Homework 3 : Problem 4.17

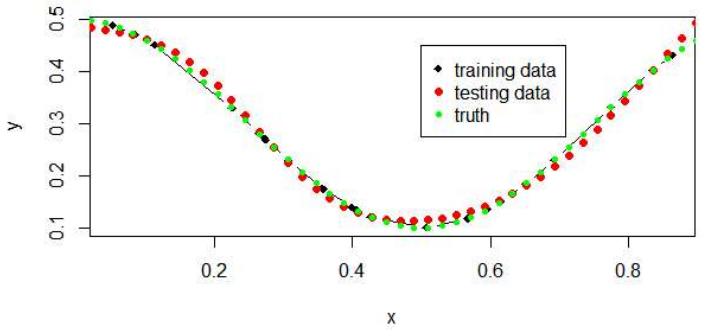
Two-layer perceptron with a linear output unit to approximate the function

$y(x) = 0.3 + 0.2 \cos(2 \pi x)$, $x \in [0, 1]$

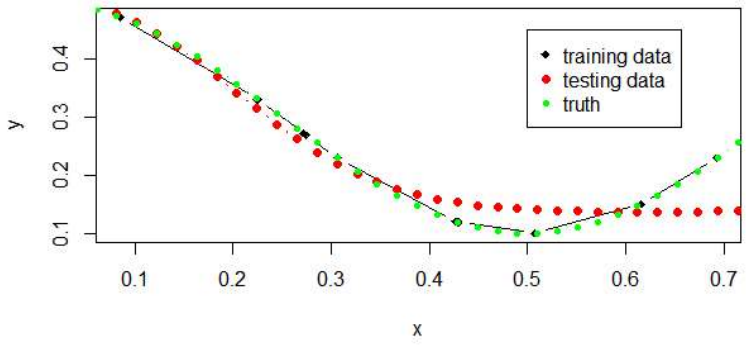
Use the backpropagation algorithm in one of its forms to train the network.

```
library(MASS)
library(nnet)

set.seed(10)

# Generate the data suggested (basically the XOR data set):
# Since x belongs to [0,1] we sample it using Uniform Distribution
nTrain = 30
X      = runif( nTrain )
X      = sort(X)
X      = data.frame(X)
YTrain = 0.3 + 0.2 * cos( 2 * pi * X )

# Generate NTest new vectors and predict their response:
XTest = seq(0.,1.,length=20)
XTest = sort( XTest )
XTest = data.frame( XTest )
YTest = 0.3 + 0.2 * cos( 2 * pi * XTest )

# Train a neural network on this data
regNet = nnet( X, YTrain, size=2, trace=FALSE ) # the example calling sequence
YPredict = predict( regNet, XTest ) # the predictions

matplot( X, YTrain, xlab='x', ylab='y', type="b", pch=18, col="black" )
matplot( XTest, YPredict, type="b", pch=19, col="red", add=T )
matplot( XTest, YTest, type="b", pch=20, col="green", add=T ) # the truth

legend( 0.5, 0.45, legend=c( "training data", "testing data", "truth" ), pch=c(18,19,20), col=c("blac

if( savePlots ){
  dev.off()
}
```

Use of 25 data points from this function for the training with 2 Hidden Units



Resulting output of 50 samples fed into the trained network with 2 Hidden Units



Resulting output of 50 samples fed into the trained network with 5 Hidden Units