解释和利用对抗样本

Ian J. Goodfellow, Jonathon Shlens & Christian Szegedy
Google Inc., Mountain View, CA
{goodfellow,shlens,szegedy}@google.com

摘要

包括神经网络在内的数种机器学习模型一致地将对抗样本——也就是将微小的人为设计的扰动叠加到数据集中的样例所得到的的输入——错误分类，这种样本会导致模型以很高的置信度给出错误的输出。早前对这一现象的解释集中关注非线性以及过拟合。与之相对，我们认为神经网络对对抗样本的脆弱性主要源于它们的线性本质。这一解释可被新的量化结果佐证，我们同时对这些结果最有趣的事实，也就是对抗样本跨架构和训练集的泛化性，给出了首个解释。此外，这一观点给出了一种简单快速的对抗样本生成方法。通过使用这一方法获得用于对抗训练的样本，我们在 MNIST 数据集上减小了 maxout 网络的测试集误差。

1 引言

Szegedy 等人（2014b）发现了一个有趣的现象：包括最先进的神经网络在内的几种机器学习模型对于对抗样本非常脆弱。也就是说，这些机器学习模型对样本进行了错误分类，这些样本与从数据分布中得到的被正确分类的示例仅略有不同。在许多情况下，在训练数据的不同子集上训练的具有不同架构的各种模型会将同一对抗样本误分类。这表明对抗样本暴露了我们训练算法在基础上的盲点。

这些对抗样本的成因是一个谜，而推测性解释认为，这是由于深度神经网络的极端非线性所致，可能还有模型平均不足和纯监督学习问题的正则化不足的原因。我们证明这些推测性假设是不必要的。高维空间中的线性行为足以形成对抗样本。这种观点使我们能够设计一种快速生成对抗样本的方法，从而使对抗性训练切实可行。我们证明，对抗性训练可以提供比单独使用 dropout（Srivastava 等，2014）更多的正则化收益。通用的正则化策略（例如 dropout，预训练和模型平均）并不能显着降低模型对于对抗样本的脆弱性，但改用非线性模型族（如 RBF 网络）可以做到这一点。

我们的解释表明，利用线性加速训练与使用非线性效应来抵抗对抗扰动是一对根本矛盾。从长远来看，通过设计可以成功训练更多非线性模型的更强大的优化方法，可以避免这种折衷。

2 相关工作

Szegedy 等人（2014b）展示了神经网络和相关模型的多种有趣特性。与本文最为相关的那些包括：

- 可以可靠地对盒约束 L-BFGS 找到对抗样本。
- 在某些数据集上，例如 ImageNet（Deng 等，2009），对抗样本与原始样例非常接近，以至于人眼无法区分这些差异。
- 相同的对抗样本经常会被不同架构或在训练数据的不同子集上训练的各种分类器错误分类。

- 浅 softmax 回归模型也容易受到对抗样本的影响。
- 用对抗样本训练可以使模型规则化，但是，由于需要在内部循环中进行复杂的约束优化，因此这在当时尚不实用。

这些结果表明，基于现代机器学习技术的分类器，甚至是那些在测试集上性能出色的分类器，也不能学习到确定正确输出标签所需的真实内在概念。这些分类器只是浮于表面，它们可以很好地处理自然出现的数据，但是当人们选取样本空间中出现可能性不高的点作为输入时，它们就会无所适从。这尤其令人失望，因为计算机视觉中一种流行的方法是将卷积网络特征空间视作欧氏距离近似于感知距离的空间。如果感知距离很小的图像对应于网络表示中的完全不同的类别，则这种方法显然是有缺陷的。

尽管线性分类器具有相同的问题，但过去通常将这些结果解释为深度网络特有的缺陷。我们认为对这一缺陷的了解是修复它的机会。实际上，Gu 与 Rigazio（2014）和 Chalupka 等人（2014 年）已经开始着手设计抵抗对抗扰动的模型，尽管目前还没有任何模型能在成功地做到这一点的同时保持对干净输入的最优准确性。

3 对抗样本的线性解释

我们从解释线性模型存在对抗样本入手。

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占 8 位，因此它们会丢弃低于动态范围 1/255 的所有信息。因为特征的精度是有限的，所以如果对抗样本 $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\eta}$ 的扰动 $\boldsymbol{\eta}$ 的每个分量都小于特征精度，则分类器对输入 $\boldsymbol{x}$ 的响应不应当与 $\tilde{\boldsymbol{x}}$不同。形式上，对于类别分离良好的问题，$在||\boldsymbol{\eta}||_\infty < E$ 的条件下，我们期望分类器为 $\boldsymbol{x}$ 和 $\tilde{\boldsymbol{x}}$分配相同的类别，其中 E 足够小因而会被与我们的问题相关的传感器或数据存储设备丢弃。 考虑权值向量 $\boldsymbol{w}$ 和对抗样本 $\tilde{\boldsymbol{x}}$的点积：

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

对抗扰动使激活值增加 $\boldsymbol{w}^T\boldsymbol{\eta}$。通过赋值 $\eta = \text{sign}(\boldsymbol{w})$，我们可以在 η 受到最大范数约束的条件下最大化此增加值。如果 $\boldsymbol{w}$ 为 n 维向量，并且权值向量的元素的平均大小为 m，则激活值增长量为 Emn。由于$||\eta||_\infty$并不随问题维数的增长而增长，但是由扰动 η 引起的激活值变化量可以随 n 线性增长，因此对于高维问题，我们可以在输入上叠加无穷小的变化量，这些小变化累加起来将引起输出的巨大变化。我们可以将其视为一种“偶然隐写术”，也就是说线性模型被迫专门处理最接近其权值向量的信号，即使存在多个信号且其他信号的幅度更大。

这种解释表明，如果输入具有足够的维数，则简单的线性模型也可能存在对抗样本。先前对于对抗样本的解释引用了神经网络的假想性质，例如高度非线性性质。我们基于线性的假设更简单，同时还能解释为什么 softmax 回归容易受到对抗样本的影响。

4 非线性模型的线性扰动

对抗样本的线性理论提出了一种快速生成对抗样本的方法。我们假设神经网络是因为线性太强而无法抵抗线性对抗扰动。LSTM（Hochreiter 与 Schmidhuber，1997），ReLU（Jarrett 等，2009；Glorot 等，2011）和 maxout 网络（Good-other 等，2013c）都被故意设计成高度线性模型，以使它们更易于优化。出于相同的原因，诸如 Sigmoid 网络之类的非线性模型需要仔细调整，以将其大部分时间工作在非饱和区（也就是线性更强的状态）。这些线性行为表明，针对线性模型的廉价、解析性扰动也会妨害神经网络的工作。

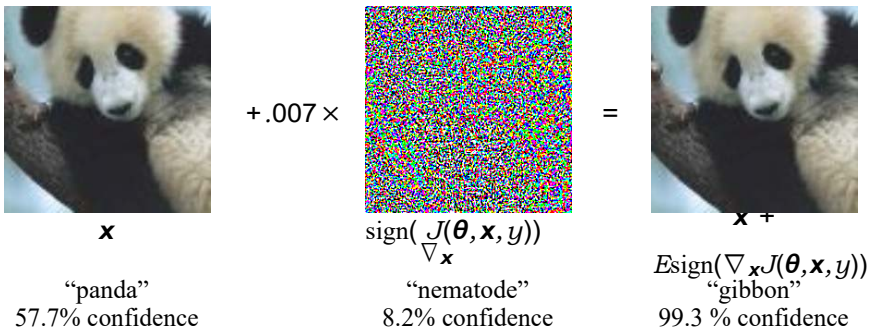


图 1：展示在 ImageNet 上将快速对抗样本生成算法应用于 GoogLeNet（Szegedy 等人，2014a）的效果。通过叠加一个不可感知的小向量，该向量的各元素等于代价函数对输入的梯度的各元素的符号，我们可以更改 GoogLeNet 对图像的分类结果。此处我们取 E 为 0.007，对应于将 GoogLeNet 转换到实数域后 8 位图像编码的最小量化大小。

设 θ 为模型参数，x 为模型输入，y 为与 x 相关联的目标（对于具有目标的机器学习任务），而J(θ, x, y)为用于训练神经网络的代价函数。我们可以在当前 θ 值附近将代价函数局部线性化，以获得最大范数约束下的最优扰动

$$\boldsymbol{\eta} = \text{Esign}\,(\nabla_{\mathrm{x}} \mathrm{J}(\boldsymbol{\theta}, \mathbf{x}, \mathrm{y}))$$

我们称该方法为生成对抗样本的“快速梯度符号法”。我们可以使用反向传播算法有效地计算所需的梯度。

我们发现该方法可靠地导致了各种模型将其输入错误分类。有关在 ImageNet 上的演示，请参见图 1。我们发现，当 E＝0.25 时，浅 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%。在相同设置下，maxout 网络将 89.4%的对抗样本错误分类，平均置信度为 97.6%。同样，当 E=0.1 时，在预处理版本的 CIFAR-10（Krizhevsky 与 Hinton，2009）测试集[2]上应用卷积 maxout 网络时，我们得到 87.15%的错误率和 96.6%的错误标签平均概率。生成对抗样本的简单方法并不唯一。例如，我们还发现，通过将 x 沿梯度方向旋转一个小角度能可靠地获得对抗样本。

这些简单、快速的算法能够生成被错误分类的样例，这一事实证明了我们对于对抗样本的线性解释的正确性。该算法还可以用作加速对抗训练的方法，或用于对受过训练的网络进行分析。

5 线性模型的对抗训练与权值衰减的对比

也许我们可以考虑的最简单的模型是逻辑回归。在这种情况下，快速梯度符号法可以得到精确的对抗样本。我们可以使用这种情况来直观了解在简单的设置下如何生成对抗样本。有关指导性图像，请参见图 2。

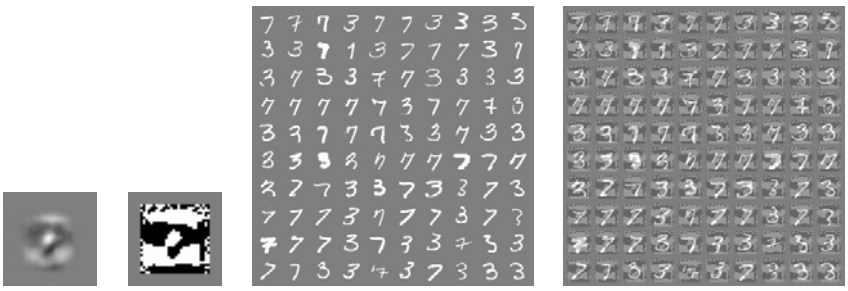
如果我们训练一个模型来识别标签$\mathrm{y} \in \{-1,1\}$，而$\mathrm{P}(\mathrm{y}=1)=\sigma(\mathrm{w}^{\mathrm{T}}\mathrm{x}+\mathrm{b})$，其中$\sigma(\mathrm{z})$是对数 Sigmoid 函数，则训练过程包括在下列函数上的梯度下降

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

其中$\zeta(\mathrm{z})=\log(1+exp(z))$是 softplus 函数。我们可以基于快速梯度符号法得出简单的解析式，用于训练叠加了最坏情况对抗扰动的 x（而不是 x 本身）。

[1] 该测试集使用[0，1]区间中的 MNIST 像素值。MNIST 数据确实包含非 0 或 1 的值，但是图像本质上是二进制的。每个像素大致编码“墨水”或“无墨水”。这表明我们可以期望分类器有能力处理宽度 0.5 范围内的扰动，并且实际上人类观察者可以轻松读取此类图像。

[2] 欲获取预处理代码，请参见 https://github.com/lisa-lab/pylearn2/tree/master/pylearn2/scripts/papers/maxout，数据标准差约为 0.5。



(a) (b) (c) (d)

图 2：快速梯度符号法应用于逻辑回归（不是近似解，而是最大范数盒约束下最具破坏性的精确解）。a）在 MNIST 上训练的逻辑回归模型的权值。b）在 MNIST 上训练的逻辑回归模型的权值符号。这就是最优扰动。即使模型的参数量较低且拟合良好，人类观察者也看不出这种扰动与 3 和 7 之间的关系存在关联。c）MNIST 数据集中的 3 和 7。 在鉴别这些样例是 3 还是 7 的任务中，逻辑回归模型的错误率为 1.6%。d）逻辑回归模型的快速梯度符号对抗样本（E=0.25）。逻辑回归模型在这些样本上的错误率为 99%。

注意，梯度的符号是$-\text{sign}(\text{w})$，而$\text{w}^{\text{T}}\text{sign}(\text{w}) = ||w||_1$。因此，对抗版本逻辑回归要求最小化下列函数

$$\mathsf{E}_{\boldsymbol{x},y\sim p_{\text{data}}}\zeta(y(E||\boldsymbol{w}||_1 - \boldsymbol{w}^T\boldsymbol{x} - b))$$

这有点类似于 L1 正则化。但是，两者有一些重大区别。最显著的是，L1 正则化是在训练期间从模型激活值中减去 L1 惩罚，而不是加 L1 惩罚。这意味着，如果模型学会做出足以使 ζ 饱和的预测，则对抗惩罚最终可能开始消失。这种情况不一定会发生——在拟合不足的情况下，对抗训练只会使拟合不足的状况恶化。因此，我们可以认为 L1 权值衰减提供比对抗训练更“糟糕”的情况，因为在收益良好的情况下，它不会像对抗惩罚一样失效。

如果我们从逻辑回归转向多分类 softmax 回归，则 L1 权值衰减的效果会更令人感到悲观，因为它会将 softmax 的每个输出视为独立可扰动的，而实际上通常不可能找到与所有类的权值向量都一致的扰动η。对具有多个隐藏单元的深层网络而言，权值衰减高估了扰动可能带来的损害。由于 L1 权值衰减高估了对抗扰动可能造成的损害，因此有必要使用比 E 更小的 L1 权值衰减系数，而 E 与我们的特征精度有关。当我们在 MNIST 上训练 maxout 网络时，我们在取 E=0.25 进行对抗训练时获得了良好的效果。当我们将 L1 权值衰减应用于第一层时，我们发现甚至 0.0025 的系数也显得太大了，并且模型在训练集上会卡在 5%误差水平上。较小的权值衰减系数可以成功完成训练，但没有带来正则化收益。

6 深层网络的对抗训练

对深层网络很容易受到对抗样本攻击的批评具有误导性，因为与浅层线性模型不同，深层网络至少能够表达可以抵抗对抗扰动的函数。通用近似定理（Hornik 等人，1989）保证，只要允许其隐藏层具有足够的单位，具有至少一个隐藏层的神经网络就可以按任意精度表达任何函数。浅层线性模型无法在训练点附近变得稳定，同时还会对不同的训练点产生不同的输出

当然，通用近似定理并没有说明特定的训练算法是否能够发现具有所有所需特性的函数。显然，标准的有监督训练并未指明所选函数要能抵抗对抗样本。必须以某种方式将这一要求编码在训练过程中。

Szegedy 等人（2014b）表明，通过在对抗样本和干净样本的混合数据集上进行训练，神经网络会在一定程度上被正则化。用对抗样本进行训练与其他数据增强方案有所不同。通常，人们会通过转换（例如某种预期会在测试集中实际发生的变换）来扩充数据。相反，这种形式的数据增强使用了不太可能自然发生的输入，而这些输入暴露了模型概念化其决策函数的方式中的缺陷。当时未能证明此过程可以在最优基准水平之上获得超过 dropout 的提升。但是，未能证明的部分原因是因为很难对基于 L-BFGS 的计算代价高昂的对抗样本进行广泛的实验。

我们发现用基于快速梯度符号法的对抗目标函数进行训练可以有效地起到正则化作用：

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

在我们所有的实验中，我们设置 α= 0.5。其他值可能会更好，但是我们对这个超参数的最初猜测已经足够好，以至于我们不需要探索更多的值。这种方法意味着我们会不断更新我们提供的对抗样本，让它们能对抗当前版本的模型。进行 dropout 正则化的同时使用这种方法来训练 maxout 网络，我们能够将错误率从无对抗训练时的 0.94％降低到有对抗训练时的 0.84％。

我们观察到在训练集上引入对抗样本后我们没有达到零错误率。我们通过进行两项更改来解决此问题。首先，我们使模型更大，每层使用 1600 个单元，而不是原来的 maxout 网络使用的 240 个单元。如果没有对抗训练，这会使模型略微过拟合，并在测试集上获得 1.14％的错误率。在对抗训练过程中，我们发现验证集错误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，如果验证集错误率在 100 轮内未降低则终止学习。我们发现，虽然验证集错误率曲线非常平坦，但对抗验证集错误率曲线却并非如此。因此，我们又在对抗验证集错误率上使用了早停法。使用前述标准选择要训练的轮数，我们随后对所有 60,000 个样本进行了再训练。我们使用不同种子的随机数生成器进行的五次不同训练，这些随机数生成器用于选择训练样本的小批量，初始化模型权值并生成 dropout 掩模。五次训练生成的模型中四个在测试集上的错误率为 0.77％，一个错误率为 0.83％。其平均值 0.782％是在 MNIST 的排列不变版本上报道的最佳结果，尽管与带 dropout 的微调 DBM 所获得的结果 0. 79%（Srivastava 等人，2014）在统计上没有区别。

我们的模型在某种程度上也可以抵抗对抗样本。回想一下，在没有对抗训练的情况下，在快速梯度符号法生成的对抗样本上，同一模型的错误率为 89.4％。经过对抗训练，错误率降至 17.9％。对抗样本可在两个模型之间迁移，但经过对抗训练的模型显示出更高的鲁棒性。针对原始模型生成的对抗样本在对抗训练模型上导致的错误率为 19.6％，而针对新模型生成的对抗样本在原始模型上导致的错误率为 40.9％。当经过对抗训练的模型确实对一个对抗样本分类错误时，不幸的是，其预测置信度依然非常高。错误分类样本的平均置信度为 81.4％。我们还发现，训练出来的模型的权值发生了显着变化，经过对抗训练的模型的权值明显更具局域性和可解释性（见图 3）。

对抗训练过程可以看作是使最坏情况对抗干扰误差最小化。这可以解释为学习玩对抗游戏，也可以解释为在输入含有$U(-\epsilon, \epsilon)$噪声时，最小化带噪样本的期望代价的上界。对抗训练也可以看作是主动学习的一种形式，其中模型能够要求新数据点的标签。在这种情况下，人类标签者将被替换为从附近点复制标签的启发式标签器。

我们还可以通过对 ε 最大范数盒内的所有数据点进行训练，或对该盒内的许多点进行采样，来对模型进行正则化，使其对小于 ε 精度的特征变化不敏感。这相当于在训练期间以最大范数 ε 叠加噪声。但是，均值为零且协方差为零的噪声在抵抗对抗样本方面非常低效。任何参考矢量与此类噪声矢量之间的点积期望为零。这意味着在许多情况下，除了让输入更难看外噪声基本上不会产生影响。

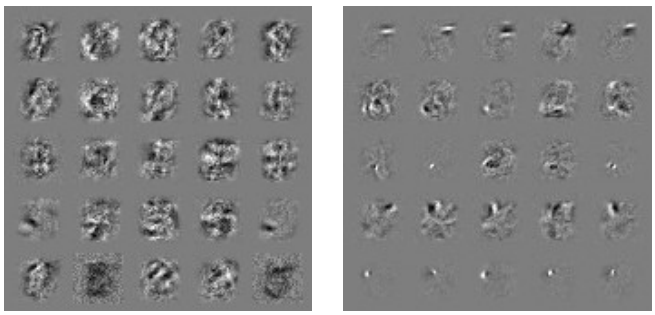


图 3：在 MNIST 上训练的 maxout 网络的权值可视化。每行显示的是单个 maxout 单元的滤波器。左）以朴素方式训练的模型。右）经过对抗训练的模型。

实际上，在许多情况下，噪声确实会导致较低的目标函数值。我们可以将对抗训练视为在带噪输入中进行硬样本挖掘，仅考虑那些强烈抵抗分类的噪点来更有效地进行训练。作为控制变量实验，我们训练了两个 maxout 网络，其中一个网络对每个像素随机添加±E 的噪声；另一个网络对每个像素添加$\mathrm{U}(-\mathrm{E},\mathrm{E})$的噪声。在快速梯度符号对抗样本上，它们分别以 97.3％的置信度取得了 86.2％的错误率，以及以 97.8％的置信度取得了 90.4％的错误率。

由于符号函数的导数在任何地方要么为零要么无定义，因此基于快速梯度符号法的对抗目标函数的梯度下降无法使模型预测对抗样本对参数变化的反应。如果我们取而代之使用基于小旋转或按比例缩放梯度的对抗样本，则扰动过程本身是可导的，学习算法可以将对抗样本的反应考虑在内。但是，我们发现此过程的正则化效果不够强大，也许是因为这类对抗样本并不难对付。

一个自然的问题是，扰动输入层或隐藏层或同时扰动两者，这几种方案中哪种更好。这方面的结果不一致。Szegedy 等人（2014b）报道，当将对抗扰动应用于隐藏层时，其规则化效果最佳。该结果是在 Sigmoid 网络上获得的。在我们使用快速梯度符号法的实验中，我们发现隐藏单元的激活不受限制的网络只是通过使其隐藏单元的激活非常大来做出响应，因此通常最好只扰动原始输入。在诸如 Rust 模型的饱和模型中，我们发现输入的扰动与隐藏层的扰动效果相当。基于旋转隐藏层的扰动解决了无界激活无限制增长从而使加性扰动相对较小这一问题。我们能够成功地利用隐藏层的旋转扰动训练 maxout 网络。但是，这并未产生与输入层加性扰动同样强的正则化效果。我们对对抗训练的观点是，只有在模型有能力学会抵抗对抗样本时，对抗训练才明显有用。仅当应用通用近似定理时，情况才很明显。由于神经网络的最后一层，即线性 sigmoid 或线性 softmax 层，并不是最后一层隐藏层通用近似函数，因此在对最后一层隐藏层施加对抗扰动时，很可能会遇到拟合不足的问题。我们确实发现了这种效应。使用隐藏层扰动进行训练的最佳结果从未涉及最后一层隐藏层的扰动。

7 不同类型的模型容量

对抗样本的存在似乎违反直觉的原因之一是，我们大多数人对高维空间的直觉很糟糕。我们生活在三维空间中，因此我们不习惯将数百个维度中的小效果加起来产生大效果。我们的直觉还以另一种方式误导我们。许多人认为小容量模型无法做出许多不同的自信预测。这是不正确的。某些小容量模型确实做出了不同的自信预测。例如，满足如下条件的浅 RBF 网络

$$p(y = 1 \mid \boldsymbol{x}) = \exp((\boldsymbol{x} - \mu)^T \boldsymbol{\beta}(\boldsymbol{x} - \mu))$$

只能对 μ附近的阳性样本做出自信的预测。在别的地方，它要么默认无有效分类，要么只能做出低置信度的预测。

RBF 网络天生就不受对抗样本的影响，因为它们被欺骗时信心不足。取 E =0.25 时，快速梯度符号法生成的对抗样本可以使无隐藏层的浅层 RBF 网络在 MNIST 上达到 55.4％错误率。但是，它对错误分类样本的置信度仅为 1.2％。它对干净测试样本的平均置信度为 60.6％。我们不能期望具有如此小容量的模型对所有数据点都能获得正确的答案，但是它确实可以通过大幅降低其对“无法理解”的点的置信度来做出正确的响应。

不幸的是，RBF 单元不具有显著变换下的不变性，因此它们不能很好地泛化。我们可以将线性单元和 RBF 单元视为精确度-召回率折衷曲线上的不同点。线性单元通过在特定方向上响应每个输入来实现较高的召回率，但由于在不熟悉的情况下响应过强而导致精确度较低。RBF 单元通过仅对空间中的特定点做出响应来达到高精确度，但这样做会牺牲召回率。受此想法的启发，我们决定探索各种涉及二次单元的模型，包括深层 RBF 网络。我们发现这是一项艰巨的任务——使用 SGD 训练时，具有足够二次抑制能力以抵抗对抗干扰的模型却得到了很高的训练集错误率。

8 为什么对抗样本能够泛化？

对抗样本的一个有趣特性是，为一个模型生成的对抗样本经常被其他模型错误分类，即使它们具有不同的架构或在不同训练集上训练。此外，当这些不同的模型将同一个对抗样本错误分类时，它们通常在分类结果上彼此一致。基于极端非线性和过拟合的假说难以解释这种现象——为什么具有过高容量的多个极端非线性模型必须以相同的方式一致地标记这些超出自然分布的数据点？从“对抗样本像有理数铺填实数一样精细地铺填数据空间”这一假设来看，这种行为尤其令人惊讶，因为在这种观点中，对抗性示例很常见，但仅在非常精确的位置出现。

在线性视角下，对抗样本出现在一个较宽的子空间中。只需保证方向 η 与代价函数的梯度点积为正， E 足够大即可。图 4 展示了这种现象。通过追踪 E 的不同值，我们看到对抗样本出现在由快速梯度符号法确定的 1 维连续区域中，而不是在细小口袋中。这就解释了为什么对抗样本如此丰富，以及为什么被一个分类器错误分类的样本会以较高的先验概率被另一分类器错误分类的原因。

为了解释为什么不同的分类器将相同的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同一训练集上训练的线性分类器。当在训练集的不同子集上训练时，该参考分类器能够学习大约相同的分类权值，这是因为机器学习算法能够泛化。内在分类权值的稳定性反过来又导致对抗样本的稳定性。

为了检验该假设，我们在深层 maxout 网络上生成了对抗样本，并使用浅层 softmax 网络和浅层 RBF 网络分类这些样本。在被 maxout 网络错误分类的样本上，RBF 网络的预测结果仅有 16.0％与 maxout 网络的预测结果相同，而 softmax 分类器的预测结果仅有 54.6％与 maxout 网络的预测结果相同。但是这些数字很大程度上是由不同模型的不同错误率导致的。如果我们将注意力集中在两个模型都出错的情况下，则 softmax 回归的预测结果有 84.6％与 maxout 相同，而 RBF 网络只有 54.3％的预测结果与 maxout 相同。相比之下，RBF 网络有 53.6％的预测结果与 softmax 回归相同，因此它的行为中具有很强的线性部分。我们的假设无法解释 maxout 网络的所有错误或跨模型泛化的所有错误，但显然其中很大一部分是因为线性行为是跨模型泛化的主要原因。

9 备择假说

我们现在考虑并反驳一些对抗样本假说。首先，一个假说是，生成训练可以对训练过程提供更多约束，

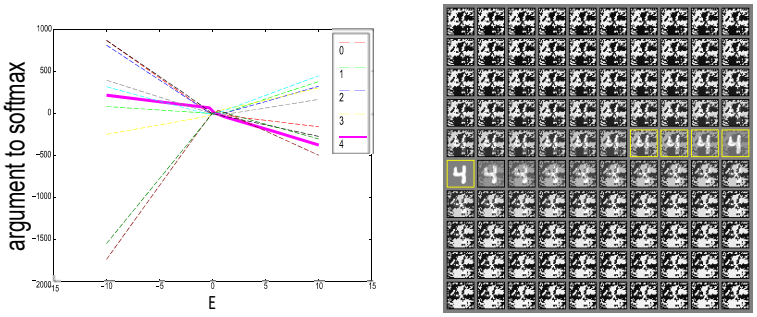


图 4：通过追踪不同的 E 值，我们可以看到，只要我们沿正确的方向移动，几乎所有足够大的 E 值都会可靠地产生对抗样本。正确的分类仅在数据中出现 x 的细歧管上发生。R^n空间大部分由对抗样本和垃圾分类样本组成（请参阅附录）。本图是用一个经朴素方法训练的 maxout 网络制作的。左）显示了在我们对单个输入样本改变 E 时，对所有 10 个 MNIST 类的 softmax 层参数。正确的类别是 4。我们可以看到每个类别的未归一化对数概率与 E 明显成分段线性关系，并且错误分类在 E 值的宽范围内都是稳定的。此外，随着我们增大 E 至足以进入垃圾输入区，这种预测变得非常极端。右）用于生成曲线的输入（左上方=负 E，右下方=正 E，黄色框表示被正确分类的输入）。

或者使模型学习区分“真”数据与“假”数据的方法，并仅对“真”数据有信心。MP-DBM（Goodfellow 等，2013a）提供了一个很好的模型来检验这一假说。它的推断过程在 MNIST 上具有良好的分类精度（错误率 0.88%）。该推断过程是可导的。其他生成模型或是推断过程不可导从而难以计算对抗样本，或是需要额外的非生成性鉴别器模型才能在 MNIST 上获得良好的分类精度。对于 MP-DBM，我们可以确定是生成模型本身对对抗样本有响应，而不是最上面的非生成分类器模型。我们发现该模型容易受到对抗样本的攻击。当 E 为 0.25 时，我们发现模型对从 MNIST 测试集生成的对抗样本的错误率为 97.5%。其他形式的生成训练仍然有可能赋予抵抗力，但显然，生成性这一事实本身并不足够。

关于为何存在对抗样本的另一个假说是，单个模型具有奇怪的怪癖，但许多模型的平均能排除对抗样本。为了检验这一假设，我们在 MNIST 上训练了 12 个 maxout 网络。每个网络都使用不同的随机数生成器种子进行训练，随机数生成器用于初始化权值，生成 dropout 掩模以及选择数据的小批次以进行随机梯度下降。在旨在干扰的所有模型的 E =0.25 的对抗样本上，模型的错误率为 91.1%。如果我们改用旨在干扰单一模型的对抗样本，则错误率将降至 87.9%。多模型集合仅提供有限的抵抗对抗干扰的能力。

10 总结与讨论

作为总结，本文有如下观察结果：

- 对抗样本可以解释为高维点积的属性。它们是模型过于线性而不是非线性的结果。
- 可以将对抗样本在不同模型上的泛化解释为：对抗扰动与模型的权值向量高度一致，并且不同模型在训练执行相同任务时会学习到相似的函数。
- 扰动的方向而不是空间中的特定点最为重要。空间中对抗样本并非像有理数精细地填充实数一样填充数据空间。
- 因为方向最重要，所以对抗扰动对不同的干净样本普遍存在。

- 我们引入了一系列生成对抗样本的快速方法。
- 我们展示了对抗训练可以带来比 dropout 更强的的正则化效果。
- 我们进行的控制实验没能用更简单但是效果更差的正则化方法，包括 L1 权值衰减和加噪声，复现对抗训练的效果。
- 容易训练的模型也容易被干扰。
- 线性模型缺乏抵抗对抗干扰的能力；只有带隐藏层的架构（此时通用近似定理适用）能被训练抵抗对抗干扰。
- RBF 网络能抵抗对抗样本。
- 对输入分布建模的模型不能抵抗对抗样本。
- 模型集合不能抵抗对抗样本。

其他一些对垃圾类样本的进一步观察参见附录：

- 垃圾类样本无处不在，易于生成。
- 浅层线性模型不能抵抗垃圾类样本。
- RBF 网络能抵抗垃圾类样本。

基于梯度的优化是现代 AI 的主力军。通过使用足够线性的网络——无论是 ReLU 或是 maxout 网络，LSTM 还是经过精心配置不过饱和的 Sigmoid 网络——我们能够解决我们关心的大多数问题（至少在训练集上可以）。对抗样本的存在表明，能够解释训练数据，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了我们要求它们执行的任务。取而代之的是，它们的线性响应对数据分布中未出现的点过于自信，并且这些自信的预测通常非常不正确。这项工作表明，我们可以通过显式标识有问题的点并在每个点处校正模型来部分纠正此问题。但是，可能还会得出一个结论，即我们使用的模型族存在本质缺陷。易于优化的代价是模型容易被误导。这激励了新的优化程序的开发，使之能够训练行为在局部更稳定的模型。

致谢

感谢 Geoffrey Hinton 和 Ilya Sutskever 的有益讨论。感谢 Jeff Dean，Greg Corrado 和 Oriol Vinyals 对本文草案的反馈。感谢 Theano（Bergstra 等，2010; Bastien 等，2012），Pylearn2（Goodfellow 等，2013b）和 DistBelief（Dean 等，2012）的开发者。

参考文献

Bastien, Fre´de´ric, 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, Fre´de´ric, Lamblin, Pascal, Pascanu, 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, Fre´de´ric, 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, Dumitru, 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`.

附录 垃圾类样本

与对抗样本相关的概念是从“垃圾类”中抽取的样本。这些样本是退化输入，人类会将其归类为不属于训练集中的任何类别。如果我们称训练集中的类别为“阳性类别”，那么我们要小心避免垃圾输入的伪阳性——即，我们不想将退化的输入归类为真实的东西。在每个类别使用单独的二分类器的情况下，我们希望所有类别的输出均接近出现该类别的可能性为零，并且在仅阳性类服从多元分布的情况下，我们希望分类器输出类的高熵（几乎是均匀的）分布。减少对垃圾输入的脆弱性的传统方法是向表示垃圾类别（？）的模型引入额外的恒定输出。Nguyen 等人（2014 年）最近在计算机视觉的背景下以“愚弄图像”的名义重新普及了垃圾分类的概念。与对抗样本一样，人们一直误以为垃圾类别的假阳性很难找到，并且它们主要是深层网络所面临的问题。

我们对线性和高维空间产生对抗样本的解释也适用于分析该模型在垃圾类样本中的行为。线性模型在远离训练数据的点比在靠近训练数据的点产生更多的极端预测。为了找到这种模型的高置信度垃圾假阳性，我们只需要生成一个远离数据的点，更大的范数就会产生更高的置信度。RBF 网络无法对远离训练数据的点给出高置信度的类别预测，因此不会被这种现象所愚弄。

我们从 $N(0, I_{784})$ 中生成了 10,000 个样本，并将它们输入到 MNIST 数据集的各个分类器中。在这种情况下，我们认为对任何类别预测大于 0.5 的概率都是错误。一个朴素地经训练的 maxout 网络（顶部具有 softmax 层）对高斯垃圾样本的分类错误率为 98.35%，错误分类时的平均置信度为 92.8%。将顶层更改为独立的 Sigmoid 单元后错误率降至 68%，平均错误置信度为 87.9%。在 CIFAR-10 上，使用来自 $N(0, I_{3072})$ 的 1,000 个样本，卷积 maxout 网络获得 93.4%的错误率，平均置信度为 84.4%。

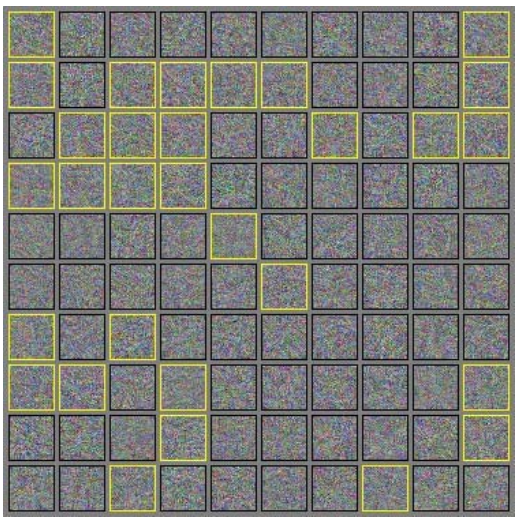


图 5：在 CIFAR-10 上训练的卷积网络的随机欺骗图像。这些样本是通过从各向同性的高斯分布中抽取样本，然后在增加“飞机”类别概率的方向上采取梯度符号步骤生成的。黄色框表示样本成功欺骗了模型，并认为飞机存在的置信度至少为 50%。“飞机”是 CIFAR-10 中最难构造欺骗图像的类别，因此从成功率的角度来看，这个数字代表最坏的情况。

这些实验表明 Nguyen 等人（2014）采用的优化算法矫枉过正（或者可能只在 ImageNet 上有必要），并且愚弄的图像中丰富的几何结构是由于其搜索过程中编码的先验，而不是那些只能导致假阳性的结构。

虽然 Nguyen 等人(2014)将注意力集中在深度网络上，浅层线性模型也存在同样的问题。对垃圾样本，softmax 回归模型的错误率为 59.8%，错误的平均置信度为 70.8%。如果我们改用行为不像线性函数的 RBF 网络，则会发现错误率为 0%。注意，当错误率为零时，对错误的平均置信度是不确定的。

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以高置信度分类为任何类别的伪造点更难。MNIST 和 CIFAR-10 上的上述方法往往在类上具有非常不均匀的分布。在 MNIST 上，朴素训练的 maxout 网络的假阳性的 45.3%被分类为 5，没有一个被分类为 8。同样，在 CIFAR-10 上，卷积网络的假阳性中有 49.7%被归类为青蛙，而没有一个被归类为飞机，汽车，马匹，轮船或卡车。

为了解决 Nguyen 等人(2014)提出的针对特定类别生成虚假图像这一问题，我们建议将 $E\nabla_{\boldsymbol{x}} p(y=i/\boldsymbol{x})$ 加到高斯样本 x 中，作为生成分类为 i 类虚假图像的快速方法。如果我们重复此采样过程直到成功，我们将获得具有可变运行时间的随机算法。在 CIFAR-10 上，我们发现单一采样步骤对青蛙和卡车的成功率为 100%，而最困难的类别是飞机，每个采样步骤的成功率为 24.7%。对所有十个类别取平均，该方法的平均每步成功率为 75.3%。因此，我们可以生成任何所需类，只用少量样本且无需特殊先验，而无需数万代的进化。为了确认得到的样本确实是愚弄图像，而不是由梯度符号方法产生的真实类的图像，请参见图 5。对于包含更多类的数据集，此方法在生成类 i 成员方面的成功率可能会降低，因为在这种情况下，无意中增加了不同类 j 激活的风险更高。我们发现，我们能够训练一个 maxout 网络，使高斯垃圾样本的错误率达到零（它仍然容易受到对高斯样本应用快速梯度符号迭代生成的垃圾样本的影响），而对其分类干净样本的能力没有负面影响。不幸的是，与对抗训练不同，这并未导致模型测试集错误率的显着降低。

总之，用线性零件构建的深度或浅层模型似乎很可能会错误地处理随机选择的输入，并且这些模型仅在包含训练数据的非常薄的流形上表现合理。