Projet 3 Data Science Lab
Attaques adversariales et robustesse des réseaux de neurones

Meriem DEHMAS - Aurèle GOETZ - Julien SENTUC
Master IASD

9 Décembre 2020

1 Introduction

Les réseaux de neurones artificiels sont devenus depuis leur avènement très usités, pour des tâches diverses. Il a été remarqué qu'ils présentent une vulnérabilité à ce qui est désigné par *attaques adversriales* et plusieurs travaux ont été menés pour comprendre cette vulnérabilité, définir des méchanismes de défense ou pour introduire de nouvelles attaques adversariales. L'objectif du projet, pour lequel le présent document est le rapport, est d'avoir une première introduction à ce domaine, d'implémenter quelques attaques connues ainsi qu'une méthode de défense et d'investiguer d'autres directions (méthodes d'attaques ou de défenses ou autre) pouvant potentiellement évoluer vers des travaux de recherche soutenables.

En section 2 sont introduites les attaques implémentées: FGSM, PGD et C&W et une méthode de défense : l'entrainement adversarial, en section 3 sont présentés les résultats des expérimentations faites sur ces méthodes. Nous décrivons en dernière section le travail prospectif mené, à savoir "la double attaque adversariale" et l'inspection du comportement du réseau de neurones dans un contexte adversarial.

Les descriptions données dans ce texte se rapportent au contexte de classification *d'images.*

2 Attaques implémentées et entrainement adversarial

Definition 1. Exemple advrsarial :

Un exemple adversarial construit à partir de l'exemple x de classe y est un exemple $x + \delta$ tel que :

. $\delta \leq \epsilon$ ($x + \delta$ imperceptiblement différent de x)

. $f(x + \delta) \neq y$ (ou dans le cas d'un exemple adversarial avec une classe cible t, $f(x + \delta) = t \neq y$)

. Et $x + \delta$ valide, ce qui revient, dans le contexte d'images dont les entrées sont entre 0 et 1 par exemple, à: $(x + \delta)_i \in [0, 1] \quad \forall i \in \{0 \dots Nb\,de\,pixels\,de\,x\}$

Une attaque adversariale contre un réseau de neurones revient donc à la génération de tels exemples , lesquels, tel que le suggère leur définition, ont pour effet de faire prédire au réseau autre que la classe originelle d'un exemple donné sur un exemple lui étant visuellement similaire (qui serait identifié par une personne comme étant toujours de la classe originelle).

2.1 Attaque FGSM

L'attaque FGSM a été le premier grand succès des attaques adversariales au sein de la littérature scientifique. En 2015, Goodfellow et al. publient leur article sur le sujet [2]. Le principe simple de l'attaque FGSM consiste à simplifier le problème de maximisation de l'erreur de prédiction en le linéarisant au voisinage du point considéré. Avec cette approche, l'exemple adversarial est obtenu par un simple pas de montée de gradient:

$$x = x + \epsilon \, sign(\nabla_x L(\theta, x, y))$$

ϵ est limité par un critère de norme comme on pourra le voir pour les autres attaques décrites ci-dessous. Pour des raisons de compacité de ce rapport, cette attaque très simple ne sera pas plus détaillée ici.

2.2 Attaque PGD

L'attaque Projected Gradient Descent (PGD) est une version itérative de FGSM.

Dans le cas d'une attaque ciblée, le processus consiste en la minimisation, avec une descente de gradient, de la fonction de perte entre la prédiction pour l'exemple adversarial et la classe cible.

$$x_{t+1} = \Pi_S \left(x_t - \alpha \, sign(\nabla_x L(\theta, x, y)) \right)$$

Avec S l'espace des perturbation admises, α le 'stepsize' de la descente de gradient, L la fonction de perte, x l'exemple adversarial, θ les paramètres du modèle et y, la classe cible.

Pour une attaque non ciblée, une montée de gradient est opérée pour maximiser la perte entre la prédiction pour l'exemple adversarial et la classe réelle (classe de l'exemple à partir duquel la version adversariale a été générée).

$$x_{t+1} = \Pi_S \left(x_t + \alpha \, sign(\nabla_x L(\theta, x, y)) \right)$$

Avec y la classe réelle.

La perturbation trouvée avec un pas de gradient $(x_{t+1} - x_t)$ est projetée dans S, l'espace des perturbations admises: la boule L_p autour de x, selon la norme choisie.

Seules les normes L_2 et L_∞ ont été implémentées.

2.3 Attaque CW

Cette section présente l'attaque introduite par Carlini et Wagner pour la norme 2[1]. L'attaque est formulé ainsi :

$$minimiser \; D(x + \delta, x) \; tel \; que \; C(x, x + \delta) = tx + \delta \in [0, 1]^n$$

où D est la mesure de distance entre l'image adversarial et l'image réelle (ici la norme L2), C la fonction de classification et t la classe d'image voulu. La première contrainte assure la mauvaise classification de l'image, tandis que la seconde assure que l'image adversarial généré est valide. La contrainte $C(x, x + \delta) = t$ est hautement non linéaire, elle est donc formulé sous la forme d'une fonction objectif f tel que $C(x, x + \delta) = t$ si et seulement si $f(x + \delta) \leq 0$.Dans leur papier, Carlini et Wagner testent 7 fonctions objectifs différentes, la meilleur est :

$$f(x) = max(max\{Z(x)_i : i \neq t\} - Z(x)_t, -k)$$

où $Z(x)_k$ est la probabilité de la classe k. Le terme k permet de définir à quel point nous voulons que notre image adversarial soit classifié en tant que classe t.

Le problème peut ainsi être reformulé comme ceci:

$$minimiser \; ||\delta||_2 + c \times f(x + \delta) \; tel \; que \; x + \delta \in [0, 1]^n$$

Si la valeur de c est petite, l'attaque réussit rarement, si elle est grande, elle réussit souvent mais la distance L2 peut être grande. La valeur de c est donc choisi par recherche dichotomique.

Afin de garantir que la modification produise une image valide, nous avons la contrainte $0 \leq x_i + \delta_i \leq 1$ pour tout i. Cette contrainte est connu sous le nom de "box constraint". Carlini utilise alors un changement de variable, on introduit w tel que $x + \delta = \frac{1}{2} \times (\tanh(w) + 1)$. Le problème d'optimisation est donc le suivant :

$$minimiser\ ||\frac{1}{2} \times (\tanh(w) + 1)) - x||_2 + c \times f(\frac{1}{2} \times (\tanh(w) + 1))\ tel\ que\ \tanh(w) \in [-1, 1]$$

Il est à noter que Carlini et Wagner présentent une attaque pour la norme infinie et une autre pour la norme 0, qui n'ont pas été implémentées dans le cadre de ce projet.

Toutes les attaques ci-haut ont été implémentées en leurs versions ciblée et non ciblée.

2.4 Entrainement adversarial

L'entrainement adversarial a été introduit par Goodfellow et al[2] puis alélioré par Madry et al[3]. Il consiste à enrichir le jeu de données d'entrainement avec des exemples adversariaux. À chaque étape de l'entrainement, la procédure devient donc :

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\max_{||\tau||_p \leq \epsilon} \mathcal{L}(f_\theta(x + \tau), y) \right].$$

Dans le cas où $p = \infty$, l'entrainement adversarial permet une bonne robustesse contre les attaques l_∞. Néanmoins, un entrainement sur la norme l_∞ est une protection faible contre les attaques l_2 et inversement.

3 Résultats

Le jeu de données utilisé pour le projet est CIFAR-10, un ensemble d'images de 32 par 32 pixels. Le réseau de neurones utilisé est constitué de 3 blocs composés d'une couche de convolution et d'une couche de "max pooling", suivi d'une couche d'Average Pooling, d'une couche fully connected avec Relu et d'une couche fully connected avec un Soft max.
Le tableau suivant présente la précision du modèle simple et du modèle avec entrainement adversarial pour les différentes attaques.

Modèles	sans attaques	FGSM	PGD l_∞	CW
sans entrainement adv.	0.76	0.041	0	0*
avec entrainement adv.	0.69	0.26	x	x

Table 1: Précision des modèles avec et sans entrainement adversarial sur les différentes attaques

L'attaque FGSM a été évaluée avec un epsilon à 0.05. PGD avec un epsilon de 0.09 pour la norme l_∞, un stepsize de 0.01, sur 45 itérations. L'attaque C&W étant plus lente, elle a été appliquée sur un échantillon réduit de 200 images avec un taux d'apprentissage 0.001 sur 1000 itérations.

Nous n'avons présenté ici que quelques résultats, les notebooks de démonstration peuvent servir de réference.

4 Travail prospectif

4.1 Double attaque adversariale

4.1.1 Idées générales

En tant que sujet d'exploration, nous avons voulu observer ce qui se produit quand on cherche à adversarialiser un exemple lui-même déjà adversarial. Ce thème de recherche n'apparait pas à notre connaissance dans la littérature. L'idée naïve à la base de cette exploration est qu'il serait peut-être possible de retourner à la classe d'origine d'un exemple adversarial en l'adverserialisant à nouveau. Par exemple, une image à l'origine de la classe A déguisée en B sera étonnamment facile à adversarialiser en A. La notion de facilité peut se traduire par le niveau de confiance que l'on arrive à atteindre dans une classe cible quand on réalise une attaque ciblée vers cette classe.
Sur la figure 1, on observe le nombre d'exemple de chaque classe donnée en abscisse qui ont pour classe d'aversarialisation privilégiée la classe en ordonnée. On peut alors faire une remarque évidente : pour chaque classe d'origine, certaines classes cibles sont naturellement privilégiées de par des similarités entre les classes (truck et automobile, bird et airplane etc...). En partant de ce constat très simple on peut arriver à une autre idée concernant les attaques adversariales doubles: si une attaque adversariale ne parvient pas à changer la facilité d'un exemple à être adversarialisé vers les autres classes, on peut reconnaitre des anomalies par ce principe. On appellera « signature adversariale », le vecteur constitué des confiances vers les différentes classes d'arrivée lors d'une attaque adversariale vers ces classes (pour CIFAR10 ce vecteur sera de taille 10). Ce qui est espéré, c'est qu'il est possible de reconnaitre la signature adversariale d'un exemple et que celle-ci est robuste aux attaques.

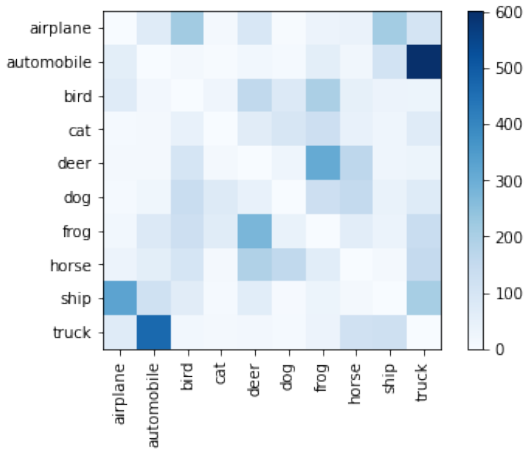

Figure 1: Matrice adversariale M pour une attaque FGSM ciblée. M est définie comme suit:
$M_{i,j}$ = nombre d'exemples de la classe i dont la meilleure classe cible est la classe j (meilleure au sens de la meilleure confiance dans la classe cible après attaque)

4.1.2 Etude de cas particulier : mode opératoire

L'expérience mise en place dans le cadre de ce projet est la suivante :

- Pour chaque couple de classes origine/cible, on sélectionne les exemples de la classe d'origine dont la classe cible est la classe cible privilégiée pour une attaque (au sens de la figure 1).
- On ne garde alors que les échantillons de taille assez importante pour que les résultats statistiques suivant aient du sens (par exemple on ne s'intéresse pas au cas airplane/dog qui ferait intervenir moins de 10 exemples). Pour commencer, on décide même de ne considérer comme classe cible, que la classe cible majoritaire de chaque classe d'origine (ex : airplane/bird, automobile/truck, etc...)

- On réalise une attaque FGSM ciblée (epsilon = 0.05) sur les exemples sélectionnés, de la classe d'origine vers la classe cible. On a alors constitué une population d'exemples adversariaux.

- On se munit des exemples de la classe d'origine, des exemples de la classe cible, et des exemples adversariaux construits à l'étape précédente. Pour ces trois populations, on calcule la signature adversariale de chaque exemple (on attaque vers toutes les classes et on mesure la confiance dans la classe à l'arrivée), et on réalise des histogrammes de ces signatures adversariales pour les 3 populations.

4.1.3 Résultats : validation des intuitions

Un exemple de résultat de la démarche décrite dans la section précédente est fourni en figure 2. Il s'agit du cas airplane/bird qui valide bien les idées générales annoncées précédemment.

Tout d'abord, examinons le premier histogramme. Il s'agit de la confiance en la classe airplane quand on adversarialise vers celle-ci. Sans surprise, la population airplane affiche des valeurs très proches de 1. Les exemples de la classe bird s'adversarialisent eux moins bien en airplane. Ce qui est particulièrement intéressant, c'est de voir que les exemples adversarialisés de la classe airplane vers la classe bird reviennent très bien vers la classe airplane et beaucoup mieux que les birds originaux. On confirme donc sur cet exemple le premier postulat : l'adversarial d'un adversarial est étonnamment bon.

Par ailleurs, il est également aisément observable que les exemples adversarialisés de la classe airplane vers la classe bird s'adversarialisent particulièrement mal vers les classes animales (dog, cat, frog). On retrouve ici une caractéristique de la classe airplane d'origine. Malgré l'attaque, la signature adversariale reste visible dans cet exemple.

Cet exemple présenté en figure 2 semble donc valider les idées générales présentées en première partie. Néanmoins, un tel succès n'est pas observé pour tous les couples de classes origine/cible. Un second exemple est donné en figure 3. Il s'agit du couple automobile/truck qui occupe une place un peu particulière, les deux classes étant très proches par nature et étant chacune la classe adversariale privilégiée de l'autre. On observe sur la figure 3, que le retour à la classe d'origine se fait facilement (plus facilement que pour les vrais exemples de la classe truck), cependant aucune autre différence notable de signature adversariale n'est observée entre les exemples de la classe truck et les automobiles adversarialisés en truck.

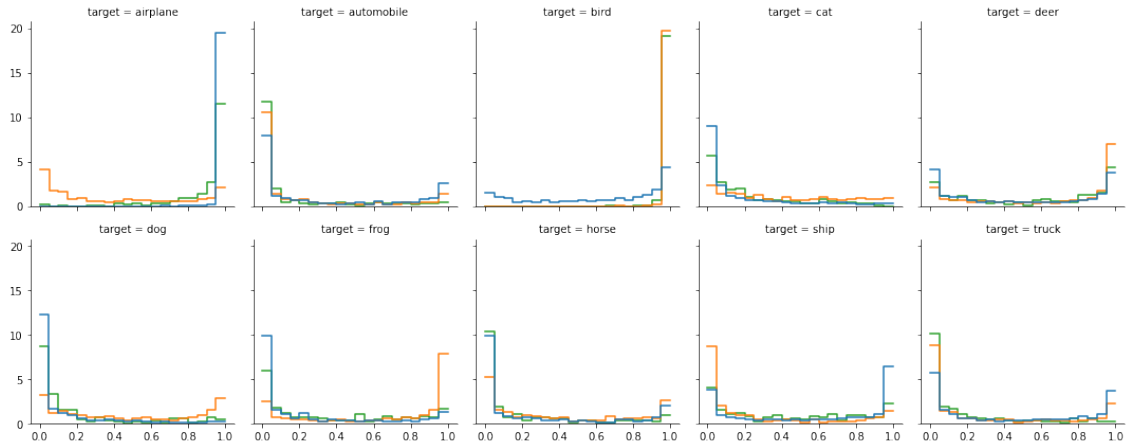

Figure 2: Pour chaque classe cible, histogramme des confiances en la classe cible après attaque FGSM ciblé vers cette classe. Trois populations sont représentées: en bleu, les exemples de la classe airplane, en orange ceux de la classe bird et en vert des exemples de la classe airplane adversarialisés en birds.

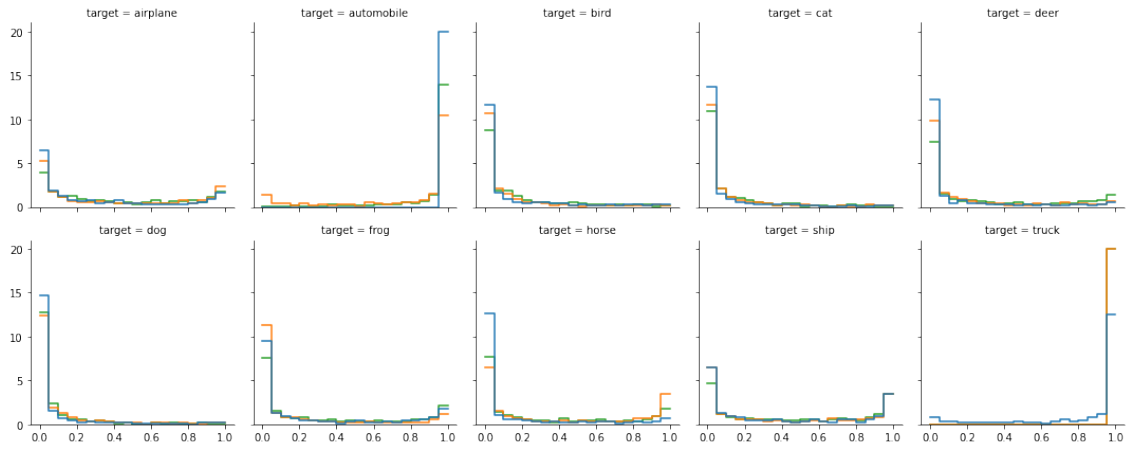

Figure 3: Pour chaque classe cible, histogramme des confiances en la classe cible après attaque FGSM ciblé vers cette classe. Trois populations sont représentées: en bleu, les exemples de la classe automobile, en orange ceux de la classe truck et en vert des exemples de la classe automobile adversarialisés en trucks.

4.1.4 Perspectives

Les résultats présentés précédemment mettent en perspective un moyen de détecter des exemple adversariaux comme des anomalies de signature adversariale. Plus le nombre de classes d'un data set sera important, plus on peut espérer que cette signature apportera de l'information et aura des chances de permettre la distinction entre un exemple naturel et adversarial. Néanmoins, le coût computationnel augmente alors également. Il serait alors envisageable d'interroger un sous-ensemble de classes cibles bien choisies. Si les différences entre signatures adversariales d'exemples naturels et adversariaux sont assez notables, il est alors possible d'entrainer un classifieur basé sur les signatures adversariales pour différencier les deux populations.

4.2 Inspection des espaces latents du réseau de neurones

Le but de cette inspection est en premier lieu de déterminer à quel niveau du réseau l'attaque agirait (si elle agit sur une couche en particulier) pour potentiellement limiter le calcul des gradients à ces couches lors d'une attaque ou d'un entrainement adversarial pour le gain que ceci representerait; ce n'est toutefois pas cette perspective que nous explorons, en effet, après avoir identifié cette couche en question nous avons plutot tenté de determiner une différence entre exemples adversariaux et exemples normaux, étant la direction que les observations suggéraient.

L'attaque utilisée ici est FGSM.

4.2.1 Identification de la couche d'action de l'attaque: K plus proches voisins

Afin de visualiser les changements le long du réseau de neurones, l'algorithme des "k-plus proches voisins" (KNN) est appliqué aux images de l'ensemble de test, leur version bruitée et leur version adversarialisée sur l'ensemble d'entrainement.Nous avons aussi étudié le comportement de données bruitées pour vérifier que la différence de comportement observée entre données normales et données adversariales est bien due à leur caractère "d'adversariales". Les tables 2 à 5 présentent les résultats d'une image de cheval. L'image est adversarialisée en chat, le nombre de voisins est fixé à 100.

Images	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
normale	0	0	20	15	33	8	13	8	2	1
bruitée	0	0	20	10	39	4	19	7	0	1
adversariale	0	0	19	12	38	5	16	6	3	1

Table 2: 100 plus proches vosins après la deuxième couche de convolution + max pooling

Images	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
normale	1	0	11	15	39	11	11	9	3	0
bruitée	1	0	8	13	44	12	14	7	1	0
adversariale	2	0	6	16	42	5	16	6	4	3

Table 3: 100 plus proches vosins après la troisième couche de convolution + max pooling

Images	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
normale	2	1	4	16	22	17	11	20	0	7
bruitée	1	1	5	15	23	11	17	18	0	9
adversariale	1	9	2	18	19	9	7	7	0	28

Table 4: 100 plus proches vosins après la couche "Average Pooling"

Images	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
normale	3	0	2	33	6	12	3	41	0	0
bruitée	2	0	5	27	12	10	4	40	0	0
adversariale	6	2	3	59	2	14	5	1	1	7

Table 5: 100 plus proches vosins après la première couche entièrement connectée

4.2.2 Différenciation entre exemples adversariaux et exemples originaux selon leurs composantes: tentative

Les résultats montrent donc des changements notables au niveau de la dernière couche de convolution et de la première couche entièrement connectée, laissant comprendre que la première couche entièrement connectée amplifie des différences justement présentes entre les vecteurs adversariaux et originaux en entrée, (c'est-à-dire qu'elle aurait des poids pour des composantes qui étaient justement différentes entre ces vecteurs, amplification suite à laquelle la divergence d'un exemple adversarial de ses plus proches voisins est plus notable (résultats de l'opération KNN dans l'espace latent en sortie de cette couche)).

La direction prise était donc en premier lieu de vérifier l'existence de telles composantes sur lesquelles les exemples adversariaux diffèreraient des exemples originaux et ensuite d'inspecter à quoi correspondraient ces composantes, en supposant qu'elles correspondraient à des caractéristiques des images, sur lesquelles l'attaque ayant généré les exemples adversariaux agirait.

Pour la première vérification nous opérons une ACP sur les vecteurs en question[1] et extrayons les composantes pour lesquelles la différence absolue entre un vecteur adversarial et le vecteur original dépasse un

[1]L'ACP est d'une part pour réduire la dimension des vecteurs (de 512 à 40) et ayant aussi pour effet, en quelque sorte, de regrouper les composantes, se qui serait plus proche de notre vision : les composantes correspondraient à des caractéristiques dans les images

certain seuil [2].

Nous observons qu'un certain groupe de composantes revient dans les comparaisons, nous pourions les considérer comme celles caractérisant la différence entre un exemple advesarial et un exemple original (mais pas entre un exemple **effectivement** adversarial (i.e qui aurait effectivement "trompé" le réseau) et un original, car elles sont aussi observées sur des exemples adversarieux qui ont toutefois été correctement classés (deux dernières paires de la table 5)), ou du moins, comme celles sur lesquelles l'attaque semble agir .

Paire (exemple adversarial - prediction)	composantes différentes entre l'exemple adversarial et l'original
cat-frog	[1]
deer-horse	[1 29]
cat-airplane	[1 9 13 29]
cat-dog	[1 9 13 29]
cat-ship	[1 9 13 29]
airplane-ship	[0 1 4 5 8 9 13 18 24 29 36 37]
airplane-airplane	[1 9 13 29]
ship-ship	[1 29]

Table 6: Composantes sur lesquelles des exemples adversariaux diffèrent des originaux au delà d'un certain seuil. Les composantes 1,29,9,13 ou encore 0,4,5,8...etc pour les classes 'ship' et 'automobile' étaient les seules (sur 40 composantes) qui revenaient sur tous les exemples.

Nous avons tenté d'inspecter à quoi correspondraient ces composantes en visualisant les matrices en sortie des couches précédentes du réseau (les trois couches convolutives) sur tous les plans, or la comparaison était non conclusive puisque nous n'avons pu observer aucune différence entre les matrices adversariales et celles originales.

Une autre tentative d'inspection de l'évolution des distances à travers le réseau consistait en le calcul de la distance moyenne entre les vecteurs latents adversariaux et ceux des classes que le réseau prédit pour ceux-ci, normalisés et tous réduits en une même dimensionalité (10)[3] . Les courbes obtenues suggèrent une décroissance de cette distance à travers les couches du réseau, pouvant laisser penser que plus on avance dans les couches plus l'exemple est confondu avec la fausse prédiction que le réseau fera, pouvant rejoindre le fait que plus un réseau est profond plus il est ≪ attaquable ≫ , toutefois nous ne postulons pas ceci, car non sûrs de pouvoir étudier l'évolution d'une distance sur des espaces différents (bien que, comme mentionné, ceux-ci ont été réduits en la même dimension) et aussi parce que nous observons la même décroissance sur des paires ≪ vecteur adversarial, vecteur original ≫ pour lesquelles la courbe de la distance devrait être croissante (si la supposition que l'exemple s'approcherait de la fausse prédiction en avançant dans le réseau il s'éloignerait de l'exemple original).

5 Conclusion

Une conclusion serait que de résumer le travail effectué: Nous avons pu observer des conclusions reportées dans la littérature sur les trois attaques implémentées et l'entrainement adversarial, avons introduit la notion de "double attaque" comme potentiellement donnant une caractérisation des exemples adversariaux et donc

[2]le seuil est pris comme étant deux fois la moyenne du vecteur original, mais son choix est non important, il suffit de prendre un seuil à partir duquel l'on commence à avoir des sorties mais qui n'est pas trop petit de sorte à avoir toutes les composantes par exemple)

[3]La normalisation est un centrage-réduction, la dimensionalité de 10 est prise comme étant le minimum des dimensionalités des espaces latents. La moyenne est sur les exemples induisant une fausse prédiction sur les 10000 exemples test

un moyen de les détecter en vue d'une défense et avons tenté d'inspecter le comportement du réseau (ou alors des exemples adversariaux dans celui-ci), toutefois de manière non conclusive.

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017.

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[3] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.