McGill University

COMP 551 - 001

Applied Machine Learning

Kaggle Competition Report

Garcia La Rotta, Camilo. camilo.e.garcia@mail.mcgill.ca (260657037)
Shnaidman, Jacob. YYYYY@mail.mcgill.ca (260655643)
Wiltzer, Harley. harley.wiltzer@mail.mcgill.ca (260690006)

Group YYYYY

March 20, 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). The linear learners each achieved accuracies of approximately 60%. As for the hand-made NN, results of about 65% were observed, for networks of both one and two hidden layers. Finally, the official results submitted to the Kaggle competition where those of the CNN which XXXXX.

Feature Design

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

OpenCV image transformation libraries were leveraged to enhance the samples' features. To remove the background pattern a binary threshold filter was used. To remove the remaining noise from darker regions of the background pattern we dilated and eroded the black pixels. We also experimented with Gaussian and standard pixel blurring, but cross validation showed blurring reduced the model's capacity to correctly classify the samples. To finalize preprocessing, we extracted the largest area contour into a new image and cropped it at 32X32, vastly reducing the input dimension and ensuring the only relevant information was the binary array representing the largest area digit. The visualization of the aforementioned transformations can be found in the Appendix, Figure 1.

Before training, the images binary values were transformed from (0,255) representing black and white to (1,0). This normalization proved to be effective at increasing the rate of convergence in some models.

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

Algorithms

Linear SVM

Linear SVM was used as a supervised classification algorithm, which inherently performs feature selection by choosing the subset of features with maximal variance. It handles multiclass classification through the 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 logistic 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

The Neural Network is a supervised classification algorithm. The two main processes are Feed-Forward, where features are forwarded through layers of nodes fully connected to the previous layer each one containing a weight. For this reason, Neural Networks are called Multi-Layer Perceptrons, as each node updates its weight parameters based the error seen by the network. At each layer a nonlinear activation function is applied, allowing the model to learn arbitrarily nonlinear relationships from its training data. Next, Back-Propagation is used to adjust the weights of the network by leveraging dynamic programming to perform efficient chain rule computations.

Convolutional Neural Network

A variant of NNs optimized for image analysis, Convolutional Neural Networks (CNN) reduce the connectivity of nodes to a local range (local connectivity), also constraining all nodes in a depth slice to the same weights (parameter sharing). These constraints vastly reduce the resources required to learn distinct features of images, allowing for far better classification for a given amount of processing power and time.

Methodology

To commence the process, the models were run through small subsets of the data (5000 samples) in order to get a grasp of the optimal ranges of hyperparameters without sacrificing too much time. For the SVM classifier, for example, it was seen that the model performed best for penalty parameters in the range of $5*10^{-4}$ to 10^{-3}. For Logistic Regression, each logical ordered set of hyperparameters from broad ranges were attempted. It was seen that the lbfgs solver was always vastly superior, and the penalty parameter allowed the model to perform best in the range of $5*10^{-3}$ and $2*10^{-2}$. This allowed us to efficiently perform cross validation on a concentrated range of possibilities. For the linear learners, computation was fairly fast so we could afford to do 3-fold cross validation. It was seen that the best penalty parameter for SVM was $0.0009\overline{4}$, and for Logitic Regression the best penalty parameter was 0.01. With these hyperparameters (and the lbfgs solver for Logistic Regression), the SVM Classifier and Logistic Regression Classifier scored accuracies of 60% and 57% respectively.

For the fully-connected feed-forward neural network, there were an enormous amount of hyperparameters and the model was very slow to train. Once again, it was necessary to run the model on smaller datasets in order to gain an intuition of the ranges and possibilities of best performers. Due to the enormous amount of time spent on training, it was decided to limit the NN to at most two hidden layers. Furthermore, the amount of hidden units at each layer was another hyperparameter to tune. Furthermore, the *layer types* had to be chosen, as they define the nonlinear activation function executed at each step. The logistic sigmoid, hyperbolic tangent, softplus rectifier, and ReLU activations were tested, but it was seen that for both the first and second hidden layers, the sigmoid and hyperbolic tangent activations performed much better. The final cross validation involved testing neural network architectures with one and two hidden layers in all permutations of logistic sigmoid and hyperbolic tangent layers.
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:** Image preprocessing, 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

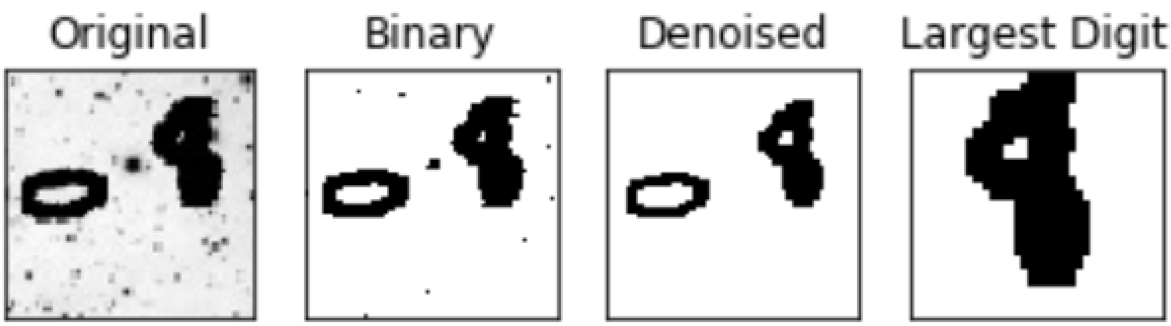

Figure 1: Image preprocessing with OpenCV