Machine Learning Milestone

Deric Pang
dericp@cs.washington.edu

Saidutt Nimmagadda
nimmas@cs.washington.edu

Abstract

Our project investigates the task of recognizing handwritten digits. Our initial project proposal involved reading three papers [1]-[3] and determining which techniques to use to achieve our task.

1 Project Description

The data for our project was taken from the MNIST dataset. As written on the Kaggle competition website, “The MNIST (’Modified National Institute of Standards and Technology’) dataset is a classic within the Machine Learning community that has been extensively studied. More detail about the dataset, including Machine Learning algorithms that have been tried on it and their levels of success, can be found at http://yann.lecun.com/exdb/mnist/index.html.”

Simply put, we aim to take pixel data from gray-scale images of hand-drawn digits and classify the digit as a number from 0 to 9.

2 Progress

2.1 Readings

The readings were helpful in directing us towards the techniques that we should try to implement. We understood from the conclusions of the Lecun paper [1] that the k-Nearest Neighbors algorithm would not only pose serious scaleability difficulties when it came to runtime and memory usage, but it would also be a comparatively unreliable classifier.

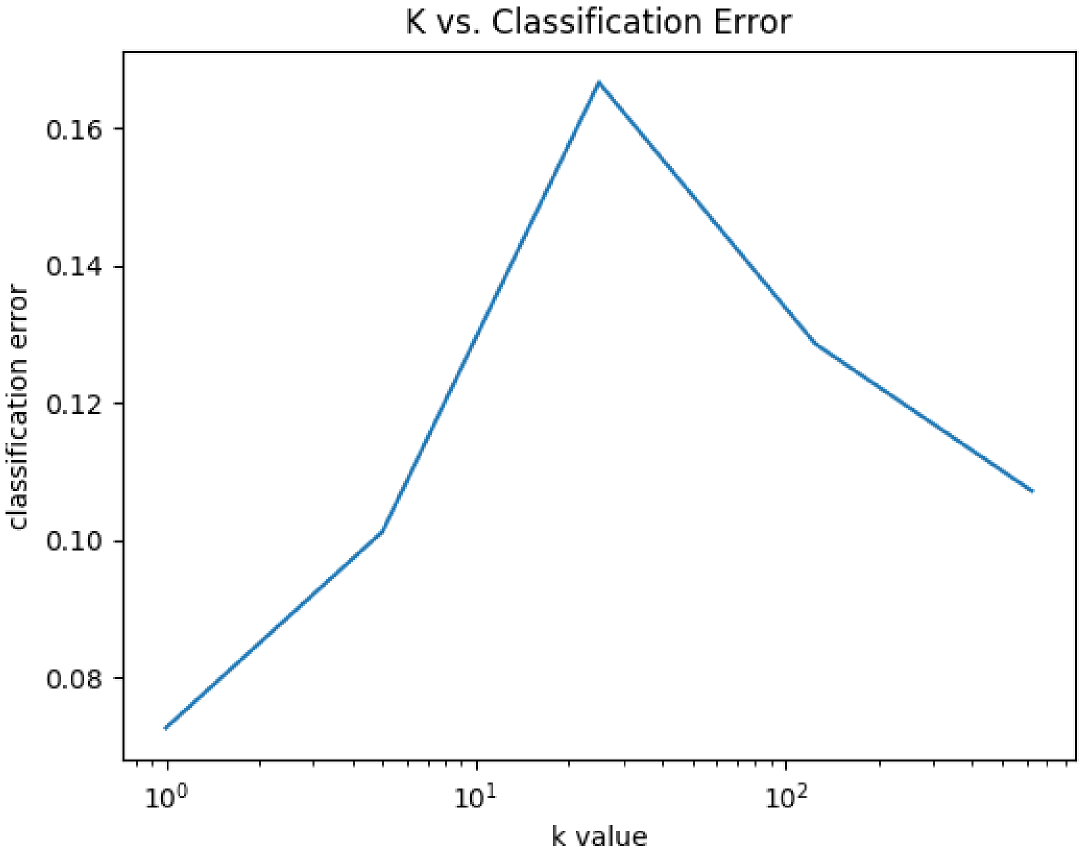
Our findings from the milestone portion of the project confirms this, with our implementation of k-NN taking a good amount of time (roughly 5 minutes) to run over less than 10 percent of the total dataset. In addition, we only achieved classification errors of roughly 10 to 20 percent. When we implement a convolutional neural network, like LeCun did in his paper, we expect a decrease in runtime and memory usage (which would promote scaleability), as well as a large decrease in classification error.

Additionally, Maji and Malik [2] found that “with improved features a low complexity classifier, in particular an additive-kernel SVM, can achieve state of the art performance.” We hope to attempt to implement this as well.

2.2 k-NN

We have implemented a classifier using k-nearest neighbors regression. In order to determine the best value of k to use, we implemented cross-validation using an 80-20 split.

The first issue we ran into was the massive runtime needed to run k-nn over such a large model (the training set has 42,000 points). Given runtime constraints, we ran the algorithm on 10% of the training set. For each value of k, we found the classification error on the validation set.



Surprisingly, we can see that with $k = 1$ we achieve the lowest classification error. This is most likely due to the lack of a kernel to weigh the points according to how far they are from the query.

3 What's Next

The first thing to try next is to add a kernel to our current k-NN implementation. We suspect that this will decrease our classification error significantly, epecially for larger values of k. However, this will not solve the problem of the massive runtime needed for k-nn.

We hope to implement an SVM based on what was discussed by Maji and Malik [2]. This will significantly decrase the runtime of our classifier.

Finally, we want to implement a convolutional neural network. As Lecun [1] found, the nueral network performs the best out of all the conventional classification methods on hand-drawn digits.

References

[1] LeCun, Yann, et al. ”Comparison of learning algorithms for handwritten digit recognition.” International conference on artificial neural networks. Vol. 60. 1995.

[2] Maji, Subhransu, and Jitendra Malik. ”Fast and accurate digit classification.” EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-159 (2009).

[3] Sundaresan, Vishnu, and Jasper Lin. ”Recognizing Handwritten Digits and Characters.” (1998).