**Logistic Regression in Python and TensorFlow**

Michael Stewart

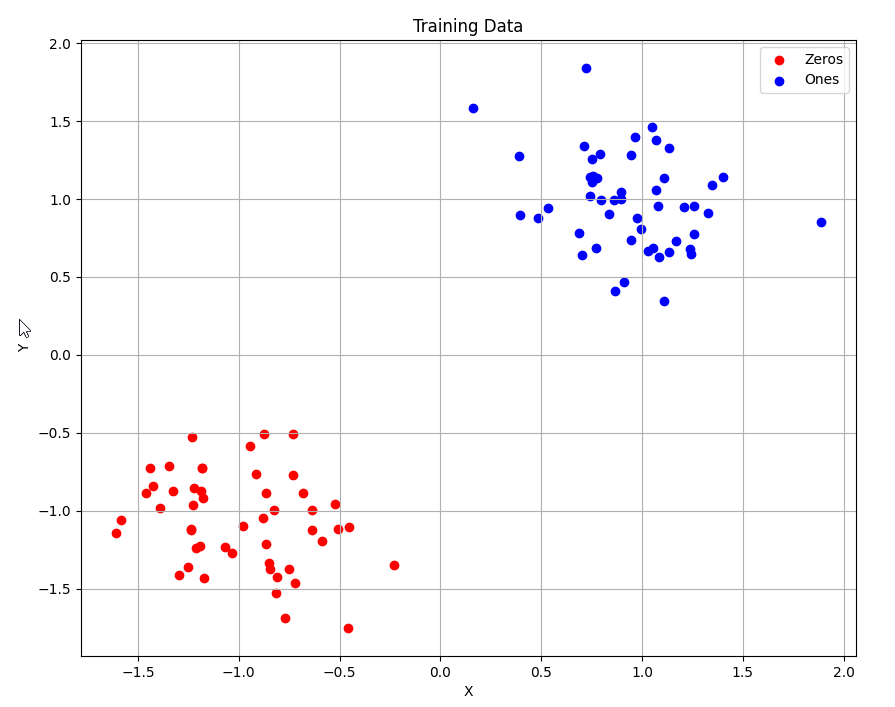
Colorado State University Global

CSC580-1: Applying Machine Learning and Neural Networks - Capstone

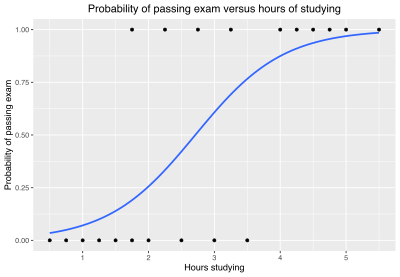
Professor Isaac Gang

October 12th, 2024

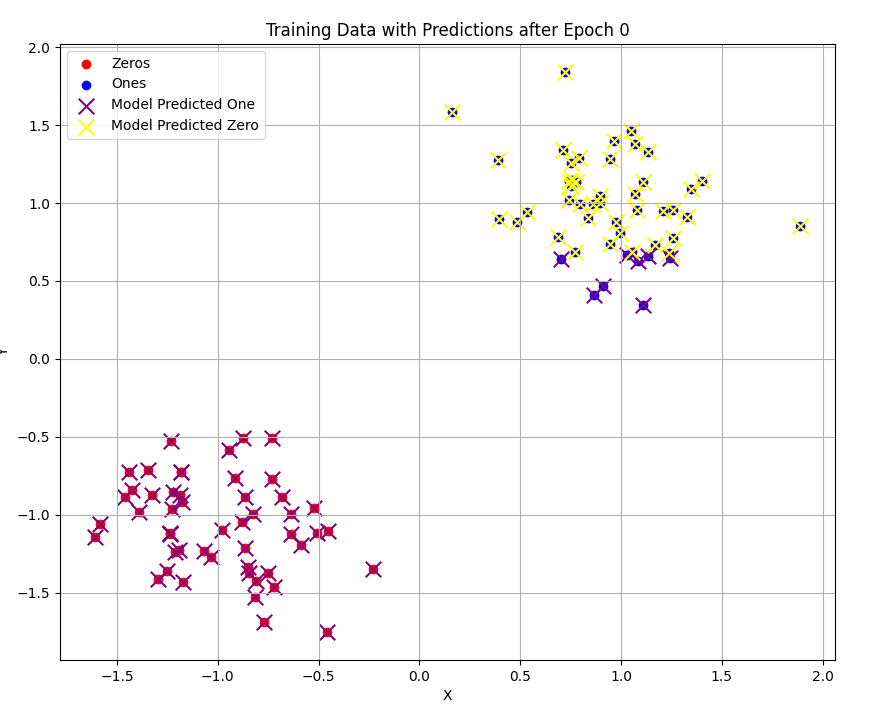
**Logistic Regression in Python and TensorFlow**

****

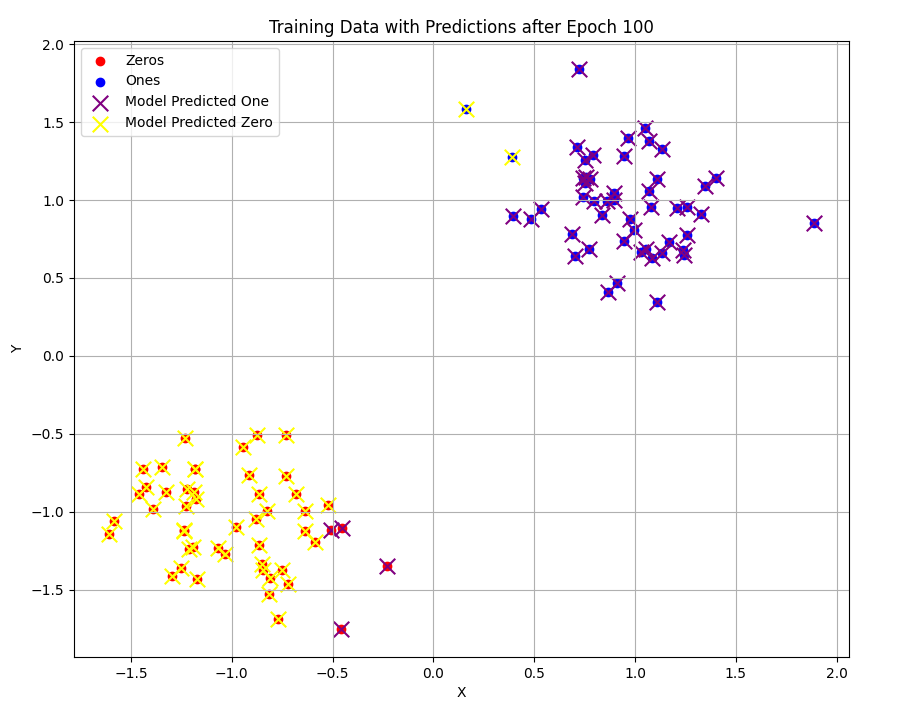
Logistic regression is the method of applying a logarithmic function to a dataset to predict a binary classification. In this dataset, we have two Gaussian formations. We have one Gaussian formation around the point -1,-1 and the point 1,1. To apply logistic regression with TensorFlow on this dataset, we have to cover a few things first. Normally, logistic regression has only one variable and it will look like the graph below:



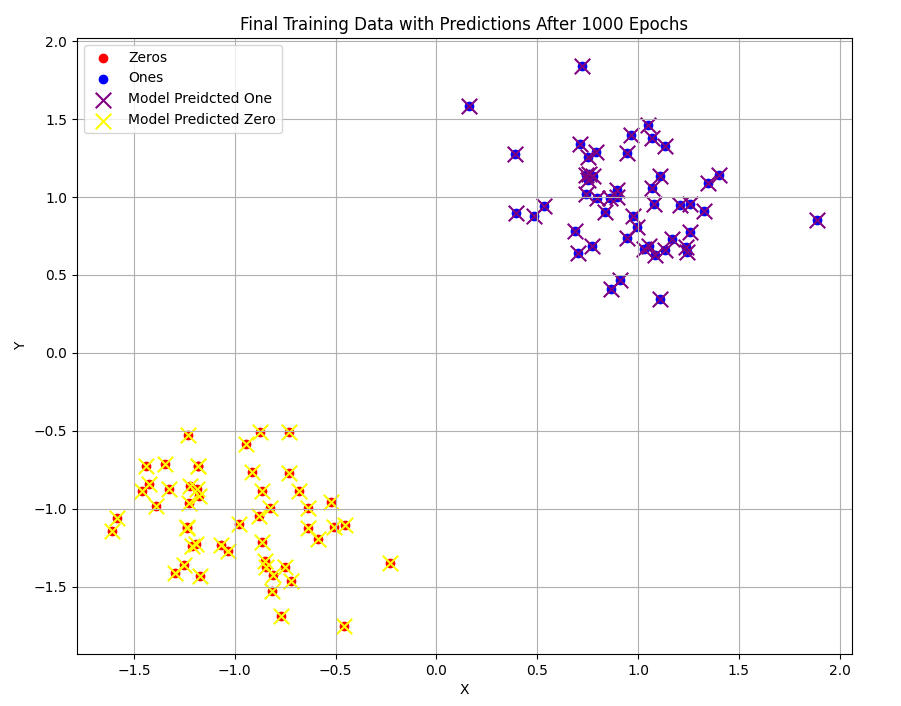
You can see above that the black dots indicate the dataset of hours studying versus how they did on the exam. The blue line is the logistic regression fit and it shows how the variable “Hours Studying” equates to a confidence that the student will pass the exam. In our example, we have two binaries, but they each have XY coordinates. How will we be able to make a fit of data with an extra dimension? We will make a 3D graph that will have the plot points on their respective planes with a sheet having the shape of a logistic regression in between them.

****

This is the graph of the predictions after epoch 0. This means that the model is only trained a small amount. You can tell that this model is not trained at all because it is completely misclassifying both clusters. It believes that the zeros cluster is the ones cluster and vice versa. As epochs continue, it uses a loss function and optimizes it over time.

****

This is about 1/10 of the way through training. You can see that the model is still classifying some of the nodes incorrectly, but it is much better than the first epoch. You can see that the model is struggling to identify the nodes that are close to X = 0. This means that the model in this training session has noticed that the X value for the clusters is a strong indicator of which class they belong to. Then as small improvements continue, it will notice the strong correlation with Y values and their clusters as well.

****

**Epoch 0, Loss: 159.6503448486328**

**Epoch 999, Loss: 1.7942259311676025**

Here is the final model predicting every node correctly. The model successfully completed this after about 200 epochs, but the loss function still decreased a significant amount throughout 1000 epochs. Listed below the graph are the values for loss and epochs. You can see the big difference between the graphs and their loss functions. This means that our loss function fits the model really well. If our model was not predicting the classes correctly but the loss function was doing really well, that would mean we should revise our loss function and figure out a better way to indicate how wrong the model is.

**References**

Contributors to Wikimedia projects. (2024, September 18). *Logistic regression*. Wikipedia. <https://en.wikipedia.org/wiki/Logistic_regression>

Ganegedara, T. (2022). *TensorFlow in action*. Simon and Schuster.