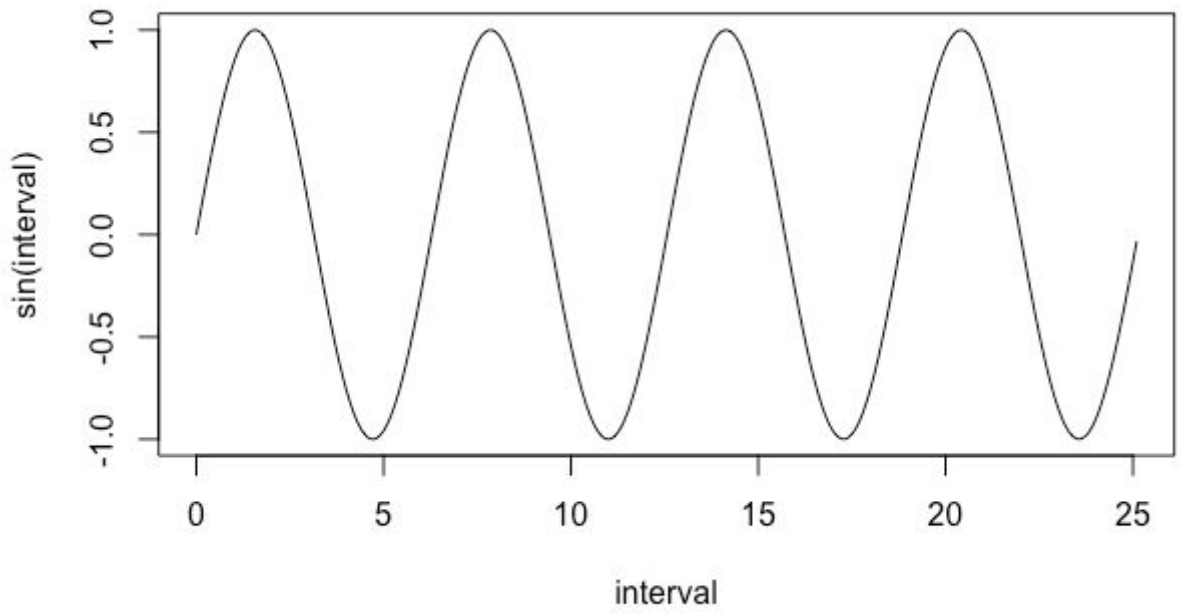
TDDE01: Lab 6

A report by

Adam Nyberg
adany869

Introduction

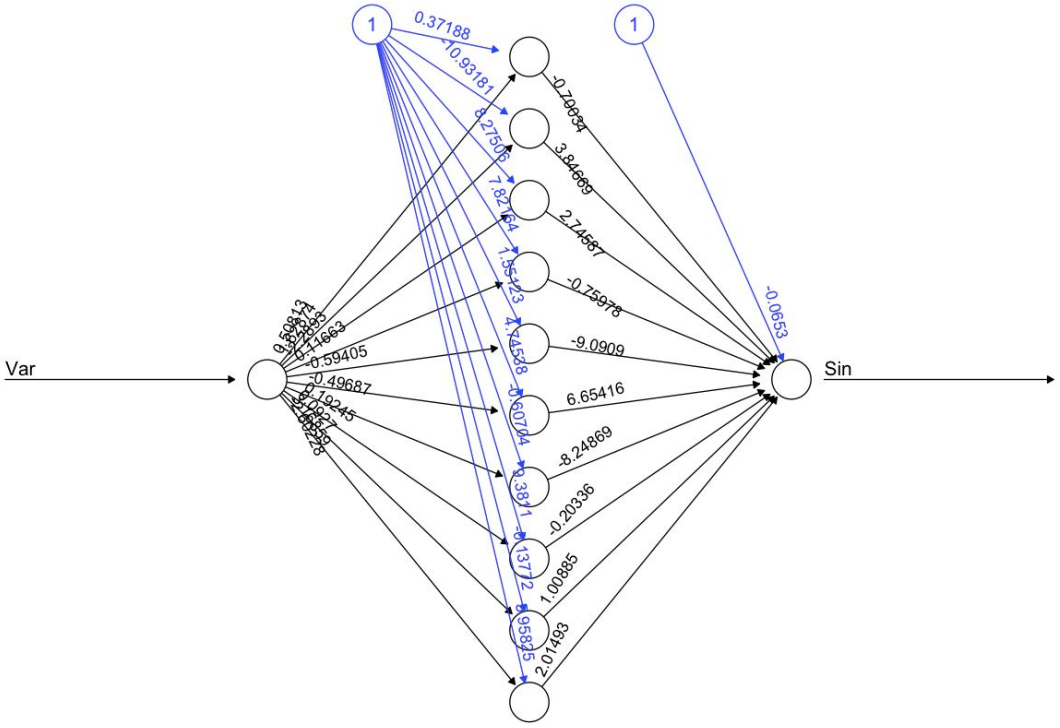
In this assignment I was to predict values for the trigonometric function $sin(x)$ by training a neural network. I did that by using the library neuralnets. When training the neural net I use the following parameters. One hidden layer with 10 units and sigmoid function as the activation function.



Choosing weight

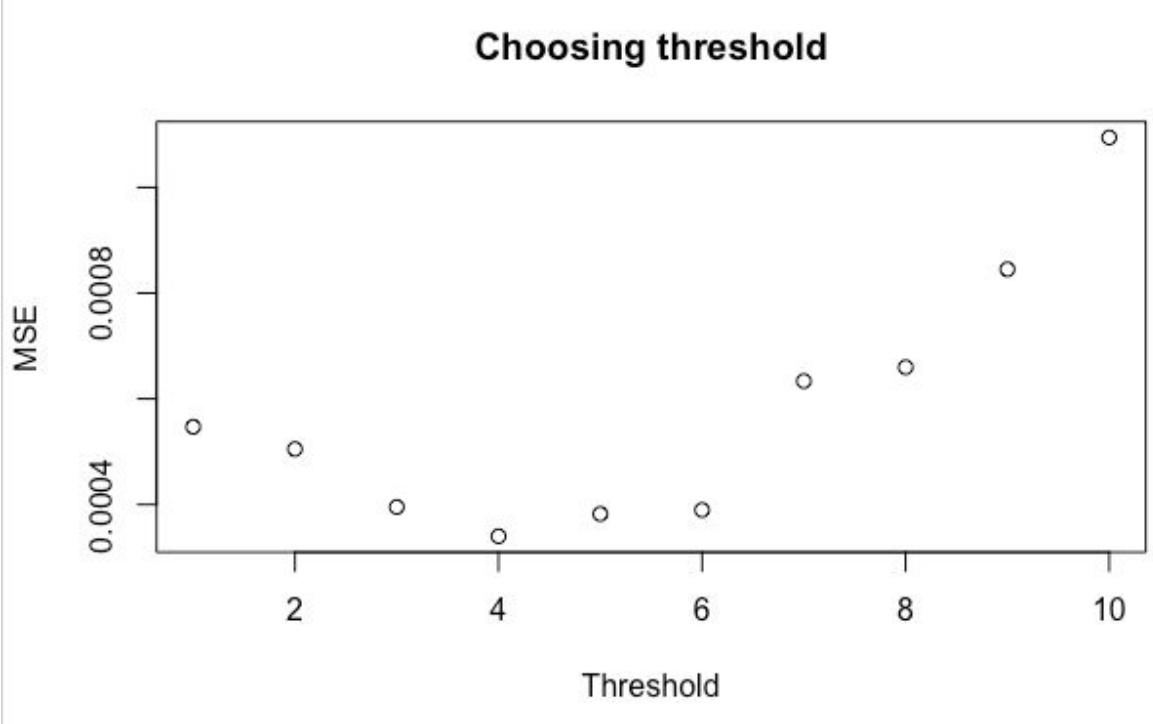
The task now was to chose the appropriate starting weights. I first counted the synaptic constants in the neural network and then created a vector containing 31 values between -1 and 1.

The plot of the neural network below is new. Previously I thought that setting hidden=c(10,1) would create a neural network with one hidden layer with 10 units. That's now corrected by setting hidden=10.



Choosing the threshold

Now I chose the best threshold and I did that by testing different values between 0.001 and 0.01 and then calculating MSE. That gave me the following plot.

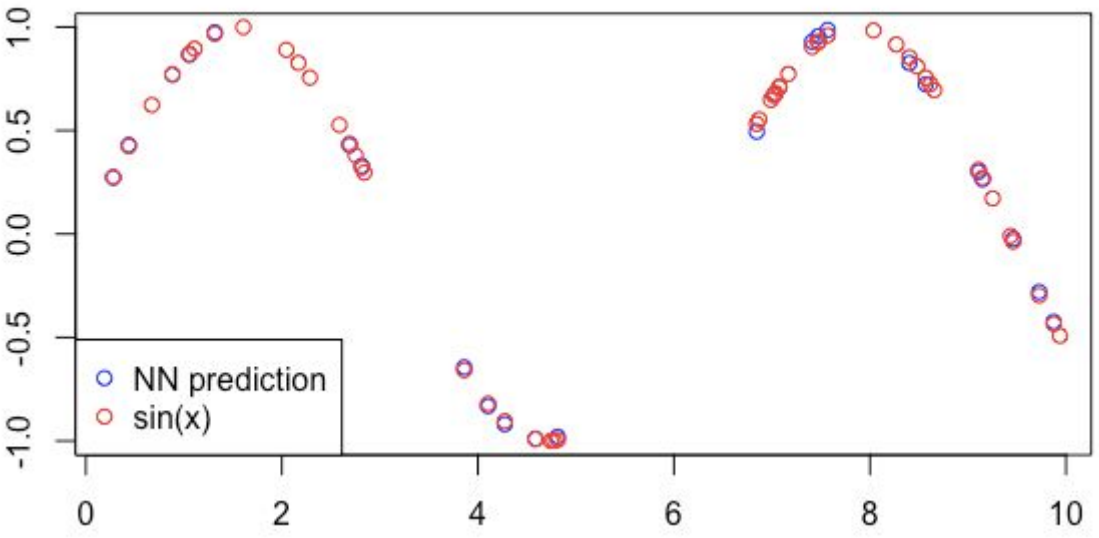


After inspecting the plot I chose the threshold 0.004 as the optimal.

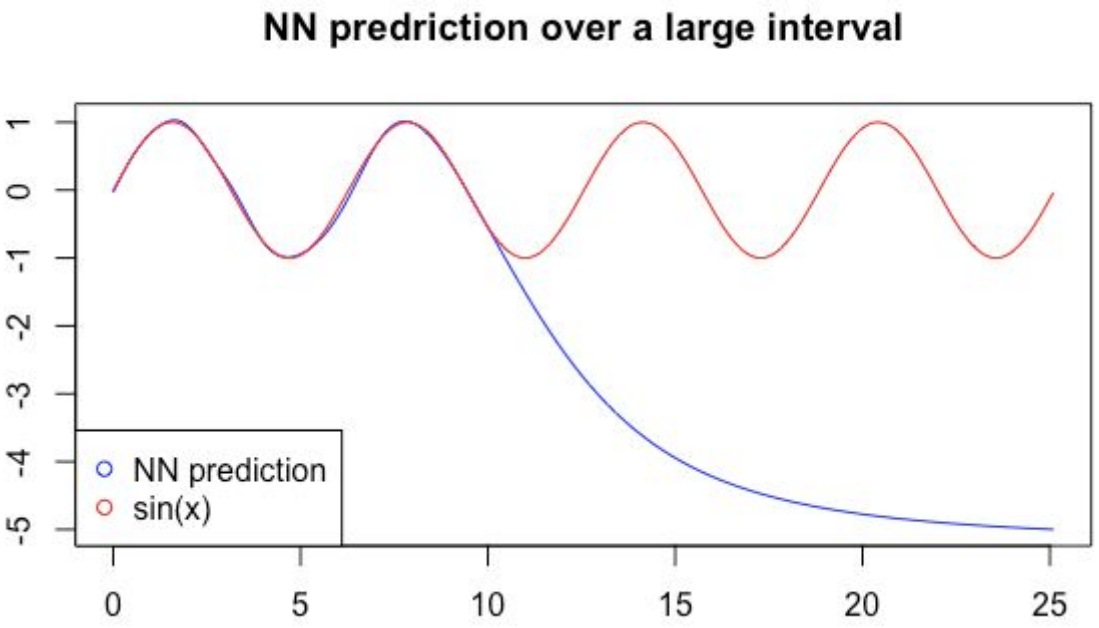
This plot above is new because the change of the neural network also changed this plot. The numbers changed so now the optimal threshold is 0.004.

Result

In the plot below we can see the final result of the sin function created by the neural net.



As we can see it looks pretty good. Just for fun I decided to try to use the neural net to create the sin function over a larger set of variables and that resulted in the plot below.



In this plot we see that the neurla net are only able to perdict values within the training data.

```
library(neuralnet)

set.seed(1234567890)
Var = runif(50, 0, 10)
trva = data.frame(Var, Sin=sin(Var))

train = trva[1:25,] # Training
validation = trva[26:50,] # Validation


# Random initializaiton of the weights in the interval [-1, 1]
winit = runif(31, -1, 1)# Your code here

MSEs = c()
bestMSE = Inf
bestNN = Inf
bestThreshold = 0

for(i in 1:10) {
  nn = neuralnet(Sin~Var, data=train, hidden = 10, threshold = (i/1000), startweights = winit)

  # Your code here
  preds = compute(nn, validation$Var)$net.result
  real = validation$Sin
  MSEs[i] = sum((real - preds)^2)/nrow(validation)

  if (MSEs[i] < bestMSE) {
    bestNN = nn
    bestMSE = MSE
    bestThreshold = i/1000
  }
}

plot(bestNN)

plot(1:10, MSEs, ylab = "MSE", xlab="Threshold", main = "Choosing threshold")

# Plot of the predictions (black dots) and the data (red dots)
plot(prediction(bestNN)$rep1, col="blue", ylab="", xlab="")
points(trva, col = "red")
legend("bottomleft", pch=c(1,1), col=c("blue", "red"), legend = c("NN prediction", "sin(x)"),
xpd = TRUE)


# Plot predicted over sequence
interval = seq(0, 8*pi, 0.1)
```

```
plot(interval, compute(bestNN, interval)$net.result, type="l", col="blue",  ylab="", xlab="",
main="NN predriction over a large interval")
points(interval, sin(interval), type="l", col="red")
legend("bottomleft", pch=c(1,1), col=c("blue", "red"), legend = c("NN prediction", "sin(x)"),
xpd = TRUE)
```