Міністерство освіти і науки України
Національний Університет «Києво-Могилянська Академія»
Кафедра математики факультету інформатики

Властивості розподілів штучно згенерованих зображень. Теоретична частина

Курсова робота за спеціальністю «113 Прикладна математика»

Керівник курсової роботи
к. фіз.-мат. наук, доцент
Крюкова Г.В.

к. фіз.-мат. наук, доцент
Дмитришин А.
Університет Еребро, Швеція

Виконав студент
Іванюк-Скульський Б.В.

Київ – 2020

Зміст

1 Introduction

In recent years, machine learning and, in particular, deep learning (DL) models have improved their performance in various tasks, e.g., image classification, speech recognition, natural language processing. However, even state-of-the-art models are vulnerable to so called adversarial perturbations. These perturbations applied to a correctly classified sample aren't visible for a human eye but lead to misclassification of the sample [5, 12, 13, 18, 19]. Clearly that such an issue may cause serious consequences in the applications where safety and security are priority, for example, autonomous driving. There have been recent attempts to explain this phenomenon, see e.g., [5], but a consistent theory is still missing.

In this paper, we propose a new approach to adversarial image detection. Our approach relies on the assumption that an adversarial perturbation pushes a sample away from a manifold where the correctly classified samples are concentrated. This allows us to use distributions of certain distances for detecting adversarial samples.

2 Background

We provide some basic background on machine learning, Generative Adversarial Networks (GAN), and autoencoders.

2.1 Deep Learning Classifiers

A deep learning classifier can be expressed as a mapping from an input space, denoted by R^{D}, to the space of classes, denoted by R^{C}, $f(X, \theta_c) : R^{\mathrm{D}} \to R^{\mathrm{C}}$ with trainable parameters θ_c. We will focus on neural networks with sigmoid output layer. Given the input x (i.e. sample from X) the predicted label for x is either 0 or 1, and denoted as $\hat{y} = \arg\max_{i \in [0;1]} f(x)_i$, where $f(x) = sigmoid(W_o Z + b_o)$ is a vector output from neural network bounded by 0 and 1. Common training objective in this setting is to minimize the binary cross-entropy (BCE) loss, defined as:

$$\mathcal{L}_{\mathrm{BCE}}(\mathrm{x}, \mathrm{y}) = -y \log F(x) - (1 - y) \log(1 - F(x)) \tag{1}$$

for a single input pair (x, y).

2.2 Adversarial Networks

Originally adversarial neural networks were represented as a multilayer perceptrons [3] that were composed from two parts: Generator network G(z, θ_g) which is a differentiable function with parameters θ_g that learns the mapping from input noise variable $p_z(z)$ to data space, and a discriminator (i.e., a deep learning classifier) network D(x, θ_d) with parameters θ_d that represents the probability of input x coming from real distribution rather than p_g (i.e. adversarial

Рис. 1: Sample images from real MNIST (1st row) and CelebA (3d row) datasets and corresponding adversarial images (2nd and 4th rows respectively) generated by DCGAN for 10 epochs

sample). In general we are aiming to maximize correct classification of training and generated samples for discriminator network and simultaneously minimize $\log(1 - D(G(z)))$:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \quad (2)$$

given $V(G, D)$ as a value function of a two-player minimax game.

In 2016 Radford, Metz, and Chintala [17] introduced an extension of GAN described above, known as Deep Convolutional Generative Adversarial Network (DCGAN). DCGAN consist of a discriminator network, made of stridden convolutional layers, batch normalization layers, and LeakyReLU activations; as well as a generator network, comprised of convolutional-transpose layers, batch normalization layers, and ReLU activations.

2.3 Autoencoders

Autoencoder is a type of neural network that aims to copy its input to its output. The network is composed of two parts: encoder function $h = f(x)$ and a decoder function that performs reconstruction $r = g(h)$. Autoencoders are designed to copy an input approximately, meaning the model is forced to prioritize which aspects of the input should be copied, it often learns useful properties of the data [4].

$$\mathcal{L}(\mathrm{x}, \mathrm{g}(\mathrm{f}(\mathrm{x}))) = \min ||x - (f \circ g)x||^2 \quad (3)$$

given $\mathcal{L}$ a loss function penalizing $g(f(x))$ being dissimilar from input x.

3 Methods

Many learning algorithms exploit the idea that data concentrates around a low dimensional manifold. In the following method, we use geometrical characteristics of such a manifold for detecting adversarial examples.

We make the following assumptions on our data:

(a) Data points belong to a smooth manifold;

(b) For every point on this manifold (data point or not), the centroid of its k nearest neighbors (KNN) is approximately at the same distance from this point;

(c) For a point that does not belong to the manifold the set of its KNN coincides with the set of KNN of its projection on the manifold.

We suggest a method for detecting adversarial examples by comparing the distances to the centroids for a given point and its KNN. This method is base on the following theoretical justification.

Let x be a point outside of a given manifold, and M_x be the centroid of its k nearest neighbors. Using the Euclidean distance, $|M_x x|^2 = |x\mathrm{Pr}_x|^2 + |\mathrm{Pr}_x M_x|^2$, where Pr_x is the projection of x on the manifold.

From our assumption (c) we have that if M_x is a centroid for KNN of x then M_x is also a centroid for KNN of Pr_x. Combining this with the assumption (b) we have that $|\mathrm{Pr}_x M_x| = |x' M'_{x'}|$, for any points x' and its KNN's centroid $M'_{x'}$. Therefore $|x M_x|^2 = |x\mathrm{Pr}_x|^2 + |x' M'_{x'}|^2$.

Now since X is outside of our manifold we have $|x\mathrm{Pr}_x| > 0$ and thus $|x M_x| > |x' M'_{x'}|$. Moreover, $|\mathrm{x\ Pr}_x| = \sqrt{|x M_x|^2 - |x' M'_{x'}|^2}$.

Therefore we may use the following statement as a criterion if a given example is adversarial or not:

For every point outside of a given (differential) manifold and large enough integer k, the distance to the centroid of its k nearest neighbors is significantly larger than the distance from a point on this manifold to the centroid of its k nearest neighbors.

Obtaining similar theoretical results with relaxed conditions (a)–(c) is a part of our future work.

The above theory results in the following method for detecting adversarial examples. For a given image x we find the set of its nearest neighbors. Then for this set, we find the centroid point M_x and the distance $M_x x$. We do the same for each neighbor-image from selected set of neighbour images, resulting in the distance distribution of neighbor images to their neighbor-sets centroids. These distances suggest us whether the input image x is adversarial or not. This procedure for our experiments is described in more details in the following section, see also Algorithm 1.

Adversarial images are generated using DCGAN introduced in Section 2.2. The dimensionality of the images is reduced using autoencoders, Section 2.3.

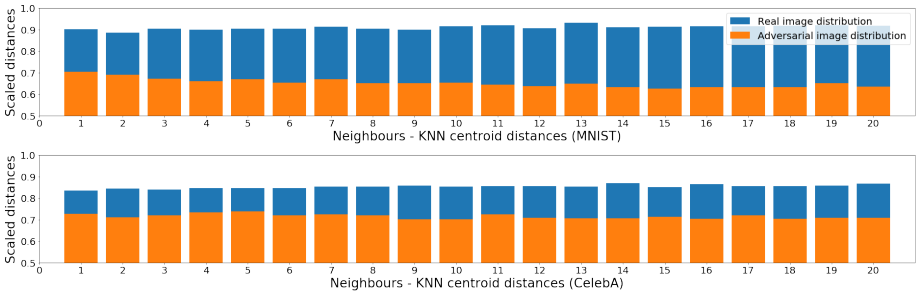

Рис. 2: Distance distribution to centroid for KNN (K = 20) of input images x_i scaled by the distance $M_{x_i}x_i$ (where $M_{x_i}x_i$ is a centroid point of KNN's of x_i) averaged after 1000 experiments. Both for real and adversarial samples

4 Experiments

In this section we present results which we have received after performing our theoretical assumptions having both real and adversarial image samples. We start by describing how the experiments were performed.

For our experiments we took **MNIST** (60000 images) [8] and **CelebA** (200000 images) [10] datasets. The pixel values of images were scaled to be in the range of $[-1.0, 1.0]$. Given this data, we have generated 10000 adversarial images both for MNIST and CelebA data using DCGAN introduced in Section 2.2. Few samples of generated images both for MNIST and CelebA datasets can be viewed in Fig. 1.

Afterward, we started to perform our experiment. In order to reduce the dimensionality and keep the key features of the images, we use autoencoders trained on real images. To be exact, in the first experiment, we reduce dimensions of an MNIST sample image from $1 \times 28 \times 28$ to $8 \times 4 \times 4$ and then flatten it to obtain the 128-dimensional vector. In the second experiment, we reduce the input CelebA image from $3 \times 218 \times 178$ to $8 \times 4 \times 4$ and flatten it like in the previous experiment. After obtaining 128-dimensional vectors, we are following with Algorithm 1, described below.

The first step is to find neighbouring images of a given image x. For this purpose, we have tried euclidean distance and cosine similarity that results in almost the same arrays of neighbors S_x. Resulting array of neighbours S_x is in sorted order starting from the nearest to x. For a selected array of neighbor images, we calculate coordinates of the centroid point M_x and the corresponding $M_x x$ distance. The rednext step, we perform the same operation with each selected neighbor-image from the array S_x, resulting in the distance distribution of neighbor images to their neighbor-arrays centroids. These distances give us a piece of the evidence whether the input image x is adversarial or not.

In practice we have randomly selected 1000 samples from both real and adversarial images and performed algorithm described above on them. As after

Algorithm 1 comparing the distances to the centroids for a given point x and its KNN

Input: sample image x
Output: distance distribution
Initialisation : M_{distr} empty array
$S_x \leftarrow$ search for neighbours of x
$M_x \leftarrow$ centroid point of obtained set S_x
$M_x x \leftarrow$ distance between x and M_x
for s_i from set S_x **do**
  $S_i \leftarrow$ search for neighbours of s_i
  $M_{s_i} \leftarrow$ centroid point of obtained set S_i
  $M_{s_i} s_i \leftarrow$ distance between M_{s_i} and s_i
  $M_{distr}[i] \leftarrow M_{s_i} s_i$
end for
return $M_x x$, M_{distr}

each iteration we receive distance $M_{x_i} x_i$ and distribution array M_{distr_i}, for the sake of generality we have divided each value from distribution array by distance $M_{x_i} x_i$ resulting in a scaled array that can be interpreted as percentage distance of $M_{x_i} x_i$. Results can be view in Fig. 2. From two plots we can see that in all the cases scaled distance for adversarial samples (orange bar plots) is lower by a certain percentage comparing to real samples (blue bar plots). For the MNIST dataset, this difference of distance means is about 20% - 25% and for CelebA data around 15%. This difference in percentages for MNIST and CelebA data can be explained by the nature of images. In MNIST they are greyscaled, meaning much lower variance in pixel values comparing to images taken from CelebA data. Given information obtained from our experiment, we can see that for both MNIST and CelebA images, real samples are more concentrated in the same manifold comparing to the adversarial images, that is proving that our assumptions from 3 Methods (a) - (c) are True.

5 Related Work

Investigation of adversarial perturbations is one of the most active areas of deep learning research. A fundamental understanding of both the adversarial attacks and defenses is crucial and has already resulted in a large number of publications.

Here we list just a few, most relevant results on the detection of adversarial examples. In [6, 9] the authors show that adversarial examples are not drawn from the same distribution as the original data, and can thus be detected using statistical tests. The dimensional properties of adversarial regions are characterized in [11] using so-called Local Intrinsic Dimensionality. In [15] the authors propose to use the knowledge extracted from a DNN to improve its resilience to adversarial samples.

Unfortunately, the defense strategies are typically not successful if an attacker is allowed to modify the attack algorithm using the information about the defense mechanism of a particular network, see e.g., [1] and [2]. This results in a large number of open problems and a need for further research.

6 Conclusion

In this paper, we present a method of detecting adversarial examples. To be exact, we compare the distance from a given example to a centroid of its nearest neighbors with the distribution of the distances from these neighbors to the centroids of their neighbors. For an adversarial example, such a distance is much larger than the corresponding distances for its neighbors while for a real example it is not.

The presented method is a step towards a better understanding of adversarial examples using the geometrical properties of data. We hope that such geometrical methods can make a significant contribution to our ability of detecting adversarial examples. Moreover, we expect the methods based on geometrical properties to be effective in the cases where the architecture-based-methods fail. Thus there is a good potential that combining both these types of methods will create a powerful tool for detection of adversarial examples.

Another argument in favor of the methods based on the geometry of the data is that adversarial examples seem to be generalizable across different models, see e.g., [18] and [14]. Therefore methods based on the geometry of the data may be more universal but also more dependent on the available training data.

As a part of our future research we plan to test our method on the sets of adversarial examples that are generated in different ways [16] as well as on natural adversarial examples [7].

Література

[1] Anish Athalye, Nicholas Carlini та David Wagner. *Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples*. 2018.

[2] Nicholas Carlini та David Wagner. "Adversarial examples are not easily detected: Bypassing ten detection methods". в: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM. 2017, с. 3—14.

[3] Ian J. Goodfellow та ін. "Generative Adversarial Nets". в: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14. Montreal, Canada: MIT Press, 2014, с. 2672—2680.

[4] Ian Goodfellow, Yoshua Bengio та Aaron Courville. *Deep Learning*. MIT Press, 2016.

[5] Ian Goodfellow, Jonathon Shlens та Christian Szegedy. "Explaining and Harnessing Adversarial Examples". в: *ICLR* abs/1412.6572 (2015).

[6] Kathrin Grosse та ін. "On the (statistical) detection of adversarial examples". в: *arXiv preprint arXiv:1702.06280* (2017).

[7] Dan Hendrycks та ін. "Natural Adversarial Examples". в: *arXiv preprint arXiv:1907.07174* (2019).

[8] Y. Lecun та ін. "Gradient-based learning applied to document recognition". в: *Proceedings of the IEEE* 86.11 (1998), с. 2278—2324.

[9] Xin Li та Fuxin Li. "Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics". в: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), с. 5775—5783.

[10] Ziwei Liu та ін. "Deep Learning Face Attributes in the Wild". в: *Proceedings of International Conference on Computer Vision (ICCV)*. груд. 2015.

[11] Xingjun Ma та ін. "Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality". в: *Proceedings of the 6th International Conference on Learning Representations (ICLR), Vancouver, BC, Canada* (2018).

[12] Anh Nguyen, Jason Yosinski та Jeff Clune. "Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images". в: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2015.

[13] N. Papernot та ін. "The Limitations of Deep Learning in Adversarial Settings". в: *2016 IEEE European Symposium on Security and Privacy (EuroS P)*. бер. 2016, с. 372—387. DOI: 10.1109/EuroSP.2016.36.

[14] Nicolas Papernot, Patrick McDaniel та Ian Goodfellow. "Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples". в: *arXiv preprint arXiv:1605.07277* (2016).

[15] Nicolas Papernot та ін. "Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks". в: *Proceedings of the 37th IEEE Symposium on Security and Privacy, San Jose, CA.* (2016).

[16] Nicolas Papernot та ін. "Technical Report on the CleverHans v2.1.0 Adversarial Examples Library". в: *arXiv preprint arXiv:1610.00768* (2018).

[17] Alec Radford, Luke Metz та Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". в: *arXiv:1511.06434* (2016).

[18] Christian Szegedy та ін. "Intriguing properties of neural networks". в: *CoRR* abs/1312.6199 (2013). arXiv: 1312.6199. URL: http://arxiv.org/abs/1312.6199.

[19] Yevgeniy Vorobeychik та Murat Kantarcioglu. "Adversarial Machine Learning". в: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12.3 (2018), с. 1—169. DOI: 10.2200/S00861ED1V01Y201806AIM039. eprint: https://doi.org/10.2200/S00861ED1V01Y201806AIM039. URL: https://doi.org/10.2200/S00861ED1V01Y201806AIM039.