Statistical Machine Learning
Statistics GR 5241 — Spring 2022

Data Analysis Project

1 Background

The MNIST database of handwritten digits is one of the most commonly used dataset for training various image processing systems and machine learning algorithms. MNIST has a training set of 60,000 examples, and a test set of 10,000 examples. It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting.

MNIST is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. The original black and white (bilevel) images from NIST were size normalized. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. The images were centered in a 28×28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28×28 field.

2 Data Processing

A lot of packages in python has MNIST database. Let's use the one in `torch.utils.data`. Using the following code to download the data from the server.

```python
# Data download and preprocessing

DOWNLOAD_MNIST = True # If already download, set as False
train_data = torchvision.datasets.MNIST(
    root='./mnist/',
    train=True,  # this is training data
     #transform=torchvision.transforms.ToTensor(),
    download=DOWNLOAD_MNIST,
)
test_data = torchvision.datasets.MNIST(root='./mnist/', train=False)

# change the features to numpy
X_train = train_data.train_data.numpy()
X_test  = test_data.test_data.numpy()

# change the labels to numpy
Y_train = train_data.train_labels.numpy()
```

```
Y_test  = test_data.test_labels.numpy()
```

1. (3 points) Familiarize yourself with the data, including

   (a) Plot one sample in `X_train`. What is the number you see from the 28×28 pixel-field? Does it match with the label in `Y_train`?

   (b) What is the dimension of `X_train` and `X_test`? Normalize `X_train` and `X_test` such that the value of each element lies in $[0, 1]$.

   (c) A popular choice to deal with the labels is to use the *one-hot* embedding. Represent `Y_train` and `Y_test` using one-hot embedding. List the benefit of such transformation.

3 Before Deep Learning

Historically, many methods have been tested with MNIST training and test sets. Yann LeCun, Corinna Cortes, and Christopher J.C. Burges maintained the progress on the accuracy of the classification task until 2012, where you can find the details here: http://yann.lecun.com/exdb/mnist/. Among them, there are 4 reports on the following methods:

Methods	Feature Engineering	Test Error
KNN	None	5%
AdaBoost.M1 / C4.5	None	4.05%
SVM with Gaussian Kernel	None	1.4%

2. (12 points) A key property for a good machine learning algorithm is the *reproducibility*, meaning that one can repeatedly run the algorithm on certain datasets and obtain the same (or similar) results on a particular project.

   (a) (8 points) Try to implement and train the above mentioned classifier on the training dataset, and report the test errors of them using the test dataset. Can you reproduce the results? If not, please justify your reason.

   (b) (4 points) Pick your favorite classifier (not limited to the above mentioned algorithms) and try to implement it on the training set and report the test error using the test dataset. Turn the hyperparameters until it out perform all three of the classifier you implemented in part 2(a).

4 Deep Learning

In this part, you are asked to implement different artificial neural network structures with MNIST training and test sets.

3. (5 points) Train a single layer neural network with 100 hidden units (e.g. with architecture: $784 \rightarrow 100 \rightarrow 10$). You should use the initialization scheme discussed in class and choose a reasonable learning rate (i.e. 0.1). Train the network repeatedly (more than 5 times) using different random seeds, so that each time, you start with a slightly different initialization of the weights. Run the optimization for at least 150 epochs each time. If you observe underfitting, continue training the network for more epochs until you start seeing overfitting.

   (a) Plot the average training cross-entropy error (sum of the cross-entropy error terms over the training dataset divided by the total number of training example) on the y-axis vs. the epoch number (x-axis). On the same figure, plot the average validation cross-entropy error function. Examine the plots of training error and validation/test error (generalization). How does the network's performance differ on the training set versus the validation set during learning? Use the plot of training and testing error curves to support your argument.

   (b) We could implement an alternative performance measure to the cross entropy, the mean miss-classification error. We can consider the output correct if the correct label is given a higher probability than the incorrect label, then count up the total number of examples that are classified incorrectly (divided by the total number of examples) according to this criterion for training and validation respectively, and maintain this statistic at the end of each epoch. Plot the classification error (in percentage) vs. number of epochs, for both training and testing. Do you observe a different behavior compared to the behavior of the cross-entropy error function?

   (c) Visualize your best results of the learned W as one hundred 28×28 images (plot all filters as one image, as we have seen in class). Do the learned features exhibit any structure?

   (d) Try different values of the learning rate. You should start with a learning rate of 0.1. You should then reduce it to .01, and increase it to 0.2 and 0.5. What happens to the convergence properties of the algorithm (looking at both average cross entropy and % incorrect)? Try momentum of 0.0, 0.5, 0.9. How does momentum affect convergence rate? How would you choose the best value of these parameters?

4. (5 points) Redo part 3(a) - 3(d) with a CNN i.e. with one 2-D convolutional layers $\rightarrow$ Relu activation $\rightarrow$ Maxpooling with appropriate hyperparameters. Compare the best result from the single layer neural network and the CNN, what could you conclude?

5. (5 points) Redo part 3(a) - 3(d) with your favorite deep learning architecture (e.g., introducing batch normalization, introducing dropout in training) to beat the performance of **SVM with Gaussian Kernel**, i.e., to have a test error rate lower than 1.4%.

5 More about Deep Learning

In this part, you are working with `train.txt`, `val.txt` and `test.txt`. In particular, `train.txt` contains 20,000 lines and `val.txt` and `test.txt` contains 5000 lines in the same format. Each line contains 1569 coordinates, with the first 784 real-valued numbers correspond to the 784 pixel values for the first digit, next 784 real valued numbers correspond to the pixel values for the second digit.

6. (1 points) As a warm up question, load the data and plot a few examples. Decide if the pixels were scanned out in row-major or column-major order. What is the relationship between the 2 digits and the last coordinate of each line?

7. (8 points) Repeat part 3(a) - 3(d) with at least two of your favorite deep learning architecture (e.g., introducing batch normalization, introducing dropout in training) with respect to with `train.txt`, `val.txt` and `test.txt`. In particular,

   (a) Using `train.txt` to train your models.

   (b) Using the validation error (i.e., the performance on `val.txt`) to select the best model.

   (c) Report the generalization error (i.e., the performance on `test.txt`) for the model you picked. How would you compare the test errors you obtained with respect to the original MNIST data? Explain why you cannot obtain a test error lower than 1%.

Format of the Final Report

The goal of your final report is to document **ALL** the experiments with respect to the MNIST and related datasets you have done and your main findings, so be sure to explain the results. Be concise and to the point. In particular:

1. The answers to all questions should be in pdf form.

2. Please include a README file with instructions on **ALL** your code.

3. Package your code and README document using zip or tar.gz in a file called **STAT5241-*yourCUid*.[zip—tar.gz].**