An Optimization view on Squash Function of CapsNet

Zhaobin Li*

College of Computer Science & Engineering, Northwest Normal University, Lanzhou, China

*Corresponding author: zhaobinli@nwnu.edu.cn

Abstract. In CapsNet, a bounded measure of the modulus length of the feature is needed, so Squash function is used to compress the feature vector. This paper discusses the definition of Squash function, redefines Squash function based on the idea of information gain rate of decision tree, and constructs CapsNet model on this function. By testing on MNIST, Fashion-MNIST and Cifar-10 datasets, the experimental results show that the Squash function defined in this paper has better classification performance than the traditional Squash function in CapsNet model.

Keywords: Squash function, CapsNet, Capsule.

1. Introduction

Compared with the traditional neural network, the traditional neural network completes the layer-by-layer abstraction from input to output through the layer-by-layer superposition of neurons, thereby completing the output to the final target, and has a better generalization ability, but the capsule network is based on the idea of go forward layer-by-layer with capsules. A capsule is a group neurons, which represents the instantiation parameters of a specific type of an object or an object part. The capsule network model uses the concept of capsule to improve the characterization limitations of CNN and RNN. It uses vector output capsule instead of scalar output feature detector of CNN, and uses dynamic routing instead of maximum pooling, and achieves good results on MINIST data sets.

Since the release of CapsNet in 2017[1], many experts and scholars have carried out research in this area. Such as Dilin Wang[2] proposed a dynamic routing method based on K-means cluster. Sabour[3] proposed matrix capsules with EM routing based on GMM. Rajasegaran[4] proposed a deep capsule network architecture which uses a novel 3D convolution based dynamic routing algorithm. Chang[5] proposed a multi-lane capsule network with strict-squash (MLSCN) to solve the problem with Capsule is that it turns to account for everything in the image. Shi[6] proposed sparse CapsNet with explicit regularizer to solve the problem with CapsNet carries a large number of parameters, leading to the challenge of heavy memory and computational cost.

However, these models are mainly focused on model architecture design, routing design, model application and so on. The research on feature compression has not yet been seen. Therefore, this paper focuses on CapsNet's feature compression(Squash function).

2. An Optimization view on Squash

From the structure of dynamic routing in CapsNet, as shown in Algorithm 1:

Algorithm 1 The Dynamic Routing Procedure[1]

1: **procedure** ROUTING ($\hat{u}_{j|i}, r, l$)**:**
2: for all capsule i in layer l and capsule j in layer (l+1): $b_{j|i} \leftarrow 0$
3: **for** r iterations **do**
4: for all capsule i in layer l: $c_{j|i} \leftarrow \text{softmax}(b_{j|i})$
5: for all capsule j in layer (l+1): $s_j \leftarrow \sum_i c_{j|i}\hat{u}_{j|i}$, $v_j \leftarrow squash(s_j)$
6: for all capsule i in layer l and capsule j in layer $l+1$: $b_{j|i} = b_{j|i} + \langle \hat{u}_{j|i}, v_j \rangle$
7: **end for**
8: **Return** v_j

The Squash function is formula (1).In CapsNet, the significant degree of characteristics is mainly expressed by the length of capsules. The larger the length of capsules, the more significant the characteristics are. However, the feature module length needs to exist as a bounded index, so the feature module length needs to be compressed. The compression formula used in the original paper[1] is as follows:

$$Squash(s_j) = \frac{||s_j||^2}{\mathbf{1}+|| s_j ||^2} \frac{s_j}{|| s_j ||} \tag{1}$$

One of the understandings of this formula is that, the formula changes the length of vector s_j to 1 through $\frac{s_j}{|| s_j ||}$, module length is compressed to 0 ~ 1 by $\frac{||s_j||^2}{\mathbf{1}+|| s_j ||^2}$. That is to say, the main purpose of the compression function is to compress the module length while preserving the difference of features as much as possible. There are many functions that can reflect the feature module length and compress the feature module length between 0 and 1, for example:

$$\mathbf{1} - e^{-||s||} \tag{2}$$

$$\mathbf{tanh}(|| s ||) \tag{3}$$

$$\frac{||s||}{\alpha+|| s ||} \tag{4}$$

$$\frac{||s||^2}{\alpha+|| s ||^2} \tag{5}$$

Among them, α is a constant larger than 0. We will discuss the advantages and disadvantages of these compression functions in the experimental part of this paper. But intuitively, the compression function of formula (1) ~ (5) is only related to the length of vector v. We can also use graphs to express the algorithm more clearly. When the number of iterations r = 3, the dynamic routing of CapsNet is shown in Figure 1.

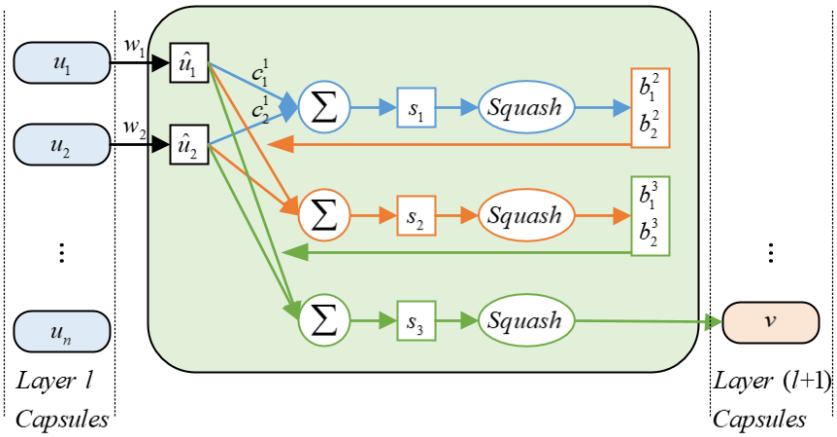


Fig 1. Capsule Dynamic Routing, when iter_routing = 3

As can be seen from Fig.1, the saliency of features is relative to other features and needs to be considered as a whole. However, Squash in traditional algorithms is only related to the size of features, and can not reflect the overall trend of features. As shown in Figure2:

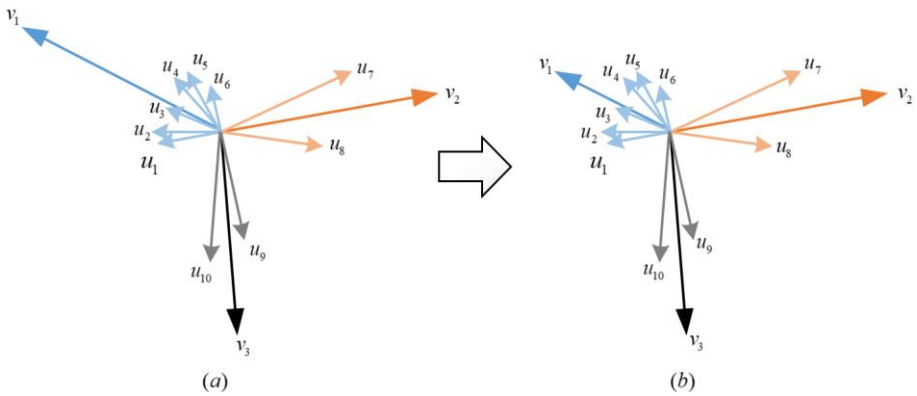


Fig 2. Capsule describes features by clustering eigenvectors

In essence, CapsNet represents the input itself as several eigenvectors, and then clusters them. However, as can be seen from Fig. 2-(a), although the module length of u_1 to u_6 is not large, because there are a large number of classes, the v_1 and v_2 and v_3 obtained after clustering are similar, and it is possible to retain or strengthen v_1 as the main feature. However, for datasets with large amounts of noise data, the class represented by v_1 may not be available. Representativeness, we do not want v_1 to be the main feature and have a larger module length. We only want to embody v_1, but the module length is not too long, similar to Fig.2-(b). Based on the idea of information gain of decision tree, we redesign Squash function as follows:

$$Squash(s_j) = \frac{||s_j||}{\sum_{j=1}^{r} || s_j ||} s_j \tag{6}$$

Dynamic routing also changes, as shown in Fig.3:

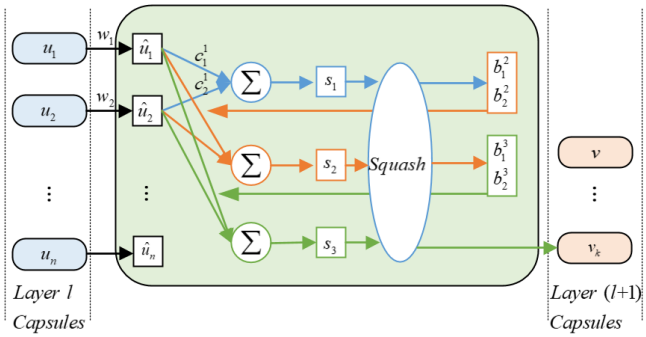


Fig 3. Capsule Dynamic Routing with new Squash function, when iter_routing = 3

3. CapsNet architecture

CapsNet architecture is shown in Fig. 4.

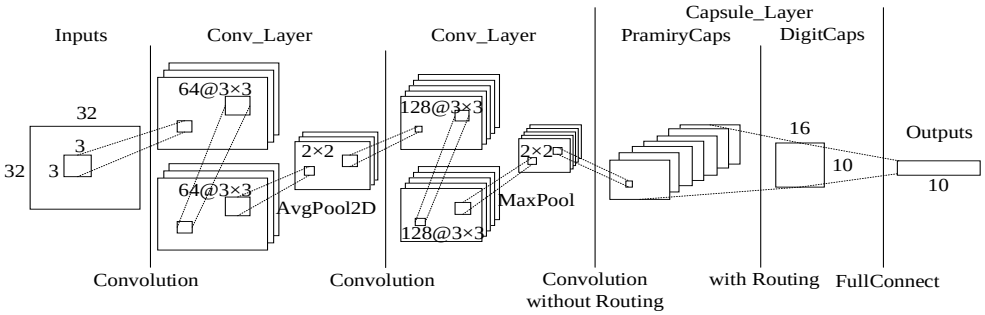


Fig 4. CapsNet architecture on FashionMNIST dataset

4. Experimental Result

Firstly, we experimented on datasets with formulas (1-5). The results are shown in Table 1.

Table 1. Accuracy on test data set of dataset

	Eq. 1	Eq. 2	Eq. 3	Eq. 4 $\alpha=1$	Eq. 5 $\alpha=0.5$
MNIST	0.9832	0.9713	0.9783	0.9801	**0.9956**
FashionMNIST	0.9689	0.9576	0.9403	0.9691	**0.9893**
Carfar10	0.9330	0.9277	0.9310	0.9301	**0.9427**

It can be seen that the results of formulas (1 ~ 5) are not very different, which shows that these formulas are feasible. At the same time, experiments show that for formulas 4 and 5, when $\alpha = \mathbf{0.5}$, the accuracy is the highest, but too large or too small α will make the accuracy decline. We experimented on the model (Fig. 4) on MNIST, Fashion-MNIST and Cifar-10 datasets, epoch = 100. We can see that the Squash function defined in this paper has achieved better results on the test sets of three datasets, which shows that the method is effective.

5. Conclusion

This article defines a new extrusion method. Experiments on multiple datasets using multiple squeezing methods have shown that the proposed method can achieve better classification results than traditional algorithms on CapsNet. How to recognize feature sequences and improve the expression ability of features will be our future work.

Acknowledgement

This work is supported in part by the Key Research and Development Program of Gansu (22YF7GA171).

References

[1] Sabour S, Frosst N, Hinton G E. Dynamic routing between capsules. [C]//Advances in Neural Information Processing Systems. 2017: 3856-3866.

[2] Wang, Dilin, Qiang Liu. An optimization view on dynamic routing between capsules. [C]//ICLR 2018.

[3] Sabour, Sara, Nicholas Frosst, and G. Hinton. Matrix capsules with EM routing. [C]// 6th International Conference on Learning Representations, ICLR. 2018.

[4] Rajasegaran, Jathushan, et al. DeepCaps: Going Deeper with Capsule Networks. [C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.

[5] Siwei Chang, Jin Liu. Multi-Lane Capsule Network for Classifying Images With Complex Background[J]. IEEE ACCESS. 2020.8.

[6] Ruiyang Shi, Lingfeng Niu , Ruizhi Zhou. Sparse CapsNet with explicit regularizer [J]. 2022.124,108486.