Stochastic smoothing of the top-K calibrated hinge loss for deep imbalanced classification

Camille Garcin[1,2]

Maximilien Servajean[3,4]
Alexis Joly[2]
Joseph Salmon[1,5]

[1] IMAG, Univ Montpellier, CNRS
[2] Inria, LIRMM, Univ Montpellier, CNRS
[3] LIRMM, Univ Montpellier, CNRS
[4] AMIS, Univ Paul Valery
[5] Institut Universitaire de France (IUF)

Inter-class ambiguity

different labels but similar images

Cirsium rivulare

Chaerophyllum aromaticum

Conostomium kenyense

Adenostyles leucophylla

Sedum montanum

Cirsium tuberosum

Chaerophyllum temulum

Conostomium quadrangulare

Adenostyles alliariae

Sedum rupestre

Some species are visually similar

Necessary to return several classes

Motivation of Top-K

From a single to many predicted labels

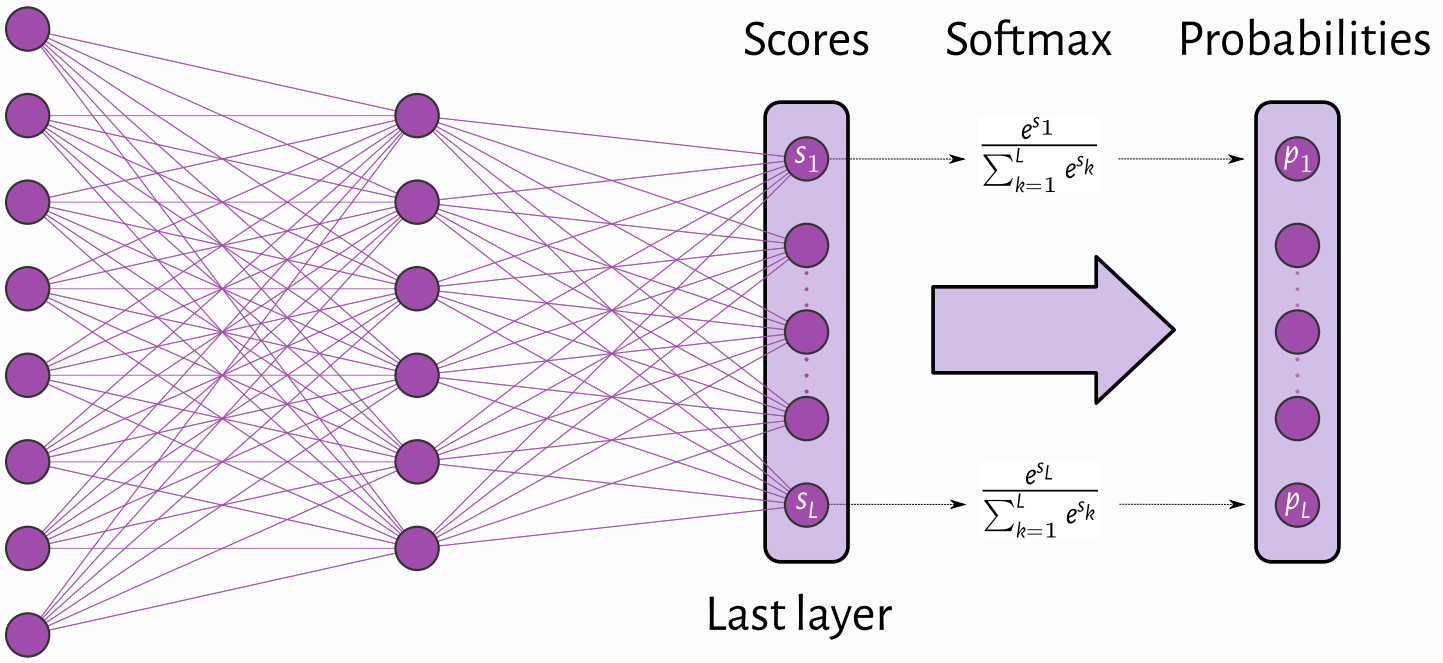
Possible solution: return the K "most likely" classes for all images

- Pros for a small K:
  model output must remain informative

- Pros for a large K:
  ensure the true class lies in the K returned classes

Choice of K :

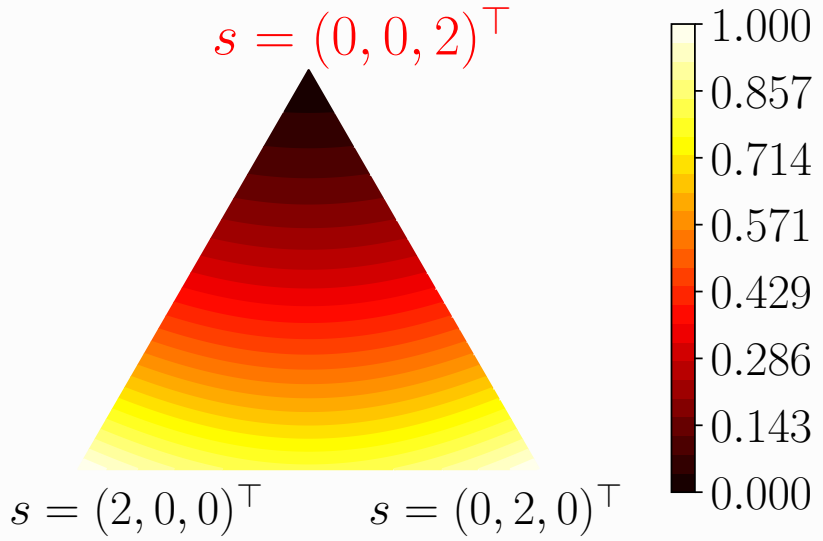
- task-dependant, often $K = 3, 5, \ldots$ or even larger for challenging tasks
- considered fixed by the user for the talk (not tuned)



- L: number of **classes**, $[L] := \{1, \ldots, L\}$, label space
- $K \in [L]$ is a fixed parameter used for top-K
- From an image, get a score vector $\mathbf{s} = (s_1, \ldots, s_L)^\top \in \mathbb{R}^L$ (aka logits)
- s_k : score for class k
- Prediction: output the K classes with the K highest scores

- Training: cross-entropy (CE) loss + Stochastic Gradient Descent (SGD)
- $\ell_{\mathrm{CE}}(\mathbf{s}, y) = -\log\left(\frac{e^{s_y}}{\sum_{k\in[L]} e^{s_k}}\right)$

Example : $L = 3, K = 2, y = 3$
(Normalized) level set of $\mathbf{s} \mapsto \ell_{\mathrm{CE}}(\mathbf{s}, y)$:



- Not designed to optimize top-K accuracy
- Can we do better than cross entropy ?

Notation and properties[1] for top-K

For a score vector $\mathbf{s} \in \mathbb{R}^L$:

Definition

$$\mathrm{top}_K : \mathbf{s} \mapsto s_{(K)} \qquad \text{(K-th largest score)}$$

Properties

- $\nabla \mathrm{top}_K(\mathbf{s}) = \arg \mathrm{top}_K(\mathbf{s}) \in \mathbb{R}^L$:
  vector with a single 1 at the K-th largest coordinate of $\mathbf{s}$, 0 o.w.

$$\text{For } \mathbf{s} = \begin{bmatrix} 4.0 \\ -1.5 \\ 2.5 \\ 1.0 \end{bmatrix}, \quad \mathrm{top}_2(\mathbf{s}) = 2.5 \quad \nabla \mathrm{top}_2(\mathbf{s}) := \arg \mathrm{top}_2(\mathbf{s}) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

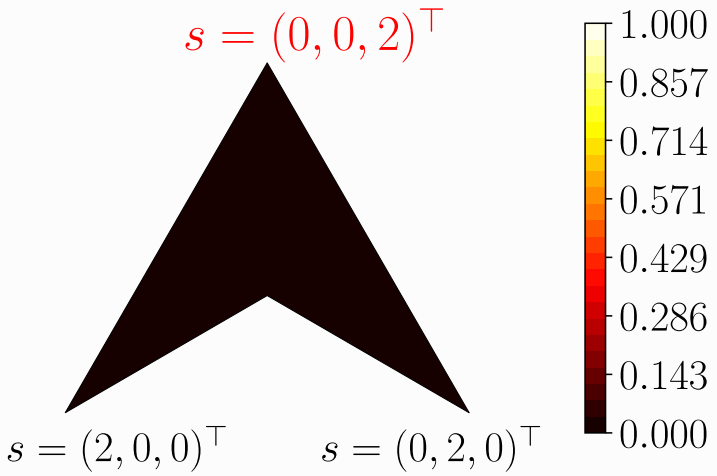
[1] F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". In: *ICML*. vol. 119, pp. 10727–10735.

Top-K error

Objective: minimize top-K error (0/1 loss):

$$\ell^K(\mathbf{s}, y) = \mathbb{1}_{\{\text{top}_K(\mathbf{s}) > s_y\}}$$

Problem: piecewise constant function w.r.t. $\mathbf{s}$, hard to optimize!!!

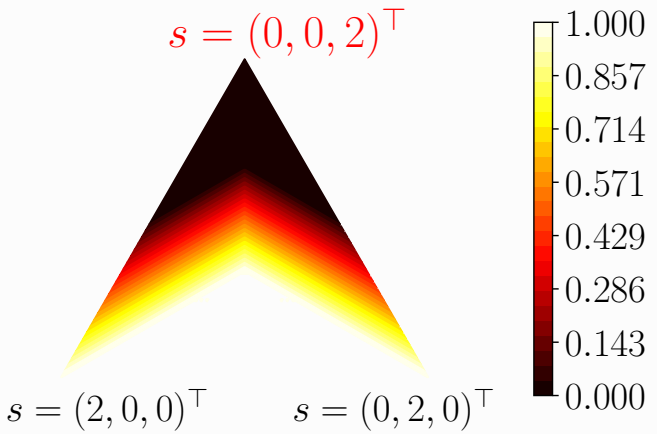


(Normalized) Level sets of $\mathbf{s} \mapsto \ell^K(\mathbf{s}, y)$, $L = 3$, $K = 2$, $y = 3$.

Top-K calibrated hinge loss[2]

A top-K hinge-loss that is top-K calibrated:

$$\ell^{K}_{\text{Cal. Hinge}}(\mathbf{s}, y) = (1 + \text{top}_{K+1}(\mathbf{s}) - s_y)_+$$



Better theoretical properties, but still fails with deep learning (more later)

Problem: $\mathbf{s} \to \text{top}_K(\mathbf{s})$ non-smooth and sparse gradient

[2] F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". In: *ICML*. vol. 119, pp. 10727–10735.

Smooth top-K [3]
Definition

Motivation: top_K is a non-smooth, function, smooth it!

- smoothing parameter $\epsilon > 0$
- score $\mathbf{s} \in \mathbb{R}^L$

Definition

The ϵ-smoothed version of top_K:

$$\mathrm{top}_{K,\epsilon}(\mathbf{s}) \triangleq \mathbb{E}_Z[\mathrm{top}_K(\mathbf{s} + \epsilon Z)]$$

Z : standard normal random vector, $Z \sim \mathcal{N}(0, \mathrm{Id}_L)$

[3] Q. Berthet et al. (2020). "Learning with differentiable perturbed optimizers". In: *NeurIPS*.

Smooth top-K
Proposition

Proposition

For a smoothing parameter $\epsilon > 0$,

- $\mathrm{top}_{K,\epsilon}$ is $\frac{4\sqrt{KL}}{\epsilon}$-smooth.
- For any $\mathbf{s} \in \mathbb{R}^L$, $|\mathrm{top}_{K,\epsilon}(\mathbf{s}) - \mathrm{top}_K(\mathbf{s})| \leq \epsilon \cdot C_{K,L}$, where $C_{K,L} = K\sqrt{2 \log L}$.
- The gradient of $\mathrm{top}_{K,\epsilon}$ reads:

$$\nabla_{\mathbf{s}} \mathrm{top}_{K,\epsilon}(\mathbf{s}) = \mathbb{E}[\arg \mathrm{top}_K(\mathbf{s} + \epsilon Z)]$$

- From non-smooth to smooth function with simple stochastic perturbation
- When $\epsilon \to 0$, recover the original function

Practical implementation: forward pass

Solution: Draw B noise vectors $Z_1, \dots, Z_B$, with $Z_b \overset{i.i.d.}{\sim} \mathcal{N}(0, \mathrm{Id}_L)$ for $b \in [B]$.

$$\mathrm{top}_{K,\epsilon}(\mathbf{s}) = \mathbb{E}_Z[\mathrm{top}_K(\mathbf{s} + \epsilon Z)]$$

Monte Carlo estimation :

$$\widehat{\mathrm{top}}_{K,\epsilon,B}(\mathbf{s}) = \frac{1}{B}\sum_{b=1}^{B} \mathrm{top}_K(\mathbf{s} + \epsilon Z_b)$$

Easy implementation with deep learning libraries *e.g.*, Pytorch, Tensorflow

Practical implementation: backward pass

Solution: Draw B noise vectors $Z_1, \dots, Z_B$, with $Z_b \overset{i.i.d.}{\sim} \mathcal{N}(0, \mathrm{Id}_L)$ for $b \in [B]$.

$$\nabla_{\mathbf{s}} \mathrm{top}_{K,\epsilon}(\mathbf{s}) = \mathbb{E}[\arg \mathrm{top}_K(\mathbf{s} + \epsilon Z)]$$

Monte Carlo estimation :

$$\widehat{\nabla \mathrm{top}}_{K,\epsilon,B}(\mathbf{s}) = \frac{1}{B} \sum_{b=1}^{B} \arg \mathrm{top}_K(\mathbf{s} + \epsilon Z_b)$$

Pytorch implementation available at:

https://github.com/garcinc/noised-topk

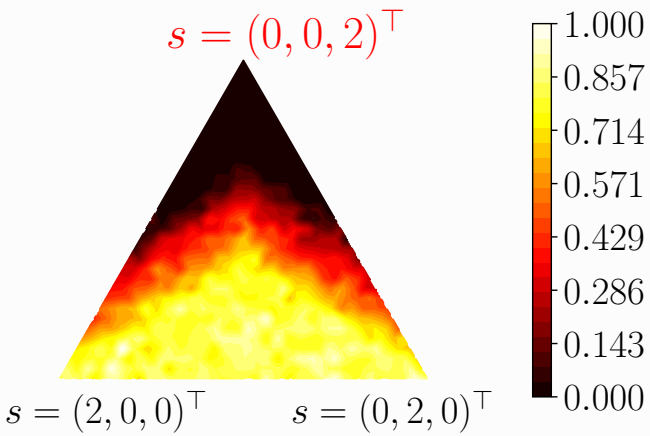
Imbalanced top-K loss

Modification: use larger margins for classes with few examples[4]:

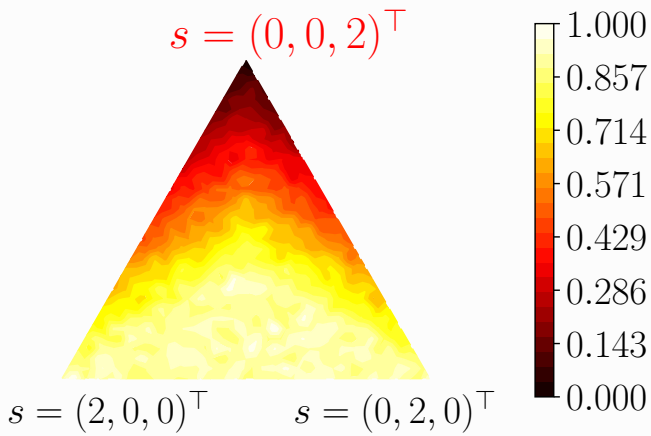
$$\ell^{K,\epsilon,B,m_y}_{\text{Noised Imbal.}}(\mathbf{s}, y) = (m_y + \widehat{\text{top}}_{K+1,\epsilon,B}(\mathbf{s}) - s_y)_+ \tag{1}$$

Set $m_y = C/n_y^{1/4}$, with n_y the number of samples in the training set with class y, and C a hyperparameter to be tuned on a validation set.

Intuition: add more emphasis on rarely seen examples



(a) $\ell^{K,0.3,30}_{\text{Noised bal.}}$.



(b) $\ell^{K,1,30}_{\text{Noised bal.}}$.

[4] K. Cao et al. (2019). "Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss". In: *NeurIPS*. vol. 32, pp. 1565–1576.

Comparison of several losses

K	ℓ_{CE}	$\ell^{K,0.1}_{\text{Smoothed Hinge}}$	$\ell^{K,1.0,5}_{\text{Noised bal.}}$	focal ($\gamma = 2.0$)	$\ell^{\max m_y=0.2}_{\mathrm{LDAM}}$	$\ell^{K,0.01,5,\max m_y=0.2}_{\text{Noised imbal.}}$
1	36.3±0.3	35.7±0.2	35.8±0.3	37.6±0.3	40.6±0.1	**42.4**±0.3
3	58.8±0.4	50.3±0.2	58.7±0.4	60.4±0.3	63.3±0.3	**64.9**±0.4
5	68.7±0.2	50.9±0.3	66.4±0.5	69.7±0.2	71.9±0.3	**73.2**±0.5

Macro-average test top-K accuracy on Pl@ntNet-300K, ResNet-50.

- Imbalanced losses fare better than balanced losses
- Class-wise margin is effective compared to constant margin
- $\ell^{K,\epsilon,B,m_y}_{\text{Noised imbal.}}$ outperforms other losses on Pl@ntNet-300K
- Many more experiments in the paper !

Thank you for your attention !

Contact: camille.garcin@umontpellier.fr

Bibliographie

- Berthet, Q. et al. (2020). “Learning with differentiable perturbed optimizers”. In: *NeurIPS*.
- Cao, K. et al. (2019). “Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss”. In: *NeurIPS*. Vol. 32, pp. 1565–1576.
- Yang, F. and S. Koyejo (2020). “On the consistency of top-k surrogate losses”. In: *ICML*. Vol. 119, pp. 10727–10735.