Отчёт по поводу вакансии дрессировщика нейросетей

[Дмитрий Пасько](https://github.com/PasaOpasen)

01 06 2020

# Задание 1

Диаграмма алгоритма:

Реализация алгоритма:

f = function(a, b, c) {  
 if (a > c & b > c) {  
 cat("no solutions \n")  
 return()  
 }  
   
 exist\_sol = F  
   
 for (apples in 0:(c%/%a)) for (pears in 0:((c - apples \* a)%/%b)) if (apples \*   
 a + pears \* b == c) {  
 cat(apples, " apples, ", pears, " pears\n")  
 exist\_sol = T  
 }  
   
 if (!exist\_sol) {  
 cat("no solutions \n")  
 }  
}

Примеры:

f(1, 2, 3)

## 1 apples, 1 pears  
## 3 apples, 0 pears

f(15, 30, 150)

## 0 apples, 5 pears  
## 2 apples, 4 pears  
## 4 apples, 3 pears  
## 6 apples, 2 pears  
## 8 apples, 1 pears  
## 10 apples, 0 pears

f(10, 20, 50)

## 1 apples, 2 pears  
## 3 apples, 1 pears  
## 5 apples, 0 pears

f(100, 140, 220)

## no solutions

# Задание 2

Загрузим данные и отберём подходящие по времени:

library(readxl)  
library(dplyr)  
  
data = read\_excel("data.xlsx") %>% mutate(Date = as.Date(Date))  
  
data2 = data %>% filter(Date >= as.Date("2016-03-01") & Date < as.Date("2018-11-26"))

Посмотрим первые строки и статистику по данным:

head(data2, 10)

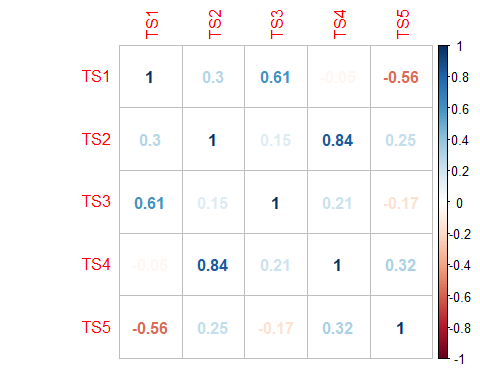
## # A tibble: 10 x 6  
## Date TS1 TS2 TS3 TS4 TS5  
## <date> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2016-03-01 -0.405 -0.478 -1.21 -1.47 1.20  
## 2 2016-03-02 -0.397 -0.682 -1.17 -1.57 1.11  
## 3 2016-03-03 -0.355 -0.774 -1.22 -1.61 1.31  
## 4 2016-03-04 -0.259 -0.506 -1.35 -1.56 1.35  
## 5 2016-03-05 -0.278 -0.788 -1.32 -1.68 1.15  
## 6 2016-03-06 -0.376 -0.598 -1.04 -1.53 1.12  
## 7 2016-03-07 -0.474 -0.788 -1.05 -1.62 1.22  
## 8 2016-03-08 -0.530 -1.01 -0.757 -1.64 1.11  
## 9 2016-03-09 -0.614 -0.932 -0.670 -1.58 1.30  
## 10 2016-03-10 -0.718 -1.20 -0.765 -1.47 1.12

summary(data2)

## Date TS1 TS2 TS3   
## Min. :2016-03-01 Min. :-1.4211 Min. :-2.3861 Min. :-2.44687   
## 1st Qu.:2016-11-05 1st Qu.:-0.7762 1st Qu.:-0.9051 1st Qu.:-0.68272   
## Median :2017-07-13 Median :-0.3382 Median : 0.2077 Median :-0.06417   
## Mean :2017-07-13 Mean : 0.0000 Mean : 0.0000 Mean : 0.00000   
## 3rd Qu.:2018-03-20 3rd Qu.: 0.3980 3rd Qu.: 0.8573 3rd Qu.: 0.72861   
## Max. :2018-11-25 Max. : 2.3416 Max. : 2.1531 Max. : 2.69313   
## TS4 TS5   
## Min. :-2.07362 Min. :-1.6934   
## 1st Qu.:-0.88311 1st Qu.:-0.8591   
## Median : 0.02046 Median :-0.1111   
## Mean : 0.00000 Mean : 0.0000   
## 3rd Qu.: 0.90027 3rd Qu.: 0.6723   
## Max. : 1.94343 Max. : 2.5865

Корреляции между временными рядами:

library(corrplot)  
  
corrplot(cor(data2[, -1]), method = "number")



Проведём регрессионный анализ для каждого временного ряда, взяв в качестве регрессоров остальные ряды. При этом, раз идёт работа с временными рядами, важно проводить тесты на автокорреляцию.

Summary = function(fit) {  
 print(summary(fit))  
 cat("-----> ТЕСТ НА АВТОКОРРЕЛЯЦИЮ:\n")  
 cat("\n")  
 print(car::durbinWatsonTest(fit))  
 cat("\n")  
}  
  
Summary(lm(TS1 ~ TS5 + TS2 + TS3 + TS4 - 1, data2))

##   
## Call:  
## lm(formula = TS1 ~ TS5 + TS2 + TS3 + TS4 - 1, data = data2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.230608 -0.101022 -0.001432 0.099504 0.239774   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## TS5 -0.433382 0.003984 -108.8 <2e-16 \*\*\*  
## TS2 1.153539 0.006657 173.3 <2e-16 \*\*\*  
## TS3 0.573787 0.003859 148.7 <2e-16 \*\*\*  
## TS4 -0.995131 0.006964 -142.9 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1151 on 996 degrees of freedom  
## Multiple R-squared: 0.9868, Adjusted R-squared: 0.9868   
## F-statistic: 1.864e+04 on 4 and 996 DF, p-value: < 2.2e-16  
##   
## -----> ТЕСТ НА АВТОКОРРЕЛЯЦИЮ:  
##   
## lag Autocorrelation D-W Statistic p-value  
## 1 -0.02797872 2.055192 0.42  
## Alternative hypothesis: rho != 0

Summary(lm(TS2 ~ TS1 + TS5 + TS3 + TS4 - 1, data2))

##   
## Call:  
## lm(formula = TS2 ~ TS1 + TS5 + TS3 + TS4 - 1, data = data2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.189383 -0.085380 0.001091 0.083734 0.179242   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## TS1 0.839069 0.004842 173.29 <2e-16 \*\*\*  
## TS5 0.362554 0.004091 88.62 <2e-16 \*\*\*  
## TS3 -0.482661 0.004168 -115.80 <2e-16 \*\*\*  
## TS4 0.862438 0.003421 252.11 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09813 on 996 degrees of freedom  
## Multiple R-squared: 0.9904, Adjusted R-squared: 0.9904   
## F-statistic: 2.571e+04 on 4 and 996 DF, p-value: < 2.2e-16  
##   
## -----> ТЕСТ НА АВТОКОРРЕЛЯЦИЮ:  
##   
## lag Autocorrelation D-W Statistic p-value  
## 1 -0.05983911 2.118307 0.08  
## Alternative hypothesis: rho != 0

Summary(lm(TS3 ~ TS1 + TS2 + TS5 + TS4 - 1, data2))

##   
## Call:  
## lm(formula = TS3 ~ TS1 + TS2 + TS5 + TS4 - 1, data = data2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.44493 -0.16521 -0.00181 0.17123 0.43947   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## TS1 1.66769 0.01121 148.71 <2e-16 \*\*\*  
## TS2 -1.92860 0.01665 -115.80 <2e-16 \*\*\*  
## TS5 0.71124 0.00929 76.56 <2e-16 \*\*\*  
## TS4 1.67644 0.01446 115.96 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1961 on 996 degrees of freedom  
## Multiple R-squared: 0.9617, Adjusted R-squared: 0.9615   
## F-statistic: 6249 on 4 and 996 DF, p-value: < 2.2e-16  
##   
## -----> ТЕСТ НА АВТОКОРРЕЛЯЦИЮ:  
##   
## lag Autocorrelation D-W Statistic p-value  
## 1 0.009567536 1.980152 0.684  
## Alternative hypothesis: rho != 0

Summary(lm(TS4 ~ TS1 + TS2 + TS3 + TS5 - 1, data2))

##   
## Call:  
## lm(formula = TS4 ~ TS1 + TS2 + TS3 + TS5 - 1, data = data2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.219446 -0.097332 0.001204 0.098493 0.243865   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## TS1 -0.958159 0.006705 -142.90 <2e-16 \*\*\*  
## TS2 1.141614 0.004528 252.11 <2e-16 \*\*\*  
## TS3 0.555368 0.004789 115.96 <2e-16 \*\*\*  
## TS5 -0.408324 0.005427 -75.24 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1129 on 996 degrees of freedom  
## Multiple R-squared: 0.9873, Adjusted R-squared: 0.9873   
## F-statistic: 1.937e+04 on 4 and 996 DF, p-value: < 2.2e-16  
##   
## -----> ТЕСТ НА АВТОКОРРЕЛЯЦИЮ:  
##   
## lag Autocorrelation D-W Statistic p-value  
## 1 -0.04315816 2.084443 0.224  
## Alternative hypothesis: rho != 0

Summary(lm(TS5 ~ TS1 + TS2 + TS3 + TS4 - 1, data2))

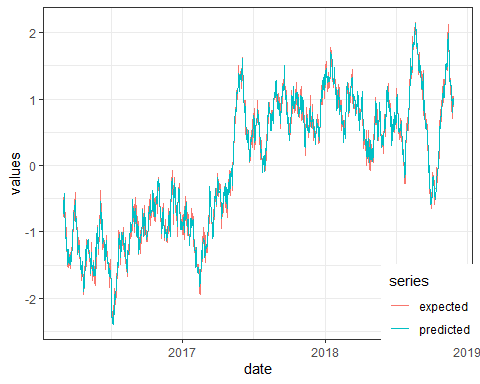
##   
## Call:  
## lm(formula = TS5 ~ TS1 + TS2 + TS3 + TS4 - 1, data = data2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.52367 -0.21172 -0.00317 0.20786 0.58423   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## TS1 -2.12832 0.01956 -108.79 <2e-16 \*\*\*  
## TS2 2.44779 0.02762 88.62 <2e-16 \*\*\*  
## TS3 1.20176 0.01570 76.56 <2e-16 \*\*\*  
## TS4 -2.08264 0.02768 -75.24 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.255 on 996 degrees of freedom  
## Multiple R-squared: 0.9353, Adjusted R-squared: 0.935   
## F-statistic: 3597 on 4 and 996 DF, p-value: < 2.2e-16  
##   
## -----> ТЕСТ НА АВТОКОРРЕЛЯЦИЮ:  
##   
## lag Autocorrelation D-W Statistic p-value  
## 1 0.0553368 1.889172 0.068  
## Alternative hypothesis: rho != 0

Во всех представленных моделях превышает 90% нет оснований полагать наличие автокорреляции в ошибках. Реграссантом является второй временной ряд, поскольку его модель имеет

1. наибольший
2. наименьшую стандартную ошибку
3. наибольшее значение F-статистики, то есть наибольшую значимость

Сравнение второго временного ряда и его оценки:

library(ggplot2)  
  
fit = lm(TS2 ~ TS1 + TS5 + TS3 + TS4 - 1, data2)  
  
ggplot(tibble(date = rep(data2$Date, 2), values = c(data2$TS2, predict(fit, data2)),   
 type = sort(rep(c("expected", "predicted"), nrow(data2)))), aes(x = date, y = values,   
 col = factor(type))) + geom\_line() + theme\_bw() + theme(legend.position = c(0.9,   
 0.1)) + labs(col = "series")



# Задание 3

Запишем производную функции бинарной кроссэнтропии по переменной :

Если учесть, что:

Придём к выражению:

Запрограммируем это выражение и сделаем спуск из начальных параметров в течение 10 итераций:

library(readxl)  
  
data = read\_excel("data.xlsx", sheet = 2)  
  
print(data)

## # A tibble: 16 x 3  
## X1 X2 Label  
## <dbl> <dbl> <dbl>  
## 1 0.8 -1.1 0  
## 2 1.5 0.5 1  
## 3 -2 -1 0  
## 4 1 -3 1  
## 5 3 1 1  
## 6 -3 -0.2 0  
## 7 -3 -2 0  
## 8 -3.5 0.5 0  
## 9 0.5 -2.5 0  
## 10 -1 1 1  
## 11 1 0 1  
## 12 -3 2.1 1  
## 13 3 -0.5 1  
## 14 1.5 -1.5 1  
## 15 -2 -3 0  
## 16 -1.7 0.3 0

loss = function(pred, expect) {  
 tmp = -expect \* log(pred) - (1 - expect) \* log(1 - pred)  
 return(mean(tmp))  
}  
  
  
w1 = 0  
w2 = 0  
for (iter in 1:10) {  
 z = w1 \* data$X1 + w2 \* data$X2  
 s = 1/(1 + exp(-z))  
 cat("iter = ", iter, " loss = ", loss(s, data$Label), "\n")  
   
 w1 = w1 + mean(data$X1 \* s \* (data$Label \* exp(-z) + data$Label - 1))  
 w2 = w2 + mean(data$X2 \* s \* (data$Label \* exp(-z) + data$Label - 1))  
}

## iter = 1 loss = 0.6931472   
## iter = 2 loss = 0.4352326   
## iter = 3 loss = 0.4138624   
## iter = 4 loss = 0.4046549   
## iter = 5 loss = 0.399987   
## iter = 6 loss = 0.3973808   
## iter = 7 loss = 0.3958349   
## iter = 8 loss = 0.39488   
## iter = 9 loss = 0.3942728   
## iter = 10 loss = 0.3938784

z = w1 \* data$X1 + w2 \* data$X2  
s = 1/(1 + exp(-z))  
cat("end loss = ", loss(s, data$Label), "\n")

## end loss = 0.393618

cat("w1 = ", w1, " w2 = ", w2, "\n")

## w1 = 0.8808708 w2 = 0.7460919