Deep learning Code :

FOREST TYPE MAPPING DEEP LEARNING :

```
-----------------------------------------------------
#install.packages("h2o")
#install.packages("itsmr")
library(itsmr)
library(h2o)
localH2o = h2o.init(ip = "localhost", port = 54321, startH2O = TRUE)

##########################################
#Forest Type mapping

traindata = read.csv("forestTraining.csv", header = T)
testdata=read.csv("forestTesting.csv", header = T)
data<-rbind(traindata,testdata)
labels<-data[,1]
labels<-as.factor(labels)
for(x in c(0.50,0.40,0.30,0.20,0.10)){
 smp_size <- floor(x * nrow(data))
 train_ind <- sample(seq_len(nrow(data)), size = smp_size)
 traindata<-data[train_ind,]
 #Randomly shuffle the data
 traindata<-traindata[sample(nrow(traindata)),]
 testdata<-data[-train_ind,]
 trainlabels<-labels[train_ind]
 testlabels<-labels[-train_ind]
 #Converting into h2o type data

 traindata_h2o<-as.h2o(traindata)
 trainlabels_h2o<-as.h2o(trainlabels)
 testdata_h2o<-as.h2o(testdata)
 testlabels_h2o<-as.h2o(testlabels)
 #Training the deep learning model
 dlModel<-h2o.deeplearning(x=2:28,y=1,training_frame = traindata_h2o,
                           activation = "Tanh",
                           hidden = c(50,50,50),  ## three hidden layers
                           epochs = 500)
 testPrediction<-h2o.predict(dlModel,testdata_h2o[,-1])
 predictResult<-as.vector(testPrediction$predict)
 label<-as.vector(testlabels_h2o)
 count=0
 ###################################
 #Accuracy
 for(i in 1:nrow(testdata_h2o)){
  if(predictResult[i]==label[i])count<-count+1
```

```
  }
  print(cat("accuracy percentage for",x*100,"%  of the training data is
",(count*100)/(nrow(testdata_h2o)),"%"))

}
------------------------------------------------------------
BREAST CANCER DATASET :
#install.packages("h2o")
library(h2o)
localH2o = h2o.init(nthreads=-1,ip = "localhost", port = 54321, startH2O = TRUE)

##############################################
#Breast Cancer dataset

data<-read.table("breast-cancer-wisconsin.data.txt",sep=',')
data<-data[,-1]
data<-na.omit(data)
labels<-data[,10]
labels<-as.factor(labels)
#Separating Data into test and training datasets

set.seed(1234)
for(x in c(0.50,0.40,0.30,0.20,0.10)){
smp_size <- floor(x * nrow(data))
train_ind <- sample(seq_len(nrow(data)), size = smp_size)
traindata<-data[train_ind,]
#Randomly shuffle the data
#traindata<-traindata[sample(nrow(traindata)),]
testdata<-data[-train_ind,]
trainlabels<-labels[train_ind]
testlabels<-labels[-train_ind]
#Converting into h2o type data

traindata_h2o<-as.h2o(traindata)
trainlabels_h2o<-as.h2o(trainlabels)
testdata_h2o<-as.h2o(testdata)
testlabels_h2o<-as.h2o(testlabels)
#Training the deep learning model
dlModel<-h2o.deeplearning(x=1:9,y=10,training_frame = traindata_h2o,
                  activation = "Tanh",
                  hidden = c(50,50,50),  ## three hidden layers
                  epochs = 500)
testPrediction<-h2o.predict(dlModel,testdata_h2o[,-10])
predictResult<-as.vector(testPrediction$predict)
label<-as.vector(testlabels_h2o)
count=0
###################################
#Accuracy
```

```
for(i in 1:nrow(testdata_h2o)){
 if(round(predictResult[i])==label[i])count<-count+1
}
print(cat("accuracy percentage for",x*100,"%  of the training data is
",(count*100)/(nrow(testdata_h2o)),"%"))

}
```

OPTICAL HANDWRITTEN DATASET

```
#install.packages("h2o")
library(h2o)
localH2o = h2o.init(ip = "localhost", port = 54321, startH2O = TRUE)

##########################################
#Optical handwritten digits dataset

data<-read.table("optdigits.tra.txt",sep=',')
testdata<-read.table("optdigits.tes.txt",sep=',')
data<-rbind(data,testdata)
labels<-data[,65]
set.seed(1234)
for(x in c(0.50,0.40,0.30,0.20,0.10)){
 smp_size <- floor(x * nrow(data))
 train_ind <- sample(seq_len(nrow(data)), size = smp_size)
 traindata<-data[train_ind,]
 #Randomly shuffle the data
 #traindata<-traindata[sample(nrow(traindata)),]
 testdata<-data[-train_ind,]
 trainlabels<-labels[train_ind]
 testlabels<-labels[-train_ind]
 #Converting into h2o type data

 traindata_h2o<-as.h2o(traindata)
 trainlabels_h2o<-as.h2o(trainlabels)
 testdata_h2o<-as.h2o(testdata)
 testlabels_h2o<-as.h2o(testlabels)
 #Training the deep learning model
 dlModel<-h2o.deeplearning(x=1:64,y=65,training_frame = traindata_h2o,
                           activation = "Tanh",
                           hidden = c(50,50,50),  ## three hidden layers
                           epochs = 500)
 testPrediction<-h2o.predict(dlModel,testdata_h2o[,-65])
 predictResult<-as.vector(testPrediction$predict)
 label<-as.vector(testlabels_h2o)
 count=0
 ###################################
 #Accuracy
```

```
  for(i in 1:nrow(testdata_h2o)){
    if(round(predictResult[i])==label[i])count<-count+1
  }
  print(cat("accuracy percentage for",x*100,"%  of the training data is
",(count*100)/(nrow(testdata_h2o)),"%"))

}
```

IRIS DEEP LEARNING :

```
#install.packages("h2o")
library(h2o)
localH2o = h2o.init(nthreads=-1,ip = "localhost", port = 54321, startH2O = TRUE)

##############################################
#Breast Cancer dataset
data(iris)
#data<-read.table("breast-cancer-wisconsin.data.txt",sep=',')
data<-iris
data<-na.omit(data)
labels<-iris[,5]
labels<-as.factor(labels)
#Separating Data into test and training datasets

set.seed(1234)
for(x in c(0.50,0.40,0.30,0.20,0.10)){
smp_size <- floor(x * nrow(data))
train_ind <- sample(seq_len(nrow(data)), size = smp_size)
traindata<-data[train_ind,]
#Randomly shuffle the data
traindata<-traindata[sample(nrow(traindata)),]
testdata<-data[-train_ind,]
trainlabels<-labels[train_ind]
testlabels<-labels[-train_ind]
#Converting into h2o type data

traindata_h2o<-as.h2o(traindata)
trainlabels_h2o<-as.h2o(trainlabels)
testdata_h2o<-as.h2o(testdata)
testlabels_h2o<-as.h2o(testlabels)
#Training the deep learning model
dlModel<-h2o.deeplearning(x=1:4,y=5,training_frame = traindata_h2o,
                  activation = "Tanh",
                  hidden = c(150,150,150),  ## three hidden layers
                  epochs = 500)
trainPrediction<-h2o.predict(dlModel,traindata_h2o)
testPrediction<-h2o.predict(dlModel,testdata_h2o[,-5])
predictResult<-as.vector(testPrediction$predict)
```

```
label<-as.vector(testlabels_h2o)
count=0
###################################
#Accuracy
for(i in 1:nrow(testdata_h2o)){
  if(predictResult[i]==label[i])count<-count+1
}
print(cat("accuracy percentage for",x*100,"%  of the training data is
",(count*100)/(nrow(testdata_h2o)),"%"))
}
```

ALPHABET RECOGNITION :

```
#install.packages("h2o")
library(h2o)
localH2o = h2o.init(ip = "localhost", port = 54321, startH2O = TRUE)

###############################################
#Alphabets data Set

data<-read.table("letter-recognition.data",sep=',')
data<-na.omit(data)
labels<-data[,1]
labels<-as.factor(labels)
#Separating Data into test and training datasets

set.seed(1234)
for(x in c(0.50,0.40,0.30,0.20,0.10)){
smp_size <- floor(x* nrow(data))
train_ind <- sample(seq_len(nrow(data)), size = smp_size)
traindata<-data[train_ind,]
#Randomly shuffle the data
traindata<-traindata[sample(nrow(traindata)),]
testdata<-data[-train_ind,]
trainlabels<-labels[train_ind]
testlabels<-labels[-train_ind]
#Converting into h2o type data

traindata_h2o<-as.h2o(traindata)
trainlabels_h2o<-as.h2o(trainlabels)
testdata_h2o<-as.h2o(testdata)
testlabels_h2o<-as.h2o(testlabels)
#Training the deep learning model
dlModel<-h2o.deeplearning(x=2:17,y=1,training_frame = traindata_h2o,
                          activation = "Tanh",
                          hidden = c(50,50,50),  ## three hidden layers
                          epochs = 10)
```

```
testPrediction<-h2o.predict(dlModel,testdata_h2o[,-1])
predictResult<-as.vector(testPrediction$predict)
label<-as.vector(testlabels_h2o)
count=0
####################################
#Accuracy
for(i in 1:nrow(testdata_h2o)){
  if(predictResult[i]==label[i])count<-count+1
}
print(cat("accuracy percentage for",x*100,"%  of the training data is
",(count*100)/(nrow(testdata_h2o)),"%"))
}

------------------------------------------------------------
TICTACTOE DATASET#install.packages("h2o")
library(h2o)
localH2o = h2o.init(nthreads=-1,ip = "localhost", port = 54321, startH2O = TRUE)

##############################################
#Tic Tac dataset

data<-read.table("ticTacToe.txt")
data<-na.omit(data)
labels<-data[,28]
labels<-as.factor(labels)
#Separating Data into test and training datasets

set.seed(1234)
for(x in c(0.50,0.40,0.30,0.20,0.10)){
  smp_size <- floor(x * nrow(data))
  train_ind <- sample(seq_len(nrow(data)), size = smp_size)
  traindata<-data[train_ind,]
  #Randomly shuffle the data
  traindata<-traindata[sample(nrow(traindata)),]
  testdata<-data[-train_ind,]
  trainlabels<-labels[train_ind]
  testlabels<-labels[-train_ind]
  #Converting into h2o type data

  traindata_h2o<-as.h2o(traindata)
  trainlabels_h2o<-as.h2o(trainlabels)
  testdata_h2o<-as.h2o(testdata)
  testlabels_h2o<-as.h2o(testlabels)
  #Training the deep learning model
  dlModel<-h2o.deeplearning(x=1:27,y=28,training_frame = traindata_h2o,
                            activation = "Tanh",
                            hidden = c(50,50,50),  ## three hidden layers
                            epochs = 500)
```

```
 testPrediction<-h2o.predict(dlModel,testdata_h2o[,-28])
 predictResult<-as.vector(testPrediction$predict)
 label<-as.vector(testlabels_h2o)
 count=0
 ###################################
 #Accuracy
 for(i in 1:nrow(testdata_h2o)){
  if(predictResult[i]==label[i])count<-count+1
 }
 print(cat("accuracy percentage for",x*100,"%  of the training data is
",(count*100)/(nrow(testdata_h2o)),"%"))

}


-------------------------------------------------------------
MAGIC DEEP LEARNING :
#install.packages("h2o")
library(h2o)
localH2o = h2o.init(ip = "localhost", port = 54321, startH2O = TRUE)

###############################################
#Tic Tac dataset

data<-read.table("magic04.data.txt",sep=",")
data<-na.omit(data)
labels<-data[,11]
labels<-as.factor(labels)
#Separating Data into test and training datasets

set.seed(1234)
for(x in c(0.50,0.40,0.30,0.20,0.10)){
 smp_size <- floor(x * nrow(data))
 train_ind <- sample(seq_len(nrow(data)), size = smp_size)
 traindata<-data[train_ind,]
 #Randomly shuffle the data
 traindata<-traindata[sample(nrow(traindata)),]
 testdata<-data[-train_ind,]
 trainlabels<-labels[train_ind]
 testlabels<-labels[-train_ind]
 #Converting into h2o type data

 traindata_h2o<-as.h2o(traindata)
 trainlabels_h2o<-as.h2o(trainlabels)
 testdata_h2o<-as.h2o(testdata)
 testlabels_h2o<-as.h2o(testlabels)
 #Training the deep learning model
 dlModel<-h2o.deeplearning(x=1:10,y=11,training_frame = traindata_h2o,
```

```
                    activation = "Tanh",
                    hidden = c(50,50,50),  ## three hidden layers
                    epochs = 500)
  testPrediction<-h2o.predict(dlModel,testdata_h2o[,-11])
  predictResult<-as.vector(testPrediction$predict)
  label<-as.vector(testlabels_h2o)
  count=0
  ####################################
  #Accuracy
  for(i in 1:nrow(testdata_h2o)){
    if(predictResult[i]==label[i])count<-count+1
  }
  print(cat("accuracy percentage for",x*100,"%  of the training data is
",(count*100)/(nrow(testdata_h2o)),"%"))

}
```