Compressing Convolutional Neural Networks in the Frequency Domain

ABSTRACT

Convolutional neural networks(CNN)are increasingly used in many areas of computer vision. They are particularly attractive because of their ability to “absorb” great quantities of labeled data through millions of parameters. However, as model sizes increase, so do the storage and memory requirements of the classifiers, hindering many applications such as image and speech recognition on mobile phones and other devices. In this paper, we present a novel network architecture, Frequency-Sensitive Hashed Nets (FreshNets), which exploits inherent redundancy in both convolutional layers and fully-connected layers of a deep learning model, leading to dramatic savings in memory and storage consumption. Based on the key observation that the weights of learned convolutional filters are typically smooth and low-frequency, we first convert filter weights to the frequency domain with a discrete cosine transform (DCT) and use a low-cost hash function to randomly group frequency parameters into hash buckets. All parameters assigned the same hash bucket share a single value learned with standard backpropagation. To further reduce model size, we allocate fewer hash buckets to high-frequency components, which are generally less important. We evaluate FreshNets on eight data sets, and show that it leads to better compressed performance than several relevant baselines.

卷积神经网络（CNN）越来越多地用于计算机视觉的许多领域。它们特别有吸引力，因为它们能够通过数百万个参数“吸收”大量的标记数据。然而，随着模型尺寸增加，分类器的存储和存储器需求也增加，阻碍了诸如移动电话和其他设备上的图像和语音识别的许多应用。在本文中，我们提出一个新的网络架构，频率敏感散列网络（FreshNets），利用卷积层和深度学习模型的完全连接层的固有冗余，导致显着节省内存和存储消耗。基于学习的卷积滤波器的权重通常是平滑和低频的关键观察，我们首先利用离散余弦变换（DCT）将滤波器权重转换到频域，并使用低成本哈希函数来随机地对频率参数进行分组到哈希桶。分配了相同散列桶的所有参数共享使用标准反向传播学习的单个值。为了进一步减小模型大小，我们将较少的哈希桶分配给高频分量，这通常不太重要。我们评估FreshNets上的八个数据集，并表明它导致比几个相关基线更好的压缩性能。

CCS Concepts

•Information systems → Data mining;

•Computing methodologies→Supervised learning; Neural networks;

Keywords

Model compression; convolutional neural networks; hashing

CCS概念

•信息系统→数据挖掘; •计算方法→监督学习;神经网络;

关键词

模型压缩; 卷积神经网络; 散列

1. INTRODUCTION

In the recent years convolutional neural networks (CNN) have led to impressive results in image classiﬁcation [15,30,48,63], object detection [19,58], image retrieval [43], image caption generation[26,37,58], face veriﬁcation[47,55], video understanding[27] and audio classiﬁcation [35]. Problems that seemed impossibly hard only ﬁve years ago can now be solved at better than human accuracy [24]. Although CNNs have been known for a quarter of a century [17], only recently have their superb generalization abilities been accepted widely across the machine learning and computer vision communities. This broad acceptance coincides with the release of very large collections of labeled data [12]. Deep networks and CNNs are particularly well suited to learn from large quantities of data, in part because they can have arbitrarily many parameters. As data sets grow, so do model sizes. In 2012, the ﬁrst winner of the ImageNet competition that used a CNN had already 240MB of parameters and the most recent winning model,in 2014, required 567MB [50].

1.引言

近年来卷积神经网络（CNN）在图像分类[15,30,48,63]，对象检测[19,58]，图像检索[43]，图像标题生成[26,37] 58]，面部验证[47,55]，视频理解[27]和音频分类[35]。几年前似乎不可能困难的问题现在可以比人类的准确性更好地解决[24]。尽管CNNs已经有四分之一世纪的历史[17]，但是最近才有机会学习和计算机视觉社区广泛接受其极好的泛化能力。这种广泛的接受与发布非常大的标记数据集合[12]。深度网络和CNN特别适合于从大量数据中学习，部分是因为它们可以具有任意多个参数。随着数据集增长，模型大小也随之增长。 2012年，使用CNN的ImageNet竞争的第一个赢家已经有240MB的参数，最近的获奖模型，在2014年，需要567MB [50]。

Independently, there has been another parallel shift of computing from servers and workstations to mobile platforms. As of January 2014 there have already been more web searches through smart phones than computers. Today speech recognition is primarily used on cell phones with intelligent assistants such as Apple’s Siri, Google Now or Microsoft’s Cortana. As this trend continues, we are expecting machine learning applications to also shift increasingly towards mobile devices. However, the disjunction of deep learning with ever increasing model sizes and mobile computing reveals an inherent dilemma. Mobile devices have tight memory and storage limitations. For example, even the most recent iPhone 6 only features 1GB of RAM, most of which must be used by the operating system or the application itself. In addition, developers must make their apps compatible with the most limited phone still in circulation, often restricting models to just a few megabytes of parameters. In addition to memory limitations, there are also low-power restrictions. Loading 500MB of parameters for a deep network from SSD drive to memory, to maybe just classify a single image, requires a significant amount of energy.

独立地，计算从服务器和工作站到移动平台的另一个并行转移。截至2014年1月，通过智能手机已经有更多的网络搜索，而不是计算机。今天，语音识别主要用于具有智能助手的手机，例如苹果的Siri，Google Now或Microsoft的Cortana。随着这种趋势的继续，我们期待机器学习应用程序也越来越趋向于移动设备。然而，深度学习与不断增加的模型大小和移动计算的分离揭示了固有的困境。移动设备具有严格的内存和存储限制。例如，即使最近的iPhone 6只有1GB的RAM，其中大部分必须由操作系统或应用程序本身使用。此外，开发人员必须使他们的应用程序与仍在传播中的最有限的手机兼容，通常将模型限制为仅几兆字节的参数。除了内存限制，还有低功耗限制。从SSD驱动器到内存，将深度网络的500MB参数加载到内存，或者只是分类单个映像，需要大量的能量。

In response, there has been a recent interest in reducing the model sizes of deep networks. Denil et al. [13] use low-rank decomposition of the weight matrices to reduce the effective number of parameters in the network. Bucilua et al. [5] and Ba et al. [2] show that complex models can be compressed into 1-layer neural networks. Independently, the model size of neural networks can be reduced effectively through reduced bit precision [9]. Meanwhile, the conventional wisdom that fully connected layers contain more parameters is no longer true. More and more parameters are shifting towards convolutional layers. For example, GoogleNet [54] contains 11 layers involving convolutions with only one single fully connected layer on the top. As a result, 85% of the parameters lie in the convolutional layers. A recent work proposed by Long et al. [36] advocates fully convolutional training which replaces the fully connected layers with the convolutional layer. All of these make the compression of the convolutional layers a must.

作为响应，近来对减小深度网络的模型大小感兴趣。 Denil et al。 [13]使用权重矩阵的低秩分解来减少网络中的参数的有效数量。 Bucilua et al。 [5]和Ba et al。 [2]表明复杂的模型可以压缩成1层神经网络。独立地，神经网络的模型大小可以通过降低位精度有效地减少[9]。同时，完全连接的层包含更多参数的传统观点不再是真实的。越来越多的参数转向卷积层。例如，GoogleNet [54]包含11层，包括卷积，顶部只有一个完全连接的层。结果，85％的参数位于卷积层中。 Long等人提出的最近的工作[36]提倡完全卷积训练，用卷积层代替完全连接的层。所有这些使卷积层的压缩成为必须。

In this paper we propose a novel approach for neural network compression targeted especially for CNNs. We build on recent work by Chen et al. [6], who show that weights of fully connected networks can be effectively compressed with the hashing trick[60]. Due to the nature of local pixel correlation in images (i.e. spatial locality), ﬁlters in CNNs tend to be smooth. We transform these ﬁlters into frequency domain with the discrete cosine transform (DCT) [42]. In frequency space, the ﬁlters are naturally dominated by low frequency components. Our compression takes this smoothness property into account and randomly hashes the frequency components of all CNN ﬁlters at a given layer into one common set of hash buckets. All components inside one hash bucket share the same value. As lower frequency components are more pronounced than higher frequencies, we allow collisions only between similar frequencies and allocate fewer hash buckets for the high frequencies (which are less important).

在本文中，我们提出一种特别针对CNN的神经网络压缩的新方法。我们基于Chen等人最近的工作。 [6]，他们表明，完全连接的网络的权重可以有效地压缩与哈希的技巧[60]。由于图像中的局部像素相关性（即空间局部性）的性质，CNN中的滤波器倾向于是平滑的。我们用离散余弦变换（DCT）将这些滤波器变换到频域[42]。在频率空间中，滤波器自然地由低频分量占据。我们的压缩考虑了这个平滑性属性，并且将给定层处的所有CNN过滤器的频率分量随机地散列成一个公共的哈希桶。一个哈希桶内的所有组件共享相同的值。因为较低频率分量比较高频率更显着，所以仅允许在相似频率之间的冲突，并且为高频（其不那么重要）分配较少的哈希桶。

Our approach has several compelling properties:

1. The number of parameters in the CNN is independent of the number of convolutional ﬁlters;

2. During testing we only need to add a low-cost hash function and the inverse DCT transformation to any existing CNN code for ﬁlter reconstruction;

3. During training, the hashed weights can be learned with simple back-propagation[3]—the gradient of a hash bucket value is the sum of gradients of all hashed frequency components in that bucket.

We evaluate our compression scheme on eight deep learning image benchmark data sets and compare against four competitive baselines. Although all compression schemes lead to lower test accuracy as the compression increases, our FreshNets method is by far the most effective compression method and yields the lowest generalization error rates on almost all tested tasks. The rest of the paper is organized as follows. Section 2 introduces background on feature hashing and discrete cosine transformation. We describe FreshNets in Section 3 and review the literature in Section 5. Experimental results are presented in Section 6. Section 7 draws the conclusion.

我们的方法有几个引人注目的属性：

CNN中的参数的数量与卷积滤波器的数量无关;

在测试期间，我们只需要为任何现有的CNN代码添加一个低成本哈希函数和逆DCT变换，用于滤波器重建;

在训练期间，可以通过简单的反向传播学习散列权重[3] - 散列桶值的梯度是该桶中所有散列频率分量的梯度之和。

我们对八个深度学习图像基准数据集的压缩方案进行评估，并与四个竞争性基线进行比较。虽然所有压缩方案导致压缩增加时测试精度降低，但我们的FreshNets方法是最有效的压缩方法，并且在几乎所有测试任务上产生最低的广义误差率。本文的其余部分安排如下。第2节介绍了特征散列和离散余弦变换的背景。我们在第3节中描述FreshNets并且在第5节中回顾文献。实验结果在第6节中给出。第7节得出结论。

2. BACKGROUND

2.1 Feature Hashing

Feature Hashing (a.k.a the hashing trick) [11,49,60] has been previously studied as a technique for reducing model storage size. In general, it can be regarded as a dimensionality reduction method that maps an input vector x ∈ Rd to a much smaller feature space via a mapping φ:Rd → Rk where k d. The mapping φ is a composite of two approximately uniform auxiliary hash functions h:N→{1,...,k}and ξ:N→{−1,+1}. The jth element of the k-dimensional hashed input is deﬁned as

φj(x) = X i:h(i)=j ξ(i) xi.

As shown in [60], a key property of feature hashing is its preservation of inner product operations,where inner products after hashing produce the correct pre-hash inner product in expectation: E[φ(x)>φ(y)]φ = x>y. This property holds because of the bias correcting sign factor ξ(i). With feature hashing, models are directly learned in the much smaller space Rk, which not only speeds up training and evaluation but also signiﬁcantly conserves memory. For example, a linear classiﬁer in the original space could occupy O(d) memory for model parameters, but when learned in the hashed space only requires O(k)parameters. The information loss induced by hash collisionis much less severe for sparse feature vectors and can be counteracted through multiple hashing [49] or larger hash tables [60].

2.背景

2.1特征哈希

特征哈希（a.k.a哈希技巧）[11,49,60]先前已经被研究作为用于减少模型存储大小的技术。一般来说，可以将其视为一种维数降低方法，其通过映射将输入向量x∈Rd映射到小得多的特征空间：其中， d。映射φ是两个近似均匀的辅助散列函数h：N→{1，...，k}和ξ：N→{-1，+ 1}的合成。 k维散列输入的第j个元素定义为

 φj（x）= Xi：h（i）=jξ（i）xi。

 如[60]所示，特征散列的关键特性是其保留内积运算，其中散列后的内积产生期望的正确预散列内积：E [φ（x）>φ（y）]φ = x> y。该属性由于偏差校正符号因子ξ（i）而成立。使用特征哈希，模型直接在更小的空间Rk中学习，这不仅加快了训练和评估，还显着节省了内存。例如，原始空间中的线性分类器可以占用模型参数的O（d）存储器，但是当在哈希空间中学习时仅需要O（k）参数。哈希冲突引起的信息丢失对稀疏特征向量的影响要小得多，并且可以通过多重哈希[49]或更大的哈希表来抵消[60]。

2.2 Discrete Cosine Transform(DCT)

Methods built on the DCT [1,42] are widely used for compressing images and movies, including forming the standard technique for JPEG [41,59]. DCT expresses a function as a weighted combination of sinusoids of different phases/ frequencies where the weight of each sinusoid reﬂects the magnitude of the corresponding frequency in the input. When employed with sufﬁcient numerical precision and without quantization or other compression operations, the DCT and inverse DCT (projecting frequency inputs back to the spatial domain) are lossless. Compression is made possible inimagesbylocalsmoothnessofpixels(e.g. abluesky)whichcan be well represented regionally by fewer non-zero frequency components. Though highly related to the discrete Fourier transformation(DFT), DCT is often preferable for compression tasks because of its spectral compaction property where weights for most images tend to be concentrated in a few low-frequency components of the DCT [42]. Further, the DCT transformation yields a real-valued representation ,unlike the DFT whose representation has imaginary components. Given an input matrix V ∈Rd×d, the corresponding matrixV∈Rd×d in frequency domain after DCT is deﬁned as:

Vj1j2 = sj1sj2 d−1 X i1=0 d−1 X i2=0 c(i1,i2,j1,j2) Vi1i2, (1) where c(i1,i2,j1,j2) = cosπ di1 + 1 2j1cosπ di2 + 1 2j2 is the cosine basis function,and sj =q1 d when j=0 and sj =q2 d otherwise. We use the shorthand fdct to denote the DCT operation in Eq. (1), i.e. V = fdct(V ). The inverse DCT convertsV from the frequency domain back to the spatial domain,reconstructing V without loss:

Vi1i2 =

d−1 X j1=0

d−1 X j2=0

sj1sj2 c(i1,i2,j1,j2)Vj1j2. (2)

We denote the inverse DCT function in Eq. (2) as f−1 dct, i.e. V = f−1 dct(V).

2.2离散余弦变换（DCT）

基于DCT的方法[1,42]广泛用于压缩图像和电影，包括形成JPEG的标准技术[41,59]。 DCT表示作为不同相位/频率的正弦曲线的加权组合的函数，其中每个正弦曲线的权重反映输入中的相应频率的幅度。当使用具有足够的数值精度和没有量化或其他压缩操作时，DCT和反DCT（投影频率输入回到空间域）是无损的。通过局部地由较少的非零频率分量可以很好地表示的像素（例如，abluesky）的局部平滑性使得压缩成为可能。虽然与离散傅立叶变换（DFT）高度相关，但DCT通常优选用于压缩任务，因为其频谱压缩属性，其中大多数图像的权重倾向于集中在DCT的几个低频分量中[42]。此外，与其表示具有虚分量的DFT不同，DCT变换产生实值表示。给定输入矩阵V∈Rd×d，DCT后的频域中对应的矩阵V∈Rd×d定义为：

Vj1j2 = sj1sj2 d-1 X i1 = 0 d-1 X i2 = 0 c（i1，i2，j1，j2）Vi1i2其中c（i1，i2，j1，j2）= cos + 1 2＆pi; j 1 cos＆phi; d d i 2 + 1 2 j 2＆是余弦基函数，并且当j = 0和sj = q2 d时sj = q1 d否则。我们使用简写fdct来表示等式1中的DCT运算。 （1），即V = fdct（V）。逆DCT将V从频域变换回空间域，无损地重建V：

Vi1i2 =

d-1 X j1 = 0

d-1 X j2 = 0

sj1sj2 c（i1，i2，j1，j2）Vj1j2。 （2）

我们表示等式中的逆DCT函数。 （2）表示为f-1 dct，即V = f-1 dct（V）。

3. Frequency-Sensitive Hashed Nets

Here we present FreshNets, a method for using weight sharing to reduce the model size (and memory demands) of convolutional neural networks. Similar to the work of Chen et al. [6], we achieve smaller models by randomly forcing weights throughout the network to share identical values. Unlike previous work, we implement the weight sharing and gradient updates of convolutional ﬁlters in the frequency domain. These sharing constraints are made prior to training, and we learn frequency weights under the sharing assignments. Since the assignments are made with a hash function, they incur no additional storage. We begin by deriving the equivalent ﬁlter representation after the DCT, and describe an efﬁcient random weight-sharing scheme in the frequency space implemented using the hashing trick. Next we show how to learn the parameters in the frequency domain with standard back-propagation. At last, we describe a scheme to take advantage of ﬁlter smoothness by allocating more shared weights to low frequency components

频率敏感的哈希网

在这里，我们提出FreshNets，一种使用权重共享减少卷积神经网络的模型大小（和内存需求）的方法。类似于Chen等人的工作。 [6]，我们通过随机强迫整个网络的权重来实现更小的模型，以共享相同的值。与以前的工作不同，我们在频域中实现卷积滤波器的权重共享和梯度更新。这些共享约束在训练之前进行，并且在共享分配下学习频率权重。由于分配是使用散列函数进行的，它们不会产生额外的存储空间。我们开始通过在DCT之后导出等效滤波器表示，并且描述在使用散列技巧实现的频率空间中的有效随机权重共享方案。接下来，我们将展示如何使用标准反向传播学习频域中的参数。最后，我们描述了一种利用滤波器​​平滑性的方案，通过向低频分量分配更多的共享权重

3.1 Filters in spatial and frequency domain

3.1空间和频域滤波器

Let the matrix V denote the weight matrix of the d×d convolutional ﬁlter that connects the k input plane to the l output plane. (For notational convenience we assume square ﬁlters and only consider the ﬁlters in a single layer of the network.) The weights of all ﬁlters in a convolutional layer can be denoted by a 4-dimensional tensor V ∈Rm×n×d×d where m and n are the number of input planes and output planes, respectively, resulting in a total of m×n×d2 parameters. Convolutional ﬁlters can be represented equivalently in either the spatial or frequency domain, mapping between the two via the DCT and its inverse. We denote the ﬁlter in frequency domain as Vk` = fdct(V k`) ∈Rd

令矩阵V表示将k输入平面连接到l输出平面的d×d卷积滤波器的权重矩阵。 （为了标记方便，我们假设平方滤波器，并且仅考虑网络的单个层中的滤波器。）卷积层中的所有滤波器的权重可以由4维张量V∈Rm×n×d×d来表示，其中 m和n分别是输入平面和输出平面的数量，导致总共m×n×d2个参数。 卷积滤波器可以在空间或频域中等效地表示，通过DCT及其逆在两者之间进行映射。 我们将频域中的滤波器表示为Vk`= fdct（V k`）∈Rd

3.2 Random Weight Sharing by Hashing

3.2通过哈希的随机权重分配

After the lossless conversion via DCT, ﬁlters in frequency domain V remain the same size as equivalent ﬁlters in the spatial domain V. We propose to use weight sharing to reduce the number of parameters in the frequency domain. With ﬁlters in a frequency representation, V, we would like to reduce the number of model parameters to exactly K values stored in a weight vector w∈RK, where K m×n×d2. To achieve this, we randomly assign a value from w to each ﬁlter frequency weight in V. A naïve implementation of this random weight sharing would introduce an auxiliary matrix for V to track the weight assignments, using to signiﬁcant additional memory. To address this problem, Chen et al. [6] advocate use of the hashing trick to (pseudo-)randomly assign shared parameters. Using the hashing trick, we tie each ﬁlter weight Vk` j1j2 to an element of w indexed by the output of a hash function h(·):

在通过DCT的无损转换之后，频域V中的滤波器保持与空间域V中的等效滤波器相同的大小。我们提出使用权重共享来减少频域中的参数的数量。 使用在频率表示中的滤波器V，我们希望将模型参数的数目精确地减少到存储在权重向量w∈RK中的K个值， m×n×d2。 为了实现这一点，我们从w中随机分配一个值到每个滤波器频率权重。这个随机权重共享的一个简单的实现将引入一个辅助矩阵V跟踪权重分配，使用显着的额外内存。 为了解决这个问题，Chen et al。 [6]倡导使用散列技巧（伪）随机分配共享参数。 使用散列技巧，我们将每个过滤器权重Vkjjj2绑定到由散列函数h（·）的输出索引的w的元素：

where h(k,`,j1,j2)∈{1,··· ,K}, and ξ(k,`,j1,j2)∈{±1} is a sign factor computed by a second hash function ξ(·) to preserve inner-products in expectation as described in Section 2. With the mapping in Eq. (5), we can implement shared parameter assignments with no additional storage cost. For a schematic illustration, see Figure 1. The ﬁgure also incorporates a frequency sensitive hashing scheme discussed later in this section. Note that the same K weights in w are shared across all ﬁlters in the convolutional layer. This way,we compress the whole Vk` which contains mnd2 parametersintoa K-dimensional weight vector w. In other words, adding more convolutional ﬁlters does not change the model size, as all ﬁlter values are “recycled” from already existing ﬁlters. We can arbitrarily control the number of effective parameters in each layer simply by adjusting K.

其中h（k，`，j1，j2）∈{1，...，K}和ξ（k，`，j1，j2）∈{±1}是由第二散列函数ξ ·）保留内部产品的期望，如第2节所述。 （5），我们可以实现共享的参数分配，没有额外的存储成本。 有关示意图，请参见图1.该图还包含本节后面讨论的频率敏感散列方案。 注意，w中相同的K权重在卷积层中的所有滤波器之间共享。 这样，我们将包含mnd2个参数的整个Vk`压缩到K维权重向量w。 换句话说，添加更多的卷积滤波器不会改变模型大小，因为所有的滤波器值都是从已经存在的滤波器“回收”。 我们可以通过调整K来任意控制每一层中的有效参数的数量。

3.3 Gradients over Shared Frequency Weights

3.3共享频率上的梯度

Typical convolutional neural networks learn ﬁlters in the spatial domain. As our shared weights are stored in the frequency domain, we derive the gradient with respect to the ﬁlter parameters in the frequency space. Following Eq. (2), we express the gradient of parameters in the spatial domain with respect to their counterparts in the frequency domain:

典型的卷积神经网络学习空间域中的滤波器。 由于我们的共享权重存储在频域，我们导出相对于频率空间中的滤波器参数的梯度。 按照公式 （2）中，我们表示在空间域中相对于它们在频域中的对应物的参数的梯度：

Let L be the loss function adopted for training. Using standard back-propagation, we can derive the gradient with respect to the ﬁlter parameters in the spatial domain,

令L为训练采用的损失函数。 使用标准反向传播，我们可以导出相对于空间域中的滤波器参数的梯度

3.4 Frequency Sensitive Hashing

频率敏感哈希

Figure 2 shows a ﬁlter in spatial (left) and frequency (right) domains. In the spatial domain CNN ﬁlters are smooth [30] due to the local pixel smoothness in natural images. In the frequency domain this corresponds to components with large magnitudes in the low frequencies, depicted in the upper left half of Vk` in Figure 2, with small indices (j1,j2). Correspondingly, the high frequencies, in the bottom right half of Vk`, with large indices (j1,j2), have magnitudes near zero.

As stated earlier, the ﬁlters in the spatial domain V k` are typically “smooth” and in the frequency domain most weight intensities are in the low-frequency regions. The low frequency regions correspond to entries with small indices (j1,j2) and the low frequency entries typically have much larger norms than higher frequency values (entries with larger indices (j1,j2).

图2显示了在空间（左）和频率（右）域中的滤波器。 在空间域CNN滤波器是平滑的[30]由于自然图像中的局部像素平滑。 在频域中，这对应于在低频中具有大幅度的分量，在图2中的Vk'的左上半部中示出，具有小的索引（j1，j2）。 相应地，在具有大指数（j1，j2）的Vk'的右下半部中的高频具有接近零的量值。

如前所述，空间域V k'中的滤波器通常是“平滑的”，并且在频域中，大多数权重强度在低频率区域中。 低频区域对应于具有小索引（j1，j2）的条目，并且低频条目通常具有比较高频率值（具有较大索引（j1，j2）的条目）大得多的范数。

As components of different frequency regions tend to be of different magnitudes (and thereby varying importance to the spatial structure of the ﬁlter), we want to avoid collisions between high and low frequency components. Therefore, we assign separate hash spaces to different frequency regions

由于不同频率区域的分量倾向于具有不同的幅度（并且由此对滤波器的空间结构具有不同的重要性），因此我们希望避免高频分量和低频分量之间的冲突。 因此，我们将不同的散列空间分配给不同的频率区域

We deﬁne a compression rate rj ∈ (0,1] for each frequency region j and assign Kj = rjNj where Nj is the number of virtual parameters in the jth frequency regions. A smaller rj induces more collisions during hashing, leading to increased weight sharing. Since lower frequency components tend to be of higher importance, making collisions more hurtful, we commonly assign larger rj (fewer collisions) to low-frequency regions. Intuitively, given a size budget for the whole convolutional layer, we want to squeeze the hash space of high frequency region to save space for low frequency regions. These compression rates can either be assigned by hand or determined programmatically by cross-validation, as demonstrated in Section 6.

我们为每个频率区域j定义压缩率rj∈（0,1]，并且分配Kj = rjNj，其中Nj是第j个频率区域中的虚拟参数的数量，较小的rj在散列期间引发更多的冲突，导致权重共享 由于较低频率分量往往具有较高的重要性，使得冲突更有害，我们通常将较大的rj（较少的冲突）分配给低频区域。直观地，给定整个卷积层的大小预算，我们要挤压散列 高频区域的空间，以节省低频区域的空间，这些压缩率可以手动分配或通过交叉验证程序化确定，如第6节所示。

4. OVERVIEW

In a nutshell, we summarize the training procedure of FreshNets as follows. Initialization phase

• Given the ﬁlter size d and an overall budget for the number of parameters K, determine the budget for each frequency region Kj such that

• Randomly initialize weight vector w which consists of 2d−1 number of sub-vectors wj.

Feedforward phase:

• Construct Vk` according to Eq (11).

• Use inverse DCT to convert Vk` to its spatial domain V k` according to (4).

• Use V k` to perform convolution on the inputs, and generate outputs.

Backpropagation phase:

• Compute the gradient w.r.t. the convolutional ﬁlters V k` in the spatial domain using normal backpropagation.

• Compute the gradient w.r.t. Vk` using DCT according to (8).

• Compute the gradient w.r.t. the real weight vector w according to Eq(12) and update w using any gradient descent methods such as SGD with momentum [53], Adagrad [16] and RMSprop [56].

All the above training operations can be implemented in existing CNN packages by modifying the computational procedure for back propagation.

4.概述

简而言之，我们总结FreshNets的培训程序如下。初始化阶段

·给定滤波器尺寸d和参数K的数量的总预算，确定每个频率区域K j的预算，使得

·随机初始化由2d-1个子向量wj组成的权重向量w。

前馈阶段：

•根据式（11）构造Vk'。

·根据（4）使用逆DCT将V k'转换为其空间域V k'。

•使用V k`对输入执行卷积，并生成输出。

反向传播阶段：

•计算梯度w.r.t.卷积滤波器V k'在空间域中使用正常反向传播。

•计算梯度w.r.t. Vk'使用根据（8）的DCT。

•计算梯度w.r.t.根据等式（12）的实权重向量w和使用任何梯度下降方法（诸如具有动量的SGD [53]，Adagrad [16]和RMSprop [56]）更新w。

所有上述训练操作可以通过修改反向传播的计算程序在现有的CNN包中实现。

5. RELATED WORK

Several recent studies have conﬁrmed that there is signiﬁcant redundancy in the parameters learned in deep neural networks. Recent work by Denil et al. [13] learns parameters in fully-connected layers after decomposition into two low-rank matrices, i.e. W = AB where W ∈Rm×n, A∈Rm×k and B ∈ Rk×n. In this way, theoriginal O(mn) parameterscouldbestoredwith O(k(m+n)) storage, where k min(m,n).

Several works apply related approaches to speed up the evaluation time with convolutional neural networks. Two works propose to approximate convolutional ﬁlters by a weighted linear combination of basis ﬁlters [25,45]. In this setting, the convolution operation only needs to be performed with the small set of basis ﬁlters instead of all the ﬁlters. The desired output feature maps are computed by matrix multiplication as the weighted sum of these basis convolutions.

Further speedup can be achieved by learning rank-one basis ﬁlters so that the convolution operations are very cheap to compute[14, 33]. Based on this idea, Denton et al. [14] advocate decomposing the four-dimensional tensor of the ﬁlter weights into a sum of different rank-one, four-dimensional tensors and show some encouraging results. In addition, they adopt bi-clustering to group ﬁlters such that each subgroup can be better approximated by rank-one tensors. Courbariaux et al. [10] introduce BinaryConnect that enforces weights in neural networks to take on binary values. This replaces many multiply-accumulate operations by simple accumulations, leading to less power-hungry and fast computation.

There is a distinctive difference between FreshNets and the above works. In each of the above works, evaluation time is the main focus, with any resulting storage reduction achieved merely as a side effect. However, with the trend toward architectures with fewer fully connected layers and additional convolutional layers[54],compression of ﬁlters is of increasing importance.

5.相关工作

最近的一些研究证实，在深层神经网络中学习的参数有重要的冗余。最近的工作Denil et al。 [13]在分解成两个低秩矩阵之后，在完全连接的层中学习参数，即W = AB，其中W∈Rm×n，A∈Rm×k和B∈Rk×n。以这种方式，原始O（mn）参数可以与O（k（m + n））存储，其中k min（m，n）。

几个工程应用相关的方法来加快卷积神经网络的评估时间。两个工作提出通过基本滤波器的加权线性组合近似卷积滤波器[25,45]。在这个设置中，卷积运算只需要使用一小组基本过滤器而不是所有过滤器来执行。通过矩阵乘法计算期望的输出特征映射作为这些基本卷积的加权和。

进一步的加速可以通过学习一级基本过滤器来实现，使卷积运算非常便宜计算[14，33]。基于这个想法，Denton et al。 [14]主张将滤波器权重的四维张量分解为不同一阶四维张量的总和，并显示一些令人鼓舞的结果。此外，他们采用双聚类分组过滤器，使每个子组可以更好地近似一级张量。 Courbariaux et al。 [10]介绍BinaryConnect，它在神经网络中实施权重以获取二进制值。这通过简单的累加代替了许多乘法累加运算，导致更少的功耗和快速的计算。

FreshNets和上述作品之间有明显的区别。在上述每个工作中，评估时间是主要焦点，任何所得到的存储减少仅仅作为副作用实现。然而，随着具有更少的完全连接层和额外的卷积分布的架构的趋势[54]​​，对滤波器的压缩越来越重要。

Another technique for speeding up convolutional neural network evaluation is computing convolutions in the Fourier frequency domain, as convolution in the spatial domain is equivalent to (comparatively lower-cost) element-wise multiplication in the frequency domain [38,57]. Unlike FreshNets, for a ﬁlter of size d×d and an image of size n×n where n > d, Mathieu et al. [38] convert the ﬁlter to its frequency domain of size n × n by oversampling the frequencies, which is necessary for doing element-wise multiplication with a larger image but also increases the memory overhead at test time. Training in the Fourier frequency domain may be advantageous for similar reasons, particularly when convolutions are being performed over large 3-D volumes [4].

Other works focus entirely on compressing the fully-connected layers of CNNs. A branch of these works propose to post-process a trained convolutional net. Leveraging the similarity between connections, Gong et al. [20] learns a convolutional net in advance, and then applies kmeans clustering on the weight values for quantization. These clusters form a smaller-size codebook for all the weights. With similar spirit, another work by Han et al. [23] recursively train a neural network and prune unimportant connections based on their weight magnitude. Han et al. [22] further combine the techniques of pruning and quantization to achieve more compression.

A few recent works [52,61] also adopt the quantization techniques. However, with quantization, each connection still needs to store the index to the cluster it belongs to,which limits its potential for compression. Other works focus on matrix or tensor decomposition for compression. For example, Yang et al. [62] adopt the fastfood transformation [32] for compressing the fully connected layers, and Kim et al. [28] investigate general tensor decomposition with rank selection for compressing the entire network. Most relevant to this work is HashedNets [6] which compresses the fully connected layers of deep neural networks. This method uses the hashing trick to efﬁciently implement parameter sharing prior to learning, achieving notable compression with less loss of accuracy than the competing baselines which relied on low-rank decomposition or learning in randomly sparse architectures.

用于加速卷积神经网络评估的另一种技术是在傅里叶频域中的计算卷积，因为空间域中的卷积等价于频域中的（相对较低成本的）逐元素乘法[38,57]。与FreshNets不同，对于大小为d×d的滤波器和大小为n×n的图像，其中n> d，Mathieu et al。 [38]通过对频率进行过采样将滤波器转换到大小为n×n的频域，这对于与较大图像进行元素级乘法是必要的，但也增加了测试时间的存储器开销。在傅里叶频域中的训练可能有利于类似的原因，特别是当卷积在大的3-D卷上执行时[4]。

其他工作完全侧重于压缩CNN的完全连接的层。这些工作的一个分支建议后处理一个训练的卷积网。利用连接之间的相似性，Gong等人[20]提前学习卷积网络，然后对量化值应用k mean聚类。这些簇为所有权重形成较小尺寸的码本。用类似的精神，Han等人的另一项工作递归地训练神经网络并基于它们的权重大小修剪不重要的连接。 Han et al。 [22]进一步组合修剪和量化的技术来实现更多的压缩。

一些最近的作品[52,61]也采用量化技术。然而，使用量化，每个连接仍然需要保存在集群中，这限制了压缩的可能性。其他着作集中在矩阵或张量分解压缩。例如，Yang et al。 [62]采用快速变换[32]压缩完全连接的层，Kim等人[28]调查一般张量分解与秩选择压缩整个网络。与这项工作最相关的是HashedNets [6]，它压缩了深层神经网络的完全连接层。该方法使用散列技巧在学习之前有效地实现参数共享，实现显着的压缩，与基于低秩分解或在随机稀疏结构中学习的竞争基线相比具有更少的精度损失。

Another recent work by Rippel et al.[46] proposes to learn weight parameters in the frequency domain for faster convergence of learning, which is a fundamentally different goal compared with FreshNets. In addition, they adopts discrete Fourier transformation(DFT) [8] while FreshNets uses DCT.

Rippel等人最近的另一项研究[46] 提出在频域中学习更快的收敛学习的权重参数，这是一个根本不同的目标与FreshNets相比。 此外，他们采用离散傅立叶变换（DFT）[8]，而FreshNets使用DCT。

6. EXPERIMENTALRESULTS

In this section, we conduct several comprehensive experiments on several benchmark datasets to evaluate the compression performance of FreshNets.

6.1 Datasets

We experiment with eight benchmark datasets: CIFAR10 [29], CIFAR100[29], SVHN[39] and ﬁve challenging variants of MNIST[31, 34,44]. The CIFAR10 dataset contains 60000 images of 32 × 32 pixels with three color channels. Images are selected from ten classes with each class consisting of 6000 unique instances.

The CIFAR100 dataset also contains 60000 32×32 images, but is more challenging since the images are selected from 100 classes (each class has 600 images). For both CIFAR datasets, 50000 images are designated for training and the remaining 10000 images for testing. To improve accuracy on CIFAR100, we augment by horizontal reﬂection and cropping [30], resulting in 0.8M training images.

The SVHN dataset is a large collection of digits (10 classes) cropped from real-world scenes, consisting of 73257 training images,26032 testing images and 531131 less difﬁcult images for additional training.

In our experiments, we use all available training images, for a total of 604388 training samples. For the MNIST variants [31], each variation either reduces the training size (MNIST-07) or amends the original digits by rotation (ROT), background superimposition (BGRAND and BG-IMG), or a combination thereof (BG-ROT). We preprocess all datasets with whitening (except CIFAR100 and SVHN which were prohibitively large).

6.实验结果

在本节中，我们对几个基准数据集进行几个综合实验，以评估FreshNets的压缩性能。

6.1数据集

我们对八个基准数据集进行实验：CIFAR10 [29]，CIFAR100 [29]，SVHN [39]和五个具有挑战性的MNIST [31,34,44]变体。 CIFAR10数据集包含60000个32×32像素的图像，具有三个颜色通道。从十个类中选择图像，每个类由6000个唯一实例组成。

CIFAR100数据集还包含60000个32×32图像，但是更具挑战性，因为图像是从100个类中选择的（每个类有600个图像）。对于两个CIFAR数据集，指定50000个图像用于训练，剩余10000个图像用于测试。为了提高CIFAR100的精度，我们增加水平反射和裁剪[30]，导致0.8M的训练图像。

SVHN数据集是从真实世界场景裁剪的数字（10类）的大集合，包括73257个训练图像，26032测试图像和531131更少的难看的图像用于额外的训练。

在我们的实验中，我们使用所有可用的训练图像，总共604388个训练样本。对于MNIST变体[31]，每个变体通过旋转（ROT），背景重叠（BGRAND和BG-IMG）或其组合（BG-ROT）减少初始化（MNIST-07）我们预处理所有带有白化的数据集（除了CIFAR100和SVHN，这是非常大的）。

6.2 Baselines

We compare the proposed FreshNets with four baseline methods: HashedNets [6], low-rank decomposition (LRD) [13], ﬁlter dropping (DropFilt) and frequency dropping (DropFreq). HashedNets was ﬁrst proposed to compress fully-connected layers in deep neural networks via the hashing trick. In this baseline, we apply the hashing trick directly to the convolutional layer by hashing ﬁlter weights in the spatial domain. This induces random weight sharing across all ﬁlters in a single convolutional layer.

Additionally, we compare against low-rank decomposition of the convolutional ﬁlters [13]. Following the method in [14], we unfold the four-dimensional ﬁlter tensor to form a two dimensional matrix on which we apply the low-rank decomposition. The parameters of the decomposition are ﬁne-tuned via back-propagation. DropFreq learns parameters in the DCT frequency domain but sets high frequency components to 0 to meet the compression requirement. DropFilt compresses simply by reducing the number of ﬁlters in each convolutional layer.

The experimental environment is an off-the-shelve desktop with two 8-core Intel(R) Xeon(R) processors of 2.67 GHz and 128GB RAM.All methods were implemented usingTorch7 [7] and run on NVIDIA GTX TITAN graphics cards with 2688 cores and 6GB of global memory. Model parameters are stored and updated as 32 bit ﬂoating-point values. We further hold out 20% data from the training set as the validation set for early stopping. In particular, if the validation error does not decrease for a number of epochs, the training would stop and output the model with the lowest validation error. Hyperparameters are selected for all algorithms with Bayesian optimization [40,51] and hand tuning on the validation set. We use the open source Bayesian Optimization MATLAB implementation “bayesopt.m” from Gardner et al. [18].4 All our reported results are based on the test error performance.

6.2基线

我们将所提出的FreshNets与四种基线方法：HashedNets [6]，低秩分解（LRD）[13]，滤波器丢弃（DropFilt）和频率丢弃（DropFreq）进行比较。 HashedNets首先提出通过哈希技巧在深层神经网络中压缩全连接层。在这个基线中，我们通过在空间域中对滤波器权重进行哈希处理，将散列技巧直接应用于卷积层。这导致在单个卷积层中的所有滤波器之间的随机权重共享。

此外，我们与卷积滤波器的低秩分解进行比较[13]。根据[14]中的方法，我们展开四维滤波张量形成一个二维矩阵，我们在其上应用低秩分解。分解的参数通过反向传播进行调整。 DropFreq在DCT频域中学习参数，但将高频分量设置为0以满足压缩要求。 DropFilt通过简单地减少每个卷积层中的过滤器的数量来压缩。

实验环境是一个现成的桌面，有两个8核英特尔（R）Xeon处理器2.67 GHz和128GB RAM.All方法使用Torch7 [7]实现，运行在NVIDIA GTX TITAN显卡2688核心和6GB的全局内存。模型参数被存储和更新为32位浮点值。我们还将来自训练集的20％数据作为早期停止的验证集。特别地，如果验证错误不能减少多个时段，则训练将停止并输出具有最低验证错误的模型。选择贝叶斯优化[40,51]和所有算法的超参数。使用来自Gardner等人的开源贝叶斯优化MATLAB实现“bayesopt.m” [18] .4我们所有的报告结果都是基于测试误差性能。

6.3 Comprehensive evaluation

We adopt the neural network architecture shown in Table 1 for all datasets. The architecture is a deep convolutional neural network consisting of ﬁve convolutional layers (with 5×5 ﬁlters) and one fully-connected layer. Before convolution, input feature maps are zero-padded such that output maps remain the same size as the (unpadded) input maps after convolution. Max-pooling is performed after convolutions in layers 2, 4 and 5 with ﬁlter size 2 × 2 and stride 2, reducing both input map dimensions by half. Rectiﬁed linear units are adopted as the activation function throughout. The output of the network is a softmax function over labels.

In this architecture, the convolutional layers hold the majority of parameters (1.2 million in convolutional layer v.s. 40 thousand in the fully connected layer with 10 output classes). During training, we optimize parameters using mini-batch gradient descent with batch size 64 and momentum 0.9. We use 20 percent of the training set as a validation set for early stopping. For FreshNets, we use a frequency-sensitive compression scheme which increases weight sharing among higher frequency components. For all baselines, we apply HashedNets [6] to the fully connected layer at the corresponding level of compression. All error results are reported on the test set.

6.3综合评价

我们对所有数据集采用表1所示的神经网络架构。该架构是一个深层卷积神经网络，由五个卷积层（5×5个滤波器）和一个完全连接的层组成。在卷积之前，输入特征映射被零填充，使得输出映射保持与卷积之后的（未添加的）输入映射相同的大小。在具有过滤器大小2×2和步幅2的层2,4和5中的卷积之后执行最大池，将输入图尺寸减小一半。采用整流线性单元作为激活函数。网络的输出是标签上的softmax函数。

在这种架构中，卷积层拥有大多数参数（120万卷积层v.s. 40万在具有10个输出类的完全连接层中）。在训练期间，我们使用批次大小64和动量0.9的小批量梯度下降来优化参数。我们使用20％的训练集作为早期停止的验证集。对于FreshNets，我们使用频率敏感的压缩方案，增加高频分量之间的权重分配。对于所有基线，我们在相应的压缩级别将HashedNets [6]应用于完全连接的层。所有错误结果都在测试集上报告。

Table2(a) and (b) show the comprehensive evaluation of all methods under compression ratios 1/16 and 1/64, respectively. We exclude DropFilt and DropFreq in Table 2(b) because neither supports 1/64 compression in this architecture for all layers. For all methods, the fully connected layer (top layer) is compressed by HashedNets [6] at the corresponding compression rate. In this way, the ﬁnal size of the entire network respects the speciﬁed compression ratio. For reference, we also show the error rate of a standard convolutional neural network (CNN, columns 2 and 8) with the fully-connected layer compressed by HashedNets and no compression in the convolutional layers. Excluding this reference, we highlight the method with best test error on each dataset in bold.

We discern several general trends. In Table 2(a), we observe the performance of the DropFilt and DropFreq at 1/16 compression. At this compression rate, DropFilt corresponds to a network 1/16 ﬁlters at each layer: 2, 4, 4, 8, 16 at layers 1−5 respectively. This architecture yields particularly poor test accuracy, including essentially random predictions on three datasets. DropFreq, which at 1/16 compression parameterizes each ﬁlter in the original network by only 1 or 2 low-frequency values in the DCT frequency space, performs with similarly poor accuracy. Low rank decomposition (LRD) and HashedNets each yield similar performance at both 1/16 and 1/64 compression. Neither explicitly considers the smoothness inherent in learned convolutional ﬁlters, instead compressing the ﬁlters in the spatial domain. Our method, FreshNets, consistently outperforms all baselines, particularly at the higher compression rate as shown in Table 2(b).

表2（a）和（b）分别示出了在压缩比为1/16和1/64时所有方法的综合评价。我们在表2（b）中排除DropFilt和DropFreq，因为在这种架构中对所有层都不支持1/64压缩。对于所有方法，完全连接的层（顶层）由HashedNets [6]以相应的压缩率压缩。以这种方式，整个网络的最终大小遵守指定的压缩比。作为参考，我们还显示标准卷积神经网络（CNN，第2列和第8列）的错误率与完全连接的层压缩HashedNets和卷积层中没有压缩。除了这个参考，我们突出显示每个数据集上最好的测试错误的方法，以粗体显示。

我们看到几个总趋势。在表2（a）中，我们观察了在1/16压缩下DropFilt和DropFreq的性能。在该压缩率下，DropFilt对应于每层的网络1/16滤波器：分别为在层1-5处的2,4，4,8,16。这种架构产生特别差的测试精度，包括基本上对三个数据集的随机预测。 DropFreq，其在1/16压缩参数化在原始网络中的每个滤波器仅在DCT频率空间中的1或2个低频值，执行具有类似的低精度。低级分解（LRD）和HashedNets在1/16和1/64压缩都产生类似的性能。既没有明确考虑学习的卷积滤波器固有的平滑，而是压缩空间域中的滤波器。我们的方法FreshNets，始终优于所有基线，特别是在较高的压缩率，如表2（b）所示。

Using the same model in Table 1, Figure 5 shows more complete curves of test errors with multiple compression factors on the CIFAR10 and ROT datasets.

在表1中使用相同的模型，图5示出了在CIFAR10和ROT数据集上具有多个压缩因子的测试误差的更完整的曲线。

6.4 Varying compression by frequency

As mentioned in Section 3.4, we allow a higher collision rate in the high frequency components than in the low frequency components for each ﬁlter.

To demonstrate the utility of this scheme, we evaluate several hash compression schemes. Systematically, we set the compression rate of the jth frequencyb and rj with a parameterized function,i.e. rj = f(j). In this experiment, we use the beta distribution:

f(j;α,β) = Zxα−1(1−x)β−1,

where x = j+1 2k−1 is a real number between 0 and 1, k is the ﬁlter size, and Z is a normalizing factor such that the resulting distribution of parameters meets the target parameter budget K, i.e. when:

2k−2 X j=0 rjNj = K.

We adjust α and β to control the compression rate for each frequency region. As shown in Figure 6, we have multiple pairs of α and β, each of which results in a different compression scheme. For example, if α = 0.25 and β = 2.5, the compression rate monotonically decreases as a function of component frequency, meaning more parameter sharing among high frequency components (blue curve in Figure 6).

To quickly evaluate the performance of each scheme, we use a simple four-layer FreshNets where the ﬁrst two layers are DCT hashed convolutional layers (with 5×5 ﬁlters) containing 32 and 64 feature maps respectively, and the last two layers are fully connected layers.

We test FreshNets on CIFAR10 with each of the compression schemes shown in Figure 6. In each,weightsharingislimitedtobe within groups of similar frequencies, as described in Section 3.4, however number of unique weights shared within each group is varied. We denote the compression scheme with α,β = 1 (red curve) as a frequency-oblivious scheme since it produces a uniform compression independent of frequency.

In the inset bar plot in Figure 6, we report test error normalized by the test error of the frequency-oblivious scheme and averaged over compression rates 1, 1/2, 1/4, 1/16, 1/64, and 1/256. We can see that the proposed scheme with fewer shared weights allocated to high frequency components (represented by the blue curve) outperforms all other compression schemes. An inverse scheme where the high frequency regions have the lowest collision rate (purple curve) performs the worst. These empirical results ﬁt our assumption that the low frequency components of a ﬁlter are more important than the high frequency components.

6.4频率变化压缩

如3.4节所述，对于每个滤波器，在高频​​分量中的碰撞率要高于在低频分量中的碰撞率。

为了演示此方案的效用，我们评估几个哈希压缩方案。系统地，我们设置第j个频率b和rj的压缩率与参数化函数， rj = f（j）。在这个实验中，我们使用beta分布：

 f（j;α，β）=Zxα-1（1-x）β-1，

其中x = j + 1 2k-1是0和1之间的实数，k是滤波器大小，Z是归一化因子，使得所得到的参数分布满足目标参数预算K，即当：

 2k-2 X j = 0 rjN j = K

我们调整α和β以控制每个频率区域的压缩率。如图6所示，我们有多对α和β，每一对导致不同的压缩方案。例如，如果α= 0.25和β= 2.5，压缩率作为分量频率的函数单调减小，这意味着在高频分量（图6中的蓝色曲线）之间更多的参数共享。

为了快速评估每个方案的性能，我们使用一个简单的四层FreshNets，其中第一个两层是分别包含32和64个特征图的DCT散列卷积层（具有5×5个滤波器），并且最后两个层是完全连接的层。

我们使用CIFAR10上的FreshNets测试每种压缩方案，如图6所示。在每一种情况下，权重分配限制在类似频率的组内，如第3.4节所述，但是每组中共享的唯一权重的数量是变化的。我们将α，β= 1（红色曲线）的压缩方案表示为频率不变方案，因为其产生独立于频率的均匀压缩。

在图6的插入条形图中，我们报告了由频率忽略方案的测试误差归一化的测试误差，并且在压缩率1，1/2，1/4，1/16，1/64和1 / 256。我们可以看到，分配给高频分量（由bluecurve表示）的较少共享权重的所提出的方案优于所有其他压缩方案。其中高频区域具有最低冲突率（紫色曲线）的逆方案执行最差。这些经验结果证明了滤波器的低频分量比高频分量更重要的假设。

6.5 Filter visualization

We investigate the smoothness of the learned convolutional ﬁlters in Figure 7 by visualizing the ﬁlter weights (ﬁrst layer) of (a) a standard, uncompressed CNN, (b) FreshNets, and (c) HashedNets (with weight sharing in the spatial domain). For this experiment, we again apply a four layer network with two convolutional layers but adopt larger ﬁlters (11×11) for better visualization.

All three networks are trained on MNIST, and both FreshNets and HashedNets have 1/16 compression on the ﬁrst convolutional layer. When plotting, we scale the values in each ﬁlter matrix to the range [0,255]. Therefore, white and black pixels stand for large positive and negative weights, respectively. We observe that, although they are more blurry due to the compression, the ﬁlter weights of FreshNets are still smooth while weights in HashedNets appear more chaotic.

6.5过滤器可视化

我们通过可视化（a）一个标准的，未压缩的CNN，（b）FreshNets和（c）HashedNets（在空间域中的权重共享）的滤波器权重（第一层）来研究图7中学习的卷积滤波器的平滑度， 。 对于这个实验，我们再次应用具有两个卷积层的四层网络，但采用更大的滤波器（11×11）更好的可视化。

  所有三个网络在MNIST上训练，FreshNets和HashedNets在第一卷积层上具有1/16压缩。 当绘制时，我们将每个滤波器矩阵中的值缩放到范围[0,255]。 因此，白色和黑色像素分别代表大的正负权重。 我们观察到，虽然他们由于压缩更模糊，FreshNets的过滤器权重仍然平滑，而HashedNets的权重显得更混乱。

7. CONCLUSIONS

In this paper we present FreshNets, a method for learning convolutional neural networks with dramatically compressed model storage. We introduce negligible efﬁciency overhead (since most running time is still spent on convolutional operations) but obtain a dramatically smaller convolutional neural network. Harnessing the hashing trick for parameter-free random weight sharing and leveraging the smoothness inherent in convolutional ﬁlters, FreshNets compresses parameters in a frequency-sensitive fashion such that signiﬁcant model parameters (e.g. low-frequency components) are better preserved. As such , FreshNets preserves prediction accuracy better than competing baselines at high compression rates.

We believe that the proposed compression techniques will have broad impacts on real-world applications, such as better image and speech recognition on mobile phones and other devices. In the future, we will investigate more on the frequency structure of FreshNets and explore additional hashing schemes. Moreover, we will investigate its integration with other implementation schemes such as low precision representation of weights.

7.结论

在本文中，我们提出FreshNets，一种学习卷积神经网络与显着压缩模型存储的方法。我们引入可忽略的效率开销（因为大多数运行时间仍然用于卷积运算），但获得一个显着更小的卷积神经网络。利用散列技巧无参数随机权重共享和利用卷积滤波器固有的平滑性，FreshNets以频率敏感的方式压缩参数，使得重要的模型参数（例如低频分量）被更好地保留。因此，FreshNets在高压缩率下保留预测精度比竞争基线更好。

我们认为所提出的压缩技术将对真实世界的应用产生广泛的影响，如在手机和其他设备上更好的图像和语音识别。在未来，我们将更多地研究FreshNets的频率结构，并探讨额外的哈希方案。此外，我们将调查其与其他实现方案的集成，例如权重的低精度表示。