Session 1: Utilizing the maximum computation power of your computer on R

Sanjay Rathee

04 October 2019

# Note

Please open your R-Studio as an administrator.

# Overview

In this tutorial, we introduce a package to use multiple cores of your system.

After this tutorial you will be able to:

* Train a neural network model without using multicores
* Train a neural network model using multicores
* Compare results

# Using R Markdown

The tutorial that follows was created using R Markdown. As an exercise in using R Markdown, here, we ask you to save your work in an R Markdown file. By the end of this session, you will have recreated the .Rmd file of this tutorial. You don’t have to retype the instruction and comments. Focus only on the R code and add notes that are useful for your own understanding.

To create an R Markdown file:

* Choose **File** -> **New File** -> **R Markdown…** from RStudio menu.
* Save this file with the name **Rcode.Rmd** in your working directory. Don’t change the other R Markdown options.
* From R studio menu use **Code** -> **Insert Chunk** to add boxes to save R code.

You may use the R Markdown cheat sheet <https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf> to further format the output of the R Markdown file.

# R Markdown setup

This is generated by default when we create an R Markdown file.

# Change this to match the directory you have stored your scripts and data  
root\_directory <- "/media/sanju/B47A9D517A9D10EA/sanju-personal/rcode-clinic/rcode"  
  
knitr::opts\_chunk$set(echo = TRUE)  
knitr::opts\_knit$set(root.dir = root\_directory)

# Required packages

Uncomment and install a package if you didn’t already.

#install.packages("caret")  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

## For Linux and Mac users  
#install.packages("doMC")  
require(doMC)

## Loading required package: doMC

## Loading required package: foreach

## Loading required package: iterators

## Loading required package: parallel

## For Windows  
#install.packages("doParallel")  
#require(doParallel)  
#install.packages("e1071")  
#library(e1097)

# Downloading Data

The data we are going to use is stored as a csv file (having gene expressions for patients) and clinical data (having response) from a breast cancer study. The data are available in folder “session-1” of this link (<https://github.com/sanjaysinghrathi/Rcode-clinic/blob/master/Data.csv>).

The csv data file includes the gene expression data for 19 samples (patients) as well as clinical parameters like response referring to the samples.

print("Welcome to Session 1")

## [1] "Welcome to Session 1"

# Load data

Put the data.csv file in current working directory.

## Load data file  
data <- read.csv("Data.csv", header = TRUE, row.names = 1)

# Train the neural network model

Put the timer on to check the time used by model for training.

start\_time <- Sys.time()  
numFolds <- trainControl(method = 'LOOCV', allowParallel = TRUE, verbose=TRUE , search = "grid")  
grid <- expand.grid(size=c(seq(from = 1, to = 10, by = 1)),  
 decay=c(seq(from = 0.0, to = 0.9, by = 0.1)))  
set.seed(567)  
garbage <- capture.output(suppressWarnings(model <- train(response ~ ., data, method='nnet', trace = FALSE, preProcess = c('center', 'scale'), metric="Accuracy", trControl = numFolds, linout=FALSE, tuneGrid=grid)))  
max(model$results$Accuracy, na.rm = TRUE)

## [1] 1

training\_time <- Sys.time()  
print("Training time...")

## [1] "Training time..."

training\_time-start\_time

## Time difference of 1.375069 mins

**Question**: How much time is used by the model to train?

# Set the number of cores for DoMC

Find the number of cores available on your computer.

## For Linux and Mac users  
registerDoMC(cores=6)  
  
## For window users  
#registerdoParallel(cores=6)

# Train the neural network model with maximum possible cores on computer

Put the timer on to check the time used by model for training.

start\_time <- Sys.time()  
numFolds <- trainControl(method = 'LOOCV', allowParallel = TRUE, verbose=TRUE , search = "grid")  
grid <- expand.grid(size=c(seq(from = 1, to = 10, by = 1)),  
 decay=c(seq(from = 0.0, to = 0.9, by = 0.1)))  
set.seed(567)  
garbage <- capture.output(suppressWarnings(model <- train(response ~ ., data, method='nnet', trace = FALSE, preProcess = c('center', 'scale'), metric="Accuracy", trControl = numFolds, linout=FALSE, tuneGrid=grid)))  
max(model$results$Accuracy, na.rm = TRUE)

## [1] 1

training\_time <- Sys.time()  
print("Training time...")

## [1] "Training time..."

training\_time-start\_time

## Time difference of 29.91197 secs

**Question:** How much time is taken by the model to train with 6 cores?

# Cleaning

Remove unwanted data.

rm(numFolds)  
rm(grid)  
rm(model)

**Question:** Repeat the same to find out time required to train model with different number of cores.