Differentiable and Sparse Top-k: a Convex Analysis Perspective

Mathieu Blondel
Motivation for the research

- The top-k operator is increasingly used as a **building block** in neural networks (top-k classification, mixtures of expert, weight pruning).

- However, it is a **discontinuous** operation, making it difficult to use in end-to-end trainable networks.

- A crucial property of the top-k is its **sparsity** but many existing differentiable top-k relaxations are **dense**.

- **Smooth optimization** is known to enjoy **faster** convergence rates.

- However, sparsity is crucial in certain applications as a **selection mechanism**: mixtures of experts, weight pruning.
Related work

A large body of work on relaxations of sorting, ranking and top-k...

- Using **unimodal row-stochastic matrices** (Grover et al, 2019; Prillo and Eisenschlos, 2020)
- Using **optimal transport** (Cuturi et al, 2019)
- Using the **permutahedron** (Blondel et al, 2020)
- Using **perturbations** (Berthet et al, 2020)
- Using **sorting networks** (Petersen et al, 2021)
Contributions

- A **general** top-k framework, including top-k in magnitude
- Differentiable and sparse relaxations thanks to $p$-norm regularization
- Reduction to **isotonic optimization**, for computation and differentiation
- **GPU/TPU-friendly** algorithm based on Dykstra’s algorithm
- Applications to top-k classification, mixtures of experts, weight pruning
Top-k mask
Top-k mask operator

Bit-encoding of the top-k indices ("k-hot encoding")

\[
\text{topkmask}(x)_i := \begin{cases} 
1, & \text{if } \text{rank}(x)_i \leq k \\
0, & \text{otherwise.}
\end{cases} 
\]

\[x = (1.7, 3.2, -2.4)\]

\[\text{top1mask}(x) = (0, 1, 0)\]
\[\text{top2mask}(x) = (1, 1, 0)\]
\[\text{top3mask}(x) = (1, 1, 1)\]

Discontinuous, piecewise constant with null derivatives
Top-k operator

Sparse vector containing the top-k values

\[ \text{topk}(x) := x \cdot \text{topkmask}(x) \in \mathbb{R}^n \]

\[ x = (1.7, 3.2, -2.4) \]
\[ \text{top1}(x) = (0.0, 3.2, 0) \]
\[ \text{top2}(x) = (1.7, 3.2, 0) \]
\[ \text{top3}(x) = (1.7, 3.2, -2.4) \]

Discontinuous, piecewise affine with constant derivatives
Regularized top-k mask: overview of the approach

- Rewrite top-k mask as a linear program solution
  \[
  \text{topkmask}(x) = y(x) := \arg\max_{y \in C} \langle x, y \rangle
  \]

- Add regularization $R$
  \[
  \text{topkmask}_R(x) = y_R(x) := \arg\max_{y \in C} \langle x, y \rangle - R(y)
  \]

- Use a reduction to isotonic optimization to easily compute and differentiate $\text{topkmask}_R(x)$
Our framework is significantly more general than the existing one and allows for example to express isotonic optimization problems. We successfully use our operators to prune weights in neural networks, for instance the case in sparse mixture of experts, where the router maps non-zero values correspond to the affine function with derivatives either undefined or constant a.e. Moreover, our operators are adaptable on the algorithmic side, in addition to pool adjacency structures, and to sparsify a neural network, by removing weights with the smallest magnitude (e.g., top-

$$\theta(s) = (3, 1, -1 + s, s) \in \mathbb{R}^4$$

$$k = 2$$
Top-k mask as a linear program

- With $C = \{ y \in \mathbb{R}^n : y \in [0, 1]^n, y^\top 1 = k \}$, we get
  \[
  \text{topkmask}(x) = y(x) = \arg\max_{y \in C} \langle x, y \rangle
  \]
- The vertices of $C$ are all possible bit encodings of cardinality $k$
- Relation with the **capped probability simplex**
  \[
  C/k = \left\{ y \in \mathbb{R}^n : y \in [0, 1/k]^n, y^\top 1 = 1 \right\}
  \]
Relation with the permutahedron

- The convex hull of all permutations of \( w \)

\[
P(w) := \text{conv}(\{(w_{\sigma_1}, \ldots, w_{\sigma_n}) : \sigma \in \Sigma\})
\]

- With \( w = 1_k := (1, \ldots, 1, 0, \ldots, 0) \)\(^{\underline{k}}\) \((n-k)^{\underline{n-k}}\), we get

\[
P(w) = C = \{ y \in \mathbb{R}^n : y \in [0, 1]^n, y^\top 1 = k \}.
\]
Top-k mask: value function and its conjugate

- Value function: support function of $\mathcal{C}$

\[
f(x) := \max_{y \in \mathcal{C}} \langle x, y \rangle = \text{topksum}(x) := \sum_{i=1}^{k} x_{\sigma_i} = \langle x_{\sigma}, 1_k \rangle
\]

where $\sigma = \text{argsort}(x) \iff x_{\sigma_1} \geq \cdots \geq x_{\sigma_n}$ and $x_{\sigma} := (x_{\sigma_1}, \ldots, x_{\sigma_n})$

- Conjugate: indicator function of $\mathcal{C}$

\[
f^*(y) := \sup_{x \in \mathbb{R}^n} \langle x, y \rangle - f(x) = \delta_{\mathcal{C}}(y) := \begin{cases} 0, & \text{if } y \in \mathcal{C} \\ \infty, & \text{if } y \notin \mathcal{C} \end{cases}
\]
Regularized version

- The regularized version

\[ \text{topkmask}_R(x) = y_R(x) := y^* \]

is defined using the dual solution

\[
y^* = \arg\max_{y \in C} \langle x, y \rangle - R(y)
= \arg\max_{y \in \mathbb{R}^n} \langle x, y \rangle - f^*(y) - R(y)
\]

- Equivalently, if we define the primal solution (infimal convolution)

\[
u^* = \arg\min_{u \in \mathbb{R}^n} R^*(x - u) + f(u)
\]

then \( y^* = \nabla R^*(x - u^*) \)
Regularized version

\[
R(y) = \frac{1}{p} \|y\|_p^p = \frac{1}{p} \sum_{i=1}^{n} |y_i|^p
\]

with our relaxed operator. We use a differentiable top-k operator in magnitude. However, it is easy to see that the relaxed approach would also lead to a differentiable top-k (gradient a.e. top-k) or a differentiable (or differentiable a.e.) and sparse top-k mask operators, including operators that are made of a succession of self-attention layers and MLP. This alternative would also lead to a convolutional optimization, we showed that these operators can be computed using either the Pool Adjacent Violators (PAV) or a bi-conjugate interpretation, which we believe has an interest point of view, the introduction of a non-linearity allows us to draw connections with the mathematical discussions. We also thank the anonymous reviewers for their feedback.
Computing the regularized version

- Recall that the primal solution is

\[ u^* = \arg\min_{u \in \mathbb{R}^n} R^*(x - u) + f(u) \]

\[ = \arg\min_{u \in \mathbb{R}^n} R^*(x - u) + \langle u_\pi(u), 1_k \rangle \]

where \( \pi(u) = \text{argsort}(u) \)

- Reduction to isotonic optimization

\[ u^*_\sigma = \arg\min_{v_1 \geq \cdots \geq v_n} R^*(x_\sigma - v) + f(v) \]

\[ = \arg\min_{v_1 \geq \cdots \geq v_n} R^*(x_\sigma - v) + \langle v, 1_k \rangle \]

where \( \sigma = \text{argsort}(x) \)

- Differentiation available in closed form (implicit diff not needed) given \( v^* \)
Pool Adjacent Violators (PAV)

\[
\argmin_{v_1 \geq \cdots \geq v_n} \sum_{i=1}^{n} h_i(v_i)
\]

- Partitions the set \([n]\) into disjoint sets \((B_1, \cdots B_m)\), starting from \(m = n\) and \(B_i = \{i\}\)

- Merges these sets until the isotonic condition is met

- Needs to be able to solve \(\argmin_{\gamma \in \mathbb{R}} \sum_{i \in B_j} h_i(\gamma)\) in constant time to get \(O(n)\) total complexity
  - \(p\)-norm regularization case: we need to find the root of a polynomial (easy when \(p = 2\) or \(p = 4/3\))
Using Dykstra’s algorithm

- Key idea: alternate projections between $C_1$ and $C_2$

\[
\{ \mathbf{v} \in \mathbb{R}^n : v_1 \geq \cdots \geq v_n \} = \left\{ \mathbf{v} \in \mathbb{R}^n : v_1 \geq v_2, v_3 \geq v_4, \ldots \right\} \cap \left\{ \mathbf{v} \in \mathbb{R}^n : v_2 \geq v_3, v_4 \geq v_5, \ldots \right\}
\]

- Huge speedup on TPU

![Graph showing running time comparison between PAV, Dykstra, and Hard methods across varying dimensions.](image)
Top-k in magnitude
Top-k in magnitude operator

Same as top-k operator but selects elements with largest **absolute value**

\[
\text{topkmag}(x) := x \cdot \text{topkmask}(|x|)
\]

\[
x = (1.7, 3.2, -2.4)
\]

\[
\text{top1mag}(x) = (0.0, 3.2, 0)
\]

\[
\text{top2mag}(x) = (0, 3.2, -2.4)
\]

\[
\text{top3mag}(x) = (1.7, 3.2, -2.4)
\]
Top-k in magnitude as a gradient

- We introduce a **nonlinearity** \( \varphi(x) = (\phi(x_1), \ldots, \phi(x_n)) \)

\[
f_\varphi(x) := f(\varphi(x)) = \max_{y \in C} \langle \varphi(x), y \rangle
\]

- With \( \phi(x) = \frac{1}{2}x^2 \), we have

\[
\nabla f_\varphi(x) = \text{topkmag}(x)
\]

- With \( \phi(x) = x \), we have

\[
\nabla f_\varphi(x) = \nabla f(x) = \text{topkmask}(x)
\]
Regularized version

\[ \text{topkmag}_R(x) := y^* = \nabla R^*(x - u^*) \]

where we defined the **dual** solution

\[ y^* := \arg\max_{y \in \mathbb{R}^n} \langle x, y \rangle - f^*_\varphi(y) - R(y) \]

and the **primal** solution (inf–convolution)

\[ u^* = \arg\min_{u \in \mathbb{R}^n} R^*(x - u) + f_\varphi(u) \]
Conjugate

- For $\varphi(x) = x$: **indicator function** of $C$
  \[ f_{\varphi}^*(y) = f^*(y) = \delta_C(y) \]

- For $\varphi(x) = \frac{1}{2}x^2$: **squared k-support norm**
  \[ f_{\varphi}^*(y) = \frac{1}{2} \min_{z \in C} \sum_{i=1}^n \frac{y_i^2}{z_i} \]

- For general $\varphi$: **minimum distance to** $C$
  \[ f_{\varphi}^*(y) = \min_{z \in C} D_{\varphi}^*(y, z) \]
  where we defined the **$f$-divergence**
  \[ D_f(y, z) := \sum_{i=1}^n z_i f(y_i/z_i) \]
Computing the regularized version

• Primal solution

\[ u^* = \arg\min_{u \in \mathbb{R}^n} R^*(x - u) + f_\phi(u) \]

• Reduction to isotonic optimization

\[ u^*_\sigma = \arg\min_{v_1 \geq \cdots \geq v_n \geq 0} R^*(x_\sigma - v) + f_\phi(v) \]

\[ = \arg\min_{v_1 \geq \cdots \geq v_n \geq 0} R^*(x_\sigma - v) + f(\phi(v)) \]

\[ = \arg\min_{v_1 \geq \cdots \geq v_n \geq 0} R^*(x_\sigma - v) + \langle \phi(v), 1_k \rangle \]

where \( \sigma = \text{argsort}(|x|) \) and assuming \( \phi(x) = \phi(-x) \)
Nonconvex viewpoint: connection with the $\ell_0$ pseudo-norm

- We have
  
  \[ f_\varphi(x) = \max_{y \in S_k} \langle x, y \rangle - \sum_{i=1}^{n} \phi(y_i) \]

  where
  
  \[ \varphi(x) = (\phi(x_1), \ldots, \phi(x_n)) \quad \text{and} \quad S_k := \{ y \in \mathbb{R}^n : \|y\|_0 \leq k \} \]

- $f_\varphi^*(y)$ is the convex envelope of
  
  \[ y \mapsto \sum_{i=1}^{n} \phi^*(y_i) + \delta_{S_k}(y) \]

  With $\phi(x) = \frac{1}{2}x^2$, $f_\varphi^*(y)$ is the squared $k$-support norm
Applications
Weight pruning (multilayer perceptron, MNIST)

\[ W_i \leftarrow \text{topkmag}(W_i), \ i \in \{2, 3\} \]

\[ W_3 \sigma(W_2 \sigma(W_1 a + b_1) + b_2) + b_3 \]
We now demonstrate the applicability of our top-k operator to a linear classification head of width $3 \times 32 \times 784$, where $W$ and $b$ are set to 0 by magnitude pruning, using the framework proposed in Petersen et al. 2020a. We set $\theta$ and learn $\beta$.

For each $\left(\theta_i, y\right)$, we use the corresponding pretrained ViT model for a test error of $1/3$.

We find that the ViT finetuned with the smooth loss instead of the cross-entropy loss. We use a smooth top-k loss, whereas the performance of the network is evaluated to a test error of $1/3$.
Sparse Mixture of Vision Transformers

![Graph showing precision-at-1 vs training steps for differentiable top-k vs hard top-k]

JFT-300M dataset (305 million images)
Sparsity-constrained OT

\[
\min_{T \in \mathcal{U}(a,b)} \langle T, C \rangle + \sum_{j=1}^{n} f_{\varphi}(t_j)
\]

![Diagram showing OT formulations comparison](image_url)

**Unregularized**

**Entropy regularized**

**Squared 2-norm regularized**

**Sparsity constrained (ours)**

- For **entropy regularized OT**, the plans are always fully dense, meaning that all points are fractionally matched with one another (nonzeros of a transportation plan are indicated by small squares).
- With **entropy-regularized OT**, plans are on average sparser than with unregularized OT in the limit case.
- In the limit case, it can be thought as a middle ground between unregularized OT (recovered when \(k\) is large enough) and with uniform source and target distributions.
- For **sparsity-constrained OT**, we show that the dual and semi-dual formulations are tractable and that smoothness of the objective increases as \(k\) increases.
- Despite the nonconvexity of cardinality constraints, we show that the corresponding dual and semi-dual formulations are tractable and can be solved with first-order gradient methods. Our method can be thought of as a framework for OT with nonconvex regularization, based on the dual and semi-dual formulations.
Thank you!