TDDE01 – Machine Learning
Laboration Report 6

Martin Estgren <mares480>
Linköping University (LiU), Sweden

December 18, 2016

For this assignment we will examine how a *neural net* can be used to approximate the *sinusoidal function* with *resilient back propagation* using the *batched gradient descent function* (equation:2). The *neural net* will consist of one hidden layer with 10 *perceptions* (equation:3), all using the *logistic function* (equation:1) as *activation function*.

$$f(x) = \frac{1}{1 + e^{-k(x-x_0)}} \tag{1}$$

$$\boldsymbol{w}_{(i)} = \boldsymbol{w}_{(i-1)} - \eta_k \nabla E(\boldsymbol{w}_{(i-1)}) \tag{2}$$

$$\hat{y}_j(\boldsymbol{x}) = \sigma(w_0 + \sum_{h=1}^{H} \sigma(w_{0h} + \boldsymbol{w}_h^\intercal \boldsymbol{x})) \tag{3}$$

First a sample data set is generated from a uniform random distribution in the range 0...10, 50 samples are generated and the sinusoidal value stored as for evaluation and training. THe set of 50 observations are then split into one training set and one validation set of equal size.

```
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
```

To find the best *neural net* we create 10 nets with different *threshold* as and initial weights set as uniform random samples between -1 and 1. For each of the nets the *means squared error* (M.S.E) is calculated with the validation data set.

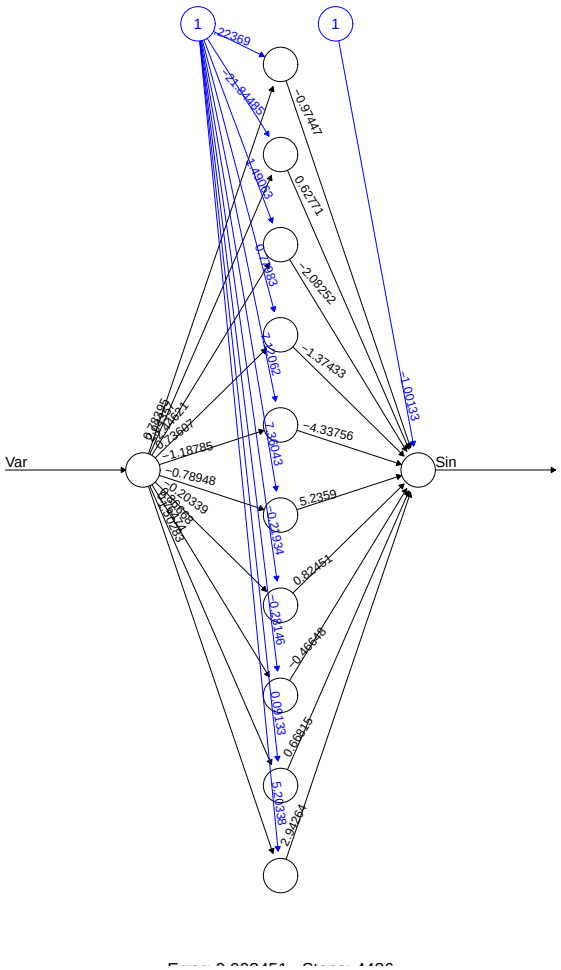
```
winit <- runif(250,-1,1)
  for(i in 1:10) {
    nn <- neuralnet(formula = Sin ~ Var, data=tr, hidden = 10,
    threshold = i/1000 ,startweights = winit)
    result = compute(nn, va$Var)$net.result
    results[i] = mean((result - va$Sin)^2)
}
```

The best net is selected and re-run on the entire generated data set. For this data set the optimal threshold was found to be 0.004.

```
best = which.min(results)
nn <- neuralnet(formula = Sin ~ Var , data=trva, hidden = 10, threshold
     = best/1000, startweights = winit)
```

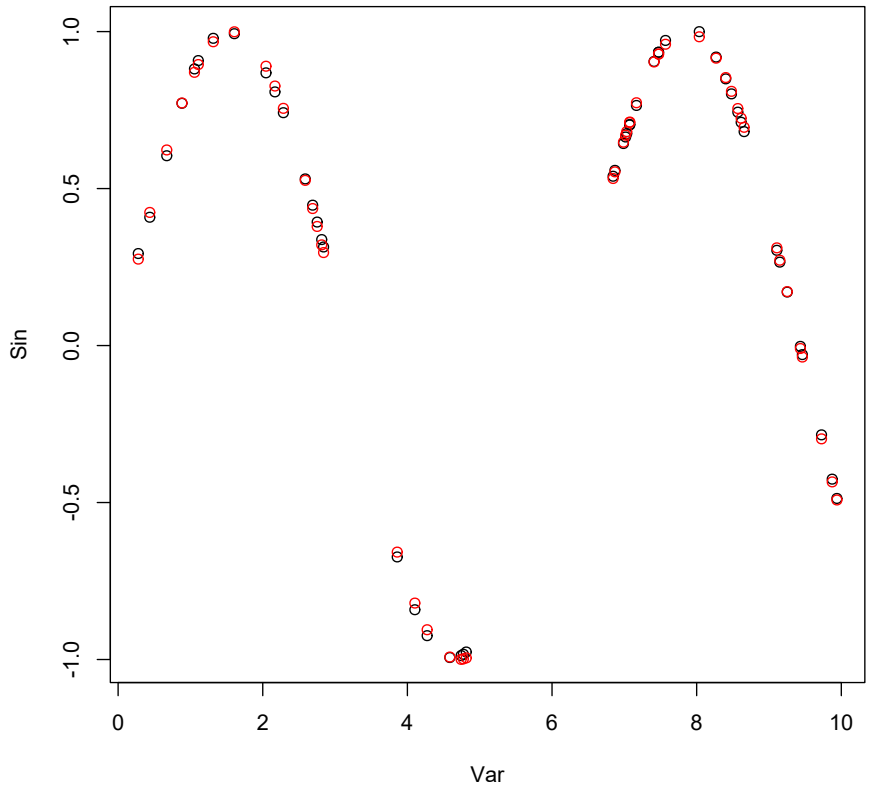
The resulting prediction of the net can be seen in figure 2 and a visual representation of the best neural net in figure 1.

Figure 1: (Poor) Visual representation of the best neural network.



The figure above shows a graph representation of the neural net. The black numbers are the individual weights and the blue ones are the intercept for each node.

Figure 2: Neural net predictions, the red values are the predictions and the black predicted.



The resulting predictions are very close the ones in the original data, indicating that the neural net has produced a good approximation of the sinusoidal function.

The full code for this assignment can be found in the appendix on the next page.

References

[FHT09] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning*. Springer series in statistics, Berlin, second (11th) edition, 2009.

[GF10] Frauke Günther and Stefan Fritsch. neuralnet: Training of Neural Networks. *The R Journal*, 2(1):30–38, 2010.

Appendix

Listing 1: Code for the assignment

```
library(neuralnet)
library("grDevices")

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]
results = rep(0,10)
print(results)
winit <- runif(250,-1,1)
  for(i in 1:10) {
    nn <- neuralnet(formula = Sin ~ Var, data=tr, hidden = 10,
    threshold = i/1000 ,startweights = winit)
    result = compute(nn, va$Var)$net.result
    results[i] = mean((result - va$Sin)^2)
}
best = which.min(results)
nn <- neuralnet(formula = Sin ~ Var , data=trva, hidden = 10, threshold
     = best/1000, startweights = winit)
plot(nn)
# Plot of the predictions (black dots) and the data (red dots)
setEPS()
cairo_ps("predictions.eps")
  plot(prediction(nn)$rep1, col="Black")
  points(trva, col = "red")
  dev.off()
```