**Sentiment Analysis:**

Team: Spiderman

Term: Fall 2022

Members: Sumanth Sureshbabu (sumanth4@illinois.edu)

Vasantha Kumar Dhanasekaran \* (vkd2@illinois.edu)

Rajesh Grandhi (rajeshg2@illinois.edu)

The main purpose to do sentiment Analysis to predict sarcasm. The steps involved are build and train a classifier for the sarcasm dataset.The classifier should have a final layer with 1 neuron activated by sigmoid as shown. It will be tested against a number of sentences that the network hasn't previously seen and you will be scored on whether sarcasm was correctly detected in those sentences.

**Setup:**

Install the Tensorflow and other libraries.

*pip install tensorflow*

*pip install requests*

*pip install numpy*

As an alternate we have included the dockerfile to do a 1 step install and run the program

*Make sure the docker in running*

*See appendix how to run the python program using docker*

**Implementation:**

First we import all the libraries

import json  
import tensorflow as tf  
import numpy as np  
import urllib  
from tensorflow.keras.preprocessing.text import Tokenizer  
from tensorflow.keras.preprocessing.sequence import pad\_sequences  
import tensorflow as tf  
import numpy as np  
import requests

* We can download the dataset from the google datastore.
* Define the tuning parameters

example

vocab\_size = 1000  
embedding\_dim = 16  
max\_length = 120  
trunc\_type='post'  
padding\_type='post'  
oov\_tok = "<OOV>"  
training\_size = 20000

* Segregating the training data into sentences and labels.
* Splitting the data into training and test
* Next step is tokenization. This object contains token objects from the tokenization process.
* Train the data with model.fit method.
* Save the model and test it with test data.

This step helps us to determine the hyperparameters required to tune and attain high accuracy of the model

Output:

Predicting the sarcasm in a dataset using random sentences.