On the Machine Illusion

by

Zhitao Gong

A dissertation proposal submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
April 12, 2018

Keywords: Adversarial, Security, Deep Learning, Computer Vision, NLP

Copyright 2018 by Zhitao Gong

Approved by

Wei-Shinn Ku, Associate Professor of Computer Science and Software Engineering
Anh Nguyen, Assistant Professor of Computer Science and Software Engineering
Shiwen Mao, Professor of Electrical and Computer Engineering
Xiao Qin, Professor of Computer Science and Software Engineering

Abstract

The existence of adversarial samples reveals yet another quirk in neural nets. The clean samples, when perturbed with very small carefully chosen noise (e.g., change of color for a few pixels in an image, or replacement of a few words in a text piece), may be turned into adversarial ones. Despite that they are almost the same (visually or semantically) as the original one from the perspective of human beings, the adversarial samples will trick the well-trained neural nets into wrong predictions with very high confidence (probabilities). The investigation into this phenomenon has important implications both in practice and in theory. In the real word, more and more tasks are automated by neural nets, e.g., inappropriate comments and images filtering, computer virus detection, spam filtering, etc. For example, the adversarial samples might be wrongly leveraged to bypass the machine models. The replacement of a few seemingly non-important words in a sentence could turn a hateful comment into a "good" one, from the machine's point of view. This may potentially cause severe social problems if our models are not made more robust and intelligent. On the other hand, to explain this phenomenon rigorously, some of our preconceived intuitions and hypothesis about neural nets need to be revised, e.g., the generalization hypothesis, training dynamics, etc. We are still far away from a unified theory explaining how neural nets work, this study will at least provide us more insight towards our final goal.

Some pieces of my work are finished. In our first work, we propose a simple yet effective binary classifier to filter out the adversarial samples. Furthermore, we also discuss in details the limitations of our approach, which is, unfortunately, shared among many other proposed defense methods. In our second and ongoing work, we propose a model gradient-based framework to generate adversarial samples for text models. The main difficulty to generate adversarial texts with model gradient-based methods is that the input space is discrete, which makes it unclear to how to accumulate the small noise directly on the inputs. We work around this problem by searching for adversarials in the embedding space and then reconstruct the adversarial texts from the noise embeddings. Our third work is yet concretized. The high level direction will be that we first study the adversarial samples in classical machine learning models (e.g., linear models, support vector machine (SVM), nearest neighbors), for which the training dynamics and solutions are well-understood, and for which the solutions can usually be laid out in explicit forms. With intuitions and ideas gathered from these models, we then search for possible analogies in the realm of neural network.

Table of Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1
Problem Overview

Artificial intelligence (AI) helps us in many challenging tasks, e.g., recommendation systems, search, computer vision (CV), natural language processing (NLP), machine translations (MT), etc. The workhorse behind AI are the numerous machine learning models, including classical models (e.g., generalized linear models, SVM) and deep learning models (e.g., neural nets, deep reinforcement learning). Deep learning models, especially neural nets-based models, achieve state-of-the-art results in many fields. However, these models are not well understood yet.

[51] shows that the state-of-the-art image models may be tricked into wrong predictions when the test images are perturbed with carefully crafted noise. Furthermore, these perturbed images appear visually almost the same as the original ones from the perspective of human beings. These images are called *adversarial images* [51]. Many followup work show that the adversarial samples are more universal than expected. Figure 1.1 gives an example of the adversarial images. In Figure 1.1, the first row of column FGSM, FGVM, JSMA and DeepFool show adversarial images crafted from a clean image from MNIST [22], the first image in the first row. The second row visualizes the noise scale (i.e., the pixel difference between adversarial image and the original one) in heatmap. Since the pixel values are normalized to $(0, 1)$ before being fed into the classification models, the noise value range is $(-1, 1)$. The predicted label and confidence for each image is shown at the bottom of each column.

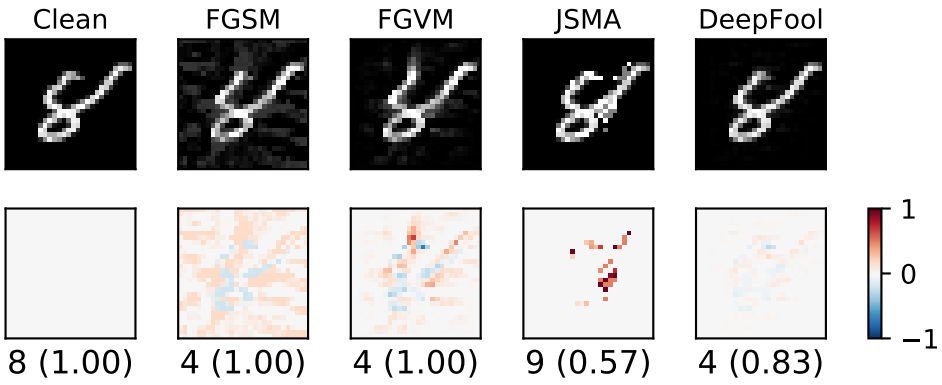

Figure 1.1: Adversarial images from MNIST dataset.

All the neural nets-based image classification models are vulnerable to adversarial samples. Another study [38] shows that even classical machine learning models are affected by the adversarial samples to some degree. Furthermore, many work on generating adversarial samples indicate that it is very cheap and easy to compute the noise. Worse still, [38] demonstrates that the adversarial samples exhibit transferability. In other words, adversarial

samples crafted for one model are likely to be adversarial for a different model, e.g., another model with different hyperparameters, or even different techniques.

Chapter 2
Road Maps

In a series of work, we plan to investigate the adversarial phenomenon in detail, both empirically and theoretically. Concretely, we have planned out three projects, each with a different focus regarding this problem.

1. **Generate adversarial texts**. Lots of work in literature focus on generating images adversarials. The difficulty of generating adversarial texts are two-folds. First, the input space is discrete, which makes it unclear how to perturb the input by iteratively accumulating small noise, as is commonly done in generating adversarial images. Second, the quality evaluation of the generated texts are intrinsically difficult. Besides human evaluation, we have yet found better ways to compare the quality of two text piece. We propose a framework to workaround the first problem. Preliminary results show that our framework can be applied to a wide range of text models. The details are discussed in Part II.

2. **Defend adversarials**. As the adversarial samples may severely degrade the performance of machine learning models, we are empirically evaluating different defensive methods to detect adversarial samples. Specifically, in one of our work, we are experimenting with binary classifier to separate adversarial samples from clean ones. The preliminary results demonstrate that it works well in practice. However, there are limitations to this binary-classifier approach. The details are in Part III.

3. **Input space topology**. A few work [52, 11, 53] have empirically studied the distribution of adversarial samples. We plan to start off from the previous work and follow this direction further. The details are still vague and need further refinement.

Part II

Generating Adversarial Samples

Chapter 3
Introduction

3.1 Motivation

Many work in literature focus on the study of adversarial images. These work have provided us new perspectives to understand the mechanism of deep neural nets, e.g., linear hypothesis [11], rethinking smoothness assumptions [51]. In addition, many algorithms have been proposed to enhance the robustness of the deep neural nets, e.g., adversarial training [11]. Only a few attempts have been made in the text domain. We think it is worthwhile to investigate adversarial samples for text models as well. In this work, we propose a simple yet effective framework to adapt the adversarial methods for images to text domain. Specifically, we first focus on model gradient-based methods since they are very fast in practice.

3.2 Problems

The major problems to generate adversarial texts are two-folds:

1. *The input space is discrete.* As a result, it is not possible to accumulate small noise computed from gradient directly in the input space. However, the algorithms rely on accumulated gradients to perturb the samples. They work well in image domain as image models usually take inputs in a continuous domain $[0, 1]$.

2. *The quality evaluation is intrinsically difficult.* This is mainly because the quality of a sentence is rather subjective. Besides, the judging criteria, if any, may vary based on situations, and even evolve over time. To illustrate the intricacies involved, let's compare an actual utterance from Master Yoda, *Much to learn, you still have*, with the usually way of speaking, *You still have much to learn*. Which is better? Are they equally good? Start Wars fans will definitely favor the Yoda-style, while both sentences successfully convey the same meaning.

The first problem is relatively easier to solve. In this work, we propose a general framework with slightly modified model gradient-based methods. We first search for adversarials in the text embedding space [31] via gradient-based methods, and then reconstruct the adversarial texts. For word-level models, (approximate) nearest search is used to align noisy embeddings to legit ones. For character-level models, we directly reverse the noisy embeddings by cosine distance. The embedding space is continuous, thus allowing us to accumulate small noise iteratively.

The second problem is open-ended. We employ Word Mover's Distance (WMD) [21] to measure the difference between an adversarial text piece with the original one. Obtaining a score from a trained language model was used in [1]. In machine translation, BLEU score [41] is used to quantify the quality of translated text piece. However, it is rather difficult (impossible, at least from my point of view) to argue which metric is better.

3.3 Outline

We briefly review recent work on generating adversarial images and texts in Chapter 4. Our adversarial text framework is detailed in Chapter 5. Experiments and preliminary results are reported in Chapter 6.

Chapter 4
Related Work

The existence of adversarial samples was first discussed in [51]. There has been an abundance of work on methods to generate adversarial images. These adversarial images raise concerns about the wide applications of deep neural nets [19]. As a result, many work have investigated the defense against these adversarial samples. However, so far as we see in literature, the attacking is much easier and cheaper than defense.

For notation, x denotes the input, y the prediction, f the target model such that $y = f(x)$, L the loss function, x^* the adversarial sample generated based on x. $\|\cdot\|_p$ denotes the p-norm. We slightly abuse the notation here, L_x denotes the loss with x as the input.

4.1 Generate Adversarial Images

The methods mainly fall into three categories, *model gradient-based*, *adversarial noise minimization* and *generative model-based*. Generally speaking, model gradient-based is much faster than the rest. However, the other two require much less knowledge about the model, thus more practical. In addition, noise minimization are usually more effective and generate more subtle noise.

4.1.1 Model Gradient-based

This class of methods compute the noise based on Jacobian or loss gradients of the target models, thus requiring a full knowledge of the target model. Fast gradient method (FGM) [11] and its variants, iterative FGM and targeted FGM [19], add to the whole image the noise that is proportional to either ∇L_x (FGVM) or $\text{sign}(\nabla L_x)$ (FGSM). Jacobian-based saliency map approach (JSMA) [40], on the contrary, perturbs one pixel at a time. It chooses the pixel with the highest saliency score, which is calculated as $-\nabla y_t \cdot \nabla y_o$ subject to $\nabla y_t > 0$, where y_t is the probability for the target class, and y_o is the sum of probabilities of all other classes. Intuitively, JSMA tries to increase the probability of the target class while decreasing others. DeepFool [34] iteratively finds the optimal direction in which we need to *travel* the minimum distance to cross the decision boundary of the target model. Although in non-linear case, the optimality is not guaranteed, in practice, however, DeepFool usually finds very subtle noise compared to other gradient methods.

4.1.2 Adversarial Noise Minimization

This class of methods are usually black-box, i.e., using the target model only as an oracle without knowledge of its parameters, gradients, etc. Generally speaking, this class minimizes $\|x^* - x\|_p$ subject to $f(x^*) \neq f(x)$ and x^* is in the input domain. This formulation is a box-constrained optimization problem, L-BFGS is used in [51] to solve it. In order to utilize

more optimization methods, [6] proposes a refinement of the above formulation. Instead of working in the original image space, it searches for noise in a transformed space by minimizing $\|s(x^*) - x\|_p - L_{s(x^*)}$, where s is a squashing function that keeps x^* within the input domain, e.g., `sigmoid` for images in the domain $[0, 1]$. Based on the reformulation, many work provide interesting insight into adversarial images. [35] shows that, instead of applying different noise to each image, it is possible to apply the same noise, i.e., a universal perturbation, to different images, such that the resulting images still trick the target model in most cases. The one-pixel change may also turn a clean image into an adversarial one [50].

4.1.3 Generative Model-based

Similar to the noise minimization, this class also formulates the problem as an optimization problem. The difference is that, instead of performing the optimization directly, this class trains a separate model to map the input to noise or adversarial samples. Adversarial transformation network (ATN) [4] trains a separate model g that minimizes $\beta\|x^* - x\|_p + \|f(x^*) - f(x)\|_{p'}$, where $g(x) = x^*$. The ATN may be used to generate adversarial noise or samples from the clean input. [59] proposes to first create a mapping between the input space and a random noise space, and then search in the noise space for potential adversarials which are verified by being mapped back to the input space. To create the mapping between input and noise space, the authors propose an autoencoder structure which consists of a) an encoder G, a generator network that maps the random noise z to the input x, $G(z) = x$, and b) a decoder I (referred to as *inverter*), another generator network that maps the input to the random noise, $I(x) = z$. Generative Adversarial Network (GAN) [12] is used for both generator networks. The whole network is trained end-to-end by minimizing the loss $\mathbb{E}_x\|G(I(z)) - x\|_p + \lambda\mathbb{E}_z\|I(G(x)) - z\|_p$.

4.2 Generate Adversarial Texts

Most work in the previous section focus on image models. As we have discussed, the main problem to generate adversarial texts are the discrete input space and the lack of quality measurement. The aforementioned model attack [59] is a viable workaround for the first problem since the noise space is smooth. However, the disadvantage with their method is that they do not have an explicit control of the quality of the generated adversarial samples. As we have seen in [59], the generated adversarial images on complex dataset usually have large visual changes. Generally, the proposal methods in literature can be classified into two categories by the space where they search for the adversarial texts. The first class of methods work in the raw input text space, while the other in a transformed space.

There are, in general, three ways to alter a sentence, *replacement*, *deletion* and *insertion*. Each has its own traits. Replacement is most straightforward and widely used since it is relatively easier to maintain the grammar and syntax correctness compared to the other two. Deletion is easier to implement since we only need to identify in some way the important features. Insertion is much more difficult mostly because we need to carefully find the word or construct a legit sentence that does not interfere with the meaning of original text piece. The sequence generation is, in itself, an active research area.

4.2.1 Text-space Methods

This class of methods follow a similar strategy.

1. Identify the features (e.g., characters, words) that have the most influence on the prediction, and then
2. follow different strategies to perturb these features according to a pool of candidates.

In essence, this class of methods are similar to JSMA [40], in which the intensity of the pixel with the highest score is increased or decreased. The Jacobian value ∇f or the loss gradient ∇L are usually employed to construct a measurement for the feature importance, e.g., ∇L is used in [24] to select important characters and phrases to perturb. The perturbation candidates usually include typos, synonyms, antonyms, frequent words in each category, and other task-dependent features. For example, typos, synonyms, and important adverbs and adjectives are used as candidates for insertion and replacement in [47]. [16] manually construct distracting yet legit sentences to overshadow the important sentences. [1] iteratively replace each word with its nearest neighbors in the embedding space until success or a threshold is reached.

Despite being intuitive, this class of methods are computationally expensive, mainly because searching in a large yet discrete space is intrinsically difficult. Some of these methods also heavily rely on manual features which does not scale in practice.

4.2.2 Transformed-space Methods

In order to employ the powerful search strategies that only work in continuous space, the other line of work follows a different strategy.

1. Map the raw texts into a continuous space,
2. search for potential adversarial in the transformed space,
3. reconstruct adversarial texts from the transformed space.

There are usually two ways to map from text space to a continuous space:

1. Word level encoding, i.e., text embedding [29].
2. Character level encoding, e.g., [17].

In [24], the authors attempt applying FGM directly on character-level CNN [58]. Although the labels of the text pieces are altered, the texts are changed to random stream of characters beyond recognition. A black-box attack based on GAN is proposed [54]. Hotflip [8] focuses on character-level model. It replaces one character at a time which maximizes the increase in loss. This is in principle the same as JSMA. The aforementioned work [59] employs autoencoder structure, in which the encoder maps the input texts to a Gaussian noise space, while the decoder maps the noise back to text space to reconstruct the potential adversarial texts.

Chapter 5
Adversarial Text Framework

Our method is more general than aforementioned methods, beside, the computation is really fast. In this section, we present our framework that generates adversarial texts by noise generated computed from model gradients.

5.1 Discrete Input Space

In order to work in a continuous space, our framework first searches for adversarial texts in the text or character embedding space, then reconstructs the adversarial sentences with nearest neighbor search. Searching for adversarials in the embedding space is similar in principle to searching for adversarial images. However, the generated noisy embedding vectors usually do not correspond to any tokens in the text space. To construct the adversarial texts, we align each embedding to its nearest one. We can use (approximate) nearest neighbor search if the vocabulary size is large, or direct embedding reverse by cosine distance if the embedding matrix is relative small. This reconstructing process can be seen as a strong *denoising* process. With appropriate noise scale, we would expect most of the tokens/characters remain unchanged, with only few replaced. This framework builds upon the following observations.

1. In the gradient-based methods, the input features (e.g., pixels, tokens, characters) that are relatively more important for the final predictions will receive more noise, while others relatively less noise. The is actually the core property of the gradient-based methods.

2. The embedded word vectors preserve the subtle semantic relationships among words [31, 30]. For example, `vec("clothing")` is closer to `vec("shirt")` as `vec("dish")` to `vec("bowl")`, while `vec("clothing")` is far away from `vec("dish")` in terms of p-norm, since they are not semantically related [29]. This property assures that it is more likely to replace the victim words with a semantically related one rather than a random one.

5.2 Word Mover's Distance (WMD)

The second problem we need to resolve is the choice of quality metric for generated adversarial texts, so that we have a scalable way to measure the effectiveness of our framework. We employ the Word Mover's Distance (WMD) [21] as the metric. WMD measures the dissimilarity between two text documents as the minimum amount of distance that the embedded words of one document need to *travel* to reach the embedded words of another document. WMD can be considered as a special case of Earth Mover's Distance (EMD) [46]. Intuitively, it quantifies the semantic similarity between two text bodies. In this work, WMD

is closely related to the ratio of number of words changed to the sentence length. However, we plan to extend our framework with paraphrasing and insertion/deletion, where the sentence length may change. In that case, WMD is more flexible and accurate.

Chapter 6
Experiment

We evaluate our framework on three text classification problems. Section 6.1 details on the data preprocessing. The adversarial algorithms we use are (FGM) [11] and DeepFool [34]. We tried JSMA, however, due to the mechanism of JSMA, it is not directly applicable in our framework. We report in Section 6.4 the original model accuracy, accuracy on adversarial embeddings, and accuracy on reconstructed adversarial texts in our experiment. Only a few examples of generated adversarial texts are shown in this paper due to the space constraint. The complete sets of adversarial texts under different parameter settings and the code to reproduce the experiment are available online[1].

Computation-wise, the bottleneck in our framework is the nearest neighbor search. Word vector spaces, such as GloVe [43], usually have millions or billions of tokens embedded in very high dimensions. The vanilla nearest neighbor search is almost impractical. Instead, we employ the an approximate nearest neighbor (ANN) technique in our experiment. The ANN implementation which we use in our experiment is Approximate Nearest Neighbors Oh Yeah (`annoy`)[2], which is well integrated into `gensim` [44] package.

6.1 Dataset

We use three text datasets in our experiments. The datasets are summarized in Table 6.1. The last column shows our target model accuracy on clean test data.

Table 6.1: Dataset Summary

Dataset	Labels	Training	Testing	Max Length	Accuracy
IMDB	2	25000	25000	300	0.8787
Reuters-2	2	3300	1438	100	0.9854
Reuters-5	5	1735	585	100	0.8701

6.1.1 IMDB Movie Reviews

This is a dataset for binary sentiment classification [26]. It contains a set of 25,000 highly polar (positive or negative) movie reviews for training, and 25,000 for testing. No special preprocessing is used for this dataset except that we truncate/pad all the sentences to a fixed maximum length, 400. This max length is chosen empirically.

[1] https://github.com/gongzhitaao/adversarial-text
[2] https://github.com/spotify/annoy

6.1.2 Reuters

This is a dataset of 11,228 newswires from Reuters, labeled over 90 topics. We load this dataset through the NLTK [5] package. The raw Reuters dataset is highly unbalanced. Some categories contain over a thousand samples, while others may contain only a few. The problem with such highly unbalanced data is that the texts that belong to under-populated categories are almost always get classified incorrectly. Even though our model may still achieve high accuracy with 90 labels, it would be meaningless to include these under-populated categories in the experiment since we are mainly interested in perturbation of those samples that are already being classified correctly. Keras[3] uses 46 categories out of 90. However, the 46 categories are still highly unbalanced. In our experiment, we preprocess Reuters and extract two datasets from it.

1. Reuters-2

   It contains two most populous categories, i.e., `acq` and `earn`. The `acq` category contains 1650 training samples and 719 test samples. Over 71% sentences in the `acq` category have less than 160 tokens. The `earn` category contains 2877 training samples and 1087 test samples. Over 83% sentences in `earn` category have less then 160 tokens. In order to balance the two categories, for `earn`, we use 1650 samples out of 2877 for training, and 719 for testing. The maximum sentence length of this binary classification dataset is set to 160.

2. Reuters-5

   It contains five categories, i.e., `crude`, `grain`, `interest`, `money-fx` and `trade`. Similar to Reuters-2, we balance the five categories by using 347 examples (the size of `interest` categories) for each category during training, and 117 each for testing. The maximum sentence length is set to 350.

6.2 Embedding

Our framework relies heavily on the *size* and *quality* of the embedding space. More semantic alternatives would be helpful to improve the quality of generated adversarial texts. As a result, we use the GloVe [43] pre-trained embedding in our experiment. Specifically, we use the largest GloVe embedding, `glove.840B.300d`, which embeds 840 billion tokens (approximately 2.2 million cased vocabularies) into a vector space of 300 dimensions. The value range of the word vectors are roughly $(-5.161, 5.0408)$.

6.3 Model

In this work, we focus on feedforward architectures. Specifically, we use CNN model for the classification tasks. The model structure is summarized in Figure 6.1.

Where B denotes batch size, L the maximum sentence length, D the word vector space dimension. In our experiment, we have $B = 128$, and $D = 300$ since we are using the pre-trained embedding `glove.840B.300d`.

[3] https://keras.io/

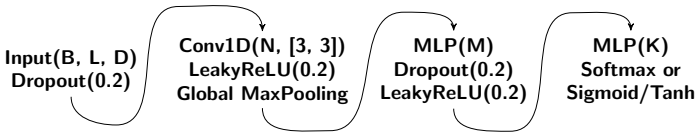

Figure 6.1: CNN model for text classification.

Note that for models trained for binary classification tasks, DeepFool assumes the output in the range $[-1, 1]$, instead of $[0, 1]$. Thus we have two slightly different models for each of the binary classification task (IMDB and Reuters-2), one with `sigmoid` output, and the other with `tanh`. The model with `tahn` output is trained with Adam [18] by minimizing the mean squared error (MSE), while all the other models are trained with Adam by minimizing the cross-entropy loss. Despite the small difference in architecture, `sigmoid`- and `tanh`-models on the same task have almost identical accuracy. As a result, in Table 6.1, we report only one result for IMDB and Reuters-2.

All our models have $N = 256$ and $M = 512$, except for the one with `tanh` output on the IMDB classification task, in which we have $N = 128$ and $M = 256$. The reason that we change to a smaller model is that the larger one always gets stuck during the training. We are not yet clear what causes this problem and why a smaller model helps.

6.4 Preliminary Results

Table 6.2: Model accuracy under different parameter settings.

Method	Dataset	acc_1/acc_2			
		ϵ	0.40	0.35	0.30	0.25
FGSM	IMDB		0.1213 / 0.1334	0.1213 / 0.1990	0.1213 / 0.4074	0.1213 / 0.6770
	Reuters-2		0.0146 / 0.6495	0.0146 / 0.7928	0.0146 / 0.9110	0.0146 / 0.9680
	Reuters-5		0.1128 / 0.5880	0.1128 / 0.7162	0.1128 / 0.7949	0.1128 / 0.8462
		ϵ	15	30	50	100
FGVM	IMDB		0.6888 / 0.8538	0.6549 / 0.8354	0.6277 / 0.8207	0.5925 / 0.7964
	Reuters-2		0.7747 / 0.7990	0.7337 / 0.7538	0.6975 / 0.7156	0.6349 / 0.6523
	Reuters-5		0.5915 / 0.7983	0.5368 / 0.6872	0.4786 / 0.6085	0.4000 / 0.5111
		ϵ	20	30	40	50
DeepFool	IMDB		0.5569 / 0.8298	0.5508 / 0.7225	0.5472 / 0.6678	0.5453 / 0.6416
	Reuters-2		0.4416 / 0.6766	0.4416 / 0.5236	0.4416 / 0.4910	0.4416 / 0.4715
	Reuters-5		0.1163 / 0.4034	0.1162 / 0.2222	0.1162 / 0.1641	0.1162 / 0.1402

The model accuracy on adversarial embeddings before and after the nearest neighbor search under different parameter settings are summarized in Table 6.2. ϵ is the noise scaling factor. We report two accuracy measurements per parameter setting in the format acc_1/acc_2, where acc_1 is the model accuracy on adversarial embeddings before nearest neighbor search,

Clean Text	Label	WMD (n/L)	Adversarial Text
I was n't really interested in seeing Step Up , but my friend just kept bugging and bugging **[. . .]** where are you when we need you ? ! < br / > < br / > 4/10 (IMDB)	1→0	0.0089 (0.0027)	I was n't really interested in seeing Step Up , but my friend just kept bugging and bugging **[. . .]** where are you when we need you ? ! < br / > < br / > 7/10
SAN MIGUEL DEAL HIT BY MORE LAWSUITS A bid by San Miguel Corp (SMC) & lt ; SANM . MN > to buy back 38 . 1 mln sequestered shares from United Coconut Planters Bank (UCPB) has been hit by two new lawsuits , sources in the Philippine food and brewery company said . A Manila court yesterday issued an injunction barring UCPB from selling the shares , which represent 31 pct of SMC ' s outstanding capital stock of 121 mln shares , until hearings on April 21 on a petition filed by Eduardo Cojuangco , a former chairman of both SMC and UCPB . Cojuangco said the Coconut Industry Investment Fund (CIIF) and 1 . 4 mln farmers were the rightful owners of the shares . Cojuangco said the shares were held in trust by UCPB and represented a blue chip investment . His petition said UCPB ' s plans to sell (REUTERS-2)	0→1	0.5851 (0.0750)	SAN MIGUEL DEAL YEAR RESULT MORE LOSSES A bid by San Miguel Corp (SMC) & lt ; SANM . MN > to get back 38 . 1 mln sequestered earnings from United Coconut Compost Bank (UCPB) has been hit by two new lawsuits , sources in the Philippine food and brewery company said . A Manila court yesterday issued an injunction barring UME from selling the shares , which represent 31 pct of SMC ' s outstanding capital earnings of 121 mln shares , until hearings on April 21 on a petition filed by Eduardo Cojuangco , a former chairman of both SMC and Bolivariano . Cojuangco said the Coconut Industry Investment Fund [Odyss) and 1 . 4 mln farmers were the rightful owners of the shares . Cojuangco said the shares were held in trust by Hyvinkaa and represented a blue chip investment . His petition said UCPB ' s plans to sell
CHINA DAILY SAYS VERMIN EAT 7 - 12 PCT GRAIN STOCKS A survey of 19 provinces and seven cities showed vermin consume between seven and 12 pct of China ' s grain stocks , the China Daily said **[. . .]** (REUTERS-5)	1→3	0.1249 (0.0153)	CHINA DAILY SAYS VERMIN EAT 7 - 12 PCT CARRYING STOCKS A survey of 19 provinces and seven cities showed vermin consume between seven and 12 pct of China ' s gas stocks , the China Daily said **[. . .]**

Figure 6.2: Adversarial texts generated via FGVM.

acc_2 the accuracy on adversarial embeddings that are reconstructed by nearest neighbor search. In other words, acc_2 is the model accuracy on generated adversarial texts.

In the adversarial text examples, to aid reading, we omit the parts that are not changed, denoted by **[. . .]** in the texts. The "(IMDB)" at the end of each clean text piece denotes the dataset that this piece of text belongs to. In addition to Word Mover's Distance (WMD), we also report the change rate, $\frac{n}{L}$, where n is the number of changed words, L the sentence length. The corresponding changed words are highlighted in the figures.

6.4.1 Fast Gradient Method

We first evaluate two versions of FGM, i.e., FGSM and FGVM. Their example results are shown in Figure 6.3 and Figure 6.2, respectively. For FGVM, it was proposed in [32] to use $\frac{\nabla L}{\|\nabla L\|_2}$ to FGVM usually needs much larger noise scaling factor since most gradients are close to zero.

6.4.2 DeepFool

Adversarial examples are shown in Figure 6.4. We experiment with different overshoot values (also denoted as ϵ in the table). Usually, for images, we tend to use very small overshoot values, e.g., 1.02, which creates just enough noise to cross the decision boundary. However, in our framework, the reconstructing process is a very strong denoising process, where much of the subtle noise will be smoothed. To compensate for this, we experiment with very large overshoot values. In practice, this works very well. As we can see, labels are altered by replacing just one word in many cases.

Clean Text	Label	WMD (n/L)	Adversarial Text
One of those TV films you saw in the seventies that scared the hell out of you when you were a kid but still gives you an eerie feeling . No great actors or expensive production but everytime that phone rings ... ... (IMDB)	0→1	0.3979 (0.0930)	One of those TV films you saw in the seventies that scared the hell out of you when you were a kid but not gives you considered unnerving feeling . No great actors and/or expensive production but everytime that phone rings ... ...
CIS & It ; CISIF . O > AGREES TO SECOND EXTENSION CIS Technologies Inc said that it and the Swiss Reinsurance Co of Zurich , Switzerland agreed to a second extension of two dates for the final part of their share purchase agreement . It said the June one election date has been extended to June 15 and the June 30 closing date has been changed to July 31 . (Reuters-2)	0→1	0.9414 (0.1972)	CIS & It – Un-Idle _ O > RESULT TO SECOND EXTENSION CIS Technologies Inc saying that it and the Swiss Reinsurance Co of Zurich run Switzerland stated to single second extension of two dates cost the final part the their share purchase agreement . When said the July one election date has been extended to June 15 both the June 30 closing date has been changed to July 29 .
FED EXPECTED TO ADD RESERVES The Federal Reserve will enter the government securities market to supply reserves via either a large round of customer repurchase agreements or by overnight or possibly four - day system repurchases , economists said . **[. . .]** tax payments swell Treasury balances at the Fed . Fed funds hovered at 6 - 3 / 4 pct after averaging 6 . 80 pct yesterday . (Reuters-5)	2→3	0.6628 (0.1146)	FED EXPECTED TO ADD RESERVES The Federal Reserve 'll enter the government securities market able supply reserves via either a large round of customer repurchase agreements or directed overnight or possibly four - day system repurchases books economists said . **[. . .]** tax payments surf Treasury balances at the futures well Fed funds hovered at 6 - 3 Category 4 Q2 after averaged 6 ..... 80 pct yesterday .

Figure 6.3: Adversarial texts generated via FGSM.

Clean Text	Label	WMD (n/L)	Adversarial Text
Quick summary of the book : **[. . .]** The book was n't bad , but was soooooooo cliché < br / > < br / > Now about the movie **[. . .]** (IMDB)	0→1	0.0317 (0.0050)	Quick summary of the book : **[. . .]** The book was n't bad , but was soooooooo TahitiNut < br / > < br / > Now about the movie **[. . .]**
zulchzulu < SM > TO OFFER SPECIAL DIVIDEND Southmark Corp said it will issue its shareholders a special dividend right **[. . .]** (Reuters-2)	1→0	0.0817 (0.0125)	zulchzulu < SM > TO OFFER OFFERS SHARES Southmark Corp said it will issue its shareh olders a special dividend right **[. . .]**
U . K . MONEY MARKET GIVEN FURTHER 68 MLN STG HELP The Bank of England said it provided the market with a further **[. . .]** (Reuters-5)	3→2	0.0556 (0.0077)	U . K . MONEY MARKET GIVEN FURTHER 68 ARL STG HELP The Bank of England said it provided the market with a further **[. . .]**

Figure 6.4: Adversarial texts generated via DeepFool.

Chapter 7
Next Step

1. Test our framework against seq2seq models, possibly in the domain of machine translation. The hallucination in neural machine translation (NMT) is a related question. When fed with some meaningless and repeated tokens, NMT models may output some legit translations. As far as I know, searching for such triggering tokens are mainly by trial-and-error. It is possible that we could generate such triggering tokens in our framework.

2. Incorporate other classes of image adversarial methods into our framework.

3. Provide more strong arguments for our choice of WMD as the evaluation metric.

Part III

Defending Adversarial Samples

Chapter 8
Introduction

8.1 Motivation

Deep neural networks have been successfully adopted to many life critical areas, e.g., skin cancer detection [9], auto-driving [48], traffic sign classification [7], etc. A recent study [51], however, discovered that deep neural networks are susceptible to adversarial images. Figure 8.1 shows an example of adversarial images generated via fast gradient sign method [19, 20] on MNIST. The adversarial images (second row) are generated from the first row via iterative FGSM. The label of each image is shown below with prediction probability in parenthesis. Our model achieves less then 1% error rate on the clean data. As we can see that although the adversarial and original clean images are almost identical from the perspective of human beings, the deep neural network will produce wrong predictions with very high confidence. Similar techniques can easily fool the image system into mistaking a stop sign for a yield sign, a dog for a automobile, for example. When leveraged by malicious users, these adversarial images pose a great threat to the deep neural network systems.

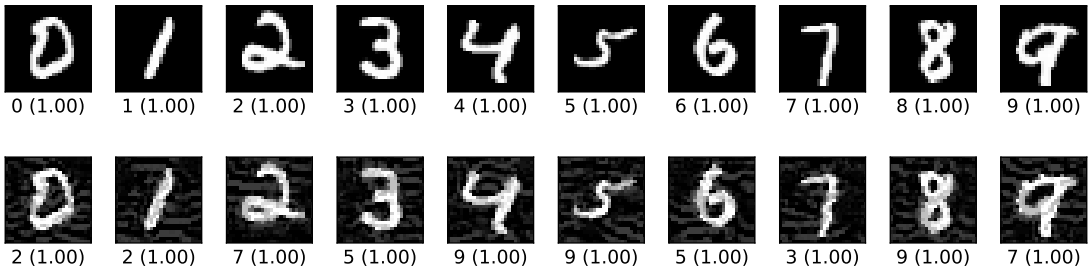

Figure 8.1: Adversarial samples.

8.2 Proposed Defense

Although adversarial and clean images appear visually indiscernible, their subtle differences can successfully fool the deep neural networks. This means that deep neural networks are very sensitive to these subtle differences. So an intuitively question to ask is: can we leverage the neural nets' sensitivity to subtle differences to distinguish between adversarial and clean images? Our preliminary results suggest that the answer is positive. In this work we demonstrate that a simple binary classifier can separate the adversarial from the original clean images with very high accuracy (over 99%). However, we also show that the binary classifier suffers from the generalization limitation, i.e., it is sensitive 1) to hyperparameters in

adversarial methods, and 2) to different ways the adversarials are crafted. In addition to that, we also discover that this limitation is also shared among other proposed defense methods, e.g., defensive training [15, 20], knowledge distillation [39], etc. We empirically investigate the limitation and propose the hypothesis that the adversarial and original dataset are, in effect, two completely *different* datasets, despite being visually similar.

This part is organized as follows. In Chapter 9, we give an overview of the current research in adversarial defense. The adversarial generating algorithms are already surveyed in 4. Chapter 10 presents our preliminary results and discussions.

Chapter 9
Related Work

Generally speaking, generating adversarial samples are much cheaper and easier than defending them. Despite numerous work on defense in literature, little progress has been made toward understanding and defending adversarial samples. There are two orthogonal lines of work towards the defense of adversarial samples. The first direction focuses on the improvement of the model, while the other focuses on preprocessing the inputs, e.g, detecting or recovering the adversarial samples.

9.1 Enhance Models

The main idea to improve models' robustness to adversarial samples is to include adversarial samples during training process [23, 20, 15, 33]. However, as discussed in Section 10.2, this class of methods suffer from, what we call, the *generalization limitations*. Our experiment suggests this seems to be an intrinsic property of adversarial datasets.

[57] suggests that the activation bounded ReLU may provide some robustness.

Knowledge distillation [14] in itself is also shown to be an viable method in some cases [39]. The restrictions of knowledge distillation are 1) that it only applies to models that produce categorical probabilities, and 2) that it needs model re-training.

9.2 Preprocess Inputs

WGAN [3] is used in [2] to denoise the adversarial images. [27] uses GAN (called reformer in the paper) to move adversarial samples closer to the clean ones. [13, 19, 55] performs various image transformations (e.g., applying Gaussian noise/Gaussian filters, JPEG compression [19], resizing [55], image quilting [13]) to all incoming images, be it clean or adversarial. However in our experiment, the performance of the transformation-based defense varies a lot across different datasets.

Adversarial image/noise are detected based on images statistics, e.g, entropy [25, 37], psychometric perceptual adversarial similarity score (PASS) [45]. [49] detects the adversarial samples by hypothesis testing. [56] proposes a defense method for text models based on the score computed from a language model.

Around the same time of our study, another work [28] also proposes the binary classifier as a defense. The difference is that they are too optimistic about the effective of this approach.

Chapter 10
Experiment

Generally, we follow the steps below to test the effectiveness and limitation of the binary classifier approach.

1. Train a deep neural network f_1 on the original clean training data X_{train}, and craft adversarial dataset from the original clean data, $X_{train} \rightarrow X_{train}^{adv(f_1)}$, $X_{test} \rightarrow X_{test}^{adv(f_1)}$. f_1 is used to generate the attacking adversarial dataset which we want to filter out.

2. Train a binary classifier f_2 on the combined (shuffled) training data $\{X_{train}, X_{train}^{adv(f_1)}\}$, where X_{train} is labeled 0 and $X_{train}^{adv(f_1)}$ labeled 1.

3. Test the accuracy of f_2 on X_{test} and $X_{test}^{adv(f_1)}$, respectively.

4. Construct second-round adversarial test data, $\{X_{test}, X_{test}^{adv(f_1)}\} \rightarrow \{X_{test}, X_{test}^{adv(f_1)}\}^{adv(f_2)}$ and test f_2 accuracy on this new adversarial dataset. Concretely, we want to test whether we could find adversarial samples 1) that can successfully bypass the binary classifier f_2, and 2) that can still fool the target model f_1 if they bypass the binary classifier. Since adversarial datasets are shown to be transferable among different machine learning techniques [38], the binary classifier approach will be seriously flawed if f_2 failed this second-round attacking test.

The code to reproduce our experiment are available https://github.com/gongzhitaao/adversarial-classifier.

10.1 Efficiency and Robustness of the Classifier

Table 10.1: Accuracy on adversarial samples generated with FGSM/TGSM.

	f_1		f_2			
Dataset	X_{test}	$X_{test}^{adv(f_1)}$	X_{test}	$X_{test}^{adv(f_1)}$	$\{X_{test}\}^{adv(f_2)}$	$\{X_{test}^{adv(f_1)}\}^{adv(f_2)}$
MNIST	0.9914	0.0213	1.00	1.00	0.00	1.00
CIFAR10	0.8279	0.1500	0.99	1.00	0.01	1.00
SVHN	0.9378	0.2453	1.00	1.00	0.00	1.00

We evaluate the binary classifier approach on MNIST, CIFAR10, and SVHN datasets. Of all the datasets, the binary classifier achieved accuracy over 99% and was shown to be robust to a second-round adversarial attack. The results are summarized in Table 10.1. Each column denotes the model accuracy on the corresponding dataset. The direct conclusions from Table 10.1 are summarized as follows.

1. Accuracy on X_{test} and $X_{test}^{adv(f_1)}$ suggests that the binary classifier is very effective at separating adversarial from clean dataset. Actually during our experiment, the accuracy on X_{test} is always near 1, while the accuracy on $X_{test}^{adv(f_1)}$ is either near 1 (successful) or near 0 (unsuccessful). Which means that the classifier either successfully detects the subtle difference completely or fails completely. We did not observe any values in between.

2. Accuracy on $\{X_{test}^{adv(f_1)}\}^{adv(f_2)}$ suggests that we were not successful in disguising adversarial samples to bypass the the classifier. In other words, the binary classifier approach is robust to a second-round adversarial attack.

3. Accuracy on $\{X_{test}\}^{adv(f_2)}$ suggests that in case of the second-round attack, the binary classifier has very high false negative. In other words, it tends to recognize them all as adversarials. This, does not pose a problem in our opinion. Since our main focus is to block adversarial samples.

10.2 Generalization Limitation

Before we conclude too optimistic about the binary classifier approach performance, however, we discover that it suffers from the *generalization limitation*.

1. When trained to recognize adversarial dataset generated via FGSM/TGSM, the binary classifier is sensitive to the hyper-parameter ϵ.

2. The binary classifier is also sensitive to the adversarial crafting algorithm.

In out experiment, the aforementioned limitations also apply to adversarial training [20, 15] and defensive distillation [39].

10.2.1 Sensitivity to ϵ

Table 10.2 summarizes our tests on CIFAR10. For brevity, we use $f_2\big|_{\epsilon=\epsilon_0}$ to denote that the classifier f_2 is trained on adversarial data generated on f_1 with $\epsilon = \epsilon_0$. The binary classifier is trained on mixed clean data and adversarial dataset which is generated via FGSM with $\epsilon = 0.03$. Then we re-generate adversarial dataset via FGSM/TGSM with different ϵ values.

Table 10.2: ϵ sensitivity on CIFAR10

	$f_2\big\|_{\epsilon=0.03}$	
ϵ	X_{test}	$X_{test}^{adv(f_1)}$
0.3	0.9996	1.0000
0.1	0.9996	1.0000
0.03	0.9996	0.9997
0.01	0.9996	**0.0030**

As shown in Table 10.2, $f_2\big|_{\epsilon=\epsilon_0}$ can correctly filter out adversarial dataset generated with $\epsilon \geq \epsilon_0$, but fails when adversarial data are generated with $\epsilon < \epsilon_1$. Results on MNIST and SVHN are similar. This phenomenon was also observed in defensive training [20]. To overcome this issue, they proposed to use mixed ϵ values to generate the adversarial datasets. However, Table 10.2 suggests that adversarial datasets generated with smaller ϵ are *superset* of those generated with larger ϵ. This hypothesis could be well explained by the linearity hypothesis [19, 53]. The same conclusion also applies to adversarial training. In our experiment, the results of defensive training are similar to the binary classifier approach.

10.2.2 Disparity among Adversarial Samples

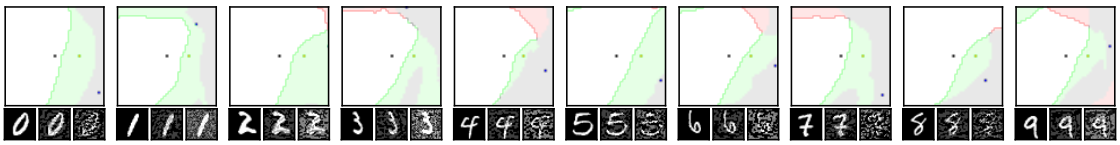

Figure 10.1: Adversarial training [15, 20] does not work properly.

Figure 10.1 is a church window plot [53]. Each pixel (i, j) (row index and column index pair) represents a data point $\tilde{x}$ in the input space and $\tilde{x} = x + \mathbf{h}\epsilon_j + \mathbf{v}\epsilon_i$, where $\mathbf{h}$ is the direction computed by FGSM and $\mathbf{v}$ is a random direction orthogonal to $\mathbf{h}$. The ϵ ranges from $[-0.5, 0.5]$ and $\epsilon_{(\cdot)}$ is the interpolated value in between. The central black dot • represents the original data point x, the orange dot (on the right of the center dot) • represents the last adversarial sample created from x via FGSM that is used in the adversarial training and the blue dot • represents a random adversarial sample created from x that cannot be recognized with adversarial training. The three digits below each image, from left to right, are the data samples that correspond to the black dot, orange dot and blue dot, respectively. □ ( ) represents the data samples that are always correctly (incorrectly) recognized by the model. represents the adversarial samples that can be correctly recognized without adversarial training only. And represents the data points that were correctly recognized with adversarial training only, i.e., the side effect of adversarial training.

In our experiment, we also discovered that the binary classifier is also sensitive to the algorithms used to generate the adversarial datasets.

Specifically, the binary classifier trained on FGSM adversarial dataset achieves good accuracy (over 99%) on FGSM adversarial dataset, but not on adversarial generated via JSMA, and vise versa. However, when binary classifier is trained on a mixed adversarial dataset from FGSM and JSMA, it performs well (with accuracy over 99%) on both datasets. This suggests that FGSM and JSMA generate adversarial datasets that are *far away* from each other. It is too vague without defining precisely what is *being far away*. In our opinion, they are *far away* in the same way that CIFAR10 is *far away* from SVHN. A well-trained model on CIFAR10 will perform poorly on SVHN, and vise versa. However, a well-trained model on the the mixed dataset of CIFAR10 and SVHN will perform just as well, if not better, on both datasets, as if it is trained solely on one dataset.

The adversarial datasets generated via FGSM and TGSM are, however, *compatible* with each other. In other words, the classifier trained on one adversarial datasets performs well on

adversarials from the other algorithm. They are compatible in the same way that training set and test set are compatible. Usually we expect a model, when properly trained, should generalize well to the unseen data from the same distribution, e.g., the test dataset.

In effect, it is not just FGSM and JSMA are incompatible. We can generate adversarial data samples by a linear combination of the direction computed by FGSM and another random orthogonal direction, as illustrated in a church plot [53] Figure 10.1. Figure 10.1 visually shows the effect of adversarial training [20]. Each image represents adversarial samples generated from *one* data sample, which is represented as a black dot in the center of each image, the last adversarial sample used in adversarial training is represented as an orange dot (on the right of black dot, i.e., in the direction computed by FGSM). The green area represents the adversarial samples that cannot be correctly recognized without adversarial training but can be correctly recognized with adversarial training. The red area represents data samples that can be correctly recognized without adversarial training but cannot be correctly recognized with adversarial training. In other words, it represents the side effect of adversarial training, i.e., slightly reducing the model accuracy. The white (gray) area represents the data samples that are always correctly (incorrectly) recognized with or without adversarial training.

As we can see from Figure 10.1, adversarial training does make the model more robust against the adversarial sample (and adversarial samples around it to some extent) used for training (green area). However, it does not rule out all adversarials. There are still adversarial samples (gray area) that are not affected by the adversarial training. Further more, we could observe that the green area largely distributes along the horizontal direction, i.e., the FGSM direction. In [36], they observed similar results for fooling images. In their experiment, adversarial training with fooling images, deep neural network models are more robust against a limited set of fooling images. However they can still be fooled by other fooling images easily.

Chapter 11
Next Step

1. Make a thorough study of all the defense methods.
2. Despite of the ever proliferating papers on defense, few make any concrete contribution to the understanding of adversarial samples. We plan to investigate this phenomenon from a theoretical point of view, following some of recent work [42, 10].
3. Study the defense for adversarial texts.

Bibliography

[1] Anonymous. "Adversarial Examples for Natural Language Classification Problems". In: *International Conference on Learning Representations* (2018). URL: https://openreview.net/forum?id=r1QZ3zbAZ.

[2] Anonymous. "Defense-Gan: Protecting Classifiers Against Adversarial Attacks Using Generative Models". In: *International Conference on Learning Representations* (2018). URL: https://openreview.net/forum?id=BkJ3ibb0-.

[3] M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein GAN". In: *ArXiv e-prints* (Jan. 2017). arXiv: 1701.07875 [stat.ML].

[4] Shumeet Baluja and Ian Fischer. "Adversarial Transformation Networks: Learning To Generate Adversarial Examples". In: *CoRR* abs/1703.09387 (2017). URL: http://arxiv.org/abs/1703.09387.

[5] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[6] Nicholas Carlini and David Wagner. "Towards Evaluating the Robustness of Neural Networks". In: *CoRR* abs/1608.04644 (2016). URL: http://arxiv.org/abs/1608.04644.

[7] Dan CireşAn et al. "Multi-Column Deep Neural Network for Traffic Sign Classification". In: *Neural Networks* 32 (2012), pp. 333–338.

[8] J. Ebrahimi et al. "HotFlip: White-Box Adversarial Examples for NLP". In: *ArXiv e-prints* (Dec. 2017). arXiv: 1712.06751 [cs.CL].

[9] Andre Esteva et al. "Dermatologist-Level Classification of Skin Cancer With Deep Neural Networks". In: *Nature* advance online publication (Jan. 2017). Letter. ISSN: 1476-4687. URL: http://dx.doi.org/10.1038/nature21056.

[10] J. Gilmer et al. "Adversarial Spheres". In: *ArXiv e-prints* (Jan. 2018). arXiv: 1801.02774 [cs.CV].

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and Harnessing Adversarial Examples". In: *ArXiv e-prints* (Dec. 2014). arXiv: 1412.6572 [stat.ML].

[12] I. J. Goodfellow et al. "Generative Adversarial Networks". In: *ArXiv e-prints* (June 2014). arXiv: 1406.2661 [stat.ML].

[13] C. Guo et al. "Countering Adversarial Images Using Input Transformations". In: *ArXiv e-prints* (Oct. 2017). arXiv: 1711.00117 [cs.CV].

[14] G. Hinton, O. Vinyals, and J. Dean. "Distilling the Knowledge in a Neural Network". In: *ArXiv e-prints* (Mar. 2015). arXiv: 1503.02531 [stat.ML].

[15] Ruitong Huang et al. "Learning With a Strong Adversary". In: *CoRR* abs/1511.03034 (2015). URL: http://arxiv.org/abs/1511.03034.

[16] Robin Jia and Percy Liang. "Adversarial Examples for Evaluating Reading Comprehension Systems". In: *arXiv preprint arXiv:1707.07328* (2017).

[17] Yoon Kim et al. "Character-Aware Neural Language Models". In: *arXiv preprint arXiv:1508.06615* (2015).

[18] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014). URL: http://arxiv.org/abs/1412.6980.

[19] A. Kurakin, I. Goodfellow, and S. Bengio. "Adversarial Examples in the Physical world". In: *ArXiv e-prints* (July 2016). arXiv: 1607.02533 [cs.CV].

[20] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. "Adversarial Machine Learning At Scale". In: *CoRR* abs/1611.01236 (2016). URL: http://arxiv.org/abs/1611.01236.

[21] Matt Kusner et al. "From word embeddings to document distances". In: *International Conference on Machine Learning*. 2015, pp. 957–966.

[22] Yann LeCun and Corinna Cortes. *MNIST handwritten digit database*. http://yann.lecun.com/exdb/mn 2010. URL: http://yann.lecun.com/exdb/mnist/.

[23] Hyeungill Lee, Sungyeob Han, and Jungwoo Lee. "Generative Adversarial Trainer: Defense To Adversarial Perturbations With GAN". In: *CoRR* abs/1705.03387 (2017). arXiv: 1705.03387. URL: http://arxiv.org/abs/1705.03387.

[24] Bin Liang et al. "Deep Text Classification Can Be Fooled". In: *arXiv preprint arXiv:1704.08006* (2017).

[25] Bin Liang et al. "Detecting Adversarial Examples in Deep Networks With Adaptive Noise Reduction". In: *arXiv preprint arXiv:1705.08378* (2017).

[26] Andrew L. Maas et al. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. URL: http://www.aclweb.org/anthology/P11-1015.

[27] Dongyu Meng and Hao Chen. "Magnet: a Two-Pronged Defense Against Adversarial Examples". In: *arXiv preprint arXiv:1705.09064* (2017).

[28] Jan Hendrik Metzen et al. "On Detecting Adversarial Perturbations". In: *arXiv preprint arXiv:1702.04267* (2017).

[29] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. "Linguistic regularities in continuous space word representations." In: *hlt-Naacl*. Vol. 13. 2013, pp. 746–751.

[30] Tomas Mikolov et al. "Distributed Representations of Words and Phrases and Their Compositionality". In: *CoRR* abs/1310.4546 (2013). URL: http://arxiv.org/abs/1310.4546.

[31] Tomas Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *CoRR* abs/1301.3781 (2013). URL: http://arxiv.org/abs/1301.3781.

[32] Takeru Miyato et al. "Distributional Smoothing With Virtual Adversarial Training". In: *stat* 1050 (2015), p. 25.

[33] T. Miyato et al. "Virtual Adversarial Training: a Regularization Method for Supervised and Semi-Supervised Learning". In: *ArXiv e-prints* (Apr. 2017). arXiv: 1704.03976 [stat.ML].

[34] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a Simple and Accurate Method To Fool Deep Neural Networks". In: *CoRR* abs/1511.04599 (2015). arXiv: 1511.04599. URL: http://arxiv.org/abs/1511.04599.

[35] Seyed-Mohsen Moosavi-Dezfooli et al. "Universal Adversarial Perturbations". In: *arXiv preprint arXiv:1610.08401* (2016).

[36] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. "Deep Neural Networks Are Easily Fooled: High Confidence Predictions for Unrecognizable Images". In: *CoRR* abs/1412.1897 (2014). URL: http://arxiv.org/abs/1412.1897.

[37] Tianyu Pang, Chao Du, and Jun Zhu. "Towards Robust Detection of Adversarial Examples". In: *CoRR* abs/1706.00633 (2017).

[38] N. Papernot, P. McDaniel, and I. Goodfellow. "Transferability in Machine Learning: From Phenomena To Black-Box Attacks Using Adversarial Samples". In: *ArXiv e-prints* (May 2016). arXiv: 1605.07277 [cs.CR].

[39] Nicolas Papernot et al. "Distillation As a Defense To Adversarial Perturbations Against Deep Neural Networks". In: *CoRR* abs/1511.04508 (2015). URL: http://arxiv.org/abs/1511.04508.

[40] Nicolas Papernot et al. "The Limitations of Deep Learning in Adversarial Settings". In: *CoRR* abs/1511.07528 (2015). URL: http://arxiv.org/abs/1511.07528.

[41] Kishore Papineni et al. "BLEU: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics. 2002, pp. 311–318.

[42] Jonathan Peck et al. "Lower bounds on the robustness to adversarial perturbations". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 804–813. URL: http://papers.nips.cc/paper/6682-lower-bounds-on-the-robustness-to-adversarial-perturbations.pdf.

[43] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[44] Radim Řehůřek and Petr Sojka. "Software Framework for Topic Modelling with Large Corpora". English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[45] Andras Rozsa, Ethan M. Rudd, and Terrance E. Boult. "Adversarial Diversity and Hard Positive Generation". In: *CoRR* abs/1605.01775 (2016).

[46] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. "The Earth Mover's Distance As a Metric for Image Retrieval". In: *International journal of computer vision* 40.2 (2000), pp. 99–121.

[47] Suranjana Samanta and Sameep Mehta. "Towards Crafting Text Adversarial Samples". In: *arXiv preprint arXiv:1707.02812* (2017).

[48] Eder Santana and George Hotz. "Learning a Driving Simulator". In: *CoRR* abs/1608.01230 (2016). URL: http://arxiv.org/abs/1608.01230.

[49] Y. Song et al. "PixelDefend: Leveraging Generative Models To Understand and Defend Against Adversarial Examples". In: *ArXiv e-prints* (Oct. 2017). arXiv: 1710.10766 [cs.LG].

[50] J. Su, D. Vasconcellos Vargas, and S. Kouichi. "One Pixel Attack for Fooling Deep Neural networks". In: *ArXiv e-prints* (Oct. 2017). arXiv: 1710.08864 [cs.LG].

[51] Christian Szegedy et al. "Intriguing Properties of Neural Networks". In: *CoRR* abs/1312.6199 (2013). URL: http://arxiv.org/abs/1312.6199.

[52] Pedro Tabacof and Eduardo Valle. "Exploring the Space of Adversarial Images". In: *arXiv preprint arXiv:1510.05328* (2015).

[53] D Warde-Farley and I Goodfellow. "Adversarial Perturbations of Deep Neural Networks". In: *Perturbation, Optimization and Statistics*. Ed. by Tamir Hazan, George Papandreou, and Daniel Tarlow. 2016.

[54] C. Wong. "DANCin Seq2seq: Fooling Text Classifiers With Adversarial Text Example Generation". In: *ArXiv e-prints* (Dec. 2017). arXiv: 1712.05419 [cs.LG].

[55] Cihang Xie et al. "Mitigating Adversarial Effects Through Randomization". In: *CoRR* abs/1711.01991 (2017).

[56] Yuanshun Yao et al. "Automated Crowdturfing Attacks and Defenses in Online Review Systems". In: *CoRR* abs/1708.08151 (2017).

[57] V. Zantedeschi, M.-I. Nicolae, and A. Rawat. "Efficient Defenses Against Adversarial Attacks". In: *ArXiv e-prints* (July 2017). arXiv: 1707.06728 [cs.LG].

[58] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-level Convolutional Networks for Text Classification". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes et al. Curran Associates, Inc., 2015, pp. 649–657. URL: http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification.pdf.

[59] Z. Zhao, D. Dua, and S. Singh. "Generating Natural Adversarial Examples". In: *ArXiv e-prints* (Oct. 2017). arXiv: 1710.11342 [cs.LG].