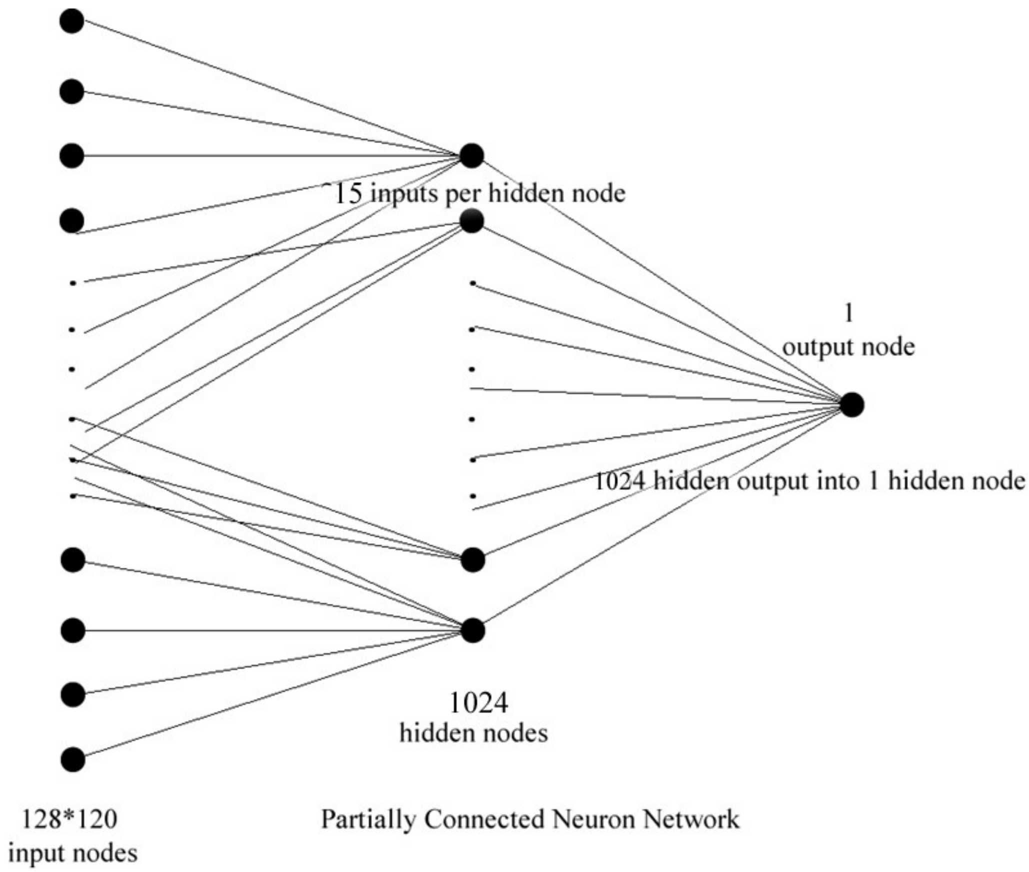
/* ECS170 Assignment #3: Connectionist Architectures and Ensemble Techniques

* Casey Wilson 997995396

* Anh Le 998513706

*/

Part I.



We choose this architecture because it reflects 1024 equally divided subregions of the whole picture. Each subregion is corresponding to one hidden node with 15 input nodes go into it(128*120/1024 = 15). The network takes in each pixel as input and will use back propagation to correct the decision error. Some hidden nodes are **not important** to the decision. They will have sigmoid function output very close to 0.5(all input with value 0, all black) and hardly change after each update. The **important** hidden nodes are in the center. They will change a lot because the white part of the face make the sigmoid outputs greatly vary. Since this is partially connected architecture, we can separate the important and unimportant parts (they're **independent**!) for training and thus obtain a higher accuracy.

Part II.

Average Accuracy: 0.9481481671333313 Standard Deviation: 0.02962963581085205

Average Accuracy: 0.9259259223937988 Standard Deviation: 0.054934815068409436

Average Accuracy: 0.9592592597007752 Standard Deviation: 0.013858007549404607

Average Accuracy: 0.9333333373069763 Standard Deviation: 0.04319223709102788

Average Accuracy: 0.9703703761100769 Standard Deviation: 0.014814826846127237

Average Accuracy: 0.9777777910232544 Standard Deviation: 0.013857982061379065

Average Accuracy: 0.9148148179054261 Standard Deviation: 0.03989010029492023

Average Accuracy: 0.9629629731178284 Standard Deviation: 0.020286030321470257

Average Accuracy: 0.9703703761100769 Standard Deviation: 0.01888526256647399

Average Accuracy: 0.9444444417953491 Standard Deviation: 0.020286030321477265

Part IV.

Each hidden node weight is representative weight of its subregion. In other words, the weights of 15 pixels in one subregion can be grouped as calculated as one. The input nodes pass through the hidden node should have similar characteristic due to the nature of the adjacent input pixel. They can have different randomized weights at first but as the training go on they will have value very close to each other and their hidden node. Hidden node weight changing scheme defines the importance of that subregion. If that subregion is not important, weight won't change much after many iterations. In short, a neural network which makes decision through hidden nodes gives better result than one that making decision directly from input nodes.