Domain Shift in Capsule Networks

Rajath S.*, Sumukh Aithal K.* and S. Natarajan†
Department of Computer Science, PES University, Bangalore, India

Keywords: Capsule Networks, Domain Shift, Convolutional Neural Networks.

Abstract: Capsule Networks are an exciting deep learning architecture which overcomes some of the shortcomings of Convolutional Neural Networks (CNNs). Capsule networks aim to capture spatial relationships between parts of an object and exhibits viewpoint invariance. In practical computer vision, the training data distribution is different from the test distribution and the covariate shift affects the performance of the model. This problem is called Domain Shift. In this paper, we analyze how well capsule networks adapt to new domains by experimenting with multiple routing algorithms and comparing it with CNNs.

1 INTRODUCTION

Collecting and labelling datasets for every new machine learning task and domain is extremely expensive and time-consuming. There will be scenarios where sufficient training data will not be available. Luckily, in this era, there are lots of open-source datasets available for many domains and tasks. However, due to a lot of reasons, there is always a distribution change or domain shift between two domains that can degrade the performance.

Domain shift occurs when the train distribution is different from the actual test distribution. Several works have been proposed to learn indiscriminate features from the source distribution and the target distribution. Thus domain shift is an important problem in practical computer vision. Despite CNNs working very well in the deep learning paradigm, there are a lot of concerns regarding its robustness to shape, rotation, and noise. The idea of capsules and viewpoint invariance properties of Capsule Networks and a complete model was first introduced by Sabour et al. (Sabour et al., 2017). Capsule Networks have recently been applied to many domains, such as Generative Models (Jaiswal et al., 2018), Object Localization (Liu et al., 2018), and Graph Networks (Verma and Zhang, 2018). There have been many unsupervised and supervised methods to route information between layers in Capsule nets. Several routing techniques like EM-routing, Self routing, and Dynamic Routing have been previously proposed.

CNNs are said to learn local features and not the global object shape (Baker et al., 2018). CNNs are also said to be highly texture biased and do not rely on shape as humans do (Geirhos et al., 2018). In contrast, Capsule networks are said to explain global visual processing (Doerig et al., 2019).

In this paper, we analyze the domain shift properties of capsule networks on a few popular routing algorithms and compare it with Convolutional Neural Networks.

2 RELATED WORK

2.1 Capsule Nets

Convolutional Neural Networks do not capture spatial and hierarchical relations between the parts of an object. This problem is addressed by Capsule Networks A capsule consists of multiple neurons that together depict different properties of the same entity. A group of capsules makes a layer in the capsule network. The output of a capsule is a vector that describes various properties of the entity in the image like pose, skew, texture and the length of the vector denotes the probability that the object is in the image. A non-linear process between layers are used in Capsule networks in order to convert activation probabilities and poses of capsules in a lower layer into the activation probabilities and poses of capsules in the higher layer. Due to this structure, Capsule Networks have properties like equivariance and viewpoint invariance unlike traditional CNNs.

*Student, equal contribution
†Professor

S., R., K., S. and Natarajan, S.
Domain Shift in Capsule Networks.
DOI: 10.5220/0010252002750278
In *Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2021)*, pages 275-278
ISBN: 978-989-758-486-2
Copyright © 2021 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

2.2 Formation of Capsules

If Ω_l shows the set of capsules contained in layer l, then for each capsule $i \in \Omega_l$ there is a pose vector u_i and an activation scalar α_i associated with it. Along with this, there exists a weight matrix W_{ij}^{pose} which predicts pose changes for every capsule $j \in \Omega_{l+1}$ as shown in equation 1

$$\hat{u}_{j|i} = W_{ij}^{pose} u_i \tag{1}$$

The pose vector of capsule j is a linear combination of the prediction vectors as shown in equation 2

$$u_j = \sum_i c_{ij} \hat{u}_{j|i} \tag{2}$$

In equation 2, c_{ij} is a routing coefficient which is determined by the routing algorithm used in the capsule network. Hence capsules are formed based on routing algorithms.

2.3 Routing Techniques

The routing algorithm decides how to assign each lower-level capsule to one higher level capsule. These routing techniques are crucial as they enable upper-level capsules to learn higher-level features by combining the features of capsules at the lower layer. Dynamic Routing (Sabour et al., 2017) Self Routing (Hahn et al., 2019) and EM routing (Hinton et al., 2018) are recent and popular routing algorithms used on Capsule Nets.

In Dynamic routing, the pose is represented using a vector, and the length of the vector determines its activation. In contrast, the EM routing technique has a matrix that is used to denote the pose and a separate activation scalar is defined. In the Self Routing method, a vector is used to represent the pose and a separate activation scalar is defined.

EM routing (Hinton et al., 2018) is an unsupervised routing technique where the routing procedure is based on the Expectation Maximization algorithm. In this technique, higher-level features are determined based on the votes from lower level features. The vote of a capsule is calculated by multiplying the pose matrix with a learnable transformation invariant matrix W. The viewpoint invariant transformation matrix is learnt discriminatively and the coefficients are iteratively updated by the EM-Algorithm. The paper (Hinton et al., 2018) also shows a reduction in error rates on datasets suitable for shape recognition tasks when compared to CNNs.

Dynamic Routing is an unsupervised routing technique that was initially introduced by Sabour et al.(Sabour et al., 2017). In this algorithm, an iterative routing-by-agreement technique is used. A capsule in a lower layer is influenced to send its output to capsules in the higher layer whose activity vectors have a big scalar product with the prediction incoming from capsules in the lower layer.

Unlike Dynamic Routing, Self Routing(Hahn et al., 2019) is a supervised routing algorithm, where agreement between capsules is not required. Instead, every capsule is routed independently based on the subordinates in the same layer. Hence, the way activations and poses of higher capsules are obtained is similar to that of Mixture Of Experts.

2.4 Domain Shift

Domain Adaptation aims to minimize the domain shift. Several Deep Domain Adaptation techniques have been proposed (Wang and Deng, 2018) based on the concept of adversarial training. Domain Adversarial Neural Networks (Ganin et al., 2016) aims to achieve domain transfer by learning a domain invariant feature representation. A domain classifier is trained to discriminate whether the feature belongs to the source domain or the target domain. The feature extractor must extract features such that the domain classifier cannot classify whether the sample belongs to the source domain or the target domain. Essentially, the network should not contain discriminative information about the origin of the sample.

Other methods aim to minimize the divergence between the source and target data distribution by using divergence measures like Maximum Mean Discrepancy and Correlation Alignment (Sun et al., 2017). Maximum Mean Discrepancy aims is a divergence measure which compares whether two samples belong to the same distribution by comparing the means of the features after mapping them to a reproducible Kernel Hilbert Space.

3 MOTIVATION

Our hypothesis is that capsule networks will have a smaller domain shift as compared to CNNs. The motivation behind this hypothesis is that since capsule networks claim to capture the spatial relationship between parts of an object (Sabour et al., 2017), the network should be less susceptible to domain shift when compared to CNNs.

4 EXPERIMENTS AND EVALUATION

4.1 Architecture

To compare the domain shift on different datasets, we use a common architecture to train both CIFAR-10 and SVHN. A ResNet-20 block is used for the base CNN architecture. As the ResNet-20 block consists of 19 convolution layers followed by average pooling and fully connected layers, a Capsule Network is built on top of it by replacing the last two layers with a primary capsule and fully-connected capsule layer. In order to have an equal comparison between all routing techniques, we use the same base CNN architecture for all of them.

The main reason for choosing a ResNet-20 block was that it is applied in various architectures and performs very well on most vision-related tasks.

To compare domain shift, we also have a CNN baseline which consists of a standard ResNet-18 block. This CNN baseline network trained on datasets like CIFAR-10 and SVHN.

4.2 Implementational Details

We have used a Stochastic Gradient Descent(SGD) Optimizer to optimize our parameters with an initial learning rate of 0.1, momentum with value 0.9, and a learning rate decay factor of 10^{-4}. Negative log likelihood loss is used while training the models. All models were trained for 100 epochs and the model with the best validation accuracy was chosen for predicting on the test set. The number of capsules per layer is set to 16 and routing is performed once between the primary capsule layer and the fully connected layer in the case of Capsule Networks.

4.3 Datasets

Datasets used to examine domain shift are CIFAR-10 (Krizhevsky et al., 2009), STL-10 (Coates et al., 2011), SVHN (Netzer et al., 2011) and MNIST (LeCun et al., 2010).

MNIST is a database of handwritten digits has a dataset size of 70,000 samples.

MNIST-M is a dataset that is synthetically generated by randomly replacing the foreground and background of MNIST samples with natural images.

The Street View House Numbers dataset (SVHN) (Coates et al., 2011) used, contains around 100,000 digit images procured from Google Street View Images. This real-world dataset has images which are of size 32X32.

The CIFAR-10 (Krizhevsky et al., 2009) dataset contains around 60,000 coloured images belonging to 10 classes which are 32x32 in size.

The STL-10 (Coates et al., 2011) consists of the same 10 classes as that of the CIFAR-10 dataset, but with higher resolution images of size 96X96.

While training on CIFAR-10 and SVHN, we use augmentation techniques like random flip and random crop.

5 RESULTS AND ANALYSIS

We analyze different routing algorithms and their domain shift when trained on two important datasets: CIFAR-10 and SVHN. Capsule Networks with different routing techniques are trained on the source dataset and tested on the target dataset. In all the experiments, the test accuracies on the source and target domains are reported.

In the first experiment, as shown in Table 1 the model is trained on the SVHN dataset and predicted on the MNIST dataset. From this experiment, we show that EM-Routing has minimal domain shift when compared to other routing techniques and CNNs.

In the second experiment as shown in Table 2, we choose the SVHN dataset as our source domain and MNIST-M as the target domain. The domain shift of EM-Routing and CNNs are comparable. Self-routing and Dynamic Routing algorithms underperform in terms of domain shift.

Finally, in the last experiment as depicted in Table 3, CIFAR-10 is trained as the source dataset and its performance is evaluated on the target STL-10 dataset. Domain shift of EM-Routing is on par with CNN. Dynamic and Self-Routing techniques slightly underperform.

We can hence show that EM-routing performs well amongst all routing techniques, and most of the time performing better than CNNs in terms of minimizing domain shift. Dynamic Routing technique and Self-Routing are more susceptible to domain shift depending on the experiments performed.

6 CONCLUSION AND FUTURE WORK

In this paper, we have carried out a comprehensive analysis of Domain Shift in Capsule Networks by considering different routing algorithms. Using a Capsule network with different routing techniques,

Table 1: Source SVHN Target MNIST.

Model	Source Accuracy	Target Accuracy	Domain Shift
Dynamic Routing	95.25	69.79	25.46
EM-Routing	94.3	75.13	19.17
Self-Routing	92.91	60.03	32.88
CNN	96.11	74.01	22.1

Table 2: Source SVHN Target MNIST-M.

Model	Source Accuracy	Target Accuracy	Domain Shift
Dynamic Routing	95.25	47.94	47.31
EM-Routing	94.30	51.31	42.99
Self-Routing	92.91	46.92	45.99
CNN	96.11	53.23	42.88

Table 3: Source CIFAR-10 Target STL10.

Model	Source Accuracy	Target Accuracy	Domain Shift
Dynamic Routing	85.15	30.62	54.53
EM-Routing	82.67	39	43.67
Self-Routing	79.63	38.55	41.08
CNN	91.88	47.06	44.82

we examined how well these models adapt to new domains. These Capsule network models are then compared with a baseline CNN architecture to prove the former's superiority in adapting to new domains. A lower domain shift hence proves the Capsule network's viewpoint invariance and equivariance properties. This can be further enhanced by experimenting on larger different datasets and routing techniques to better understand the Domain Shift in Capsule Networks. Further work can be done to use Capsule networks for domain adaptation and domain generalization.

REFERENCES

Baker, N., Lu, H., Erlikhman, G., and Kellman, P. J. (2018). Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 14(12):e1006613.

Coates, A., Ng, A., and Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223.

Doerig, A., Schmittwilken, L., Sayim, B., Manassi, M., and Herzog, M. H. (2019). Capsule networks but not classic cnns explain global visual processing. *bioRxiv*, page 747394.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. S. (2016). Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2018). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*.

Hahn, T., Pyeon, M., and Kim, G. (2019). Self-routing capsule networks. In *Advances in Neural Information Processing Systems*, pages 7658–7667.

Hinton, G. E., Sabour, S., and Frosst, N. (2018). Matrix capsules with EM routing. In *International conference on learning representations*.

Jaiswal, A., AbdAlmageed, W., Wu, Y., and Natarajan, P. (2018). Capsulegan: Generative adversarial capsule network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

LeCun, Y., Cortes, C., and Burges, C. (2010). MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Liu, W., Barsoum, E., and Owens, J. D. (2018). Object localization with a weakly supervised capsnet. *arXiv preprint arXiv:1805.07706*.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning.

Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866.

Sun, B., Feng, J., and Saenko, K. (2017). *Correlation Alignment for Unsupervised Domain Adaptation*, pages 153–171. Springer International Publishing, Cham.

Verma, S. and Zhang, Z.-L. (2018). Graph capsule convolutional neural networks. *arXiv preprint arXiv:1805.08090*.

Wang, M. and Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135 – 153.