QPLayer: a generic and efficient approach for differentiating quadratic programming problems

A. Bambade[1,2,3] F. Schramm[1,2] A. Taylor[1,2] J. Carpentier[1,2]

[1]Inria Paris, France [2]Département d'informatique de l'ENS, PSL Research University, Paris, France [3]École des Ponts, Marne-la-Vallée, France

Quadratic Programming Differentiation

Objective: differentiate closest feasible **Quadratic Programs** (QPs) solution, under **robustness**, **accuracy** and **speed** requirements

$$s^\star(\theta) = \arg\min_{s\in\mathbb{R}^{n_i}} \tfrac{1}{2}\|s\|_2^2 \quad \text{s.t. } x^\star(\theta), z^\star(\theta) \in \arg\min_{x\in\mathbb{R}^n} \max_{z\in\mathbb{R}^{n_i}_+} L(x,z,s;\theta), \qquad \text{(QP-H}(\theta))$$

with $L(x,z,s;\theta) \triangleq \frac{1}{2}x^\top H(\theta)\,x + x^\top g(\theta) + z^\top (C(\theta)x - u(\theta) - s)$, $H \in \mathcal{S}_+(\mathbb{R}^d)$ (i.e., symmetric positive semi-definite), $g \in \mathbb{R}^d$ $C \in \mathbb{R}^{n_i \times d}$, $u \in \mathbb{R}^{n_i}$ and $s \in \mathbb{R}^{n_i}$ smallest in ℓ_2 norm. When feasibility enforced: $s^\star(\theta) = 0$, and equivalent to classic convex QPs.
Typically: $\theta = \{H, g, C, u\}$, which is not necessarily **feasible**.

Why is it relevant?

- QP: ubiquitous tool, optimization differentiation provides inductive knowledge
- Yet, differentiation is often not possible, which restricts layer architecture and the approach expressivity

Practical relevance of Quadratic Programming

- Robotics (e.g., motion planning, whole-body control, optimization-based control, etc.)
- beyond Robotics (control, machine learning, etc.)

Contributions

- **An extended and unified approach** for differentiating robustly and quickly closest feasible QP solutions
  - Working guarantees under current used assumptions [1, 2]
  - Least square estimate otherwise [3],
- **A dedicated C++ solver** with forward and backward modes integrated in **ProxSuite** and linkable with PyTorch.

Our approach

Design an extensive conservative Jacobian definition

- **Closest feasible QP KKT**
  We show that canceling this path differentiable map G is equivalent to solving Problem QP-H(θ)

$$G(x,z,t;\ \theta) := \begin{bmatrix} H(\theta)x + g + C(\theta)^\top z \\ C(\theta)x - u(\theta) - t \\ [[t]_- + z]_+ - z \\ C(\theta)^\top [t]_+ \end{bmatrix}, \qquad \text{(G)}$$

  At optimality, $s^\star = [t^\star]_+$.
- **Closest feasible solution conservative Jacobians**
  Considering, $v^\star := (x^\star, z^\star, t^\star)$ a solution to QP-H(θ), we define its Extended Conservative Jacobians (ECJ) as follows

$$\frac{\partial x^\star}{\partial \theta}, \frac{\partial z^\star}{\partial \theta}, \frac{\partial t^\star}{\partial \theta} \in \arg\min_w \left\| \frac{\partial G(x^\star, z^\star, t^\star;\ \theta)}{\partial v^\star} w + \frac{\partial G(x^\star, z^\star, t^\star;\ \theta)}{\partial \theta} \right\|_2^2, \qquad (1)$$

- **Well-posedness**
  If the Problem (**??**) is feasible and satisfies standard assumption, we recover standard implicit differentiation.

Efficient derivation

- **Forward pass**
  We leverage Augmented Lagrangian capability to converge naturally towards the closest feasible problem solution [4].
  We use ProxQP as a backend [5] based on revisited primal-dual augmented Lagrangian methods.
- **Backward pass general case**
  Considering a loss $\mathcal{L}$, if we can solve the equality constrained QP, then following [1]

$$\min_{b_1,b_2,b_3,b_4} \ 0 \qquad (2)$$

$$\text{s.t.} \begin{bmatrix} H & C^\top & 0 & 0 \\ C & 0 & (I-\Pi_1) & 0 \\ 0 & -I & -\Pi_1\Pi_2 & (1-\Pi_2)C \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = -\begin{bmatrix} \frac{\delta \mathcal{L}}{\delta x} \\ \frac{\delta \mathcal{L}}{\delta z} \\ \frac{\delta \mathcal{L}}{\delta t} \end{bmatrix}, \qquad (3)$$

  the ECJs are recovered with simple update rules (Π_1 and Π_2 are diagonal matrices). Otherwise, we use forward mode. If primal feasibility is enforced we show it simplifies to a small well conditioned linear system.

Benchmarks

	OptNet	QPLayer
Forward pass (ms)	615.84 $\pm$ 16.15	55.2 $\pm$ 6.93
Backward pass (ms)	61.26 $\pm$ 2.84	39.27 $\pm$ 2.17
Final Loss	0.02604	0.02556

Table: Average computational times (over 800 epochs) for solving a cart-pole example with friction when using OptNet or QPLayer. Randomized smoothing is used for obtaining informative gradients [6].

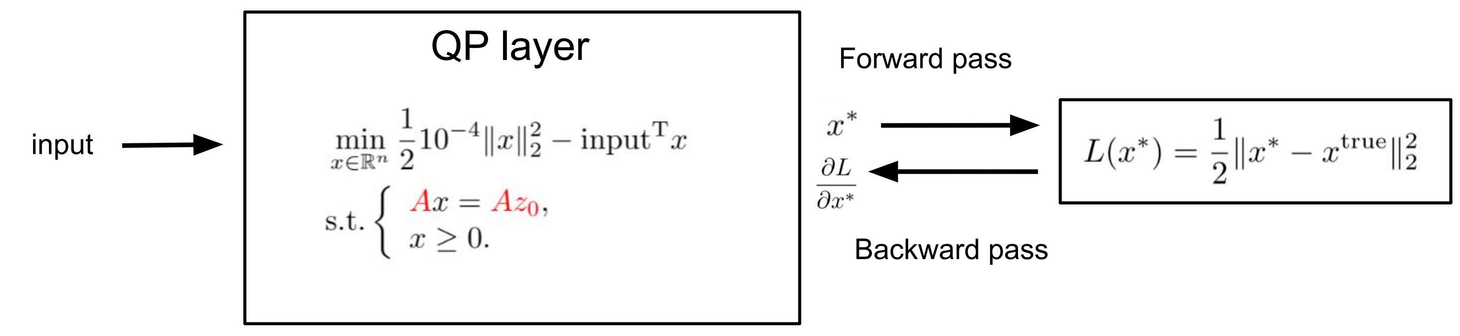

Figure: Strictly convex QP layer enforced to be primal feasible. We learn here A and $z_0 > 0$ to solve Sudoku problems.

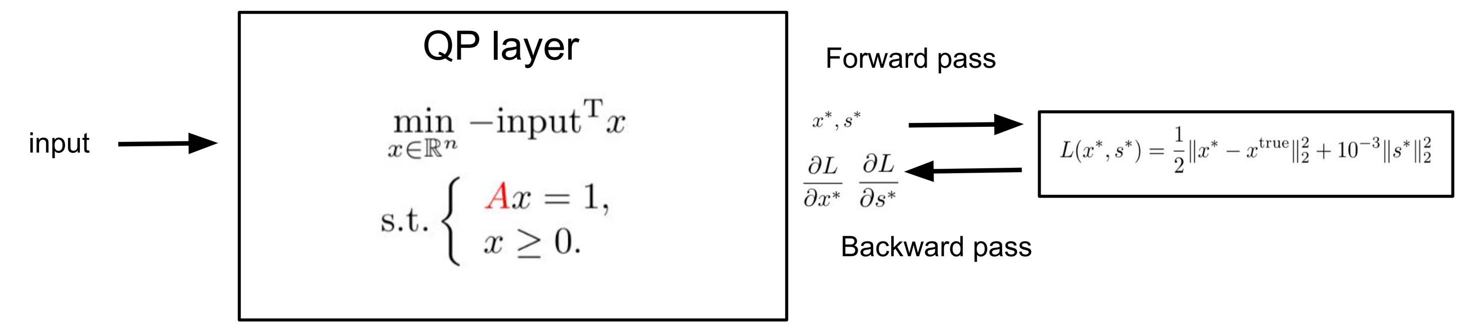

Figure: LP layer. We learn here only A to solve Sudoku problems. The QPLayer approach enables learning a feasible layer.

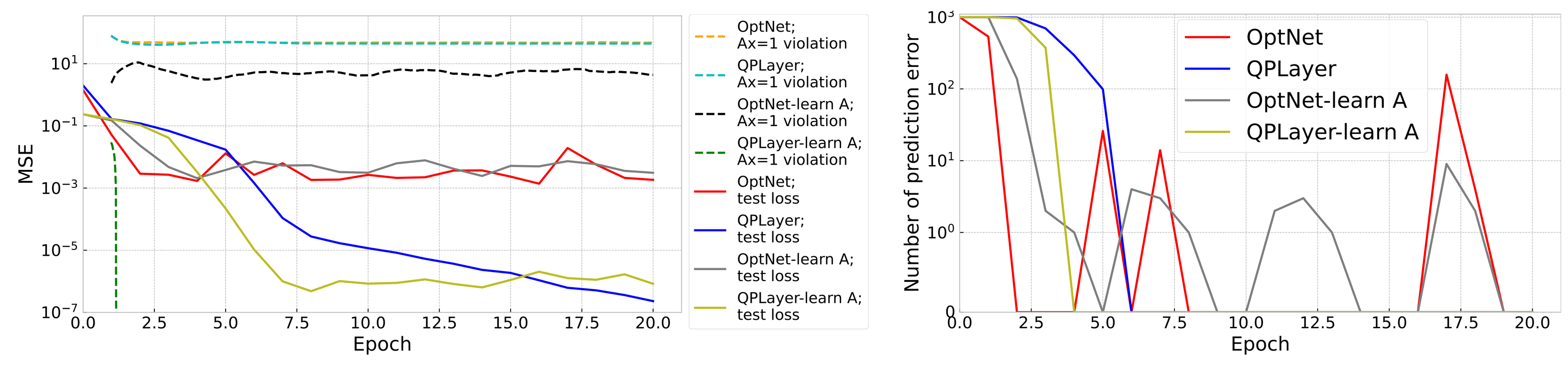

Figure: Sudoku training and test plots using QPLayer and OptNet layers. QPLayer can learn LPs, whereas OptNet is restricted to strictly convex QPs, which limits its representational power. QPLayer can also be specialized to learn models satisfying specific linear constraints.

Future work

- Extension to GPUs, multi CPUs,
- Beyond LPs and QPs: SOCPs, SDPs, etc..

References

1. Amos, B. & Kolter, J. Z. *OptNet: Differentiable Optimization as a Layer in Neural Networks.* 2021.
2. Barratt, S. T. On the Differentiability of the Solution to Convex Optimization Problems. *arXiv: Optimization and Control* (2018).
3. Agrawal, A. *et al.* *Differentiable Convex Optimization Layers.* in *Advances in Neural Information Processing Systems* (eds Wallach, H. *et al.*) **32** (Curran Associates, Inc., 2019).
4. Chiche, A. & Gilbert, J. C. How the augmented Lagrangian algorithm can deal with an infeasible convex quadratic optimization problem. *Journal of Convex Analysis* **23.** https://hal.inria.fr/hal-01057577 (2016).
5. Bambade, A., El-Kazdadi, S., Taylor, A. & Carpentier, J. *PROX-QP: Yet another Quadratic Programming Solver for Robotics and beyond.* in *RSS 2022 - Robotics: Science and Systems* (New York, United States, June 2022). https://hal.inria.fr/hal-03683733.
6. Lidec, Q. L., Montaut, L., Schmid, C., Laptev, I. & Carpentier, J. *Leveraging Randomized Smoothing for Optimal Control of Nonsmooth Dynamical Systems.* 2022.