

来自今日头条

# Dynamic Routing Between Capsules

Sara Sabour

Nicholas Frosst

Geoffrey E. Hinton  
Google Brain  
Toronto

胶囊间的动态路由

## 摘要

本论文所研究的胶囊意为一组神经元，其激活向量反映了某类特定实体（可能是整体也可能是部分）的表征。本论文使用激活向量的模长来描述实体存在的概率，用激活向量的方向表征对应实例的参数。某一层级的活跃胶囊通过矩阵变换做出预测，预测结果会用来给更高层级的胶囊提供实例参数。当多个预测值达成一致时，一个高层级的胶囊就会被激活。论文中展示了差异化训练的多层胶囊系统可以在MNIST上达到当前最高水平的表现，在识别高度重叠的数字上也要比卷积网络要好得多。网络的实现中运用迭代的一致性路由机制：当低层级的胶囊的预测向量和高层级胶囊的激活向量有较大的标量积时，这个低层级胶囊就会倾向于向高层级胶囊输出。

## 一、简介

人类视觉通过使用仔细确定的固定点序列来忽略不相关的细节，以确保只有极小部分的光学阵列以最高的分辨率被处理。要理解我们对场景的多少知识来自固定序列，以及我们从单个固定点中能收集到多少知识，内省不是一个好的指导，但是在本文中，我们假设单个固定点给我们提供的不仅仅是一个单一的识别对象及其属性。我们假设多层视觉系统在每个固定点上都会创建一个类似解析树这样的东西，并且单一固定解析树在多个固定点中如何协调的问题会被我们忽略掉。

解析树通常通过动态分配内存来快速构建，但根据Hinton等人的论文「Learning to parse images, 2000」，我们假设，对于单个固定点，从固定的多层神经网络中构建出一个解析树，就像从一块岩石雕刻出一个雕塑一样（雷锋网 AI 科技评论注：意为只保留了部分树枝）。每个层被分成许多神经元组，这些组被称为“胶囊”（Hinton等人「Transforming auto-encoders, 2011」），解析树中的每个节点就对应着一个活动的胶囊。通过一个迭代路由过程，每个活动胶囊将在更高的层中选择一个胶囊作为其在树中的父结点。对于更高层次的视觉系统，这样的迭代过程就很有潜力解决一个物体的部分如何层层组合成整体的问题。

一个活动的胶囊内的神经元活动表示了图像中出现的特定实体的各种属性。这些属性可以包括许多不同类型的实例化参数，例如姿态（位置，大小，方向），变形，速度，反照率，色相，纹理等。一个非常特殊的属性是图像中某个类别的实例的存在。表示存在的一个简明的方法是使用一个单独的逻辑回归单元，它的输出数值大小就是实体存在的概率（雷锋网 AI 科技评论注：输出范围在0到1之间，0就是没出现，1就是出现了）。在本文中，作者们探索了一个有趣的替代方法，用实例的参数向量的模长来表示实体存在的概率，同时要求网络用向量的方向表示实体的属性。为了确保胶囊的向量输出的模长不超过1，通过应用一个非线性的方式使矢量的方向保持不变，同时缩小其模长。

胶囊的输出是一个向量，这一设定使得用强大的动态路由机制来确保胶囊的输出被发送到上述层中的适当的父节点成为可能。最初，输出经过耦合总和为1的系数缩小后，路由到所有可能的父节点。对于每个可能的父节点，胶囊通过将其自身的输出乘以权重矩阵来计算“预测向量”。如果这一预测向量和一个可能的父节点的输出的标量积很大，则存在自上而下的反馈，其具有加大该父节点的耦合系数并减小其他父节点耦合系数的效果。这就加大了胶囊对那一个父节点的贡献，并进一步增加了胶囊预测向量和该父节点输出的标量积。这种类型的“按协议路由”应该比通过最大池化实现的非常原始的路由形式更有效，其中除了保留本地池中最活跃的特征检测器外，忽略了下一层中所有的特征检测器。作者们论证了，对于实现分割高度重叠对象所需的“解释”，动态路由机制是一个有效的方式。

卷积神经网络（CNN）使用学习得到的特征检测器的转移副本，这使得他们能够将图片中一个位置获得的有关好的权重值的知识，迁移到其他位置。这对图像解释的极大帮助已经得到证明。尽管作者们此次用矢量输出胶囊和按协议路由的最大池化替代CNN的标量输出特征检测器，他们仍然希望能够在整个空间中复制已习得的知识，所以文中构建的模型除了最后一层胶囊之外，其余的胶囊层都是卷积。与CNN一样，更高级别的胶囊得以覆盖较大的图像区域，但与最大池化不同，胶囊中不会丢弃该区域内实体精确位置的信息。对于低层级的胶囊，位置信息通过活跃的胶囊来进行“地点编码”。当来到越高的层级，越多的位置信息在胶囊输出向量的实值分量中被“速率编码”。这种从位置编码到速率编码的转变，加上高级别胶囊能够用更多自由度、表征更复杂实体的特性，表明更高层级的胶囊也相应地需要更高的维度。

## 二、如何计算一个胶囊的向量输入和输出

已经有很多方法可以实现胶囊的大致思路。这篇文章的目的，不是去探究所有可能的方法，而只是表明非常简单直接的方式就可以取得很好的效果，而且动态路由也可以起到帮助。

作者们用胶囊输出向量的模长来表示一个胶囊所表征的实体在输入中出现的概率。因此作者们采用一个非线性函数对向量进行“压缩”，短向量被压缩到几乎为零，长向量也被压缩到1以下长度。判别学习中充分利用这个非线性函数。

$$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1 + \|\mathbf{s}_j\|^2} \frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}$$

（式1）

其中 $\mathbf{v}_j$ 是胶囊 $j$ 的输出向量， $\mathbf{s}_j$ 是它的全部输入。

除了第一层胶囊，胶囊 $\mathbf{s}_j$ 的全部输入是对预测向量 $\mathbf{u}_{ji}$ 的加权求和。这些预测向量都是由低一层的胶囊产生，通过胶囊的输出 $\mathbf{u}_i$  和一个权重矩阵 $\mathbf{W}_{ij}$ 相乘得来。

（式2）

其中 $c_{ij}$ 是由迭代的动态路径过程决定的耦合系数。

胶囊 $i$ 和其上一层中所有胶囊的耦合系数的和为1，并由“routing softmax” 决定。这个“routing softmax” 的初始逻辑值 $b_{ij}$  是胶囊 $i$ 耦合于胶囊 $j$ 的对数先验概率。

(式3)

这个对数先验可以和其他权重一起被判别学习。他们由两个胶囊的位置和类型决定，而不是当前的输入图像决定。耦合系数会从初始值开始迭代，通过测量每个高一层胶囊 $j$ 的当前输出 $v_i$ 和低一层胶囊 $i$ 的预测值 $u_{ij}$ 之间的一致性。

所述一致性是简单的点积 $a_{ij} = v_j \cdot u_{ij}$ 。这个一致性可被看做最大似然值，并在计算出所有将胶囊 $i$ 连接到更高层胶囊得到的新耦合值前，加到初始逻辑值 $b_{ij}$ 上。

在卷积胶囊层中，胶囊内每一个单元都是一个卷积单元。因此每一个胶囊都会输出一个向量网格而不是一个简单的向量。

路由计算的伪码如下图

#### Procedure 1 Routing algorithm.

```

1: procedure ROUTING( $\hat{u}_{j|i}$ ,  $r$ ,  $l$ )
2:   for all capsule  $i$  in layer  $l$  and capsule  $j$  in layer  $(l+1)$ :  $b_{ij} \leftarrow 0$ .
3:   for  $r$  iterations do
4:     for all capsule  $i$  in layer  $l$ :  $c_i \leftarrow \text{softmax}(b_i)$  ▷ softmax computes Eq. 3
5:     for all capsule  $j$  in layer  $(l+1)$ :  $s_j \leftarrow \sum_i c_{ij} \hat{u}_{j|i}$ 
6:     for all capsule  $j$  in layer  $(l+1)$ :  $v_j \leftarrow \text{squash}(s_j)$  ▷ squash computes Eq. 1
7:     for all capsule  $i$  in layer  $l$  and capsule  $j$  in layer  $(l+1)$ :  $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j$ 
   return  $v_j$ 

```

### 三、某类数字是否存在的边缘损失

作者们用实例化向量的模长来表示胶囊要表征的实体是否存在。所以当且仅当图片里出现属于类别 $k$ 的数字时，作者们希望类别 $k$ 的最高层胶囊的实例化向量模长很大。为了允许一张图里有多多个数字，作者们对每一个表征数字 $k$ 的胶囊分别给出单独的边缘损失函数(margin loss):

(式4)

其中 $T_c=1$ 当且仅当图片中有属于类别 $C$ 的数字， $m_+=0.9$ ， $m_-=0.1$ 。是为了减小某类的数字没有出现时的损失，防止刚开始学习就把所有数字胶囊的激活向量模长都压缩了。作者们推荐选用 $\lambda = 0.5$ 。总损失就是简单地把每个数字胶囊的损失加起来的总和。

### 四、CapsNet 结构

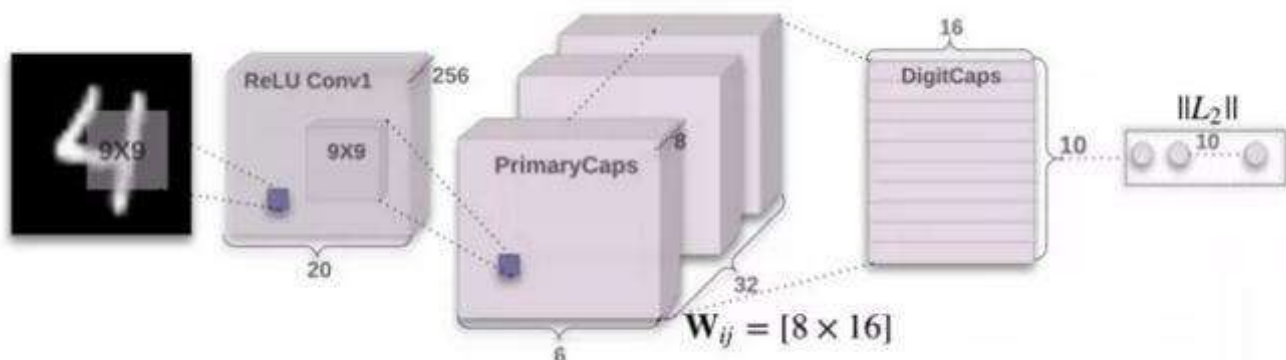
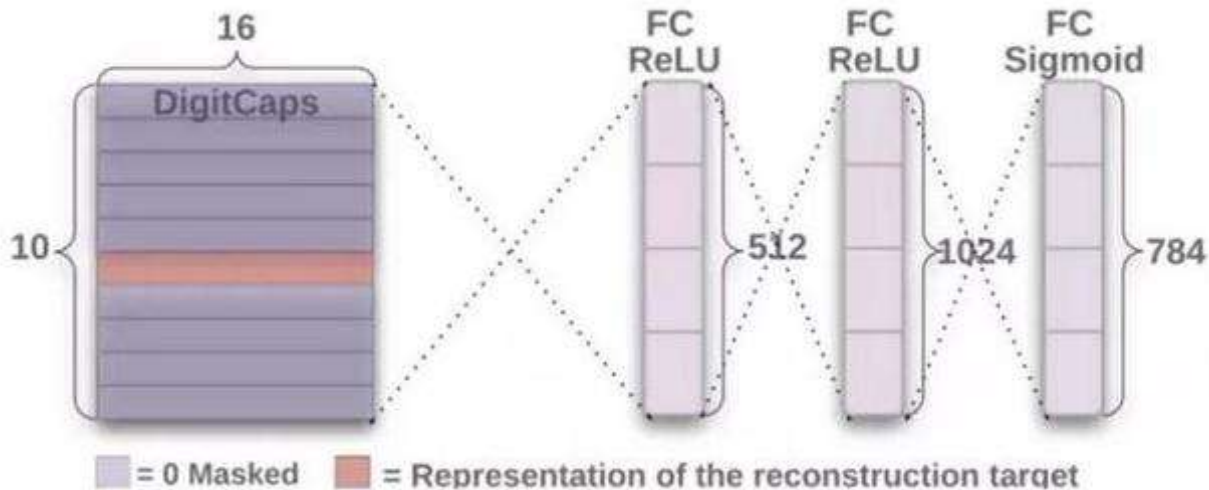


图1：一个简单的3层CapsNet。这个模型的结果能和深层卷积网络（比如. Batch-normalized maxout network in network, 2015）的结果媲美。DigitCaps层每个胶囊的激活向量模长给出

了每个类的实例是否存在，并且用来计算分类损失。是PrimaryCapsules中连接每个  $u_i, i \in (1, 32 \times 6 \times 6)$  和每个  $v_j, j \in (1, 10)$  的权重矩阵。



**图2：从DigitCaps层来重构数字的解码结构。训练过程中极小化图像和Sigmoid层的输出之间的欧氏距离。训练中作者们用真实的标签作为重构的目标。**

图1展示的是一个简单的CapsNet结构。这是一个很浅的网络，只有2个卷积层和1个全连接层。Conv1有256个 $9 \times 9$ 的卷积核，步长取1，激活函数为ReLU。这层把像素亮度转化成局部特征检测器的激活，接下去这个值会被用来作为原始胶囊(primary capsules)的输入。

原始胶囊是多维实体的最底层。这个过程和图形生成的视角相反，激活了一个原始胶囊就和刚好是图形渲染的逆过程。与先分别计算实例的不同部分再拼在一起形成熟悉的总体理解（图像中的每个区域都会首先激活整个网络而后再进行组合）不同，这是一种非常不同的计算方式。而胶囊的设计就很适合这样的计算。

第二层PrimaryCapsules是一个卷积胶囊层，有32个通道，每个通道有一个8维卷积胶囊（也就是说原始胶囊有8个卷积单元， $9 \times 9$ 的卷积核，步长为2）。这一层中的胶囊能看到感受野和这个胶囊的中心重合的所有 $256 \times 81$  Conv1单元的输出。PrimaryCapsules一共有 $[32, 6, 6]$ 个输出（每个输出是一个8维向量）， $[6, 6]$ 网格中的每个胶囊彼此共享权重。由于具有区块非线性，可以把PrimaryCapsules视作一个符合式1的卷积层。最后一层（DigitCaps）有对每个数字类有一个16维的胶囊，所有低一层的胶囊都可以是这一层胶囊的输入。

作者们只在两个连续的胶囊层（比如PrimaryCapsules和DigitCaps）之间做路由。因为Conv1的输出是1维的，它所在的空间中不存在方向可以和高层的向量方向达成一致性。所以在Conv1和PrimaryCapsules之间没有路由。所有的路由逻辑值( $b_{ij}$ )被初始化为0。因此，一开始一个胶囊的输出( $u_i$ )会以相同的概率( $c_{ij}$ )传入到所有的母胶囊( $v_0, v_1, \dots, v_{10}$ )。作者们用TensorFlow实现了这个网络，选择了Adam优化器和TensorFlow的默认参数，包括指数衰减的学习率用来优化式4的边缘损失的总和。

#### 4.1 为了正则化效果而做的重构工作

作者们使用了一个额外的重构损失，希望数字胶囊能对输入数字的实例化参数做编码。在训练过程中，作者们用掩蔽的方法只把正确的数字胶囊的激活向量保留下来。然后用这个激活向量来做重构。数字胶囊的输出会传入一个由3个全连接层组成的解码器，它的结构如图2，用来建模像素密度。



作者们极小化回归单元的输出和原来图片的像素亮度之间的平方误差，并把重构误差收缩到原来的0.0005倍，这样才不会在训练过程中盖过边缘误差的作用。如图3所示，CapsNet的16维输出的重构是鲁棒的，同时也只保留了重要的细节。

五、把 Capsule 用在MNIST上

使用 28×28 MNIST的图片集进行训练，训练前这些图片在每个方向不留白地平移了2个像素。除此之外，没有进行其他的数据增改或者转换。在MNIST数据库中，6万张图片用于训练，另外1万张用于测试。

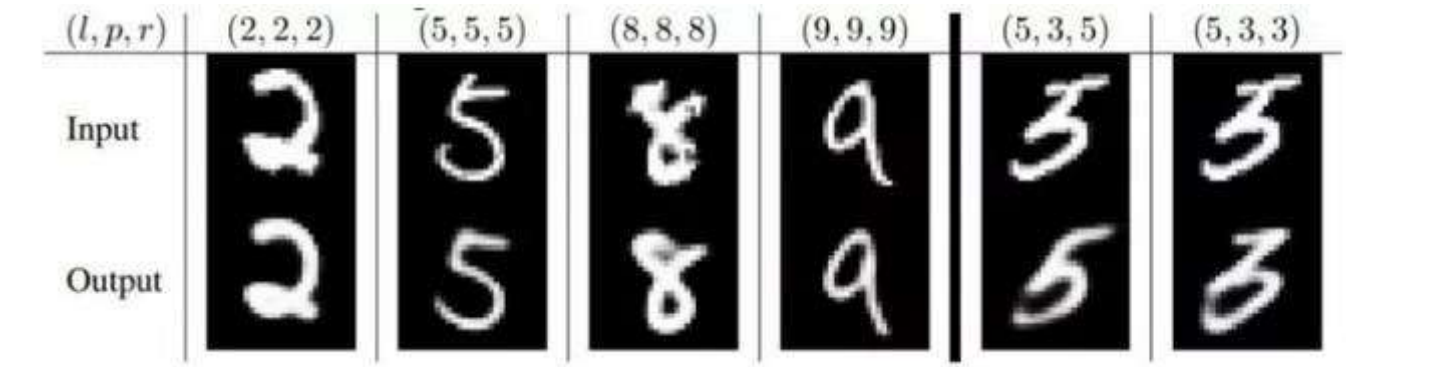


图3：利用3次路由迭代学习的CapsNet对MNIST中的测试照片进行重构。(l, p, r)分别代表真实标签、模型预测和重建结果。最右两列展示的是重建失败的例子，解释了模型是如何混淆了图片中的“5”和“3”。其他列属于被正确分类了的，展示了模型可以识别图像中的细节，同时降低噪声。

Method	Routing	Reconstruction	MNIST (%)	MultiMNIST (%)
Baseline	-	-	0.39	8
CapsNet	1	no	0.34±0.032	-
CapsNet	1	yes	0.29±0.011	7
CapsNet	3	no	0.35±0.036	-
CapsNet	3	yes	0.25±0.005	5


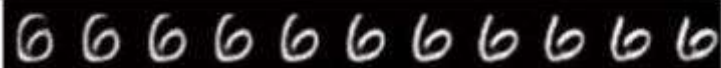


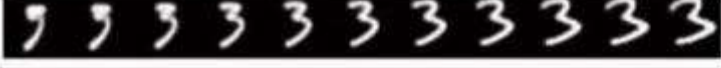
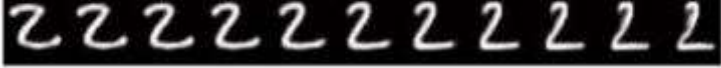
表1：CapsNet 分类MNIST数字测试准确度。结果包含了三次测试得到的平均数和标准差。

测试中作者使用的是单一模型，没有进行“综合”或者明显的数据扩增方法。（Wan等人在「Regularization of neural networks using dropconnect」中通过“综合”及数据扩增实现了0.21%的错误率，而未使用这两种方法时的错误率是0.57%）作者们通过3层神经网络实现了较低的错误率（0.25%），这一错误率以往只有更深的网络才能达到。表1展现的是不同设置的CasNet在NMIST数据库上的测试错误率，表明了路由以及正则器重构的重要性。其基线是一个标准的三层神经网络（CNN），分别具有256、256及128个通道。每个通道具有5×5的卷积核，卷积步长为1。接着有两个全连接层，大小分别为328、192。最后的全连接层通过dropout连接到带有交叉熵损失的10个分类输出的softmax层。

5.1 capsule的单个维度表示什么

由于模型中只向DigitCaps层的胶囊传递一个数字的编码并置零其他数字，所以这些胶囊应该学会了在这个类别已经具有一个实例的基础上拓展了变化空间。这些变化包括笔画粗细、倾斜和宽度。还

包括不同数字中特定的变化，如数字2尾部的长度。通过使用解码器网络可以看到单个维度表示什么。在计算正确的数字胶囊的激活向量之后，可以将这个激活向量的扰动反馈给解码器网络，并观察扰动如何影响重建。这些扰动的例子如图4所示。可以看到，胶囊的一个维度（总数为16）几乎总是代表数字的宽度。有些维度表示了全局变化的组合，而有些维度表示数字的局部变化。例如，字母6上部分的长度和下部分圈的大小使用了不同的维度。

Scale and thickness	
Localized part	
Stroke thickness	
Localized skew	
Width and translation	
Localized part	

**图4：维度扰动。** 每一行表示DigitCaps16个维度表示中的一个维度在 $[-0.25, 0.25]$ 范围，步长0.05时的重构结果

## 5.2 仿射变换的鲁棒性

实验表明，每个DigitCaps层的胶囊都比传统卷积网络学到了每个类的更鲁棒的表示。由于手写数字的倾斜、旋转、风格等方面存在自然差异，训练好的CapsNet对训练数据小范围的仿射变换具有一定的鲁棒性。

为了测试CapsNet对仿真变换的鲁棒性，作者们首先基于MNIST训练集创造了一个新的训练集，其中每个样本都是随机放在 $40 \times 40$ 像素的黑色背景上的MNIST数字。然后用这样的训练集训练了一个CapsNet和一个传统的卷积网络（包含MaxPooling和DropOut）。

然后，作者们在affNIST数据集上测试了这个网络，其中，每个样本都是一个具有随机小范围仿射变换的MNIST数字。模型并没有在任何放射变换，甚至标准MNIST自然变换的训练集合上训练过，但一个训练好的带有早期停止机制（early stop）的CapsNet，在拓展的MNIST测试集上实现了99.23%的准确度，在仿射测试集上实现了79%的准确性。具有类似参数数量的传统卷积模型在扩展的MNIST测试集上实现了类似的准确度（99.22%），在仿射测试集上却只达到了66%。

## 六、高度重叠数字的分割

动态路由可以视为平行的注意力机制，允许同层级的胶囊参与处理低层级的活动胶囊，并忽略其他胶囊。理论上允许模型识别图像中的多个对象，即使对象重叠。Hinton等人的目的是分割并识别高度重合数字对象（「Learning to parse images, 2000」中提出，其它人也在类似的领域实验过他们的网络，Goodfellow等人在「Multi-digit number recognition from street view imagery using deep convolutional neural networks, 2013」中，Ba等人在「Multiple object recognition with visual attention, 2014」中，Greff等人在「Tagger: Deep unsupervised perceptual grouping, 2016」中）。一致性路由使利用对象的形状的先验知识帮助进行分割成为了可能，并避免在像素领域进行更高级别的细分。

### 6.1 MultiMNIST数据集

作者们通过在数字上覆盖另一个来自相同集合（训练或测试）但不同类别的数字来生成MultiMNIST训练测试数据集。每个数字在每个方向上最多移动4个像素，产生 $36 \times 36$ 像素的图像。考虑到 $28 \times 28$ 像素图像中的数字是以 $20 \times 20$ 像素的范围作为边框，两个数字的边框内范围平均有80%的重合部分。MNIST数据集中的每个数字都会生成1K MultiMNIST示例。训练集的大小为60M，测试集的大小为10M。

## 6.2 MultiMNIST数据集上的结果

作者用MultiMNIST的训练数据中重新训练得到的3层CapsNet模型，比基线卷积模型获得了更高的分类测试准确率。相较于Ba等人在「Multiple object recognition with visual attention, 2014」的序列注意力模型，他们执行的是更简单的、数字交叠远远更小的任务（本文的测试数据中，两个数字的外框交叠率达到80%，而Ba等人的只有4%），而本文的模型在高度交叠的数字对中获得了与他们同样的5%的错误率。测试图片由测试集中的成对的图片构成。作者们把两个最活跃的数字胶囊看作胶囊网络产生的分类结果。在重建过程中，作者们每次选择一个数字，用它对应的数字胶囊的激活向量来重建这个数字的图像（已经知道这个图像是什么，因为作者们预先用它来生成合成的图像）。与上文MNIST测试中模型的唯一不同在于，现在把将学习率的衰减步数提高到了原来的10倍，这是因为训练数据集更大。

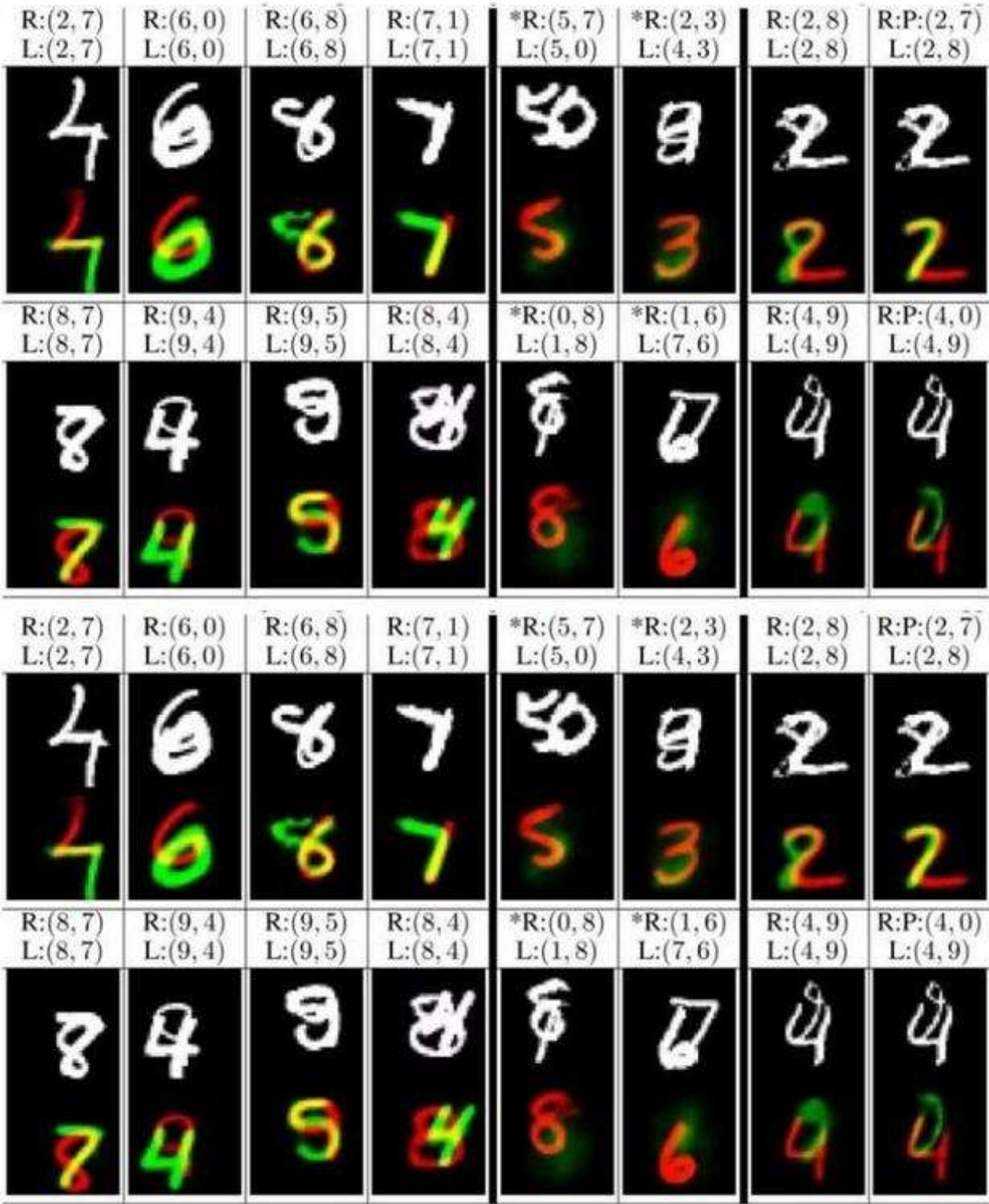


图5：一个经3次路由迭代的CapsNet在MultiMNIST测试数据集上的样本重建结果

如图中靠下的图像所示，两个重建出的互相交叠的数字分别显示为绿色和红色的。靠上的图显示的是输入的图像。表示图像中两个数字的标签；表示用于重建的两个数字。最右边的两列显示了从标签和从预测重建的两个错误分类样例。在例子中，模型将8错判成7；在的例子中，模型将9错判成0。其他的列都分类正确并且显示了模型不仅仅考虑了所有的像素同时能够在非常困难的场景下将一个像素分配给两个数字（1-4列）。值得说明的是，在数据集产生的过程中，像素的值都会被剪裁到1以内。两个含“\*”的列显示了重建的数字既不是标签值也不是预测值。这些列显示模型不仅仅找到了所有存在的数字的最佳匹配，甚至还考虑了图像中不存在的数字。所以在的例子中，模型并不能重建数字7，是因为模型知道数字对5和0是最佳匹配，而且也已经用到了所有的像素。的例子也是类似的，数字8的环并没有触发为0的判断，因为该数字已经被当做8了。因此，如果两个数字都没有其他额外的支持的话，模型并不会将一个像素分配给这两个数字。



图5中的重构表明，CapsNet 能够把图片分割成两个原来的数字。因为这一分割并非是直接的像素分割，所以可以观察到，模型可以准确处理重叠的部分(即一个像素同时出现在多个数字上)，同时也利用到所有像素。每个数字的位置和风格在DigitCaps中都得到了编码。给定一个被编码数字，解码器也学会了去重构这一数字。解码器能够无视重叠进行重构的特性表明，每个数字胶囊都能从PrimaryCapsules层接收到的不同激活向量来获取位置和风格。

表1 也着重表现了这一任务中胶囊之间路由的重要性。作为CapsNet分类器准确率的对比基线，作者们一开始先训练了带有两层卷积层和两层全连接层的卷积神经网络。第一层有512个大小为 $9 \times 9$ 的卷积核，步长为1；第二层有256个大小为 $5 \times 5$ 的卷积核，步长为1。在每个卷积层后，模型都连接了一个 $2 \times 2$ 大小，步长2的池化层。第三层是一个1024维的全连接层。

所有的这三层都有ReLU非线性处理。最后10个单元的层也是全连接。我们用TF默认的Adam优化器来训练最后输出层的Sigmoid交叉熵损失。这一模型有24.56M参数，是CapsNet的11.36M参数的两倍多。作者们从一个小点的CNN(32和64个大小为 $5 \times 5$ 的卷积核，步幅为1，以及一个512维的全连接层)开始，然后逐渐增大网络的宽度，直到他们在MultiMNIST的10K子集上达到最好的测试精度。他们也在10K的验证集上搜索了正确的学习率衰减步数。

作者们一次解码了两个最活跃的DigitCaps胶囊，得到了两张图片。然后把所有非零的像素分配给不同的数字，就得到了每个数字的分割结果。

## 七、其它数据集

作者们在 CIFAR10 的数据及上测试了胶囊模型，在用了不同的超参和7个模型集成（其中每个模型都通过图像中 $24 \times 24$ 的小块进行三次路由迭代）后得到10.6%的错误率。这里的图片都是三个颜色通道的，作者们一共用了64种不同的 primary capsule，除此之外每个模型都和在 MNIST 数据集中用的一模一样。作者们还发现胶囊能够帮助路由softmax增加一个“以上皆非”的分类种类，因为不能指望10个 capsules 的最后一层就能够解释图片里的一切信息。在测试集上有 10.6% 的错误率差不多也是标准的卷积网络初次应用到 CIFAR10 上能达到的效果。

和生成模型一个一样的缺点是，Capsules 倾向于解释图片中的一切。所以当能够对杂乱的背景建模时，它比在动态路由中只用一个额外的类别来的效果好。在 CIFAR-10 中，背景对大小固定的模型来说变化太大，因此模型表现也不好。

作者们还用了和 MNIST 中一样的模型测试了 smallNORB 数据集，可以得到目前最好的 2.7% 的错误率。smallNORB 数据集由  $96 \times 96$  的双通道灰度图组成。作者们把图片缩放到  $48 \times 48$  像素，并且在训练时从中随机裁剪  $32 \times 32$  的大小。而在测试时，直接取中间  $32 \times 32$  的部分。

作者们还在 SVHN 的 73257 张图片的小训练集上训练了一个小型网络。我们把第一个卷积层的通道数减少到 64个，primary capsule 层为 16 个 6维胶囊，最后一个胶囊层为8维的。最后测试集错误率为 4.3%。

## 八、讨论以及以往工作

30年来，语音识别的最新进展使用了以高斯混合作为输出分布的隐马尔可夫模型。这些模型虽然易于在一些计算机上学习，但是存在一个致命的缺陷：他们使用的“n种中的某一种”的表示方法的效率是呈指数下降的，分布式递归神经网络的效率就比这种方法高得多。为了使隐马尔可夫模型能够记住的迄今它所生成字符的信息倍增，需要使用的隐藏节点数目需要增加到原来的平方。而对于循环神经网络来说，只需要两倍的隐藏神经元的数量即可。

现在卷积神经网络已经成为物体识别的主流方法，理所当然要问是其中是否也会有效率指数下降，从而引发这种方法的式微。一个可能性是卷积网络在新类别上泛化能力的困难度。卷积网络中处理平移变换的能力是内置的，但对于仿射变换的其他维度就必须进行选择，要么在网格中复制特

征检测器，网格的大小随着维度数目指数增长，要么同样以指数方式增加的标注训练集的大小。胶囊通过将像素强度转换为识别到的片段中的实例化参数向量，然后将变换矩阵应用于片段，以预测更大的片段的实例化参数，从而避免了效率的指数下降。学到了部分和整体之间固有的空间关系的转换矩阵构成了具有视角不变性的知识，从而可以自动泛化到的视角中。

胶囊使得我们可以做出一个非常具有表征意义的假设：在图像的每一个位置，至多只有一个胶囊所表征的实体的实例。这种假设是由一种称为“crowding” (Pelli等人「Crowding is unlike ordinary masking: Distinguishing feature integration from detection, 2004」) 的感知现象驱动的，它消除了绑定问题，并允许一个胶囊使用分布式表示(它的激活向量)来对给定位置的该类型实体的实例化参数进行编码。这种分布式表示比通过在高维网格上激活一个点来编码实例化参数的效率要高得多，并且通过正确的分布式表示，胶囊可以充分利用空间关系可以由矩阵乘法来建模的特点。

胶囊中采用的神经活动会随着视角的变化而变化，而不是试图消除神经活动中视角变化带来的影响。这使它们比“归一化”法(如Jaderberg等「Spatial transformer networks, 2015」)更具有优势:它们可以同时处理多个不同仿射变换或不同对象的不同部件。

胶囊同时也非常擅长处理图像分割这样的另一种视觉上最困难的问题之一，因为实例化参数的矢量允许它们使用在本文中演示的那样的一致性路由。对胶囊的研究目前正处于一个与本世纪初研究用于语音识别的递归神经网络类似的阶段。根据基础表征性的特点，已经有理由相信这是一种更好的方法，但它可能需要一些更多的在细节上的洞察力才能把它变成一种可以投入应用的高度发达的技术。一个简单的胶囊系统已经在分割数字图像上提供了无与伦比的表现，这表明了胶囊是一个值得探索的方向。

(完)