Grounding LTLf specifications in image sequences

Elena Umili

Sapienza University of Rome

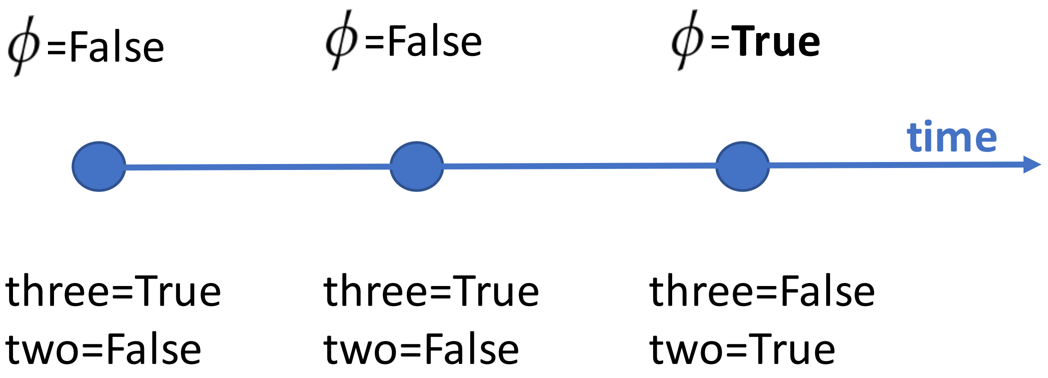
Reference

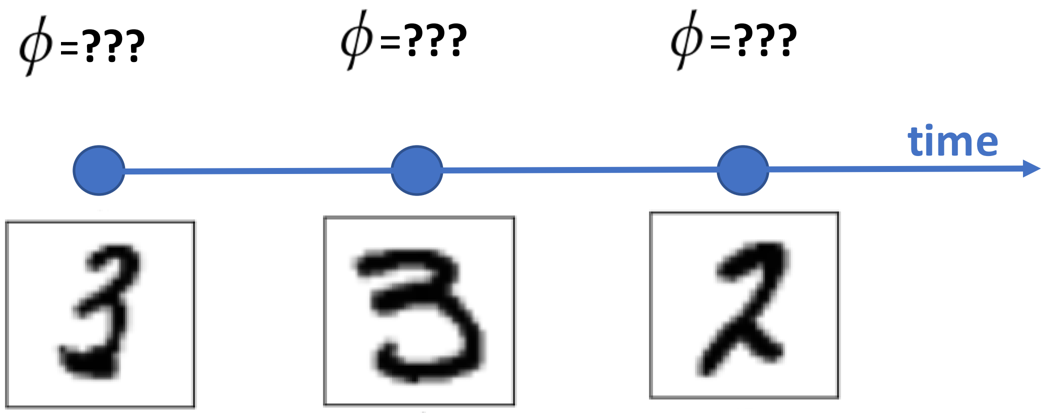
- Elena Umili, Roberto Capobianco, and Giuseppe De Giacomo. **Grounding LTLf specifications in images**. In Proceedings of the 16th International Workshop on Neural-Symbolic Learning and Reasoning as part of the 2nd International Joint Conference on Learning & Reasoning (IJCLR 2022), Cumberland Lodge, Windsor Great Park, UK, September 28-30, 2022, pages 45–63, 2022.

Objective

- Exploit high-level symbolic **temporal knowledge** to increase performances of a sequence classifier in **visual tasks**
- Logical knowledge: **LTLf formula** over a **symbolic set** that is **not grounded in the data**

ϕ **: three until two**

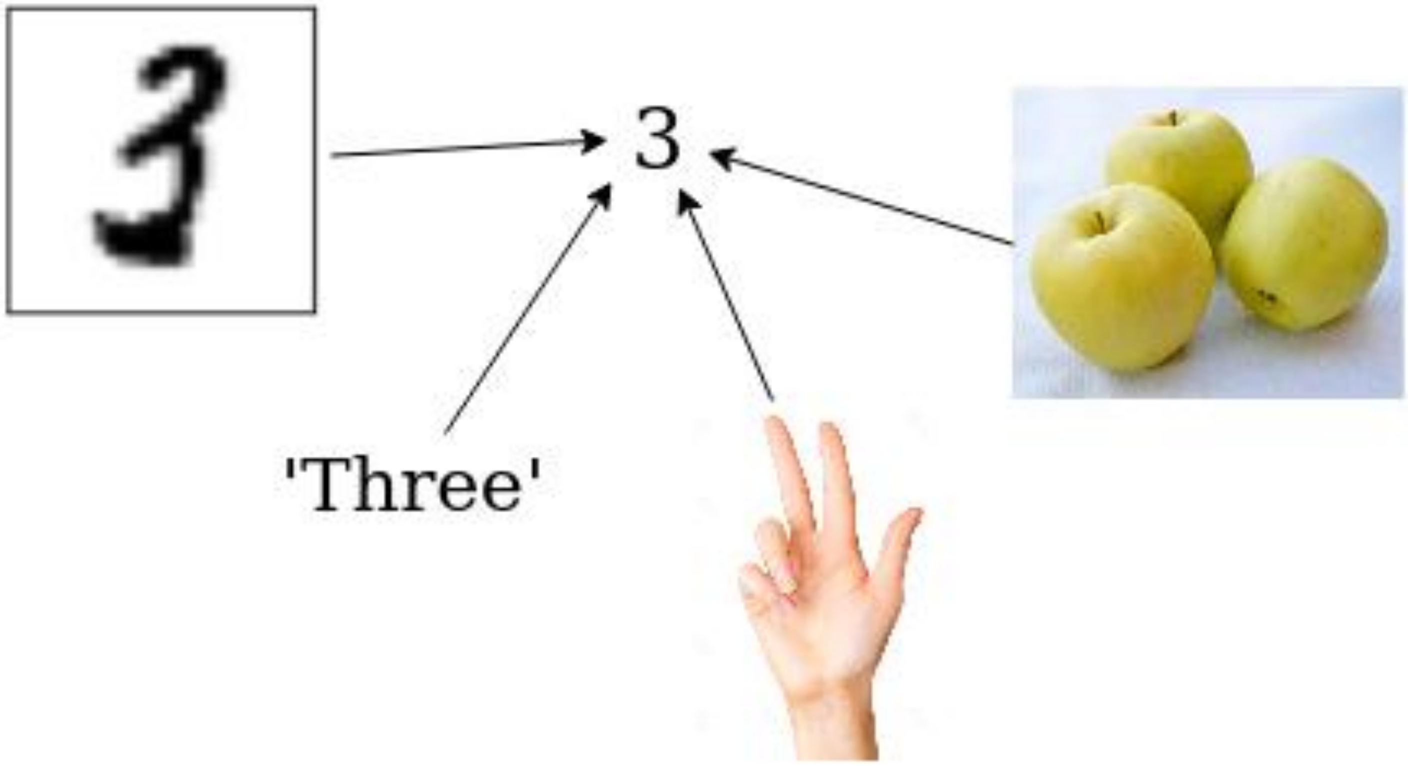




Symbol grounding

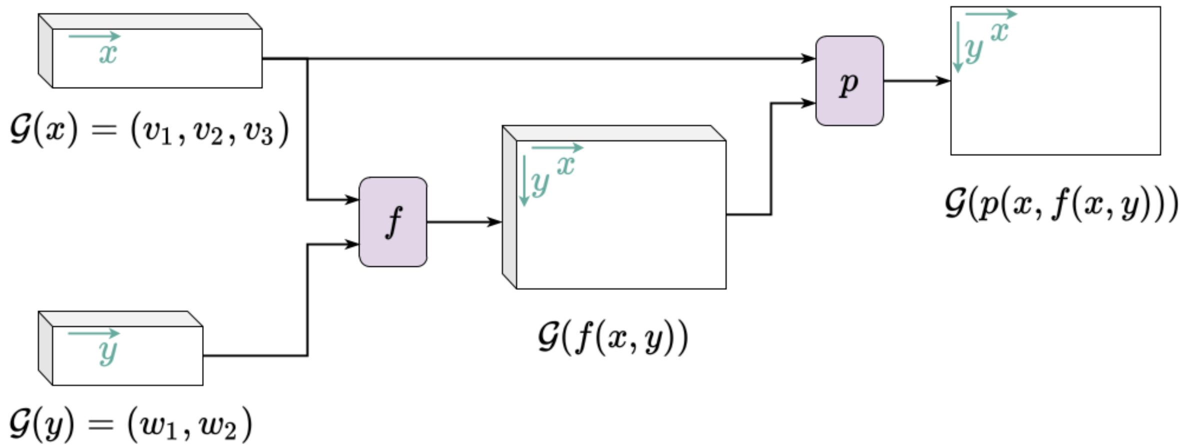
Mapping raw data into a finite set of boolean symbols with a known meaning

A symbol is **grounded** in a dataset if we know the method to recognize it in the data



Background

Logic Tensor Networks (LTN)



- LTN can reason and learn by using **both symbolic knowledge** and **raw data**
- It implements a logic called **Real Logic**, containing constants, function and predicate symbols, as First Order Logic (FOL)
- Any logic formula is interpreted in **fuzzy logic**
- Any piece of the logic can be **implemented** as a **neural network**
- Learning by **best satisfiability**
- We design a **recurrent** LTN, so to impose logic specifications that are extended in the time dimension

Method

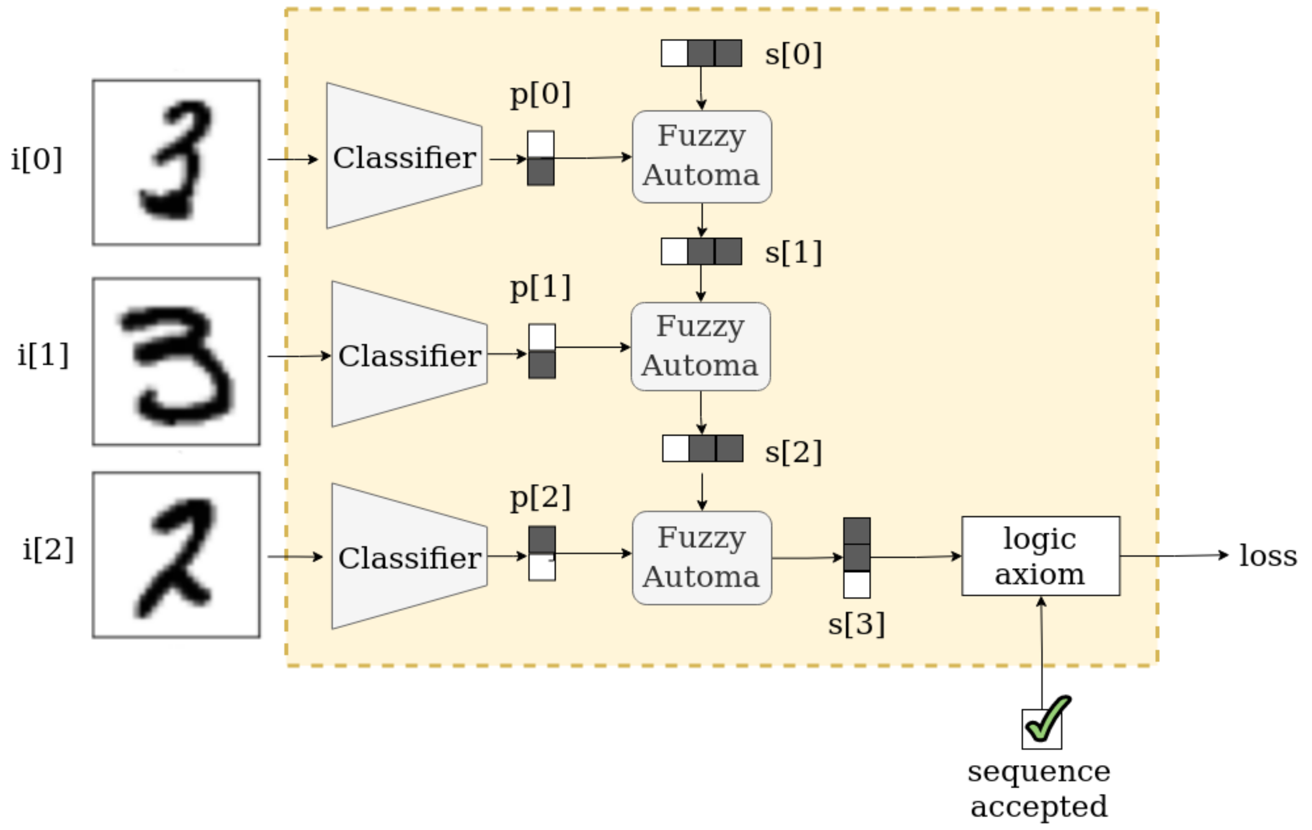
Problem formulation

- **Objective**: Given a sequence of images i[0], i[1],…,i[l] we want to classify the sequence as compliant or not with a given LTLf formula ϕ
- Input:
  - The formula ϕ
  - A set of annotated sequences $D = \{< x_1, y_1 >, < x_2, y_2 >, ..., < x_n, y_n >\}$ where
    - x is a sequence of images
    - y is 1 if the sequence satisfy the formula, 0 otherwise

Method overview

1. Translate the LTLf formula to a DFA
2. Transform the DFA in a recurrent Logic Tensor Network
3. Train the network with image sequence labels so to maximize satisfiability

Recurrent LTN architecture



The neural network is composed of two parts:

- A **perception** part, implemented by a **trainable** convolutional neural network that classifies symbols from images,
- A **logic** part, represented by a **fixed** recurrent structure, that is a fuzzy correspondent of the automaton

Fuzzy automaton

Predicates used:

- $P_s(s_i, t)$ means: the automaton is in state i at time t
- $P_c(c_i, t)$ means symbol i is in image at time t
- $Accepted(x)$ means sequence of images x is accepted by the formula

Other notations:

State at time t $s[t] = [P_s(s_0, t), P_s(s_1, t), ..., P_s(s_{|S|}, t)$

Interpretation at time t

$$p[t] = [P_c(c_0, t), P_c(c_1, t), ..P_c(c_{|P|}, t)]$$

Truth value of the automaton edge between state i and j at time t

$$e_{i,j}(p[t])$$

Fuzzy automaton working:

- Initial condition

$$P_s(s_0, 0) = \top \wedge (P_s(s_i, 0) = \bot \; \forall 1 \leq i \leq |S|)$$

- Transition rule

$$P_s(s_j, t+1) = \bigcup_{i:(i,j) \text{ is an edge of } A_\phi} P_s(s_i, t) \wedge e_{i,j}(p[t])$$

- Final condition:

$$\forall x Accepted(x) \leftrightarrow \bigcup_{s_i \in F} P_s(s_i, l)$$

Logic loss:

$$L = \sum_{i=0}^{i=n} (1 - s(x_i, y_i))$$

Where s(x, y) is the truth value of the final condition

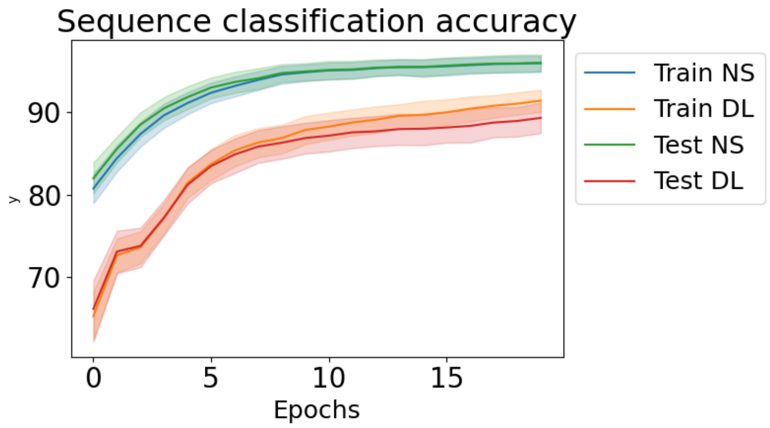
Experiments

DECLARE

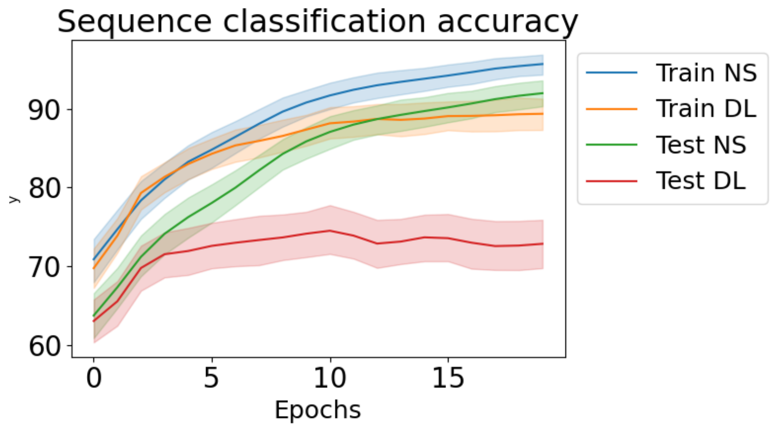
- Declare a declarative language used to model process in Business Process Management (BPM)
- It is composed of 20 types of activity constraints expressed as LTLf formulas
- We tested our approach on the declare constraints in mutually exclusive symbols setting and not mutually exclusive symbols setting

Sequence classification accuracy

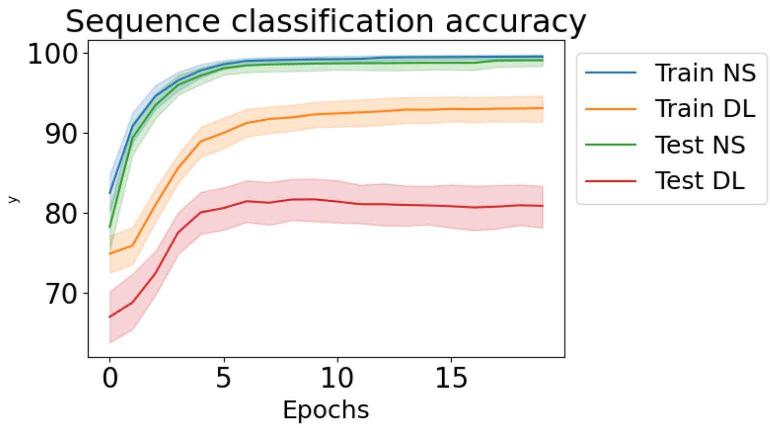
Our approach (NS) achieves better performances of an lstm-based neural network (DL)



Big dataset
(50% training traces ,
85% training images,
mutex symbols)



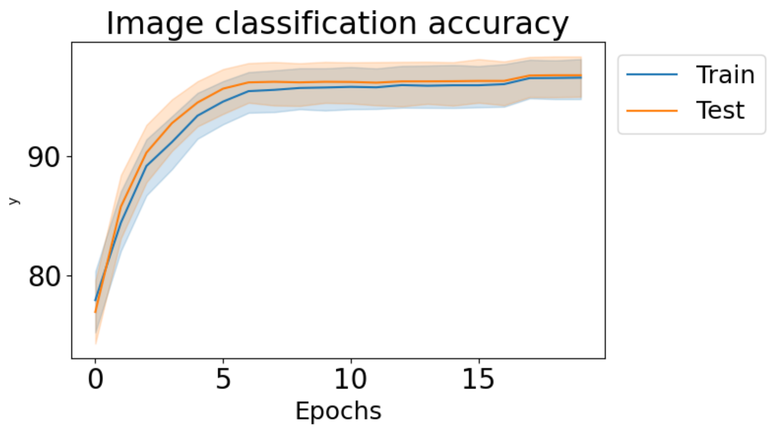
Small dataset
(40% training traces,
15% training images,
mutex symbols)



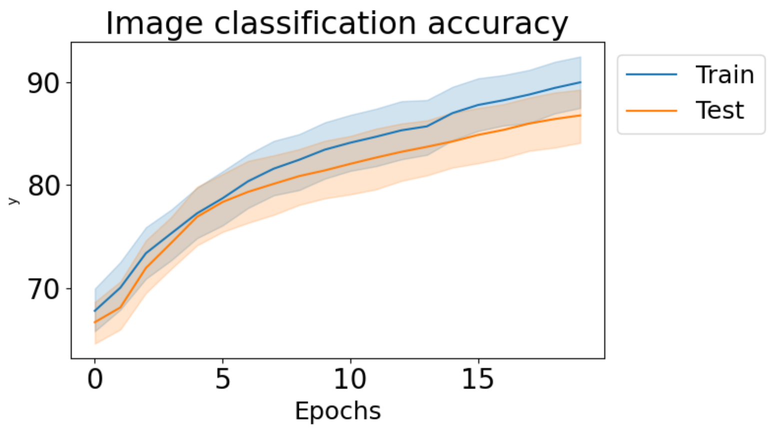
Big non mutex dataset
(50% training traces ,
85% training images,
non-mutex symbols)

Image classification accuracy

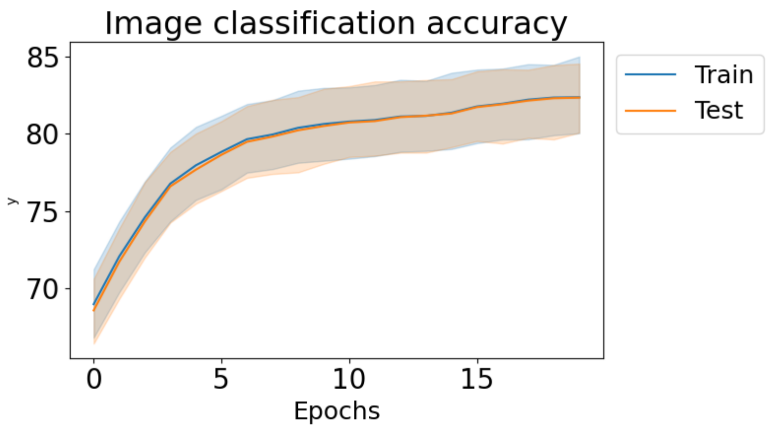
We can reach high performances in single image classification without using any single image label



Big dataset



Small dataset



Big non-mutex datset

Conclusions

Conclusions

- we propose a framework for exploiting high-level logical knowledge in the form of LTLf formulas
- we use this knowledge to map images into a set of symbols with a known meaning without any single image label
- our approach outperforms the end-to-end approach based on recurrent neural networks

Current and future work

- Apply to non-symbolic non-markovian **Reinforcement Learning** tasks
- Extend to tasks expressed in **natural language**