Dallas Alberti 11/22/2020

Canvas discussion 5

Neural Network Hidden Layer Sizes

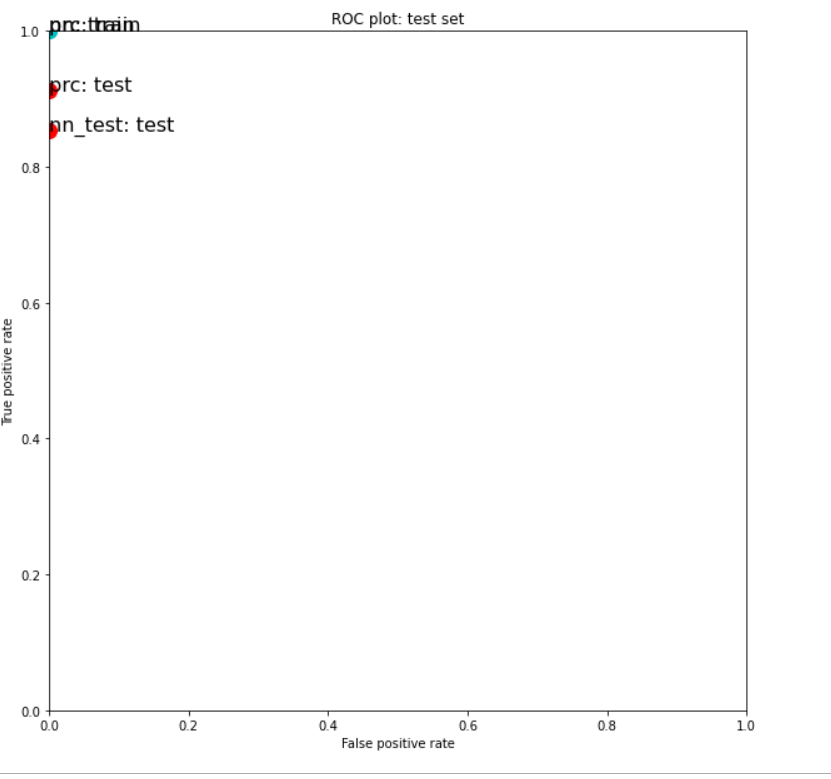
Histogram of oriented gradients (HOG) feature set.

fd, hog\_image = hog(img\_raw, orientations=8, pixels\_per\_cell=(10, 10),

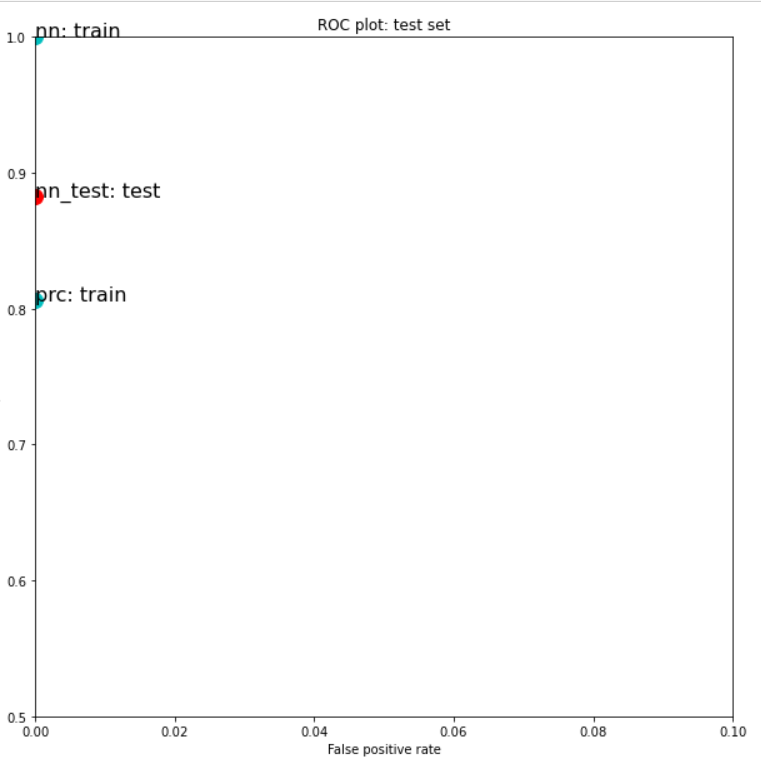
cells\_per\_block=(3, 3), visualize=True, multichannel=True)

The following images show changes to the layer sizes of a neural network model. ROC plots are not very informative so a table of changes made is shown at the end of the document.

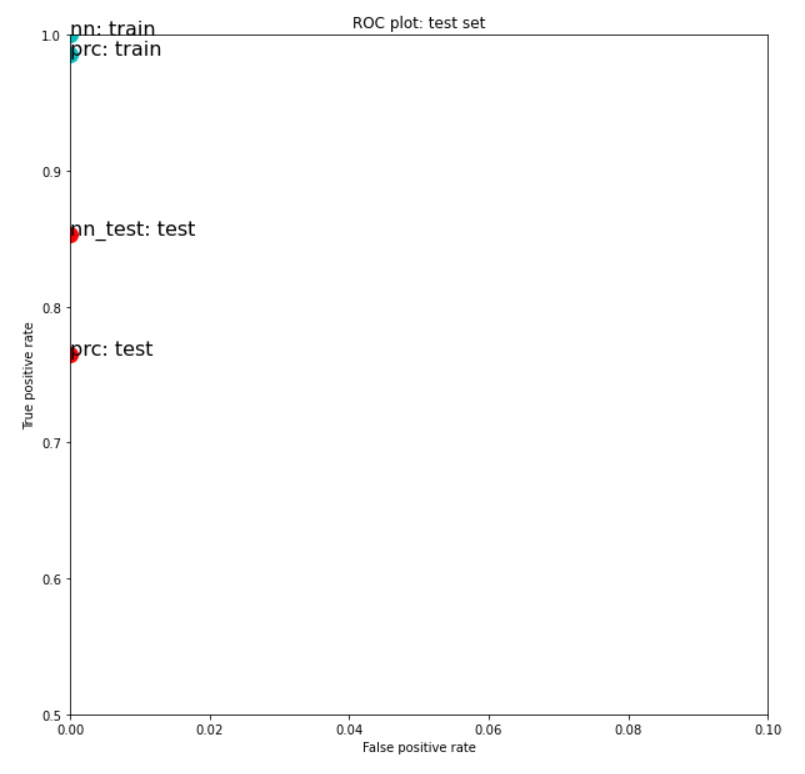
1. Default IMG 1 - Hidden layers = (100) .{'Pos': 34, 'Neg': 1656, 'TP': 29, 'TN': 1656, 'FP': 0, 'FN': 5, 'Accuracy': 0.9970414201183432, 'Precision': 1.0, 'Recall': 0.8529411764705882, 'desc': 'nn\_test', 'set': 'test'}

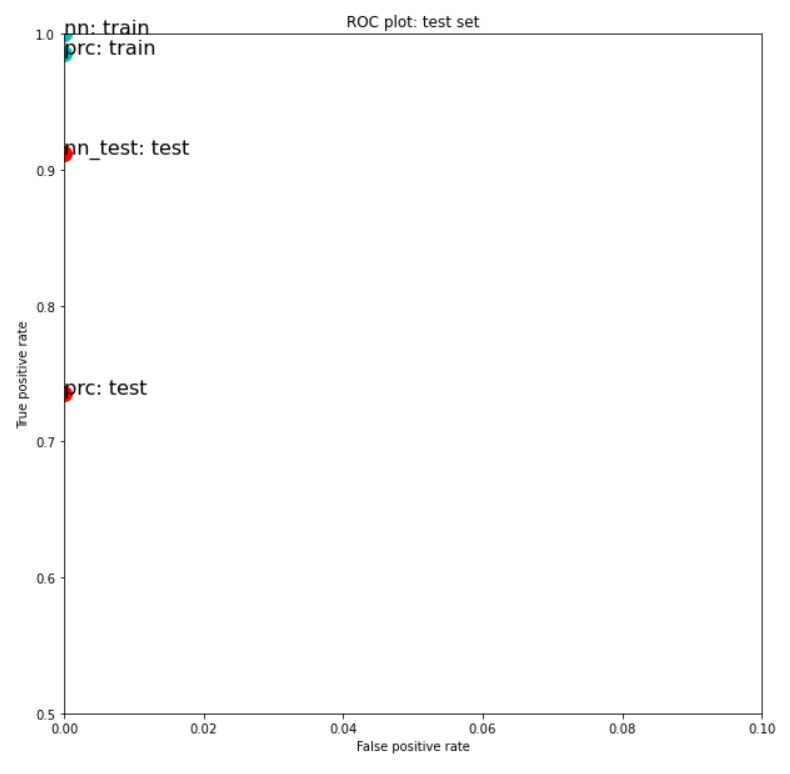


1. IMG 2- Hidden layers = (100,100) {'Pos': 34, 'Neg': 1656, 'TP': 30, 'TN': 1656, 'FP': 0, 'FN': 4, 'Accuracy': 0.9976331360946745, 'Precision': 1.0, 'Recall': 0.8823529411764706, 'desc': 'nn\_test', 'set': 'test'}

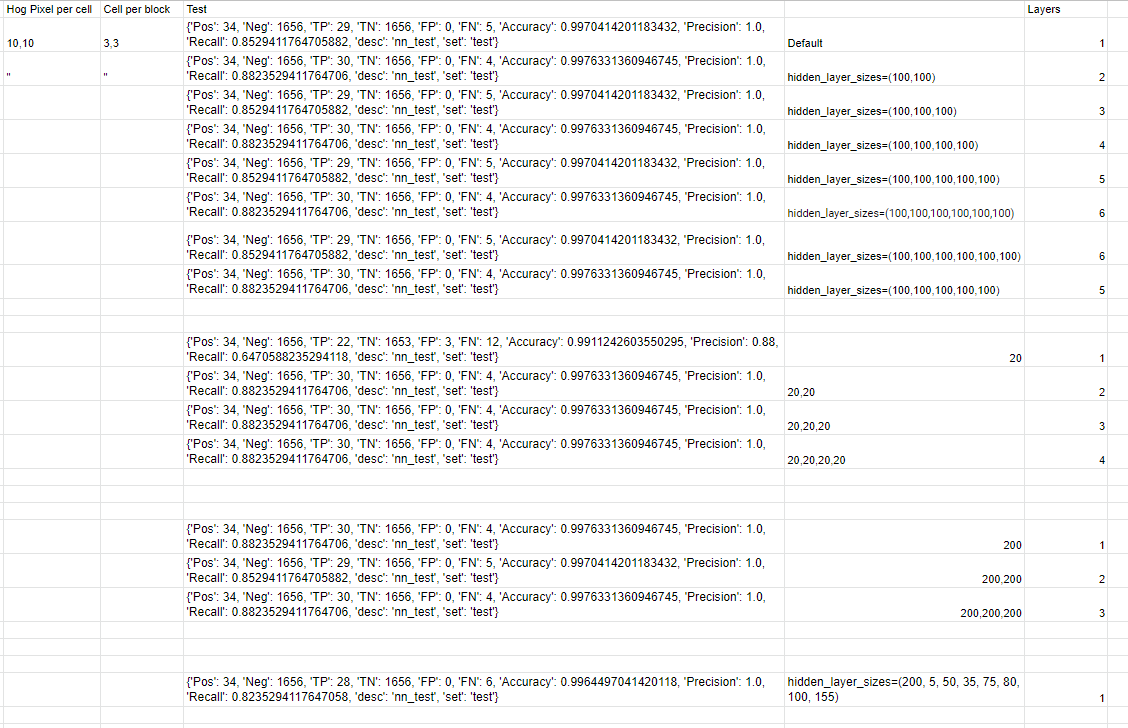


1. IMG 3 - Hidden layers = (100,100,100) {'Pos': 34, 'Neg': 1656, 'TP': 29, 'TN': 1656, 'FP': 0, 'FN': 5, 'Accuracy': 0.9970414201183432, 'Precision': 1.0, 'Recall': 0.8529411764705882, 'desc': 'nn\_test', 'set': 'test'}



1. IMG 4 - Hidden layers (100,100,100,100) {'Pos': 34, 'Neg': 1656, 'TP': 30, 'TN': 1656, 'FP': 0, 'FN': 4, 'Accuracy': 0.9976331360946745, 'Precision': 1.0, 'Recall': 0.8823529411764706, 'desc': 'nn\_test', 'set': 'test'}

These ROC plots were not very informative and even a change to show Accuracy vs Recall would not have been that enlightening. I did however discover some patterns in my tests that are worth showing in this excel spreadsheet image.



IMG 5- All changes made. Alternating FN rate of 4 and 5 depending on the number of layers present.

Summary of findings.

After making several changes to the layers of the neural network model, I started to notice that the False Negative rate was fluctuating between 4 and 5. Each layer added with 100 neurons seemed to be causing this alteration such that the default single layer (100) was producing 5 FNs, the second layer(100) was 4 FNs and so on repeating. This led me to believe that an even number of layers would always produce ‘better’ test results. Unfortunately I reran the program a few times at 6 layers expecting to replicate this result however, as seen in the chart, this was reversed after a re-train and the even number of layers began to produce 5 FNs while the odd layers produced 4 FNs. This was not observed when using 20 neurons in each layer. I restored this effect by bringing the number of neurons up to 200. I am hoping that we can discuss the reasons behind the ‘alternating effect’ that I have recorded here.