Machine learning II
Master Data Science
Winter Semester 2019/20

BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Prof. Tim Downie

Workshop 11
Fitting simple neural networks in R

There are many packages to fit artificial neural networks in *R*, the most important are `nnet`[1], `neuralnet`[2], which we will be using today, `RSNNS`: Neural Networks in R using the Stuttgart Neural Network Simulator (SNNS)[3],and `mxnet`[4]

`nnet` is well written code, but is quite limited, e.g. it only allows for one hidden layer and does not graphically display the network.

`neuralnet` has more options, but has some annoying aspects such as factor variables needed to be recoded as binary variables (see Exercise 2)

`RSNNS` and `mxnet` are both front end R packages which connect to compiled neural network libraries, designed to be accessed by different platforms. As a result both are more comprehensive than `nnet` and `neuralnet`, but are less robust in terms of installation and version updates, and the code is less intuitive.

This week at least we keep to the simpler `neuralnet` package. After Christmas you will be using the `mxnet` package.

In Moodle you can find the data set `cereals.csv` as well as the source code for Exercise 2 `Diabetes-NN.R`

You will be using the packages `neuralnet` and `plyr` if necessary install these packages.

Exercise 1

(a) Read all of part (a) before loading and reading through the web page (long address but click here) on the Analytics Vidhya website and running the code.

Comments:

The scaling of the data is coded so that the minimum value for each variable is 0 and the maximum

[1] *Venables W.N. & Ripley B.D.*, Modern Applied Statistics with S, Springer. 2002 (MASS Library)
[2] *Fritsch S., Günther F., Suling M., & Müller S.M.* (2016)
[3] *Bergmei C. and Benítez, J.M.* (2012)
[4] *Tianqi Chen, Qiang Kou, Tong H.*(2017)

value is 1. There is however a minor problem with the data used for the scaling, can you see what it is?

Note that the section *Cross Validation of a Neural Network* has commands to install and load the package `boot` this is not required for this code to run!

Furthermore the code for the `plyr` package is missing. This allows a progress-bar to be set up, so that we can see how a lengthy loop is progressing. Before the for loop add the text

```
library(plyr)
pbar <- create_progress_bar('text')
pbar$init(j)
```

Just before the end of the for loop, add

```
pbar$step()
```

(b) Once you have got to the end of the site, return and look at the code in the section *Cross Validation of a Neural Network*. What is wrong with the "k-fold cross validation" code?

(c) Adapt the code so that a leave-one-out cross validation is run. The data set is small so this should not be a problem. Use your LOOCV code to find the cross-validation RMSE (root mean square error) for 1 to 10 nodes (neurons) in the hidden layer. This will take a couple of minutes to run. What seems to be the optimal number?

(d) Multiple hidden layers can be fitted. For example 3 hidden layers with 3, 2 and 3 nodes would be specified using `hidden=c(3,2,3)`. Copy the LOOCV code and use `hidden = rep(??,j)` with j between 1 and 10 and a sensibe value for `??` to investigate the effect of the number of hidden layers on the data.

Exercise 2 Classifier

In Machine Learning 1 Week 1 we looked at the NHANES dataset for imputing missing data. Today we will use the same data set but use the quick and dirty method of dealing with missing data, namely to remove all observations which contain one or more missing value.

```
> library(NHANES)
> Diabetes<-data.frame(NHANES[,c("Diabetes","Gender","Race1","BMI","Age","Pulse","
    BPSysAve","BPDiaAve","HealthGen","DaysPhysHlthBad","DaysMentHlthBad","
    LittleInterest","Depressed")])
> Diabetes<-na.omit(Diabetes)
> names(Diabetes)
> dim(Diabetes)
```

The number of rows with complete data is $n = 6492$ observations. The synopsis of these data from that workshop are given below.

A source code file `Diabetes-NN.R` has been provided. A practical problem is getting the data into the right form for `neuralnet()`. The NHANES data include many factor variables, and `neuralnet()`

cannot process factor variables directly (unlike `nnet()`). Instead one level of each factor variable is taken as the baseline, for all other levels a binary indicator is created. E.g. for the variable `Race1` a binary variable is created for the levels `Hispanic`, `Mexican`, `White` and `Other`. The quickest way to do this is via the function `model.matrix`. The steps to do this are as follows:

- Use `model.matrix()` to convert all factor variables into binary variables for each level.
- The output from `model.matrix()` includes an "intercept" column consisting entirely of ones. We don't need this for our neural network so we drop it.
- Scale each variable as in Exercise 1, but using only the rows from the training data.
- Define the training and validation data sets.

(a) Work through the commands in the source code.

(b) The code to plot the ROC curve and the AUC for the validation data is included.

(c) Once you have fitted the NN and got the results, investigate using a different number of nodes in the hidden layer, and multiple hidden layers.

(d) Read the help page and try investigating other options such as `algorithm` (string name), `learningrate` (numeric, for back propagation only), and `err.fct="ce"` for a cross entropy loss function. You will probably need to increase the stopping `threshold` so that the code runs faster.

I was unable to find any configuration that gave a substantially better AUC for the validation data, than the first model with 3 nodes in one hidden layer. Let me know if you find one that does.

Synopsis of the Diabetes2 dataset

The NHANES data set (American National Health and Nutrition Examination surveys) contains many variables. Thirteen variables were chosen and the cases with missing data removed. giving $n = 6492$ observations.

`Gender` Gender (sex) of study participant coded as male or female

`Age` Age in years at screening of study participant. Note: Subjects 80 years or older were recorded as 80.

`Race1` Reported race of study participant: Mexican, Hispanic, White, Black, or Other.

`BMI` Body mass index (weight/height2 in kg/m2). Reported for participants aged 2 years or older.

`Pulse` 60 second pulse rate

`BPSysAve` Average systolic blood pressure reading.

`BPDiaAve` Average diastolic blood pressure reading.

`Diabetes` Recoded to (`Yes/No`) Study participant told by a doctor or health professional that they have diabetes. Reported for participants aged 1 year or older as Yes or No.

`HealthGen` Self-reported rating of participant's health in general Reported for participants aged 12 years or older. One of Excellent, Vgood, Good, Fair, or Poor.

`DaysPhysHlthBad` Self-reported number of days participant's physical health was not good out of the past 30 days. Reported for participants aged 12 years or older.

`DaysMentHlthBad` Self-reported number of days participant's mental health was not good out of the past 30 days. Reported for participants aged 12 years or older.

`LittleInterest` Self-reported number of days where participant had little interest in doing things. Reported for participants aged 18 years or older. One of None, Several, Majority (more than half the days), or AlmostAll.

`Depressed` Self-reported number of days where participant felt down, depressed or hopeless. Reported for participants aged 18 years or older. One of None, Several, Majority (more than half the days), or AlmostAll.