# Improvements on CNNs

## Convolutional layer

卷积滤波器是一个针对底层图像块的广义线性模型，在潜在概念的实例是线性可分的时候抽象化效果非常好。以下是一些提高其表达能力的工作。

#### Tiled Convolution

CNN中的权值共享可以大大减少参数的数量，并且可以提供一定的平移不变性，但是在其他不变性上，模型容易被约束。Tiled Convolution是一种能够平铺和并联来学习旋转和缩放不变性特征的CNN变种。

在普通的CNN中，对于同一层同一个特征映射，对不同感受野的权值是共享的，而在TCNN中依旧保留了局部感知的特点，但只有间隔为（人为设定）的感受野会共享权值，其他情况均相互独立。如果等于1，那么将退化为传统的CNN。

在NORB和CIFAR-10数据集上，实践表现当时结果最佳，且在小型时间序列数据集上比传统CNN表现更佳。

#### Transposed Convolution

反卷积可以被看作传统卷积的反向传播过程，也被称作或者。大多数文献使用的称呼。和传统卷积层将多个输入与卷积层的一个输出相连接不同，反卷积将一个输入与多个输出相联。相当于输入特征映射，还原原图像。反卷积的步长给了特征映射一个放大因子。特别的，反卷积会先对输入基于步长和边距填充值进行上采样，然后对上采样后的输入进行卷积操作。近年来，反卷积在可视化、识别、定位、语义分割、视觉问答、超分辨率得到了广泛应用。

#### Dilated Convolution

扩张卷积是近期一种将额外的超参数（相对于需要通过学习获得的参数，超参数由人预先设定）引入CNN的进展，通过在滤波器周围嵌入零元素，在参数数目不变的前提下，增大了感受野的大小，让网络可以覆盖更多相关联的信息。这对于哪些需要巨大的感受野以做出预测的任务而言非常重要。

以一维卷积为例，扩张倍数为，表示倍扩张卷积运算，卷积核为，大小为的卷积核对信号做卷积运算，可以表示为因为卷积核扩张的间隙均为0，所以求和的时候只计算有效的位。这种计算也可以直接应用在2D卷积运算中，如下图所示。

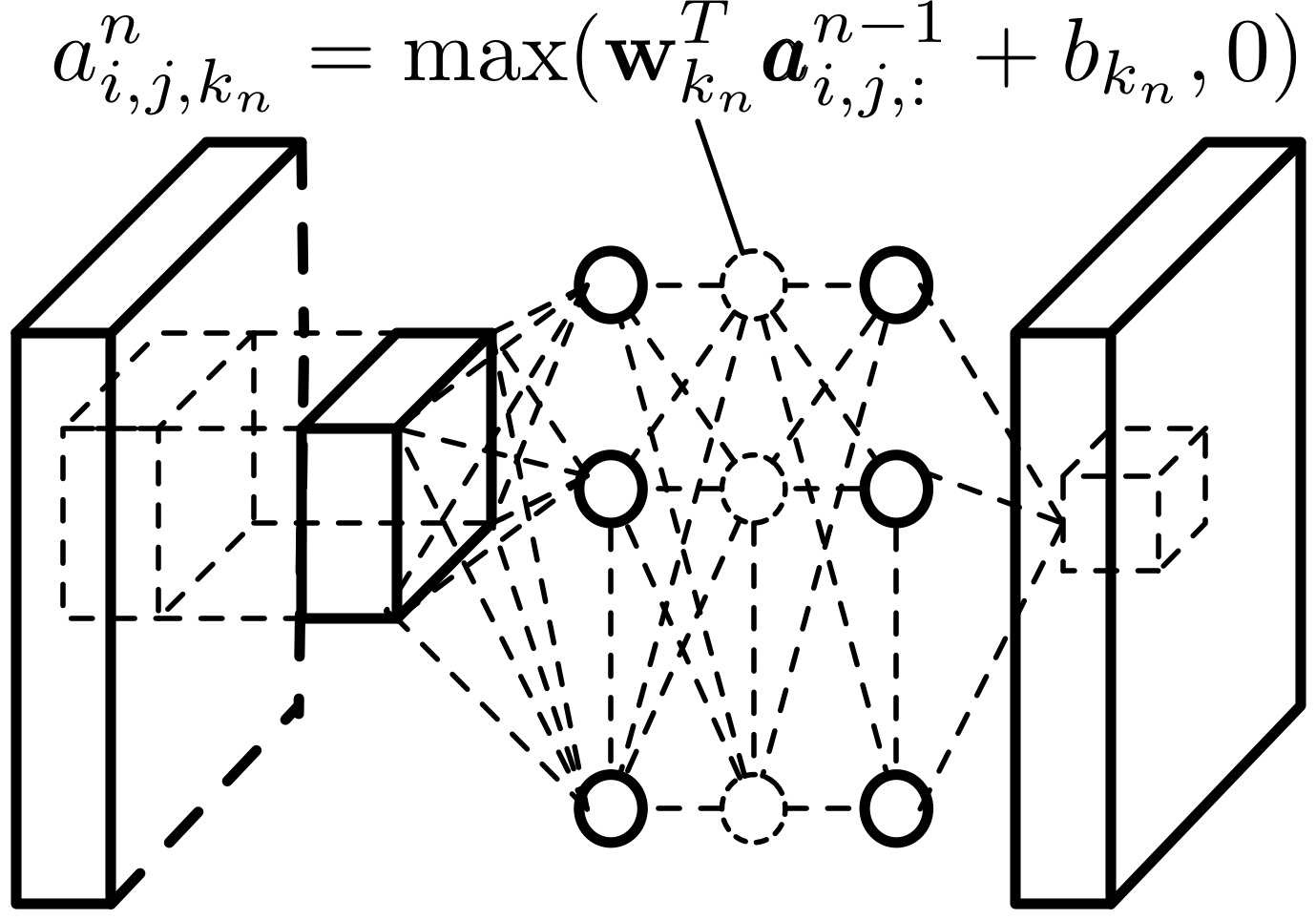
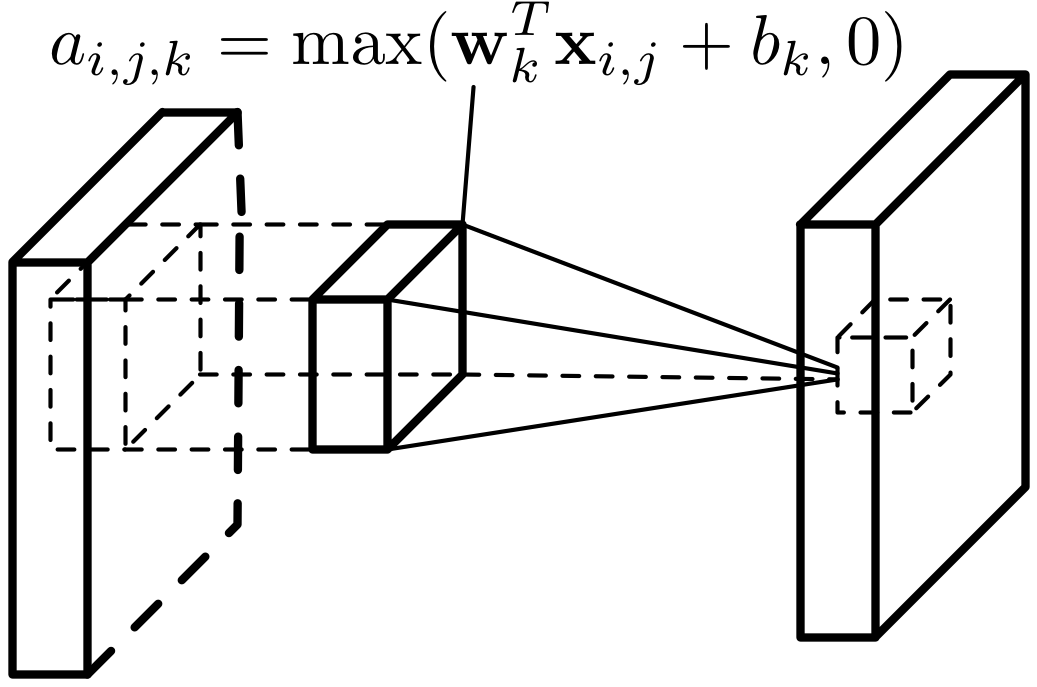


图是一个，图是一个，图是，可见感受野大小为，为放大倍数，为卷积核大小。

扩张卷积广泛应用于场景分隔、机器翻译、语音合成和语音识别。

#### Network in Network

是一种综合网络结构，其将线性的卷积层替换成一个小型网络，本文介绍的层即是其中之一。其有能力进一步接近哪些潜在概念的抽象表示。NIN的整体结构是小型网络的栈式堆叠，如下图所示。



卷积层搭配非线性激活函数(如)的特征映射可以表示为

其中是第个特征映射在位置的值，是在位置的输入块，分别是第个滤波器的权重向量和偏置。

而层表示为

其中，表示层的层数，等于，在层中，的卷积被放置在传统的卷积层之后，相当于跨通道的参数池化操作，这个在中被成功运用。因此层可以被认为是在普通卷积层后的一个级联的跨通道参数池化操作。在最后，设置一个全局均值池化层，用于在最后一层将各特征映射进行空间上的均值，然后直接将输出交给层，相比全连接层，全局均值池化拥有更少的参数并且可以降低过拟合风险和计算量。

#### Inception Module

使用可变的滤波器来捕捉大小不同的可视样式，并且尽可能形成最简洁的结构。包含一个池化操作和三个卷积操作，而在非卷积前放置卷积可以在不增加计算复杂度的情况下增加网络的深度，可以大量减少参数,相比于，可以将参数降低至

为了获取高性能的网络，把计算消耗控制在适中的水平，建议计算中，的大小从输入到输出应该缓慢地下降，且低维度的可以在表示能力没有太大损失的情况下在空间上聚合。最优化的网络性能应该是在每层的滤波器数量和网络的深度中找到平衡。

从中获取灵感，他们最新的组合了体系和 ，他们发现可以显著加速网络的训练，在数据集中，通过组合三个和一个达到了的错误率

## Pooling Layer

池化层是CNN的重要组件，其一方面可以通过减少卷积层之间的链接，从而降低计算成本，另一方面还可以将卷积层的特征映射进一步优化。池化层只有若干个超参数，其没有需要训练的参数。

最基础的池化层有两种，一种是均值池化，一种是最大值池化，前者在区块中计算均值得到输出，后者计算最大值。以下介绍一些近期在使用的池化技术。

#### LP Pooling

由受生物学启发，经理论研究，其比max pooling具备更加的泛化效果。其可以表示为

其中是第各特征映射在对应的池化区域的激活值。当时，就是在对求和，作用类似于均值池化，当，利用极限计算可以发现结果为中的最大值。其作用为最大值池化。

#### Mixed Pooling

由随即丢弃的思想所启发，其结合了最大值池化和均值池化，表示为

其中每次会随机取以此决定是使用最值池化还是均值池化，在反向传播中也会使用相同的进行计算。Mixed Pooling在修正过拟合问题中比最大值池化和均值池化表现得都更好。

#### Stochastic Pooling

同样由随即丢弃思想所启发，不同于直接选择最大值，会根据的多项分布来选择输出值，这样保证了非最大值也有机会被利用。

首先，先计算每个单元的概率

其概率分布

根据其概率分布选择一个位置，使

相比于最大值池化，可以有效抑制过拟合。

#### Spectral Pooling

是通过在频域截去输入的一些表示来降低维度的池化层，可以将的输入降低为的输出。首先其先计算输入特征映射的DFT，然后只截取中间的子矩阵，再通过iDFT还原原图像。相比于，的低通滤波能在相同的维度下保留更多的信息，且此方法也不会像其他方法一样遭遇维数锐减的情况，并且其运算可以用矩阵运算来实现，计算成本也很低。

#### Spatial Pyramid Pooling

可以生成一个定长的表示，与输入的大小无关。输入的特征映射会划分成大小和输入大小成比例的空间容器中，所以划分的容器数是固定的，池化出来的结果也是固定的大小。这与滑动窗口不同，滑动窗口的方法产生的块数与输入大小呈正相关。将最后一层池化层换成，构成的SPP网络可以处理不同大小的图片

#### Multi-scale Orderless Pooling

使用多尺度的弱结构化的Pooling层，在不丢失CNN识别能力的基础上提高不变性。它使用若干尺度从整个图片和局部区块提取深度激活特征，全局特征和传统CNN一致，而区块的用VLAD编码进行聚合，用以捕获更多局部的良好粒度的图片细节，并且提高了不变性。新的图片表示由全局激活值和VLAD的局部特征激活值联系在一起获得。

## Activation Function

适当的激活函数能显著提升CNN性能，一下有一些近期广泛使用的激活函数。

#### ReLU

修正的线性单元（Rectified linear unit），这是一个不饱和的激活函数，其函数定义为

其是一个分段线性函数，分为和，计算速度比要快很多，并且其可以引入稀疏的隐藏单元，从而允许网络更加轻松地产生稀疏表示。在缺少预先训练的时候，也可以高效地训练。尽管在0处地间断可能影响反向传播的性能，许多实际的经验说明ReLU还是表现得更好。

ReLU在正值部分是线性的，导数恒为1，可以规避梯度消失带来的训练困难，但是由于负值部分恒为0，所以一旦有一个很大的负梯度流过，很可能未来的输出都处于负值区域，难以矫正，神经元即死亡，不再有能力更新权值。

#### Leaky ReLU

ReLU的一个劣势是无论一个单元是否激活，都可能存在0梯度，这会导致在初始化未激活的情况下，使用梯度优化算法，不会改变这些单元的权重。并且由于固定在零梯度，训练的速度将变得缓慢。为了缓解这个问题，引入Leaky ReLU，定义如下

其中是之间的一个随机数，相比于ReLU直接把负值映射为0，Leaky ReLU将负值予以压缩，这使得即是单元未激活也可以有一个小的非零的梯度。

#### Parametric ReLU

相比使用一个预先固定的值，Parametric ReLU使每个通道使用不同的可以被学习一提高准确率的，其计算可以被定义为

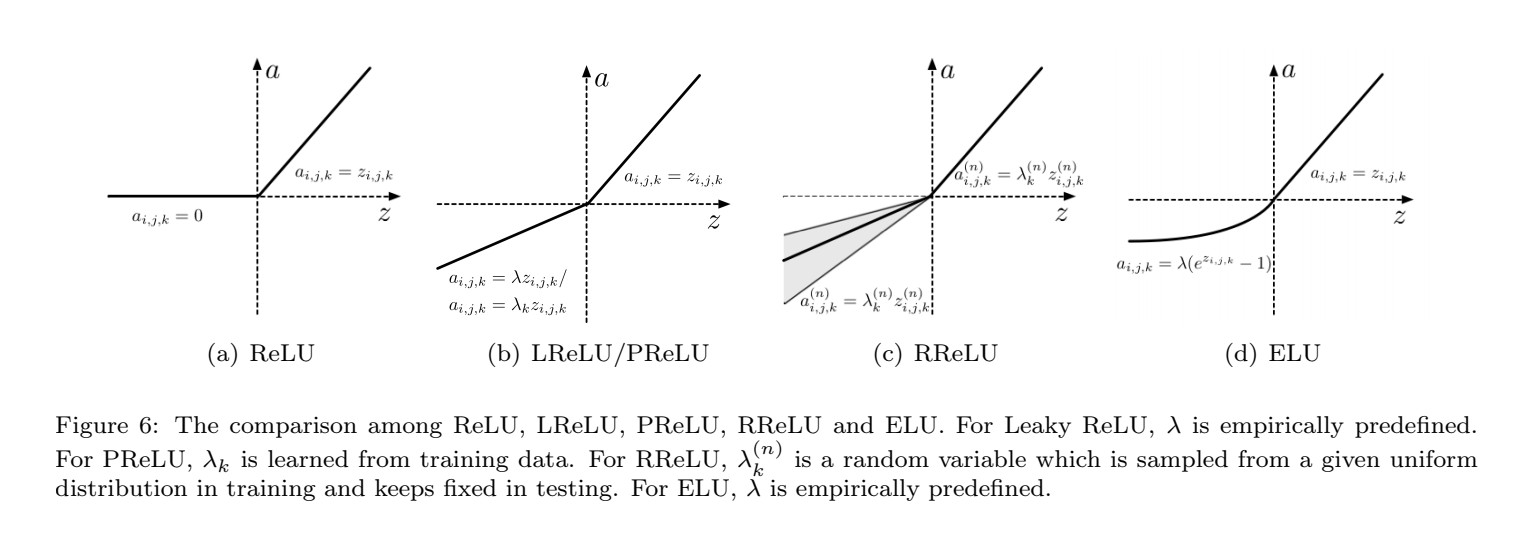
其中是第个通道中可学习的参数，这样对于整个网络来说只引入了少量额外参数（数量和通道数一致），基本没有过拟合的风险和额外的计算消耗（通过一些数据对比，PReLU还是有一定过拟合倾向的），且也能和其他参数一起同时在反向传播中被训练。

#### Randomized ReLU

在RReLU中，负值部分在训练中每计算一组样本都使用均匀分布随机一个采样比例，同样每一个通道使用的参数相互独立。而在测试集中，这个参数是一个固定值。

其中表示的是第个测试样本，表示的是此时对第个通道使用的随机参数。在训练中，指定随机值的上下界，然后每次从中选出随即参数，在测试时不再使用随机值，而是选择上下界的均值作为一个固定参数。

因为其随机化的训练过程，可以避免过拟合。经评估，让负值区域获得一个非0的洒落式的贡献，能提高性能。



#### ELU

ELU可以加速神经网络，还可以提高分类的准确率。像ReLU及变种一样，将正值部分设置为相同的值以规避梯度消失的问题，但和ReLU相反的是，ELU存在负值部分并且可以加速学习。

和LReLU等变种相比，虽然它们都含有一个不饱和的负值部分，但是ELU给负值部分应用了一个饱和函数，这降低了未激活单元的变化幅度，使得ELU对噪音有更强的鲁棒性。ELU计算可以被定义为

其中是一个预定义参数，用来控制负值输入的饱和值。

#### Maxout

Maxout是一个备选的非线性函数，对于空间中的每一个位置，其将会取各通道中的最大值作为输出。可以定义为

其中表示第个通道。值得注意的是，maxout拥有ReLU所有的优点，实际上ReLU是maxout的一个特殊情况，比如并使得为0向量为0。除此之外，maxout非常适合和Dropout配合使用。

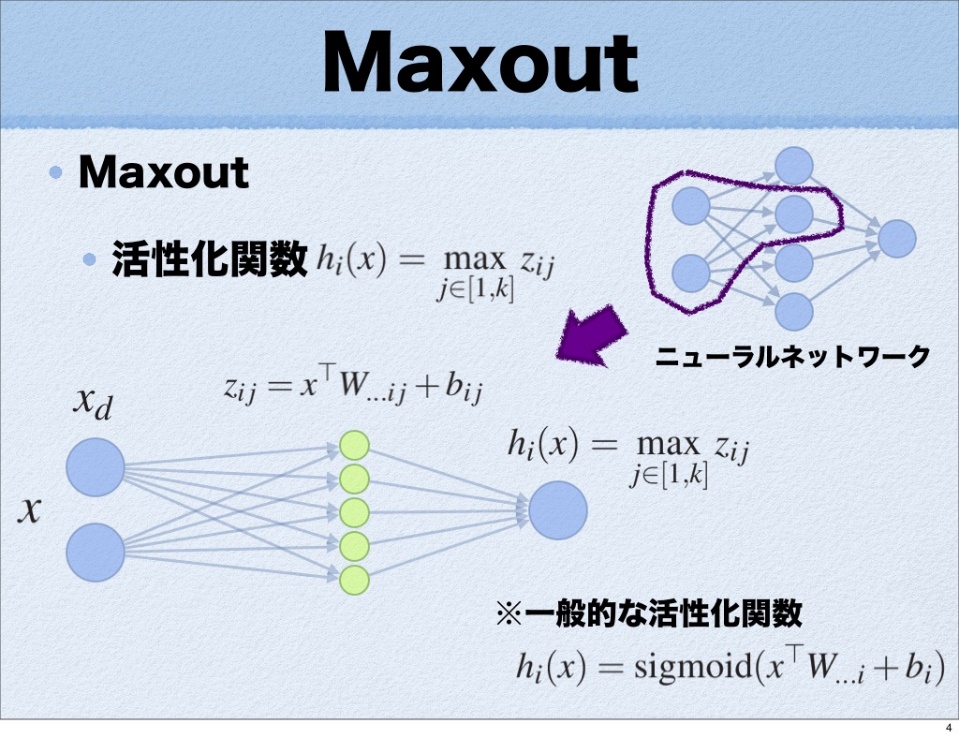
优点：

* 拟合能力强，可以拟合任意凸函数
* 具有ReLU的所有优点
* 没有ReLU的神经元死亡的缺点

缺点：

* Maxout相当于增加了k个神经元作为中介，然后在其中求出最大值，因此参数个数是原先的k倍。

以MLP为例展示Maxout的计算过程。



#### Probout

将maxout转化成基于概率的变种，称为probout。把原先求最大值的操作修改成一个概率采样的过程。首先，为每一个线性单元分配概率值，概率的计算为

其中是一个预定义参数，用来控制概率分布的变化，然后根据概率分布随机选择一个单元的值作为输出。很容易发现，如果，那么其等价于maxout。

为了整合dropout，我们将概率分布定义为

激活函数表示为

其中服从分布

Probout可以平衡maxout获得的令人满意的特性，同时通过让每个通道都产生贡献以提高不变性。然而在实践中，probout因为额外增加的概率计算，其计算成本要高于maxout。

## Loss Function

#### Hinge Loss

Hinge Loss是针对SVM设计的Loss函数，其被定义为

公式中分别表示样本数和分类数，表示样本正确的分类，表示和当前分类的关系，当相同的时候值为-1，反之为1。如果则为，则为

#### Softmax Loss

Softmax Loss是一种用于多分类逻辑回归问题和Softmax激活函数的Loss

回顾Softmax

Softmax Loss可以表示为

其中是网络对样本对分类的概率预测，当两者相等时取1.

近期为了实现Large-Margin的Softmax，出现了L-Softmax loss，其引入了权重与激活值的夹角边距，使得其分类更加明晰。定义为

其中，是用来控制分类之间的边距的参数，当，其退化为普通的softmax loss。通过设置，算法有了一个更困难的学习目标，这样可以避免过拟合，同时表现得比传统的softmax更好。

#### Contrastive Loss

Contrastive Loss是一种典型的用于训练Siamese network（一种用来学习被标记上匹配与否的数据对，并识别相似度的弱监督学习算法）。我们把数据对表示为，并让表示第层的输出对，并将最终的特征向量用于计算cost。Contrastive Loss可被定义为

其中=，意为2范数也就是模。表示的是loss function在处理不匹配样本时影响贡献的margin参数。如果匹配则反之。

Lin发现这种单边距的loss function，如果网络对所有样本做fine-tuning，效果会大幅下降。同时如果只对不匹配的样本做fine-tuning则性能被更多地保留。这表明loss function对匹配样本的处理要为此负责。因为非匹配样本的recall rate还是稳定的，说明匹配样本是主要原因。为了解决这个问题，出现了双margin的loss function

其中对匹配样本也引入了margin参数，当两者距离较近时loss为0，而不纯粹是模。另外不同于只计算最后一层的loss，每一层都有贡献，且反向传播对每一层是同时进行的。

在实践中，发现两个参数可以设置为相等的值，并通过匹配和非匹配样本分布而学习得到。

#### Kullback-Leibler Divergence

Kullback-Leibler Divergence是一种测量基于同一个离散变量的不同分布的差异的方法。的KLD可以定义为

其中是的信息熵，是之间的交叉熵。

KLD被广泛应用于各类自编码器目标函数的loss函数中，比较出名的自编码器变体有sparse AE、Denoising AE and Variational AE。VAE通过贝叶斯推断来解释潜在的表示，其包含编码和解码两部分，编码部分将输入样本转化为潜在的表示，jiemaqi1将表示还原到原数据的空间，并使其尽可能接近原输入。其中是加密和解密各自的参数。

VAE试图最大化对数似然估计的variational lower bound。

其中第一项是重建成本，KLD项是在提议分布的基础上强制执行，通常是一个标准正态分布、离散分布或者是一些带有几何解释的分布。基于VAE，许多变种也被提出。Conditional VAE从条件分布中生成样本。Denoising VAE从损坏的输入中还原原始输入

Jensen-Shannon Divergence 是KLD的一种对称形式，其合良两个分布的相似程度。

通过最小化JSD，可以使得两个分布尽可能地接近。JSD被成功用于Generative Adversarial Network(GAN 生成对抗网络)。不同于VAE将和之间的关系予以建模，GAN明确与优化生成任务，GAN的目标是找到一个分类器D，在生成数据和真实数据之间做一个良好的区分，同时促进生成器G拟合真实数据的分布。

分类器D和生成器G之间的博弈可以构建以下目标函数

原始的GAN论文中表示，对于一个给定的生成器，我们有一个优化过的分类器

优化的目标是缩小之间的JSD。如果G和D具有足够的容量，那么

会收敛于。像条件VAE一样，条件GAN也将带有附加信息的y作为输入，以构造以y为条件的输入样本。在实践中，GAN以训练不稳定著称。

## Regularization

过拟合在深度学习中是不可忽视的问题，不过其可以通过正则化大幅度降低。以下是一些有效的正则化技巧。

#### Lp-norm Regularization

这种正则化在目标函数中添加对模型复杂度的惩罚项，如果原先的Loss定义为，那么正则化后的函数如下

其中是正则项，是正则项的比重。

lp-norm的正则项通常表示为。当时，这是一个凸函数，这让训练变得简单。时，L2正则通常被表示为weight decay。还有一种原则性更强的可选L2正则叫做Tikhonov 正则，其对噪声输入有一定不变性。当，Lp正则可以更多地利用权重的稀疏效应，但会导致其称为非凸函数。

#### Dropout

Dropout被应用于全连接层中，其输出被表示为其中是输入，是权重矩阵，而是一个二进制向量，其值服从伯努利分布（0-1分布），Dropout可以阻止网络过于依赖某一个神经元，并且强制让网络即使缺乏某一特定信息的时候也是准确的。

改进Dropout的方法

* 通过高斯近似进行采样或者整合
* 一个有适应性的Dropout。每个隐藏单元的丢弃概率由一个和深度网络共享参数的二元信度网络所计算
* 在卷积后应用Dropout只会增加训练耗时，而不会抑制过拟合。因此，被称作SpatialDropout的方法，其将Dropout value扩展到整合特征映射中，其在训练小数据的时候表现非常好。

#### DropoutConnect

DropoutConnect在Dropout的基础上更进一步，其不再设置输出值为0，而是随机设置权重参与运算时的值为0，换言之其中服从伯努利分布。

## Optimization

#### Data Augmentation

深度卷积神经网络具有非常多的参数，需要大量可用的数据。一种优雅的减缓数据匮乏的方案是Data Augmentation。其包含对可用数据在不改变其性质的前提下进行变换。热门的方法有简单的几何变换，形如镜像、旋转、平移或者对亮度进行变换。Paulin提出在一个给定的转换集中贪心地选择最佳的变换。然而这种策略当可选择的变换集非常大的时候，会带来巨额的计算开销。Hauberg提出通过随机生成diffeomorphisms，还有一种从网络中收集图像以促进训练细粒度识别的学习。

#### Weight Initialization

深度卷积神经网络具有非常大量的参数，而且其loss函数是非凸的，这让训练变得很困难。为了加快收敛且避免梯度消失，一个适当的网络初始化时最重要的环节之一。偏置单元的值可以被设置为0，但权重参数要仔细初始化以破坏同一层网络的对称结构。如果没有合理地初始化，假设每一层把输入进行了倍拉伸，而最终的输出则是输入的，其中是网络层数。无论，最终的输出都会陷入极大或接近0的状态。

一种初始化方法是以0均值的0.01标准差的高斯分布来初始化，并将第二、四、五层卷积层以及所有全连接层的偏置单元设置为常数1。

另一种随机初始化方法称为 Xavier，其从一个零均值、方差为的高斯分布来选取权重，其中分别为这一层的输入单元数和输出单元数。这种方法可以通过输入输出的单元数目自动地决定初始化的规模，并且使信号在层之间传递时维持在合理的范围内。Caffe中的一个变种只使用作为方差，实现更加简单。Xavier近期还被扩展以考虑非线性整流，这种鲁棒性的初始化方法考虑了ReLU的非线性，允许网络训练极深的模型，而原先的Xavier是不行的。

此外，Saxe表示，在线性网络中，标准正交矩阵初始化要比高斯初始化表现得更好，在非线性网络中，其也可以工作。其还被扩展为一个迭代的过程，详细一点，这是一种以层序进行的单位方差处理方案，其可以被看作是标准正交初始化和Batch Normalization的结合，仅在第一个mini-batch执行。其类似于Batch Normalization，因为都经历一个单位方差标准化的流程，不同的是其使用标准正交标准化来初始化权重，可以有效接触激活值之间的相关性。

#### Stochastic Gradient Descent

反向传播是训练的标准方法，其使用梯度下降来更新参数值。其中随机梯度下降是众多优化算法之一。标准梯度下降更新参数的方法是，其中是在全体训练集中的期望值，是学习速率。随机梯度下降不计算完整的期望值，而是通过随机从训练集中选择一个样本来预测梯度。

在实践中，SGD更新参数都是依靠mini-batch而不是单个样本，这可以降低参数更新时的偏差，并且可以更加稳定地收敛。收敛速度由控制。

然而SGD并不能保证非常良好地收敛，仍旧有非常多的挑战需要被解决。

学习速度难以选择，一种通用的方法是在最开始使用固定的学习速率，使其可以稳定地收敛，随后降低学习速率让收敛速度变慢。

一种称作momentum的方法被提出以加快学习速率，其还可以受训练过的batches所影响。其提出在相关的方向上累计一个速度向量，然后用这个速度作为更新的增量。

是当前的速度向量，是momentum参数，通常被设置为0.9。

Nesterov momentum是另一种使用momentum的梯度下降优化，其修改了速度变更中Loss函数的参数

相比于常规momentum先计算梯度，再向累计的更新梯度方向移动不同，Nesterov momentum先利用上一步计算出来的梯度移动，再计算梯度更新。这种预见性的更新，避免了移动过快并且可以提升性能。

并行化的SGD方法在并行性方面提升了大规模的机器使用SGD的性能，不像标准SGD，在训练时会被某一个非常慢的机器所拖延，这种并行化方法使用异步机制，这样就其他优化手段就不会被延迟，除非自己就是最慢的电脑。

Downpour SGD用来加速具有CPU集群的分布式训练，也有一些在GPU上使用异步SGD。相比在一台机器上训练，这种异步手段成倍加速了训练效率。

因为SGD最后不一定会收敛，当现有的训练效能不再提升时，可以结束训练。一种补救过训练的方法就是使用early stopping，其以在验证集上的表现作为训练过程中中断训练的依据。还有很多停止规则可以考虑，比如，训练只进行固定的步数，或者达到一个预定的误差值，停止策略的设定需要非常仔细，需要让训练在不过拟合的前提下尽可能进行以提高泛化能力。

#### Batch Normalization

数据标准化是数据准备的第一步。全局数据标准化把数据转化为均值为0且单位方差的分布，然而数据在流过深度网络时，对于内部层的分布会改变，这会使得学习能力下降且拖慢网络。Batch Normalization可以部分减缓这种现象。其通过正规化操作，固定了均值和方差，解决了被称作covariate shift的问题。其中所估计的均值和方差是由mini-batch而不是整个训练集计算出来的。假设输入的维度是，，首先对其进行如下计算

其中分别是这个batch的均值和方差，是一个固定的值。为了提高表示能力，标准化的输入还要进一步转化为

其中是可学习的参数。BN算法相对于全局数据标准化有非常多的优势，首先，其减缓了convariant shift。其次其减少了梯度对参数的尺度和初始化值得影响，使得梯度在网络中得流动更有效率。其使得我们可以使用更高的学习速率而不必担心带来发散的风险。而且BN对模型有正则化的效果，可以较少Dropout的需求。最后，BN使我们可以使用饱和的非线性激活函数而不必担心神经元死亡。

#### Shortcut Connections

深度神经网络的梯度消失问题可以通过设置正规化的初始值和BN算法缓解，尽管它们成功阻止网络过拟合，同时也引入了训练困难的问题，最终比更浅的网络效果更差。这种更加深的网络所遭受的优化问题被认为是退化问题。

LSTM网络通过门函数来决定多少激活值要参与变换或直接通过，受其启发，一种能够使得几乎任意深度的网络都可优化的高速路网络结构被提出，其输出表示如下

其中表示第个高速块的输入和输出。是门函数，通常是一个仿射变换伴随非线性激活函数。这种门机制迫使输入输出的尺寸是相同的，且让十至上百层的网络得以高效训练，这证明其并不是只学到了一个固定的结构，而是根据具体的样本有动态的。

ResNet（残差网络）的核心思想和LSTM一样，但不同于直接将可学习的权重应用于特定的神经特异性门中，捷径是非选通的，未经转化的输入直接传播给输出，并且携带非常少的参数。ResNet的输出可以表示为

是权重层，可以试混合而成的函数，比如卷积、BN、激活、池化等等。对于一个残差块，任意更深层网络的激活值可以表示成浅层网络的激活值和残差函数的求和。这也说明梯度可以直接传播给更浅的单元，这使得深度残差网络可以更简单地被优化和训练。而传统前馈网络，梯度本质上是一系列矩阵向量的连乘，当网络过深则会梯度消失。

在传统ResNet之后，通过对预激活的ResNet安排一系列实验，表明恒等捷径对网络来说最容易学习。此外，还发现先使用Batch Normalization比在添加捷径后再使用表现得更好。在比较中，使用BN+ReLU预激活可以达到比常规ResNet更高的准确性。此外还可将权重因子引入卷积层后的输出。

ResNet可以非常深，但一般很瘦。较宽的ResNet缩减了深度，尽管在CIFAR-10, CIFAR-100, and SVHN中获取了不错的结果，但对于大尺寸的图像分类任务，其主张并没有被证实。

随机深度的ResNet随机地丢弃一些子层，并且用恒等映射建立旁路传送他们，而不通过层内运算。这种操作是以mini-batch为单位的。通过结合随机深度地ResNet和Dropout，其被概括为集成ResNet、Dropout ResNet、Stochastic Depth ResNet。

ResNet in ResNet（简称RiR），其结构是在传统的卷积网络基础上整合ResNet的特性。RiR的每一个块都包含残差单元和非残差块。这样RiR就可以学习到需要使用多少卷积层了（对于冗余的层，网络自然会调整权重以使其更多地走捷径）。

ResNet of ResNet其使用多级捷径，而DenseNet则更加激进，将一层的输出和之前每个子层都相联。

对于所有ResNet，使用捷径来训练非常深的网络都是有帮助的。

## Fast Processing of CNNs

#### FFT

在傅里叶域中进行卷积操作。

优点：

* 傅里叶变换后的滤波器在对一个batch中的多张图片做卷积时可以重用。
* 傅里叶变换后的梯度可以在对滤波器和输入图像反向传播梯度时复用。
* 输入通道的求和可以在傅里叶域中执行，因此反卷积操作只需要在每条通道、每张图片中执行一次。

缺点：

* 耗费额外的内存以存储傅里叶域的特征映射（原因是滤波器必须通过padding保持和input一样的大小），当striding参数大于1时，这个耗费尤其之大。

广泛运用于小尺寸卷积滤波器。

#### Structured Transforms

* **Low-rank matrix factorization（低秩矩阵分解）**

给定且秩为r的矩阵，其存在因子其中是一个的列满秩矩阵，而是一个行满秩矩阵，我们可以将替换成以减少参数。可以保证，其中是一个缩小因子。复杂度降低至。

* **Adaptive Fastfood transform**

其是Fastfood 变换在逼近矩阵上的泛化，其将全连接层的权重矩阵替换成，其中是参数的斜对角矩阵，是随机置换矩阵，表示的是Walsh-Hadamard矩阵。此算法空间复杂度是，时间复杂度是。

* **Circulant Matrix**

因为其计算的复杂度和空间消耗都非常良好，通过在权重矩阵上引入circulant结构，以加速计算。更进一步可以用FFT加速。计算过程，其中是circulant矩阵，是输入。

可以用FFT和iFFT表示为，其中代表点乘，由决定，是随机信号反转矩阵，以提高模型容量。其将复杂度由降低至，空间复杂度由降低至。

#### Low Precision

背景：浮点数可以微调参数，但是信息的冗余度非常高。

BNN：将一些或者所有的参与计算输出的运算限制为二进制值运算。

其二进制表现在三个方面：

* 二进制输入激活值
* 二进制权重
* 二进制输出激活值

Full binarization需要这三部分同时二进制化，而Partial binarization则只要有一部分是二进制的即可。全二进制化的网络只需要XNOR和bit计数操作即可。

#### Weight Compression

* Vector Quantization

Vector Quantization（矢量量化）可以用来压缩全连接层，VQ将数值组成集体，而不是一个个寻址。

CNN的参数被论证存在冗余，其可以用VQ来显著降低动态参数的数目。

* Pruning

Pruning（剪枝）通过永久性丢弃重要性较低的连接以减少参数数量。这样更小的网络既可以继承大型网络的知识，还可以保持较为不错的性能。

一种基于量级的剪枝

如果任意权重的量级低于某个阈值，则剪掉该权重。其一种扩展可以在先前的迭代中恢复被修剪的权重。在其基础上，考虑权重之间的关联，提出一种基于能量感知的剪枝算法，其直接使用对CNN能量消耗的估计来引导剪枝。

* Hash

将连接上的权重hash进buckets，而在同一个hash buckets的连接共享一个hash参数，其压缩了空间，但是保持了图像分类任务的泛化能力。

#### Sparse Convolution

* 稀疏基本卷积核的表示，可以通过卷积核内和卷积核间的冗余实现稀疏化。
* 优化超参数（卷积核大小、深度、局部连通性）。
* 通过少量查找，将卷积编码成丰富的字典集，其被用于覆盖CNN中权重的空间。解压操作只需要一个字典和两个tensors（张量）。字典被同一层的所有卷积核所共享。在小样本学习中效果很好。

## Applications of CNNs

#### Image Classification

#### Object Detection

#### Object Tracking

#### Pose Estimation（姿态估计）

#### Text Detection and Recognition

#### Visual Saliency Detection

#### Action Recognition

#### Scene Labeling

#### Speech Processing

#### Natural Language Processing