Mini Project 5: Dataset Generality and Transferability

Siddhant Prakash
1211092724
sprakas9@asu.edu

***Abstract*—Report for mini-project 5 to study the generality of datasets and transferability of a trained network on a given dataset.**

1. Introduction

In this mini-project we are provided with the YANN [1] toolbox using which, we pre-train a neural network on three different variation of MNIST [2] data-set, and use the pre-trained networks to find the most general dataset of the three. The aim of the project is to use this technique for measuring the dataset generality of the three variants of MNIST dataset provided to us. We will follow up with a discussion of the results we obtain, trying to provide suitable justifications for obtaining such results.

2. Approach

To test the generality of a dataset, we pre-train a network on that dataset, saving the parameters of the network learned as a '.pkl' file, and we note the test accuracy on that dataset. Then, we load the pre-trained network on a new network, initializing it with the weights of the pre-trained network and keeping the layers frozen, i.e. not learnable. We then train the classifier of this new network with the weights of the pre-trained network and note the test accuracy. The generality of the dataset is a ratio of the accuracy from the pre-trained network to the original accuracy. Thus, we can summarize our approach from [3] in the following steps:

1) Initialize and train a network with random weights on dataset i (D_i). The network is called as base network and notated as $n(D_i|r)$.
2) Retrain the network using weights of $n(D_i|r)$ with dataset j (D_j) and k layers frozen to obtain $n_k(D_j|D_i)$.
3) The dataset generality of D_i with respect to D_j at layer k is given by the following equation,

$$g_k(D_i, D_j) = \frac{\Psi_k(D_j|D_i)}{\Psi(D_j|r)} \tag{1}$$

The network architecture consist of two conv-pool layers and two fully connected layer. The first conv-pool layer has filter size of 5X5 with a max-pool layer of size 2X2 and has 20 neurons. The second conv-pool layer has filter size of 3X3 with a max-pool layer of size 2X2 consisting of 50 neurons. The fully connected layers has 800 neurons each with a dropout [4] rate of 0.5. We use nesterov [5] momentum with rmsprop [6] optimizer and the activation for every layer is 'ReLU' with batch normalization [7] on for the conv-pool layers. We use a softmax classifier to predict the classes. In re-training of new network from the pre-trained networks, all layers except the softmax layer is frozen.

The training was done on an Ubuntu 14.04 system with 1.7 GHz Intel Core i5 processor, 8GB 1600 MHz DDR3L SD-RAM and hosting Nvidia GeForce 840M GPU. All training and testing, memory and time is reported on GPU.

3. Experiments

We have three dataset, viz, MNIST_Bg, MNIST_Rotated and MNIST_Noisy. We first train the net on each of the three and note the test accuracy along with saving the network parameters. The results of the trained dataset can be seen in Table 1.

Dataset	Test Accuracy
MNIST_Bg	96.73
MNIST_Rotated	93.88
MNIST_Noisy	92.9

TABLE 1. ACCURACY ON DATASET D_i ($\Psi(D_i|r)$

Then we load each of the trained network and obtain the test accuracy on other datasets with the pre-trained network. The accuracy as obtained with different pre-trained networks is reported in Table 2. The rows signifies the dataset D_i whose pre-trained network was loaded and trained on the corresponding column dataset D_j.

	MNIST_Bg	MNIST_Rotated	MNIST_Noisy
MNIST_Bg	96.78	58.21	95.4
MNIST_Rotated	46.16	94.08	40.85
MNIST_Noisy	91.36	52.38	92.9

TABLE 2. ACCURACY ON DATASET D_j ($\Psi(D_j|D_i)$

4. Results

The generality of dataset obtained as given by equation 1 is summarized in Table 3. The rows denote dataset D_i and the columns denote the dataset D_j. Thus, we obtain dataset generality for all pair $g(D_i|D_j)$.

	MNIST_Bg	MNIST_Rotated	MNIST_Noisy
MNIST_Bg	1.0005	0.62	**1.03**
MNIST_Rotated	0.48	1.002	0.44
MNIST_Noisy	0.95	0.56	1

TABLE 3. GENERALITY OF DATASET ($g(D_i|D_j)$

5. Discussions

From Table 3, we can see the highest generality score is for MNIST_Bg dataset which provide most general features to MNIST_Noisy dataset than any other pair. While the next best score is obtained for a network trained on MNIST_Bg dataset with the pre-trained weights from MNIST_Noisy dataset. Thus, they clearly show that the dataset for both MNIST_Bg dataset and MNIST_Noisy dataset share features more common amongst each other than the MNIST_Rotated dataset.

References

[1] Yet Another Neural Network [YANN] Toolbox. https://github.com/ragavvenkatesan/yann

[2] LeCun, Yann, Corinna Cortes, and Christopher JC Burges. ”The MNIST database of handwritten digits.” (1998).

[3] Venkatesan, Ragav, Vijetha Gattupalli, and Baoxin Li. ”Neural Dataset Generality.” arXiv preprint arXiv:1605.04369 (2016).

[4] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. 2012.

[5] Nesterov, Yurii. ”A method of solving a convex programming problem with convergence rate O (1/k2).” Soviet Mathematics Doklady. Vol. 27. No. 2. 1983.

[6] Tieleman, Tijmen, and Geoffrey Hinton. ”Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude.” COURSERA: Neural networks for machine learning 4.2 (2012).

[7] Ioffe, Sergey, and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015).