Seminar thesis

Adversarial Label Flips (sexier title?)

Matthias Dellago & Maximilian Samsinger

June 15, 2021

Abstract

Given a neural network (NN) trained to classify and a untargeted evasion attack [], in what class does the adversarial example fall? In the following, we will answer this question by evaluating some state of the art attacks, on a simple NN trained on industry standard datasets [**?**, **?**, **?**, **?**]. We discover that intuitively similar classes are more likely to be confused with another.

1 Introduction

Adversarial examples exist. Targeted and untargeted. What does that look like? We use confusion matrices. example. how did we do it? -¿ NN + foolbox + mnist cifar (section methods) what did we get out of it? -¿ symmetric matrices and interesting insight about noise. (section results)

In 2013 Szegedy et al. demonstrated that attacking deep neural networks (NN) are susceptible to attacks [1]. These adversarial examples consist of a small perturbation applied to a otherwise innocuous input, engineered to cause the NN to misbehave.

In our case, the inputs will be images and the attack will be small changes to said image, designed to cause the classifier NN to misclassify the target.

These attacks on classifiers come in two different variations: targeted and untargeted. In targeted attacks[] the attacker aims to have the adversarial example identified as a specific class by the NN. (Say, misclassify a dog as a cat.) An untargeted attack meanwhile, only tries to evade correct classification. (Make this dog appear as anything, apart from a dog.)

Now, when considering untargeted attacks, the question as what the adversarial image *actually* is then classified, naturally arises. This is what we will experimentally answer in this paper.

We will present our results in terms of confusion matrices. In figure !! you can see an example. In larger matrices numbers become more difficult to process, so we will display our results in heatmap-style images (fig. !!). (Section results)

In our experiments we used the foolbox framework [], and simple NNs trained on the MNIST, FashionMNIST, and CIFAR-10 datasets []. We applied three state of the art attacks: Projected Gradient Decent (PDG)[], Carlini-Wagner [] and ...-Bethge[].

We show that, for the CIFAR-10 dataset the confusion matrices are surprisingly symmetric, and intuitively similar classes are often confused with each other. Furthermore we observe that for attacks which can choose large pertubations, there exist certain attractor classes, which most of the adversarial images are classified as. (Section Results.)

2 Background and related work

Existence of adversarial examples Since being first demonstrated [1] a large body of literature has flourished around adversarial examples. Blablabla...

Notions of similarity As mentioned previously we want to fool the NN with a small perturbation, that is to say a image which is very similar. But what do we precisely mean by "small pertubation" and "similar"? We could argue that an image with only one pixel changed is very similar, but also that an image in which each pixel was altered only very slightly is similar.

These different intuitive notions of distance and similarity are captured by norms.

The distance of two images (x,y) (i.e. vectors) is described by a metric D. This metric can be defined via different norms, most commonly the L^∞-, L^1- and L^2-norms.(The L^0-"norm" is also frequently used, though not a norm in the strict sense.) The distance between these the two images, is then defined as the L^p-norm of their difference, for a given p:

$$D(x, y) := \|y - x\|_p \tag{1}$$

... Hmmm probably not that important, maybe if there is time and space after everything else in finished.

2.1 Attacks

Fast gradient sign method Goodfellow et al. famously developed the fast gradient sign method (FGSM) [2], making attacks fast and easy. Its key insight was that the backpropagation commonly used to update the weights and biases can be applied all the way back to the input data itself to yields the gradient of the cost function. They then apply gradient decent to find an adversarial example. Since they optimise for the L^∞-norm, all entries of the perturbance are scaled to the same magnitude.

Projected gradient descent Projected gradient descent (PGD) was first shown in [3]. Conceptually it is very similar to iterating FGSM until converging in a local misclassification optimum. The "projected" part of the name, derives from the fact that upon leaving a ball of radius ϵ instead of continuing iteration, they project back onto said ball. From iterated FGSM resumes. This attack leads to formidable results, especially using the L^∞-norm.

Carlini-Wagner attack Carlini and Wagner [4] invented a different style of attack, where the cost function of the classifier and the distance of the adversarial example are wrapped into one function. They can then simultaneously optimise for both using the Adam stochastic optimiser [5].

Brendel-Bethge attack Brendel and Bethe invented a quite different method, which they aptly named Brendel & Bethge attacks [6]. **zu spicy? :D** Their method works by starting from a image deep inside the misclassification region and then preforming binary search between it and the benign, target image, to find the decision boundary. Once there, they move along the boundary

to minimise the distance to the benign image, yielding a powerful adversarial example.

Foolbox A Python library with lots of attacks [7]. They include the attacks above.

3 Methods

Reference to our github.

3.1 Datasets

MNIST, Fashion MNIST, CIFAR-10

3.2

We probably use `https://arxiv.org/pdf/1608.04644.pdf` Table 1 as a neural network for MNIST & Fashion-MNIST.

4 Experiments

5 Results

pictures, pictures, and maybe a graph or two

6 Discussion

Symmetry of matrices -¿ maybe find a way to quantify symmetry?
-¿ NN can recognise "similarity"

Attractor classes -¿ manage with an extra "noise"-class or so?

In Figures X, Y and Z one can observe that adversarial examples computed with large perturbation budgets ϵ are misclassified as "8", "TODO" and "frog" for MNIST, Fashion-MNIST and CIFAR-10 respectively. In order to shed light onto this phenomenon we generate and classify 10000 white noise images sampled from a uniform distribution on the input domain. Figure A shows that these randomly generated images are also, most commonly, classified as "8", "TODO" and "frog" respectively. This result suggests that the neural networks in question have a default output for low probability images with respect to distribution of the input domain, which in turn affects adversarial examples computed with large perturbation budgets.

7 Conclusion

What was the main idea.

8 Contribution Statement

This is joint work from Maximilian Samsinger and Matthias Dellago. Max wrote the code... We thank Alexander Schlögl for the research idea.

References

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

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

[3] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[6] Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation, 2019.

[7] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017.