Can C2C-Siamese Networks be used for Outlier Detection and Zero Shot Learning?

Palash Chatterjee
Department of Data Science
Indiana University
Bloomington, IN 47408
palchatt@iu.edu

Pauravi Wagh
Department of Data Science
Indiana University
Bloomington, IN 47408
pwagh@iu.edu

Vibhas Vats
Department of Data Science
Indiana University
Bloomington, IN 47408
vkvats@iu.edu

Vidit Mohaniya
Department of Data Science
Indiana University
Bloomington, IN 47408
vimohan@iu.edu

Abstract

Siamese Networks consists of two identical sub-networks which are joined at their outputs. The sub-networks capture a representation of the inputs while the joining neuron captures the similarity between the two representations. Class-to-Class Siamese Network is an architecture based on these Siamese Networks. It is a difference-based approach in which classification is based on learning patterns of both similarities and differences between classes. A C2C-SN learns patterns from one class C_i to another class C_j. Since, C2C-SN are trained between pairs of classes, they can be used to learn the patterns of a class and might help in performing zero-shot learning. The ability to perform case-by-case classification gives rise to the possibility of using this architecture to perform outlier detection as well. In this work, we explore these two possibilities and study the extent to which C2C-SN can be used to perform outlier detection and zero-shot learning. We show that C2C-SN can be used for outlier detection and we were able to achieve an accuracy of about 71.48% with a confidence threshold value of 0.999 on the MNIST dataset. We also show that C2C-SN fails to perform zero-shot learning with MNIST dataset and with its attributes.

1 Introduction

Class-to-Class Siamese Networks is a novel framework proposed by Xiaomeng et al.[9] to perform classification based on learning patterns of both similarity and difference between two classes. In the original paper, the authors show the ability of such networks to perform classification on a case-by-case basis and for one-shot learning.

The ability to perform case-by-case classification gives rise to the possibility to using this architecture to perform outlier detection. If any of the case-by-case classifiers fail to classify a given test image, then it can be safely termed as an outlier.

Since, C2C-SN are trained between pair of classes, they can be used to learn the patterns of a class and might help in performing zero-shot learning. The representations of the images can be extracted using the inner layers of a C2C-SN and can be substituted in place of the feature representation of image in a zero-shot learning framework.

In this work, we explore these two possibilities. We study the extent to which C2C-SN can be used to perform outlier detection and zero-shot learning. We also present the findings of some additional experiments that we performed in the Appendix. The code for the paper is available at IU Github.[1]

2 Background

2.1 Siamese Networks

The idea of Siamese networks was introduced by Lecun et al. in the early 90s[2]. Siamese Networks [2] consists of two identical sub-networks which are joined at their outputs. The sub-networks capture a representation of the inputs while the joining neuron captures the similarity between the two representations. These networks have same weights attached to them. Using same weights for both networks ensures that two extremely similar images cannot be mapped by their respective networks to a very different locations in feature space as the function computed by each system is the same[4]. As per the original paper by Lecun [2], we make use of the contrastive energy function to measure this similarity and dissimilarity. This energy is high for dissimilar pairs and low for similar pairs.

2.2 C2C Siamese Networks

Class-to-Class Siamese Network is an architecture based on the Siamese Networks. It is a difference-based approach in which classification is based on learning patterns of both similarity and difference between classes. A C2C-SN learns patterns from one class C_i to another class C_j. The network can then be used, given two cases, to identify whether their similarity and difference conform to the learned pattern. It is based on the assumption that there exist consistent similarities and differences in patterns between different classes.

This method uses different approach from a traditional weighting method. Traditional methods focus on finding the pattern of features within a class, but C2C weighting method aims to find the patterns between a pair of classes. After learning these patterns it applies these patterns as an additional information source for classification. In general, the traditional weighing methods assume that similar cases share similar (non)important features. C2C weighting adds another assumption: that cases of different classes differ from each other, with respect to certain features, in a consistent manner.

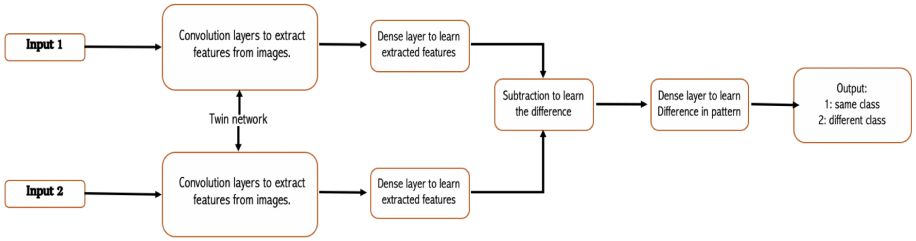


Figure 1: Network representation of C2C Siamese network

The architecture of a C2C-SN is shown in figure(1). The twin network is either CNN or FCN that extract the features from the input image. The outputs from these two networks are subtracted and are input to another FCN which learns to distinguish, if the difference of the two networks conforms to two different classes or slight variations among the same class.

2.3 Zero Shot Learning

Traditionally, classification problems have been solved by training networks on large datasets that are made up of a fixed number of classes. The trained model could classify a given test sample into one of the trained classes only.

There are two problems with this approach. Firstly, it is not always the case that we have sufficient data for training. Secondly, the addition of a new class to the training set would mean that the model

[1] https://github.iu.edu/palchatt/B657-Computer-Vision-Final-Project

would need to be re-trained. Zero-shot learning[3] aims at solving these problems. It tries to predict classes for which no training data is available based on the description of these new classes[5].

The description of a class is also referred to as its *"semantic embedding"*. These embeddings are generally captured using an unsupervised approach[3] in the specific domain, or by using publicly available embeddings, such as word2vec or WordNet [7].

The network takes a test image and its corresponding description to a shared semantic space and tries to perform a nearest neighbour classification to predict its class.

3 Experiments

3.1 Semantic Embedding for MNIST

Zero Shot Learning tries to learn the relationships between the features of an image and its semantic embedding. Since we were using MNIST data for our experiments, for which we could not find any semantic embedding, we had to handcraft them. We designed an attribute set based on the seven-segment display as the representation for each digit.

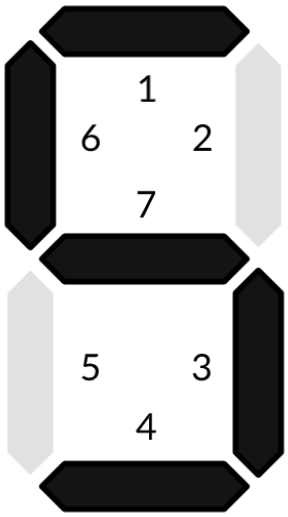


Figure 2: Seven-segment display. The embedding for 5 would be [1,0,1,1,0,1,1]

In order to ensure that the embeddings, we handcrafted, were actually good, we performed Zero Shot Learning on MNIST data using these embeddings. We used a convolutional neural network to extract features from the images and trained a fully connected network to act as the bridge between the feature representations, captured by the convolution network, and the semantic embeddings.

We used images of digits in the range of $0 - 5$ as the training set, along with the attributes we handcrafted and tested using images of digits in the range $6 - 9$. In this setting, we were able to achieve an accuracy of 51.5%.

While this shows that there is indeed a potential for developing better embeddings, this was the best we could achieve on the MNIST dataset.

We tried to introduce various other handcrafted features inspired by the symmetry of the digits, and the shapes that one might use to describe a digit, but didn't get an improvement in performance. We also used GloVe embeddings for the corresponding digits but it performed poorly. A detailed description regarding this can be found in the Appendix A.1.

3.2 Outlier Detection

Class to Class Siamese Networks learns the similarity and difference in patterns between two classes of images. This allows for the potential of using this for performing outlier detection. We created

a training set using labels $0-5$ and trained a C2C-SN per label. Each C2C-SN was trained to distinguish between the classes it was being trained on and any other classes in the training data set.

The testing set for this experiment was a mixture of all the labels from $0-9$. I consisted of images that the algorithm had not seen during the training phase. Every test image was run against all the 6 trained C2C-SN and if all the C2C-SNs agreed with each other that the image doesn't belong to any of the trained classes, then it was marked as an outlier.

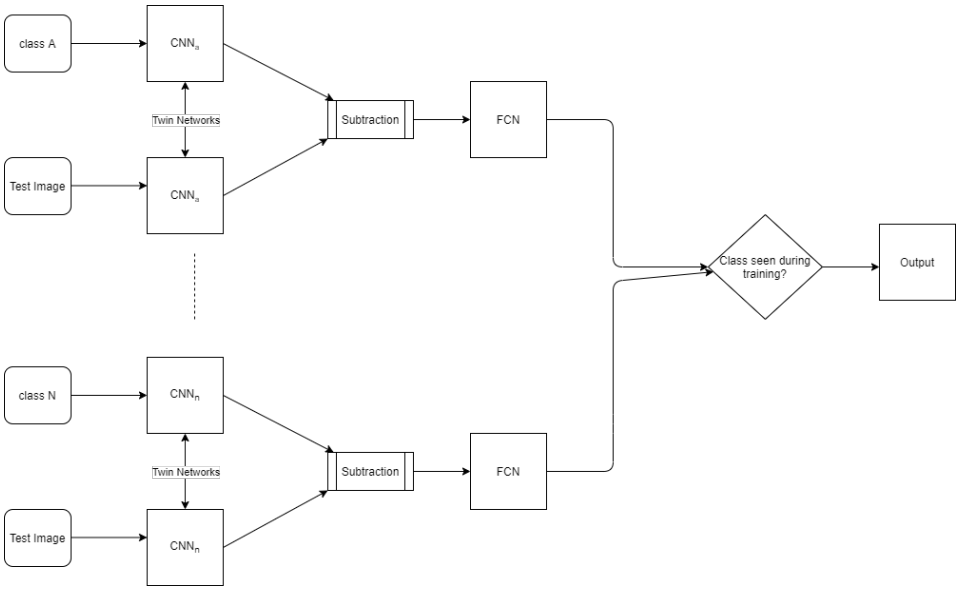


Figure 3: Outlier Detection using C2C-SN

The complete architecture is shown in Fig(3). A key component of this architecture was the confidence threshold, which decides if an image was an outlier or not. If a particular C2C-SN thought that the image being tested belongs to a class it was trained on, then the output from that C2C-SN is a number close to 1.

When deciding whether or not an image is an outlier, we have introduced this threshold which we call the *"confidence threshold"*. A higher value for confidence threshold would mean that a C2C-SN would have to be extremely confident regarding the class of a test image when deciding whether the image is an outlier or not.

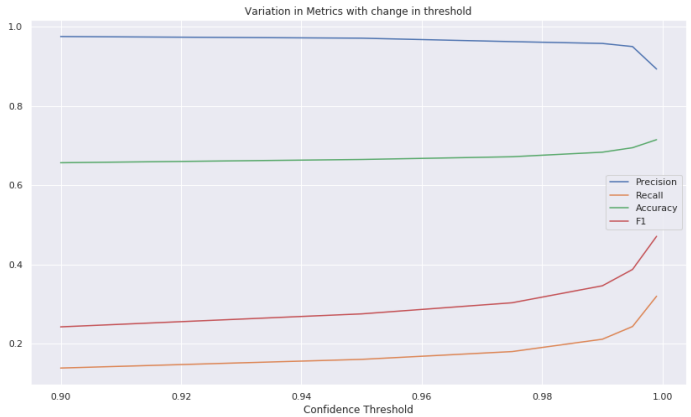


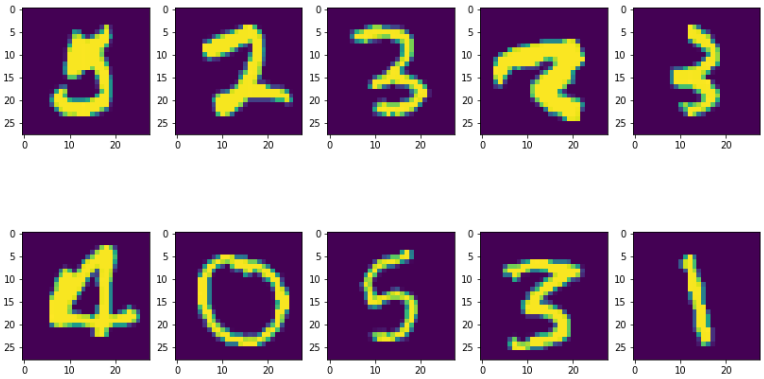
Figure 4: Performance of the model with varying confidence threshold

3.2.1 Results

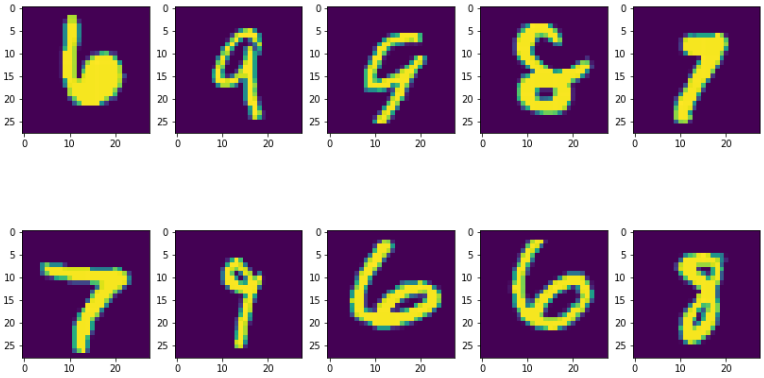
The model was evaluated using precision, recall and accuracy as the metrics. In context of outlier detection, precision would measure the percentage of true outliers among the images marked as outliers by the model. Recall would measure percentage of true outliers that were correctly detected by the model. Accuracy would determine the percentage of images correctly identified as outliers or in-lier by the model.

The performance was measured by varying the confidence threshold. The results are shown in figure(4). We were able to achieve an accuracy of about 71.48% with a confidence threshold value of 0.999.

We present some examples of errors, made by the model in detecting outliers, in figure (5). While some of the images are not clear, the model also fails in some cases when the images are relatively clear to us.



(a) Examples of images which are not outliers but marked as outliers by the model



(b) Examples of images which are outliers but not marked as outliers by the model

Figure 5: Examples of mistakes made by the model

3.3 Zero Shot Learning

Since Class-to-Class Siamese Networks learn the patterns between classes, there seemed to be a potential to perform Zero-Shot Learning.

As we train the twin networks, we needed a way to capture the representations from both the networks. We chose to sum up the representations from the two networks as the representation of the images. However, this would have caused a problem if the images belonged to the different classes. We decided to simplify this by assuming that we know beforehand which images belong to the same class, but we don't know the label of some of these classes.

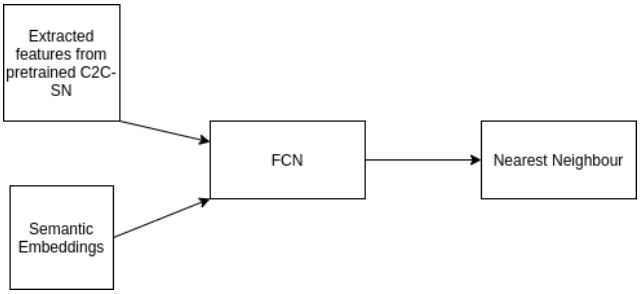


Figure 6: Zero Shot Learning using features extracted from C2C-SN

We thought that this could be achieved in practise by training the C2C-SN on the training classes and by performing one-shot learning using an image which doesn't belong to the training classes. That would help to bucket the unseen images into groups without labels.

We pre-trained one C2C-SN per digit and used their intermediate layers as the representation of image. The digits $0-5$ constituted the training set, while the digits $6-9$ constituted the testing set. The model was trained to learn the connection between representations of images of a given class extracted using the pre-trained C2C-SN models and the semantic embeddings.

While the model performed exceedingly well on classes that it saw during the training, it performed extremely poorly on classes that it had not seen. It was able to predict the correct class with an accuracy of 24.45% and the top-2 classes with an accuracy of 49.42%.

4 Conclusion

We show that C2C-SN can be used for outlier detection. The confidence threshold parameter can be tuned depending on whether the goal is to improve precision or recall. We also note that the although the results show high precision values, the recall remained low.

Additionally, we show that C2C-SN fails to perform zero-shot learning with MNIST dataset and the attributes we created. Further investigation would be required to conclude on the ability of C2C-SN to perform Zero-Shot Learning.

5 Future Work

While performing Zero-Shot Learning using C2C-SN, we note that the validation accuracy is extremely high, which might indicate that in our case, the network is overfitting to the classes it is seeing during training. As we have used a single dense layer to bridge the image representations with the semantic embeddings, it might be overfitting due to the simple embeddings we handcrafted. We can try to perform a similar experiment on a dataset that has a rich set of attributes and see if the results are consistent with our findings.

Acknowledgments

We would like to thank Prof. David Crandall and Xiaomeng Ye for their guidance.

References

[1] Matching loss function for tanh units in a neural net. URL https://stats.stackexchange.com/questions/12754/matching-loss-function-for-tanh-units-in-a-neural-net.

[2] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.

[3] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[4] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. 2015.

[5] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, page 646–651. AAAI Press, 2008. ISBN 9781577353683.

[6] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[7] Jie Song, Chengchao Shen, Yezhou Yang, Yang Liu, and Mingli Song. Transductive unbiased embedding for zero-shot learning. In *Proceedings - 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 1024–1033. IEEE Computer Society, 2018.

[8] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, Sep. 2019. ISSN 1939-3539. doi: 10.1109/TPAMI.2018.2857768.

[9] William Huibregtse Mehmet Dalkilic Xiaomeng Ye, David Leake. Applying class-to-class siamese networks to explain classification with supportive and contrastive cases. 2019.

A Experiments Tried

A.1 Using GloVe embeddings as "semantic" embeddings

Global Vectors for Word Representation [6] provide vector representation of words in various dimensions. We used the 50 dimension embeddings and used the vector representation of the words corresponding to the digits (e.g, one, two, three) as the embedding. While the values might not have meaning semantically, the intention behind using them as embeddings was to get a rich representation of the digits.

Since GloVe values are between the range of -1 and 1, we used hyperbolic tangent as the activation function for the CNN instead of sigmoid and used a corresponding loss function[1].

The experimental setup was same as described in Section 3.1. We trained on classes $0-5$ and tested on classes $6-9$ and were able to achieve an accuracy of 30%.

A.2 Joint training of image and semantic embedding

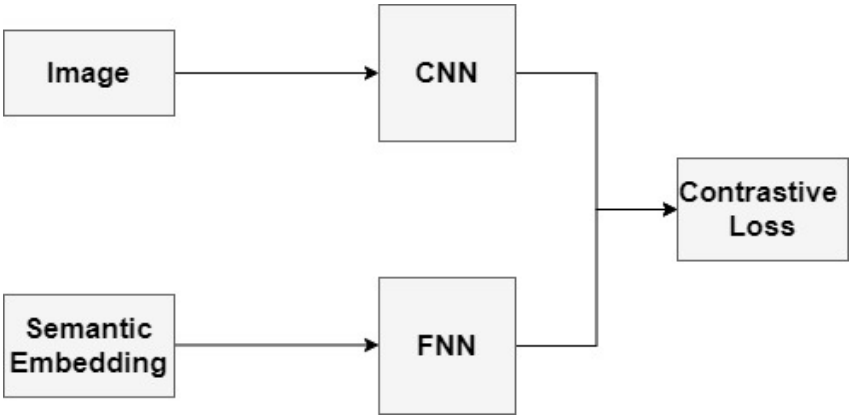


Figure 7: Network for joint training of image and semantic embedding

In Siamese network, the model learns to represent similar images by embeddings that are closer to each other, and dissimilar images by embeddings that are far from each other. This architecture has proven to work well for one-shot learning[4]. Inspired by this, we decided to try to use an architecture as shown in Figure(7) and simultaneously, train images and their semantic embedding to learn a new space in which the image features and their semantic embedding are close to each other.

The CNN was made of two convolution layers and a linear layer and output a $5-D$ vector. The FCN consisted of two fully connected layers, and took as input a $7-D$ semantic embedding which were handcrafted as shown in Figure(2) and returned a $5-D$ vector. The contrastive loss would try to bring the two outputs of CNN and FCN close to each other for similar classes.

While performing zero-shot learning, we pre-extracted the $5-D$ vector from the trained FCN for all the unseen classes. Then, we used the CNN to extract the $5-D$ image features for the test image. Later, we performed 1-nearest neighbor using the $5-D$ image feature on $5-D$ vectors for unseen classes.

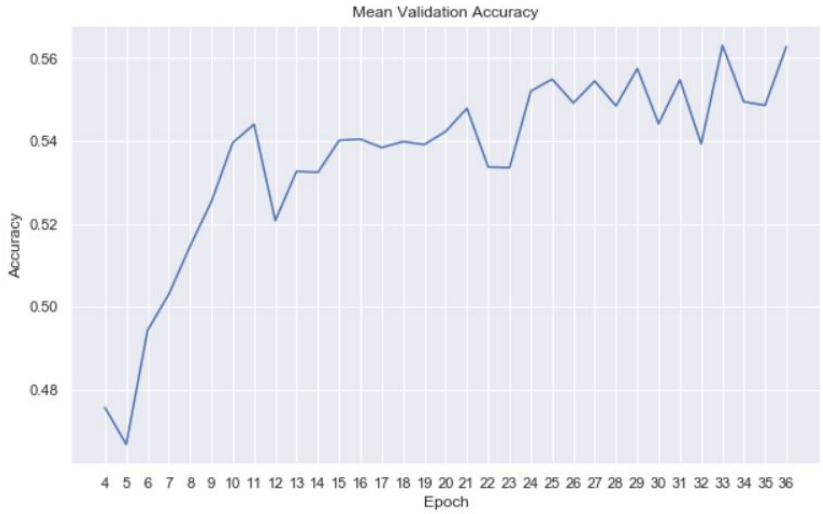
Although we did see some promising results, we discovered that the network was highly unstable as the accuracy of the network kept fluctuating drastically after each epoch. We suspect this is due to the weak $7-D$ representation of semantic embedding.

A.3 Zero Shot Learning on AWA2 Dataset

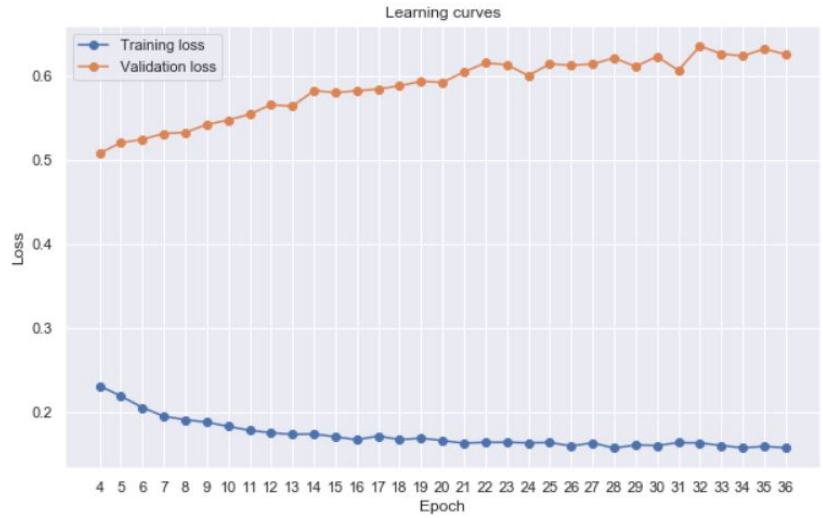
We had initially decided to use the AWA2 dataset[8] for our project. Before approaching the problem of performing Zero-Shot Learning using Class-to-Class Siamese Networks, we tried to perform Zero-Shot Learning in the traditional way to have a baseline of the performance. AWA2 dataset consists of 37322 images of 50 animals classes with pre-extracted feature representations for each image. Each

Figure 8: Semantic Embedding representation in AWA2[8]

class is also represented with a $85 - D$ semantic embedding vector. We used a pre-trained ResNet-34 model to act as our base layer and added a linear layer to reduce the output to the required dimensions. We trained the network on 40 classes and evaluated the zero shot accuracy over the unseen 10 classes. Although we saw promising results with an accuracy of around 60% as shown in Figure(9a), it took too long to train given the resources we had. The model took around 2 hours per epoch and since ours was a research project, we needed a quicker feedback cycle to test out our ideas. Hence, we decided to use the MNIST dataset for our project.



(a) Accuracy of Zero Shot Learning on AWA2 dataset



(b) Training and Validation Losses of Zero Shot Learning on AWA2 dataset

Figure 9: Results for Zero Shot Learning on AWA2 Dataset