HMLoss: Une fonction de coût robuste au déséquilibre des classes

Arthur Cartel FOAHOM GOUABOU[1], Rabah IGUERNAISSI[1], Jean Luc DAMOISEAUX[1], Abdellatif MOUDAFI[1], Djamal MERAD[1]

[1] Laboratoire d'Informatique et Systèmes UMR 7020, Aix-Marseille University
163 Avenue de Luminy, CEDEX 09, 13288 Marseille, France
cartel.gouabou@lis-lab.fr, rabah.iguernaissi@lis-lab.fr, jean-luc.damoiseaux@lis-lab.fr
abdellatif.moudafi@univ-amu.fr, djamal.merad@lis-lab.fr

Résumé – Ce travail propose une résolution de la problématique du biais induit durant l'apprentissage des modèles neuronaux sur des bases déséquilibrées. Pour cela, nous introduisons une nouvelle fonction de coût dénommée *'Hard Mining Loss'* (HMLoss) permettant de réduire simultanément la contribution des exemples faciles et aberrants durant l'apprentissage tout en augmentant la contribution des exemples difficiles, permettant ainsi au modèle de se focaliser sur les échantillons discriminants. La fonction HMLoss surclasse les méthodes courantes pour résoudre ce problème dans des applications de classification d'images. Les bases de données, codes et architectures utilisés sont disponibles à l'adresse: https://github.com/cartelgouabou/HMLoss.

Abstract – This work adresses the class imbalances issue in deep learning. We introduce a new cost function called 'Hard Mining Loss' (HMLoss) allowing to reduce simultaneously the contribution of both easy examples and outliers while increasing the contribution of hard examples during learning, thus allowing the model to focus on informative samples. HMLoss outperforms common methods for solving this problem in image classification applications. Datasets, code and models are publicly available at https://github.com/cartelgouabou/HMLoss.

1 Introduction

La majorité des jeux de données utilisés pour entraîner des réseaux de neurones convolutifs (CNNs) présentent une distribution asymétrique [1], ceci crée un biais affectant négativement la performance des CNNs entraînés [2]. Plusieurs contributions tentent de résoudre cette problématique [1, 2]. Ces méthodes se regroupent en deux groupes : les approches du point de vue des données et les approches du point de vue des algorithmes.

L'approche du point de vue des données utilise des méthodes de rééchantillonnage pour réduire le déséquilibre directement sur les données. Elles présentent l'inconvénient d' introduire de grandes quantités d'échantillons dupliqués ou de supprimer des exemples précieux, ce qui peut entraîner un sur-apprentissage du modèle.

Les approches du point de vue de l'algorithme consistent à ajuster le processus d'apprentissage de manière à faciliter la tâche d'apprentissage spécifiquement en ce qui concerne les échantillons de la classe minoritaire. L'apprentissage sensible aux coûts fait partie de cette dernière approche et vise à modifier la fonction de coût pour rendre le modèle plus sensible aux classes minoritaires. L'état de l'art actuel de ces approches [3, 4] conçoit des fonctions de coût basées sur la pondération des échantillons pour gérer le déséquilibre des classes. Ces méthodes réduisent la pondération des échantillons correctement classés. De cette façon, les échantillons faciles à classer ont des poids inférieurs à ceux des échantillons difficiles. Ces méthodes ont permis de réduire le biais induit sur les modèles entraînés sur des bases déséquilibrées. Nous émettons l'hypothè se que la principale limite de ces méthodes est qu'elles réduisent également le poids des échantillons difficiles à classer, ce qui conduit à une pondération sous-optimale des pertes. D'autre part, [5] a découvert que les échantillons avec de très grands gradients (un échantillon très difficile) affectent la stabilité du modèle. Cette étude révèle que ces échantillons existent de manière stable même lorsque le modèle converge, ce qui suggère qu'il s'agit principalement de valeurs aberrantes. Ces intuitions suggèrent que la prise en compte des niveaux des gradients est nécessaire pour construire des modèles plus robustes.

Dans cet article, nous introduisons une nouvelle fonction de coût, à savoir la perte minière dure dérivée de l'expression anglaise hard mining loss (HMLoss). HMLoss est une alternative remarquable aux approches précédentes pour traiter les problèmes de déséquilibre de classe. Cette nouvelle fonction de perte permet d'améliorer les erreurs de généralisation des classes minoritaires sans compromettre la capacité du modèle à s'adapter aux classes fréquentes. De plus, inspirée par la théorie de [5], la fonction de perte proposée régularise les valeurs aberrantes en minimisant les gradients importants. Intuitivement, la fonction de perte proposée réduit la contribution des échantil lons faciles et des valeurs aberrantes pendant l'entraînement et augmente la contribution des échantillons difficiles, ce qui permet au modèle de se focaliser sur les échantillons informatifs. Une visualisation simple de l'effet de HMLoss est présentée à la Figure 1.

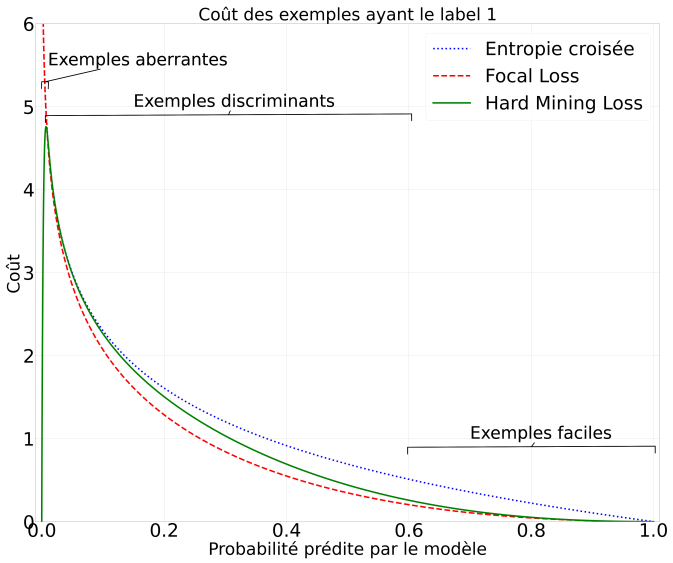

FIGURE 1 – Illustration de la nouvelle fonction de coût HMLoss comparée aux fonctions d'entropie croisée et *Focal Loss*.

2 Méthodes

2.1 Notion de difficulté d'un exemple

Soit x une image d'un jeu de données D, et l son étiquette correspondante dans D, $p_l \in [0,1]$ est la probabilité prédite par un modèle que x corresponde à son étiquette l. Lorsqu'un modèle fait une bonne prédiction avec une probabilité très élevée ($p_l >> 0.5$), l'image x peut être considérée comme un exemple facile ; à contrario, lorsque la probabilité prédite est faible (p_l proche de 0.5), l'image x peut être assimilée à un exemple difficile. Les échantillons pour lesquels le modèle a prédit des probabilités très faibles ($p_l << 0.5$) sont considérés comme des valeurs aberrantes [5].

2.2 La fonction HMLoss

Considérons une classification binaire pour simplification. Soit une image x, et $p \in [0,1]$ la probabilité prédite par un modèle que x appartienne à la classe $y \in \{0,1\}$. Définissons la variable p_t ci-dessous :

$$p_t = \begin{cases} p & si\ y = 1 \\ 1-p & sinon \end{cases} \quad (1)$$

Nous introduisons la fonction de coût HMLoss pour résoudre le problème de déséquilibre de classes. L'idée principale de notre approche est d'augmenter la contribution sur le coût d'apprentissage des exemples plus difficiles tout en maintenant une contribution faible des exemples faciles et des valeurs aberrantes. Pour définir une telle fonction continue et dérivable, nous nous sommes inspirés de la théorie du signal en empruntant la fonction sinus cardinale. Formellement, nous introduisons un terme de pondération w_t à la fonction originale d'entropie croisée. La fonction HMLoss est définie de la manière suivante :

$$HMLoss(p,y) = -w_t log(p_t) \quad (2)$$

Avec le terme de pondération w_t calculé de la manière suivante :

$$\frac{\sin(\pi p_t)}{\pi p_t} - \frac{\sin(\delta \pi p_t)}{\delta \pi p_t} \exp^{-\delta p_t} \quad (3)$$

Dans l'équation 3, le premier terme définit une distribution des coûts suivant la même dynamique que l'entropie croisée. Afin de pouvoir atténuer la contribution des larges gradients, nous avons soustrait du premier terme un sinus cardinal de fréquence plus élevée. La fonction de perte résultante présentait des oscillations qui pouvaient affecter la stabilité de l'apprentissage. Par conséquent, nous avons intégré un facteur d'amortissement à travers la fonction exponentielle pour lisser la distribution. Le paramètre δ permet de déplacer le maximum de la fonction, modifiant ainsi le seuil à partir duquel un échantillon peut être considéré comme aberrant. La variabilité d'un tel seuil est pertinente dans la mesure où la proximité des caractéristiques qui peuvent exister entre les différentes classes d'un jeu de données est corrélée au seuil des échantillons qui se situent dans la gamme des larges gradients et ne nécessitant donc pas d'attention particulière durant la phase d'apprentissage. Nous avons observé empiriquement que les modèles obtiennent généralement les meilleures performances pour $\delta \geq 1000$. Le choix du paramètre δ se fait par grid-search avec une valeur de pas égale à des puissances de 10.

Algorithm 1 Protocole d'apprentissage

Require: Base d'entraînement $D = (x_i, y_i)_{i=1}^{N}$, modèle $f(x,w)$, Nombre d'époque T, Epoque de transition T_0, Taille mini-batch m, Taille de la base N, facteur de réduction τ

Initialisation aléatoire de w
for *t*=1 **to** T_0 **do**
  $D_m \leftarrow$ SampleMiniBatch (D, m)
  $L(w) \leftarrow \frac{1}{m} CE(f(x,y); y)$
  $w_t \leftarrow w_{t-1}$ - $\eta \nabla_w L(w)$
end for
$\eta \leftarrow \eta / \tau$
for *t*=T_0 **to** T **do**
  $D_m \leftarrow$ SampleMiniBatch (D, m)
  $L(w) \leftarrow \frac{1}{m} HMLoss(f(x,y); y)$
  $w_t \leftarrow w_{t-1}$ - $\eta \nabla_w L(w)$
end for

2.3 Protocole d'entraînement en deux phases

Nous avons observé empiriquement que de meilleures caractéristiques étaient produites lors de l'initialisation du modèle avec la fonction d'entropie croisée. Ceci est similaire à l'observation faite par Cao et al.[8]. En nous inspirant de cela, nous avons défini un protocole d'entraînement (voir Algorithme 1).

TABLE 1 – Taux d'erreur de validation de ResNet-32 avec differentes fonctions de coût sur des versions déséquilibrées de type LT et STEP des bases de données CIFAR-10 et CIFAR-100. Le ratio de déséquilibre ρ est de 200. Le signe de la dague indique que les résultats sont tirés des articles originaux. La meilleure valeur de l'hyper-paramètre δ obtenue par grid search est indiquée. HMLoss surclasse toutes les autres méthodes.

BASE	CIFAR-10		CIFAR-100	
TYPE DE DÉSÉQUILIBRE	LT	STEP	LT	STEP
CE	23.1 ± 0.5	28.9 ± 0.4	56.6 ± 0.6	61.8 ± 2.7
CSCE [6]	22.1 ± 0.4	29.7 ± 1.1	61.8 ± 2.0	69.9 ± 1.7
FL [3]	21.0 ± 0.4	27.0 ± 0.5	59.1 ± 0.9	68.5 ± 0.8
CBLOSS [7]	21.8 ± 0.6	28.9 ± 0.8	61.3 ± 1.8	70.8 ± 1.7
LDAMLOSS [8]	21.5 ± 0.1	28.2 ± 0.3	58.5 ± 0.3	56.7 ± 0.4
$SEQL^{\dagger}$ [4]	-	-	56.6	-
$BALMS^{\dagger}$ [9]	18.5 ± 0.0	-	54.5 ± 0.1	-
HMLOSS	21.8 ± 0.3	27.2 ± 0.3	54.7 ± 0.3	56.1 ± 0.1
HMLOSS + CS	$\mathbf{17.3 \pm 0.3}$	$\mathbf{17.9 \pm 0.5}$	52.7 ± 0.3	$\mathbf{50.4 \pm 0.4}$
HMLOSS + CB	18.1 ± 0.2	18.3 ± 0.3	$\mathbf{52.6 \pm 0.4}$	50.5 ± 0.3
Best Hyper	$\delta = 10^5$	$\delta = 10^5$	$\delta = 10^4$	$\delta = 10^4$

En pratique, la fonction HMLoss se combine avec des stratégies standard de pondération [6, 7]. Empiriquement, cela conduit à une amélioration des performances.

3 Cadre expérimental

Afin d'évaluer notre méthode, nous l'avons comparée à la fonction d'entropie croisée (CE) ainsi qu'à l'état de l'art des fonctions de coût présentes dans la littérature et communément adoptées pour résoudre le déséquilibre des classes. Des expéri mentations approfondies sont pour cela menées sur les jeux de données déséquilibrées de CIFAR-10 et CIFAR-100, et sur le jeu de données médicales ISIC 2019. **Jeu de données** : les jeux de données de CIFAR-10 et CIFAR-100 contiennent tous les deux 50 000 images d'entraînement et 10 000 images de validation de la taille 32 * 32 pixels avec respectivement 10 et 100 classes. En suivant le même protocole que les travaux antérieux sur la problématique, nous avons créé deux versions déséquilibrées des jeux de données CIFAR : la version LT [7] et la version STEP [2]. Le ratio de déséquilibre étant fixé à 200 pour les deux jeux de données dans nos expériences. S'agissant de la base ISIC2019, elle comprend 25 331 images dermoscopiques à haute résolution réparties dans huit classes. Pour évaluer notre approche dans un cadre binaire, nous avons centré notre étude sur la classification des classes mélanomes et naevus. La base de données ainsi réduite contient 15 397 images, dont 4 522 mélanomes (classe minoritaire) et 12 875 nævus (classe majoritaire), avec un ratio de déséquilibre ρ de 2,84. La base a été répartie en base d'entraînement (80%), base de validation (10%) et base de test (10%). **Détails de l'entraînement** : Pour l'implémentation sur les jeux de données de CIFAR-10 et CIFAR-100, les images de la base d'entraînement et de validation ont été normalisées dans la plage $[0, 1]$. L'architecture ResNet32 a été utilisée pour réaliser toutes les expérimentations. Tous les modèles ont été entraînés avec une taille de lot de 128 durant 200 époques. Similairement à la procédure suivie par Cui et Al. [7], nous avons linéairement augmenté notre taux d'apprentissage durant les 5 premières époques, puis nous avons fixé notre taux d'apprentissage initial à 0,1, diminué par la suite de 0,01 à l'époque 160 et 180. S'agissant de l'implémentation sur le jeu de données ISIC 2019, nous avons appliqué les techniques de prétraitement standard aux images de lésions cutanées [10]. L'architecture pré-entraînée EfficienetB3 a été utilisée pour réaliser toutes nos expérimentations. Afin d'adapter l'architecture à une classification binaire, nous avons modifié la dernière couche en la remplaçant par une couche dense ayant un neurone.

4 Résultats

L'entraînement d'un réseau de neurones suit un processus stochastique, raison pour laquelle nous avons exécuté chacune de nos expériences à dix reprises et nous reportons par la suite les moyennes des résultats obtenus. Nous noterons par *HMLoss + CS* et *HMLoss + CB* la combinaison de la fonction HMLoss avec respectivement les approches de pondération présentées en [6] et [7].

Les résultats obtenus sur les jeux de données de validation de CIFAR-10 et CIFAR-100 sont détaillés dans la Table1. Nous évaluons d'abord notre approche sans stratégies de pondération. Les résultats montrent que notre approche surpasse largement la fonction CE classique et atteint des performances comparables aux méthodes antérieures utilisées pour l'apprentissage du déséquilibre. Seul l'état de l'art actuel sur le déséquilibre des classes, à savoir la fonction BALMS, parvient à obtenir de meilleures performances que la fonction HMLoss sans pondération. Deuxièmement, nous avons comparé la fonction HMLoss combinée avec les deux approches de pondération retenues. Nous avons observé que cela a permis à la fonction HMLoss de surpasser toutes les approches mentionnées précédemment,

TABLE 2 – Performance de l'architecture pré-entrainée EfficientNetB3 avec differentes fonction de coût sur la base ISIC2019. AUC représente l'aire en dessous de la courbe caractéristique.

Fonction de coût	AUC
CE	$94{,}8 \pm 0.4$
CSCE [6]	$94{,}3 \pm 0.7$
FL [3]	95.1 ± 0.4
CBLoss [7]	$94{,}2 \pm 0.9$
LDAM [8]	93.4 ± 0.5
HMLoss	$95{,}8 \pm 0.4$
HMLoss + CS	$\mathbf{96{,}1 \pm 0.4}$
HMLoss + CB	$95{,}7 \pm 0.3$
δ	10^4

y compris la fonction BALMS, avec une marge moyenne de 1,2%.
La Table2 présente les résultats obtenus sur le jeu de données ISIC2019. Notre méthode HMLoss a obtenu la meilleure performance, atteignant une aire en dessous de la courbe caractéristiques (AUC) de 96,1%. Parmi les fonctions de coût communément utilisées, FL a obtenu la meilleure performance juste derrière notre approche avec une AUC de 95,1%.

5 Conclusion

Dans ce travail, nous avons introduit une méthode innovante permettant de traiter le déséquilibre des classes pour l'apprentissage profond. La fonction de coût HMLoss proposée réduit la contribution des exemples faciles et aberrants tout en augmentant la contribution des exemples discriminatifs sur le coût total durant l'apprentissage. Nous avons réalisé de nombreuses expériences sur des applications de classification d'images à partir des jeux de données CIFAR10, CIFAR100 et ISIC 2019 afin de valider notre approche. Les résultats obtenus nous ont permis de démontrer que notre méthode est plus performante que les fonctions de coût existantes présentes dans la littérature. En outre, notre approche est simple, facile à implémenter et efficace. Cependant, il existe encore un potentiel de recherche prometteur sur HMLoss. Son efficacité par exemple dans d'autres applications d'apprentissage automatique, telles que la segmentation d'images, la détection de fraudes et le traitement du langage naturel, pourrait être étudiée, puisque toutes ces applications souffrent également du problème de déséquilibre de classe dans les jeux de données existantes.

6 Remerciement

Ce travail a été en partie soutenu par l'Agence Nationale de la Recherche (ANR) dans le cadre du projet DIAMELEX. Ces travaux ont bénéficié d'un accès aux ressources en HPC/IA de l'IDRIS au travers de l'allocation de ressources 2022-AD01101-2626R1 attribuée par GENCI.

Références

[1] Y. Zhang *et al.*, "Deep long-tailed learning : A survey," *arXiv preprint arXiv :2110.04596*, 2021. [Online]. Available : https://arxiv.org/abs/2110.04596

[2] M. Buda *et al.*, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, Oct. 2018. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S0893608018302107

[3] T.-Y. Lin *et al.*, "Focal Loss for Dense Object Detection," 2017, pp. 2980–2988. [Online]. Available : https://openaccess.thecvf.com/content_iccv_2017/html/Lin_Focal_Loss_for_ICCV_2017_paper.html

[4] J. Tan *et al.*, "Equalization loss for long-tailed object recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 662–11 671. [Online]. Available : https://arxiv.org/abs/2003.05176

[5] B. Li *et al.*, "Gradient Harmonized Single-Stage Detector," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 8577–8584, Jul. 2019, number : 01. [Online]. Available : https://ojs.aaai.org/index.php/AAAI/article/view/4877

[6] Y. S. Aurelio *et al.*, "Learning from Imbalanced Data Sets with Weighted Cross-Entropy Function," *Neural Process Lett*, vol. 50, no. 2, pp. 1937–1949, Oct. 2019. [Online]. Available : https://doi.org/10.1007/s11063-018-09977-1

[7] Y. Cui *et al.*, "Class-Balanced Loss Based on Effective Number of Samples," 2019, pp. 9268–9277. [Online]. Available : https://openaccess.thecvf.com/content_CVPR_2019/html/Cui_Class-Balanced_Loss_Based_on_Effective_Number_of_Samples_CVPR_2019_paper.html

[8] K. Cao *et al.*, "Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss," *NeurIPS*, 2019. [Online]. Available : https://arxiv.org/abs/1906.07413

[9] J. Ren *et al.*, "Balanced meta-softmax for long-tailed visual recognition," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4175–4186, 2020. [Online]. Available : https://arxiv.org/abs/2007.10740

[10] A. C. Foahom Gouabou *et al.*, "Ensemble Method of Convolutional Neural Networks with Directed Acyclic Graph Using Dermoscopic Images : Melanoma Detection Application," *Sensors*, vol. 21, no. 12, p. 3999, Jan. 2021, number : 12 Publisher : Multidisciplinary Digital Publishing Institute. [Online]. Available : https://www.mdpi.com/1424-8220/21/12/3999