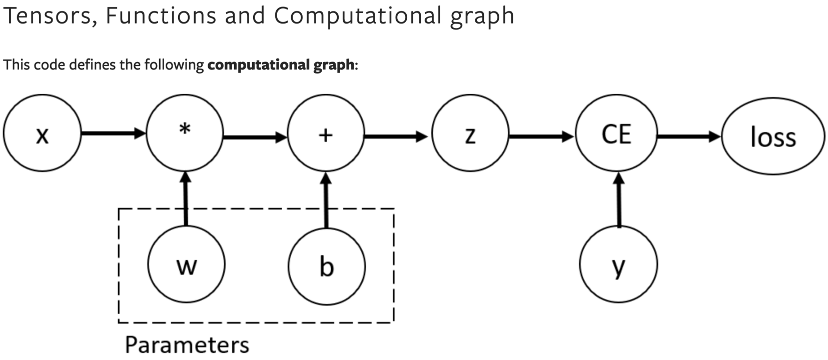
Straight-Through Meets Sparse Recovery: the Support Exploration Algorithm

Mimoun MOHAMED, François MALGOUYRES, Valentin EMIYA and Caroline CHAUX

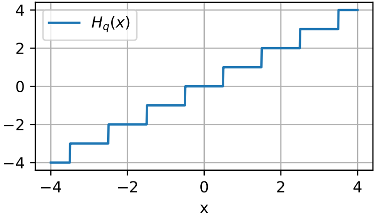
Today, learning relies on differentiability

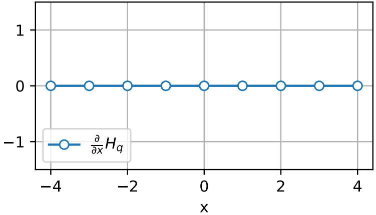


[Source: `https://pytorch.org/tutorials/beginner/basics/autogradqs_tutorial.html`]

Learning by gradient descent: repeat $\begin{cases} w & \leftarrow w - \eta \frac{\partial loss}{\partial w} \\ b & \leftarrow b - \eta \frac{\partial loss}{\partial b} \end{cases}$

Quantization is not differentiable

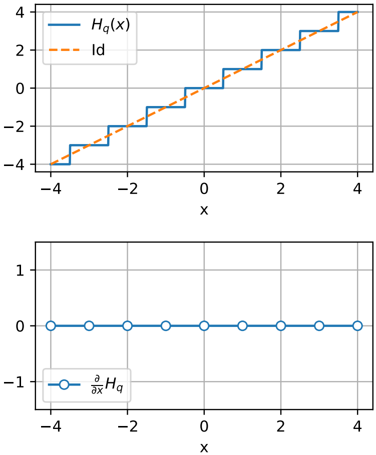




Quantization $H_q(x)$:

- not differentiable at some points
- gradient = 0

Quantization is not differentiable



Quantization $H_q(x)$**:**

- not differentiable at some points
- gradient = 0

Straight-Through Estimator[1, 2, 3] (STE):
replace $\frac{\partial H_q}{\partial x}$ by derivative of Id.

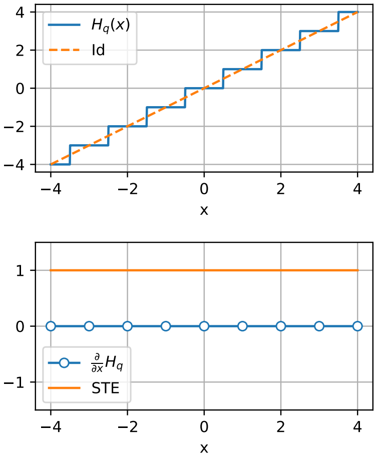
$$\frac{\partial H_q}{\partial x} \approx 1$$

[1] G. E. Hinton, *Neural networks for machine learning*, Coursera, video lectures, Lecture 15b, 2012

[2] Y. Bengio et al., "Estimating or propagating gradients through stochastic neurons for conditional computation", CoRR **arXiv:1308.3432** (2013)

[3] M. Courbariaux et al., "Binaryconnect: training deep neural networks with binary weights during propagations", Advances in neural information processing systems **28** (2015)

Quantization is not differentiable



Quantization $H_q(x)$**:**

- not differentiable at some points
- gradient = 0

Straight-Through Estimator[1, 2, 3] (STE):
replace $\frac{\partial H_q}{\partial x}$ by derivative of Id.

$$\frac{\partial H_q}{\partial x} \approx 1$$

[1] G. E. Hinton, *Neural networks for machine learning*, Coursera, video lectures, Lecture 15b, 2012

[2] Y. Bengio et al., “Estimating or propagating gradients through stochastic neurons for conditional computation”, CoRR **arXiv:1308.3432** (2013)

[3] M. Courbariaux et al., “Binaryconnect: training deep neural networks with binary weights during propagations”, Advances in neural information processing systems **28** (2015)

Sparse Support Recovery in Linear Models

Goal: Recover $S^* = \text{supp}(x^*)$ from observation

$$y = Ax^* + e \in \mathbb{R}^m$$

with $x^* \in \mathbb{R}^n$ s.t. $\|x^*\|_0 \leq k$ and $A \in \mathbb{R}^{m \times n}$

Problem formulation:

$$\underset{x \in \mathbb{R}^n,\ \|x\|_0 \leq k}{\text{Minimize}} \ F(x) := \frac{1}{2}\|Ax - y\|_2^2 \qquad \text{(SPARSE)}$$

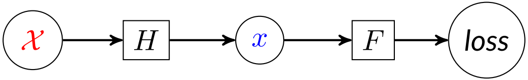
ℓ_0 sparsity constraint $\Rightarrow$ NP-Hard[4] problem

- Non-differentiable
- Non-convex
- Combinatorial $\rightarrow \binom{n}{k}$ possible supports
- Trivial if A orthogonal, difficult if A is *coherent*.

---

[4] G. Davis et al., "Adaptive greedy approximations", Constr. Approx. **13**, 57–98 (1997)

Reformulation of the Sparse Support Recovery Problem



We can prove[5] an equivalence between

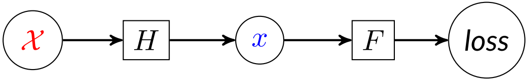
$$\underset{x\in\mathbb{R}^n,\ \|x\|_0\leq k}{\text{Minimize}}\ F(x) = \frac{1}{2}\|Ax - y\|_2^2 \qquad \text{and} \qquad \underset{\mathcal{X}\in\mathbb{R}^n}{\text{Minimize}}\, F(H(\mathcal{X}))$$

where H is the sparsification operator $\quad H(\mathcal{X}) \in \underset{\substack{x\in\mathbb{R}^n \\ \text{supp}(x)\ \subseteq\ \text{largest}_k(\mathcal{X})}}{\text{argmin}} \|Ax - y\|_2^2$

H is not differentiable $\rightarrow$ Straight-Through Estimator (STE): $\frac{\partial H}{\partial \mathcal{X}}(\mathcal{X}) \approx 1$

[5] M. Mohamed et al., "Straight-through meets sparse recovery: the support exploration algorithm", ICML (2024)

Straight-Through Estimator for Sparse Support Recovery



$$\mathcal{X}^{t+1} = \mathcal{X}^t - \eta \frac{\partial (F \circ H)}{\partial \mathcal{X}}(\mathcal{X}^t)$$

$$= \mathcal{X}^t - \eta \frac{\partial F}{\partial x}(H(\mathcal{X}^t)) \cdot \frac{\partial H}{\partial \mathcal{X}}(\mathcal{X}^t) \qquad \text{(chain rule)}$$

$$\approx \mathcal{X}^t - \eta \frac{\partial F}{\partial x}(H(\mathcal{X}^t)) \cdot 1 \qquad \text{(STE update)}$$

$$= \mathcal{X}^t - \eta A^T(Ax^t - y) \qquad \left(H(\mathcal{X}^t) = x^t \;\&\; F(x) = \frac{1}{2}\|Ax - y\|_2^2\right)$$

SEA iterative scheme: $\mathcal{X}^{t+1} = \mathcal{X}^t - \eta A^T(Ax^t - y)$

Contributions in the paper

- SEA: a new algorithm for sparse support recovery in linear models
- An STE derived for sparse recovery (not quantization)
- Generates ability to explore beyond local minima
- Good performance in difficult settings (A strongly coherent, e.g., in spike deconvolution)
- Theoretical recovery guarantees under RIP hypothesis

Paper

Straight-Through Meets Sparse Recovery: the Support Exploration Algorithm

Poster session 2: Tue 23 Jul 1:30 p.m. - 3 p.m., Hall C 4-9 #1105