Exercise 5 - Neural Networks and Support Vector Machines

The goal of this exercise is to explore ANN's (Artificial Neural Networks), and SVM's (Support Vector Machines) with different kernels. This is to better understand the methods, and to get an introduction to black box methods.

- Literature

- Chapter 7 (Black Box Methods - Neural Networks and Support Vector Machines) of "Machine Learning with R" (First Edition) by Brett Lantz, Packt Publishing Ltd., second edition, 2015.
- Chapter 9 (Support Vector Machines) of "An Introduction to Statistical Learning with Applications in R" from G. James, D. Witten, D., T. Hastie, R. Tibshirani. Springer 2013.

You have to decide how you use the datasets, and setup your training and test datasets.

In R, various functions can simplify the implementations such as:

mlp: An R implementation of neural networks. library("RSNNS").

Understand the "size = c(20,20,20)" and experiment with the amount of hidden layers and hidden nodes.
use: learnFunc="Std_Backpropagation" (standard back propagation)
understand and experiment with the learning parameters:
learnFuncParams

You can visualize the training by:
plotIterativeError(networkModel)

ksvm: An R implementation of SVM. library("kernlab")

understand the kernel: "kernel ="rbfdot""
understand the various kernel parameters: "kpar = "automatic""
understand the various learning parameters: "C = 1"

Statistical Machine Learning 2020
– Norbert Krüger – William Kristian Juel
Exercise 5 - Neural Networks and Support Vector Machines

Exercise 5.1: Neural Networks:

5.1.1 Format the training classes so it matches a neural net with N inputs and 10 outputs where each of the outputs matches a given class. For inspiration see (*1).

5.1.2 Train a neural network with N inputs and 10 outputs, based on the modified training data.

5.1.3 Determine how to classify data based on the 10 outputs, and evaluate the neural network, on other data. For inspiration see (*2).

5.1.4 Experiment with various parameters; The hidden structure (layers and nodes) and the internal learning parameters (for standard back-propagation “learning rate” and “maximum output difference”). You should also consider the data you feed to the network, a simple options is the N most important PCA components.

Exercise 5.2: SVM:

5.2.1 Train a SVM classifier on the data, and evaluate it. (Understand how the binary SVM classifier can be used for multi class classification – the ‘one-against-one’-approach is fine which is the standard used in R.)

5.2.2 Experiment with various parameters.

The kernel type; the most common is linear, polynomial and radial basis kernel, understand and test them.

Each of these kernels has different internal parameters, understand these and experiment with them.

Experiment with the learning parameters: if using C-svc (the default) there is only one parameter, “C - cost of constraints violation”.

Statistical Machine Learning 2020
– Norbert Krüger – William Kristian Juel
Exercise 5 - Neural Networks and Support Vector Machines

***1) Inspiration for formatting class labels for neural networks (where id is your data frame):**

```
lev <- levels(id$X1) # Number of classes

nnTrainingClass <- matrix(nrow = length(id$X1), ncol = 10, data = 0) # Create a list probabilities, for all labels

for(i in 1:length(id$X1)) { # Set probabilities to one for matching class
        matchList <- match(lev,toString(id$X1[i]))
        matchList[is.na(matchList)] <- 0
        nnTrainingClass[i,] <- matchList
}

trainingClass <- as.data.frame(nnTrainingClass)
```

***2) Inspiration for formatting mlp output into class labels:**

```
#Using the ”predict” function we have recieved "predictions"
responselist <- matrix(nrow = length(predictions[,1]), ncol = 1, data = "Na")

for(i in 1:nrow(predictions)) {
        responselist[i,] <- toString( which(predictions[i,]==max(predictions[i,])) - 1 )
}
responselist <- data.frame(responselist)
responselist[,1] <- as.factor(responselist[,1])

# Calculating the accuracy
agreement_rbf <- responselist[,1] == test[,1]
table(agreement_rbf)
prop.table(table(agreement_rbf))
```