Handwritten Digit Recognition using Capsule Networks

Dhanaprakaash Jayabrata Jayateja Joji

Abstract

Capsule networks are an improvement on CNNs. A capsule is a group of neurons whose outputs represent different properties of the same entity. Activity vector of capsule represents the instantiation parameters of a specific type of entity such as an object or object part. Capsnet replace scalar-output feature detectors with vector-output capsules and max-pooling with routing-by-agreement mechanism: A lower-level capsule will send its output to higher level capsules whose activity vectors have a big similarity with prediction coming from lower level capsule. We have implemented capsule network for handwritten digit recognition on MNIST dataset. It has produced testing accuracy of 99.19%. We also created a web interface to see our network in action.

1. Problem Description

Convolutional Neural Networks are now widely used in image recognition. Steps for CNN are as follows:

1. Given an input image, a set of filters scan it and perform convolution operation.

2. This creates a feature map inside the network.These features will next pass via activation(ex. ReLu) and pooling layers. Activation gives nonlinearity. Pooling helps in reducing the training time. Pooling make summaries of each sub-region

3. At the end, it will pass via a sigmoid classifier.

Here training is based on back propagation of error. Pooling is supposed to obtain positional,rotational or orientational invariance. But in reality it removes all sorts of positional invariance. Limitation of CNN: CNN does not take account of orientation of a object in image. In some cases this creates difficulty in recognition(ex: classifying images as faces, where position of eyes, nose, lips can be different so that we can say it is not a real face but CNN will classify it as a face). Capsnet solves the problem of CNN.

2. Literature Review

Capsule Network offers solution for overcoming the disadvantages of CNN. These Solutions are inspired from Human Vision system. Human vision uses saccades that ignores irrelevant details of image by a careful sequence of fixations. Our multilayer visual system creates a parse tree like structure on each fixation. The activation level of a neuron can be interpreted as likelihood of detecting a specific feature. Each node in parse tree will correspond to an active capsule. Each active capsule will choose a capsule in the layer above to be its parent in the tree using an iterative routing-by-agreement mechanism. For each possible parent, the capsule computes a "prediction vector" by multiplying its own output by a weight matrix. If this prediction vector has a large scalar product with the output of a possible parent, there is top-down feedback which increases the coupling coefficient for that parent and decreasing it for other parents. The low level features should be arranged in a certain order for the object for classifying the objects correctly. The order of features are taken into account for classifying objects where in CNN order has no role to play in classification problem. The Capsule Network learns the order during training. The Capsule network not only looks for features but also the relationship between them. The architecture of the network is as follows.

2.1. Architecture

The figure 1 shows the architecture of our capsule network(Sabour et al., 2017).

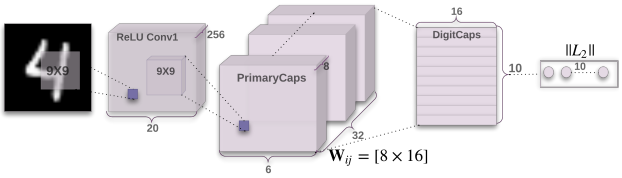

Figure 1. Capsule network architecture for handwritten digit recognition

The first layer is a convolution layer with 256,9X9 kernels with stride of 1 and ReLU activation. Second layer is primary capsule network.Primary capsule layer has 32 channels of convolutional 8D capsules. Third layer is secondary capsule network known as DigitCaps.This layer has one 16D

capsule per digit class. The output from secondary capsule is passed to a decoder which is also a neural network.

Figure 2 shows the architecture of the decoder(Sabour et al., 2017).

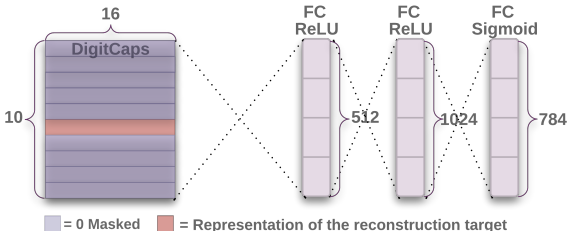

Figure 2. Decoder architecture along with secondary capsule network

We reshape the 1D output vector to produce reconstructed 28x28 image.

We implemented the capsule network for handwritten digit recognition using PyTorch(Paszke et al., 2019).

3. Training

We trained using 60000 images and tested using 10000 images from MNIST dataset with a batch size of 100.

4. Results

4.1. Testing on standard MNIST Database

It gave an accuracy of 99.19% for the architecture given in the paper (Sabour et al., 2017) after running 10 epochs.

We also modified the network by reducing number of output channels of first convolution layer to 4 and trained it for 1 epoch and it produced an accuracy of 96%.

4.2. Reconstructions

Some test images and corresponding reconstructed images are given in figure 3.

(a) Input images

(b) Reconstructed images

Figure 3. Reconstruction of images

4.3. Testing Deployed Model

Figure 4 shows some screenshots of testing deployed model.We have one wrong prediction as shown.

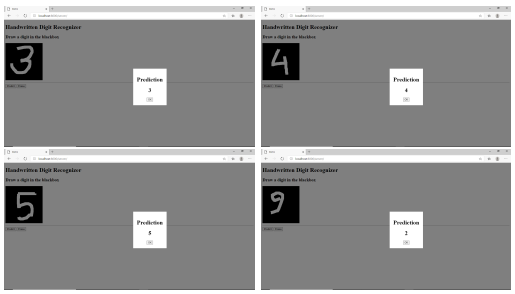

Figure 4. Predictions by deployed model

References

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8026–8037. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Sabour, S., Frosst, N., and Hinton, G. E. Dynamic routing between capsules, 2017.