Adversarial Examples and Adversarial Training

Jun Yan

University of Southern California
yanjun@usc.edu

October 30, 2019

1 Introduction

Deep Neural Networks play an important role in the rapid development of Artificial Intelligence. Deep learning models have achieved excellent results in many real world applications. However, it has been shown that deep learning models can be easily fooled by well-designed input samples, which casts doubt on their robustness.

In this paper, we mainly follow the chronological order to introduce some important works on adversarial examples and adversarial training.

2 Adversarial Examples

Adversarial examples are malicious inputs purposely designed to fool a machine learning model. They are first studied on the image classification task. The goal of these works is to slightly modify the input image to make it misclassified by the model.

Formally, we denote the **classifier** mapping image pixel vectors to a discrete label set as $f : \mathbb{R}^m \to \{1 \cdots k\}$, the **loss function** measuring the gap between predicted label and ground-truth label as $J_\theta(x_{input}, l_{gt})$, where θ are model parameters. The **original image** from the dataset is denoted as x. An **adversarial example** x' is obtained by applying **perturbation** η on x, i.e., $x' = x + \eta$. Each element of x and x' is in the range of $[0, 1]$. Let $l = f(x)$.

2.1 L-BFGS Attack

[1] is the first work to evaluate image classification models with generated perturbations on the input images. Intuitively, if image x is correctly classified, for a small enough radium $\eta > 0$, $x + r$ satisfying $\|\eta\|_2 \leq \varepsilon$ should also be correctly classified. They argue that this kind of smoothness assumption doesn't naturally hold for deep neural networks, and they develop a method to find such adversarial examples.

For given x and target label $l' \in \{1 \cdots k\}$ satisfying $l' \neq l$, they formulate the attack as a box-constrained optimization problem:

$$\begin{aligned} \text{minimize} \quad & \|\eta\|_2 \\ \text{subject to} \quad & f(x + \eta) = l' \\ & x + \eta \in [0, 1]^m \end{aligned}$$

It can be approximated by:

$$\begin{aligned} \text{minimize} \quad & c\|\eta\|_2 + J_\theta(x+\eta, l') \\ \text{subject to} \quad & x+\eta \in [0,1]^m \end{aligned}$$

L-BFGS (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) method can be used to estimate the solution. Therefore, this attack method is named as "L-BFGS Attack". It's slow but has high success rate.

They attribute the neural networks' vulnerability to their nonlinearity. They also find that those adversarial examples generalize across model architectures and training sets. Specifically, an adversarial example will still be possibly misclassified by networks trained with different hyper-parameters (number of layers, regularization or initial weights) or on a disjoint training set.

2.2 Fast Gradient Sign Method

[2] goes deeper into the adversarial example problem. They propose a faster method to generate adversarial examples. Opposed to [1], they argue that the primary cause of neural networks' vulnerability is their linear nature. For examples, ReLUs are designed to behave in a linear way. Even non-linear functions like sigmoid are tuned to spend most of their time in the non-saturating, which is approximately linear.

Given the locally linearity, we use a simple example[1] to demonstrate their idea. Consider a binary linear classifier which uses a logistic regression with a zero bias term:

$$P(y=1 \mid x) = \sigma(w^T x) = \frac{1}{1+e^{-w^T x}}$$

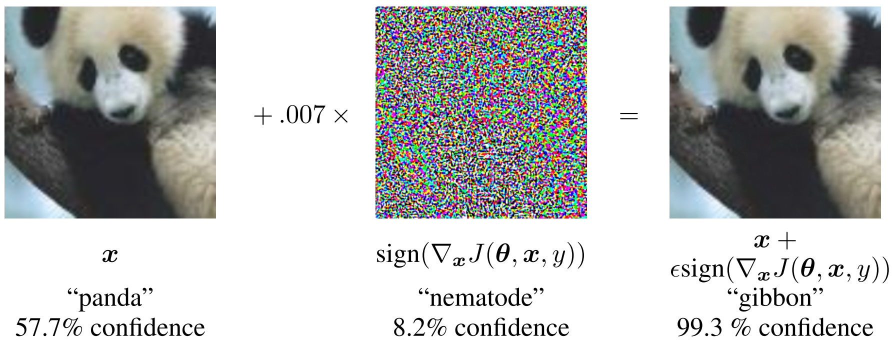
Suppose a trained model has $w = [\ \text{-1 -1 \ 1 -1 \ 1 -1 \ 1 \ 1 -1 \ 1}]^T$ and note that $w^T(x+\eta) = w^T x + w^T\eta$. Therefore, given the l -∞ norm budget (the maximum absolute change in a single pixel) of $\|\eta\|_\infty \leq \varepsilon$, we can maximize the increase on $w^T x$ by assigning $\eta = \varepsilon$ sign(w).

For a input $x = [\ \text{2 -1 \ 3 -2 \ 2 \ 2 \ 1 -4 \ 5 \ 1}]^T$, we set $\eta = 0.5$ sign$(w) = 0.5w$. By adding η to x, we can improve the class 1 probability from 5% to 88%. Here we have only 10 input dimensions while an image can usually have thousands of dimensions, which makes adversarial attacks easier and less perceptible (with much smaller ε).

Based on that, they propose "fast gradient sign method" to generate adversarial examples. The perturbation can be expressed as:

$$\eta = \varepsilon \ \text{sign}(\nabla_x J_\theta(x, l))$$

Note that this is an untargeted attack because they don't specify l'. A successful attack is showed in Figure 1.



+ .007 ×
=
$\boldsymbol{x}$
"panda"
57.7% confidence
sign$(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x}$ + ϵsign$(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure 1: An adversarial image generated by Fast Gradient Sign Method.

This method is one of the fastest and computationally cheapest to implement, but its success rate is lower than L-BFGS due to too strong assumption.

[1] This example is borrowed from slides of Stanford CS231n Lecture 9: *Understanding and Visualizing Convolutional Neural Networks*, 2016.

As for the generalization of adversarial examples, they give explanations based on neural networks' linear behavior. They also find that ensembling different models provides only limited help to defend adversarial examples.

2.3 Adversarial Examples for NLP

Above mentioned perturbation methods for images cannot be directly applied to text data as they are discrete in nature. There is also a line of work aiming at generating adversarial examples in NLP. We take [3] as an example. They propose to generate adversarial examples for evaluation on the SQuAD reading comprehension task. Below is an example of a successful attack. After appending an an adversarial distracting sentence (shown in bold), the model is fooled.

> Article: Super Bowl 50
>
> Paragraph: *Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager.* ***Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.***
>
> Question: What is the name of the quarterback who was 38 in Super Bowl XXXIII?
>
> Original Prediction: John Elway (√)
>
> Prediction under adversary: Jeff Dean (×)

They develop a 3-step procedure to construct such distracting sentences, as shown in Figure 2.

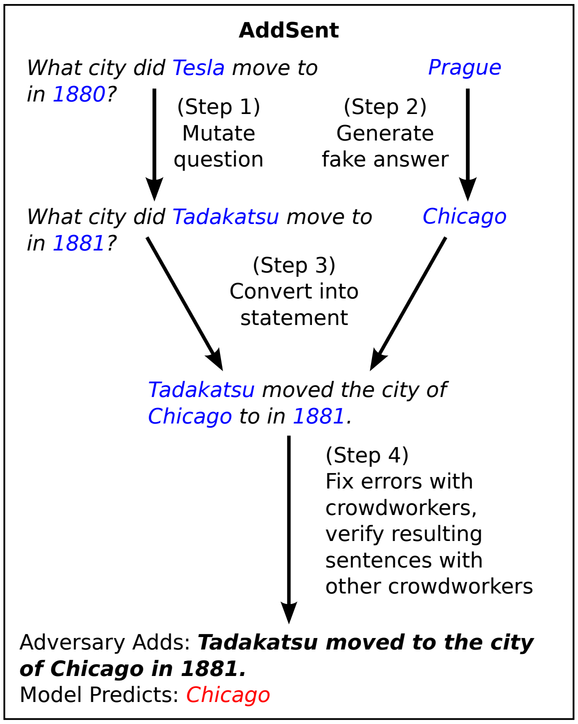

AddSent
What city did Tesla move to in 1880?
Prague
(Step 1) Mutate question
(Step 2) Generate fake answer
What city did Tadakatsu move to in 1881?
Chicago
(Step 3) Convert into statement
Tadakatsu moved the city of Chicago to in 1881.
(Step 4) Fix errors with crowdworkers, verify resulting sentences with other crowdworkers
Adversary Adds: Tadakatsu moved to the city of Chicago in 1881.
Model Predicts: Chicago

Figure 2: An illustration of how to generate a distracting sentence.

3 Adversarial Training

3.1 Basic Idea

Adversarial training means training a machine learning models on adversarial examples. This concept is first proposed in [2], as a method to help the neural networks resist adversarial perturbation.

Standard supervised training does not specify that the chosen function be resistant to adversarial examples, so they add an adversarial objective function based on the fast gradient sign method as a regularizer:

$$\widetilde{J}_\theta(x, l) = \alpha J_\theta(x, l) + (1 - \alpha) J_\theta(x + \varepsilon\ \mathrm{sign}(\nabla_x J_\theta(x, l)), l)$$

By setting $\alpha = 0.5$ to train a maxout network that is also regularized with dropout, they reduce the error rate on the original test set from 0.94% without adversarial training to 0.84% with adversarial training. On the test set of adversarial examples, the error rate falls from 89.4% to 17.9%.

It can be concluded that adversarial training can serve both as a regularization strategy and as defense against an adversary who can supply malicious inputs.

3.2 Virtual Adversarial Training

In Section 3.1, to perform adversarial training, we need the gold label l for datapoint x. In other words, the gold label determines the "adversarial direction" (to increase the loss with respect to the gold label). To further adopt adversarial training on unlabeled instances, [4] propose virtual adversarial training (VAT) as a semi-supervised learning method.

They term the regularization in adversarial training as local distributional smoothness (LDS). For unlabeled datapoint, the LDS is defined as the KL-divergence based robustness of the model predicted distribution against local perturbation.

With the hyperparameter $\varepsilon > 0$, they define:

$$\Delta_{KL}(\eta, x) = KL[p_\theta(\cdot \mid x) \mid\mid p_\theta(\cdot \mid x + \eta)]$$

$$\eta_{adv} = \underset{\eta}{argmax}\{\Delta_{KL}(\eta, x) \mid \|\eta\|_2 \le \varepsilon\}$$

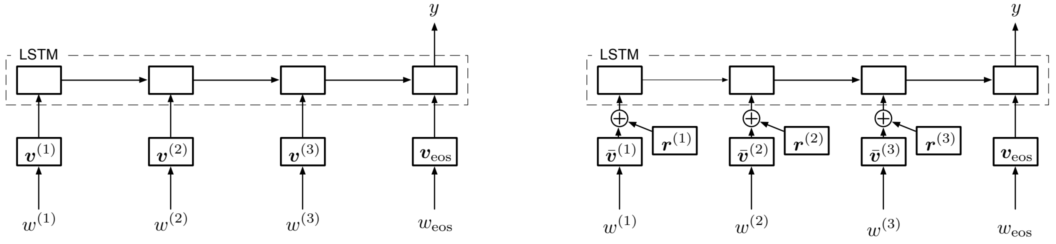
Then the LDS for datapoint x is calculated as:

$$LDS(x) = \Delta_{KL}(\eta_{adv}, x)$$

Adding the LDS of all observed datapoints to the loss function can thus provide regularization for both labeled data and unlabeled data. It outperforms nearly all semi-supervised learning methods when it's published.

3.3 Adversarial Training for NLP

Both above-mentioned methods (adversarial training and virtual adversarial training) require making small perturbations to the input vector, which is inappropriate to discrete text input. [5] is the first work to use adversarial and virtual adversarial training to improve an NLP model.



(a) LSTM-based text classification model.

(b) The model with perturbed embeddings.

Figure 3: Text classification models with clean embeddings (a) and with perturbed embeddings (b).

The main idea is to add perturbation to the input word embeddings, as illustrated in Figure 3. Different from the input of image classification, we don't have a limit (e.g. $[0, 1]$) on the range of each embedding entry. So to help ε norm constraint take effect, they normalize each embedding v_k to $\overline{v}_k$:

$$\overline{v}_k = \frac{v_k - \mathrm{E}(v)}{\sqrt{\mathrm{Var}(v)}}$$

where $\mathrm{E}(v)$ and $\mathrm{Var}(v)$ are statistics of all training examples.

After that, they directly adopt adversarial training, virtual adversarial training and their combination to the text classification model and demonstrate their effectiveness.

	'*good*'				'*bad*'			
	Baseline	**Random**	**Adversarial**	**Virtual Adversarial**	**Baseline**	**Random**	**Adversarial**	**Virtual Adversarial**
1	great	great	decent	decent	terrible	terrible	terrible	terrible
2	decent	decent	great	great	awful	awful	awful	awful
3	×bad	excellent	nice	nice	horrible	horrible	horrible	horrible
4	excellent	nice	fine	fine	×good	×good	poor	poor
5	Good	Good	entertaining	entertaining	Bad	poor	BAD	BAD
6	fine	×bad	interesting	interesting	BAD	BAD	stupid	stupid
7	nice	fine	Good	Good	poor	Bad	Bad	Bad
8	interesting	interesting	excellent	cool	stupid	stupid	laughable	laughable
9	solid	entertaining	solid	enjoyable	Horrible	Horrible	lame	lame
10	entertaining	solid	cool	excellent	horrendous	horrendous	Horrible	Horrible

Figure 4: 10 top nearest neighbors to "good" and "bad" with different training methods. "Random" means training with random perturbation with labeled examples.

Another interesting finding is shown in Figure 4. With (virtual) adversarial training, words can be better distinguished by their opposite semantic meanings instead of confused by their similar grammatical roles.

3.4 Generative Adversarial Network

Generative Adversarial Network (GAN) is another big topic, which is not going to be covered in this paper. Some people may be confused by its relation with adversarial training. As pointed out by Ian Goodfellow, GAN training can be regarded as a special case of adversarial training. While adversarial training refers to training a model on adversarial examples, GAN also involves training a classifier (discriminator) on adversarial examples (from the generator).

References

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[3] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*, 2017.

[4] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015.

[5] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

Reading Materials

[More1] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[More2] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[More3] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018.

[More4] Marco T Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

[More5] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*, 2017.

[More6] Maximin Coavoux, Shashi Narayan, and Shay B Cohen. Privacy-preserving neural representations of text. *arXiv preprint arXiv:1808.09408*, 2018.

[More7] Javid Ebrahimi, Daniel Lowd, and Dejing Dou. On adversarial examples for character-level neural machine translation. *arXiv preprint arXiv:1806.09030*, 2018.

[More8] Yi Wu, David Bamman, and Stuart Russell. Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1778–1783, 2017.

[More9] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 2019.

[More10] Wei Emma Zhang, Quan Z Sheng, A Alhazmi, and C Li. Adversarial attacks on deep learning models in natural language processing: A survey. *arXiv preprint arXiv:1901.06796*, 2019.