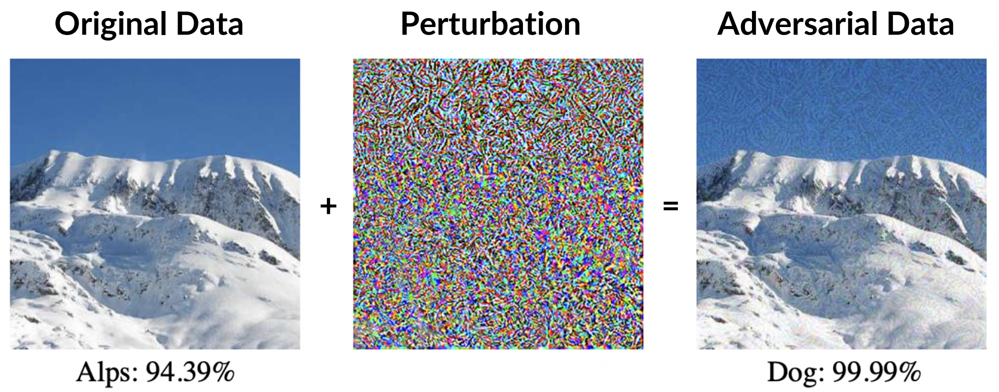
Adversarial Attack and Defense

Jaechul Roh (ID: 20473590)

YouTube Video Link: **https://youtu.be/maMC93Lf-mY**

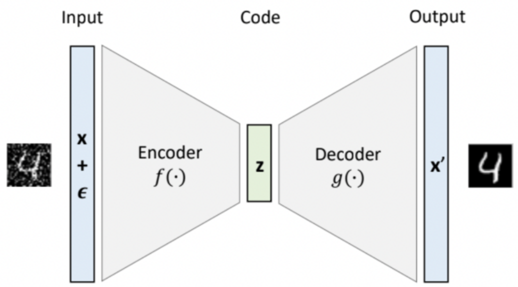
Introduction

- **Adversarial Examples** input data with an imperceptible change
- **Adversarial Examples** = Original data (x) + Perturbation with noise (ϵ)
- **Adversarial Attack** induce misclassification in purpose to make machine learning models more **ROBUST**



Course related material

Stacked Denoising Autoencoder = The **NOISY INPUT** will be inputted to denoising autoencoder, which will learn how to recover the original input (x). Such method will help to create a **MORE ROBUST CODE**, so that the model will **NOT BE SENSITIVE TOWARDS NOISY INPUTS**.

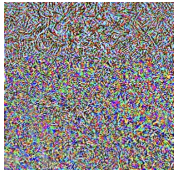


Real-life adversarial attack example

(1) INPUT

Original Data

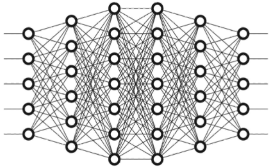
+



Noise

(2) ADVERSARIAL EXAMPLE

(3) DEEP LEARNING MODEL



”Green”

(4) OUTPUT

MOVE FORWARD

(5) ACTION

(6) CONSEQUENCE

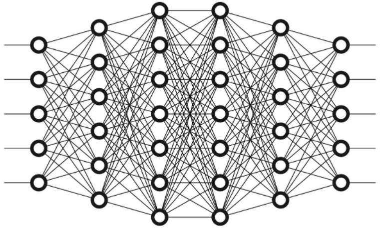
Adversarial Defense

ATTACK (Step 1)

Input

Adversarial Examples

Deep learning Model



Output

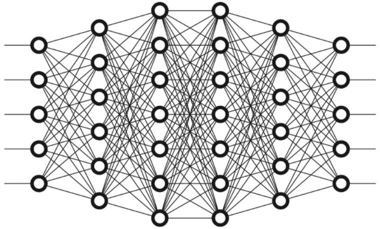
<u>**”Green”**</u>

DEFENSE (Step 2)

Clean data

+

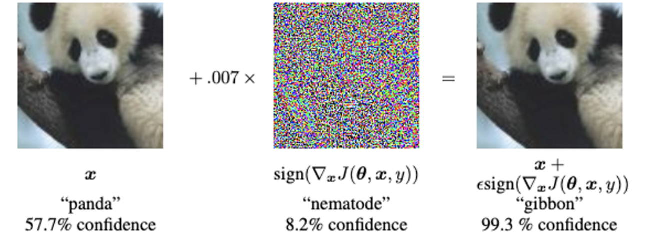
Adversarial Examples



<u>**“Red”**</u>

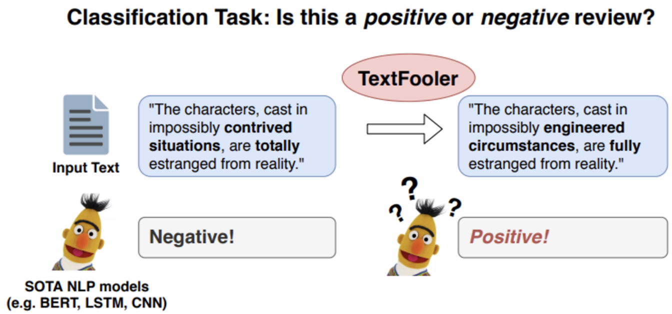
Representative models/algorithms

(Image Credit: (Goodfellow et al. 2014b))



- *Explaining and Harnessing Adversarial Examples (2015)*
  - *Ian.J.Goodfellow, Jonathon Shlens & Christian Szegedy*

- *Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment*
  - *Di Jin, Zhijing Jin, Joey Tianyi Zhou, Peter Szolovits*



Explaining and Harnessing Adversarial Examples

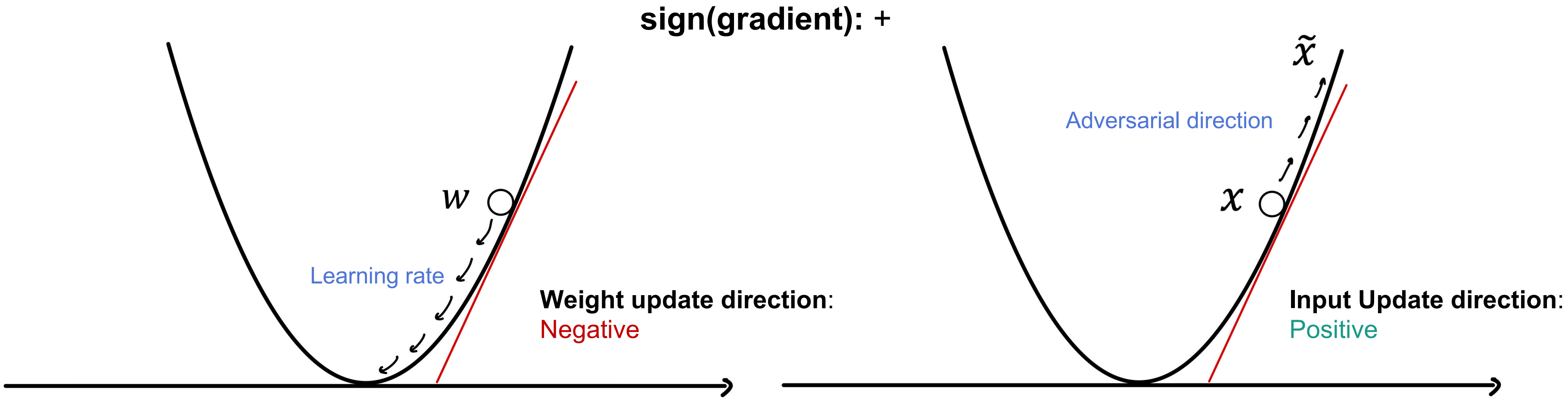
Fast Gradient Sign Method (FGSM)

- Gradient Descent Method

  OPPOSITE direction of the gradient of the cost function

- Fast Gradient Sign Method (FGSM)

  SAME direction of the gradient of the cost function



How adversarial example is formed

$$\underbrace{x + \epsilon \cdot sign\left(\underbrace{\nabla_x \underbrace{J(\theta, x, y)}_{\text{Cost Function}}}_{\text{Gradient}}\right)}_{\text{Adversarial Example}}$$

FGSM (Continued)

input
(panda image)

x ("panda")

$$+ \quad \epsilon \cdot sign(\nabla_x J(\theta, x, y)) \quad =$$

x^* ("gibbon")

Parameters
of the model

output
("panda" label)

Mathematical Notation and Concepts

activation value
(with perturbation)

$$w^T \tilde{x} = w^T x + w^T \eta$$

activation growth

original desired output

perturbation

$$\eta = \epsilon \cdot sign(\nabla_{x^*} J(\theta, x, y))$$

Deciding perturbation

FGSM uses the *"max norm constraint":*

$$(In\ all\ definitions, x = (x_1, x_1, \dots, x_n))$$

$$L^{\infty}\ distance:\ \| x \|_{\infty} = \max_{1 \le i \le n} |x_i|$$

$$L^{1}\ distance:\ \| x \|_{1} = \sum_{i=1}^{n} |x_i|$$

L^{∞} : moving as many pixels as possible but only by a small number

L^{1} : summed absolute value difference between *x* and *x**

Example 1: 1-Dimensional Calculation

$$w^T\tilde{x} = w^Tx + w^T\eta = w^T(x + \eta)$$

$$\begin{array}{ccccc} x & & w & & w^Tx \\ \begin{pmatrix} 3 \\ -2 \\ 5 \end{pmatrix} & \begin{matrix} * \\ * \\ * \end{matrix} & \begin{pmatrix} 7 \\ 10 \\ 20 \end{pmatrix} & = & \begin{pmatrix} 21 \\ -20 \\ 100 \end{pmatrix} \end{array} \Rightarrow 101$$

activation value
(WITHOUT perturbation)

$$\eta = sign(w)$$

$$sign\left(\begin{pmatrix} 7 \\ 10 \\ 20 \end{pmatrix}\right) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{array}{ccccc} & x & & \eta & & w & & w^T\tilde{x} \\ \Bigg[& \begin{pmatrix} 3 \\ -2 \\ 5 \end{pmatrix} & + & \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} & \Bigg] * & \begin{pmatrix} 7 \\ 10 \\ 20 \end{pmatrix} & = & \begin{pmatrix} 28 \\ -10 \\ 120 \end{pmatrix} \end{array} \Rightarrow 138$$

activation value
(WITH perturbation)

Example 2: 3-Dimensional Calculation

$$x + \epsilon \cdot sign\big(\nabla_x J(\theta, x, y)\big)$$

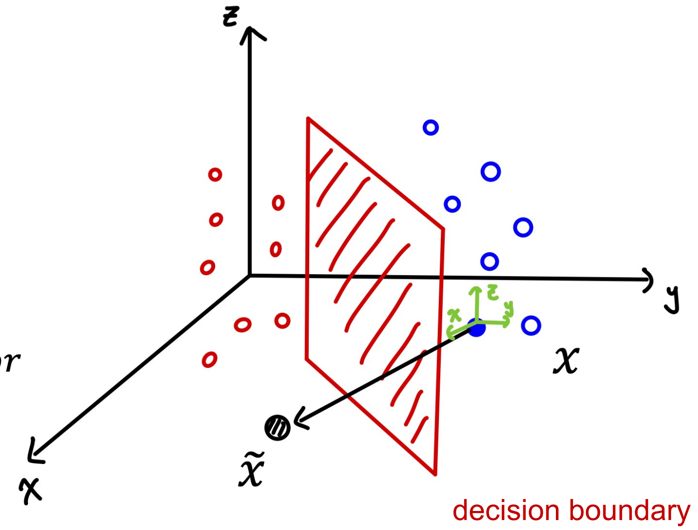
$sign(w_x) \rightarrow$ POSITIVE

$sign(w_y) \rightarrow$ NEGATIVE x ϵ_{vector} = $-\epsilon$ + x_{vector}

$sign(w_z) \rightarrow$ POSITIVE

$$\begin{bmatrix} +\epsilon \\ -\epsilon \\ +\epsilon \end{bmatrix}$$

= x^*_{vector}



Adversarial Defense (FGSM)

$$\underset{(3)}{\tilde{J}(\theta, x, y)} = \alpha \cdot \underset{(1)}{J(\theta, x, y)} + (1 - \alpha) \cdot \underset{(2)}{J(\theta, \tilde{x}, y)}$$

α : **proportion** to use between the original data and the adversarial example

(1) $\tilde{J}(\theta, x, y)$: cost function of the **original data**

(2) $J(\theta, \tilde{x}, y)$: cost function of the **adversarial example**

(3) $J(\theta, x, y)$: cost function of both **original data AND adversarial example**

Adversarial Attack in Natural Language Processing

Hardship of natural language adversarial attack

Image domain (CONTINUOUS values)

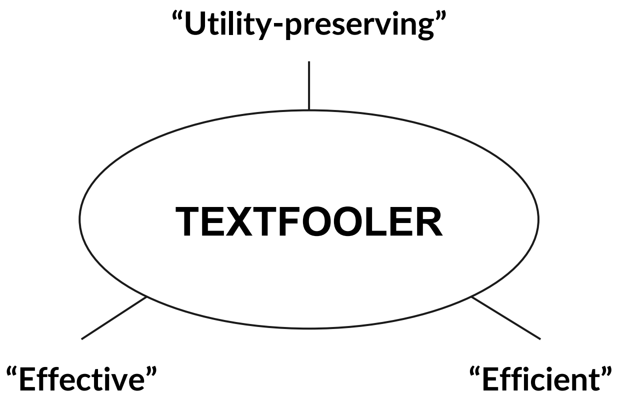
Adding a minimal noise to the pixels is not noticeable through naked eyes

Text domain (DISCRETE values)

The difference between the original text and the adversarial example is easily recognizable

Introduction

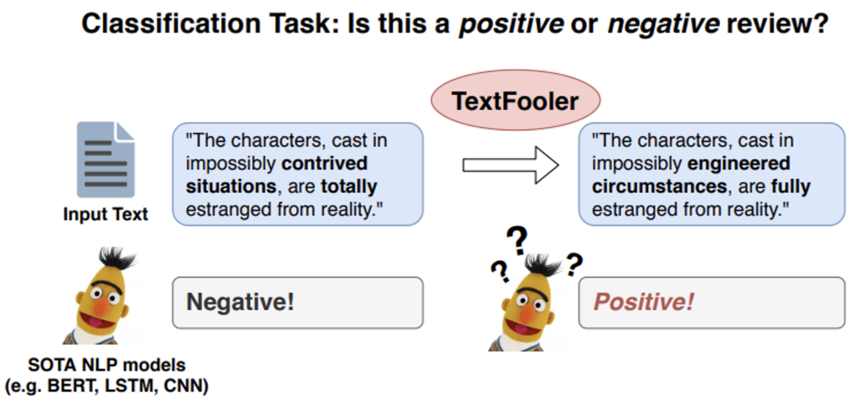
- Proposing *TextFooler*



- How to test such the robustness?

Models: 1. WordLSTM 2. WordCNN 3. BERT

Task: 5 classification tasks and 2 textual entailment tasks



Classification & Recognizing Text Entailment (NLP tasks)

INPUT SENTENCE (*X*)

“The movie seems to very boring, but quite interesting. However, I won’t watch it again”

P (Positive) = 0.05

P (Negative) = 0.85

P (Neutral) = 0.10

Classification label (*Y*) = -1: “NEGATIVE”

HYPOTHESIS SENTENCE

Input: “It was a very touching movie”

Entailment	Contradiction	Neutral
“The movie made me cry”	“The movie made me laugh”	“There was no discount for the movie ticket”

Forming text adversarial example

- **Text adversarial example** need to meet the following **requirement**:

$$\underbrace{F(X_{\mathrm{adv}})}_{\text{Classification result of adversarial text}} \neq \underbrace{F(X)}_{\text{Classification result of original data}}, \text{and } \underbrace{\mathrm{Sim}(X_{\mathrm{adv}}, X)}_{\text{Semantic similarity between } X_{adv} \text{ and } X} \geq \underbrace{\epsilon}_{\text{Minimum similarity}},$$

TEXTFOOLER Attack Algorithm

Steps:

1. **Word Importance Score**
2. **Word Transformer**
   a. Adversarial example candidates POS (Part of Speech) checking
   b. Semantic Similarity Filter
   c. Finalizing Adversarial Example

Algorithm 1 Adversarial Attack by TEXTFOOLER

Input: Sentence example $X = \{w_1, w_2, \ldots, w_n\}$, the corresponding ground truth label Y, target model F, sentence similarity function $\mathrm{Sim}(\cdot)$, sentence similarity threshold ϵ, word embeddings Emb over the vocabulary Vocab.

Output: Adversarial example X_{adv}

Initialization: $X_{\mathrm{adv}} \leftarrow X$
for each word w_i in X **do**
    Compute the importance score I_{w_i} via Eq. (2)
end for

Create a set W of all words $w_i \in X$ sorted by the descending order of their importance score I_{w_i}.
Filter out the stop words in W.
for each word w_j in W **do**
    Initiate the set of candidates CANDIDATES by extracting the top N synonyms using CosSim(Emb_{w_j}, $\mathrm{Emb}_{\mathrm{word}}$) for each word in Vocab.
    CANDIDATES $\leftarrow$ POSFilter(CANDIDATES)
    FINCANDIDATES $\leftarrow \{\ \}$
    for c_k in CANDIDATES **do**
        $X' \leftarrow$ Replace w_j with c_k in X_{adv}
        if $\mathrm{Sim}(X', X_{\mathrm{adv}}) > \epsilon$ **then**
            Add c_k to the set FINCANDIDATES
            $Y_k \leftarrow F(X')$
            $P_k \leftarrow F_{Y_k}(X')$
        end if
    end for
    if there exists c_k whose prediction result $Y_k \neq Y$ **then**
        In FINCANDIDATES, only keep the candidates c_k whose prediction result $Y_k \neq Y$
        $c^* \leftarrow \underset{c \in \mathrm{FINCANDIDATES}}{\operatorname{argmax}} \mathrm{Sim}(X, X'_{w_j \rightarrow c})$
        $X_{\mathrm{adv}} \leftarrow$ Replace w_j with c^* in X_{adv}
        return X_{adv}
    else if $P_{Y_k}(X_{\mathrm{adv}}) > \underset{c_k \in \mathrm{FINCANDIDATES}}{\min} P_k$ **then**
        $c^* \leftarrow \underset{c_k \in \mathrm{FINCANDIDATES}}{\operatorname{argmin}} P_k$
        $X_{\mathrm{adv}} \leftarrow$ Replace w_j with c^* in X_{adv}
    end if
end for
return None

1. Word Importance Ranking

"Measuring the influence the word, w_i "

Input without the word, w_i

$$I_{w_i} = \begin{cases} F_Y(X) - F_Y(X_{\backslash w_i}), & \text{if } F(X) = F(X_{\backslash w_i}) = Y \\ (F_Y(X) - F_Y(X_{\backslash w_i})) + (F_{\bar{Y}}(X_{\backslash w_i}) - F_{\bar{Y}}(X)), \\ \quad \text{if } F(X) = Y, F(X_{\backslash w_i}) = \bar{Y}, \text{ and } Y \neq \bar{Y}. \end{cases}$$

Classification output

Two different labels

"Prediction change before, and after the word, w_i "

a. Candidates and POS Checking

Output: Adversarial example X_{adv}

Initialization: $X_{\text{adv}} \leftarrow X$
for each word w_i in X **do**
(1) Compute the importance score I_{w_i} via Eq. (2)
end for

for each word w_j in W **do** (3)
Initiate the set of candidates CANDIDATES by extracting the top N synonyms using CosSim(Emb_{w_j}, Emb_{word}) for each word in Vocab. (2)
CANDIDATES $\leftarrow$ POSFilter(CANDIDATES) (4)

(1): Process **importance score** for every word in the **sentence example**

(2): **Cosine Similarity Score** between the Embedding(deleting word) and Embedding(Vocab)

(3): Extract top N synonyms and append to **CANDIDATES** list

(4): Check **POS** (Part of Speech) for every candidate word and filter

b. Semantic Similarity Filter

(3) **Cosine Similarity** between ***X (original sentence) and Xadv (Adversarial Example)***

FINCANDIDATES $\leftarrow \{\}$
for c_k in CANDIDATES **do** (2)
(1) $X' \leftarrow$ Replace w_j with c_k in X_{adv}
if $Sim(X', X_{adv}) > \epsilon$ **then** (4)
(3) Add c_k to the set FINCANDIDATES
$Y_k \leftarrow F(X')$
$P_k \leftarrow F_{Y_k}(X')$
end if
end for

(1) Substitute that specific word in the sentence with each of the words in the **CANDIDATES**

(2) Such sentence becomes *Xadv (Adversarial Example)*

(4) Words with **similarity score > ϵ** (defined by the programmer) will be stored in **FINCANDIDATES** list

c. Finalizing the Adversarial Example

In the **descending order** of similarity scores, replace the word:

(1)

if there exists c_k whose prediction result $Y_k \neq Y$ **then**
  In FinCandidates, only keep the candidates c_k whose prediction result $Y_k \neq Y$
  $c^* \leftarrow \underset{c \in \text{FinCandidates}}{\operatorname{argmax}} \; \text{Sim}(X, X'_{w_j \rightarrow c})$
  $X_{\text{adv}} \leftarrow$ Replace w_j with c^* in X_{adv}
  return X_{adv}
else if $P_{Y_k}(X_{\text{adv}}) > \underset{c_k \in \text{FinCandidates}}{\min} P_k$ **then**
  $c^* \leftarrow \underset{c_k \in \text{FinCandidates}}{\operatorname{argmin}} \; P_k$
  $X_{\text{adv}} \leftarrow$ Replace w_j with c^* in X_{adv}
end if
end for
return None

(2)

(1) IF the **prediction of the target model changes:**

- Within those candidates that **changed the output of the target model**
- Select the word that had the **highest similarity score between X and *Xadv.***

(2) ELSE IF choose the word with the **least confidence level**

(word that is most likely to change the prediction of the model)

- **Prediction changed → Attack Success!**

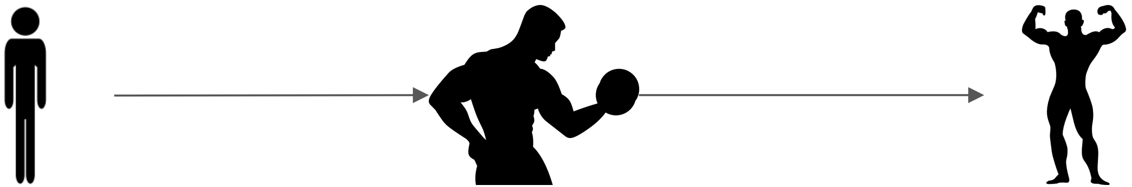
Run through this process in the descending order of importance score of each word.

Summary

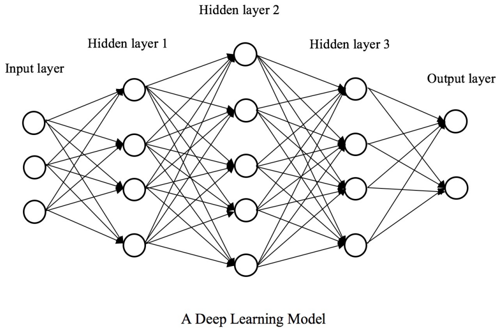
Adversarial ATTACK

Adversarial TRAINING

ROBUST
deep learning model



x^* ("gibbon")



x ("panda")

Reference

- Attack & Defense (1): Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572* (2014).
- Attack & Defense (2): Jin, Di, et al. "Is bert really robust? a strong baseline for natural language attack on text classification and entailment." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. No. 05. 2020.
- Kariya, Mahendra. “Paper Discussion: Explaining and Harnessing Adversarial Examples.” *Medium*, Medium, 16 Nov. 2018, https://medium.com/@mahendrakariya/paper-discussion-explaining-and-harnessing-adversarial-examples-908a1b7123b5.
- Dong, Yinpeng, et al. "Boosting adversarial attacks with momentum." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- **1-dimensional example:** “Explaining And Harnessing Adversarial Examples 논문-리뷰.” *Explaining And Harnessing Adversarial Examples 논문-리뷰*, 10 July 2020, velog.io/@miai0112/Explaining-And-Harnessing-Adversarial-Examples-%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0.
- “[텍스트 분류 모델 공격 기법] TextFooler: Is BERT Really Robust?” *YouTube*, 22 Nov. 2020, www.youtube.com/watch?v=EF-IYFTKZiE&t=992s.
- Pan, Zhixin, and Prabhat Mishra. "Fast Approximate Spectral Normalization for Robust Deep Neural Networks." *arXiv preprint arXiv:2103.13815* (2021).
- Cho, Yoon Sang. “Adversarial Attacks and Defenses in Deep Learing.” *Adversarial Attacks and Defenses in Deep Learing*, 2020, pp. 5–32, dmqm.korea.ac.kr/activity/seminar/289.

Thank you!

Jaechul Roh (ID: 20473590)