CPSC 393 HW 2: Spam Email Classification NN

Daniel Dinh

Introduction

Spam emails are a common occurance, however in more recent years they've become a new tool for hackers to use to find potential weaknesses in companies's and individuals online presence. Therefore, it is imperitive to potentially find a way to accurately classify whether or not an email is spam. This can be done by using a prediction model such as a Neural Network, however a prediction model as complex and computationally heavy as a Neural Network may not be necessary which raises the question: is a Neural Network required to classify whether a email is spam or can a simpler prediction model be used instead to the same efficiency. To do this a small dataset was used taken from UCI's open Machine Learning Repository that takes reported spam and non-spam emails and records the frequency of certain words and phrases.

Analysis

Spambase is a dataset compiled and presented by UCI's Machine Learning Repository. The dataset is a collection of spam and non-spam emails compiled from work and personal emails collected by the creators to provide a dataset to determine if a classification method of spam emails could be done by analyzing the frequency of certain characters and phrases in emails. Within the data set there are 58 columns in regards to features and a total of 4601 entries within the dataset.

Within the dataset the first 54 columns are all continuous and measure the frequency of either a specific key word or a special character ('[',';','(',etc.) with these values ranging from 0 - 1, the next columns 55 - 57 are continuous and measure the frequency, length, and average amount of occurences of capital letters appearing in the email, and the last column is a the target variable of spam which is a binary value where 0 represents a normal email and 1 represents a spam email.

When looking through the data, there were no values that were NULL so no values needed to dropped. The entire dataset was z-scored primarily for the columns measuring capital letters as these columns differed from the other columns as they did not range from 0-1. All columns used as predictors were z-scored to normalize the data and to ensure that all were on the same scale for predicting spam emails.

Methods

As mentioned before the overall prediction model we wll be using is a Neural Network, this Neural Network will take in the 57 prediction values and then output whether or not the email is spam or not. Taking the 57 input features mentioned earlier we will input these through a Neural Network with 10 hidden layers, outputting to a single output layer with the activation function sigmoid. This activation function will take the probability and convert this to either a 0 for a normal email and 1 for a spam.

There will be 10 hidden layers and between these hidden layers there will also be dropout layers, these dropout layers are set to a value of 0.2, meaning that every two hidden layers, 20% of the input data in the layer will be ignored for that iteration of the training process. This will allow our model to become more generalized as different sections of the data will be dropped and ignored each iteration prevent the model from becoming overfit to the specific training data.

To better train our Neural Network, the Neural Network is fitted over 25 epochs or iterations which will allow for differences between the dropout nodes and the hidden layers that will allow our model to experience different splits and scenarios to make it more accurate. In addition between each layer a loss function will be given and applied and the model will make adjustments through gradient descent to minimize our loss and maximize the accuracy of our prediction model. In this case the loss function we will be using will be binary cross-entropy which tracks how often do we have false positives or negatives and penalizes the model to reduce these incorrect predictions.

With each iteration the accuracy of the model for training data and a validation set of data will be recorded by matching the predicted output with our known output given as well as the loss calculated by our loss function. This pretty much tells us how accurate is our model overall and how bad are we at incorrectly predicting whether a email is spam or not.

The other model that we will be comparing our Neural Network to will be a simple Logistic Regression which would be equal to a Neural Network with only a single layer. Similar to the Neural Network the input of the Logisitic Regression will be the 57 columns that we've labeled as predictors, all continuous columns, and the output will be a binary and will be either 1 for a spam email or 0 for a normal email.

Similar to the Neural Network the Logistic Regression model will have the accuracies for both the training set and a test set to validate and measure how the model performs on the data

it was trained on and the data it has yet to see. These scores will be compared to the Neural Network to determine whether or not the usage of the Neural Network is justified.

Results

As seen below in Table 1, the Neural had an average training accuracy of 0.926 and validation accuracy of 0.925. In comparison the Logistic Regression had a training accuracy of 0.932 and a validation accuracy of 0.929. As seen from these values both models performed very close to one another. What this tells is that the Logistic Regression is able to predict whether or not an email is spam to the same efficiency as a Neural Network if not very minutely better. This means that the Neural Network would NOT be required to predict whether or not an email is spam, as the Logistic Regression is able to predict to the same accuracy while being more efficient both space and time wise. A Neural Network is computationally heavy with the addition of more data and layers, while a Logistic Regression is equivalent to a single layered Neural Network so therefore takes less space and computation resources. So in the end it would be more better both in terms of cost and efficiency to use a Neural Network to predict whether an email is spam.

Table 1

Model	Train	Test
NN	0.9266	0.9251
LR	0.9326	0.9294

Reflection

With this assignment, I was able to understand the importance of activation functions and the loss functions. While working on this assignment I was able to adjust and tweak a Neural Network to try and improve its efficiency to as high as possible. It also allowed me to familiarize more of the activation functions and regularization through the keras and tensorflow documentation, when building the Neural Network I basically had these two tabs open constantly to cross-reference what I could do to try and adjust my accuracy of the model. One of the tabs that I should have had open was the one for loss functions as for the longest time I had the wrong loss function of categorical cross-entropy instead of binary cross-entropy and was wondering why my loss was so low and not improving, I was able to overcome this as one of my classmates Sam B. looked over my shoulder and basically told me the loss function was wrong after I was struggling for a bit. This assignment also introduced me to yet another way of fetching data to analyze this time through UCI's Machine Learnine Repository.

In the future with assignments and tasks like this I would for sure have multiple pages of documentation open and keep going back and forth cross-referencing what each does and

switching things to improve the performance of my Neural Network. I had this for one or two tabs of the documentation but I should have probably gone for as many as I needed to ensure there weren't any issues. Another thing to note would probably be the report ahead of time and to meet up with the professor if there's any questions because there were areas where I was a little unsure of what should be in each section.

Overall this assignment provided a good way to gain experience in Neural Networks and how to make and adjust them to improve performance, a better way to navigate and reference documentation for keras and tensorflow, and finally how to present and write information in the format of a technical report.