# Deep Learning
# Virtual Adersarial Training

Mohamed Niang
Fernanda Tchouacheu
Sokhna Penda Touré

# Sommaire

# 1 Introduction

L'apprentissage semi-supervisé traite du problème à savoir comment tirer parti d'une énorme quantité de données non classifiées, pour effectuer une classification dans des situations où, généralement, il y a peu de données étiquetées. Dans ce projet, nous fessons face à un probleme d'apprentissage semi-supervisé portant sur les données MNIST. Le but ici est de faire une prédiction sur les données MNIST en utilisant uniquement 100 labels. Pour ce faire, nous allons utiliser une méthode de régularisation appelé Virtual Adversarial Training. Il s'agit d'une méthode de régularisation basée sur la perte virtuelle d'adversaires : une nouvelle mesure de l'uniformité locale de la distribution conditionnelle de l'étiquette donnée en entrée. La perte virtuelle antagoniste est définie comme la robustesse de la distribution conditionnelle de l'étiquette autour de chaque point de données d'entrée par rapport aux perturbations locales. Contrairement à la formation contradictoire, cette méthode définit l'orientation contradictoire sans information sur l'étiquette et est donc applicable à l'apprentissage semi-supervisé. Dans ce qui suit, après une présentation plus approfondie de la méthode, nous allons effectuer un apprentissage supervisé sur nos données. Nous allons par la suite utiliser le VAT pour de l'apprentissage semi supervisé sur ces mêmes données et ainsi pouvoir comparer nos différents résultats et tirer des conclusions sur l'efficacité du VAT.

# 2 Virtual Adversarial Traning

## 2.1 Méthodologie

L'approche VAT est une technique d'entraînement permettant de régulariser un modèle profond en utilisant des données sans annotation. Le principe de cette technique consiste à optimiser un bruit adapté qui une fois ajouté aux images non annotées les rendent plus difficiles à classer pour le modèle, car plus éloignées du centre de distribution de la classe. Le bruit virtuellement adversaire est calculé, pour chaque image, par rétropropagation de la fonction non supervisée des distributions des scores entre l'image bruitée et l'image sans bruit. Cette distance correspond à la prédiction du réseau (après softmax) entre l'image d'entrée $x$ et $x + \xi_{VAT}d$ , d étant un bruit issu d'une distribution gaussienne permettant le tirage de variables indépendantes et identiquement distribuées.La variable $\xi_{VAT}$ est un hyperparamètre généralement très petit ($\xi_{VAT} = 1e - 6$). La divergence de **Kullback-Leibler** permet de mesurer la dissimilarité entre deux lois de probabilité.Dans notre cas, la fonction de coût non supervisée calcule la distance entre les distributions issues du réseau à partir des images avec et sans bruit adversaire.

$$KLD(p, q) = D[p(y|x), p(y|x + r)] := \sum_{y \in Q} p(y|x) log(\frac{p(y|x)}{p(y|x + r)})$$

La distance ainsi obtenue est normalisée (en la divisant par sa norme L2) pour ensuite être retropropagé dans le réseau. Ce calcul nous permet de savoir comment modifier le bruit en entrée pour maximiser la fonction de coût et ainsi produire un exemple particulièrement difficile pour le réseau. Ce bruit adversaire(pertubations), noté $r_{adv}$, est pondéré par un deuxième hyperparamètre $\xi_{VAT}$ . Les auteurs indiquent qu'empiriquement une seule itération est suffisante pour obtenir un exemple adversaire efficace. Les courbes présentées dans **la section 4.3 de l'article de Miyato et al. (2018)** montrent que le bruit généré reste relativement le même après la première itération.
À chaque itération, la partie du batch possédant des labels est utilisée de manière classique pour calculer la fonction de **coût supervisée**. Toutes les images du batch sont utilisées pour générer des exemples adversaires afin de calculer la fonction de **coût non supervisée** (une deuxième divergence de Kullback-Leibler dans notre étude).

## 2.2 Différentes étapes

Les étapes du VAT sont les suivantes :
1) commencez par choisir M echantillon de $x^n(n = 1, ....., M)$ du jeu de donnée d'entrée x ;
2) transformez x en ajoutant une petite perturbation r , d'où le point de données transformé sera $T(x) = x + r$ ;
3) la perturbation r doit être dans le sens opposé - la sortie du modèle de l'entrée perturbée $T(x)$ doit être différente de la sortie de l'entrée non perturbée. En particulier, la divergence KL entre les deux distributions de sortie devrait être maximale, tout en garantissant que la norme L2 de r soit petite. De toutes les perturbations r, soit $r\_vadv$ la perturbation dans le sens contradictoire.

$$\Delta_{KL}(r, x^{(n)}, \theta) = D[q(y|x^{(n)}), p(y|x^{(n)} + r_{vadv}, \theta)]$$

avec

$$r_{adv} := argmax_{r,||r||\leq\epsilon} D[q(y|x^{(n)}), p(y|x^{(n)} + r, \theta)$$

4) on retourne :

$$\nabla_\theta(\frac{1}{M}\sum_{i=1}^{M} D[q(y|x^{(n)}), p(y|x^{(n)} + r^{(n)}_{vadv}, \theta)])$$

Après avoir trouvé la perturbation contradictoire et l'entrée transformée, mettez à jour les poids du modèle de sorte que la divergence KL soit minimisée. Cela rendrait le modèle robuste face à différentes perturbations. La perte suivante est minimisée par la descente de gradient :

$$LDS(x^{(n)}, \theta) = -\Delta_{KL}(r_{adv}, x^{(n)}, \theta)$$

Pendant l'entraînement virtuel contradictoire, le modèle devient plus robuste contre différentes perturbations d'entrée. À mesure que le modèle devient plus robuste, il devient plus difficile de générer des perturbations et une baisse de la perte est observée.
Cette méthode peut être considérée comme similaire aux réseaux contradictoires génératifs.

## 2.3 Méthodes d'approximation rapide pour $r_{vadv}$ et la dérivée de la fonction objective

Une fois la perturbation virtuelle accusatoire $r_{vadv}$ calculée, l'évaluation de $LDS(x, \theta)$ devient simplement le calcul de la divergence D entre les distributions de sortie $p(y|x_*, \hat{\theta})$ et $p(y|x_* + r_{vadv}, \theta)$.
Cependant, l'évaluation de $r_{vadv}$ ne peut pas être effectué avec l'approximation linéaire comme dans la formation contradictoire initiale parce que le gradient de $D[p(y|x_*, \hat{\theta}), p(y|x_* + r_{vadv}, \theta)$ par rapport à r est toujours 0 à r = 0. Dans ce qui suit, nous proposons une méthode efficace du calcul de rvadv, pour lequel il n'y a pas forme fermée évidente.
Pour simplifier, nous notons :

$$D[p(y|x_*, \hat{\theta}), p(y|x_* + r_{vadv}, \theta) = D(r, x_*, \theta)$$

Nous supposons que $p(y|x_*, \theta)$ est deux fois différentiable par rapport à $\theta$ et $x$ presque partout. Car $D(r, x_*, \hat{\theta})$ prend la valeur minimale à r = 0, l'hypothèse de différentiabilité dicte que sa première dérivée est :

$$\nabla_r D(r, x_*, \hat{\theta})|_{r=0}$$

L'approximation D de taylor du second ordre est :

$$D(r, x_*, \hat{\theta}) \approx \frac{1}{2} r^T H(x, \hat{\theta}) r$$

où $H(x, \hat{\theta})$ est la matrice hessienne donné par :

$$H(x, \hat{\theta}) := \nabla \nabla_r D(r, x_*, \hat{\theta})|_{r=0}$$

Sous cette approximation, $r_{vadv}$ apparaît comme le premier vecteur propre dominant $u(x, \hat{\theta})$ de $H(x, \hat{\theta})$ avec magnitude $\epsilon$ :

$$r_{vadv} \approx argmax(r^T H(x, \hat{\theta}) r; ||r||_2 \leq \epsilon)$$

### 2.4 Particularités et avantages

Dans la formation contradictoire, des étiquettes sont également utilisées pour générer les perturbations contradictoires. La perturbation est générée de telle sorte que l'étiquette prédite y du classificateur devient différente de l'étiquette réelle y .
Dans l'entraînement virtuel contradictoire, aucune information d'étiquette n'est utilisée et la perturbation est générée en utilisant uniquement les sorties du modèle. La perturbation est générée de telle sorte que la sortie de l'entrée perturbée est différente de la sortie du modèle de l'entrée d'origine (par opposition à l'étiquette de vérité).
L'un des avantages de cette méthode d'entraînement est qu'elle peut s'appliquer directement aux architectures profondes et possède peu d'hyperparamètres (la recherche est focalisée sur $\xi_{VAT}$). Une variante est aussi proposée par les auteurs en ajoutant un terme d'entropie à la fonction de coût total. Cet ajout contraint le modèle à fournir des prédictions plus catégoriques (l'entropie est maximale lorsque la probabilité est égale entre les différentes classes). Cette régularisation par l'entropie a montré ses preuves dans d'autres approches et permet généralement d'obtenir de meilleures performances.

## 3 Implémentation

Pour ce projet, nous avons effectué deux types d'apprentissage :

- Un apprentissage supervisé sur les données MNIST en considérant un échantillon de train de taille 100.
- Un apprentissage semi-supervisé avec 100 données labélisés et 59900 non labélisés en utilisant le VAT.

L'objectif ici est de voir l'apport qu'a le VAT et l'utilisation de données non labélisés sur les performances du model. En ce qui concerne l'apprentissage supervisé, nous obtenons un accuracy de l'ordre de 67,56%.

```
score1 = model1.evaluate(X_test, y_test, verbose=0)
print('Test loss:', score1[0])
print('Test accuracy:', score1[1])
```

```
Test loss: 1.0817532661437987
Test accuracy: 0.6756
```

Pour l'apprentissage semi-supervisé avec le VAT nous parvenons à avoir un accuracy de 92,15%. Nous voyons ainsi l'apport en terme de performance qu'aura eu

l'apprentissage semi-supervisé avec utilisation du VAT sur notre modèle.

```
score2 = model2.evaluate(X_test, y_test)
print('Test loss:', score2[0])
print('Test accuracy:', score2[1])
```

```
10000/10000 [==============================] - 1s 69us/step
Test loss: 0.5893054841025295
Test accuracy: 0.9215
```

# 4 Conclusion

Nous avons pu voir que les résultats obtenus en apprentissage semi-supervisé avec le VAT sur les données MNIST sont assez concluant. En effet, nous constatons clairement la valeur ajoutée de cette démarche.

Le Virtual Adversarial Training présente aussi d'autre nombreux avantages dans son utilisation :

- Peut être mis en oeuvre avec un faible coût de calcul ;
- Possède unique deux hyperparamètres ;
- Applicable à de nombreux modèles indépendamment de leur architecture.

Comme indiqué dans l'article étudié, une demarche à venir est voir comment combiner le VAT à des modèles génératifs pour des résultats encore plus performants.

# Références


[1] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii *Virtual Adversarial Training : A Regularization Method for Supervised and Semi-Supervised Learning.*

[2] Divam Gupta *An Introduction to Virtual Adversarial Training* https ://divamgupta.com/unsupervised-learning/semi-supervised-learning/2019/05/31/introduction-to-virtual-adversarial-training.html

[3] Takeru Miyato https ://takerum.github.io