

Image Handling

To better handle the massive nature of the MURA dataset, I wrote a class to read in the image file paths stored in a CSV file and assign an arraylist of BufferedImage objects by accessing the database. This was further specialized to exclusively include those files found under `/res/wrist_train_image_paths.csv` so as to minimize excess data caching. The class works as expected and provides ease of access to any given image, and the approach can be duplicated in conjunction with the use of a wrapper class to ensure that training images are correctly assigned a positive or negative label.

Changes to Convolutional Layers

The structure of the layers has been changed to allow for variable amounts of stride; this will likely allow for more control over the final scale of the image in order to maximize detail while minimizing memory expense. The current arrangement of layers is as follows:

Convolution [$\sigma = 2$]



Activation [ReLU]



Convolution [$\sigma = 2$]



Activation [ReLU]

The final scale of the image is approximately 6.25% of the area of the input image, which appears to allow for significant data points to remain visible in the image but minimizes the amount of potential perceptrons for the feedforward network. It is important to note, however, that visibility to the human eye does not necessarily dictate the ability of a network to learn features of the image.



Python

I have been reading up on and working in Python using Spyder to better understand how code might be transferred from one language to another, making use of the data-based optimizations that Python offers. I successfully rewrote the kernel operations as they stand in the Java version to work in Python with minimal difficulty and apparent success. I hope to further restrict the domain of the project to Python, provided that I ensure a full understanding of the code necessary to build a functional network.

Next Steps

I have begun transitioning over to Python, employing examples from the Keras API to gauge hardware and software capabilities as well as potentially beneficial network configurations. I am now attempting to implement this API for use with the MURA dataset, which will require extensive image manipulation and format conversion. This will ultimately shed light on the potential accuracy of a fully structured network, provide insight into useful network structures, and influence the final formatting and nature of the CNN.