Problem Description:

Use a convolutional neural network to classify the faces and molecule data set. Employ synthetic data of reversed images to further aid in classifying the faces data set, and with the model used for molecule detection find the probability of a molecule in an electron microscope image.

Algorithm Description:

Convolutional Neural Network: Hierarchical neural network composed of input, output, and hidden layers. The hidden layers are made up of convolutional, pooling, fully connected, and normalization layers. The input is fed into the network which is then fed into the convolutional layers that learn the information in the data through matrix convolution to avoid a fully connected layer which would require many more nodes for connection. This is then fed into the pooling layer which creates a single layer that connects to the next layer, further minimizing the connections required in the network. Fully connected layers and weights are present in multi-layer perceptron’s and function the same in giving a weight of importance to each connected node.

Experimental Results:

Attached on back

When running the cnn it became clear that the results returned were somewhat inconsistent. Different runs could return different results with no changes made in between runs, but it also became clear that a particular score would appear more often than the others. As such the data attached and discussed is the most consistently appearing data that appeared over multiple runs.

Analysis of results:

The cnn performed well on the people in the wild data set, taking about half as long as the mlp to classify the data and returning a score just as accurate. Due to the nature of the data and the fact that a cnn is a less dense mlp it stands to reason that the score returned isn’t much better than that of the mlp. Having said that with less layers and less epochs, the best mlp had (100,100) hidden layers, the fact that the cnn preformed just as good is the crucial takeaway. With a less dense network, building more complex networks requires less computational power as less is needed to achieve the same results in a more connected neural network. Flipping the images improved the accuracy by three percent on one run, but on all subsequent runs it made no difference to the accuracy. This is most likely since the only images that benefit from more data are the ones already present in the training data, so images only found in the test set see no benefit and faces that have sparse representation in the training data see minimal improvement.

The molecule data set saw very high accuracy with quick model fitting even at fifty epochs. The object detection worked to varying results with some of the images getting high probability in their detection with others not breaking ten percent. The difference in noise in some of the images played a role, with many of the images with lower scores having more noise present in the picture. The images with very high probability on the other hand clearly had a molecule where the image found them, which indicates how well the object detection was able to find the trained image, in this case the molecules, in other images of different size.

Conclusion:

CNNs work very effectively for the same reason computation is less expensive. By having less connections, it allows similar nodes to connect to similar weights, allowing for the extraction of information along those similarities. This lack of density in the makeup of the network also results in less expensive computing, as there are less connections to compute, resulting in faster run times compared to a standard mlp. This importance placed on similarity allows for object detection that can that can be very accurate, and with a working model is very easy to code and understand what is happening. By focusing on the hierarchy of the data being processed through convolution, this allows for the network to pick apart the data and make what would be assumptions about similarity through max pooling and make connections out of the similar data. In turn, accuracy can be preserved and improved by trading off less for computational run time.