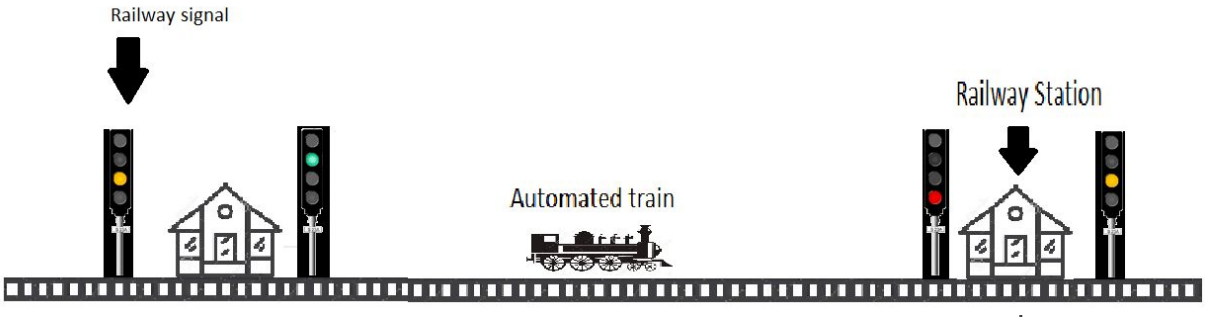
18CSE352T Neuro Fuzzy

AUTOMATED TRAIN
FUZZY CONTROL SYSTEM



Introduction :

So , basically this fuzzy control system is just an AI prototype for an automated train i e a train which runs by its own on the basis of 2 input parameters

- Distance from the signal (in km)
- Color of the signal (red , green or yellow)

On the basis of given input parameters , this system dynamically changes the speed of train which is the output (in kmph)

Basically If the signal color is red , then train stops just before it , If it is yellow then it stops at the upcoming station and last case when it turns green it surpasses the upcoming station

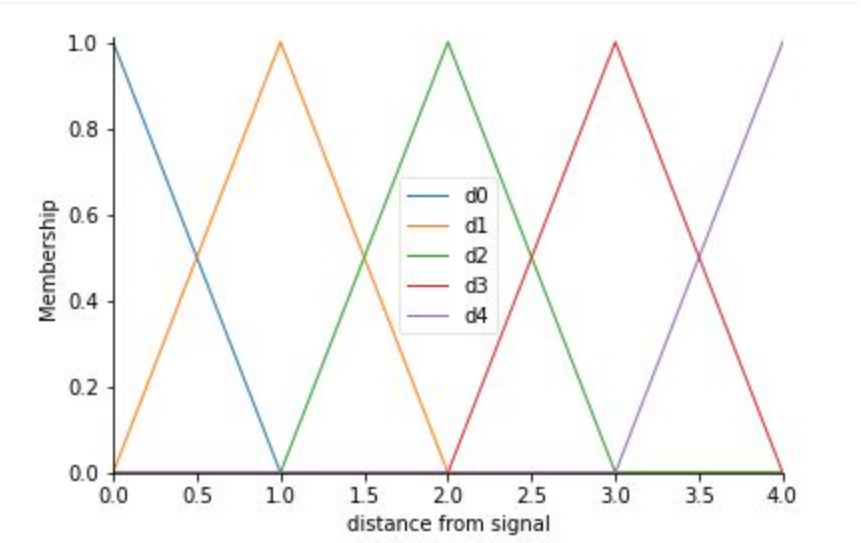
Principle:

- 1. Fuzzification of the input variables
- 2. Application of fuzzy operator(AND) on the antecedent parts of the rules
- 3. Evaluation of the fuzzy rules
- 4. Aggregation of fuzzy sets across the rules

- 5. Defuzzification of the resultant fuzzy set (i e speed)

Working :

Distance

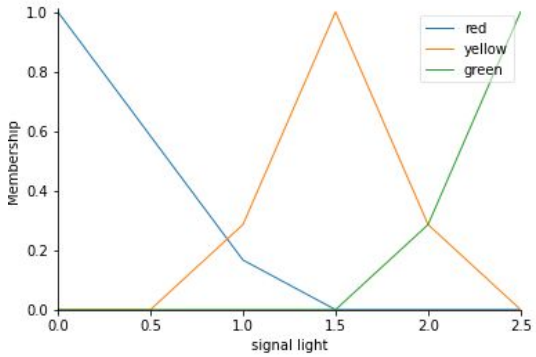


We assume that our train has sensor or radar which has capacity to measure its distance from upcoming signal in the range of 4 km

Accordingly we have segmented this range into 5 linguistic values

- d0 (0 - 1 peak at 0)
- d1 (0 - 2 peak at 1)
- d2 (1 - 3 peak at 2)
- d3 (2 - 3 peak at 3)
- d4 (3 - 4 peak at 4)

Signal :



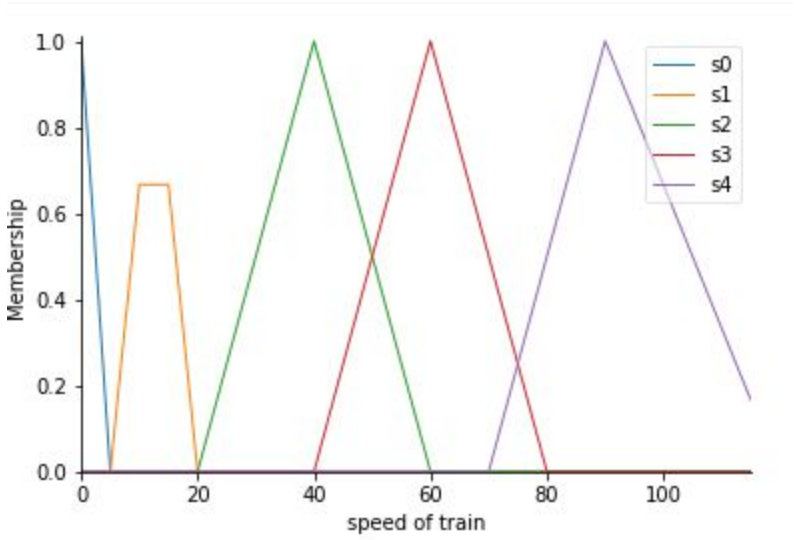
Well , color of signals lie under nominal category , but I don't know to assign membership value to nominal value since I am new to this module (scikit fuzzy) , so i decided to declare range for each of them

Here median membership value for Red signal is 0.4 , 1.3 for yellow light and 2.2 for green light

Note :

Any changes like intersection were made based on analysis of output

Speed :



Here we have speed of train ranging upto 120kmph which is segmented into 5 non-uniformly distributed membership function.

Unlike other parameter which have uniform triangular membership function .
This mixed membership function is has mixed geometry of triangular and trapezoidal shape

Rules of Inference :

Rule 1 :

If the distance from signal lies in the range d3 and color of signal is red then the speed of train should lie in the range of s1

Rule 2 :

If the distance from signal lies in the range d2 and color of signal is red then the speed of train should lie in the range of s0

Rule 3 :

If the distance from signal lies in the range d1 and color of signal is red then the speed of train should lie in the range of s0

Rule 4 :

If the distance from signal lies in the range d0 and color of signal is red then the speed of train should lie in the range of s0

Rule 5 :

If the distance from signal lies in the range d3 and color of signal is yellow then the speed of train should lie in the range of s2

Rule 6 :

If the distance from signal lies in the range d2 and color of signal is yellow then the speed of train should lie in the range of s2

Rule 7 :

If the distance from signal lies in the range d1 and color of signal is yellow then the speed of train should lie in the range of s1

Rule 8 :

If the distance from signal lies in the range d3 and color of signal is green then the speed of train should lie in the range of s4

Rule 9 :

If the distance from signal lies in the range d2 and color of signal is green then the speed of train should lie in the range of s4

Rule 10 :

If the distance from signal lies in the range d1 and color of signal is green then the speed of train should lie in the range of s3

Final Defuzzified Result :

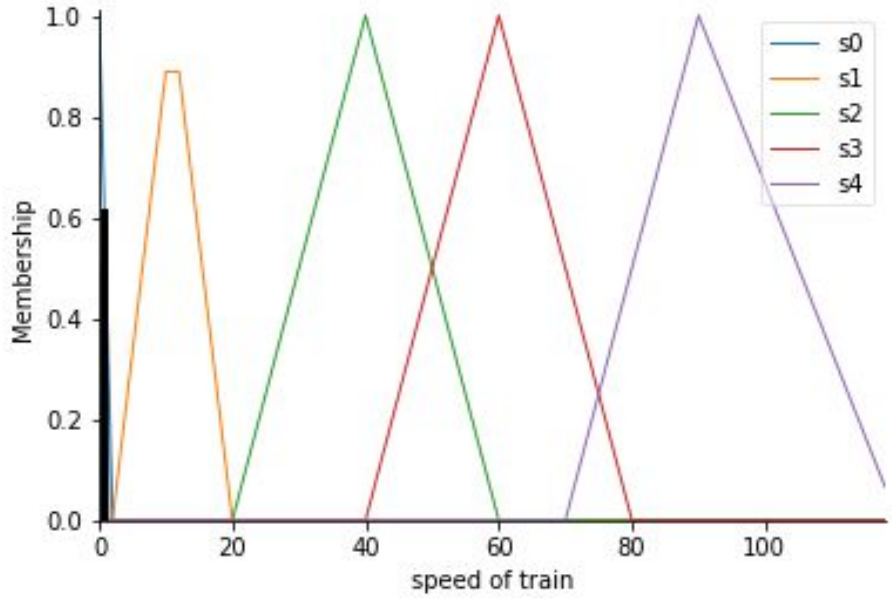
We have tested it for some single tuple of input (mostly boundaries) :

- **When distance is less than 1 km and signal display red color**

```
spding.input['distance from signal'] = 0.5
spding.input['signal light'] = 0.5
spding.compute()
```

```
print(spding.output['speed of train'])
spd.view(sim=spding)
```

```
0.7777777777777777
```



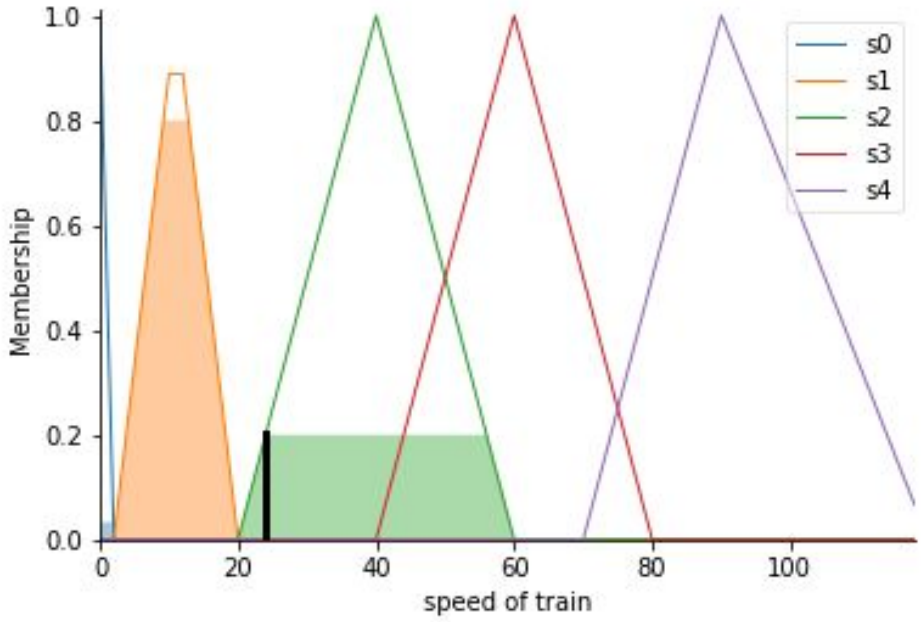
That is approx 780mph (meters per hour)

- **When distance is less than 2 km and signal display yellow color**

```
spding.input['distance from signal'] = 1.2
spding.input['signal light'] = 1.4
spding.compute()
```

```
print(spding.output['speed of train'])
spd.view(sim=spding)
```

```
24.08620483564248
```



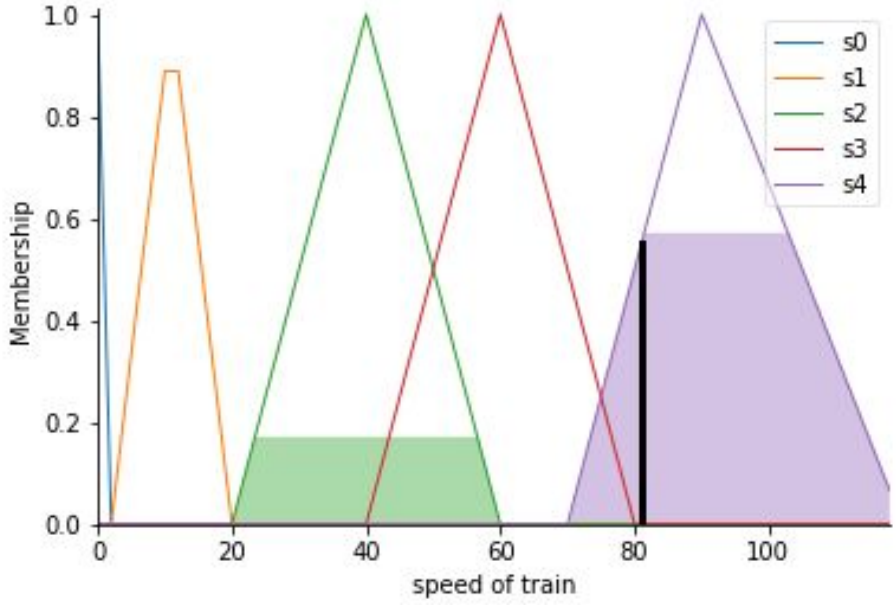
Speed of train at that instance can be approx 24 kmph

- **When distance is less than 3 km and signal display green color**

```
spding.input['distance from signal'] = 2.6
spding.input['signal light'] = 2.2
spding.compute()
```

```
print(spding.output['speed of train'])
spd.view(sim=spding)
```

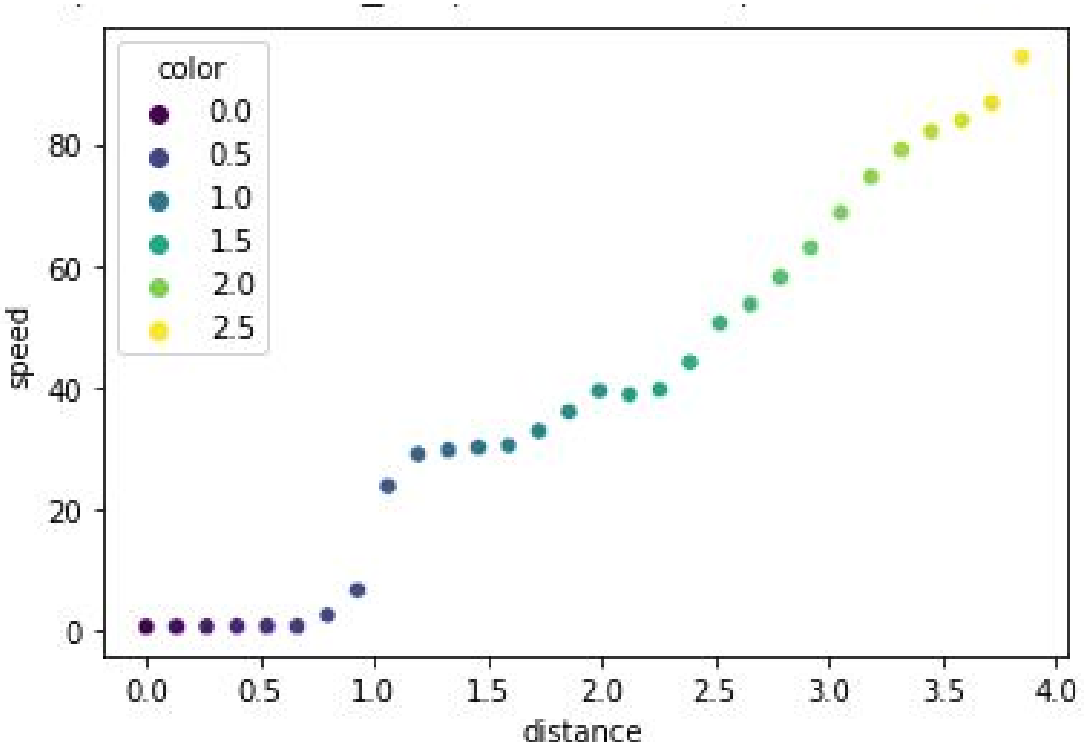
```
81.03348355038118
```



Speed of train at that instance can be approx 81kmph

On the basis of these boundary condition , we prepared a dataframe consisting of distance from signal , color of signal and speed of train of about 30 observation which depict a motion Of train when it first encounters red signal then yellow and then green .

And plotted distance against speed in from of scatterplot with hue of colour



This depict that on approaching red signal it's speed tends to zero , on the other hand when it approaches yellow or green signal its speed increases .

SUBMITTED BY :

GIRISH PANDEY
Btech CSE
SRM Institute Of Science and Technology
2018-22