Reflection: CSC630 Masters Independent Study

Garima Garima

January 2020 - April 2020

1 Introduction

Humans learn to rely on information differently based on the source of information, called the "Social Provenance" of information. These days we rely on Deep Networks [6] for taking critical decisions and Deep Networks rely on data from numerous sources. It is very important to identify the outliers and reliable source of information. Not only the source of information but there could be many other challenges with data like mislabeled, legally restricted, privacy issues and other. In this study we explored different network mechanisms to account for social cues when estimating the reliability of conclusions from deep networks. Like humans, neural networks should be able to train on information differently based on the source of the information, known as "Social Provenance in Deep Networks."

There exists various mechanism to identify outliers before using that data for training the network known as "Noise elimination". We will identify the technique in which a network should be able to identify outliers itself and their source to exclude them from training data. A mechanism of robust network handling overfitting is known as "Noise Tolerance". First and foremost, we need to understand the different types of deep neural networks and their architecture. An in depth understanding of how network train and predict the conclusions is required. A network can be trained by creating batches from huge amount of data. To find out how each of the data point is used by network is quite a challenging task. Once network identifies a data point to be an outlier, it should be able to trace back to the source of that data point. Finding the source is another challenging task that we need to explore to solve this problem.

In this study we will be focusing on particular type of outliers, that will be mislabeled data. We divided this independent study program in two parts. The first part has two goals: learning tools to create and test deep networks, and articulating a formal version of the research problem. The second part will focus on designing new models and creating testable hypothesis. We initiated this research with understanding of different network architectures using The MNIST database for handwritten digits by Yann-et-al [9]. This dataset consists

of 70,000 images of handwritten digits with their labels. We will manually and intentionally mislabel some of the images from this dataset use them as outliers to train the model. After that we will create a mechanism for network to identify these mislabeled images and trace their source.

2 Summary

In first part of the study, we identified the tools and dataset to be used. Initially we thought to go with Jupyter Notebooks and PyTorch because it has dynamic graphs feature and would give us more flexibility to explore different network architectures. We did a lot of hands-on to understand the basic concepts of Neural Network and their architecture like input, hidden and output layers, number of neurons in one layer, optimizers, loss functions, activation function, dropout, hyperparameter tuning, image preprocessing. Understood working of Neural Network like training, testing, backpropagation, gradient descent, overfitting, underfitting, graphical representation of the model training process with Visdom library [5] of PyTorch for dynamic graphing of the model. We also try to improve our knowledge and make meetings engaging by discussing various Artificial Intelligence podcasts by Lex Fridman [1] and watching videos by YouTube channel "Two Minutes Paper" [4] to see the real life application of artificial intelligence, Machine Learning algorithms. We spent some time in understanding the effect of number of layers and number of neurons in layers on accuracy of the model. We developed an artificial neural network using the MNIST digit dataset and PyTorch with 99.8% test accuracy.

During this learning process we observed there was not enough official documentation of PyTorch and because of that it was really tedious and time consuming to solve some basic errors and implement the model in PyTorch. We decided to explore TensorFlow and Keras libraries of Python. While going through TensorFlow documentation we came across Google Colab [2] where we could use free GPU services provided by Google for fast processing and also we do not have to install of TensorFlow on our local systems. Learning TensorFlow after a basic understanding of creating neural networks was an easy process. We were able to accomplish all the work done in TensorFlow and Google Colab.

After learning the basics of neural networks we started to focus on our problem statement. Instead of three different sources of data we used one source MNIST (60,000 train dataset and 10,000 test dataset) for digit recognition data and divided it in three equal parts A, B and C with 20,000 images each. To create outliers we mislabelled all the images in part C by incrementing the value of label by 1. So, 1 becomes 2 and 2 becomes 3 and so on. After that we merged all there parts again into one part and shuffled the data to train the model. We want to create a mechanism that network should identify the mislabelled images and trace it back to part C as their source. We have five models at the end. First model trained with no mislabeled data, second model in which 20,000 images are mislabeled, third model trained with part A dataset, fourth with part

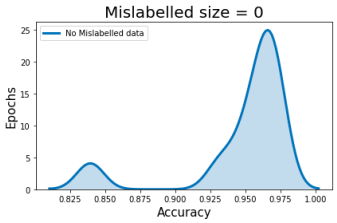

Figure 1: No Mislabelled Data

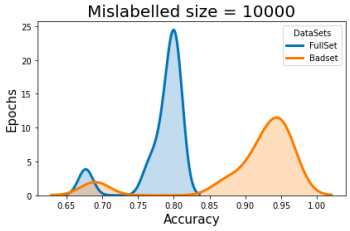

Figure 2: 10,000 Mislabelled Data

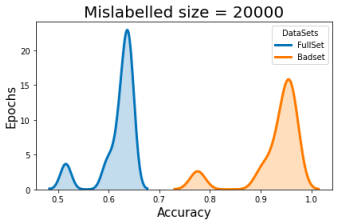

Figure 3: 20,000 Mislabelled Data

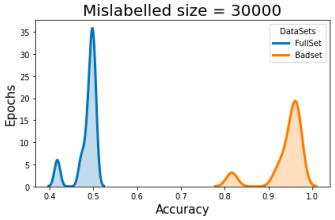

Figure 4: 30,000 Mislabelled Data

B dataset and fifth with part C. We tried to observe training and testing loss and accuracy of all the five models to compare and find any relation between them. All the models behaved independently as we assumed but we were not able to find any significant relation between them.At this point we also changed our dataset from MNIST digit recognition to FMNIST which is Fashion MNIST dataset [10], having 10 different classes representing 10 different types of fashion items. Then we also try to see if the amount of mislabelled data will give us some insights. We had a model with no mislabelled data and then we trained various models with 10,000, 20,000, 30,000 and 40,000 mislabelled data and plotted their training history as shown in Figure 1, 2, 3, 4, 5 and test history as to compare their accuracy with each other as shown in figure 6 and find any correlation between them.

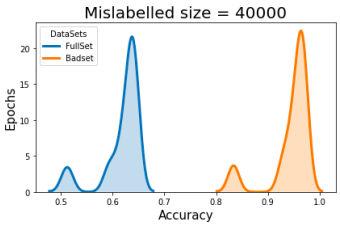

Figure 5: 20,000 Mislabelled Data

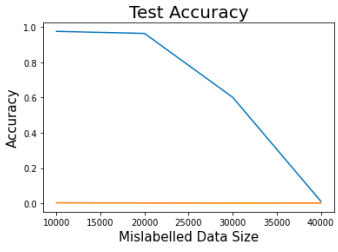

Figure 6: Test Accuracy

From Artificial Neural Network (ANN) we understood that they do not retain any semantic information of the image and process it but Convolutional Neural Network (CNN) tries to retain the semantic information of the image. So, we moved towards exploring CNN and find a way that helps us to achieve our goal. We also changed the dataset to Cifar10 [7] because it has complex structures and lots of tiny details and it would work best on CNN. In CNN learning process we came across new concepts of convolutional layers, kernels, maxpool layer and batchnormalization layer. We created the basic CNN model using the baseline VGG structure and came across very less accuracy while training and testing the model. Then we try to find out various ways to increase the accuracy of the model and we learnt about data augmentation. Data Augmentation follows the paradigm that the more data to train the better the accuracy of the model. Data Augmentation helps to generate new images by changing some features of original image like height, width, contrast etc. These new images are generated at run time when training the model and we do not have to save these images anywhere. This improved the accuracy of model from 33% to 84%. Now we have some idea about CNN and we know all these layers used in CNN. All these layers try to find the relevant the semantic information of the image before passing it to fully connected network. We were curious to see how at each layer image is being processed by CNN. So, we created the mechanism to visualize the output of all the layers in CNN and understand their purpose.

Now the challenge remains for us to see how can use these convolutional layers to identify which image network has already seen and predicted as some class but sometimes that same image have different label. Network should be able to find the which label is in majority and should use that label of the image rather than the mislabelled one.

3 Reflection

This Independent Study has been very informative journey for me. This was an unconventional and interesting way of learning. I learnt by experimenting first and than by going through text books [6] [8] [3]. I had guidance and freedom to learn at my own pace. I got great resources to learn more about deep learning field. It will help me navigate my career in Computer Vision space. We also focused on understanding the real life applications, understood summary of few papers, shared knowledge on the subject, explored future opportunities in this area. Things that really helped me during independent study is open communication of how things are going which helped us stay flexible and see what is working best for us. We focused more on understanding the basic concepts very well and not just trying to solve problem statement. If I have an option to go for independent study again the only thing I would like to change for myself is that having a better schedule of what and how much time I have to spend during the week on independent study, better structure of things to be achieved for the week. This independent study was in true sense independent

study of deep networks.

References

[1] *Artificial Intelligence Podcast can be found at.* https://www.youtube.com/playlist?list=PLrAXtmErZgOdP_8GztsuKi9nrraNbKKp4.

[2] *Google Colab.* https://colab.research.google.com/notebooks/intro.ipynb.

[3] *Neural Networks and Deep Learning Book.* http://neuralnetworksanddeeplearning.com/.

[4] *Two Minute Paper videos can be found at.* https://www.youtube.com/user/keeroyz.

[5] *Visdom Library.* https://pypi.org/project/visdom/0.1.7/.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org.

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[8] Yann LeCun, Yoshua Bengio, and Geoffery Hilton. *Deep Learning.* Nature 521, 436–444, 2015. https://doi.org/10.1038/nature14539.

[9] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[10] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[9] [10] [7] [6] [8] [3] [2] [1] [4] [5]