Objective

The objective of this lab is to gain experience in the use of machine learning methods including: the preparation of a documented dataset with image preprocessing, the use several important machine learning models, and the use of machine learning system analysis tools.

Overview

In part A of this lab you will explore the practical process of curating images for a dataset from the initial image creation and acquisition with a camera including necessary image preprocessing. This dataset will modeled after the MNIST dataset. You will use this dataset to evaluate a logistic regression classifier that has been trained on the MNIST dataset. In part B of this lab you will also optimize the performance of several fundamental machine learning models implemented by in the scikit-learn system. Two week are allowed for this lab.

Lab 7 Preparation

1. Use the lab7.ipynb jupyter notebook for your report template; this includes both part A and part B. If you do any sequences of Linux commands to process the image data make sure that you include these commands in your report (in either code or markdown cells).
2. The lab 7 directory contains the following resources:

File	Contents
lab7.ipynb	The jupyter notebook report template
lab7a.ipynb	The Lab 7 Tutorial (Scikit-learn models)
lab7b.ipynb	The Lab 7 Dataset Loading Tutorial
lab7c.ipynb	The Lab 7 Utilities (data prep and training examples)
vcorner	A python demonstration program for clipping images

Lab 7 Tasks:

1. Review the ML python tutorial. This includes python programming hints and ML experiment organization
2. Review the Lab7 Tutorial. This includes program code and functions that you will be using in Lab 7. Copy and modify program sections from this tutorial as needed. Also review the Lab 7 Utilities that contain useful programs for both parts of this lab.
3. **Part A.** Create and normalize a new dataset that conforms to the MNIST dataset
   Create a new set of 60 images of characters:
   50 of these are the digits 0-9 handwritten (5 of each)
   10 of these to be symbols that are not the characters 0-9
   Create a second copy of this dataset with modified line thickness
4. **Part B.** Evaluate several different classifiers that are:
   a. Trained with the MNIST training set
   b. Evaluated with both the MNIST test set and your test set

---

Part A

1. Data set creation.

The MNIST dataset is described as follows: "The original black and white (bilevel) images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. 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 28x28 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 28x28 field."

Your task is to create an additional dataset that is compatible with the MNIST dataset; i.e., that con be used with a classifier trained on the MNIST dataset. There are a number of ways that you could implement the preprocessing. If you choose to do a different approach to that suggested than give a full description with image documentation of your approach.

Do the following:

1. Create 60 handwritten characters (50 of these are the digits 0-9 handwritten (5 of each) number; 10 of these to be symbols that are not the characters 0-9. Note, these characters must be written by hand on paper! It is not permitted to create the characters using an electronic device and then printing the outcome. It is also not permitted to directly make an image from text created on an electronic device.
2. Photograph these characters into images. Transfer these images to jupyterhub.ece.cornell.edu where you can process them. If you have a number of image files to transfer then put them into a directory on your local computer and create a .zip file of that directory. Use the jupyter lab upload facility to transfer the .zip file to jupyter lab. Then unzip the uploaded file.
   Avoid using any apps. in this process that may modify the properties of the images (such as photoshop). Although with smart devices some modification may be unavoidable. For efficiency you may choose to do a batch of many characters in a single image and then clip individual characters during preprocessing

3. Preprocess these images as outlined above.
   1. First, threshold the images to make binary (bilevel) images (**vpix**), also (**vfmt**) to convert from original image format to vx.
   2. For each character image, find the bounding of the character, then map the image into a 20 x 20 new image space. For interpolation of the binary image consider using (**vimag**)
   3. for each image compute the center of mass (COM) then translate copy your image to an image of size 28 x 28 such that the COM is now in the center.
   4. Invert images if necessary (**vpix**) to have light foreground and dark background.

   These transformations may be done in the VisionX framework. It is permitted to use the scipy or numpy equivalents in python to do the steps of thresholding, image resizing with interpolation and image centering to the COM; it is not permitted to use other python libraries. The goal of the lab is to gain the experience of doing it yourself.
   A python example program vcorner for clipping a small image from a larger image is available that may be useful for this lab.
   The commands **vfmt** and **vxport** may be used to convert between VisionX image file format and other image file formats.

4. Your final images must be stored in .png format in a local directory called *idata* with a csv index file *labels.csv* to describe the contents (see the data importing tutorial for details).
5. To explore the impact of line thickness on performance, make a second dataset with a different line thickness. First visually compare your dataset with examples form the MNIST dataset. Is the line thickness of the characters different? Make your second data set more similar to MNIST if possible, otherwise just change the line thickness by a small amount. Repeat the preprocessing of the above dataset with a new step between thresholding and interpolation that modifies the thickness of the character lines The suggest program is **vmorph**. The line thickness may be modified by specifying an appropriate kernel (s=, t=) on the command line (erode -e to make thinner, or dilate -d to make thicker). The kernel determines the amount of change. Save the second dataset in a directory called *idata2*.
6. For all steps outlined above, in your lab report for idata, be sure to give a complete description of your method with multiple image examples. Also give key information and some image examples for dataset idata2.

2. Experiments with Logistic Regression classifier

For this section of the lab you should use code from the Lab 7 tutorial. Also, when reading the image dataset that you have created you should use code from the Lab 7 Data importing Tutorial

1. Starting with the classifier described in the Lab tutorial evaluate the performance of the classifier for both the standard test dataset and also your own datasets (idata and idata2).
2. Plot the images from the standard MNIST test set of the first 40 images with errors together with the correct and incorrect labels. Discuss these results.
3. Plot all your test images with truth and predicted labels. Discuss these results.

Part B

3. Experiments with classifiers

Your are to specify train and test different classifiers with the MNIST dataset. See the Lab 7 Utilities for how to effectively conduct this part of the lab.

1. Be sure to **clearly** report the training time, testing time, and score for each evaluation.
2. Report the score, time for training, and time for testing for each classifier into a table and discuss results
3. Give confusion matrix for each classifier and dataset

Do the following:

1. Evaluate a K-nearest-neighbor (K-NN) classifier for 1 and 3 near neighbors. Also, select a number of near neighbors for an additional test.
   The K-NN classifier may take a long time to execute on the full MNIST dataset. Intelligently modify the size of your training and testing datasets so that the evaluation time on your computing platform is less than 10 minutes.

   Explore at least two approaches: (a) train on just part of the training set, and (b) use PCA. Try at least two different parameters for each method (for PC start with number of components = 20).; Show the time in seconds taken for training and testing. Discuss the expected impact of reducing the size of the dataset on your results and compare with the results you achieved.

2. Evaluate two multi-layer perceptron classifiers one with a single hidden layer and one with two hidden layers. As with the K-NN classifier, intelligently modify the number the dataset to complete each evaluation within 10 minutes.
3. Evaluate two support vector machine classifiers; one with a linear function, and one with radial basis functions. Note, you only need to do a single run with the radial basis functions as that classifier may take more time than other methods. As with the K-NN classifier intelligently modify the number in the training dataset to complete each evaluation within 10 minutes.

Important Reference material

Scikit-Learn Reference

1. Getting Started
   This introduction provides important information on the scikit-learn conventions and data formats such as the scikit-learn fit (training) and predict (testing) functions. See also the scikit-learn Examples and Users Guide.
2. MLP with weight visualization
   This tutorial shows how a MLP classifier is used.
3. SVM (not MNIST) example
   This tutorial gives an example to using the performance measures in scikit-learn including a confusion matrix

MNIST Introduction

1. MNIST Database
   This reference provides information from Yann LeCun on the formation of the MNIST dataset and the current status of the success of different machine learning methods.