Report

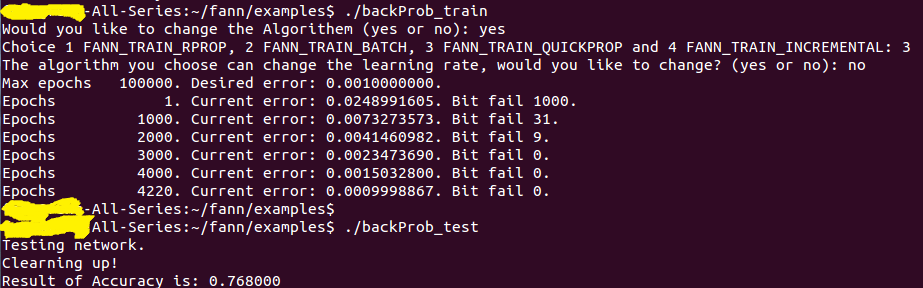
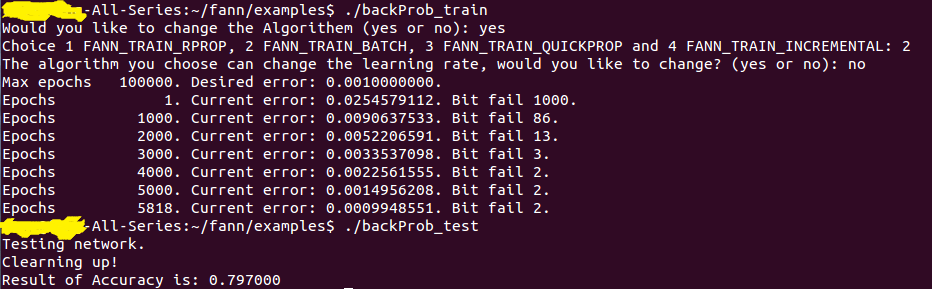
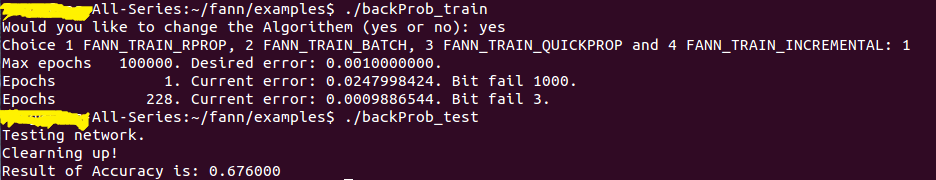
Purpose:

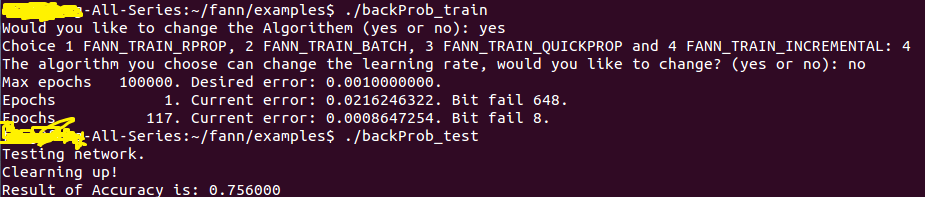
In this project, we are told to recognize the handwritten digits. We need to use FANN as the tool to do that so. The given files are a set of training and testing datasets with the information about the handwritten digits and its corresponding value. Then, those information can be trained and used to guess new handwritten digits.

Description:

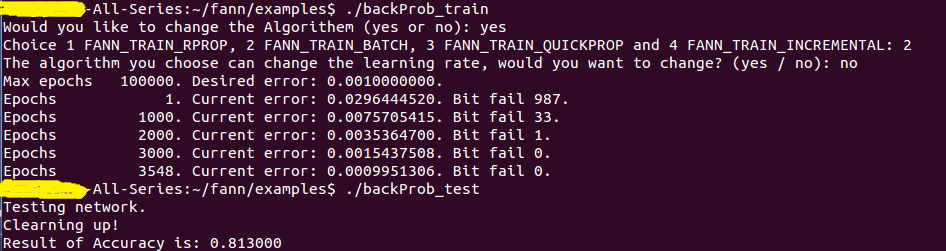
In order to accurately recognize the handwritten digits, the accurate rate must be as high as possible because that number tells how likely that the training data has been evaluated. How to increase the value of accurate rate requires knowledge of how to set the factors of the FANN.

Different parameters or algorithms for this project vary a lot. For instance, the size of the ANN can’t be too small. The experiment with three layers containing only three neurons for each layer does not tell us anything about the new handwritten digit, which means the accuracy in this type of ANN would be small, and experimentally proved (about 23 percent accurate). How large the size of the ANN should be depends on experience. After training with different size of the ANN (about 100 or 150 neurons for each layer), with different algorithms, RPROP, ATCH, QUICKPROP, and INCREMENTAL, desired rate 0.001, learning rate 0.7, the size of the ANN in this project should be 100 neurons for each layer with 3 hidden layers. The good one with BATCH algorithm has accuracy about 80%, which means the error rate is 20%. However, with the same setup, the probabilities from two different times does not have to be the same as the pictures given. That could be FANN trains little different in different training processes. Whatever reason is, the accurate rate would be closed to each other with the same setup.





After testing a few algorithms and some other factors, it is a need to increase the size of the ANN a lot. After changing the values of the hidden layers to 1000, the accurate rate improves by above 1 percent (it took about 4 hours to complete the training process).



In conclusion, there are many factors to change that can improve the accurate rate. Or probably, some factors work in a particular algorithm, in a particular size of the ANN with a particular learning rate.

How to use:

All files need to be located in the example of the fann folder. **This project is done in the Ubuntu system**. Both of the training file and the testing file are in the zip file. There is also a make file in the zip file that help to create the train and test executable files. After all files have been located properly, use make to generate the executable files. Moreover, remember to put both the training and testing dataset in the same directory as other files. User is allowed to hard code to change the size of the ANN, desired rate, and other values. The algorithm and learning rate can change during execution time.