Projet 3
Attaques adversariales

Mohamed Zineddine Chedadi, Félix Breton, Charlotte Durand

18 Décembre 2020

Table des matières

1 Introduction

Dans ce projet, nous allons vous présenter nos travaux sur les attaques adversariales de réseaux de neurones. Le principe des attaques adversariales est simple : on cherche à tromper des réseaux de neurones en ajoutant aux données en entrée une petite perturbation, imperceptible à l'oeil humain qui permet la modification de la classe proposée par le modèle en sortie. Nous allons vous présenter plusieurs algorithmes d'attaques adversariales ainsi que des méthodes d'entraînement de modèles permettant d'apprendre au modèle à contrer ces attaques.

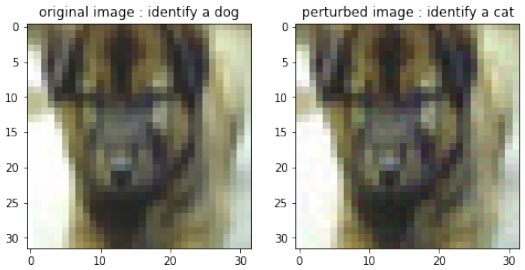


FIGURE 1 – Une image correctement étiquetée comme un chien est transformée en chat en subissant une attaque adversariale

2 Algorithmes

Nous allons dans ce projet considérer des entrées x, dans notre cas des images issues de la base de données CIFAR 10. Ce sont des images en couleurs de taille $32 * 32$ réparties en 10 classes. On va alors pouvoir évaluer nos entrées x, à travers notre modèle avec des poids w.

$$x_{adversarial} = \tilde{x} = x + \eta \tag{1}$$

Lorsque l'on crée un exemple adversarial, on va ajouter à nos entrées x une grandeur η, suffisamment petite pour qu'elle ne soit pas perceptible à l'oeil nu. En pratique pour CIFAR10, on impose que pour chaque pixel, la perturbation maximale soit de $\eta = \frac{8}{256} \sim 0.031$.

2.1 Attaques adversariales

Nous avons décider de travailler sur deux types d'attaques adversariales : "Projected Gradient Descent" (PGD) et "Fast Gradient Signed Method" (FGSM). Ces deux méthodes sont relativement semblables, elles s'appuient sur le même principe : on calcule le signe du gradient par rapport aux images de la fonction de perte du réseau profond. Cela signifie donc que l'on est sur des attaques nécessitant de connaître les poids du modèle entraîné.

Pour simplifier, on peut dire que PGD consiste à itérer FGSM, tout en effectuant une projection afin de garder la perturbation limitée.

2.1.1 FGSM

Il s'agit du type d'attaque le plus simple :
On calcule :

$$D = \text{signe}(\Delta_x L(\theta, x, y)) \tag{2}$$

Ce gradient est alors multiplié par la valeur η qui conditionne la perceptibilité de la perturbation.

Algorithm 1 FGSM

Require: images d'entrée x, classe y de x, θ (poids du modèle), η
On convertit les images en tenseurs
On calcule $D = \text{signe}(\Delta_x L(\theta, x, y))$
On calcule les images perturbées $\tilde{x} = x + \eta * D$
return La prédiction du modèle sur les images perturbées $\tilde{x}$

2.1.2 PGD

L'algorithme PGD repose sur le même calcul du gradient (Eq. 2) sur un nombre d'itérations que l'on définit. Après avoir calculé le gradient et avoir calculé la perturbation modulé par η, on projette l'image perturbée $\tilde{x}$ sur une boule centrée en x et de rayon ϵ qui correspond au maximum que l'on s'impose pour ne pas détecter visuellement la perturbation.

Algorithm 2 PGD

Require: images d'entrée x, classe y de x, θ (poids du modèle), ϵ, η, nombre d'itérations n
for i in range (1,n) **do**
On convertit les images en tenseurs
On calcule $D = \text{signe}(\Delta_x L(\theta, x, y))$
On calcule les images perturbées $\tilde{x} = x + \eta * D$
On projette $\tilde{x}$ sur une boule centrée en x et de rayon ϵ
end for
return La prédiction du modèle sur les images perturbées $\tilde{x}$

2.2 Entraînement face aux attaques adversariales

2.2.1 Entraînement adversariale

Lors d'un entraînement adversariale, on va entraîner un modèle à la fois sur les images perturbées et sur les images non perturbées. On utilise ici la même structure que pour le modèle classique, mais avec deux branches pour chacun des types d'images (poids de réseau partagés entre les deux branches). Ensuite, la fonction de perte du modèle est calculée en utilisant un poids α pour jouer sur le rapport entre les deux types d'images.

$$\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x + \eta \,\text{signe}(\Delta_x J(\theta, x, y)), y) \tag{3}$$

Dans l'équation 3, on utilise l'attaque FGSM. On peut également utiliser l'attaque PGD lors de cet entraînement. En injectant des images perturbées dans le modèle, on souhaite lui permettre de se défendre face à ces attaques.

2.2.2 Entraînement ensembliste

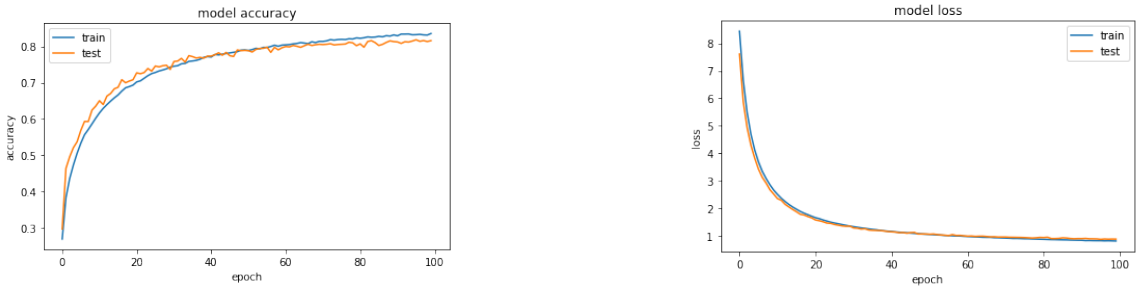
Lors d'un entraînement ensembliste, au lieu de juste entraîner un seul modèle, on entraîne adversarialement plusieurs modèles, et lors de la prédiction on prend le vote majoritaire. Le fait d'avoir plusieurs modèles a comme conséquence d'affaiblir les attaques. En effet, lors d'une attaque, il faut prendre le signe du gradient de la fonction de perte de notre modèle. Or que quand on a plusieurs modèle il s'avère que produire une attaque sur un seul de ces sous-modèles n'aura pas de grand effet sur le modèle ensembliste. Ainsi pour l'attaquer il est recommandé de prendre la moyenne des gradients ou la moyenne des signes des gradients. Dans notre projet on a opté pour la moyenne des signes des gradients vu qu'elle nous a donné des attaques plus puissantes comparant à l'autre approche.

3 Expériences

Dans cette partie, nous allons voir les résultats expérimentaux que nous avons trouvé.

3.1 CIFAR 10

Dans un premier temps, nous entraînons un modèle sur CIFAR 10. Le modèle comprend trois fois de suite une Conv2D, une normalisation et un MaxPooling, puis un AveragePooling suivi en sortie de deux couches denses activées par une ReLU puis une softmax. On utilise après les Conv2D du dropout, ainsi qu'une régularisation $l2$ afin de limiter le sur-apprentissage. La fonction de perte est une "categorical cross-entropy". Les courbes de la fonction de perte et de l'accuracy sont présentées dans la Fig. 2. On observe bien l'absence de sur-apprentissage, puis que la fonction de perte n'augmente pas sur la validation lorsque le nombre d'epochs augmente. On obtient les valeurs suivantes :



(a) Accuracy

(b) Fonction de perte

FIGURE 2 – Fonction de perte et accuracy du modèle entraîné sur CIFAR-10

Set de données	Accuracy	Loss
Entraînement	91,06 %	0.632
Test	81,51%	0.868

3.2 PGD et FGSM

Nous allons comparer nos deux attaques FGSM et PGD sur le modèle entraîné ci-dessus. Sur la figure de gauche, on observe l'accuracy du modèle calculée avec les images perturbées ainsi que l'évolution de la fonction de perte du modèle en fonction de la valeur de η, c'est-à-dire le poids que l'on accorde à la perturbation. L'attaque PGD est proposée avec 3, 4 et 5 itérations. On remarque que PGD offre de meilleurs résultats, malgré un faible nombre d'itérations. De plus, PGD donne une baisse importante de l'accuracy pour des valeurs de η plus faible.

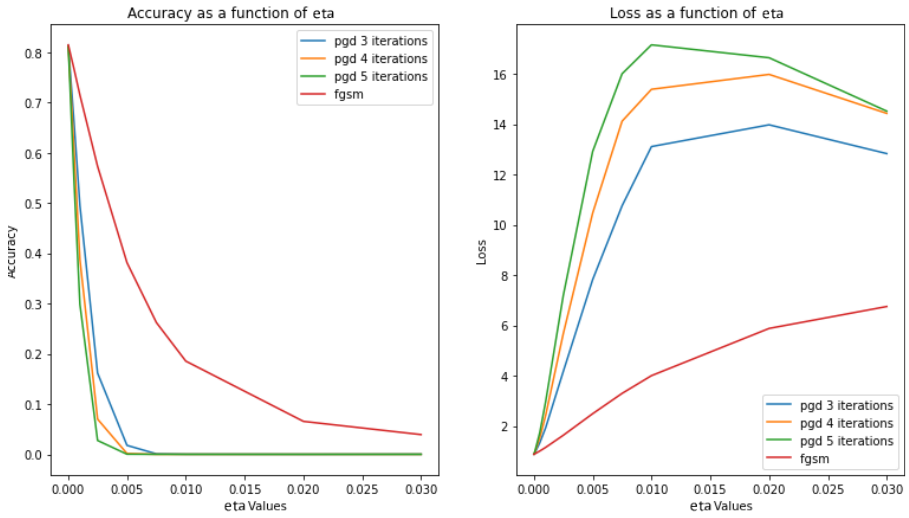


FIGURE 3 – Comparaison entre les attaques PGD et FGSM

En choisissant la valeur maximale de $\eta = 0.03$ on obtient les résultats suivants :

Attaque	Accuracy
Sans attaque	81,5%
Avec FGSM	3,9%
Avec PGD (4 itérations)	0%

3.3 Entraînement adversarial

3.3.1 Entraînement adversarial classique

Nous avons commencé par entraîner notre modèle à résister aux attaques adversariales en lui injectant des images perturbées par des attaques PGD et FGSM. Dans toute la suite des résultats, nous avons utilisé un facteur α de 0.5. Comme on peut l'observer dans le tableau ci-dessous, On perd environ 12% d'accuracy lorsque l'on entraîne avec des images perturbées, un peu moins pour l'attaque PGD. Ces calculs sont fait sur le set d'images test.

Modèle	CIFAR-10	+ FGSM	+PGD
Accuracy	81,52 %	68,84 %	70,38 %

Nous pouvons maintenant lancer des attaques PGD et FGSM sur ces nouveaux modèles :

Attaque / Entrainement	FGSM	PGD
CIFAR-10	3,9 %	0 %
FGSM	6,46 %	1,75 %
PGD	19,79 %	14,42 %

On remarque que dans tous les cas, les modèles résistent mieux aux attaques puisque l'accuracy augmente. Le modèle entraîné avec une attaque PGD marche plutôt bien sur une attaque PGD et encore mieux sur une attaque FGSM.

3.3.2 Entraînement adversarial ensembliste

Pour notre modèle ensembliste, on a choisit d'entraîner 9 modèles adversariales en utilisant l'algortihme du PGD. Le nombre est fixé à cause de la limitation de GPU de google colab. Il serait intéressant de voir l'influence d'un grand nombre des sous-modèles. Pour les 9 sous-modèles, on a choisit de les entrainer avec des hyperparametres, η et le nombre d'itérations, différents. Ce choix était arbitraire vu qu'on avait pas assez de temps pour faire plusieurs expériences. Le tableau ci-dessous montre les performances des différents sous-modèles :

Modèle	Ensemble	0	1	2	3	4	5	6	7	8	9
Accuracy (%)	78,7	74,4	78,1	77,5	75,4	76,0	75,0	74,0	70,5	68,4	60,1

Pour voir l'influence du nombre de sous-modèles, on a attaqué plusieurs modèles ensemblistes avec des nombres de sous-modèles différents. Il importe de noter que dans cette expérience on a utilisé une attaque FGSM avec des valeurs de η différentes. Sur la gauche on trouve les performances des modèles ensemblistes avec des sous-modèles entraînes adversarialement, or sur la droite on trouve des sous-modèles entraînes d'une manière normale. Il est clair que dans les deux cas, l'augmentation du nombre de sous-modèles rend le modèle ensembliste plus robuste.

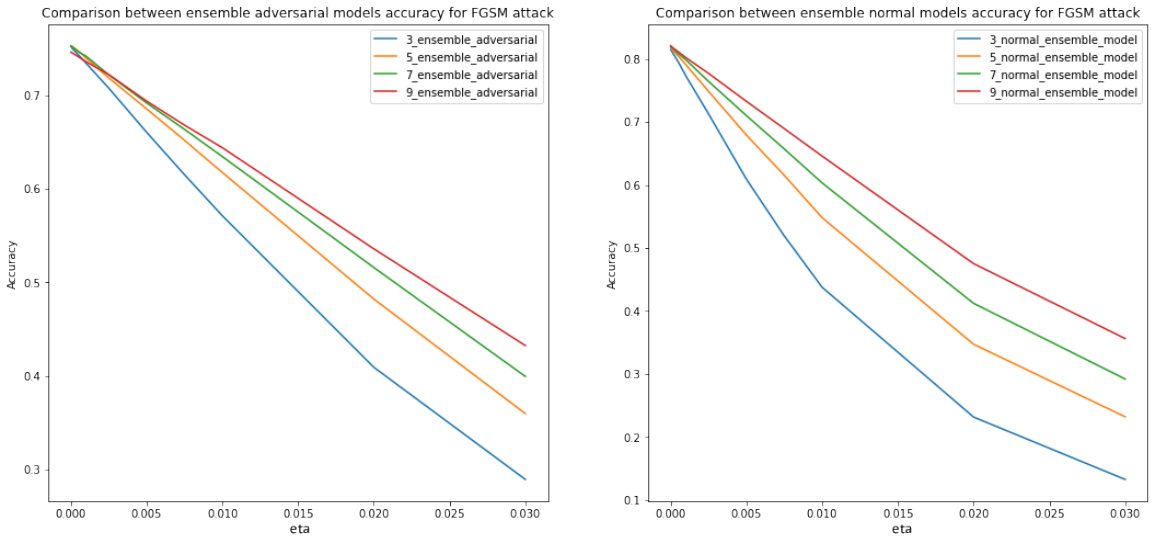


FIGURE 4 – Influence du nombre de sous-modèles sur l'accuracy face à une attaque FGSM

3.3.3 Comparaison des deux entraînements adversariaux

Nous allons maintenant pouvoir comparer l'efficacité de nos différents entraînements. Dans un premier temps, tous les modèles sont attaqués par une FGSM pour différentes valeurs de η. Les résultats sont présentés dans la Fig. 5. On remarque en premier lieu que le modèle classique est bien moins robuste que les autres, suivi du modèle entraîné avec des images perturbées par FGSM. Le modèle entraîné avec des images attaquées par PGD est globalement équivalent au modèle ensembliste avec 3 itérations non attaquées. On commence à obtenir des résultats intéressants pour le modèle à 3 itérations attaquées. Les modèles les plus robustes sont ceux ensembliste avec 9 itérations, ce qui est cohérent avec les résultats de la Fig. 4. La Fig. 6 présente une comparaison de tous ces modèles face à des attaques PGD avec 5 ou 6 itérations. On remarque tout d'abord que les résultats sont moins bon que face à une attaque FGSM. Il semble simplement que l'entraînement adversarial avec une attaque PGD simple soit plus robuste que tous les autres modèles pour des valeurs de η faibles. Pour des η élevés, les modèles adversariaux ensemblistes sont les plus robustes.

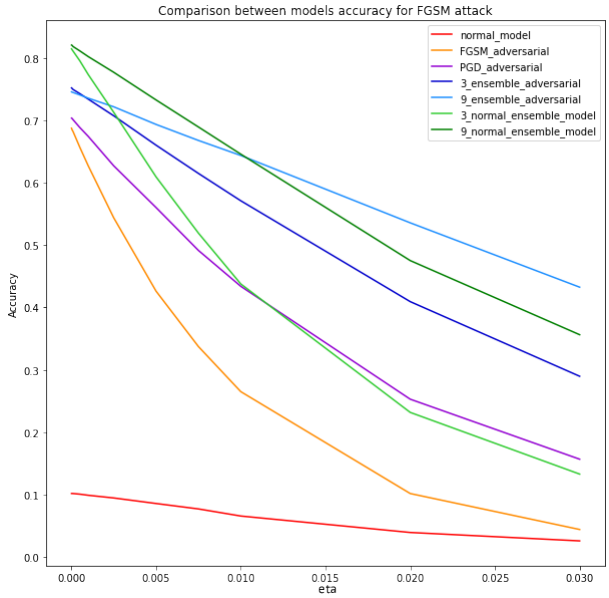


FIGURE 5 – Comparaison des entraînements sur une attaque FGSM

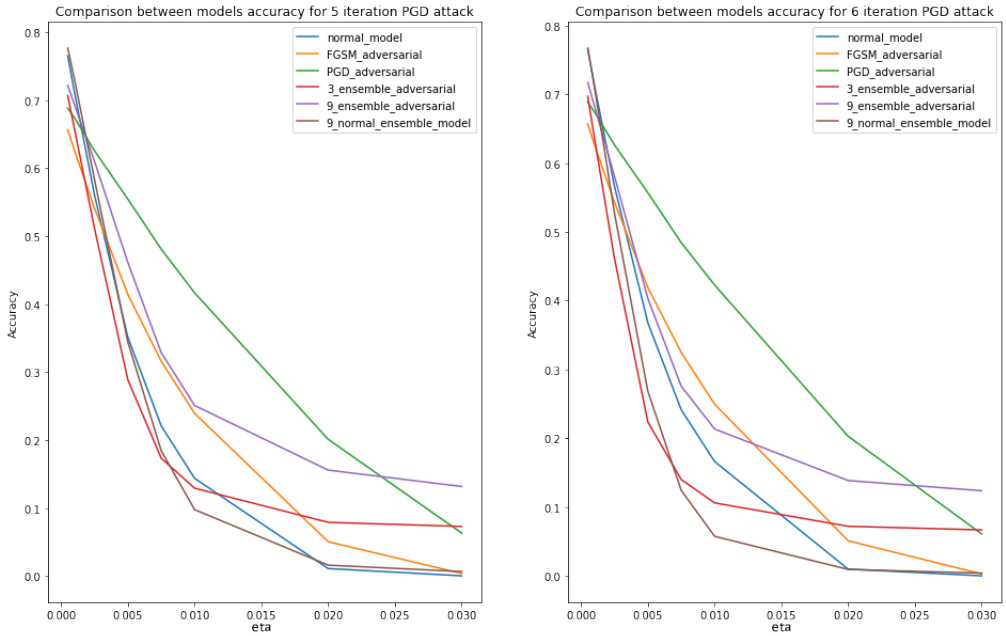


FIGURE 6 – Comparaison des entraînements sur une attaque PGD avec 5 et 6 itérations

4 Conclusion

Pour conclure, on note que notre méthode ensembliste s'avère la plus robuste face aux attaques FGSM et PGD. Force est de noter que cette méthode est à revoir et à expérimenter encore plus pour conclure de son efficacité. Il serait aussi intéressant d'expérimenter d'autres méthodes d'attaques pour ce type de défense. Une première attaque qu'on envisageait à tester constitue de prendre la moyenne des gradients des modèles qui classifient correctement l'image ou l'exemple en question au lieu de prendre la moyenne des signes des gradients de tous les modèles.