lab6_EmilKlassonSvensson

Emil K Svensson

17 December 2016

Assignment

```
library(neuralnet)
set.seed(1234567890)
Var <- runif(50, 0, 10)
trva <- data.frame(Var, Sin=sin(Var))


tr <- trva[1:25,] # Training
va <- trva[26:50,] # Validation

# # Random initializaiton of the weights in the interval [-1, 1]
winit <- runif(31,-1,1)

MSE.nn <-c()
#MSE.nn.train <-c()
for(i in 1:10) {

  nn<- neuralnet(Sin ~ Var,data = tr, hidden = 10, startweights = winit,
                 threshold = i/1000)

  pr.nn <- compute(nn,va$Var)
  pr.nn.tr <- compute(nn,tr$Var)

  # MSE.nn.train[i] <- sum((tr$Sin - pr.nn.tr$net.result)^2) / nrow(tr)
  MSE.nn[i] <- sum((va$Sin - pr.nn$net.result)^2) /nrow(va)

  if (i > 1 && MSE.nn[i] > MSE.nn[i - 1]) {
    paste("Gradiant descent has decended at itteration: ",i)
  break()
  }
}


nn <- neuralnet(Sin ~ Var,data = trva, hidden = 10, startweights = winit,
                threshold = 4/1000)
#plot(nn) dosn't work in markdown.
```

Here we can see a plot over the neural network and how data is feeded through the inuput nodes to the hidden layers and forward to the output. The number of randomly choosen initialization weights were chosen from the number of input nodes, hidden layers, output nodes and one extra for the bias-function (also called intercept sometimes). In total there were 31 of these and so the number of generated weights were 31.

```
# Your code here
# Plot of the predictions (black dots) and the data (red dots)
x <- prediction(nn)$rep1[,1]
```

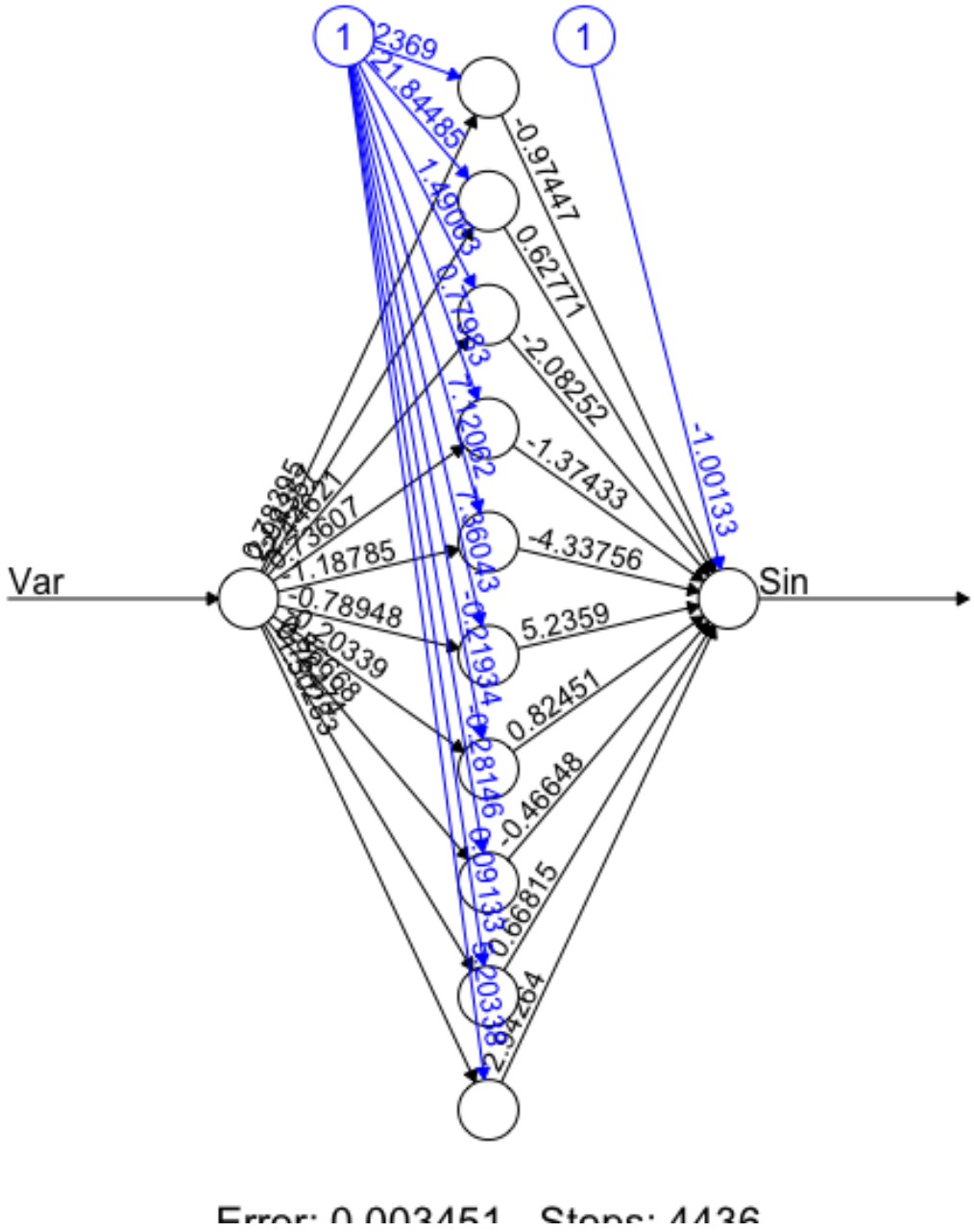

1
1
Var
Sin
-0.97447
0.62771
-2.08252
-1.37433
-4.33756
5.2359
0.82451
-0.46648
0.66815
-1.00133
-1.18785
-0.78948
-0.20339
Error: 0.003451 Steps: 4436

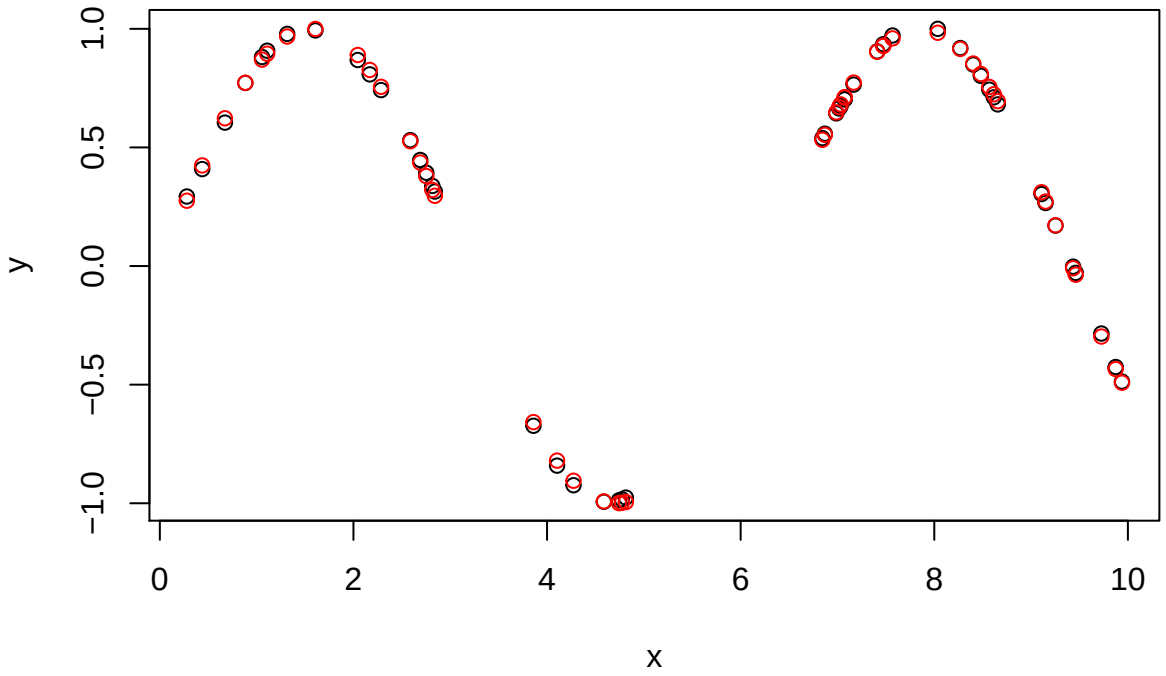
Figure 1:

```
## Data Error:  0;
```

```
y<-prediction(nn)$rep1[,2]
```

```
## Data Error:  0;
```

```
plot(x,y)
points(trva, col = "red")
```



The fitted values seem to follow the sine-function wery wery well.