QPLayer: Leveraging augmented-Lagrangian techniques for differentiating over infeasible quadratic programs

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

[1]Willow team
Inria and ENS Paris

[2]Sierra team
Inria and ENS Paris

PGMO days, 29 November 2023

Table of Contents

Table of Contents

Introduction: QP layers v.s. neural networks

Outputs of current learning pipelines are explicit function of the inputs.

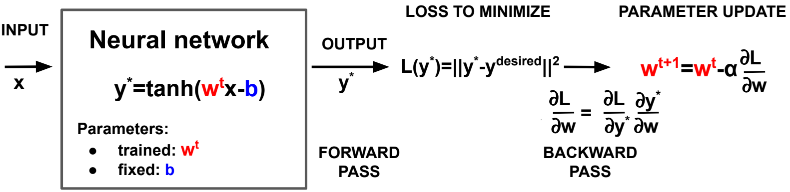


Figure: Example of a feedforward neural network.

Introduction: QP layers v.s. neural networks

More recent literature considers differentiable optimization problems as layers.

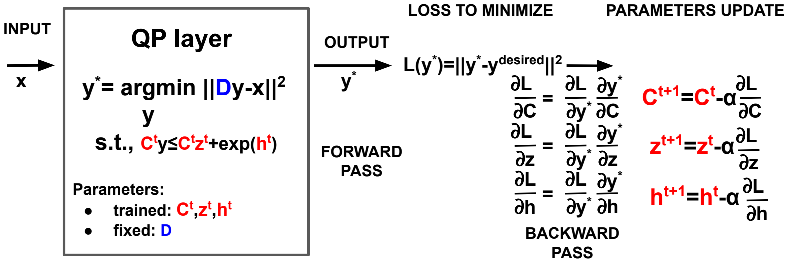


Figure: Example of a Quadratic Programming layer (with D nonsingular).

Introduction: QP layers pros (representative power)

Why using such more complex architecture ?

- For some class of layers, the representative power is similar to those of standard Neural Networks,

Theorem (B. Amos & Z. Kolter (2021))

Let $f : \mathbb{R}^n \to \mathbb{R}^n$ *be an elementwise piecewise linear function with* k *linear regions. Then function can be represented as an OptNet layer using* $O(nk)$ *parameters. Additionally, the layer* $z_{i+1} = \max(Wz_i + b, 0)$ *for* $W \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^m$ *can be represented by an OptNet layer with* $O(mn)$ *parameters.*

Why using such more complex architecture ?

- For some class of layers, the representative power is similar to those of standard Neural Networks,
- For some optimization based problem, it performs better.

QP layers in machine learning: example 1

Using a convex QP as a deep learning layer performs better than a ConvNet for solving Sudokus.

			3
1			
		4	
4			1

2	4	1	3
1	3	2	4
3	1	4	2
4	2	3	1

Figure: Example of Sudoku.

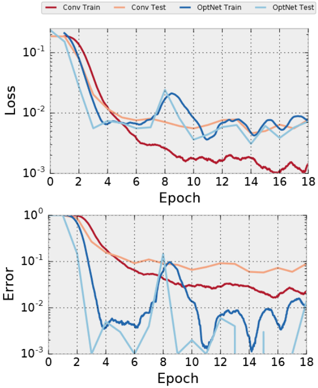


Figure: Sudoku Training plots[1].

[1] Brandon Amos and J. Zico Kolter (2021). OptNet:Differentiable Optimization as a Layer in Neural Networks.

Introduction: QP layers pros (practical speed)

Why using such more complex architecture ?

- For some class of layers, the representative power is similar to those of standard Neural Networks,
- For some optimization based problem, it performs better,
- Achievable practical use.

Introduction: QP layers pros (practical speed)

Trick: use the Implicit Function Theorem.

$$\min_{x \in \mathbb{R}^n} \left\{ f(x;\, \theta) \triangleq \frac{1}{2} x^\top H(\theta)\, x + x^\top g(\theta) \right\} \qquad \text{(QP}(\theta))$$
$$\text{s.t. } C(\theta)\, x \leq u(\theta),$$

Introduction: QP layers pros (practical speed)

Trick: use the Implicit Function Theorem.

$$\min_{x \in \mathbb{R}^n} \left\{ f(x;\,\theta) \triangleq \frac{1}{2} x^\top H(\theta)\, x + x^\top g(\theta) \right\} \quad \text{s.t. } C(\theta)\, x \leq u(\theta), \tag{QP(θ)}$$

Noting $(x^\star(\theta), z^\star(\theta))$ a solution to (QP(θ)), applying the IFT outputs:

$$\begin{bmatrix} H & C^\top \\ D(z^\star)C & D(Cx^\star - u) \end{bmatrix} \begin{bmatrix} \frac{\partial x^\star}{\partial \theta} \\ \frac{\partial z^\star}{\partial \theta} \end{bmatrix} = - \begin{bmatrix} \frac{\partial H}{\partial \theta} x^\star + \frac{\partial g}{\partial \theta} + \frac{\partial C}{\partial \theta}^\top z^\star \\ D(z^\star)(\frac{\partial C}{\partial \theta} x^\star - \frac{\partial u}{\partial \theta}) \end{bmatrix}$$

Introduction: QP layers pros (practical speed)

Trick: use the Implicit Function Theorem.

$$\min_{x\in\mathbb{R}^n} \left\{ f(x;\,\theta) \triangleq \frac{1}{2} x^\top H(\theta)\, x + x^\top g(\theta) \right\} \quad \text{s.t. } C(\theta)\, x \leq u(\theta), \tag{QP(θ)}$$

Noting $(x^\star(\theta), z^\star(\theta))$ a solution to (QP(θ)), applying the IFT outputs:

$$\begin{bmatrix} H & C^\top \\ D(z^\star)C & D(Cx^\star - u) \end{bmatrix} \begin{bmatrix} \frac{\partial x^\star}{\partial \theta} \\ \frac{\partial z^\star}{\partial \theta} \end{bmatrix} = -\begin{bmatrix} \frac{\partial H}{\partial \theta} x^\star + \frac{\partial g}{\partial \theta} + \frac{\partial C}{\partial \theta}^\top z^\star \\ D(z^\star)(\frac{\partial C}{\partial \theta} x^\star - \frac{\partial u}{\partial \theta}) \end{bmatrix}$$

$$\frac{\partial \mathcal{L}}{\partial \theta} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial x} \\ \frac{\partial \mathcal{L}}{\partial z} \end{bmatrix}^\top \begin{bmatrix} \frac{\partial x}{\partial \theta} \\ \frac{\partial z}{\partial \theta} \end{bmatrix} = (-\begin{bmatrix} H & C^\top D(z^\star) \\ C & D(Cx^\star - u) \end{bmatrix} \begin{bmatrix} d_x \\ d_z \end{bmatrix})^\top \begin{bmatrix} \frac{\partial x}{\partial \theta} \\ \frac{\partial z}{\partial \theta} \end{bmatrix}$$

Introduction: cons (limited trainable architectures)

Two issues

- IFT assumptions (non singularity of KKT matrix etc.),
- Structural feasibility at training and test time.

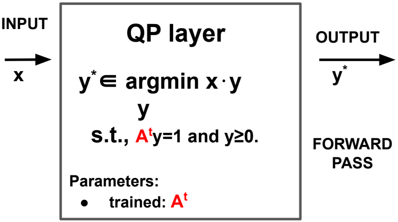


Figure: A LP layer. Nothing guarantees during training that the vector of 1 lies in the range space of the A^t.

Table of Contents

Solution outline: ideal pipeline

General idea: consider a "broader" problem.

Solution outline: ideal pipeline

General idea: consider a "broader" problem.

- Forward pass: solve the closest feasible QP.

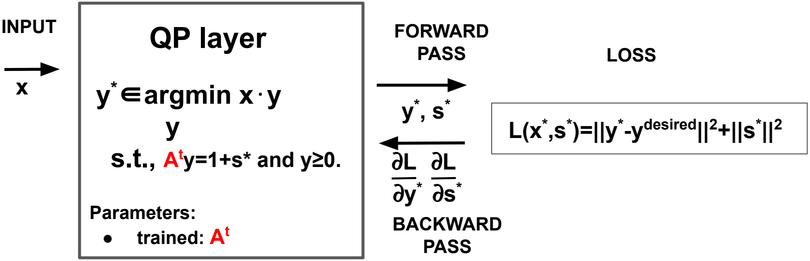


Figure: General solution outline.

Solution outline: ideal pipeline

General idea: consider a "broader" problem.

- Forward pass: solve the closest feasible QP.
- Backward pass: differentiate through one solution.

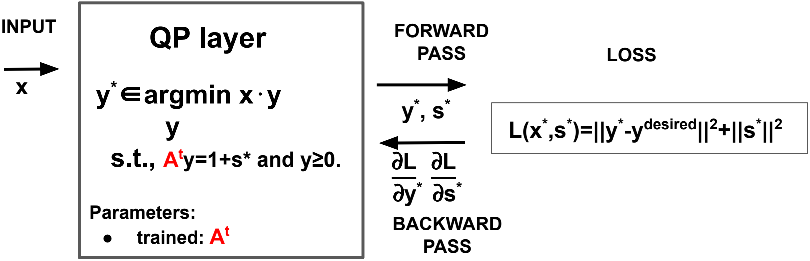


Figure: General solution outline.

Solution outline: plan

General plan:

- Forward pass: solve the closest feasible QP.

Solution outline: plan

General plan:

- Forward pass: solve the closest feasible QP.
  - Define the closest feasible problem.
  - Propose an efficient algorithm to solve it.

Solution outline: plan

General plan:

- Forward pass: solve the closest feasible QP.
  - Define the closest feasible problem.
  - Propose an efficient algorithm to solve it.
- Backward pass: differentiate through one solution.

Solution outline: plan

General plan:

- Forward pass: solve the closest feasible QP.
  - Define the closest feasible problem.
  - Propose an efficient algorithm to solve it.
- Backward pass: differentiate through one solution.
  - Prove applying the IFT makes sense.
  - Provide tractable algorithms.
  - In non differentiable case, provide alternatives.

Table of Contents

The closest feasible QP problem: definition

$$x^\star(\theta) \in \operatorname*{argmin}_{x\in\mathbb{R}^n} \left\{ f(x;\,\theta) \triangleq \frac{1}{2}x^\top H(\theta)\,x + x^\top g(\theta) \right\} \quad (\mathrm{QP}(\theta))$$
$$\text{s.t. } C(\theta)\,x \leq u(\theta),$$

where $H(\theta) \in \mathcal{S}_+^n(\mathbb{R})$, $g(\theta) \in \mathbb{R}^n$, $C(\theta) \in \mathbb{R}^{n_i \times n}$ and $u(\theta) \in \mathbb{R}^{n_i}$.

The closest feasible QP problem: definition

$$
\begin{aligned}
x^\star(\theta) \in \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} & \left\{ f(x;\,\theta) \triangleq \frac{1}{2} x^\top H(\theta)\, x + x^\top g(\theta) \right\} \\
\text{s.t.}\ \ & C(\theta)\, x \leq u(\theta),
\end{aligned}
\qquad (\mathrm{QP}(\theta))
$$

where $H(\theta) \in \mathcal{S}_+^n(\mathbb{R})$, $g(\theta) \in \mathbb{R}^n$, $C(\theta) \in \mathbb{R}^{n_i \times n}$ and $u(\theta) \in \mathbb{R}^{n_i}$.

Assumption

$H(\theta)$ is symmetric positive definite in the direction of $g(\theta)$ or $g(\theta)$ is orthogonal to the recession cone of $QP(\theta)$, i.e.,
$g(\theta) \perp \mathcal{C}^\infty(\theta) \triangleq \{y \in \mathbb{R}^n | C(\theta)[x + \tau y] \leq u(\theta) \text{ s.t. } C(\theta)x \leq u(\theta), \tau \geq 0\}.$

The closest feasible QP problem: definition

Under Assumption (1) the closest feasible problem (QP-H(θ)) is well-posed:

$$\begin{aligned} s^\star(\theta) = \arg \min_{s \in \mathbb{R}^{n_i}} \ & \frac{1}{2}\|s\|_2^2 \\ \text{s.t. } & x^\star(\theta), z^\star(\theta) \in \arg \min_{x \in \mathbb{R}^n} \max_{z \in \mathbb{R}_+^{n_i}} L(x, z, s; \theta), \end{aligned} \tag{QP-H(θ)}$$

with $L(x, z, s; \theta) \triangleq \frac{1}{2} x^\top H(\theta)\, x + x^\top g(\theta) + z^\top (C(\theta)x - u(\theta) - s)$

Under Assumption (1) the closest feasible problem (QP-H(θ)) is well-posed:

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

with $L(x,z,s;\theta) \triangleq \frac{1}{2}x^\top H(\theta)\,x + x^\top g(\theta) + z^\top (C(\theta)x - u(\theta) - s)$

Remark

$s^\star \triangleq \arg\min_{s\in\mathcal{R}(C)+]-\infty,u]} \|s\|_2^2$

The closest feasible QP problem: solution method

The Augmented Lagrangian

$$\mathcal{L}_A(x, z; \mu) \triangleq f(x) + \frac{1}{2\mu}(\|[Cx - u + \mu z]_+\|_2^2 - \|\mu z\|_2^2)$$

The AL method

$$x^{k+1} = \arg\min_x \mathcal{L}_A(x, z; \mu)$$
$$z^{k+1} = [\frac{1}{\mu}(Cx^{k+1} - u) + z^k]_+$$

Theorem (A. Chiche, JC Gilbert (2016)[1])

Under Assumption (1), the revised AL algorithm does not terminate with a direction of unboundedness and generates a sequence $\{(x^k, z^k)\}$ converging towards a solution to (QP-H(θ)).

Remark

(i) revised AL algorithm = modified stopping criterion; (ii) exact AL method!

[1] Chiche, A., Gilbert, J. C. (2016). How the augmented Lagrangian algorithm can deal with an infeasible convex quadratic optimization problem.

The closest feasible QP problem: solution method in practice

The ProxSuite library

ProxSuite
ADVANCED OPTIMIZERS FOR ROBOTICS AND BEYOND

- ✓ **fast:** C++ implementation, with homemade linear Cholesky solver
- ✓ **scalable:** various backends for dense, sparse and matrix-free optimization
- ✓ **easy-to-use:** API closed to OSQP, Python and Julia bindings
- ✓ **open-source:** BSD-license, easily installable

github.com/Simple-Robotics/proxsuite

```
conda install -c conda-forge proxsuite
```

Table of Contents

Differentiating (QP-H(θ)) solutions: methodology

- Analyze the properties of KKT map for (QP-H(θ)),
- Extended Conservative Jacobian definition.
- Practical algorithms.

Differentiating (QP-H(θ)) solutions: KKT conditions of (QP-H(θ))

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

Lemma

It holds that $(x^\star, z^\star, s^\star)$ *solves QP-H(*θ*) iff there exists* $t^\star \in \mathbb{R}^{n_i}$ *s.t.* $G(x^\star, z^\star, t^\star;\ \theta) = 0$ *and* $s^\star = [t^\star]_+$.

Differentiating (QP-H(θ)) solutions: path differentiability of G

Lemma

G is path differentiable w.r.t. $x^\star$, $z^\star$ and $t^\star$. Furthermore, if $H(\theta)$, $g(\theta)$, $C(\theta)$ and $u(\theta)$ are differentiable w.r.t. θ, then G is path differentiable w.r.t. θ.

It means G has a **conservative**[1] Jacobian. It is equivalent to having a chain rule for the Clarke subdifferential

$$Jac^c G(w) \triangleq conv\{\lim_{k\to\infty} Jac\ G(w_k) \in diff_F, w_k \to w\},$$

with $diff_G$ the full measure set where G is differentiable, and $Jac\ G$ the standard Jacobian of G.

[1] Bolte, J., Le, T., Pauwels, E., Silveti-Falls, T. (2021). Nonsmooth implicit differentiation for machine-learning and optimization.

Differentiating (QP-H(θ)) solutions: Extended Conservative Jacobian

Let $v^\star = (x^\star, z^\star, t^\star) \in \mathbb{R}^n \times \mathbb{R}_+^{n_i} \times \mathbb{R}^{n_i}$ s.t. $G(x^\star, z^\star, t^\star;\ \theta) = 0$.

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

$\Pi \frac{\partial t^\star}{\partial \theta} \in \frac{\partial s^\star}{\partial \theta}$, with $\Pi \in \partial([.]_+)(t^\star)$.

Differentiating (QP-H(θ)) solutions: Extended Conservative Jacobian

Why do we use a least-square ?

- It makes sense in "standard" cases:
  - If (QP-H(θ)) reduces to (QP(θ)) (i.e., feasible problem), and if the IFT applies, we recover standard Jacobian.

Differentiating (QP-H(θ)) solutions: Extended Conservative Jacobian

Why do we use a least-square ?

- It makes sense in "standard" cases:
  - If (QP-H(θ)) reduces to (QP(θ)) (i.e., feasible problem), and if the IFT applies, we recover standard Jacobian.
  - If (QP-H(θ)) is infeasible and satisfies non-singularity conditions for a "conservative" IFT, we recover conservative Jacobians

Lemma

Let $C(\theta), u(\theta)$ be differentiable w.r.t. θ, and $H = 0$, and g be fixed w.r.t. θ and satisfying Assumption 1. If $s^\star > 0$ and $z^\star = 0$ (i.e., it does not satisfy strict complementarity) and C is full row rank, then the ECJs of $x^\star$, $z^\star$, $t^\star$, and $s^\star$ correspond to conservative Jacobians.

Differentiating (QP-H(θ)) solutions: Extended Conservative Jacobian

Why do we use a least-square ?

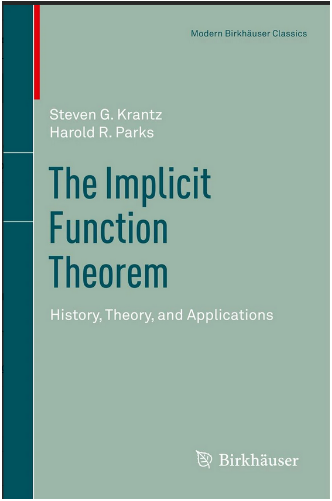
- It makes sense in "standard" cases:
  - If (QP-H(θ)) reduces to (QP(θ)) (i.e., feasible problem), and if the IFT applies, we recover standard Jacobian.
  - If (QP-H(θ)) is infeasible and satisfies non-singularity conditions for a "conservative" IFT, we recover conservative Jacobians
- In non standard cases (i.e., IFT does not apply):
  - Solutions may still have a notion of derivatives

Differentiating (QP-H(θ)) solutions: Extended Conservative Jacobian

Why do we use a least-square ?

- It makes sense in "standard" cases:
  - If (QP-H(θ)) reduces to (QP(θ)) (i.e., feasible problem), and if the IFT applies, we recover standard Jacobian.
  - If (QP-H(θ)) is infeasible and satisfies non-singularity conditions for a "conservative" IFT, we recover conservative Jacobians
- In non standard cases (i.e., IFT does not apply):
  - Solutions may still have a notion of derivatives
  - If not, it is a form of smoothing, hopefully, the "ECJ" is informative.

Differentiating (QP-H(θ)) solutions: other examples



Differentiating (QP-H(θ)) solutions: a toy example

Consider the feasible LP parameterized by $\theta \in (0, 1)$

$$\begin{aligned} x^\star(\theta) \in \underset{x_1, x_2 \in \mathbb{R}^2}{\arg\min}\, & x_1 + x_2 \\ \text{s.t.}\;\; & \theta \leq x_1 + x_2, \\ & 0 \leq x_1 \leq 1, \\ & 0 \leq x_2 \leq 1. \end{aligned}$$

Differentiating (QP-H(θ)) solutions: a toy example

Some differential calculus:
$\frac{\partial H}{\partial \theta} = 0$, $\frac{\partial g}{\partial \theta} = 0$, $\frac{\partial C}{\partial \theta} = 0$, $\frac{\partial u}{\partial \theta} = (-1\ 0\ 0\ 0\ 0)^\top$.

Differentiating (QP-H(θ)) solutions: a toy example

Some differential calculus:
$\frac{\partial H}{\partial \theta} = 0$, $\frac{\partial g}{\partial \theta} = 0$, $\frac{\partial C}{\partial \theta} = 0$, $\frac{\partial u}{\partial \theta} = (-1\ 0\ 0\ 0\ 0)^\top$.

$$\begin{bmatrix} H & C^\top & 0 \\ C & 0 & -I \\ 0 & \Pi_1 - I & \Pi_1 \Pi_2 \\ 0 & 0 & C^\top \Pi_3 \end{bmatrix} \in \frac{\partial G(x^\star, z^\star, t^\star;\ \theta)}{\partial v^\star},$$

for some $\Pi_1 \in \partial[.]_+([t^\star]_- + z^\star)$, $\Pi_2 \in \partial[.]_-(t^\star)$ and $\Pi_3 \in \partial[.]_+(t^\star)$.

Differentiating (QP-H(θ)) solutions: a toy example

Some differential calculus:
$\frac{\partial H}{\partial \theta} = 0$, $\frac{\partial g}{\partial \theta} = 0$, $\frac{\partial C}{\partial \theta} = 0$, $\frac{\partial u}{\partial \theta} = (-1\ 0\ 0\ 0\ 0)^\top$.

$$\begin{bmatrix} H & C^\top & 0 \\ C & 0 & -I \\ 0 & \Pi_1 - I & \Pi_1\Pi_2 \\ 0 & 0 & C^\top \Pi_3 \end{bmatrix} \in \frac{\partial G(x^\star, z^\star, t^\star;\ \theta)}{\partial v^\star},$$

for some $\Pi_1 \in \partial[.]_+([t^\star]_- + z^\star)$, $\Pi_2 \in \partial[.]_-(t^\star)$ and $\Pi_3 \in \partial[.]_+(t^\star)$.
Feasible problem: $\Pi_2 = I$, $\Pi_3 = 0$

Differentiating (QP-H(θ)) solutions: a toy example

Some differential calculus:
$\frac{\partial H}{\partial \theta} = 0$, $\frac{\partial g}{\partial \theta} = 0$, $\frac{\partial C}{\partial \theta} = 0$, $\frac{\partial u}{\partial \theta} = (-1\ 0\ 0\ 0\ 0)^\top$.

$$\begin{bmatrix} H & C^\top & 0 \\ C & 0 & -I \\ 0 & \Pi_1 - I & \Pi_1 \Pi_2 \\ 0 & 0 & C^\top \Pi_3 \end{bmatrix} \in \frac{\partial G(x^\star, z^\star, t^\star;\ \theta)}{\partial v^\star},$$

for some $\Pi_1 \in \partial[.]_+([t^\star]_- + z^\star)$, $\Pi_2 \in \partial[.]_-(t^\star)$ and $\Pi_3 \in \partial[.]_+(t^\star)$.
Feasible problem: $\Pi_2 = I$, $\Pi_3 = 0$
Strict complementarity+unique active constraint:
$\Pi_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

Differentiating (QP-H(θ)) solutions: a toy example

The problem simplifies to

$$\frac{\partial x^\star}{\partial \theta}, \frac{\partial z^\star}{\theta} \in \arg\min_{b_x, b_z} \left\| \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} (b_x)_1 \\ (b_x)_2 \\ b_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\|_2^2$$

with $\frac{\partial t^\star}{\partial \theta} = 0$.

The IFT does not apply (degenerate constraints)!

Yet, the least square provides solutions given by the equations:

$$(b_x)_1 + (b_x)_2 = \frac{1}{2},$$
$$b_z \in \mathbb{R}.$$

Differentiating (QP-H(θ)) solutions: Backward AD algorithms (generic case)

$$\frac{\partial \mathcal{L}}{\partial \theta} = (b_1^\star)^\top \frac{\partial H}{\partial \theta} x^\star + (b_1^\star)^\top \frac{\partial g}{\partial \theta} + (b_2^\star)^\top \frac{\partial C}{\partial \theta} x^\star + (z^\star)^\top \frac{\partial C}{\partial \theta} b_1^\star + (s^\star)^\top \frac{\partial C}{\partial \theta} b_4^\star - (b_2^\star)^\top \frac{\partial u}{\partial \theta},$$

with $b_1^\star$, $b_2^\star$, $b_3^\star$ and $b_4^\star$ with solutions of

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

Differentiating (QP-H(θ)) solutions: Backward AD algorithms (feasible case)

$$\frac{\partial \mathcal{L}}{\partial \theta} = (b_x^\star)^\top \frac{\partial H}{\partial \theta} x^\star + (b_x^\star)^\top \frac{\partial g}{\partial \theta} + (\Pi_1 b_z^\star)^\top \frac{\partial C}{\partial \theta} x^\star + (z^\star)^\top \frac{\partial C}{\partial \theta} b_x^\star - (\Pi_1 b_z^\star)^\top \frac{\partial u}{\partial \theta},$$

with $b_x^\star$, $b_z^\star$, the solution of the following linear system

$$\begin{bmatrix} H & C^\top \Pi_1 \\ C & -(I - \Pi_1) \end{bmatrix} \begin{bmatrix} b_x \\ b_z \end{bmatrix} = - \begin{bmatrix} \frac{\delta \mathcal{L}}{\delta x^\star} \\ \frac{\delta \mathcal{L}}{\delta z^\star} \end{bmatrix}.$$

Differentiating (QP-H(θ)) solutions: toy example (continued!)

LP parameterized by $\theta \in (0,1)$:

$$\begin{aligned} x^\star(\theta) \in & \arg\min_{x_1, x_2 \in \mathbb{R}^2} x_1 + x_2 \\ \text{s.t. } & \theta \leq x_1 + x_2, \\ & 0 \leq x_1 \leq 1, \\ & 0 \leq x_2 \leq 1. \end{aligned}$$

We choose: $x_1^\star = x_2^\star = \frac{\theta}{2}$.

The problem simplifies to

$$\frac{\partial x^\star}{\partial \theta}, \frac{\partial z^\star}{\theta} \in \arg\min_{b_x, b_z} \left\| \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} (b_x)_1 \\ (b_x)_2 \\ b_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\|_2^2$$

with $\frac{\partial t^\star}{\partial \theta} = 0$.

The IFT does not apply (degenerate constraints)!

Yet, the least square provides solutions given by the equations:

$$\begin{aligned} & (b_x)_1 + (b_x)_2 = \frac{1}{2}, \\ & b_z \in \mathbb{R}. \end{aligned}$$

Differentiating (QP-H(θ)) solutions: toy example (continued!)

Let's minimize $\mathcal{L}(\theta) = x_1^\star(\theta)$.

Differentiating (QP-H(θ)) solutions: toy example (continued!)

Let's minimize $\mathcal{L}(\theta) = x_1^\star(\theta)$. If we try applying the IFT we get

$$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} (b_x)_1 \\ (b_x)_2 \\ b_z \end{bmatrix} = - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

which is infeasible!

Differentiating (QP-H(θ)) solutions: toy example (continued!)

Let's minimize $\mathcal{L}(\theta) = x_1^\star(\theta)$. If we try applying the IFT we get

$$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} (b_x)_1 \\ (b_x)_2 \\ b_z \end{bmatrix} = - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

which is infeasible! The least-square solution, yet provides

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} (b_x^\star)_1 \\ (b_x^\star)_2 \\ b_z^\star \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

which outputs as ECJ for $\frac{\partial \mathcal{L}}{\partial \theta} = b_z^\star = \frac{1}{2}$. It is coherent with $\nabla_\theta(\theta/2) = 1/2$, if we choose $x_1^\star(\theta) = \theta/2$.

Differentiating (QP-H(θ)) solutions: toy example (continued!)

Let's minimize $\mathcal{L}(\theta) = x_1^\star(\theta)$. If we try applying the IFT we get

$$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} (b_x)_1 \\ (b_x)_2 \\ b_z \end{bmatrix} = - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

which is infeasible! The least-square solution, yet provides

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} (b_x^\star)_1 \\ (b_x^\star)_2 \\ b_z^\star \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

which outputs as ECJ for $\frac{\partial \mathcal{L}}{\partial \theta} = b_z^\star = \frac{1}{2}$. It is coherent with $\nabla_\theta(\theta/2) = 1/2$, if we choose $x_1^\star(\theta) = \theta/2$.

The problem simplifies to

$$\frac{\partial x^\star}{\partial \theta}, \frac{\partial z^\star}{\theta} \in \underset{b_x, b_z}{\arg\min} \left\| \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} (b_x)_1 \\ (b_x)_2 \\ b_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\|_2^2$$

with $\frac{\partial t^\star}{\partial \theta} = 0$.
The IFT does not apply (degenerate constraints)!
Yet, the least square provides solutions given by the equations:

$$(b_x)_1 + (b_x)_2 = \frac{1}{2},$$
$$b_z \in \mathbb{R}.$$

Differentiating (QP-H(θ)) solutions: toy example (continued!)

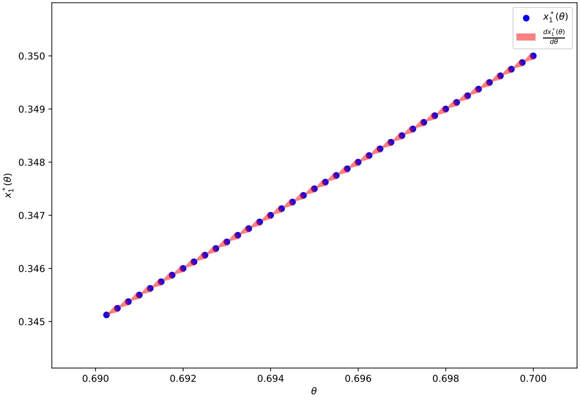


Figure: 40 iterations of gradient descent for minimizing $x_1^\star(\theta)$ using ECJs.

Table of Contents

Back to the Sudoku problem: setup

Three architectures compared:

Back to the Sudoku problem: setup

Three architectures compared:

- structurally feasible with strictly convex loss (standard model of OptNet);

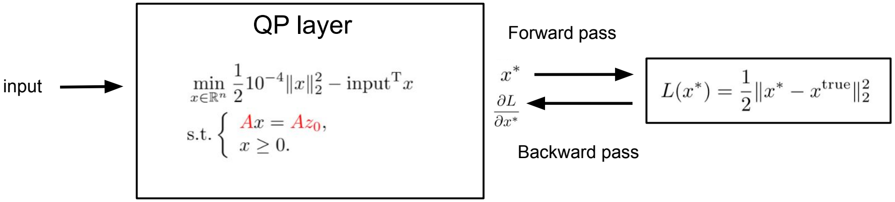


Figure: OptNet layer (structurally feasible at training time).

Three architectures compared:

- structurally feasible with strictly convex loss (standard of OptNet);
- training infeasible LP towards feasibility (with QPLayer, ours).

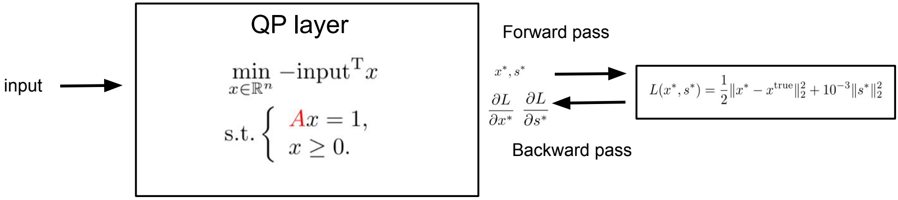


Figure: QPLayer training an infeasible LP.

Back to the Sudoku problem: setup

Three architectures compared:

- structurally feasible with strictly convex loss (standard model of OptNet);
- training infeasible LP towards feasibility (with QPLayer, ours).
- Reformulation of (QP-H(θ)) as a convex QP (non standard)

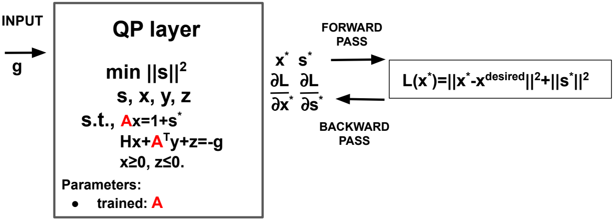


Figure: OptNet layer used for trying to learn A via a reformulation of (QP-H(θ)).

Back to the sudoku: results

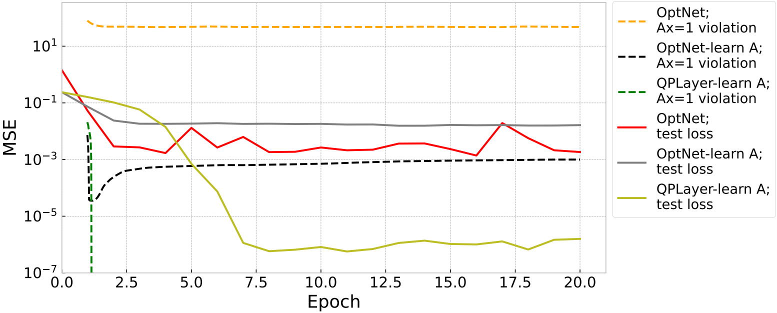


Figure: Test MSE loss of QPLayer, OptNet, QPLayer-learn A, and OptNet-learn A specialized for learning A. It includes Sudoku $Ax = 1$ violation.

Back to the sudoku: results

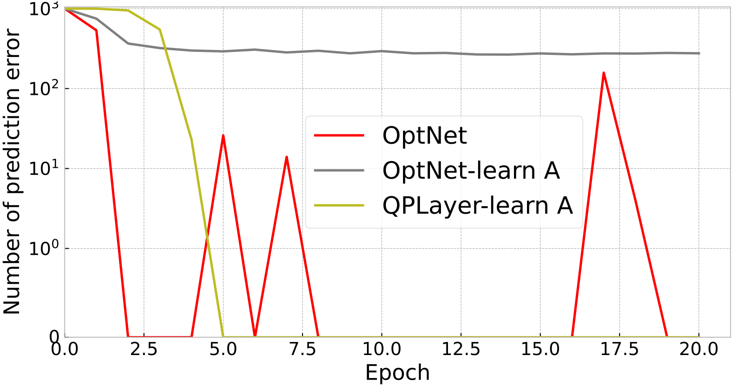


Figure: Test prediction errors over 1000 puzzles of OptNet, QPLayer, QPLayer-learn A and OptNet-learn A specialized for learning A.

Table of Contents

Conclusion

We introduced:

- QPLayer: framework for learning new types of QP layers.
- Practical concept of "Extended" Conservative Jacobian.
- Practical algorithms making use of powerful AL properties.

Appendix: Augmented Lagrangian method

The Augmented Lagrangian

$$\mathcal{L}_A(x, z; \mu) \triangleq f(x)$$
$$+ \frac{1}{2\mu}(\|[Cx - u + \mu z]_+\|_2^2 - \|\mu z\|_2^2)$$

The AL method

$$x^{k+1} \approx_{\epsilon^k} \arg\min_x \mathcal{L}_A(x, z; \mu)$$
$$z^{k+1} = [\frac{1}{\mu}(Cx^{k+1} - u) + z^k]_+$$

Magnus Hestenes

Michael J.D. Powell

Appendix: QP layers in machine learning (example 2)

Coupling visual sensing and calibration of control algorithms in robotics.

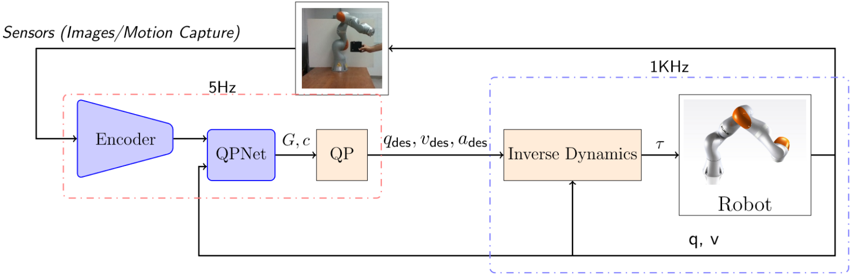


Figure: Pipeline of image measurements for learning QP cost functions (for optimal reaching tasks).

The closest feasible QP problem: solution method

Augmented Lagrangian-based methods have the property of converging towards a solution to (QP-H(θ)) if (QP(θ)) is primal infeasible.

Theorem (A. Chiche, JC Gilbert (2016))

Under Assumption (1)*, the revised AL algorithm does not terminate with a direction of unboundedness and generates a sequence* $\{(x^k, z^k)\}$ *converging towards a solution to* (QP-H(θ))*.*

Remark

revised AL algorithm means here two things: (i) the AL algorithm takes into account a modified stopping criterion (closest feasible optimality); (ii) the AL method is exact!

Differentiating (QP-H(θ)) solutions: Backward AD algorithms

Lemma

Let $h : \mathbb{R}^n \times (\mathbb{R}^{n_i})^2 \to \mathbb{R}$ be a differentiable function, and let $H(\theta)$, $g(\theta)$, $C(\theta)$ and $u(\theta)$ be differentiable w.r.t. θ and satisfying 1. Then, denoting $\mathcal{L}(\theta) \triangleq h(x^\star(\theta), z^\star(\theta), s^\star(\theta))$ and under IFT assumptions, we have that $\frac{\partial \mathcal{L}}{\partial \theta}$ can be derived as follows

$$\frac{\partial \mathcal{L}}{\partial \theta} = (b_1^\star)^\top \frac{\partial H}{\partial \theta} x^\star + (b_1^\star)^\top \frac{\partial g}{\partial \theta} + (b_2^\star)^\top \frac{\partial C}{\partial \theta} x^\star + (z^\star)^\top \frac{\partial C}{\partial \theta} b_1^\star + (s^\star)^\top \frac{\partial C}{\partial \theta} b_4^\star - (b_2^\star)^\top \frac{\partial u}{\partial \theta},$$

where $b_1^\star$, $b_2^\star$, $b_3^\star$ and $b_4^\star$ are the solutions of the linear system

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

Differentiating (QP-H(θ)) solutions: Backward AD algorithms (feasible case)

Lemma

Let $h : \mathbb{R}^n \times (\mathbb{R}^{n_i}) \to \mathbb{R}$ be a differentiable function, and let $H(\theta)$, $g(\theta)$, $C(\theta)$ and $u(\theta)$ be differentiable w.r.t. θ and satisfying Assumption 1. Then, denoting $\mathcal{L}(\theta) \triangleq h(x^\star(\theta), z^\star(\theta))$, we have under IFT assumptions that $\frac{\partial \mathcal{L}}{\partial \theta}$ can be derived as follows

$$\frac{\partial \mathcal{L}}{\partial \theta} = (b_x^\star)^\top \frac{\partial H}{\partial \theta} x^\star + (b_x^\star)^\top \frac{\partial g}{\partial \theta} + (\Pi_1 b_z^\star)^\top \frac{\partial C}{\partial \theta} x^\star + (z^\star)^\top \frac{\partial C}{\partial \theta} b_x^\star - (\Pi_1 b_z^\star)^\top \frac{\partial u}{\partial \theta},$$

with $b_x^\star$, $b_z^\star$, the solution of the following linear system

$$\begin{bmatrix} H & C^\top \Pi_1 \\ C & -(I - \Pi_1) \end{bmatrix} \begin{bmatrix} b_x \\ b_z \end{bmatrix} = - \begin{bmatrix} \frac{\delta \mathcal{L}}{\delta x^\star} \\ \frac{\delta \mathcal{L}}{\delta z^\star} \end{bmatrix}.$$