Convolutional Neural Network

For the third method, we created a Convolutional Neural Network to train the model to recognize the type of fruit from a given image. This network can be divided into two separate parts to clearly understand how it works and was implemented. The first part of the model works by passing the data into 2 convolution layers and then a maxpool layer. Then it is passed into a fully connected hidden layer that leads to the output layer.

The convolution layers work by applying filters to the image data to extract information from it. This works by multiplying a smaller matrix of numbers to the image dataset and striding through the image one by one both horizontally and vertically. The resulting matrix is a smaller one which reduces dimensionality and makes it faster and more feasible for the fully connected layer to operate. The convolution is done by all the initialized filters. The dimensions of the input to the first convolution layer is number of channels \* height \* width. The number of channels are usually three for the RGB values of an image but for our case it is the number of methods of normalization for the orientation gradients which are 4 in our case. When passed through the layer the dimensions become, number of filters \* height \* width. The feed forward layer equations are:

Where

For the back propagation we use the gradients of the functions.

And

For updating our algorithm we used a version of the stochastic gradient descent known as adam gradient descent. This estimates the value of the mean and variance using parameters β1 and β2 and uses these to update the value of the weights/filters/biases. The formula used for updating these rules is:

Where

The activation functions for the convolution, maxpooling and first hidden layers were ReLu while the output was the softmax function since we wanted the probabilities to sum to one for the output. The hidden layer had 100 nodes. They were all fully connected.

The time taken to train the dataset is 11.6 hours for 2 epochs. The batch size was taken as 32 datapoints from a training dataset of 48,905 images.

When the accuracy of the model was tested it came to 2% which implied that the model could not train properly. It will require a lot more epochs than two to train completely. This is because of high number of classes. This will need a much more powerful computer to train but we can still show some results.

