Machine Learning Lab3

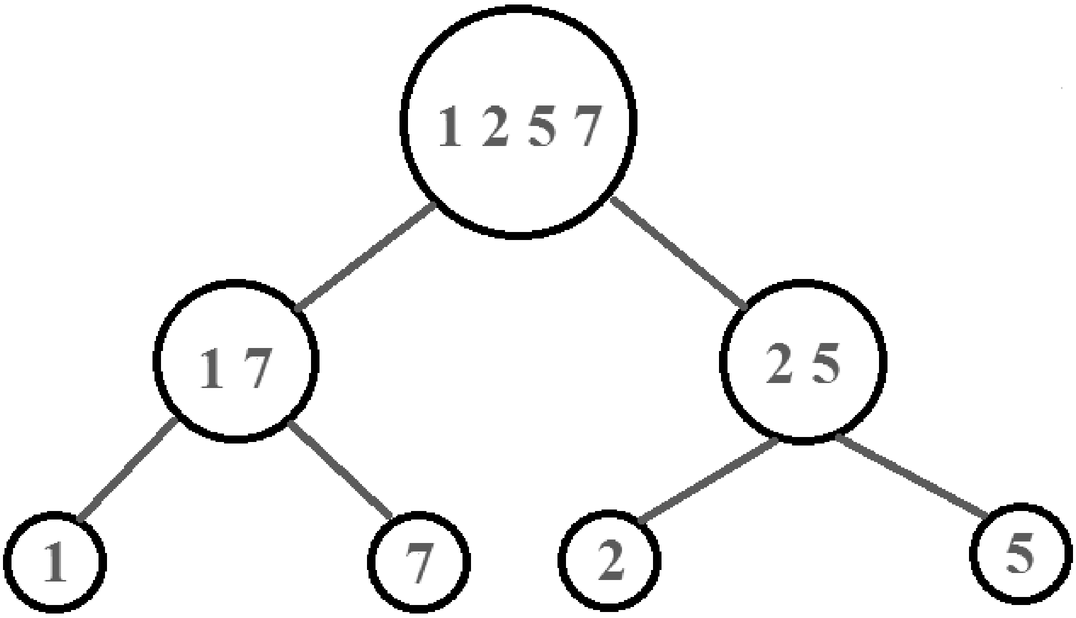
Designing a Perceptron:

Then it is your job to implement the perceptron algorithm. Please try to encapsulate your algorithm as a function. A possible form of the function can take two input arguments, one is the feature vector (tr_feats) and the other is the label (tr_label), and it will output the learned weights:

```
                function weight=Perceptron(feats,label)
```

After you have implemented the perceptron function, try to finish the following tasks:

1. Treat digit ‘2’ as positive, digit ‘5’ as negative, learn a percetron. Try to visualize the samples that are most likely to be ‘2’ or most likely to be‘5’. Also try to visualize the wrongly classified samples.
2. Now consider four digits, ‘2’, ‘5’ ‘1’, and ‘7’. If we consider ‘2’ and ‘5’ are more visually alike, and the same for ‘1’ and ‘7’, then we could design a ‘parent’ perceptron that consider ‘2’ and ‘5’ to be positive, ‘1’ and ‘7’ to be negative. We can also design two ‘child’ perceptrons, one for classifying ‘2’ against ‘5’, and the other distinguishes ‘1’ from ‘7’. Therefore, in total we need three perceptrons to classify these four digits. Do you have some other ideas? Try to implement them and compare the results.



3. Now consider all the 10 digits. Please think by yourself that how many perceptrons do you need to recognize them. Then implement it!