Derivation of Constraints from Machine Learning Models and Applications to Security and Privacy

Moreno Falaschi Catuscia Palamidessi Marco Romanelli

Summary

- Machine learning and explainability
- FOL theories from ML models
  - Decision trees
  - Support vector machines
  - Nearest neighbors
  - Neural networks
- Applications to security and privacy
  - Side channel attack
  - Malware
  - Model inversion

Nowadays ML is pervasive, and performs better than humans

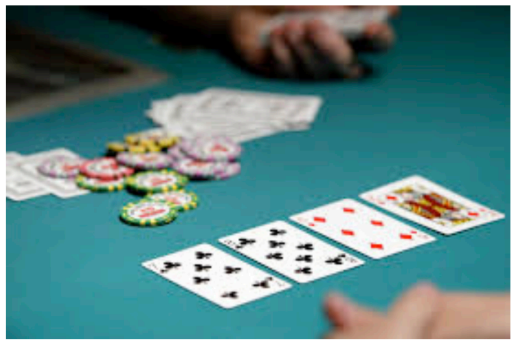
AlphaGo beats Go human champ

Deep Net outperforms humans in image classification

IMAGENET

Autonomous search-and-rescue drones outperform humans

DeepStack beats professional poker players



Computer out-plays humans in "doom"

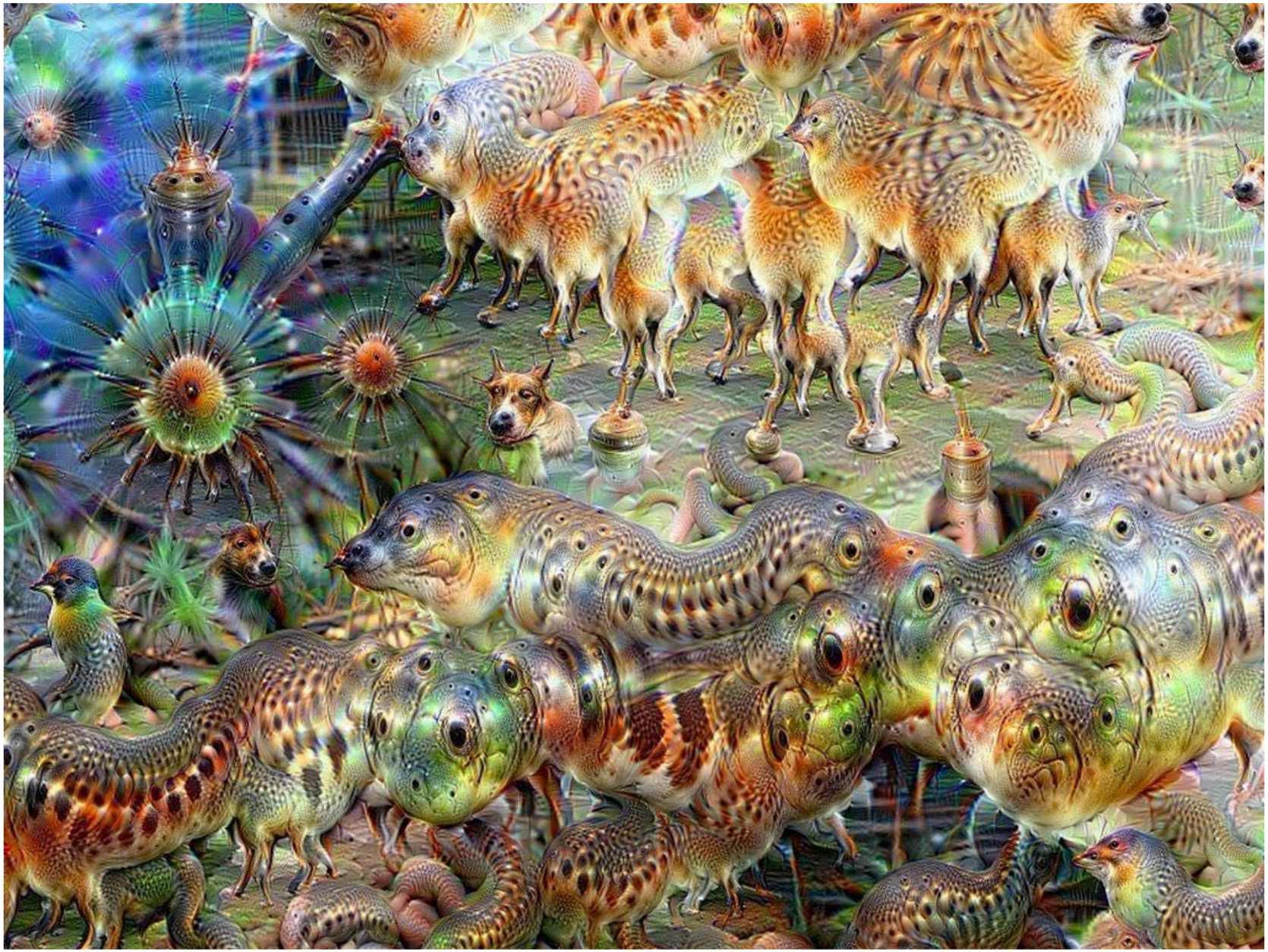
IBM's Watson destroys humans in jeopardy

Deep Net beats human at recognizing traffic signs

Generation of new examples via a GAN

Synthetic images of human faces generated using the CELEBA-HQ dataset

Machine learning and art



An artwork created using DeepDream, developed by researchers at Google in 2015

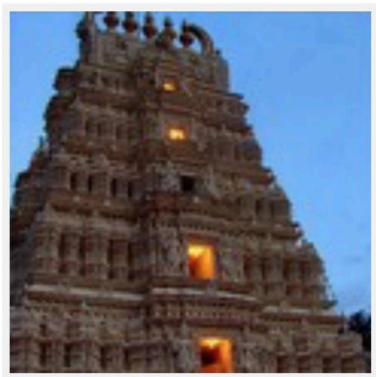
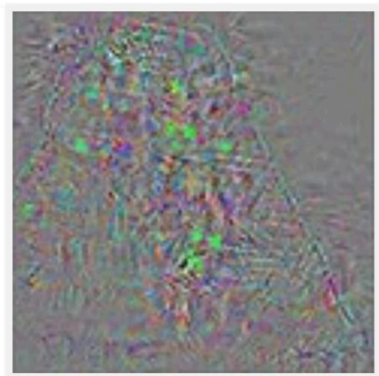
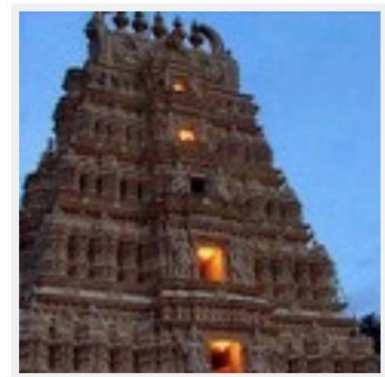
On the other hand...

Original image | Perturbation | Adversarial example



Panda
57.7% confidence

Gibbon
98.3% confidence

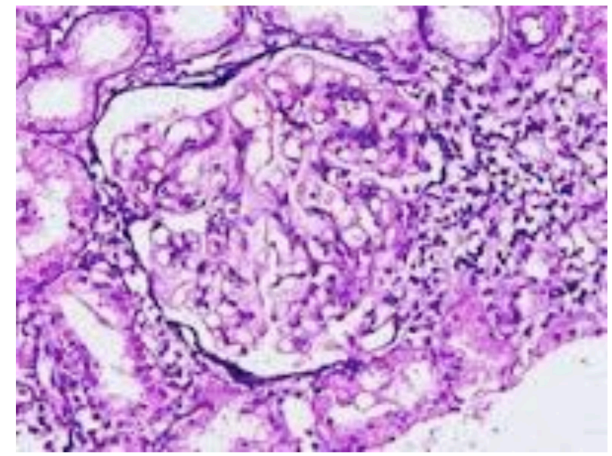
Temple
97% confidence

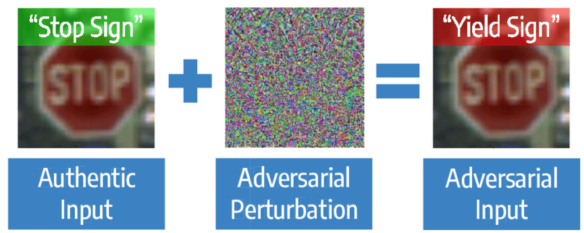
Ostrich
98% confidence

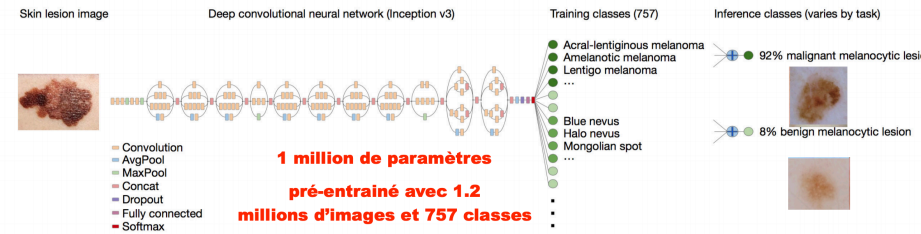
Wrong decision can be costly and dangerous

"Autonomous car crashes, because it wrongly recognizes …"

"AI medical diagnosis system misclassifies patient's disease …"

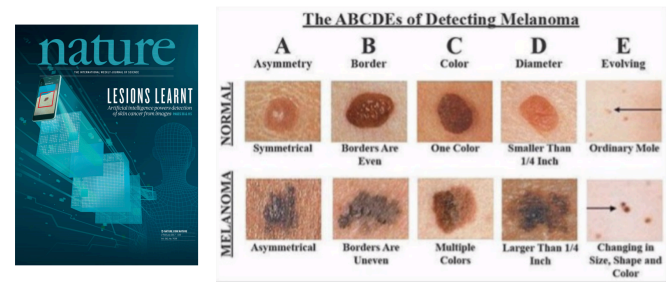






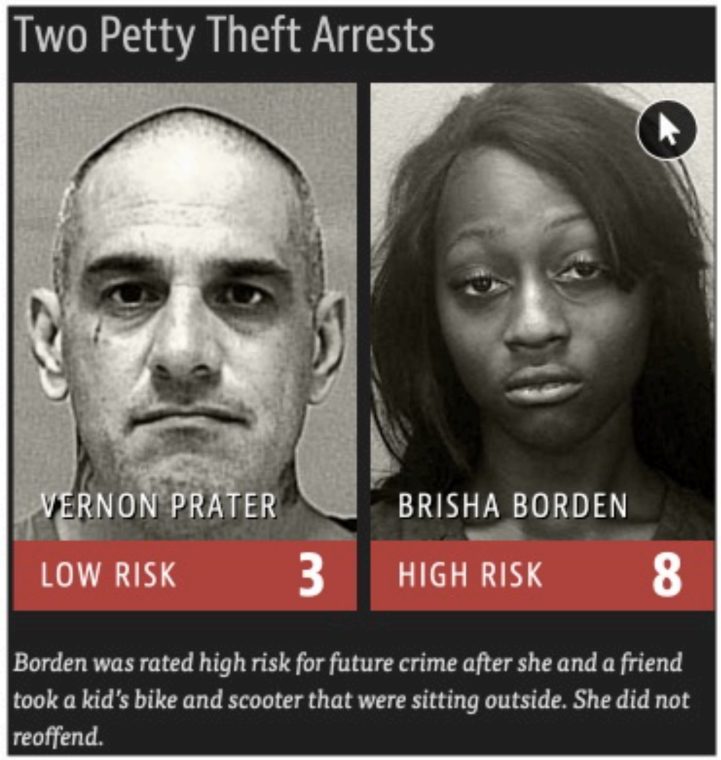
The misclassification may be due to :

- insertion of special crafted noise
- wrong correlation learned in training phase



A. Esteva, et al., "Dermatologist-level classification of skin cancer with deep neural networks," Nature, 2017.

A related problem: fairness
ML can amplify bias



Non recidivating black people twice as likely to be labelled high risk than non recidivating white people

https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Necessity for explanability

2018

The data subject should have the right not to be subject to a decision, which may include a measure, evaluating personal aspects relating to him or her which is based solely on automated processing and which produces legal effects concerning him or her or similarly significantly affects him or her, such as automatic refusal of an online credit application or e-recruiting practices without any human intervention.

...

In any case, such processing should be subject to suitable safeguards, which should include specific information to the data subject and the right to obtain human intervention, to express his or her point of view, to obtain an explanation of the decision reached after such assessment and to challenge the decision.

ML models as FOL constraints

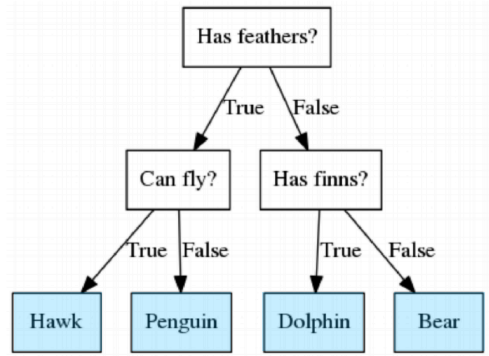
- Help understanding the model
- Constraints can be manipulated
  - queries
  - check implications (derivation of new constraints)
  - satisfiability
  - ....
- Addition of new constraints
  - new knowlege
  - potential attackers
  - fairness desiderata
  - ...

Summary

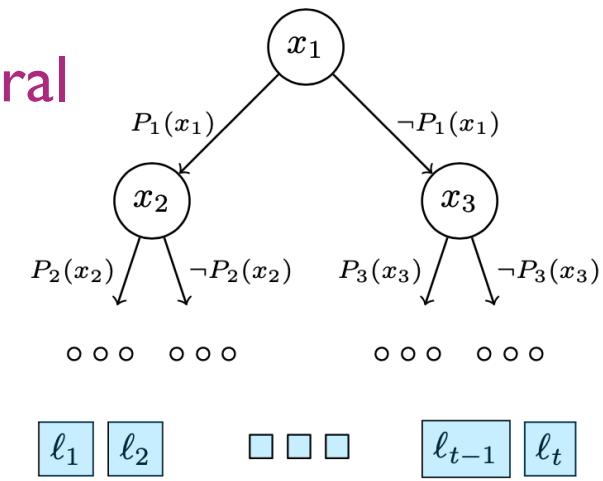
- Machine learning and explainability
- FOL theories from ML models
  - Decision trees
  - Support vector machines
  - Nearest neighbors
  - Neural networks
- Applications to security and privacy
  - Side channel attack
  - Malware
  - Model inversion

Decision trees

Example



General case



(a) A binary decision tree.

$$P_1(X_1) \wedge P_2(X_2) \wedge \ldots \wedge P_l(X_l) \implies L = \ell_1$$

$$P_1(X_1) \wedge P_2(X_2) \wedge \ldots \wedge \neg P_l(X_l) \implies L = \ell_2$$

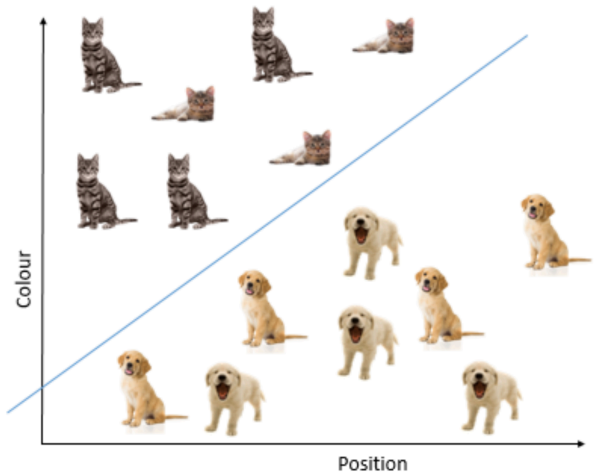
$$\vdots$$

$$\neg P_1(X_1) \wedge \neg P_3(X_3) \wedge \ldots \wedge P_r(X_r) \implies L = \ell_{t-1}$$

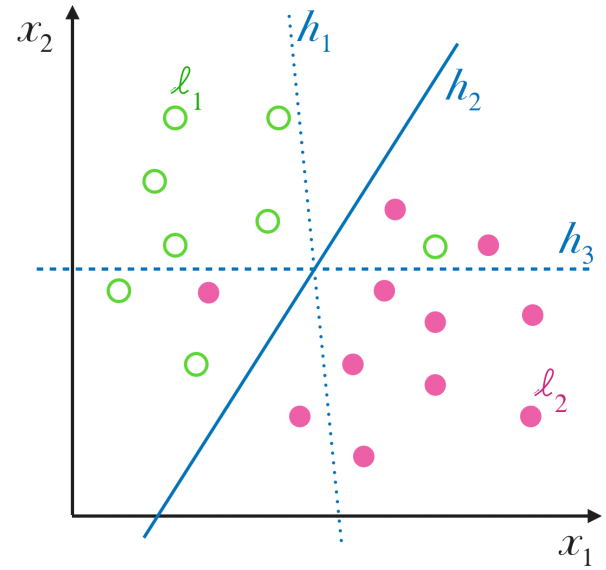
$$\neg P_1(X_1) \wedge \neg P_3(X_3) \wedge \ldots \wedge \neg P_r(X_r) \implies L = \ell_t$$

(b) The formulae derived from the decision tree.

Support vector machines



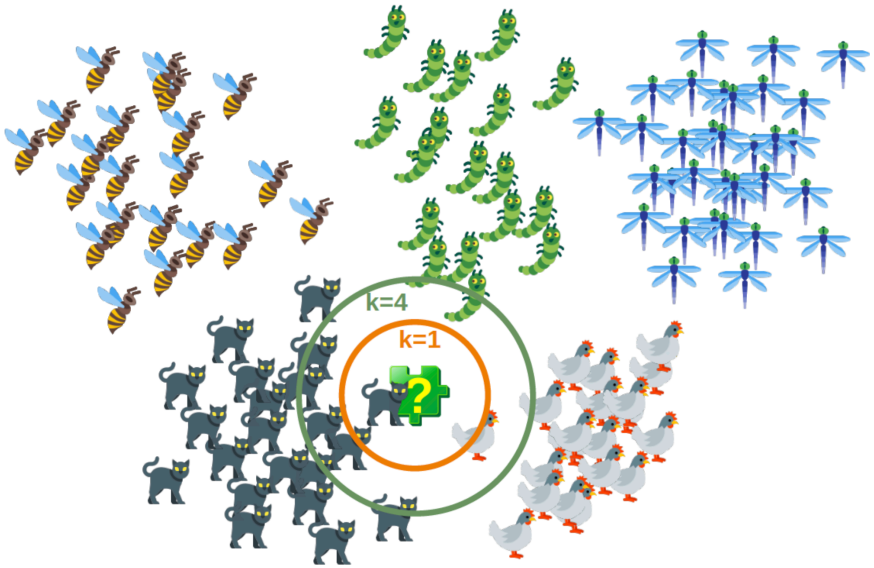
Example



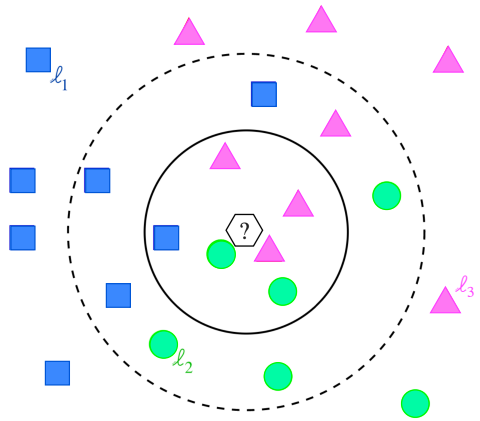
General case

Constraints $\vec{w} \cdot \vec{X} < a \Rightarrow L = \ell_1$ $\vec{w} \cdot \vec{X} > a \Rightarrow L = \ell_2$

Nearest neighbors



Example



General case

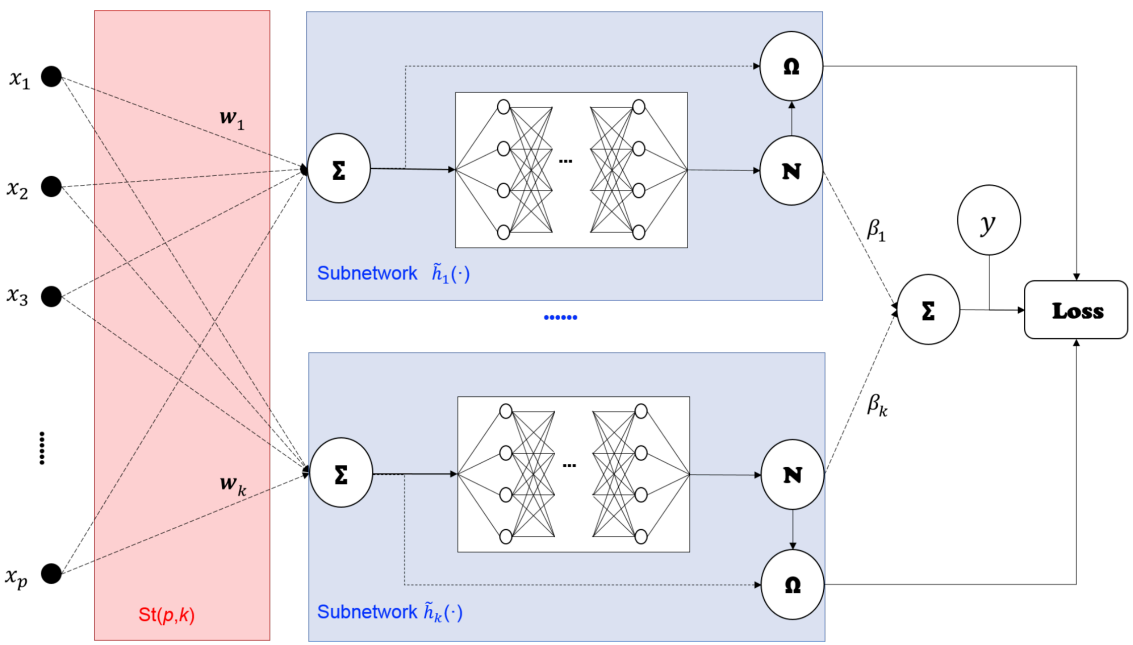
Constraints $\forall \vec{w} \neq \vec{v}_1, \ldots \vec{v}_k.\ (d(\vec{X}, \vec{v}_1) \leq d(\vec{X}, \vec{w}) \wedge \ldots \wedge d(\vec{X}, \vec{v}_k) \leq d(\vec{X}, \vec{w})) \Rightarrow L = \ell$

Neural networks

Yang et al. *Explanaibility of NN through architecture constraints*. IEEE TNNLS 2020

Idea: decompose the network in different sub-networks

The weights of the sub-networks are orthogonal, so that the contribution of each sub-network can be recognized in the result



Loss function

$$\mathbb{E}(y|\boldsymbol{x})) = \mu + \sum_{j=1}^{k} \beta_j \tilde{h}_j(\boldsymbol{w}_j^T \boldsymbol{x}) + \varepsilon,$$

Constraints

$$\sum_{i=1}^{p} |W_{ij}| \leq T_1,$$

$$\sum_{j=1}^{k} |\beta_j| \leq T_2,$$

$$\boldsymbol{W}^T \boldsymbol{W} = \boldsymbol{I}_k,$$

Summary

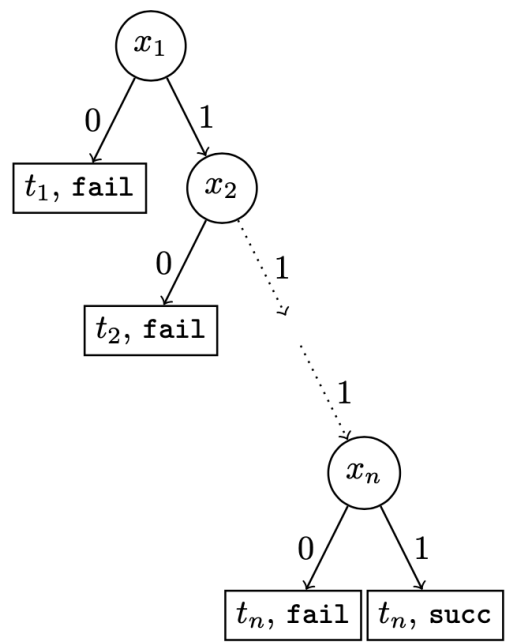
- Machine learning and explainability
- FOL theories from ML models
  - Decision trees
  - Support vector machines
  - Nearest neighbors
  - Neural networks
- Applications to security and privacy
  - Side channel attack
  - Malware
  - Model inversion

Side channel attack

Example: Password checker

The program checks the enterest string bit by bit against the stored password, and fails as soon as it finds a mismatch.

Of course, if the attacker is capable to measure the time until failure, then there is a timing attack. The goal is to detect it authomatically



(a) The decision tree for the password example.

$$X_1 = 0 \implies Time = t_1 \wedge Result = \texttt{fail}$$

$$X_1 = 1 \wedge X_2 = 0 \implies Time = t_2 \wedge Result = \texttt{fail}$$

$$\vdots$$

$$X_1 = 1 \wedge X_2 = 1 \wedge \ldots \wedge X_n = 0 \implies Time = t_n \wedge Result = \texttt{fail}$$

$$X_1 = 1 \wedge X_2 = 1 \wedge \ldots \wedge X_n = 1 \implies Time = t_n \wedge Result = \texttt{succ}$$

(b) The set of constraints derived from the decision tree for the password example.

Side channel attack

Idea: represent the capabilities and the knowledge of the attacker by contraints

$$(X_1 = 1 \wedge X_2 = 1 \wedge \ldots \wedge X_m = 1) \wedge$$
$$(Time = t_1 \vee Time = t_2 \vee \ldots \vee Time = t_n) \wedge (Result = \texttt{fail} \vee Result = \texttt{succ})$$

There are n solutions for this constraint (in the context of the theory expressed in previous slide), which means that the complexity of the attack is linear.

If the time information were not present, then the number of solution (and hence the attack) would be exponential

Malware

Example: a smart healtcare System

$$Pulse > 160 - 0.5\,Age \quad \Rightarrow \quad Hearth\_alarm = \texttt{on}$$

$$\left(\begin{array}{c} ((Sex = \texttt{female} \wedge Weight > .85(Height - 100)) \\ \vee \\ (Sex = \texttt{male} \wedge Weight > .90(Height - 100))) \\ \wedge \\ Blood\_pressure > 130 \end{array}\right) \quad \Rightarrow \quad Diabetes\_alarm = \texttt{on}$$

A possible attacker able to tamper with the Blood-pressure indicator

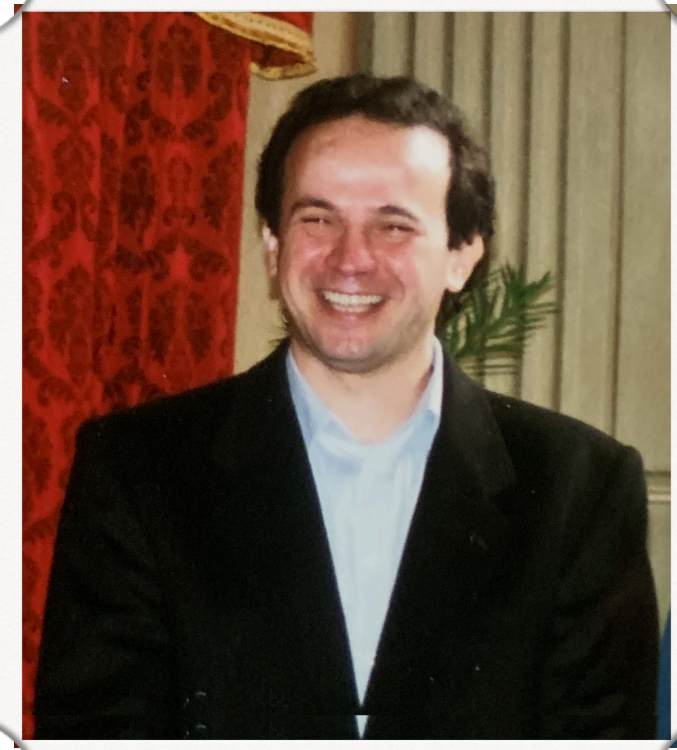
$$R - 20 \leq Blood\_pressure \leq R + 20 \wedge Diabetes\_alarm = \texttt{off}$$

Security breach:
the following scenario becomes satisfiable for any 130< r < 150

$$Sex \doteq \texttt{female}\ ,\ Weight \doteq 80\ ,\ Height \doteq 165\ ,\ Age \doteq 40\ ,$$
$$R \doteq r\ ,\ Blood\_pressure \doteq r - 20\ ,\ Diabetes\_alarm \doteq \texttt{off}$$

Thank you !

Questions?



And, of course,

Happy Birthday, Mau!