Bilkent University

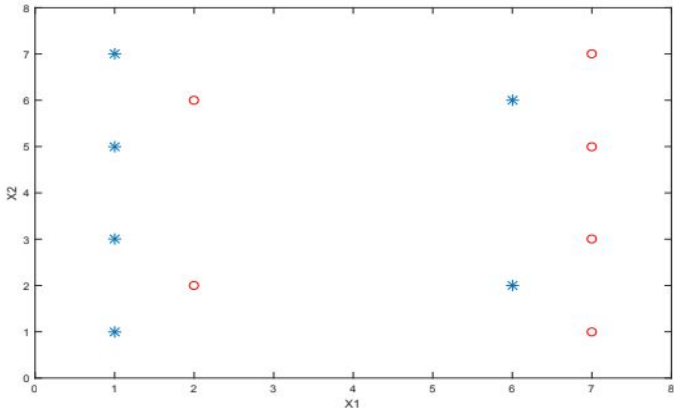
Department of Computer Engineering

CS - 464
Introduction to Machine Learning

Homework 3

Name : **Osman Burak İntişah**
Id : **2160243**
Section : **2**
Date : **22.05.2020**

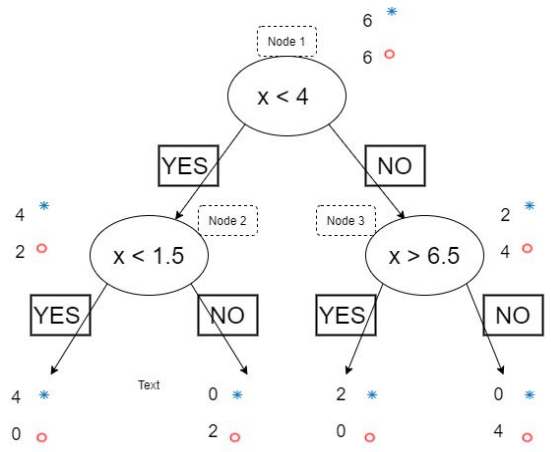
Question 1)



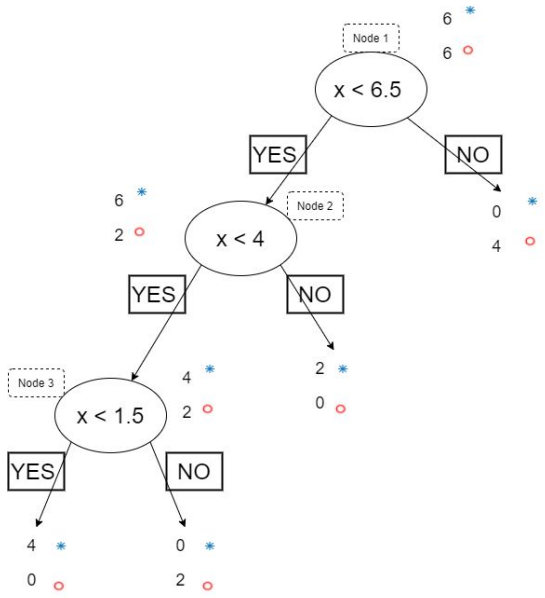
a)

I have found two different decision trees which are given below

1st Decision Tree



2nd Decision Tree



Entropy Calculations

The formulation that is provided in the class is used for this calculations

$$H(X) = E(I(X)) = \sum_i p(x_i) I(x_i) = -\sum_i p(x_i) \log_2 p(x_i)$$

For the 1st Decision Tree

Node 1

$$H(x) = \left(\frac{6}{12}\log_2 2 + \frac{6}{12}\log_2 2\right) = \log_2 2 = 1$$

Node 2

$$H(x) = -\left(-\frac{4}{6}\log_2\left(\frac{4}{6}\right) - \frac{2}{6}\log_2\left(\frac{2}{6}\right)\right) = \frac{4}{6}\cdot(0.58) + \frac{2}{6}\cdot(1,6) = 0.92$$

Node 3

$$H(x) = -\left(-\frac{2}{6}\log_2\left(\frac{2}{6}\right) - \frac{4}{6}\log_2\left(\frac{4}{6}\right)\right) = \frac{2}{6}\cdot(1,6) + \frac{4}{6}\cdot(0.58) = 0.92$$

For the 2nd Decision Tree

Node 1

$$H(x) = \left(\frac{6}{12}\log_2 2 + \frac{6}{12}\log_2 2\right) = \log_2 2 = 1$$

Node 2

$$H(x) = -\left(-\frac{6}{8}\log_2\left(\frac{6}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right)\right) = \frac{6}{8}\cdot(0.42) + \frac{2}{8}\cdot(2) = 0.815$$

Node 3

$$H(x) = -\left(-\frac{2}{6}\log_2\left(\frac{2}{6}\right) - \frac{4}{6}\log_2\left(\frac{4}{6}\right)\right) = \frac{2}{6}\cdot(1,6) + \frac{4}{6}\cdot(0.58) = 0.92$$

Leaf Nodes

For the leaf nodes the entropy is equal to 0. Because all of the leaf nodes are pure. Which means they include just one class.

Information Gain Calculations

Information Gain = entropy(parent) – [average entropy(children)]

For the 1st Decision Tree

Node 1

$$IG = 1 - \left(\frac{1}{2}\cdot 0.92\right) = 0.54$$

Node 2

$$IG = 0.92 - 0 = 0.92$$

Node 3

$$IG = 0.92 - 0 = 0.92$$

For the 2nd Decision Tree

Node 1

$$IG = 1 - \left(\frac{1}{2}\cdot(0.815 + 0)\right) = 0.59$$

Node 2

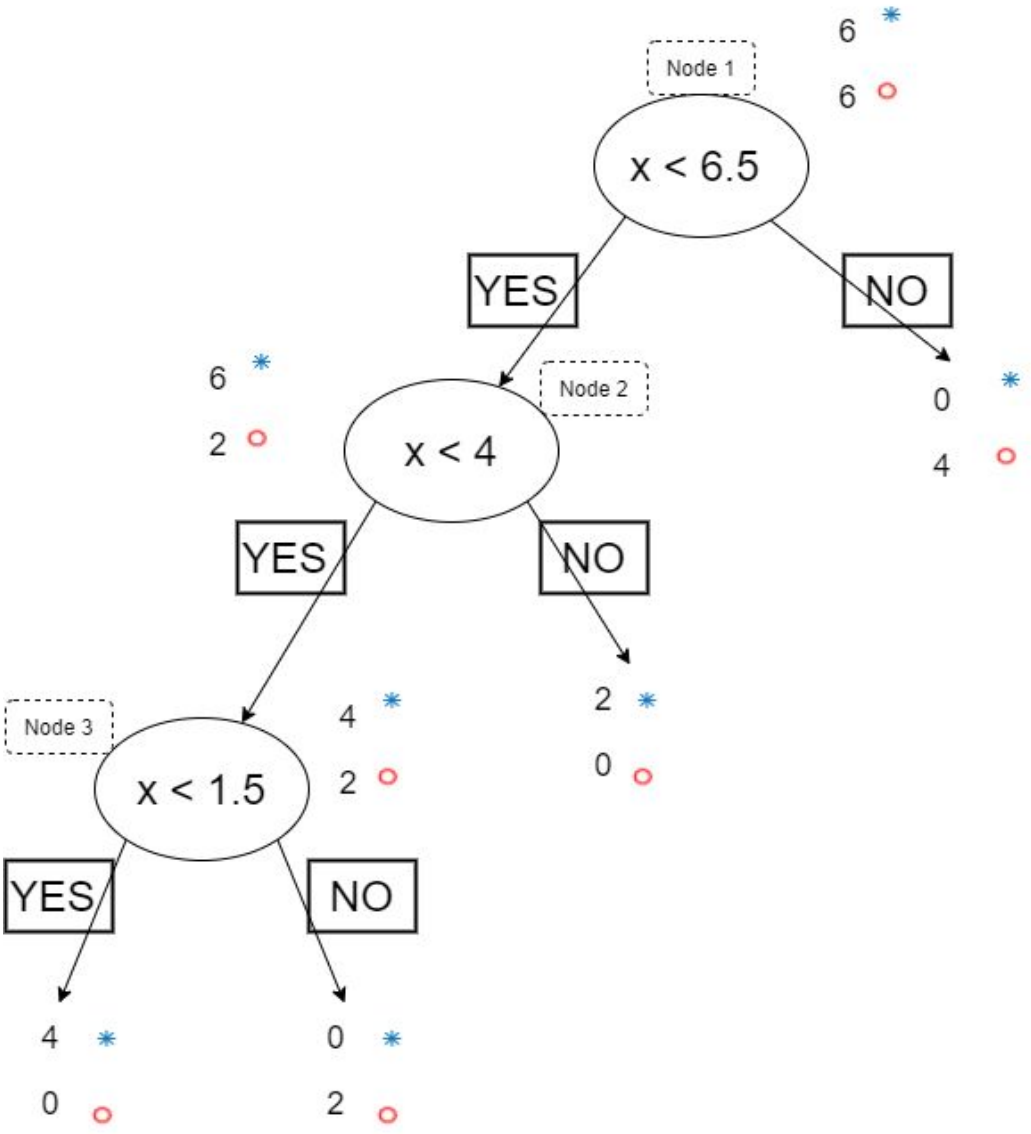
$$IG = 0.815 - \frac{(0 + 0.92)}{2} = 0.355$$

Node 3

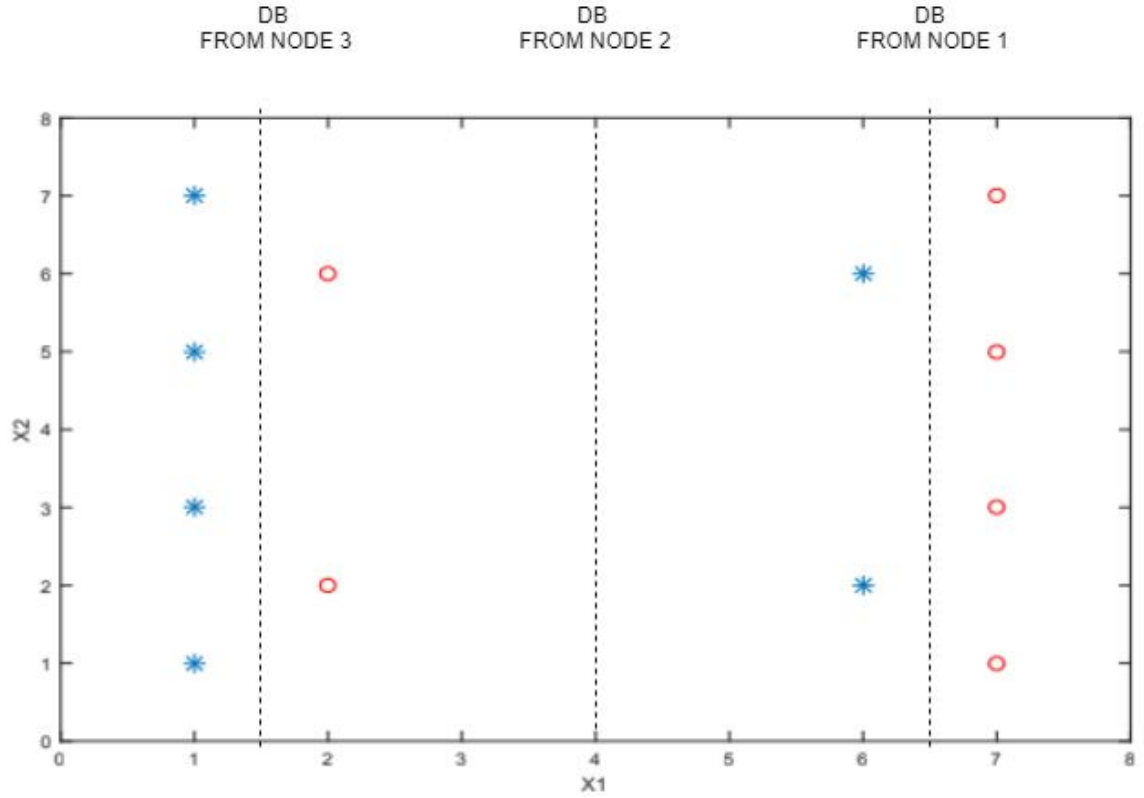
$$IG = 0.92 - 0 = 0.92$$

Finally,

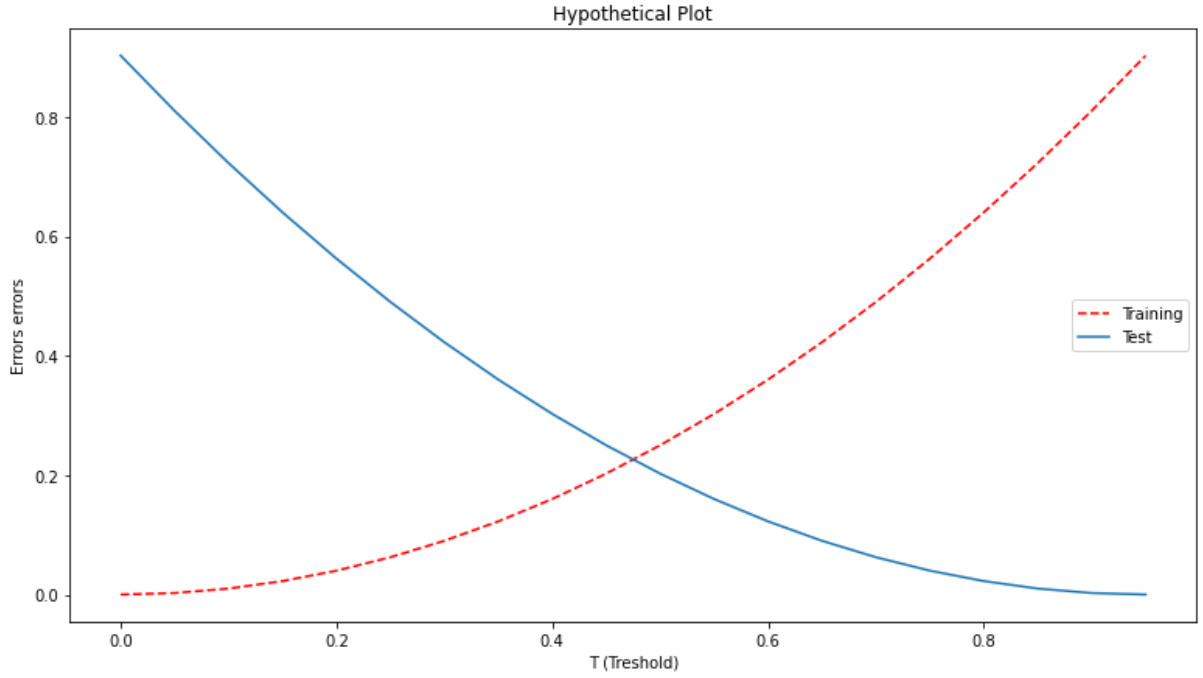
According to given calculations above since we are having more information gain from the first node and we are using ID3 algorithm which chooses the best one for splitting the second tree (which is given below) is the answer of our question.



b)


DB
FROM NODE 3
DB
FROM NODE 2
DB
FROM NODE 1
X2
X1
0
1
2
3
4
5
6
7
8

c)


Hypothetical Plot
Errors errors
T (Treshold)
Training
Test
0.0
0.2
0.4
0.6
0.8

Training errors should be decrease when T is becoming large. Because we start to classify the training set better and better and at the point T=0 we would have 0 training error since every point is classified truly.

However, for the test errors, it ca does not increase and reach to 0. Because we can not know the test set and there might be some misclassifications. And making T smaller may overfit the model. And as it is stated in the lecture we having some misclassified training set is good.

I know that the curves may not seem exponentially like this but I wanted to show the increase and decrease of these errors according to some threshold.

d)

ID3 algorithm is not optimal. Because as it is seem from the calculations in the first question we are using a greedy approach and selecting the best one. However, this approach may cause to converge to a local optima.

e)

$$GINI(t) = 1 - \sum_i p(i|t)^2$$

$p(i \mid t)$ is the relative frequency of class i at node t

Gini Index Calculations
For the 1st Decision Tree

$$G(N1) = 1 - \left(\frac{6}{12}\right)^2 - \left(\frac{6}{12}\right)^2 = 0.5$$

$$G(N2) = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = 0.444$$

$$G(N3) = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = 0.444$$

For the 2nd Decision Tree

$$G(N1) = 1 - \left(\frac{6}{12}\right)^2 - \left(\frac{6}{12}\right)^2 = 0.5$$

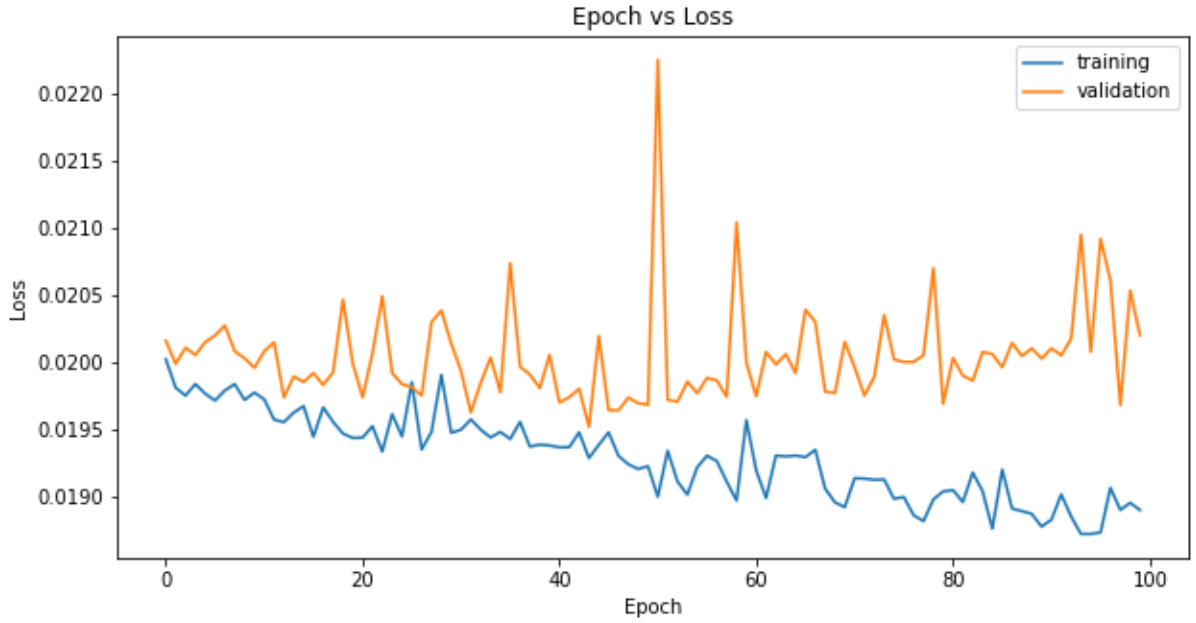
$$G(N2) = 1 - \left(\frac{6}{8}\right)^2 - \left(\frac{2}{8}\right)^2 = 0,375$$

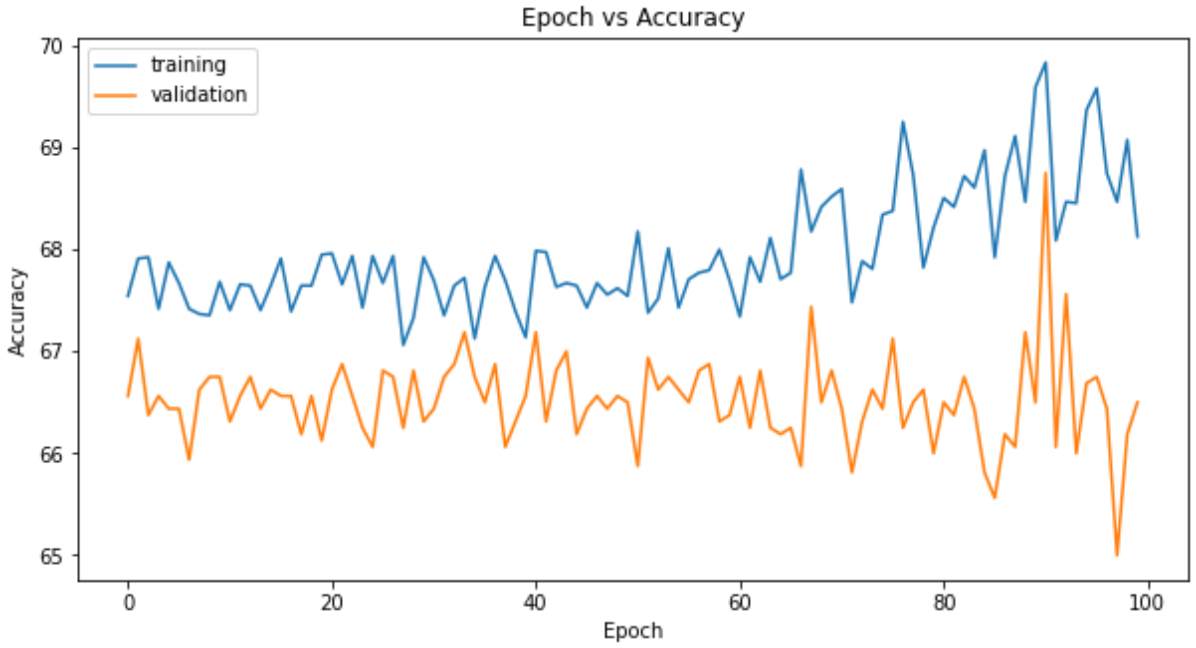
$$G(N3) = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = 0.444$$

Questin 2.1)

Plotting Results:

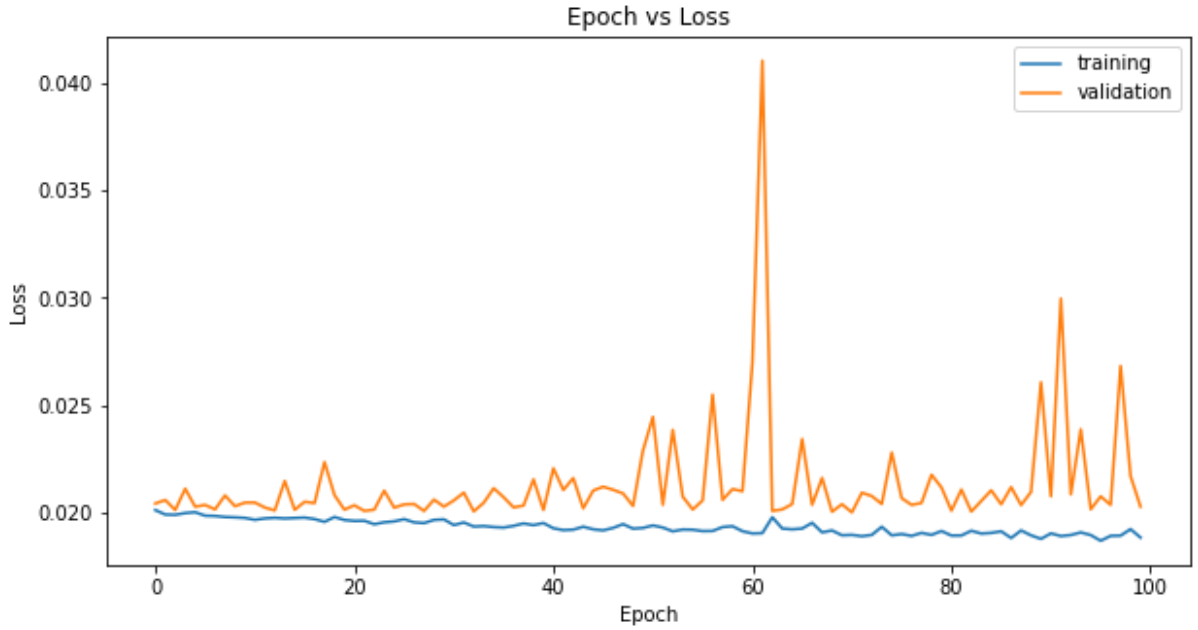
When learning rate = 0.005 momentum = 0.9 weight_decay = 5e-04

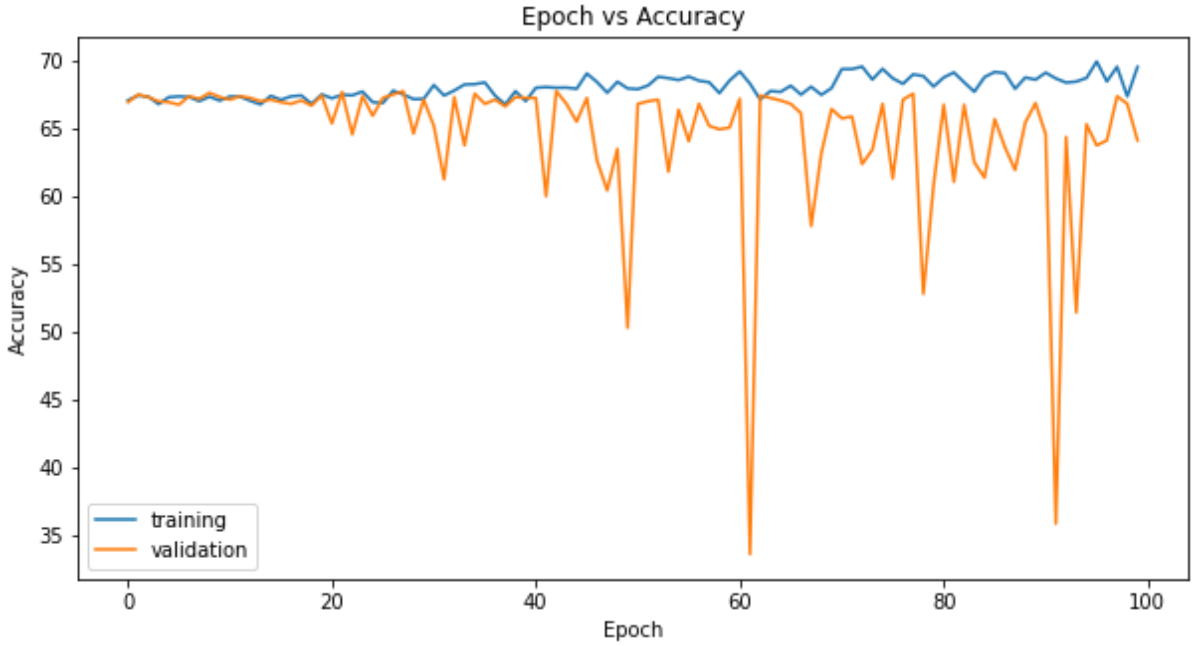




When learning rate = 0.05 momentum = 0.4 weight_decay = 2e-04

Plotting Results:

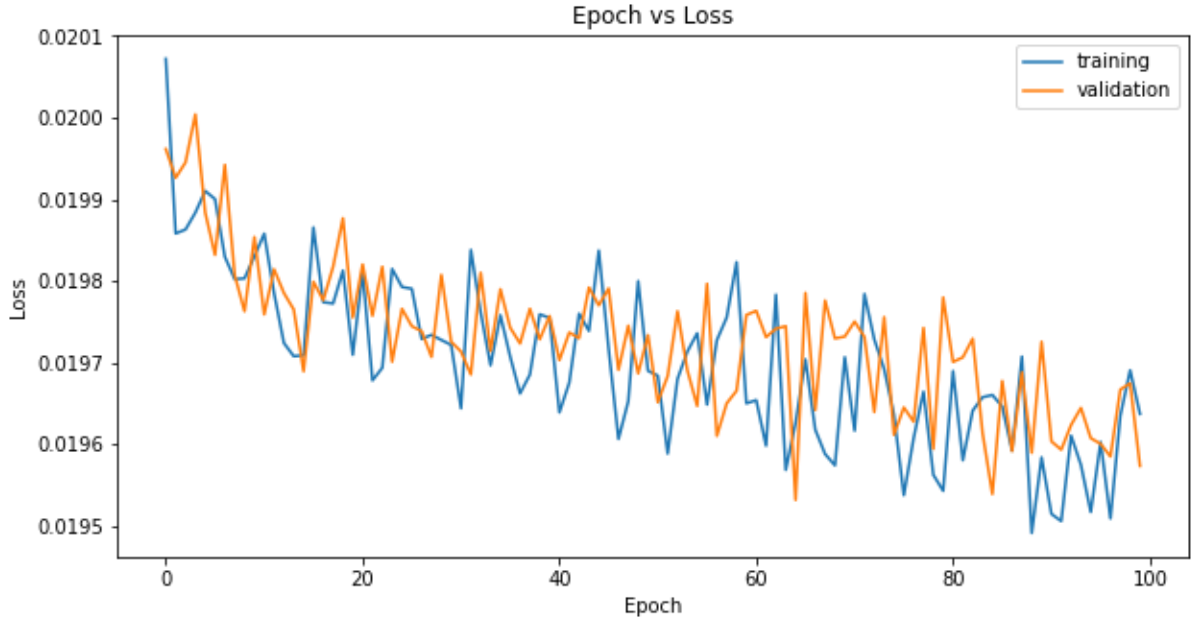


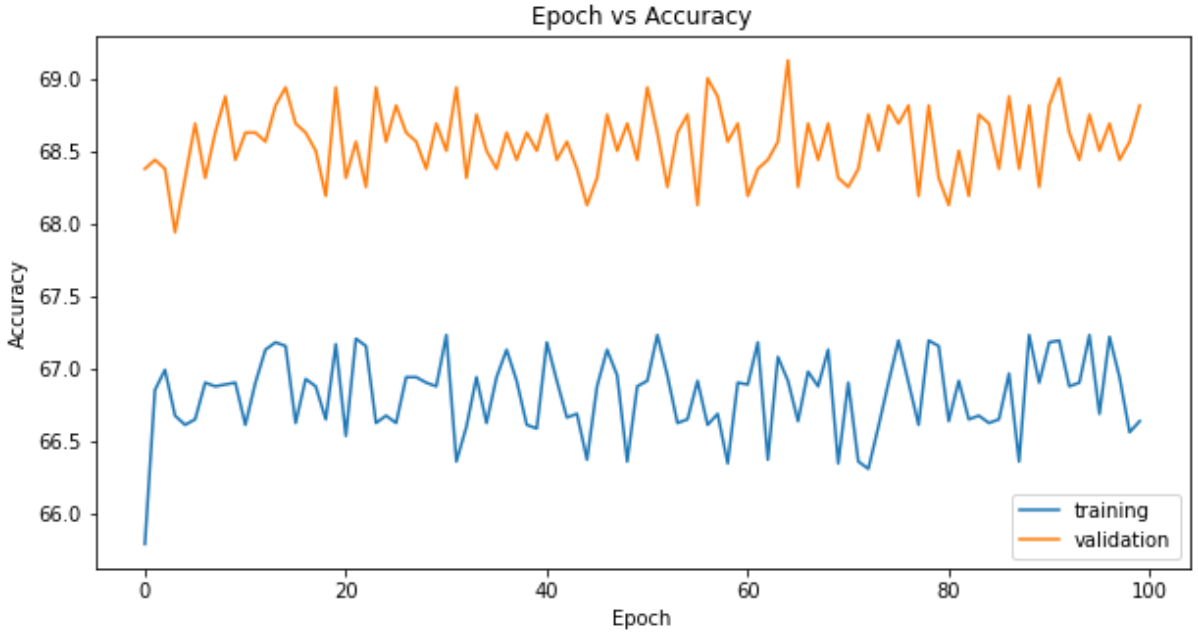


Testing Results:

Test set: Average loss: 0.6084, Accuracy:(68.19999694824219)

When learning rate = 0.0005 momentum = 0.8 weight_decay = 3e-02



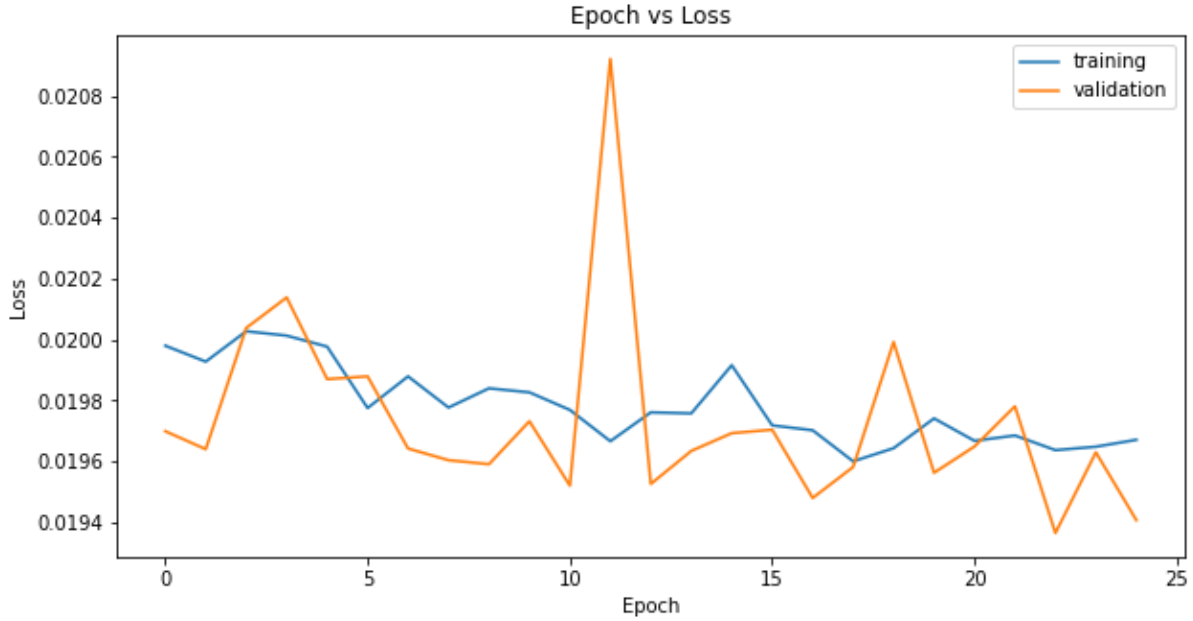


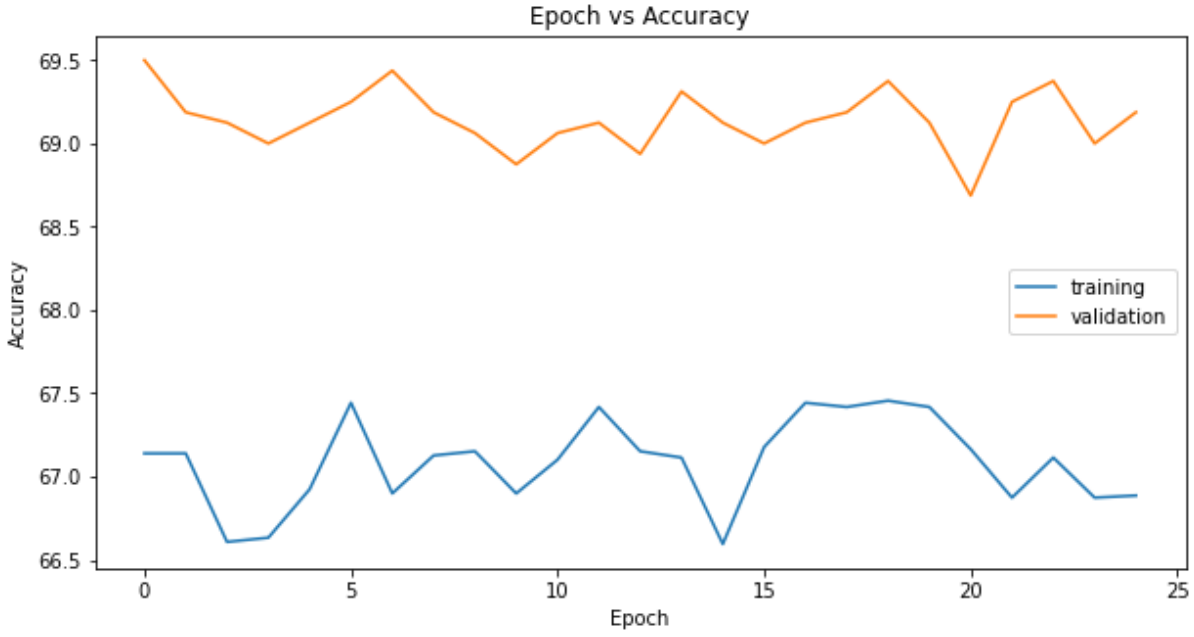
Test set: Average loss: 0.6078, Accuracy:(69.19999694824219)

Questin 2.2)

When learning rate = 0.005 momentum = 0.5 weight_decay = 2e-04

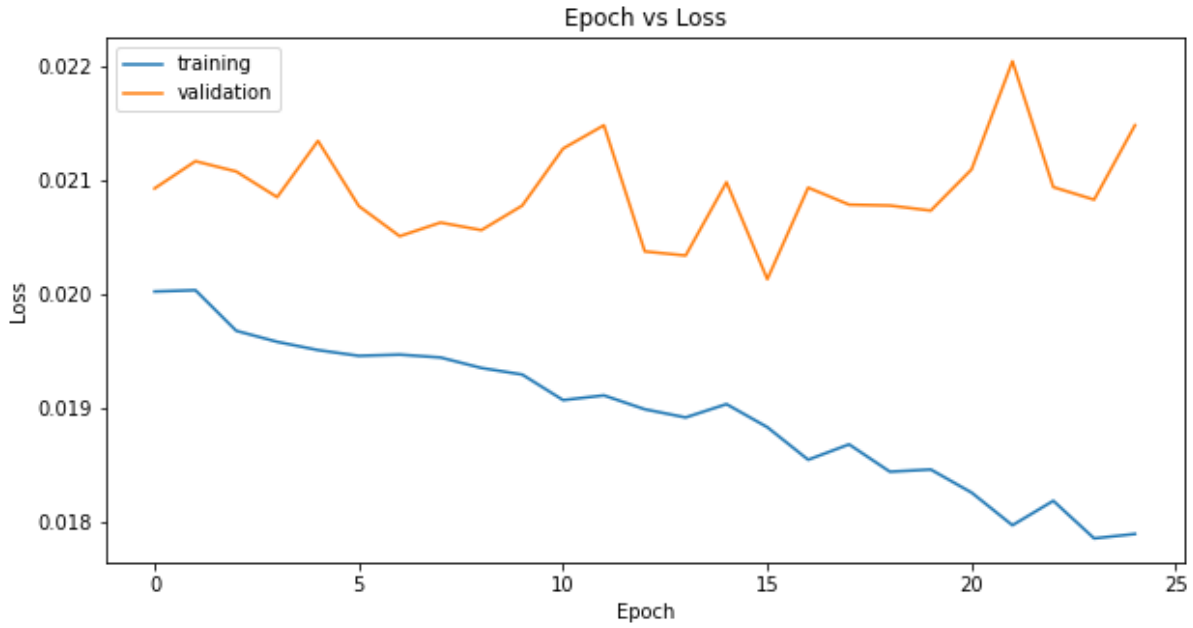
Plotting Results:

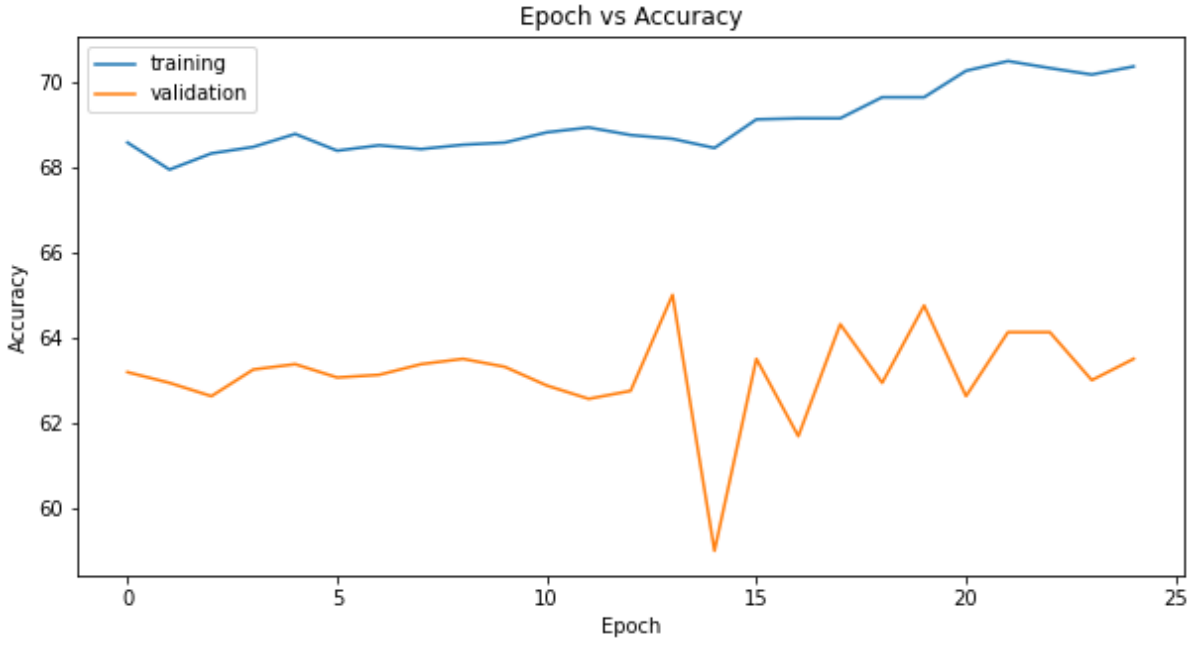




When learning rate = 0.05 momentum = 0.9 weight_decay = 6e-04

Plotting Results:





Questin 2.3)

1)

Since I was not enough time to see results for 100 epoch for CNN model I have not clear answers for this. But I will interpret according to result I have obtained.

- With CNN we could get better if I could see 100th epoch.

2)

Strengths of CNN

- Better accuracy

Weaknesses of CNN

- Comparatively slower than MLP
- Higher computational cost than MLP
- There might be overfitting in my results since the data provided may not be large enough for CNN model

Strengths of MLP

- A lot faster than CNN
- Computational inexpensive

3)

With max pooling operation we can extract the most important features (in other words eliminating most of the feature); with average pooling operation we keep more information and feature in comparison to max pooling.
We are using max pooling in order to get the most important features that affects the label of the image. If we used average pooling we may not be able to get the important features that helps us the predict the image correctly. Which might result with lower accuracy.