

CSC447/547: BPNN Forest Fire Predictor - Questions

Luke Meyer, Savoy Schuler, Steven Huerta

February 24, 2016

How well does your network train?

Our team feels that our Backpropagation Neural Network (BPNN) trains fairly well. Our team implemented a momentum term that reduced the training time. This makes sense, as the momentum term causes the gradient descent to skip over narrow local minimums that may add training epochs or cause the descent to terminate inside the local minimum. We found that training was reduced by a few dozen to a few hundred epochs with the momentum term. Reading additional literature, it may have been helpful to also add a local adjusting learning rate to curb aggressiveness in the descent. However, we could not implement this in time.

BPNN training is always subject to the dataset it is training on. For example if we were to apply our BPNN to classify a 4×6 binary representation of characters (which we did), it is able to classify rather quickly, taking, on average, less than 300 epochs to train on a set of 8 characters with a minimum error threshold of .001. Each character can be classified as they appear distinctly different. Forest Fire severity is much different, as conditions in two sample of data may very similar values for inputs, but the severity of the two are very different. Classifying these two samples to a small error threshold becomes difficult or impossible for this case.

The difficulty in classification became very apparent when training, testing, and cross-validating. For example, we trained on the Black Hills data set with the associated parameter file. It would not reach the min error, and terminate for reaching the maximum epochs or getting stuck in a minimum. Testing the weights generated through training provided very promising results, as only one or two of our answers were incorrect. Cross-validation was much different, providing a success rate around 40%. This does make sense. Our testing confirms that we can recognize patterns on which we have trained. The cross validation informs us that we are usually incorrect recognizing patterns of fire severity. We did find that the data for the Northwest trained faster (smaller set) and classified with greater accuracy than the Black Hills data. The cross validation for both sets were both similarly inaccurate.

What is the impact of network topology (i.e., changing the number of hidden layer nodes) on training?

In our observations, there is a direct and inverse correlation between the number of hidden layers and the mean square error (MSE), however it is not linear.

However, there reaches a point for each data set where the correlation begins acting asymptotically and the MSE difference between layers becomes marginal. It is interpreted that this is because we are converging on the lowest maintainable MSE for each data set.

Additionally, adding more nodes within the hidden layer did not improve the classification. We are sure that there likely exists an optimal solution for the number of nodes within the layer, but this is dependent upon the data set being classified, and would not generalize well to other data sets. The example of the wolf, granny, and woodsman classifier provides a good example of why three nodes within the hidden layer would provide the best conditions for classification. However, with PDSI and area burned serving as the data, there does not seem to be an easy guess for how these pieces may be grouped together. Perhaps we count the number of summers and winters in the sample size and make that the number of nodes within the hidden layer. We could additionally add another for the area burned. We do not know, however, how the data is being classified, and we are left with trial and error.

How well does the network generalize from training data to testing data?

As discussed previously, generalizing from training data to testing data, our network retains, on average, the same mean square error as what was converged to during the training process. The MSEs of the testing results were reasonably within the same power difference of the MSE produced during training. This allows us to infer that our network maintains reasonable consistency between training and testing phase.

The cross-validation testing greatly reduces our confidence in this network being used to predict fires. The removal of the sample to be tested in cross-validation removes the opportunity of training on that piece of data, preventing our network from learning that sample as part of the set. This reduced our accuracy from around 90% to 40%.

Lastly, we attempted to try our training on one set to test on the other set. Results were extremely poor, even after adjusting the classification parameters to be the same. This does make sense, as the two environments have different trees, different soils, different understory vegetation, as well as moisture. We likened it to generalizing what a dog is from only looking at poodles. It generalizes very poorly.