EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES

Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy

Bogdashevskaya Mariya
27.04.18

Adversarial examples

Several machine learning models, including neural networks, consistently misclassify adversarial examples—inputs formed by applying small but intentionally worst-case perturbations to examples from the dataset, such that the perturbed input results in the model outputting an incorrect answer with high confidence.

On some datasets, such as ImageNet (Deng et al., 2009), the adversarial examples were so close to the original examples that the differences were indistinguishable to the human eye.

The same adversarial example is often misclassified by a variety of classifiers with different architectures or trained on different subsets of the training data.

Training on adversarial examples can regularize the model.

Szegedy et al. (2014b)

The linear explanation of adversarial examples

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\eta}$$

the same class to $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ so long as $||\boldsymbol{\eta}||_\infty < \epsilon$

Consider the dot product between a weight vector $\boldsymbol{w}$ and an adversarial example $\tilde{\boldsymbol{x}}$:

$$\boldsymbol{w}^\top \tilde{\boldsymbol{x}} = \boldsymbol{w}^\top \boldsymbol{x} + \boldsymbol{w}^\top \boldsymbol{\eta}.$$

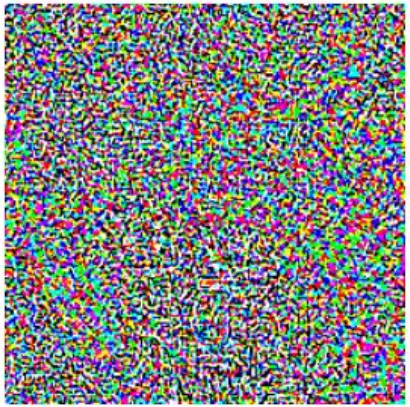
The adversarial perturbation causes the activation to grow by $\boldsymbol{w}^\top \boldsymbol{\eta}$.We can maximize this increase subject to the max norm constraint on $\boldsymbol{\eta}$ by assigning $\eta = \text{sign}(\boldsymbol{w})$. If $\boldsymbol{w}$ has n dimensions and the average magnitude of an element of the weight vector is m, then the activation will grow by ϵmn. Since $||\eta||_\infty$ does not grow with the dimensionality of the problem but the change in activation caused by perturbation by η can grow linearly with n, then for high dimensional problems, we can make many infinitesimal changes to the input that add up to one large change to the output.

Linear perturbation of non-linear models

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$=$

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

Let $\boldsymbol{\theta}$ be the parameters of a model, $\boldsymbol{x}$ the input to the model, y the targets associated with $\boldsymbol{x}$ (for machine learning tasks that have targets) and $J(\boldsymbol{\theta}, \boldsymbol{x}, y)$ be the cost used to train the neural network. We can linearize the cost function around the current value of $\boldsymbol{\theta}$, obtaining an optimal max-norm constrained pertubation of

$$\boldsymbol{\eta} = \epsilon \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right).$$

and panda -> gibbon :(

Adversarial training of linear models versus weight decay

If we train a single model to recognize labels $y \in \{-1, 1\}$ with $P(y = 1) = \sigma\left(\boldsymbol{w}^\top \boldsymbol{x} + b\right)$ where $\sigma(z)$ is the logistic sigmoid function, then training consists of gradient descent on

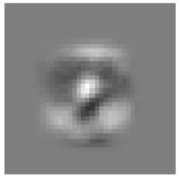
$$\mathbb{E}_{\boldsymbol{x}, y \sim p_{\text{data}}} \zeta(-y(\boldsymbol{w}^\top \boldsymbol{x} + b))$$

where $\zeta(z) = \log\left(1 + \exp(z)\right)$ is the softplus function.

The adversarial version of logistic regression is therefore to minimize

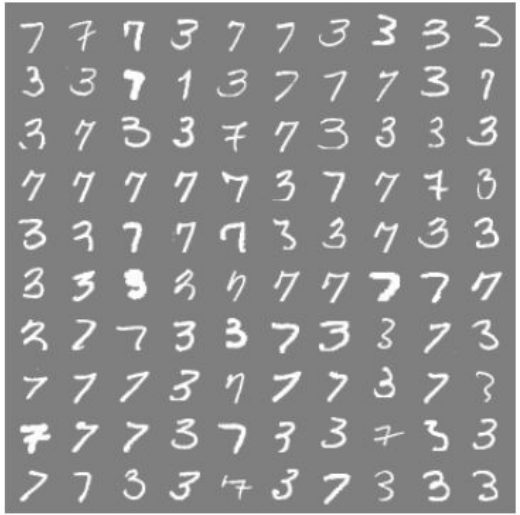
$$\mathbb{E}_{\boldsymbol{x}, y \sim p_{\text{data}}} \zeta(y(\epsilon||\boldsymbol{w}||_1 - \boldsymbol{w}^\top \boldsymbol{x} - b)).$$

OBJ

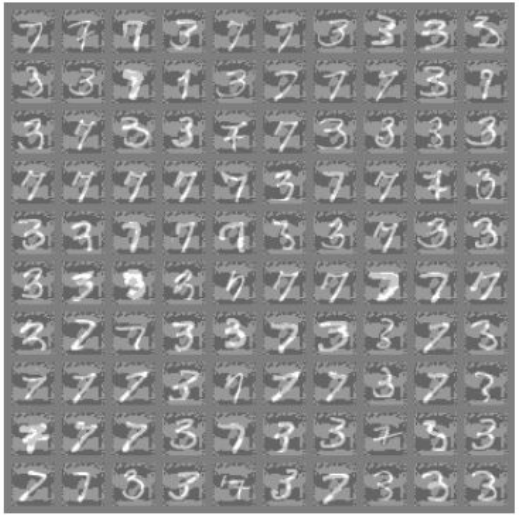


(a)

(b)



(c)

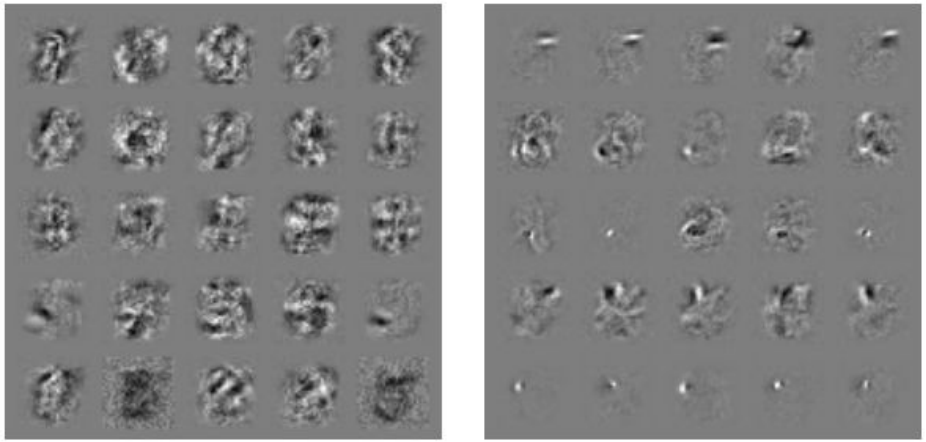


(d)

Adversarial training of deep networks

We found that training with an adversarial objective function based on the fast gradient sign method was an effective regularizer:

$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1 - \alpha) J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right)).$$



Summary

Adversarial examples can be explained as a property of high-dimensional dot products. They are a result of models being too linear, rather than too nonlinear.

The generalization of adversarial examples across different models can be explained as a result of adversarial perturbations being highly aligned with the weight vectors of a model, and different models learning similar functions when trained to perform the same task.

Models that are easy to optimize are easy to perturb.

Linear models lack the capacity to resist adversarial perturbation; only structures with a hidden layer (where the universal approximator theorem applies) should be trained to resist adversarial perturbation