Your 598 ID: 5107

Title of paper: "'Explaining and Harnessing Adversarial Examples" by Goodfellow, Shlens, and Szegedy.
What is their primary result? The authors argue that a neural network's vulnerability to adversarial examples comes from its linear nature and no the nonlinearity and overfitting.
Why is this important? This work focuses on an aspect of the problem which has not been previously addressed. Moreover, the point of view yields simple and fast methods of generating adversarial examples and provide examples for adversarial training.
What are their key ideas? The authors show that a simple linear model can have adversarial examples if its input has sufficient dimensionality. This comes from the following observation. If we consider the product between a weight vector $\mathbf{w}$ and an adversarial example $\bar{\mathbf{x}}$, we have

$$\mathbf{w} \cdot \bar{\mathbf{x}} = \mathbf{w} \cdot \mathbf{x} + \mathbf{w} \cdot \boldsymbol{\eta}$$

with adversarial perturbation $\mathbf{w} \cdot \boldsymbol{\eta}$, we can maximize the increase subject to the max norm constraint on $\boldsymbol{\eta}$ by assigning $\boldsymbol{\eta} = \text{sgn}(\mathbf{w})$. Assuming $\mathbf{w}$ is n-dimensional, and the average magnitude of an element of the weight vector is m, we have a growth in the activation of ϵmn. Since $\|\eta\|_\infty$ grows linearly, for high dimensional problems, infinitesimal changes to the input can add up to one large change to the output.

This observation leads to their *fast gradient sign method* of generating adversarial examples. The method causes a wide variety of models to misclassify their input.

The authors then consider training their adversarial model on logistic regression and analyze that.

The authors question why adversarial examples also generalize; that is, an example generated for a specific model is often misclassified by other models. They hypothesize that neural networks trained no current methodologies all resemble the linear classifier learned on the same training set. The reference classifier learns approximately the same classification weights.
Did they implement something? The authors implement a fast gradient sign method: Given $\boldsymbol{\theta}$ the parameters of the model, $\mathbf{x}$ the input to the model, y the targets associated with $\mathbf{x}$ and $J(\boldsymbol{\theta}, \mathbf{x}, y)$ the cost, the fast gradient methods is defined as

$$\boldsymbol{\eta} = \epsilon \, \text{sgn}(\nabla_{\mathbf{x}} J(\boldsymbol{\theta}, \mathbf{x}, y)).$$

How did they train and evaluate it? Using $\epsilon = 0.25$, the authors cause a shallow softmax classifier to have an error rate of 99.9% with average confidence of 79.3% on the MNIST test set. Similarly, using $\epsilon = 0.1$, they obtain an error rate of 87.15% and average probability of 96.6% assigned to incorrect labels with a convolutional maxout network on a preprocessed version of CIFAR-10.
Grader's 598 ID: