Deep Learning: Virtual Adversarial Training

Mohamed Niang
Fernanda Tchouacheu
Sokhna Penda Touré

MSc in Data Science: University of Paris-Saclay

Le 6 Janvier 2020

Sommaire

Introduction

Article étudié

Virtual Adversarial Training : A Regularization Method for Supervised and Semi-Supervised Learning

Auteurs

Takeru Miyato,Shin-ichi Maeda, Masanori Koyama et Shin Ishii

Objectif du projet

- Apprentissage semi-supervisé sur les données MNIST avec 100 labels
- Utilisation du Virtual Adversarial Training
- Comparaison des différents résultats obtenus

L'apprentissage semi-supervisé

Apprentissage semi-supervisé pour améliorer les performances en combinant les données avec labels (peu) et sans labels (beaucoup)

- Classification semi-supervisée : entraîner sur des données avec labels et exploiter les données (beaucoup) sans labels
- Clustering semi-supervisé : clustering des données sans labels en s'aidant des données avec labels

Hypothèse de base pour la plupart des algorithmes d'apprentissage semi-supervisés

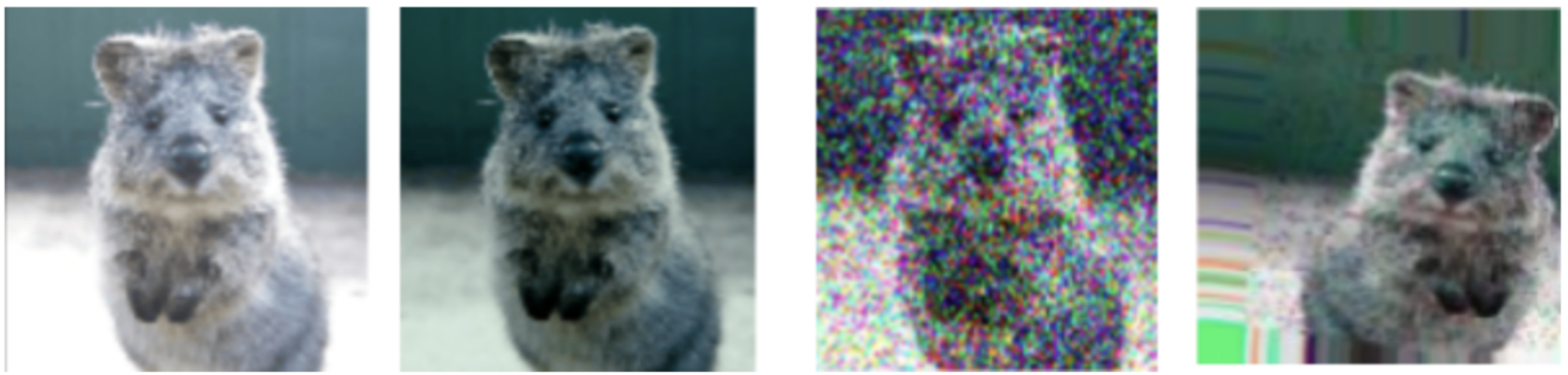
- Points proches ont probablement le même label de classe
- Deux points qui sont connectés par un chemin traversant des régions de forte densités doivent avoir le même label

Local Distributional Smoothness(LDS)

Définition

Il s'agit de la régularité de la distribution de sortie du modèle, par rapport à l'entrée. Un moyen facile pour la régularisation LDS est de générer des points de données artificiels en appliquant de petites perturbations aléatoires sur des points de données réels. Après cela, encouragez le modèle à avoir des sorties similaires pour les points de données réels et perturbés.

Exemple de transformation de données d'entrée



Le VAT : Généralités

- **VAT** : Technique efficace pour la LDS
- **Principe** : sélection d'une entrée et d'une perturbation pour laquelle le modèle donne une sortie très différente $\rightarrow$ pénalisation du modèle pour sa sensibilité à la perturbation.

Le VAT : Méthodologie

Etapes

- Point de données d'entrée x
- Transformation de x en T(x) = x+r (r étant la perturbation)
- r étant dans la direction adversaire

La sortie du modèle de l'entrée perturbée T(x) doit être différente de la sortie de l'entrée non perturbée $\rightarrow$ distance de Kullback-Leibler maximale

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

avec

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

Le VAT : Méthodologie

Robustesse

Après avoir trouvé la perturbation adversaire et l'entrée transformée nous mettons à jour les poids du modèle de sorte que la divergence KL soit minimisée. Cela rend le modèle robuste face à différentes perturbations.

La perte suivante est minimisée par la descente de gradient :

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

À mesure que le modèle devient plus robuste, une baisse de la perte est observée.

Le VAT : Terme de régularisation

Mesure de régularité

$LDS(x, \theta)$ peut être considéré comme une mesure négative de la régularité locale du modèle à chaque donnée d'entrée x, et sa réduction rendrait le modèle lisse à chaque point de données.

Le terme de régularisation est la moyenne des $LDS(x, \theta))$ sur toutes les données d'entrée :

$$R_{vadv}(D_l, D_{ul}, \theta) = \frac{1}{N_l + N_{ul}} \sum_{x \in D_l, D_u l} LDS(x, \theta)$$

Le VAT : Paramètres

Nombre de paramètres

Un des avantages du VAT est son nombre réduit d'hyperparamètres (2) rendant ainsi l'optimisation basé sur ces derniers beaucoup moins complexe qu'avec d'autres modèles génératifs d'apprentissage supervisé et semi-supervisé.

- $\epsilon > 0$: contrainte de norme pour la direction adversaire

$$||r_{vadv}|| \leq \epsilon$$

- $\alpha > 0$: contrôle l'équilibre relatif entre la log-vraisemblance négative et le terme de régularisation dans la fonction objectif :

$$l(D_l, \theta) + \alpha R_{vadv}(D_l, D_u l, \theta)$$

$l(D_l, \theta)$: log-vraisemblance négative des données labélisés

Données MNIST

- Base de données de chiffres écrits à la main
- 60 000 images d'apprentissage et 10 000 images de test
- Images en noir et blanc, normalisées centrées de 28 pixels de côté

Résultats obtenus

Apprentissage supervisé

Obtention d'une accuracy de 67,56% avec un modèle supervisé sur nos données.

```
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
```

Apprentissage semi-supervisé

Accuracy de 92,15% obtenu avec 100 labels sur les données MNIST.

```
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
```

Impact des hyperparamètres

- Pour de faibles valeurs de ϵ les deux paramètres jouent à peu près le même rôle $\rightarrow$ optimisation sur uniquement l'un d'entre eux
- Lorsque ϵ est grand $\rightarrow$ recherche de la paire d'hyperparamètres donnant des résultats optimaux
- Résultats satisfaisants obtenus avec α fixé à 1 et $\epsilon \in [1, 3]$

Conclusion

Résultats concluant en apprentissage semi-supervisé sur les données MNIST comme vu dans l'article d'etude.
Le Virtual Adversarial Training présente de nombreux avantages :

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

Merci de votre attention !