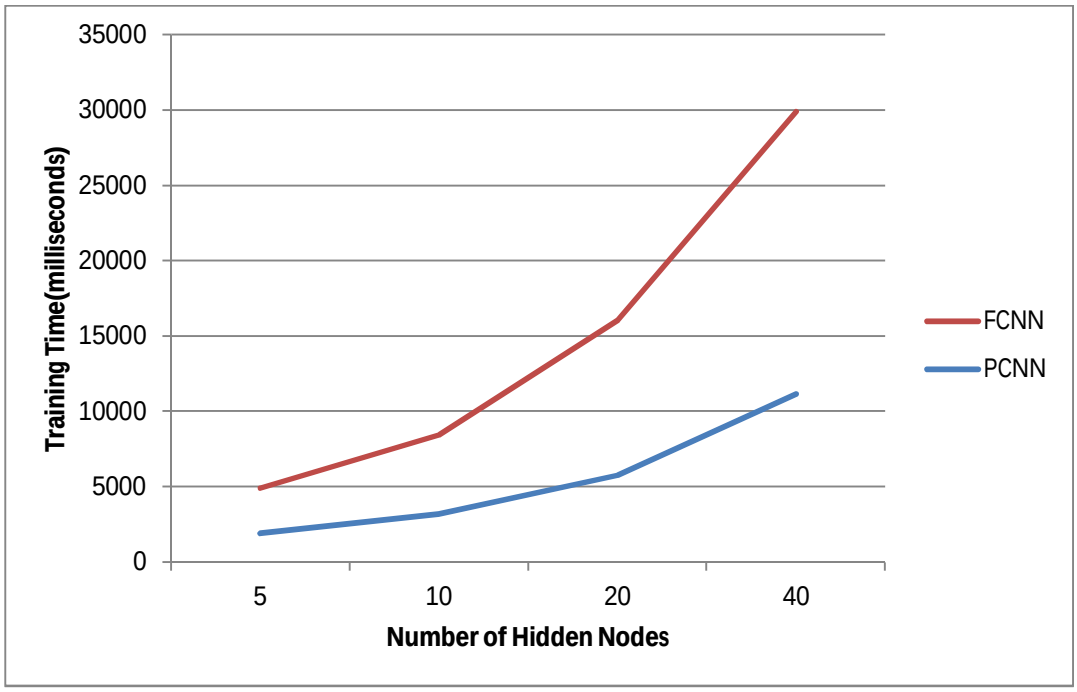
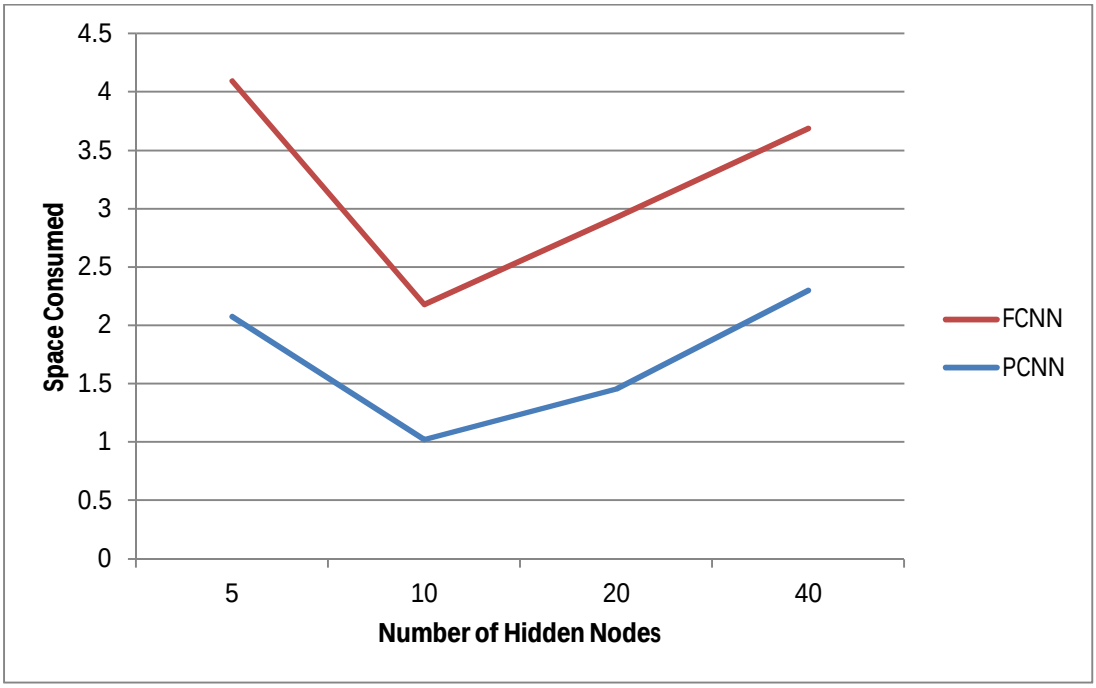
Neural Network Project Submission 04:
Improvements in ANN Performance
Submitted by Raj D. Devrukhkar

In this part of the project we will try to increase the performance of a Neural Network by converting it into a Partially Connected Neural Network (PCNN). A PCNN is a neural network which has the same architecture as the FCNN but with less number of connections between the nodes. In an FCNN, each node of one layer is connected to every node in the other, whereas in PCNN, some of the connections are dropped. To create a PCNN, we usually start with the FCNN and eventually drop connections that are not that important, during the training period. Since we drop connections during the training period, the training time reduces, since once dropped, the network does not have to perform weight updates on this connection saving time and space. Also once the training is done and all the not so important connections the dropped, the calculations to find a result of an input will be reduced since we will have less connections and hence less calculations (for sum for activation function) as compared to an FCNN. However, before we can say that using an FCNN is faster or consumes less space, we have evaluate as to the amount of time and space that the algorithm consumes to check if a connection should be dropped or not. The resources consumed by the algorithm should be less than the total resources saved by the conversion.

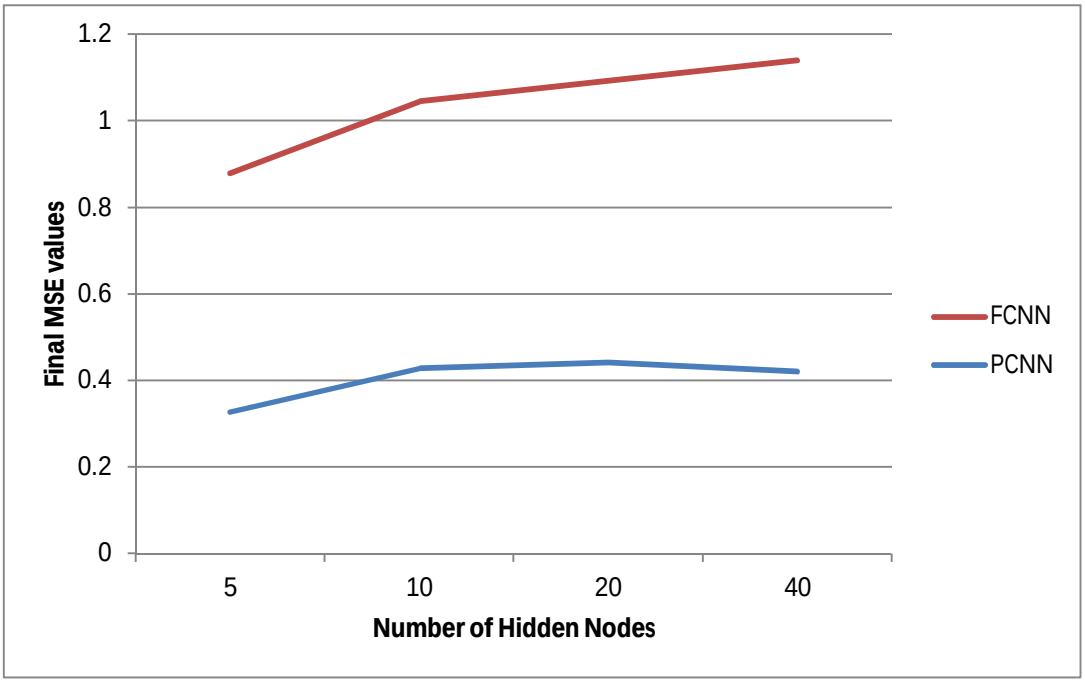
We will be using a simple method to evaluate if the connection between two nodes is important or not. After every sample has been used for training, out algorithm will check if any of the weights are less than the specified threshold. If any of the weights are less than the threshold then they are dropped from the network. (Note, there are some methods that regain connections, one of these methods is explained in brief in the later part of this report). Following are the results produced on the Liver Patient Database [1].



Graph 1: PCNN vs. FCNN w.r.t Training Time



Graph 2: PCNN vs. FCNN w.r.t Space Consumed



Graph 3: PCNN vs. FCNN w.r.t Final MSE values

As you can see from the graphs shown above, PCNNs give better results as compared to FCNN. One notable fact to see would be from graph 1. One can easily see that in each case PCNN takes less time as compared to FCNN but one should also notice that the slope of the FCNN curve is greater than the PCNN's curve. Implying that with increase in the number of hidden neurons, the increase in training time is higher in FCNN than in PCNN.

From graph 2 we can only infer that PCNNs consume less space as compared to FCNN, the slope of the curve is more or less the same so the rate in change seems to be similar.

We can see from graph 3 that even the mean error between the target and the actual output has decreased for this dataset. This is an unexpected result and hence I am specifying that this result is for this dataset. Evaluating the MSEs of other datasets on PCNN would provide a better understanding or generalization of this behaviour. This result seems odd because as we remove certain connections, the resultant output value shifts a bit because of a few missing values in its network sum. But we are trying to get the right output by adjusting the weights. So if our process of adjusting the weights is trying to get the output closer to target then the result of these missing values would change this shift towards the target. But we do not know if the shift happens towards the target result or away from the target result. But since removing these not so important connections actually reduces the error in the network then I would assume that they actually shift in the direction of the target output; which is exactly what we want. Hence, PCNN not just save resources but also gives a better result. However, as mentioned before, this may be a characteristic of this particular dataset and hence we cannot generalize this behaviour.

The graph shown above give a good understanding of the powers of PCNN with respect to resource consumption, however, they do not tell us if the results provided by the network are accurate or not. Graph 3 gives some indication, but it is not enough to validate that the PCNN gives better results. To check if PCNN gives us better or at least similar results to FCNN, we should test the network with new dataset. Unfortunately, the program that I have written for PCNN has some bugs and for now I cannot output the confusion matrices and hence cannot evaluate the accuracy of the results.

There are other methods which can be used to remove connections and ensure that the resultant network still gives an accurate result. One of those methods is described in [2]. The authors in this paper use the idea of plasticity from Biological Neural Networks and apply it to Artificial Neural Network. Plasticity defines the measure of importance of a connection to the final output; if the importance measure is less, then the connection should be dropped. The authors first create a fully connected neural network and then prune connections depending on the value of the plasticity measure. Note that in this method, a connection once dropped can be retained later. To check if the connection is relevant for the output, calculations are made for each connection and hence, the overall time and space consumption is high. Hence, concept-wise this method works but defeats the purpose of increasing the performance of the network. Hence, the method to create a PCNN should be carefully evaluated to ensure that they actually give you the results that are needed.

REFERENCES:

[1] Bendi Venkata Ramana, Prof. M. S. Prasad Babu and Prof. N. B. Venkateswarlu, “A Critical Comparative Study of Liver Patients from USA and INDIA: An Exploratory Analysis”, International Journal of Computer Science Issues, ISSN :1694-0784, May 2012.

[2] Piazza, F.; Marchesi, M.; Orlandi, G.; Uncini, A.; , "An algorithm for dynamically adapting neural network topologies," *Circuits and Systems, 1991. Conference Proceedings, China., 1991 International Conference on* , vol., no., pp.56-59 vol.1, 16-17 Jun 1991.

[3] Rajen Bhatt, Abhinav Dhall, 'Skin Segmentation Dataset', UCI Machine Learning Repository.