centralelille
ÉCOLE CENTRALE DE LILLE

Etude de la robustesse des réseaux de neurones en réponse à des méthodes de sparsification et de régularisation

Louis Maestrati

14 septembre 2020

Superviseurs: Jared Tanner (Oxford), Constantin Puiu

Plan

- Introduction aux réseaux de neurones
- Attaques adversarielles et robustesse
- Magnitude Pruning (méthode de sparsification)
- Sparsification et Robustesse
- Low-Rank Regularization et Robustesse
- Entraînement à haute température et Robustesse
- Conclusion

Introduction aux réseaux de neurones (DNN)

Un DNN est un estimateur issu de la composition en série de fonctions affines -paramétrés par des poids- et de fonctions d'activations choisis expressement non linéaires:

$$f(x) := \sigma_d\big(W_d^T \sigma_{d-1}(...(\sigma_1(W_1^T x + b_1))) + b_{d-1}) + b_d\big), \quad (1)$$

où $f(x) \in \mathbb{R}^{n_d}$ est la sortie du réseau, $\{W_i \in \mathbb{R}^{n_{i-1} \times n_i}\}_{i=2}^d$, $W_1 \in \mathbb{R}^{n \times n_1}$ les matrices de poids des couches, $\{b_i \in \mathbb{R}^{n_i}\}_{i=1}^d$ les biais, n_i est le nombre de neurones de la couche i, et n est la taille de l'entrée x du réseau.

- On s'intéresse à la classification $f(x) = p(x) \in \mathbb{R}^c$, où $c = n_d$ est le nombre de classes
- Les logits sont définis comme:

$$g(x) = W_d^T f(x) \in \mathbb{R}^c \quad (2)$$

- la classe prédite est $\hat{y}(x) = indmax(p(x))$

Introduction aux DNNs: la fonction de perte/loss

Les logits sont:

$$g(x) = W_d^T f(x) \in \mathbb{R}^c \tag{3}$$

Le vecteur de probabilité est obtenu en appliquant une fonction σ_d particulière, le softmax (noté s):

$$s_k(g(x)) := \frac{\exp(g_k(x)/T)}{\sum_{i=1}^{c} \exp(g_i(x)/T)}, \quad T > 0, \quad \forall k \in \{1, 2, ..., c\}, \tag{4}$$

où T est le paramètre de température.

La loss est la cross-entropie, définis comme:

$$\frac{1}{N} \sum_{i=1}^{N} L(x_i, y_i; \theta) = \frac{1}{N} \sum_{i=1}^{N} \left(- \sum_{k=1}^{c} y_i^{(k)} \log(s_k(g(x_i))) \right). \tag{5}$$

Attaques adversarielles

- Intuition: de faibles perturbations en entrée peuvent mener à de grands changements en sortie:

$$\|f(x+\delta) - f(x)\| \leq \|\delta\| \prod_{i=1}^{d} \big(L_{\sigma_i}\|W_j\|\big), \tag{6}$$

- Ideallement, les "attackers" utiliseraient $\tilde{x} = x + \delta_x^*$ où

$$\delta_x^* = \arg\min_{\delta_x} \|\delta_x\| \qquad \text{s.c.} \qquad \hat{y}(x+\delta_x) \neq \hat{y}(x). \tag{7}$$

- En pratique on utilise l'attaque Fast Gradient Sign Method (FGSM) qui exploite le gradient de la loss p.r. à l'image en entrée:

$$\tilde{x} = x + \epsilon \,\text{sign}\big(\nabla_x L(x, y; \theta)\big) \quad \text{pour un faible} \quad \epsilon > 0, \tag{8}$$

- Et l'on continue à augmenter ϵ jusu'à observer un changement de classe.

Robustesse

Il y a de nombreuses manières de définir la robustesse R d'un réseau. On la définit comme la résistance aux attaques FGSM:

$$R := \mathbb{E}_{x,y}[1_{y=\hat{y}(\hat{x}(\epsilon))}] = acc(\epsilon)$$

Sparsification: Magnitude Pruning (MP)

centralelille
ÉCOLE CENTRALE DE LILLE

Qu'est ce qu'un DNN "sparse"? Un DNN est dit "sparse" si un nombre important de ces poids peuvent être supprimés / capable d'effectuer sa tâche en utilisant qu'un nombre réduit de poids

- La littérature lie la robustesse et la "sparsity" d'un DNN [1], en usant du MP [2], [3], [4].
- MP: Élaguer les poids du DNN les plus faibles.
- Nous avons élaguer graudellement les poids au cours de l'entraînement (comme proposé par Zhu et. al.,[5]).

MP et DNNs: Résultats

centralelille
ÉCOLE CENTRALE DE LILLE

En règle générale, MP fait décroître la robustesse du réseau, avec quelques exceptions. On a conduit des tests avec les modèles Resnet18, Alexnet et un perceptron à 5 couches. Divers batchsizes ont été testés.

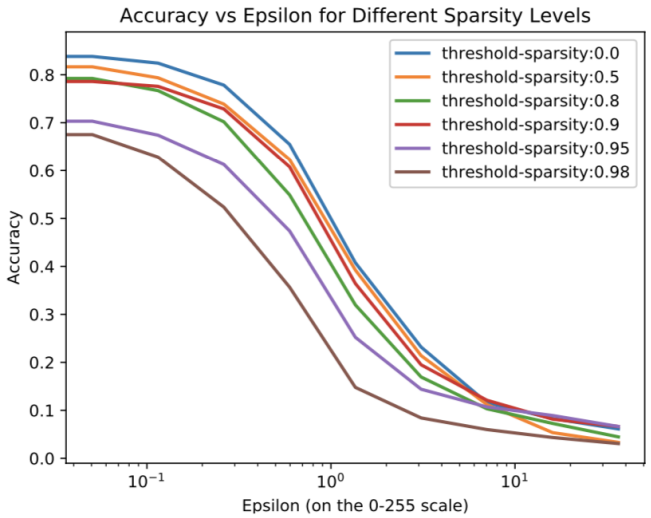

Figure 1: *Accuracy du Resnet18 pour différentes valeurs de ϵ, suivant la méthode FGSM. Le batchsize utilisé est 512.*

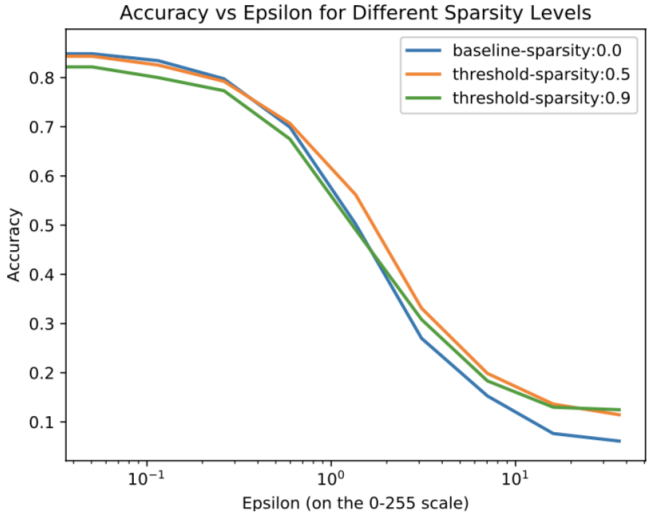

Figure 2: *Accuracy du Resnet18 pour différentes valeurs de ϵ, suivant la méthode FGSM. Le batchsize utilisé est 128.*

MP et DNNs: Résultats

centralelille
ÉCOLE CENTRALE DE LILLE

D'autres résultats négatifs:

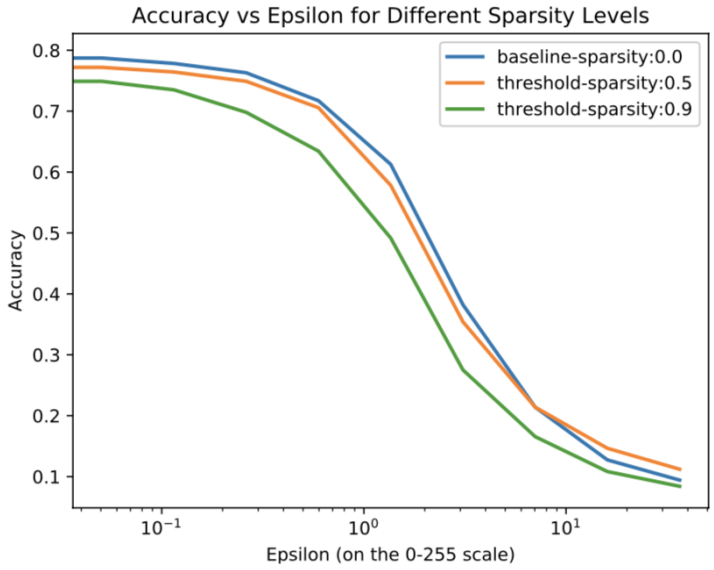

Figure 3: *Accuracy du Alexnet pour différentes valeurs de ϵ, suivant la méthode FGSM. Le batchsize utilisé est 128.*

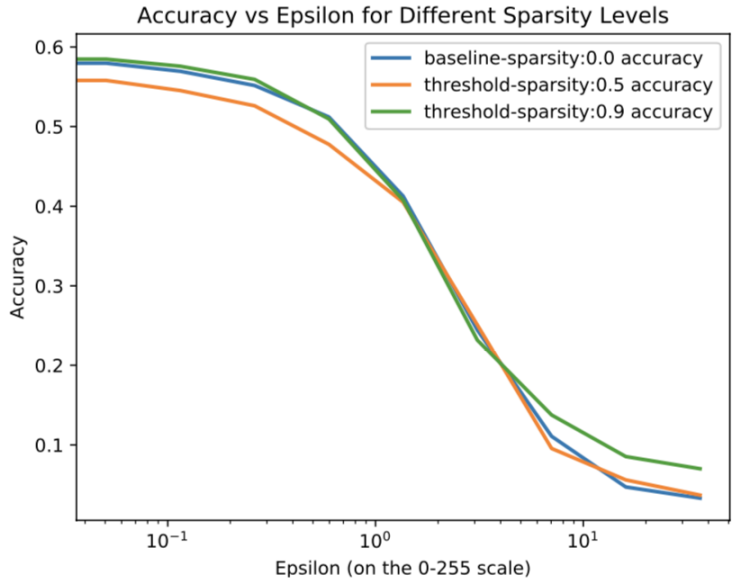

Figure 4: *Accuracy du FC5 pour différentes valeurs de ϵ, suivant la méthode FGSM. Le batchsize utilisé est 512.*

- Conclusion: robuste $\Rightarrow$ sparse mais sparse $\not\Rightarrow$ robuste!

Low-Rank Regularization

- La littérature suggère que l'adversarial training induit des matrices de poids à faible rang [6].
- L'utilisation de la norme nucléaire (diminuant le rang des poids du modèle) dans la loss améliore la robustesse du réseau, mais pas autant que l'adversarial training [6].
- On montre qu'une méthode de low-rank regularization différente, contraignant le rang du modèle, n'amène pas à plus de robustesse.

Low-Rank Regularization: Résultats

- On a utilisé une méthode basée sur une décomposition des matrices de poids pour effectuer de la low-rank regularization ([7])
- Les résultats sont négatifs:

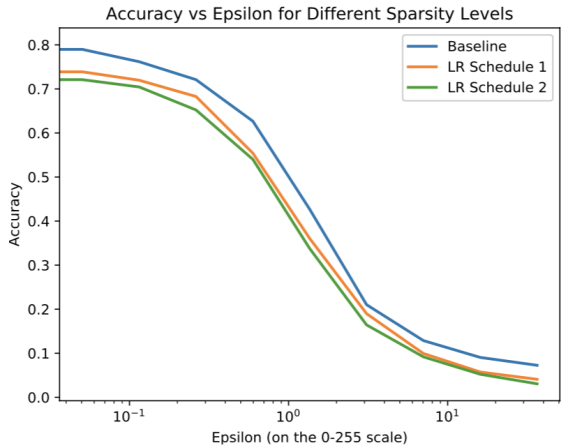

Figure 5: *Accuracy de l'Alexnet pour différentes valeurs de ϵ, suivant la méthode FGSM. Le batchsize utilisé est 512.*

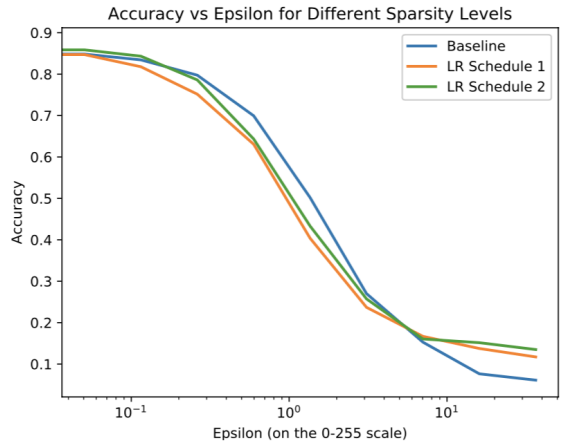

Figure 6: *Accuracy du Resnet18 pour différentes valeurs de ϵ, suivant la méthode FGSM. Le batchsize utilisé est 128.*

- Conclusion: robuste $\Rightarrow$ low-rank mais low-rank $\not\Rightarrow$ robuste!

L'entraînement à haute température

centralelille
ÉCOLE CENTRALE DE LILLE

- On fait varier le paramètre $T \in [20, 100]$
- Les résultats sont positifs

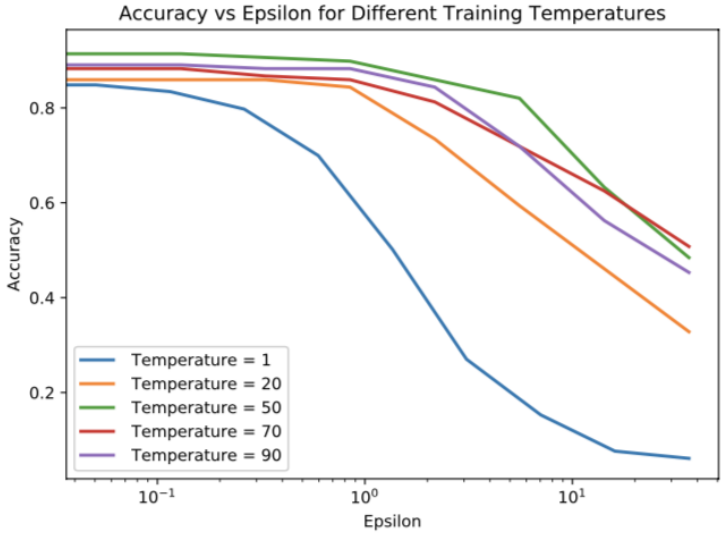

Figure 7: *ResNet18, l' attacker ne connaît pas la température d'entraînement T.*

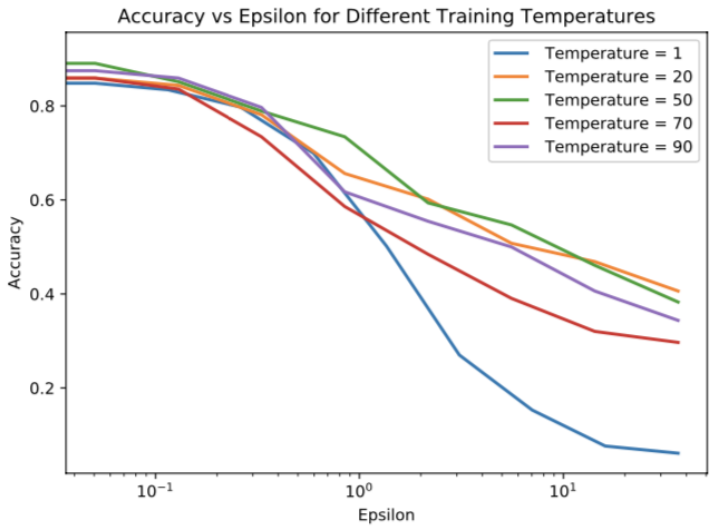

Figure 8: *ResNet18, l' attacker connaît la température d'entraînement T.*

- Conclusion: Est utile face aux attaques FGSM, encore plus lorsque ϵ est grand.

Conclusions

1. Un réseau robuste semble adopter des matrices de poids à faible rang et sparses. Cependant, un modèle à faible rang et sparse n'est pas nécessairement plus robuste que son jumeau non régularisé.
2. L'entraînement à haute température constitue un bon moyen de prévention aux attaques FGSM

References

[1] Justin Cosentino, Federico Zaiter, Dan Pei, Jun Zhu, The Search for Sparse, Robust Neural Networks, *arXiv:1912.02386* , 2019.

[2] Luyu Wang, L.; Ding, G. W.; Huang, R.; Cao, Y.; Lui, Y. C. Adversarial robustness of pruned neural networks, *ICLR Workshop submission*, 2018.

[3] Ye, S.; Wang, S.; Wang, X.; Yuan, B.; Wen, W.; Lin, X. Defending DNN adversarial attacks with pruning and logits augmentation, *ICLR Workshop submission*, 2018.

[4] Guo, Y.; Zhang, Chao.; Zhang, Changsui.; Chen, Y. Sparse DNNs with Improved Adversarial Robustness, *Advances in neural information processing systems*, 2019.

[5] Zhu, M.; Gupta, S. To prune, or not to prune: exploring the efficacy of pruning for model compression, *CoRR, abs/1710.01878*, 2017. URL: http://arxiv.org/abs/1710.01878.

[6] Langenberg, P.; Balda, E. R.; Behboodi, A.; Mathar, R. On the Effect of Low-Rank Weights on Adversarial Robustness of Neural Networks, 2019.

[7] Tai, C.;, Xiao, T.; Zhang, Y.; Wang, X.; Weinan E. Convolutional Neural Networks with Low-rank Regularization, 2016.