CSC2515-Introduction to Machine Learning

Digit Recognition in Street View House Numbers

Bai, Min–mbai@cs.toronto.edu

Loyzer, Mark–loyzer@cs.toronto.edu

Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
December 21, 2015

Abstract

Sections and Chapters

Min Bai, Mark Loyzer

December 21, 2015

Contents

1 Abbreviations, Symbols, Nomenclature

???

2 Introduction

This project focuses on classifying digits from street view images. Towards this goal, download the Format 2 images from the Street View House numbers dataset [1] with train 32x32.mat, test 32x32.mat data, which you can find at http://ufldl.stanford.edu/housenumbers/. In this task, all the images have a fixed 32 32 resolution with character-level ground truth labels. For each example, the labeled character is centered at the image. There are ten classes in total 1 for each digit. Divide the training into train and validation (e.g., 80% and 20%).

Note that the data is collected from street-view images, thus there exist vast intra-class variations. To generate competitive performance, you may want to consider exploiting good feature representations that are robust to those variations, whether they should be hand-crafted features or learned features. In order to boost performance, you may also want to consider augmenting the training data with extra 32x32.mat. The purpose of this project is to investigate machine learning techniques to solve this task. Try things that we have seen in class, or other techniques if you feel like it. Write in your report what you have done, what you observe, what you have tried, why you did what you did, etc.

3 Techniques

3.1 Convolutional Neural Networks

3.2 K-Nearest Neighbours

K-Nearest Neighbours (K-NN) has traditionally performed very well in a variety of tasks so we will use K-NN with a variety of preprocessing techniques. These techniques include:

- Each RGB colour channel as added features. Considering how K-NN prevails in situations where the dataset has a smooth distribution (ie.

the number of dimensions can't be too large unless you have a plethora of data), we expect this to produce poor results.

- Each RGB colour channel as an additional example. The idea behind this technique is to expect that colours do not matter, but intensities do. Additionally we can gain three times the number of examples.
- Converting examples to grayscale. Similarly, if the colours do not influence the class then converting to grayscale should be an efficient way to reduce the dimensionality.
- Normalizing grayscale examples assuming again that intensities contain information.
- Normalizing grayscale examples with a threshold so that all pixels below the threshold will be 0.
- Extracting hard coded feature detectors ()

All techniques will be evaluated for $K \in 1, 3, 5, 7, 9, 11, 13, 15, 17, 19$.

3.2.1 RGB Channels as Features

Figure 1 shows the performance of

4 Conclusion

5 References

Figure 1: