SYNOPSYS CHAMPIONSHIP 2016 PROJECT ABSTRACT

THE ABSTRACT IS A REQUIRED PART OF YOUR PROJECT

Bring your abstract with you to project check-in on Wednesday, March 16, 2016.

122-H30-E1 Sanjay Mohan

You should bring at least ten copies of your abstract with you when you come to the Championship. One copy should remain on display with your project during the Championship. You will want others to give to the judges. Your abstract should be written after you finish your research and experimentation and should include:

- Your project title, the full name(s) of all team members, and your school (all centered)
- The purpose of your project
- Your hypothesis or evaluation criteria
- A brief statement about the procedures and equipment you used
- Your results (analysis of data)
- Your conclusions

Type or print neatly using 10- or 12- point black type. Single space throughou
Center your project title, your name(s), and school.
You must use this form. Your abstract should be less than 500 words, and it should fit within the lines on this form.

Application of Artificial Neural Networks in Handwritten Text Processing

Sanjay Mohan
Los Gatos High School

In the past few decades, the distribution and usage of personal computers has increased exponentially. These computers rely on simple, intuitive methods of input. The current input standard of a typical laptop computer utilizes a QWERTY keyboard layout for text processing, while other variants include AZERTY, Dvorak, and alphabetical layouts. However, with letter keys in seemingly random positions in different keyboards, learning to type often takes immense practice; this practice is often integrated into multiyear elementary school programs. In other instances, keyboards may not be available or may be more impractical.

This project seeks to augment and thus improve upon the current keyboard standards by introducing a method of text processing through the touchpad, an almost universal feature of modern laptop computers. Touchpad text input is based on a simple extension of handwriting skill, nullifies the need for extensive keyboard practice, and offers other improvements on current text input.

This project relies on the use of feedforward artificial neural networks (ANNs), a method of machine learning designed to classify numerical inputs through various computations. Based on the neural connections of the brain, ANNs use sets of training data to adjust numerical coefficients in order to approximate an unknown function. In this implantation, a user draws on the touchpad; the pixel values of the resulting image are input into the ANN, a character classification is output, and that determined character is displayed in an output text field.

The software was designed in Python. The graphical user interface makes use of the tkinter package, and the ANN makes use of Anaconda Accelerate, a distribution of Python from Continuum Analytics that features low level optimizations to NumPy matrix operations. Training and testing of the software was focused on Arabic numerals 0-9 using the MNIST database of handwritten digits, a subset of a dataset released by the National Institute of Standards and Technology.

As ANNs are approximations of one-to-one maps from an input space to an output space, accuracy of output classification relies on consistency of input data. For this reason, before drawn images are input into the ANN they must be standardized to a common size and position. Some improvements that significantly increased classification accuracy included modifications to the standardization algorithm, artificial expansion of training data, and tuning of ANN hyperparameters.

The goal of this project was to achieve 95% classification accuracy on test images generated through simulation of typical usage of the graphical user interface. While consistent improvements to classification accuracy were achieved, 95% accuracy was not; the best performing network achieved 85% accuracy on the test set. Further improvements may be made through generation of a larger dataset of digits drawn through the graphical user interface; these digits will more closely resemble digits generated during actual usage. After this, the software can be generalized to alphabetic characters and even characters of other languages with a similar method. ANNs have great potential applications through integration with everyday activities.