网络中的网络

Min Lin[1,2], Qiang Chen[2], Shuicheng Yan[2]

综合科学与工程研究生院

新加坡国立大学电子与计算机工程系

{linmin,chenqiang,eleyans}@nus.edu.sg

摘要

我们提出了一种名为“Network In Network”（NIN）的新型深度网络结构，以增强接受域内局部补丁的模型可区分性。传统的卷积层使用线性滤波器，然后是非线性激活函数来扫描输入。相反，我们构建了更复杂结构的微型神经网络，以抽象接受域内的数据。我们用多层感知器实例化了微型神经网络，这是一种强大的函数逼近器。通过类似 CNN 的方式在输入上滑动微型网络来获得特征图，然后将其输入到下一层。可以通过堆叠多个上述结构来实现深层的 NIN。通过微型网络增强局部建模，我们能够在分类层上利用特征图上的全局平均池化，这比传统的全连接层更容易解释，且不太容易过拟合。我们在 CIFAR-10 和 CIFAR-100 上展示了 NIN 的最先进分类性能，并在 SVHN 和 MNIST 数据集上展示了合理的性能。

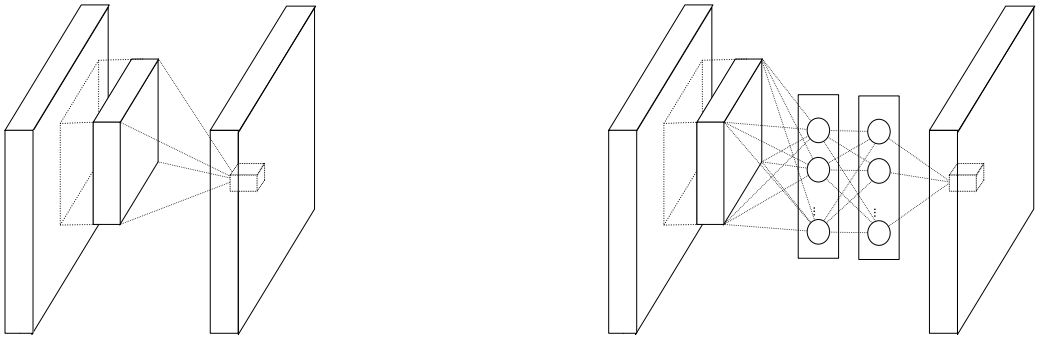
1 引言

卷积神经网络（CNN）[1] 由交替的卷积层和池化层组成。卷积层对输入的每个局部区域进行线性滤波器和底层感受域的内积运算，然后经过非线性激活函数处理。最终的输出被称为特征图。

在 CNN 中，卷积滤波器是底层数据补丁的广义线性模型（GLM），我们认为 GLM 的抽象程度较低。这里的抽象是指特征对于相同概念的变体是不变的[2]。通过用更强大的非线性函数逼近器替换 GLM，可以增强局部模型的抽象能力。当潜在概念的样本是线性可分离的时，GLM 可以实现较高程度的抽象，即概念的变体都位于 GLM 定义的分隔平面的一侧。因此，传统的 CNN 隐含地假设潜在概念是线性可分离的。然而，同一概念的数据通常位于非线性流形上，因此捕捉这些概念的表示通常是输入的高度非线性函数。在 NIN 中，GLM 被“微型网络”结构取代，这是一个通用的非线性函数

逼近器。在这项工作中，我们选择多层感知器[3]作为微型网络的实例化，它是一个通用的函数逼近器，可以通过反向传播进行训练。

以下是 Figure 1 的内容描述： 我们称之为 mlpconv 层的结果结构与 CNN 进行了比较。线性卷积层和 mlpconv 层都将局部感受域映射到输出特征向量。mlpconv 通过由多个具有非线性激活函数的全连接层组成的多层感知器（MLP）将输入的局部补丁映射到输出特征向量。MLP 在所有局部感受域之间共享。特征图是通过滑动 MLP 获得的。



(a) 线性卷积层　　(b) 多层感知器卷积层

图 1：线性卷积层和 mlpconv 层的比较。线性卷积层包括线性滤波器，而 mlpconv 层包括一个微型网络（本文选择多层感知器）。这两个层都将局部感受域映射到潜在概念的置信度值。

在 NIN 中，特征图是通过类似 CNN 的方式滑动在输入上获得，并被馈送到下一层。NIN 的整体结构是多个 mlpconv 层的堆叠。它被称为 “Network In Network”（NIN），因为在 mlpconv 层中我们有微型网络（MLP），它们是组成整个深度网络的构成元素。

在 NIN 中，我们不采用传统的全连接层进行分类，而是直接通过全局平均池化层输出最后一个 mlpconv 层的特征图的空间平均值作为各类别的置信度，然后将得到的向量馈送到 softmax 层进行分类。在传统的 CNN 中，由于全连接层充当了中间的黑盒子，很难解释目标成本层中的类别级信息是如何传递回先前的卷积层的。相比之下，全局平均池化更有意义和可解释性，因为它强化了特征图和类别之间的对应关系，这得益于使用微型网络进行更强大的局部建模。此外，全连接层容易出现过拟合，并且严重依赖于 dropout 正则化[4][5]，而全局平均池化本身就是一种结构正则化器，天然地防止整体结构的过拟合。

2 卷积神经网络

经典的卷积神经网络[1]由交替堆叠的卷积层和空间池化层组成。卷积层通过线性卷积滤波器生成特征图，然后经过非线性激活函数（如整流器、Sigmoid、tanh 等）。以线性整流器为例，特征图的计算如下所示：

$$f_{i,j,k} = \max(w_k^T x_{i,j}, 0).$$ (1)

这里的 (i,j) 是特征图中的像素索引，x_ij 代表以位置 (i,j) 为中心的输入补丁，k 用于索引特征图的通道。

这种线性卷积在潜在概念的实例是线性可分离的情况下足够进行抽象。然而，实现良好抽象的表示通常是输入数据的高度非线性函数。在传统的 CNN 中，这可能通过利用一个过度完备的滤波器集[6]来弥补。即，可以学习单独的线性滤波器来检测同一概念的不同变体。然而，为单个概念拥有过多的滤波器会给下一层带来额外负担，因为它需要考虑来自前一层的所有变体的组合[7]。就像在 CNN 中，来自更高层的滤波器映射到原始输入中的更大区域。它通过将下一层的低层概念组合起来生成更高级的概念。因此，我们认为在将它们组合成更高级概念之前，对每个局部补丁进行更好的抽象将是有益的。

在最近的 maxout 网络[8]中，通过对仿射特征图进行最大池化（仿射特征图是线性卷积的直接结果，没有应用激活函数），减少了特征图的数量。对线性函数的最大化使得可以得到一个分段线性逼近器，能够逼近任何凸函数。与执行线性分离的传统卷积层相比，maxout 网络更为强大，因为它可以分离位于凸集内的概念。这种改进使 maxout 网络在几个基准数据集上表现最佳。

然而，maxout 网络假设潜在概念的实例位于输入空间中的凸集内，这并不一定成立。当潜在概念的分布更加复杂时，需要使用更一般的函数逼近器。为了实现这一点，我们引入了新颖的"网络内网络"结构，在每个卷积层中引入微型网络，以计算局部补丁的更抽象特征。

在之前的几项研究中，已经提出了在输入上滑动微型网络的方法。例如，结构化多层感知器（Structured Multilayer Perceptron，SMLP）[9]将共享的多层感知器应用于输入图像的不同补丁；在另一项工作中，训练了一个基于神经网络的过滤器用于人脸检测[10]。然而，它们都是针对特定问题设计的，并且都只包含一个层的滑动网络结构。NIN 从更一般的角度提出，将微型网络整合到 CNN 结构中，以更好地抽象所有层级的特征。

3 网络中的网络

我们首先强调我们提出的"网络内网络"结构的关键组成部分：MLP 卷积层和全局平均池化层，分别在第 3.1 节和第 3.2 节中进行详细介绍。然后我们在第 3.3 节详细介绍整体的 NIN 结构。

3.1 多层感知器卷积层

鉴于对潜在概念分布没有先验知识，我们希望使用通用函数逼近器来提取局部补丁的特征，因为它能够逼近潜在概念的更抽象表示。径向基网络和多层感知机是两种众所周知的通用函数逼近器。我们选择多层感知机有两个原因。首先，多层感知机与卷积神经网络的结构兼容，可以使用反向传播进行训练。其次，多层感知机本身可以是一个深度模型，这与特征重用的精神是一致的[2]。本文中将这种新型层称为 mlpconv，其中多层感知机替换 GLM 对输入进行卷积。图 1 说明了线性卷积层和 mlpconv 层之间的差异。mlpconv 层的计算如下所示：

$$
\begin{aligned}
f_{i,j,k_1}^{1} &= \max({w_{k_1}^{1}}^{T} x_{i,j} + b_{k_1}, 0). \\
&\vdots \\
f_{i,j,k_n}^{n} &= \max({w_{k_n}^{n}}^{T} f_{i,j}^{n-1} + b_{k_n}, 0).
\end{aligned}
$$

这里的 n 是多层感知机中的层数。在多层感知机中，使用修正线性单元作为激活函数。

从跨通道（跨特征图）池化的角度来看，方程 2 相当于在普通卷积层上进行级联的跨通道参数化池化。每个池化层对输入特征图进行加权线性重组，然后经过修正线性单元。跨通道池化的特征图在下一层中再次进行跨通道池化。这种级联的跨通道参数化池化结构允许跨通道信息进行复杂且可学习的交互。

跨通道参数化池化层也等价于具有 1x1 卷积核的卷积层。这种解释使得理解 NIN 的结构变得直观。

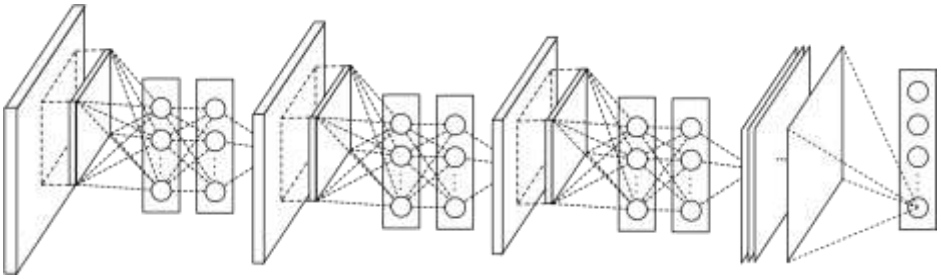


图 2：网络内网络的整体结构。在本文中，NIN 包括三个 mlpconv 层和一个全局平均池化层的堆叠。

与 maxout 层的比较：maxout 网络中的 maxout 层在多个仿射特征图上执行最大池化[8]。maxout 层的特征图计算如下：

$$f_{i,j,k} = \max_m \left(w_{k_m}^T x_{i,j} \right) \quad (3)$$

对线性函数进行 maxout 形成了一个分段线性函数，能够模拟任何凸函数。对于凸函数，函数值低于特定阈值的样本形成一个凸集。因此，通过逼近局部补丁的凸函数，maxout 具有形成分隔超平面的能力，用于样本位于凸集内的概念（例如 l2 球体，凸锥体）。Mlpconv 层与 maxout 层的不同之处在于，凸函数逼近器被通用函数逼近器取代，后者在建模潜在概念的各种分布方面具有更强的能力。

3.2 全局平均池化

传统的卷积神经网络在网络的较低层进行卷积操作。在分类任务中，最后一个卷积层的特征图被向量化，然后输入到全连接层，最后是一个 softmax 逻辑回归层[4] [8] [11]。这种结构将卷积结构与传统的神经网络分类器相连接。它将卷积层视为特征提取器，然后对得到的特征进行传统的分类。

然而，全连接层容易出现过拟合问题，从而影响整个网络的泛化能力。Hinton 等人提出了 Dropout 作为一种正则化方法，它在训练过程中随机将全连接层中一半的激活置零。这种方法提高了泛化能力并且在很大程度上防止了过拟合[4]。

在本文中，我们提出了另一种策略，称为全局平均池化，来替代 CNN 中的传统全连接层。其思想是在最后的 mlpconv 层为分类任务中的每个类别生成一个特征图。我们不是在特征图之上添加全连接层，而是取每个特征图的平均值，然后将得到的向量直接输入到 softmax 层。全局平均池化相对于全连接层的一个优势是，它更符合卷积结构，通过强制特征图和类别之间的对应关系。因此，特征图可以更容易地被解释为类别置信度图。另一个优势是，在全局平均池化中没有需要优化的参数，因此可以避免在这一层出现过拟合。此外，全局平均池化消除了空间信息，因此对输入的空间平移更加鲁棒。

我们可以将全局平均池化视为一种结构正则化器，它明确强制特征图成为概念（类别）的置信度图。这得益于 mlpconv 层，因为它们比 GLM 更好地逼近了置信度图。

3.3 网络中的网络结构

NIN 的整体结构是一堆叠的 mlpconv 层，其上是全局平均池化层和目标成本层。在 mlpconv 层之间可以像 CNN 和 maxout 网络那样添加子采样层。图 2 显示了一个具有三个 mlpconv 层的 NIN。在每个 mlpconv 层内部，有一个三层感知器。NIN 和微型网络中的层数是灵活的，可以根据特定任务进行调整。

4 实验

4.1 概述

我们在四个基准数据集上评估了 NIN：CIFAR-10 [12]、CIFAR-100 [12]、SVHN [13] 和 MNIST [1]。用于这些数据集的网络都由三个堆叠的 mlpconv 层组成，而且在所有实验中，mlpconv 层后面都跟着一个空间最大池化层，将输入图像降采样两倍。作为正则化器，我们在除了最后一个 mlpconv 层之外的所有输出上都应用了 dropout。除非另有说明，实验部分中使用的所有网络都使用全局平均池化，而不是在网络顶部使用全连接层。另一个应用的正则化器是权重衰减，这是 Krizhevsky 等人所使用的[4]。图 2 说明了本节中使用的 NIN 网络的整体结构。参数的详细设置在附录中提供。我们在由 Alex Krizhevsky 开发的超快 cuda-convnet 代码上实现了我们的网络。数据集的预处理、训练集和验证集的分割都遵循 Goodfellow 等人的方法[8]。

我们采用了 Krizhevsky 等人[4]使用的训练过程。也就是说，我们手动设置了权重和学习率的适当初始化。网络使用大小为 128 的小批量进行训练。训练过程从初始权重和学习率开始，直到在训练集上的准确度停止改善，然后学习率降低一个数量级。这个过程重复一次，最终的学习率是初始值的百分之一。

4.2 CIFAR-10

CIFAR-10 数据集[12]包含 10 类自然图像，总共有 50,000 张训练图像和 10,000 张测试图像。每个图像都是大小为 32x32 的 RGB 图像。对于这个数据集，我们应用了与 Goodfellow 等人在 maxout 网络中使用的相同的全局对比度归一化和 ZCA 白化。我们使用训练集中的最后 10,000 张图像作为验证数据。

在这个实验中，每个 mlpconv 层的特征图数量设置为与相应的 maxout 网络中的数量相同。我们使用验证集调整了两个超参数，即局部感受野大小和权重衰减。之后，固定了超参数，我们从头开始使用训练集和验证集重新训练网络。得到的模型用于测试。在这个数据集上，我们获得了 10.41%的测试错误率，这比最先进的方法提高了超过 1%。与以前的方法的比较见表 1。

Table 1: Test set error rates for CIFAR-10 of various methods.

Method	Test Error
Stochastic Pooling [11]	15.13%
CNN + Spearmint [14]	14.98%
Conv. maxout + Dropout [8]	11.68%
NIN + Dropout	**10.41%**

CNN + Spearmint + Data Augmentation [14]	9.50%
Conv. maxout + Dropout + Data Augmentation [8]	9.38%
DropConnect + 12 networks + Data Augmentation [15]	9.32%
NIN + Dropout + Data Augmentation	**8.81%**

在我们的实验中发现，NIN 中在 mlpconv 层之间使用 dropout 可以提高网络的性能，改善模型的泛化能力。如图 3 所示，引入 dropout 层在 mlpconv 层之间将测试错误率降低了超过 20%。这一观察结果与 Goodfellow 等人的研究[8]一致。因此，在本文中使用的所有模型都在 mlpconv 层之间添加了 dropout 正则化器。没有 dropout 正则化器的模型在 CIFAR-10 数据集上的错误率为 14.51%，这已经超过了许多以前有正则化器的最新技术（除了 maxout）。由于没有 maxout 没有 dropout 的性能数据，因此本文只比较有 dropout 正则化的版本。

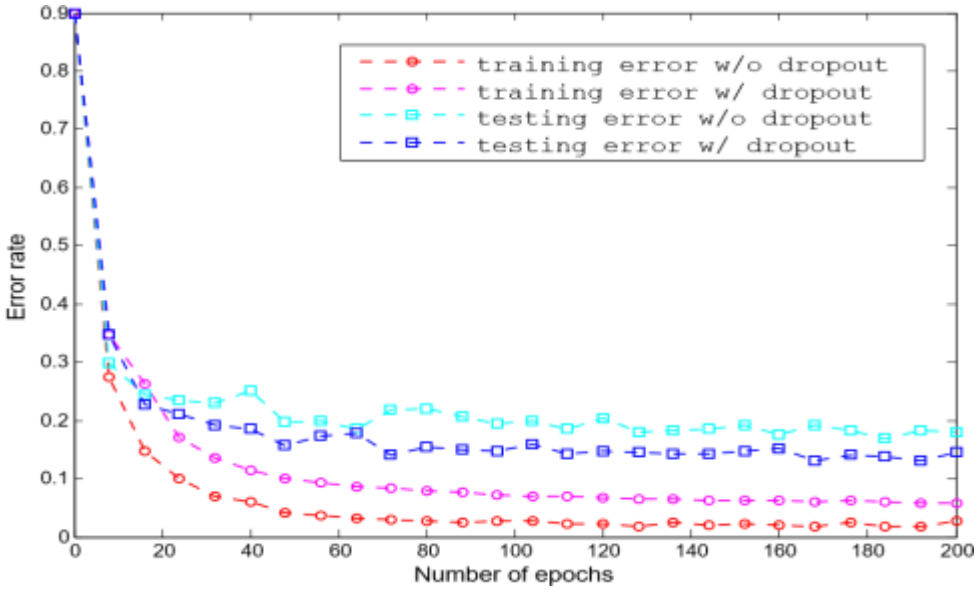


图 3：mlpconv 层之间使用 dropout 的正则化效果。显示了 NIN 在训练的前 200 个 epoch 中使用和不使用 dropout 的训练和测试错误。

为了与先前的研究保持一致，我们还在 CIFAR-10 数据集上使用平移和水平翻转增强来评估我们的方法。我们能够实现 8.81%的测试错误率，这创造了新的最先进性能。

4.3 CIFAR-100

CIFAR-100 数据集[12]与 CIFAR-10 数据集在大小和格式上相同，但包含 100 个类别。因此，每个类别中的图像数量只有 CIFAR-10 数据集的十分之一。对于 CIFAR-100，我们没有调整超参数，而是使用与 CIFAR-10 数据集相同的设置。唯一的区别是最后一个 mlpconv 层输出 100 个特征图。在 CIFAR-100 上获得了 35.68%的测试错误率，这超过了当前最佳性能（没有数据增强）超过 1%。性能比较的详细信息见表 2。

Table 2: Test set error rates for CIFAR-100 of various methods.

Method	Test Error
Learned Pooling [16]	43.71%
Stochastic Pooling [11]	42.51%
Conv. maxout + Dropout [8]	38.57%
Tree based priors [17]	36.85%
NIN + Dropout	**35.68%**

4.4 Street View House Numbers

SVHN 数据集[13]由 630,420 张 32x32 的彩色图像组成，分为训练集、测试集和额外集。该数据集的任务是对每幅图像中央的数字进行分类。训练和测试过程遵循 Goodfellow 等人的方法[8]。即从训练集中每个类别选择 400 个样本，并从额外集中每个类别选择 200 个样本用于验证。训练集和额外集的其余部分用于训练。验证集仅用作超参数选择的指导，但从不用于训练模型。

数据集的预处理再次遵循 Goodfellow 等人的方法[8]，即局部对比度归一化。在 SVHN 中使用的结构和参数与用于 CIFAR-10 的类似，包括三个 mlpconv 层，后跟全局平均池化。对于这个数据集，我们获得了表 3：各种方法在 SVHN 上的测试集错误率。

Method	Test Error
Stochastic Pooling [11]	2.80%
Rectifier + Dropout [18]	2.78%
Rectifier + Dropout + Synthetic Translation [18]	2.68%
Conv. maxout + Dropout [8]	2.47%
NIN + Dropout	2.35%
Multi-digit Number Recognition [19]	2.16%
DropConnect [15]	1.94%

测试错误率为 2.35%。我们将我们的结果与未增强数据的方法进行比较，比较结果如表 3 所示。

4.5 MNIST

MNIST[1]数据集由手写的数字 0-9 组成，每个数字的大小为 28x28。总共有 60,000 张训练图像和 10,000 张测试图像。对于这个数据集，采用了与 CIFAR-10 相同的网络结构。但是，从每个 mlpconv 层生成的特征图的数量减少了。因为 MNIST 与 CIFAR-10 相比是一个更简单的数据集，所以需要更少的参数。我们在这个数据集上进行了没有数据增强的测试。结果与之前采用卷积结构的方法进行了比较，如表 4 所示。

表 4：不同方法在 MNIST 上的测试集错误率。

Method	Test Error
2-Layer CNN + 2-Layer NN [11]	0.53%

Stochastic Pooling [11]	0.47%
NIN + Dropout	0.47%
Conv. maxout + Dropout [8]	0.45%

由于 MNIST 已被调整到非常低的错误率，因此我们取得的性能（0.47%）与当前最佳性能（0.45%）相当，但并不更好。

4.6 全局平均池化作为正则化器

全局平均池化层类似于全连接层，因为它们都对向量化的特征图执行线性变换。不同之处在于变换矩阵。对于全局平均池化，变换矩阵是预先确定的，并且仅在共享相同值的块对角元素上非零。全连接层可以有密集的变换矩阵，并且这些值会经过反向传播优化。为了研究全局平均池化的正则化效果，我们用全连接层替换了全局平均池化层，而模型的其他部分保持不变。我们评估了这两个模型，在全连接线性层之前使用和不使用 dropout。这两个模型都在 CIFAR-10 数据集上进行了测试，并比较了它们的性能，结果见表 5。

表 5：全局平均池化与全连接层的比较。

Method	Testing Error
mlpconv + Fully Connected	11.59%
mlpconv + Fully Connected + Dropout	10.88%
mlpconv + Global Average Pooling	10.41%

如表 5 所示，没有使用 dropout 正则化的全连接层表现最差（11.59%）。这是可以预料的，因为如果不应用正则化器，全连接层会对训练数据过拟合。在全连接层之前添加 dropout 减少了测试错误（10.88%）。全局平均池化在这三者中取得了最低的测试错误率（10.41%）。

然后，我们探讨全局平均池化是否对传统 CNN 具有相同的正则化效果。我们实例化了一个传统 CNN，其结构如 Hinton 等人[5]所述，包括三个卷积层和一个局部连接层。局部连接层生成 16 个特征图，这些特征图被馈送到一个带有 dropout 的全连接层。为了公平比较，我们将局部连接层的特征图数量从 16 减少到 10，因为全局平均池化方案中每个类别只允许一个特征图。然后，通过用全局平均池化替换 dropout + 全连接层，创建了一个等效的网络。这些模型在 CIFAR-10 数据集上进行了性能测试。

这个带有全连接层的 CNN 模型只能达到 17.56%的错误率。当添加 dropout 后，我们获得了与 Hinton 等人[5]报告的类似性能（15.99%）。通过在该模型中用全局平均池化替换全连接层，我们获得了 16.46%的错误率，相比没有 dropout 的 CNN 模型提高了 1%。这再次验证了全局平均池化层作为正则化器的有效性。尽管它略逊于 dropout 正则化器的结果，但我们认为全局平均池化对于线性卷积层可能要求过高，因为它需要具有修正激活的线性滤波器来建模各个类别的置信度图。

4.7 NIN 的可视化

我们通过全局平均池化明确强化了 NIN 的最后一个 mlpconv 层的特征图，使其成为各个类别的置信度图，这仅仅是通过更强的局部感受野建模来实现的，例如 NIN 中的 mlpconv。为了了解这一目的的实现程度，我们提取并直接可视化了 CIFAR-10 训练模型的最后一个 mlpconv 层的特征图。

图 4 显示了从 CIFAR-10 测试集中选取的十个类别的一些示例图像及其对应的特征图。预期会在与输入图像的实际类别相对应的特征图中观察到最大的激活，这是通过全局平均池化明确强化的。在实际类别的特征图中，可以观察到最强的激活大致出现在原始图像中物体的相同区域。对于结构化物体，比如图 4 的第二行中的汽车，这一点尤为明显。需要注意的是，这些类别的特征图是仅通过类别信息进行训练的。如果使用物体的边界框进行细粒度标注，预计会获得更好的结果。

可视化再次证明了 NIN（Network in Network）的有效性。这是通过使用 mlpconv（多层感知器卷积）层来更强烈地建模局部感受野实现的。然后，全局平均池化强制学习类别级别的特征图。进一步的探索可以朝着通用物体检测方向进行。检测结果可以根据类别级别的特征图实现，类似于 Farabet 等人[20]在场景标记工作中的做法。

5 结论

我们提出了一种名为“网络中的网络”（Network In Network，简称 NIN）的新型深度网络，用于分类任务。这种新结构包含使用多层感知器来卷积输入的 mlpconv 层，以及作为传统 CNN 中全连接层替代品的全局平均池化层。mlpconv 层更好地建模局部块，而全局平均池化层作为结构正则化器，全局防止过拟合。通过 NIN 的这两个组件，我们在 CIFAR-10、CIFAR-100 和 SVHN 数据集上展示了最先进的性能。通过特征图的可视化，我们证明了 NIN 最后一个 mlpconv 层的特征图是类别的置信度图，这激发了通过 NIN 进行物体检测的可能性。

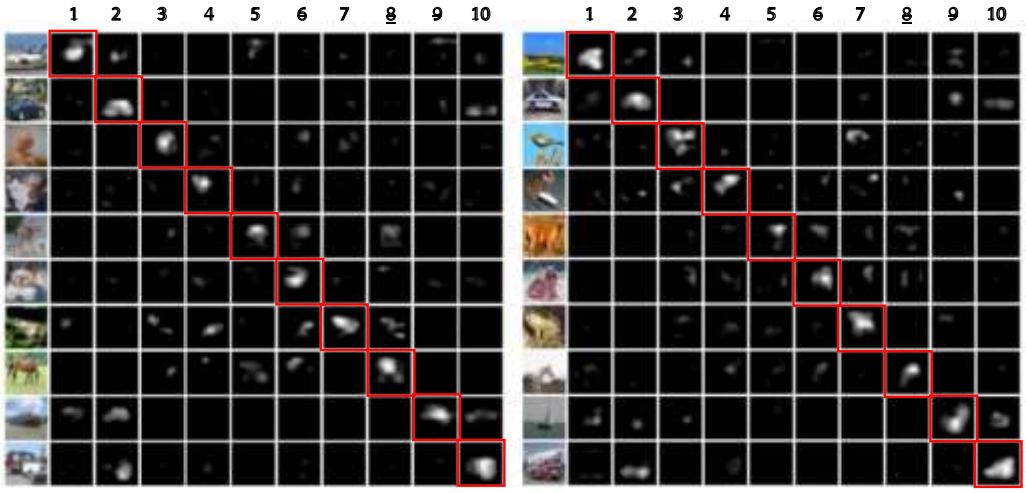


图 4：来自最后一个 mlpconv 层的特征图可视化。仅显示了特征图中激活程度最高的 10%。与特征图对应的类别为：1. 飞机，2. 汽车，3. 鸟，4. 猫，5. 鹿，6. 狗，7. 青蛙，8. 马，9. 船，10. 卡车。对应于输入图像真实标签的特征图被突出显示。左侧面板和右侧面板只是不同的示例。

参考资料

[1] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner.´ Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[2] Y Bengio, A Courville, and P Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35:1798–1828, 2013.

[3] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, DTIC Document, 1961.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.

[5] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[6] Quoc V Le, Alexandre Karpenko, Jiquan Ngiam, and Andrew Ng. Ica with reconstruction cost for efficient overcomplete feature learning. In *Advances in Neural Information Processing Systems*, pages 1017–1025, 2011.

[7] Ian J Goodfellow. Piecewise linear multilayer perceptrons and dropout. *arXiv preprint arXiv:1301.5088*, 2013.

[8] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

[9] C¸aglar G˘ ulc¸ehre and Yoshua Bengio. Knowledge matters: Importance of prior information for¨ optimization. *arXiv preprint arXiv:1301.4083*, 2013.

[10] Henry A Rowley, Shumeet Baluja, Takeo Kanade, et al. *Human face detection in visual scenes*. School of Computer Science, Carnegie Mellon University Pittsburgh, PA, 1995.

[11] Matthew D Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013.

[12] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[13] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, volume 2011, 2011.

[14] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *arXiv preprint arXiv:1206.2944*, 2012.

[15] Li Wan, Matthew Zeiler, Sixin Zhang, Yann L Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1058–1066, 2013.

[16] Mateusz Malinowski and Mario Fritz. Learnable pooling regions for image classification. *arXiv preprint arXiv:1301.3516*, 2013.

[17] Nitish Srivastava and Ruslan Salakhutdinov. Discriminative transfer learning with tree-based priors. In *Advances in Neural Information Processing Systems*, pages 2094–2102, 2013.

[18] Nitish Srivastava. *Improving neural networks with dropout*. PhD thesis, University of Toronto, 2013.

[19] Ian J Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv preprint arXiv:1312.6082*, 2013.

[20] Clement Farabet, Camille Couprie, Laurent Najman, Yann Lecun, et al. Learning hierarchical´ features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1915–1929, 2013.