Ataque adversarios

Enrique David Guzmán Ramírez
IIMAS, Universidad Nacional Autónoma de México

Introducción al Aprendizaje Profundo
M. en C. Berenice Montalvo Lezama
M. en C. Ricardo Montalvo Lezama

14 de junio de 2021

Resumen

Las redes neuronales profundas son modelos poderosos que se han utilizado ampliamente para lograr un rendimiento cercano al nivel humano en una variedad de tareas. Sin embargo, a pesar de su desempeño superior estudios recientes han encontrado que incluso los modelos del estado del arte son sumamente vulnerables a ataques adversarios. En el presente trabajo se exploran algunos de los algoritmos de ataques adversarios en algunas de las arquitecturas más recientes de visión computacional.

1. Introducción

Un ataque adversario es una muestra de datos de entrada que ha sido perturbada levemente con la intención de hacer fallar a un clasificador
En muchos casos, estas modificaciones pueden ser tan sutiles que un observador humano ni siquiera nota la modificación, pero el clasificador comete un error. Los ejemplos de adversarios plantean problemas de seguridad porque podrían usarse para realizar un ataque a los sistemas de aprendizaje automático, incluso si el atacante no tiene acceso al modelo subyacente.
Con la introducción de modelos de aprendizaje profundo en cada vez más distintos aspectos de nuestra vida, los problemas que estos ataques adversarios pueden ocasionar son preocupantes.

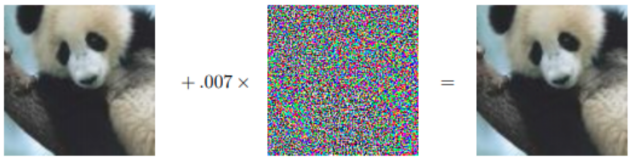


Figura 1: Ejemplo de ataque adversario. Imagen tomada de *Explaining and harnessing adversarial examples.*

Los escenarios de posibles ataques adversarios se pueden clasificar en diferentes maneras:

- **Ataque no dirigido:** el objetivo es hacer que el clasificador prediga una etiqueta incorrecta, la etiqueta incorrecta específica no importa.
- **Ataque dirigido:** el objetivo es cambiar la predicción del clasificador a alguna clase objetivo específica.

En segundo lugar, los escenarios de ataque se pueden clasificar por la cantidad de conocimiento que el adversario tiene sobre el modelo:

- **Caja negra:** el atacante no sabe mucho sobre el modelo, pero puede sondear o consultar el modelo, es decir, darle algunas entradas y observar salidas.
- **Caja blanca:** el atacante tiene pleno conocimiento del modelo, como la arquitectura del modelo y los valores de todos los parámetros y pesos entrenables.

1.1. Algoritmos de ataques adversarios

Hay una multitud de ataques adversarios, pero particularmente usaré los siguientes cuatro:

1. **FGSM (Fast Gradient Sign Method)** [1]: la idea es generar el ejemplo adversario $\boldsymbol{x}_{\text{adv}}$ de la siguiente forma

$$\boldsymbol{x}_{\text{adv}} = \boldsymbol{x} + \epsilon \cdot \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right).$$

   donde ϵ es el orden de la perturbación, J es la función de pérdida (usualmente cross-entropy), $\boldsymbol{\theta}$ los pesos del modelo, $\boldsymbol{x}$ la imagen original y y la etiqueta.

2. **PGD (Projected Gradient Descent)** [2]: podemos verlo como una variante de varios pasos, donde α es la magnitud de la perturbación en cada paso

$$\boldsymbol{x}_{\text{adv}}^{t+1} = \text{Proj}\left(\boldsymbol{x}_{\text{adv}}^{t} + \alpha \cdot \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right)\right)$$

3. **MIFGSM (Momentum Iterative Fast Gradient Sign Method)** [3]: algoritmo ganador de

esta competencia, inspirados por los optimizadores con momento proponen

$$\boldsymbol{x}_{\text{adv}}^{t+1} = \text{Clip}\left(\boldsymbol{x}_{\text{adv}}^{t} + \alpha \cdot \text{sign}(g_{t+1})\right),$$

donde $g_{t+1} = \xi \cdot g_t + \dfrac{\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)}{\|J(\boldsymbol{\theta}, \boldsymbol{x}, y)\|}$, donde ξ es un factor de decaimiento.

4. **One Pixel Attack** [4]: plantea el problema como un problema de optimización

$$\max_{\boldsymbol{x}_{\text{adv}}} f_{\text{adv}}(\boldsymbol{x} + \boldsymbol{x}_{\text{adv}}),$$
$$\text{con la restricción } \|\boldsymbol{x}_{\text{adv}}\| \leq d,$$

donde $d = 1$ para el caso de un ataque a un pixel. Particularmente proponen resolver este problema de optimización con *evolución diferencial*.

2. Objetivo

El objetivo del proyecto es explorar y desarrollar algunas de las técnicas usadas para generar ataques adversarios así como explorar algunos de los métodos de defensa que existen.

3. Propuesta de solución

Hay una variedad de arquitecturas preentrenadas en PyTorch con el conjunto de datos de ImageNet que sirven a la perfección para esta tarea, particularmente usaré las siguientes

- AlexNet
- Resnet18
- Inception v3
- MobileNet v2

Con estas arquitecturas y métodos se exploraron los ataques y defensas adversarias.

- Para los ataques de caja blanca los ejemplos adversarios se hacen a la medida para cada modelo, posteriormente podemos evaluar el accuracy con cada ataque.
- Para el ataque de caja negra se generaron los ejemplos adversarios con Inception v3 y se le pasaron a MobileNet v2.
- Para las defensas usé una versión modificada de MobileNet v2 para que use CIFAR10. Entrené desde cero usando el dataset de entrenamiento de CIFAR10 y posteriomente entrené otro modelo agregando 20,000 ejemplos adversarios al dataset, lo que hace que el modelo sea más robusto a ataques adversarios.

4. Resultados

Con los ataques de caja blanca, donde los ejemplos adversarios se hacen a la medida de cada modelo, la disminución en el accuracy es enorme.

Modelo	Limpio (acc@1/acc@5)	FGSM (acc@1/acc@5)	PGD (acc@1/acc@5)	MIFGSM (acc@1/acc@5)	OnePixel (acc@1/acc@5)
AlexNet	60.9 / 84.6	6.0 / 28.6	2.9 / 19.3	3.5 / 21.0	58.3 / 83.4
ResNet-18	82.5 / 95.4	3.5 / 24.3	0.8 / 14.6	1.0 / 13.5	78.7 / 94.5
Inception v3	76.5 / 93.1	10.8 / 39.3	3.9 / 28.1	5.1 / 27.6	70.1 / 91.8
MobileNet v2	85.0 / 97.3	3.5 / 24.4	0.5 / 11.3	0.6 / 8.8	81.4 / 96.6

Tabla 1: Ataques de caja blanca.

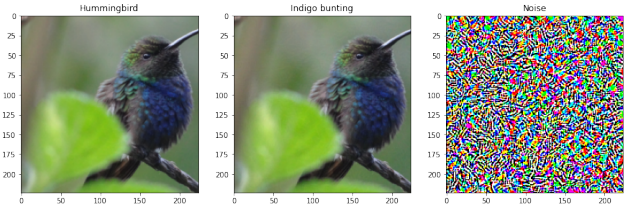


Figura 2: Ataque de caja blanca a AlexNet con FGSM.

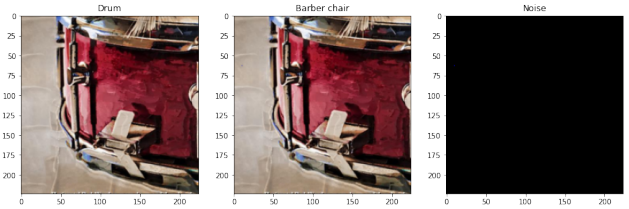


Figura 3: Ataque de caja blanca a Inception v3 con OnePixel.

En el caso del ataque de caja negra hice los ataques más agresivos, aunque no es tan efectivo como un ataque de caja blanca, se logra bajar considerablemente el accuracy del modelo.

Modelo	Limpio (acc@1/acc@5)	FGSM (acc@1/acc@5)	PGD (acc@1/acc@5)	MIFGSM (acc@1/acc@5)	OnePixel (acc@1/acc@5)
MobileNet v2	85.0 / 97.3	51.7 / 78.3	59.4 / 82.9	56.3 / 81.5	83.5 / 97.0

Tabla 2: Ataque de caja negra a MobileNet v2.

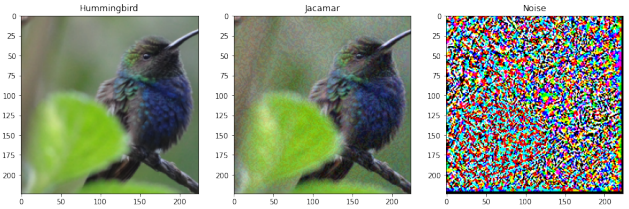


Figura 4: Ataque de caja negra a MobileNet v2 con FGSM.

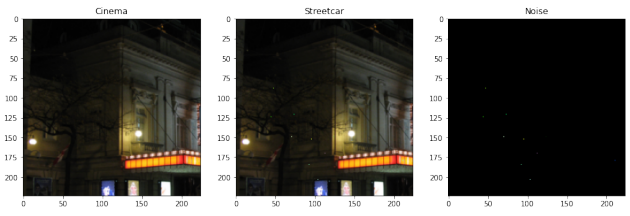
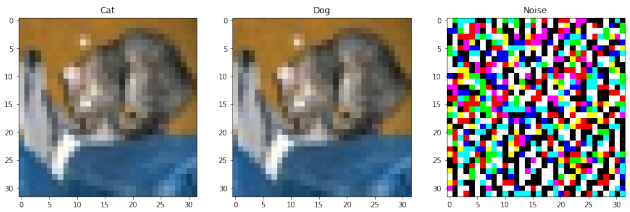


Figura 5: Ataque de caja negra a MobileNet v2 con OnePixel.

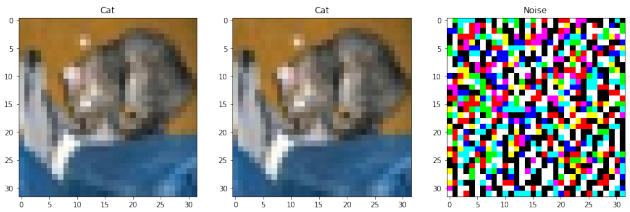
En cuanto al entrenamiento con ejemplos adversarios, se confirma que es una técnica sencilla pero eficiente.

Modelo	Limpio (accuracy)	FGSM (accuracy)	PGD (accuracy)	MIFGSM (accuracy)	OnePixel (accuracy)
MobileNet v2	81.68	37.03	28.29	31.27	63.17
MobileNet v2 adversarial	80.62	77.44	77.49	77.35	77.58

Tabla 3: Entrenamiento sin y con ejemplos adversarios en CIFAR10 con una versión modificada de MobileNet v2.



(a) Ataque adversario con FGSM a MobileNet v2 entrenada sin ejemplos adversarios.



(b) Ataque adversario con FGSM a MobileNet v2 entrenada con ejemplos adversarios.

Figura 6: Entrenamiento adversario.

5. Conclusiones

- Los ataques adversarios son un fenómeno interesante y un problema importante en la seguridad del aprendizaje automático, por lo que es relevante hacer notar a la comunidad de este problema.
- Ataques relativamente simples pueden engañar fácilmente incluso a los modelos más recientes sin que un humano llegue a notarlos.
- El estudio de estos ataques a su vez nos puede ayudar a generar defensas para hacer a los modelos más robustos, confiables y seguros.

Referencias

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

[2] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2019.

[3] Y. Dong, F. Liao, T. Pang, X. Hu, and J. Zhu, "Discovering adversarial examples with momentum," *CoRR*, vol. abs/1710.06081, 2017.

[4] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *CoRR*, vol. abs/1710.08864, 2017.