McGill University

COMP 551 - 001

Applied Machine Learning

Kaggle Competition Report

Garcia La Rotta, Camilo. camilo.e.garcia@mail.mcgill.ca (260657037)
Schnaidman, Jacob. YYYYY@mail.mcgill.ca (XXXXX)
Wiltzer, Harley. harley.wiltzer@mail.mcgill.ca (260690006)

Group YYYYY

March 16, 2018

CONSTRAINTS (THIS IS NOT A SECTION)

The main text of the report should not exceed 6 pages.

References and appendix can be in excess of the 6 pages.

The format should be doublecolumn, 10pt font, min. 1” margins.

You can use the standard IEEE conference format

Introduction

This report discusses the applied methods and obtained results regarding the image analysis Kaggle competition. The problem at hand was to automatically recognize randomly scaled hand-written digits. The input is a 64x64 grey-scale image and the output is the number corresponding to the digit with maximum area. To address this challenge, we performed supervised training through LinearSVMs (SVM), Logistic Regression (LR), Neural Networks (NN) and Convolutional Neural Networks (CNN). As it was expected, the performance of the linear learners was almost as bad as that of a random classifier. As for the hand-made NN, XXXXXXXX. Finally, the official results submitted to the Kaggle competition where those of the CNN which YYYYYYYY.

Feature Design

We did not generate new features based on combinations of existent features, nor did we leveraged external data during the feature design/model selection/training process. Because he images we are working with are grey-scale, there is no use in applying dimensionality reduction techniques to the color channels: e.g. collapsing collapse the RGB channels into a single gray-scale channel.

For the linear learners, to reduce computational time and eliminate potentially irrelevant features, we performed PCA for the LR model. PCA reduces the number of interested dimensions in the data space, leaving those with maximal variance. We did use this technique for the SVM model because by definition it already perform feature selection.

As for the NN, we leveraged OpenCV pytorch .... **TODO**

Algorithms

Linear SVM

Supervised classification algorithm, which inheritently performs feature selection by choosing the subset of featres with maximal variance. It handles multiclass classification through one-vs-rest scheme. Its tuned hyper-parameters are:

- The penalty which controls the penalization of the regularization and optimization problem norm.
- The dual/primal optimization problem. We have more samples than features, hence we prefer the dual.
- The penalty parameter of the error term (c) which determines he influence of the miss-classification on the objective function. the larger C, the model will choose a smaller the hyper-plane margin if it means it can correctly classify more points.

Logistic Regression

Supervised algorithm which leverages the sigmoid function and one-vs-rest scheme to perform multiclass classification. Its tuned hyper-parameters are:

- The penalty and dual/primal, C which serve the same purpose as for SVMs.
- The solver, which is the algorithm used in the optimization problem. For multiclass problems we cross validated newton-cg, sag, saga and lbfgs.

Neural Network

Supervised classification algorithm. The two main processes are: Feed-Forward, where features are slided through layers of nodes fully connected to the previous layer each one containing a weight. At each layer an activation function is applied. Back-Propagation, which is implemented through the chain-rule to efficiently compute the gradients of the cost function with respect to the weight parameters of the network.

Convolutional Neural Network

A variant of NNs optimized for image analysis, CNNs reduce the connectivity of nodes to a local range (parameter sharing), also constraining all nodes in a depth slice to the same weights. These constraints vastly reduce the ressources required to learn distinct features of images.

Methodology

For all models, a very small subset of samples (100) was used to better understand the behaviour and performance improvements of every model through every logical permutation of hyper-parameters. The official model selection process was done through K-fold cross validation with 3 folds and the complete training data-set. As for the ranges of values used, we leveraged Numpy's logspace package, which generates an equally distant set of numbers between two specified boundaries in log-space.

To visualize and understand the behaviour of each hyper-parameter on the performance of the SVM and LR models, we cross validated with every valid permutation. Note that for example that for SVM's the combination: DUAL, L1 penalty and Squared Hinge loss is invalid.

For LinearSVM .... (TALK ABOUT FINAL BEST HYPERPARMS)

For LR .... (TALK ABOUT FINAL BEST HYPERPARMS)

Results

Discussion

It was to be expected that linear learners would be vastly sub-optimal for this type of image analysis problems. While the preprocessing techniques we used are very rudimentary, due to the mathematical nature of linear algorithms we would not vast improvements even if we spent more ressources on feature and model selection. With respec to NNs and CNNs, becoming more proefficient in image preprocessing libraries such as OpenCV and Tensor libraries such as PyTorch and TensorFlow would be interesting aread of future work to improve the performance and ressource requirements of our models. Another area in which we could see noticeable improvements is applying the image preprocessing technique of data augmentation through: uniform rotation, centering, translation, rescaling, flipping, shearing and stretching.

Statement of Contributions

All members helped writing the report. In broad terms, the focus of each student was the following:

- **Harley Wiltzer: TODO**
- **Jacob Schnaidman: TODO**
- **Camilo Garcia La Rotta:** Linear Learners (Feature/Model selection, coding and analysis)

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

References (OPTIONAL)

Appendix (OPTIONAL)