École Nationale Supérieure d'Informatique pour l'Industrie et l'Entreprise

Université Paris-Saclay

Rapport du projet de deep learning

Noisy Student Training

Étudiant : **Etienne Gaucher**

Matière : **Deep learning**

Professeur : **Blaise Hanczar**

Sommaire

1 Introduction

La classification d'images a été et est toujours un challenge. Pour réussir à affecter une étiquette à une image, il existe un très grand nombre de méthodes supervisées et non supervisées, sans compter celles qui se trouvent entre ces 2 types.
La difficulté du projet réside dans le fait que nous possédons très peu d'images labélisées mais beaucoup d'images non-labélisées. Bien que nous ne connaissons pas les labels d'un grand nombre d'images, il faudrait d'une certaine manière réussir à en tirer profit.
Dans ce projet, nous allons nous inspirer d'un article publié en 2019 et intitulé *"Self-training with Noisy Student improves ImageNet classification"* où les auteurs présentent une méthode semi-supervisée nommée Noisy Student Training basée sur une structure Professeur-Élève. Ils appliquent cette méthode sur le modèle EfficientNet. Une fois la méthode implémentée, ils ont testé ses performances en comparaison d'autres modèles de l'état de l'art sur le jeu de données ImageNet. Les résultats étant très convaincants, nous allons donc l'implémenter et la tester sur le jeu de données MNIST composé de chiffres manuscrits en blanc sur fond noir. Le jeu de données MNIST contient un jeu de données d'apprentissage de 60 000 images et un jeu de données de test avec 10 000 images. On divise tout d'abord le jeu d'apprentissage en 2 ensembles :

- un ensemble d'entraînement de 50 000 images
- un ensemble de validation de 10 000 images

Le jeu de validation nous permettra de vérifier l'efficacité de notre modèle lors de l'entraînement. Pour rappel, la contrainte de notre projet est la connaissance d'uniquement 100 labels. L'ensemble d'entraînement est donc divisé en 2 sous-ensembles :

- un ensemble d'entraînement de 49 900 images dont on suppose qu'on ne connaît pas les labels
- un ensemble d'entraînement de 100 images dont on connaît les labels

Puisque le nombre d'images labelisées est faible, la méthode Noisy Student Training nous permettra d'utiliser les images non-labelisées pour la prédiction des étiquettes de l'ensemble de test.

2 La méthode Noisy Student Training

La méthode Noisy Student Training est une méthode semi-supervisée. En effet, le modèle commence avec un apprentissage supervisé des 100 images étiquetées. Ce réseau est appelé professeur puisqu'il va ensuite prédire des pseudo-étiquettes aux 49 900 images du deuxième sous-ensemble du jeu d'entraînement. Ensuite, on rassemble les images étiquetées et pseudo-étiquetées dans un même jeu de données, et on entraîne un nouveau réseau sur celui-ci. Ce réseau est appelé élève. De l'apprentissage de cet ensemble, il devient professeur et va prédire de nouvelles pseudo-étiquettes aux 49 900 images dont on ne connaît pas les étiquettes. Ainsi de suite, des enchaînements professeur-élève vont se succéder pour tenter de prédire correctement les étiquettes des images.
Cependant, un tel réseau ne permettrait pas vraiment d'améliorer les performances puisque l'élève apprendrait exactement les connaissances du professeur. L'enchaînement professeur-élève convergerait uniquement dans une même direction. C'est pourquoi la méthode repose sur le fait de bruiter

l'apprentissage de l'élève. En ajoutant du bruit lors de l'apprentissage, cela force l'élève à avoir plus de robustesse, à généraliser de façon plus profonde que le professeur. Le bruit a pour but de "lisser" les frontières de décision. L'article indique les 3 types de bruit ajoutés :

- la **Data augmentation** qui contraint l'élève à apprendre plus profondément que le professeur car il a des images plus difficiles et moins habituelles
- du **Dropout** et du **Stochastic depth** qui contraignent l'élève à être plus robuste

Pour la création de la base d'apprentissage de l'étudiant, les auteurs ont de plus ajouté une étape de **filtrage** et de **stratification**. L'idée du filtrage est de ne pas utiliser toutes les images de l'ensemble non étiqueté mais d'en sélectionner seulement une partie. Pour cela, ils se basent sur la probabilité de sortie pour la pseudo-étiquette et choisissent les images dont la probabilité fait partie des plus élevées. En plus de cela, pour éviter que l'ensemble ne soit composé principalement d'images d'un même chiffre, ils stratifient la sélection pour s'assurer que chaque chiffre est présent avec de bonnes proportions dans l'ensemble d'apprentissage.

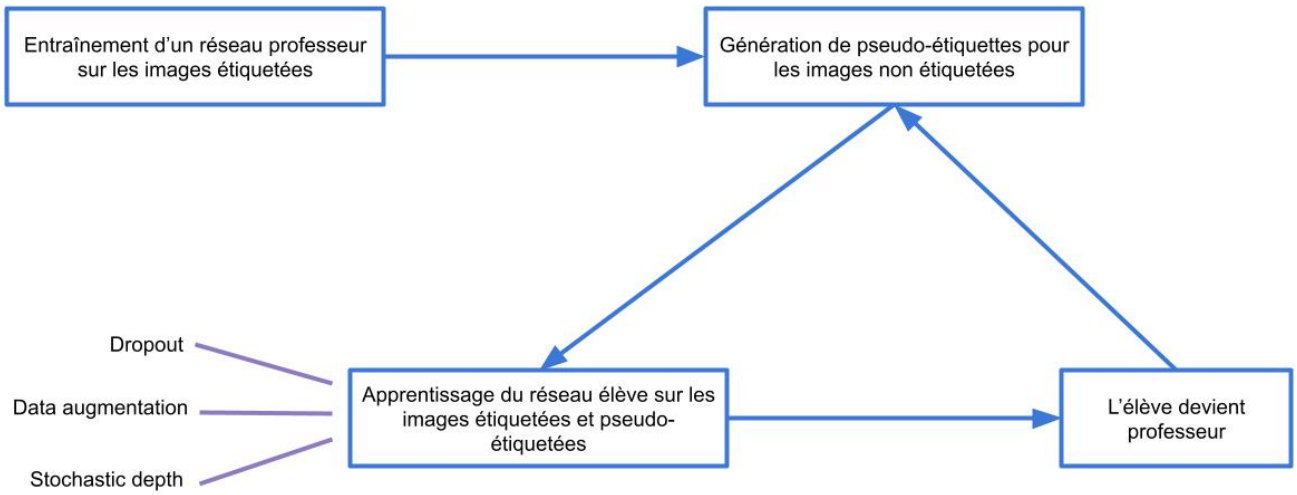

Figure 1: Méthode Noisy Student Training

3 Définition du modèle de base

Bien que les auteurs de l'article utilisent le modèle EfficientNet, il s'avère inapproprié dans notre cas. Nous possédons beaucoup moins d'images, les images sont petites (28×28 pixels) et en noir et blanc (donc un seul channel). De plus, si on utilisait le modèle EfficientNet, l'apprentissage serait très long.
Pour notre modèle de base, nous avons choisi d'implémenter un modèle beaucoup plus simple avec 4 couches : 2 couches de convolution et 2 couches de neurones complètement connectés.

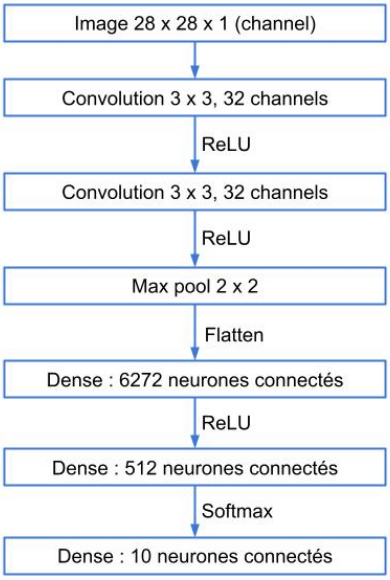

Figure 2: Réseau de base

4 Implémentation du Early Stopping

La méthode Early Stopping est une forme de régularisation utilisée pour éviter l'overfitting sur le jeu de données d'entraînement. Cette dernière suit la validation loss : si la validation loss cesse de diminuer pendant plusieurs époques consécutives, l'entraînement s'arrête.

Notre fonction procède de la manière suivante : elle sauvegarde le modèle chaque fois que la validation loss diminue. Ainsi, le modèle avec la validation loss la plus faible est enregistré. L'argument *patience* définit le nombre d'époques que nous voulons attendre après la dernière amélioration de la validation loss avant d'interrompre la boucle d'apprentissage.

5 Implémentation de la méthode

Nous avons implémenté la méthode progressivement : nous avons ajouté à la suite le DropOut puis la Data Augmentation. A chaque fois, nous avons réalisé des simulations pour comparer les performances. Nous rapportons les résultats de la façon suivante :

- la baseline correspond à l'accuracy sur les 10 000 images du jeu de données de test avant d'entrer dans la boucle Professeur-Elève. On utilise donc le réseau entraîné avec les 100 images étiquetées.
- la meilleure amélioration correspond à l'accuracy la plus élevée sur les 10 000 images du jeu de données de test après avoir effectué un certain nombre de fois la boucle Professeur-Elève. Nous avons implémenté une sorte de Early Stopping qui enregistre le modèle élève avec la validation loss la plus faible et stoppe les itérations Professeur-Elève lorsque la validation loss de l'élève ne diminue plus. Le nombre de boucle Professeur-Elève effectué avant d'obtenir le meilleur élève est également indiqué.

- on calcule la différence entre la baseline et la meilleure amélioration.

5.1 Elève sans bruit

Tout d'abord, vérifions les performances du modèle lorsqu'aucun bruit n'est ajouté à l'élève. Nous avons effectué 5 simulations dont les résultats sont présentés ci-dessous.

Numéro de simulation	1	2	3	4	5
Baseline (accuracy)	0.720	0.810	0.760	0.795	0.766
Meilleure amélioration	0.725	0.817	0.774	0.797	0.756
Différence	0.5 %	0.7 %	1.4 %	0.2 %	-1.0 %
Nombre boucle Professeur-Elève	1	1	4	1	1

Il semble évident que cette méthode n'est pas tellement plus efficace que le modèle baseline correspondant au modèle issu d'un apprentissage supervisé sur les 100 images étiquetées. Nous allons donc regarder les résultats lorsque l'apprentissage de l'élève est bruité.

5.2 Elève avec Dropout

Le premier bruit ajouté lors de l'apprentissage de l'élève est le **DropOut**. Dans l'article, les auteurs conseillent d'appliquer le DropOut sur la couche finale avec une probabilité de 0.5 ce que nous avons suivi. L'exécution du code est d'environ 20 minutes. Nous avons réalisé 5 simulations et avons obtenu les résultats suivants :

Numéro de simulation	1	2	3	4	5
Baseline (accuracy)	0.772	0.779	0.738	0.822	0.785
Meilleure amélioration	0.806	0.766	0.740	0.84	0.80
Différence	3.4 %	-1.3 %	0.2 %	1.8 %	1.5 %
Nombre boucle Professeur-Elève	3	1	1	2	1

Comme nous pouvons le voir, le DropOut ne permet pas d'obtenir tout seul une amélioration notable des performances. La baseline est très légèrement améliorée, parfois diminuée. Le DropOut doit donc être combiné avec d'autres types de bruit.

5.3 Elève avec DropOut et RandAugment

Le **RandAugment** est une technique de data augmentation que les auteurs ont choisi pour aider à améliorer l'invariance translationnelle de leur réseau de neurones. Cette augmentation est composée en temps normal de transformations comme des fluctuations de couleurs, des flous gaussiens,

des saturations (etc) ainsi que de transformations plus traditionnelles comme des random crops, resize ou rotation.

Cette méthode comporte deux paramètres :

- n, qui indique le nombre de transformations d'augmentation appliqué
- m, la force de ces augmentations

Les auteurs de l'article ont utilisé le couple ($n = 2$, $m = 27$). Nous avons pris les mêmes paramètres. L'élève est donc bruité à la fois par le DropOut et le RandAugment. L'utilisation de RandAugment augmente de façon considérable le temps d'exécution. L'exécution du code était en moyenne de 2 heures pour 3 à 10 boucles Professeur-Elève. Nous avons donc réalisé 3 simulations.

Numéro de simulation	1	2	3
Baseline (accuracy)	0.782	0.753	0.791
Meilleure amélioration	0.884	0.882	0.930
Différence	10.2 %	12.9 %	13.9 %
Nombre boucle Professeur-Elève	5	8	4

Les résultats obtenus montrent une vraie amélioration de la précision en associant RandAugment et DropOut. A chaque fois, la baseline est améliorée d'au moins 10 % ce qui est très satisfaisant. Cela prouve que la méthode Noisy Student Training permet d'améliorer l'accuracy en utilisant des images non étiquetées.

6 Résultats

On compare les résultats d'un élève sans bruit (premier tableau) et d'un élève avec RandAugment et DropOut (second tableau).

Numéro de simulation	1	2	3	4	5
Baseline (accuracy)	0.720	0.810	0.760	0.795	0.766
Meilleure amélioration	0.725	0.817	0.774	0.797	0.756
Différence	0.5 %	0.7 %	1.4 %	0.2 %	-1.0 %
Nombre boucle Professeur-Elève	1	1	4	1	1

Numéro de simulation	1	2	3
Baseline (accuracy)	0.782	0.753	0.791
Meilleure amélioration	0.884	0.882	0.930
Différence	10.2 %	12.9 %	13.9 %
Nombre boucle Professeur-Elève	5	8	4

On voit bien qu'un élève non bruité n'améliore quasiment pas la baseline, il n'y a pas vraiment d'amélioration. Comme évoqué dans l'article, la méthode repose vraiment sur le fait de bruiter l'élève.

Les résultats de l'implémentation de Noisy Student Training sont très satisfaisants. L'accuracy augmente d'au moins 10 % en associant DropOut et RandAugment. L'optimisation des paramètres permettrait peut-être d'aller encore plus loin. La méthode Noisy Student Training prend en compte les images non étiquetées et a un fort potentiel.

7 Améliorations envisageables

Par manque de temps, nous avons choisi de renoncer à certaines tâches. Par exemple, nous aurions aimé implémenter les fonctions de filtrage et stratification ou modifier notre modèle de base en ajoutant ou changeant les couches. De même, nous aurions voulu tester si l'association DropOut - RandAugment est plus bénéfique que RandAugment tout seul. Enfin, nous pourrions utiliser un réseau résiduel pour observer l'influence du Stochastic Depth.