Apprentissage supervisé et contrastif de représentations à l'aide de réseaux siamois pour la classification hiérarchique

Ilyass MOUMMAD[1] Nicolas FARRUGIA[1]

[1]IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-2923 Brest, France

prénom.nom@imt-atlantique.fr

Résumé – Nous proposons une nouvelle fonction de coût pour l'apprentissage supervisé de représentations, inspiré de la méthode auto-supervisée SimSiam. L'extension que nous proposons SupSiam permet d'apprendre de meilleures représentations en incorporant l'information des labels. Nous étendons SupSiam au cadre des classes hiérarchiques en introduisant HierSupSiam. Nous expérimentons sur le dataset CIFAR-100 avec les modèles ResNet18 et ResNet50. Nos expériences montrent que l'apprentissage d'un simple classifieur linéaire sur les représentations apprises par notre approche obtient des performances supérieures à l'apprentissage classique par l'entropie croisée quand la capacité du modèle est grande.

Abstract – We propose a new cost function for supervised learning of representations, inspired by the self-supervised method SimSiam. The extension we propose SupSiam allows learning better representations by incorporating label information. We extend SupSiam to the hierarchical class setting by proposing HierSupSiam. We experiment on the CIFAR-100 dataset with the ResNet18 and ResNet50 models. Our experiments show that training a simple linear classifier on the representations learned by our approach outperforms classical cross-entropy learning when the model capacity is large.

1 Introduction

[1] L'apprentissage profond est devenu l'un des principaux fondements des systèmes d'intelligence artificielle. Grâce aux annotations humaines, les réseaux de neurones profonds ont la capacité de reconnaître des formes à partir d'une grande quantité de données dans de nombreux domaines tels que la vision par ordinateur, le traitement du langage naturel ou le traitement des signaux acoustiques [12]. Le succès de l'apprentissage profond est lié à l'apprentissage de modèles comportant plusieurs centaines de milliers à quelques dizaines millions de paramètres, sur de grandes quantités de données annotées. Plus récemment, de nouvelles méthodes d'apprentissages, dites "auto-supervisées", ont été proposées, en raison de la disponibilité de grandes quantités de données non étiquetées. Précisément, l'apprentissage auto-supervisé (AAS) est un paradigme d'apprentissage de représentations (i.e. apprentissage d'une fonction d'un espace de données brutes vers un espace dit "latent" de plus faible dimension) à partir de données non étiquetées où les données elles-mêmes fournissent la supervision, que nous appelons une tâche de prétexte [7].

Deux familles communes de tâches de prétextes sont utilisées en AAS. La première est l'auto-prédiction, qui consiste à prédire une partie ou la totalité des données originales à partir d'une partie ou d'une version modifiée des mêmes données (par exemple, la restauration ou le débruitage pour les tâches liées à l'image) [4]. La deuxième famille d'AAS regroupe les approches contrastives, qui consistent à apprendre des représentations similaires pour différentes versions (ou "vues") des mêmes données (aussi appelées paires positives ou exemples positifs) et des représentations dissimilaires pour des vues de données différentes (aussi appellées paires négatives ou exemples négatifs) [2]. Dans la plupart des approches contrastives, la construction des vues s'appuie fortement sur l'augmentation stochastique des données, c'est-à-dire l'application de transformations sur les données brutes, par exemple le recadrage et redimensionnement des images, le retournement horizontal ou la distortion des couleurs. L'apprentissage contrastif est également très performant pour l'apprentissage cross-modal, par exemple en contrastant des représentations issues de descriptions textuelles d'images [11] ou de sons [13].

Les recherches récentes en AAS se concentrent davantage sur les approches contrastives, car elles semblent plus performantes que les approches d'auto-prédiction [10, 2, 5]. L'apprentissage contrastif supervisé ("Supervised Contrastive Learning", SCL) [8] est une approche supervisée, donc nécessitant des annotations, qui s'inspire des approches d'AAS contrastives, en découplant l'apprentissage supervisé en deux étapes. La première étape est l'apprentissage de la représentation à l'aide d'informations sur les labels en plus de l'augmentation des données, et la deuxième étape consiste à entraîner un classificateur, en figeant les représentations apprises lors de la première étape.

Le succès de l'apprentissage contrastif est lié au fait que les paires négatives sont éloignées les unes des autres dans l'espace latent, afin d'éviter l'apprentissage de représentations où un modèle produit un vecteur constant pour toutes les données. L'apprentissage contrastif nécessite typiquement de travailler avec des grands batchs de données pour la descente de gradients stochastique lors de l'apprentissage, ou de garder une grande quantité de données en mémoire pour échantillonner des exemples négatifs [5]. Le travail de SimSiam [3] aborde ce problème en proposant une approche qui n'utilise pas d'exemples négatifs, en se basant sur les réseaux siamois. Nous étendons le travail de SimSiam au cadre supervisé et proposons SupSiam, illustré dans la figure 1.

Dans certains problèmes de classification, une hiérarchie des classes peut être définie afin de représenter les relations entre

[1]Ce travail a été financé par le programme de l'Agence Nationale de la Recherche "AI@IMT", ainsi que par l'entreprise OSO-AI.

différentes classes [14]. Un apprentissage de représentations préservant et exploitant la hiérarchie pourrait permettre une meilleure généralisation, notamment pour transférer l'apprentissage à d'autres tâches où on souhaiterait un riche extracteur de caractéristiques. Nous proposons HierSupSiam, une extension de SupSiam hiérarchique.

À notre connaissance, il s'agit de la première approche d'apprentissage de représentation sans exemple négatif qui est plus performante que l'entropie croisée sur le problème de classification. De plus, notre travail permet de fournir une fonction de coût unifiée qui peut être utilisée pour l'apprentissage auto-supervisé ou supervisé, et permet également d'exploiter la hiérarchie des classes.

2 Travaux Connexes

SimSiam [3] est une méthode d'AAS qui permet d'apprendre des représentations sans labels à l'aide de réseaux siamois en rapprochant deux représentations des vues (ou augmentations) provenant de la même image, sans la nécessité d'éloigner les représentations des exemples négatifs dans l'espace latent comme fait la méthode SimCLR [2]. Ceci est réalisé en ne retropropageant que sur la représentation d'une vue à la fois en fixant l'autre représentation, Les flèches bleues dans la figure 1 illustrent cette méthode. Notre approche SupSiam étend SimSiam [3] dans le cadre supervisé en s'inspirant de l'apprentissage contrastif supervisé (SCL) [8], qui unifie les paradigmes d'apprentissage supervisé et d'apprentissage auto-supervisé.

La hiérarchie des classes peut être exploitée pour apprendre des représentations plus riches. Le travail de [1] rajoute au réseau convolutif une couche de classification après chaque couche de convolution avec différents groupes de classes, en incrémentant le nombre de groupes à chaque couche suivante, ce qui impose une hiérarchie de classe. [14], qui étend SCL [8] au cadre hiérarchique, est proche de notre extension hiérarchique HierSupSiam.

3 Méthode

Notre contribution consiste à considérer de nombreux exemples positifs par échantillon en utilisant une approche sans exemple négatif. Ce procédé n'est pas évident car la tâche de prétexte est plus difficile que simplement apprendre l'invariance aux augmentations, car nous cherchons également à apprendre l'invariance à l'étiquette de classe. La deuxième contribution de ce travail consiste à étendre notre approche au cadre des classes hiérarchiques.

Soit f un encodeur réseau de neurones constitué d'un extracteur convolutionnel (réseau de neurones convolutionnel) et d'un perceptron multi couches (PMC) ici appellé "projecteur", et h un PMC appelé prédicteur. Le pipeline de SimSiam contient 2 branches : une première pour f, et une seconde pour $f \circ h$, il prend en entrée deux vues x_1 et x_2 d'une donnée x dans les deux branches, et qui sort $f(x_1) = z_1$, $h(f(x_1)) = p_1$, et $f(x_2) = z_2$, $h(f(x_2)) = p_2$, de f et de $f \circ h$ respectivement. L'idée centrale SimSiam est de rapprocher p_1 pour qu'il soit proche de z_2, et aussi de rapprocher p_2 pour qu'il soit proche de z_1 en propageant le gradient sur la deuxième branche ($f \circ h$) et en empêchant la retropropagation du gradient (ci-dessous dénommée operation `stopgrad`) sur la première branche (f). On note ici que les deux branches partagent les mêmes paramètres de l'encodeur. Fig. 1 illustre cette procédure.

La fonction de coût $\mathcal{L}$ est l'opposé de la similarité cosinus entre les sorties de la première et de la deuxième branche, et vice-versa (fonction symmétrique), definie pour chaque exemple :

$$\mathcal{L} = \frac{1}{2} D(p_1, \texttt{stopgrad}(z_2)) + \frac{1}{2} D(p_2, \texttt{stopgrad}(z_1)) \tag{1}$$

où D est l'opposé du produit scalaire défini comme :

$$D(p_i, z_j) = -\frac{p_i}{\|p_i\|_2} \cdot \frac{z_j}{\|z_j\|_2} \tag{2}$$

et $\|.\|_2$ est la norme l_2.

On étend SimSiam au cadre supervisé où on rapproche la représentation d'une vue d'un exemple non seulement à la représentation d'une autre vue du même exemple mais aussi aux représentations des autres vues des exemples appartenant à la même classe du batch multivues, on définit la nouvelle fonction de coût supervisée $\mathcal{L}^{sup}$ pour chaque vue $i \in I = \{1...2N\}$ comme suit :

$$\begin{aligned} \mathcal{L}^{sup} &= \frac{1}{|P(i)|} \sum_{k \in P(i)} \Big(\frac{1}{2} D(p_i, \texttt{stopgrad}(z_k)) \\ &+ \frac{1}{2} D(p_k, \texttt{stopgrad}(z_i)) \Big) \end{aligned} \tag{3}$$

où $P(i) = \{j \in I : y_i = y_j\}$ est l'ensemble des indices des exemples de la même catégorie que i dans le batch multivues (indices des exemples positifs), et $|P(i)|$ sa cardinalité.

Nous étendons ensuite cette fonction de coût pour incorporer une hiérarchie des classes. Soit $\mathbf{L}$ l'ensemble des niveaux de labels, et soit $l \in \mathbf{L}$ un niveau de label. On définit la nouvelle fonction de coût hiérarchique $\mathcal{L}^{hier}$ pour chaque vue $i \in I = \{1...2N\}$ comme :

$$\mathcal{L}^{hier} = \sum_{l \in L} \alpha_l \mathcal{L}_l^{sup} \tag{4}$$

où α_l est un coefficient de pénalité pour le niveau l.

Les fonctions de coût SupSiam et HierSupSiam sont simples à implémenter, nous donnons un pseudocode du style de PyTorch dans Algorithme 1.

Algorithme 1 : Pseucode PyTorch de SupSiam

```
# f: backbone + projection mlp
# h: prediction mlp

for x, y in loader:
# load a minibatch x, y with n labeled samples
    x1, x2 = aug(x), aug(x)  # random augmentation
    z1, z2 = f(x1), f(x2)  # projections, n-by-d
    p1, p2 = h(z1), h(z2)  # predictions, n-by-d

    L = D(p1, z2, y)/2 + D(p2, z1, y)/2  # loss

    L.backward()  # back-propagate
    update(f, h)  # SGD update

def D(p, z, y):  # negative cosine similarity
    z = z.detach()  # stop gradient
    p = normalize(p, dim=1)  # l2-normalize
    z = normalize(z, dim=1)  # l2-normalize
    m = eq(y, y.T).float()  # positive pairs mask
    sim = mm(p, z.T)  # sim matrix
    return -sim[m==1].mean()  # select positive sim
```

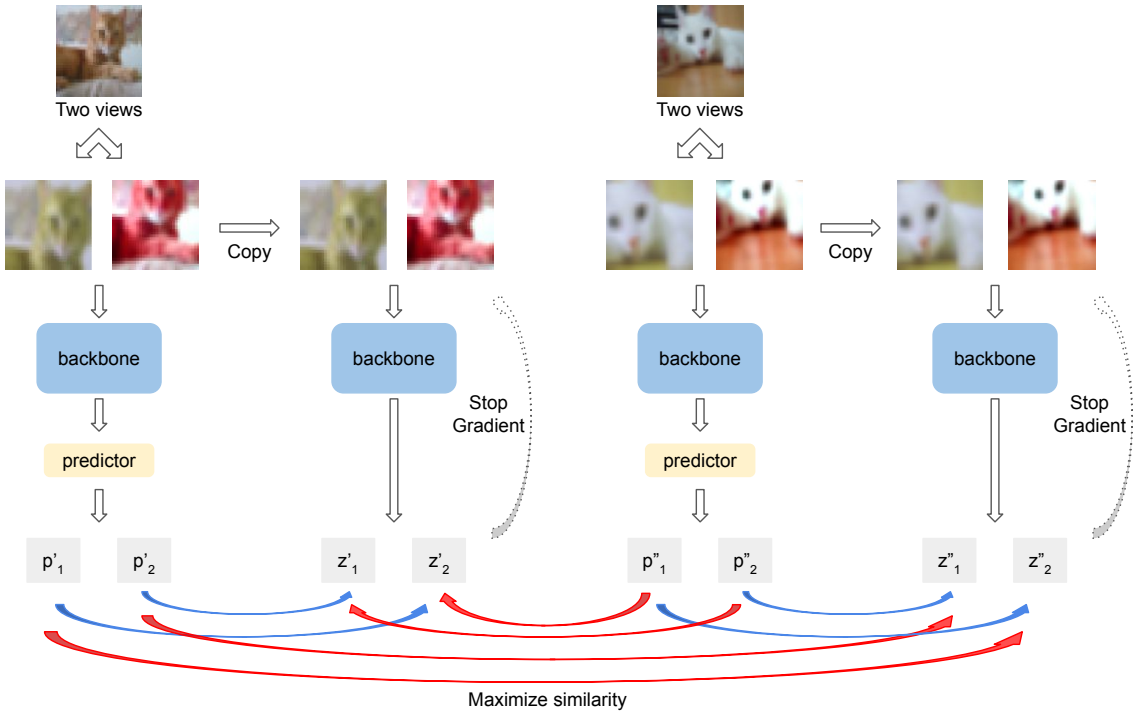

FIGURE 1 : SimSiam : La représentation de la première vue (sortie de $f \circ h$) est rapprochée de la représentation de la deuxième vue (sortie de f), et vice versa (flêches bleues). SupSiam : la représentation d'une vue d'une image est rapprochée de la représentation d'une vue d'une autre image de la même classe, ainsi qu'une autre vue de la même image (flêches bleues et rouges).

4 Experimentations

Nous avons testé notre approche sur le jeu de données CIFAR-100, qui comporte des images de taille 32 par 32 pixels, avec 100 classes et 500 exemples par classe pour l'entrainement. Les 100 classes de CIFAR-100 sont elles-mêmes regroupées en 20 superclasses, ce qui définit deux nivaeaux de hiérarchie que nous exploitons dans HierSupSiam. Nous avons utilisé les mêmes augmentations de données que SimSiam [3] : recadrage redimensionné, retournement horizontal, distorsion de couleur et conversion en niveaux de gris.

Nous avons expérimenté avec les encodeurs ResNet18 et ResNet50 [6] avec un PMC à une couche cachée de dimension 2048 et une dimension de sortie de 2048 (avec une couche de normalisation des batchs après chaque fonction d'activation) comme extracteur de caractéristiques. Le prédicteur est un PMC à une couche cachée de dimension 512 (avec une normalization des batchs uniquement dans la couche cachée) et une dimension de sortie de 2048.

Nous utilisons un pré-entraînement avec la descente de gradient stochastique sur 350 époques avec un taux d'apprentissage variant selon un recuit simulé basé sur la fonction cosinus ("Cosine Annealing learning rate scheduler" [9]), un momentum de 0,9, des batchs de données de taille 256, une dégradation des pondération ("weight decay") de 0,0001 et un taux d'apprentissage initial de 0,2 pour toutes les expérimentations avec ResNet18. Pour les expériences sur ResNet50, nous avons utilisé un taux d'apprentissage initial de 0,4 pour l'entropie croisée, de 0,2 pour SimSiam/SupSiam, et 0,3 pour HierSupSiam. Pour Hierarchical Supervised SimSiam (HierSupSiam), nous optons pour le choix de coefficients à une combinaison linéaire des deux fonctions de coûts des deux niveaux de CIFAR-100 avec $\alpha_1 = 0.95$ pour les classes de bas niveau et $\alpha_5 = 0,05$ pour les superclasses de CIFAR-100 (une meilleure exploration des coefficients des niveaux sera investiguée dans des travaux futurs).

Après l'entraînement, les deux PMC ne sont plus utilisés et seul l'encodeur de ResNet est gardé. Nous figeons les paramètres de l'encodeur et entraînons un classificateur linéaire dessus pour 100 épochs avec un taux d'apprentissage de 30 et un momentum de 0,9 sans dégradation des pondérations, pour mesurer la qualité du pré-entraînement.

Comme mentionné précédemment, la tâche de prétexte que nous essayons d'apprendre est difficile car nous entraînons un encodeur à être invariant à des augmentations ainsi qu'à la classe de l'étiquette. Avec les mêmes hyperparamètres ci-dessus, nous avons constaté que l'entraînement de SupSiam (et HierSupSiam) est instable ; une exécution peut converger vers un bon extracteur de caractéristiques, tandis qu'une autre peut diverger en un extracteur de caractéristiques inutile. Pour remédier à ce problème, nous effectuons une phase initiale de 10 époques avec la méthode SimSiam (sans étiquettes) afin que le modèle commence avec une bonne initialisation, puis nous passons à SupSiam ou HierSupSiam pour le reste des époques. Cette astuce simple s'est avérée efficace et stabilise le pré-entraînement.

Tableau 4 résume les résultats de nos expérimentations sur 350 époques (moyennés sur 3 runs). Pour la première colonne ResNet18, nous pouvons observer que SupSiam est plus performant (72.91±0.46) que SimSiam (63.76±0.13) puisqu'il incorpore l'information des étiquettes dans l'apprentissage de la représentation. HierSupSiam améliore encore légèrement les performances (72.98±0.11) et est proche de l'entropie croisée (73.59±0.25). Nous nous attendons à obtenir une amélioration

Méthode	ResNet18	ResNet50
Approches auto-supervisées		
SimSiam	63.76±0.13	66.65±1.38
Approches supervisées		
SupSiam	72.91±0.46	76.24±0.18
HierSupSiam	72.98±0.11	76.49±0.26
Cross-Entropy (our repro)	73.59±0.25	74.79±1.19
SupCon [8] (1000 époques)	-	76.5
Cross-Entropy [8] (500 époques)	-	75.3

TABLE 1 : Résultats sur CIFAR-100

plus importante si nous explorons un meilleur choix pour les coefficients de la fonction de coût de niveau hiérarchique α_l de Eq. 4.

Sur la deuxième colonne, nous résumons nos résultats avec ResNet50, nous obtenons un score de classification de 66.65±1.38 pour SimSiam, 76.24±0.18 pour SupSiam, et 76.49±0.26 pour HierSupSiam, dépassant l'entropie croisée (74.79±1.19). En comparant avec la méthode SupCon [8], nos résultats sont très proches alors que notre méthode n'est entraînée que pour 350 époques, alors que SCL est entraîné pendant 1000 époches. Enfin, nous constatons que notre approche bénéficie de la capacité des gros modèles tels que ResNet50 comme pour les approches AAS dans la littérature [2, 5].

5 Conclusion

Dans ce travail, nous avons proposé SupSiam, une extension de SimSiam dans un cadre supervisé pour l'apprentissage de représentation, ainsi que HierSupSiam, une extension de SupSiam dans le cas où une hiérarchie de classes est présente. Nous avons montré expérimentalement qu'avec un modèle de grande capacité, notre approche dépasse l'entropie croisée en terme de performances de classification. Comme perspectives, nous avons pour objectif de mieux explorer la hiérarchie dans l'espace latent en inférant par exemple un graphe d'hiérarchie à partir des données pendant l'entraînement. Nous explorerons également différents choix de coefficients de pondération des fonctions de coût de chaque niveau dans le cadre hiérarchique avec d'autres datasets avec hiérarchie. Nous étendrons également nos expériences sur d'autres domaines tels que l'audio, ainsi que le transfert d'apprentissage vers d'autres tâches.

Références

[1] Alsallakh BILAL, Amin JOURABLOO, Mao YE, Xiaoming LIU et Liu REN : Do convolutional neural networks learn class hierarchy ? *IEEE transactions on visualization and computer graphics*, 24(1):152–162, 2017.

[2] Ting CHEN, Simon KORNBLITH, Mohammad NOROUZI et Geoffrey HINTON : A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv :2002.05709*, 2020.

[3] Xinlei CHEN et Kaiming HE : Exploring simple siamese representation learning. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[4] Kaiming HE, Xinlei CHEN, Saining XIE, Yanghao LI, Piotr DOLLÁR et Ross GIRSHICK : Masked autoencoders are scalable vision learners. *arXiv :2111.06377*, 2021.

[5] Kaiming HE, Haoqi FAN, Yuxin WU, Saining XIE et Ross GIRSHICK : Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv :1911.05722*, 2019.

[6] Kaiming HE, Xiangyu ZHANG, Shaoqing REN et Jian SUN : Deep residual learning for image recognition. *In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[7] Ashish JAISWAL, Ashwin Ramesh BABU, Mohammad Zaki ZADEH, Debapriya BANERJEE et Fillia MAKEDON : A survey on contrastive self-supervised learning. *Technologies*, 9(1), 2021.

[8] Prannay KHOSLA, Piotr TETERWAK, Chen WANG, Aaron SARNA, Yonglong TIAN, Phillip ISOLA, Aaron MASCHINOT, Ce LIU et Dilip KRISHNAN : Supervised contrastive learning. *In* H. LAROCHELLE, M. RANZATO, R. HADSELL, M.F. BALCAN et H. LIN, éditeurs : *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020.

[9] Ilya LOSHCHILOV et Frank HUTTER : Sgdr : Stochastic gradient descent with warm restarts. *arXiv preprint arXiv :1608.03983*, 2016.

[10] Ishan MISRA et Laurens van der MAATEN : Self-supervised learning of pretext-invariant representations. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6707–6717, 2020.

[11] Alec RADFORD, Jong Wook KIM, Chris HALLACY, Aditya RAMESH, Gabriel GOH, Sandhini AGARWAL, Girish SASTRY, Amanda ASKELL, Pamela MISHKIN, Jack CLARK *et al.* : Learning transferable visual models from natural language supervision. *In International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[12] Jürgen SCHMIDHUBER : Deep learning in neural networks : An overview. *Neural Networks*, 61:85–117, 2015.

[13] Yusong WU, Ke CHEN, Tianyu ZHANG, Yuchen HUI, Taylor BERG-KIRKPATRICK et Shlomo DUBNOV : Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. *arXiv preprint arXiv :2211.06687*, 2022.

[14] Shu ZHANG, Ran XU, Caiming XIONG et Chetan RAMAIAH : Use all the labels : A hierarchical multi-label contrastive learning framework. *In CVPR*, 2022.