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

Joseph Salmon
IMAG, Univ Montpellier, CNRS
Institut Universitaire de France (IUF)

Mainly joint work with:

Camille Garcin (Univ. Montpellier, IMAG)
Maximilien Servajean (Univ. Paul-Valéry-Montpellier, LIRMM, Univ. Montpellier)
Alexis Joly (Inria, LIRMM, Univ. Montpellier)

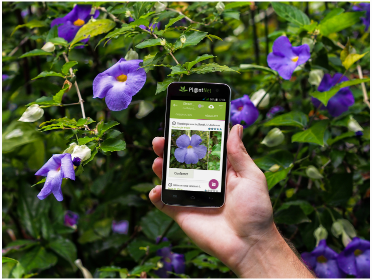
and:

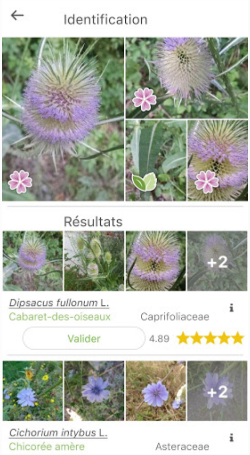
Pierre Bonnet (CIRAD, AMAP)
Antoine Affouard, **J-C. Lombardo**, **Titouan Lorieul**, **Mathias Chouet** (Inria, LIRMM, Univ. Montpellier)

- C. Garcin, A. Joly, et al. (2021). "Pl@ntNet-300K: a plant image dataset with high label ambiguity and a long-tailed distribution". In: *NeurIPS Datasets and Benchmarks 2021*
- C. Garcin, M. Servajean, et al. (2022). "Stochastic smoothing of the top-K calibrated hinge loss for deep imbalanced classification". In: *ICML*

Plant classification with Pl@ntNet

https://plantnet.org/

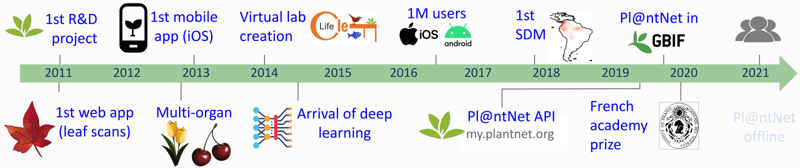




- ML assisted citizen science
- **> 40,000 species**
- **> 10,000,000 annotated images**
- **> 1Tb** of data $\Longrightarrow$ Reduction to share with community

PL@NTNET HISTORY

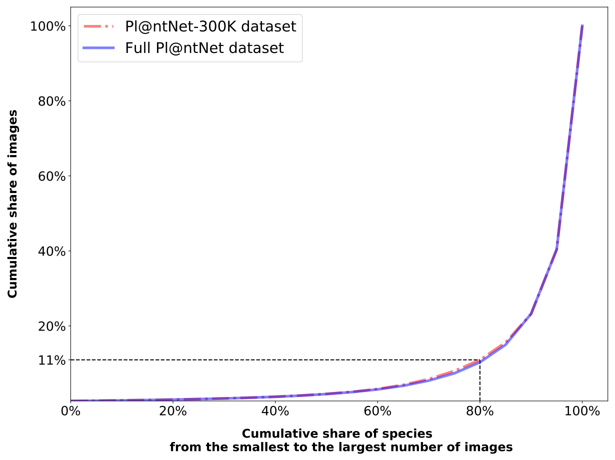
Pl@ntNet Key milestones



inria
cirad AGRICULTURAL RESEARCH FOR DEVELOPMENT
IRD Institut de Recherche pour le Développement FRANCE
INRAE
agropolis fondation Supporting agricultural research for sustainable development

Table of Contents

Long tailed distribution
Preserved with sampling of genera



80% of species account for only 11% of images

Intra-class variability

same label/species but very diverse images

Guizotia abyssinica

Diascia rigescens

Lapageria rosea

Casuarina cunninghamiana

Freesia alba

Plant species are challenging to model based on pictures only!

Inter-class ambiguity

different labels/species but similar images

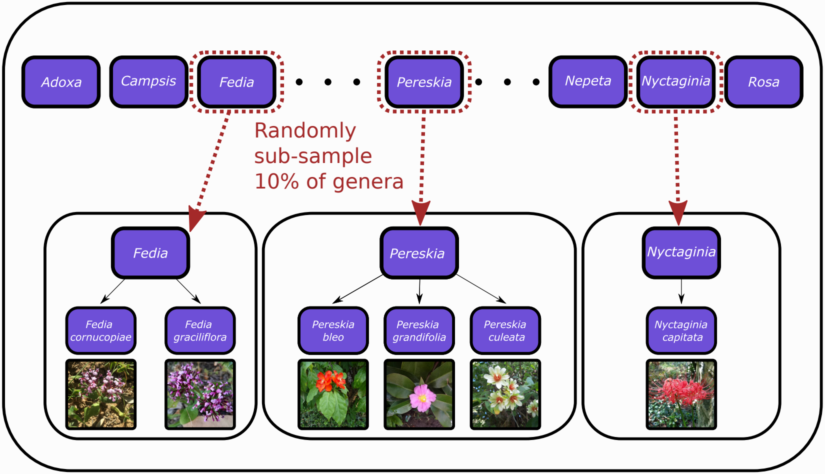
Cirsium rivulare | *Chaerophyllum aromaticum* | *Conostomium kenyense* | *Adenostyles leucophylla* | *Sedum montanum*

Cirsium tuberosum | *Chaerophyllum temulum* | *Conostomium quadrangulare* | *Adenostyles alliariae* | *Sedum rupestre*

Some species are visually similar (especially within genus)

Table of Contents

Construction of Pl@ntNet-300K
Subsampling of genera



Sample at genus level to preserve intra-genus ambiguity

Links

Zenodo, 1 click download

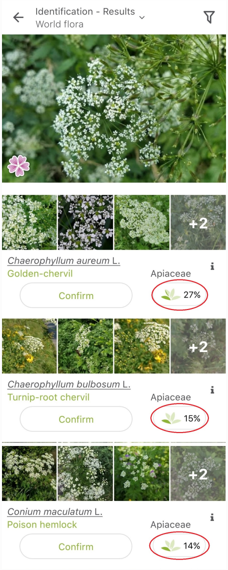
https://zenodo.org/record/5645731

Code to train models:

https://github.com/plantnet/PlantNet-300K

Table of Contents

Limitation of a single proposition



With high class ambiguity, returning a single class is hazardous

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

- Pros for a small K:
  ease user experience, handle screen size constraints (mobiles)

  Pl@ntNet returns **species names** + **most similar images** to the query: narrows down the ambiguity

- 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)

Table of Contents

Notation: multi-class setting

- L: number of **classes**, $[L] := \{1, \ldots, L\}$, label space
  Pl@ntNet-300K: $L = \mathbf{1\,081}$ species
- $\mathcal{X}$: Feature space
  Pl@ntNet-300K: $\mathcal{X} = \mathbb{R}^{256 \times 256 \times 3}$
- $(X_i, Y_i) \in \mathcal{X} \times [L], i = 1, \ldots, n$ *i.i.d.* according to $\mathbb{P}$ (unknown)
  Pl@ntNet-300K: **306 146** images
- $K \in [L]$ is a fixed parameter used for top-K
- **Set-valued classifier**
  $\Gamma : \mathcal{X} \to 2^{[L]}$; $2^{[L]}$: set of all subsets of $[L]$

Mathematical goal:
minimize the risk $\mathbb{P}(Y \notin \Gamma(X))$ with cardinality constraints on $\Gamma(X)$

Notation:

- $p_\ell(x) \triangleq \mathbb{P}(Y = \ell | X = x)$: conditional label probability given an input x
- Decreasing ordering : $p_{(1)}(x) \geq \cdots \geq p_{(L)}(x)$,
  i.e., (1) is the most likely class for x, (2) the second most likely class, etc.

  Below we also use: $p_{(1)}(x) = p_{i_1(x)}(x), \ldots, p_{(L)}(x) = p_{i_L(x)}(x)$

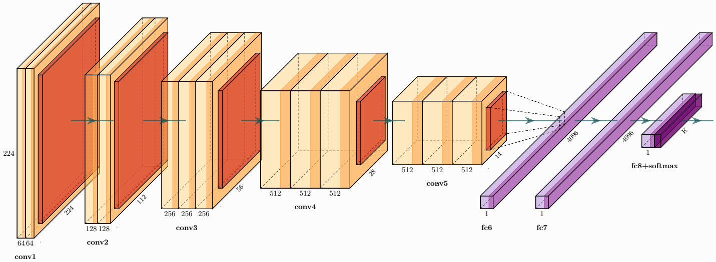
- **Top-K classification**:

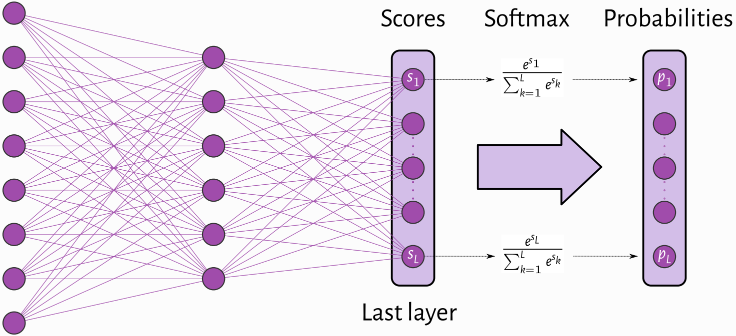
$$\Gamma^*_{\text{top-}K} \in \arg\min_{\Gamma} \mathbb{P}(Y \notin \Gamma(X)) \quad \text{s.t. } |\Gamma(x)| \leq K, \forall x \in \mathcal{X} \qquad \Longrightarrow \qquad \Gamma^*_{\text{top-}K}(x) = \{i_1(x), \ldots, i_K(x)\}$$

Interpretation:
the optimal top-K classifier returns the K most likely classes

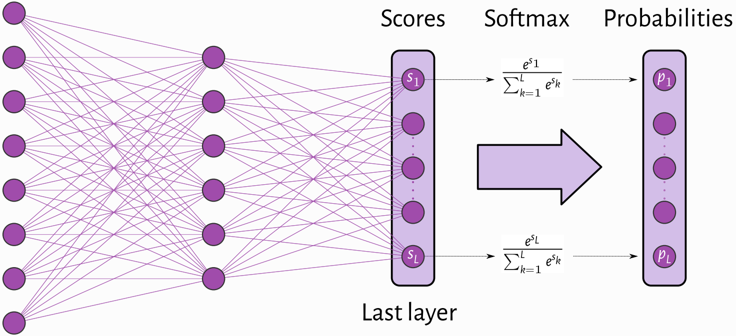
[1] M. Lapin, M. Hein, and B. Schiele (2015). "Top-k multiclass SVM". In: *NeurIPS*, pp. 325–333.

Deep learning
Notation mostly





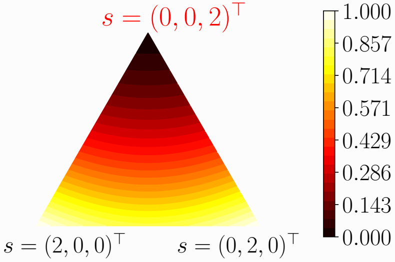
Scores
Softmax
Probabilities
s_1
s_L
$\frac{e^{s_1}}{\sum_{k=1}^{L} e^{s_k}}$
$\frac{e^{s_L}}{\sum_{k=1}^{L} e^{s_k}}$
p_1
p_L
Last layer



- 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
- Reordered scores: $s_{(1)} \geq s_{(2)} \geq \cdots \geq s_{(L)}$
- (Top-1) prediction: output the "most likely" class, associated to $s_{(1)}$ or $p_{(1)}$

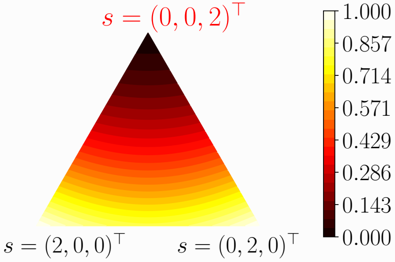
- Training: cross-entropy (CE) loss + Stochastic Gradient Descent (SGD)
- $\ell_{\mathrm{CE}}(\mathbf{s}, y) = -\log\left(\frac{e^{s_y}}{\sum_{k\in[L]} e^{s_k}}\right)$

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



- 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 ?

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

Definition

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

$$\mathrm{top}\Sigma_K : \mathbf{s} \mapsto \sum_{k \in [K]} s_{(k)} \qquad \text{(sum of K largest scores)}$$

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

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

Example on the following score vector: $$\mathbf{s} = \begin{bmatrix} 4.0 \\ -1.5 \\ 2.5 \\ 1.0 \end{bmatrix}$$

We have

$$\operatorname{top}_2(\mathbf{s}) = 2.5 \qquad \nabla \operatorname{top}_2(\mathbf{s}) := \arg\operatorname{top}_2(\mathbf{s}) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Illustration of top-K notation

Example on the following score vector: $\mathbf{s} = \begin{bmatrix} 4.0 \\ -1.5 \\ 2.5 \\ 1.0 \end{bmatrix}$

We have

$$\operatorname{top}_2(\mathbf{s}) = 2.5 \qquad \nabla \operatorname{top}_2(\mathbf{s}) := \arg\operatorname{top}_2(\mathbf{s}) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

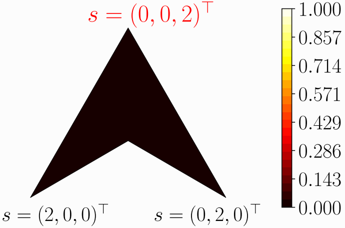
$$\operatorname{top}\Sigma_2(\mathbf{s}) = 4.0 + 2.5 = 6.5 \qquad \nabla \operatorname{top}\Sigma_2(\mathbf{s}) := \arg\operatorname{top}\Sigma_2(\mathbf{s}) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Table of Contents

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

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

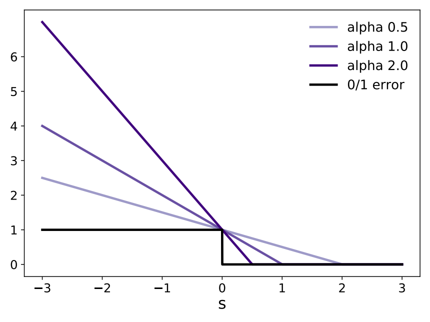
Problem: piecewise constant function w.r.t. $\mathbf{s}$, hard to optimize!!!



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

- Binary case ($L = 2$): $y = 1, y = -1$
- Score s: predict $y = 1$ if $s > 0$, $y = -1$ otherwise

Objective: Minimize binary 0/1 error $\ell^{0/1}(s, y) = \mathbb{1}[sy < 0]$.
Upper bound of $\ell^{0/1}$: $\ell^{\text{Hinge}}(s, y) = \alpha \max(0, 1 - \frac{1}{\alpha} sy) = \alpha(1 - \frac{1}{\alpha} sy)_+$



Larger margins ($\frac{1}{\alpha}$) require more confident predictions to achieve a zero loss

Top-K hinge loss[3]

Motivation: surrogate top-K loss, similar to hinge loss in binary classification

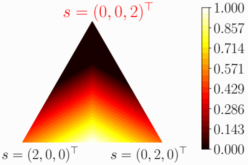
$$\ell^K_{\mathrm{Hinge}}(\mathbf{s}, y) = \left(1 + \mathrm{top}_K(\mathbf{s}_{\setminus y}) - s_y\right)_+$$

where $\mathbf{s}_{\setminus y}$ is the vector $\mathbf{s}$ with coordinate y removed

Remark: 1 acts as a *margin* above

Limitations:

- Experimental: poor performance
- Theoretical: ℓ^K_{Hinge} is not top-K calibrated (more later)



[3] M. Lapin, M. Hein, and B. Schiele (2015). "Top-k multiclass SVM". In: *NeurIPS*, pp. 325–333.

Table of Contents

Question:
When minimizing a surrogate loss ℓ implies minimizing the top-K error ℓ^K ?

Answer: Yes, if ℓ is top-K **calibrated**
i.e., if the Bayes risk can only be attained by a score sharing the same top-K as the underlying conditional probability distribution)

Integrated ℓ-Risk for classifier f $\quad \mathcal{R}_\ell(f) \triangleq \mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(f(x),y)]$

Integrated Bayes Risk $\quad \mathcal{R}_\ell^* \triangleq \inf_{f:\mathcal{X}\to\mathbb{R}^L} \mathcal{R}_\ell(f)$

Top-K consistency

Theorem [4]

$$\ell \text{ is top-}K\text{ calibrated} \implies \ell \text{ is top-}K\text{ consistent:}$$

i.e., for any sequence of measurable functions $f^{(n)} : \mathcal{X} \to \mathbb{R}^L$, we have:

$$\mathcal{R}_\ell\left(f^{(n)}\right) \to \mathcal{R}_\ell^* \implies \mathcal{R}_{\ell^K}\left(f^{(n)}\right) \to \mathcal{R}_{\ell^K}^*$$

where ℓ^K is the (0/1) top-K loss

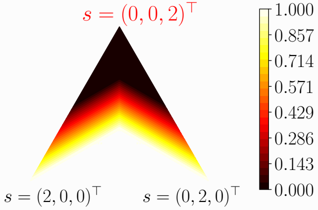
Interpretation:
Minimizing a top-K calibrated loss implies minimizing the top-K error

Note: ℓ_{CE} is top-K calibrated, but not when restricted to **linear classifiers** (for $d \leq 3$, $L \leq 3$, $K \leq 2$).

[4] F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". In: *ICML*. vol. 119, pp. 10727–10735, Theorem 2.2.

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

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



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

<u>Problem</u>: $\mathbf{s} \to \mathrm{top}_K(\mathbf{s})$ non-smooth and sparse gradient

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

Table of Contents

<u>Motivation</u>: $\mathrm{top}\Sigma_K$ is a non-smooth, function, smooth it!

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

Definition

The ϵ-smoothed version of $\mathrm{top}\Sigma_K$:

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

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

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

Proposition

For a smoothing parameter $\epsilon > 0$,

- The function $\mathrm{top}\Sigma_{K,\epsilon} : \mathbb{R}^L \to \mathbb{R}$ is strictly convex, twice differentiable and $\sqrt{K}$-Lipschitz continuous.
- The gradient of $\mathrm{top}\Sigma_{K,\epsilon}$ reads:
$$\nabla_{\mathbf{s}} \mathrm{top}\Sigma_{K,\epsilon}(\mathbf{s}) = \mathbb{E}[\arg \mathrm{top}\Sigma_K(\mathbf{s} + \epsilon Z)]$$
- $\nabla_{\mathbf{s}} \mathrm{top}\Sigma_{K,\epsilon}$ is $\frac{\sqrt{KL}}{\epsilon}$-Lipschitz.
- When $\epsilon \to 0$, $\mathrm{top}\Sigma_{K,\epsilon}(\mathbf{s}) \to \mathrm{top}\Sigma_K(\mathbf{s})$.

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

Reminder: $\mathrm{top}_K(\mathbf{s}) \triangleq \mathrm{top}\Sigma_K(\mathbf{s}) - \mathrm{top}\Sigma_{K-1}(\mathbf{s})$

Definition

For any $s \in \mathbb{R}^L$ and $K \in [L]$, the smoothed top-K at level ϵ is:

$$\mathrm{top}_{K,\epsilon}(\mathbf{s}) \triangleq \mathrm{top}\Sigma_{K,\epsilon}(\mathbf{s}) - \mathrm{top}\Sigma_{K-1,\epsilon}(\mathbf{s})$$

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 $C_{K,L} = K\sqrt{2 \log L}$.

- Smooth approximation of top_K.
- Smoothness constant depending on ϵ and problem constants.
- When $\epsilon \to 0$, recover initial top-K

Table of Contents

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

Definition

We define $\ell^{K,\epsilon}_{\text{Noised bal.}}$ the noised balanced top-K hinge loss as:

$$\ell^{K,\epsilon}_{\text{Noised bal.}}(\mathbf{s}, y) = (1 + \mathrm{top}_{K+1,\epsilon}(\mathbf{s}) - s_y)_+$$

Problem: Untractable: how to deal with the expectation in $\mathrm{top}_{K+1,\epsilon}(\mathbf{s})$?

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 $b \in [B]$.

$$\begin{aligned} \mathrm{top}_{K,\epsilon}(\mathbf{s}) &= \mathrm{top}\Sigma_{K,\epsilon}(\mathbf{s}) - \mathrm{top}\Sigma_{K-1,\epsilon}(\mathbf{s}) \\ &= \mathbb{E}_Z[\mathrm{top}\Sigma_K(\mathbf{s} + \epsilon Z)] - \mathbb{E}_Z[\mathrm{top}\Sigma_{K-1}(\mathbf{s} + \epsilon Z)] \end{aligned}$$

Monte Carlo estimation :

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

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 $b \in [B]$.

$$\begin{aligned}\nabla_{\mathbf{s}} \mathrm{top}_{K,\epsilon}(\mathbf{s}) &= \nabla_{\mathbf{s}} \mathrm{top}\Sigma_{K,\epsilon}(\mathbf{s}) - \nabla_{\mathbf{s}} \mathrm{top}\Sigma_{K-1,\epsilon}(\mathbf{s}) \\ &= \mathbb{E}[\arg \mathrm{top}\Sigma_K(\mathbf{s} + \epsilon Z)] - \mathbb{E}[\arg \mathrm{top}\Sigma_{K-1}(\mathbf{s} + \epsilon Z)]\end{aligned}$$

Monte Carlo estimation :

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

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

$L = 4, K = 2, B = 3, \epsilon = 1.0, \mathbf{s} = \begin{bmatrix} \mathbf{2.4} \\ 2.6 \\ 2.3 \\ 0.5 \end{bmatrix}$. We have $\mathrm{top}_K(\mathbf{s}) = \mathbf{2.4}$ and $\arg\mathrm{top}_K(\mathbf{s}) = \begin{bmatrix} \mathbf{1} \\ 0 \\ 0 \\ 0 \end{bmatrix}$. Assume the three noise vectors sampled are:

$$Z_1 = \begin{bmatrix} 0.2 \\ -0.1 \\ 0.1 \\ 0.3 \end{bmatrix}, Z_2 = \begin{bmatrix} 0.1 \\ 0.1 \\ -0.1 \\ 0.1 \end{bmatrix}, Z_3 = \begin{bmatrix} -0.1 \\ -0.1 \\ 0.1 \\ -0.1 \end{bmatrix}.$$

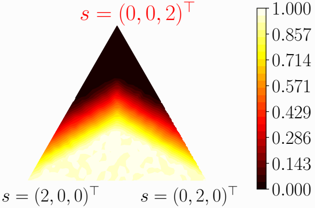
The perturbed vectors are now:

$$\mathbf{s} + \epsilon Z_1 = \begin{bmatrix} 2.6 \\ \mathbf{2.5} \\ 2.4 \\ 0.8 \end{bmatrix}, \mathbf{s} + \epsilon Z_2 = \begin{bmatrix} \mathbf{2.5} \\ 2.7 \\ 2.2 \\ 0.6 \end{bmatrix}, \mathbf{s} + \epsilon Z_3 = \begin{bmatrix} 2.3 \\ 2.5 \\ \mathbf{2.4} \\ 0.4 \end{bmatrix}.$$

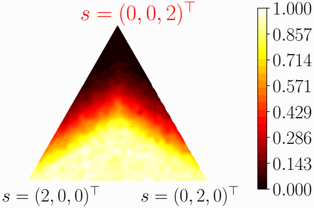
$$\widehat{\mathrm{top}}_{K,\epsilon,B}(s) = (\mathbf{2.5} + \mathbf{2.5} + \mathbf{2.4})/3 = 2.47 \quad ,$$

$$\widehat{\nabla \mathrm{top}}_{K,\epsilon,B}(s) = \frac{1}{3}\left(\begin{bmatrix} 0 \\ \mathbf{1} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{1} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \mathbf{1} \\ 0 \end{bmatrix} \right) = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ 0 \end{bmatrix}.$$

Noised top-K loss
Visualization



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



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

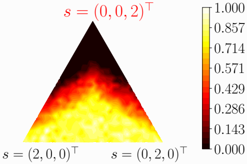
Table of Contents

Modification: use larger margins for classes with few examples[7]:

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

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



$\ell^{K,0.1,30,1}_{\text{Noised Imbal.}}$.

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

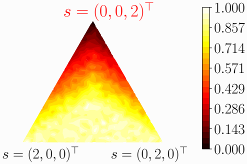
Imbalanced top-K loss

Modification: use larger margins for classes with few examples[7]:

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

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



$\ell^{K,0.1,30,5}_{\text{Noised Imbal.}}$.

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

Table of Contents

CIFAR100 DATASET

- 100 classes, 500 training images per class and 100 test images per class

Superclass	**Classes**
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchids, poppies, roses, sunflowers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

`https://www.cs.toronto.edu/~kriz/cifar.html`

Table of Contents

Influence of ϵ on top-K accuracy

ϵ	0.0	1e-4	1e-3	1e-2	1e-1	1.0	10.0	100.0
Top-5 acc.	19.38	14.84	11.4	93.36	94.46	94.24	93.78	93.12

CIFAR-100 best validation top-5 accuracy, DenseNet 40-40, $\ell^{K=5,\epsilon,B=10}_{\text{Noised bal.}}$.

- $\epsilon = 0$ recovers $\ell^{K}_{\mathrm{Cal.\ Hinge}}$: bad performance
- ϵ large enough, relevant coordinates are updated, learning occurs
- Optimization robust to large values of ϵ

Influence of B

B	1	2	3	5	10	50	100
Top-5 acc	94.28	94.2	94.46	94.52	94.24	94.64	94.52

- $\ell^{5,0.2,B}_{\text{Noised bal.}}$, CIFAR-100 dataset, DenseNet 40-40 model.
- B has little influence
- Using SGD increases the randomness (B noise vectors per batch)
- In practice set B to a small value *e.g.*, $B = 3$

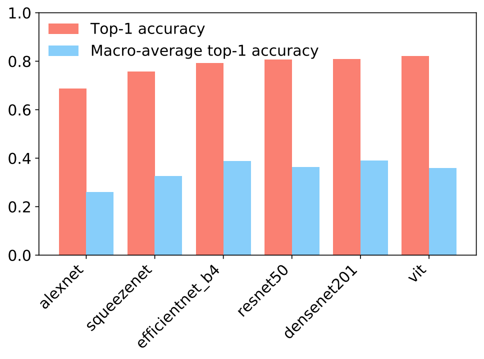
Table of Contents

- Test set of examples $S_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- $\Gamma_K : \mathcal{X} \to 2^{[K]}$ learned top-K classifier (model) to evaluate
- $\mathcal{C}_j$ set of examples of class j: $\mathcal{C}_j = \{\ell \in [L], y_\ell = j\}$

Top-K accuracy: $\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[y_i \in \Gamma_K(x_i)]$
Reflects the performance on classes with lots of examples

Macro-average Top-K accuracy: $\frac{1}{L} \sum_{j=1}^{L} \frac{1}{|\mathcal{C}_j|} \sum_{\ell \in \mathcal{C}_j} \mathbb{1}[y_\ell \in \Gamma_K(x_\ell)]$
Reflects the performance on all classes regardless of number of examples

Cross-entropy baseline
Accuracy vs Macro-average accuracy



Pl@ntNet-300K test performance for several neural networks: large gaps due to long-tailed distribution

Number of images	Mean bin accuracy
$0 - 10$	0.09
$10 - 50$	0.35
$50 - 500$	0.59
$500 - 2000$	0.79
> 2000	0.93

Test accuracy (ResNet50) w.r.t. number of images per class at training...

... (many) classes with few examples have low accuracy (hard to learn)

Comparison of several losses

K	ℓ_{CE}	$\ell^{K,\tau}_{\text{Smoothed Hinge}}$ (8)	$\ell^{K,\epsilon,B}_{\text{Noised bal.}}$	focal (9)	LDAM (10)	$\ell^{K,\epsilon,B,m_y}_{\text{Noised imbal.}}$
1	35.91	NA	35.44	37.87	40.54	**42.36**
3	58.91	50.41	59.06	59.96	63.50	**64.77**
5	69.05	50.71	66.97	69.91	72.23	**72.95**
10	78.08	46.23	76.08	78.88	80.69	**80.85**

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

- $\ell^{K,\tau}_{\text{Smoothed Hinge}}$ gives unsatisfactory for imbalanced datasets
- Imbalanced losses: far 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

(8) L. Berrada, A. Zisserman, and M. P. Kumar (2018). "Smooth Loss Functions for Deep Top-k Classification". In: *ICLR*.

(9) T.-Y. Lin et al. (2017). "Focal Loss for Dense Object Detection". In: *ICCV*, pp. 2999–3007.

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

Conclusion

- A new loss for top-K classification: smooth a top-K calibrated one
- Suitable for training deep learning models
- Significant performance gains on real databases such as Pl@ntNet (with high ambiguity & a long tail distribution)

Perpectives

- A fixed set size K is not ideal in practice
  - Some species are easy to recognize while others are ambiguous
  - Some images are very informative while others are not
- Set-valued classification with a varying set size could be more effective

Contact information

Contact:

joseph.salmon@umontpellier.fr
http://josephsalmon.eu

Github: @josephsalmon

Twitter: @salmonjsph

Bibliographie

- Berrada, L., A. Zisserman, and M. P. Kumar (2018). "Smooth Loss Functions for Deep Top-k Classification". In: *ICLR*.
- 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.
- Garcin, C., A. Joly, et al. (2021). "Pl@ntNet-300K: a plant image dataset with high label ambiguity and a long-tailed distribution". In: *NeurIPS Datasets and Benchmarks 2021*.
- Garcin, C., M. Servajean, et al. (2022). "Stochastic smoothing of the top-K calibrated hinge loss for deep imbalanced classification". In: *ICML*.
- Lapin, M., M. Hein, and B. Schiele (2015). "Top-k multiclass SVM". In: *NeurIPS*, pp. 325–333.
- Lin, T.-Y. et al. (2017). "Focal Loss for Dense Object Detection". In: *ICCV*, pp. 2999–3007.
- Yang, F. and S. Koyejo (2020). "On the consistency of top-k surrogate losses". In: *ICML*. Vol. 119, pp. 10727–10735.

CIFAR100: LABEL NOISE INJECTION

- Reminder: 20 superclasses each containing 5 classes
- Ex: Super class large carnivors contains the classes "bear", "leopard", "lion", "tiger", "wolf"

For each image in the training set:

- With probability p, randomly sample label <u>within</u> the superclass
- With probability $1 - p$, keep the label unchanged

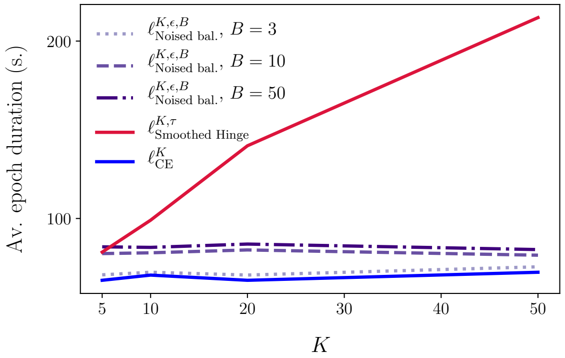
Possibly wrong class, but same superclass as original dataset.

CIFAR100 RESULTS

Label noise p	ℓ_{CE}	$\ell^{5,1.0}_{\text{Smoothed Hinge}}$	$\ell^{5,0.2,10}_{\text{Noised bal.}}$
0.0	94.24	94.34	**94.35**
0.1	90.39	**92.08**	92.03
0.2	87.67	90.22	**90.68**
0.3	85.93	88.82	**89.58**
0.4	83.74	87.40	**87.48**

- CIFAR-100 test Top-5 accuracy, DenseNet 40-40.
- When $p > 0$, ℓ_{CE} tries to fit corrupted labels while top-K losses merely strives to get the super-class right.
- $\ell^{K,\epsilon,B}_{\text{Noised bal.}}$ gives good performance and faster to train than $\ell^{K,\tau}_{\text{Smoothed Hinge}}$

Loss : $\ell(\mathbf{s}, y)$	**Expression**	**Param.**	**Reference**
$\ell^K(\mathbf{s}, y)$	$\mathbb{1}_{\{\mathrm{top}_K(\mathbf{s}) > s_y\}}$	K	
$\ell_{\mathrm{CE}}(\mathbf{s}, y)$	$-\ln\left(e^{s_y} / \sum_{k \in [L]} e^{s_k}\right)$	—	
$\ell^K_{\mathrm{Hinge}}(\mathbf{s}, y)$	$\left(1 + \mathrm{top}_K(\mathbf{s}_{\setminus y}) - s_y\right)_+$	K	(Lapin, Hein, and Schiele 2015)
$\ell^K_{\mathrm{CVXHinge}}(\mathbf{s}, y)$	$\left(\frac{1}{K} \sum_{k \in [K]} \mathrm{top}_k(\mathbf{1}_L - \delta_y + \mathbf{s}) - s_y\right)_+$	K	(Lapin, Hein, and Schiele 2015)
$\ell^K_{\mathrm{Cal.\ Hinge}}(\mathbf{s}, y)$	$(1 + \mathrm{top}_{K+1}(\mathbf{s}) - s_y)_+$	K	(Yang and Koyejo 2020)
$\ell^{K,\tau}_{\text{Smoothed Hinge}}(\mathbf{s}, y)$	$\tau \ln \left[\sum_{A \subset [L], \lvert A \rvert = K} e^{\frac{\mathbb{1}_{\{y \notin A\}}}{\tau} + \sum_{j \in A} \frac{s_j}{K\tau}} \right] - \tau \ln \left[\sum_{A \subset [L], \lvert A \rvert = K} e^{\sum_{j \in A} \frac{s_j}{K\tau}} \right]$	K, τ	(Berrada, Zisserman, and Kumar 2018)
$\ell^{K,\epsilon,B}_{\text{Noised bal.}}(\mathbf{s}, y)$	$(1 + \widehat{\mathrm{top}}_{K+1,\epsilon,B}(\mathbf{s}) - s_y)_+,$	K, ϵ, B	**proposed**
$\ell^{K,\epsilon,B,m_y}_{\text{Noised Imbal.}}(\mathbf{s}, y)$	$(m_y + \widehat{\mathrm{top}}_{K+1,\epsilon,B}(\mathbf{s}) - s_y)_+,$	K, ϵ, B, m_y	**proposed**



- CIFAR-100 dataset, DenseNet 40-40 model
- $\ell^{K,\epsilon,B}_{\text{Noised bal.}}$ insensitive to K unlike $\ell^{K,\tau}_{\text{Smoothed Hinge}}$

Proposition

For a smoothing parameter $\epsilon > 0$ and a label $y \in [L]$:

- $\ell^{K,\epsilon}_{\text{Noised bal.}}(\cdot, y)$ is continuous and differentiable almost everywhere
- The gradient of $\ell(\cdot, y) \triangleq \ell^{K,\epsilon}_{\text{Noised bal.}}(\cdot, y)$ is given by:

$$\nabla \ell(\mathbf{s}, y) = \mathbb{1}_{\{1+\mathrm{top}_{K+1,\epsilon}(\mathbf{s}) \geq s_y\}} \cdot (\nabla \mathrm{top}_{K+1,\epsilon}(\mathbf{s}) - \delta_y),$$

where $\delta_y \in \mathbb{R}^L$ is the vector with 1 at coordinate y and 0 elsewhere.

$\Delta_L \triangleq \{\boldsymbol{\pi} \in \mathbb{R}^L : \sum_{k\in[L]} \pi_k = 1, \pi_k \geq 0\}$: probability simplex of size L

Risks

- Conditional risk: for $x \in \mathcal{X}$, $\boldsymbol{\pi} \in \Delta_L$, $\mathcal{R}_{\ell|x}(\mathbf{s}, \boldsymbol{\pi}) = \mathbb{E}_{y|x\sim\pi}(\ell(\mathbf{s}, y))$
- Integrated risk for a scoring function f: $\mathcal{R}_\ell(f) \triangleq \mathbb{E}_{(x,y)\sim\mathbb{P}}[\ell(f(x), y)]$

Bayes risks :

$$\mathcal{R}^*_{\ell|x}(\boldsymbol{\pi}) \triangleq \inf_{\mathbf{s}\in\mathbb{R}^L} \mathcal{R}_{\ell|x}(\mathbf{s}, \boldsymbol{\pi})$$

$$\mathcal{R}^*_\ell \triangleq \inf_{f:\mathcal{X}\to\mathbb{R}^L} \mathcal{R}_\ell(f)$$

Definition [11]

For a fixed $K \in [L]$, and given $\mathbf{s} \in \mathbb{R}^L$ and $\tilde{\mathbf{s}} \in \mathbb{R}^L$, we say that $\mathbf{s}$ is top-K preserving w.r.t. $\tilde{\mathbf{s}}$, denoted $P_K(\mathbf{s}, \tilde{\mathbf{s}})$, if for all $k \in [L]$,

$$\tilde{s}_k > \mathrm{top}_{K+1}(\tilde{\mathbf{s}}) \implies s_k > \mathrm{top}_{K+1}(\mathbf{s})$$
$$\tilde{s}_k < \mathrm{top}_{K}(\tilde{\mathbf{s}}) \implies s_k < \mathrm{top}_{K}(\mathbf{s})$$

The negation of this statement is $\neg P_k(\mathbf{s}, \tilde{\mathbf{s}})$.

Roughly speaking: the top-K coordinates of the two vectors are the same

[11] F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". In: *ICML*. vol. 119, pp. 10727–10735, Definition 2.3.

Example:

- Consider the vectors $\mathbf{s} = \begin{bmatrix} 4.0 \\ -1.5 \\ 2.5 \\ 1.0 \end{bmatrix}$ and $\tilde{\mathbf{s}}_1 = \begin{bmatrix} 5.0 \\ 1.0 \\ 6.0 \\ 3.0 \end{bmatrix}$.

  $\mathbf{s}$ is top-2 preserving with respect to $\tilde{\mathbf{s}}_1$ because it preserves its top-2 components (the first and third components).

- Consider the vectors $\mathbf{s} = \begin{bmatrix} 4.0 \\ -1.5 \\ 2.5 \\ 1.0 \end{bmatrix}$ and $\tilde{\mathbf{s}}_2 = \begin{bmatrix} 5.0 \\ 5.5 \\ -1.0 \\ 3.0 \end{bmatrix}$.

  $\mathbf{s}$ is not top-2 preserving with respect to $\tilde{\mathbf{s}}_2$ because it changes its top-2 components.

Top-K calibrated loss

Definition[12]

A loss $\ell : \mathbb{R}^L \times \mathcal{Y} \to \mathbb{R}$ is top-K calibrated if for all $\boldsymbol{\pi} \in \Delta_L$ and $x \in \mathcal{X}$:

$$\inf_{\mathbf{s} \in \mathbb{R}^L : \neg P_k(\mathbf{s}, \boldsymbol{\pi})} \mathcal{R}_{\ell|x}(\mathbf{s}, \boldsymbol{\pi}) > \mathcal{R}^*_{\ell|x}(\boldsymbol{\pi})$$

Interpretation:
ℓ is top-K calibrated if the Bayes risk can only be attained among top-K preserving vectors w.r.t. the conditional probability distribution

[12] F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". In: *ICML*. vol. 119, pp. 10727–10735, Definition 2.4.