R Notebook

## Exercício de Redes Neurais

**O propósito do exercício é analisar uma série temporal de dados e fazer a partir desse análisse, uma previsão de comportamento da variável à futuro.**

A serie temporal é do IBGE e tem dados sobre o volume de vendas do comercio varejista ampliado entre Janeiro de 2003 e Dezembro de 2011. O formato é arquivo texto, csv.

Serie em: <http://seriesestatisticas.ibge.gov.br/series.aspx?no=2&op=0&vcodigo=MC67&t=volume-vendas-comercio-varejista-ampliado-tipos> , extraída em 02/12/2016

### Carregar pacotes necessários ao análise

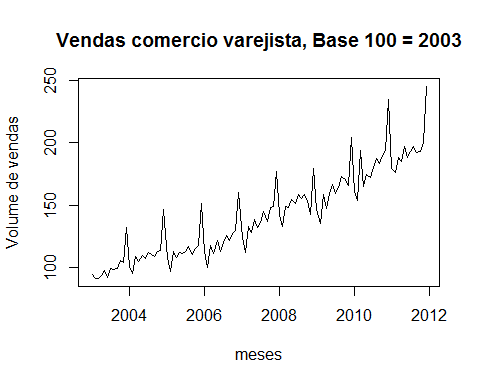
## Warning: package 'neuralnet' was built under R version 3.4.3

carregando arquivo de dados

vendas <- read.csv("IBGE\_VVCVA.csv",   
 header=FALSE,   
 stringsAsFactors = F)  
  
#transposição de linha a coluna em dataframe  
vendas2 <- as.data.frame(t(vendas))  
  
# transformação em st\_Vendas temporal  
st\_Vendas <- ts(vendas2,   
 start = c(2003,1),   
 frequency = 12)

Preparando a ST para análise

plot (st\_Vendas,  
 xlab = "meses",  
 ylab = "Volume de vendas",  
 main = "Vendas comercio varejista, Base 100 = 2003")

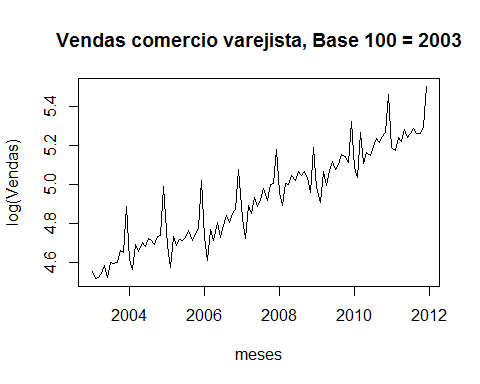


# ——————————————————————————

Se observa que a série tem uma **tendencia crescente** ao longo do tempo, com **variação sazonal forte nos meses de dezembro e fevereiro** de cada ano

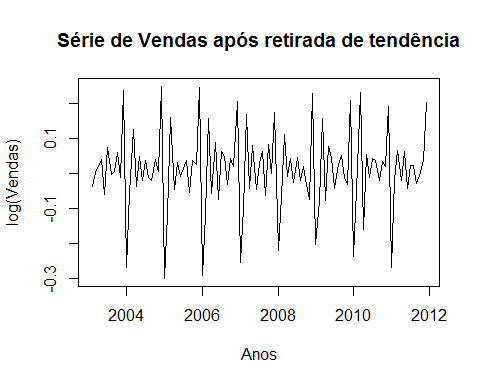
Ao aplicar a função log na série, vamos a estacionar a variância primeiramente.

# número de observações  
n <- length(st\_Vendas)  
  
#   
logst\_Vendas <- log(st\_Vendas)  
  
plot (logst\_Vendas,  
 xlab = "meses",  
 ylab = "log(Vendas)",  
 main = "Vendas comercio varejista, Base 100 = 2003")



A tendencia da série, será retirada fazendo uma diferença entre observações com lag=1 e diferenças de primeira ordem

delta <- diff(logst\_Vendas,lag = 1, differences = 1)  
  
plot (delta,  
 xlab = "Anos",  
 ylab = "log(Vendas)",  
 main = " Série de Vendas após retirada de tendência ")



*Normalização da série* para deixar ela entre 0 e 1

maximo <- max(delta)  
minimo <- min(delta)  
normalizado <- (delta-minimo)/(maximo-minimo)

# ——————————————————————————

PREPARA CONJUNTO DE PADROES ENTRADA/SAIDA

Os lags definidos são o valor prévio (1), o segundo valor anterior (2), o quarto valor anterior (4) e o 12 valor anterior (12), caraterizando um modelo que leva em conta os dois meses anteriores, o valor do trimestre e do ano. O arquivo de saída vair ter 4 variáveis explicativas (X1 a X4) e 1 de resposta (Y), exemplo:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1 | X2 | X3 | X4 | Y |
| Jan04 | Dec03 | Out03 | Fev03 | Fev04 |
| Fev04 | Jan04 | Nov03 | Mar03 | Mar04 |
| … | … | … | … | … |
| … | … | … | … | … |

lags <- c(1,2,4,12) # INFORME OS LAGS

lags <- c(1,2,4,12) # INFORME OS LAGS  
inic <- max(lags)+1 # inicar em 13  
  
nobs <- length(normalizado) # número de observações: 107  
# iniciar vetores nulos  
inputs <- c() # PADROES DE ENTRADA  
output <- c() # PADROES DE SAIDA  
indice <- c() # posição de cada output no tempo  
  
# iteração que preenche os vetores com as observações  
for (i in inic:nobs) {  
 inputs = rbind(inputs,normalizado[i-lags])  
 output = c(output,normalizado[i])  
 indice = c(indice,i+1)  
  
data.train <- cbind(inputs,output) # PADRÕES ENTRADA/SAÍDA  
  
nomes <- c()  
for(i in 1:length(lags)) {  
 nomes[i] = paste("X",i,sep="")  
 }  
  
colnames(data.train)=c(nomes,"Y")  
  
limites <- c(0,1)   
p <- dim(inputs)[2]+1 # número de variáveis no modelo  
  
#original tem um p demais  
#range.data <- matrix(rep(limites,p),2,p) # intervalo de cada variável  
range.data <- matrix(rep(limites,p),2)  
}

SELECIONA AMOSTRAS TREINO E TESTE

npadroes <- dim(data.train)[1] # calcula o número de padrões para o análise: 95  
  
noutsample <- 12 # escolhe o tamanho da previsão...12  
  
ninsample <- npadroes-noutsample # número de padrões de treino

# ——————————————————————————

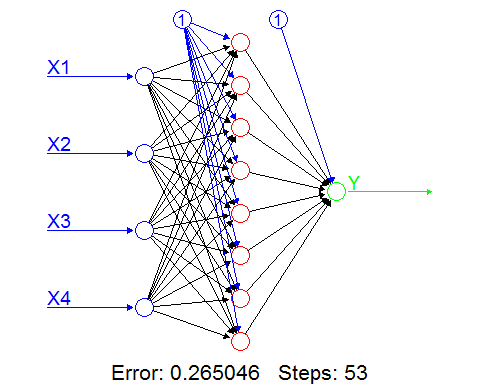
Processamento da ST com rede neural, usando o pacote neuralnet versão 1.33. O pacote treina redes neurais com backpropagation

TREINA REDE FEEDFORWARD  
Serão usadas 8 camadas escondidas, valor que foi o melhor encontrado depois de testar de 3 a 10 camadas. Os padrões a serem usados são os primeiros 84 (1:ninsample) valores observados.

modelo\_nn <- neuralnet(formula = Y ~ X1 + X2 + X3 + X4,  
 data = data.train[1:ninsample,],   
 hidden = 8,  
 linear.output = F)

## Gráfico da rede neural

plot(modelo\_nn,  
 rep ="best",  
 col.hidden = "red",  
 col.entry = "blue",  
 col.entry.synapse = "blue",  
 col.out = "green",  
 col.out.synapse = "green",  
 show.weights = F,  
 fontsize = 14)

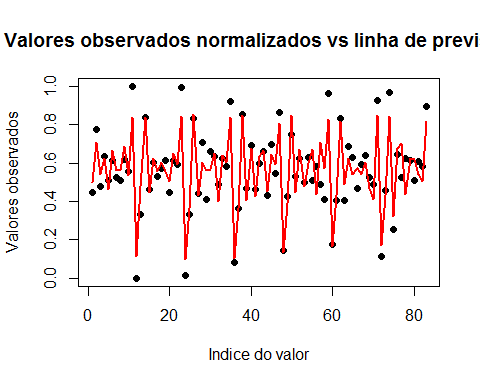


PREVISÃO INSAMPLE

Feito o modelo, vamos fazer uma previsão com o modelo calculado, sobre os próprios dados de treinamento (previsão insample de 84 dados).

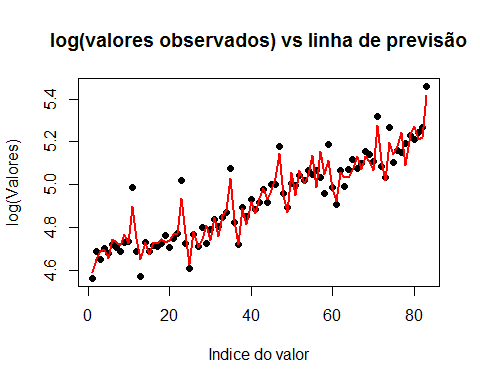
Essa previsão vai servir para testar a precisão do modelo calculado

previsao <- compute (modelo\_nn,   
 inputs[1:ninsample,])$net.result   
  
plot (output[1:ninsample],  
 xlab = "Indice do valor",  
 ylab = "Valores observados",  
 main = "Valores observados normalizados vs linha de previsão",  
 pch = 19)  
  
lines (1:ninsample,   
 previsao,  
 lwd = 2,  
 col = "red")



Vamos agora fazer a previsão da série sobre o log de valores reais previsão do log passageiros 1 passo a frente

previsaolog <- c()  
for (i in 1:ninsample) {  
 x0 = logst\_Vendas[indice[i]-1]  
 cumprevisao = x0 + previsao[i] \* (maximo-minimo) + minimo  
 previsaolog = c(previsaolog,cumprevisao)  
}  
  
plot(logst\_Vendas[indice[1]:indice[ninsample]],  
 xlab = "Indice do valor",  
 ylab = "log(Valores)",  
 main = "log(valores observados) vs linha de previsão",  
 pch = 19)  
  
lines(1:ninsample,   
 previsaolog,  
 lwd = 2,  
 col = "red")

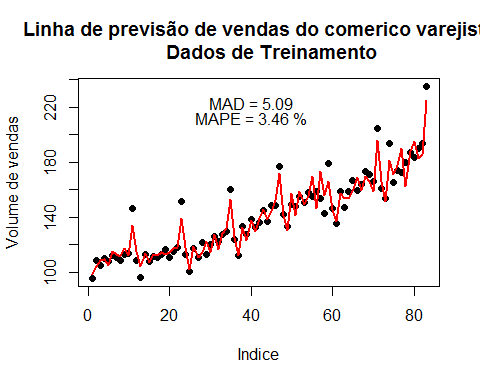


Previsão de vendas do comercio varejista 1 passo a frente

previsao\_vendas\_treino = exp(previsaolog)  
  
MAD\_INSAMPLE = round(mean(abs(st\_Vendas[indice[1]:indice[ninsample]] - previsao\_vendas\_treino)),2)  
  
MAPE\_INSAMPLE = round(100\*mean(abs(st\_Vendas[indice[1]:indice[ninsample]] -   
 previsao\_vendas\_treino)/st\_Vendas[indice[1]:indice[ninsample]]),2)

Plotagem da série com indicadores de precisão

x\_plot <- mean(indice[1]:indice[ninsample])-15  
  
y\_plot <- max(st\_Vendas[indice[1]:indice[ninsample]])\*.95  
  
y\_plot2 <- y\_plot\*.95  
  
plot (st\_Vendas[indice[1]:indice[ninsample]],  
 xlab = "Indice",  
 ylab = "Volume de vendas",  
 main = "Linha de previsão de vendas do comerico varejista  
 Dados de Treinamento",  
 pch = 19)  
  
lines (1:ninsample,   
 previsao\_vendas\_treino,  
 lwd = 2,  
 col = "red")  
  
text(x\_plot,y\_plot, paste0("MAD = ",MAD\_INSAMPLE))  
  
text(x\_plot,y\_plot2, paste0("MAPE = ",MAPE\_INSAMPLE," %"))

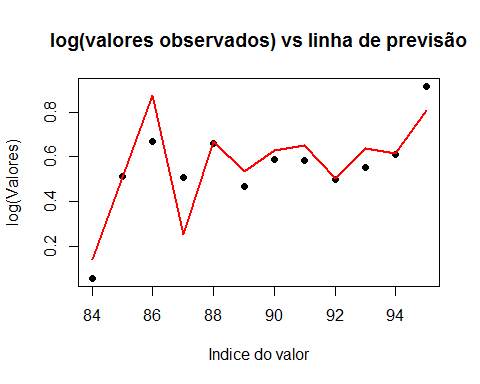


# ——————————————————————————

PREVISÃO OUTSAMPLE

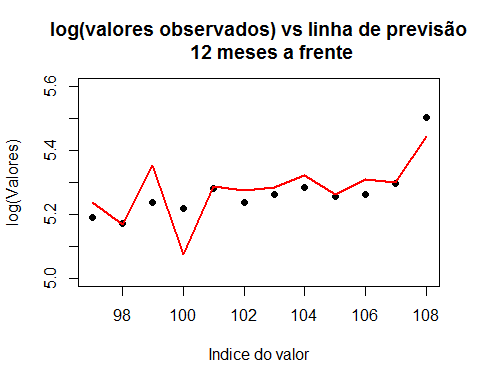
Vamos fazer agora a previsão sobre os valores que não foram usados para treinar o modelo, ou seja os últimos 12 valores do ano 2011 (indices 84 a 95 dos dados de data.train) previsão da diferença do log

previsao\_vendas\_teste <- compute(modelo\_nn,  
 inputs[(ninsample + 1):(ninsample + noutsample),])$net.result   
  
plot((ninsample + 1):(ninsample + noutsample),  
 output[(ninsample + 1):(ninsample + noutsample)],  
 xlab = "Indice do valor",  
 ylab = "log(Valores)",  
 main = "log(valores observados) vs linha de previsão",  
 pch = 19)  
  
lines((ninsample+1):(ninsample+noutsample),  
 previsao\_vendas\_teste,  
 lwd = 2,  
 col = "red")



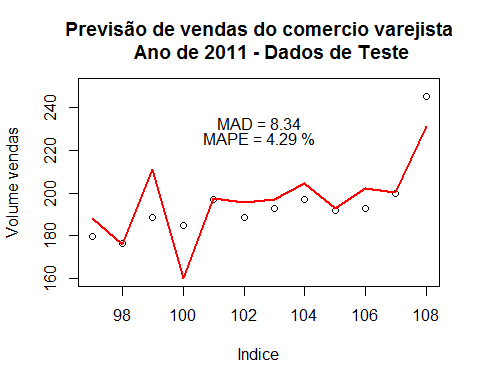
previsão do log 1 passo a frente

previsaolog = c()  
for (i in (ninsample+1):(ninsample + noutsample)){  
 x0 = logst\_Vendas[indice[i] - 1]  
 cumprevisao = x0 + previsao\_vendas\_teste[i-ninsample]\*(maximo-minimo)+minimo  
 previsaolog = c(previsaolog,cumprevisao)}  
   
  
plot(indice[ninsample+1]:indice[ninsample+noutsample],  
 logst\_Vendas[indice[ninsample+1]:indice[ninsample+noutsample]],  
 ylim = c(5,5.6),  
 xlab = "Indice do valor",  
 ylab = "log(Valores)",  
 main = "log(valores observados) vs linha de previsão  
 12 meses a frente",  
 pch = 19)  
  
lines(indice[ninsample+1]:indice[ninsample+noutsample],  
 previsaolog,  
 lwd = 2,  
 col = "red")



previsão de vendas 1 passo a frente

previsao\_vendas\_futuro <- exp(previsaolog)  
  
MAD <- round(mean(abs(st\_Vendas[indice[ninsample + 1]:indice[ninsample + noutsample]] -   
 previsao\_vendas\_futuro)),2)  
MAPE <- round(100\*mean(abs(st\_Vendas[indice[ninsample +1 ]:indice[ninsample + noutsample]] -   
 previsao\_vendas\_futuro) / st\_Vendas[indice[ninsample + 1]:indice[ninsample+noutsample]]),2)  
  
# coordenadas de plotagem dos indicadores de qualidade  
x\_plot <- mean(indice[ninsample+1]:indice[ninsample+noutsample])  
  
y\_plot <- max(st\_Vendas[indice[ninsample+1]:indice[ninsample+noutsample]])\*.95  
  
y\_plot2 <- y\_plot\*.97  
  
  
plot (indice[ninsample+1]:indice[ninsample+noutsample],  
 st\_Vendas[indice[ninsample + 1]:indice[ninsample + noutsample]],  
 xlab = "Indice",  
 ylab = "Volume vendas",  
 main = "Previsão de vendas do comercio varejista  
 Ano de 2011 - Dados de Teste",  
 ylim = c(160,250))  
  
lines(indice[ninsample + 1]:indice[ninsample + noutsample],  
 previsao\_vendas\_futuro,  
 lwd = 2,  
 col = "red")  
  
  
# Plotar Indicadores de precisão  
text(x\_plot, y\_plot,   
 paste0("MAD = ",MAD))  
text(x\_plot,y\_plot2,   
 paste0("MAPE = ",MAPE," %"))



Fazer uma série temporal com a previsão e exportar o arquivo

Vendas\_2011 <- ts (round((previsao\_vendas\_futuro),2),   
 start = c(2011,1),   
 frequency = 12)  
  
# salvar como data frame e exportar  
meses <- c("Jan","Fev","Mar","Abr","Mai","Jun",  
 "Jul", "Ago", "Set","Out","Nov","Dec")  
  
nome\_col <- c( "Vendas Previstas")   
  
prev\_vendas\_2011 <- as.data.frame(Vendas\_2011,   
 row.names = meses)  
  
names(prev\_vendas\_2011) <- nome\_col  
  
write.csv2(prev\_vendas\_2011,   
 file = "previsao\_vendas\_2012.csv" )

# Fim do análise

date()

## [1] "Mon Feb 19 07:51:41 2018"