Analyzing the performance of Capsule Neural Networks (Capsnet)

CSE 676 DEEP LEARNING, FALL 2017
University at Buffalo
Project 2 Report

Aditya Pravin Walke
awalke@buffalo.edu (5020-5384)

Sunil Kunjappan Vasu
sunilkun@buffalo.edu (5020-5673)

Abstract

Recent advances in deep learning and its applications have been applied to many fields.The increase in performance of the machine learning techniques in the past few years has helped in this advancement. One of the recent advances in this field has been Capsule Neural Networks (CapsNets), which contains capsules that the creators believe are similar to human brain modules. These capsules are particularly good at handling different types of visual stimulus and encodings like pose (position, size, orientation), deformation, velocity, albedo, hue, texture, etc. These CapsNets provide state of the art performance on the MNIST dataset. Their performance as compared to Convolutional Neural Networks is observed in this report on different datasets.

1. Introduction

Geoffrey Hinton introduced the idea for a new neural network framework that tries and improves on Convolutional Neural Networks called Capsule Neural Networks (CapsNets). The idea of capsules, which are used in CapsNets have been around for a long time but were limited in their implementation because there was just no technical way to make it work before, one of the reasons for this is that computers were just not powerful enough and another reason is that there was no algorithm that allowed to implement and successfully learn a capsule network. The Capsule Neural Networks tries to overcome the drawbacks of the Convolutional Neural Networks by placing an emphasis on achieving equivariance instead of invariance. Therefore instead of just looking for two eyes, one nose and one mouth to classify an image as face the capsules also handle different types of visual stimulus and encodings like pose (position, size, orientation) , hue, texture, etc. This enables Capsule Networks to give us the ability to take full advantage of the spatial relationship in an image. In the report we test the performance of the CapsNet with the CNN. Baseline models for both the networks were used to ease the requirement of computational resources as well as time constraints.

2. Type of Task and Task Description

The subtask in the project can be divided into the following:

1. Set up a Generative adversarial network (GAN) and train on MNIST dataset. Create a dataset of forged imaged for the MNIST dataset.
2. Set up a Convolutional Neural Network (CNN) trained on the MNIST dataset.
3. Set up a Capsule Neural Network (CapsNet) and trained on the MNIST dataset.
4. Test the performance of the CapsNet as compared to the CNN on different datasets as well as images generated from the GAN.

Generative adversarial network (GAN)
We implemented two types of GANs: mnist_dcgan and mnist_gan. mnist_dcgan consist of a deep generative adversarial network implementation and mnist_gan consist of a fully connected layer implementation.

Convolutional Neural Network (CNN)
We trained a simple Convolutional Neural Network with 2 convolutional layers, trained on the MNIST dataset.

Capsule Neural Network (CapsNet)
We trained a Capsule Neural Network with 2 capsule layers, trained on the MNIST dataset.

3. Datasets

Multiple datasets were used for the training and testing of the different models.

MNIST:
The traditional MNIST dataset was used for the training of all the models (GAN, ConvNet & CapsNet). The MNIST dataset is used as baseline for testing performance in the world of machine learning and suitable for training models using limited computational resources as well as time.

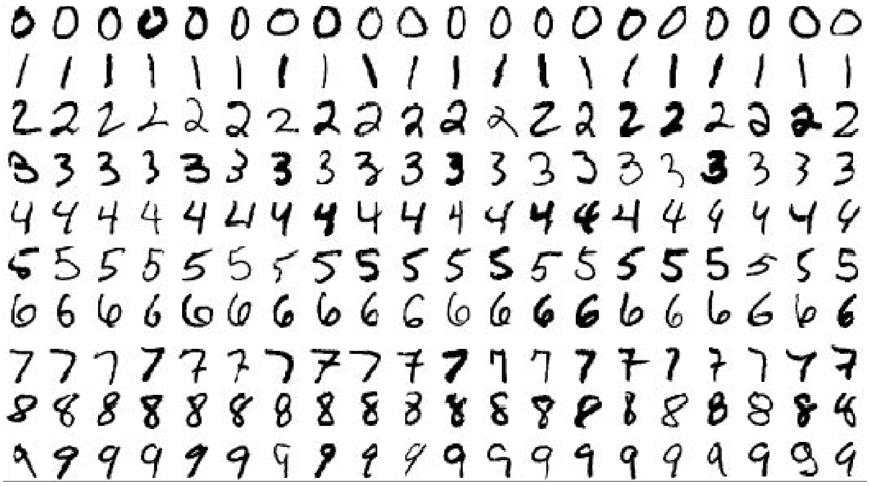


Fig: Original MNIST dataset

Fashion - MNIST:

The original MNIST dataset contains a lot of handwritten digits and is used as a benchmark to validate many algorithms. But there is debate that the MNIST dataset is too easy and is very overused and does not represent modern computer vision tasks well. Therefore the MNIST dataset is used to set the performance of the models to provide additional validation.

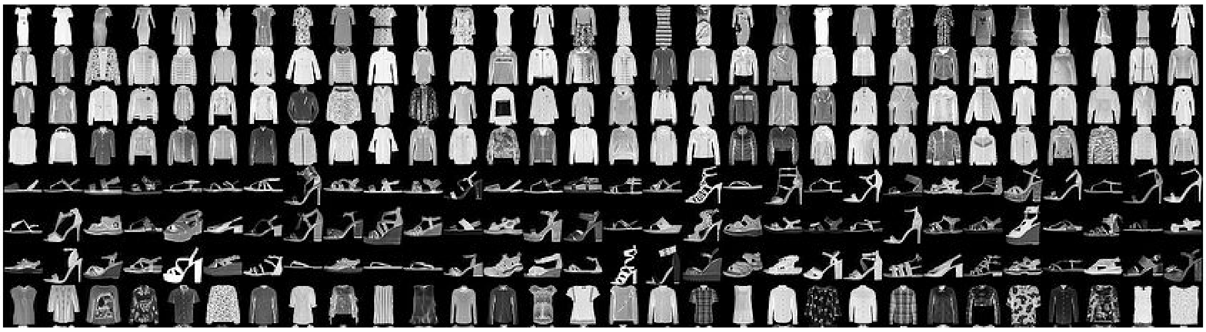


Fig: Fashion-MNIST Dataset

Overlapping Digits:

We generated samples of overlapping digits similar to the MultiMNIST dataset used by creators of CapsNet to test the segmenting of highly overlapping digits using dynamic routing.

Fig: Overlapping Digit Samples (Sample 1: 5 and 0, Sample 2: 4 and 9)
Resized for better viewing.

DCGAN generated images:

We generated images using a DC GAN to test the performance of the CapsNet.

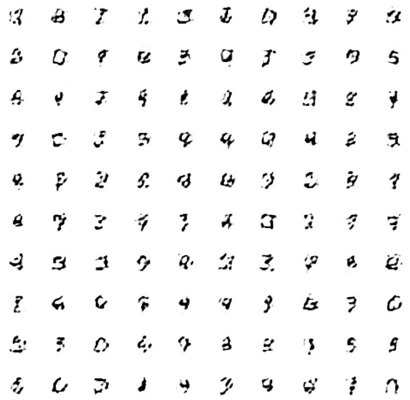


Fig: DC GAN generated Image at epoch-1

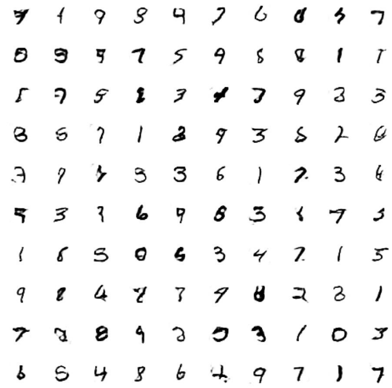


Fig: DC GAN generated Image at epoch-25

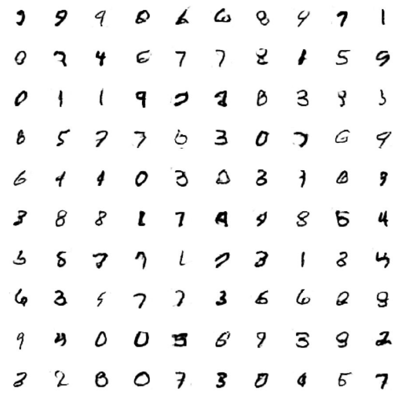


Fig: DC GAN generated Image at epoch-50

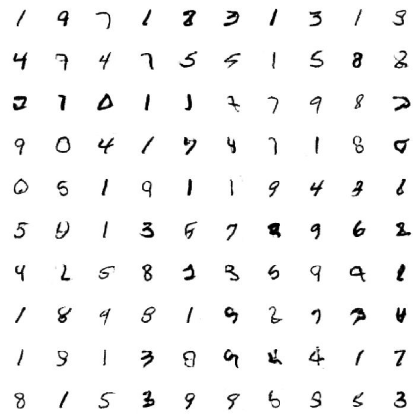


Fig: DC GAN generated Image at epoch-100

Fig: Deep GAN generated Images

4. Machine Learning Models and Methods used

Generative adversarial network (GAN)

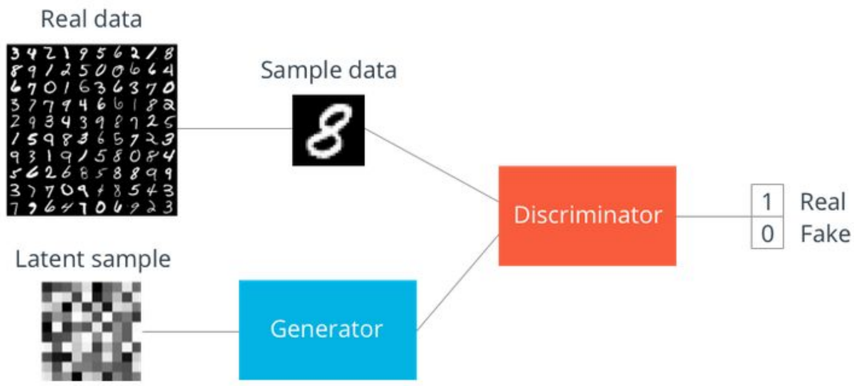


Fig: GAN Implementation

We ran the GAN for 400 epoch and we could see that the Generative loss was reduced in the beginning but the loss start to increase gradually from 50 epoch to 200 epoch and then remains constant. The Discriminative loss remain constant for deep GAN.

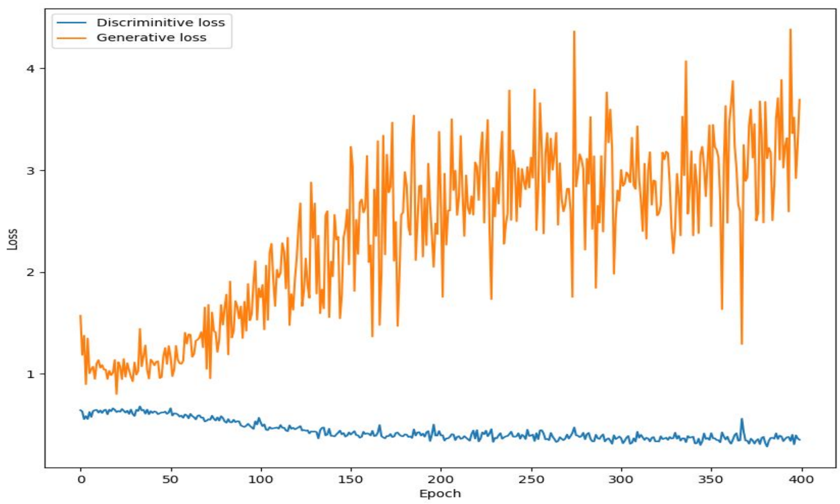


Fig: GAN Discriminative and Generative Loss

However as the quality of the images generated by the GAN even after training for 400 epochs weren't good enough we switched to DCGAN.

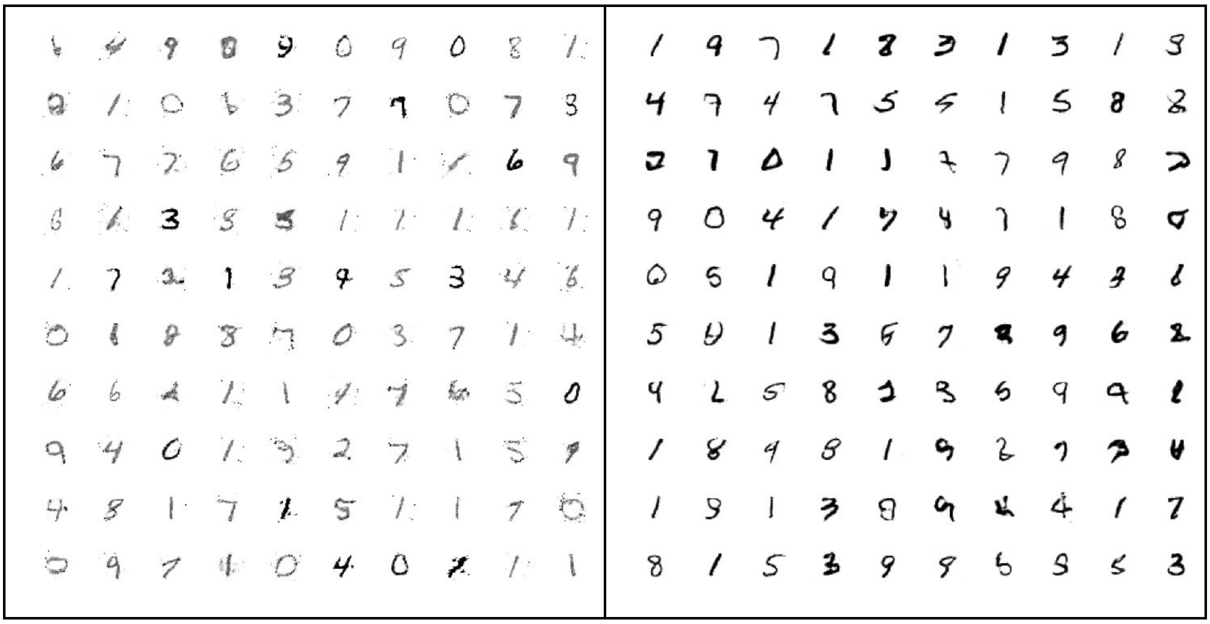


Fig: GAN generated image at epoch-400 vs DCGAN generated image at epoch-100

The Deep Convolutional GAN was run for 100 epochs and we could see that the Generative and Discriminative loss remain constant and there are no much variations.

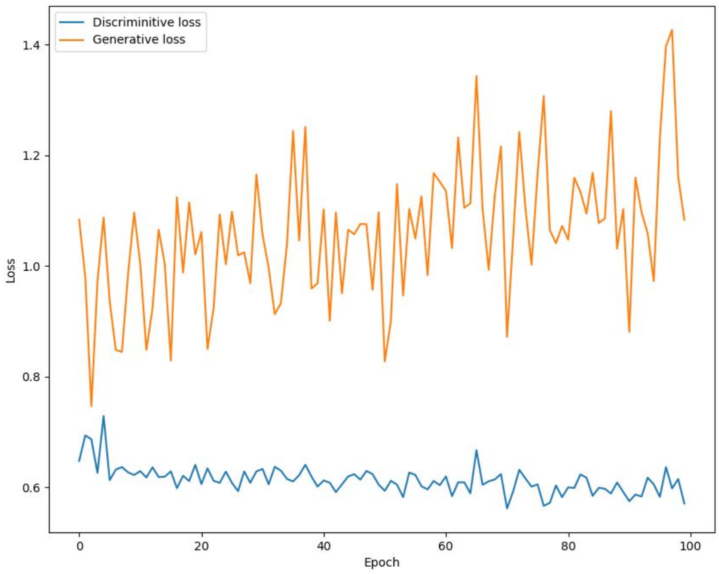


Fig: Deep Convolutional GAN Discriminator and Generative loss

Capsule Neural Network (CapsNet)

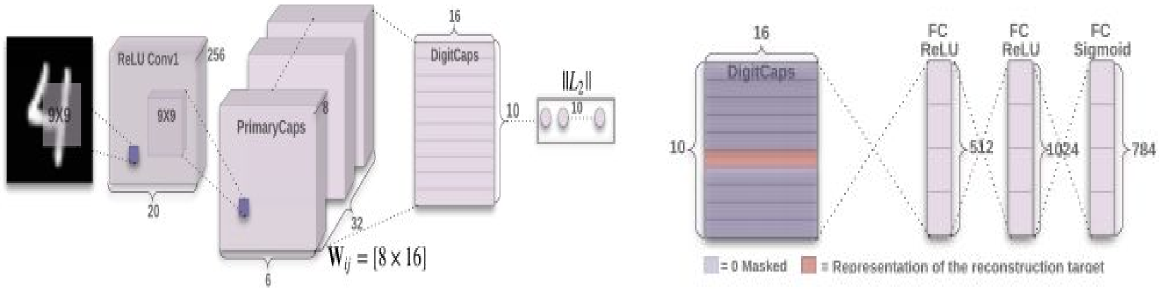


Fig: Architecture of Capsule Neural Network

We trained a Capsule Neural Network with two capsule layers. The primary capsule layer is composed of 32 maps of 6x6 capsules each and each capsule outputs an 8D activation vector. A special squash function was applied to these vectors.

$$\text{squash}(\mathbf{s}) = \frac{||\mathbf{s}||^2}{1 + ||\mathbf{s}||^2} \frac{\mathbf{s}}{||\mathbf{s}||}$$

The next capsule layer that is the digit capsule layer is composed of 10 capsules of 16 dimensions each. Next the routing by agreement was done by initializing all routing logits to zero and an Adam optimizer was used.

5. Performance Comparison

MNIST

The Capsule Neural Network and Convolutional Neural Network were trained on the MNIST dataset and the performance was compared.

	Capsule Neural Network		Convolution Neural Network	
Epoch	**Validation Accuracy**	**Loss**	**Validation Accuracy**	**Loss**
1	**98.7400%**	**0.421774**	**96.6200%**	**0.111225**
2	**99.0800%**	**0.297736**	**97.8000%**	**0.071902**
3	**99.1600%**	**0.244209**	**98.2800%**	**0.056789**
4	**99.3200%**	**0.217923**	**98.4000%**	**0.050923**
5	**99.3400%**	**0.201941**	**98.7800%**	**0.042795**
6	**99.2800%**	**0.189707**	**98.7400%**	**0.042699**

7	**99.3400%**	**0.182829**	**98.8200%**	**0.039232**
8	**99.3800%**	**0.175820**	**99.0000%**	**0.036245**
9	**99.3400%**	**0.172991**	**99.0200%**	**0.033640**
10	**99.3800%**	**0.167206**	**99.0600%**	**0.033676**

Fig: Training comparison of CapsNet and CNN

The Capsule Neural Network achieved a test accuracy of 99.39% whereas Convolutional Neural Network achieved a test accuracy of 99.03%. So there was a marginal increase in accuracy observed in the CapsNet.

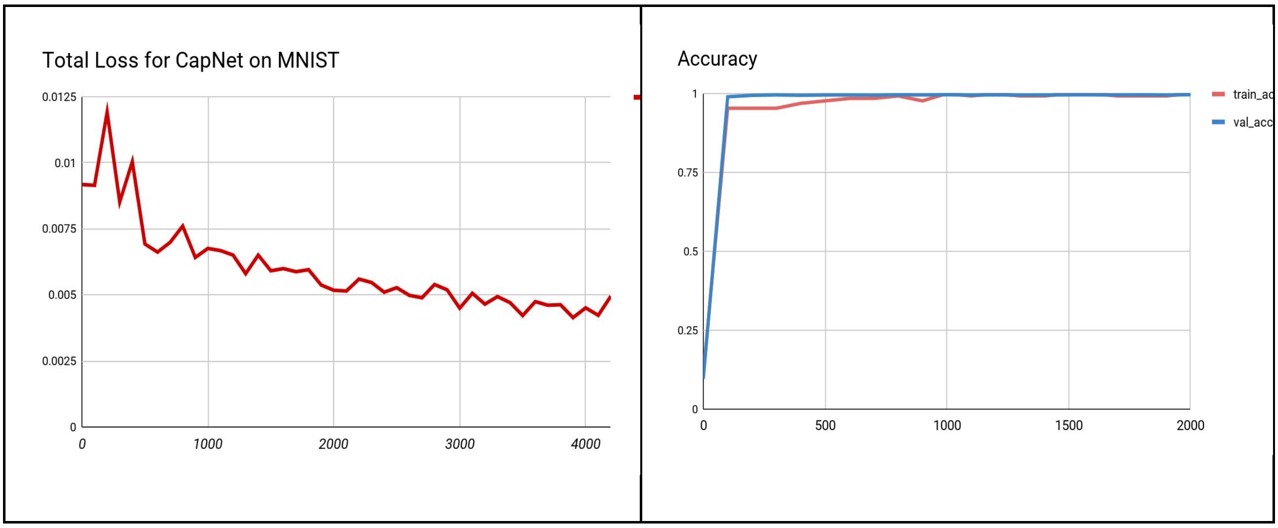


Fig: Loss and Training and Validation Accuracies of CapsNet on MNIST

Fashion-MNIST

The CapsNet and CNN were trained and tested on the Fashion-MNIST dataset, the CapsNet achieved a test accuracy of 90.1% compared to CNN's test accuracy of 89.6%.

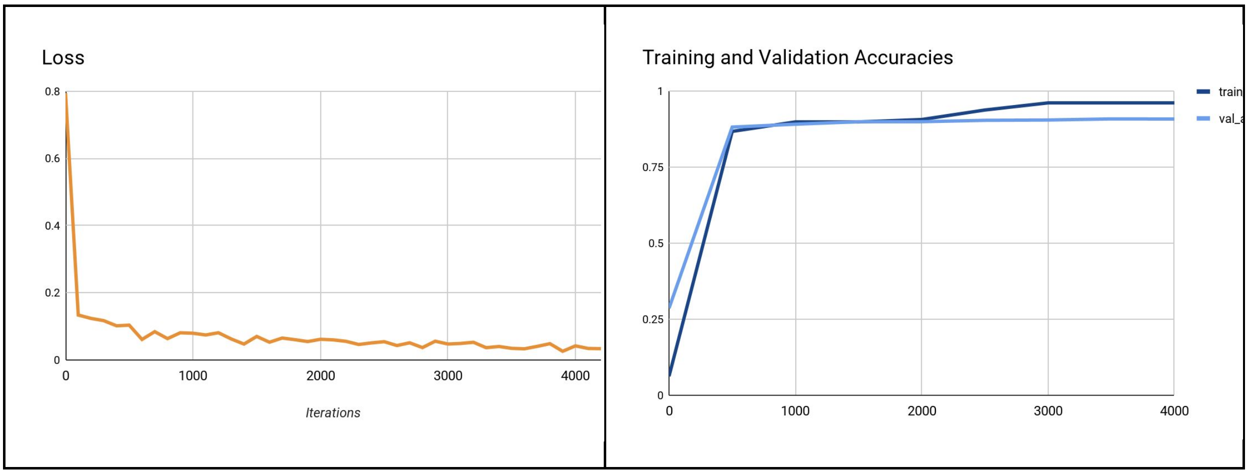
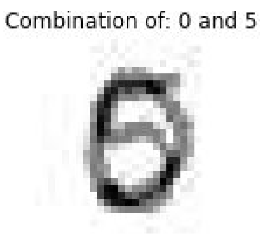
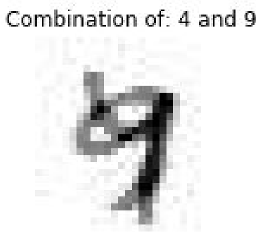
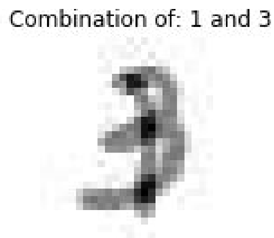
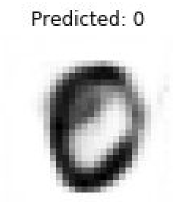
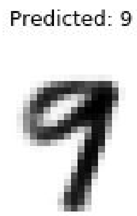


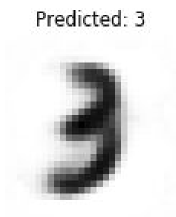
Fig: Loss and Training And Validation Accuracies of CapsNet on Fashion-MNIST

Overlapping Digits

The CapsNet and CNN achieved relatable accuracy on both the MNIST and Fashion-MNIST dataset. The next test was performed to see the performance of both these networks on overlapping digits. The routing by agreement in CapsNet should make it possible to use a prior about shape of objects to help segmentation and it should obviate the need to make higher level segmentation decisions in the domain of pixels. As the MultiMNIST dataset is not made available, we created a sample of overlapping digits and compared the performance of CapsNets and CNNs.

<table>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>

CapsNet Prediction</td>
<td>

CapsNet Prediction</td>
<td>

CapsNet Prediction</td>
</tr>
<tr>
<td>CNN Prediction: 0</td>
<td>CNN Prediction: 9</td>
<td>CNN Prediction: 3</td>
</tr>
</table>

Fig: Prediction of CapsNet and CNN on Overlapped Digits

The CapsNet and CNN both were able to classify the images well into either of the two overlapped digits and were generally in agreement with each other. However the reconstruction shown in the above figure shows that the CapsNet is able to segment the image into the original digit. The fact that it is able to reconstruct the digits regardless of the overlap shows that each digit capsule can pick up the style and position from the votes it is receiving from the primary capsule layers.

GAN Generated Images

It is widely suggested that CNNs are highly susceptible to adversarial attacks and that even small translations in the image pixels can lead to a misclassification of a trained image by the CNN. After the creation of CapsNet, there is a suggestion that they could be used to tackle this problem faced by CNNs better.

MODEL	PERFORMANCE	
Deep Convolutional GAN	With 20% Probability	Classified = 1929 Input = 2000 Performance = 96.45%
	With 50% Probability	Classified = 999 Input = 2000 Performance = 49.95%
GAN	With 20% Probability	Classified = 1146 Input = 2000 Performance = 57.3%
	With 50% Probability	Classified = 131 Input = 2000 Performance = 6.55%

Fig: Performance of CNN on classification of GAN generated images

CLASS	SAMPLE GAN GENERATED IMAGES			
Digit 3				
Digit 4				
Digit 7				
Digit 9				

Fig: Example of Major misclassification of GAN generated imaged by CNN with more than 50% accuracies.

The above figures illustrate the performance of a CNN on GAN generated images. To compare the performance of CapsNet and CNN, we trained both the CapsNet and the CNN as well as the DCGAN on MNIST dataset. We used a small sample of 100 DCGAN generated images 10 from each class to test the performance of both the networks, CNN gave a test accuracy of about 30% whereas CapsNet gave a test accuracy of 37.45%. Although the CapsNet performed better, it is not as big an improvement as was expected.

6. Conclusion and Scope for Improvement

In this project we tried to understand the capabilities of a Capsule Neural Network. We looked at how a CapsNet delivers rotational and other invariances, it does that by being equivariant to the spatial setup of each entity inside an image, and understood the intuition behind them trying to model hierarchical relationships inside of internal knowledge representation of a network. We also observed its performance in comparison to a Convolutional Neural Network, where the CapsNet was able to achieve state of the art performance on a simple dataset. But the obvious drawback being the that current implementation is much slower as compared to the CNN model. Going forward the capsules are definitely able to tackle a lot of the problems and provide some improvement over CNN but the best implementation of capsules is the main question and yet to be explored.

7. References

[1] https://arxiv.org/pdf/1710.09829v1.pdf
[2] https://hackernoon.com/what-is-a-capsnet-or-capsule-network-2bfbe48769cc
[3] https://en.wikipedia.org/wiki/Generative_adversarial_network
[4] https://en.wikipedia.org/wiki/Zero-sum_game
[5] https://hackernoon.com/capsule-networks-are-shaking-up-ai-heres-how-to-use-them-c233a0971952
[6] https://www.youtube.com/watch?v=2Kawrd5szHE&feature=youtu.be
[7] https://www.youtube.com/watch?v=VKoLGnq15RM
[8] https://github.com/zalandoresearch/fashion-mnist