Demystifying and Generalizing BinaryConnect

Yao-Liang Yu
(Joint work with Tim Dockhorn, Eyyub Sari, Mahdi Zolnouri, Vahid Partovi Nia)

24th Midwest Optimization Meeting
October 29, 2022

UNIVERSITY OF WATERLOO

The “Big” Cost

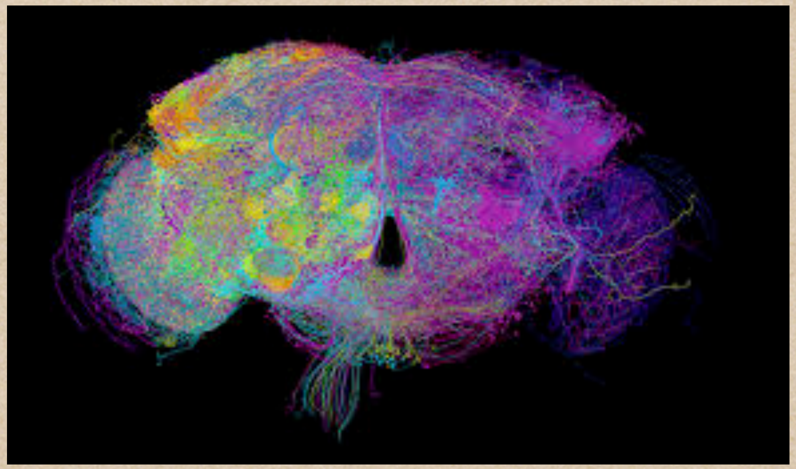
The "Big" Cost

big data

The "Big" Cost

big data

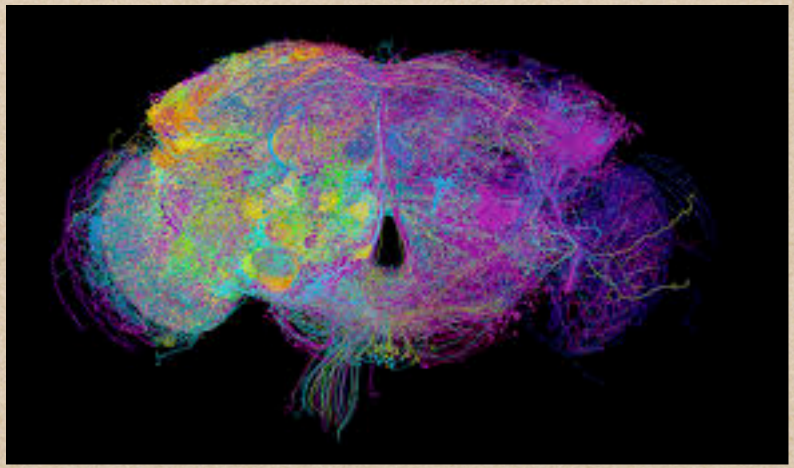
big model



The "Big" Cost

big data

big model

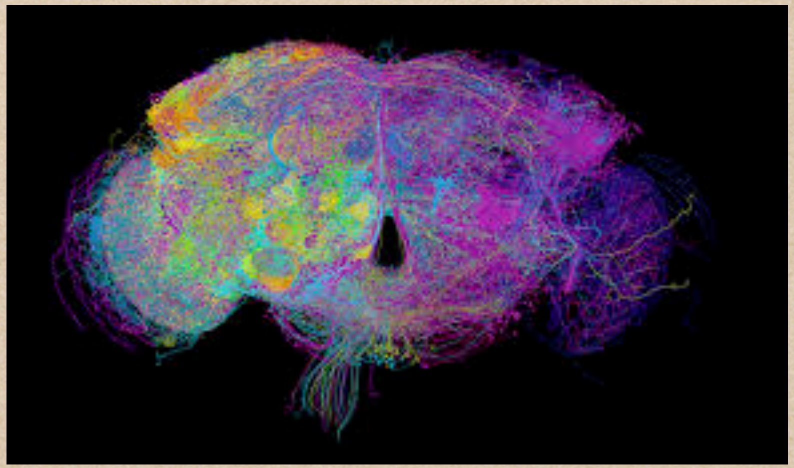


big computing

The "Big" Cost

big data

big model



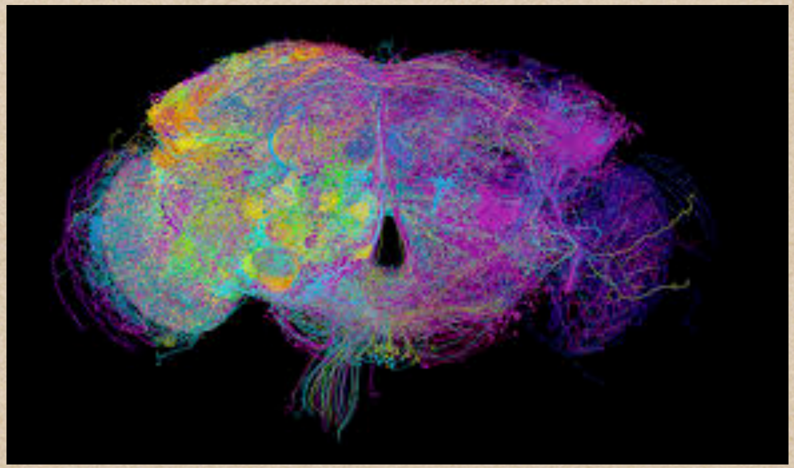
big computing

big consumption

The "Big" Cost

big data

big model



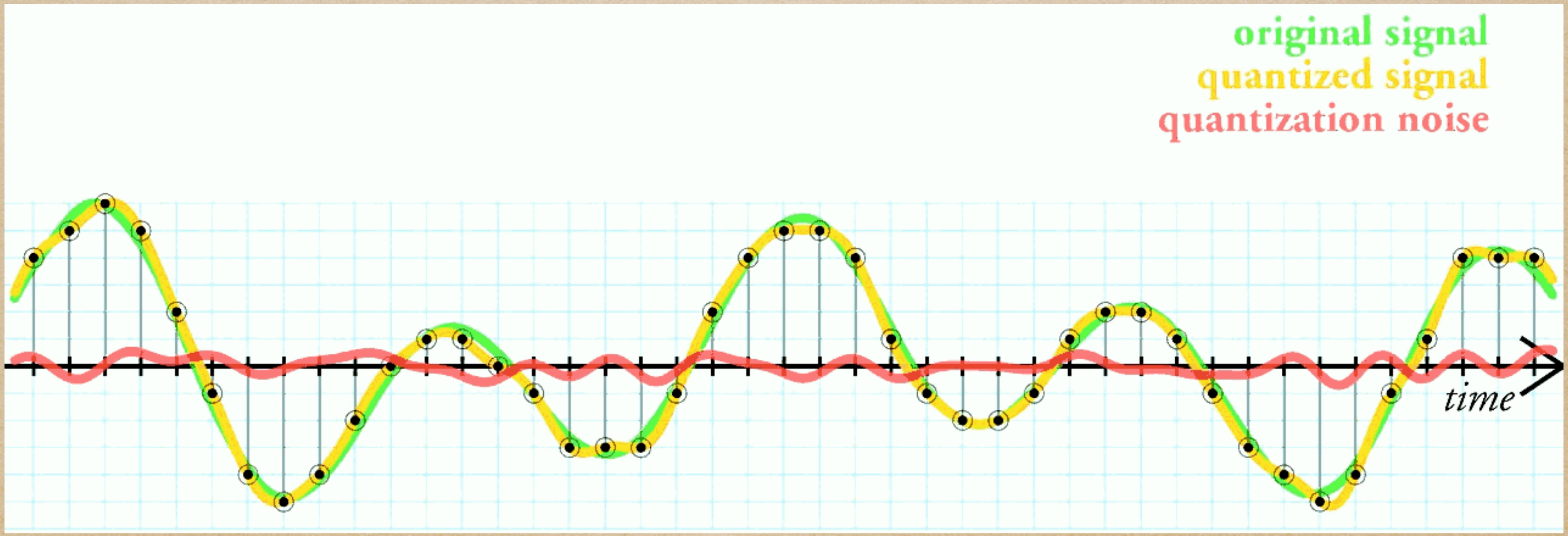
small devices

big computing

big consumption

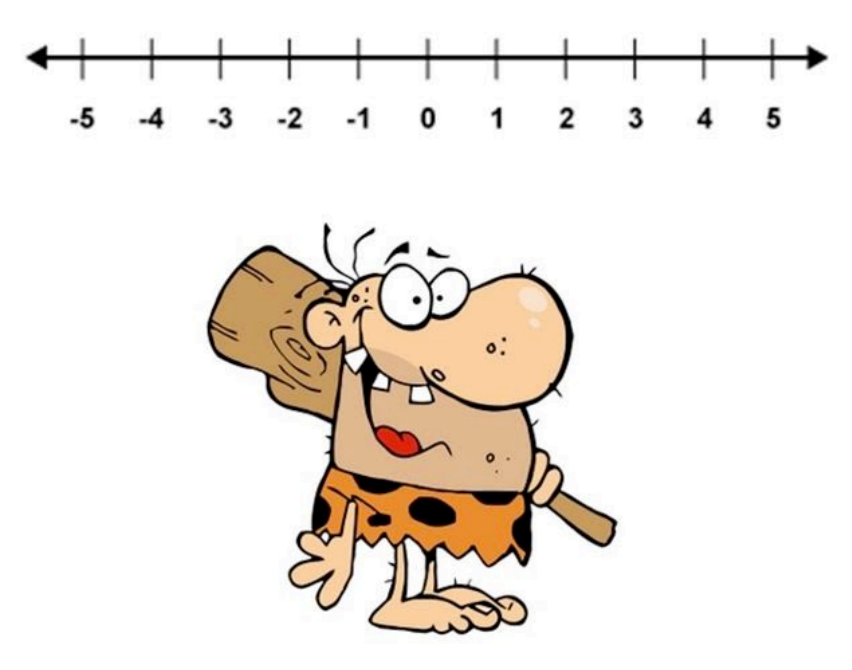
Quantization

- Full precision weights —> low precision
- Immediately reduced memory and energy
- Can also quantize activations/gradients/etc.



An Inherent Problem

- Real numbers do not really exist in physical world, or at least in current digital computers
- Create lots of headaches, a.k.a. numerical analysis
- We all deal with it, often by ignoring it and hoping things do not break…



Background

QNN Problem

$$\min_{\mathbf{w} \in Q} \ \ell(\mathbf{w})$$

- ℓ : loss we aim to minimize, e.g. cross-entropy as a surrogate for misclassification error
- Q: quantization set, usually consists of finitely many values, e.g. $Q = \{\pm 1\}^d$ for binary nets
- Solving QNN is very challenging, but we "just" aim to compete against the continuous network:

$$\min_{\mathbf{w}^* \in \mathbb{R}^d} \ \ell(\mathbf{w}^*)$$

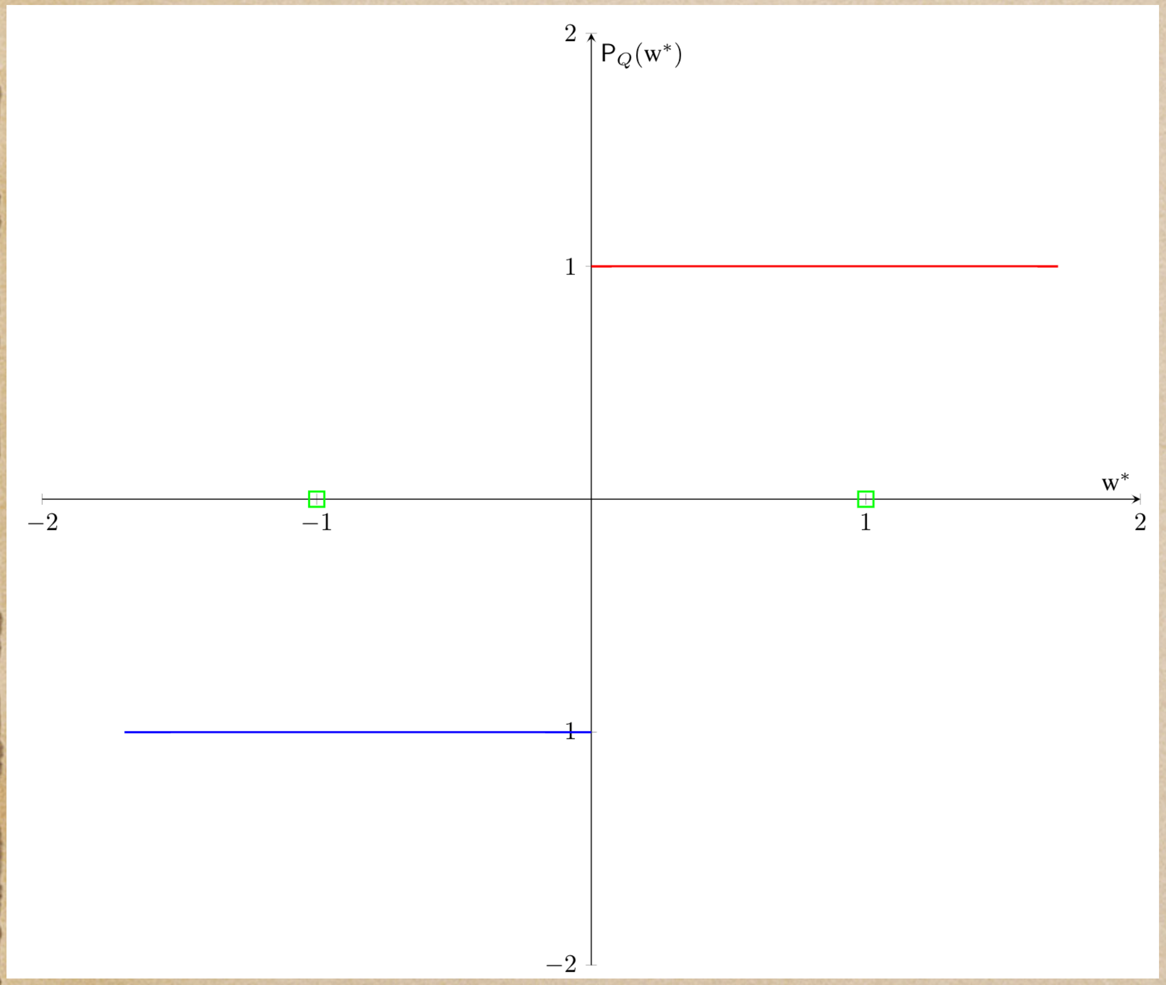
Binary Connect

$$\mathbf{w}^*_{t+1} = \mathbf{w}^*_t - \eta_t \widetilde{\nabla} \ell\left(\mathbf{P}(\mathbf{w}^*_t)\right)$$

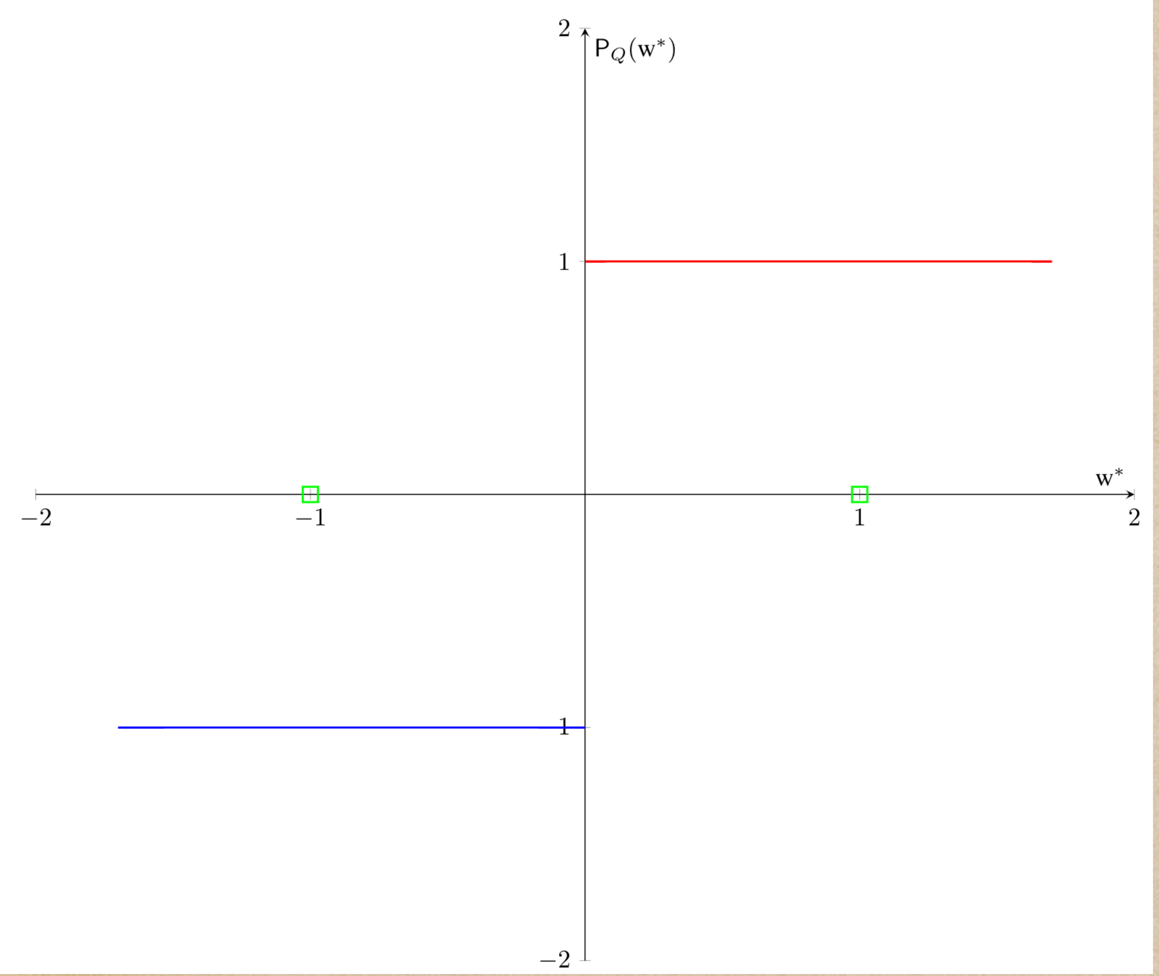
- The quantizer P, e.g. the sign function, thresholds continuous weights $\mathbf{w}^*$ into discrete (binary) ones.
- Many other choices of P have been invented since.
- Setting P = id, we recover the usual SGD for training NNs.

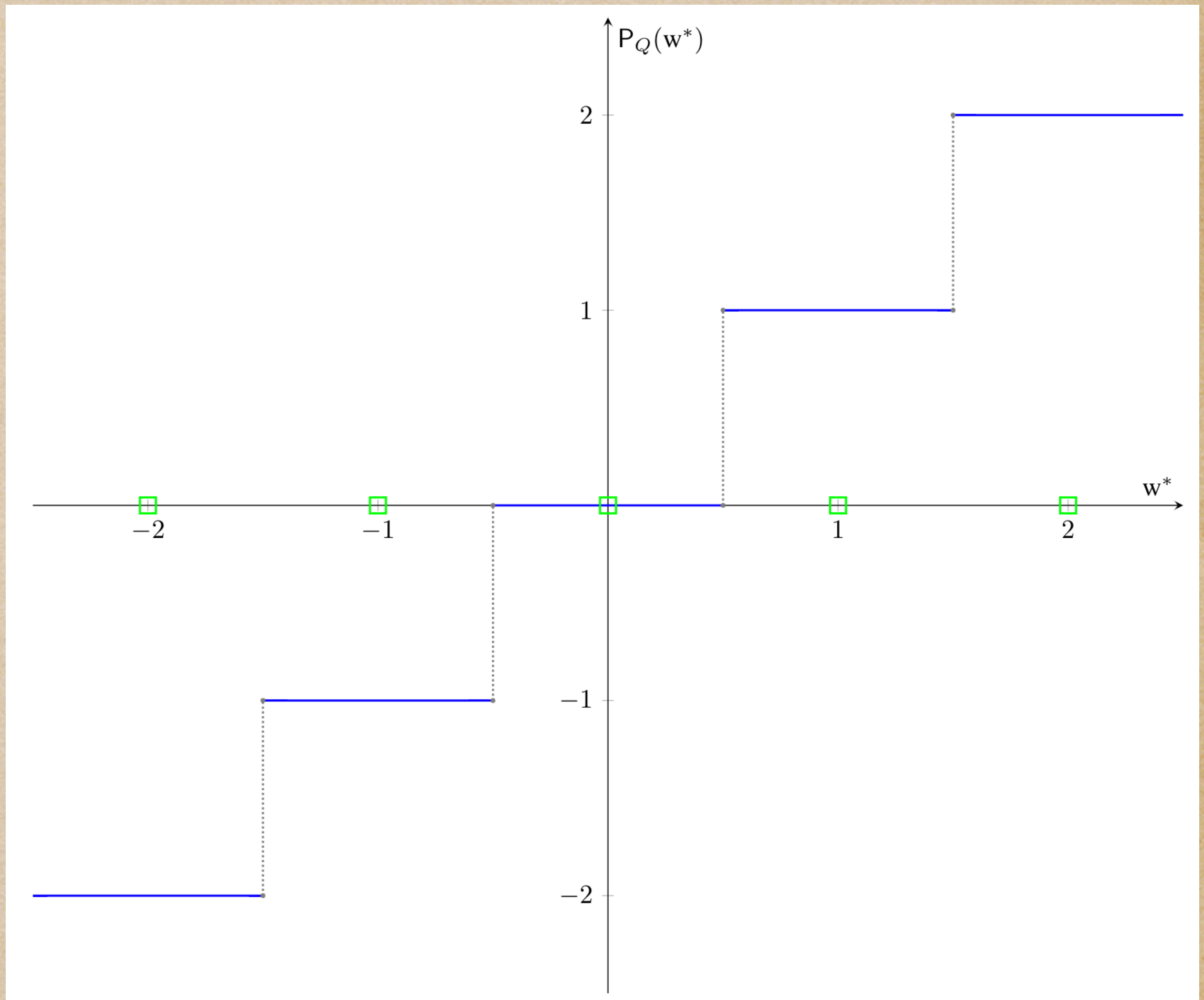
- M. Courbariaux, Y. Bengio, and J.-P. David. *BinaryConnect: Training Deep Neural Networks with Binary Weights during Propagations*. NeurIPS (2015).
- I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. *Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations*. JMLR (2017).

Projector



Projector





“Straight-through”

$$\min_{\mathbf{w}\in Q} \ell(\mathbf{w}) \xrightarrow{\text{reparameterize}} \min_{\mathbf{w}^*\in\mathbb{R}^d} \ell\left(\mathbf{P}(\mathbf{w}^*)\right)$$

- Y. Bengio, N. Léonard, and A. Courville. *Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation*. arXiv (2013).

"Straight-through"

$$\min_{\mathbf{w}\in Q} \ell(\mathbf{w}) \xrightarrow{\text{reparameterize}} \min_{\mathbf{w}^*\in\mathbb{R}^d} \ell\big(\mathbf{P}(\mathbf{w}^*)\big)$$

- Forward pass: apply quantizer P

- Y. Bengio, N. Léonard, and A. Courville. *Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation*. arXiv (2013).

“Straight-through”

$$\min_{\mathbf{w}\in Q} \ell(\mathbf{w}) \xrightarrow{\text{reparameterize}} \min_{\mathbf{w}^*\in \mathbb{R}^d} \ell\big(\mathbf{P}(\mathbf{w}^*)\big)$$

- Forward pass: apply quantizer P
- Backward pass: ignore quantizer P

- Y. Bengio, N. Léonard, and A. Courville. *Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation*. arXiv (2013).

"Straight-through"

$$\min_{\mathbf{w}\in Q} \ell(\mathbf{w}) \xrightarrow{\text{reparameterize}} \min_{\mathbf{w}^*\in\mathbb{R}^d} \ell\big(\mathbf{P}(\mathbf{w}^*)\big)$$

- Forward pass: apply quantizer P
- Backward pass: ignore quantizer P

$$\mathbf{w}^*_{t+1} = \mathbf{w}^*_t - \eta_t \, \cancel{\nabla \mathbf{P}(\mathbf{w}^*_t)} \cdot \widetilde{\nabla} \ell\big(\mathbf{P}(\mathbf{w}^*_t)\big)$$

- Y. Bengio, N. Léonard, and A. Courville. *Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation*. arXiv (2013).

"Straight-through"

$$\min_{\mathbf{w}\in Q} \ \ell(\mathbf{w}) \xrightarrow{\text{reparameterize}} \min_{\mathbf{w}^*\in\mathbb{R}^d} \ \ell\big(\mathbf{P}(\mathbf{w}^*)\big)$$

- Forward pass: apply quantizer P
- Backward pass: ignore quantizer P

$$\mathbf{w}^*_{t+1} = \mathbf{w}^*_t - \eta_t \, \cancel{\nabla \mathbf{P}(\mathbf{w}^*_t)} \cdot \widetilde{\nabla}\, \ell\big(\mathbf{P}(\mathbf{w}^*_t)\big)$$

- Black magic is necessary? ∇P does not exist!

- Y. Bengio, N. Léonard, and A. Courville. *Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation*. arXiv (2013).

Proximal Quantization

$$\mathbf{w}_{t+1} = \mathbf{P}\left(\mathbf{w}_t - \eta_t \widetilde{\nabla}\, \ell(\mathbf{w}_t)\right)$$

- Y. Bai, Y.-X. Wang, and E. Liberty. *ProxQuant: Quantized Neural Networks via Proximal Operators*. ICLR (2018).

Proximal Quantization

$$\mathbf{w}_{t+1} = \mathbf{P}\left(\mathbf{w}_t - \eta_t \widetilde{\nabla}\, \ell(\mathbf{w}_t)\right)$$

- Constantly studied since 60s, to this day

• Y. Bai, Y.-X. Wang, and E. Liberty. *ProxQuant: Quantized Neural Networks via Proximal Operators*. ICLR (2018).

Proximal Quantization

$$\mathbf{w}_{t+1} = \mathbf{P}\left(\mathbf{w}_t - \eta_t \widetilde{\nabla} \ell(\mathbf{w}_t)\right)$$

- Constantly studied since 60s, to this day
- Again, with P = id, we recover the usual SGD for training NNs

- Y. Bai, Y.-X. Wang, and E. Liberty. *ProxQuant: Quantized Neural Networks via Proximal Operators*. ICLR (2018).

Proximal Quantization

$$\mathbf{w}_{t+1} = \mathbf{P}\left(\mathbf{w}_t - \eta_t \widetilde{\nabla}\, \ell(\mathbf{w}_t)\right)$$

- Constantly studied since 60s, to this day
- Again, with P = id, we recover the usual SGD for training NNs
- If ℓ is smooth and $\eta_t \leq \eta_0$, converges (warning: may not mean much!)

- Y. Bai, Y.-X. Wang, and E. Liberty. *ProxQuant: Quantized Neural Networks via Proximal Operators*. ICLR (2018).

Proximal Quantization

$$\mathbf{w}_{t+1} = \mathbf{P}\big(\mathbf{w}_t - \eta_t \widetilde{\nabla}\,\ell(\mathbf{w}_t)\big)$$

- Constantly studied since 60s, to this day
- Again, with P = id, we recover the usual SGD for training NNs
- If ℓ is smooth and $\eta_t \leq \eta_0$, converges (warning: may not mean much!)
- In implementation: $\eta_t \to \infty$!

- Y. Bai, Y.-X. Wang, and E. Liberty. *ProxQuant: Quantized Neural Networks via Proximal Operators*. ICLR (2018).

The Similarity and Subtlety

The Similarity and Subtlety

- BC: $\mathbf{w}_t = \mathrm{P}(\mathbf{w}_t^*), \;\; \mathbf{w}_{t+1}^* = \mathbf{w}_t^* - \eta_t \widetilde{\nabla} \ell(\mathbf{w}_t)$
- PQ: $\mathbf{w}_t = \mathrm{P}(\mathbf{w}_t^*), \;\; \mathbf{w}_{t+1}^* = \mathbf{w}_t - \eta_t \widetilde{\nabla} \ell(\mathbf{w}_t)$
- PT: $\mathbf{w}_t = \mathrm{P}(\mathbf{w}_t^*), \;\; \mathbf{w}_{t+1}^* = \mathbf{w}_t^* - \eta_t \widetilde{\nabla} \ell(\mathbf{w}_t^*)$

The Similarity and Subtlety

- BC: $\mathbf{w}_t = \mathrm{P}(\mathbf{w}_t^*), \quad \mathbf{w}_{t+1}^* = \mathbf{w}_t^* - \eta_t \widetilde{\nabla} \ell(\mathbf{w}_t)$
- PQ: $\mathbf{w}_t = \mathrm{P}(\mathbf{w}_t^*), \quad \mathbf{w}_{t+1}^* = \mathbf{w}_t - \eta_t \widetilde{\nabla} \ell(\mathbf{w}_t)$
- rBC: $\mathbf{w}_t = \mathrm{P}(\mathbf{w}_t^*), \quad \mathbf{w}_{t+1}^* = \mathbf{w}_t - \eta_t \widetilde{\nabla} \ell(\mathbf{w}_t^*)$
- PT: $\mathbf{w}_t = \mathrm{P}(\mathbf{w}_t^*), \quad \mathbf{w}_{t+1}^* = \mathbf{w}_t^* - \eta_t \widetilde{\nabla} \ell(\mathbf{w}_t^*)$

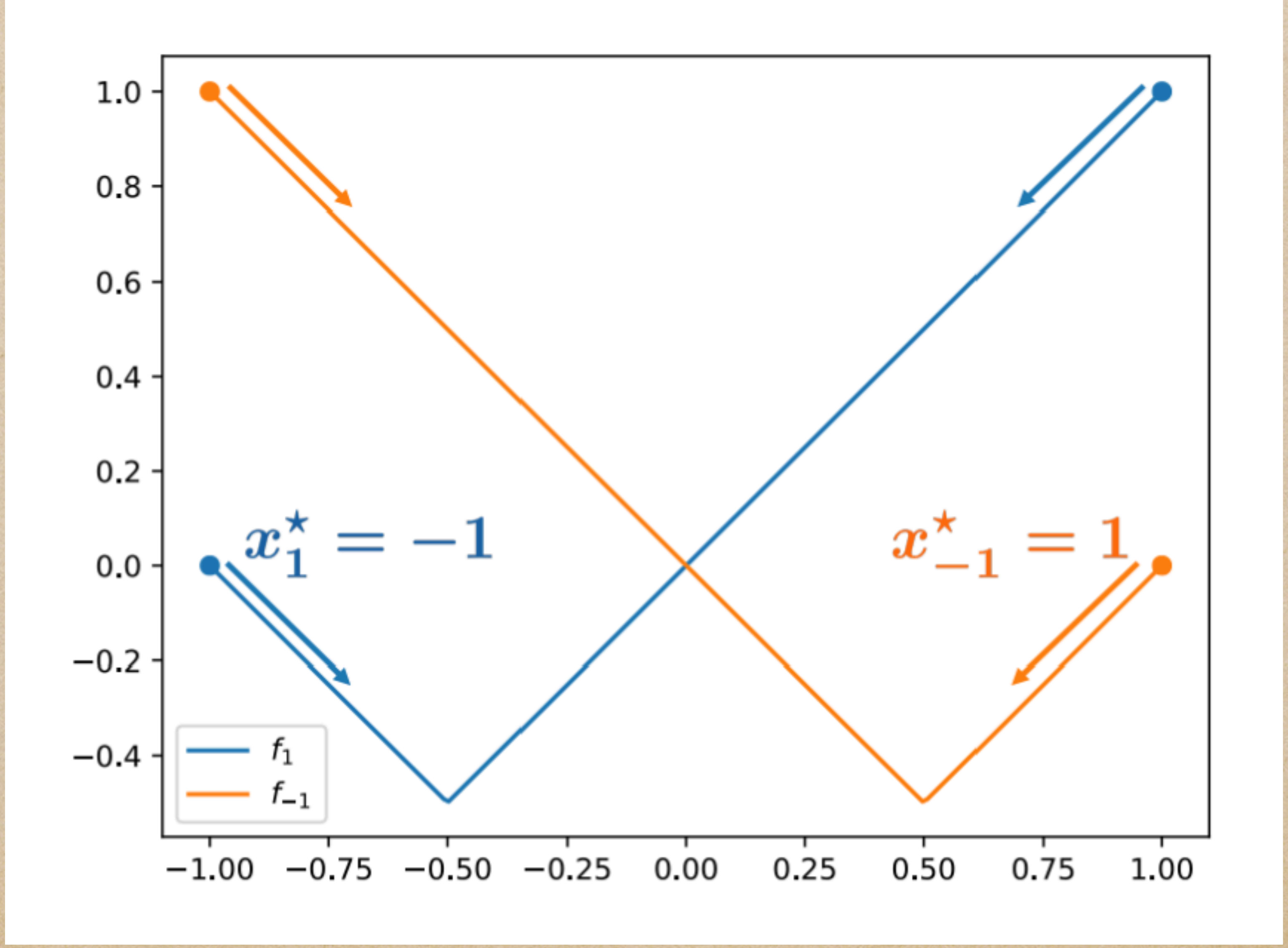
PT Doesn't Work At ALL

Model	Dataset	Full Precision	Binary		Ternary		Quaternary	
			without cal	with cal	without cal	with cal	without cal	with cal
ResNet20	CIFAR-10	92.01	10.17	17.71	10.00	11.89	9.30	35.41
ResNet56	CIFAR-10	93.01	10.41	39.15	10.00	9.99	10.06	58.31

- Cal: Training the BatchNorm layers for 1 epoch

The Problem of BC

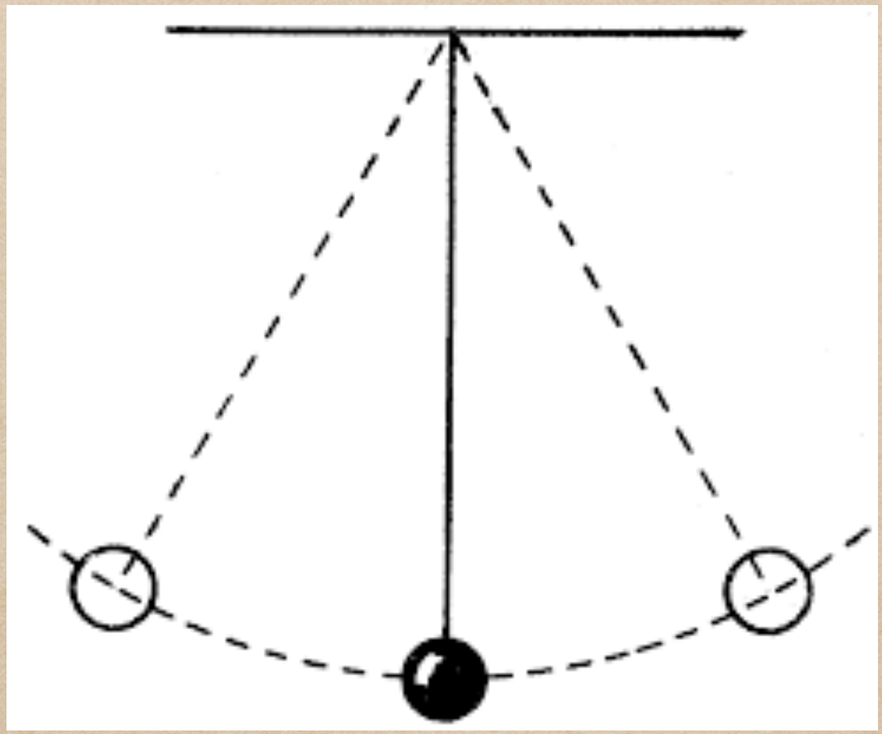
$$\mathbf{w}_t = \mathrm{P}(\mathbf{w}_t^*), \quad \mathbf{w}_{t+1}^* = \mathbf{w}_t^* - \eta_t \widetilde{\nabla} \ell(\mathbf{w}_t)$$



- Y. Bai, Y.-X. Wang, and E. Liberty. *ProxQuant: Quantized Neural Networks via Proximal Operators*. ICLR (2018).

The Problem of PQ

$$\mathbf{w}_t = \mathrm{P}(\mathbf{w}_t^*), \quad \mathbf{w}_{t+1}^* = \mathbf{w}_t - \eta_t \widetilde{\nabla} \ell(\mathbf{w}_t)$$



- Small update $\eta_t \widetilde{\nabla} \ell(\mathbf{w}_t)$ leads to $\mathbf{w}_{t+1} = \mathbf{w}_t$

What Makes a Good Quantizer?

Proximal Quantizer

$$\mathrm{P}_{\mathrm{r}}^{\nu}(\mathbf{w}^*) := \operatorname*{argmin}_{\mathbf{w}} \frac{1}{2\nu} \|\mathbf{w}^* - \mathbf{w}\|_2^2 + \mathrm{r}(\mathbf{w})$$

- If $\mathrm{r}(\mathbf{w}) = \iota_Q(\mathbf{w})$, reduce to projection: finding the discrete weight $\mathbf{w}$ in Q that is closest to the continuous weight $\mathbf{w}^*$.
- More generally, the regularizer $\mathrm{r}(\mathbf{w})$ penalizes deviation from discreteness (i.e. from Q).
- ν: allowing gradual transitioning from continuous to discrete.

How to choose the regularizer r?

- Typical choices: $\iota_Q(\mathbf{w})$, $\mathrm{dist}_Q(\mathbf{w})$, $\mathrm{dist}_Q^2(\mathbf{w})$
- From r to P: tedious and uninspiring calculation

$$\mathsf{P}_{\mathsf{r}}^{\nu}(\mathbf{w}^*) := \operatorname*{argmin}_{\mathbf{w}} \frac{1}{2\nu}\|\mathbf{w}^* - \mathbf{w}\|_2^2 + \mathsf{r}(\mathbf{w})$$

How to choose the regularizer r?

- Typical choices: $\iota_Q(\mathbf{w})$, $\text{dist}_Q(\mathbf{w})$, $\text{dist}_Q^2(\mathbf{w})$
- From r to P: tedious and uninspiring calculation

$$\mathrm{P}_{\mathrm{r}}^{\nu}(\mathbf{w}^*) := \operatorname*{argmin}_{\mathbf{w}} \frac{1}{2\nu}\|\mathbf{w}^* - \mathbf{w}\|_2^2 + \mathrm{r}(\mathbf{w})$$

- Don't!

A Direct Design Approach

Theorem 3.1 ([41, Proposition 3]). *A (possibly multi-valued) map* $\mathsf{P} : \mathbb{R} \rightrightarrows \mathbb{R}$ *is a proximal map (of some function* r*) if and only if it is (nonempty) compact-valued, monotone and has a closed graph. The underlying function* r *is unique (up to addition of constants) iff* P *is convex-valued, while* r *is convex iff* P *is nonexpansive (i.e.* 1*-Lipschitz continuous).*

Theorem 3.2. *Let* $\mathsf{P}_i : \mathbb{R}^d \rightrightarrows \mathbb{R}^d, i = 1, \ldots, k$ *be proximal maps. Then, the averaged map*

$$\mathsf{P} := \sum_{i=1}^k \alpha_i \mathsf{P}_i, \qquad \text{where } \alpha_i \geq 0, \ \ \sum_{i=1}^k \alpha_i = 1, \tag{9}$$

is also a proximal map. Similarly, the product map

$$\mathsf{P} := \mathsf{P}_1 \times \mathsf{P}_2 \times \cdots \times \mathsf{P}_k, \ \ \mathbf{w}^* = (\mathbf{w}_1^*, \ldots, \mathbf{w}_k^*) \mapsto \big(\mathsf{P}_1(\mathbf{w}_1^*), \ldots, \mathsf{P}_k(\mathbf{w}_k^*)\big) \tag{10}$$

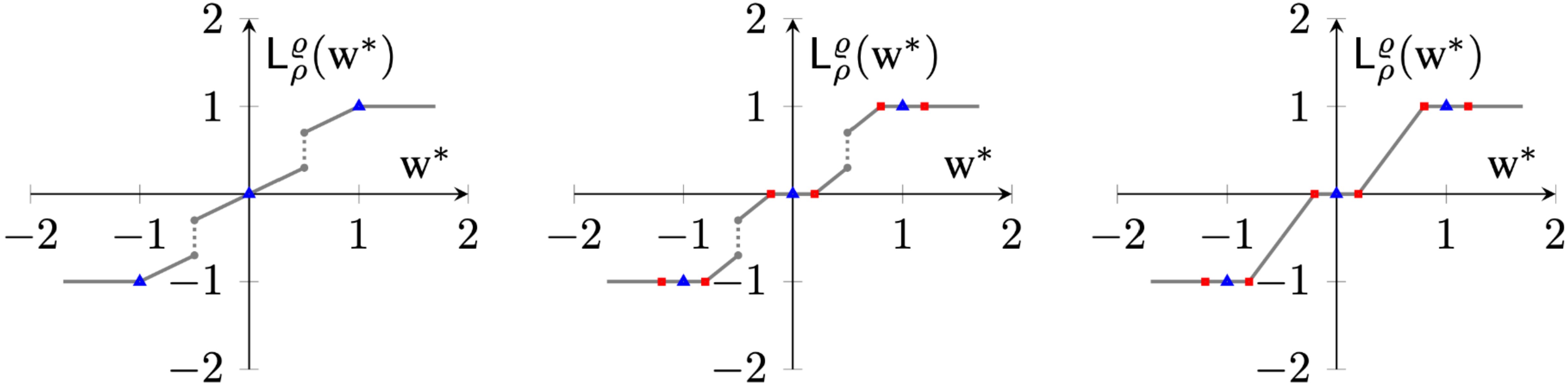
is a proximal map (from $\mathbb{R}^{dk}$ *to* $\mathbb{R}^{dk}$*).*

- Operationally, all we need to know is P.
- r is required for theoretical analysis: existence suffices

- Y. Yu, X. Zheng, M. Bowick and E. Xing. *Minimizing nonconvex and non-separable functions. AISTATS* (2015).

Example

$$\rho = \varrho = \frac{\nu}{2(\nu+1)}$$



(a) $\rho = 0, \varrho = 0.2$. (b) $\rho = \varrho = 0.2$. (c) $\rho = 0.2, \varrho = 0$.

Figure 1: Different instantiations of the proximal map L_ρ^ϱ in (13) for $Q = \{-1, 0, 1\}$.

- $\rho, \varrho \to 0$: P —> id, reduce to standard SGD
- $\rho, \varrho \to \infty$: P —> projection to Q, reduce to BC

Demystifying BC with 3 technical tools

Generalized Conditional Gradient

$$\min_{\mathbf{w}^*} f(\mathbf{w}^*) + g(\mathbf{w}^*)$$

- Y. Yu, X. Zhang and D. Schuurmans. *Generalized Conditional Gradient for Structured Sparse Estimation. JMLR* (2017).
- K. Bredies and D. Lorenz. *Iterated Hard Shrinkage for Minimization Problems with Sparsity Constraints*. SIAM SC (2008).

Generalized Conditional Gradient

$$\min_{\mathbf{w}^*} f(\mathbf{w}^*) + g(\mathbf{w}^*)$$

- Step 1: linearize f at current iterate $\mathbf{w}_t^*$

$$\min_{\mathbf{w}^*} f(\mathbf{w}_t^*) + (\mathbf{w}^* - \mathbf{w}_t^*) \cdot \nabla f(\mathbf{w}_t^*) + g(\mathbf{w}^*)$$

- Y. Yu, X. Zhang and D. Schuurmans. *Generalized Conditional Gradient for Structured Sparse Estimation. JMLR* (2017).
- K. Bredies and D. Lorenz. *Iterated Hard Shrinkage for Minimization Problems with Sparsity Constraints*. SIAM SC (2008).

Generalized Conditional Gradient

$$\min_{\mathbf{w}^*} f(\mathbf{w}^*) + g(\mathbf{w}^*)$$

- Step 1: linearize f at current iterate $\mathbf{w}_t^*$

$$\min_{\mathbf{w}^*} f(\mathbf{w}_t^*) + (\mathbf{w}^* - \mathbf{w}_t^*) \cdot \nabla f(\mathbf{w}_t^*) + g(\mathbf{w}^*)$$

- Y. Yu, X. Zhang and D. Schuurmans. *Generalized Conditional Gradient for Structured Sparse Estimation. JMLR* (2017).
- K. Bredies and D. Lorenz. *Iterated Hard Shrinkage for Minimization Problems with Sparsity Constraints*. SIAM SC (2008).

Generalized Conditional Gradient

$$\min_{\mathbf{w}^*} f(\mathbf{w}^*) + g(\mathbf{w}^*)$$

- Step 1: linearize f at current iterate $\mathbf{w}_t^*$

$$\min_{\mathbf{w}^*} f(\mathbf{w}_t^*) + (\mathbf{w}^* - \mathbf{w}_t^*) \cdot \nabla f(\mathbf{w}_t^*) + g(\mathbf{w}^*)$$

- Step 2: solve above to obtain $\mathbf{z}_t^* = \nabla g^*\big(-\nabla f(\mathbf{w}_t^*)\big)$

- Y. Yu, X. Zhang and D. Schuurmans. *Generalized Conditional Gradient for Structured Sparse Estimation. JMLR* (2017).
- K. Bredies and D. Lorenz. *Iterated Hard Shrinkage for Minimization Problems with Sparsity Constraints*. SIAM SC (2008).

Generalized Conditional Gradient

$$\min_{\mathbf{w}^*} f(\mathbf{w}^*) + g(\mathbf{w}^*)$$

- Step 1: linearize f at current iterate $\mathbf{w}_t^*$

$$\min_{\mathbf{w}^*} f(\mathbf{w}_t^*) + (\mathbf{w}^* - \mathbf{w}_t^*) \cdot \nabla f(\mathbf{w}_t^*) + g(\mathbf{w}^*)$$

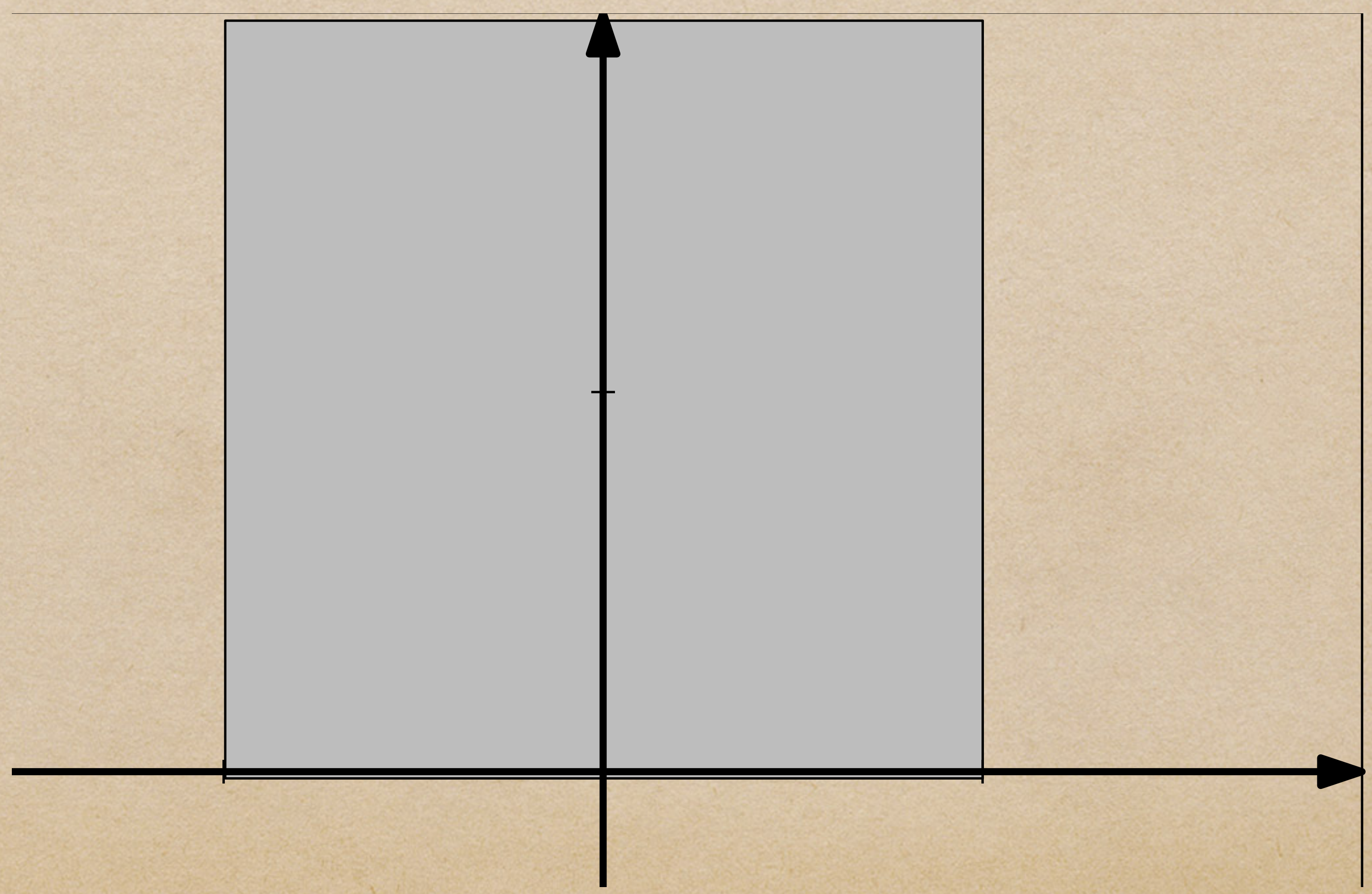
- Step 2: solve above to obtain $\mathbf{z}_t^* = \nabla g^*\big(-\nabla f(\mathbf{w}_t^*)\big)$
- Step 3: $\mathbf{w}_{t+1}^* = (1 - \lambda_t)\mathbf{w}_t^* + \lambda_t \mathbf{z}_t^*, \quad \lambda_t \in [0,1]$

- Y. Yu, X. Zhang and D. Schuurmans. *Generalized Conditional Gradient for Structured Sparse Estimation. JMLR* (2017).
- K. Bredies and D. Lorenz. *Iterated Hard Shrinkage for Minimization Problems with Sparsity Constraints.* SIAM SC (2008).

Example

$$\min_{|a|\leq 1, 0\leq b\leq 2} a^2 + (b+1)^2$$

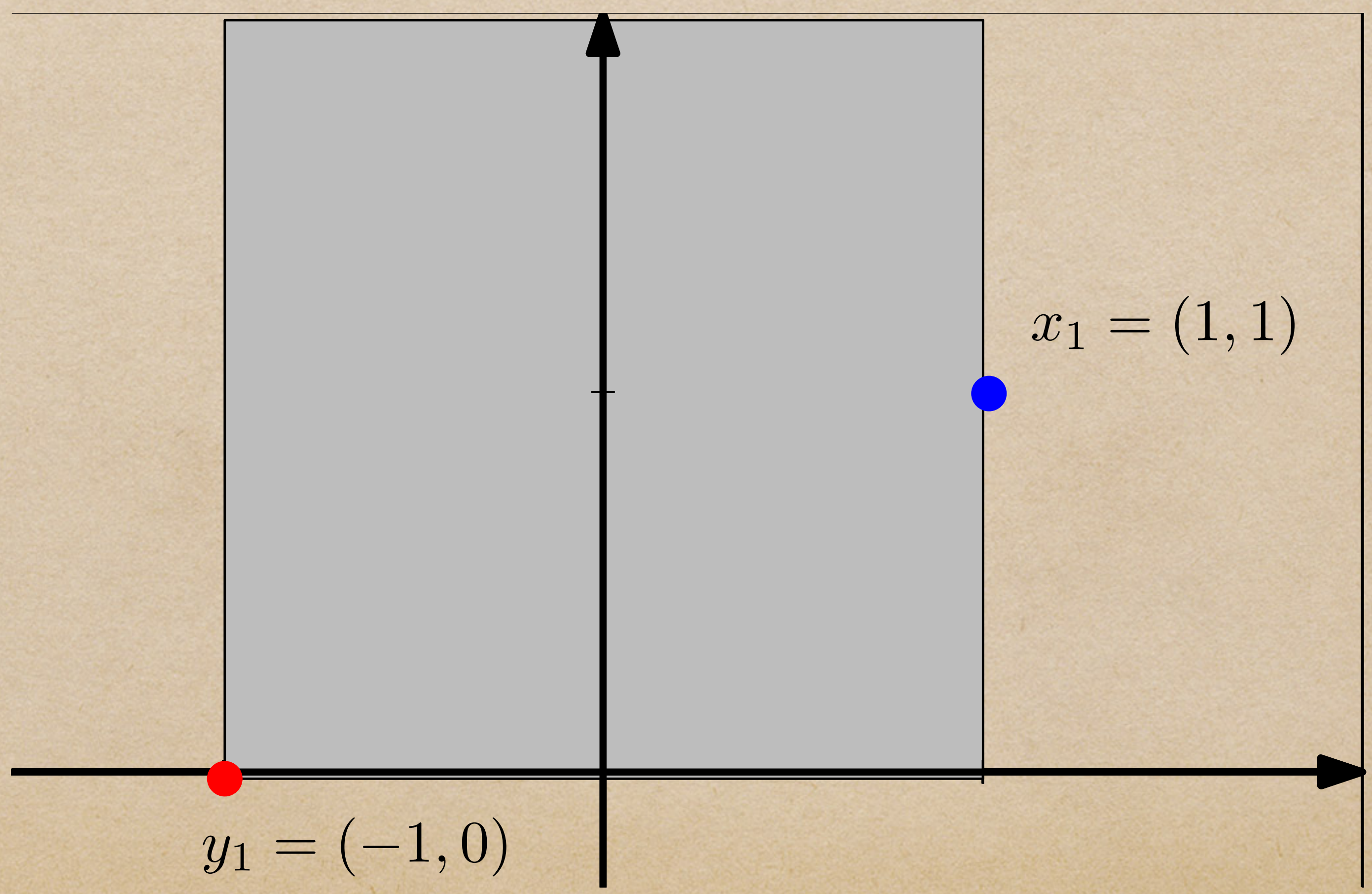
g f



Example

$$\min_{|a|\leq 1, 0\leq b\leq 2} a^2 + (b+1)^2$$

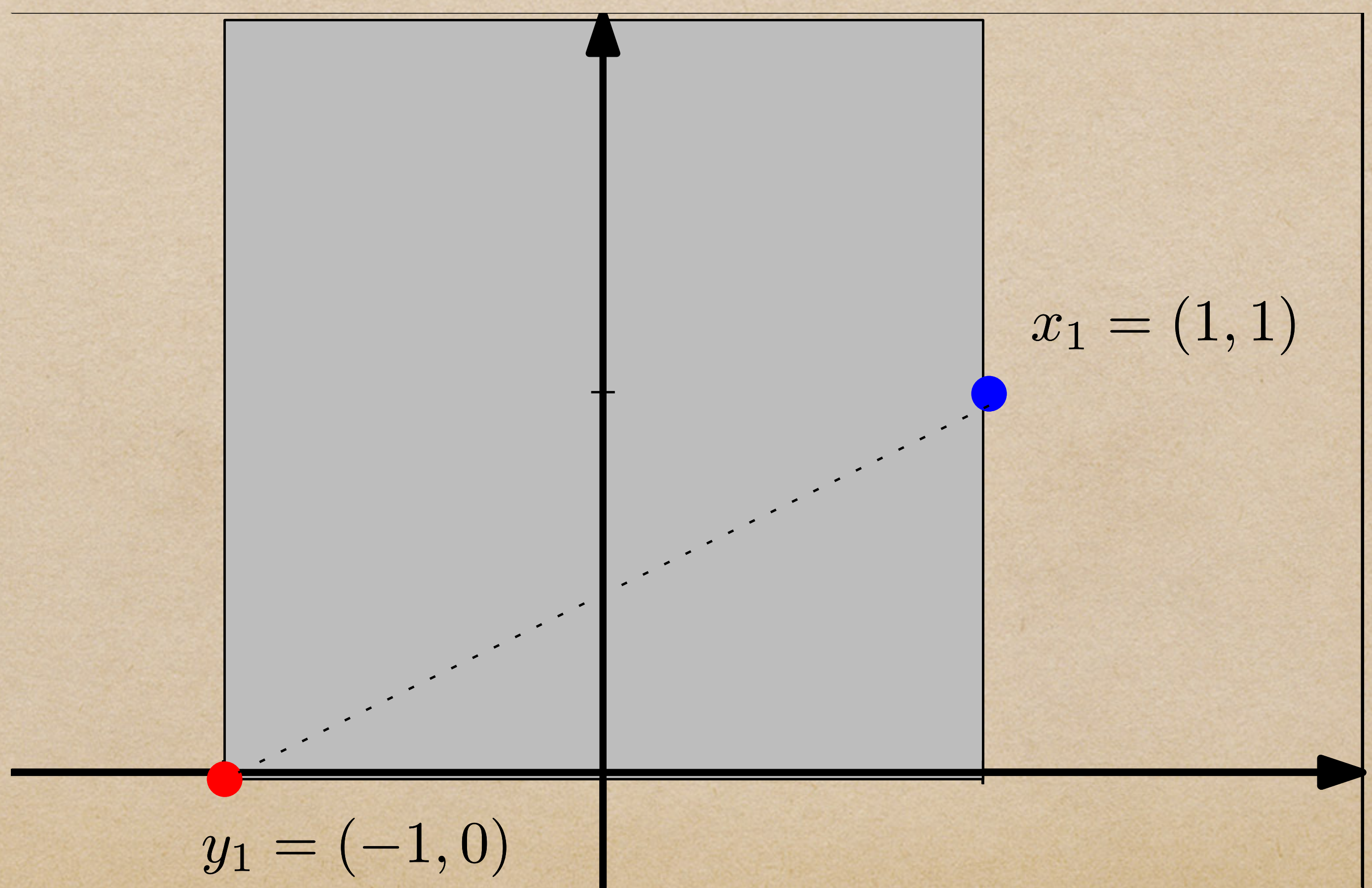
g f



Example

$$\min_{|a|\leq 1, 0\leq b\leq 2} a^2 + (b+1)^2$$

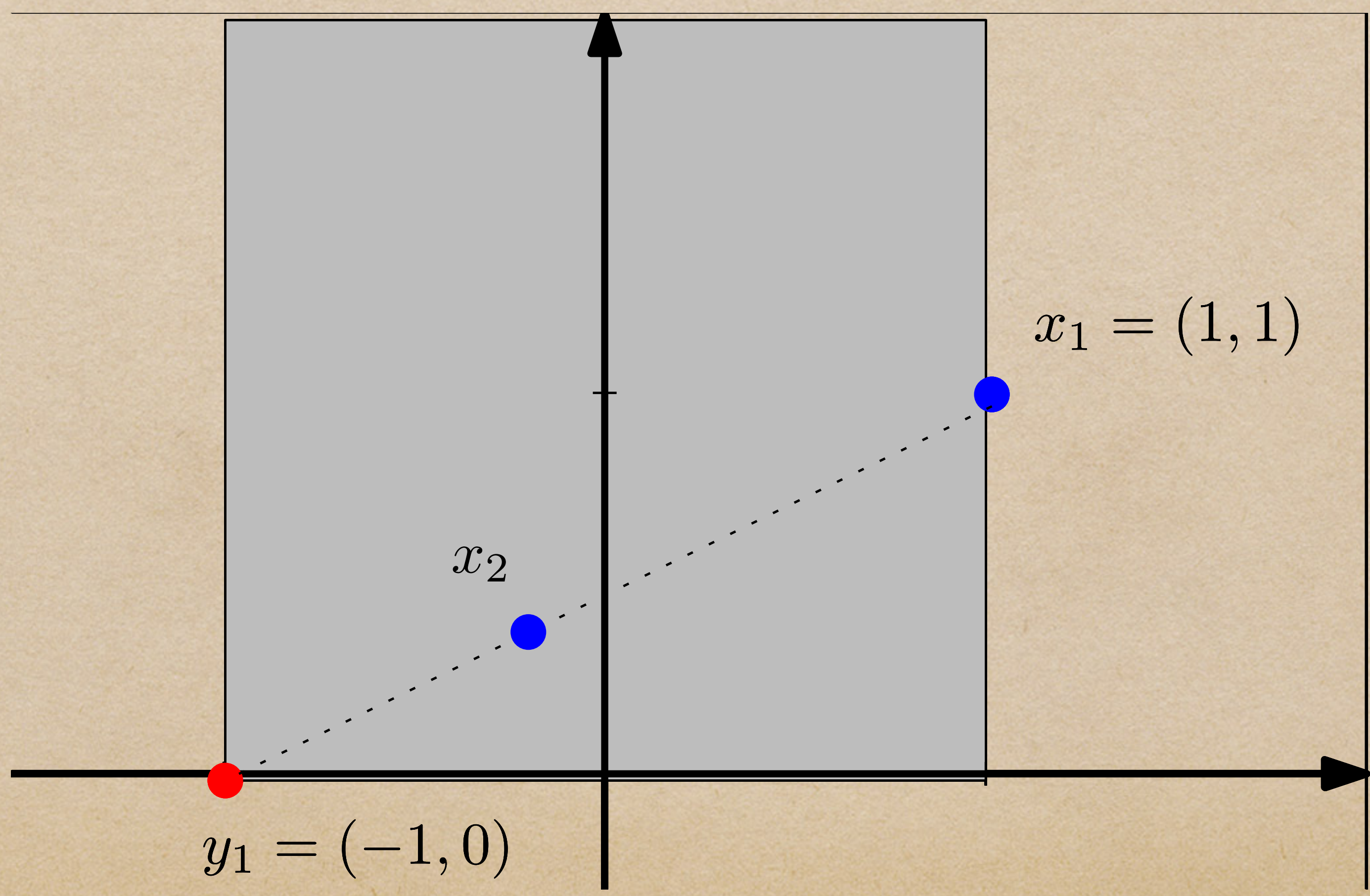
g f



Example

$$\min_{|a|\leq 1, 0\leq b\leq 2} a^2 + (b+1)^2$$

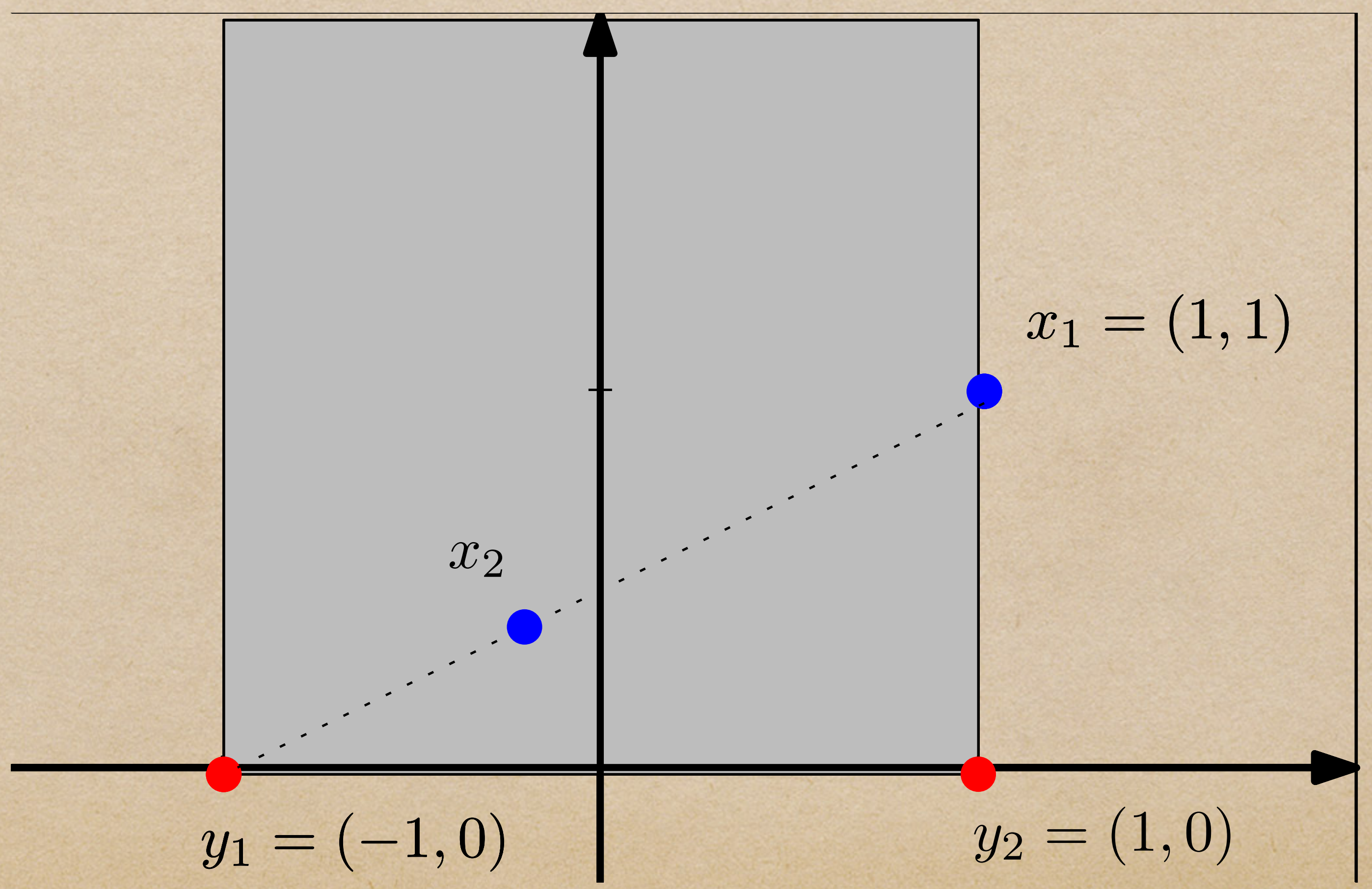
g f



Example

$$\min_{|a|\leq 1, 0\leq b\leq 2} a^2 + (b+1)^2$$

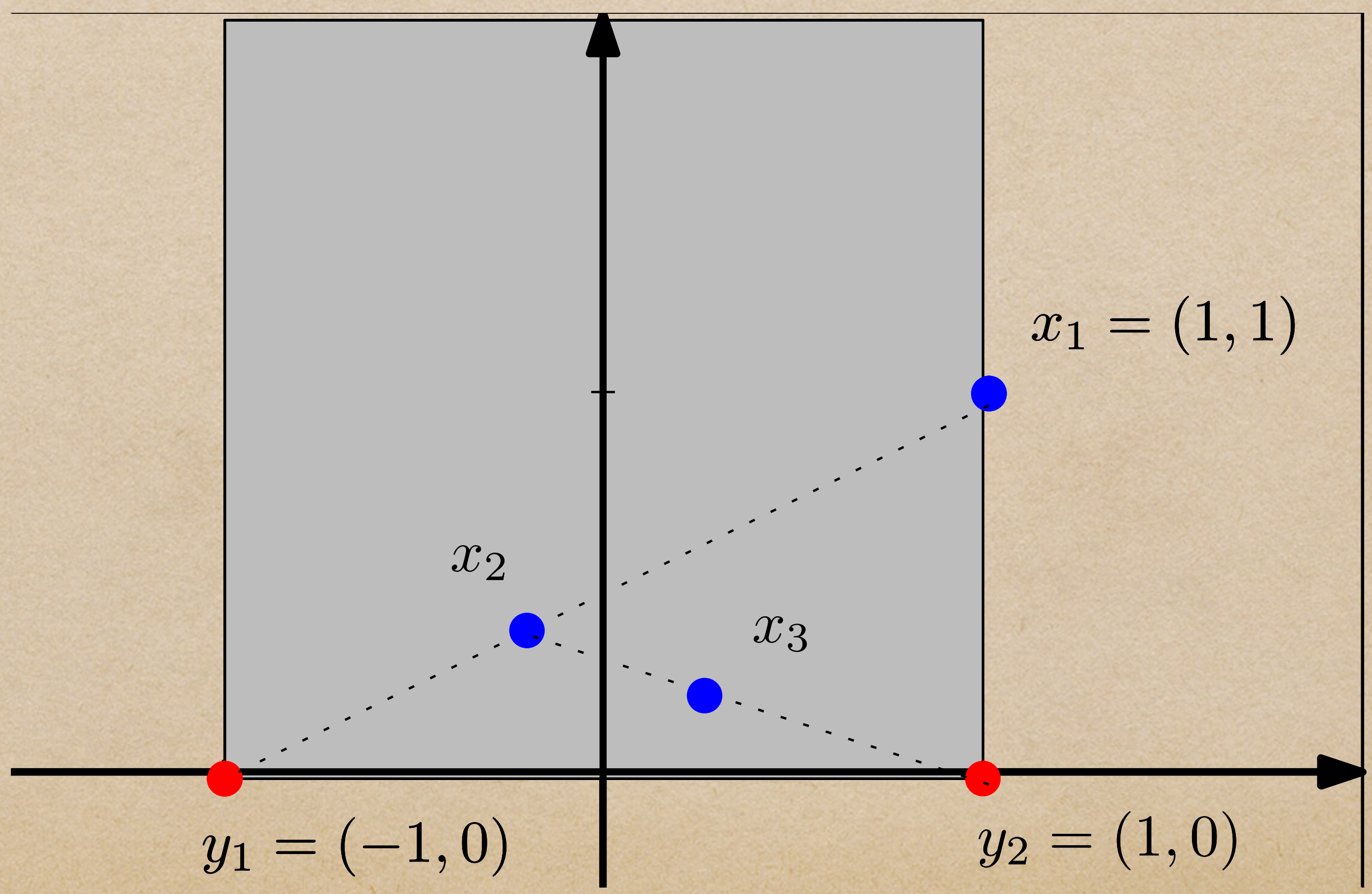
g f



Example

$$\min_{|a|\leq 1, 0\leq b\leq 2} a^2 + (b+1)^2$$

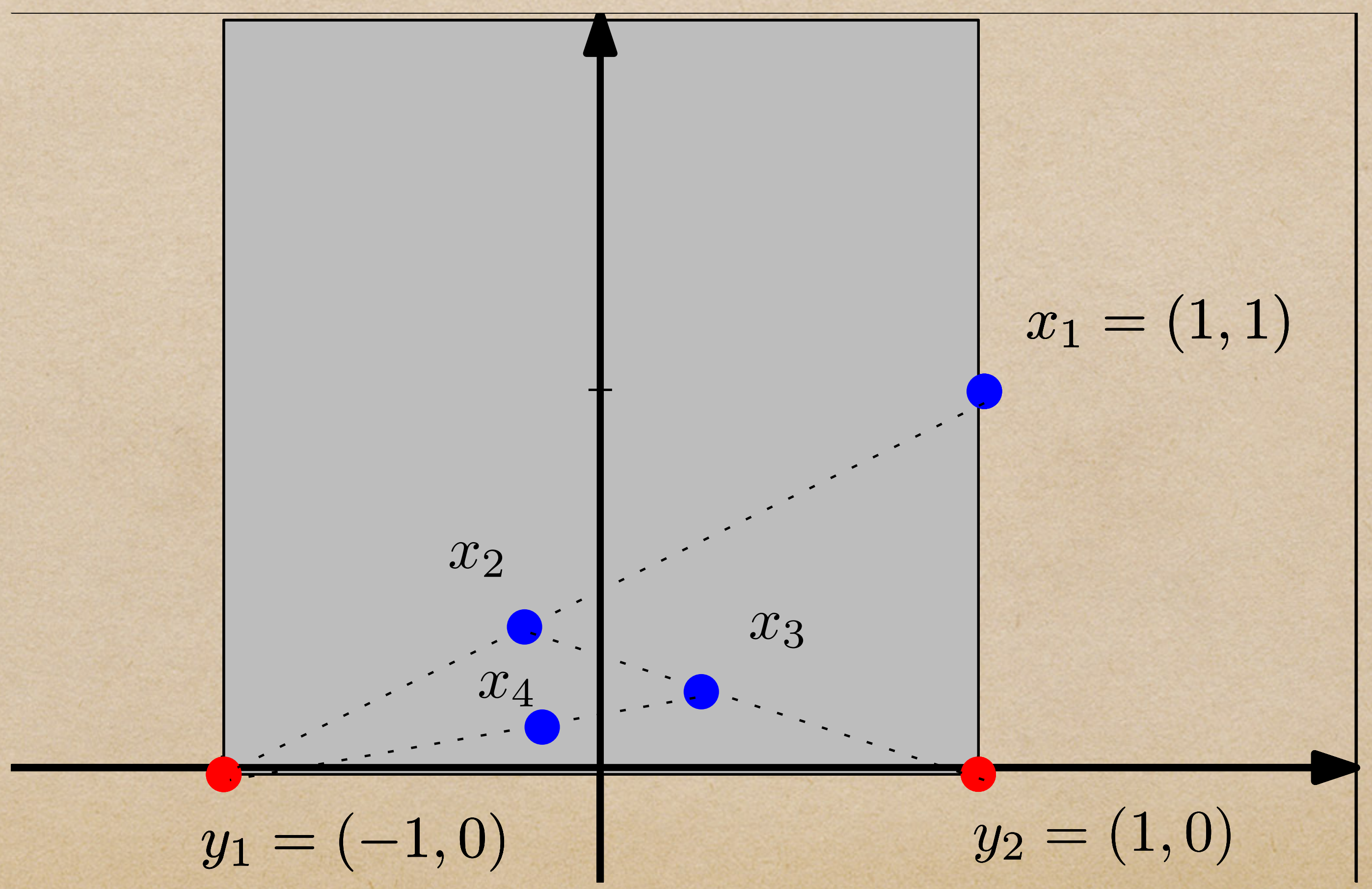
g f



Example

$$\min_{|a|\leq 1, 0\leq b\leq 2} a^2 + (b+1)^2$$

g f



Primal and Dual

$$\min_{\mathbf{w}} \; \ell(\mathbf{w}) + \mathrm{r}(\mathbf{w}) \quad \Longleftrightarrow \quad \min_{\mathbf{w}^*} \; \ell^*(-\mathbf{w}^*) + \mathrm{r}^*(\mathbf{w}^*)$$

- Fenchel conjugate: $\ell^*(\mathbf{w}^*) := \max_{\mathbf{w}} \; \mathbf{w}^\top \mathbf{w}^* - \ell(\mathbf{w})$
- Twins: solving one helps solving the other.
- Example: linear programming duality.
- Note: The dual is always a convex problem.

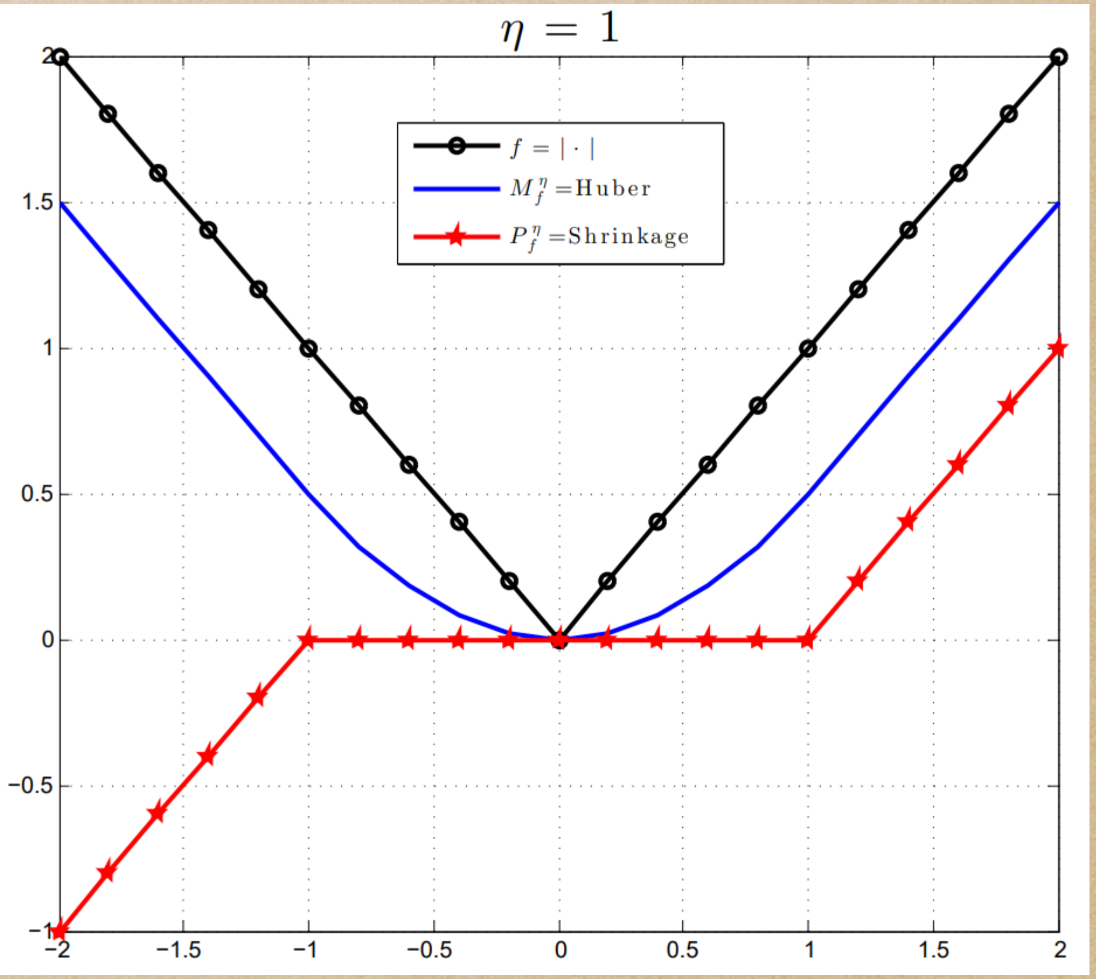
Moreau Envelope

$$\mathrm{M}_{\mathrm{r}}^{\mu}(\mathbf{w}^*) := \min_{\mathbf{w}} \frac{1}{2\mu}\|\mathbf{w}^* - \mathbf{w}\|_2^2 + \mathrm{r}(\mathbf{w})$$

- Smooth approximation
- μ controls error:

$$\mathrm{M}_{\mathrm{r}}^{\mu} \to \mathrm{r} \ \text{ as } \ \mu \downarrow 0$$

$$\nabla \mathrm{M}_{\mathrm{r}}^{\mu} = \frac{\mathrm{id} - \mathrm{P}_{\mathrm{r}}^{\mu}}{\mu} = \mathrm{P}_{\mathrm{r}^*}^{1/\mu}(\,\cdot\,/\mu)$$



Jean J. Moreau. *Fonctions convexes duales et points proximaux dans un espace hilbertien.* C.R.A.S. (1962)
Peter J. Huber. *Robust estimation of a location parameter.* Annals of Statistics (1964).

GCG —> Smoothened Dual

$$\min_{\mathbf{w}} \ \ell(\mathbf{w}) + \mathrm{r}(\mathbf{w})$$

$$\Downarrow$$

$$\min_{\mathbf{w}^*} \ \ell^*(-\mathbf{w}^*) + \mathrm{r}^*(\mathbf{w}^*)$$

$$\Downarrow$$

$$\min_{\mathbf{w}^*} \ \ell^*(-\mathbf{w}^*) + \mathrm{M}^{\mu}_{\mathrm{r}^*}(\mathbf{w}^*)$$

- As $\mu \downarrow 0$ we approach the original dual.
- $\mathrm{M}^{\mu}_{\mathrm{r}^*}$ is differentiable, with Lipschitz cont grad.

GCG —> Smoothened Dual

$$\min_{\mathbf{w}} \ \ell(\mathbf{w}) + r(\mathbf{w})$$

↓

$$\min_{\mathbf{w}^*} \ \ell^*(-\mathbf{w}^*) + r^*(\mathbf{w}^*)$$

↓

g

$$\min_{\mathbf{w}^*} \ \boxed{\ell^*(-\mathbf{w}^*)} + M^{\mu}_{r^*}(\mathbf{w}^*)$$

- As $\mu \downarrow 0$ we approach the original dual.
- $M^{\mu}_{r^*}$ is differentiable, with Lipschitz cont grad.

GCG —> Smoothened Dual

$$\min_{\mathbf{w}} \ \ell(\mathbf{w}) + \mathrm{r}(\mathbf{w})$$

$$\min_{\mathbf{w}^*} \ \ell^*(-\mathbf{w}^*) + \mathrm{r}^*(\mathbf{w}^*)$$

g

$$\min_{\mathbf{w}^*} \ \ell^*(-\mathbf{w}^*) + \mathrm{M}^{\mu}_{\mathrm{r}^*}(\mathbf{w}^*)$$

f

- As $\mu \downarrow 0$ we approach the original dual.
- $\mathrm{M}^{\mu}_{\mathrm{r}^*}$ is differentiable, with Lipschitz cont grad.

Unpacking

- Step 1: $\mathbf{w}_t := \nabla \mathrm{M}^{\mu}_{\mathrm{r}*}(\mathbf{w}^*_t) = \mathrm{P}^{1/\mu}_{\mathrm{r}**}(\mathbf{w}^*_t/\mu)$
- Step 2: $\mathbf{z}^*_t := -\nabla \ell^{**}(\mathbf{w}_t)$
- Step 3: $\mathbf{w}^*_{t+1} = (1-\lambda_t)\mathbf{w}^*_t + \lambda_t \mathbf{z}^*_t, \quad \lambda_t \in [0,1]$
- f^{**} is the convex hull of f, the best convex approximation in some sense

Simplifying

$$\pi_t = \prod_{s=1}^{t} (1 - \lambda_s)$$

$$\eta_t = \frac{\lambda_t}{\pi_t}$$

- Upon change-of-variable:

$$\mathbf{w}_t = \mathrm{P}^{1/\mu}_{\mathrm{r}^{**}}(\pi_{t-1}\mathbf{w}_t^*/\mu), \quad \mathbf{w}_{t+1}^* = \mathbf{w}_t^* - \eta_t \nabla \ell^{**}(\mathbf{w}_t)$$

- Allow $\mu = \mu_t = \pi_{t-1}$ to adapt with iteration:

$$\mathbf{w}_t = \mathrm{P}^{1/\pi_{t-1}}_{\mathrm{r}^{**}}(\mathbf{w}_t^*), \quad \mathbf{w}_{t+1}^* = \mathbf{w}_t^* - \eta_t \nabla \ell^{**}(\mathbf{w}_t)$$

- Replace with non convex originals:

$$\mathbf{w}_t = \mathrm{P}^{1/\pi_{t-1}}_{\mathrm{r}}(\mathbf{w}_t^*), \quad \mathbf{w}_{t+1}^* = \mathbf{w}_t^* - \eta_t \nabla \ell(\mathbf{w}_t)$$

The Nut is Cracked

- If $\mathbf{r}(\mathbf{w}) = \iota_Q(\mathbf{w})$, reduce to projection

$$\mathbf{w}_t = \mathrm{P}_{\mathsf{r}}^{1/\tau_{t-1}}(\mathbf{w}_t^*), \quad \mathbf{w}_{t+1}^* = \mathbf{w}_t^* - \eta_t \nabla \ell(\mathbf{w}_t)$$

- This is exactly BinaryConnect!

- Even for convex ℓ and r, new interpretation of the (regularized) dual averaging algorithm.

- Y. Nesterov. *Primal-dual subgradient methods for convex problems*. MP (2009).
- L. Xiao. *Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization*. JMLR (2010).

Proximal Connect

Proximal Connect

$$\mathbf{w}_t = \mathrm{P}_{\mathrm{r}}^{\nu_t}(\mathbf{w}_t^*) \qquad \mathbf{w}_{t+1}^* = \mathbf{w}_t^* - \eta_t \widetilde{\nabla} \ell(\mathbf{w}_t)$$

$$\nu_t = 1 + \sum_{\tau=1}^{t-1} \eta_\tau$$

- $\nu_t = 1/\mu_t \to \infty$ and $\mathrm{P}_{\mathrm{r}}^{\nu_t} \to \mathrm{P}_Q$.
- Diverging ν_t was a crucial hack prior to our justification here.
- Easily derive improved convergence guarantees.

Experiments

Fine-tuning

Table 2: Fine-tuning pretrained ResNets. Final test accuracy: mean and std over three runs.

Model	Quantization	BC [10]	PQ [5]	rPC (ours)	PC (ours)
ResNet20	Binary	**90.31** (0.00)	89.94 (0.10)	89.98 (0.17)	**90.31** (0.21)
	Ternary	74.95 (0.16)	91.46 (0.06)	**91.47** (0.19)	91.37 (0.18)
	Quaternary	91.43 (0.07)	91.43 (0.21)	91.43 (0.06)	**91.81** (0.14)
ResNet56	Binary	92.22 (0.12)	92.33 (0.06)	92.47 (0.29)	**92.65** (0.16)
	Ternary	74.68 (1.4)	93.07 (0.02)	92.84 (0.11)	**93.25** (0.12)
	Quaternary	93.20 (0.06)	92.82 (0.16)	92.91 (0.26)	**93.42** (0.12)

End-to-end Training

Table 3: End-to-end training of ResNets. Final test accuracy: mean and std over three runs.

Model	Quantization	BC [10]	PQ [5]	rPC (ours)	PC (ours)
ResNet20	Binary	87.51 (0.21)	81.59 (0.75)	81.82 (0.32)	**89.92** (0.65)
	Ternary	27.10 (0.21)	47.98 (1.30)	47.17 (1.94)	**84.09** (0.16)
	Quaternary	89.91 (0.09)	85.29 (0.09)	85.05 (0.27)	**90.17** (0.14)
ResNet56	Binary	89.79 (0.45)	86.13 (1.71)	86.25 (1.50)	**91.26** (0.59)
	Ternary	30.31 (7.79)	50.54 (3.68)	42.95 (1.57)	**84.36** (0.75)
	Quaternary	90.69 (0.57)	87.81 (1.60)	87.30 (1.02)	**91.70** (0.14)

Conclusion

Summary

- Existing quantization algorithms are not that different from each other
- A convenient design of proximal quantizers
- BC is GCG applied to the smoothened dual
- ProxConnect unifies and extends SOTA
- Open possibilities for acceleration and new applications

Thank you!

[redacted]