Student: Jason Zhou
Mentor: Dr. Dongjin Song
Status Report #: 2
Time Spent on Research This Week: 3
Cumulative Time Spent on Research: 10
Miles Traveled to/from Mentor This Week: 0
Cumulative Miles Traveled to/from Mentor: 0
==========

Thursday, September 16th, 2021 (1 hour):

Last week, during my Zoom meeting with my mentor, he recommended that I begin learning TensorFlow, SciKit Learn, and Numpy. As such, today I found a video course on Youtube which taught TensorFlow 2.0 (an updated version of TensorFlow). Although the beginning part was a review on supervised, unsupervised, and reinforcement learning, the later parts had an abundance of things for me to learn. For example, from what I understand, a tensor is a vector computation that is partial and will eventually produce a result. Each tensor has a data type. This means the tensor could be an integer, a string (a collection of letters), a float (decimal numbers), and other types of data. Additionally, all tensors also have their own shape, which is how they are represented in terms of dimensions (2D, 3D, etc). Finally, each tensor has a value assigned to them. They can have one value (which would make them a scalar) or they can have an array of values (this contributes towards the shape of the tensor).

Overall, the video is seven hours long, so I will be watching the video over the course of a week or two to gain a good understanding of TensorFlow.

On this day, I also received an email from my mentor saying that he would be unable to attend our weekly Friday meeting due to some other circumstances.

Friday, September 17th, 2021 (1 hour):

Picking up where I left off the previous day, I continued to watch the TensorFlow video course. The instructor focused on linear regression, which is a machine learning algorithm (essentially the linear regression program that can be found on calculators). To provide a simple explanation, you provide a couple of data points and plot those points on a graph. Then, draw a straight line of best fit. Although it's easier to explain, it is harder to execute in code. First, one must have a data set that they can use and import the data into a variable so that it can be stored and manipulated. The next step would be to separate the data into categorical and numerical data. Categorical data would be anything that can fall under a word (for example, male or female). Numerical data is just anything that can be represented as a number (like age or height). Both types of data will act as features or inputs for the linear regression model.

I also decided to change my IDE (integrated development environment). Previously I had been using a software called Jupyter to run as an IDE to learn machine learning; however, I

have decided to switch to Google Collaborator. Google Collaborator does not need to be installed and can be run on a website that accesses the clouds for all of its needs. As such, this would allow me to work on the GHS school computers without downloading anything, which is convenient.

Saturday, September 18th, 2021 (1 hour):

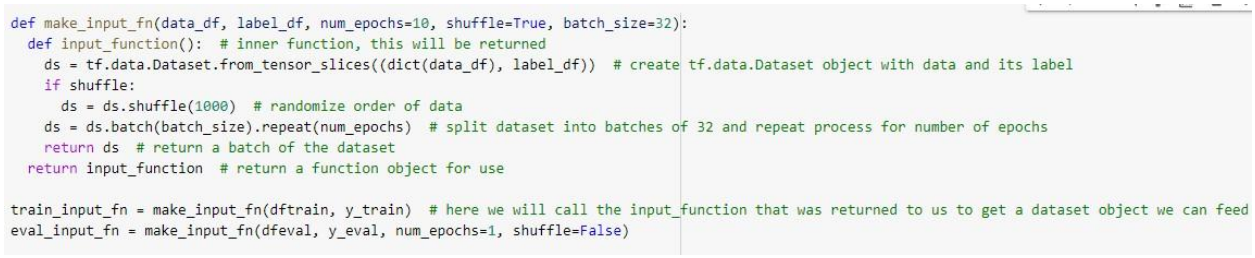
Just like yesterday, I continued to watch the TensorFlow video course. For this part of the video, I ended up learning a lot of technical terms such as categorical and numerical data, batches, epoch, overfitting, and input function.

First off, when handling data, one must prepare batches of data. For instance, instead of passing in all of the data into a model, only certain amounts of data would be passed in at a time. Ultimately, this makes processing the data faster. This would be similar to moving heavy luggage around. Instead of trying to carry all of it at once, one might try to carry portions of the luggage at a time because it is easier.

Associated with batches, there are also epochs that can be defined. An epoch is simply how many times the model will look over the data. For example, it's like setting a music playlist to repeat "n" number of times, so that one can hear each song played nth times. Setting an epoch value helps the model improve through repetition just like humans might.

Usually, it is good for the model to repeatedly look at the data (setting an epoch); however, this can also be harmful to the learning process. When the epoch value is too high and the model has looked at the data too many times, it begins to memorize the output values instead of understanding the concepts or patterns that make up the data, which is bad for a program that is supposed to make predictions. This is what would be called overfitting. So, for example, an epoch value of 100,000,000,000,000 would definitely overfit a model for most datasets.

The concept of batches and epochs makes sense; however, to actually apply these ideas, one must use an input function. This is simply a method (function) that defines how a dataset will handle batches and how many times it will loop through the data (epoch).



```
def make_input_fn(data_df, label_df, num_epochs=10, shuffle=True, batch_size=32):
  def input_function():  # inner function, this will be returned
    ds = tf.data.Dataset.from_tensor_slices((dict(data_df), label_df))  # create tf.data.Dataset object with data and its label
    if shuffle:
      ds = ds.shuffle(1000)  # randomize order of data
    ds = ds.batch(batch_size).repeat(num_epochs)  # split dataset into batches of 32 and repeat process for number of epochs
    return ds  # return a batch of the dataset
  return input_function  # return a function object for use

train_input_fn = make_input_fn(dftrain, y_train)  # here we will call the input_function that was returned to us to get a dataset object we can feed
eval_input_fn = make_input_fn(dfeval, y_eval, num_epochs=1, shuffle=False)
```

(This is a picture of code I wrote as a reference to input functions, which came from the TensorFlow video course I have cited in the reference section. It shows the steps to making a basic input function that will define the batches and epochs for a dataset.)

Besides what I learned today, I also received an email from Doctor Jinbo Bi (who I believe is Niteesh's mentor) explaining the details about a high school group research program at UConn which I had talked about in my previous status report. Essentially, it's a study group involving students from both GHS and EOSmith who have similar interests in computer science. This group was primarily made to help the students gain a better understanding of concepts or skills that we need (linear algebra, machine learning, coding, etc.). Additionally, there will also be demos of research projects from various research teams that the students will be able to attend.

As of this moment, this is all I currently know, and I hope to learn more in the coming week!

References

freeCodeCamp.org. (2020, March 3). *TensorFlow 2.0 complete course - python neural networks for beginners tutorial* [Video]. *Youtube*. https://www.youtube.com/watch?v=tPYj3fFJGjk&t=6284s