Optimisation des perturbations pour l'apprentissage contrastif

Camille RUPPLI[1,3], Pietro GORI[1], Roberto ARDON[3], Isabelle BLOCH[2,1]

[1]LTCI, Télécom Paris, Institut Polytechnique de Paris, Paris, France

[2]Sorbonne Université, CNRS, LIP6, Paris, France

[3]Incepto Medical, Paris, France

camille.ruppli@incepto-medical.com, pietro.gori@telecom-paris.fr
roberto.ardon@incepto-medical.com, isabelle.bloch@sorbonne-universite.fr

Résumé – Les méthodes actuelles d'apprentissage contrastif utilisent des perturbations aléatoires échantillonnées dans une grande liste de transformations avec des hyperparamètres fixes pour apprendre des invariances à partir d'une base de données non annotées. Dans la lignée de travaux précédents introduisant une petite quantité de supervision, nous proposons une méthode de recherche des perturbations optimales pour l'apprentissage contrastif en utilisant un réseau de perturbations. Avec peu de données annotées, notre méthode améliore les résultats de précision et converge plus rapidement. Contrairement aux travaux précédents, aucun modèle génératif n'est nécessaire pour l'optimisation des perturbations. Les images perturbées conservent les informations pertinentes pour résoudre la tâche de classification supervisée. Les expériences ont été réalisées sur 34000 coupes 2D d'images de résonance magnétique du cerveau. Avec 30% de données annotées, notre modèle atteint des performances similaires à celles d'un modèle entièrement supervisé avec 100% d'annotations.

Abstract – Current contrastive learning methods use random perturbations sampled from a large list of transformations with fixed hyperparameters to learn invariances from an unannotated database. Following previous works that introduce a small amount of supervision, we propose a framework to find optimal perturbations for contrastive learning using a perturbation network. Our method increases performances at low annotated data regime both in supervision accuracy and convergence speed. In contrast to previous work, no generative model is needed for perturbation optimization. Perturbed images keep relevant information to solve the supervised task, here classification. Experiments were performed on 34000 2D slices of brain Magnetic Resonance Images. With 30% of labeled data our model achieves similar performances to those of a fully supervised model with 100% of labels.

1 Introduction

En imagerie médicale, les données sont largement accessibles, mais les annotations sont moins nombreuses et coûteuses à obtenir. Des méthodes d'apprentissage auto-supervisé ont été développées pour tirer profit des données non annotées et augmenter les performances sur une tâche supervisée avec peu d'annotations. Parmi ces méthodes, celles par apprentissage contrastif [1, 2, 8] entraînent un encodeur à apprendre des invariances entre des perturbations des données non annotées.

Dans la plupart des travaux, les perturbations utilisées pour apprendre l'invariance sont échantillonnées de façon aléatoire à partir d'une liste donnée. Alors que de nombreux travaux étudient l'impact de la suppression de certaines perturbations sur la performance des tâches supervisées [2, 8], peu de recherches ont été menées sur l'optimisation des perturbations et de leurs hyper-paramètres. Certains auteurs [7, 10] se concentrent sur le rôle des perturbations mais sans optimisation explicite des perturbations. Dans [7], une analyse formelle de la composition des transformations est proposée pour sélectionner les perturbations admissibles, tandis que dans [10] des espaces latents spécifiques aux perturbations sont explorés.

En apprentissage supervisé, les auteurs de [3] proposent une optimisation sur les perturbations utilisées pour l'augmentation des données, mais une étape de pré-entraînement est nécessaire. Alors que la supervision est également introduite dans l'apprentissage contrastif dans [4, 11], peu d'auteurs l'utilisent pour influencer le choix des perturbations. Parmi eux, ceux de [9] introduisent un générateur de perturbations (reposant sur [5]) pour générer des images perturbées, dans de nouveaux espaces de couleurs, minimisant l'information mutuelle, tout en gardant suffisamment d'information pour la tâche supervisée. Comme ces perturbations sont limitées aux espaces de couleurs, elles ne sont pas pertinentes pour les images médicales.

Comme dans [9], ce travail utilise une petite quantité de supervision pour l'optimisation des perturbations. Nous introduisons une méthode d'optimisation de perturbations qui ne nécessite pas de pré-entraînement, et contrairement à [9], est applicable aussi bien aux images en couleurs qu'aux images en niveaux de gris.

Nos contributions sont les suivantes :

— Nous proposons un cadre différentiable semi-supervisé pour optimiser les perturbations de l'apprentissage contrastif, qui ne nécessite pas de pré-entraînement.

— Nous montrons que notre méthode trouve des perturbations pertinentes pour la tâche supervisée, et qui sont fa-

cilement interprétables.
— Nous montrons que notre méthode converge plus rapidement que d'autres algorithmes d'apprentissage contrastif, et présente de bonnes performances avec peu de données annotées.

2 Méthode proposée

Les méthodes d'apprentissage contrastif entraînent un encodeur à rapprocher les représentations, dans l'espace latent, des images d'une paire positive tout en éloignant celles des images des paires négatives. Les paires positives sont deux perturbations de la même image et les paires négatives sont des versions perturbées d'images différentes. Dans la plupart des méthodes telles que simCLR [2], durant l'entraînement, chaque image est perturbée deux fois avec deux perturbations différentes.

Les perturbations utilisées dans la plupart des méthodes sont choisies au hasard dans une liste fixe donnée. L'information mutuelle entre les images d'une paire positive n'est pas contrôlée, et, comme montré dans [9], cela peut être sous-optimal. Intuitivement, une paire positive avec une information mutuelle élevée n'apporte pas d'information supplémentaire à l'encodeur. Inversement, avec une petite quantité de supervision, elle doit contenir suffisamment d'informations pour la tâche supervisée objectif.

Nous nous concentrons ici sur les tâches de classification. Nous introduisons un réseau de perturbation M qui minimise l'information mutuelle entre les images d'une paire positive sans compromettre les performances de la tâche supervisée. Pour chaque image de la base d'entraînement, un réseau de neurones M génère un ensemble de paramètres Λ définissant la perturbation à appliquer T_{Λ_M}. Comme dans [2, 10], l'espace latent de l'encodeur f est optimisé en utilisant une tête de projection z dans un espace de dimension inférieure où une fonction de coût I_{NCE} est minimisée. La supervision est ajoutée sur l'espace latent en utilisant un classificateur linéaire p qui minimise une fonction de coût de classification $\mathcal{L}$. La figure 1 présente une vue schématique de notre approche (X représente une image de l'ensemble d'apprentissage et X_M sa version perturbée).

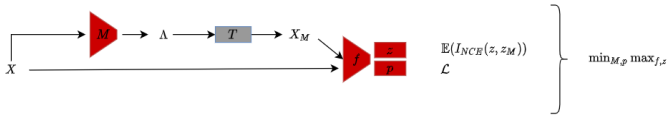

FIGURE 1 – Schéma de l'optimisation proposée (en rouge les éléments entraînables, en gris les éléments non entraînables).

2.1 Optimisation des perturbations

Nous considérons un ensemble fini de transformations d'intensité et géométriques appliquées aux images : symétries axiales horizontales et verticales, ajout de bruit gaussien, ajout de bruit additif, rotation et cadrage. Chaque transformation est paramétrée par un vecteur de paramètres (le vecteur de paramètres d'une rotation autour d'un point fixe ne contient que son angle). La fonction de perturbation à appliquer T_{Λ_M} est la composition de transformations dans un ordre fixé. Le réseau de perturbation M génère les paramètres de la fonction de perturbation. Nous proposons d'entraîner M à trouver la perturbation optimale pour le problème d'apprentissage contrastif semi-supervisé. Le réseau M associe une image à l'espace des vecteurs de paramètres, normalisés dans $[0, 1]$.

Soit λ_k le vecteur de paramètres pour une transformation donnée, alors la fonction de perturbation, notée T_{Λ_M}, est paramétrée par $\Lambda = [\lambda_1, \cdots, \lambda_K]$ (où $K = 7$ pour les transformations considérées).

La perturbation optimale pour notre problème d'apprentissage contrastif semi-supervisé est alors obtenue via M, optimisé afin de trouver le Λ_M^* optimal. Contrairement à [2], nous ne perturbons qu'une seule version des images. Avec ce choix, nos expériences produisent de meilleurs résultats. L'optimisation se déroule comme suit.

Optimisation du réseau de perturbations : (i) M génère un ensemble de vecteurs Λ_M définissant une perturbation T_{Λ_M}. Pour chaque image X, une version perturbée est générée : $X_M = T_{\Lambda_M}(X)$. (ii) Les ensembles de données perturbées et non perturbées passent par l'encodeur f suivi de la tête de projection z. (iii) Le gradient de la fonction de coût $-I_{NCE}$ (voir ci-dessous, équation 2) est calculé pour mettre à jour les poids du réseau M visant à minimiser l'information mutuelle. Notons que nous avons développé des expressions différentiables des perturbations utilisées.

Optimisation de l'encodeur : (i) à partir de l'étape d'optimisation précédente de M, une version perturbée des données est générée. Les projections latentes des données perturbées et non perturbées sont générées à l'aide de l'encodeur f et de la tête de projection z. (ii) Le gradient de la fonction de coût est calculé et les paramètres de f et z sont mis à jour. Cela permet de rapprocher les paires positives et d'éloigner les paires négatives.

Notre problème d'optimisation est défini comme suit :

$$\min_{M,p} \max_{f,z} \begin{cases} & \alpha_0 I_{NCE}\Big(z\big(f(X_M)\big), z\big(f(X)\big)\Big) \\ + & \alpha_1 \mathcal{L}\Big(p\big(f(X_M)\big), y\Big) \\ + & \alpha_2 \mathcal{L}\Big(p\big(f(X)\big), y\Big) \end{cases} \quad (1)$$

Les α_i sont des poids équilibrant les termes de la somme et les y sont les annotations pour la classification lorsqu'elles sont disponibles. La fonction de coût contrastive I_{NCE} est celle introduite dans [2] :

$$I_{NCE}(X_{Mi}, X_i) = -\sum_i \log\left(\frac{e^{sim(z(f(X_{Mi})),z(f(X_i)))}}{\sum_{j,j\neq i} e^{sim(z(f(X_{Mi})),z(f(X_j)))}}\right) \quad (2)$$

où l'indice i définit les paires positives et j les paires négatives. Enfin, sim est une mesure de similarité définie par $sim(x, x') = \frac{x^T x'}{\tau}$ où τ est un scalaire fixé. $\mathcal{L}$ est la fonction de coût d'entropie croisée binaire pour la contrainte supervisée.

2.2 Paramètres expérimentaux

Données Les expériences ont été réalisées sur le jeu de données BraTS [6]. Les coupes 2D le long de l'axe axial ont été obtenues à partir des volumes 3D. Seules les images comportant moins de 80% de pixels noirs ont été conservées. Nous avons ainsi obtenu 34000 images. Nous avons étudié la tâche supervisée de classification de présence de tumeurs (classification binaire, présente/non présente). Nous avons sélectionné aléatoirement quatre jeux de test de 1000 images.

Implémentation Pour chaque expérience, l'encodeur f est un réseau convolutif composé de quatre blocs de convolution avec deux couches de convolution dans chaque bloc. Le réseau M est un réseau convolutif composé de deux blocs de convolution avec une couche convolutive. La tête de projection z est un perceptron à deux couches comme proposé dans [2]. Le modèle est entraîné avec des ensembles de taille 32 durant 100 époques. Les taux d'apprentissage sont fixés à 10^{-2}, 10^{-3} et 10^{-4} pour M, M avec supervision et f, respectivement.

Nous avons réalisé des expériences avec $\alpha_0 \in \{0, 1, 1, 0\}$ et $(\alpha_1, \alpha_2) \in \{10, 1, 0\}^2$ pour les optimisations supervisées. Pour les expériences sans contrainte de supervision, α_0 est fixé à 1.

Les expériences entièrement supervisées décrites dans la section 3 sont optimisées avec la même architecture d'encodeur et une couche dense suivie d'une fonction d'activation sigmoïde pour la tâche de classification. Pour les expériences entièrement supervisées, nous avons utilisé un taux d'apprentissage de 10^{-4}.

Infrastructure informatique Les optimisations ont été exécutées sur des cartes Tesla NVIDIA V100.

2.3 Évaluation linéaire

Pour évaluer la qualité de la représentation apprise par l'encodeur, nous suivons le protocole d'évaluation linéaire utilisé dans la littérature [2, 8, 9]. L'encodeur est gelé avec les poids appris durant l'optimisation. Une couche linéaire est ajoutée, après avoir retiré la tête de projection z, et entraînée avec les données de test annotées et non utilisées dans la phase d'entraînement précédente. Cela signifie que nous projetons d'abord les données de test dans l'espace latent du modèle figé, puis nous estimons le modèle linéaire le plus discriminant. Le raisonnement sous-jacent est le suivant : une bonne représentation doit rendre les classes des données de test linéairement séparables.

3 Résultats et discussion

Afin d'évaluer l'impact de chaque terme de l'équation 1, nous avons optimisé les approches suivantes :
Aléatoire (sans M, sans supervision) : chaque image est perturbée avec des paramètres générés par une distribution uniforme : $\Lambda = \mathcal{U}\left([0, 1]^7\right)$, et $\alpha_1 = \alpha_2 = 0$.
Aléatoire avec supervision (sans M, avec supervision) : nous ajoutons la contrainte de supervision à la stratégie aléatoire. Nous fixons $\alpha_i = 1, \forall i$.
Auto-supervisé (avec M, sans supervision) : en fixant α_1 et α_2 à 0, nous optimisons l'équation 1.
Auto-supervisé avec contrainte de supervision (avec M et supervision) : en fixant $\alpha_0 = 0, 1$, $\alpha_1 = \alpha_2 = 10$, nous optimisons l'équation 1.

Évaluation linéaire - Une évaluation linéaire a été effectuée pour les quatre stratégies d'optimisation avec les données de test. Les performances ont été évaluées avec les poids obtenus à différentes époques. Nous cherchons à évaluer si notre méthode produit de meilleures représentations plus tôt lors de l'entraînement. Nous calculons la moyenne et l'écart-type des performances sur trois ensembles de test différents.

Nous avons également entraîné l'encodeur pour la tâche de classification dans un cadre supervisé avec une quantité croissante de données annotées. Pour l'entraînement supervisé, nous avons augmenté les données en composant aléatoirement les transformations testées. Chaque transformation avait une probabilité de 0,5 d'être tirée. Nous avons effectué l'évaluation linéaire sur l'encodeur gelé et les données de test et reportons les valeurs d'AUC (aire sous la courbe ROC) obtenues sous forme de lignes horizontales dans la figure 2. La figure 2 montre également les résultats de l'évaluation linéaire de l'optimisation simCLR classique comme dans [2] où une seule image est perturbée par une composition aléatoire des transformations testées. Comme pour les expériences entièrement supervisées, chaque transformation avait une probabilité de 0,5 d'être appliquée.

La figure 2 montre que la stratégie auto-supervisée avec contrainte de supervision permet d'obtenir de meilleures représentations beaucoup plus rapidement pendant l'entraînement. Elle montre également que l'optimisation avec seulement 30% des données annotées nous permet d'atteindre pratiquement la même qualité de représentation que l'entraînement entièrement supervisé avec 100% d'annotations. La contrainte de supervision joue un rôle important car le générateur de perturbations aléatoires avec supervision converge vers les résultats de M après 20 époques.

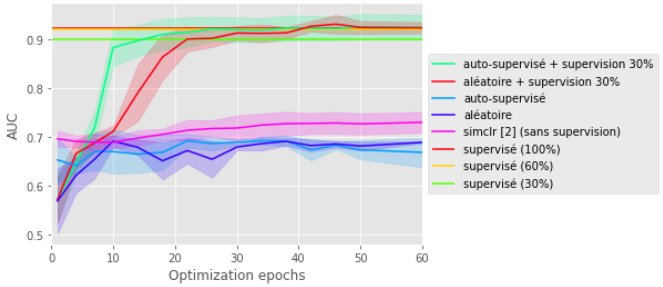

FIGURE 2 – Résultat de l'évaluation linéaire pour les différentes méthodes (variabilité sur le jeu de test).

Mesures d'évaluation - Les images perturbées doivent être

aussi différentes que possible, afin de réduire l'information mutuelle, mais doivent également contenir des informations pertinentes pour la tâche supervisée objectif. Pour la tâche proposée, à savoir la classification de la présence de tumeurs cérébrales, nous faisons l'hypothèse que les informations pertinentes se trouvent à l'intérieur et autour de la tumeur. Nous proposons donc de vérifier dans quelle mesure cette zone a été préservée après la perturbation. Dans le jeu de données BraTS, les masques de segmentation des tumeurs sont disponibles. Nous utilisons la mesure d'évaluation suivante : le pourcentage d'erreur de la taille du masque entre les masques de segmentation d'origine et perturbé (seuillé à 0,8), défini comme suit :

$$\frac{|size(T_{\Lambda^*}(mask)) - size(mask)|}{size(mask)}$$

.

La table 1 montre que l'optimisation des paramètres de transformation avec M préserve davantage de pixels de tumeur après les perturbations. La figure 3 montre que cela se traduit par une meilleure compréhension et interprétabilité de l'importance de la transformation par rapport à la tâche de classification.

TABLE 1 – Évaluation moyenne sur les trois jeux de test, écart type entre parenthèses.

	Erreur sur le masque
M optim. + supervision	**0,52** (0,04)
M optim. sans supervision	0,95 (0,00)
M aléatoire	0,95 (0,00)

Lors de l'optimisation sans supervision, pour minimiser l'information mutuelle, le réseau M peut générer des perturbations qui créent des images très différentes des images non perturbées mais qui ne contiennent pas d'informations pertinentes pour la tâche de classification. Sans la contrainte de supervision, l'extraction optimale génère une image avec une majorité de valeurs nulles, inutiles pour l'entraînement supervisé. La figure 3 montre un exemple de perturbations trouvées lors de différentes optimisations. Nous voyons que la contrainte de supervision aide à générer des images pertinentes qui conservent les pixels de la tumeur.

4 Conclusion

Nous avons proposé une méthode pour optimiser les perturbations utilisées dans l'apprentissage contrastif avec une certaine quantité de supervision. Par rapport aux travaux précédents, notre méthode ne nécessite pas d'entraînement supplémentaire. Nous trouvons des perturbations interprétables et obtenons de meilleures représentations, surtout pendant les premières époques d'optimisation. Les travaux futurs permettront d'étudier d'autres fonctions de coût et d'optimiser M pour générer deux versions d'images perturbées.

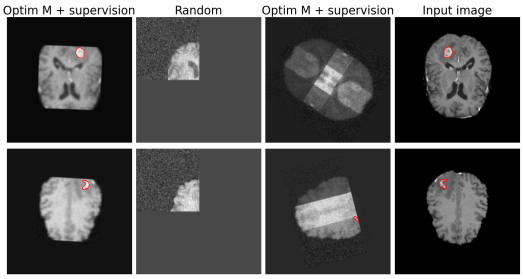

FIGURE 3 – Exemple de perturbations générées avec les différentes optimisations (le contour rouge indique la tumeur).

Références

[1] Krishna Chaitanya et al. Contrastive learning of global and local features for medical image segmentation with limited annotations. In *NeurIPS*, volume 33, pages 12546–12558, 2020.

[2] Ting Chen et al. A simple framework for contrastive learning of visual representations. In *37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

[3] Ekin D. Cubuk et al. AutoAugment : Learning Augmentation Strategies From Data. In *CVPR*, pages 113–123, 2019.

[4] Prannay Khosla et al. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673, 2020.

[5] Durk P Kingma and Prafulla Dhariwal. Glow : Generative Flow with Invertible 1x1 Convolutions. In *NeurIPS*, volume 31, 2018.

[6] Bjoern H. Menze et al. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10) :1993–2024, 2015.

[7] Mandela Patrick et al. On compositions of transformations in contrastive self-supervised learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9577–9587, 2021.

[8] Alexis Perakis et al. Contrastive Learning of Single-Cell Phenotypic Representations for Treatment Classification. In *MLMI - MICCAI*, pages 565–575, 2021.

[9] Yonglong Tian et al. What Makes for Good Views for Contrastive Learning ? In *NeurIPS*, 2020.

[10] Tete Xiao et al. What should not be contrastive in contrastive learning. In *International Conference on Learning Representations*, 2021.

[11] Xiangyun Zhao et al. Contrastive learning for label efficient semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10623–10633, 2021.