Group Reconstruction and Max-Pooling Residual Capsule Network

Xinpeng Ding^{1,2}, **Nannan Wang**^{1,3*}, **Xinbo Gao**^{1,2}, **Jie Li**^{1,2} and **Xiaoyu Wang**^{4}
^{1}State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China
^{2}School of Electronic Engineering, Xidian University,Xi'an, China
^{3}School of Telecommunications Engineering, Xidian University, Xi'an, China
^{4}IntelliFusion, Shenzhen, China
xpding.xidian@gmail.com, nnwang@xidian.edu.cn, {xbgao, leejie}@mail.xidian.edu.cn, fanghuaxue@gmail.com

Abstract

In capsule networks, the mapping of low-level capsules to high-level capsules is achieved by a routing-by-agreement algorithm. Since the capsule is made up of collections of neurons and the routing mechanism involves all the capsules instead of simply discarding some of the neurons like Max-Pooling, the capsule network has stronger representation ability than the traditional neural network. However, considering too much low-level capsules' information will cause its corresponding upper layer capsules to be interfered by other irrelevant information or noise capsules. Therefore, the original capsule network does not perform well on complex data structure. What's worse, computational complexity becomes a bottleneck in dealing with large data networks. In order to solve these shortcomings, this paper proposes a group reconstruction and max-pooling residual capsule network (GRMR-CapsNet). We build a block in which all capsules are divided into different groups and perform group reconstruction routing algorithm to obtain the corresponding high-level capsules. Between the lower and higher layers, Capsule Max-Pooling is adopted to prevent overfitting. We conduct experiments on CIFAR-10/100 and SVHN datasets and the results show that our method can perform better against state-of-the-arts.

1 Introduction

In recent years, convolutional neural network (CNN) has been a main deep learning model for computer vision tasks [Krizhevsky *et al.*, 2012]. However, internal data representation of a convolutional neural network does not take into account important spatial hierarchies between simple and complex objects [Hinton *et al.*, 2018]. Especially, viewpoint invariance is achieved by pooling [Simonyan and Zisserman, 2014] so that the small perturbations in the input can not effect the output too much [Cohen and Welling, 2016], [Cohen *et al.*, 2018]. This results in the loss of information about the internal properties of present entities (e.g location, orientation, shape and pose) in an image and relationships between them. Capsule network is a new type network introduced in [Sabour *et al.*, 2017] for image classification. There are two key features distinguishing them from CNNs.

On one hand, rather than using a scalar, a capsule is a group of neurons which can denote many more properties such as pose, viewpoint, velocity, grain or regions, etc. In this way, the network can not only extract features, but also extract variants of them. The introduction of capsules provides a new method of feature learning, which is represented by the parameterization of the entity vector (named capsule) to represent a wide variety of attributes, so that better robustness can be obtained.

On the other hand, pooling is replaced with a routing scheme to send lower-level capsule (e.g nose, mouth, ears etc.) outputs as input to parent capsule (e.g face) that represent part-whole relationships to achieve translation equivariance and untangles the coordinate frame of an entity through linear transformations. Such a scheme can be seen as a feature clustering and be optimized by coordinate descent through several iterations. However, the computational complexity in the routing scheme is fairly high [Chen and Crandall, 2018]. The capsule network has shown its potential by achieving a state-of-the-art result of 0.25% test error on MNIST , better than the previous baseline of 0.39%. However, its performance on complex data, such as CIFAR-10 [Krizhevsky and Hinton, 2009], seems not so good, only 68.93% [Xi *et al.*, 2017].

Compared with pooling, routing will take every lower capsule learned into account when mapping from lower capsules to higher counterparts. This scheme works well on MNIST [LeCun, 1998] according to handwritten digits hardly have noise so that every capsule in lower layer represents the desired features. However, normal images like CIFAR-10 [Krizhevsky and Hinton, 2009] have much more noise which can cause that the higher capsules achieve too much useless information such as background. This will disrupt useful lower capsule routing to its higher capsules. The computational complexity becomes too high while dealing with large datasets [Ben-Israel and Greville, 2003], [Li *et al.*, 2018].

To tackle these drawbacks, we propose a group reconstruction and max-pooling residual capsule network (GRMR-CapsNet). It can significantly reduce computational complex-

*Correspondingauthor:NannanWang(nnwang@xidian.edu.cn) ity and improve the recognition accuracy for complex data such as images in CIFAR-10/100. There are three main contributions in our paper:

- Different from the method of multiplying the low-level capsule dimension to the high-level capsule by multiplying the viewing angle invariant matrix in [Sabour *et al.*, 2017], [Hinton *et al.*, 2018], we use a convolutional layer named *Convolutional Transformation Layer* to compute capsules within a local receptive field so as to reduce the complexity of computation.
- We propose a new routing scheme named *Group-Reconstruction Routing Algorithm* to map from low-level capsules to high-level capsules in a supervised way. Our method firstly divides all capsules in the lower layer into different groups and then routes them separately in each group. By doing this, some capsules that are divided into the same group will be given a very low weight (almost zero) in the route of the next layer, while other capsule groups important can achieve high weight. For all capsules in one group share the parameters in routing, the computational complexity can also be reduced. For all capsules in one group share the parameters in routing, the computational complexity can also be reduced.
- We design a *Capsule Max-Pooling Residual Block* which can adaptively choose the routing or max-pooling. Specifically, we add capsule-based max-pooling as a skip connection into each block. Formally, denoting the routing mapping as $R(x)$. We can get the higher level capsules u from the lower level capsules v by the mapping of $u = R(v)$. Specifically, we let $M(x)$ denote the max-pooling, the original mapping is changed into $u = R(v) + M(v)$. It is conceivable that the mapping of the network tends to $u \approx M(v)$ when the pooling effect is better, and the mapping of that tends to $u \approx M(v)$ when the routing effect is better.

We firstly conduct experiments on CIFAR-10 to show that our proposed group capsule routing have explicit improvement over original routing algorithm. Then, we present comprehensive experiments on CIFAR-10/100 and SVHN to evaluate our network, which shows that our method have a positively improvement against state-of-the-arts.

2 Related Work

2.1 Capsules

[Sabour *et al.*, 2017] presented a vector consisting of neurons, called capsule, whose orientation represents the properties of the entity and whose length represents the probability of it. In [Hinton *et al.*, 2018], a capsule was introduced by a 4x4 pose matrix and an activation probability which can respectively evaluate the properties and existence of entities.

2.2 Routing

According to [Sabour *et al.*, 2017], let u_i, v_j denote the lower and higher capsules in a layer respectively, where i, j indicate the index of capsules. Use d_1, d_2, n_1, n_2 to represent the dimensions and number of capsules in the lower and higher layers, i.e., $\{u_i \in R^{d_1}\}_{i=1}^{n_1}$, $\{v_j \in R^{d_2}\}_{j=1}^{n_2}$. The first step is mapping from the dimension of u_i to that of v_j by $\nu_{j|i} = w_{ij} \cdot u_i$ where $w_{ij} \in R^{d_2 \times d_1}$ is a transform matrix. The second step is an agreement routing method to cluster all lower capsules into higher ones. The transformed capsules are multiplied by a routing coefficient:

$$s_j^{(r)} = \sum_i c_{ij}^{(r)} \cdot \nu_{j|i}. \tag{1}$$

The process of dynamically updating c_{ij} could be deemed as voting. Firstly, given $b_{ij}^{(r)} \leftarrow 0, r \leftarrow 0$, then:

$$b_{ij}^{(r+1)} \leftarrow b_{ij}^{(r)} + \nu_{j|i} \cdot v_j^{(r)}, \tag{2}$$

where $v^{(r)}$ is computed from $s^{(r)}$ via squash($\cdot$):

$$v = \frac{\|s\|^2}{1 + \|s\|^2} \frac{s}{\|s\|}. \tag{3}$$

The coefficients in the routing algorithm is updated by an unsupervised approach iteratively which finds the best voting c [Sabour *et al.*, 2017]. In [Hinton *et al.*, 2018], the routing algorithm was replaced by a modified EM-algorithm to fit a gaussian mixture model (GMM). Given $R_{ij} \leftarrow 1/n_2$, the EM routing clusters n_1 capsules the dimensions of d_1 into n_2 capsules the dimensions of d_2 iteratively:

$$\alpha_j^{(r)}, \mu_j^{(r)}, \sigma_j^r \leftarrow M - step[\alpha_i, R_{ij}^{(r)}, \nu_{j|i}], \tag{4}$$

$$R_{ij}^{(r+1)} \leftarrow E - step[\alpha_j^{(r)}, P_{j|i}(\nu_{j|i}, \mu_j^{(r)}, \sigma_j^r)]. \tag{5}$$

where α_i represents the activation probability of low-level capsules u_j. $M - step$ produces the activation α_j together with the mean μ_j and std standard deviation σ_j in high layer. Then, the co-efficients R_{ij} is updated by the the $E - step$.

In [Lenssen *et al.*, 2018], they proposed a group equivariant capsule network (GECN) which can guarantee equivariance and invariance properties just like the group convolution. That paper also introduced a scheme for routing by agreement algorithms, a spatial aggregation method, and the ability to integrate group convolutions. In [Zhang *et al.*, 2018a], they proposed a orthogonal projections onto capsule subspaces method to capture features. Compared with introducing some brand new architectures for capsule nets,that paper tends to learn a group of capsule subspaces to represent a set of entity classes. In [Zhang *et al.*, 2018b], they tried to extract multi-labeled relation with the capsule network by add attention scheme into routing-by-agreement algorithm.

3 Methodology

3.1 Convolutional Transformation Layer

As mentioned in Section 2, before they are fed into routing-by-agreement algorithm, capsules in low layer u with the dimension of d_1 should be mapped to ν with the dimension of d_2 by a transformation matrix $w_{ij} \in R^{d_2 \times d_1}$. However, the number of parameters would become very large with an increasing dimension d_1 and d_2. According to [Hinton *et al.*,

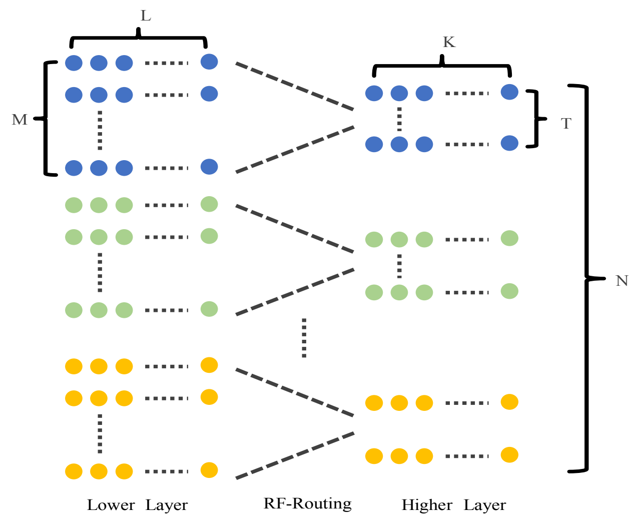

Figure 1: Group Capsule Layer. Capsules in the lower layer is divided into N groups. Capsules in the higher layer are obtained by the *Reconstruction Feedback Routing* (RF-Routing) in the same group.

2011], [Sabour *et al.*, 2017], We can summarize the two main roles of the matrix w_{ij}. Firstly, the matrix w_{ij} encodes important spatial and other relationships between u_i which represent lower level feature (eyes, mouth and nose) and v_j higher level feature (face). Secondly, matrices attempt to learn to transform low-layer capsules into high-layer capsules from a different viewpoint.

Based on the above analysis that only information between adjacent capsules is beneficial, we propose *Convolutional Transformation Layer* (CTL) to extract the local perspective and relationship of all capsules. Specifically, the voting $\nu_{t|m}$ is obtained by u_m fed into convolutional capsule layer with 32 group convolution [Krizhevsky *et al.*, 2012] which indicates 32 capsule types, a receptive field of 3×3, a padding of 1 and a stride of 1×1.

3.2 Group Reconstruction Routing Algorithm

In this section, we will detail the proposed Group Reconstruction Routing Algorithm which can effectively avoid information confusion during the mapping from the low-level capsule to the high-level capsule. The low-level capsules firstly are fed into Group Capsule Layer, in which capsules are divided into several groups. Following is the Reconstruction Feedback Routing which can use back propagation to update the coefficients compared with iterative ones.

Group Capsule Layer

According to [Sabour *et al.*, 2017] , the routing-by-agreement algorithm could find a potential relationship between the capsules in lower layers and those in higher layers, that is, if the two capsules are more similar, their cosine similarity will be higher. Nevertheless, for a capsule, it is not directly related to all other capsules, so it is not reasonable to route the entire capsules in lower layers. What's worse, all capsules learned can hardly be used to represent features, because a significant

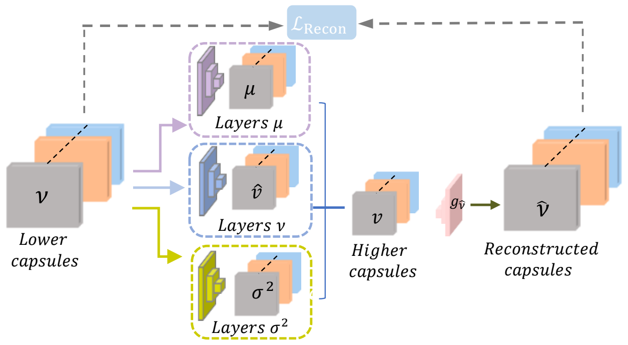

Figure 2: Reconstruction Feedback Routing. The input are lower layer capsules ν transformed by *Convolutional Transformation Layer* (CTL). Following are three stacked layers to learn μ, $\hat{v}$ and σ^2 which directly express higher-layer capsules v as the output. Then, we use generation layers $g_{\hat{u}}$ to reconstruct $\hat{u}$ and compare the distance between u and $\hat{u}$ and use this distance as feedback to train the network.

portion of the capsules are noise ones. Capsules in higher layers achieving the noisy-mixed capsules in lower layers, capsule networks works not very well on complex data.

Taking this into account, we propose a method which divides capsules into several groups $\{G_n\}$, trying to learn the capsules in each group are as similar as possible. As shown in Figure 1, we divide capsules into N groups of which M capsules make up. The length of each capsule is L. Capsules in different groups are present by different colors. For example, the first capsule group is shown by blue color and the second one is present by green. In each group n, we aggregate M capsules into T capsules with the length of K by the Group Reconstruction Routing algorithm which shares the same variables, so the computational complexity in routing could be reduced. Totally, we can obtain $N \times T$ capsules in the higher layer. For example, for a face, we should learn capsule groups that represent the eyes, nose, ears, mouth and so on respectively.

Reconstruction Feedback Routing

In each group which is described in the above section, what we get are capsules with similar information. In [Hinton *et al.*, 2018], Gaussian Mixed Model (GMM) plays the role of the routing-by-agreement algorithm. Formally, let ν_{ij}, R_{ij}, α_{ij} denote voting, the pose entity and the activation, then we can get:

$$\begin{aligned}&\mathcal{N}(\boldsymbol{\nu_{ij}}; \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2)\\&= \prod_{l=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_j^l} \exp\left(-\frac{1}{2(\sigma_j^l)^2}(\nu_{ij}^l - \mu_j^l)^2\right),\end{aligned} \tag{6}$$

$$p_{ij} \leftarrow \mathcal{N}(\boldsymbol{\nu}_{ij}; \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2), \tag{7}$$

$$R_{ij} \leftarrow \frac{\alpha_j P_{ij}}{\sum_{j=1}^{T} \alpha_j P_{ij}}, \tag{8}$$

where μ_j and σ_j^2 is the the mean and std standard deviation of v_j.

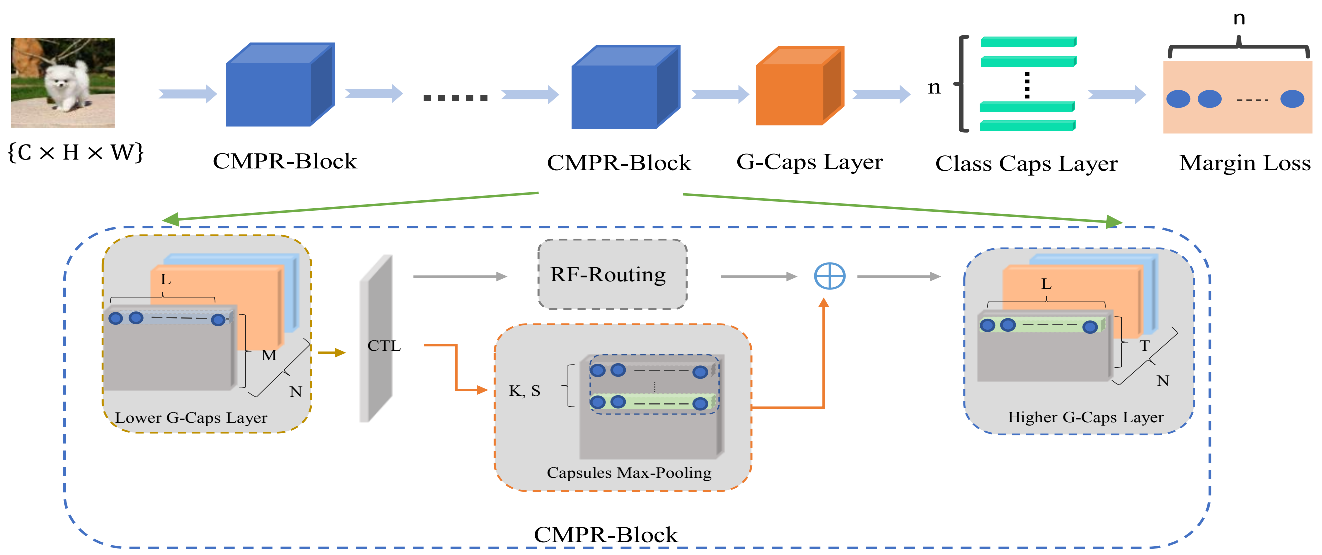

Figure 3: GRMR-CapsNet. The network firstly consists of four CMPR-Blocks, following is the Group Capsule Layer. The last layer is a normal capsule layer which generates n class capsules. Margin loss is used in our paper.

In this paper, we directly learn μ_j and σ_j^2 by back propagation algorithm. Similarly to [Kingma and Welling, 2013], let us consider capsules in high layers $\boldsymbol{V} = \{\boldsymbol{v}_t\}_{t=1}^{T}$ which are generated by capsules in low layers $\boldsymbol{U} = \{\boldsymbol{u}_m\}_{m=1}^{M}$. Given the $p(\boldsymbol{V}|\boldsymbol{U})$ which represents the posterior probability of V for U and $p(\boldsymbol{U})$ the distribution of U, the distribution $p(\boldsymbol{V})$ can easily be obtained by the corresponding formula:

$$p(\boldsymbol{V}) = \sum p(\boldsymbol{V}|\boldsymbol{U})p(\boldsymbol{U}). \tag{9}$$

Therefore, if $\{\boldsymbol{u}_m\}_{i=m}^{M}$ and $\{\boldsymbol{v}_t\}_{t=1}^{T}$ are i.i.d. and obey the Gaussian distribution, we can know that $p(\boldsymbol{v_t}|\boldsymbol{u_m}) \sim \mathcal{N}(\boldsymbol{v_t}; \boldsymbol{u_m}, \boldsymbol{\sigma}_m^2)$.

Based on the above analysis, we build two neural networks: $\mu_t = f_\mu(\nu_{t|m}), \sigma_t^2 = f_\sigma(\nu_{t|m})$. so as to express $p(\boldsymbol{V})$ directly by formula:

$$v_t = \mathcal{N}(\boldsymbol{\hat{v}_t}; \boldsymbol{u_{t|m}}, \boldsymbol{\sigma}_{t|m}^2), \tag{10}$$

where $\hat{v}_t$ is computed by a neural network $f_v(\nu_{t|m})$. We can also get the reconstructed U through a generator $\hat{U} = g(V)$. Let us define $\mathcal{D}(\hat{u}_m, u_m)^2$ as the distance of the $\hat{u}_m$ and u_m, then we can get the Reconstruction Loss:

$$\begin{aligned}\mathcal{L}_{Recon} &= \sum_{m}^{M} \mathcal{D}(\hat{u}_m, u_m)^2 \\ &= \sum_{m}^{M} \cos(\Theta_m) = \sum_{m}^{M} \frac{u_m \cdot \hat{u}_m}{|u_m|\,|\hat{u}_m|}.\end{aligned} \tag{11}$$

We use cosine similarity to describe the distance between the reconstructed capsules and the original capsules.

Then, in each group, capsules are aggregated into higher layer capsules by Group Reconstruction Feedback Routing Algorithm with $\mathcal{L}_{Recon}$ restricted shown in Algorithm1. Firstly, the transformed capsules in lower layers are are evenly divided into N groups, each group corresponding to the same type of capsules, which will avoid higher layer capsules mixing in useless information in the routing. In order to reduce the number of parameters to build a deeper network, we improve the original method [Hinton *et al.*, 2018] which iteratively obtains higher layer capsules using the EM algorithm. Instead, we design three stacked layers: f_μ, f_σ and f_v to directly fit the corresponding mean μ, variance σ^2 and capsules $\hat{v}$. Then, the higher-layer capsules are obtained by the Equation 10. Finally, based on the hypothesis that the higher layer capsules can reconstruct the lower layer capsules, we design generation layers $g_{\hat{u}}$ to reconstruct the lower layer capsules, and compare their similarity by Equation 11, and add this similarity loss to the BP algorithm to update f_μ, f_σ and f_v and $g_{\hat{u}}$ in a feedback manner. Compared to [Sabour *et al.*, 2017][Hinton *et al.*, 2018], our method can have a better performance in a supervised way.

Algorithm 1 Group Reconstruction Routing Algorithm.

```
Input: low-level capsules $u_m$
Output: high-level capsules $v_t$
 1: for all capsules $u_m$: $\nu_{t|m} \leftarrow CTL_t(u_m)$
 2: divide all capsule $u_m$ into groups $G_n$;
 3: for all groups $G_n$ do
 4:     for all capsule $\nu_{t|m} \in G_n$:
        $\mu_{t|m} \leftarrow f_\mu(\nu_{t|m})$, $\sigma_{t|m}^2 \leftarrow f_\sigma(\nu_{t|m})$, $\hat{v}_t \leftarrow f_v(\nu_{t|m})$
 5:     for all capsules $\hat{v}_t$: $v_t \leftarrow \mathcal{N}(\boldsymbol{\hat{v}_t}; \boldsymbol{u_{t|m}}, \boldsymbol{\sigma}_{t|m}^2)$
 6:     for all capsules $v_t$: $\hat{u}_m \leftarrow g_{\hat{u}}(v_t)$
 7: end for
 8: for all capsules $\hat{u}_m$: $\mathcal{L}_{Recon} = \mathcal{D}(\hat{u}_m, u_m)^2$
 9: $f_\mu, f_\sigma, f_v, g_{\hat{u}} \leftarrow BP(\mathcal{L}_{Recon})$
10: return $v_t$
```

3.3 Capsule Max-Pooling Residual Block

As we know, once the pooling is proposed, it has been widely used because of its good performance on the reduction of parameters to avoid overfitting. Considering its advantages that routing does not have, we propose a capsule-based max-pooling and add it to every few stacked layers which is shown in Figure 3. Formally, we define the block named *Capsule Max-Pooling Residual Block* (CMPR-Block) as:

$$v = R(\nu) + M(\nu), \tag{12}$$

here ν are the capsules transformed by CTL in the lower layer and v are higher layer capsules and the dimensions of $M(x)$ and $R(x)$ are equal. The function $R(\nu)$ represents the RF-Routing which is described in section 3.2 and the function $M(\nu)$ indicates the capsule max-pooling. As Figure 3 shows, in each group, we firstly compute the l_2 norm for all capsules: $\|v\|_2$, then we choose the max value of these l_2 norms with the kernel size of k and stride of s. The whole process can be simply described as: $v_{res} = max(\|v\|)_2$. The operation $R + M$ is performed by element-wise addition just like [He *et al.*, 2016]. The shortcut connections shown in Equation 12 will not bring extra calculations and parameters, which will be shown in Section 4.3.

3.4 Network Architecture

The architecture of *Group Reconstruction and Max-Pooling Residual Capsule Network* (GRMR-CapsNet) is shown in Figure 3. The input to the network is a $3 \times 32 \times 32$ image. The network begins with the four CMPR-Block described in Section 3.3 . Each block contains 32 groups in which the number of capsules is equal. In the first CMPR-Block, each group has 32×32 capsules with a dimension of 2. In the following layers, the number of capsules in each group is gradually halved, while the dimensions of the capsule are gradually doubled (ig, the second layer has 16×16 capsules with a dimension of 4, and the third layer has 8×8 capsules with a dimension of 8 .....). In the end, we will get 32 groups of 4×4 capsules each with a dimensions of 16 and these capsules are fed into the Group Capsule Layer. After routing by G-Caps Layer, there is only one capsule left in each group. Then, the capsules of these groups are sent to the Class Caps Layer, similarly to the primary caps layer in [Sabour *et al.*, 2017], which is a fully connected capsule layer. Capsules in different groups are clustered into n types of capsules by the dynamic routing algorithm, and the length of each obtained capsule represents the activation. In this paper, the loss we used is the margin loss proposed by [Sabour *et al.*, 2017].

3.5 Margin Loss

Just like [Sabour *et al.*, 2017], we use the length of the capsule to represent the probability of its entity existence. We want the capsule in the top layer for image class n to have the most significant length if and only if that the n class image is fed into the network. To allow for multiple classes, a separate margin loss L_n follows, for each image capsule, n:

$$L_n = \sum_{n=1}^{N} (T_n max\left(0, m^+ - \|v_n\|\right)^2 + \lambda \left(1 - T_n\right) max\left(0, \|v_n\| - m^-\right)^2) \tag{13}$$

where T_n = 1 if a image of class n is present. m^+ = 0.9 means that the length of the vector v_n is at least 0.9 to be activated, and m^- = 0.1 means that the length of the vector should be less than 0.1 without the presence of class n images. The λ down-weighting of the loss for absent image classes stops the initial learning from shrinking the lengths of the activity vectors of all the image capsules. We use λ = 0.5. The total loss is simply the sum of the losses of all image capsules.

4 Experiments

4.1 Datesets

We do experiments on both CIFAR [Krizhevsky and Hinton, 2009] and SVHN [Netzer *et al.*, 2011] datasets to evaluate the performance of our network. CIFAR-10 and CIFAR-100 are labeled as a subset of 80 million tiny image datasets. The Street View House Number dataset has 73257 colored digits for training, 26032 digits for testing, with an additional 531131 training images available. We adapt the standard data augmentation: horizontal flipping and shifting to the datasets.

4.2 Implementation Details

Convolutional Transformation Layer. As methioned in 3.1, we use the convolutional capsule layer with 64 group convolution [Krizhevsky *et al.*, 2012] which indicates 64 capsule types, a receptive field of 3×3, a padding of 1 and a stride of 1×1.

Group Reconstruction Routing Algorithm. We use group convolution to implement group capsule. We define three convolutional layers as a function layer and use three funtion layers to learn the f_μ, f_σ, f_v and $g_{\hat{u}}$. Instead directly computing the v by Equation 10, we let α, β and γ as the coefficient to the μ, σ and $\hat{u}$ so that the network can coverage better. Contrary to the function layer, we use three deconvolutional layers each of which has a kernel size of 3×3, a padding of 1 and a stride of 1.

Capsule Max-Pooling Residual Block. We use the method introduced in 3.3 with a kernel of 2×2 and a stride of 2 in each group. After the operation $M + R$ performed by element-wise addition, we use the squash expressed by Equation 3 to normalize the capsules cascaded with ReLu [Nair and Hinton, 2010].

4.3 Results

We perform experiments with the capsule network [Sabour *et al.*, 2017] named CapsNetI as baseline for comparison with the proposed G-CapsNet. In particular, the baseline model is shallow with only two convolutional layers and one fully connected layer. Conv1 has 256, 9×9 convolution kernels with a stride of 1 and ReLU activation. The second layer (PrimaryCapsules) is a convolutional capsule layer. In the third layer (DigitsCapsules) with the matrix W_{ij} the size of 8×16, using routing to cluster the class capsules. In our first experiment which is shown in Table 1, we replace DigitsCapsules Layer with our method. Specifically, dividing the capsules into 16, 32, 64 groups respectively, we obtain three models: CapsuleNet-16, CapsuleNet-32 and CapsuleNet-64. We conduct experiments on CIFAR-10 datasets to evaluate

Model	Number of parameters	CIFAR-10
CapsNetI	8.2M	68.93
G-CapsNet-16	1.5M	69.32
G-CapsNet-32	1.5M	70.45
G-CapsNet-64	1.4M	71.77
CMPR-Block	**1.4M**	72. 87
Block with $\mathcal{L}_{Recon}$	1.6M	**75.62**

Table 1: Top 1 accuracy and number of parameters on CIFAR-10. The best results are highlighted in bold for different network architectures.

Model	CIFAR-10	CIFAR-100	SVHN
ResNet110	93.37	72.78	97.99
VGG16	92.63	70.44	96.04
GoodInit	94.16	72.6	98.21
Batch NIN	93.25	71.14	98.19
Maxout	90.65	61.43	97.53
CapsNetII	87.34	60.44	95.7
Our Method	**94.91**	**76.21**	**98.64**

Table 2: Top 1 accuracy of GRMR-CapsNet compared with state-of-the-arts on CIFAR-10/100 and SVHN .The best results are highlighted in bold for different network architectures.

the proposed G-CapsNets compared with the capsule network [Sabour *et al.*, 2017]. For CapsNetI, the number of parameters is about 8.2M and the accuracy is 68.93%. Compared with the baseline model, we proposed model G-CapsNet-16 which contains 16 groups has the 1.5M numbers of parameters and the accuracy of 70.45% as shown in Table 1. At the same time, when the capsule is divided into 32 and 64 groups, the number of parameters decreases to 1.5M and 1.4M and the accuracy improves to 71.77% and 72.87% respectively. This result indicates that the more the number of groups divided, the denser the capsule relationship within the group, and the more able to distinguish the discrimination of different groups of capsules. Then, we add Capsule Max-Pooling to the G-CapsNet-64, scilicet CMPR-Block proposeed in Section 3.3. The result shows that Capsule Max-Pooling can slightly improve the performance without enduring the increasing computation complexity. Lastly, considering $\mathcal{L}_{Recon}$, we will get the top 1 accuracy of 75.62% which indicates that the reconstruction loss will indeed benefit the routing.

Comparison with State-of-the-arts

In order to demonstrate the superiority of our model, CRMR-CapsNet presented in 3.4 , we also conduct a experiment compared with the state-of-art network: ResNet110 [He *et al.*, 2016], VGG16 [Simonyan and Zisserman, 2014], GoodInit [Mishkin and Matas, 2015], Batch NIN [Chang and Chen, 2015], Maxout [Goodfellow *et al.*, 2013] and CapsNetII. In this paper, our purpose is to improve the performance of Capsule Networks. We just choose the works which are widely used to compare with. Instead the CapsNetI, we use six layers to extract features to build capsules before the PrimaryCapsules Layer. Each layer contains two convolutional layers with the kernel of 2×2, the padding of 1×1 and the stride of 1. Following is Batch Normalization and ReLu. Initial learning rate is set to 0.0001 and maximum epoch is 80. Adam [Kingma and Ba, 2014] is used with momentum 0.9. Batch size [Ioffe and Szegedy, 2015] is 128. We conduct the experiments on CIFAR-10/100 and SVHN. It is clear from Table 2 that the best top 1 accuracy of state-of-the-arts on three datesets are GoodInit's 94.16%, ResNet110's 72.78% and GoodInit's 98.21% respectively. Our proposed method achieves the top 1 accuracy of **94.91%**, **76.21%** and **98.64%**, which is better than theirs on all three datasets. And we have a **7.57%**, **15.77%** and **2.94%** increasing of the basic CapsNet which just uses dynamic routing or EM routing.

5 Conclusion

In this paper, we propose a Group Reconstruction and Max-Pooling Residual Capsule Network (GRMR-CapsNet) that consists of Capsule Max-Pooling Residual Block (CMPR-Block). In each block, all capsules transformed by Convolutional Transformation Layer (CTL) are fed into Group Capsule Layer (GCL) in which the capsules are devided into N groups. To reduce the computational complexity and mapping many more useful information to higher layer capsules, we introduce Reconstruction Feedback Routing that construct three funtion to obtain μ, σ_2 and $\hat{v}$ and define a $\mathcal{L}_{Recon}$ as a feedback to the block. The proposed network takes a image as input and predicts its classification just like the capsule network. The class capsules the length of which represents the probability of the entity can be obtained through Class Caps Layer. We do experiments to prove that the proposed CMPR-Block with $\mathcal{L}_{Recon}$ could greatly improve the performance of capssule network and reduce its computational complexity. Simultaneously, we find different numbers of group can have an effect on the performance. Experiment results on CIFAR-10/100 and SVHN show that our network is competitive compared with state-of-the-arts.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (under Grant 61876142, 61432014, U1605252, 61772402, and 61671339), in part by the National Key Research and Development Program of China under Grant 2016QY01W0200, in part by the National High-Level Talents Special Support Program of China under Grant CS31117200001, in part by the Young Elite Scientists Sponsorship Program by CAST under Grant 2016QNRC001, in part by the Young Talent fund of University Association for Science and Technology in Shaanxi, China, in part by the CCF-Tencent Open Research Fund (No. RAGR20180105) and Tencent AI Lab Rhino-Bird Focused Research Program (No. JR201923), and in part by the Xidian University-Intellifusion Joint Innovation Laboratory of Artificial Intelligence, in part by the Fundamental Research Funds for the Central Universities under Grant JB190117.

References

[Ben-Israel and Greville, 2003] Adi Ben-Israel and Thomas NE Greville. *Generalized inverses: theory and applications*, volume 15. Springer Science & Business Media, 2003.

[Chang and Chen, 2015] Jia-Ren Chang and Yong-Sheng Chen. Batch-normalized maxout network in network. *arXiv preprint arXiv:1511.02583*, 2015.

[Chen and Crandall, 2018] Zhenhua Chen and David Crandall. Generalized capsule networks with trainable routing procedure. *arXiv preprint arXiv:1808.08692*, 2018.

[Cohen and Welling, 2016] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999, 2016.

[Cohen *et al.*, 2018] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

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

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Hinton *et al.*, 2011] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011.

[Hinton *et al.*, 2018] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. 2018.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[LeCun, 1998] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[Lenssen *et al.*, 2018] Jan Eric Lenssen, Matthias Fey, and Pascal Libuschewski. Group equivariant capsule networks. *CoRR*, abs/1806.05086, 2018.

[Li *et al.*, 2018] Hongyang Li, Xiaoyang Guo, Bo Dai, Wanli Ouyang, and Xiaogang Wang. Neural network encapsulation. In *European Conference on Computer Vision*, pages 266–282. Springer, 2018.

[Mishkin and Matas, 2015] Dmytro Mishkin and Jiri Matas. All you need is a good init. *CoRR*, abs/1511.06422, 2015.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[Netzer *et al.*, 2011] 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, page 5, 2011.

[Sabour *et al.*, 2017] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3856–3866. Curran Associates, Inc., 2017.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Xi *et al.*, 2017] Edgar Xi, Selina Bing, and Yang Jin. Capsule network performance on complex data. *arXiv preprint arXiv:1712.03480*, 2017.

[Zhang *et al.*, 2018a] Liheng Zhang, Marzieh Edraki, and Guo-Jun Qi. Cappronet: Deep feature learning via orthogonal projections onto capsule subspaces. *arXiv preprint arXiv:1805.07621*, 2018.

[Zhang *et al.*, 2018b] Xinsong Zhang, Pengshuai Li, Weijia Jia, and Hai Zhao. Multi-labeled relation extraction with attentive capsule network. *arXiv preprint arXiv:1811.04354*, 2018.