DEEP LEARNING PROJECT WEEK-2

1.Topic In this project I have built a Neural Network to detect Tremor/Shaking of Upper or Lower Body of people. I have used a deep neural network with Long-Short-Term memory cells.

2. DNN Model

2.1 Architecture The model is built with a linear stack of layers with the Sequential model. I have used two LSTM layers with a dropout layer on the top and then a dense layer with regularization and the output layer. The code snippet and the model architecture are shown below.

```
import tensorflow as tf
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.LSTM(units=45,return_sequences=True,input_shape=(45,30)))
model.add(tf.keras.layers.LSTM(units=45,return_sequences=True,input_shape=(45,30)))
model.add(tf.keras.layers.TimeDistributed(tf.keras.layers.Dropout(0.2)))
model.add(tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(units=90,activation='relu')))
model.add(tf.keras.layers.BatchNormalization())
model.add(tf.keras.layers.TimeDistributed(tf.keras.layers.Dropout(0.2)))
model.add(tf.keras.layers.TimeDistributed(tf.keras.layers.Dense(units=1, activation='sigmoid')))
```

2.2 Input:
Shape of tensor Before reshaping:
X_train shape is (80010,30):(Json files,features)
X_test shape is (18000,30):(Json files,features)
After reshaping:
X_train shape is (889,45,30):(samples,Timesteps,features)
X_test shape is (400,45,30):(samples,Timesteps,features)

2.3 Output:
Shape of tensor Y_train is (80010,1) Y_test is (18000,1)

3.Hyperparameters

3.1 List of Hyperparameters used In this model the hyperparameters I have used are: epochs, batch size, learning rate, number of LSTM layers and depth of the network, dropout rate.

3.2 Range of Hyperparameters tried

Batch size	1,3,5,10,15,20,32
Epochs	10,50,100,200,500,700
Learning rate	0.1,0.01,0.001,0.002,0.0001,0.0002
Dropout	0.1,0.2,0.4,0.5,0.7
Depth of LSTM layers	1,2,3,4,5,6
Number of time stamps	90,45

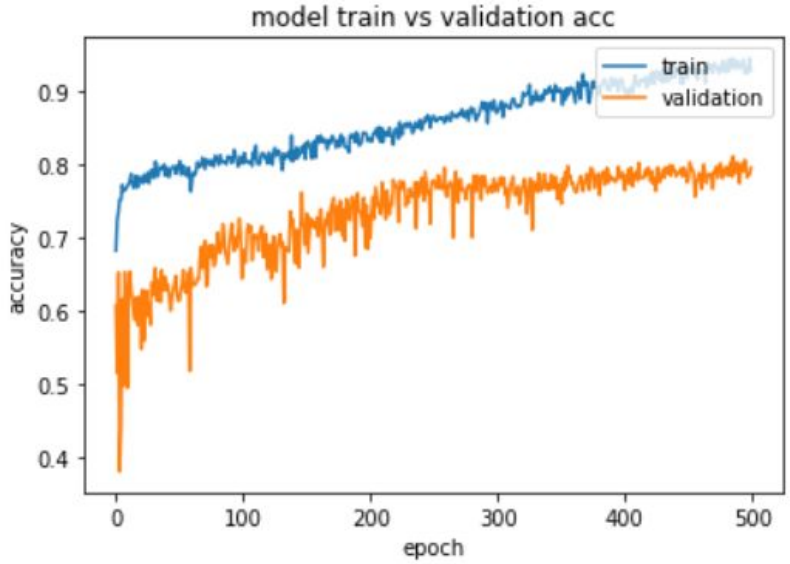
3.3 Optimal Hyperparameters found

Batch size	5
Epochs	500
Learning rate	0.002
Dropout	0.2
Depth of LSTM layers	2
Number of time stamps	45

4. Training and Testing performances

4.1 Accuracies The below graph shows how the training and validation accuracies change for each epoch

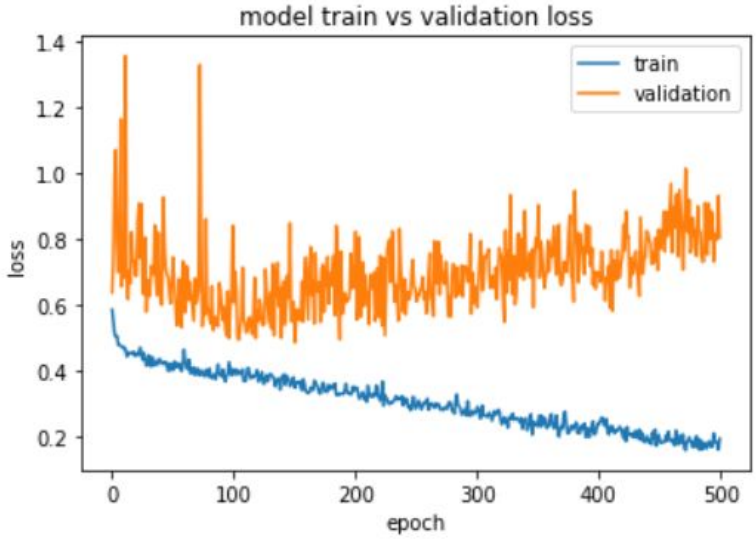
Training acc: 91.89% validation acc: 80.01%



5.2 Losses

The below graph shows how the training and validation Losses change for each epoch

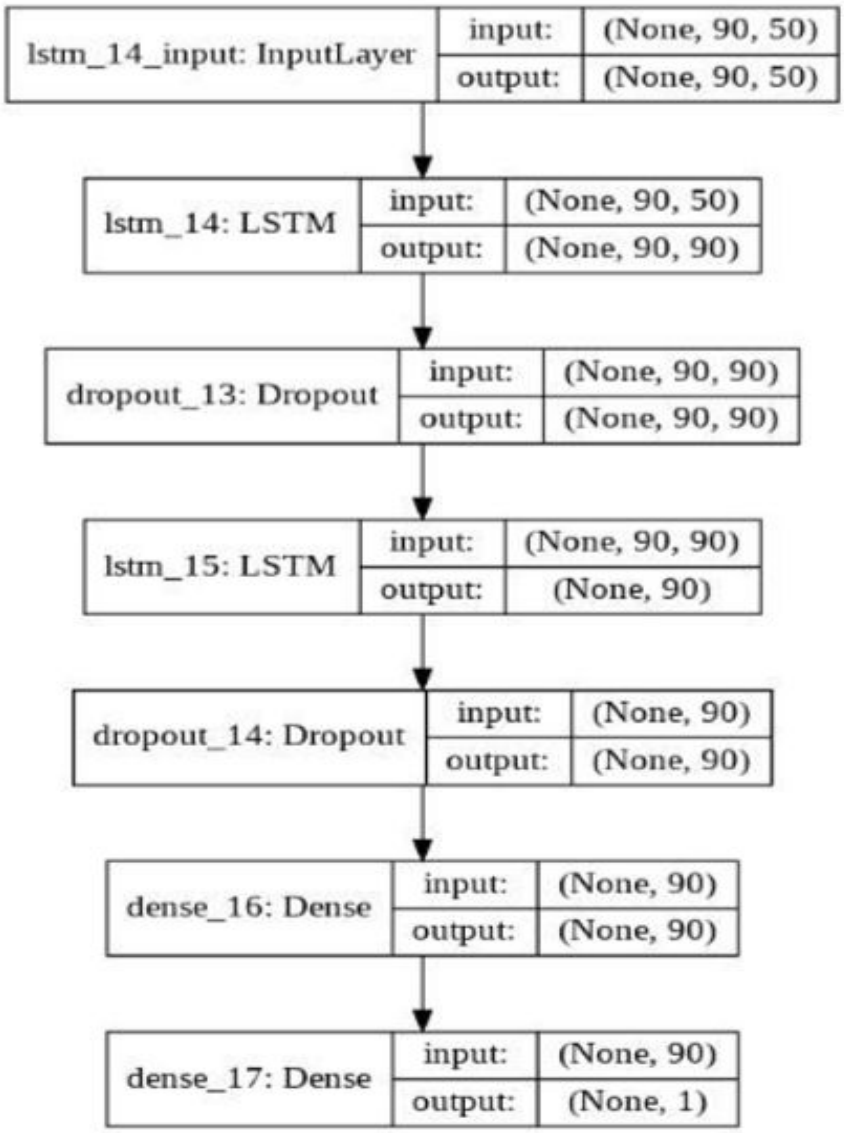
Training loss: 0.2201 validation loss: 0.5601



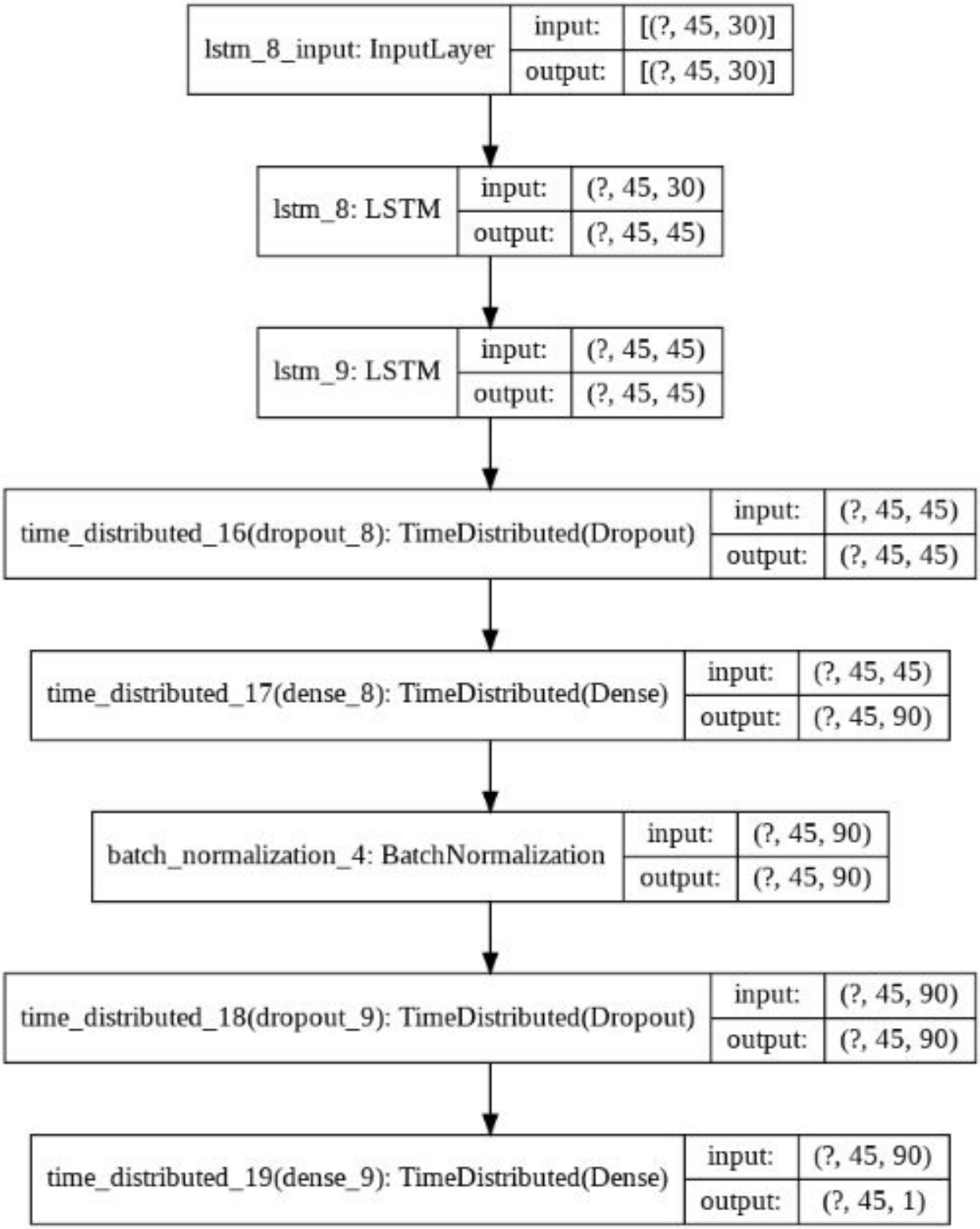
5. IMPROVEMENTS:

1. **Model Architecture:**After reading several papers and experimenting with several layers and several numbers of nodes. I have changed the model architecture to the following.

   Previous model



Current Model:



2. **More Data:** I do not have any data set available online hence I need to make videos on my own. In the previous week I have trained my model with 50000 frames and validated it on 10000 frames. This time I have trained the model with 90,000 frames and validated it on 18,000 frames. I have noticed that my model was giving one as output when I was standing as well without shivering. I have added many more negative samples of standing. You can clearly see the difference in a few videos, the accuracy has improved heavily.

Data taken	**Previous Week 1**	**Present Week 2**
Training	50,000 frames	90,000 frames
Testing	10,000 frames	18,000 frames

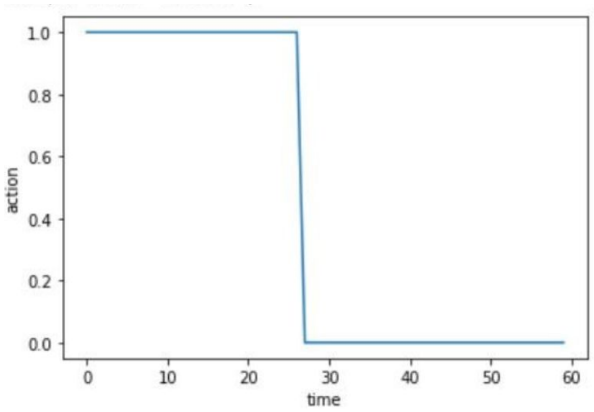
3. **Using Time Distributed Layer**: Previously, I have got a single output value after 90 frames, but now My model gives the output for every frame. Hence there is no generalization. So for every second I get 30 outputs and hence I take the mode of all the 30 outputs to plot the graph.

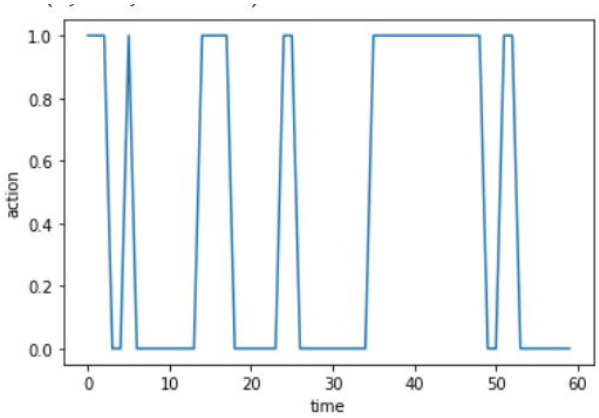
Previous Week 1	90 frames one output
Present Week 2	1 frame 1 output

4. **Features:** I have taken only 30 features of the Openpose output as the last 20 coordinates of the openpose output are of the toes and head. As my detection model does not depend on these coordinates I have removed them. This has improved the validation and training accuracy by a great extent.

5. **Performance:** The model performance has improved by 6%. I have improved the validation accuracy of the model from 73.77% to ~80%.

Video 1: first 15 seconds shivering and next 45 seconds no shivering.

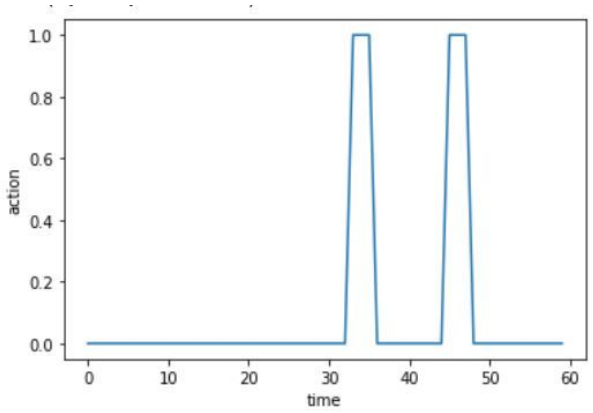


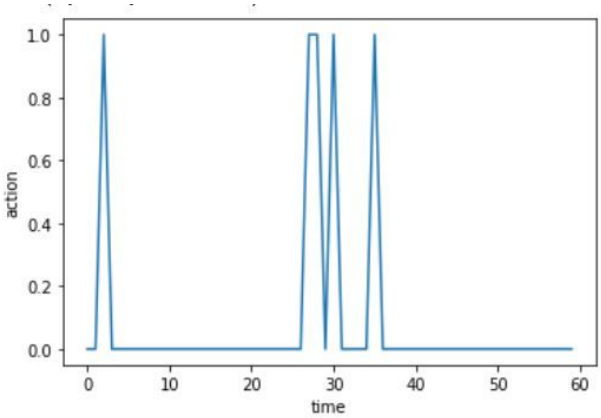


Week 1 Week 2

Comment: This is a very poor validation video where the difference between shivering and not shivering is not portrayed properly. The model performs extremely well with the rest of the validation videos.

Video 2: This is completely a negative video where there is no shivering at all.

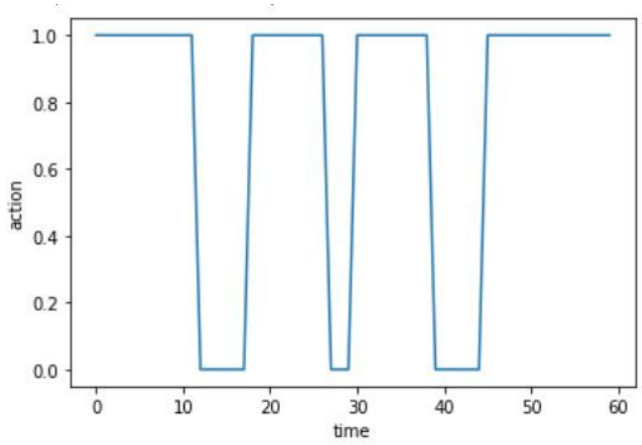


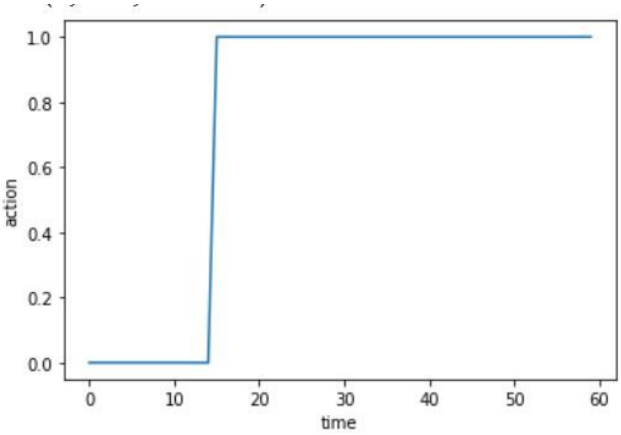


Week 1 Week 2

Comment: Last time I got one output for 3 seconds so for all the 3 seconds I have put the same value. But this time I have got output for every single frame. Hence the accuracy is more and very small parts of them like 1 second spikes have appeared. I will be improving the model with negative samples further in the next submission.

Video 3: In this video I have shivered for the last 45 seconds.

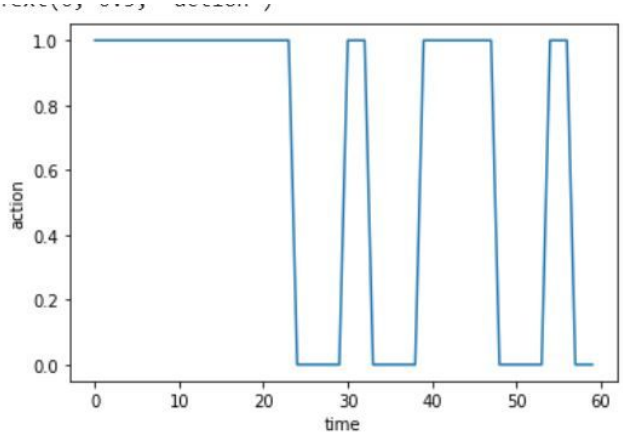


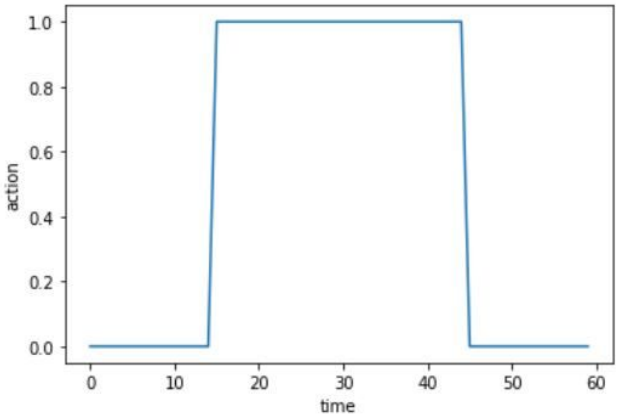


Week 1 Week 2

Comments: This week I have completely focussed on this improvement. I have seen the places where the video was predicting wrongly and have rectified it. The above output is exactly accurate.

Video 4: 15 seconds: shivering, 30 seconds: no shivering, 15 seconds: shivering

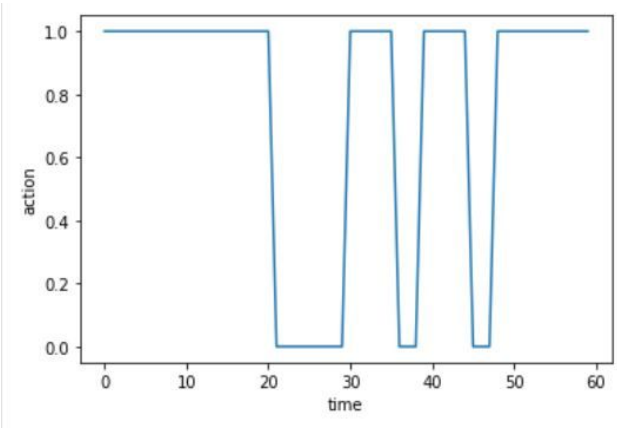


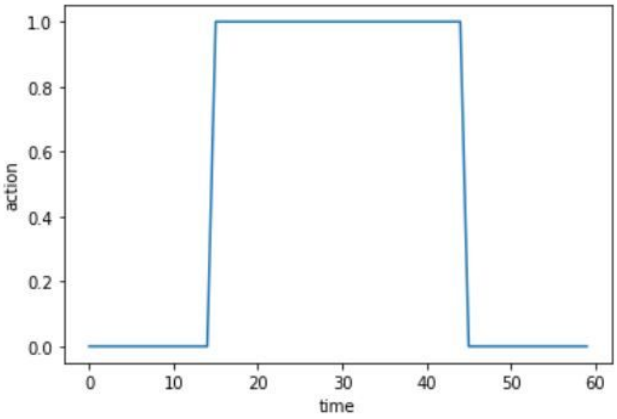


Week 1 Week 2

Comments: The graph is exactly accurate and has given correct results.

Video 5:





Week 1 Week 2

Comments: The graph is exactly accurate and has given correct results.