ISSN: 2458-8989

Natural and Engineering Sciences

Supplement, 2019, 4(3): 100-106

Applying Capsule Network on Kannada-MNIST Handwritten Digit Dataset

Emine UÇAR[1], Murat UÇAR[2]

[1,2] Department of Management Information Systems, Iskenderun Technical University, Turkey

emine.ucar@iste.edu.tr, murat.ucar@iste.edu.tr

Abstract

Convolutional neural networks (CNNs) are applied in many different fields such as image processing, natural language processing (NLP) and biomedical. In recent years, a new CNN architecture known as Capsule Network (Capsule-Net) has been developed to reduce some disadvantages of convolutional neural networks and improve the performance. Architecture of capsule networks is inspired by the human brain's inverse graphics and hierarchical mapping concepts. In this study, the architecture and working of the capsule networks were examined and comparative analysis of Capsule Network and CNN on the new handwritten digits dataset called Kannada-MNIST were defined. Our experimental results show that the accuracy of capsule network model is better than the accuracy of CNNs.

Keyword(s): Capsule networks, deep learning, image processing, Kannada-MNIST.

Introduction

The convolutional neural networks have large learning ability and can infer the nature of an input image without any prior knowledge, which makes them an appropriate method for image classification (Krizhevsky et al., 2012). Although CNNs are considered as the pioneer of deep learning, they have some limitations and problems. It is a well-known fact that the convolutional neural networks cannot understand the spatial relationships between parts of an image. And also it needs large amount of data for training. Due to these constraints, the interest of researchers is directed towards capsule networks in recent years.

To overcome above the limitations of CNN, Sabour and Hinton et al. have recently proposed Capsule networks (Sabour et al., 2017). In computer graphics, the image is created by some parameters like width, height and angle. In inverse graphics, these parameters are defined from the image and used for equivariance in capsule. Capsule networks use the dynamic routing algorithm for recognizing the object successfully by the capsules consisting of a group

of neurons. For object recognition, classification and segmentation, a more robust model was obtained by using the capsule structure, dynamic orientation and squash function in the capsule network.

The purpose of the present work is to classify images of Kannada-MNIST data set using capsule network and CNN models. The rest of this paper is arranged as follows: In Section 2, related works are discussed. In Section 3, after presenting the properties of data set, convolutional neural networks and capsule networks are explained in detail. In Section 4, obtained results and observations are evaluated comparatively and conclusion is placed at Section 5.

Related Works

Capsules are a new network architecture in Deep Learning and are producing amazing results by comparison to convolutional neural networks and traditional neural networks. The applications carried out with the capsule network reached by literature review are listed below.

Mandal et al., have implemented capsule network and LeNet and AlexNet architectures on handwritten Indic digits and character datasets. They have also combined capsule networks wtih other networks such as LeNet and AlexNet. Their study has showed that AlexNet with capsule networks were achieved the best performance on most of the dataset (Mandal et al., 2018).

Haque et al., have proposed a model for testing effectiveness of capsule network on Bangla handwritten recognition. Their results show that capsule networks offers acceptable accuracy (Haque et al., 2018).

Nair et al., have used several dataset such as MNIST, fashion-MNIST, SVHN, etc. for comparing CNN and Capsule networks. For comparison they utilized the AlexNet model with the routing capsule network model. As a result they found that capsule networks were able to perform better than AlexNet on more complicated datasets but they were not as good at dealing with deformations (Nair et al., 2018).

Engelin has used Sellpy's dataset to test the rotational views comprehension in capsule networks. This dataset includes images of clothes with a white background and are divided into two parts such as Sellpy Face Forward (SFF) and Sellpy Rotated Objects (SRO). They tested on traditional CNN architecture with different numbers of convolutional layers and observed that the error rate for capsule networks is lesser than that of CNN. As well the results show that capsule networks perform well on SRO images rather than SFF images (Engelin, 2018).

Mehta and Parmar, have used CIFAR10 dataset for comparing the CNN and Capsule network. According to their results accuracy of capsule network is better than CNN but for larger and complex images, it is not that promising like CNN (Mehta and Parmar, 2019).

Mukhometzianov and Carrillo investigated the performance of capsule networks with three well-known classifiers (Fisherfaces, LeNet, and ResNet). They tested the accuracy of classification on four datasets that includes images of faces, traffic signs, and everyday objects. The evaluation results show that CapsNet appears to be a new important image classification technique but requires significant computational resources for simple architectures Mukhometzianov and Carrillo, 2018).

Materials and Methods

In this part of the study description of dataset and methods are presented in detail.

Dataset

In this study we used a new handwritten digits dataset called Kannada-MNIST (Prabhu, 2019). With nearly 60 million speakers worldwide, Kannada is the official language of Karnataka State in India. This dataset contains a training set and a test set that respectively consist of 60000 28x28 gray scale sample images and 10000 sample images equally distributed into the 10 classes. And also contains a more challenging test dataset called Dig-MNIST dataset that consists of 10240 28x28 gray-scale images (Figure 1).

Figure 1 – Sample images of Kannada-MNIST Dataset (Prabhu, 2019).

Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a deep learning algorithm consisting of one or more convolutional layers, subsampling layers and followed by one or more fully connected layers (Lecun et al., 2015). The task of convolutional layers is to define both low level and high level complex features in a given input. Pooling layers are usually used after the convolutional layers and their purpose is to simplify information outputted by the convolutional layer. The last layer of the CNN model is fully connected layer. It is responsible for taking the results of the convolution or pooling layer and using them to classify the image into an output label (Srinivas et al., 2017).

Capsule Neural Networks

Capsule neural network is an enhanced model of classical CNN. Firstly the capsules have been introduced by Hinton et al. (2017) for addressing the limitations of CNN (Sabour et al., 2017). The main weakness of the CNN is mostly related to the pooling layers. Because the pooling layers of convolutional neural network use sub-sampling which loses the precise spatial relationship. If the images are rotated or tilted, CNN will not test such images

properly. Capsule networks can overcome such limitations because they use a dynamic routing algorithm and have a 16-dimensional vector that stores the pose parameters and orientation details (Mehta and Parmar, 2019). Capsules with transformation matrices allow networks to learn part-whole relationships automatically.

A Capsule-Net is structured in several layers too. The lowest layer capsule is called primary capsule and receives a small area of the image as input and tries to determine the presence and pose of particular pattern. Simultaneously the upper layer capsules called routing capsules, trace massive and highly complex objects. With a few convolutional layers, the primary capsule layer is applied, there after the output is remodeled to a vector, where these vectors are given values ranging from 0 to 1 to represent a probability using a squashing function. This generates the output of the primary capsules. An algorithm called routing by agreement is used in the subsequent layers for tracing objects and their pose. This algorithm maintains a routing weight for each connection, when there is an agreement the routing weight increases otherwise it decreases (Sabour et al., 2017).

Results and Discussion

In this study, classification was performed on Kannada-MNIST database images using capsule network and CNN. For the classification, 60000 28x28 gray scale sample images are used as the training data and Dig-MNIST dataset is used as the testing data.

Deep neural network models are created in Python programming language with using Keras library. Keras is an open source library that is strong and easy to use (Gulli and Pal, 2017). For faster training process, TESLA K80 hardware is used via Google Colaboratory free GPU service.

Designed CapsNet

The training setup is 15 epochs with Adam optimizer and batch size 1000. In the first layer of the proposed model, 9x9 convolution is applied. As a result of this process, the tensor of 26 × 26 × 256 is transmitted to the primary capsule layer. The processes performed in the capsule layer and obtained output dimensions are as follows;

- Inputs to the model are 28x28 Kannada-MNIST dataset images.
- Convolution : 9 x 9 x 256
- Reshape : 1152 x 8
- Squash : 1152 x 8
- Capsule layer output (DigitsCaps): 10 x 16

In the output layer of the capsule network model (DigitsCaps), the 16-length capsule for each class is calculated based on the previous layer. The decoder part is composed of three fully connected layers having 512, 1024 and 784 neurons and ReLU, ReLU and sigmoid activation functions respectively. The sigmoid function is used to reconstruct the image. At this stage, α

value which is the loss of reconstruction of the image was selected as 0.005. The capsule network achieved 81.63% on the dig-MNIST dataset.

Table 1 – Classification reports of deep learning models for the dig dataset**.**

	Convolutional Neural			**Capsule Neural Networks**		
Class	**precisio**	**recall**	**f1-**	**precisio**	**recall**	**f1-**
0	0.8360	0.6074	0.7036	0.8133	0.7275	0.7680
1	0.9013	0.7578	0.8233	0.9385	0.7460	0.8313
2	0.6385	0.9434	0.7615	0.7218	0.9531	0.8215
3	0.8787	0.6016	0.7142	0.9248	0.7333	0.8180
4	0.9191	0.7432	0.8218	0.8914	0.7861	0.8354
5	0.7441	0.9541	0.8361	0.7772	0.9404	0.8510
6	0.5511	0.7949	0.6509	0.7279	0.7158	0.7218
7	0.8918	0.5713	0.6964	0.8873	0.7617	0.8197
8	0.7732	0.7725	0.7728	0.8251	0.8847	0.8539
9	0.7936	0.8711	0.8305	0.7672	0.9140	0.8342
Accurac			**0.7617**			**0.8163**

Designed CNN

CNN model consisted of an input layer, five convolution layers, one fully connected layer followed by an output layer. Softmax activation function was used for output layer. A dropout layer was used to prevent overfitting and set to 0.3. Root mean square error was used for measuring the loss error. The Adam optimizer was used for training with learning rate of 10^{-3} and batch size 1000. This pre-trained CNN achieved 76.17% accuracy on the dig-MNIST dataset.

Table 1 and Table 2 provide the per-class classification reports and the confusion matrixes of deep learning models for the dig dataset.

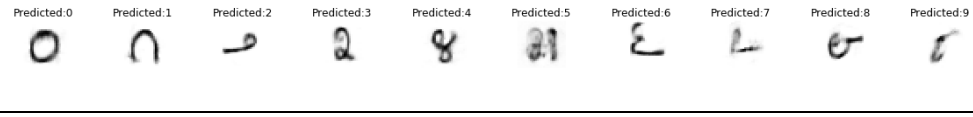
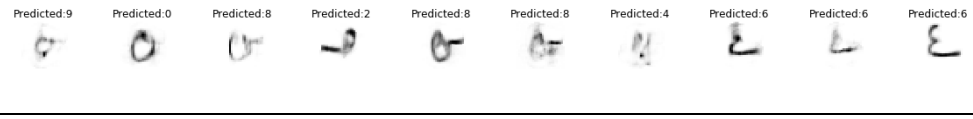
Table 2 – Confusion matrixes of deep learning models for the dig dataset.

	CONVOLUTIONAL NEURAL NETWORKS									
TRUE LABEL	০	62	51	75	36	0	15	92	5	39	89
	১	10	77	66	17	0	39	3	4	8	9
	২	5	1	96	3	2	13	28	0	1	5
	৩	0	1	21	61	4	13	35	17	3	5
	৪	1	6	28	11	76	67	5	1	11	31
	৯	0	8	13	1	7	97	0	1	16	1
	৫	5	1	25	7	49	28	814	31	8	56
	২	3	4	60	7	1	23	318	58	4	19
	৩	5	6	63	3	2	17	119	1	79	17
	৮	1	7	6	0	2	2	63	11	40	89
		০	১	৭	৪	৪	৯	৫	২	৩	৬

	CAPSULE NEURAL NETWORKS									
೦	74	16	66	15	1	10	11	3	24	13
೧	12	76	53	7	11	10	2	26	10	14
೨	7	2	97	12	7	9	2	0	8	1
೩	6	2	11	75	8	10	7	22	4	5
೪	2	7	35	3	80	41	7	4	86	34
೫	0	12	5	2	13	96	0	1	28	0
೬	20	0	31	9	51	72	733	39	6	63
೭	3	2	37	1	1	15	162	78	10	13
೮	5	4	27	10	5	11	33	2	90	21
೯	1	5	9	2	1	2	50	2	16	93
	೦	೧	೨	೩	೪	೫	೬	೭	೮	೯
	PREDICTED LABEL									

Table 3 provides the correctly classified and misclassified examples of each label for the dig dataset.

Table 3 – Correctly classified and misclassified examples of each label.

	**CORRECTLY CLASSIFIED**
	
	**MISCLASSIFIED**
	

Conclusion

In this study we analyzed the performance of convolutional neural networks and capsule network model, which is very new in the literature. For measuring the performance, we used Kannada-MNIST dataset that has never been tried before. Experimental results showed that the overall accuracy of Capsule networks obtained by the combination of the best conditions was found to be 81.63% while the overall accuracy of CNN was found to be 76.17%. As a result of the analysis carried out that Capsule networks achieves better results than CNN. In the future it is aimed to measure the performance of the model on different datasets.

Conflict of Interest: The authors declare that they have no conflict of interest.
Note: This paper is presented in the International Conference on Artificial Intelligence towards Industry 4.0 held on November 14 - 16, 2019 at Iskenderun Technical University, Iskenderun, Turkey.

References

Engelin M., (2018). CapsNet Comprehension of Objects in Different Rotational Views, A comparative study of capsule and convolutional networks, Degree Project in Computer Science and Engineering, Stockholm, Sweden.

Gulli, A. and Pal. S. (2017). "Deep Learning with Keras". Packt Publishing Ltd, Birmingham - Mumbai, pp. 71-105, 2017.

Haque, S., Rabby, A.S., Islam, M.S., & Hossain, S.A. (2018). ShonkhaNet: A Dynamic Routing for Bangla Handwritten Digits Recognition Using Capsule Network. 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), 1-12.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. NIPS.

LeCun Y, Bengio Y, Hinton G. (2015). Deep learning. Nature, 521(7553):436–44.

Mandal, B., Dubey, S., Ghosh, S., Sarkhel, R., & Das, N. (2018). Handwritten Indic Character Recognition using Capsule Networks. 2018 IEEE Applied Signal Processing Conference (ASPCON), 304-308.

Mehta, A. and Parmar, V. D. (2019). A Study on Capsule Networks with the Comparative Analysis of Capsule Networks and CNN, International Journal of Computer Sciences and Engineering, 7(4), 105-108.

Mukhometzianov R. and Carrillo, J. (2018). "CapsNet comparative performance evaluation for image classification," arXiv:1805.11195v1.

Nair, P.Q., Doshi, R., & Keselj, S. (2018). Pushing the Limits of Capsule Networks. Technical note.

Prabhu, V.U. (2019). Kannada-MNIST: A new handwritten digits dataset for the Kannada language. ArXiv, abs/1908.01242.

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In Advances in neural information processing systems (pp. 3856-3866).

Srinivas S, Sarvadevabhatla RK, Mopuri KR, Prabhu N, Kruthiventi SS, Babu RV. (2017). An Introduction to Deep Convolutional Neural Nets for Computer Vision. In: Deep Learning for Medical Image Analysis, 25–52.