Handwritten Digits Classification

[Team Y.E.S.: COMP 598 Group Project 3] *

Yuting Wen
McGill University
yuting.wen@mail.mcgill.ca

Emmanuel Bengio
McGill University
emmanuel.bengio@mail.mcgill.ca

Sherry Ruan
McGill University
shanshan.ruan@mail.mcgill.ca

ABSTRACT

In this project, we aim at classify a much more difficult variation of the MNIST dataset of handwritten digits. We adopt feature selection and construction techniques together with four main machine learning algorithms: Gaussian Naive Bayes, fully connected Feedforward Neural Network, Linear Support Vector Machine, and XXX (name of the 4th algorithm here). We analyze and assess the parameter selection process and the performance of each algorithm. We conclude the report with discussion and suggestions for further improvement.

1. INTRODUCTION

The MNIST database of handwritten digits [4] is a standard touchstone of effective image classification algorithms. It is extensively studied and tested by many machine learning techniques [3, 2, 1, 7] . The original dataset consists of more than 60,000 handwritten digits from 0 to 9, normalized to a 28x28 fixed image size [4].

The dataset we are dealing with in this project is more challenging. Modifications of the original dataset include embossing, rotation, rescaling, and texture pattern. These artificial alterations introduce a great amount of noise and undoubtedly increase the level of difficulty of the digit classification task. The modified dataset contains 50,000 training examples of 48x48 fixed size, and the test set comprises 20,000 instances which require classification [6].

We decided to apply four different algorithms: Naive Bayes, Feedforward Neural Networks, Linear Support Vector Machine, and XXX to the modified MNIST dataset. For the baseline algorithm, we chose Gaussian Naive Bayes since features given as float numbers are continuous. For Neural Networks, ... For Linear SVM, ... For XXX, ... (one or two sentences summarizing each algorithm)

The performance of algorithms varies widely. The base line algorithm, Naive Bayes, provides around 40% accuracy, this may due to the fact that the Naive Bayes assumption does not hold in the digit classification task in general. NN..... . SVM..... XXX..... (one or two sentences summarizing the performance)

Our empirical results, though preliminary, provide considerably accurate predictions (especially XXX) for the modified MNIST digit classification. Thus, we are optimistic of applying the algorithms and analysis presented in this report to other real-world classification problems. In particular, this can motivate the further study on more specialized machine learning algorithms on image classification tasks.

*The dataset and the implementation of the algorithm described in this report is available at https://github.com/yutingyw/imageClassification

2. RELATED WORK

This section is optional this time

3. DATA PREPROCESSING

We adopted different data preprocessing methods based on the characteristic of each machine learning algorithm. Since the dataset was given in a relatively organized format (csv files containing float numbers), we spared little effort to format data or extract numerical data from images. Most data proprecessing methods we used were adapted for a specialized algorithm.

In Naive Bayes, we adopted normalization to make it suitable for the algorithm. We obtained a set of scaled examples of unit norm after the normalization. We chose L2 norm since it resulted in the greatest improvement in terms of accuracy. We will give more details including the graph showing accuracy versus data preprocessing methods in later section (testing and validation).

For Neural Networks, ...

For Linear SVM, ...

For XXX, ...

4. FEATURE DESIGN AND SELECTION

feature design and selection methods

5. ALGORITHM SELECTION

5.1 Baseline: Naive Bayes

[5]

5.2 Neural Net

5.3 Open:

algorithm selection for each of the 3 categories (baseline, neural net, open)

6. OPTIMIZATION

6.1 Baseline:

6.2 Neural Net

6.3 Open:

optimization methods for each of the 3 categories (baseline, neural net, open)

7. PARAMETER SELECTION

7.1 Baseline:

7.2 Neural Net

7.3 Open:

model order, learning rate, etc. for each of the 3 categories (baseline, neural net, open)

8. TESTING AND VALIDATION

8.1 Baseline:

8.2 Neural Net

8.3 Open:

detailed analysis of your results, outside of Kaggle for each of the 3 categories (baseline, neural net, open)

9. DISCUSSION

pros and cons of your approach and methodology)

We hereby state that all the work presented in this report is that of the authors

10. REFERENCES

[1] Y. Huang, J. Zhao, M. Tian, Q. Zou, and S. Luo. Slow feature discriminant analysis and its application on handwritten digit recognition. In *Proceedings of the 2009 International Joint Conference on Neural Networks*, IJCNN'09, pages 132–135, Piscataway, NJ, USA, 2009. IEEE Press.

[2] C. Jou and H.-C. Lee. Handwritten numeral recognition based on simplified feature extraction, structural classification and fuzzy memberships. In *Proceedings of the 17th International Conference on Innovations in Applied Artificial Intelligence*, IEA/AIE'2004, pages 372–381. Springer Springer Verlag Inc, 2004.

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[4] Y. Lecun and C. Cortes. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/. Accessed: 2014-10-30.

[5] J. Pineau. Comp 598 - applied machine learning lecture 5: Naive bayes classification. http://www.cs.mcgill.ca/~jpineau/comp598/Lectures/05NaiveBayes-publish.pdf. Accessed: 2014-10-11.

[6] J. Pineau and A. Leigh. Comp-598: Applied machine learning mini-project 3: Difficult digits. http://www.cs.mcgill.ca/~jpineau/comp598/Projects/Project3_instructions.pdf. Accessed: 2014-10-30.

[7] Theano. Classifying mnist digits using logistic regression. http://www.deeplearning.net/tutorial/logreg.html. Accessed: 2014-10-30.