**Mixup : BEYOND EMPIRICAL RISK MINIMIZATION**

会议：ICLR2018

代码：<https://github.com/facebookresearch/mixup-cifar10>.

**ABSTRACT**

大型深层神经网络功能强大，但表现出不尽如人意的行为，例如记忆和对对抗示例的敏感性。 在这项工作中，我们提出了***mixup***，一种简单的学习原理来缓解这些问题。本质上，***mixup***在一对示例及其标签的凸组合上训练了一个神经网络。通过这样做，***mixup***使神经网络规则化，从而在训练示例之间倾向于简单的线性行为。 我们在ImageNet-2012，CIFAR-10，CIFAR-100，Google命令和UCI数据集上进行的实验表明，mixup提高了最新的神经网络体系结构的通用性。 我们还发现，***mixup***减少了损坏标签的记忆，提高了对抗示例的鲁棒性，并稳定了生成对抗网络的训练。

**1 INTRODUCTION**

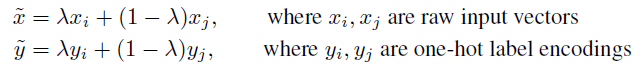
大型深度神经网络在计算机视觉（Krizhevsky等，2012），语音识别（Hinton等，2012）和强化学习（Silver等，2016）等领域取得了突破。在大多数成功的应用中，这些神经网络具有两个共同点。 首先，对他们进行训练以使其在训练数据上的平均误差最小，这是一种学习规则，也称为经验风险最小化（ERM）原理（Vapnik，1998年）。 其次，这些最新的神经网络的规模与训练示例的数量呈线性比例关系。 例如，Springenberg等人的网络（2015）使用10^6个参数对CIFAR-10数据集中的5X10^4张图像进行建模，（Simonyan＆Zisserman，2015）的网络使用108个参数对ImageNet-2012数据集中的10^6张图像进行建模，并使用Chelba网络等（2013年）使用2个10^10参数对One Billion Word数据集中的10^9个字进行建模。

令人惊讶的是，学习理论的经典结果（Vapnik＆Chervonenkis，1971）告诉我们，只要学习机器（例如神经网络）的大小不随训练数据的数量而增加，就可以保证ERM的收敛性。 在这里，学习机的大小是根据其参数数量或相关的VC复杂性来衡量的（Harvey等人，2017）。

正如最近的研究所强调的那样，这一矛盾挑战了ERM是否适合训练我们当前的神经网络模型。 一方面，即使在强正则化的情况下，或者在标签随机分配的分类问题中，ERM仍允许大型神经网络记住训练数据（而不是从训练数据中泛化）（Zhang et al。，2017）。 另一方面，用ERM训练的神经网络在训练分布之外的示例（Szegedy等人，2014）（也称为对抗性示例）上进行评估时，其预测会急剧改变。 这些证据表明，ERM无法解释或提供与培训数据仅略有不同的测试分布的概括。 但是，ERM的替代方法是什么？

选择与训练数据相似但不同的示例进行训练的选择方法称为数据增强（Simard等，1998），该方法已通过Vicinal Risk Minimization（VRM）原则正式化（Chapelle等，2000）。 在VRM中，需要人类知识来描述训练数据中每个示例周围的邻域。 然后，可以从训练示例的附近分布中提取其他虚拟示例，以扩大对训练分布的支持。 例如，在执行图像分类时，通常将一个图像的附近定义为其水平反射，轻微旋转和轻微缩放的集合。 尽管数据扩充始终可以提高泛化性（Simard等，1998），但该过程是依赖于数据集的，因此需要使用专家知识。 此外，数据扩充假设附近的示例共享同一类，并且不会跨不同类的示例对附近关系建模。

贡献 受这些问题的影响，我们引入了一个简单且与数据不可知的数据增强例程，称为mixup（第2节）。 简而言之，mixup构建了虚拟培训示例



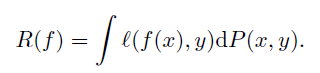
（xi; yi）和（xj; yj）是从我们的训练数据中随机抽取的两个示例，其中。 因此，mixup通过合并现有的知识（特征向量的线性插值应导致相关目标的线性插值）来扩展训练分布。 mixup可以用几行代码来实现，并且引入了最小的计算开销。

尽管它很简单，但是它可以在CIFAR-10，CIFAR100和ImageNet-2012图像分类数据集（第3.1和3.2节）中提供最新的最新性能。 此外，当从损坏的标签中学习（第3.4节）或面对对抗示例时（第3.5节），mixup提高了神经网络的鲁棒性。 最后，mixup提高了语音（第3.3节）和表格（第3.6节）数据的泛化能力，可用于稳定GAN的训练（第3.7节）。

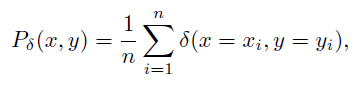
为了了解各种设计选择对mixup的影响，我们进行了一套完整的消融研究实验（第3.8节）。 结果表明，mixup效果比以前的工作中的相关方法要好得多，并且每种设计选择都有助于最终性能。 最后，我们探索与先前工作的联系（第4节），并提供一些讨论要点（第5节）。

**2 FROM EMPIRICAL RISK MINIMIZATION TO mixup**

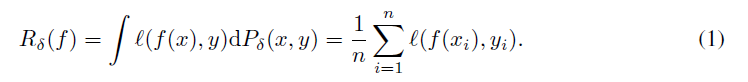
在监督学习中，我们感兴趣的是找到一个函数，该函数描述遵循联合分布P（X, Y）的随机特征向量X和随机目标向量Y之间的关系。 为此，我们首先定义一个损失函数，它会惩罚预测f（x）与实际目标y之间的差异，例如。然后，将数据分布P上损失函数的平均值最小化，也称为预期风险：



不幸的是，在大多数实际情况下分布P是未知的。 相反，我们通常可以访问一组训练数据，其中对所有的（xi; yi）。 使用训练数据D，我们可以通过经验分布来近似P

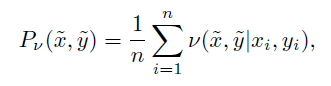


其中（x = xi; y = yi）是一个以（xi; yi）为中心的狄拉克质量。 使用经验分布P，我们现在可以通过经验风险来近似预期风险：

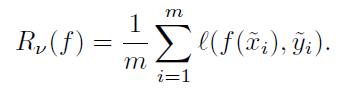


通过最小化（1）学习函数f被称为经验风险最小化（ERM）原理（Vapnik，1998）。 尽管计算效率很高，但经验风险（1）仅在有限的n个示例集合上监视f的行为。 当考虑具有与n相当的数量参数的函数（例如大型神经网络）时，最小化（1）的一种简单方法是记忆训练数据（Zhang et al。，2017）。 记忆反过来会导致f在训练数据之外的不良行为（Szegedy等，2014）。

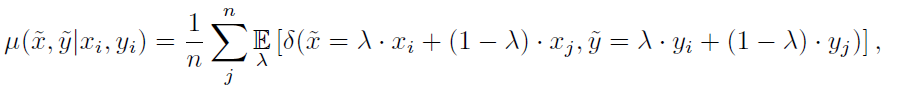
但是，朴素的估计值是逼近真实分布P的众多可能选择之一。例如，在“最小风险”原则（Chapelle et al。，2000）中，分布P近似为



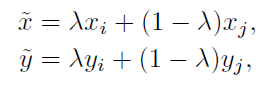
其中为邻域分布，用于度量在训练特征目标对（xi，yi）附近找到虚拟特征目标对的可能性。 特别是Chapelle等（2000年）认为高斯近邻，这相当于用加性高斯噪声来扩充训练数据。 要学习使用VRM，我们对邻域分布进行抽样，以构建数据集，并最大程度地减少经验上邻近的风险：

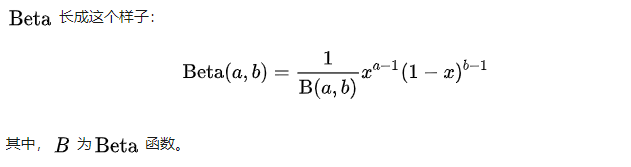


本文的作用是提出一种通用的邻域分布，称为mixup：



其中。 简而言之，从mixup邻域分布中采样会生成虚拟特征目标向量





其中（xi; yi）和（xj; yj）是从训练数据中随机绘制的两个特征目标向量，而。 mixup超参数控制特征对之间的插值强度，时ERM原理恢复为。

mixup训练的实现很简单，并且引入了最小的计算开销。 图1a显示了在PyTorch中实施mixup训练所需的几行代码。最后，我们提到了替代设计选择。 首先，在初步实验中，我们发现三个或更多示例的凸组合与从Dirichlet分布中采样的权重不会提供进一步的收益，而是会增加mixup的计算成本。 其次，我们当前的实现使用单个数据加载器来获取一个小批量，然后在随机混洗之后将mixup应用于同一小批量。 我们发现该策略同样有效，同时减少了I / O需求。第三，仅在具有相等标签的输入之间进行插值并不会导致在续篇中讨论的mixup性能提高。 可以在第3.8节中找到更多的经验比较。

*# y1, y2 should be one-hot vectors*

***for*** *(x1, y1), (x2, y2)* ***in zip****(loader1, loader2):*

*lam = numpy.random.beta(alpha, alpha)*

*x = Variable(lam \* x1 + (1. - lam) \* x2)*

*y = Variable(lam \* y1 + (1. - lam) \* y2)*

*optimizer.zero\_grad()*

*loss(net(x), y).backward()*

*optimizer.step()*

*(a) One epoch of mixup training in PyTorch.*

**mixup在做什么？** mixup邻域分布可以理解为一种数据增强形式，它鼓励模型f在训练示例之间线性表现。 我们认为，当在训练样本之外进行预测时，这种线性行为会减少不希望有的振荡。 而且，从奥卡姆剃须刀的角度来看，线性度是一个很好的感应偏差，因为这是最简单的行为之一。 图1b显示，mixup导致决策边界在类之间线性过渡，从而提供了更平滑的不确定性估计。图2说明了使用ERM和mixup在CIFAR-10数据集上训练的两个神经网络模型的平均行为。 两种模型具有相同的架构，使用相同的程序进行训练，并在随机采样的训练数据之间的相同点进行评估。 就训练样本之间的模型预测和梯度规范而言，使用mixup训练的模型更稳定。

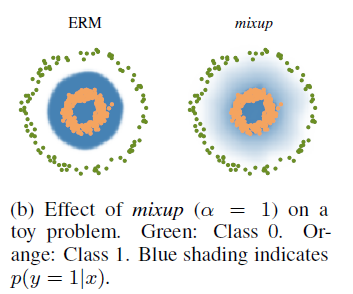
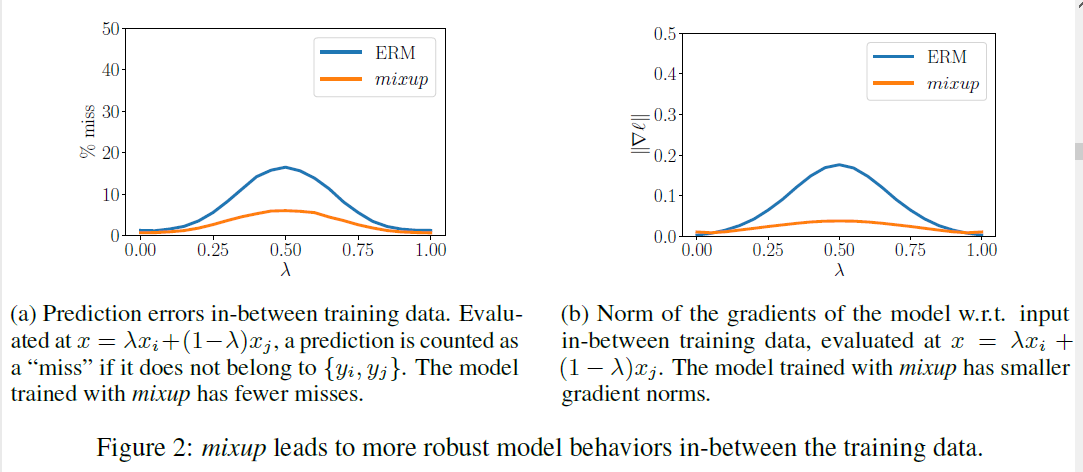


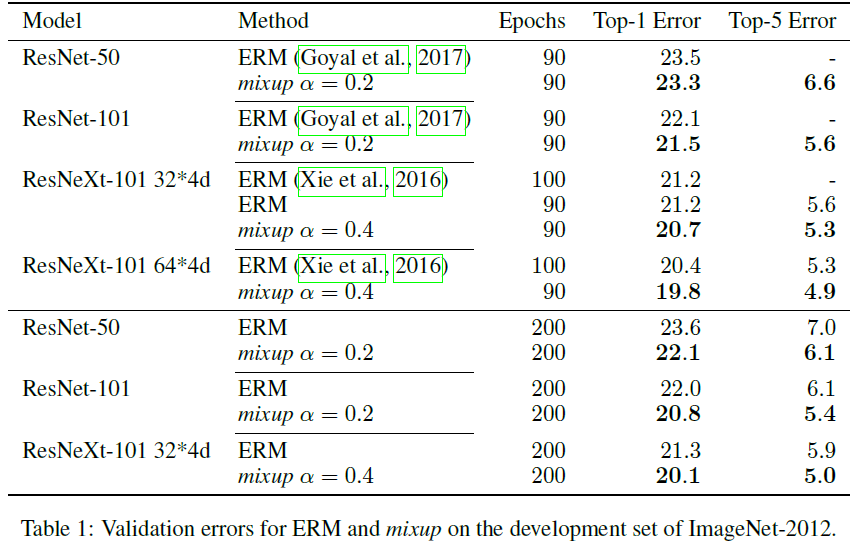
图1：mixup说明图，收敛为ERM。



**3实验**

**3.1 IMAGENET分类**

我们在ImageNet-2012分类数据集上评估mixup（Russakovsky等人，2015）。 该数据集包含总共1,000个类别的130万个训练图像和50,000个验证图像。 在训练方面，我们遵循标准的数据增强实践：比例和纵横比失真，随机裁剪和水平翻转（Goyal等人，2017）。 在评估期间，仅测试每个图像的224X224中心。 我们使用mixup和ERM训练了几种最新的ImageNet-2012分类模型，并在表1中报告了top-1和top-5错误率。



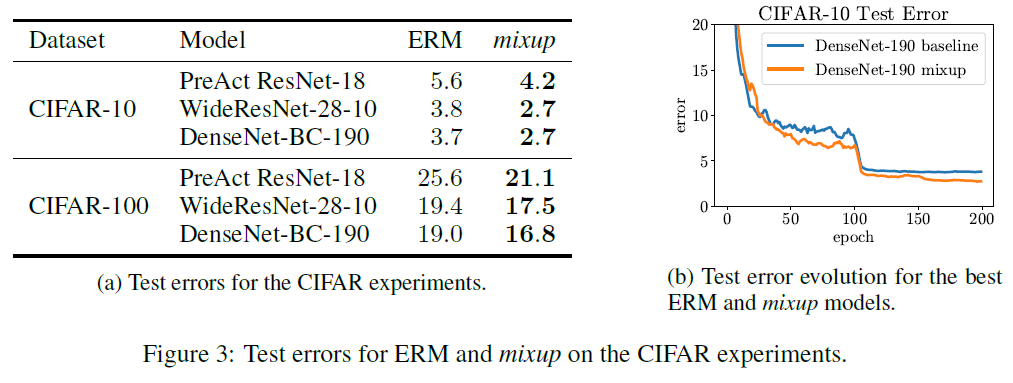
对于本节中的所有实验，我们在Caffe2中使用数据并行分布式训练，最小批量大小为1,024。 我们使用（Goyal et al。，2017）中描述的学习率时间表。 具体地说，学习率在前5个时期内从0.1线性增加到0.4，然后在训练90个时期后在30、60和80个时期后除以10。 或训练60个时期后的60、120和180个时期。

对于mixup，我们发现相对于ERM可以提高性能，而对于大，mixup会导致拟合不足。我们还发现，具有更高能力和/或更长训练时间的模型是从mixup中受益最大的模型。 例如，经过90个纪元的训练后，ResNet-101和ResNeXt-101的mixup变体比其较小的模型（如ResNet-50）（0.2％）获得了比其ERM类似物更大的改进（0.5％至0.6％）。 经过200个纪元的训练后，与90个纪元的运行相比，ResNet-50的mixup变量的top-1错误进一步降低了1.2％，而其ERM类似物保持不变。

**3.2 CIFAR-10 AND CIFAR-100**

我们在CIFAR-10和CIFAR-100数据集上进行了其他图像分类实验，以进一步评估mixup的泛化性能。 特别是，我们比较了（Liu，2017）实施的PreAct ResNet-18（He等人，2016），（Zagoruyko＆Komodakis，2016a）的WideResNet28-10（Zagoruyko＆Komodakis，2016a）的ERM和mixup培训。 （2016b），以及（Veit，2017）中实施的DenseNet（Huang等，2017）。 对于DenseNet，我们将增长率更改为40以遵循（Huang等人，2017）的DenseNet-BC-190规范。 对于mixup，我们固定= 1，这将导致插值均匀地分布在零和一之间。 所有模型都在单个Nvidia Tesla P100 GPU上使用PyTorch2在训练集上进行了200个时期的训练，每个小批量处理有128个示例，并在测试集上进行了评估。 对于所有型号，除了WideResNet，学习率从0.1开始并在100和150个时期后除以10。 对于WideResNet，我们遵循（Zagoruyko＆Komodakis，2016a），在60、120和180个时期之后将学习率除以10。 重量衰减设置为10􀀀4。 在这些实验中，我们不使用辍学。

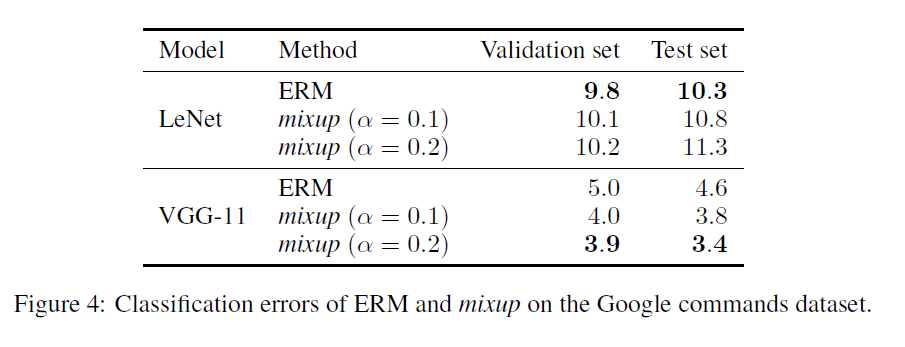
我们在图3a中总结了我们的结果。 在CIFAR-10和CIFAR-100分类问题中，使用mixup训练的模型都大大优于使用ERM训练的模型。 如图3b所示，mixup和ERM收敛速度接近最佳测试误差。 请注意，（Huang等人，2017）中的DenseNet模型训练了300个纪元，并在150和225个纪元计划了进一步的学习率衰减，这可能解释了图3a中DenseNet的性能与黄等（2017）原始结果的差异。



**3.3 SPEECH DATA**

接下来，我们使用Google命令数据集执行语音识别实验（Warden，2017）。 数据集包含65,000个语音，其中每个语音大约一秒钟长，属于30个类别中的一个。 这些类别对应于语音命令，例如“是”，“否”，“向下”，“左”，由数千个不同的说话者发音。 为了对语音进行预处理，我们首先从原始波形中提取归一化的频谱图，采样率为16 kHz。 接下来，我们在160X101处对频谱图进行零填充以使其大小相等。对于语音数据，在波形和频谱图级别同时应用mixup是合理的。 在这里，我们在将数据馈送到网络之前在频谱图级别应用mixup。

对于本实验，我们比较了LeNet（Lecun等人，2001）和VGG-11（Simonyan＆Zisserman，2015）架构，它们每个都由两个卷积层和两个完全连接的层组成。 我们使用Adam作为优化器，用100个示例的小批量训练每个模型30个纪元（Kingma＆Ba，2015）。 训练以等于3X10-3的学习率开始，然后每10个周期除以10。 对于mixup，我们使用五个纪元的预热期，在此期间我们以原始训练示例训练网络，因为我们发现它可以加快初始收敛。 表4显示，在此任务上，mixup性能胜过ERM，尤其是在使用VGG-11（容量更大的型号）时



**3.4 MEMORIZATION OF CORRUPTED LABELS**

跟随张等（2017），我们针对随机损坏的标签评估了ERM和mixup模型的鲁棒性。 我们假设增加mixup插值的强度应从训练示例中进一步生成虚拟示例，从而使记忆更加难以实现。特别是，与记忆涉及随机标签的插值相比，学习实际示例之间的插值应该更容易。 我们采用开放源代码实施方式（Zhang，2017），以生成三个CIFAR-10训练集，其中20％，50％或80％的标签分别被随机噪声代替。 所有测试标签均保持完整以进行评估。 dropout（Srivastava et al。，2014）被认为是学习带有损坏标签的最新方法（Arpit et al。，2017）。 因此，我们在这些实验中比较mixup，dropout，mixup+ dropout和ERM。 对于mixup，我们选择; 对于dropout，我们在两个卷积层之间的ReLU激活层之后的每个PreAct块中添加了一个dropout层，如（Zagoruyko＆Komodakis，2016a）中所建议。 我们选择dropout概率。 对于mixup和dropout的组合，我们选择。 这些实验使用在（Liu，2017）中实现的PreAct ResNet-18模型（He et al。，2016）。 所有其他设置与第3.2节中的相同。

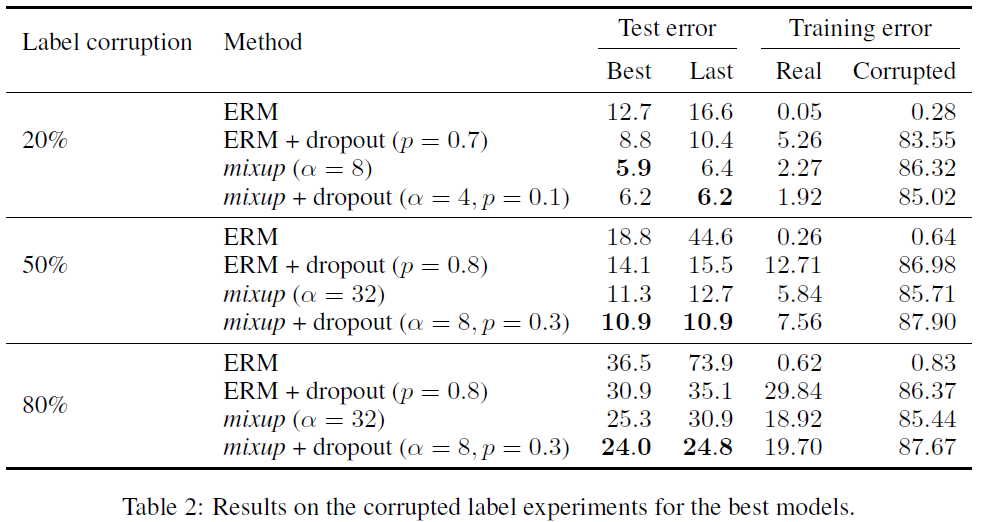
我们在表2中总结了我们的结果，其中我们注意到了在训练期间实现的最佳测试错误，以及200个时期后的最终测试错误。 为了量化记忆量，我们还评估了在最后一个时期对真实标签和损坏标签的训练错误。 随着训练以较小的学习率（例如小于0.01）进行，ERM模型开始过度拟合损坏的标签。 当使用大概率（例如0.7或0.8）时，辍学可以有效地减少过度拟合的情况。 具有较大（例如8或32）的mixup效果优于最佳和最后一个时期dropout测试错误，并且在保持对噪声标签的抵抗力的同时，在真实标签上实现了较低的训练错误。有趣的是，mixup + dropout表现最佳，表明这两种方法是兼容的。

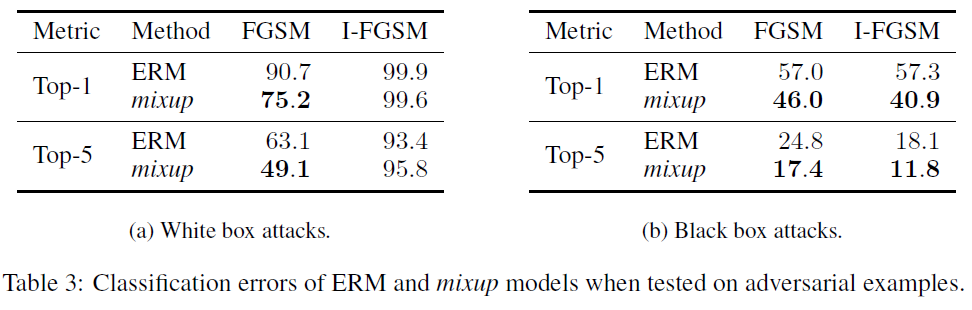
**3.5鲁棒性的示例**

使用ERM训练的模型的不良后果之一是其对对抗性示例的脆弱性（Szegedy等，2014）。 对抗性示例是通过在合法示例中添加微小（视觉上难以察觉）的扰动来获得的，从而降低了模型的性能。 相对于合法示例，通过增大损失表面的坡度会产生对抗性噪声。 提高对抗性示例的鲁棒性是积极研究的主题。

在旨在解决该问题的几种方法中，有人提出对模型的Jacobian范数进行惩罚以控制其Lipschitz常数（Drucker＆Le Cun，1992; Cisse等人，2017; Bartlett等人，2017; Hein和Andriushchenko，2017年）。 其他方法通过产生和训练对抗性示例来执行数据增强（Goodfellow等人，2015）。 不幸的是，所有这些方法都给ERM增加了可观的计算开销。 在这里，我们表明mixup可以通过惩罚沿着最合理的方向（例如到其他训练点的方向）的给定输入的损耗梯度的范数来显着提高神经网络的鲁棒性，而不会妨碍ERM的速度。 实际上，图2显示，与原始ERM相比，mixup后的模型在示例之间的损失和梯度范数较小。

为了评估mixup模型对付对抗性示例的鲁棒性，我们使用了三种ResNet-101模型：其中两种模型在ImageNet-2012上使用ERM进行了训练，而第三种使用mixup在模型上进行了训练。 在第一组实验中，我们研究了一个ERM模型和mixup模型对白盒攻击的鲁棒性。 也就是说，对于这两个模型中的每一个，我们都使用模型本身来生成对抗性示例，无论是使用快速梯度符号方法（FGSM）还是迭代FGSM（I-FGSM）方法（Goodfellow等，2015）， 每个像素的最大摄动。 对于I-FGSM，我们使用10个步长相等的迭代。 在第二组实验中，我们评估了针对黑盒攻击的鲁棒性。 也就是说，我们使用第一个ERM模型使用FGSM和I-FGSM生成对抗性示例。 然后，我们针对这些示例测试第二种ERM模型和mixup模型的鲁棒性。 表3汇总了这两种设置的结果。

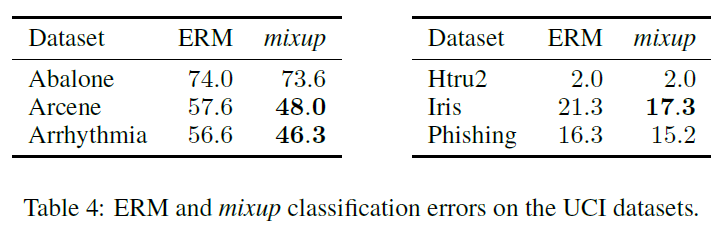




对于FGSM白盒攻击，就Top-1错误而言，mixup模型比ERM模型具有2：7倍的鲁棒性。 对于FGSM黑匣子攻击，就Top-1错误而言，mixup模型比ERM模型具有1:25倍的鲁棒性。 同样，虽然mixup和ERM都无法抵抗白盒I-FGSM攻击，但是在黑盒I-FGSM设置中，mixup的抵抗能力比ERM高40％。 总体而言，对于EMR而言，在白盒和黑场环境下对抗示例时，mixup产生的神经网络比ERM更为强大。

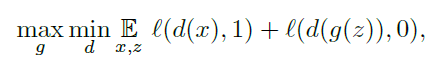
**3.6管状数据**

为了进一步探索mixup对非图像数据的性能，我们对从UCI数据集得出的六个任意分类问题进行了一系列实验（Lichman，2013）。 本节中的神经网络是完全连接的，具有两个隐藏的层，其中包含128个ReLU单元。 这些神经网络的参数是使用Adam（Kingma和Ba，2015年）使用默认的超参数来学习的，它们的大小为16个微型批次的10个历元。表4显示，mixup提高了六个数据集中四个的平均测试误差，并且绝不逊色于ERM。



**3.7 STABILIZATION OF GENERATIVE ADVERSARIAL NETWORKS (GANS)**

生成对抗网络，也称为GAN（Goodfellow等，2014），是强大的隐式生成模型家族。 在GAN中，生成器和鉴别器相互竞争以对分布P建模。一方面，生成器g竞争将噪声矢量转换为类似于真实样本的伪样本g（z）。 另一方面，鉴别器竞争区分真实样本x和假样本g（z）。 从数学上讲，训练GAN相当于解决优化问题



其中是二进制交叉熵损失。 不幸的是，解决先前的最小-最大方程是一个众所周知的困难的优化问题（Goodfellow，2016年），因为鉴别器通常会为发生器提供消失的梯度。 我们认为mixup应该使GAN训练稳定，因为它可以充当鉴别器梯度的正则化器，类似于图1b中的二元分类器。然后，鉴别器的平滑性保证了向发生器产生稳定的梯度信息源。GAN的mixup配方为：



图5说明了对两个玩具数据集（蓝色样本）建模时GAN（橙色样本）训练mixup的稳定作用。 这些实验中的神经网络是完全连接的，具有512个ReLU单元的三个隐藏层。 发生器网络接受二维高斯噪声矢量。 使用带有默认参数的Adam优化器对网络进行20,000个大小为128的迷你批次的训练，其中，在每次生成器迭代之前，对鉴别器进行五次迭代训练。 mixupGAN的训练对于超参数和体系结构选择似乎很有希望。

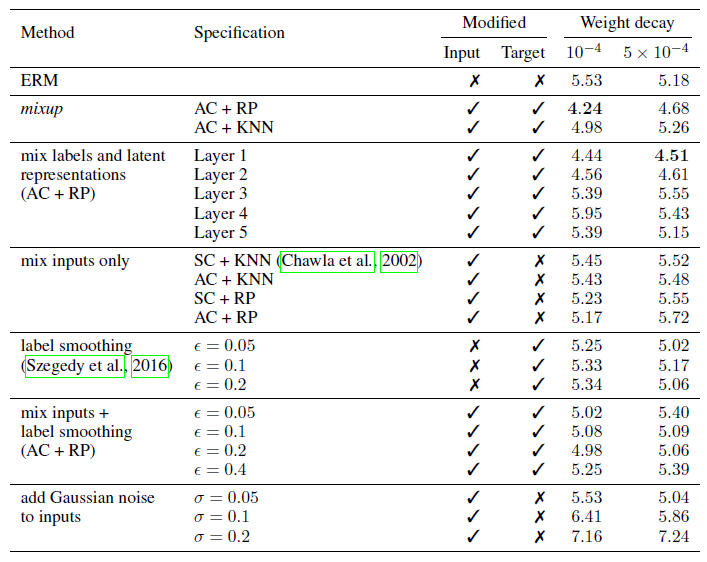


表5：在CIFAR-10数据集上的消融研究结果。 报告的是最近10个时期的中值测试错误。 勾号表示该组件与标准ERM培训不同，而叉号表示其遵循标准培训做法。 AC：在所有班级之间mixup使用。 SC：在同一个班级内混用。 RP：随机对之间mixup。 KNN：k个最近邻居之间的mixup（k = 200）。 有关实验和解释的详细信息，请参阅课文。

**3.8 ABLATION STUDIES**

mixup是一种数据增强方法，仅包括两部分：原始输入的随机凸组合，以及相应地，一热标签编码的凸组合。 但是，有几种设计选择。 例如，关于如何增加输入，我们可以选择对神经网络的潜在表示（即特征图）进行插值，并且可以选择仅在最近的邻居之间或仅在相同类别的输入之间进行插值 。当要进行插值的输入来自两个不同的类别时，我们可以选择为合成输入分配单个标签，例如，使用凸组合中权重更大的输入标签。 为了将mixup与这些替代可能性进行比较，我们在CIFAR-10数据集上使用PreAct ResNet-18架构运行了一组烧蚀研究实验。

具体来说，对于每种数据增强方法，我们测试了两个权重衰减设置（10^-4对于mixup很有效，而5X10^-4对于ERM则很有效）。 所有其他设置和超参数与3.2节中报告的相同。

为了将插值原始输入与插值潜在表示进行比较，我们在每个残差块（表示为1-4层）之前或最上方的“平均池+完全连接”层（表示为第5层）之前，对学习表示的随机凸组合进行测试。 为了比较mixup的随机输入对（RP）与mixup的最近邻居（KNN），我们首先为每个训练样本计算200个最近的邻居，这些样本是来自同一类别（SC）或来自所有类别（AC）。 然后在训练期间，对于小批量中的每个样本，我们通过凸组合并从其最近邻居随机抽取来用合成样本替换样本。 为了比较所有类别（AC）的mixup与同一类别（SC）内的mixup，我们将小批量凸面组合它的样本索引的随机排列，其中排列是基于每个批次（AC）或每个类别（SC）进行的。 为了仅将mixup输入和标签与mixup输入进行比较，我们要么将两个单热编码的凸组合用作目标，要么选择更紧密训练样本的单热编码作为目标。 对于标签平滑，我们遵循Szegedy等（2016），并使用作为错误类别的目标，并使用作为正确类别的目标。将高斯噪声添加到输入中用作另一个基准。 我们报告了最近10个时期的中值测试错误。结果示于表5。

从消融研究实验中，我们有以下观察。 首先，mixup是我们测试的最佳数据扩充方法，并且明显优于第二最佳方法（mixup输入+标签平滑）。 其次，通过将测试误差与较小的权重衰减（10􀀀4）与较大的权重衰减（510􀀀4）进行比较，可以看到正则化的效果。 例如，对于ERM，较大的重量衰减效果更好，而对于mixup而言，较小的重量衰减更可取，从而证实了其正则化效果。当在较高层的潜在表示中进行插值时，我们还看到了较大的权重衰减的增加优势，这表明正则化强度降低。 在所有输入插值方法中，mixup来自所有类别（AC + RP）的随机对具有最强的正则化效果。标签平滑和添加高斯噪声的正规化效果相对较小。 最后，我们注意到SMOTE算法（Chawla等人，2002）并没有导致明显的性能提升。

**4 RELATED WORK**

数据增强是深度学习所有成功应用的核心，从图像分类（Krizhevsky等，2012）到语音识别（Graves等，2013； Amodei等，2016）。 在所有情况下，都将充分利用领域知识来设计合适的数据转换，从而提高通用性。 例如，在图像分类中，通常会使用旋转，平移，裁剪，调整大小，翻转（Lecun等人，2001； Simonyan＆Zisserman，2015）和随机擦除（Zhong等人，2017）来实施视觉上合理的不变性。 在模型中通过训练数据。 同样，在语音识别中，噪声注入是提高训练模型的鲁棒性和准确性的普遍做法（Amodei等，2016）。

Chawla等人（2002年）与混搭更相关,建议通过内插最近的邻居来增加不平衡数据集中的稀有类别。 DeVries＆Taylor（2017）表明，在特征空间中对同一类别的最近邻居进行插值和外推可以提高泛化能力。 但是，它们的建议仅在输入/要素级别的特定类别中的最近邻居之间运行，因此不考虑相应标签的更改。 最近的方法还提出了通过标签平滑（Szegedy等人，2016）或惩罚高置信度softmax分布（Pereyra等人，2017）来规范化神经网络输出分布的方法。 这些方法与mixup具有相似之处，因为监督取决于多个平滑标签，而不是像传统ERM中那样依赖单个硬标签。 但是，这些作品中的标签平滑处理是独立于关联特征值应用或调整的。

mixup享有以前的数据扩充和正则化方案的几个理想方面，而没有缺点。 就像DeVries＆Taylor（2017）的方法一样，它不需要大量的领域知识。 像标签平滑一样，每个示例的监督也不会过度依赖于真实标签。 与这两种方法不同，mixup变换在数据增强和监督信号之间建立了线性关系。 我们相信，这会导致强大的正则化器，从而改善我们的实验所证明的泛化性。线性约束通过其对近似函数导数的影响，还将mixup与其他方法相关，例如神经网络的Sobolev训练（Czarnecki等，2017）或WGAN-GP（Gulrajani等，2017）。

**5 DISCUSSION**

我们提出了mixup，一种与数据无关的直接数据增强原理。 我们已经表明，mixup是最小化附近风险的一种形式，它可以对虚拟样本进行训练，该虚拟样本构造为来自训练集及其标签的两个随机样本的线性插值。将mixup合并到现有的训练流水线中可以减少几行代码，并且几乎没有或根本没有计算开销。 在广泛的评估中，我们证明了这种mixup, 改善了ImageNet，CIFAR，语音和表格数据集上最新模型的泛化误差。 此外，mixup有助于对抗损坏标签的记忆，对抗示例的敏感性以及对抗训练的不稳定。

在我们的实验中，以下趋势是一致的：随着大小的增加，对真实数据的训练误差会增加，而泛化差距会减小。 这支持了我们的假设，即mixup隐含地控制模型的复杂性。 但是，我们尚无一个很好的理论来理解这种偏差－偏差权衡的“最佳点”。 例如，在CIFAR-10分类中，即使（即，仅对成对的真实示例进行训练），而在ImageNet分类中，对真实数据的训练错误会随着.根据我们使用ImageNet和Google命令进行的不同模型架构的实验，我们推测增加模型容量将使训练误差对大型对象的敏感性降低，从而使mixup具有更大的优势。

mixup还为进一步的探索提供了多种可能性。 首先，是否有可能使类似的想法适用于其他类型的监督学习问题，例如回归和结构化预测？ 虽然将mixup概括为回归问题很简单，但将其应用于结构化预测问题（例如图像分割）的情况仍然不太明显。 第二，类似的方法在监督学习之外还能证明有用吗？ 插值原理似乎是合理的归纳偏差，它也可能有助于无监督，半监督和强化学习。 我们是否可以将mixup扩展到特征标签外推，以确保可靠的模型行为远离训练数据？ 尽管我们对这些方向的讨论仍是推测性的，但我们对mixup可能性打开的可能性感到兴奋，并希望我们的观察结果将对未来的发展有用。