Optimization for Learning - FRTN50

Extra Credit Assignment: Multi-Class Classification

Written by: Martin Morin

Latest update: October 18, 2019

Introduction

`MNIST`

`MNIST`[1] is a now classic data set for benchmarking image classification algorithms. It consist of 28×28 pixel images of handwritten digits and the goal is to identify the digit in a given image. It is split into two sets, one consisting of 60000 images for training and one with 10000 images for testing/benchmarking.

A function `loadmnist` have been provided in `mnistdata.j` for loading the data. It allows for a subset of the digits 0-9 to be selected and the size of the data set to be reduced. Some basic re-scaling of the images is also done. The file `mnistdata.jl` also provide a function displaying the images.

For reference, state of the art convolutional neural networks have achieved error rates of around 0.3%. Variants of one-against-rest SVM:s have achieved error rates of around 1%.[1] You are unlikely to get very close to these numbers since that is not the focus of this assignment. However, with different kernels and depending on the formulation, it should be possible.

Objective

The objective is to design multi-class classifiers based on the two different SVM formulations. One of the drawbacks of the dual based SVM formulations seen in the course is their storage requirements. The main topic of interest is therefore to examine this and see how your algorithms scale with problem size. For example, given 64 bit numbers the size of the full training set is roughly $60000 \times 28 \times 28 \times 64 \div 8 \approx 360\,\text{MB}$. This is small enough to hold in the RAM of any modern computer. However, your classification algorithms will need to keep more things in memory than just the data and will therefore potentially need much memory than this.

Tasks

Task 1 - Reduced `MNIST` and Average Confidence

Design a classifier with a reduced training set consisting of the digits 0-2 and the size reduced with a factor 3, i.e. `loadmnist(0:2,reduction=3,set=:train)`. Use the average confidence formulation in Exercise 4.12 and the following 5:th order polynomial kernel $K(x, y) = (x^T y)^5$. Solve the problem with a coordinate descent algorithm. What is the best error rate you can achieve?

Task 2 - Algorithm Complexity

Experiment with the data set size, i.e. add/remove more digits and/or smaller/larger reduction factor. Roughly, how large problems can you solve? In theory, how large problems should you be able to solve? Look at how does the following scale with the number of data points and classes/digits.

- The algorithms memory requirement.
- The computational cost for one iteration of your algorithm.
- The computational cost for classification of unseen data.

What are the bottlenecks of the algorithm?

Task 3 - Full `MNIST` and Average Confidence

Use the same SVM formulation, kernel and algorithm as the previous assignments but implement it so the storage requirement is linear in the number of training data points. Design a classifier with the full `MNIST` training set. Note, it is unlikely that you will have time to solve the problem to high accuracy. Simply run for a fixed amount of iterations that give good performance. What is the best error rate you can achieve?

The reduction in storage will not come without cost. What is the computational cost for one iteration of this implementation? Based on the iteration cost, how much slower should it be? Is it that much slower? How large problems should you now be able to solve within reasonable time?

Task 4 - Full `MNIST` and Worst Case Confidence

Design a classifier based the worst case confidence formulation from Exercise 4.12. Use the full `MNIST` training data. An ordinary coordinate descent algorithm will not work in this case so the coordinates need to be processed in blocks.

How does the memory requirement scale with the number of training data point? Compare the computational cost for one iteration with the other formulations/implementations? Roughly, how much slower should it? How much slower is it in reality? What is the computational cost for classification of unseen data?

Report

The report should be in pdf format all your code should be submitted with it. The report should contain all relevant information and enough details for someone to implement the algorithms, replicate your practical results, and verify claims. Present and interpret the results and explain any procedures you used, motivate them if necessary.

The grade will be based on the quality of the report and your work and not necessarily how much you do. Of course, the two can not necessarily be separated but try to solve the tasks to the best of your ability within the time you have. A well made and well presented report containing just a couple of tasks will be valued higher than a rushed solutions of all tasks.

References

[1] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. *The MNIST Database of Handwritten Digits*. URL: `http://yann.lecun.com/exdb/mnist/`.