Advances in Applied Mathematics 应用数学进展, 2022, 11(4), 1728-1739
Published Online April 2022 in Hans. http://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2022.114189

Hans汉斯

改进胶囊网络在图像识别中的应用

巩瑞鑫，贺　衍

太原理工大学数学学院，山西　晋中

收稿日期：2022年3月14日；录用日期：2022年4月8日；发布日期：2022年4月19日

摘　要

图像识别是指利用计算机对图像进行处理、分析和理解，以识别各种不同模式的目标和对象的技术，并对质量不佳的图像进行一系列的增强与重建技术手段，从而有效改善图像质量。本文用改进胶囊网络对MNIST数据集进行训练。胶囊是一组神经元，其活动向量表示一种特定类型的实体的实例化参数，它的长度代表实体存在的概率，方向代表实体的实例化参数，低层的活性胶囊，依据转移矩阵对高层胶囊的实例化参数进行预测，当多个预测一致时，高层胶囊被激活。本文利用spread损失来代替margin损失，避免过早出现“失活”胶囊，并且在不添加重构子网络的情况下，对不同路由迭代次数进行研究，确定路由迭代次数对分类准确率的影响，并确定模型最优参数。研究表明该模型在未做增强和扩展处理的MNIST数据集上的误分率低至0.32%。同时，改进胶囊网络在Fashion-MNIST，CIFAR-10数据集上也表现出了良好的性能。

关键词

图像识别，卷积神经网络，胶囊网络，手写数字识别

Application of Improved Capsule Network in Image Recognition

Ruixin Gong, Kan He

School of Mathematics, Taiyuan University of Technology, Jinzhong Shanxi

Received: Mar. 14th, 2022; accepted: Apr. 8th, 2022; published: Apr. 19th, 2022

Abstract

Image recognition refers to the technology of using computer to process, analyze and understand images in order to identify targets and objects in different modes. And carry out a series of enhancement and reconstruction technical means for the poor quality image, so as to effectively im-

文章引用: 巩瑞鑫, 贺衍. 改进胶囊网络在图像识别中的应用[J]. 应用数学进展, 2022, 11(4): 1728-1739.
DOI: 10.12677/aam.2022.114189

prove the image quality. In this paper, the improved capsule network is used to train MNIST data set. Capsule is a group of neurons, and its activity vector represents the instantiation parameters of a specific type of entity. Its length represents the probability of entity existence, and its direction represents the instantiation parameters of entity. For low-level active capsule, the instantiation parameters of high-level capsule are predicted according to the transfer matrix; when multiple predictions are consistent, the high-level capsule is activated. In this paper, spread loss is used to replace margin loss to avoid premature "inactivation" capsule, without adding reconstruction sub network, different routing iteration times are studied to determine the impact of routing iteration times on classification accuracy and determine the optimal parameters of the model. The research shows that the misclassification rate of the model on MNIST data set without enhancement and expansion is as low as 0.32%. At the same time, the improved capsule network also shows good performance on Fashion-MNIST and CIFAR-10 data sets.

Keywords

Image Recognition, Convolutional Neural Network, Capsule Network, Handwritten Numeral Recognition

1. 引言

图像识别是人工智能的一个重要领域，一直都受到人们的高度重视，它的发展经历了三个阶段，其中数字图像处理为图像识别技术的发展提供了强大的动力。因为 MNIST 数据集只有 10 类，因此成为了相对简单的手写识别任务。神经网络在 MNIST 数据集上进行训练，并验证模型预测的准确率，然后将该模型推广到其他图像识别任务中，改进图像识别方法，以达到更好的识别效果。图像识别的流程图如图 1 所示：

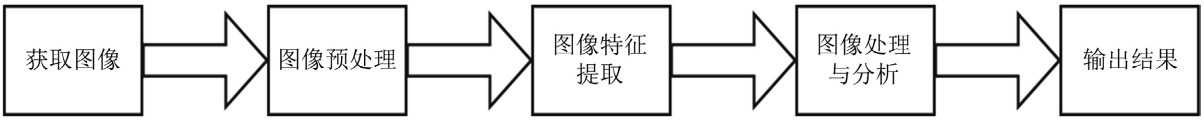

Figure 1. Flow chart of image recognition
图 1. 图像识别流程图

图像识别是通过分类并提取重要特征而排除多余的信息来识别图像。随着深度学习的发展，计算机视觉技术也迈出了很大的一步，卷积神经网络经常被用来做图像识别，其中逻辑回归与 *softmax* 回归在图像识别中被广泛应用。

逻辑回归[1] (*logistic regression*)用“*one-vs-all*”方法构建手写数字识别器，将 MNIST 输入的特征向量转化为一个标量，通过线性函数来实现分类，将线性函数的输出作为 *sigmoid* 函数的输入，得到一个概率值，训练 10 个不同的分类器，分别对应十个数字，以达到预测的目的，但逻辑回归算法不能保证输出的 10 个概率之和为 1。*Softmax* 回归算法很好地解决了这个问题。*Softmax* 回归算法将输入的特征向量转化的 10 个特征值作为 *softmax* 函数的输入，输出 10 个[0, 1]区间内的数值，且输出之和为 1。

卷积神经网络[2] [3] [4] (*Convolutional Neural Network, CNN*)利用卷积层对图像进行特征提取，可以直接用原始图像训练模型，采用不同的核函数提取出图像不同方面的特征。卷积神经网络结构如图 2 所示：

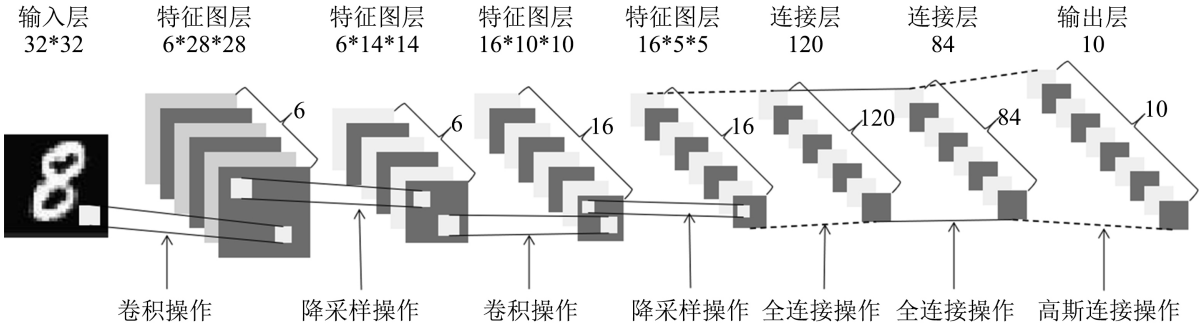

Figure 2. Convolutional neural network
图 2. 卷积神经网络

卷积运算会增加样本数据点的数量，从而大大增加了模型的运算量，因此用最大池化进行降采样，但同时会损失输入的信息。卷积神经网络对物体之间的空间关系识别能力不强，并且对物体旋转之后的图像识别能力不强，容易受到白盒攻击。

随着胶囊网络的出现，该区域内实体的位置信息不会丢失，经过旋转、平移、放缩后的手写体数字都可以有效识别，这是因为胶囊网络有两个优点：基于层的压缩以及动态路由。胶囊网络的胶囊层之间使用动态路由来代替卷积神经网络的池化层，用向量输出代替标量输出，不会丢失实体的位置信息，而且减少了计算量，提升了模型的准确率。但是该模型的损失函数固定上下限，容易过早出现“失活”胶囊。胶囊网络在动态路由和重构的共同作用下表现出了良好的性能，究竟是重构的作用还是仅仅是动态路由的作用是需要解决的一个问题。

本文的主要贡献如下：

1) 在动态路由过程中，使用 *spread* 损失来代替 *margin* 损失，使用线性增加的方式改变阈值，避免过早出现“失活”胶囊；

2) 在不加重构的情况下，对不同路由迭代次数进行研究，确定路由迭代次数对分类准确率的影响，并确定模型最优参数；

3) 研究表明，改进胶囊网络使得图像识别效果显著提升，同时改进胶囊网络在 Fashion-MNIST，CIFAR-10 数据集上也表现出了良好的性能。

2. 相关工作

2012 年，AlexNet [5]引入了 *ReLU* 函数，解决了权值消失的问题；引入了 *dropout* 的概念，来防止过拟合的问题；使用了数据增强来处理仿射变换引起的误分类问题。2014 年，VGGNet [6]使用了更小的卷积核和更深的网络；在不影响输入输出维度的情况下，引入非线性变换，降低计算量；采用了 *Multi-Scale* 的方法来训练和预测，可以增加训练的数据量，防止模型过拟合，提升预测准确率。同年，GoogLeNet [7] [8]使用 *inception* 模块结构提升训练结果，一是使用 1 × 1 的卷积来进行升降维，二是在多个尺寸上同时进行卷积再聚合，使得网络在每个层学到更好的特征表示。

这些网络层数过于多且复杂，导致计算量很大。网络的深度对模型的性能至关重要，当网络层数增加后，网络可以进行更加复杂的特征模式的提取，但是随着网络层数不断增加，可能会出现精度下降问

题和梯度消失或梯度爆炸问题。

2015 年，ResNet [9]通过残差学习解决了深度网络的退化问题，其内部的残差块使用了跳跃连接，缓解了在深度神经网络中增加深度带来的梯度消失问题，可以训练出更深的网络。2017 年，DenseNet [10]建立了不同层之间的连接关系，充分利用特征减轻梯度消失的问题；利用 *bottleneck layer*, *Translation layer* 以及较小的 *growth rate* 使得网络变得更窄，从而减少参数，有效抑制过拟合且减少计算量。

这些网络虽然一定程度上解决了梯度消失问题，但本质上依然是卷积神经网络，因此这些网络对物体之间的空间关系识别能力不强，并且对物体旋转之后的图像识别能力不强。

2017 年，Hinton 等人提出了胶囊网络[11] (*Capsule Network*, *CapsNet*)，用胶囊层之间的动态路由来代替卷积神经网络的池化层，用向量输出来代替标量输出，不会丢失关于实体在该区域内精确的位置信息，提升识别的准确率。胶囊网络不仅在 MNIST 数据集上取得了良好的分类准确率，并且对高度重叠的数字有很高的分割识别能力。为了确定重构是否在该网络中发挥了至关重要的作用，本文将在不加重构子网络的情况下，研究其分类准确率。

3. 胶囊网络

胶囊是由胶囊网络结构中的一组包含全部相关特征重要信息的向量神经元组成的，用向量的长度来表示实体存在的概率，用它的方向来表示该实体的实例化参数。胶囊输出的概率总和并不等于 1，也就是说胶囊有同时识别多个物体的能力，可以分割高度重叠的数字[12]。

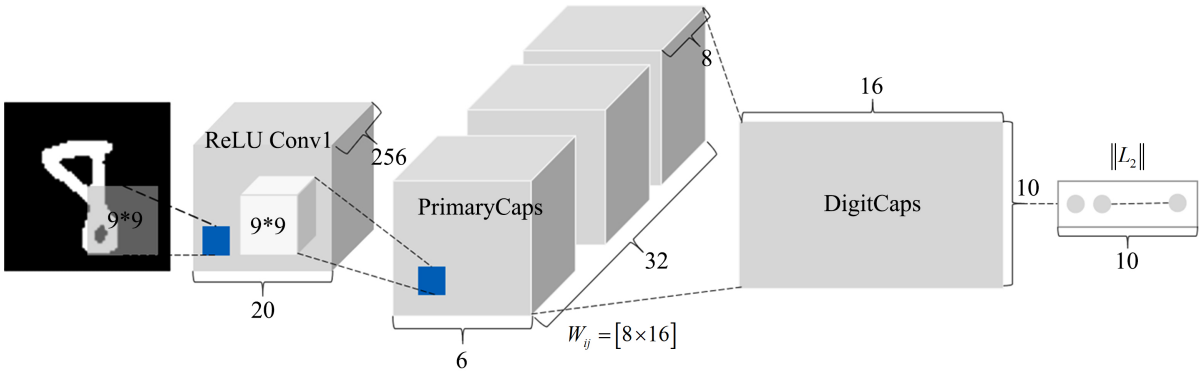

Figure 3. Capsule network
图 3. 胶囊网络

3.1. 胶囊网络结构

胶囊网络结构(图 3)包含两个卷积层(用 *Conv* 表示)和一个全连接层(用 *FC* 表示)，胶囊网络用 *CapsNet* 表示。

$$Input \to Conv1:[28,28,1] \xrightarrow{conv(9*9),stride=1,ReLU} [20,20,256] \tag{1}$$

第一个卷积层(*Conv*1)有 256 个卷积核大小为 9*9 的滤波器，步长为 1，无填充，激励函数为 *ReLU* 函数，得到一个 20*20*256 的输出，没有方向。该层主要作用就是对图像像素做一次局部特征检测，然后用作初级胶囊层的输入。

$$\begin{aligned} &Conv1 \to PrimaryCaps:[20,20,256] \xrightarrow{conv(9*9),stride=2} [6,6,8,32] \\ &\xrightarrow{reshape} [1152,8] \xrightarrow{activation} [1152,8]\left(u_i \in R^8\left(i=1,2,\cdots,1152\right)\right) \end{aligned} \tag{2}$$

第二个卷积层(*Conv*2)又叫做初级胶囊层(*PrimaryCaps*)，初级胶囊层有 32 个通道，维度是 8 维，即每个初级胶囊包含 8 个卷积单元，每个卷积单元的卷积核为 9*9，步长为 2。每个初级胶囊输出可以得到 256*9*9 个卷积单元的全部输出。初级胶囊总共有 32*6*6 个胶囊输出，每个输出是一个 8 维向量，每个胶囊在 6*6 的网格中共享权重，因此该胶囊层输出的维度是 6*6*8*32。

$$PrimaryCaps \to DigitCaps: u_i \in R^8 \xrightarrow{W_{ij}(j=0,1,\cdots,9)} \hat{u}_{j|i} = W_{ij}u_i$$
$$\xrightarrow{sum} s_j = \sum_i c_{ij}\hat{u}_{j|i} \xrightarrow{squash} v_j = squash\left(s_j\right) = \frac{\left\|s_j\right\|^2}{1+\left\|s_j\right\|^2}\frac{s_j}{\left\|s_j\right\|} \in R^{16} \tag{3}$$

最后一层是数字胶囊层(*DigitCaps*)，共有 10 个类别，每个类别是一个 16 维的胶囊，每个胶囊接受所有上一层胶囊的输出。初级胶囊层与数字胶囊层之间是全连接的，前者有 6*6*32 个元素，每个元素都是 1*8 的向量，而后者有 10 个元素(即 10 个数字)，每个元素都是一个 1*16 的向量，为了让 1*8 的向量与 1*16 的向量全连接，需要 6*6*32 个 8*16 的矩阵。初级胶囊层有 1152 个向量，而数字胶囊层有 10 个向量，根据动态路由算法进行迭代，计算耦合函数并输出 10 个向量 v_j。

因为胶囊的长度表示实体存在的概率，所以做分类时取向量的 L_2 范数即可。

$$Output\left(probability\right): v_j \xrightarrow{L_2} \left(\left\|v_0\right\|, \left\|v_1\right\|, \cdots, \left\|v_9\right\|\right) \tag{4}$$

两个连续的胶囊层之间的连接由动态路由实现，动态路由的任务是找到每一个低层胶囊的输出最有可能激活哪一个高层胶囊。

3.2. 动态路由

两个连续的胶囊层之间的连接由动态路由实现，第 l 层的胶囊 i 与第 $\left(l+1\right)$ 层的胶囊 j 之间的动态路由关系如下。

设胶囊 i 的输出为 u_i，胶囊 i 与胶囊 j 之间的权重矩阵为 W_{ij}，则胶囊 i 到胶囊 j 的预测向量 $\hat{u}_{j|i}$ 为：

$$\hat{u}_{j|i} = W_{ij}u_i \tag{5}$$

胶囊 i 与胶囊 j 之间的耦合系数 c_{ij} 为：

$$c_{ij} = \frac{\exp\left(b_{ij}\right)}{\sum_k \exp\left(b_{ik}\right)} \tag{6}$$

其中 b_{ij} 初始化为零，c_{ij} 由迭代动态路由过程决定。

b_{ij} 初始化为 0，目的是使得每个胶囊的初始耦合系数均为 $1/k$，k 为下一层胶囊个数，即低层任意一个胶囊预测的高层胶囊概率是相等的，并且和为 1。

第 $\left(l+1\right)$ 层胶囊 j 的输出总和 s_j 为：

$$s_j = \sum_i c_{ij} \cdot \hat{u}_{j|i} \tag{7}$$

对 s_j 使用压缩函数，当 s_j 为短向量时，输出向量 v_j 缩小到几乎为 0 的长度；当 s_j 为长向量时，输出向量 v_j 被压缩到略小于 1 的长度。使用的压缩函数 *squash* 为：

$$v_j = \frac{\left\|s_j\right\|^2}{1+\left\|s_j\right\|^2} \cdot \frac{s_j}{\left\|s_j\right\|} \tag{8}$$

在计算时，为了防止分母为零，在分母 $\left\|s\right\|$ 里加入小量 ε (取 10^(−7))，即：

$$\|s\| \approx \sqrt{\sum_i s_i^2 + \varepsilon} \tag{9}$$

路由迭代协议用输出向量和预测向量的点积表示，点积越大，夹角越小，一致性越好，协议 a_{ij} 为：

$$a_{ij} = v_j \cdot \hat{u}_{j|i} \tag{10}$$

在计算胶囊 i 连接到胶囊 j 的所有耦合系数的新值之前，将 a_{ij} 添加到初始 b_{ij} 中，用来更新参数 b_{ij}：

$$b_{ij} \leftarrow b_{ij} + a_{ij} \tag{11}$$

输出向量与预测向量一致时，a_{ij} 越大，耦合系数 c_{ij} 就被更新的越大，胶囊 i 被分配到胶囊 j 的可能性越大，即胶囊 j 被激活的概率越大，实体存在的概率越大。

动态路由把卷积神经网络的标量输出特征检测器替换成了向量输出，将最大池化层用路由协议机制代替，所以每个胶囊在前向传播时，优先前往下一个最相关的胶囊。

在动态路由过程中，路由迭代次数用 r 表示，本文研究随着路由迭代次数的变化，模型分类准确率的变化趋势，以确定模型的最优参数。

动态路由过程如下：

Procedure: Routing algorithm

procedure ROUTING $(\hat{u}_{j|i}, r, l)$
  for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
  for r iterations do
    for all capsule i in layer l: $c_i \leftarrow softmax(b_i)$
    for all capsule j in layer $(l+1)$: $s_j \leftarrow \sum_i c_{ij}\hat{u}_{j|i}$
    for all capsule j in layer $(l+1)$: $v_j \leftarrow squash(s_j)$
    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

3.3. 损失函数

因为胶囊允许多个分类同时存在，所以不能直接用传统的交叉熵损失，而是用间隔损失(*margin loss*)来判别模型的性能。每个胶囊分类的间隔损失为 L_k：

$$L_k = T_k \max\left(0, m^+ - \|v_k\|\right)^2 + \lambda\left(1 - T_k\right)\max\left(0, \|v_k\| - m^-\right)^2 \tag{12}$$

其中 k 是分类，T_k 是分类的指示函数，即当 k 类存在时，T_k 等于 1；当 k 类不存在时，T_k 等于 0。m^+ 为上界，此处取值为 0.9，惩罚假阳性，即预测 k 类存在但真实不存在；m^- 为下界，此处取值为 0.1，惩罚假阴性，即预测 k 类不存在但真实存在。λ 取 0.5。总的损失是各类损失之和。

由于 *margin* 损失直接固定了上下界，容易过早出现“失活”胶囊。因此在训练过程中线性增加间隔阈值，来修改模型的损失函数。

Spread 损失

为了降低训练对模型的初始化和超参数的敏感性，我们使用“扩散损失”(*spread loss*)来直接最大化目标类 a_t 的激活和其他类的激活之间的差距。如果激活一个错误的类 a_i，a_i 到目标类 a_t 的距离比阈值 m 更小，那么它的损失为：

$$L_i = \left(\max\left(0, m - \left(a_t - a_i\right)\right)^2\right), L_s = \sum_{i \neq t} L_i \tag{13}$$

其中 L_s 表示每个胶囊的 *spread* 损失之和，从较低的边距开始训练可以避免过早出现“失活”胶囊，阈值 m 从 0.2 开始，在每一代的训练后线性增加 0.1，m 增加到最大值 0.9 后停止增长。

3.4. 评价指标

用准确率作为模型的评价指标，准确率用 *acc* 表示：

$$acc = \frac{TP}{TP + FP} \tag{14}$$

其中 *TP* 表示预测正确的类别，*FP* 表示预测错误的类别。以 MNIST 数据集为例，真实数字为 1 时，只有预测数字为 1，*TP* 加 1，否则 *FP* 加 1。

3.5. 重构子网络

鲁棒性强的模型有一定的重构能力，如果模型能够重构，证明它至少有了一个良好的表示，并且从重构结果中可以看出模型存在的问题。解码器结构(图 4)用于从数字胶囊层表示重构数字，使用真实标签作为重构目标，重构时单独取出需要重构的向量，然后放到后面的三层全连接网络中重构，重构子网络最终输出的维度是 784，与最初输入的大小为 28*28 的图像维度相同，重构损失就是最终输出和最初输入的 784 个单元上的像素值的欧氏距离，用 L_c 表示。

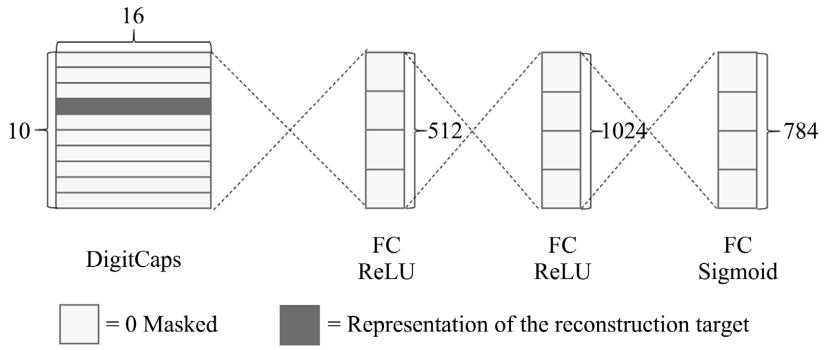

Figure 4. Decoder
图 4. 解码器

模型的总损失函数为 L：

$$L = L_s + \alpha L_c \tag{15}$$

其中 α 取 0.0005，对重构损失进行缩放，*spread* 损失依旧占主导地位。

4. 实验

4.1. 数据集

本文以 MNIST 数据集为例来验证胶囊网络在数字识别上的性能，然后在 Fashion-MNIST，CIFAR-10 数据集上也进行了验证。

MNIST 数据集[13]全称为 *Modified National Institute of Standards and Technology*，其中训练集由 250 个不同的人手写的数字构成(0~9，共 10 个类别)，其中 50%是高中学生，50%是来自人口普查局的工作人员，总共 60,000 个数字，在训练集中，55,000 个数字作为训练集，5000 个数字作为验证集来调整模型参

数；而测试集也是同样比例的手写数字数据，总共 10,000 个数字。每幅图像为一个 28*28 像素的单元。

4.2. 实验环境与参数设置

本文在 Tensorflow 环境[14]下训练网络模型，使用 *pycharm*，采用 *Adam* 优化器[15]，用于梯度下降。在梯度下降时，模型参数更新之前需要处理的样本数用 *batch* 表示[16]，它的大小在 1 到训练集总数之间；学习算法在整个训练集上的工作次数为训练轮数，用 *epoch* 表示，每一轮训练都完整的遍历一次训练集。将胶囊网络应用到数据集上，在训练时不使用旋转和缩放来集成和扩充数据。

在训练集上，*batch* 大小取 128，*epoch* 取 150，路由迭代次数分别取 1, 2, 3, 4, 5，每 100 步输出一次平均损失，每 500 步输出一次平均精度，步数用 *step* 表示，损失用 *loss* 表示，精度用 *acc* 表示，*iter_routing* 表示路由迭代次数。为了验证重构的重要性，本文在不加重构子网络的情况下进行实验。

4.3. 实验结果

当 epoch 为 150，路由迭代次数为 1, 2, 3 时，训练损失(图 5)与训练精度(图 6)如下：

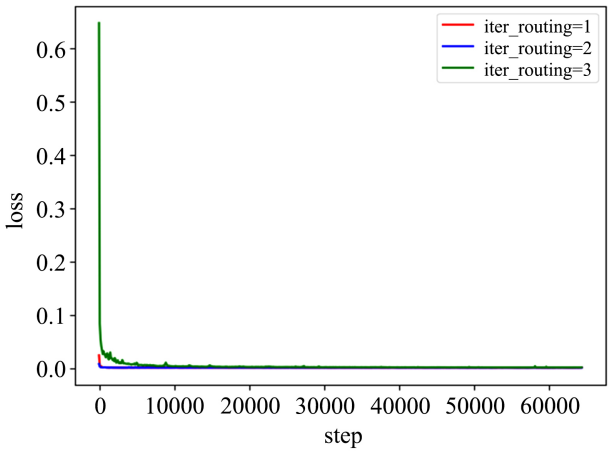

Figure 5. Training loss when epoch is equal to 150 and the number of iterations is 1, 2 and 3 respectively
图 5. Epoch = 150，迭代次数分别为 1, 2, 3 时的训练损失

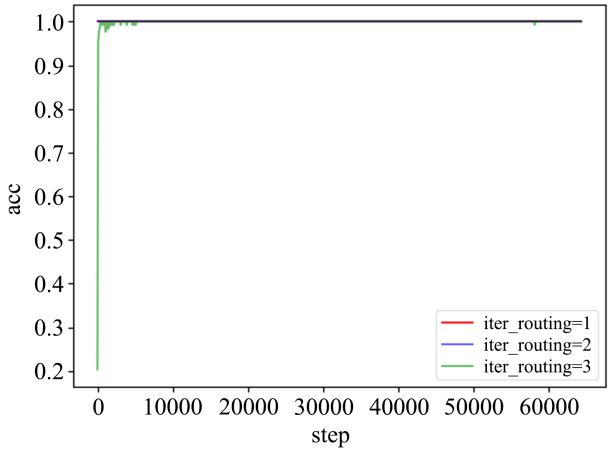

Figure 6. Training accuracy when epoch is equal to 150 and the number of iterations is 1, 2 and 3 respectively
图 6. Epoch = 150，迭代次数分别为 1, 2, 3 时的训练精度

图 5 和图 6 表明，在训练时，路由迭代次数为 1, 2, 3 时，模型的准确率均已达到 99%以上，训练轮数为 150 时，模型的训练效果已经很好，为验证模型的泛化能力，将其用于测试集。

如图 7 所示，当 *epoch* 为 150，路由迭代次数为 4 和 5 时，损失函数不收敛。图 7 表明随着迭代次数的增加，分类精度不会一直增加，达到一定的精度后，迭代次数增加，分类精度反而会下降，这是因为迭代次数的增加会导致损失函数不收敛，因此确定迭代次数的大小是非常重要的。

当 epoch 为 150，路由迭代次数分别为 1, 2, 3 时，测试精度如图 8 所示。图 8 表明在不加重构子网络的情况下，路由迭代次数为 2 时，误分率最小为 0.32%。

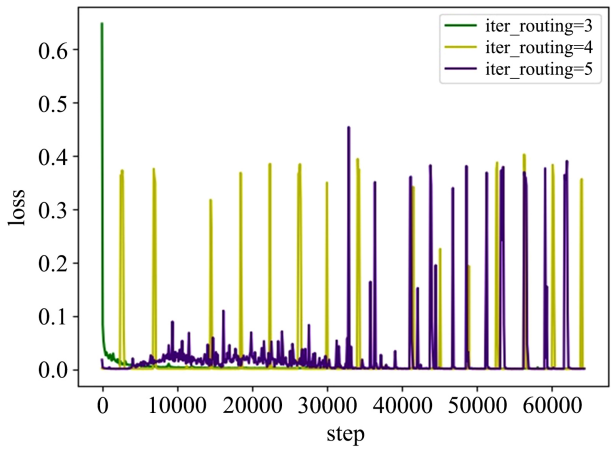

Figure 7. Training loss when epoch is equal to 150 and the number of iterations is 3, 4 and 5 respectively
图 7. Epoch = 150，迭代次数分别为 3, 4, 5 时的训练损失

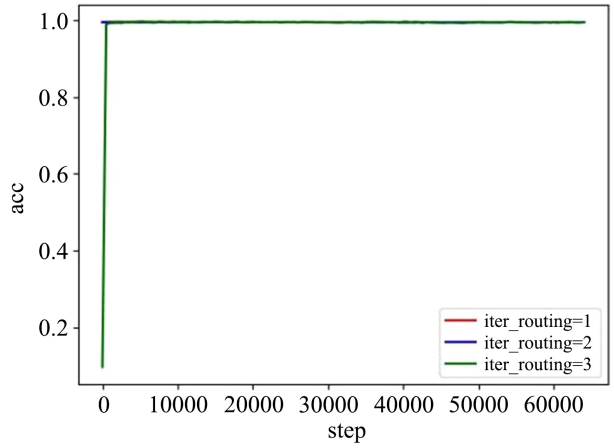

Figure 8. Test accuracy when epoch is equal to 150 and the number of iterations is 1, 2 and 3 respectively
图 8. Epoch = 150，迭代次数分别为 1, 2, 3 时的测试精度

Wan L 等[17]通过旋转和缩放来集成和扩展数据，在 MNIST 数据集上的误分率减小到了 0.21%，本文的基线模型是 Wan L 等人提出的模型在未经过增强和扩展数据时实现 0.39%的误分率。本文使用单一的模型进行测试，而没有任何模型平均，在不加重构的情况下，当 *epoch* 为 150，路由迭代次数为 2 时，该网络模型在 MNIST 数据集上的识别效果最好，测试集误分率为 0.32%，这是由于实验时并未做重构，所以模型的准确率要低于 Hinton 等人提出的胶囊网络所得到的准确率(误分率为 0.25%)，如表 1 所示。

Table 1. Misclassification rate of MNIST dataset classification test
表 1. MNIST 数据集分类测试误分率

方法	路由迭代次数	重构	误分率(%)
基线(Wan L 等)	-	-	0.39
胶囊网络	1	无	0.36
胶囊网络	2	无	**0.32**
胶囊网络	3	无	0.35
胶囊网络(Hinton 等)	3	有	**0.25**

表 1 表明了不同胶囊网络参数设置在 MNIST 数据集上的测试错误率，并显示了路由和重构正则化的重要性。添加重构可以通过在胶囊向量中强制执行姿态编码来提高路由性能。基线是一个标准的卷积神经网络，有三个卷积层，分别有 256、256、128 个通道，每个卷积核大小为 5，步长为 1，最后的卷积层之后是两个大小为 328,192 的全连接层，具有 *dropout* 的最后一个全连接层连接到具有交叉熵损失的 10 类 *softmax* 层，基线使用 *Adam* 优化器在 2-*pixel shifted* MNIST 数据集上进行训练，基线设计用于在 MNIST 上实现最佳性能的同时使计算成本尽可能接近胶囊网络，基线的参数数量有 35.4 M，胶囊网络有 8.2 M 参数，不加重构子网络的胶囊网络有 6.8 M 参数。

实验表明，与传统的卷积网络相比，每个数字胶囊对每个类别都有更鲁棒的表示。由于手写数字的倾斜、旋转、风格等存在自然差异，训练后的 CapsNet 对训练数据的仿射变换具有一定的鲁棒性。

4.4. 其他数据集

Fashion-MNIST 数据集[18]包含了 10 个类别的图像，分别是：*t-shirt* (*T* 恤)，*trouser* (裤子)，*pullover* (套衫)，*dress* (裙子)，*coat* (外套)，*sandal* (凉鞋)，*shirt* (衬衫)，*sneaker* (运动鞋)，*bag* (包)，*ankle boot* (短靴)。训练数据集每个类别含有 6000 个样本，测试数据集每个类别含有 1000 个样本，训练集共 60,000 个样本，测试集共 10,000 个样本。每幅图像为 28*28 像素的单元。

CIFAR-10 数据集包含 10 个类别的 RGB 彩色图片，分别是：飞机(*aplane*)、汽车(*automobile*)、鸟类(*bird*)、猫(*cat*)、鹿(*deer*)、狗(*dog*)、蛙类(*frog*)、马(*horse*)、船(*ship*)和卡车(*truck*)。图片的尺寸为 32*32，数据集中一共有 50,000 张训练图片和 10,000 张测试图片。

本文利用胶囊网络训练和测试 MNIST 数据集，对不同路由迭代次数、不同 *epoch* 下的准确率进行比较，以得出模型的最优参数。随着迭代次数的增加，分类精度不会一直增加，达到一定的精度后，迭代次数增加，分类精度反而会下降，这是因为迭代次数的增加会导致损失不收敛。经过训练的多层胶囊系统在 MNIST 数据集上达到了最先进的性能，并且在识别高度重叠的数字方面比卷积网络要好得多。使用重构子网络能够提高胶囊网络的性能，引进重构比没引进重构的识别误差要小；同时改进胶囊网络在 Fashion-MNIST，CIFAR-10 数据集上也表现出了良好的性能。

5. 总结与展望

在传统的神经网络中，只有一个单独单元的输出被一个非线性变换压缩，当有许多神经元被输出时，对每一个都采用非线性变换，而胶囊网络将这些输出神经元聚合在一个张量神经元中，并对整个胶囊采用非线性变换，也就是说，应用非线性变换的时候，是对整个网络进行操作，而不是对单独的神经元。动态路由把卷积神经网络的标量输出特征检测器替换成了向量输出，将最大池化层用路由协议机制代替，

所以每个胶囊在前向传播时，优先前往下一个最相关的胶囊。

这个新架构也要付出相应的代价：路由算法相比于普通的卷积神经网络，前向传播有一个额外的外层循环，它需要在所有单元上进行 r 次迭代来计算输出。对于每一个嵌套在一层里的张量神经元采用这些操作时，无论是 *softmax* 或是压缩函数，都会使得梯度更难计算，模型可能会在一些较大的数据集中遇到梯度消失的问题。

深度学习[19]本质就是一系列的张量变换，胶囊将神经元的输入和输出升级成二维向量，在接下来的研究中可以延伸为高维张量，目前胶囊网络只有三层，增加网络层数很可能提高模型性能，动态路由过程还有待改进。此外，胶囊网络在复杂数据集上实现的性能并不是很好，该网络还有很大的改进空间。

基金项目

国家自然科学基金面上项目(11771011)。

参考文献

[1] Peduzzi, P., Concato, J., Kemper, E., *et al*. (1996) A Simulation Study of the Number of Events per Variable in Logistic Regression Analysis. *Journal of Clinical Epidemiology*, **49**, 1373-1379. https://doi.org/10.1016/S0895-4356(96)00236-3

[2] Lawrence, S., Giles, C.L., *et al*. (1997) Face Recognition: A Convolutional Neural Network Approach. *IEEE Transactions on Neural Networks*, **8**, 98-113. https://doi.org/10.1109/72.554195

[3] Zeiler, M.D. and Fergus, R. (2013) Visualizing and Understanding Convolutional Neural Networks. *European Conference on Computer Vision*, Sydney, 1-8 December 2013, 818-833.

[4] Zeiler, M.D. and Fergus, R. (2013) Stochastic Pooling for Regularization of Deep Convolutional Neural Networks.

[5] Technicolor, T., Related, S., Technicolor, T., *et al*. (2012) ImageNet Classification with Deep Convolutional Neural Networks.

[6] Simonyan, K. and Zisserman, A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. 3*rd International Conference on Learning Representations*, San Diego, 7-9 May 2015, 1-12.

[7] Szegedy, C., Liu, W., Jia, Y., *et al*. (2014) Going Deeper with Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Boston, 7-12 June 2015, 1-9. https://doi.org/10.1109/CVPR.2015.7298594

[8] Szegedy, C., Vanhoucke, V., Ioffe, S., *et al*. (2016) Rethinking the Inception Architecture for Computer Vision. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 2818-2826. https://doi.org/10.1109/CVPR.2016.308

[9] He, K., Zhang, X., Ren, S., *et al*. (2016) Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90

[10] Huang, G., Liu, Z., Laurens, V., *et al*. (2016) Densely Connected Convolutional Networks. *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 2261-2269.

[11] Sabour, S., Frosst, N. and Hinton, G.E. (2017) Dynamic Routing between Capsules. arXiv:1710.09829 [cs.CV]

[12] Ba, J., Mnih, V. and Kavukcuoglu, K. (2014) Multiple Object Recognition with Visual Attention. 3*rd International Conference on Learning Representations*, *ICLR* 2015, San Diego, 7-9 May 2015, 1-10.

[13] Lecun, Y. and Cortes, C. (2010) The MNIST Database of Handwritten Digits. http://yann.lecun.com/exdb/mnist

[14] (2016) TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.

[15] Kingma, D. and Ba, J. (2014) Adam: A Method for Stochastic Optimization. 3*rd International Conference on Learning Representations*, *ICLR* 2015, San Diego, 7-9 May 2015, 1-15.

[16] Chang, J.R. and Chen, Y.S. (2015) Batch-Normalized Maxout Network in Network. *Proceedings of the* 33*rd International Conference on Machine Learning*, New York, 20-22 June 2016, 1-9.

[17] Wan, L., Zeiler, M., Zhang, S., *et al*. (2013) Regularization of Neural Networks Using Dropconnect. *International Conference on Machine Learning*, *PMLR*, Atlanta, 17-19 June 2013, 1058-1066.

[18] Xiao, H., Rasul, K. and Vollgraf, R. (2017) Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:1708.07747 [cs.LG]

[19] Lecun, Y., Fu, J.H. and Bottou, L. (2004) Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. *Proceedings of the* 2004 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington DC, 27 June-2 July 2004, II-104. https://doi.org/10.1109/CVPR.2004.1315150