A Two-Head Loss Function for Deep Average-K Classification

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

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

camille.garcin@inria.fr, maximilien.servajean@lirmm.fr
alexis.joly@inria.fr, joseph.salmon@umontpellier.fr

A. Hyperparameter sensitivity

We study the impact of the hyperparameters α and $|B|$ on average-K accuracy by conducting further experiments on CIFAR-100 (Section 6.2).

Figure 1 shows how CIFAR-100 average-5 accuracy varies as a function of the hyperparameter α. Average-5 accuracy is stable over a wide range of α values (roughly 10^{-2} to 10^{1}), which means that α does not require a precise tuning to obtain good results. It drops sharply for high α values, *i.e.* when the candidate classes have much more weight than the annotated labels of the training set in the objective function.

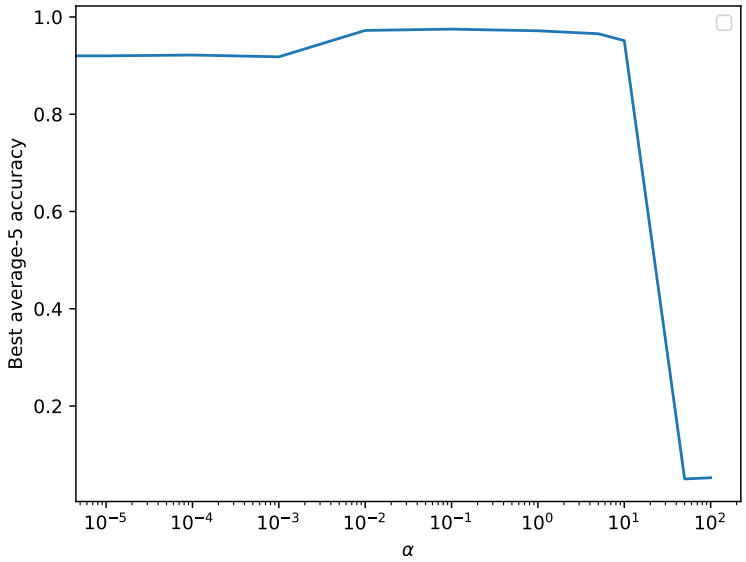

Figure 1. CIFAR-100 best validation average-5 accuracy for a DenseNet 40-40 trained with $\ell_{\text{AVG-5}}$ for different values of α.

Figure 2 shows how average-K accuracy varies with the batch size for a model trained with ℓ_{CE} or $\ell_{\text{AVG-K}}$. For a fair comparison we maintain the ratio of learning rate to batch size constant. As expected, average-K accuracy decreases for both methods when the batch size becomes too small. However, we find that $\ell_{\text{AVG-K}}$ is more robust than ℓ_{CE} to large batch size values. This can be explained by the choice of more relevant candidate classes when the batch size becomes large. This is counterbalanced by the empricical fact that SGD tends not to work well with very large batch sizes in deep learning.

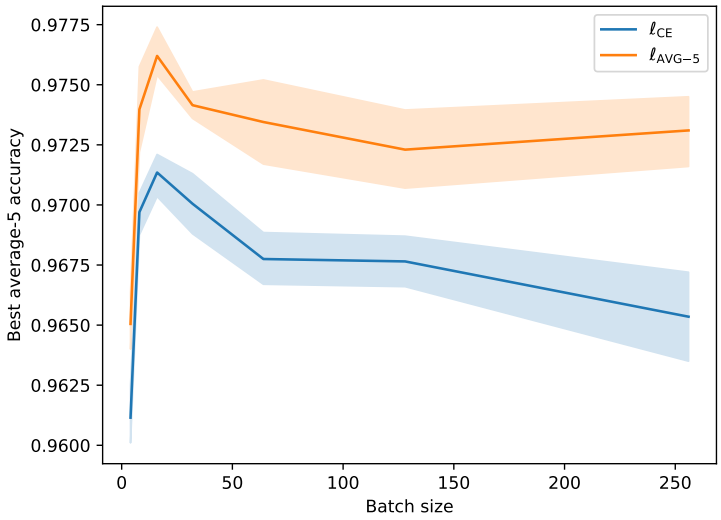

Figure 2. CIFAR-100 best validation average-5 accuracy as a function of batch size for a DenseNet 40-40 trained with $\ell_{\text{AVG-5}}$ or ℓ_{CE}. For a fair comparison we maintain the ratio of learning rate to batch size constant. The 95% confidence interval is represented.

B. Average size of output set (for the test set)

We report the average size of the output set of test instances for CIFAR-100 and Pl@ntNet-300K in Table 1 and Table 2 respectively. This size is computed as follows:

$$K_{test} = \frac{1}{|\mathcal{N}_{test}|} \sum_{i \in \mathcal{N}_{test}} \sum_{j=1}^{L} \mathbb{1}[\varsigma_j(z_i) \geq \lambda_{val}] \ . \tag{1}$$

The tables show that for both datasets, all methods lead to comparable values of K_{test}, making the comparison of *test avg-K accuracy* fair.

ℓ_{CE}	$\ell_{\text{AVG-K}}$	ℓ_{ROLE}	ℓ_{AN}	ℓ_{EPR}	$\ell_{\text{TOP-K}}$
4.85 ± 0.07	4.90 ± 0.03	4.88 ± 0.05	4.89 ± 0.08	4.88 ± 0.07	4.95 ± 0.01

Table 1. CIFAR-100 K_{test}, (DenseNet 40-40)

K	2	3	5	10
ℓ_{CE}	2.00 ± 0.01	3.01 ± 0.01	5.04 ± 0.03	10.07 ± 0.07
$\ell_{\text{TOP-K}}$	2.01 ± 0.01	3.02 ± 0.00	5.01 ± 0.01	10.02 ± 0.01
$\ell_{\text{AVG-K}}$	2.00 ± 0.01	3.01 ± 0.02	5.04 ± 0.01	10.00 ± 0.01
ℓ_{AN}	2.01 ± 0.01	3.01 ± 0.01	5.00 ± 0.01	9.97 ± 0.03
ℓ_{EPR}	2.01 ± 0.01	3.01 ± 0.01	5.03 ± 0.03	10.00 ± 0.11

Table 2. Pl@ntNet-300K K_{test}, (ResNet-18).

C. Head used for prediction

In this section, we compare the test average-K accuracy obtained with our method when using either the SCCP head or the ML head for prediction. The results for CIFAR-100 and Pl@ntNet-300K are presented in Table 3 and Table 4 respectively.

First of all, it should be noted that removing the ML head from our method and predicting with the SCCP head will give the same results as a vanilla model trained with ℓ_{CE}.

Interestingly, as can be seen in Table 3 and Table 4, using the SCCP head for prediction outperforms a simple model trained with ℓ_{CE}. This indicates that the presence of the ML head enhances the performance of the SCCP head. Here is a tentative explanation: the SCCP head, trained with ℓ_{ce}, is optimized to maximize the probability of a single class. A model trained solely with ℓ_{ce} focuses on extracting discriminative features to predict a single class. In contrast, the ML head's objective is to predict sets of classes rather than a single one, encouraging the discovery of shared patterns among ambiguous classes. As the model is trained jointly, the ML head helps create richer feature representations, mitigating the usual overfitting associated with the cross-entropy loss.

From Table 3 and Table 4, we can see that predicting with the SCCP head performs slightly worse on CIFAR-100, and either slightly better or worse on Pl@ntNet-300K, depending on K. Ultimately, using either head for prediction leads to better results than any other method. For a given problem, practitioners can either default to using the ML head for prediction or evaluate both heads and select the best performing one.

ℓ_{CE}	$\ell_{\text{AVG-K}}$ (ML)	$\ell_{\text{AVG-K}}$ (SCCP)
96.83 ± 0.16	**97.35** ± 0.06	96.93 ± 0.06

Table 3. CIFAR-100 test average-5 accuracy (%), (DenseNet 40-40). (ML) and (SCCP) indcates that the head used for prediction is respectively the ML head and the SCCP head.

K	ℓ_{CE}	$\ell_{\text{AVG-K}}$ (ML)	$\ell_{\text{AVG-K}}$ (SCCP)
2	89.63 ± 0.08	90.34 ± 0.06	**90.77** ± 0.16
3	92.64 ± 0.17	93.81 ± 0.10	**94.05** ± 0.11
5	95.11 ± 0.18	**96.42** ± 0.09	96.39 ± 0.07
10	97.11 ± 0.09	**98.23** ± 0.03	98.17 ± 0.06

Table 4. Pl@ntNet-300K test average-K accuracy (%), (ResNet-18). (ML) and (SCCP) indcates that the head used for prediction is respectively the ML head and the SCCP head.

D. Experiments details

We report in Tables 5 and 6 the hyperparameters selected after grid search for all losses, for CIFAR-100 and Pl@ntNet-300K datasets respectively.

loss	hyperparameters
ℓ_{CE}	-
$\ell_{\text{AVG-K}}$	$\alpha = 0.3$
ℓ_{AN}	-
ℓ_{EPR}	$\beta = 0.01$
$\ell_{\text{TOP-K}}$	$\epsilon = 0.2$
ℓ_{ROLE}	$\lambda = 0.0$, learning rate Θ: $\times 1.0$

Table 5. Hyperparameters selected after grid search for the CIFAR-100 experiments.

E. Additional models

We conduct further experiments on Pl@ntNet-300K: we train more models using exactly the same setting as in Section 6.3. We train a ResNet-34, a DenseNet169 and a ViT-16. We report the results in Table 7. They confirm that our loss is able to outperform other losses for various architectures, in particular for few-shot classes and medium-shot classes, which represent the vast majority of classes for Pl@ntNet-300K.

loss	hyperparameters
ℓ_{CE}	-
$\ell_{\text{AVG-K}}$	$\alpha = 5.0$
ℓ_{AN}	-
ℓ_{EPR}	$\beta = 0.001$
$\ell_{\text{TOP-K}}$	$\epsilon = 1.0$

Table 6. Hyperparameters selected after grid search for the Pl@ntNet-300K experiments.

	DenseNet169	ResNet34	ViT
ℓ_{CE}	96.75 (47.61/89.85/96.89)	95.62 (40.82/85.80/95.34)	96.74 (44.14/88.89/**97.13**)
$\ell_{\text{TOP-5}}$	97.21 (58.13/88.20/97.08)	96.25 (52.49/83.30/95.53)	96.77 (54.91/86.49/96.30)
$\ell_{\text{AVG-5}}$	**97.49** (**65.97/92.36/97.20**)	**96.80** (**61.85/89.07/96.45**)	**96.88** (**65.33/90.43**/96.59)
ℓ_{AN}	96.36 (39.55/82.56/96.57)	95.28 (33.08/76.66/94.89)	95.05 (22.53/74.54/94.79)
ℓ_{EPR}	96.08 (39.87/79.72/95.87)	94.79 (35.53/71.37/93.58)	94.82 (32.54/74.60/93.71)

Table 7. Pl@ntNet-300K test average-5 accuracy for different models.