CPSC 340 Machine Learning Take-Home Final Exam (Spring 2020)

Question 1 [70/100 points]

Recall the MNIST data set from assignment 6 which could be downloaded at `http://deeplearning.net/data/mnist/mnist.pkl.gz`. Go ahead and download this dataset, since we will be using it for this question.

MNIST contains labelled handwritten digits (i.e. 0 to 9) with 60,000 training examples and 10,000 test examples. It is a widely used dataset and with known error rates for several machine learning methods encountered in class. We will be using `http://yann.lecun.com/exdb/mnist/` as a reference for test errors.

For this question, you will implement 5 machine learning methods from class and apply them to the MNIST dataset in order to do supervised classification of digits, with the goal of minimizing the test error. The approaches to be implemented and employed are one example from each of the following types:

1. k-nearest-neighbours (KNN)
2. linear regression
3. support vector machine (SVM)
4. multi-layer perceptron (MLP)
5. convolutional neural network (CNN)

This question will be answered in a report format, provided at the end of the exam LaTeXfile `final.tex`. You will have to provide test errors achieved using your implementations, calculated as the percentage of incorrectly labeled test examples (using the default test set provided in the MNIST dataset partition). As an example, results from `http://yann.lecun.com/exdb/mnist/` for each of the above models (with particular hyper-parameter settings) are shown below::

Model	Error (%)
KNN	0.52
linear regression	7.6
SVM	0.56
MLP	0.35
CNN	0.23

Running `python.py main.py -q 1` will load the MNIST dataset into a training set and a test set (if you stored the dataset in a separate directory called `./data/`). The rest of the code (model, training, and testing procedures) must be written by you. You are not permitted to use built-in models (e.g. from PyTorch or scikit-learn), but we encourage you to use code from your assignments. Remember that in past assignments, you have had to implement all of the models listed except for CNNs.

Bundle your code along with a `.pdf` generated from the filled in LaTeXreport into a `.zip` file and submit it to Gradescope. Marks may be taken off for very messy or hard to read code, so make sure to use

descriptive variable names and include comments where appropriate. Since we are also marking based on test error, you are expected to only evaluate performance on the test set in the partition provided.

Skeleton for Question 1 Answer

1 Introduction

Three sentences describing the MNIST classification problem. MNIST is a database consists of 1,797 digits representing the numbers 0-9 written by high school students and employees of the United States Census Bureau. It is used for training various image processing systems. We are going to use differnet machine learning method to classify the images

2 Methods

2.1 KNN

Three to four sentences describing the particulars of your KNN implementation, highlighting the hyperparameter value choices you made and why.

In KNN, the input consists of the k closest training examples in the feature space, and the output depends on whether k-NN is used for classification or regression. The data is splited into training and testing set to train the k-NN classifier. There is also validation set to find the best value for k.

2.2 linear regression

Three to four sentences describing the particulars of your linear regression implementation, highlighting the hyperparameter value choices you made and why.

The data is splited into training and testing set to train the linear regression classifier. Use the cross-entropy to compute the loss. Then use gradient-based strategy for ftting the robust regression model under the og-sum-exp approximation.

2.3 SVM

Three to four sentences describing the particulars of your SVM implementation, highlighting the hyperparameter value choices you made and why.

Use SVM with RBF kernel. Firstly, standardize the data with mean=0 and std = 1. Then use cross-validation to find the best parameters C and gamma.

2.4 MLP

Three to four sentences describing the particulars of your MLP implementation, highlighting the hyperparameter value choices you made and why.

Select loss function called categorical cross entropy and Stochastic Gradient Descent as optimization algorithms. Since the running time of cross-validation is too long, I tried several value in hyperparamer depth and layer width in traning data and use the one with smallest testing error.

2.5 CNN

Three to four sentences describing the particulars of your CNN implementation, highlighting the hyperparameter value choices you made and why.

There are 2 convolution layers followed by pooling layer. Multiple filters are used at each convolution layer.The data is splited into training and testing set to train the CNN classifier

3 Results

Model	Their Error	Your Error (%)
KNN	0.52	
linear regression	7.6	
SVM	0.56	
MLP	0.35	
CNN	0.23	

4 Discussion

Up to half a page describing why you believe your reported test errors are different than those provided (and "detailed" on the MNIST website).

There choose of hyperparameters could influence error. Moreover, the data need preprocessing before fitting into model.