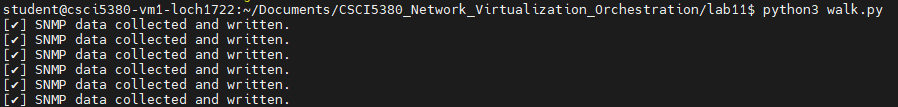
TensorFlow Application Explanation

The TensorFlow application I created was a basic way to predict the health of a device based on certain SNMP metrics polled from a host of devices. There are three main modules to this application: walk.py, train.py and predict\_health.py.

Walk.py

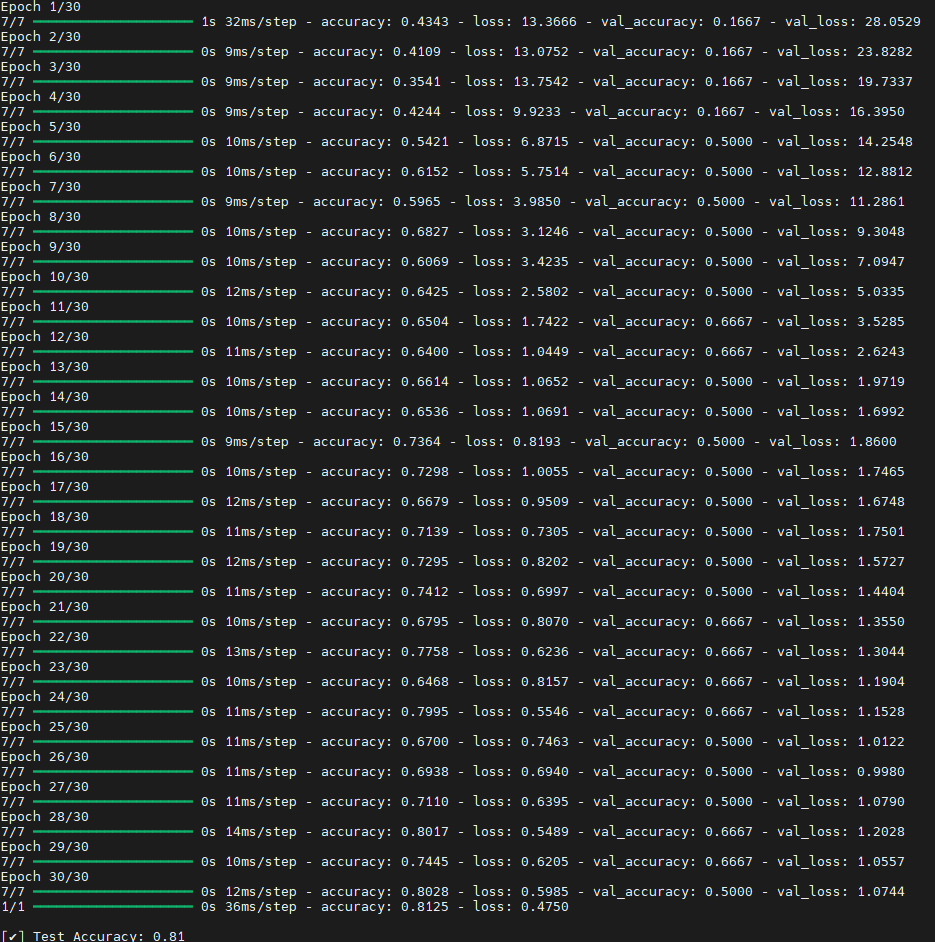
Walk.py is a simple application that polls CPU, packet in and packet out errors via SNMP on a given list of device IPs. All devices must be setup for SNMPv2c and the device OIDs are Cisco.



1st walk.py does a SNMP call on those 3 OIDs. 2nd data is then parsed and appended into a csv called: “cisco\_snmp\_health.csv”. Within the .csv there is a label header that identifies health based on certain metrics I outlined based on the CPU, packet in and pack out errors. This will then put a label of “healthy”, “warning”, or “failure”. Within walk.py will display the metrics for these labels.

Train.py

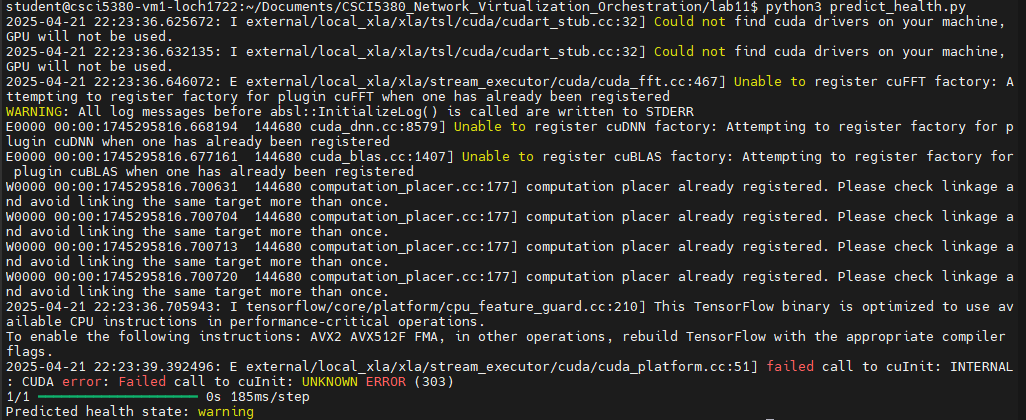
Train.py uses TensorFlow to create a model based on x inputs of CPU, packet in and packet out errors and the y value being the health itself. This model will then be used to predict device health based on the three inputs.



Predict\_health.py

Predict\_health.py simply uses the encoder and model created by the train.py file and given a sample will then predict the health of the device.

Sample: {CPU: 53, in\_errors: 20.0, out\_errors: 9.0}



Running TensorFlow Application

To run this application, first install the dependencies:

pip install tensorflow pandas scikit-learn

Next, input devices wanted in walk.py in the routers array.

Run python3 on walk.py until enough data is collected and Ctrl+C to exit the application.

Then run train.py to create the model and finally input data points in the predict\_health.py sample variables and run that for a health prediction.