Multi-class Classification and Neural Networks

Aquino, Patrica Rose
Basallote, Lawrence Andrew
Carag, Stephanie
Jacinto, Dan Emanuel
Lunasco, Jan Osbert

I. Introduction

In Biology, a neural network is an interconnection of neurons which transmit patterns, signals or data from one neuron to another. The dentrites are the ones receiving the input signals and the axons are the ones releasing the output signals. In Artificial Neural Network (ANN), the basic model of the bilogical neural network is made into a simple diagram which can be applied for computer understanding or learning. Like the biological model, the ANN also consists of nodes or ”neurons” interconnected to one another. There are inputs where data or patterns are shown. The hidden layers are synonymous to the biological dendrites which processes the input informaton. The hidden layer then fires the processed output to the output layer which is synonymous to the biological axons.

II. Data and Results

A. Code

```
import matplotlib.pyplot as plt
#Compare estimate, yHat, to actually score
plt.bar([0,1,2], y, width = 0.35,
alpha=0.8)
plt.bar([0.35,1.35,2.35],yHat, width =
0.35, color='r', alpha=0.8)
plt.grid(1)
plt.legend(['y', 'yHat'])

timeElapsed = endTime-startTime

import matplotlib.pyplot as plt
plt.plot(weightsToTry, costs)
plt.grid(1)
plt.ylabel('Cost')
plt.xlabel('Weight')

import matplotlib.pyplot as plt
testValues = np.arange(-5,5,0.01)
plt.plot(testValues, sigmoid(testValues),
linewidth=2)
plt.plot(testValues, sigmoidPrime(testVal-
ues), linewidth=2)
plt.grid(1)
plt.legend(['sigmoid', 'sigmoidPrime'])

print cost1, cost2
print cost2, cost3
numericalGradient, 2*x
numgrad = computeNumericalGradient(NN, X,
y)
grad = NN.computeGradients(X,y)
```

B. Results

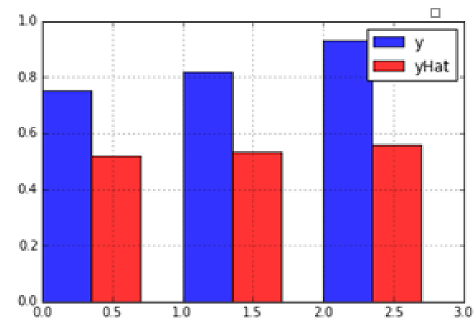

Fig. 1. Compare estimate yHat

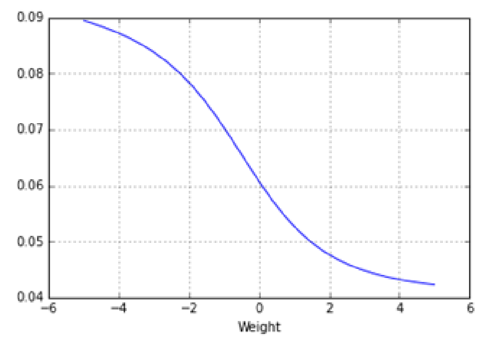

Fig. 2. Plot of the Cost vs Weight

Time elapsed: 0.025

Cost1 and Cost2 = [0.40] [0.70]

Cost2 and Cost3 = [0.70] [0.45]

Numerical Gradient: (2.999, 3.0)

numgrad = array([0.01247989, -0.01838624, -0.00560211,

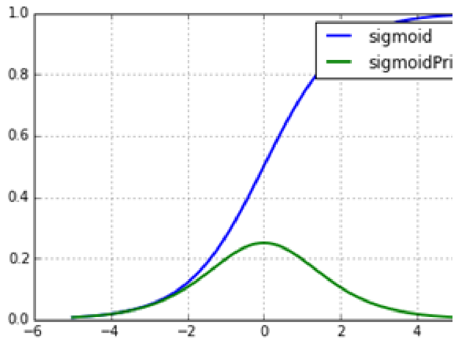

Fig. 3. Plot of the Sigmoid and sigmoidPrime

0.00574306, -0.00828112, -0.00246826, -0.05745107, -0.05743181, -0.06648923])

grad = array([0.01247989, -0.01838624, -0.00560211, 0.00574306, -0.00828112, -0.00246826, -0.05745107, -0.05743181, -0.06648923])

III. Conclusion

In this experiment, we applied the algorithm of neural networks again. However, this experiment is quite different from the previous one. The previous experiment was more of a logistic regression in a larger scale. However, artificial neural network is a completely different algorithm since it is designed to process like the neural networks of the brain. It can be observed that the accuracy for validation becomes higher if there is more data in the training data. The number of samples for validation do not matter, but if there are more samples, then the accuracy is much more reliable. During testing, it can be observed that the accuracy became smaller than in validation. This is because the testing data can be a lot different from the training data. Validating the neural network is important because this avoids the overfitting of the neural network. If the machine detects that the neural network is overfitting during the validation process, then it corrects the network. However, during testing, even if the previous neural network only fits that data, the machine will not correct the neural network. It will output its prediction based on the neural network produced previously.