Exemples passeurs en apprentissage profond

Adrien CHAN-HON-TONG
ONERA the french aerospace lab
adrien.chan_hon_tong@onera.fr

1 Résumé

Les exemples contradictoires en apprentissage profond sont la possibilité d'avoir deux images équivalentes classées de façon différentes. Classiquement, on considère que ce problème permettrait aux acheteurs d'un système d'apprentissage profond de le hacker. Je montre ici (via une experimentation sur CIFAR10) que ces exemples contradictoires pourraient aussi permettre aux vendeurs de tels systèmes de tricher (de façon indétectable) sur les performances annoncées.

2 Introduction

L'apprentissage profond [8] semble être une nouvelle révolution industrielle avec des applications qui iraient bien au delà de l'indexation du web [12]. On pense notamment aux voitures autonomes et aux diagnostics [5].
Cependant, l'apprentissage profond n'est pas acceptable sur au moins deux points : le secret des données d'apprentissage [11] et les exemples contradictoires (voir par exemple [10, 9, 15]).
La problématique du secret est qu'il a été montré qu'il peut être possible de determiner des données d'apprentissage à partir des poids d'un reseau de neurones classiques [11]. Cependant, les réseaux avec bulle d'air (on ne publie qu'un réseau n'ayant jamais vu les vraies données mais ayant appris via des réseaux privés) sont un premier moyen de réduire ce problème.
Plus problématique, les exemples contradictoires montrent que les réseaux de neurones classiques d'apprentissage profond sont hackables. Une perturbation imperceptible d'une donnée d'entrée peut conduire à une décision opposée. D'un point de vue philosophique, ce problème n'est peut être pas si critique. Au fond, la perturbation ajoutée fait sortir la donnée de la distribution visée (même en étant imperceptible). On ne peut donc pas vraiment exiger du réseaux de neurones de bien traiter cette donnée. Cependant d'un point de vue pratique, c'est un vrai problème car si ce type de technique d'apprentissage profond est mis en production, il devra interagir avec des utilisateurs intelligents, et donc, notamment il devra affronter des données sortant de la distribution initiale.
Ce problème d'exemples contradictoires est souvent perçu comme la possibilité de hacker (en tant qu'utilisateur) un système d'apprentissage profond (ces exemples pouvant exister dans le monde physique [7]).
Cependant, je montre ici (sur un exemple de classification d'images sur CIFAR10 [6]) qu'un autre hackage est possible : celui d'un vendeur qui voudrait mentir sur les performances.

3 Evaluation vs falsification

L'évaluation d'un module d'apprentissage profond est aujourd'hui un processus fondamentalement empirique (et [14] peut faire penser qu'il n'y a pas d'alternative). Pour évaluer un module, il convient donc de collecter des nouveaux échantillons de la distribution visée et de mesure la qualité du module sur ces nouveaux échantillons. Cependant, si cette procédure est suffisamment sure, sa compatibilité avec la legislation et la pratique économique n'est pas évidente !
Peut-on réellement faire un appel d'offre (typiquement d'une entreprise publique tenue de faire des appels d'offres publiques) qui ne précise pas explicitement comment le système demandé sera évalué ? Et, si oui, peut on réellement appliquer une pénalité au maitre d'oeuvre pour avoir livrer un produit insuffisant alors que celui ci n'a aucune façon de s'auto evaluer ? Et si oui, considérant que la collecte d'une base d'apprentissage est de loin l'étape la plus couteuse dans la conception d'un tel module d'appentissage profond, y aura-t-il des entreprises pour prendre le risque du payer cette collecte sans savoir si elles pourront le valoriser puisque sans maitrise sur la base de test ? D'ailleurs, il n'est pas clair que la nécessité de s'évaluer sur des données inconnus soit clairement mis en évidence dans toutes les normes qui pourraient être concernée (comme European norm CEN/TC 206 - Biological and clinical evaluation of medical devices).
Pour ces raisons purement économiques, il est possible que dans certain cas, une entreprise soit autorisé à vendre un module d'apprentissage profond évalué sur des données public utilisables par l'entreprise. Or, dans ce cas, il est connu qu'il est possible d'afficher une performance d'évaluation bien plus grande qu'elle ne devrait.
Typiquement, directement apprendre sur les données de test permet d'obtenir des performances totalement surevaluées. Cependant, une telle falsification est facile à détecter par un auditeur expert. Plus tenue, de telles fausses performances peuvent

se produire (parfois même inconsciemment) via l'utilisation de la base de test comme base d'évaluation [2]. Ce faisant, reproduire l'expérience d'apprendre puis d'évaluer ne permet pas de détecter une falsification. Seule une suspicion existe en fonction du nombre de meta paramètres plus ou moins règlés en dur.
Cependant, je montre ici qu'il y a une technique plus imaginative de falsification consistant à ajouter une perturbation indetectable aux exemples d'apprentissage (la collecte de ces exemples étant couteuses, il parait pertinent que le cout soit pris en charge par le vendeur et non par l'acheteur qui sinon aurait interêt à internaliser totalement la conception). Ce faisant, cette falsification est totalement indetectable : elle marche avec un module ayant 0 meta parametre et une relecture de la base d'apprentissage ne peut pas la mettre en lumière (ce qui est le cas, notamment, si on se contente simplement de mettre les exemples de tests dans la base d'apprentissage). Cette technique justifie de s'appeler exemples passeurs car les perturbations à ajouter se calculent comme pour les exemples contradictoires.

4 Expérimentation sur CIFAR10

4.1 Système visé

Le système 0 paramètre choisi est un réseau de neurones convolutionnel (CNN) entrainé sur IMAGENET [1] [1] utilisé comme extracteur de caractéristiques puis un SVM [13] (cette chaine est inspired de [4]). Notons que la chaine est totalement reproductible : pas d'aléatoire et des propriétes de convexité. Les poids finaux ne dependent donc que de la base d'apprentissage. Les images (train et test) sont transformées en vecteur par le CNN (0 paramètre). Un SVM est appris sur les vecteurs d'apprentissage et utilisé sur ceux de test. (Ici, ce sera LIBLINEAR avec les paramètres par défauts [3] donc virtuellement sans paramètre).
Bien sur, ce système n'est pas adapté a CIFAR (puisque les poids sont appris sur IMAGENET) et n'a que 75% de taux de bonne classification mais la falsification va le rendre (injustement) compétitif.

4.2 Justification théorique

Soit $x_1, ..., x_N$ dans $\mathbb{R}^D$ et $y_1, ..., y_N$ les labels (dans -1/+1) correspondant. Soit des poids w dans $\mathbb{R}^D$, l'énergie classique sur w est $e(w, x, y) = w^T w + \sum_n relu\left(1 - y_n x_n^T w\right)$ avec $relu$ la fonction 0 avant 0 et valant l'identité aprés ($C = 1$ avec LIBLINEAR - il manque le b pour favoriser la lecture). L'apprentissage conduit à $w^*(x, y) = \arg\min_w e(w, x, y)$. Notons $w_{train} = w^*(x_{train}, y_{train})$.
Maintenant, rappelons que si R est orthonormale (typiquement une rotation) alors $e(Rw, Rx, y) = e(w, x, y)$, et donc, $w^*(Rx, y) = Rw^*(x, y)$.
Supposons alors que l'on dispose de w_{voulu} (car l'évaluation est publique - sinon bien sur cette falsification est impossible). Si chaque $x_{train,n}$ (on omet le *train*) devient $x'_n = x_n + \delta x_n^T w_{train} w_{voulu} - \delta x_n^T w_{voulu} w_{train}$, l'apprentissage conduit alors vers $w_{hack} = w^*\left((I + \delta w_{test} w_{train}^T - \delta w_{train} w_{test}^T)x, y\right)$. Or, au premier ordre en δ, $(I + \delta w_{test} w_{train}^T - \delta w_{train} w_{test}^T)$ est orthonormale, donc $w_{hack} = w_{train} + \delta w_{train}^T w_{train} w_{test} - \delta w_{train}^T w_{test} w_{train}$ (au premier ordre). En modifiant les vecteurs d'entrée, on peut ainsi obtenir une rotation du modèle vers les poids voulus (on calcule le gradient de l'image vis à vis de la distance du vecteur voulu avec le vecteur courant). Malheureusement, avec des CNN classiques, une petite perturbation des images va conduire à de fort changement niveau vecteur (c'est ce que montrent les exemples contradictoires) et donc une forte rotation vers les poids voulus. Ainsi, des perturbations invisibles permettent de fortement biaiser l'évaluation.

4.3 Calcul pratique

En pratique, 40 lignes de pytorch suffisent pour pouvoir biaiser l'apprentissage. Il suffit de considérer le CNN avec une couche supplémentaire - initialisée avec des poids voulus. Il suffit ensuite de passer chaque image d'apprentissage dans le CNN en indiquant qu'on veut stocker le gradient de l'image. Finalement, le signe du gradient est ajouté à l'image : pour chaque pixel de l'image, la valeur du pixel est modifiée de -1,0,+1 en fonction du signe de la derivée partielle correspondante. La transformation semble alors globalement vérifier $x_n'^T w_{voulu} = x_n^T w_{voulu} + \lambda_n y_n$, et, est ici empiriquement plus efficace que la rotation de la sous section précédente.

4.4 Résultats

En apprennant après une perturbation (donc avec des pixels d'apprentissage distant d'au plus 1 des pixels bruts), le taux de bonne detection passe de 75% à 86%. Pire en appliquant seulement 3 fois cette étape, ce taux rejoint l'état de l'art avec 92%.
Ce niveau de performance est peu interessant pour CIFAR10 mais cette technique pourrait s'appliquer par exemple sur des cellules (bioloqiques) et il serait alors problématique de pouvoir passer de 75% à 92% aussi injustement...

1. https ://github.com/jcjohnson/pytorch-vgg

Remarque :

Cet article ne cherche pas à empécher la diffusion des technologies d'apprentissage profond dans la société. Je souligne juste que de telles falsifications seraient un problème. La question de savoir quelle reaction doit avoir le legislateur est hors de la portée de cet article. Cet article ne doit pas être lu comme la position de l'ONERA sur ce sujet.

Références

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet : A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[2] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout : Preserving validity in adaptive data analysis. *Science*, 349(6248) :636–638, 2015.

[3] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR : A library for large linear classification. *Journal of Machine Learning Research*, 9 :1871–1874, 2008.

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[5] Hayit Greenspan, Bram van Ginneken, and Ronald M Summers. Guest editorial deep learning in medical imaging : Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5) :1153–1159, 2016.

[6] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[7] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations (ICLR)*, 2017.

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553) :436–444, 2015.

[9] Seyed Mohsen Moosavi Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[10] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled : High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

[11] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321. ACM, 2015.

[12] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface : Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[13] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[14] David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7) :1341–1390, 1996.

[15] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.