Handwritten Digit Recognition Using Multilayered Learning and Elastic Matching

Mutian Chen and **Ryland Wheliss**
{kychen,rywheliss}@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

Abstract

The abstract should be between four and seven sentences long. Introduce the problem you are studying. Describe what you did. Summarize your results — what did you discover, what is the main take-away message? Basically, you're trying to sell your paper to the reader, so be brief and to the point. **Do *not* include any citations in the abstract.**

1 Introduction

For this assignment we were given a MNIST dataset which consisted of 60,000 handwritten letters that each were converted into 28 x 28 set of pixels displaying the color of each pixel. There was then a label for each letter that we were then asked to predict for a test set. We investigated different classifiers and looked at many resources to determine what the greatest predictors were. (Jonathan Milgram 2006)

2 Background

ated into the course of multi-layer perceptron. This is a Neural Network regression classifier that includes a hidden layer where the input is transformed into a more seperable layer. Many hidden layers can be added to the model and we used two different transformations to create the hidden data. The first is the identity which is a no-operation activation. It just returns the same out put as was inputed or $f(x) = x$. The second transformation we used was described in the paper tilte, and transformed the input into a rectified linear unit function.(Jonathan Milgram 2006) This transformation returns $f(x) = max(0, x)$ and was used in the paper as a baseline but we had not fully understood their further studys and so this became one of our better classifiers.

Looking more into the multi-layer transformations we came across Reading Checks with Multilayer Graph Transformer Netwroks (Yann Le Cun 1997). This paper went even more indepth into multilayer neural net training. It spoke a lot about its use in efficiently computing the gradients of the function as well as going into the mathematical side of the hidden layers or as they called it, the convolutional layer. We used some of their research and testing in Gradient Based Learning to determine the number of layers to use as well as which classifiers not to use.

3 Experiments

It should be noted that in the original data set if every pixel grey scale value were to be used as a feature to train for the model might result in a huge set of features. Having such set of features may cause a commnon phenomena known as "the curse of dimensionality", which implicates that the increase in the dimensional space resulted from the increase in the number of features might dilute the statisical significance of the final result.(Bellman 2003) That being said, in order to reduce the complexity of the final model in the hope of avoiding overfitting problems, a feature selection process was implemented for the experiments.(Hall 1999)

Feature Selections

According to Hall, feature selection process consists of the following steps:

- Starting point
- Search orgnization
- Evaluation strategy
- Stopping criterion

As introduced in the "Data Preparation" section, after some inspections through the data sets, it was observed that some pixels' grey scale values stayed the same all the time. Those points generally tend to have no significant impacs on the predictions according to the low-variance rule. Thus, those features could be the starting points. For this purpose,the backward elemination method was adpoted to reduce the number of features.(Hall 1999) The evaluation strategy is basically reducing the features that have zero variance one by one using the variance-based feature selection method implemented in the scikit-learn library. This process was repeated until the results showed significant decline in the models generated by the learning algorithms. For the purpose of determining the optimal set of features, the SVM algorithm was implemented because of its short running time.

Classifiers

For this particular classification problem, several models were implemented and tested out. Here

Super Vector Machines SVM serves as the baseline model for the experiments due to its simplistic nature. It is also used for finding the optimal set of features.

Multi-layered Perceptron

4 Results

Present the results of your experiments. Simply presenting the data is insufficient! You need to analyze your results. What did you discover? What is interesting about your results? Were the results what you expected? Use appropriate visualizations. Prefer graphs and charts to tables as they are easier to read (though tables are often more compact, and can be a better choice if you're squeezed for space).

Embedding Pictures

See the source code (`results.tex`) for instructions on how to insert figures (like figure 1) or plots into your document.

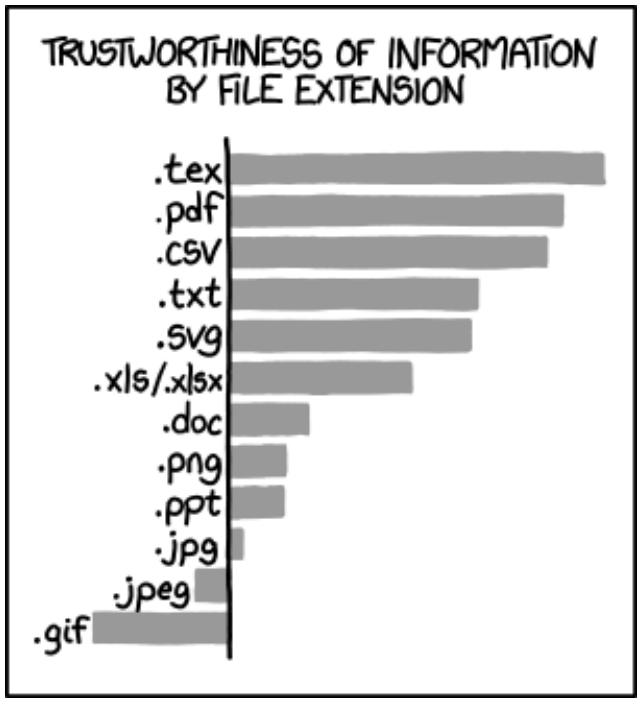

Figure 1: On the trustworthiness of LaTeX. Image courtesy of `xkcd`.

Creating Tables

Again, refer to `results.tex` to learn how to create simple tables (like table 2).

Column 1	Column 2	Column 3
1	3.1	2.7
42	−1	1729

Figure 2: An example table.

5 Conclusions

In this section, briefly summarize your paper — what problem did you start out to study, and what did you find? What is the key result / take-away message? It's also traditional to suggest one or two avenues for further work, but this is optional.

6 Contributions

Mutian wrote the SVD classifier as well as the feature selection code. Ryland wrote the MLP classifier code as well as investigated more into the testing and training sets. Ryland wrote the Introduction, the Background Information, the Citations, the Contributions, the Acknowledgements and the experiments. Mutian wrote part of the experiments, the results and the conclusion. Mutian was also big into researching what other people had done while Ryland interpreted those readings to work as classifiers.

7 Acknowledgements

This section is optional. But if there are people you'd like to thank for their help with the project — a person who contributed some insight, friends who volunteered to help out with data collection, etc. — then this is the place to thank them. Keep it short!

References

Bellman, R. E. 2003. *Dynamic Programming*. Courier Dover Publications.

Hall, M. A. 1999. *Correlation-based Feature Selection for Machine Learning*. Pearson Education.

Jonathan Milgram, Mohamed Cheriet, R. S. 2006. ?one against one? or ?one against all?: Which one is better for handwriting recognition with svms? *HAL archives*.

Yann Le Cun, Lon Bottou, Y. B. 1997. Reading checks with multilayer graph transformer networks. *Speech and Image Processing Services Research Lab*.