OPTICAL RECOGNITION OBJECTIVES

EXECUTIVE SUMMARY

In financial services one of the most daunting challenges facing the industry is embracing the seismic shift towards the analytical revolution. This has the potential for enormous impacts to the everyday jobs of people spanning various lines of business from the retail banking, mergers & acquisitions as well as sell-side investment banking. An example of such a technology is optical character recognition, which is a foundational aspect in the domain of computer vision.

Optical character recognition is a data analytics technology that is used to convert digital representations of characters and symbols, like those that would appear in a scanned document, to its native digital representation in the computer. Once the data is in the native representation of the computer, further automation and analysis can be conducted seamlessly across the organization.

For example, this technology could be used to automate the scanning and process of a deposit check for a bank teller, or to scan and classify thousands of documents made available via a data room during a merger or acquisition. Having the ability to automatically digest vast amounts of human produced content and transform it for digital and analytical processing would give any financial organization a substantial competitive edge in practically every area of its daily business and operational process.

RESEARCH DESIGN

For this problem we used a well-known, universally recognized and accepted database of hand-written digits composed of a series of 70,000, 28x28 pixelized, images that also contain an associated label that specifies the images true digit value. This database has been used in countless research studies, machine learning competitions (such as Kaggle), and academic classrooms throughout the world.

Our classification system uses the first sixty-thousand images in the database as a training dataset, and the last ten-thousand as our test and validation dataset. We will take the same sixty-thousand images and classify them with different approaches.

The first approach we will take is to establish our baseline results by using a standard random forest model on the trading set. This random forest will serve as our benchmark case throughout the rest of the study.

Once the benchmark case is solved for in terms of modeling accuracy and training-time, we will look to improve on it using a statistical procedure called principal component analysis. The big idea behind principal component analysis is to reduce the dimensionality of a dataset, making it easier to process, while preserving the variance in the data that makes each item uniquely identifiable. This technique is universally recognized as one of the most popular methods of optimization in this problem space.

TECHNICAL OVERVIEW

The model construction and benchmarking methods were conducted purely in the cloud, leveraging a pre-canned environment from a world-class provider of machine learning solutions, Google, called Colabratory. This cloud environment enables us to not only conduct research in a more efficient manner, while also allowing maximum reproducibility by taking out the variability of an individual desktops hardware configuration for the benchmark results. Additionally, we can push our research globally and allow any astute reader to reproduce our results for themselves in a sandboxed environment.

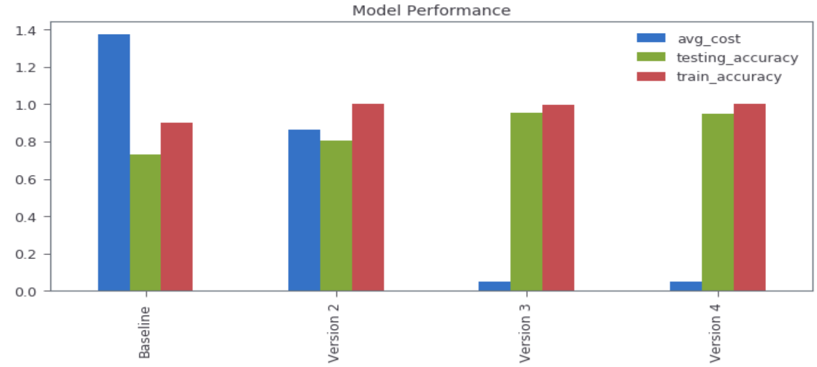
Additionally, for this research we also leveraged an industry-leading machine learning framework, TensorFlow, that is also produced by Google. TensorFlow gives us access to the same underlying technology that powers several of the most advanced analytical systems in production today.

Using this framework, we setup an artificial neural network algorithm for training our classification system. For this problem, we chose a Multilayer Perception (MLP), algorithm that is trained on the MNIST dataset with various parameters so that we can conduct research on the optimal training configuration, in which we reach maximum accuracy with minimal time spent on the algorithms I/O and computational heavy training phase.

At the heart of this study is the aforementioned Multilayer Perception algorithm, and three tuning parameters: learning rate, training epochs, and batch size. We will execute the "baseline" algorithm with the minimal specifications to get the algorithm to perform substantially better than the flip of a coin, which would be our worst case in terms of performance, as we would have simply wasted resources for absolutely no benefit. From there, we will tune the parameters based on a training time cut-off and attempt to find the equilibrium are of maximum performance in minimal time.

CONCLUSION

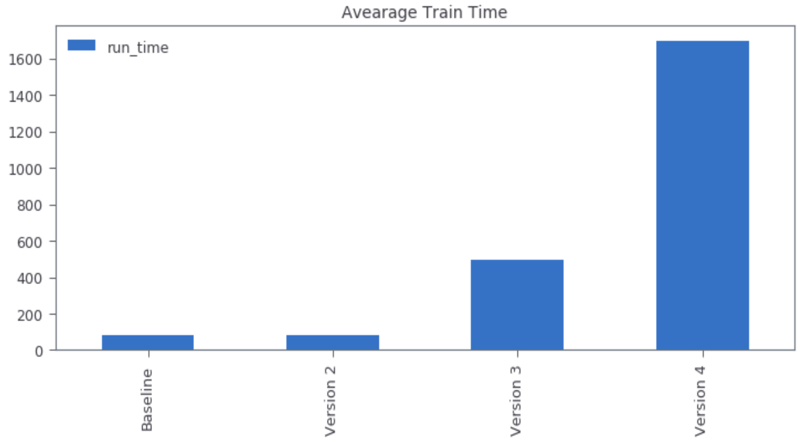
After successfully completing our trials and research, we should first define how we measured each characteristic of the model. For accuracy, we used the TensorFlow 'accuracy' metric, which is the frequency with which the model's prediction matches the test label.



The baseline model has a testing accuracy score of 73%, compared to the most robust model, version 4, which has a testing accuracy of 95%, which is an outstanding score.

We should note that the time sacrifice to get such an outstanding increase in testing accuracy, the model training time went up by a factor of 20 times. This is obviously an outlandish increase in run-time.

It turns out there is a nice equilibrium area with the parameters used for the third version of the model. For this version, we used a learning rate of 0.01 and 500 training epochs. This combination yielded an even greater training accuracy score, version 3 had .9557, while version 4 had .9476, at approximately 25% of the training cost of version 4. This was due to the increased batch size in version 3, even though model 4 had 10x more training epochs.



This research is by no means exhaustive or complete. We have merely scratched the surface on what is possible here with TensorFlow and some training data.

For additional information on how this research was conducted, please visit the Colabratory Notebook.