CPSC 340 Machine Learning Take-Home Final Exam Q1 (Spring 2020)

Margot Chen
99595167

1 Introduction

The MNIST dataset is a labelled handwritten digits dataset with 60,000 training examples and 10,000 test examples. The digit image size is 28*28. Every image was transformed into a vector of length 784. In this question we are building multi classification models using 50,000 of the training examples and testing the models on test examples

2 Methods

2.1 KNN

KNN measures similarity with distance metrics. In this case I used Euclidean distance. The only hyper-parameter is k. I tried from 3 to 6. When k=3, I got the least training error and validation error.

2.2 linear regression

I used multi-class softmax with SGD and fixed learning rate. The hyper-parameters are learning rate α, batch size, and number of epochs. Since I don't know how many epochs is enough, I set epoch = 100 and used early stopping. After each epoch, I check the validation error. The process ends when the error is not decreasing. For α I tried 0.01, 0.001, and 0.0005. For batch size I tried 500, 1000, 1500. They didn't make huge difference.

2.3 SVM

I used multi-class SVM with SGD. The loss function and subgradient were abtained from
https://cs231n.github.io/optimization-1/
Again, I used early stopping to determine when to stop SGD. The hyper-parameters are learning rate α, batch size, λ for L2 regularization, and number of epochs. I tried the same choices of α and batch size as linear regression, they didn't make much difference either. A small enough λ yields large margin and small hinge loss. So I picked $\lambda = 0.001$.

2.4 MLP

I used both GD and SGD. For GD, the max iteration is 100. For SGD, epoch=100, batch size=1000, alpha=0.001. I tried 10, 50, and 100 hidden units. 50 hidden units got the best error.

2.5 CNN

3 Results

Model	Their Error	Your Error (%)
KNN	0.52	3.19
linear regression	7.6	8
SVM	0.56	8.6
MLP	0.35	4.9
CNN	0.23	

4 Discussion

4.1 KNN

MNIST dataset has very high dimension (784 features). It took a lot of time and space to compute. I didn't do any preprocessing to the dataset, which caused the huge difference between my result and the reference. Other distance metrics such as L3 may also improve the result.

4.2 Linear regression

My result is close to the reference. Using changing step size may improve the result slightly.

4.3 SVM

Since the data is not linear separable, SVM cannot get the best result unless we use kernels to add nonlinearity. I tried but didn't figure out how to implement kernelized SVM.

4.4 MLP

I only used one hidden layer. More hidden layers and hidden units can bring better results but also require more computational resource.