Notes on Adversarial Examples

David Meyer

dmm@{1-4-5.net,uoregon.edu,...}

March 14, 2017

1 Introduction

The surprising discovery of *adversarial examples* by Szegedy et al. [6] has led to new ways of thinking about unsupervised training of deep models [2] while at the same time casuing confusion and concern about the nature of learning in such (deep) models. These notes explore the analysis of adversarial examples given in [3] and elsewhere.

What Szegedy et al. [6] discovered was that several machine learning models, including state-of-the-art neural networks, are vulnerable to adversarial examples. That is, these machine learning models can misclassify examples that are only slightly different (imperceptibly so in many cases) from correctly classified examples drawn from the data distribution. In many cases, a wide variety of models with different architectures trained on different subsets of the training data misclassify the same adversarial example (this is kind of shocking). The implication is that adversarial examples expose fundamental problems in popular training algorithms.

At the time of publication [6], the cause of these adversarial examples was a mystery, and speculative explanations have suggested it is due to extreme nonlinearity of deep neural networks, perhaps combined with insufficient model averaging and insufficient regularization of the purely supervised learning problem. What [3] shows is that these speculative hypotheses are unnecessary. As we shall see, linear behavior in high-dimensional spaces is sufficient to cause the existence of adversarial examples.

In addition, [3] shows that generic regularization strategies such as dropout, pre-training and model averaging or ensembling do not confer a significant reduction in a model's vulnerability to adversarial examples. On the other hand, changing to nonlinear model families such as RBF networks can confer some resistance (to such adversarial examples).

These explanations reveal fundamental tradeoffs between models that are easy to train[1] and models that use nonlinear effects to resist adversarial perturbation. Not surprisingly, these tradeoffs appear to be fundamental; however, Goodfellow et.al. [3] suggest that it may be possible to escape these tradeoffs by designing more powerful optimization methods that can successfully train more nonlinear models.

2 Linear explanation of Adversarial Examples

This section follows the analysis and notation of Section 3 of [3], which explains the existence of adversarial examples for linear models. The section starts with a description of the precision of the sensor or storage media. Here they use the common example digital images, which often use only eight bits per pixel (gray scales) and as a result they discard all information below $\frac{1}{255}$ of the dynamic range. The point here is that since the precision of the features is limited, it makes little sense for a classifier to respond differently to an input $\boldsymbol{x}$ than to an adversarial input $\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\eta}$, if of course every element of the perturbation $\boldsymbol{\eta}$ is smaller than the precision of the features. Let's say that ϵ is the largest value *below* the resolution of the sensor (or storage media). Then for problems with well-separated classes, we expect a classifier to assign the same class to $\mathbf{x}$ and $\hat{\mathbf{x}}$, so long as $||\boldsymbol{\eta}||_\infty < \epsilon$ (where $||\boldsymbol{x}||_\infty$ is the *max* or *infinity* norm).

Next, consider the activation induced by an adversarial example $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}} = \mathbf{x} + \boldsymbol{\eta} \qquad \text{\# adversarial example } \hat{\mathbf{x}} \text{ given perturbation } \boldsymbol{\eta} \tag{1}$$

$$\mathbf{w}^{\mathrm{T}}\hat{\boldsymbol{x}} = \mathbf{w}^{\mathrm{T}}\boldsymbol{x} + \mathbf{w}^{\mathrm{T}}\boldsymbol{\eta} \qquad \text{\# activation } \mathbf{w}^{\mathrm{T}}\hat{\boldsymbol{x}} \text{ induced by } \boldsymbol{\eta} \tag{2}$$

Notes:

- The adversarial perturbation causes the activation to grow by $\mathbf{w}^{\mathrm{T}}\boldsymbol{\eta}$
- $\mathbf{w}^{\mathrm{T}}\boldsymbol{\eta}$ is maximized, subject to $||\boldsymbol{\eta}||_\infty < \epsilon$, by assigning $\boldsymbol{\eta} = \text{sign}(\boldsymbol{w})$ **why is this true?**

Now, if $\dim(w) = n$ and $\overline{|w|} = m$, then the activation $\mathbf{w}^{\mathrm{T}}\boldsymbol{\eta}$ will grow by ϵmn What is interesting here (among other things) is that the max norm $||\boldsymbol{\eta}||_\infty$, does not grow with the dimensionality of the input, while at the same time the change in activation caused by $\boldsymbol{\eta}$ can grow linearly with n. As a result in high dimensional cases many very small changes to the input can add up to one large change to the output. So essentially we see that even a simple linear model can have adversarial examples if its input has sufficient dimensionality. This result is in stark contrast to thinking around the time of [6], when it was thought that adversarial examples were a property of highly non-linear deep neural networks.

[1] due to their linearity

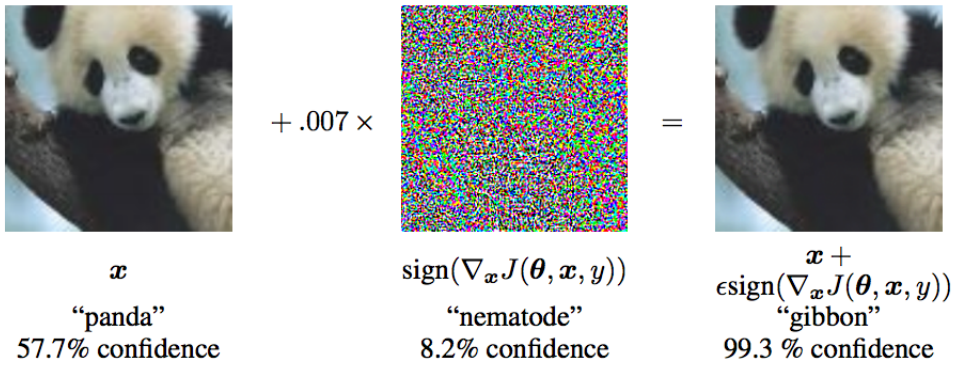

Figure 1: Fast adversarial example generation applied to GoogLeNet on ImageNet [6]. GoogLeNet's classification of this image can be changed, perhaps surprisingly, by adding an imperceptibly small vector whose elements are $\text{sign}(\nabla_x J(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y}))$, that is, the sign of the gradient of the cost function. Here $\epsilon = .007$, roughly $2 * (1/255)$, corresponds to the magnitude of the smallest bit of an 8 bit image encoding (after GoogLeNet's conversion to real numbers).

3 What about Non-Linear Models?

Goodfellow, et.al [3] hypothesizes that many neural network architectures are too linear to resist linear adversarial perturbation. These include LSTMs [5], ReLUs [1] and Maxout networks [4], which are all intentionally designed to behave in very linear ways so that they will be easier to optimize[2]. The important conclusion here is that the behavior of simple linear models under adversarial perturbation $\boldsymbol{\eta}$ described in Section 2 should also apply to ("damage") neural networks.

In typical form, let $\boldsymbol{\theta}$ be parameters of a model, $\mathbf{y}$ the targets associated with $\mathbf{x}$ (in the supervised learning case) and $J(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y})$ be the cost function (optimization objective) to train the neural network. Here we can linearize the cost function around the current value of $\boldsymbol{\theta}$, obtaining an optimal max-norm constrained perturbation of

$$\boldsymbol{\eta} = \epsilon\text{sign}(\nabla_x J(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y})) \tag{3}$$

Equation 3 is referred to as the *fast gradient sign method* for generating adversarial exam-

[2]The more nonlinear models (e.g., sigmoid networks) are carefully tuned to spend most of their time in the non-saturating, more linear regime for the same reason.

ples. Note that, importantly, the gradient in Equation 3 can be efficiently computing using back-propagation. An example of the fast gradient sign method is shown in Figure 1.

References

[1] Glorot, X., Bordes, A., and Bengio, Y. Deep sparse rectifier neural networks. In *Aistats* (2011), vol. 15, p. 275.

[2] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[3] Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *https://arxiv.org/abs/1412.6572* (2014).

[4] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A. C., and Bengio, Y. Maxout networks. *ICML (3) 28* (2013), 1319–1327.

[5] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural Computation 9*, 8 (2017/03/14 1997), 1735–1780.

[6] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *https://arxiv.org/abs/1312.6199* (2013).