SCENE GRAPH GENERATION USING CAPSULE NETWORK

Vijay Tadikamalla (CS17BTECH11040)

Advisor : Vineeth N Balasubramanian

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Introduction

Convolutional Neural Networks (CNNs) have played an essential role in the field of deep learning. Many variants' of CNNs have proven to be very successful in many Computer Vision tasks across different domains. However, there are two main drawbacks to CNN's: their failure to take into account of important spatial hierarchies between features, and their lack of rotational invariance. To address these concerns, Hinton et al. proposed a novel type of neural network using the concept of capsules. With the use of dynamic routing concept and reconstruction regularization, the capsule network model would be both rotations invariant and spatially aware.

These properties of Capsule Networks can prove out to be very useful for tasks like Scene graph generation, where understanding a visual scene goes beyond recognizing individual objects in isolation. However, there is a long way before we can use Capsule Network models for scene graph generation as training a Capsule Network model requires significant computational resources and currently they were just tested on simple and low dimensional datasets like MNIST.

In this work, we attempt to extend the Capsule network model on datasets with higher dimensionality and complexity as image classification is the first step essential step for Scene graph generation.

Salient Features of CapsNet Model

Vector-to-Vector Nonlinearity

The length of the output vector of a capsule represents the probability that the entity represented by the capsule is present in the current input. So, a non-linear **"squashing"** function is used to maintain the relationship between length of vector and probability.

$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1 + ||\mathbf{s}_j||^2} \frac{\mathbf{s}_j}{||\mathbf{s}_j||}$$

Fig. 1: Squash Function

where v_j is the vector output of capsule j and s_j is its total input.

Dynamic Routing

Dynamic routing can be viewed as a parallel attention mechanism that allows each capsule at one level to attend to some active capsules at the level below and to ignore others. This method can be though of as a replacement to backpropogation in classic CNNs.

Model Architecture

The following CapsNet architecture was used for image classification on all the datasets for image classification.

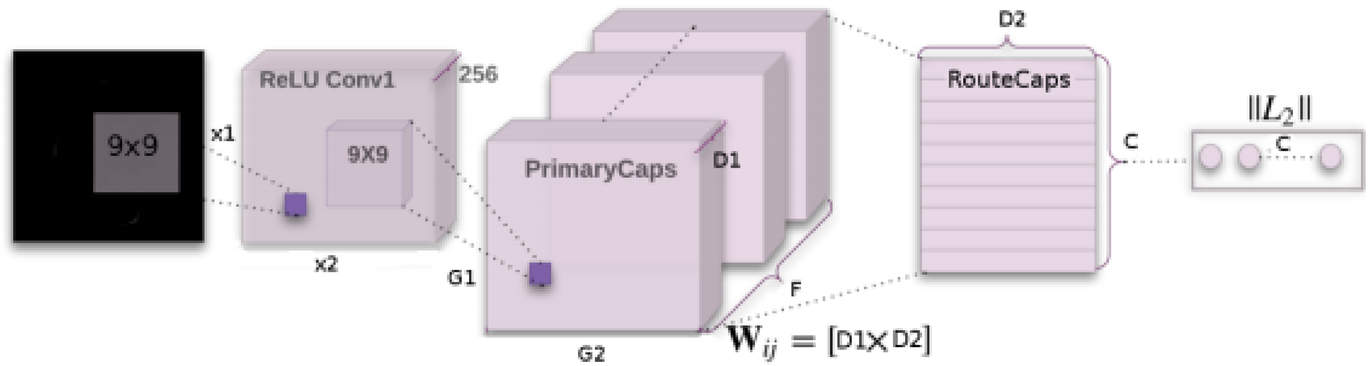

Fig. 2: Modified version of CapsNet architecture (from paper 1)

This first convolutional layer produces 256 feature maps, using a 9 x 9 kernel and valid padding. For the convolution within the primary capsules we also kept a kernel size of 9x9, x_1 and x_2 entirely depend on the size of the input image. G_1 x G_2 are the dimensions of each capsule, and these values are computed automatically based on x_1 and x_2. Other parameters we tuned for different datasets including D_1 and D_2 (the dimensions of the output vectors in primary and routing capsules), F (the number of channels in the primary capsule layer), and C (the number of classes).

Results

After experimenting with different combinations of approaches and methods like

- Stacking more capsule layers.
- Increasing the number of primary capsules
- Increasing number of convolution layers before capsule layer.
- Changing activation function.

We achieved the following Image classification accuracy on different datasets.

Model	MNIST	CIFAR10	CIFAR100
DenseNet	98.6%	98.41%	82.62%
ResNet	99.59%	93.57%	80%
Our results	99.3%	86.40%	26% *
Hinton's [1]	99.75%	89.40%	18% *
DeepCaps [3]	99.72%	91.01%	35.70% *

* tells that the results were reported after 25 epochs.

Future Work

- We will try improve the image classification accuracy on more high dimensional datasets like CIFAR-100 etc.
- We will also try to use Capsule network on more complicated computer vision tasks like object detection, image segmentation and finally scene graph generation.

References

1. Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic Routing Between Capsules." (2017).
2. Hinton, Geoffrey E., Alex Krizhevsky, and Sida D. Wang. "Transforming auto-encoders." (2011).
3. Jathushan Rajasegaran, Vinoj Jayasundara,Sandaru Jayasekara, Hirunima Jayasekara,Suranga Seneviratne, Ranga Rodrigo, "Deep-Caps:Going Deeper with Capsule Net-works" (2019)