Structured Pruning of Neural Networks for Constraint Learning

Andrea Lodi
Cornell Tech and Technion - IIT
andrea.lodi@cornell.edu
Joint work with Matteo Cacciola & Antonio Frangioni

LEANOPT-24
AAAI Workshop on Learnable Optimization
February 26, 2024

CANADA EXCELLENCE RESEARCH CHAIR
DATA SCIENCE FOR REAL-TIME DECISION-MAKING

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

CORNELL TECH | HOME OF THE JACOBS TECHNION-CORNELL INSTITUTE

Preamble

The use of Machine Learning (ML) for Combinatorial Optimization (CO) problems has been ubiquitous in the last 5 years at the very least.

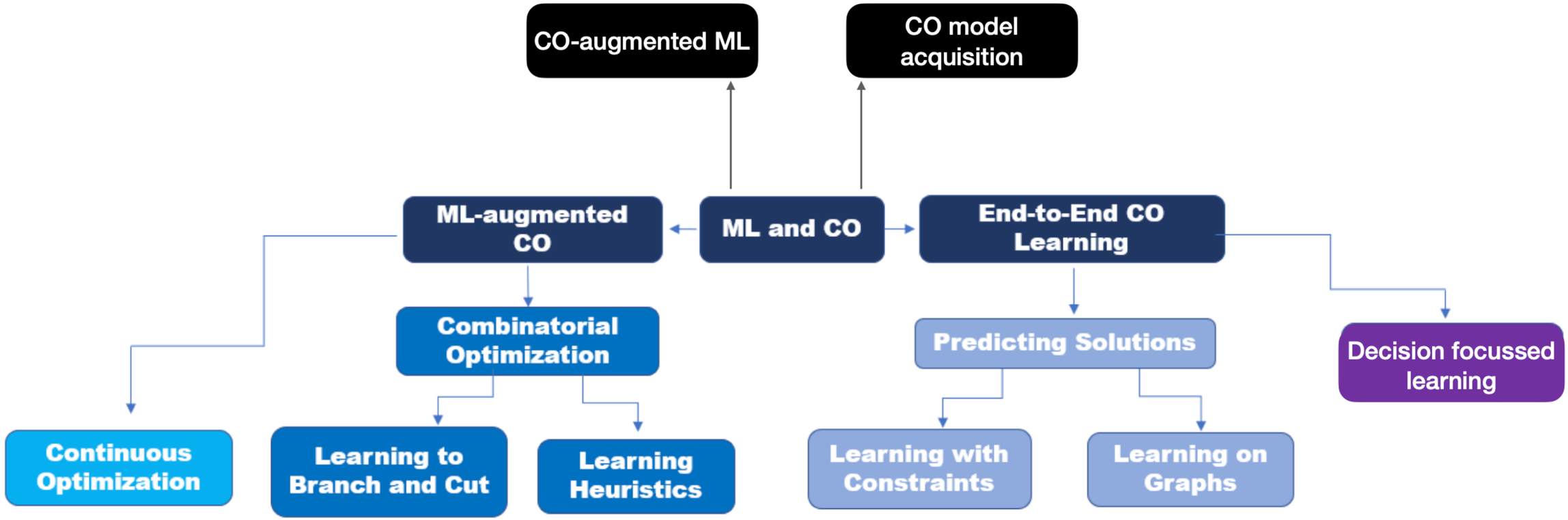
This is due to the impressive success of ML, especially deep learning, in beating human capabilities in image recognition, language processing and sequential games.

Those successes led to ask natural questions about using modern statistical learning in other disciplines, Combinatorial Optimization being one of them.

We have witnessed many examples of this trend, where both directions in the interplay between ML and CO are taken into account.

Schematic Overview

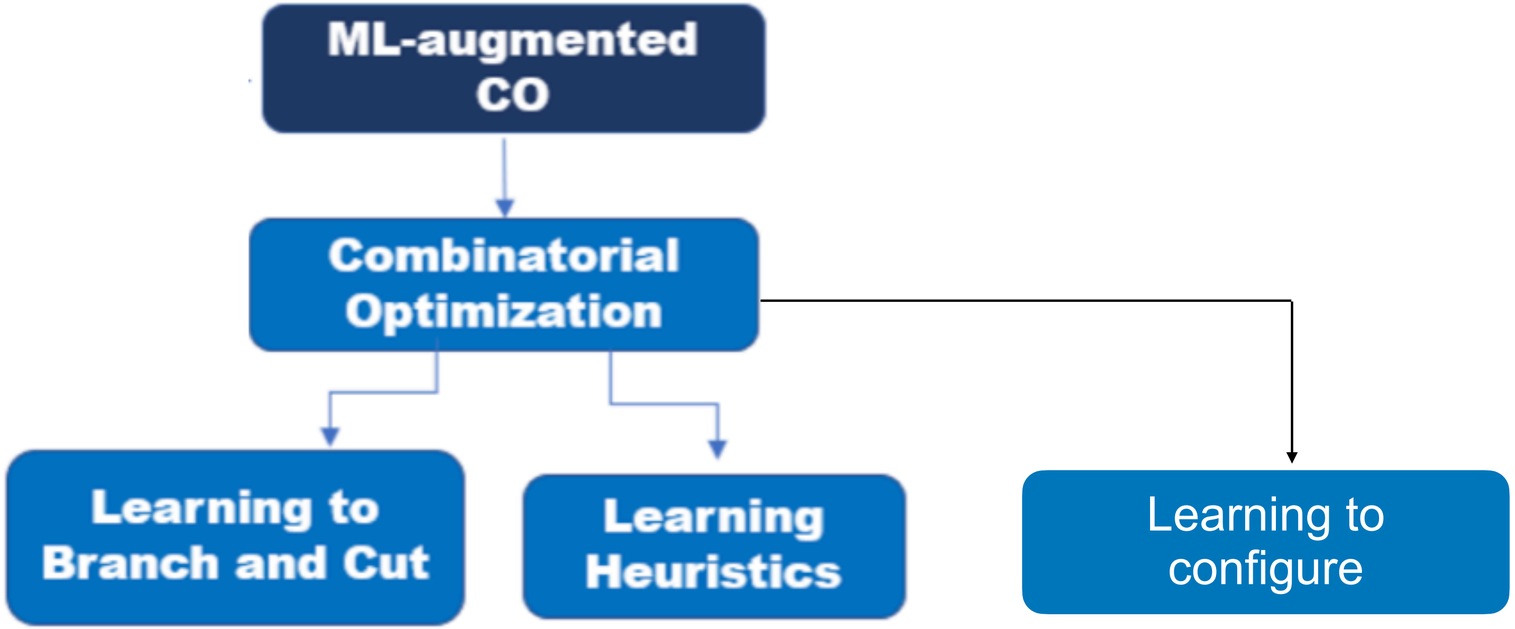
Slide courtesy of N. Yorke-Smith



Y. Bengio, A. Lodi, A. Prouvost: Machine Learning for Combinatorial Optimization: a Methodological Tour d'Horizon, EJOR 2021, 405-421

J. Kotary, F. Fioretto, P. Van Hentenryck, B. Wilder: End-to-End Constrained Optimization Learning: A Survey. IJCAI 2021:4475-4482

ML-augmented CO (or MILP)



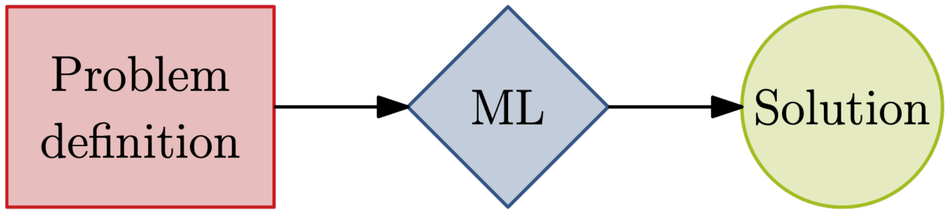
L. Scavuzzo, K. Aardal, A. Lodi, N. Yorke-Smith: Machine Learning Augmented Branch and Bound for Mixed Integer Linear Programming, arXiv:2402.05501, 2024.

Outline

- Today, we will talk about a slightly more intricated interplay, where
    - ML helps learning the CO model, but
    - CO helps reducing the size of the ML learned model without losing accuracy.

M. Cacciola, A. Frangioni, A. Lodi,
Structured Pruning of Neural Networks for Constraints Learning,
arXiv:2307.07457, 2023.

End to End Learning

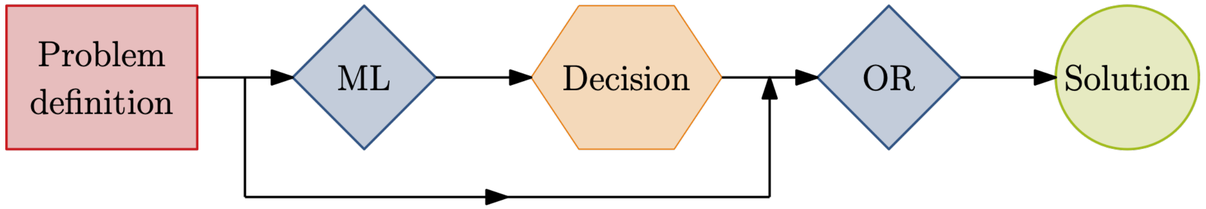


- Learning TSP solutions
  [*Bello et al. 2017*][*Kool and Welling 2018*][*Emami and Ranka 2018*]
  [*Vinyals et al. 2015*][*Nowak et al. 2017*]

- Predict aggregated solutions to MILP under partial information
  [*Larsen et al. 2018*]

- Approximate obj value to SDP (for cut selection)
  [*Baltean-Lugojan et al. 2018*]

Learning Properties



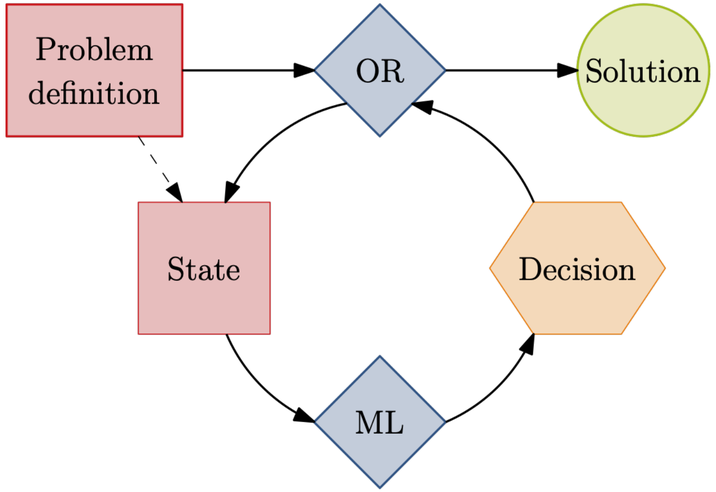
- Use a decomposition method
  [*Kruber et al. 2017*]

- Linearize an MIQP
  [*Bonami et al. 2018*]

- Provide initial cancer treatment plans to inverse optimization
  [*Mahmood et al. 2018*]

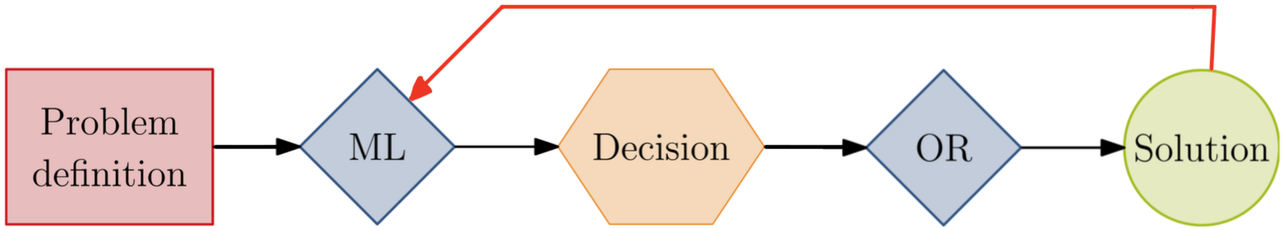
Learning Repeated Decisions

- Learning where to run heuristics in B&B
  [*Khalil et al. 2017b*]
- Learning to branch
  [*Lodi and Zarpellon 2017*] (survey)
- Learning gradient descent
  e.g. [*Andrychowicz et al. 2016*]
- Predicting booking decisions under imperfect information
  [*Larsen et al. 2018*]
- Learning cutting plane selection
  [*Baltean-Lugojan et al. 2018*]



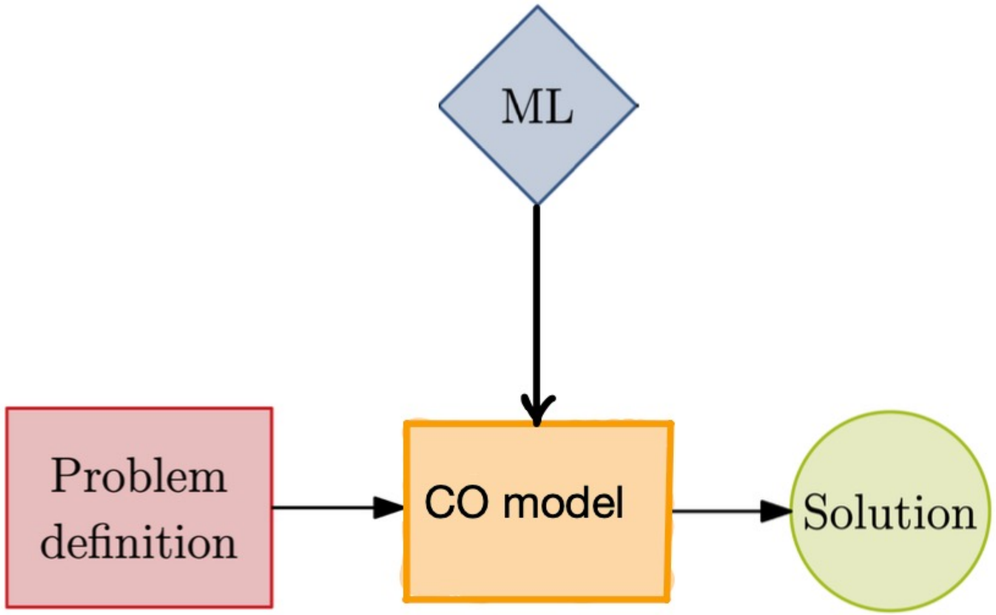
just a matter of viewpoint

Decision-aware learning



- Quadratic Programming
  [Amos and Kolter 2017]
- Novel loss functions development
  [Elmachtoub and Grigas 2022] [Mandi et al. 2022]
- Perturbation or penalization methods
  [Mandi and Guns 2020] [Blondel et al. 2020] [Wilder et al. 2019]
- Learning with combinatorial optimization layers
  [Dalle et al. 2022]

Constraint Learning



- Empirical decision model learning
  [*Lombardi and Milano 2017*]
- JANOS: integrated predictive & prescriptive modeling
  [*Bergman et al. 2020*]
- MIP with constraint learning
  [*Maragno et al. 2021*]

Constraint Learning: an Example

- Allocate scholarships to admitted students.
- Need to model the probability of a student to enroll, i.e., accepting the offer.
- This is correlated with student GPA, SAT, received scholarship offer, etc.
- An ML model can be trained to approximate this correlation:

$$p = g(GPA, SAT, offer)$$

- If we want to solve the problem by CO, we need a model of the function $g( )$.

Constraint Learning: Modeling

- If the function *g( )* is an Artificial Neural Network (ANN), we have some Mixed-Integer Programming (MIP) models from the literature (Fischetti and Jo 2018, Anderson et al. 2020).

- In most cases, the relation among the output of two consecutive layers is the ReLu activation function

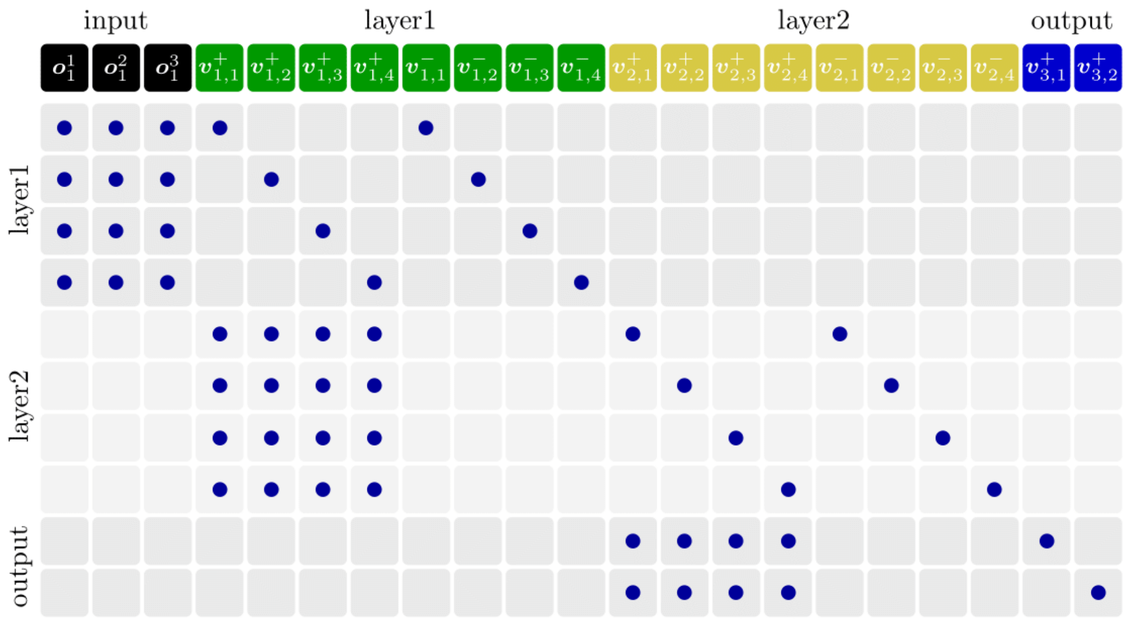
$$o_{l+1} = \max(0, W_l o_l + b_l)$$

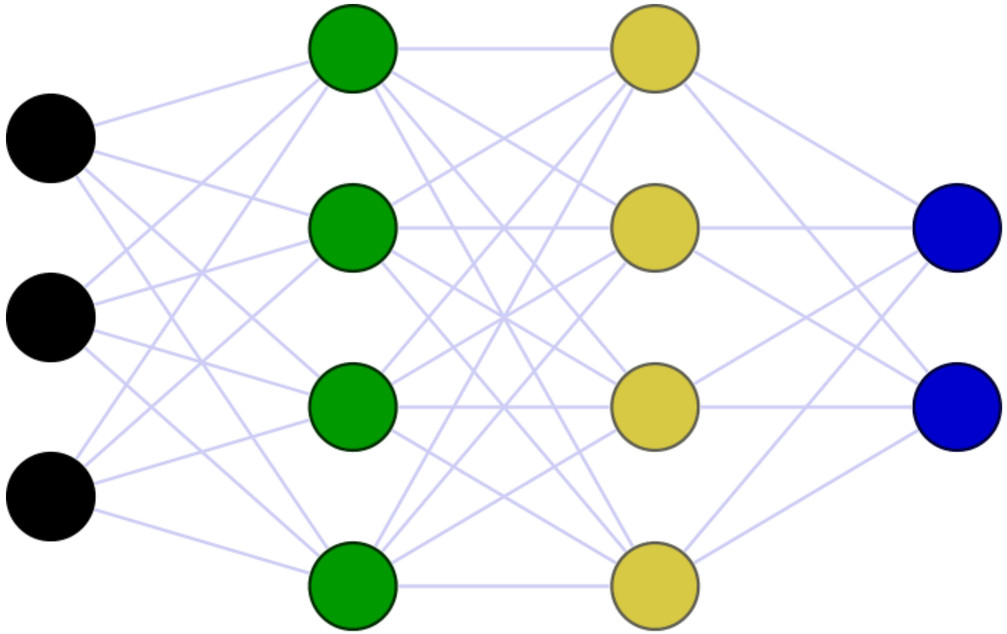
- So, each Neuron of the network has the following formulation:

$$\begin{aligned}
& v_l^+ - v_l^- = W_l o_l + b_l \\
& v_l^+ \leq M^+ z_l \\
& v_l^- \leq M^- (1 - z_l) \\
& 0 \leq v_l^+ \leq M^+ \\
& 0 \leq v_l^- \leq M^- \\
& z_l \in \{0,1\}^m
\end{aligned}$$

Constraint Learning: Modeling (cont.d)

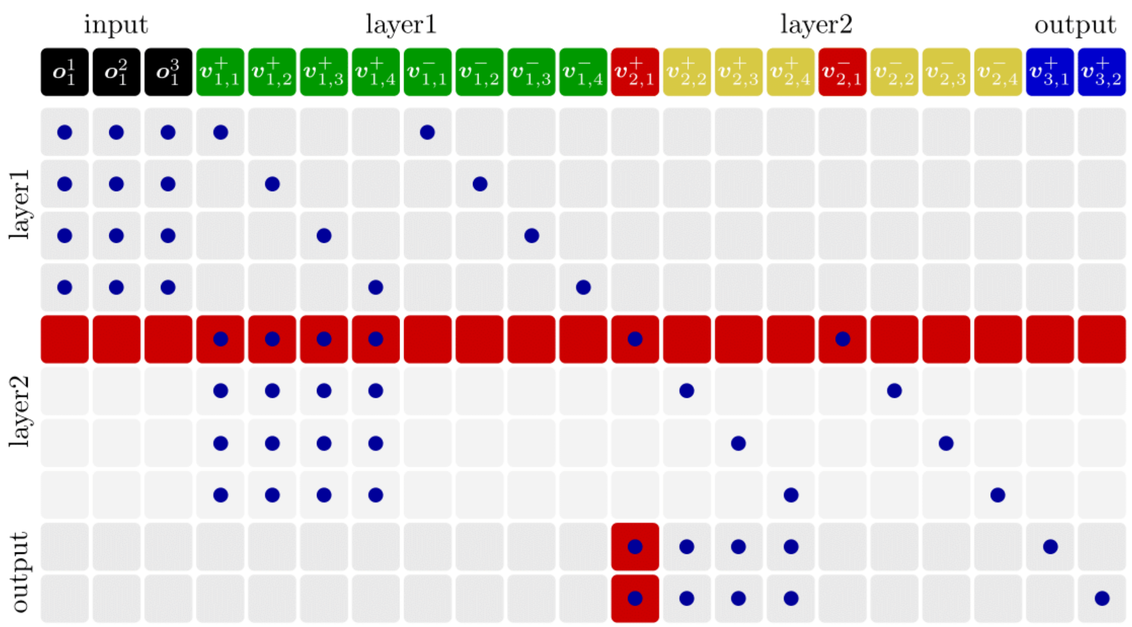
- The constraint matrix for a simple architecture looks like this

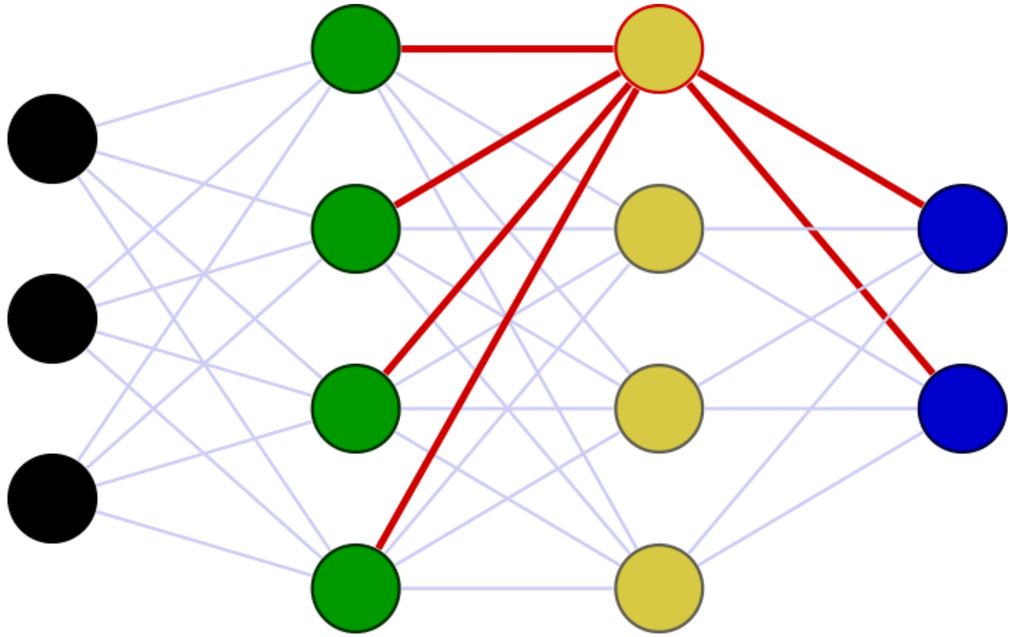




Constraint Learning: Modeling (cont.d)

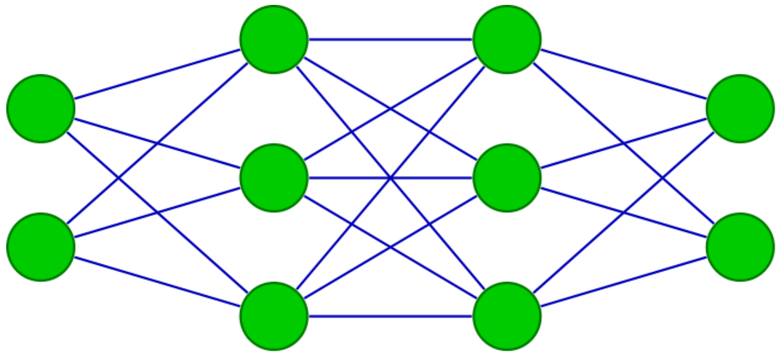
- The constraint matrix for a simple architecture looks like this

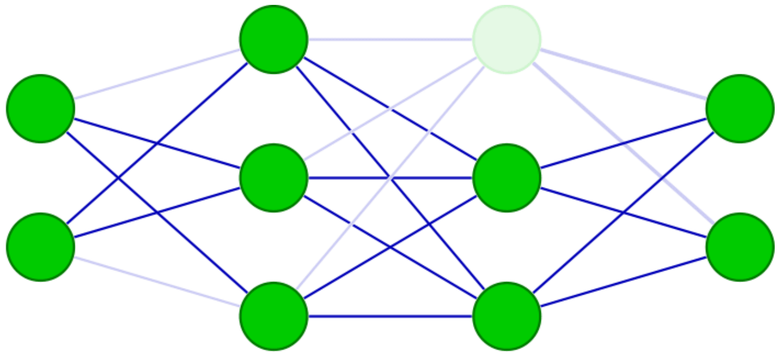




ANN Pruning

- Model Compression for Edge Computing.
- Quantization, Knowledge Distillation, etc.
- Pruning: removing parameters without affecting performance, i.e., accuracy.





ANN Pruning (cont.d)

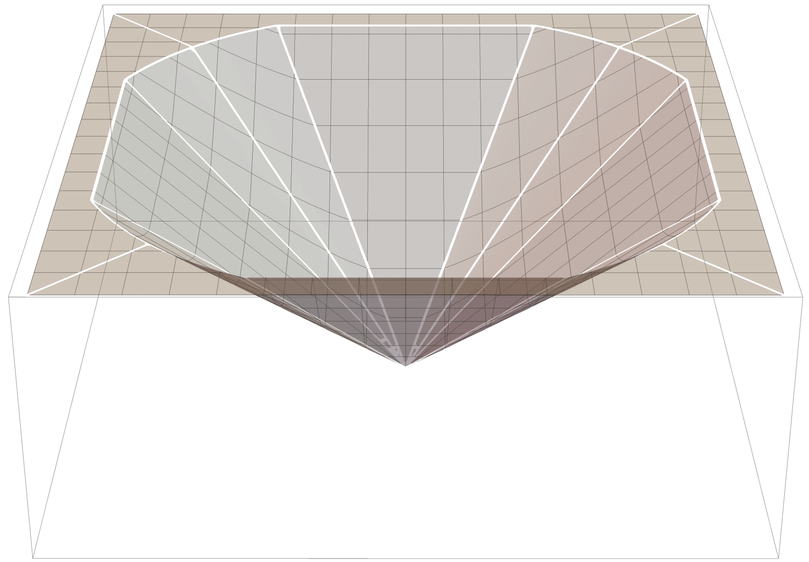
- A potential approach for pruning is Pruning through Regularization

$$\min L(g(X), Y) + R(w)$$

- In Cacciola, Frangioni and Lodi, arXiv:2206.14056, 2022, such a regularization is obtained in two steps:

- First, explicitly using the L_0 norm to penalize the nonzero weights;
- Second, taking the perspective reformulation of the linear programming relaxation of the obtained formulation.

- Structured Perspective Regularization (SPR).



Mathematical model

- $W = \cup_i E_i$ is the set of model parameters
- X is the training set

$$\min L(X, W) + \lambda \left[\alpha \sum_{j \in I} w_j^2 + (1 - \alpha) \sum_{i=1}^{N} y_i \right]$$

$$-My_i \leq w_j \leq My_i \qquad \forall w_j \in E_i,\ i = 1, \ldots, N \qquad (1)$$

$$y_i \in \{0, 1\} \qquad i = 1, \ldots, N \qquad (2)$$

Perspective reformulation

$$\min \left\{ L(X, W) + \lambda \left[\alpha \sum_{i=1}^{N} \sum_{j \in E_i} \frac{w_j^2}{y_i} + (1 - \alpha) \sum_{i=1}^{N} y_i \right] : (1), (2) \right\} \tag{3}$$

A. Frangioni and C. Gentile. Perspective cuts for a class of convex 0–1 mixed integer programs.266 Mathematical Programming, 106(2):225–236, 2006

Final model

$$\min \left\{ L(X, W) + \lambda \sum_{i=1}^{N} z_i(w_i; \alpha, M) \right\}, \tag{4}$$

where $w^i = [w_j]_{j \in E_i}$, and

$$z_i = \begin{cases} 2\sqrt{\alpha(1-\alpha)} \left\| w^i \right\|_2 & \text{if } \frac{\left\| w^i \right\|_\infty}{M} \leq \sqrt{\frac{\alpha}{1-\alpha}} \left\| w^i \right\|_2 \leq 1 \\ \frac{\alpha M}{\left\| w^i \right\|_\infty} \left\| w^i \right\|_2^2 + (1-\alpha) \frac{\left\| w^i \right\|_\infty}{M} & \text{if } \sqrt{\frac{\alpha}{1-\alpha}} \left\| w^i \right\|_2 \leq \frac{\left\| w^i \right\|_\infty}{M} \leq 1 \\ \alpha \left\| w^i \right\|_2^2 + (1-\alpha) & \text{otherwise.} \end{cases}$$

Pruning for Constraint Learning

- Recently, GUROBI has released an open-source Python interface to "import" **trained** ANNs inside MIPs.
- According to the released experiments, despite some nice algorithmic enhancements to improve the resulting MIP formulation (e.g., through OBBT), the bottleneck seems scalability.
- Model Compression can help Constraint Learning to scale better.
- Pruning seems very suited for this goal, especially structured one.
- The idea is to perform neuron pruning on ANNs before embedding them in MIP formulations in order to reduce size and, hopefully, solution times.

Adversarial Example Search

- Given a trained ANN and an input sample, find a new sample in a neighborhood of the first one that is classified differently by the ANN.
- Important problem for ML, directly linked to robustness and interpretability.
- We used some Feed-Forward Neural Networks and the MNIST dataset

$$
\begin{aligned}
&\max y_h - y_k \\
&s.t.\ y = g(\bar{x}) \\
&\quad \Delta \geq x - \bar{x} \\
&\quad \Delta \geq \bar{x} - x \\
&\quad \bar{x} \in \mathrm{R}^n
\end{aligned}
$$

- where Δ represents the max distance between the input example and the one to find.

Experiments

Δ = 5					
Arch	λ-α	Acc.	Time	Nodes	Pruned Arch
2x50 Baseline		97.55	14.88	5820	
		97.47	20.65	12040	
		97.25	8.07	9497	
2x50 Pruned	0.5-0.9	97.77	3.29	3328	1x39-1x43
		97.49	3.93	6482	1x30-1x42
		97.73	1.96	3992	1x39-1x42
2x100 Baseline		97.96	39.29	2971	
		97.76	35.01	3112	
		97.97	39.00	3019	
2x100 Pruned	0.5-0.9	98.08	15.99	3066	1x61-1x80
		98.01	15.65	3107	1x61-1x87
		98.04	17.67	2951	1x63-1x86
2x200 Baseline		98.14	1800.37	424758	
		98.04	1800.18	401361	
		97.95	781.90	58656	
2x200 Pruned	0.5-0.5	97.96	18.66	3029	1x56-1x144
		98.13	24.65	3600	1x57-1x144
		98.04	28.19	2997	1x59-1x140
6x100 Baseline		97.60	474,76	15261	
		97.77	1800.02	798306	
		97.67	818.19	14334	
6x100 Pruned	1.0-0.1	97.52	231.02	3173	1x39-1x82 2x61-1x60-1x54
		97.47	79.22	7566	1x44-1x71-1x49 -1x53-1x49-1x45
		97.21	44.24	11417	1x37-1x72-1x48 -1x51-1x48-1x46

Table 1: Results using Δ = 5.

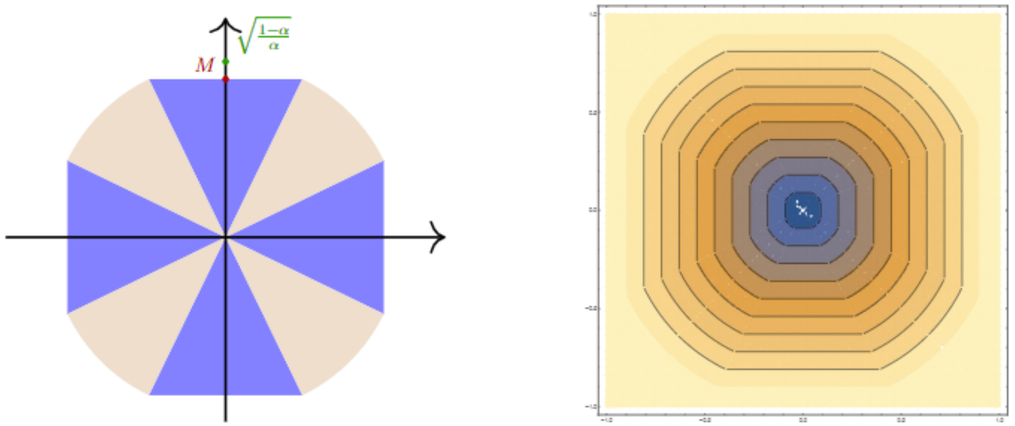
Δ = 20				
Arch	λ-α	Time	Nodes	Found
2x50 Baseline		1.85	1	YES
		5.06	1221	YES
		1.66	1	YES
2x50 Pruned	0.5-0.9	2.73	127	YES
		2.90	1128	YES
		0.64	1	YES
2x100 Baseline		17.35	1217	YES
		147.67	7532	YES
		102.67	3252	YES
2x100 Pruned	0.5-0.9	6.66	2079	YES
		6.03	127	YES
		2.16	1	YES
2x200 Baseline		439.17	40075	YES
		563.24	17597	YES
		508.14	6014	YES
2x200 Pruned	0.5-0.5	2.56	1	YES
		18.65	5433	YES
		9.60	1202	YES
6x100 Baseline		1800.06	138918	NO
		1800.03	237328	NO
		1800.10	184954	NO
6x100 Pruned	1.0-0.1	15.53	1	YES
		7.51	1	YES
		129.70	27045	YES

Table 2: Results using Δ = 20.

Summary

- Overall, we start to have (solid) evidence that ML / CO integration approaches could be effective.
- We also start to see some intricate algorithmic synergies as in the presented case of Constraint Learning and its use in MIP.
- Probably my favorite “practical” topic is to “learn” to repeatedly solve the “same instance”.

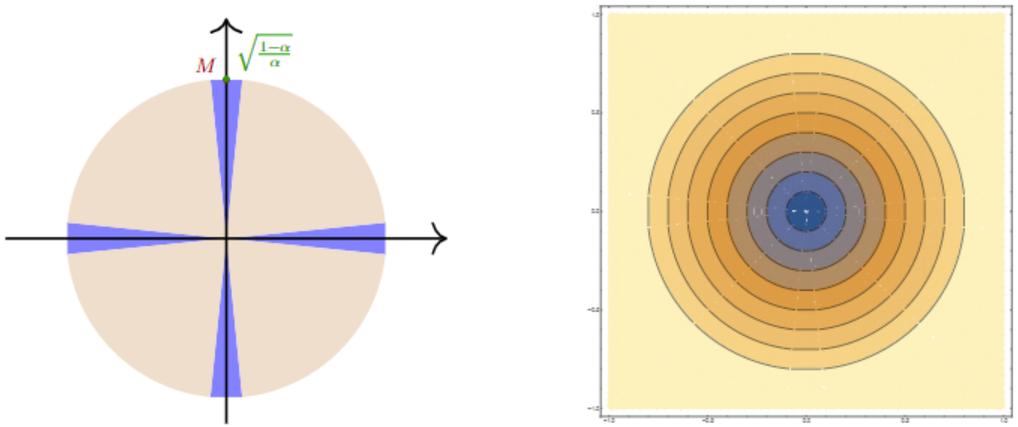
Visualization of SPR



(a) $M = 0.9$, $\alpha = 0.5$

Left, regions in which the SPR changes definition, right level sets of the structured sparsity term of the SPR

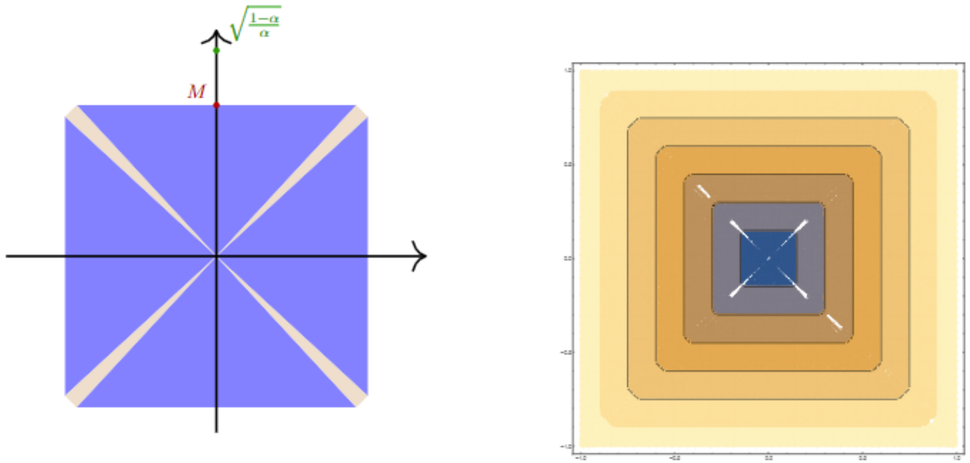
Visualization of SPR



(b) $M = 0.9$, $\alpha = 0.55$

Left, regions in which the SPR changes definition, right level sets of the structured sparsity term of the SPR

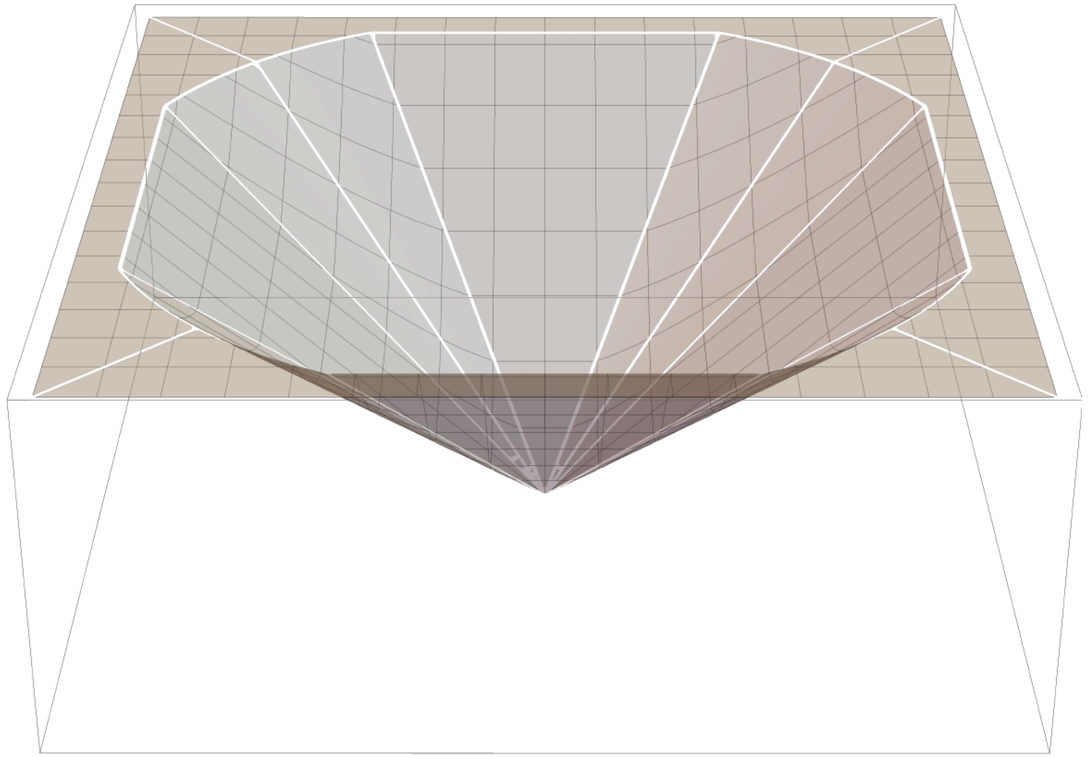
Visualization of SPR



(c) $M = 0.9$, $\alpha = 0.4$

Left, regions in which the SPR changes definition, right level sets of the structured sparsity term of the SPR

Visualization of SPR



3-dimensional plot of the structured sparsity term of the SPR. When the norm of the entity is big enough, the term is constant. Otherwise, it is more similar to the l_2 or l_∞ norm, based on how are distributed the weights in the entity.