第10章-对抗性样本

第10章-对抗性样本

本章涵盖

10.1. 对抗性样本的上下文

10.2. 谎言，该死的谎言和分布

10.3. 使用和滥用训练

10.4. 信号与噪音

10.5. 并非所有希望都已消失

10.6. GANs的缺点

10.7. 结论

总结

参考文献

本章涵盖

- GAN之前的研究领域和交织的历史
- 计算机视觉中的深度学习方法
- 我们自己的具有真实图像和噪音的对抗示例

在本书的学习过程中，您已经将GAN理解为一个明确的概念。但是，在2014年，GAN似乎上是一次质的飞跃，尤其是对于那些不熟悉新兴的对抗性领域的人来说，包括Ian Goodfellow等人在该领域的工作[1]。本章将深入研究对抗性样本(Adversarial examples)，这些样本是特殊构造的样本，这些样本使其他分类算法发生灾难性的失效。”

我们还讨论了他们与GAN的联系，以及为何对抗性学习(adversarial learning)仍然是机器学习中一个尚未解决的问题--这是当前方法的一个重要但很少讨论的缺陷。包括对抗性样本在机器学习的鲁棒性，公平性和安全性中起着重要作用

不可否认，在过去五年中，机器学习的能力正在匹及并超越人类水平,其性能方面取得了长足进步，例如在计算机视觉（CV）分类任务或玩游戏的能力上[2]。但是，仅查看指标和ROC曲线[3]不足以让我们理解:（a）神经网络为何做出决策（工作方式）以及（b）神经网络容易出错。本章涉及第一章，然后深入到第二章。在开始之前，应该说，尽管本章几乎只涉及计算机视觉的问题，但在诸如文本甚至人类的各个领域都已经发现了对抗性样本[4]。

首先，当我们谈论神经网络的性能时，我们经常读到它们在大型数据集ImageNet上的分类错误率低于人类。这是经常被引用的统计数据（比起其他任何事情都更像是一个学术笑话），其掩盖了在平均值下方隐藏的绩效差异。尽管人类的错误通常主要是由于他们无法区分在此数据集中突出显示的不同品种的狗而引起的，但机器学习失败的情况却不祥得多。经过进一步调查，出现了一些对抗性例子。

与人类不同，CV算法要解决的问题在本质上非常不同并且是与训练数据有关的问题。因为该算法必须对每张图片进行预测，所以即使我们有很多实例，也必须通过训练数据中看到的孤立和遥远的个体实例之间进行推断。

当我们训练了诸如Inception V3和VGG-19之类的网络时，我们发现了一种使图像分类器在训练数据的细流形(thin manifold)上工作的惊人方法。但是，当人们试图戳破这些算法的分类能力时，他们发现了一个宇宙级大坑-当前的机器学习算法很容易被甚至很小的扭曲所愚弄。迄今为止，几乎所有成功的主要机器学习算法都在一定程度上遭受了该缺陷的影响，并且确实有人推测这就是为什么机器学习起作用的原因。

> NOTE：在有监督的情况下，思考我们的训练集。我们有一个训练流形-描述了样本所在高维度的分布。例如，我们的300×300像素图像存在与在270,000维空间（300×300×3color）中。这使得训练非常复杂。

10.1. 对抗性样本的上下文

首先，我们想快速讲解为什么在本书结尾处包含本章：

- 通过对抗性样本，我们通常试图生成新的样本，这些样本会使我们现有的系统被蒙蔽，从而错误地分类。我们通常以邪恶的攻击者的身份或以研究人员的身份这样做，以了解我们的系统将如何强大地运行。尽管存在重大差异，但对抗性示例与GAN密切相关。
- 这将使您了解为什么GAN可能很难训练，以及为什么我们现有的系统如此脆弱。
- 对抗性示例允许GAN使用不同的应用程序集合，我们希望至少为您提供其功能的基础知识。

就应用而言，对抗性样本很有趣，原因有以下几个：

- 正如所讨论的，对抗性示例可以用于恶意目的，因此测试关键系统的鲁棒性非常重要。如果攻击者可以轻易地愚弄面部识别系统来访问您的手机怎么办?
- 它们帮助我们了解机器学习的公平性-这是一个日益重要的主题。我们可以使用对抗性学习的表示形式，这些表示形式对于分类任务很有用,使攻击者不能恢复受保护的数据，这可能是确保我们的机器学习不歧视任何人的最佳方法之一。
- 同样，我们可以使用对抗性学习来保护有关个人的敏感信息（可能是医疗或财务信息）的隐私。在这种情况下，我们仅关注与无法恢复的个人有关的信息。“

按照当前的研究现状，了解对抗性样本是开始理解对抗性防御的唯一方法，因为大多数论文都首先描述了其防御的攻击类型，然后才尝试解决它们。在撰写本书时，没有针对所有类型攻击的通用防御措施。但是，这是否是研究它们的好理由，取决于您对对抗样本的看法。我们决定不详细介绍防御措施-在本章末尾的高级思想之上-因为超出此范围的所有内容均不包括在内。

10.2. 谎言，该死的谎言和分布

要真正理解对抗性示例，我们必须回到CV分类任务的领域来了解该任务的难度。回想一下，从原始像素到最终够对图像集进行分类具有的挑战性。

为了拥有一个真正可泛化的算法，我们必须对数据做出明智的预测，而这些预测要远远超过训练集中所见的任何数据。而且，即便是我们稍微改变拍摄照片的角度，手边的图像和同一类训练集中最近的图像之间的像素级差异也很大。

当我们在RGB空间中拥有100,000个300×300图像样本的训练集时，我们必须以某种方式处理270,000维度的数据。当我们考虑所有可能的图像（不是我们实际观察到的图像，而是可能发生的图像）时，每个维度的像素值都与其他维度无关，because we can always generate a valid picture by rolling a hypothetical 256-sided dice 270,000 times. (这句话我真的不会翻译)。因此，理论上我们在8位颜色空间上有$256^{270,000}$个示例（一个数字，长650,225位）。

我们将需要很多样本来覆盖此空间的1%。当然，这些图像大多数都没有任何意义。通常，我们的训练集比这少很多，因此我们需要我们的算法来训练，使用相对有限的数据来推断甚至还没有看到的区域。这是因为该算法最有可能在训练集中没有看到任何东西。

我们知道算法必须有意义地泛化。他们必须能够有意义地填充他们从未见过任何样本的大部分空间。计算机视觉算法之所以起作用，主要是因为它们可以对大量的丢失概率做出很好的猜测，但是它们的优势也是其最大的弱点。

> Note:通常最少要有100,000个样本，深度学习算法才能真正开始发光。

10.3. 使用和滥用训练

在本节中，我们介绍两种关于对抗性样本的思考，一种是基于第一原理，另一种是通过类推。第一种思考对抗性样本的方法是从训练机器学习分类的方法开始。请记住，这些是具有数千万参数的网络。在整个培训过程中，我们会更新其中的一些内容，以使类别与训练集中提供的标签相匹配。我们只需要找到正确的参数更新，这就是随机梯度下降（SGD）允许我们执行的操作。 现在回想一下简单的分类器时代，然后您对GAN有了更多了解。这里我们有某种可学习的分类函数$f_\theta(x)$（例如，深度神经网络或DNN），其由θ（DNN的参数）进行参数化，并以x（例如，图像）作为输入，然后对它分类。在训练时，我们将其$\hat{y}$与真实的标签y进行比较，这就是我们得到损失(L)的方式。然后，我们更新$f_\theta(x)$的参数，以使损失最小化。公式(equation 10.1)、(equation 10.2)和(equation 10.3)进行了总结[5]:。

equation 10.1

$$\hat{y} = f_\theta(x) \tag{1}$$

equation 10.2

$$L = \|y - \hat{y}\| \tag{2}$$

equation 10.3

$$\min_\theta \|\mathrm{y} - \hat{y}\| \text{ s.t. } \hat{y} = f_\theta(x) \tag{3}$$

从本质上讲，我们已将神经网络的输出定义为预测得到的样本（equation 10.1）。损失是真实标签和预测标签（equation 10.2）之间的某种距离的度量.然后将整个问题表述为试图调整DNN的参数使得真实标签和预测标签之间的差异的度量最小化.然后构成样本的预测(equation 10.3).

这一切都很好，但是实际上如何使分类损失最小化呢？我们如何解决公式10.3中所述的优化问题？我们通常使用基于SGD的方法来获取 batches of *x*；然后，我们将损失函数当前参数(θ_t)的导数乘以我们的学习率（α），即我们的新参数(θ_{t+1}),参见(equation 10.4)。

$$\theta_{t+1} = \theta_t - \alpha * \frac{\partial L}{\partial \theta} \tag{4}$$

这是您将找到的最快的深度学习入门。但是，既然您已经有了这个上下文，请考虑一下是否可以将此强大工具（SGD）也用于其他目的。例如，当我们加大损失空间而不是减少损失时会发生什么？事实证明，最大化错误要比最小化错误要容易得多，同时也很重要。就像许多重大发现一样，它开始时似乎是一个BUG，后来变成了骇客：如果我们开始更新像素而不是权重，该怎么办？如果我们恶意更新它们，则会出现对抗性样本。 你们中的一些人可能会对SGD的快速回顾感到困惑，所以让我们来回顾典型的损失空间可能是什么样子？

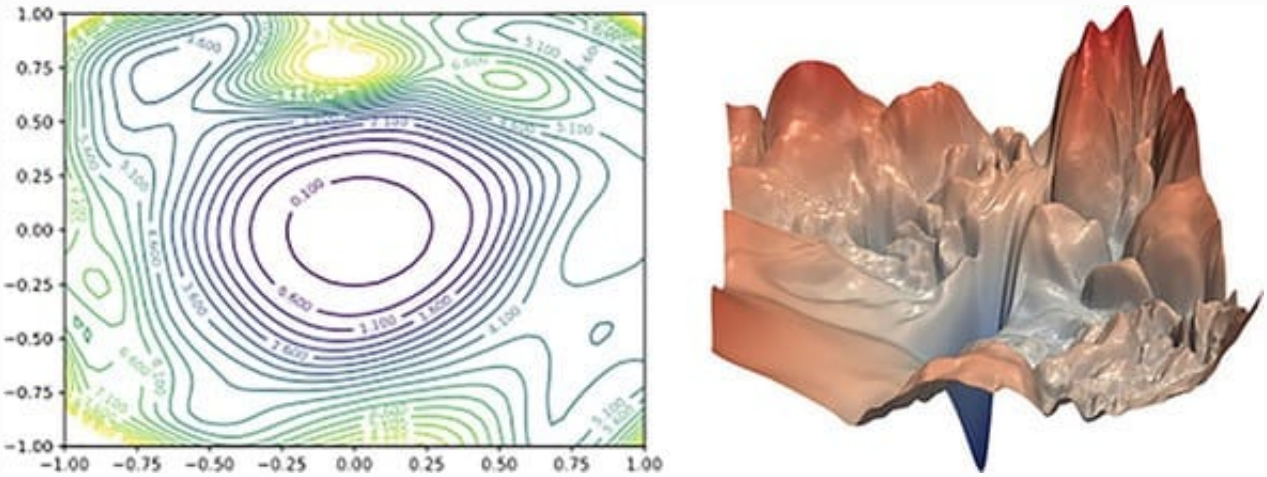

Figure 10.1. 在这种典型的损失空间中，这就是我们的深度学习算法可以切实获得的损失值的类型。在左侧，您具有相等损失的2D等高线，在右侧，您可以看到损失空间的外观的3D渲染。还记得第6章中的登山吗?

(Source: "Visualizing the Loss Landscape of Neural Nets," by Tom Goldstein et al., 2018, https://github.com/tomgoldstein/loss-landscape.)

第二个可以用来思考对抗性的有效的思维模型的例子。您可能会将对抗性样本视为条件GAN(CGAN)，就像我们在前两章中遇到的那样。借助对抗性样本，我们将以整个图像为条件，并尝试生成一个域转移或相似的图像，但在欺骗分类器的域中除外。"生成器"可以是简单的随机梯度上升，仅调整图像即可使其他分类器蒙蔽。

两种方式中的哪一种对您都有意义，现在让我们直接了解对抗性样本及其外观。它们是为了使这些变更后的图像很容易被错误地分类。实现此目标的首批方法之一是快速符号梯度法（FSGM），它与我们之前的描述一样简单。

首先从梯度更新（equation 10.4）开始，看一下符号，然后朝相反的方向走一小步。实际上，经常出现图像看起来（几乎）相同！A picture is worth a thousand words to show you how little noise is needed(俚语),见figure 10.2。

Figure10.2:一点噪音会带来很大的不同。中间的图片上有噪点（差异）。当然，正确的图片会被大量放大（大约300倍）并移动，以便可以创建有意义的图像。

现在，我们对该未修改的图像运行ResNet-50预训练分类器，并检查表10.1所示的前三个预测；

Table10.1原始图像预测

Order	Class	Confidence
First	mountain_tent	0.6873
Second	promontory	0.0736
Third	valley	0.0717

前三名都很明智，mountain_tent排在第一位。Table 10.2显示了对抗性图像的预测。前三名完全错过了mountain_tent，并提出了一些至少与户外活动相匹配的建议，但即使修改后的图像也显然不是suspension_bridge。

Table 10.2对抗图像预测

Order	Class	Confidence
First	volcano	0.5914
Second	suspension_bridge	0.1685
Third	valley	0.0869

这就是我们可以扭曲预测的程度，而预算只有大约200个像素值（相当于将一个几乎黑色的像素变成一个几乎白色的像素）分布在整个图像上。

一件令人惊讶的事情是创建整个示例所需的代码很少。在本章中，我们将使用一个名为foolbox的库，该库提供了许多方便的方法来创建对抗性示例。事不宜迟，让我们深入了解它。我们从众所周知的导入开始，再加上foolbox，这是一个专门设计用来简化对抗攻击的库。

Listing 10.1. Our trusty imports

```
import numpy as np
from keras.applications.resnet50 import ResNet50
from foolbox.criteria import Misclassification,
ConfidentMisclassification
from keras.preprocessing import image as img
from keras.applications.resnet50 import preprocess_input,
decode_predictions
import matplotlib.pyplot as plt
import foolbox
import pprint as pp
Import keras
%matplotlib inline
```

Next, we define a convenience function to load in more images. Listing 10.2. Helper function

```
def load_image(img_path: str):
  image = img.load_img(img_path, target_size=(224, 224))
  plt.imshow(image)
  x = img.img_to_array(image)
  return x
image = load_image('DSC_0897.jpg')
```

接下来，我们必须设置Keras来创建我们的模型，并从Keras的 convenience function中下载ResNet-50。

Listing 10.3. Creating [tables 10.1]and [10.2]

```
keras.backend.set_learning_phase(0) #实例化模型
kmodel = ResNet50(weights='imagenet')
preprocessing = (np.array([104, 116, 123]), 1)
#从Keras模型创建foolbox对象
fmodel = foolbox.models.KerasModel(kmodel,
                                   bounds=(0, 255),
                                   reprocessing=preprocessing)
#我们制作图像（1,2,224,224,3），使其适合ResNet-50
to_classify = np.expand_dims(image, axis=0)

preds = kmodel.predict(to_classify)          #我们称之为预测并打印结果。

print('Predicted:', pp.pprint(decode_predictions(preds, top=20)
[0]))
label = np.argmax(preds)       # 获取编号最高的索引，作为以后使用的标签
# ::-1表示反转颜色通道，因为Keras ResNet-50需要BGR而不是RGB。
```

```
image = image[:, :, ::-1]
#创建攻击对象，设置较高的错误分类标准
attack = foolbox.attacks.FGSM(fmodel, threshold=.9,


 criterion=ConfidentMisclassification(.9))
#对源图像施加攻击
adversarial = attack(image, label)
#在对抗性图像上获取新的预测
new_preds = kmodel.predict(np.expand_dims(adversarial, axis=0))
print('Predicted:', pp.pprint(decode_predictions(new_preds, top=20)
[0]))
```

使用这些示例非常简单！现在您可能在想，也许仅仅是ResNet-50受这些示例的影响。好吧，我们有一些坏消息要给您。在我们测试本章的各种代码设置时，ResNet不仅被证明是最难破解的分类器，而且在每个ImageNet类别中DAWNBench都是无可争议的赢家（这是DAWNBench的CV类别中最具挑战性的任务），见Figure 10.3[6]

Image Classification on ImageNet

Training Time

All Submissions

Objective: Time taken to train an image classification model to a top-5 validation accuracy of 93% or greater on ImageNet.

Rank	Time to 93% Accuracy	Model	Hardware	Framework
1 Dec 2018	0:09:22	ResNet-50 *ModelArts Service of Huawei Cloud* source	16 * 8 * Tesla-V100(ModelArts Service)	Huawei Optimized MXNet
2 Nov 2018	0:10:28	ResNet-50 *ModelArts Service of Huawei Cloud* source	16 nodes with RDMA (8*V100 for each node)	TensorFlow v1.8.0
3 Sep 2018	0:18:06	ResNet-50 *fast.ai/DIUx (Yaroslav Bulatov, Andrew Shaw, Jeremy Howard)* source	16 p3.16xlarge (AWS)	PyTorch 0.4.1

但是对抗性样本的最大问题是它们的普适性。对抗性样本不仅限于深度学习，还可以推广到不同的机器学习技术。如果我们针对一种技术生成一个对抗样本，那么它很有可能甚至可以在我们尝试攻击的另一种模型上运行，如 figure10.4 所示。

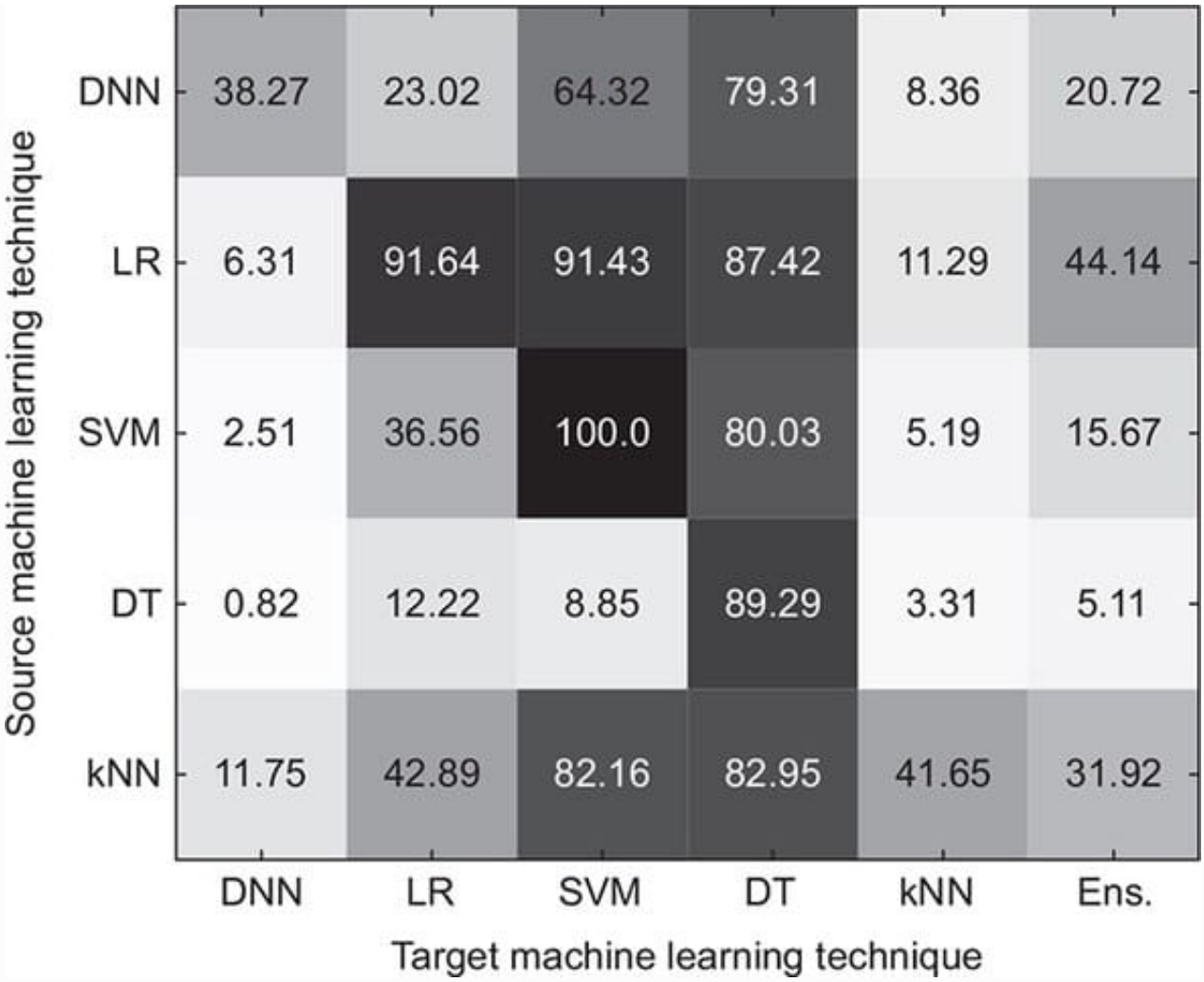

Figure 10.4。此处的数字表示为欺骗该行中的分类器而欺骗该列的分类器的对抗性样本所占的百分比。这些方法是深度神经网络（DNN），逻辑回归（LR），支持向量机（SVM），决策树（DT），最近邻居（kNN）和继承学习（Ens）。

(Source: "Transferability in Machine Learning: from Phenomena to Black-Box Attacks Using Adversarial Samples," by Nicolas Papernot et al., 2016, https://arxiv.org/pdf/1605.07277.pdf.)

10.4. 信号与噪音

更糟糕的是，许多对抗性样本都很容易被构造，以至于我们可以轻易地利用从np.random.normal中进行采样得到的高斯噪声欺骗分类器。另一方面，为了支撑我们早先的ResNet-50是一个相当鲁棒性的体系结构，我们将向您展示其他体系结构受此问题的影响更大。

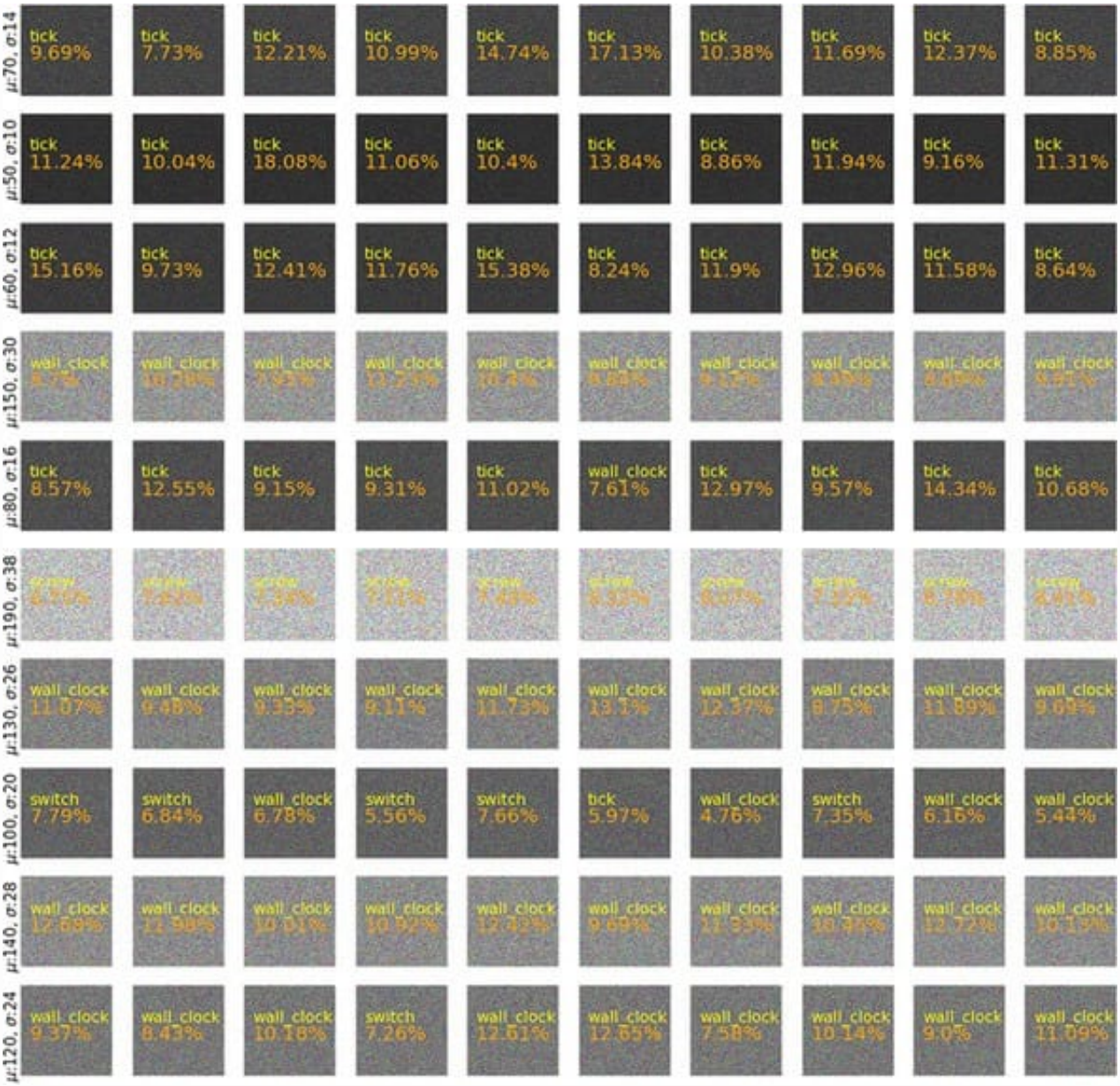

Figure 10.5,显然，在大多数情况下，仅凭简单的采样噪声，我们无法将分类归类为错误的分类。这就是ResNet-50的优点。在左侧，我们包括了所使用的均值和方差，以便您查看其影响。

Figure 10.5显示了在纯高斯噪声下运行ResNet-50的结果。但是，我们可以对噪声本身进行对抗性攻击，以查看图像分类错误的程度，而且可以更快地得到。

在Figure10.4中，我们将使用投影梯度下降（PGD）攻击，如图10.6所示。尽管这仍然是一种简单的攻击，但它需要进行高级的解释。与以前的攻击不同，我们现在正在采取一些措施，不管它可能导致我们到哪儿（甚至是"无效"的像素值），然后再投射到可行的空间上。现在，我们将PGD攻击应用于Figure 10.7中的高斯噪声，并运行ResNet-50看看我们如何做。

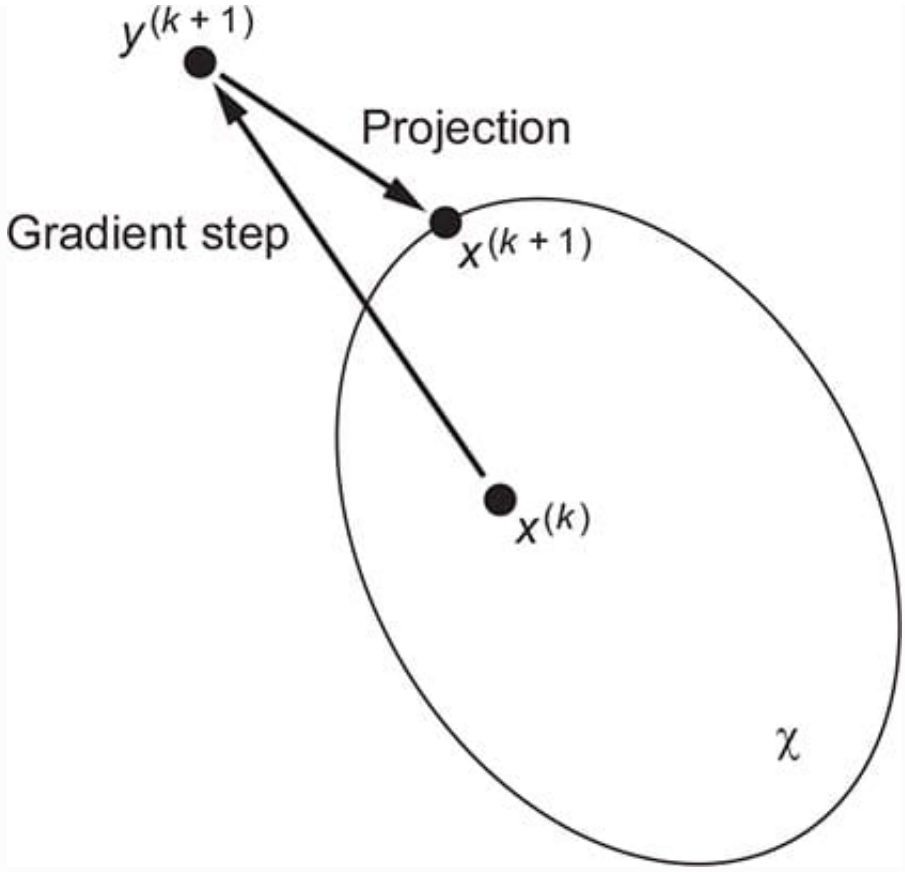

Figure 10.6。投影梯度下降无论在哪里，都朝最佳方向迈出了一步，然后使用投影来找到点集中最接近的等效点。在这种情况下，我们试图确保仍然得到有效的图片：我们以$x^{(k)}$为例，对$y^{(k+1)}$采取最佳步骤，然后将其投影到一组有效的图像x^{k+1}上。

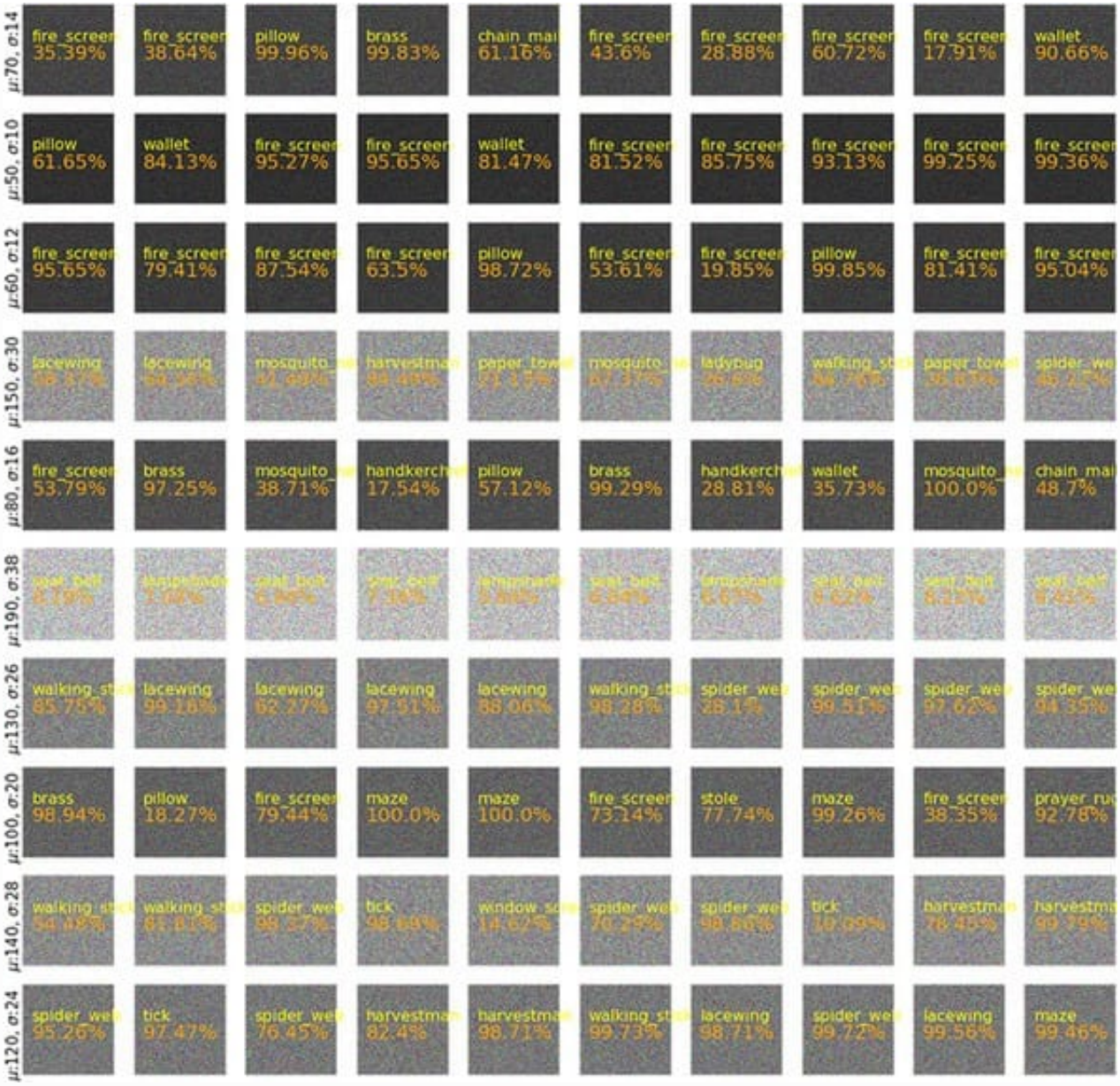

FIgure 10.7。当我们在对抗性噪声下运行ResNet-50时，我们得到了一个不同的结果：在应用PGD攻击后，大多数项目都被错误分类了,尽管 PGD 仍然是简单的攻击。

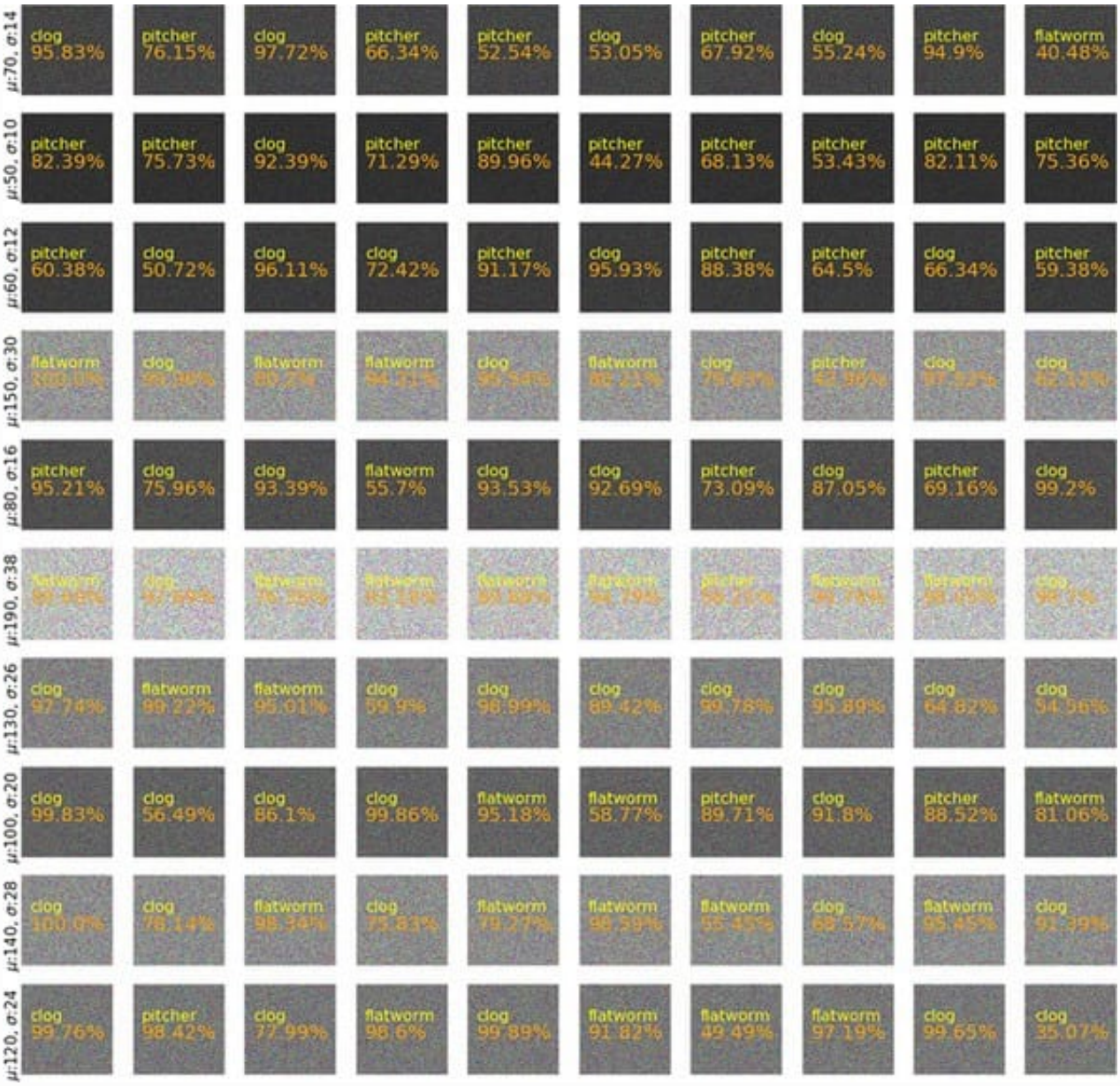

Figure 10.8. 应用高斯噪声于Inception-V3。注意，我们没有使用任何攻击。这些噪声只是从高斯分布中采样的。

为了证明大多数架构都更糟，我们将研究Inception-V3，该架构已在CV社区中非常有名。确实，该网络已经被认为是非常可靠的，以至于我们在第5章中进行了介绍。在Figure 10.8中，您可以看到，即使是产生初始得分的某些事情，在一些琐碎的例子中仍然失败。为了消除任何疑问，Inception-V3仍然是目前最好的预训练网络之一，并且确实具有超人的准确性。

> NOTE:这只是常规的高斯噪声。您可以自己在代码中看到没有应用对抗步骤。当然，您可能辩解道噪声得到了更好的预处理，但这仍然是巨大的对抗弱点。

如果您像我们一样想亲自看看。好吧，现在我们为您提供这些复现的代码。因为每个代码都是相似的，所以我们只经历一次，下一次保证使用DRYer代码。(译者注,源代码在 jupyter notebook,下面的代码可以直接跑)

> NOTE:For an explanation of *don't repeat yourself (DRY)* code, see Wikipedia at https://en.wikipedia.org/wiki/Don%27t_repeat_yourself.

```
# We get the standard Imports
import numpy as np
from keras.preprocessing import image as img
from keras.applications.resnet50 import preprocess_input,
decode_predictions, ResNet50
import foolbox
import pandas as pd
import keras

# 初始化我们的ResNet50预训练模型和DataFrame以将对象存储到
keras.backend.set_learning_phase(0)
kmodel = ResNet50(weights='imagenet')

mu_list = range(50, 200, 10)
sigma_list = range(10, 100, 2)
storage_df = pd.DataFrame()

# 在for循环中生成一堆均值和方差值
for mu, sigma in zip(mu_list, sigma_list):
  # 该特定均值的样本，以及该位置处的正常样本
  rand_noise = np.random.normal(loc=mu, scale=sigma, size=(224,224,
3))
  # 仅允许使用0–255像素值
  rand_noise = np.clip(preprocess_input(rand_noise), 0, 255.)
  # 得到我们的第一个预测
  noise_preds = kmodel.predict(np.expand_dims(rand_noise, axis=0))
  # 分别获取预测的类别和置信度
  predictions = decode_predictions(noise_preds, top=20)[0]
  # 将这些预测以数据帧格式存储
  new_df = pd.DataFrame(predictions, columns=
['id','class','prediction'])
  new_df['sigma'] = sigma
  new_df['mu'] = mu
  # 将此数据点添加到数据框
  storage_df = pd.concat([new_df, storage_df])


storage_df.to_csv('initialization_vals_for_noise.csv')
```

10.5. 并非所有希望都已消失

现在，有些人开始担心对抗性样本的安全性,但是更重要的是要保持对假想攻击者的有意义的认识。如果攻击者可以稍微更改每个像素，为什么不更改整个图像(^ 7)？为什么不传入一张完全不一样的图像呢？为什么传入的样本必须在感知上（而不是在视觉上）不同？

有些人举了无人驾驶汽车和对抗性停车标志的例子。但是如果我们能够做到这一点，为什么攻击者为什么不将喷漆完全涂在停车标志上，或者干脆用高速限速标志遮挡停车标志一会儿呢？因为与对抗性样本不同，这些“传统攻击”将在100%的时间内起作用，而对抗性攻击只有在信息传递良好且不受预处理失真的情况下才起作用。

这并不意味着当您拥有一个任务关键型机器学习应用程序时，就可以忽略此问题。但是在大多数情况下，对抗性攻击比更常见的攻击媒介需要更多的精力，因此记住这一点是值得的。

但是，与大多数安全隐患一样，对抗性攻击也具有对抗性防御，试图防御多种类型的攻击。本章介绍的攻击是一些较容易的攻击，但也存在更简单的攻击，例如通过MNIST绘制一条直线。即使这样也足以欺骗大多数分类器。 对抗性防御是一个不断发展的博弈，在这种博弈中，针对某些类型的攻击（但不是全部），可以使用许多良好的防御。但是世事无常，以至于在ICLR 2018提交截止日期后三天，八种经过审查并提案的防御中有七种就被打破了[7]。

10.6. GANs的缺点

为了使其与GAN的联系更加清晰，可以想象有一个生成对抗性样本的系统，用其中一个样本说明该样本有多好--取决于该样本是否成功欺骗了系统。这不是让您想起生成器（adversary）和判别器（分类算法）吗？这两种算法同时存在竞争：adversary试图以轻微的图像干扰来欺骗分类器，而分类器则试图不被愚弄。确实，思考GAN的方法几乎就像是ML循环式对抗性样本，最终提出了图像。

另一方面，您可以将迭代式对抗性攻击视为GAN，并不是为了是生成最现实的样本，而是为了使用生成器生成将使分类器蒙蔽的样本。当然，您必须始终记住存在重要的区别，并且通常在已部署的系统中具有固定的分类器。但这并不妨碍我们在对抗性训练中使用这种思想，在对抗性训练中，某些实现方法包括了基于欺骗它的对抗性样本对分类器进行重复训练。然后，这些技术越来越接近典型的GAN设置。举一个例子，让我们看一下已经有一段时间作为可行的防御技术的一种技术。在稳健的歧管防御中，我们采取以下步骤来防御对抗性示例[8]：

> 1. 我们选取一个真实图片 x (adversarial or regular)
>   1. 它投射到隐空间 z
>   2. 使用生成器 G 生成一个相似的样本 x,通过 G(z) 生成 x*
> 2. 我们使用分类器 C 得到生成样本的类别 C(x*),并使它偏离正确样本的趋势减小(which generally already tends to misclassify way less than running the classification directly on *x*.算了这个从句我是真的翻译不过来)

但是这种辩护仍然存在一些模棱两可的情况，在这种情况下，分类器确实会受到较小的干扰。不过我们还是鼓励您检查他们的论文，因为这些案例对于人类来说也不太清楚，这是模型可靠的标志。为了解决这个问题，我们在流形上进行对抗训练：我们将其中一些对抗情况纳入训练集中，以便分类器学会从实际训练数据中区分出来。

本文证明，使用GAN可以为我们提供分类器，即使经过某些最复杂的方法，分类器也不会在受到轻微扰动后完全崩溃。与大多数防御措施一样，下游分类器的性能确实会下降，因为现在必须对我们的分类器进行训练以隐式处理这些对抗性案件。但是，即使有这种挫折，它也不是普遍的防御。

当然,对抗性训练也有一些有趣的应用。例如，一段时间以来，通过对抗性训练，在半监督学习中取得了最好的成绩[9]（SOTA）。随后，这受到了GAN（请记住第7章）和其他方法的挑战，但这并不意味着在您阅读这些内容时，对抗性训练将不再是最新的技术。

这里我们希望这为您提供了研究GAN和对抗性样本的另一个理由-部分原因是在关键任务分类任务中，GAN可能是前进的最佳防御方法，或者是因为本书不涉及的其他应用。这是一个假设的对抗实例[10]。

综上所述，我们提出了对抗性样本的概念，并使与GAN的联系更加具体。这是一种未被充分认识的联系，但可以巩固您对这一具有挑战性的主题的理解。此外，防御对抗性样本的一种防御措施就是GAN本身[11]！因此，GAN也有潜力来解决这这个最初的差异.

10.7. 结论

对抗性样本是一个重要领域，因为即使商用计算机视觉产品也遭受了这一缺点，并且仍然容易被学术研究者愚弄[12]。除了安全性和机器学习可解释性应用程序之外，公平性和鲁棒性仍然有许多实际用途。

此外，对抗性样本是巩固您自己对深度学习和GAN的理解的绝佳方法。对抗性样本利用训练分类器一般情况下的缺点以及在一些特定情况下愚弄分类器的相对优势。分类器必须对许多图像进行预测，并且由于存在许多自由度，因此可以轻松制作一个特殊的偏移量来使分类器完全正确。结果，我们可以很容易地获得对抗噪声，该噪声完全改变了图片的标签，而不会明显改变图像。

对抗性样本可以在AI的许多领域和许多领域中找到，而不仅仅是深度学习或计算机视觉。但是，正如您在代码中看到的那样，在计算机视觉中创建代码并不困难。存在针对这些示例的防御措施，您已经看到了使用GAN的示例，但对抗性样本的问题远未完全解决。

总结

- 滥用问题空间的维度是产生的对抗性样本是机器学习的重要原因，因为它们向我们展示了GAN为何起作用以及为什么某些分类器很容易被破坏。
- 我们可以轻松地生成带有真实图像和噪声的对抗样本。
- 微小的有意义的攻击向量就可以作为对抗性样本。
- 对抗性样本的应用包括网络安全和机器学习公平性，我们可以使用GAN来抵御它们。

参考文献

1. See "Intriguing Properties of Neural Networks," by Christian Szegedy et al., 2014, https://arxiv.org/pdf/1312.6199.pdf↩

2. What constitutes human-level performance in vision-classification tasks is a complicated topic. However, at least in, for example, Dota 2 and Go, AI has beat human experts by a substantial margin.↩

3. A receiver operating characteristic (ROC) curve explains the trade-offs between false positives and negatives. We also encountered them in chapter 2. For more details, Wikipedia has an excellent explanation.↩

4. See "Adversarial Attacks on Deep Learning Models in Natural Language Processing: A Survey," by Wei Emma Zhang et al., 2019, http://arxiv.org/abs/1901.06796. See also "Adversarial Examples That Fool Both Computer Vision and Time-Limited Humans," by Gamaleldin F. Elsayed et al., 2018, http://arxiv.org/abs/1802.08195.↩

5. Please remember, this is just a quick summary, and we have to skip over some details, so if you can point them out—great. If not, we suggest picking up a book such as *Deep Learning with Python* by François Chollet (Manning, 2017) to brush up on the specifics.↩

6. See "Image Classification on ImageNet," at DAWNBench, https://dawn.cs.stanford.edu/benchmark/#imagenet.↩

7. ICLR is the *International Conference on Learning Representations*, one of the smaller but excellent machine learning conferences. See Anish Athalye on Twitter in 2018, http://mng.bz/ad77. It should be noted that there were three more defenses unexamined by the author.↩

8. See "The Robust Manifold Defense: Adversarial Training Using Generative Models," by Ajil Jalal et al., 2019, https://arxiv.org/pdf/1712.09196.pdf.↩

9. See "Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning," by Takeru Miyato et al., 2018, https://arxiv.org/pdf/1704.03976.pdf.↩

10. This was a hotly debated topic at ICLR 2019. Though most of these conversations were informal, using (pseudo) invertible generative models as a way to classify "out-of-sample"ness of an image seems like a fruitful avenue.↩

11. See Jalal et al., 2019, https://arxiv.org/pdf/1712.09196.pdf.↩

12. See "Black-Box Adversarial Attacks with Limited Queries and Information," by Andrew Ilyas et al., 2018, https://arxiv.org/abs/1804.08598.↩