Exercise 4 - Multi Layer Perceptrons & Neural Networks (in R)

October 23, 2017

1. MLP prediction of hand-drawn digits (1 or 0)

```
#Author: Thomas Hollis
#Subject: Bachelor Thesis

#1. Data Import & Global Variables
data <- read.csv("R/DigitData.csv") #import data
alpha = 0.1 #initialises model parameter (learning rate)
bias = 0 #initialises a bias of 0
weights = integer(25) #initialises up 25 weights
y = integer(8) #initialises model output
result = integer(4) #initialises array of results

#2. Data post-processing
data_train <- data[1:8, ] #split data to training set
data_test <-data[9:12, ] #split data to testing set

#3. Forward propagate the NN model
for(i in 1:8)
{
  if((sum(t(data_train[i,2:26])*weights) + bias) > 0) #perceptron binary classification
equation
  {
    y[i] = 1 #guess digit 1
  }
  else
  {
    y[i] = -1 #guess digit 0
  }
  weights = weights + alpha*(data_train$Number[i] - y[i])*data_train[i,2:26] #update weights,
adapted from H. Yin instructions
  bias = bias + alpha*(data_train$Number[i] - y[i]) #update bias, from H. Yin instructions
}

#4. Use the trained model to predict
for(i in 1:4)
{
  if((sum(t(data_test[i,2:26])*weights) + bias) > 0) #use the final weights and bias to make
prediction
  {
    result[i] = 1
  }
  else
  {
    result[i] = 0
  }
}
result #prints the result
```

Final learn weights and bias: weighted ±0.2 or 0.6, with a bias of -0.2
Average performance over 10 runs: 1.0 (always predicts “1 1 0 0”, correctly)

2. NN prediction of hand-drawn digits (1-9, MNIST dataset)

```
MNistdata <- read.csv("MNIST_train.csv")

MNistTrain <- MNistdata[1:2000, ]
MNistTest <- MNistdata[2001:2500, ]

MNistTrain <- cbind(MNistTrain[, 2:785], class.ind(MNistTrain$label))
colLabel <- c("zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine")

colnames(MNistTrain)[785:794] <- colLabel

n <- names(MNistTrain[,1:784])
f <- as.formula(paste('zero + one + two + three + four + five + six + seven + eight + nine ~',
paste(n[!n %in% ''], collapse = ' + ')))

MNistModel <- neuralnet(f, data = MNistTrain, hidden = 30, linear.output = FALSE)

model_result <- compute(MNistModel, MNistTest[ ,1:784])

predicted <- model_result$net.result


MNistTest <- cbind(MNistTest[, 2:785], class.ind(MNistTest$label))
colnames(MNistTest)[785:794] <- colLabel

picks<-(0:9)[apply(predicted,1,which.max)]

prop.table(table(MNistTest[, 785:794] == picks))

#Needs further development
```

Final performance: 0.96