# EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES

# 解释和利用对抗样本

Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy

Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy

Google Inc., Mountain View, CA
谷歌公司，加利福尼亚州山景城
{goodfellow, shlens, szegedy}@google.com

{goodfellow, shlens, szegedy}@google.com

## ABSTRACT

## 摘要

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. Early attempts at explaining this phenomenon focused on nonlinearity and overfitting. We argue instead that the primary cause of neural networks' vulnerability to adversarial perturbation is their linear nature. This explanation is supported by new quantitative results while giving the first explanation of the most intriguing fact about them: their generalization across architectures and training sets. Moreover, this view yields a simple and fast method of generating adversarial examples. Using this approach to provide examples for adversarial training, we reduce the test set error of a maxout network on the MNIST dataset.

几种机器学习模型，包括神经网络，持续错误分类对抗样本——这些输入是通过对数据集中样本施加小但故意的最坏情况扰动而形成的，以至于扰动后的输入导致模型以高置信度输出错误答案。早期对这一现象的解释集中在非线性和过拟合上。我们则认为，神经网络对对抗扰动的脆弱性主要源于其线性特性。这一解释得到了新的定量结果的支持，同时首次解释了它们最引人注目的事实: 它们在不同架构和训练集之间的泛化。此外，这一观点提供了一种简单快速的生成对抗样本的方法。利用这种方法为对抗训练提供示例，我们减少了在 MNIST 数据集上 maxout 网络的测试集错误率。

## 1 INTRODUCTION

## 1 引言

Szegedy et al. (2014b) made an intriguing discovery: several machine learning models, including state-of-the-art neural networks, are vulnerable to adversarial examples. That is, these machine learning models misclassify examples that are only slightly different 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 suggests that adversarial examples expose fundamental blind spots in our training algorithms.

Szegedy 等 (2014b) 做出了一个引人注目的发现: 几种机器学习模型，包括最先进的神经网络，易受对抗样本的影响。也就是说，这些机器学习模型错误分类与从数据分布中正确分类的样本仅有轻微差异的样本。在许多情况下，具有不同架构的各种模型在不同的训练数据子集上训练，却错误分类同一个对抗样本。这表明，对抗样本暴露了我们训练算法中的基本盲点。

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. We show that these speculative hypotheses are unnecessary. Linear behavior in high-dimensional spaces is sufficient to cause adversarial examples. This view enables us to design a fast method of generating adversarial examples that makes adversarial training practical. We show that adversarial training can provide an additional regularization benefit beyond that provided by using dropout (Srivastava et al., 2014) alone. Generic regularization strategies such as dropout, pretraining, and model averaging do not confer a significant reduction in a model's vulnerability to adversarial examples, but changing to nonlinear model families such as RBF networks can do so.

这些对抗样本的原因仍然是一个谜，推测性的解释认为这可能是由于深度神经网络的极端非线性，可能还结合了模型平均不足和纯监督学习问题的正则化不足。我们表明这些推测性的假设是不必要的。在高维空间中的线性行为足以导致对抗样本的产生。这一观点使我们能够设计出一种快速生成对抗样本的方法，使对抗训练变得可行。我们表明，对抗训练可以提供额外的正则化益处，超出仅使用 dropout (Srivastava et al., 2014) 所提供的益处。诸如 dropout、预训练和模型平均等通用正则化策略并未显著降低模型对对抗样本的脆弱性，但转向 RBF 网络等非线性模型族可以做到这一点。

Our explanation suggests a fundamental tension between designing models that are easy to train due to their linearity and designing models that use nonlinear effects to resist adversarial perturbation. In the long run, it may be possible to escape this tradeoff by designing more powerful optimization methods that can succesfully train more nonlinear models.

我们的解释暗示了在设计易于训练的线性模型与利用非线性效应抵抗对抗扰动之间存在根本性的张力。从长远来看，可能通过设计更强大的优化方法来成功训练更非线性的模型，从而逃避这种权衡。

## 2 RELATED WORK

## 2 相关工作

Szegedy et al. (2014b) demonstrated a variety of intriguing properties of neural networks and related models. Those most relevant to this paper include:

Szegedy 等人 (2014b) 展示了神经网络及相关模型的一系列引人注目的特性。与本文最相关的包括:

- Box-constrained L-BFGS can reliably find adversarial examples.

- 盒约束 L-BFGS 可以可靠地找到对抗样本。

- 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.

- 在一些数据集上，例如 ImageNet (Deng et al. 2009)，对抗样本与原始样本的差异如此之小，以至于人眼无法区分。

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

- 同一对抗样本常常被不同架构的多种分类器错误分类，或者在不同的训练数据子集上训练。

- Shallow softmax regression models are also vulnerable to adversarial examples.

- 浅层 softmax 回归模型也容易受到对抗样本的影响。

- Training on adversarial examples can regularize the model-however, this was not practical at the time due to the need for expensive constrained optimization in the inner loop.

- 在对抗样本上训练可以对模型进行正则化——然而，由于在内循环中需要昂贵的约束优化，这在当时并不实用。

These results suggest that classifiers based on modern machine learning techniques, even those that obtain excellent performance on the test set, are not learning the true underlying concepts that determine the correct output label. Instead, these algorithms have built a Potemkin village that works well on naturally occuring data, but is exposed as a fake when one visits points in space that do not have high probability in the data distribution. This is particularly disappointing because a popular approach in computer vision is to use convolutional network features as a space where Euclidean distance approximates perceptual distance. This resemblance is clearly flawed if images that have an immeasurably small perceptual distance correspond to completely different classes in the network's representation.

这些结果表明，基于现代机器学习技术的分类器，即使在测试集上表现出色，也并没有学习到决定正确输出标签的真实基本概念。相反，这些算法构建了一个波腾金村，能够在自然发生的数据上表现良好，但当访问在数据分布中概率不高的空间点时，其虚假性就暴露无遗。这尤其令人失望，因为计算机视觉中的一种流行方法是使用卷积网络特征作为一个空间，在这个空间中，欧几里得距离近似于感知距离。如果具有不可测量的小感知距离的图像在网络的表示中对应于完全不同的类别，这种相似性显然是有缺陷的。

These results have often been interpreted as being a flaw in deep networks in particular, even though linear classifiers have the same problem. We regard the knowledge of this flaw as an opportunity to fix it. Indeed, Gu & Rigazio (2014) and Chalupka et al. (2014) have already begun the first steps toward designing models that resist adversarial perturbation, though no model has yet succesfully done so while maintaining state of the art accuracy on clean inputs.

这些结果常常被解释为深度网络的缺陷，尽管线性分类器也存在同样的问题。我们将了解这一缺陷视为修复它的机会。实际上，Gu 和 Rigazio (2014) 以及 Chalupka 等人 (2014) 已经开始迈出设计能够抵抗对抗扰动的模型的第一步，尽管尚未有模型在保持对干净输入的最先进准确性的同时成功做到这一点。

# 3 THE LINEAR EXPLANATION OF ADVERSARIAL EXAMPLES

# 3 对抗示例的线性解释

We start with explaining the existence of adversarial examples for linear models.

我们首先解释线性模型中对抗示例的存在。

In many problems, the precision of an individual input feature is limited. For example, digital images often use only 8 bits per pixel so they discard all information below 1/255 of the dynamic range. Because the precision of the features is limited, it is not rational for the classifier to respond differently to an input $\mathbf{x}$ than to an adversarial input $\widetilde{\mathbf{x}} = \mathbf{x} + \eta$ if every element of the perturbation $\eta$ is smaller than the precision of the features. Formally, for problems with well-separated classes, we expect the classifier to assign the same class to $\mathbf{x}$ and $\widetilde{\mathbf{x}}$ so long as $\| \eta\|_\infty < \epsilon$ , where $\epsilon$ is small enough to be discarded by the sensor or data storage apparatus associated with our problem.

在许多问题中，单个输入特征的精度是有限的。例如，数字图像通常每个像素仅使用 8 位，因此它们丢弃了动态范围下 1/255 以下的所有信息。由于特征的精度有限，因此如果扰动的每个元素 $\eta$ 小于特征的精度，分类器对输入 $\mathbf{x}$ 和对抗输入 $\widetilde{\mathbf{x}} = \mathbf{x} + \eta$ 的响应不应有差异。从形式上讲，对于类之间分离良好的问题，我们期望分类器将相同的类分配给 $\mathbf{x}$ 和 $\widetilde{\mathbf{x}}$ ，只要 $\| \eta\|_\infty < \epsilon$ ，其中 $\epsilon$ 足够小，以至于被与我们的问题相关的传感器或数据存储设备丢弃。

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

考虑权重向量 $\mathbf{w}$ 和对抗示例 $\widetilde{\mathbf{x}}$ 之间的点积:

$$\mathbf{w}^\top \widetilde{\mathbf{x}} = \mathbf{w}^\top \mathbf{x} + \mathbf{w}^\top \eta$$

The adversarial perturbation causes the activation to grow by $\mathbf{w}^\top \eta$ .We can maximize this increase subject to the max norm constraint on $\eta$ by assigning $\eta = \operatorname{sign}(\mathbf{w})$ . If $\mathbf{w}$ has $n$ dimensions and the average magnitude of an element of the weight vector is $m$ , then the activation will grow by $\epsilon m n$ . Since $\| \eta\|_\infty$ does not grow with the dimensionality of the problem but the change in activation caused by perturbation by $\eta$ 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. We can think of this as a sort of "accidental steganography," where a linear model is forced to attend exclusively to the signal that aligns most closely with its weights, even if multiple signals are present and other signals have much greater amplitude.

对抗扰动导致激活增加 $\mathbf{w}^\top \eta$ 。我们可以通过分配 $\eta = \operatorname{sign}(\mathbf{w})$ 来最大化这一增长，同时满足对 $\eta$ 的最大范数约束。如果 $\mathbf{w}$ 有 $n$ 个维度，并且权重向量的元素的平均幅度为 $m$ ，那么激活将增加 $\epsilon m n$ 。由于 $\| \eta\|_\infty$ 不随问题的维度增长，但由 $\eta$ 引起的扰动所导致的激活变化可以随 $n$ 线性增长，因此对于高维问题，我们可以对输入进行许多微小的变化，这些变化加起来形成对输出的一次大变化。我们可以将其视为一种"意外隐写术"，在这种情况下，线性模型被迫专注于与其权重最紧密对齐的信号，即使存在多个信号且其他信号的幅度更大。

This explanation shows that a simple linear model can have adversarial examples if its input has sufficient dimensionality. Previous explanations for adversarial examples invoked hypothesized properties of neural networks, such as their supposed highly non-linear nature. Our hypothesis based on linearity is simpler, and can also explain why softmax regression is vulnerable to adversarial examples.

该解释表明，如果输入具有足够的维度，简单线性模型可能会存在对抗样本。之前对对抗样本的解释引用了神经网络的假设属性，例如它们被认为具有高度非线性的特性。我们基于线性的假设更为简单，也可以解释为什么 softmax 回归对对抗样本易受攻击。

# 4 LINEAR PERTURBATION OF NON-LINEAR MODELS

# 4 非线性模型的线性扰动

The linear view of adversarial examples suggests a fast way of generating them. We hypothesize that neural networks are too linear to resist linear adversarial perturbation. LSTMs (Hochreiter & Schmidhuber, 1997), ReLUs (Jarrett et al. 2009, Glorot et al. 2011), and maxout networks (Good-fellow et al. 2013c) are all intentionally designed to behave in very linear ways, so that they are easier to optimize. More nonlinear models such as sigmoid networks are carefully tuned to spend most of their time in the non-saturating, more linear regime for the same reason. This linear behavior suggests that cheap, analytical perturbations of a linear model should also damage neural networks.

对对抗样本的线性视角暗示了一种快速生成它们的方法。我们假设神经网络过于线性，无法抵抗线性对抗扰动。LSTM(Hochreiter & Schmidhuber, 1997)、ReLU(Jarrett et al. 2009, Glorot et al. 2011) 和 maxout 网络 (Goodfellow et al. 2013c) 都是故意设计为以非常线性的方式运行，以便更容易优化。更非线性的模型，如 sigmoid 网络，经过精心调整，以便在非饱和的、更加线性的状态下花费大部分时间，原因相同。这种线性行为表明，线性模型的廉价解析扰动也应该对神经网络造成损害。

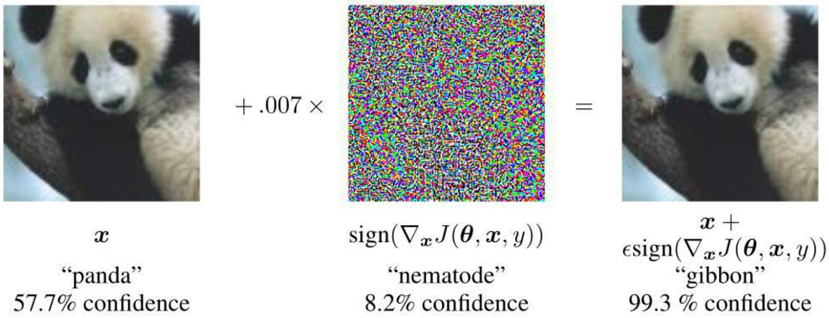

+ .007 ×
=
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} +$
$\epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure 1: A demonstration of fast adversarial example generation applied to GoogLeNet (Szegedy et al. 2014a) on ImageNet. By adding an imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input, we can change GoogLeNet' s classification of the image. Here our $\epsilon$ of .007 corresponds to the magnitude of the smallest bit of an 8 bit image encoding after GoogLeNet's conversion to real numbers.

图 1: 在 ImageNet 上应用于 GoogLeNet(Szegedy et al. 2014a) 的快速对抗样本生成演示。通过添加一个几乎不可察觉的小向量，其元素等于成本函数相对于输入的梯度元素的符号，我们可以改变 GoogLeNet 对图像的分类。这里我们的 $\epsilon$ 为 0.007，对应于 GoogLeNet 转换为实数后 8 位图像编码的最小位的大小。

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

设 $\theta$ 为模型的参数，$\mathbf{x}$ 为模型的输入，$y$ 为与 $\mathbf{x}$ 相关的目标 (对于具有目标的机器学习任务)，$J(\theta, \mathbf{x}, y)$ 为用于训练神经网络的成本。我们可以围绕当前的 $\theta$ 值线性化成本函数，从而获得一个最优的最大范数约束扰动。

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

We refer to this as the "fast gradient sign method" of generating adversarial examples. Note that the required gradient can be computed efficiently using backpropagation.

我们将其称为生成对抗样本的"快速梯度符号方法"。注意，所需的梯度可以通过反向传播高效计算。

We find that this method reliably causes a wide variety of models to misclassify their input. See Fig. 1 for a demonstration on ImageNet. We find that using $\epsilon = .25$ , we cause a shallow softmax classifier to have an error rate of 99.9% with an average confidence of 79.3% on the MNIST (?) test set [1] . In the same setting, a maxout network misclassifies 89.4% of our adversarial examples with an average confidence of 97.6% . Similarly, using $\epsilon = .1$ , we obtain an error rate of 87.15% and an average probability of 96.6% assigned to the incorrect labels when using a convolutional maxout network on a preprocessed version of the CIFAR-10 (Krizhevsky & Hinton, 2009) test set 2 Other simple methods of generating adversarial examples are possible. For example, we also found that rotating $\mathbf{x}$ by a small angle in the direction of the gradient reliably produces adversarial examples.

我们发现这种方法可靠地导致各种模型对其输入进行错误分类。请参见图 1 以了解在 ImageNet 上的演示。我们发现使用 $\epsilon = .25$ 时，我们使一个浅层 softmax 分类器在 MNIST (?) 测试集 [1] 上的错误率为 99.9% ，平均置信度为 79.3% 。在相同的设置中，一个 maxout 网络以平均置信度 97.6% 错误分类了 89.4% 的对抗样本。同样，使用 $\epsilon = .1$ 时，我们在对 CIFAR-10 (Krizhevsky & Hinton, 2009) 测试集的预处理版本上，获得了 87.15% 的错误率和分配给错误标签的平均概率 96.6% 。生成对抗样本的其他简单方法也是可能的。例如，我们还发现将 $\mathbf{x}$ 旋转一个小角度朝向梯度方向可靠地产生对抗样本。

The fact that these simple, cheap algorithms are able to generate misclassified examples serves as evidence in favor of our interpretation of adversarial examples as a result of linearity. The algorithms are also useful as a way of speeding up adversarial training or even just analysis of trained networks.

这些简单、廉价的算法能够生成错误分类样本的事实为我们将对抗样本解释为线性结果提供了证据。这些算法也有助于加速对抗训练，甚至只是对训练网络的分析。

# 5 ADVERSARIAL TRAINING OF LINEAR MODELS VERSUS WEIGHT DECAY

# 5 线性模型的对抗训练与权重衰减

Perhaps the simplest possible model we can consider is logistic regression. In this case, the fast gradient sign method is exact. We can use this case to gain some intuition for how adversarial examples are generated in a simple setting. See Fig. 2 for instructive images.

我们可以考虑的最简单的模型可能是逻辑回归。在这种情况下，快速梯度符号方法是精确的。我们可以利用这个案例来获得一些关于对抗样本如何在简单环境中生成的直觉。有关说明性图像，请参见图 2。

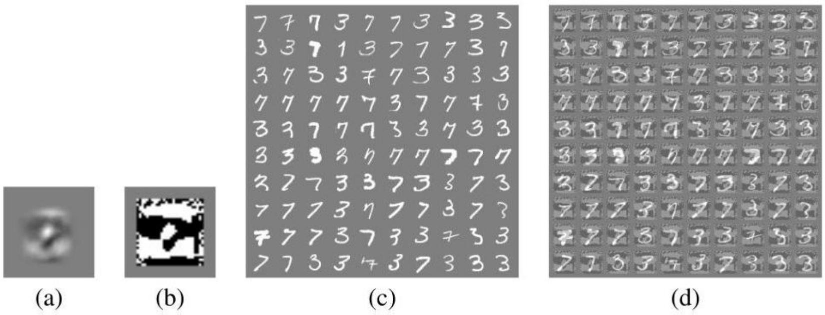
If we train a single model to recognize labels $y \in \{-1, 1\}$ with $P(y=1) = \sigma\left(\mathbf{w}^\top \mathbf{x} + b\right)$ where $\sigma(z)$ is the logistic sigmoid function, then training consists of gradient descent on

如果我们训练一个单一模型来识别标签 $y \in \{-1, 1\}$ ，使用 $P(y=1) = \sigma\left(\mathbf{w}^\top \mathbf{x} + b\right)$ ，其中 $\sigma(z)$ 是逻辑 sigmoid 函数，那么训练过程包括对

$$\mathbb{E}_{\mathbf{x}, y \sim p \text{ data}} \zeta\left(-y\left(\mathbf{w}^\top \mathbf{x} + b\right)\right)$$

where $\zeta(z) = \log(1 + \exp(z))$ is the softplus function. We can derive a simple analytical form for training on the worst-case adversarial perturbation of $\mathbf{x}$ rather than $\mathbf{x}$ itself, based on gradient sign

的梯度下降，其中 $\zeta(z) = \log(1 + \exp(z))$ 是 softplus 函数。我们可以基于梯度符号推导出一个简单的解析形式，用于在最坏情况下对 $\mathbf{x}$ 的对抗扰动进行训练，而不是直接对 $\mathbf{x}$ 进行训练。



[1] This is using MNIST pixel values in the interval $[0, 1]$ . MNIST data does contain values other than 0 or 1, but the images are essentially binary. Each pixel roughly encodes ”ink” or ”no ink”. This justifies expecting the classifier to be able to handle perturbations within a range of width 0.5, and indeed human observers can read such images without difficulty.

[1] 这是使用 MNIST 像素值在区间 $[0, 1]$ 内进行的。MNIST 数据确实包含除了 0 或 1 以外的值，但图像本质上是二元的。每个像素大致编码为“墨水”或“无墨水”。这证明了期望分类器能够处理宽度为 0.5 的扰动是合理的，实际上人类观察者可以毫不费力地读取这样的图像。

[2] See https://github.com/lisa-lab/pylearn2/tree/master/pylearn2/scripts/ papers/maxout, for the preprocessing code, which yields a standard deviation of roughly 0.5 .

[2] 有关预处理代码，请参见 https://github.com/lisa-lab/pylearn2/tree/master/pylearn2/scripts/papers/maxout，该代码产生的标准差大约为 0.5。

Figure 2: The fast gradient sign method applied to logistic regression (where it is not an approximation, but truly the most damaging adversarial example in the max norm box). a) The weights of a logistic regression model trained on MNIST. b) The sign of the weights of a logistic regression model trained on MNIST. This is the optimal perturbation. Even though the model has low capacity and is fit well, this perturbation is not readily recognizable to a human observer as having anything to do with the relationship between 3s and 7s. c) MNIST 3s and 7s. The logistic regression model has a 1.6% error rate on the 3 versus 7 discrimination task on these examples. d) Fast gradient sign adversarial examples for the logistic regression model with $\epsilon=.25$ . The logistic regression model has an error rate of 99% on these examples.

图 2: 快速梯度符号方法应用于逻辑回归 (在这里它不是近似，而是真正的在最大范数框中最具破坏性的对抗样本)。a) 在 MNIST 上训练的逻辑回归模型的权重。b) 在 MNIST 上训练的逻辑回归模型的权重符号。这是最佳扰动。尽管模型的容量较低且拟合良好，但这个扰动对于人类观察者而言并不容易识别出与 3 和 7 之间的关系有关。c) MNIST 中的 3 和 7。逻辑回归模型在这些示例的 3 与 7 区分任务上的 1.6% 错误率。d) 逻辑回归模型的快速梯度符号对抗样本，具有 $\epsilon=.25$ 。逻辑回归模型在这些示例上的错误率为 99% 。

perturbation. Note that the sign of the gradient is just $-\operatorname{sign}(\mathbf{w})$ , and that $\mathbf{w}^{\top} \operatorname{sign}(\mathbf{w})=\|\mathbf{w}\|_1$ . The adversarial version of logistic regression is therefore to minimize

扰动。请注意，梯度的符号仅为 $-\operatorname{sign}(\mathbf{w})$ ，并且 $\mathbf{w}^{\top} \operatorname{sign}(\mathbf{w})=\|\mathbf{w}\|_1$ 。因此，逻辑回归的对抗版本是最小化

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

This is somewhat similar to $L^1$ regularization. However, there are some important differences. Most significantly, the $L^1$ penalty is subtracted off the model’s activation during training, rather than added to the training cost. This means that the penalty can eventually start to disappear if the model learns to make confident enough predictions that $\zeta$ saturates. This is not guaranteed to happen-in the underfitting regime, adversarial training will simply worsen underfitting. We can thus view $L^1$ weight decay as being more "worst case" than adversarial training, because it fails to deactivate in the case of good margin.

这在某种程度上类似于 $L^1$ 正则化。然而，有一些重要的区别。最显著的是，$L^1$ 惩罚在训练过程中是从模型的激活中减去，而不是加到训练成本上。这意味着，如果模型学会做出足够自信的预测以至于 $\zeta$ 饱和，惩罚最终可能会开始消失。这并不是保证会发生的——在欠拟合的情况下，对抗训练只会加剧欠拟合。因此，我们可以将 $L^1$ 权重衰减视为比对抗训练更“最坏情况”，因为在良好边际的情况下，它未能停用。

If we move beyond logistic regression to multiclass softmax regression, $L^1$ weight decay becomes even more pessimistic, because it treats each of the softmax’s outputs as independently perturbable, when in fact it is usually not possible to find a single $\eta$ that aligns with all of the class’s weight vectors. Weight decay overestimates the damage achievable with perturbation even more in the case of a deep network with multiple hidden units. Because $L^1$ weight decay overestimates the amount of damage an adversary can do, it is necessary to use a smaller $L^1$ weight decay coefficient than the $\epsilon$ associated with the precision of our features. When training maxout networks on MNIST, we obtained good results using adversarial training with $\epsilon=.25$ . When applying $L^1$ weight decay to the first layer, we found that even a coefficient of .0025 was too large, and caused the model to get stuck with over 5% error on the training set. Smaller weight decay coefficients permitted succesful training but conferred no regularization benefit.

如果我们将逻辑回归扩展到多类软最大回归，$L^1$ 权重衰减变得更加悲观，因为它将每个软最大输出视为独立可扰动的，而实际上通常无法找到一个与所有类别的权重向量对齐的单一 $\eta$ 。在深度网络具有多个隐藏单元的情况下，权重衰减甚至更高估了扰动所能造成的损害。由于 $L^1$ 权重衰减高估了对手可以造成的损害，因此有必要使用比与我们特征精度相关的 $\epsilon$ 更小的 $L^1$ 权重衰减系数。在对 MNIST 上的 maxout 网络进行训练时，我们通过使用对抗训练与 $\epsilon=.25$ 获得了良好的结果。当将 $L^1$ 权重衰减应用于第一层时，我们发现即使是 0.0025 的系数也太大，导致模型在训练集上出现过 5% 错误。较小的权重衰减系数允许成功训练，但没有带来正则化的好处。

## 6 ADVERSARIAL TRAINING OF DEEP NETWORKS

## 6 深度网络的对抗训练

The criticism of deep networks as vulnerable to adversarial examples is somewhat misguided, because unlike shallow linear models, deep networks are at least able to represent functions that resist adversarial perturbation. The universal approximator theorem (Hornik et al. 1989) guarantees that a neural network with at least one hidden layer can represent any function to an arbitary degree of accuracy so long as its hidden layer is permitted to have enough units. Shallow linear models are not able to become constant near training points while also assigning different outputs to different training points.

对深度网络作为易受对抗样本攻击的批评有些误导，因为与浅层线性模型不同，深度网络至少能够表示抵抗对抗扰动的函数。通用逼近定理 (Hornik 等，1989) 保证了具有至少一个隐藏层的神经网络可以以任意精度表示任何函数，只要其隐藏层允许有足够的单元。浅层线性模型无法在训练点附近保持常数，同时又对不同的训练点分配不同的输出。

Of course, the universal approximator theorem does not say anything about whether a training algorithm will be able to discover a function with all of the desired properties. Obviously, standard supervised training does not specify that the chosen function be resistant to adversarial examples. This must be encoded in the training procedure somehow.

当然，通用逼近器定理并没有说明训练算法是否能够发现具有所有期望属性的函数。显然，标准的监督训练并没有规定所选择的函数必须对对抗样本具有抵抗力。这必须以某种方式编码在训练过程中。

Szegedy et al. 2014b) showed that by training on a mixture of adversarial and clean examples, a neural network could be regularized somewhat. Training on adversarial examples is somewhat different from other data augmentation schemes; usually, one augments the data with transformations such as translations that are expected to actually occur in the test set. This form of data augmentation instead uses inputs that are unlikely to occur naturally but that expose flaws in the ways that the model conceptualizes its decision function. At the time, this procedure was never demonstrated to improve beyond dropout on a state of the art benchmark. However, this was partially because it was difficult to experiment extensively with expensive adversarial examples based on L-BFGS.

Szegedy 等人 (2014b) 表明，通过在对抗样本和干净样本的混合上进行训练，神经网络可以在一定程度上进行正则化。对抗样本的训练与其他数据增强方案有所不同；通常，数据增强是通过诸如平移等预期在测试集中实际发生的变换来进行的。这种数据增强形式则使用不太可能自然发生的输入，但这些输入暴露了模型概念化其决策函数时的缺陷。当时，这一过程从未被证明能在最先进的基准上超越 dropout。然而，这在一定程度上是因为基于 L-BFGS 的昂贵对抗样本难以进行广泛实验。

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

我们发现，基于快速梯度符号法的对抗目标函数进行训练是一种有效的正则化方法:

$$\widetilde{J}(\theta, \mathbf{x}, y) = \alpha J(\theta, \mathbf{x}, y) + (1-\alpha) J(\theta, \mathbf{x} + \epsilon \operatorname{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y))).$$

In all of our experiments, we used $\alpha = 0.5$ . Other values may work better; our initial guess of this hyperparameter worked well enough that we did not feel the need to explore more. This approach means that we continually update our supply of adversarial examples, to make them resist the current version of the model. Using this approach to train a maxout network that was also regularized with dropout, we were able to reduce the error rate from 0.94% without adversarial training to 0.84% with adversarial training.

在我们所有的实验中，我们使用了 $\alpha = 0.5$ 。其他值可能效果更好；我们对这个超参数的初步猜测效果很好，以至于我们没有觉得需要进一步探索。这种方法意味着我们不断更新我们的对抗样本，以使其抵抗当前版本的模型。使用这种方法训练一个同时使用 dropout 进行正则化的 maxout 网络，我们能够将错误率从 0.94% (未进行对抗训练) 降低到 0.84% (进行对抗训练)。

We observed that we were not reaching zero error rate on adversarial examples on the training set. We fixed this problem by making two changes. First, we made the model larger, using 1600 units per layer rather than the 240 used by the original maxout network for this problem. Without adversarial training, this causes the model to overfit slightly, and get an error rate of 1.14% on the test set. With adversarial training, we found that the validation set error leveled off over time, and made very slow progress. The original maxout result uses early stopping, and terminates learning after the validation set error rate has not decreased for 100 epochs. We found that while the validation set error was very flat, the adversarial validation set error was not. We therefore used early stopping on the adversarial validation set error. Using this criterion to choose the number of epochs to train for, we then retrained on all 60,000 examples. Five different training runs using different seeds for the random number generators used to select minibatches of training examples, initialize model weights, and generate dropout masks result in four trials that each had an error rate of 0.77% on the test set and one trial that had an error rate of 0.83% . The average of 0.782% is the best result reported on the permutation invariant version of MNIST, though statistically indistinguishable from the result obtained by fine-tuning DBMs with dropout (Srivastava et al. 2014) at 0.79%.

我们观察到在训练集上的对抗样本中未能达到零错误率。我们通过进行两项更改来解决这个问题。首先，我们将模型的规模增大，每层使用 1600 个单元，而不是原始 maxout 网络为此问题使用的 240 个单元。在没有对抗训练的情况下，这导致模型略微过拟合，并在测试集上获得了 1.14% 的错误率。通过对抗训练，我们发现验证集的错误率随着时间的推移趋于平稳，并且进展非常缓慢。原始的 maxout 结果使用了早停法，在验证集错误率在 100 个周期内没有下降后终止学习。我们发现虽然验证集错误率非常平稳，但对抗验证集错误率并非如此。因此，我们在对抗验证集错误率上使用了早停法。使用这一标准选择训练的周期数后，我们在所有 60,000 个样本上进行了重新训练。五次不同的训练运行使用不同的随机数生成器种子来选择训练样本的小批量，初始化模型权重，并生成 dropout 掩码，结果产生了四次试验在测试集上均有 0.77% 的错误率，以及一次试验的错误率为 0.83% 。平均值 0.782% 是报告的在 MNIST 的置换不变版本上的最佳结果，尽管在统计上与通过 dropout 微调 DBM(Srivastava 等，2014) 获得的结果 0.79% 无显著差异。

The model also became somewhat resistant to adversarial examples. Recall that without adversarial training, this same kind of model had an error rate of 89.4% on adversarial examples based on the fast gradient sign method. With adversarial training, the error rate fell to 17.9%. Adversarial examples are transferable between the two models but with the adversarially trained model showing greater robustness. Adversarial examples generated via the original model yield an error rate of 19.6% on the adversarially trained model, while adversarial examples generated via the new model yield an error rate of 40.9% on the original model. When the adversarially trained model does misclassify an adversarial example, its predictions are unfortunately still highly confident. The average confidence on a misclassified example was 81.4% . We also found that the weights of the learned model changed significantly, with the weights of the adversarially trained model being significantly more localized and interpretable (see Fig. 3).

该模型在一定程度上对对抗性样本变得具有抵抗力。回想一下，在没有对抗训练的情况下，这种模型在基于快速梯度符号方法的对抗性样本上的错误率为 89.4% 。经过对抗训练后，错误率降至 17.9%。对抗性样本在两个模型之间是可转移的，但经过对抗训练的模型显示出更强的鲁棒性。通过原始模型生成的对抗性样本在经过对抗训练的模型上产生的错误率为 19.6% ，而通过新模型生成的对抗性样本在原始模型上产生的错误率为 40.9% 。当经过对抗训练的模型错误分类一个对抗性样本时，其预测结果不幸的是仍然非常自信。错误分类样本的平均置信度为 81.4% 。我们还发现，学习模型的权重发生了显著变化，经过对抗训练的模型的权重显著更局部化且更具可解释性 (见图 3)。

The adversarial training procedure can be seen as minimizing the worst case error when the data is perturbed by an adversary. That can be interpreted as learning to play an adversarial game, or as minimizing an upper bound on the expected cost over noisy samples with noise from $U(-\epsilon, \epsilon)$ added to the inputs. Adversarial training can also be seen as a form of active learning, where the model is able to request labels on new points. In this case the human labeler is replaced with a heuristic labeler that copies labels from nearby points.

对抗训练过程可以被视为在数据被对手扰动时最小化最坏情况错误。这可以解释为学习玩一场对抗游戏，或者最小化在带有来自 $U(-\epsilon, \epsilon)$ 的噪声添加到输入的噪声样本上的期望成本的上界。对抗训练也可以被视为一种主动学习的形式，其中模型能够请求新点的标签。在这种情况下，人类标注者被一个启发式标注者所替代，该标注者从附近的点复制标签。

We could also regularize the model to be insensitive to changes in its features that are smaller than the $\epsilon$ precision simply by training on all points within the $\epsilon$ max norm box, or sampling many points within this box. This corresponds to adding noise with max norm $\epsilon$ during training. However, noise with zero mean and zero covariance is very inefficient at preventing adversarial examples. The expected dot product between any reference vector and such a noise vector is zero. This means that in many cases the noise will have essentially no effect rather than yielding a more difficult input.

我们还可以对模型进行正则化，使其对小于 $\epsilon$ 精度的特征变化不敏感，方法是对 $\epsilon$ 最大范数框内的所有点进行训练，或在该框内采样多个点。这相当于在训练过程中添加具有最大范数 $\epsilon$ 的噪声。然而，均值为零且协方差为零的噪声在防止对抗样本方面非常低效。任何参考向量与这种噪声向量之间的预期点积为零。这意味着在许多情况下，噪声基本上不会产生影响，而不是产生更困难的输入。

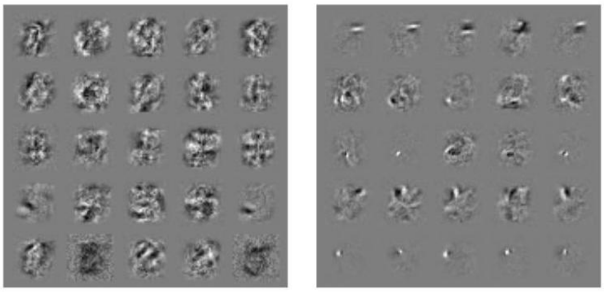


Figure 3: Weight visualizations of maxout networks trained on MNIST. Each row shows the filters for a single maxout unit. Left) Naively trained model. Right) Model with adversarial training.

图 3: 在 MNIST 上训练的 maxout 网络的权重可视化。每一行显示一个 maxout 单元的滤波器。左) 天真训练的模型。右) 具有对抗训练的模型。

In fact, in many cases the noise will actualy result in a lower objective function value. We can think of adversarial training as doing hard example mining among the set of noisy inputs, in order to train more efficiently by considering only those noisy points that strongly resist classification. As control experiments, we trained training a maxout network with noise based on randomly adding $\pm\epsilon$ to each pixel, or adding noise in $U(-\epsilon, \epsilon)$ to each pixel. These obtained an error rate of 86.2% with confidence 97.3% and an error rate of 90.4% with a confidence of 97.8% respectively on fast gradient sign adversarial examples.

实际上，在许多情况下，噪声会导致目标函数值降低。我们可以将对抗训练视为在噪声输入集中的困难样本挖掘，以便通过仅考虑那些强烈抵抗分类的噪声点来更有效地进行训练。作为对照实验，我们训练了一个基于随机向每个像素添加 $\pm\epsilon$ 的噪声的 maxout 网络，或向每个像素添加 $U(-\epsilon, \epsilon)$ 的噪声。这些在快速梯度符号对抗样本上分别获得了 86.2% 的错误率，置信度为 97.3% ，以及 90.4% 的错误率，置信度为 97.8% 。

Because the derivative of the sign function is zero or undefined everywhere, gradient descent on the adversarial objective function based on the fast gradient sign method does not allow the model to anticipate how the adversary will react to changes in the parameters. If we instead adversarial examples based on small rotations or addition of the scaled gradient, then the perturbation process is itself differentiable and the learning can take the reaction of the adversary into account. However, we did not find nearly as powerful of a regularizing result from this process, perhaps because these kinds of adversarial examples are not as difficult to solve.

由于符号函数的导数在任何地方都是零或未定义的，因此基于快速梯度符号方法的对抗目标函数上的梯度下降不允许模型预测对手如何对参数的变化做出反应。如果我们基于小旋转或缩放梯度的加法来生成对抗样本，那么扰动过程本身是可微的，学习可以考虑对手的反应。然而，我们没有发现这个过程产生的正则化效果有那么强大，可能是因为这类对抗样本并不那么难以解决。

One natural question is whether it is better to perturb the input or the hidden layers or both. Here the results are inconsistent. Szegedy et al. (2014b) reported that adversarial perturbations yield the best regularization when applied to the hidden layers. That result was obtained on a sigmoidal network. In our experiments with the fast gradient sign method, we find that networks with hidden units whose activations are unbounded simply respond by making their hidden unit activations very large, so it is usually better to just perturb the original input. On saturating models such as the Rust model we found that perturbation of the input performed comparably to perturbation of the hidden layers. Perturbations based on rotating the hidden layers solve the problem of unbounded activations growing to make additive perturbations smaller by comparison. We were able to succesfully train maxout networks with rotational perturbations of the hidden layers. However, this did not yield nearly as strong of a regularizing effect as additive perturbation of the input layer. Our view of adversarial training is that it is only clearly useful when the model has the capacity to learn to resist adversarial examples. This is only clearly the case when a universal approximator theorem applies. Because the last layer of a neural network, the linear-sigmoid or linear-softmax layer, is not a universal approximator of functions of the final hidden layer, this suggests that one is likely to encounter problems with underfitting when applying adversarial perturbations to the final hidden layer. We indeed found this effect. Our best results with training using perturbations of hidden layers never involved perturbations of the final hidden layer.

一个自然的问题是，扰动输入、隐藏层还是两者都更好。在这里，结果并不一致。Szegedy 等人 (2014b) 报告称，当对隐藏层施加对抗扰动时，能够产生最佳的正则化效果。该结果是在一个 sigmoid 网络上获得的。在我们使用快速梯度符号方法的实验中，我们发现具有无界激活的隐藏单元的网络，简单地通过使其隐藏单元激活变得非常大来响应，因此通常更好的是只扰动原始输入。在如 Rust 模型这样的饱和模型中，我们发现输入的扰动与隐藏层的扰动表现相当。基于旋转隐藏层的扰动解决了无界激活增长的问题，使得相较之下，加性扰动变得更小。我们成功地训练了具有隐藏层旋转扰动的 maxout 网络。然而，这并没有产生与输入层的加性扰动相近的强正则化效果。我们对对抗训练的看法是，当模型具备学习抵抗对抗样本的能力时，它才明显有用。这只有在通用逼近定理适用时才显而易见。由于神经网络的最后一层，即线性-sigmoid 或线性-softmax 层，并不是最终隐藏层函数的通用逼近器，这表明在对最终隐藏层施加对抗扰动时，可能会遇到欠拟合的问题。我们确实发现了这种效应。我们在使用隐藏层扰动进行训练时，最佳结果从未涉及对最终隐藏层的扰动。

# 7 DIFFERENT KINDS OF MODEL CAPACITY

# 7 不同类型的模型容量

One reason that the existence of adversarial examples can seem counter-intuitive is that most of us have poor intuitions for high dimensional spaces. We live in three dimensions, so we are not used to small effects in hundreds of dimensions adding up to create a large effect. There is another way that our intuitions serve us poorly. Many people think of models with low capacity as being unable to make many different confident predictions. This is not correct. Some models with low capacity do exhibit this behavior. For example shallow RBF networks with

对抗样本存在的一个原因看起来可能是违反直觉的，因为我们大多数人对高维空间的直觉较差。我们生活在三维空间中，因此不习惯于数百维中的小效应累加起来产生大的效果。我们的直觉还有另一种表现不佳的方式。许多人认为低容量模型无法做出许多不同的自信预测。这并不正确。一些低容量模型确实表现出这种行为。例如，浅层 RBF 网络具有

$$p\left(y=1 \mid \mathbf{x}\right)=\exp\left(\left(\mathbf{x}-\mu\right)^{\top}\beta\left(\mathbf{x}-\mu\right)\right)$$

are only able to confidently predict that the positive class is present in the vicinity of $\mu$ . Elsewhere, they default to predicting the class is absent, or have low-confidence predictions.

只能自信地预测正类在 $\mu$ 附近存在。在其他地方，它们默认预测该类不存在，或者预测的置信度较低。

RBF networks are naturally immune to adversarial examples, in the sense that they have low confidence when they are fooled. A shallow RBF network with no hidden layers gets an error rate of 55.4% on MNIST using adversarial examples generated with the fast gradient sign method and $\epsilon = .25$ . However, its confidence on mistaken examples is only 1.2% . Its average confidence on clean test examples is 60.6% . We can’t expect a model with such low capacity to get the right answer at all points of space, but it does correctly respond by reducing its confidence considerably on points it does not ”understand.”

RBF 网络在自然上对对抗样本具有免疫力，因为当它们被欺骗时信心较低。一个没有隐藏层的浅层 RBF 网络在使用快速梯度符号方法生成的对抗样本上，在 MNIST 数据集上的错误率为 55.4% 和 $\epsilon = .25$ 。然而，它对错误示例的信心仅为 1.2% 。它在干净测试示例上的平均信心为 60.6% 。我们不能期望一个如此低容量的模型在空间的所有点上都能得到正确答案，但它确实通过在它不“理解”的点上显著降低信心来做出正确反应。

RBF units are unfortunately not invariant to any significant transformations so they cannot generalize very well. We can view linear units and RBF units as different points on a precision-recall tradeoff curve. Linear units achieve high recall by responding to every input in a certain direction, but may have low precision due to responding too strongly in unfamiliar situations. RBF units achieve high precision by responding only to a specific point in space, but in doing so sacrifice recall. Motivated by this idea, we decided to explore a variety of models involving quadratic units, including deep RBF networks. We found this to be a difficult task-very model with sufficient quadratic inhibition to resist adversarial perturbation obtained high training set error when trained with SGD.

不幸的是，RBF 单元对任何显著的变换都不具有不变性，因此它们的泛化能力较差。我们可以将线性单元和 RBF 单元视为精确度-召回率权衡曲线上的不同点。线性单元通过对每个输入在某个方向上做出响应来实现高召回率，但由于在不熟悉的情况下反应过强，可能导致低精确度。RBF 单元通过仅对空间中的特定点做出响应来实现高精确度，但这样做牺牲了召回率。受到这一思想的启发，我们决定探索涉及二次单元的多种模型，包括深度 RBF 网络。我们发现这是一项困难的任务——任何具有足够二次抑制以抵御对抗性扰动的模型在使用 SGD 训练时都获得了高训练集误差。

# 8 WHY DO ADVERSARIAL EXAMPLES GENERALIZE?

# 8 为什么对抗性示例能够泛化？

An intriguing aspect of adversarial examples is that an example generated for one model is often misclassified by other models, even when they have different architecures or were trained on disjoint training sets. Moreover, when these different models misclassify an adversarial example, they often agree with each other on its class. Explanations based on extreme non-linearity and over-fitting cannot readily account for this behavior-why should multiple extremely non-linear model with excess capacity consistently label out-of-distribution points in the same way? This behavior is especially surprising from the view of the hypothesis that adversarial examples finely tile space like the rational numbers among the reals, because in this view adversarial examples are common but occur only at very precise locations.

对抗性示例的一个有趣方面是，为一个模型生成的示例往往会被其他模型错误分类，即使它们具有不同的架构或是在不相交的训练集上训练的。此外，当这些不同的模型错误分类一个对抗性示例时，它们通常在其类别上达成一致。基于极端非线性和过拟合的解释无法轻易解释这种行为——为什么多个具有过剩容量的极端非线性模型会始终以相同的方式标记分布外的点？从对抗性示例像有理数在实数中那样精细划分空间的假设来看，这种行为尤其令人惊讶，因为在这种观点下，对抗性示例是常见的，但仅在非常精确的位置出现。

Under the linear view, adversarial examples occur in broad subspaces. The direction $\eta$ need only have positive dot product with the gradient of the cost function, and $\epsilon$ need only be large enough. Fig. 4 demonstrates this phenomenon. By tracing out different values of $\epsilon$ we see that adversarial examples occur in contiguous regions of the 1-D subspace defined by the fast gradient sign method, not in fine pockets. This explains why adversarial examples are abundant and why an example misclassified by one classifier has a fairly high prior probability of being misclassified by another classifier.

在线性视角下，对抗样本发生在广泛的子空间中。方向 $\eta$ 只需与成本函数的梯度具有正的点积，而 $\epsilon$ 只需足够大。图 4 演示了这一现象。通过追踪不同的 $\epsilon$ 值，我们看到对抗样本出现在由快速梯度符号方法定义的 1 维子空间的连续区域中，而不是在细小的口袋中。这解释了为什么对抗样本如此丰富，以及为什么一个分类器错误分类的样本有相当高的先验概率被另一个分类器错误分类。

To explain why mutiple classifiers assign the same class to adversarial examples, we hypothesize that neural networks trained with current methodologies all resemble the linear classifier learned on the same training set. This reference classifier is able to learn approximately the same classification weights when trained on different subsets of the training set, simply because machine learning algorithms are able to generalize. The stability of the underlying classification weights in turn results in the stability of adversarial examples.

为了解释为什么多个分类器将同一类别分配给对抗样本，我们假设使用当前方法训练的神经网络都类似于在相同训练集上学习的线性分类器。这个参考分类器能够在不同的训练集子集上学习到大致相同的分类权重，仅仅因为机器学习算法能够进行泛化。基础分类权重的稳定性反过来导致了对抗样本的稳定性。

To test this hypothesis, we generated adversarial examples on a deep maxout network and classified these examples using a shallow softmax network and a shallow RBF network. On examples that were misclassified by the maxout network, the RBF network predicted the maxout network's class assignment only 16.0% of the time, while the softmax classifier predict the maxout network’ s class correctly 54.6% of the time. These numbers are largely driven by the differing error rate of the different models though. If we exclude our attention to cases where both models being compared make a mistake, then softmax regression predict’ s maxout’ s class 84.6% of the time, while the RBF network is able to predict maxout’ s class only 54.3% of the time. For comparison, the RBF network can predict softmax regression’ s class 53.6% of the time, so it does have a strong linear component to its own behavior. Our hypothesis does not explain all of the maxout network's mistakes or all of the mistakes that generalize across models, but clearly a significant proportion of them are consistent with linear behavior being a major cause of cross-model generalization.

为了验证这一假设，我们在深度最大化网络上生成了对抗样本，并使用浅层 softmax 网络和浅层 RBF 网络对这些样本进行分类。在被最大化网络错误分类的样本上，RBF 网络仅以 16.0% 的概率预测最大化网络的类别，而 softmax 分类器正确预测最大化网络的类别的概率为 54.6% 。这些数字在很大程度上是由不同模型的错误率差异驱动的。如果我们不考虑两个被比较模型都犯错的情况，那么 softmax 回归以 84.6% 的概率预测最大化的类别，而 RBF 网络仅以 54.3% 的概率预测最大化的类别。作为比较，RBF 网络以 53.6% 的概率预测 softmax 回归的类别，因此它确实在自身行为中具有强烈的线性成分。我们的假设并不能解释所有最大化网络的错误或所有跨模型的错误，但显然其中相当一部分与线性行为作为跨模型泛化的主要原因是一致的。

# 9 ALTERNATIVE HYPOTHESES

# 9 替代假设

We now consider and refute some alternative hypotheses for the existence of adversarial examples. First, one hypothesis is that generative training could provide more constraint on the training pro-

现在我们考虑并驳斥一些关于对抗样本存在的替代假设。首先，一个假设是生成训练可能对训练过程提供更多的约束。

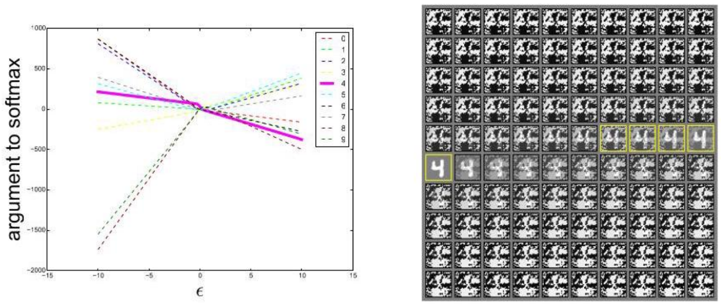


Figure 4: By tracing out different values of $\epsilon$ , we can see that adversarial examples occur reliably for almost any sufficiently large value of $\epsilon$ provided that we move in the correct direction. Correct classifications occur only on a thin manifold where $\mathbf{x}$ occurs in the data. Most of $\mathbb{R}^n$ consists of adversarial examples and rubbish class examples (see the appendix). This plot was made from a naively trained maxout network. Left) A plot showing the argument to the softmax layer for each of the 10 MNIST classes as we vary $\epsilon$ on a single input example. The correct class is 4 . We see that the unnormalized log probabilities for each class are conspicuously piecewise linear with $\epsilon$ and that the wrong classifications are stable across a wide region of $\epsilon$ values. Moreover, the predictions become very extreme as we increase $\epsilon$ enough to move into the regime of rubbish inputs. Right) The inputs used to generate the curve (upper left = negative $\epsilon$ , lower right = positive $\epsilon$ , yellow boxes indicate correctly classified inputs).

图 4: 通过追踪不同的 $\epsilon$ 值，我们可以看到对抗样本几乎在任何足够大的 $\epsilon$ 值下都可靠地出现，只要我们朝着正确的方向移动。正确的分类仅发生在 $\mathbf{x}$ 在数据中出现的一个薄流形上。大多数 $\mathbb{R}^n$ 由对抗样本和垃圾类样本组成 (见附录)。该图是从一个简单训练的 maxout 网络生成的。左) 一个图，显示了在对单个输入示例变化 $\epsilon$ 时，10 个 MNIST 类别的 softmax 层的参数。正确的类别是 4。我们看到每个类别的未归一化对数概率在 $\epsilon$ 上显著呈分段线性，并且错误分类在 $\epsilon$ 值的广泛区域内是稳定的。此外，随着我们增加 $\epsilon$ 以进入垃圾输入的范围，预测变得非常极端。右) 用于生成曲线的输入 (左上 = 负 $\epsilon$ ，右下 = 正 $\epsilon$ ，黄色框表示正确分类的输入)。

cess, or cause the model to learn what to distinguish ”real” from ”fake” data and be confident only on ”real” data. The MP-DBM (Goodfellow et al. 2013a) provides a good model to test this hypothesis. Its inference procedure gets good classification accuracy (an 0.88% error rate) on MNIST. This inference procedure is differentiable. Other generative models either have non-differentiable inference procedures, making it harder to compute adversarial examples, or require an additional non-generative discriminator model to get good classification accuracy on MNIST. In the case of the MP-DBM, we can be sure that the generative model itself is responding to adversarial examples, rather than the non-generative classifier model on top. We find that the model is vulnerable to adversarial examples. With an $\epsilon$ of 0.25, we find an error rate of 97.5% on adversarial examples generated from the MNIST test set. It remains possible that some other form of generative training could confer resistance, but clearly the mere fact of being generative is not alone sufficient.

这导致模型学习区分“真实”与“虚假”数据，并仅对“真实”数据保持信心。MP-DBM(Goodfellow et al. 2013a) 提供了一个良好的模型来测试这一假设。其推断过程在 MNIST 上获得了良好的分类准确率 (0.88% 的错误率)。这个推断过程是可微分的。其他生成模型要么具有不可微分的推断过程，使得计算对抗样本变得更加困难，要么需要额外的非生成判别模型以在 MNIST 上获得良好的分类准确率。在 MP-DBM 的情况下，我们可以确定生成模型本身对对抗样本做出反应，而不是上层的非生成分类器模型。我们发现该模型对对抗样本是脆弱的。在 $\epsilon$ 为 0.25 的情况下，我们发现从 MNIST 测试集生成的对抗样本的错误率为 97.5% 。仍然有可能其他形式的生成训练能够提供抵抗力，但显然，仅仅是生成模型并不足够。

Another hypothesis about why adversarial examples exist is that individual models have strange quirks but averaging over many models can cause adversarial examples to wash out. To test this hypothesis, we trained an ensemble of twelve maxout networks on MNIST. Each network was trained using a different seed for the random number generator used to initialize the weights, generate dropout masks, and select minibatches of data for stochastic gradient descent. The ensemble gets an error rate of 91.1% on adversarial examples designed to perturb the entire ensemble with $\epsilon = .25$ . If we instead use adversarial examples designed to perturb only one member of the ensemble, the error rate falls to 87.9% . Ensembling provides only limited resistance to adversarial perturbation.

关于对抗样本存在的另一个假设是，个别模型具有奇怪的特性，但对多个模型进行平均可能导致对抗样本被稀释。为了测试这一假设，我们在 MNIST 上训练了一个由十二个 maxout 网络组成的集成。每个网络使用不同的随机数生成器种子进行训练，以初始化权重、生成 dropout 掩码，并选择用于随机梯

度下降的小批量数据。该集成在设计用于扰动整个集成的对抗样本上获得了 91.1% 的错误率。如果我们改为使用设计用于仅扰动集成中一个成员的对抗样本，错误率降至 87.9% 。集成对对抗扰动仅提供有限的抵抗力。

# 10 SUMMARY AND DISCUSSION

# 10 总结与讨论

As a summary, this paper has made the following observations:

总结而言，本文做出了以下观察:

- 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.
- 对抗样本可以被解释为高维点积的一个特性。它们是模型过于线性而非过于非线性的结果。
- 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.
- 对抗样本在不同模型之间的泛化可以解释为对抗扰动与模型权重向量高度对齐的结果，以及不同模型在训练以执行相同任务时学习到相似的函数。
- The direction of perturbation, rather than the specific point in space, matters most. Space is not full of pockets of adversarial examples that finely tile the reals like the rational numbers.
- 扰动的方向，而非空间中的具体点，最为重要。空间并不是充满了像有理数那样精细覆盖实数的对抗样本的口袋。
- Because it is the direction that matters most, adversarial perturbations generalize across different clean examples.
- 由于方向最为重要，对抗扰动在不同的干净样本之间具有泛化性。
- We have introduced a family of fast methods for generating adversarial examples.
- 我们引入了一系列快速生成对抗样本的方法。
- We have demonstrated that adversarial training can result in regularization; even further regularization than dropout.
- 我们已经证明，对抗训练可以导致正则化；甚至比 dropout 更进一步的正则化。
- We have run control experiments that failed to reproduce this effect with simpler but less efficient regularizers including $L^1$ weight decay and adding noise.
- 我们进行了对照实验，未能用更简单但效率较低的正则化方法重现这一效果，包括 $L^1$ 权重衰减和添加噪声。
- Models that are easy to optimize are easy to perturb.
- 容易优化的模型也容易受到扰动。
- 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.
- 线性模型缺乏抵抗对抗扰动的能力；只有具有隐藏层的结构 (适用通用逼近定理) 应被训练以抵抗对抗扰动。
- RBF networks are resistant to adversarial examples.

- RBF 网络对对抗样本具有抵抗能力。
- Models trained to model the input distribution are not resistant to adversarial examples.
- 训练用于建模输入分布的模型对对抗样本并不具有抵抗力。
- Ensembles are not resistant to adversarial examples.
- 集成方法对对抗样本并不具有抵抗力。

Some further observations concerning rubbish class examples are presented in the appendix:
关于垃圾类样本的一些进一步观察在附录中提出:

- Rubbish class examples are ubiquitous and easily generated.
- 垃圾类样本无处不在且容易生成。
- Shallow linear models are not resistant to rubbish class examples.
- 浅层线性模型对垃圾类样本并不具有抵抗力。
- RBF networks are resistant to rubbish class examples.
- RBF 网络对垃圾类样本具有抵抗力。

Gradient-based optimization is the workhorse of modern AI. Using a network that has been designed to be sufficiently linear-whether it is a ReLU or maxout network, an LSTM, or a sigmoid network that has been carefully configured not to saturate too much- we are able to fit most problems we care about, at least on the training set. The existence of adversarial examples suggests that being able to explain the training data or even being able to correctly label the test data does not imply that our models truly understand the tasks we have asked them to perform. Instead, their linear responses are overly confident at points that do not occur in the data distribution, and these confident predictions are often highly incorrect. This work has shown we can partially correct for this problem by explicitly identifying problematic points and correcting the model at each of these points. However, one may also conclude that the model families we use are intrinsically flawed. Ease of optimization has come at the cost of models that are easily misled. This motivates the development of optimization procedures that are able to train models whose behavior is more locally stable.
基于梯度的优化是现代人工智能的主要驱动力。使用设计为足够线性的网络——无论是 ReLU 或 maxout 网络、LSTM，还是经过精心配置以避免过度饱和的 sigmoid 网络——我们能够拟合大多数我们关心的问题，至少在训练集上。对抗样本的存在表明，能够解释训练数据或甚至能够正确标记测试数据并不意味着我们的模型真正理解我们要求它们执行的任务。相反，它们的线性响应在数据分布中未出现的点上过于自信，而这些自信的预测往往是高度错误的。这项工作表明，我们可以通过明确识别问题点并在每个这些点上纠正模型来部分解决这个问题。然而，也可以得出结论，我们使用的模型家族本质上是有缺陷的。优化的便利性是以容易被误导的模型为代价的。这激励了能够训练行为更局部稳定的模型的优化程序的发展。

# ACKNOWLEDGMENTS

# 致谢

We would like to thank Geoffrey Hinton and Ilya Sutskever for helpful discussions. We would also like to thank Jeff Dean, Greg Corrado, and Oriol Vinyals for their feedback on drafts of this article. We would like to thank the developers of Theano(Bergstra et al., 2010, Bastien et al., 2012), Pylearn2(Goodfellow et al., 2013b), and DistBelief (Dean et al., 2012).
我们要感谢 Geoffrey Hinton 和 Ilya Sutskever 的有益讨论。我们还要感谢 Jeff Dean、Greg Corrado 和 Oriol Vinyals 对本文草稿的反馈。我们要感谢 Theano(Bergstra 等，2010；Bastien 等，2012)、Pylearn2(Goodfellow 等，2013b) 和 DistBelief(Dean 等，2012) 的开发者。

# REFERENCES

# 参考文献

Bastien, Frédéric, Lamblin, Pascal, Pascanu, Razvan, Bergstra, James, Goodfellow, Ian J., Bergeron, Arnaud, Bouchard, Nicolas, and Bengio, Yoshua. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

Bergstra, James, Breuleux, Olivier, Bastien, Frédéric, Lamblin, Pascal, Pascanu, Razvan, Razvan, Desjardins, Guillaume, Turian, Joseph, Warde-Farley, David, and Bengio, Yoshua. Theano: a CPU and GPU math expression compiler. In Proceedings of the Python for Scientific Computing Conference (SciPy), June 2010. Oral Presentation.

Chalupka, K., Perona, P., and Eberhardt, F. Visual Causal Feature Learning. ArXiv e-prints, December 2014.

Dean, Jeffrey, Corrado, Greg S., Monga, Rajat, Chen, Kai, Devin, Matthieu, Le, Quoc V., Mao, Mark Z., Ranzato, MarcAurelio, Senior, Andrew, Tucker, Paul, Yang, Ke, and Ng, Andrew Y. Large scale distributed deep networks. In NIPS, 2012.

Deng, Jia, Dong, Wei, Socher, Richard, jia Li, Li, Li, Kai, and Fei-fei, Li. Imagenet: A large-scale hierarchical image database. In In CVPR, 2009.

Glorot, Xavier, Bordes, Antoine, and Bengio, Yoshua. Deep sparse rectifier neural networks. In JMLR W&CP: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011), April 2011.

Goodfellow, Ian J., Mirza, Mehdi, Courville, Aaron, and Bengio, Yoshua. Multi-prediction deep Boltzmann machines. In Neural Information Processing Systems, December 2013a.

Goodfellow, Ian J., Warde-Farley, David, Lamblin, Pascal, Dumoulin, Vincent, Mirza, Mehdi, Pascanu, Razvan, Bergstra, James, Bastien, Frédéric, and Bengio, Yoshua. Pylearn2: a machine learning research library. arXiv preprint arXiv:1308.4214, 2013b.

Goodfellow, Ian J., Warde-Farley, David, Mirza, Mehdi, Courville, Aaron, and Bengio, Yoshua. Maxout networks. In Dasgupta, Sanjoy and McAllester, David (eds.), International Conference on Machine Learning, pp. 1319-1327, 2013c.

Gu, Shixiang and Rigazio, Luca. Towards deep neural network architectures robust to adversarial examples. In NIPS Workshop on Deep Learning and Representation Learning, 2014.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. Neural Computation, 9(8):1735-1780, 1997.

Hornik, Kurt, Stinchcombe, Maxwell, and White, Halbert. Multilayer feedforward networks are universal approximators. Neural Networks, 2:359-366, 1989.

Jarrett, Kevin, Kavukcuoglu, Koray, Ranzato, Marc'Aurelio, and LeCun, Yann. What is the best multi-stage architecture for object recognition? In Proc. International Conference on Computer Vision (ICCV'09), pp. 2146-2153. IEEE, 2009.

Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Nguyen, A., Yosinski, J., and Clune, J. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. ArXiv e-prints, December 2014.

Rust, Nicole, Schwartz, Odelia, Movshon, J. Anthony, and Simoncelli, Eero. Spatiotemporal elements of macaque V1 receptive fields. Neuron, 46(6):945-956, 2005.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15 (1):1929-1958, 2014.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Du-mitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. Technical report, arXiv preprint arXiv:1409.4842, 2014a.

Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian J., and Fergus, Rob. Intriguing properties of neural networks. ICLR, abs/1312.6199, 2014b. URL http: //arxiv.org/abs/1312.6199

# A RUBBISH CLASS EXAMPLES

# 垃圾类示例

A concept related to adversarial examples is the concept of examples drawn from a "rubbish class." These examples are degenerate inputs that a human would classify as not belonging to any of the categories in the training set. If we call these classes in the training set "the positive classes," then we want to be careful to avoid false positives on rubbish inputs-i.e., we do not want to classify a degenerate input as being something real. In the case of separate binary classifiers for each class, we want all classes output near zero probability of the class being present, and in the case of a multinoulli distribution over only the positive classes, we would prefer that the classifier output a high-entropy (nearly uniform) distribution over the classes. The traditional approach to reducing vulnerability to rubbish inputs is to introduce an extra, constant output to the model representing the rubbish class (?). Nguyen et al. 2014 recently re-popularized the concept of the rubbish class in the context of computer vision under the name fooling images. As with adversarial examples, there has been a misconception that rubbish class false positives are hard to find, and that they are primarily a problem faced by deep networks.

与对抗性示例相关的一个概念是“垃圾类”中提取的示例。这些示例是退化输入，人类会将其分类为不属于训练集中任何类别的输入。如果我们将训练集中的这些类别称为“正类”，那么我们需要小心避免在垃圾输入上出现假阳性——即，我们不希望将退化输入分类为真实的东西。在每个类别都有单独的二元分类器的情况下，我们希望所有类别的输出概率接近于零，而在仅对正类进行多项分布的情况下，我们更希望分类器输出对类别的高熵 (几乎均匀) 分布。传统上减少对垃圾输入脆弱性的方法是向模型引入一个额外的、恒定的输出，表示垃圾类 (？)。Nguyen 等人 (2014) 最近在计算机视觉的背景下重新普及了垃圾类的概念，称之为欺骗图像。与对抗性示例一样，存在一种误解，即垃圾类的假阳性很难发现，并且它们主要是深度网络面临的问题。

Our explanation of adversarial examples as the result of linearity and high dimensional spaces also applies to analyzing the behavior of the model on rubbish class examples. Linear models produce more extreme predictions at points that are far from the training data than at points that are near the training data. In order to find high confidence rubbish false positives for such a model, we need only generate a point that is far from the data, with larger norms yielding more confidence. RBF networks, which are not able to confidently predict the presence of any class far from the training data, are not fooled by this phenomenon.

我们对对抗样本的解释作为线性和高维空间的结果同样适用于分析模型在垃圾类样本上的行为。线性模型在远离训练数据的点上产生的预测比在接近训练数据的点上更为极端。为了找到这种模型的高置信度垃圾假阳性，我们只需生成一个远离数据的点，较大的范数会带来更高的置信度。RBF 网络无法自信地预测任何远离训练数据的类别，因此不会被这一现象所欺骗。

We generated 10,000 samples from $\mathcal{N}(0, \mathbf{I}_{784})$ and fed them into various classifiers on the MNIST dataset. In this context, we consider assigning a probability greater than 0.5 to any class to be an error. A naively trained maxout network with a softmax layer on top had an error rate of 98.35% on Gaussian rubbish examples with an average confidence of 92.8% on mistakes. Changing the top layer to independent sigmoids dropped the error rate to 68% with an average confidence on mistakes of 87.9% . On CIFAR-10, using 1,000 samples from $\mathcal{N}(0, \mathbf{I}_{3072})$ , a convolutional maxout net obtains an error rate of 93.4% , with an average confidence of 84.4% .

我们从 $\mathcal{N}(0, \mathbf{I}_{784})$ 生成了 10,000 个样本，并将其输入到 MNIST 数据集上的各种分类器中。在这种情况下，我们认为将概率大于 0.5 分配给任何类别都是错误的。一个简单训练的 maxout 网络在顶部有一个 softmax 层，在高斯垃圾样本上的错误率为 98.35% ，在错误上的平均置信度为 92.8% 。将顶层更改为独立的 sigmoid 后，错误率降至 68% ，在错误上的平均置信度为 87.9% 。在 CIFAR-10 上，使用来自 $\mathcal{N}(0, \mathbf{I}_{3072})$ 的 1,000 个样本，一个卷积 maxout 网络的错误率为 93.4% ，平均置信度为 84.4% 。

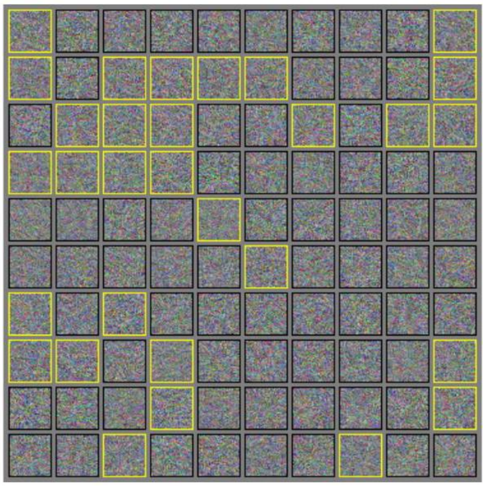


Figure 5: Randomly generated fooling images for a convolutional network trained on CIFAR- 10. These examples were generated by drawing a sample from an isotropic Gaussian, then taking a gradient sign step in the direction that increases the probability of the ”airplane” class. Yellow boxes indicate samples that successfully fool the model into believing an airplane is present with at least 50% confidence. ”Airplane” is the hardest class to construct fooling images for on CIFAR-10, so this figure represents the worst case in terms of success rate.

图 5: 为在 CIFAR-10 上训练的卷积网络随机生成的欺骗图像。这些示例是通过从各向同性高斯分布中抽取样本，然后在增加“飞机”类别概率的方向上进行梯度符号步长生成的。黄色框表示成功欺骗模型，使其相信存在飞机的样本，置信度至少为 50% 。“飞机”是 CIFAR-10 上构造欺骗图像最困难的类别，因此该图代表了成功率的最坏情况。

These experiments suggest that the optimization algorithms employed by Nguyen et al. (2014) are overkill (or perhaps only needed on ImageNet), and that the rich geometric structure in their fooling images are due to the priors encoded in their search procedures, rather than those structures being uniquely able to cause false positives.

这些实验表明，Nguyen 等人 (2014) 所采用的优化算法是过度的 (或者可能仅在 ImageNet 上需要)，而他们的欺骗图像中丰富的几何结构是由于其搜索过程中的先验编码，而不是这些结构能够独特地导致假阳性。

Though Nguyen et al. (2014) focused their attention on deep networks, shallow linear models have the same problem. A softmax regression model has an error rate of 59.8% on the rubbish examples, with an average confidence on mistakes of 70.8% . If we use instead an RBF network, which does not behave like a linear function, we find an error rate of 0% . Note that when the error rate is zero the average confidence on a mistake is undefined.

尽管 Nguyen 等人 (2014) 将注意力集中在深度网络上，但浅层线性模型也存在同样的问题。一个 softmax 回归模型在垃圾示例上的错误率为 59.8% ，在错误上的平均置信度为 70.8% 。如果我们使用一个 RBF 网络，它的行为并不像线性函数，我们发现错误率为 0% 。请注意，当错误率为零时，错误的平均置信度是未定义的。

Nguyen et al. (2014) focused on the problem of generating fooling images for a specific class, which is a harder problem than simply finding points that the network confidently classifies as belonging to any one class despite being defective. The above methods on MNIST and CIFAR-10 tend to have a very skewed distribution over classes. On MNIST, 45.3% of a naively trained maxout network’ s false positives were classified as 5 s , and none were classified as 8s. Likewise, on CIFAR-10, 49.7% of the convolutional network’s false positives were classified as frogs, and none were classified as airplanes, automobiles, horses, ships, or trucks.

Nguyen 等人 (2014) 关注于为特定类别生成欺骗图像的问题，这比简单地找到网络自信地将其分类为任何一个类别的缺陷点要困难得多。在 MNIST 和 CIFAR-10 上，上述方法往往在类别上具有非常偏斜的分布。在 MNIST 上，天真训练的 maxout 网络的假阳性中 45.3% 被分类为 5 s ，而没有被分类为 8。同样，在 CIFAR-10 上，卷积网络的假阳性中有 49.7% 被分类为青蛙，而没有被分类为飞机、汽车、马、船或卡车。

To solve the problem introduced by Nguyen et al. (2014) of generating a fooling image for a particular class, we propose adding $\epsilon \nabla_{\mathbf{x}} p(y=i \mid \mathbf{x})$ to a Gaussian sample $\mathbf{x}$ as a fast method of generating a fooling image classified as class $i$ . If we repeat this sampling process until it succeeds, we a randomized algorithm with variable runtime. On CIFAR-10, we found that one sampling step had a 100% success rate for frogs and trucks, and the hardest class was airplanes, with a success rate of 24.7% per sampling step. Averaged over all ten classes, the method has an average per-step success rate of 75.3% . We can thus generate any desired class with a handful of samples and no special priors, rather than tens of thousands of generations of evolution. To confirm that the resulting examples are indeed fooling images, and not images of real classes rendered by the gradient sign method, see Fig. 5 The success rate of this method in terms of generating members of class $i$ may degrade for datasets with more classes, since the risk of inadvertently increasing the activation of a different class $j$ increases in that case. We found that we were able to train a maxout network to have a zero percent error rate on Gaussian rubbish examples (it was still vulnerable to rubbish examples generated by applying a fast gradient sign step to a Gaussian sample) with no negative impact on its ability to classify clean examples. Unfortunately, unlike training on adversarial examples, this did not result in any significant reduction of the model's test set error rate.

为了解决 Nguyen 等人 (2014) 提出的为特定类别生成欺骗图像的问题，我们建议将 $\epsilon \nabla_{\mathbf{x}} p(y = i \mid \mathbf{x})$ 添加到高斯样本 $\mathbf{x}$ 中，作为生成被分类为类别 $i$ 的欺骗图像的快速方法。如果我们重复这个采样过程直到成功，我们就得到了一个具有可变运行时间的随机算法。在 CIFAR-10 数据集上，我们发现一次采样步骤对青蛙和卡车的成功率为 100% ，而最难的类别是飞机，其每次采样步骤的成功率为 24.7% 。在所有十个类别上平均，该方法的每步成功率为 75.3% 。因此，我们可以通过少量样本而无需特殊先验生成任何所需类别，而不是进行数万个进化生成。为了确认生成的示例确实是欺骗图像，而不是通过梯度符号方法渲染的真实类别图像，请参见图 5。就生成类别 $i$ 的成员而言，该方法的成功率可能会因类别更多的数据集而降低，因为在这种情况下，无意中增加不同类别 $j$ 的激活的风险增加。我们发现，我们能够训练一个 maxout 网络，使其在高斯垃圾示例上具有零错误率 (它仍然容易受到通过对高斯样本应用快速梯度符号步骤生成的垃圾示例的影响)，而不会对其分类干净示例的能力产生负面影响。不幸的是，与在对抗示例上训练不同，这并没有显著降低模型在测试集上的错误率。

In conclusion, it appears that a randomly selected input to deep or shallow models built from linear parts is overwhelmingly likely to be processed incorrectly, and that these models only behave reasonably on a very thin manifold encompassing the training data.

总之，随机选择的输入对于由线性部分构建的深度或浅层模型极有可能被错误处理，并且这些模型仅在一个非常狭窄的流形上合理地表现，该流形包含训练数据。