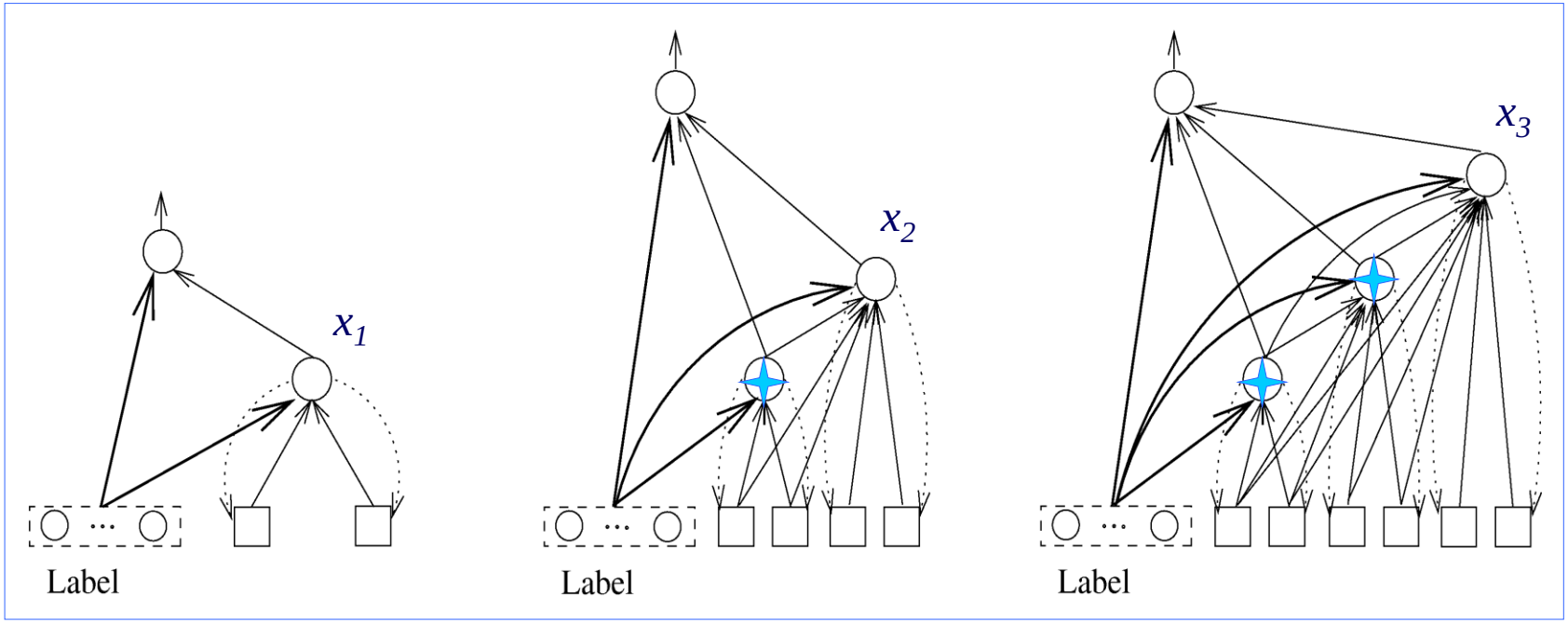
- CRCC distinguish them !
- *By CRCC we extend the domain from trees to DPAGs* !

How?
Recall the RCC Architecture

University of Pisa

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 after the training



It builds a **Deep** Neural Network!

The CRCC Contextual Approach

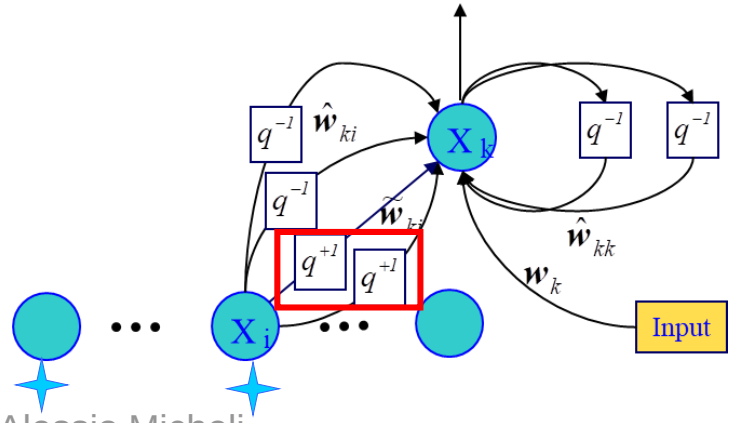
University of Pisa

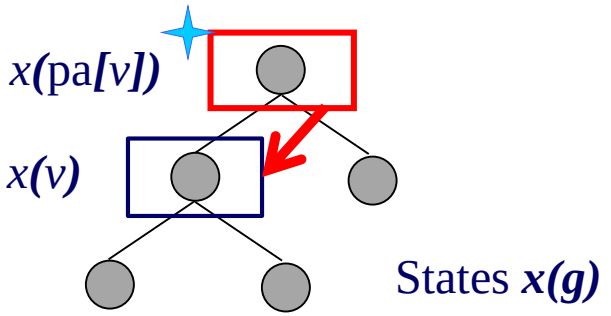
- Each time a unit is frozen, the portion of its (memorized) state encodes knowledge of "the whole" structure

$$x_1(v) = \tau_1(x_1(\mathrm{ch}[v]), \boldsymbol{l}(v))$$
$$x_2(v) = \tau_2(x_2(\mathrm{ch}[v]), x_1(\mathrm{ch}[v]), x_1(\mathrm{pa}[v]), \boldsymbol{l}(v))$$
$$\vdots \qquad \vdots \qquad \vdots$$
$$x_m(v) = \tau_m(x_m(\mathrm{ch}[v]), x_{m-1}(\mathrm{ch}[v]), x_{m-1}(\mathrm{pa}[v]), \ldots, x_1(\mathrm{ch}[v]), x_1(\mathrm{pa}[v]), \boldsymbol{l}(v))$$

unit m — frozen unit m-1 ····· frozen unit 1

pa[v]: set of parents of v



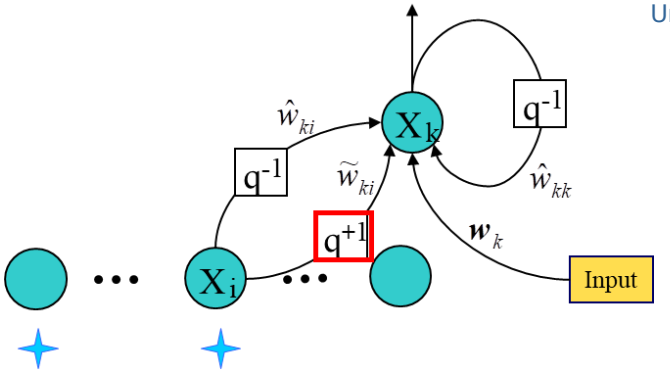


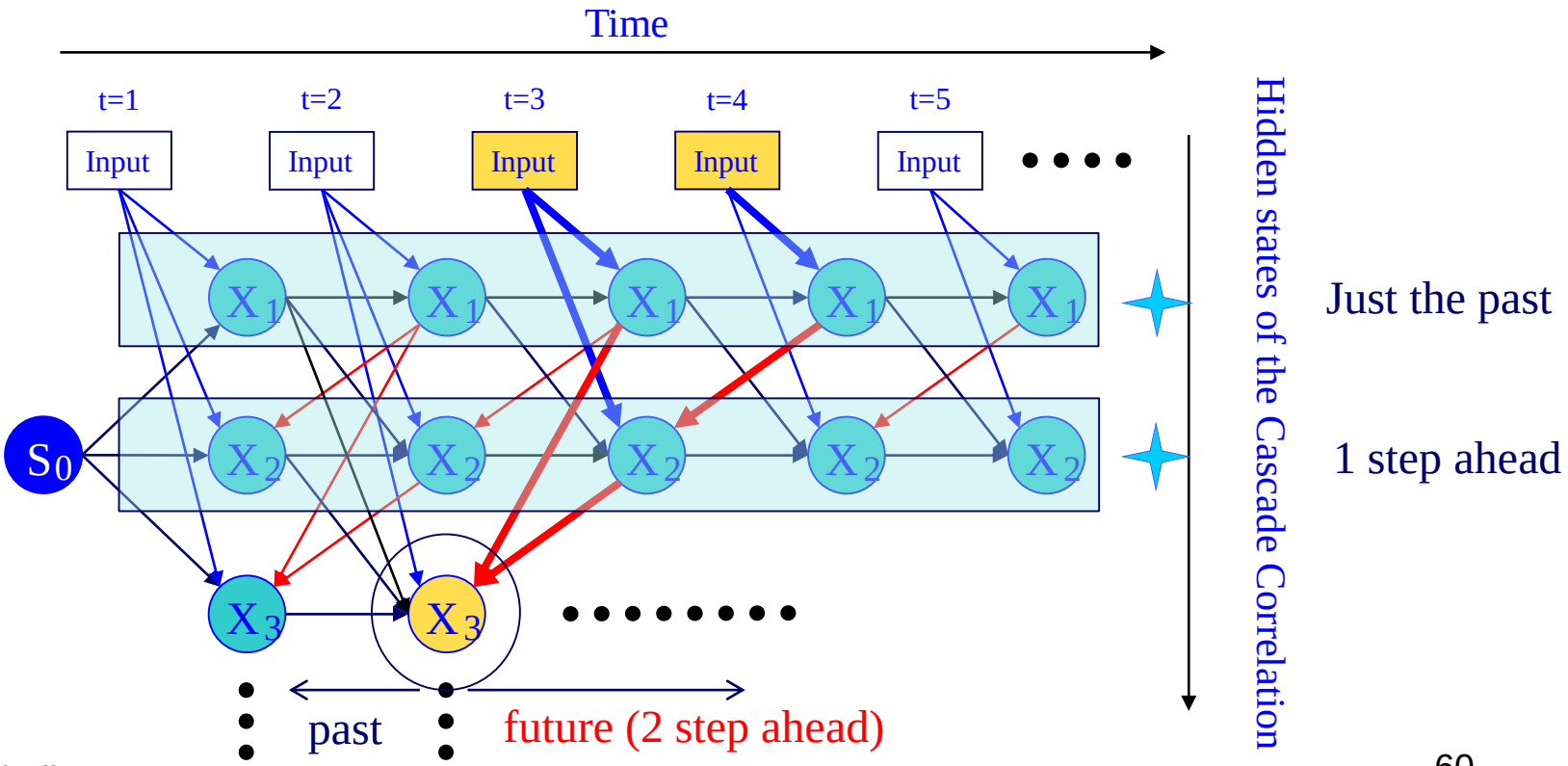
Micheli et al. IEEE TNN, 2004

Example: CRCC on a sequence

University of Pisa

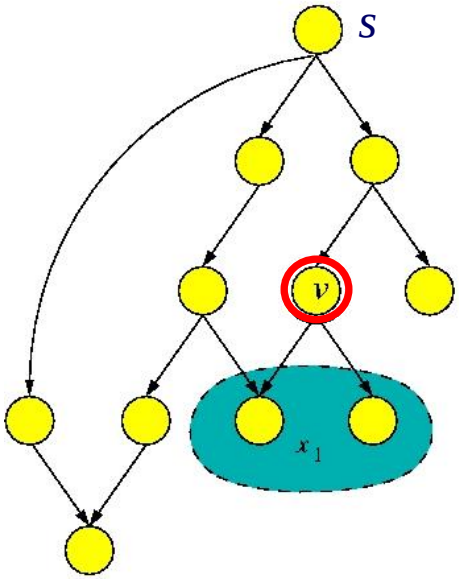
We can gain information on the "future" proportionally to the number of hidden units

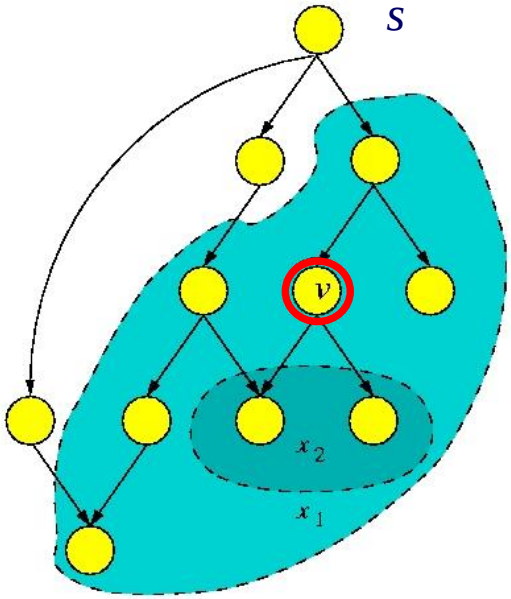


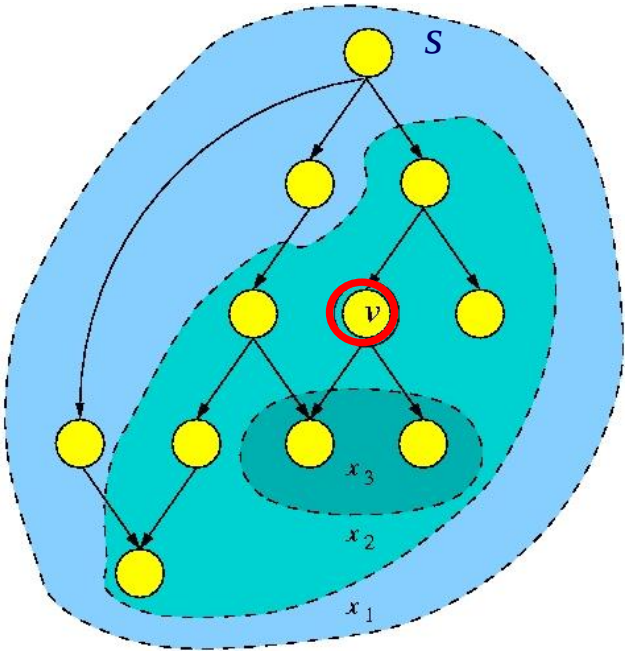


Example: *C(•) for* DPAGs

University of Pisa



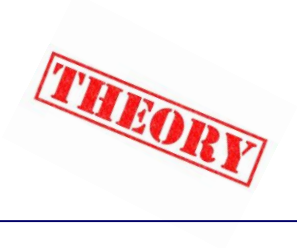




$C(x_1(v))$ $C(x_2(v))$ $C(x_3(v))$

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

Theory



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 (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**)

Micheli et al. IEEE TNN, 2004

Example: $C(\bullet)$ for Sequences

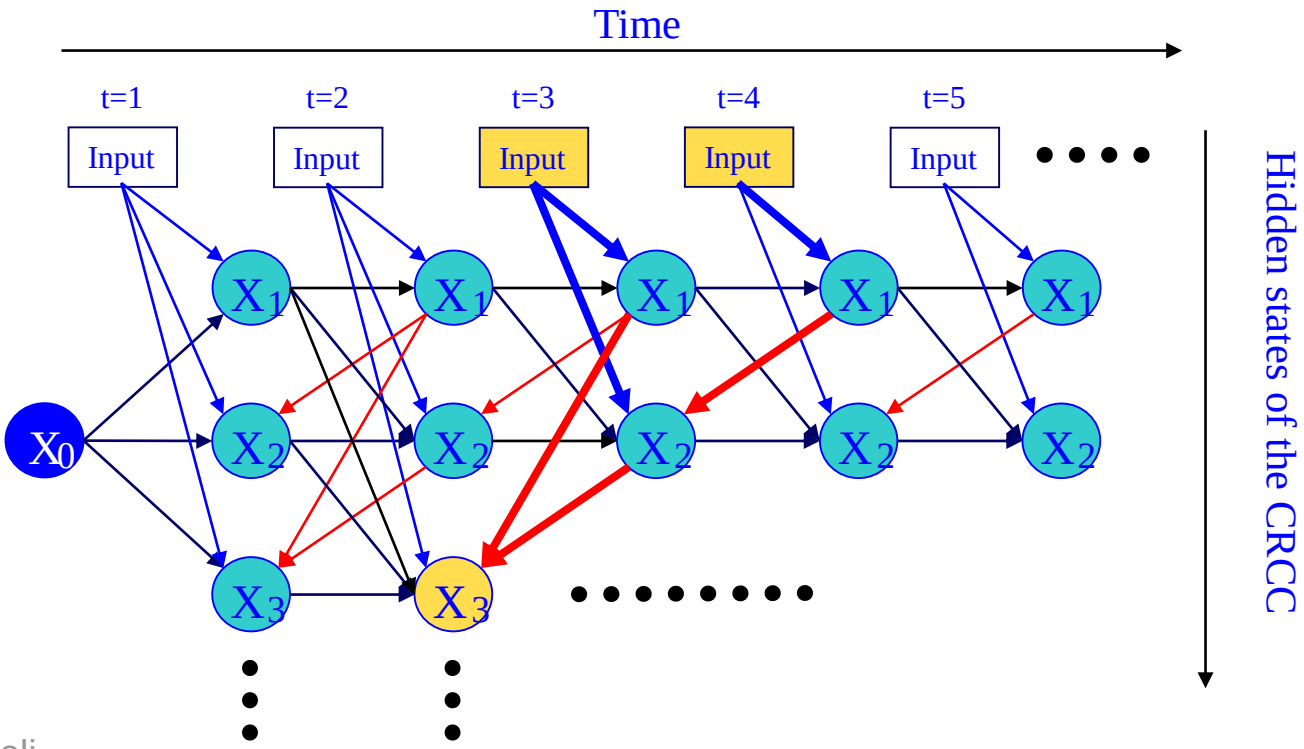
University of Pisa

THEORY

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

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*

THEORY

University of Pisa

- **Proposition 1.** Given a DPAG *G* with supersource *s*, for any vertex *v* such that *dist(s,v)=d*, the contexts $C(x_h(v))$ with $h>d$ involve all the vertecies 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 vertecies of the graph. In particular, any

$$h > max_v\ dist(s,v)$$

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)

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

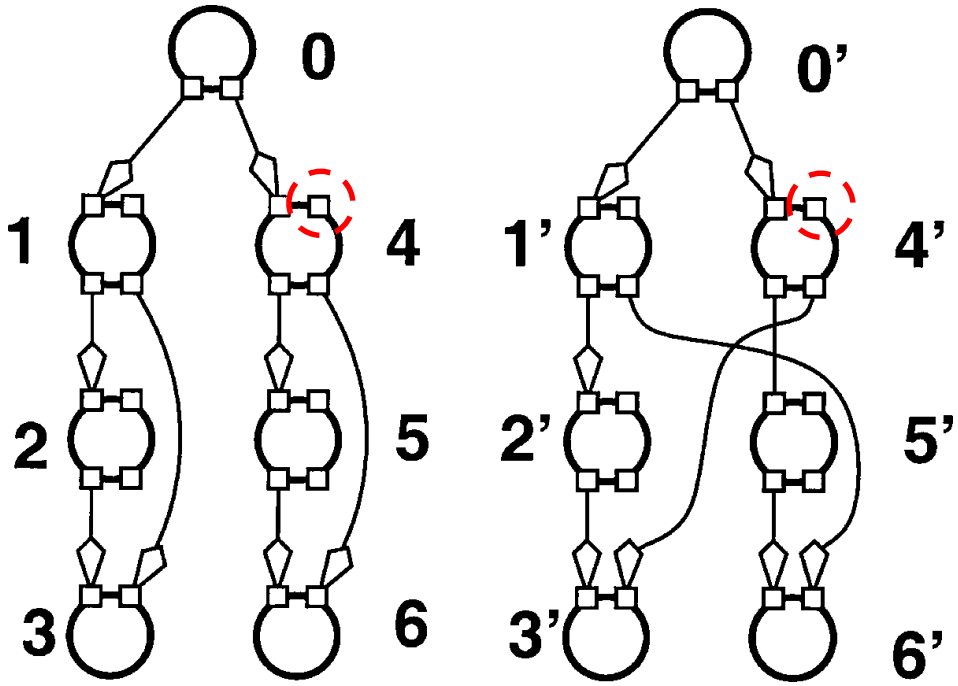
Hammer et al. Neural Computation, 2004

A Counterexample

THEORY

University of Pisa

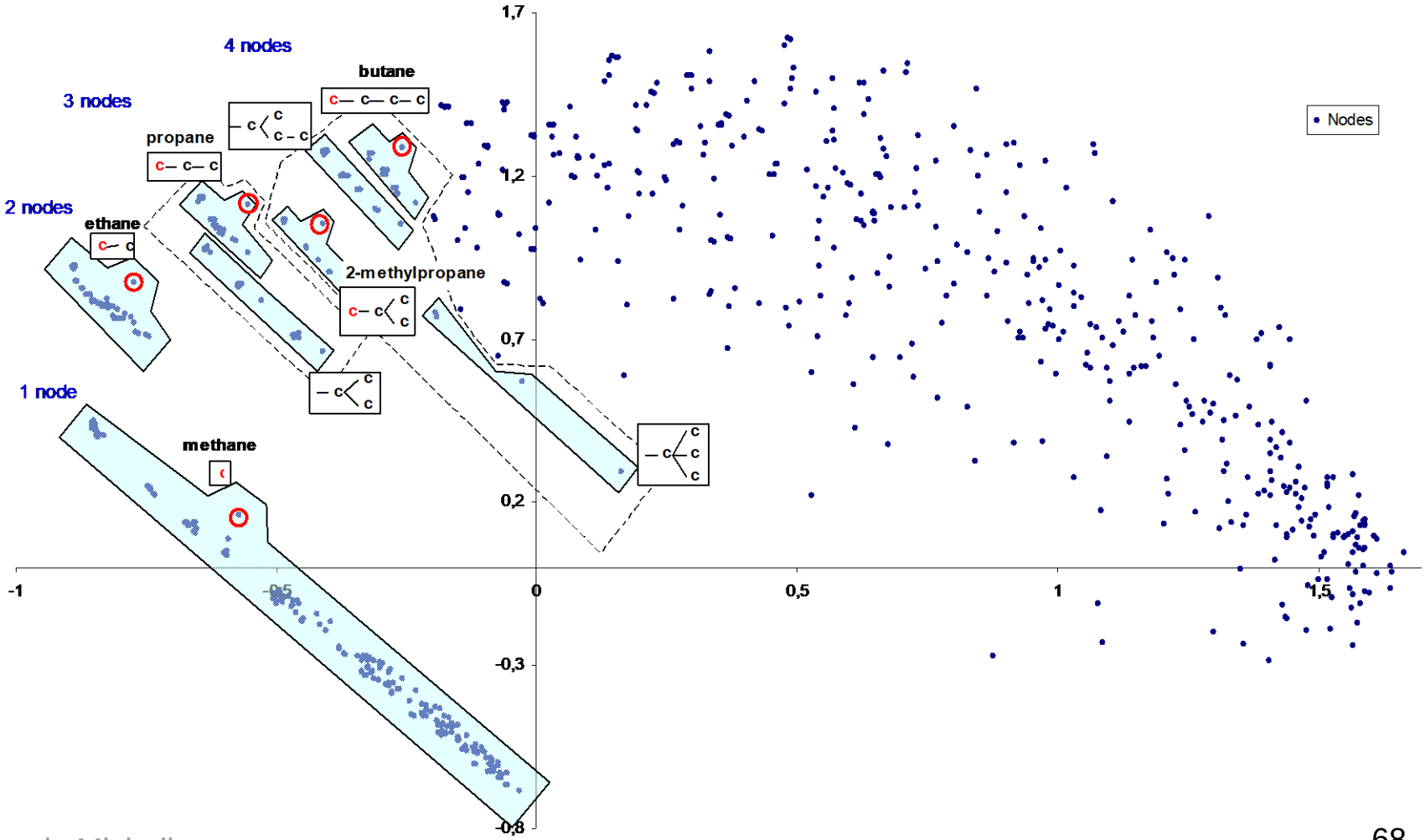
- At least one path starting from the root exists that leads to vertex with different labels or different fan-in or fan-out in two differentiable DPAGs
- E.g. position of children = position of parents (D B*ipositional* AG):



Context in a CRCC Application

University of Pisa

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



Context Relevance

University of Pisa

- **Contextual processing**: does the model process each vertex considering the context over the graph **?**

In recursive approaches we have seen the context considering the following sub-structures (due to the *causality*):

- Sub-sequences for RNN
- Sub-trees for RecNN
- Recursive Contextual models extend the context to parents (not only descending vertices) moving to DPAG, but still a *recursive causal* approach:

- New approaches by extending context, removing causality/recursion**?** Yes, ***Move to graphs!***

Learning in Structured Domain

1. Foundational models

- Introduction to structured data processing
- Recursive models:
  - From sequences to trees
  - Moving to DPAG the role of causality
- Learning variable-size graphs with cycles
  - Contractive encodings approaches for graphs
  - Contextual Multi-Layered approaches for graphs

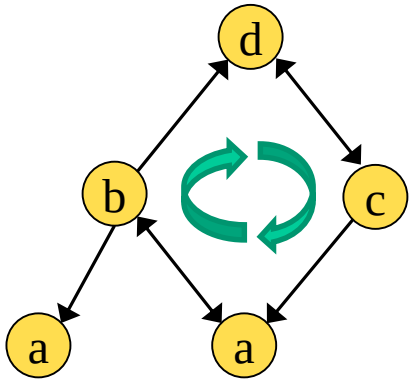
By a journey through the causality assumption!

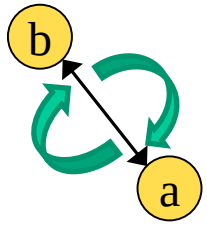
Graphs by NN: Cycles

University of Pisa

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

Cyclic graph





Occuring also for undirected edges!

How to deal with cycles and causality?

An useful view

University of Pisa

Mackassy and Provost taxonomy [JMLR 2007]:

3 main components for (node-centric) learning

- **Non-relational ("local") model:** e.g. priors
- **Relational model.** Relations in the networks are considered (e.g. Neighbors information)
- **Collective inferencing:** the entities are estimated together and 'simultaneously', considering the mutual influences
  - i.e. the cyclic dependencies described so far

Achieving the 3 components is essential to have a full **contextual processing** over networks or graphs both for inference or state-based learning approaches

Macskassy, Provost. JMLR, 2007

Main approaches for graphs by NN

University of Pisa

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]

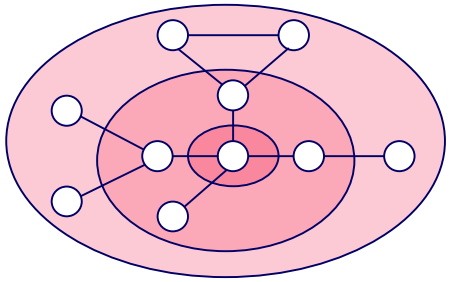
3. **Layering**: contextual non-recursive approaches (NN4G [3] /Conv. NN for graphs [4]) → **Deep NN**

1. Scarselli, Gori, Tsoi, Hagenbuchner, Monfardini. IEEE TNN, 2009.
2. Gallicchio, Micheli. IJCNN, 2010.
3. Micheli. IEEE TNN, 2009.
4. See later in the second part

2. GNN/GraphESN (2009-2010)

University of Pisa

- In GraphNN (GNN) and GraphESN 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 *GraphESN* the condition is inherited by contractivity of the reservoir dynamics (see ESP conditions): *very efficient!*
  - In *GNN* imposing constrains in the loss function (alternating learning and convergence)



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

University of Pisa

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 → bias to contractive transduction

+ GraphESN dose not require training time of the recursive part → efficient!

3. Layering

Contextual Multi-Layered approaches for graphs

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
  - In the second part

3. NN4G: Motivations

University of Pisa

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

NN4G : Neural Network for Graphs

- Two main ingredients:
  1) constructive (feedforwad) neural network approach
  2) Local and contextual information of each vertex of a graph
  - But recursive causality is removed

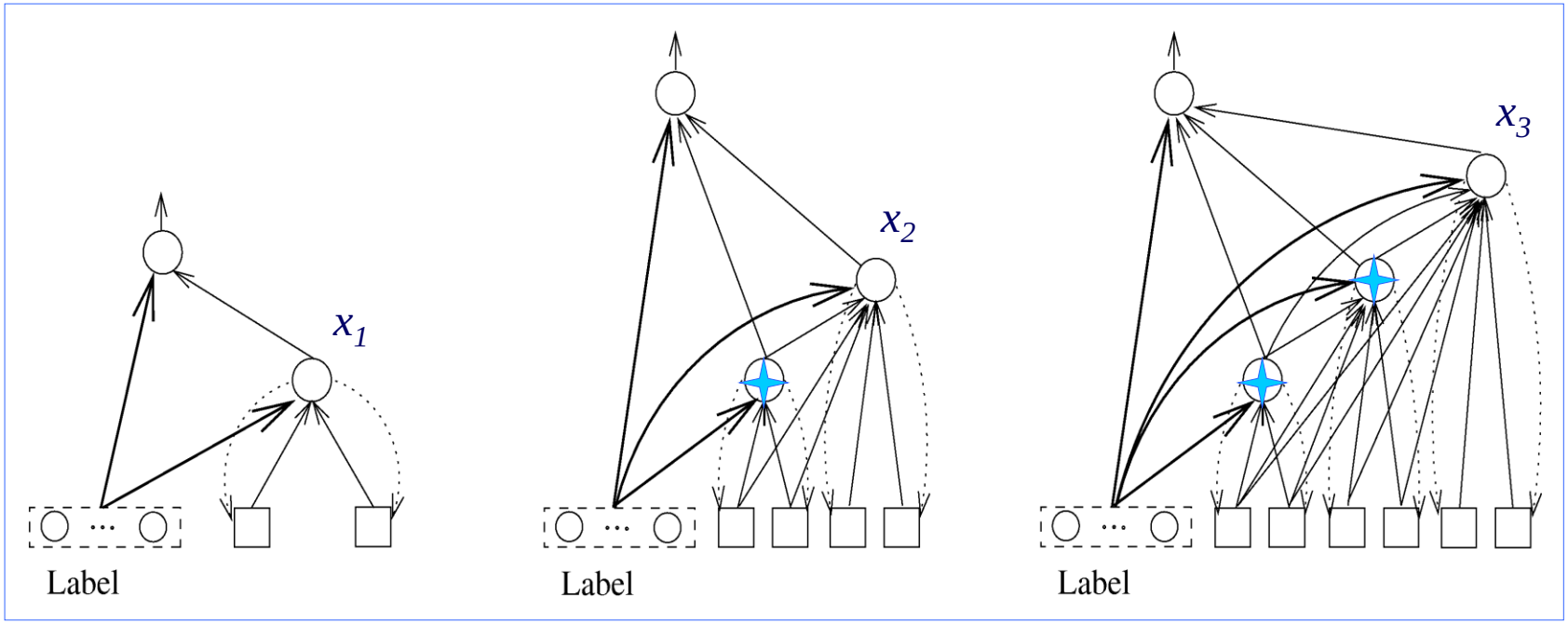
- Micheli, Sestito. WIRN 2005
- Micheli. IEEE TNN, 2009.

NN4G: 1) Constructive Approach

University of Pisa

Cascade Correlation (*RecCC* in the picture):

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



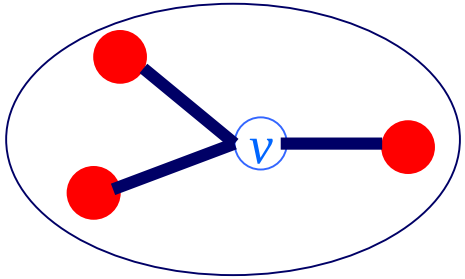
NN4G: 2) Neighbors

University of Pisa

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

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

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



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

NN4G: 2) Hidden Units and Context

University of Pisa

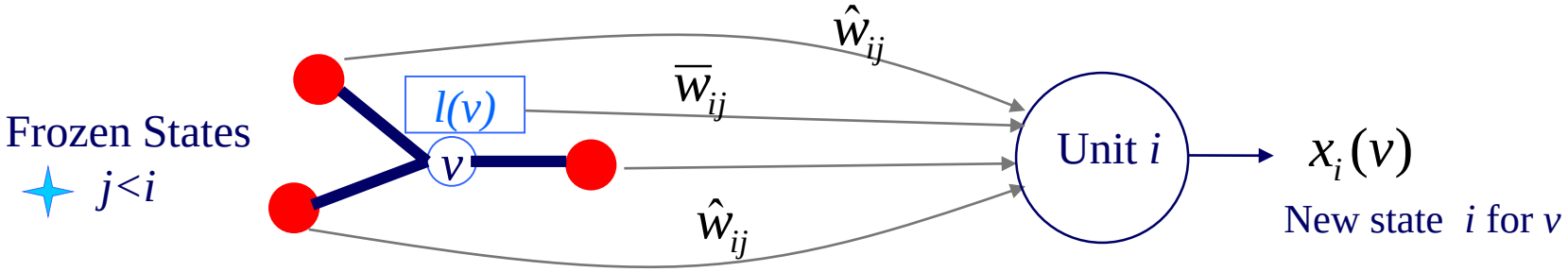
- NN4G compute a state variable for each vertex

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

State

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 vertices of g

NN4G: Hidden Units and Labeled Edges

University of Pisa

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

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

W for edge (v,u)

(v,u) is unordered (undirected graphs)

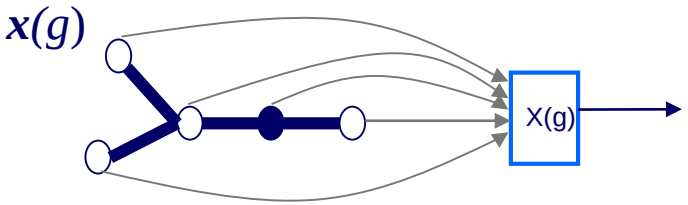
- Stationarity (weight sharing) strategy: association between weights and edges
  - **Label of the edge** (included position/orientation etc.)
  - Firs trial: **full stationarity**: 1 weight for all the edges for unordered and undirectd graphs

NN4G: Output unit

University of Pisa

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.***:

States mapping function



Can be even a simple global average or selection from relevant vertices etc.

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.

NN4G: Algorithm (inference)

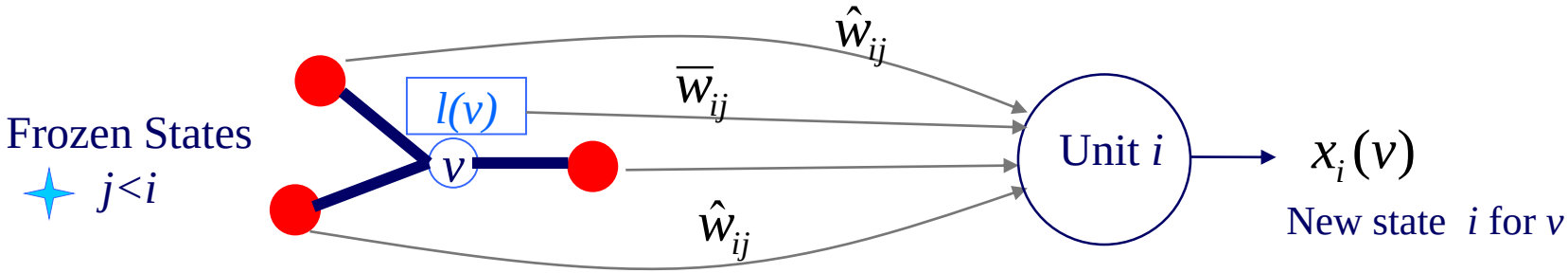
University of Pisa

1. For i=1 to N — All the created states
2. For all g in G
3. For all v in $Vert(g)$
4. Compute $x_i(v)$ *(even in parallel)*
5. Compute $X_i(g)$ — States mapping function
6. For all g in G
7. Compute $y(g)$

NN4G: 2) Hidden Units and Context

University of Pisa

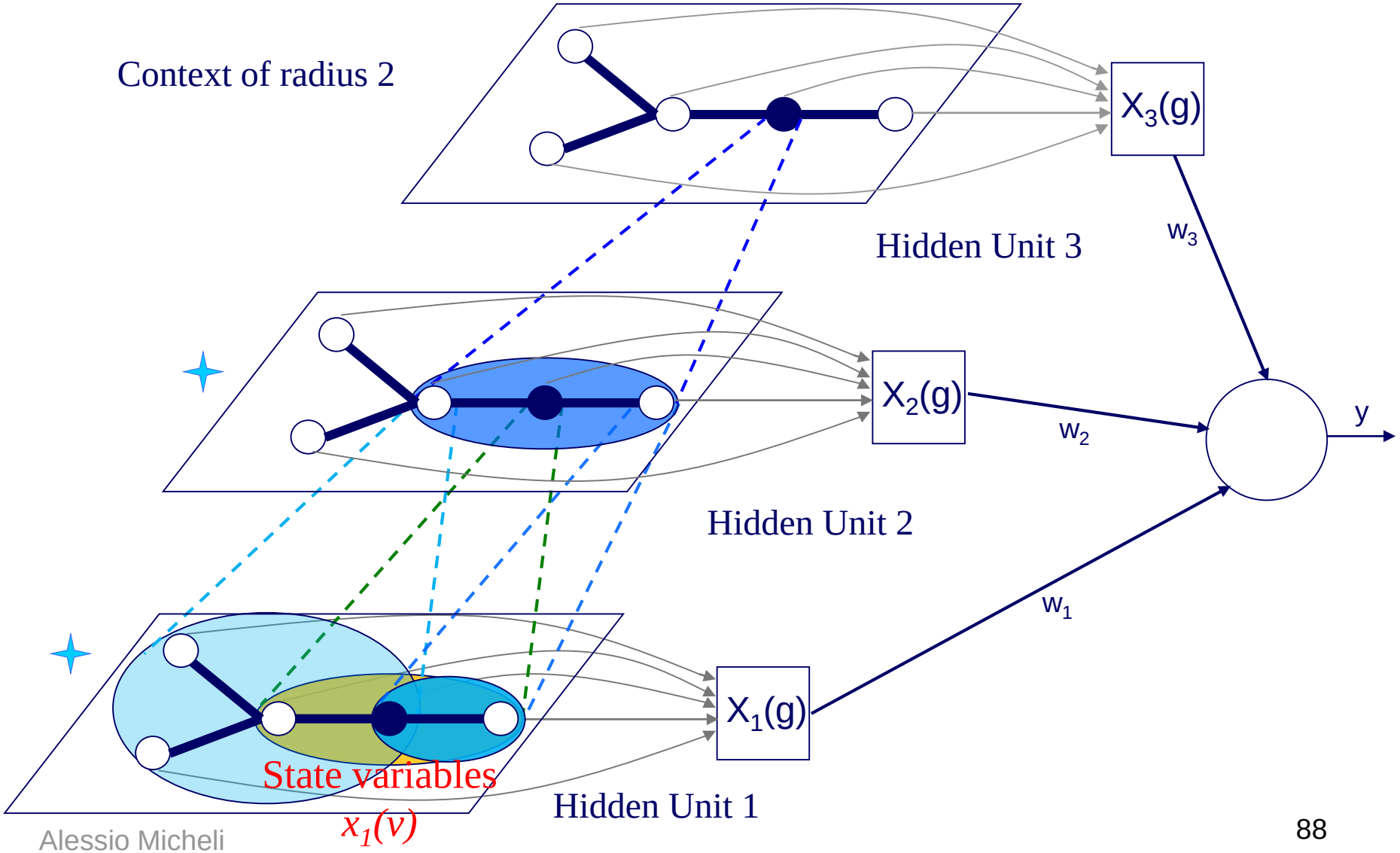
- Is NN4G just a relational approach taking only a local neighborhood (for each hidden unit) **?**



- ***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 vertecies through the context developed in the previous frozen hidden units (layers) → *see the next slide*

NN4G: Evolution of the Context (Composition through layers)

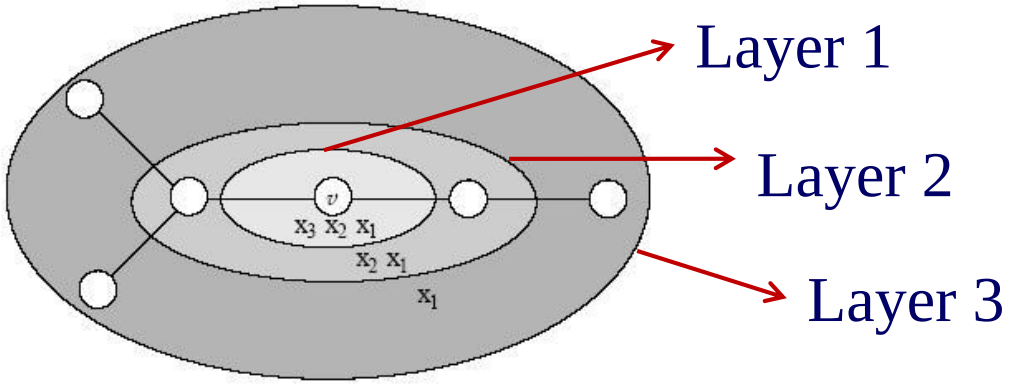
University of Pisa



NN4G: Context Growth

University of Pisa

- 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*

THEORY

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* > "diameter" of the graph satisfy the proposition.

Micheli. IEEE TNN, 2009.

NN4G Recap

University of Pisa

- NN4G: A deep model for graphs
- Characteristics:
  - Direct/undirected cyclic/acyclic labeled graphs
  - ☆ – 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|Vh² epochs):* Linear in the number of vertecies
  - ☆ – Generality: No constraints on weights values
  - Pool strategy (Cascade corr. Training): local minima avoidance, supervised architecture optimization .

Guidelines Concepts for NN4G

University of Pisa

- **Compositionality (structured encoding):** by processing of structure's vertices with composition of contexts extended to a *symmetric window* around vertices for node embedding (avoiding recursive *causality*)

- **Parsimony**: *stationary* conditions with the same possibilities (weigh-sharing at vertex or link levels, etc.) to fix the number of free-parameters

- **Adaptivity**: NN training simplified by using F.F.N.N. architectures

Analogues properties hold for contextual multilayers approaches that are apperaning through the CNN line : *Convolutional NN for graphs* (in the second part)

A first comparison NN4G / Conv.NN for Graphs

University of Pisa

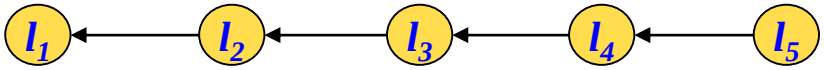
Concepts in common:

- Visiting input graphs through units with weight sharing (stationarity) (that for CNN are constrained to graph topology instead of 2D matrix)
- Layering and hence *moving to deep architecure* (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*

Main differences can be for *training*:

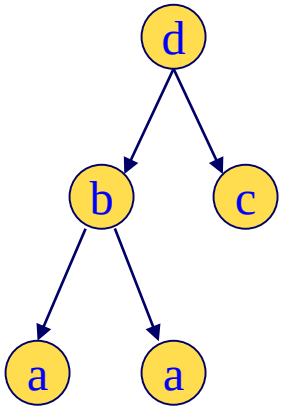
- **CNN-G**s typically use CNN training approaches, as top-down back-prop (end-to-end): see second part (can be quite demanding using many layers)
- **NN4G**: Incremental, layer by layer learning & automatic model design
  - Advantages: No gradient vanish issue, divide et impera etc.

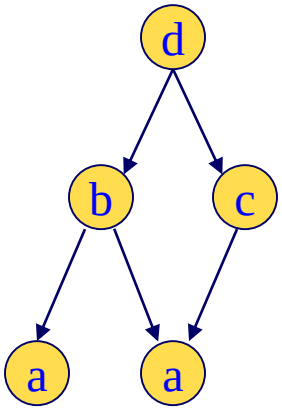
Summarizing the models panorama for SD (examples)

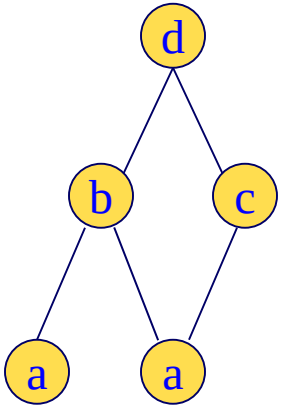


Standard ML models for flat data

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







Tree:
- Recursive NN
- Tree ESN
- HTMM
- Tree Kernels
- …

DPAG:
- CRCC

- GNN/GraphESN
- NN4G
- Conv.NN for G.
- Graph Kernels
- SRL
- …

See references for models in the bibliography slides

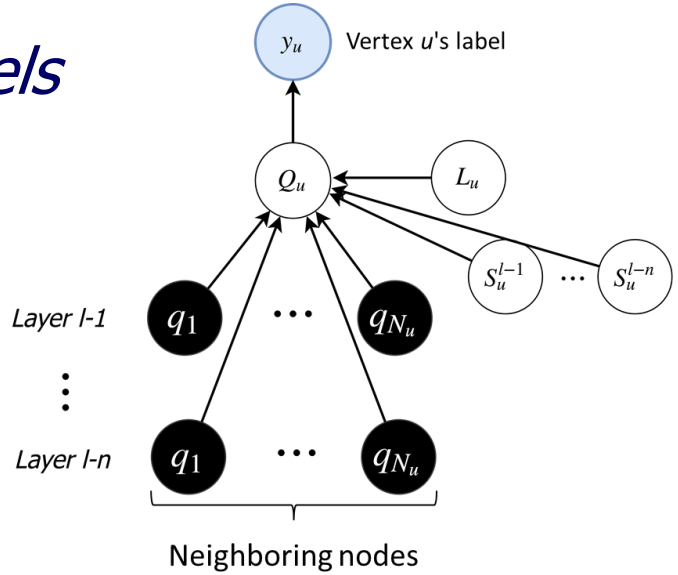
Looking ahead (I)

University of Pisa

- We can extend such contextual ideas aldso to RC and HTMM approaches making them *deep and for graphs*

- *E.g. ICML 2018 (see also next presentaion)*
- A NN4G realized by a *generative* approach
- trained by a mix of unsupervised (Markov models for hidden layer) and supervised (output layer) approaches

- *Contextual Graph Markov Models*



Looking ahead (II)

University of Pisa

- **A theoretical and practical framework for the automatic design of efficient models**
  for sequence, trees and graphs (both generative and discriminative) exploiting Deep Learning 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 NNs

  - We are open to collaboration, (model developments/new graph datasets exchange/comparisons)
  - Contact us: micheli@di.unipi.it, bacciu@di.unipi.it

Computational Intelligence & Machine Learning Group

News

University of Pisa

Forthcoming

- Special Session on "Embeddings and Representation Learning for Structured Data" at ESANN 2019

- https://www.elen.ucl.ac.be/esann/index.php?pg=specsess#structured
- http://www.esann.org

Deep Learning for Graphs
Plan in 2 lectures

1. **Foundational models:**
   Intro to Learning in Structured Domains
   - Extensions of original flat models to supervised and unsupervised learning in structured domains: from vectors to graphs

2. **Deep learning models**
   - Current view of deep learning for graph and network data

DLSD@ECML-2018:
https://sites.google.com/view/dl4sd

Main References (I)

University of Pisa

RecursiveNN (at the origin)

- 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 Journa, Vol. 12 (1/2): 117-146, 2000.
- B. Hammer, Learning with recurrent neural networks, Vol 254, Springer, 2007.

Contextual RNN (CRCC): from Trees to DPAG and properties

- 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, MIT press, 2005.

NN for Graphs:

- 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.
- C. Gallicchio, A. Micheli. Graph Echo State Networks, Proceedings of the International Joint Conference on Neural Networks (IJCNN), pages 1–8, 2010.
- 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.

Network data overview

- S.A. Macskassy, F. Provost. Classification in Networked Data: A Toolkit and a Univariate Case Study JMLR 8:935-983, 2007.

Other References (II)
(through the tutorial)

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

Deep ESN

- C Gallicchio, A Micheli, L Pedrelli. Deep reservoir computing: a critical experimental analysis, Neurocomputing, Vol. 268, Pages 87-99, 2017
- C Gallicchio, A Micheli. Echo State Property of Deep Reservoir Computing Networks, Cognitive Computation, Vol. 9, Issue 3, pp 337–350, 2017
- C. Gallicchio, A. Micheli, L. Silvestri, Local Lyapunov Exponents of Deep Echo State Networks, Neurocomputing 298 (2018) 34-45, 2018.
- C Gallicchio, A Micheli, Deep Echo State Network (DeepESN): A Brief Survey. arXiv:1712.04323 (2017).

Other References (III)
(through the tutorial)

University of Pisa

Unsupervised RecursiveNN

- M. Hagenbuchner, A. Sperduti, A.C. Tsoi, A selforganizing map for adaptive processing of structured data. IEEE Trans. on Neural Networks 14 (3), 491–505, 2003.
- 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, 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, 2004.

HTMM: further developments (generative)

- N. Gianniotis and P. Tino, "Visuali zation of tree-structured data through generative topographic mapping," IEEE Trans. Neural Netw., , vol. 19, no. 8: 1468–1493, 2008.
- 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: 1987-2002, 2012.
- D. Bacciu, A. Micheli, A. Sperduti. Compositional Generative Mapping for Tree-Structured Data - Part II: Topographic Projection Model. IEEE Transactions on Neural Networks and Learning Systems.Vol. 24 N. 2: 231 – 247, 2013.
- D. Bacciu, A. Micheli, A. Sperduti. Generative Kernels for Tree-Structured Data. IEEE Transactions on Neural Networks and Learning Systems, to appear, 2018.

Other References (IV)
(through the tutorial)

University of Pisa

- TreeESN/DeepTreeESN: efficient RecNN

- C. Gallicchio, A. Micheli. Tree Echo State Networks, Neurocomputing, volume 101, pag. 319-337, 2013.
- C. Gallicchio, A. Micheli, Deep Tree Echo State Networks, in: Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 499–506, 2018

- REecNN Recent Applications

- 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

Other References (V) (through the tutorial)

University of Pisa

- 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.
- 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.

- NN for Graphs (II):

- Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R. Gated Graph Sequence Neural Networks ArXiv e-prints, 2015.
- F. Scarselli, M. Gori, A.C.Tsoi, M. Hagenbuchner. The Vapnik–Chervonenkis dimension of graph and recursive neural Networks. Neural Networks, to appear, 2018.

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