Stanford Partnership in AI-Assisted Care

Lily Li January 12, 2018 Sample Research

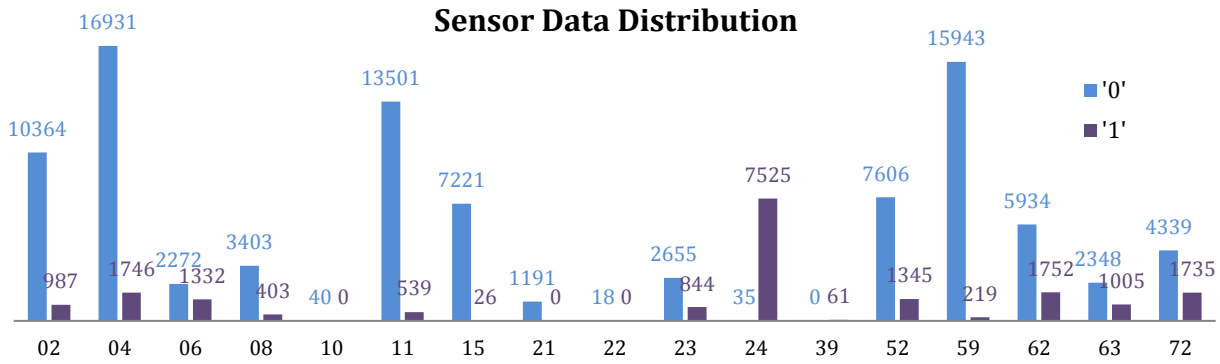
I. OVERVIEW

A 6-layered Convolutional Neural Network (3 convolutional layers, 1 flattening layer, and 2 fully connected layers) using the Tensorflow open source library was constructed to complete this research sample. This layering structure was chosen in order to have an output size of 28 x 28 x 6. After running this model three times on an Apple MacBook Pro (processor: 3.1 GHz Intel Core i5; memory: 16GB 2133 MHz LPDDR3; graphics: Intel Iris Graphics 550 1536 MB), the highest validation set classification accuracy was 96.9%. The entire dataset of files was divided into an 8:2 ratio of training and validation sub-sets respectively in order to ensure the highest accuracy possible. By using an 8:2 split for the dataset, a sufficient number of images are aside for validation while the largest possible number of images was used for training. An additional sample of 100 images from different sensors was set aside to ensure that overfitting does not occur when the model predicts new images.

II. DATA AND METHODS

In order to maximize the accuracy of the training process, the entire dataset of images was used for training, except for 50 '0' images and 50 '1' images which were saved for predicting purposes. In order to pre-process the *'.npz'* files, they were first formatted into arrays and normalized using the formula $\frac{p-imgMin}{imgMax-imgMin}$ (where *p* is each array value, *imgMin* is the minimum value of the array, and *imgMax* the maximum). Finally, the arrays were reshaped using the *resize()* method from the *scikit-image* library in order to obtain the desired dimensions of 128 x 128. This dimension was mathematically determined for the purpose of efficiently calculating the correct parameters of each layer during construction.

The following table was constructed to visualize the distribution of data in perspective to each sensor:



III. CHALLENGES AND SOLUTIONS

a. Unfamiliar syntax (python)

One of the major obstacles I faced in completing this research sample was my lack of python knowledge. Although I had learned basic python syntax through online classes[1] years prior, I had no previous experience with this language or knowledge about related applications and libraries. To complete this sample, I taught myself how to use Anaconda and Jupyter Notebooks. When constructing the model, I relied on a variety of online tutorials and GitHub repositories to learn machine-learning concepts and to modify existing open source code to fit the given prompt.

b. Unfamiliar concepts and libraries

Coming into this assignment with little knowledge about machine learning, I dedicated the majority of my time learning the concepts behind Convolutional Neural Networks[2], Tensorflow, Keras,

[1] see link for CodeCademy course information used as preparation for assignment
[2] relative articles used for research during this assignment found here and here

and AI in general, while trying to apply my knowledge by completing tutorials. Since I had very little time to construct my own model, I followed and modified tutorials as I researched. This resulted in multiple iterations of my model (see Results and Iterations).

c. Library Version Conflicts

Downloading certain libraries to be used for this assignment posed another challenge. I mistakenly downloaded Python 3.6, which was incompatible with Tensorflow and OpenCV, which were libraries that are incompatible with the latest version of Keras. To solve the problem, I created a virtual environment with Python 3.5 and its related libraries, which can be easily deleted or modified when errors occur. A peculiar problem I faced was when the IDE refused to recognize certain import statements. After much research, I found that OpenCV could not be used in conjunction with the 1.4.0 (current) version of Tensorflow; an earlier 1.0.0 version had to be used. Since Anaconda does not provide an official 1.0.0 version of Tensorflow, I installed Keras at the very end, which forcefully downgraded Tensorflow to 1.0.0 and allowed the compilation of the program.

IV. RESULTS AND ITERATIONS

a. SoftMax Regression Model

The first iteration of the model was created by following the official Tensorflow tutorial[3] on building a simple SoftMax structure. The training accuracy was very low: oscillating between 80% and 90% through multiple trials.

b. CNN Model with Two Convolutional Layers

The first attempt at building a convolutional neural network included 2 convolutional layers, 2 pooling layers, a dense and logits layer[4]. This model was not tested and discarded due to bugs involving the incorrect use of weights and parameters and version conflicts.

c. Keras Model

A Keras model[5] was attempted in order to simplify the code and efficiently train the data. This model was not tested due to issues relating to version conflicts and data importing.

d. CNN Model with Three Convolutional Layers

The last model constructed was a 6-layered Convolutional Neural Network[6] (3 convolutional layers, 1 flattening layers, and 2 fully connected layers) using the Tensorflow library. The training accuracy of all three trials came out to 96.9%.

V. FUTURE WORK

If more time were given, I would fix the bugs in all of the above models as well as research other neural network architectures. A pre-processing step I would have liked to implement was to crop the image before training. By using OpenCV, I hope to use its OCR ability to recognize the dispensers and to crop the image around it, so during training, there is less room for error. Furthermore, since I only had time to conduct very basic research on machine learning and Tensorflow syntax, I would spend more time following tutorials and code samples in order to create my own CNN neural network from scratch.

VI. PERSONAL REMARKS

Thank you for providing me with this opportunity to conduct a research sample on performance data with real-world applications. An assignment of this caliber has challenged me to code in Python, create neural networks, and apply concepts about machine learning. I am very excited to work alongside Ph.D. students and Stanford professors at the Partnership in AI-Assisted Care Lab in order to apply my knowledge in modernizing hospital technologies that will revolutionize healthcare.

---

[3] official Tensorflow tutorial on basic SoftMax regression found here
[4] official Tensorflow tutorial on creating layers found here
[5] unofficial tutorials on creating neural networks using Keras used found here and here
[6] unofficial tutorial on creating CNN model used as code sample for research sample found here