Lab 6 732A95

Anton Persson antpe404

13 december 2016

Neural network and deep learning

In this lab I implemented a neural network with a single hidden layer with 10 units. The purpose is for the neural network to learn the trigonometric sin function using the sin function on 25 random points in the interval [0, 10]. An additional 25 points is used as validation. The threshold is defined as i/1000, where i = 1,. . . ,10 as requested in the instructions. The code is visualized in the chunk below.

```
points<-50
units<-10

set.seed(1234567890)
Var <- runif(n=points, min=0, max=10)
trva <- data.frame(Var, Sin=sin(Var))

tr <- trva[1:25,] # Training
va <- trva[26:50,] # Validation
felen<-vector(, length=10)

# Random initializaiton of the weights in the interval [-1, 1]
winit <- runif(n = 3*units+1 , min=-1, max=1)
#10 för input, 10 hidden, 10 output, 1 för bias.

 for(i in 1:10) {
    nn <- neuralnet(formula = Sin~Var, data=tr, threshold = i/1000,
                    hidden=10, startweights = winit )
      # Your code here
    preds<-compute(nn,va$Var)
    felen[i]<-(sum((preds$net.result-va$Sin)**2))/nrow(va)

    if (i > 1 && felen [i] > felen[i-1]){
      cat("No more progress, iteration stops")
      break

    }
 }
```

```
## No more progress, iteration stops
```

As seen above, the loops has stopped since the MSE hasn't improved more than the set threshold, and the code seems to work so far. To report the chosen value and it's respective neural network, the following code is used.

```r
#Report the chosen value for the threshold
  thresholds<-seq(1/1000, 1/100, 0.001)
  chosen_value<-thresholds[length(felen[felen>0])-1]

  #Will now report the final NN learnt
  final_nn <- neuralnet(formula = Sin~Var, data=trva, threshold = chosen_value, hidden=10, startweights

  #plot(final_nn) Doesnt work

  chosen_value
```

```
## [1] 0.004
```

So, the output of the chunk tells me that the chosen value for the treshhold is 4/1000. The plot of the actual neural network is shown on the next page, to be able to visualize the whole picture as good as possible.

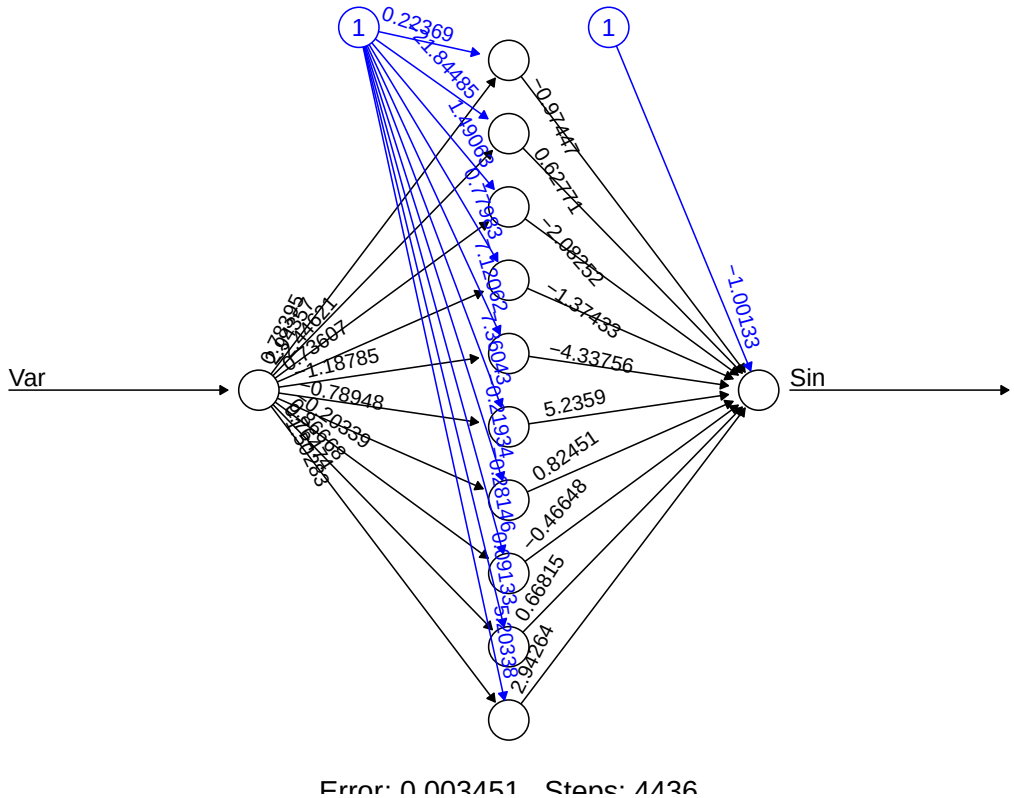


Figure 1:

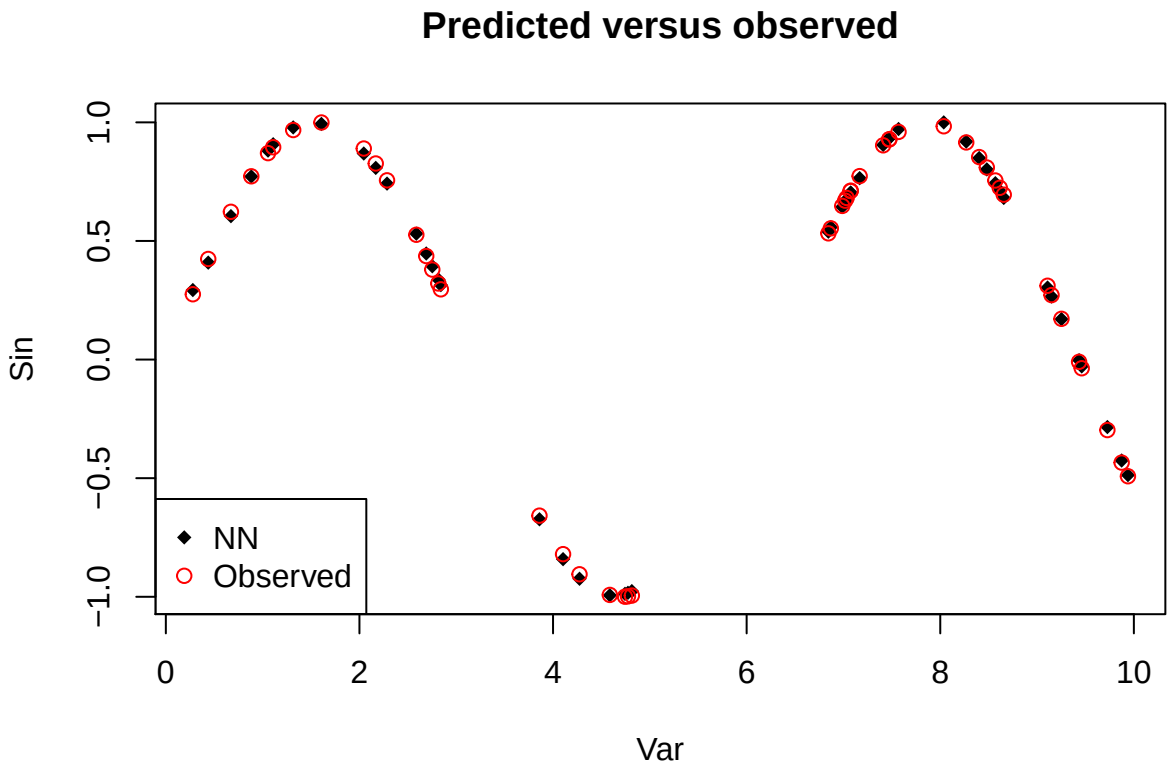
The black lines and values represent the connections and the respective weight for each connection in the network. The blue lines and their respective values represent the added bias in each step of the network.

To plot the final results, i.e. the predicted values of the neural network and the observed data, I used the code below.

```
  plot(prediction(final_nn)$rep1, pch=18)
```

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

```
  points(trva, col="red", pch=1)
  legend('bottomleft',legend=c('NN','Observed'),pch=c(18,1),col=c('black','red'))
  title("Predicted versus observed")
```



The plot above shows the observed values, all 50 points, as red circles. The black boxes in the plot represent the predicted values from the neural network. The fit seems really good, I'd say that the network seems to have learnt the trigonometric sin function.

Finally, note that all the code used to solve this task is shown in this report, why I haven't attached them in an appendix.