IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 5, 2019, accepted June 11, 2019, date of publication June 24, 2019, date of current version July 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2924548

Cv-CapsNet: Complex-Valued Capsule Network

XINMING CHENG[1], JIANGNAN HE[1], JIANBIAO HE[2], AND HONGLEI XU[3]
[1]School of Automation, Central South University, Changsha 410083, China
[2]School of Computing Sciences and Engineering, Central South University, Changsha 410083, China
[3]School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Perth, WA 6945, Australia

Corresponding author: Jianbiao He (jbhe@csu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61873287.

ABSTRACT Capsule network (CapsNet) can recognize the objects by encoding the part–whole relationships in a way similar to our human perceptual system and has already shown its great potential in image classification tasks. However, it is limited to the real domain while the complex numbers having much richer representational capacity and facilitating the noise-robust memory retrieval mechanisms. Therefore, we propose two architectures: Complex-valued Dense CapsNet (Cv-CapsNet) and Complex-valued Diverse CapsNet (Cv-CapsNet++), each of them consists of three stages. In the first stage, multi-scale complex-valued features are obtained by the restricted dense complex-valued subnetwork. Particularly, Cv-CapsNet++ utilizes a three-level Cv-CapsNet hierarchical model to extract the multi-scale high-level complex-valued features in order to adapt to the complicated datasets. In the second stage, these complex-valued features are encoded into the complex-valued primary capsules, Particularly, Cv-CapsNet++ encodes the complex-valued features from different hierarchies into the multi-dimensional complex-valued primary capsules. In the third stage, we generalize the dynamic routing algorithm to the complex-valued domain and employ it to fuse the real- and imaginary-valued information of complex-valued primary capsules. The experimental results show that the proposed architectures lead to fewer trainable parameters, better performance, and fewer iterations during training than Real-valued CapsNets (Rv-CapsNets) with similar structure and original CapsNet on FashionMNIST and CIFAR10 datasets.

INDEX TERMS Capsule network, complex-valued capsule network, CNNs, deep learning.

I. INTRODUCTION

Convolution Neural Networks (CNNs) [1] have extensive learning capacity and can infer the attributes of input images without prior knowledge, which makes them the state-of-the-art architectures in many image classification tasks. However, CNNs have several drawbacks specially related to the sub-sampling layers. Sub-sampling layers often give a small amount of translation invariance but lose the location and pose information, which leads to the fact that their parameters obtained from data training are more inclined to memorize and reproduce features rather than understand them, in particular, ignoring spatial relationships between them which can be valuable for image classification.

To address the drawbacks of CNNs, a novel architecture: CapsNet [2] abandons sub-sampling layers to preserve location and pose information, encodes the features and spatial relationships of features with capsules and transformation matrices and achieves translation equivariance. A CapsNet is more robust to attacks of samples with misled location and pose information than CNNs. However the original CapsNet also has its shortcomings, as it uses a shallow network to extract features, which makes it unsuitable for complicated datasets and its large convolutional kernels also increases its trainable parameters.

Furthermore, inspired by densely connected convolution networks [3], the dense capsule network [4] replaces the first convolutuion layer of original CapsNet with a 8-level dense convolution subnetwok to better adapt to complicated datasets and a 2-level dense convolution subnetwork is shown in Fig.1. The dense convolution subnetwork utilizes smaller convolution kernels and dense connections, which makes it capable of extracting multi-scale features including structure features and semantic features, and greatly decreases the number of trainable parameters and iterations during training. Recently, Trabelsi *et al.* [5] demonstrate that the deep complex-value networks are competitive with real-value networks. Because complex numbers exhibit a richer representational capacity [6], better generalization characteristic [7], and could also facilitate noise-robust memory retrieval

The associate editor coordinating the review of this manuscript and approving it for publication was Hugo Proenca.

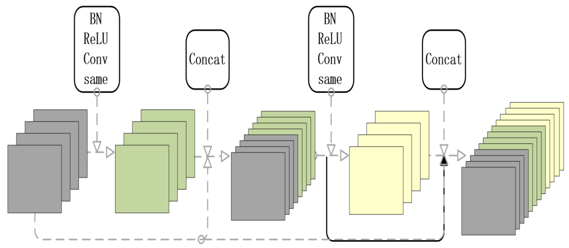


FIGURE 1. Dense convolution subnetwork.

mechanisms [8]. Experiments on image classification have shown that the real-valued networks and complex-valued networks are comparable. Further experiments have revealed that a complex-valued model with a real-valued batch normalization increases the accuracy, while a real-valued model with a complex-valued batch normalization will also increases the accuracy. We will conduct some expanded experiments on our models in this paper.

Inspired by these observations, we propose two architectures Cv-CapsNet and Cv-CapsNet++. In these frameworks, we propose the restricted dense complex-valued convolution subnetwork and complex-valued capsule encoding unit. Firstly the restricted dense complex-valued convolution subnetwork is applied to extract multi-scale features from input images. Particularly, Cv-CapsNet++ utilizes a 3-level Cv-CapsNet hierarchical model to extract multi-scale high-level complex-valued features to adapt to complicated datasets such as CIFAR10. Secondly, these complex-valued features are encoded into the complex-valued primary capsule, while in Cv-CapsNet++ complex-valued features extracted at different hierarchies are encoded into complex-valued primary capsules of different dimensions. Notably, low-level complex-valued features are encoded by high-dimensional complex-valued primary capsules, while high-level complex-valued features are encoded by low-dimensional complex-valued primary capsules. Thirdly, we generalize the dynamic routing algorithm to complex-valued domain and employ it to fuse the real-valued and imaginary-valued information of complex-valued primary capsules. The Cv-CapsNet++ fuses the features of low-level digital capsules and the high-level digital capsules into the final digital capsules to represent information of instantiation and the probability that an entity exists.

In summary, this paper has following contributions: (i) We propose restricted complex-valued dense network and complex-valued capsule encoding unit. (ii) We generalize the dynamic routing algorithm to complex-valued domain and employ it to fuse the real-valued and imaginary-valued information of complex-valued primary capsules, which greatly decreases the number of trainable parameters of complex-valued routing models than real-valued routing models with same dimension capsules. (iii) We propose Cv-CapsNet and Cv-CapsNet++, which leads to fewer trainable parameters, better performance, and fewer iterations during training than Rv-CapsNets with similar structure and original CapsNet on FashionMNIST and CIFAR10 datasets.

II. RELATED WORK

Since the CapsNet [2] was published, its great potential in image classification has received a lot of attention, as the CapsNet can encode relationships between local parts and the whole object with transformation matrices, which enables the CapsNet to understand the whole object through the part-whole relationships. Soon afterwards, Hinton *et al.* [9] propose another matrix capsule network, and this matrix capsule network can encode relationships between the entities and the viewers with transformation matrices, which equips the model with the property of viewpoint equivariance. To adapt it further for high dimension datasets, Xi *et al.* [10] give a lot of advice on improving the CapsNet and explore the effects of a variety of modified models. Experiments demonstrate that stacking more convolution layers and ensemble averaging make significant improvements. Reference [11] formulates the routing strategy of dynamic routing and proposes another routing strategy that works well but is sensitive to a dynamic hyper-parameter which makes the model training very hard. Inspired by inception block [12]–[15], [16]–[18] all utilize inception blocks to modify convolution layers of CapsNet, as the inception block can extract multi-scale information from images. However, all these improvements are limited in real-valued domain.

Moreover, the capsule network has been applied in many fields such as agricultural, transportation, industry and medical diagnosis etc. Li *et al.* [19] adopt the CapsNet to recognize the rice images captured by unmanned aerial vehicle for monitoring the growth of rice and preventing the diseases and pests. Kim *et al.* [20] incorporate the CapsNet for vehicular spatio-tempora characteristics prediction of traffic flow in complex road networks, Paoletti *et al.* [21] develop a CNN model extension that redefines the concept of capsule units to become spectral-spatial units specialized in classifying remotely sensed image data. Zhu *et al* [16] use it to diagnosis bearing fault for rotating machine health monitoring. Xu *et al.* [22] find an effective model based on capsule network to capture more discriminative features and promote gait recognition performance. Wang *et al.* [23] explore a Capsule network for protein post-translational modification site prediction. References [24], [25] successfully introduce the CapsNet to diagnose of lung cancer and brain tumor, which takes advantage of an important property of the CapsNet as it can perform well on small datasets while medical images database are scarce and precious.

III. COMPLEX-VALUED CAPSULE NETWORKS

To design a Cv-CapsNet, we first present restricted dense complex-valued convolution subnetwork, complex-valued capsule encoding unit and complex-valued dynamic routing properties.

A. RESTRICTED DENSE COMPLEX-VALUED CONVOLUTION SUBNETWORK

This paper [5] presents a method to simulate complex-valued convolution with real-valued convolution, If a complex-valued filter matrix is $W = A + iB$, and a complex-valued vector $h = x + iy$ then

$$W * h = (A + iB) * (x + iy) \tag{1}$$

We can use real-valued matrices to present the real and imaginary parts

$$\begin{bmatrix} \Re(W*h) \\ \Im(W*h) \end{bmatrix} = \begin{bmatrix} A & -B \\ B & A \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} \tag{2}$$

Since the real part and imaginary part of output of complex-valued convolution are two separate parts, in the complex-valued dense subnetwork, as the Fig.2 shows, the real part and imaginary part of every complex-valued convolution layer are separated and simulated by two real-valued parts. The real part and imaginary part of every complex-valued convolution layer are separately concatenated in generation order to the real part and imaginary part of the next complex-valued convolution layer in a feed-forward manner, and then add up to make a final complex-valued convolution layer. Only in this way can we guarantee the sustainability of complex-valued convolution operations and the correctness of complex-valued feature encoding, thus we call this subnetwork a restricted dense complex-valued convolution subnetwork. And we employ it to extract multi-scale feature including original feature, structure features and semantic features.

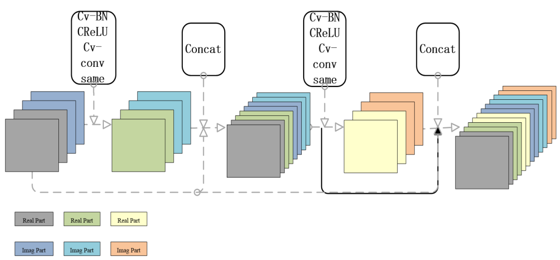


FIGURE 2. Restricted dense complex-valued convolution subnetwork.

Define CReLU as

$$\mathbb{C}ReLU(Z) = ReLU(\Re(Z)) + iReLU(\Im(Z))$$

which outperforms other complex-valued activation [5]. In this paper, we select CReLU as the activation function.

B. COMPLEX-VALUED CAPSULE ENCODING UNIT AND COMPLEX-VALUED DYNAMIC ROUTING SUBNETWORK

In the Cv-CapsNet, a complex-valued capsule is a group of complex-valued neurons and the activity vector of these neurons are denoted as complex-valued vectors u_j

$$u_j = [\Re(u_{j0})+i\Im(u_{j0}), \Re(u_{j1})+i\Im(u_{j1}), \ldots \Re(u_{jn})+i\Im(u_{jn})],$$

in which

$$\Re(u_j) = [\Re(u_{j0}), \ldots \Re(u_{jn})]$$
$$\Im(u_j) = [\Im(u_{j0}), \ldots \Im(u_{jn})]$$

The length of u_j represents the probability that the entity exists, which we actually use a real-valued vector

$$\overrightarrow{u_j} = concat(\Re(u_j), \Im(u_j)) = [\Re(u_j), \Im(u_j)]$$

to simulate for $\left\|\overrightarrow{u_j}\right\| = \left\|u_j\right\|$. The concat() function concatenates the real part and imaginary part.

The complex-valued capsule encoding unit takes the outputs of feature extracting stage as inputs. The outputs of the same complex-valued convolution kernels will be encoded into same component complex-valued capsules. As illustrated in Fig.3, the real part and imaginary part of features are evenly sorted into several components in order. Concatenating one real component and one imaginary component corresponding to the real component forms a component complex-valued primary capsules.

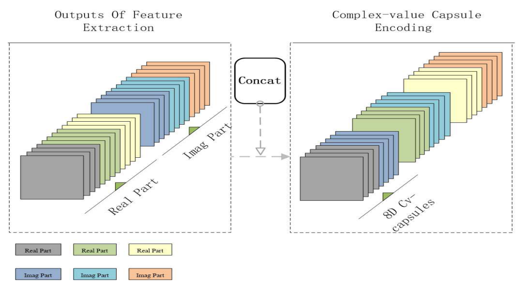


FIGURE 3. Complex-valued Capsule Encoding Unit. Twelve complex-valued features are encoded into three components 8D complex-valued primary capsule.

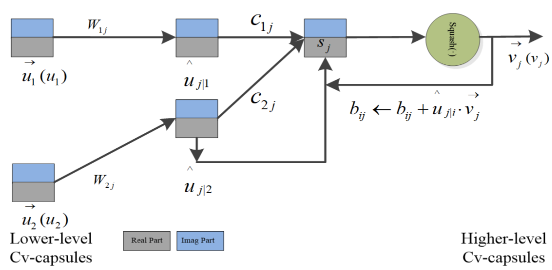


FIGURE 4. In a complex-valued dynamic routing subnetwork, all the complex-valued vectors are simulated by a twice length real-valued vectors.

In the complex-valued dynamic routing, lower-level complex-valued capsules route their information to higher-level complex-valued capsules, that agree the most with their predictions via the mechanism named complex-valued dynamic routing. Fig.4 shows two lower-level complex-valued capsules sending their outputs to the proper higher-level complex-valued capsule, as u_j represents lower-level complex-valued capsules, $\hat{u}_{j|i}$ represents the predictive complex-valued vectors obtained via Eq(3), where the real part and imaginary part from the same complex-valued

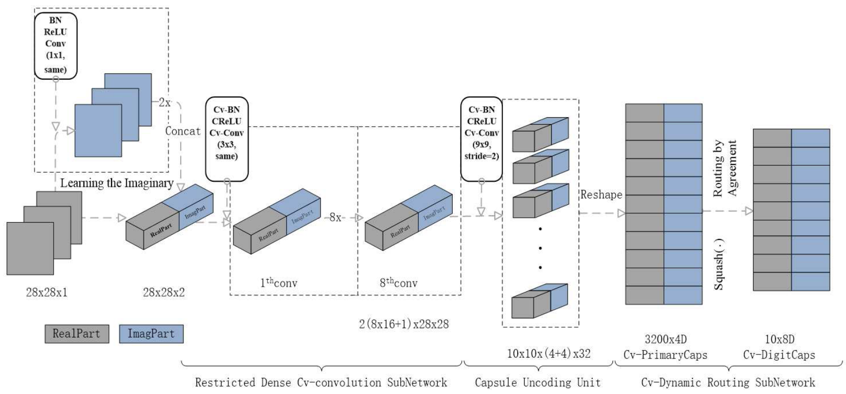


FIGURE 5. **Cv-CapNet for FashionMINST.**

primary capsule are fused by the same transformation matrix W_{ij}, which fuses the real-valued and imaginary-valued information of complex-valued primary capsules, and decreases a half of the number of trainable parameters in complex-valued routing models than real-valued routing models with same dimension capsules. In this way, it speeds up training and reduces computation.

$$\hat{u}_{j|i} = concat(W_{ij} \cdot \Re(u_j), \quad W_{ij} \cdot \Im(u_j)) = W_{ij} \cdot \overrightarrow{u_j} \tag{3}$$

Then two types of predictive complex-valued vectors are weighted by c_{ij} to sum up via Eq(4), the coefficients c_{ij} are iteratively tuned by complex-valued dynamic routing via Eq(5), where the b_{ij} initial value is 0.

$$\overrightarrow{s_j} = \sum_i c_{ij} \cdot \hat{u}_{j|i} \tag{4}$$

$$c_{ij} = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})}, b_{ij} = b_{ij} + \hat{u}_{j|i} \cdot \overrightarrow{v_j} \tag{5}$$

Define $s_j = [\Re(s_{j0}) + i\Im(s_{j0}), \ldots \Re(s_{jn}) + i\Im(s_{jn})]$, in which

$$\Re(s_j) = [\Re(s_{j0}), \ldots \Re(s_{jn})]$$
$$\Im(s_j) = [\Im(s_{j0}), \ldots \Im(s_{jn})]$$

$\overrightarrow{s_j} = [\Re(s_j), \Im(s_j)]$, $\overrightarrow{v_j} = squash(\overrightarrow{s_j})$, $v_j = squash(s_j)$.
Since we have

$$\overrightarrow{v_j} = \frac{\|\overrightarrow{s_j}\|^2}{1 + \|\overrightarrow{s_j}\|^2} \cdot \frac{[\Re(s_j), \Im(s_j)]}{\|\overrightarrow{s_j}\|} \tag{6}$$

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \cdot \frac{[\Re(s_{j0}) + i\Im(s_{j0}), \ldots \Re(s_{jn}) + i\Im(s_{jn})]}{\|s_j\|} \tag{7}$$

$$\|\overrightarrow{s_j}\| = \|s_j\| \tag{8}$$

we can use the real-valued vector $\overrightarrow{s_j}$ to simulate the process of a complex-valued vector s_j passing through the sqaush activation, which prevents the length of output vectors from exceeding 1. After that we shall achieve the final output of parent complex-valued capsules v_j.

C. DESIGN COMPLEX-VALUED CAPSULE NETWORKS

The Cv-CapsNet consists three stages. In first stage, we obtain the complex-valued input by concatenating the real component input and initial imaginary component input, and the initial imaginary components of input can learn by performing the operations presented with a single real-valued residual block [5].

$$BN \rightarrow ReLU \rightarrow Conv \rightarrow BN \rightarrow ReLU \rightarrow Conv$$

And these complex-valued samples will be fed to a restricted dense complex-valued convolution subnetwork, which contains 8 levels of complex-valued convolutions and each of those convolution levels generates 16 complex-valued features, followed by a complex-valued convolution of 9×9 with stride of 2, which results in 128 complex-valued features. In the second stage, the outputs of first stage will be inputs of complex-valued primary capsule encoding unit, and we will get 30 component complex-valued primary capsules with dimensions of 8. In the third stage, the complex-valued dynamic routing fuses the information of real part and imaginary part of complex-valued primary capsules and finally we shall get 10 16D complex-valued digit capsules. Fig.5 shows the detailed pipeline of the proposed architecture for FashionMINST.

We also designed a Cv-CapsNet++, which can extract various scales high-level information from raw images. It is a 3-level Cv-CapsNet hierarchical model, in which a Cv-CapsNet model is created and its intermediate output is used as an input to the second Cv-CapsNet which in turn generates a output fed to the third Cv-CapsNet, There is one complex-valued capsule encoding unit in each Cv-CapsNet, which results in 10 component complex-valued primary capsules. Complex-valued features extracted at different hierarchies are encoded into complex-valued primary capsules of different dimensions, low-level complex-valued features will be encoded by high-dimensional complex-valued primary capsules, high-level complex-valued features will be encoded by low-dimensional complex-valued primary capsules. we will obtain 16D, 12D, 8D complex-valued primary capsules respectively from the low-level to

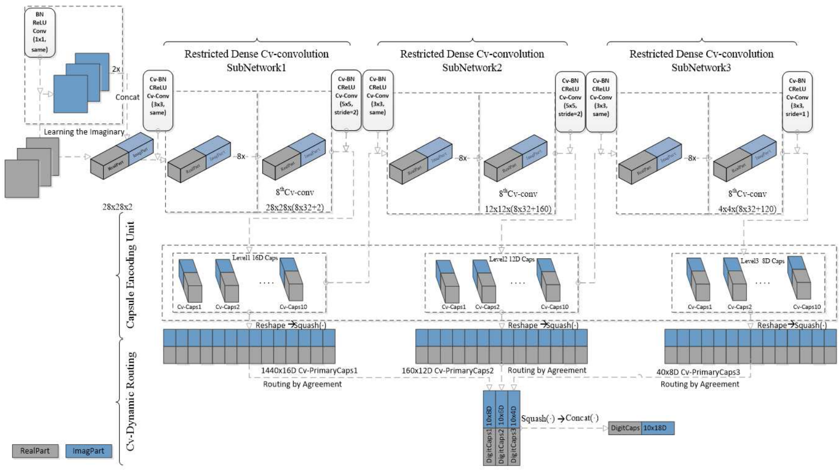


FIGURE 6. Cv-CapsNet++ for FashionMINST.

the high-level. there are two reasons for doing so. The reason one: just like the human perceptual system, the superficial cognition extracted from low-level is more diversified and expressive and the deep cognition extracted from high-level is more simple and discriminative. The second reason is that we will fuse the low-level digital capsules and the high-level digital capsules into the final digital capsules to represent the instantiation the probability that the entity exists, elements of the high-level capsules have a higher weight which means that dimension of high-level capsules should be lower. And we have do some comparison experiments to prove this encoding makes structure more robust. A complex-valued convolution of 5×5 with stride of 2 is applied to previous two Cv-CapsNets to reduce the size of features fed to the next layer, while in the third Cv-CapsNet a complex-valued convolution of 3×3 with stride of 1 is applied. 3 complex-valued primary capsule layers are fused into 8D, 6D, 4D complex-valued digit capsules respectively by complex-valued dynamic routing. Finally, we concatenate these 3 complex-valued digit capsule layers to fuse into 10 18D complex-valued capsules.

In the Cv-CapsNets, the Loss function is defined as:

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

where L_k is loss function for a complex-valued capsule k and $T_k = 1$ if a class k is present and 0 otherwise, Terms m^+, m^-, λ are hyper parameters to be indicated before the training.

IV. EXPERIMENTS

A. DATASETS

To test our proposed approach, we have used the FashionMNIST and CIFAR10 datasets, FashionMNIST includes 70K examples in size of 28x28x1, 60K examples and 10K examples are assigned into the training and testing set, which are associated with a label from 10 classes. CIFAR10 includes 60K examples in size of 28x28x3, 50K examples and 10K examples are assigned into the training and testing set, which are associated with a label from 10 classes.

B. SYSTEM SETUP

We implement the Cv-CapsNet and Cv-CapsNet++ using the Tensorflow and Keras. All the experiments were performed using GeForce GTX1080 TI with 11GB RAM. In all the experiments the mini-batch size is 128, we did not use any data augmentation scheme and repeated the experiment 3 times, the learning rate is 0.001 and decay rate 0.9 with Adam as optimizer. we set different hyper-parameters for training FashionMNIST and CIFAR10: the number of iterations is 25 and 50 in order to quickly converge to optimal solution.

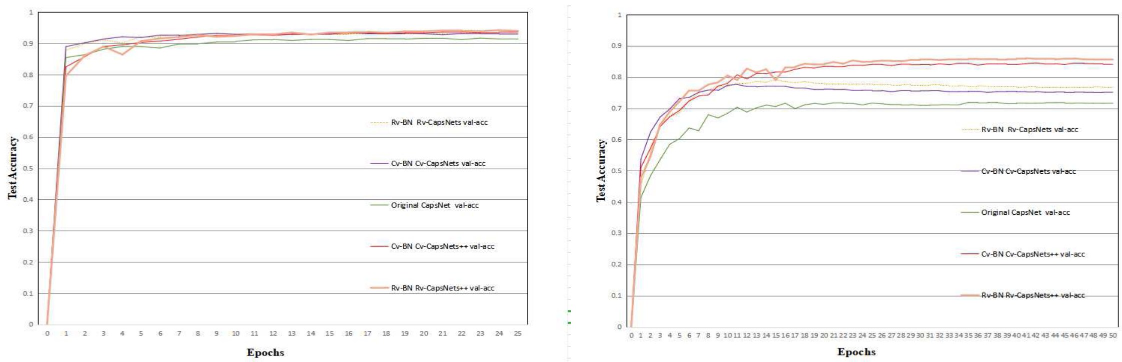
The baseline model is original CapsNet with 30 channels of primary capsules. The architecture we are trying to propose here requires a trade-off between the number of trainable parameters and width (number of convolutional filters in each layer) given a higher accuracy expectation. Specifically, our complex-valued architecture for a complex-valued dense convolution subnetwork starts with 16 complex filters (32 real filters) per convolution layer in the initial stage. In the real-valued counterpart, its dense convolution subnetwork also starts with 16 real filters per convolutional layer. Dimensions of capsules of the Cv-CapsNets are the same as Rv-CapsNets.

C. RESULTS

The accuracy is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

in which, TP means the number of true positive samples, TN means the number of true negative samples, FP means the number of false positive samples, FN means the number of false negative samples, and we use it to evaluate models in this paper. Fig.7 shows the accuracy curves of predictions provided by the Cv-CapsNets and Rv-CapsNets with similar structure and original CapsNet on FashionMNIST and



(a) FashionMNIST (b) CIFAR10

FIGURE 7. The test prediction accuracy curves on FashionMNIST and CIFAR10.

TABLE 1. Comparisons on FashionMNIST and CIFAR10. NR-No Reconstruction SubNetwork, E-Epochs.

	FashionMNIST		CIFAR10	
	# Params	Accuracy	# Params	Accuracy
Original CapsNet NR	6.43M	91.44%25E	7.99M	71.56%50E
Rv-CapsNet NR	6.41M	93.30%25E	8.20M	78.13%50E
Cv-CapsNet NR	3.60M	93.72%25E	4.07M	75.24%50E
Rv-CapsNet++ NR	4.06M	**94.00%25E**	4.83M	**85.64%50E**
Cv-CapsNet++ NR	**2.50M**	93.79%25E	**2.69M**	84.21%50E

CIFAR10 datasets. The FashionMNIST is a relatively simple dataset, as it has been normalized and each sample is a single channel image. This regularization alleviates the complexity of dataset, making it easy to learn. In contrast to FashionMNIST, CIFAR10 is a more complicated dataset, and there are a lot of complicated spatial features and noises. The experiment results show that the Rv-CapsNet and Rv-CapsNet++ performs better than Cv-CapsNet and Cv-CapsNet++ with similar structure and original CapsNet on two datasets and have a greater improvement on CIFAR10 dataset. Meanwhile, the CapsNet++ achieves a faster convergence rate than the CapsNet. The results reveal that the CapsNet++ is more expressive than the CapsNet because of its multi-hierarchy structure, which has rich feature extraction and coding capability.

Table 1 shows the comparison of the best accuracy and the number of trainable parameters. The performance of the Rv-CapsNet is better than that of the original CapsNet on two datasets. The best accuracy of Rv-CapsNet is higher than that of Cv-CapsNet by -0.42% and 2.89% on FashionMNIST and CIFAR10 datasets respectively. The best accuracy of Rv-CapsNet++ is higher than that of Cv-CapsNet++ by 0.21% and 1.43% on FashionMNIST and CIFAR10 datasets respectively. The best accuracy of Cv-CapsNet is higher than that of original CapsNet by 2.28% and 3.68%on FashionMNIST dataset and CIFAR10 datasets respectively. The best accuracy of Cv-CapsNet++ is higher than that of original CapsNet by 2.35% and 12.65% on FashionMNIST datasets and CIFAR10 datasets respectively.

In the Cv-CapsNet and Cv-CapsNet++, lots of small convolution kernels are utilized by means of dense complex-valued convolution subnetwork instead of large convolution kernels, and the modified the Complex-valued Dynamic Routing is applied. All of these help reduce the number of parameters and promote the capacity of the model to extract deep features. The number of parameters of original CapsNet is 1.78 times and 1.93 times of Cv-CapsNet on FashionMNIST dataset and CIFAR10 datasets respectively, the number of parameters of original CapsNet is nearly 2.5 times and 3 times of Cv-CapsNet++ on FashionMNIST dataset and CIFAR10 datasets respectively, and the test performance of original CapsNet is inferior to Cv-CapsNet and Cv-CapsNet++.

Ablation studies were performed in order to compare real-valued batch normalization and complex-valued batch normalization on our models. On FashionMNIST and CIFAR10 datasets, the real-valued representation performs slightly better than its complex counterpart. In general, the obtained results for both representation are quite comparable. We can observe from the Table 2 and Table 3 that replacing a complex-valued batch normalization by a regular one increased the accuracy of the complex-valued convolutional models and replacing a real-valued batch normalization by a complex-valued one increased the accuracy of the real-valued convolutional models, Table 2 and Table 3 outline the comparisons between obtained accuracies of Cv-CapsNets, Rv-CapsNets with similar structure on FashionMNIST and CIFAR10 datasets,

TABLE 2. Comparsions on FashionMNIST. BN-Batch Normalization, NR-No Reconstruction SubNetwork, E-Epochs.

Model	Descriptions	# Params	Accuracy
CapsNet	Original CapsNets NR	6.43M	91.44%25E
Rv-CapsNet	Rv-BN Rv-CapsNet NR	6.41M	93.30%25E
Rv-CapsNet	Cv-BN Rv-CapsNet NR	6.41M	93.67%25E
Cv-CapsNet	Rv-BN Cv-CapsNet NR	**3.60M**	93.52%25E
Cv-CapsNet	Cv-BN Cv-CapsNet NR	**3.60M**	**93.72%25E**
Rv-CapsNet++	Rv-BN Rv-CapsNet++ NR	4.06M	94.00%25E
Rv-CapsNet++	Cv-BN Rv-CapsNet++ NR	4.06M	93.75%25E
Cv-CapsNet++	Rv-BN Cv-CapsNet++ NR	**2.50M**	**94.40%25E**
Cv-CapsNet++	Cv-BN Cv-CapsNet++ NR	**2.50M**	93.79%25E

TABLE 3. Comparsions on CIFAR10. BN-Batch Normalization, NR-No Reconstruction SubNetwork, E-Epochs.

Model	Descriptions	# Params	Accuracy
CapsNet	Original CapsNets NR	7.99M	71.56%50E
Rv-CapsNet	Rv-BN Rv-CapsNet NR	8.20M	**78.13%50E**
Rv-CapsNet	Cv-BN Rv-CapsNet NR	8.20M	76.81%50E
Cv-CapsNet	Rv-BN Cv-CapsNet NR	**4.07M**	77.65%50E
Cv-CapsNet	Cv-BN Cv-CapsNet NR	**4.07M**	75.24%50E
Rv-CapsNet++	Rv-BN Rv-CapsNet++ NR	4.83M	85.64%50E
Rv-CapsNet++	Cv-BN Rv-CapsNet++ NR	4.83M	84.21%50E
Cv-CapsNet++	Rv-BN Cv-CapsNet++ NR	**2.69M**	**86.70%50E**
Cv-CapsNet++	Cv-BN Cv-CapsNet++ NR	**2.69M**	84.21%50E

Table 2 shows the comparison of the best test accuracy and the number of trainable parameters on Fashion-MNIST dataset. In general, the obtained results from both propositions are quite comparable. The performance of the Cv-CapsNet with complex-valued batch normalization is slightly better than that of Rv-CapsNet, while the number of parameters of Rv-CapsNet is 1.78 times of Cv-CapsNet. The best accuracy of Cv-CapsNet++ with real-valued batch normalization is slightly higher than that of Rv-CapsNet++, while the number of parameters of Rv-CapsNet is 1.62 times of Cv-CapsNet++.

Table 3 shows the comparison of the best test accuracy and the number of trainable parameters on CIFAR10 dataset. The performance of the Rv-CapsNet with real-valued batch normalization is better than that of Cv-CapsNet, while the number of parameters of Rv-CapsNet is 2.0 times of Cv-CapsNet. The best accuracy of Cv-CapsNet++ with real-valued batch normalization is higher than that of Rv-CapsNet++ by 1.06%, while the number of parameters of Rv-CapsNet++ is 1.8 times of Cv-CapsNet++.

Based on the results of all these experiments, and taking number of the parameter of the models into consideration, we believe that the complex-valued capsule model is better, Cv-CapsNets outperform Rv-CapNets with same structure and original CapsNet, and Cv-CapsNet++s outperform Rv-CapNet++s with same structure and Cv-CapsNets, particularly, a Cv-CapsNet++ model with real-valued batch normalization has a higher accuracy and fewer iterations during training and fewer trainable parameters.

V. CONCLUSION

In this work, we propose restricted complex-valued dense network and complex-valued capsule encoding unit. We generalize the dynamic routing algorithm to complex-valued domain. We propose Cv-CapsNet and Cv-CapsNet++, both leading to fewer trainable parameters, better performance, fewer iterations during training than Rv-CapsNets with similar structure and original CapsNet on FashionMNIST and CIFAR10 datasets. we have investigated and tested Cv-CapsNets, and demonstrated that the Cv-CapsNets outperforms Rv-CapNets with same structure and original CapsNet. In the future work, we plan to reduce the computational complexity of these current models.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. IEEE NIPS*, Nov. 2012, pp. 1097–1105.

[2] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2017, pp. 3856–3866.

[3] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," Aug. 2016, *arXiv:1608.06993*. [Online]. Available: https://arxiv.org/abs/1608.06993

[4] S. S. R. Phaye, A. Sikka, A. Dhall, and D. Bathula, "Dense and diverse capsule networks: Making the capsules learn better," May 2018, *arXiv:1805.04001*. [Online]. Available: https://arxiv.org/abs/1805.04001

[5] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," May 2017, *arXiv:1705.09792v2*. [Online]. Available: https://arxiv.org/abs/1705.09792v2

[6] M. Arjovsky, A. Shah, and Y. Bengio, "Unitary evolution recurrent neural networks," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, Jun. 2016, pp. 1120–1128.

[7] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 541–551, Apr. 2012.

[8] I. Danihelka, G. Wayne, B. Uria N. Kalchbrenner, and A. Graves, "Associative long short-term memory," Feb. 2016, *arXiv:1602.03032*. [Online]. Available: https://arxiv.org/abs/1602.03032

[9] G. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," presented at the 6th Int. Conf. Learn. Represent., May 2018.

[10] E. Xi, S. Bing, and Y. Jin, "Capsule network performance on complex data," Dec. 2017, *arXiv:1712.03480v1*. [Online]. Available: https://arxiv.org/abs/1712.03480v1

[11] D. Wang and Q. Liu, "An optimization view on dynamic routing between capsules," presented at the 6th Int. Conf. Learn. Represent., May 2018.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)* Jun. 2015, pp. 1–9.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[15] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," Feb. 2016, *arXiv:1602.07261*. [Online]. Available: https://arxiv.org/abs/1602.07261

[16] Z. Zhu, G. Peng, Y. Chen, and H. Gao, "A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis," *Neurocomputing*, vol. 323, pp. 62–75, Jan. 2019.

[17] C. Fang, Y. Shang, and D. Xu, "Improving protein gamma-turn prediction using inception capsule networks," *Sci. Rep.*, vol. 8, no. 1, Oct. 2018, Art. no. 15741.

[18] C. Xiang, L. Zhang, Y. Tang, W. Zou, and C. Xu, "MS-CapsNet: A novel multi-scale capsule network," *IEEE Signal Process. Lett.*, vol. 25, no. 12, pp. 1850–1854, Dec. 2018.

[19] Y. Li, M. Qian, P. Liu, Q. Cai, X. Li, J. Guo, H. Yan, F. Yu, K. Yuan, and J. Yu, "The recognition of rice images by UAV based on capsule network," *Cluster Comput.*, no. 6, pp. 1–10, Mar. 2018.

[20] Y. Kim, P. Wang, Y. Zhu, and L. Mihaylova, "A capsule network for traffic speed prediction in complex road networks," in *Proc. Sensor Data Fusion, Trends, Solutions, Appl. (SDF)*, Oct. 2018, pp. 1–6.

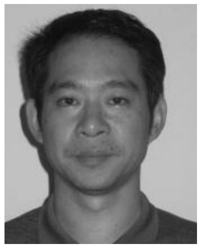
[21] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. Plaza, J. Li, and F. Pla, "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2019.

[22] Z. Xu, W. Lu, Q. Zhang, Y. Yeung, and X. Chen, "Gait recognition based on capsule network," *J. Vis. Commun. Image Represent.*, vol. 59, pp. 159–167, Feb. 2019.

[23] D. Wang, Y. Liang, and D. Xu, "Capsule network for protein post-translational modification site prediction," *Bioinformatics*, to be published. doi: 10.1093/bioinformatics/bty977.

[24] A. Mobiny and H. Van Nguyen, "Fast capsnet for lung cancer screening," Jun. 2018, *arXiv:1806.07416*. [Online]. Available: http://export.arxiv.org/abs/1806.07416

[25] P. Afshar, A. Mohammadi, and K. N. Plataniotis, "Brain tumor type classification via capsule networks," presented at the 25th IEEE Int. Conf. Image Process. (ICIP), Oct. 2018.



XINMING CHENG received the B.S. degree from Zhejiang University, China, in 1987, the M.S. degree from Hunan University, China, in 1992, and the Ph.D. degree from the Huazhong University of Science and Technology, China, in 2003, all in automatic control. He is currently a Vice Professor with the School of Automation, Central South University, China. He has published about 20 papers in international journals, such as *Automatica*, *Systems & Control Letters*, *Nonlinear Dynamics*, *International Journal of Systems Science* (ISA Transactions), and *IET Control Theory & Applications*.

JIANGNAN HE received the B.S. degree in automation from the Hunan University of Technology, Zhuzhou, China, in 2012. He is currently pursuing the M.S. degree in intelligent control and intelligent system with Central South University, Changsha, China. His research interests include computer vision and deep learning.

JIANBIAO HE received the B.S. and M.S degrees from the Huazhong University of Science and Technology, Wuhan, China, in 1986 and 1989, respectively. He is currently an Associate Professor with the School of Computer Science and Engineering, Central South University. His research interests include artificial intelligence, the Internet of Things, pattern recognition, mobile robots, and cloud computing.

HONGLEI XU received the B.Eng., M.Eng., and Ph.D. degrees from the Huazhong University of Science and Technology, China, and Curtin University, Australia, in 1997, 2002, 2005, 2009, respectively. He then joined the School of Electrical Engineering, Computing, and Mathematical Sciences, Curtin University, as a Teaching and Research Staff. He is currently a Senior Lecturer. His research interests include hybrid dynamical systems, optimal control, and blockchain systems. He serves as an Editorial Board Member for several leading international journals, including *Nonlinear Dynamics Systems Theory and Journal of Complexity*. He was a recipient of research grants from the National Natural Science Foundation of China and the Japan Society for the Promotion of Science. He also acted as the Stream Chair or the Program Member at numerous major international conferences. He acts as an Assistant Editor for the *Journal of Dynamics of Continuous*, *Discrete Impulsive Systems*, and *Series B: Application Algorithms* and a Guest Editor for other seven international journals.