

## Exercício 4 - Estudo Dirigido

---

A.P. Braga

April 26, 2017

### TREINAMENTO PERCEPTRON SIMPLES

O estudo dirigido consistirá na leitura e realização de atividades sobre o treinamento do perceptron simples.

#### PARTE 1

Ler atentamente as notas de aula da disciplina sobre Perceptron Simples, disponíveis no Moodle. Implementar as funções "yperceptron" e "trainperceptron" descritas nas notas.

#### PARTE 2

No primeiro exercício da disciplina, foi pedido para o aluno amostrar duas distribuições normais no espaço  $R^2$ , ou seja, duas distribuições com duas variáveis cada (Ex:  $x_1$  e  $x_2$ ). As distribuições são caracterizadas como  $\mathcal{N}(2, 2, \sigma^2)$  e  $\mathcal{N}(4, 4, \sigma^2)$ , como pode ser visualizado na Fig. 1.

Naquela ocasião os alunos utilizaram um vetor de pesos do perceptron igual aos coeficientes do polinômio  $x_2 = -x_1 + 6$ , ou seja,  $w_1 = 1$ ,  $w_2 = 1$  e  $\theta = -6$  ( $w_1 \cdot x_1 + w_2 \cdot x_2 + \theta = 0$ ). Todavia, nesta atividade o aluno irá fazer o treinamento do perceptron através da função "trainperceptron" implementada na etapa anterior afim de encontrar o vetor de pesos  $w$  e, utilizando a função "yperceptron" com este vetor de pesos, encontrar a superfície de separação como mostra a Figura 4.

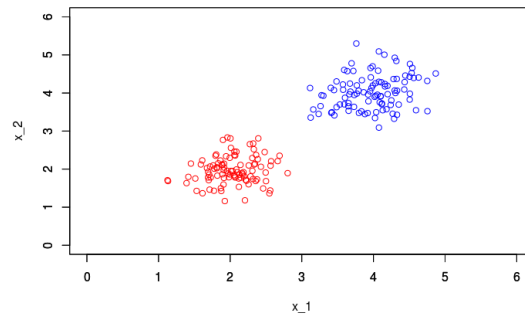


Figure 1: Dados amostrados de duas distribuições Normais com médias  $m1 = (2; 2)^T$  e  $m2 = (4; 4)^T$  e coeficiente de correlação nulo

## FORMA DE ENTREGA

Relatório em .doc ou .pdf, descrevendo o que foi feito, mostrando os gráficos e as informações pedidas e explicando os resultados obtidos, assim como as partes importantes do código. O relatório deve ser colocado em um arquivo .zip junto com os códigos utilizados e enviado via Moodle.

```

rm(list=ls())
library('plot3D')
source('yperceptron.R')
s1<-0.4
s2<-0.4
nc<-200
xc1<-matrix(rnorm(nc*2),ncol=2)*s1 + t(matrix((c(2,2)),ncol=nc,nrow=2))
xc2<-matrix(rnorm(nc*2),ncol=2)*s2 + t(matrix((c(4,4)),ncol=nc,nrow=2))
plot(xc1[,1],xc1[,2],col = 'red', xlim = c(0,6),ylim = c(0,6),xlab = 'x_1',ylab='x_2')
par(new=T)
plot(xc2[,1],xc2[,2],col = 'blue', xlim = c(0,6),ylim = c(0,6),xlab = '',ylab='')

x1_reta<-seq(6/100,6,6/100)
x2_reta<- -x1_reta+6
par(new=T)
plot(x1_reta, x2_reta , type = 'l',col = 'orange', xlim = c(0,6),ylim = c(0,6),xlab = '',ylab='')

w<-c(-6,1,1)
seqi<-seq(0,6,0.1)
seqj<-seq(0,6,0.1)
M <- matrix(0,nrow=length(seqi),ncol=length(seqj))

ci<-0
for (i in seqi){
  ci<-ci+1
  cj<-0
  for(j in seqj)
  {
    cj<-cj+1
    x<-c(i,j)
    M[ci,cj]<- yperceptron(x,w,1)
  }
}

plot(xc1[,1],xc1[,2],col = 'red', xlim = c(0,6),ylim = c(0,6),xlab = 'x_1',ylab='x_2')
par(new=T)
plot(xc2[,1],xc2[,2],col = 'blue', xlim = c(0,6),ylim = c(0,6),xlab = '',ylab='')
par(new=T)
contour(seqi,seqj,M, xlim = c(0,6),ylim = c(0,6),xlab = '',ylab='')

persp3D(seqi,seqj,M,counter=T,theta = 55, phi = 30, r = 40, d = 0.1, expand = 0.5,
  ltheta = 90, lphi = 180, shade = 0.4, ticktype = "detailed", nticks=5)

```

Figure 2: Resolução Homework 1

```
rm(list=ls())
library('plot3D')
source('trainperceptron.R')
source('yperceptron.R')
s1<-0.4
s2<-0.4
nc<-200
xc1<-matrix(rnorm(nc*2),ncol=2)*s1 + t(matrix((c(2,2)),ncol=nc,nrow=2))
xc2<-matrix(rnorm(nc*2),ncol=2)*s2 + t(matrix((c(4,4)),ncol=nc,nrow=2))
plot(xc1[,1],xc1[,2],col = 'red', xlim = c(0,6),ylim = c(0,6),xlab = 'x_1',ylab='x_2')
par(new=T)
plot(xc2[,1],xc2[,2],col = 'blue', xlim = c(0,6),ylim = c(0,6),xlab = '',ylab='')

x1_reta<-seq(6/100,6,6/100)
x2_reta<- -x1_reta+6
par(new=T)
plot(x1_reta, x2_reta , type = 'l',col = 'orange', xlim = c(0,6),ylim = c(0,6),xlab = '',ylab='')
```

Aqui você deve chamar a função trainperceptron que irá retornar o vetor de pesos  $w$

```
seqi<-seq(0,6,0.1)
seqj<-seq(0,6,0.1)
M <- matrix(0,nrow=length(seqi),ncol=length(seqj))

ci<-0
for (i in seqi){
  ci<-ci+1
  cj<-0
  for(j in seqj)
  {
    cj<-cj+1
    x<-c(i,j)
    M[ci,cj]<- yperceptron(x,w,1)
  }
}

plot(xc1[,1],xc1[,2],col = 'red', xlim = c(0,6),ylim = c(0,6),xlab = 'x_1',ylab='x_2')
par(new=T)
plot(xc2[,1],xc2[,2],col = 'blue', xlim = c(0,6),ylim = c(0,6),xlab = '',ylab='')
par(new=T)
contour(seqi,seqj,M, xlim = c(0,6),ylim = c(0,6),xlab = '',ylab='')

persp3D(seqi,seqj,M,counter=T,theta = 55, phi = 30, r = 40, d = 0.1, expand = 0.5,
  ltheta = 90, lphi = 180, shade = 0.4, ticktype = "detailed", nticks=5)
```

Figure 3: Ideia para esse exercício

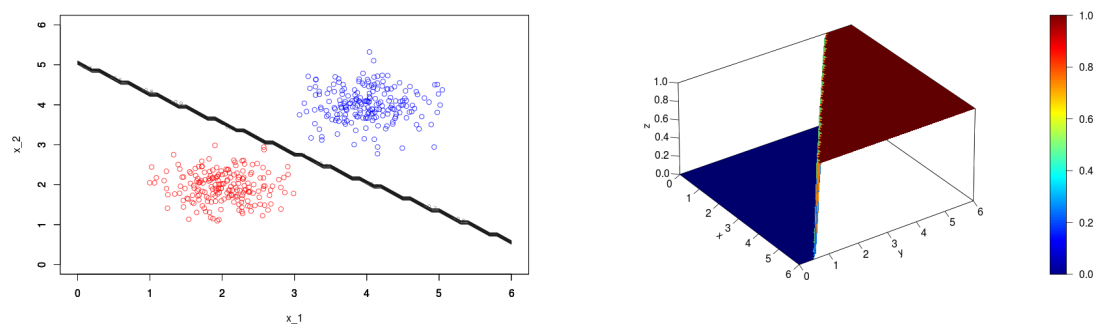


Figure 4: Resolução Ex1