Computer lab 3

Martynas Lukosevicius, Alejo Perez Gomez, Shwetha Vandagadde Chandramouly

13/12/2020

Statement of Contribution

- Assignment 1 - Alejo Perez Gomez
- Assignment 2 - Martynas Lukosevicius
- Assignment 3 - Shwetha Vandagadde Chandramouly

2. SUPPORT VECTOR MACHINES

Test results:

	error	trained on	tested on
filter0	0.07	train	validation
filter1	0.0848938826466916	train	test
filter2	0.083645443196005	train + validation	test
filter3	0.0337078651685393	all data (spam)	test

1. Which model do we return to the user ? filter0, filter1, filter2 or filter3 ? Why ?

Model should be returned with smallest error, however its hard to comapre filter3 with others because it was tested on data which was used for training. Also filter0 was not tested on test dataset, as a result error might differ.

Taking into account all arguments, filter2 should be selected as it was trained on bigger dataset (test + validation) and error is smallest between those which can be compared. (filter1 and filter2)

2. What is the estimate of the generalization error of the model selected ? err0, err1, err2 or err3 ? Why ?

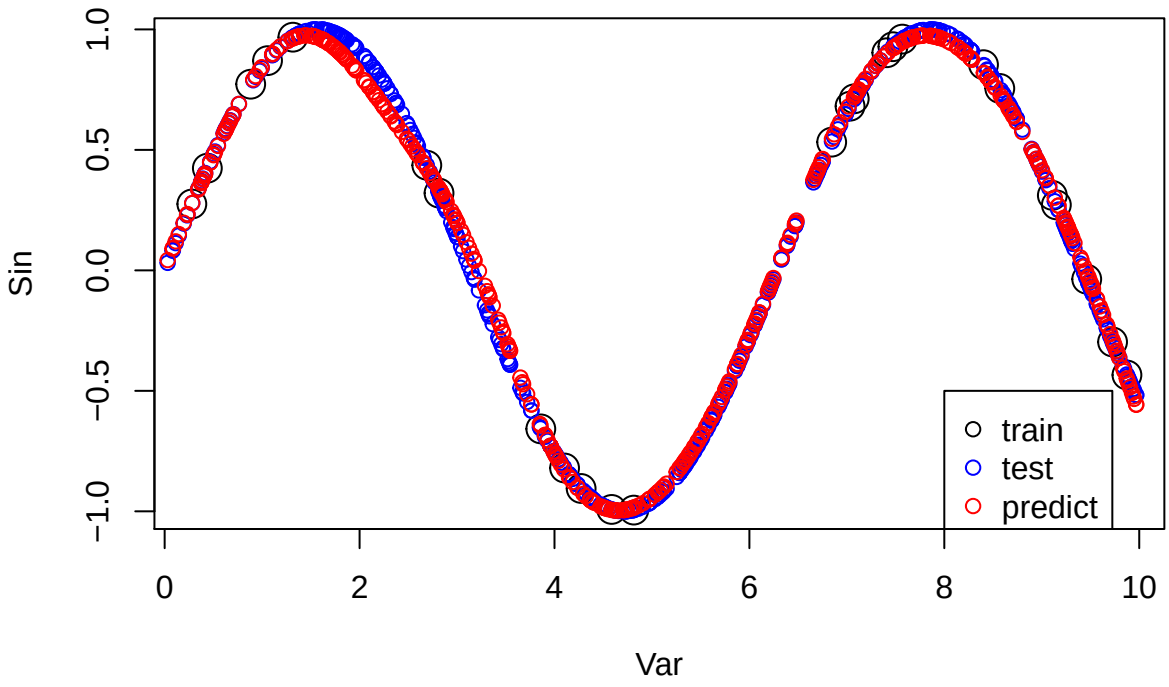
Generalization error is the error of the model on data which was not used for training, so err3 is not a generalization error. For chosen model (filter2) generalization error is err2 - 0.0836454

3. NEURAL NETWORKS

1.

Fitting neural network to learn the trigonometric sine function

```
winit = runif(46,-0.5,0.5)
nn = neuralnet(Sin~Var,tr,hidden = 15,startweights = winit)
train_predict = predict(nn,tr)
test_predict = predict(nn,te)
mse = mean(((te[,2]-test_predict)^2)/475)
```



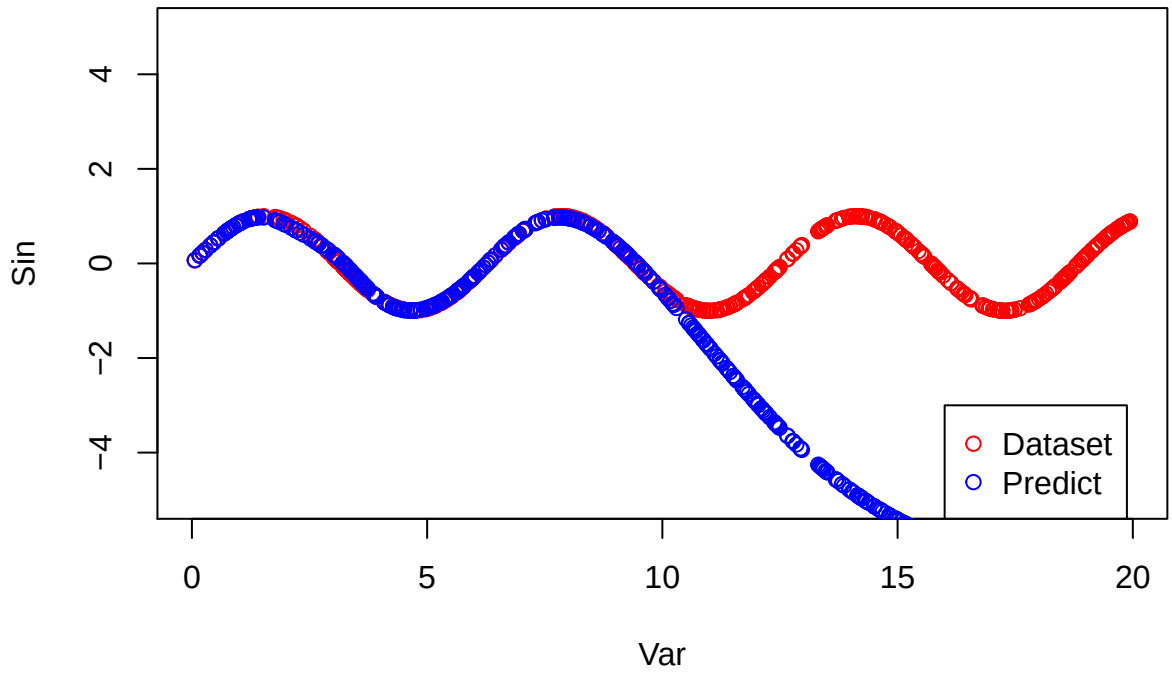
From the above graph we can see that prediction from the neural network on testdata is pretty good with a low mean square error of 2.9393626×10^{-6}

2.

Fitting the neural network for test data of range [0,20]

```
set.seed(1234567890)
Var <- runif(500, 0, 20)
newdata <- data.frame(Var, Sin=sin(Var))
new_predict = predict(nn,newdata)
plot(newdata,col="red", ylim=c(-5,5))
```

```
points(newdata$Var,new_predict,col="blue")
legend(16,-3,legend = c("Dataset","Predict"),pch = 1,col = c("red","blue"))
```

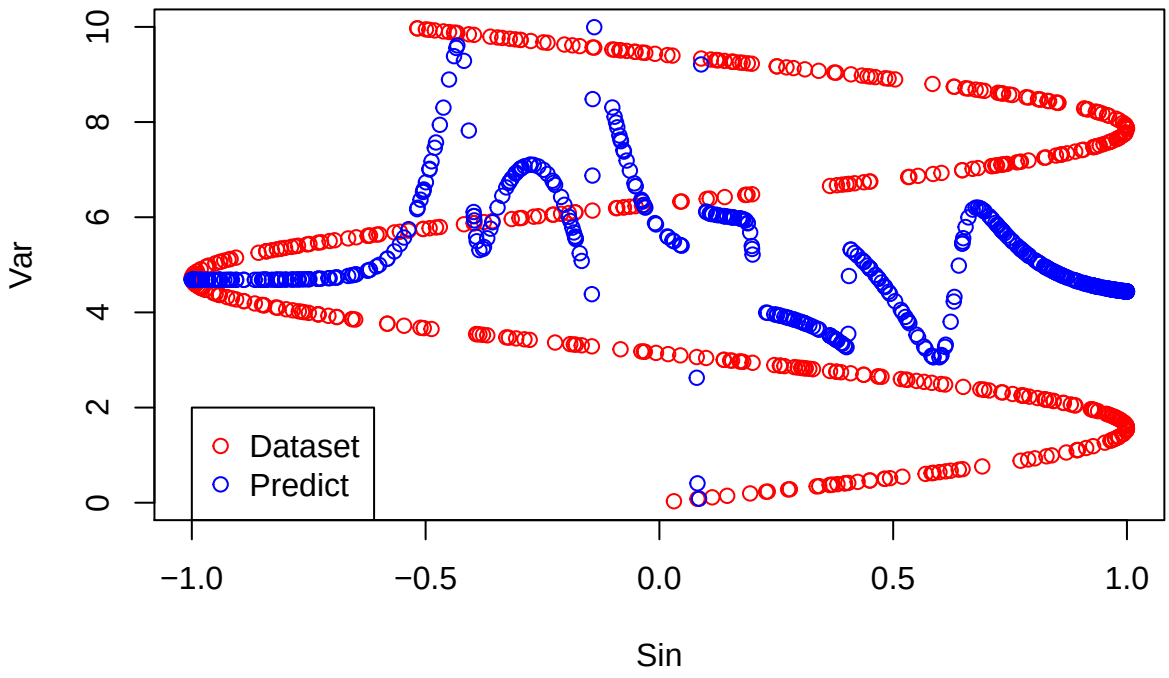


Prediction goes wrong after for the values which are not in training range this is because neural networks are not extrapolation methods. That is to say for regions of the variable space where no training data is available, the output of a neural network is not reliable.

3.

Neural net to predict x from sin(x)

```
set.seed(1234567890)
Var <- runif(500, 0, 10)
newdata <- data.frame(Var, Sin=sin(Var))
w = runif(31,-0.1,0.1)
nn = neuralnet(Var~Sin,newdata,hidden = 10,stepmax=1e8,startweights = w)
```



Sin of a variables oscillates from -1 to 1, from the graph we can see that multiple values of variables give out same output when applied with sine function, this makes it harder for neural network to predict the independent variable from the response of sine function. As result we dont get a good prediction here.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
# Lab 3 block 1 of 732A99/TDDE01 Machine Learning
# Author: jose.m.pena@liu.se
# Made for teaching purposes

library(kernlab)
set.seed(1234567890)

data(spam)

index <- sample(1:4601)
tr <- spam[index[1:3000], ]
va <- spam[index[3001:3800], ]
trva <- spam[index[1:3800], ]
te <- spam[index[3801:4601], ]

by <- 0.3
err_va <- NULL
```

```
for(i in seq(by,5,by)){
  filter <- ksvm(type~.,data=tr,kernel="rbfdot",kpar=list(sigma=0.05),C=i)
  mailtype <- predict(filter,va[,-58])
  t <- table(mailtype,va[,58])
  err_va <-c(err_va,(t[1,2]+t[2,1])/sum(t))
}

filter0 <- ksvm(type~.,data=tr,kernel="rbfdot",kpar=list(sigma=0.05),C=which.min(err_va)*by)
mailtype <- predict(filter0,va[,-58])
t <- table(mailtype,va[,58])
err0 <- (t[1,2]+t[2,1])/sum(t)


filter1 <- ksvm(type~.,data=tr,kernel="rbfdot",kpar=list(sigma=0.05),C=which.min(err_va)*by)
mailtype <- predict(filter1,te[,-58])
t <- table(mailtype,te[,58])
err1 <- (t[1,2]+t[2,1])/sum(t)


filter2 <- ksvm(type~.,data=trva,kernel="rbfdot",kpar=list(sigma=0.05),C=which.min(err_va)*by)
mailtype <- predict(filter2,te[,-58])
t <- table(mailtype,te[,58])
err2 <- (t[1,2]+t[2,1])/sum(t)


filter3 <- ksvm(type~.,data=spam,kernel="rbfdot",kpar=list(sigma=0.05),C=which.min(err_va)*by)
mailtype <- predict(filter3,te[,-58])
t <- table(mailtype,te[,58])
err3 <- (t[1,2]+t[2,1])/sum(t)



# Questions

# 1. Which model do we return to the user ? filter0, filter1, filter2 or filter3 ? Why ?

# 2. What is the estimate of the generalization error of the model selected ? err0, err1, err2 or err3
res <- rbind(c(err0, "train", "validation"),
      c(err1, "train", "test"),
      c(err2, "train + validation", "test"),
      c(err3, "all data (spam)", "test"))
row.names(res) <- c("filter0", "filter1", "filter2", "filter3")
colnames(res) <- c("error", "trained on", "tested on")
knitr::kable(res)
library(neuralnet)

set.seed(1234567890)
Var <- runif(500, 0, 10)
mydata <- data.frame(Var, Sin=sin(Var))
tr <- mydata[1:25,] # Training
te <- mydata[26:500,] # Test
winit = runif(46,-0.5,0.5)
nn = neuralnet(Sin~Var,tr,hidden = 15,startweights = winit)
```

```
train_predict = predict(nn,tr)
test_predict = predict(nn,te)
mse = mean(((te[,2]-test_predict)^2)/475)
plot(nn)
plot(tr, cex=2)
points(te, col = "blue", cex=1)
points(te[,1],predict(nn,te), col="red", cex=1)
legend(8,-0.5,legend = c("train","test","predict"),pch = 1,col = c("black","blue","red"))
set.seed(1234567890)
Var <- runif(500, 0, 20)
newdata <- data.frame(Var, Sin=sin(Var))
new_predict = predict(nn,newdata)
plot(newdata,col="red", ylim=c(-5,5))
points(newdata$Var,new_predict,col="blue")
legend(16,-3,legend = c("Dataset","Predict"),pch = 1,col = c("red","blue"))

set.seed(1234567890)
Var <- runif(500, 0, 10)
newdata <- data.frame(Var, Sin=sin(Var))
w = runif(31,-0.1,0.1)
nn = neuralnet(Var~Sin,newdata,hidden = 10,stepmax=1e8,startweights = w)
plot(nn)
plot(newdata[,2],newdata[,1],col="red",xlab = "Sin",ylab = "Var")
points(newdata[,2],predict(nn,newdata), col="blue", cex=1,xlab = "Sin", ylab = "Var")
legend(-1,2,legend = c("Dataset","Predict"),pch = 1,col = c("red","blue"))
```