Universidade de Brasília

DEPARTAMENTO DE ESTATÍSTICA

7 de outubro de 2025

Lista 2: Ajustando uma RNA "no braço".

Prof. Guilherme Rodrigues

Redes Neurais Profundas

Tópicos em Estatística 2

(A) **As questões deverão ser respondidas em um único relatório *PDF* ou *html*, produzido usando as funcionalidades do *Rmarkdown* ou outra ferramenta equivalente.**

(B) **O aluno poderá consultar materiais relevantes disponíveis na internet, tais como livros, *blogs* e artigos.**

(C) **O trabalho é individual. Suspeitas de plágio e compartilhamento de soluções serão tratadas com rigor.**

(D) **Os códigos *R* utilizados devem ser disponibilizados na integra, seja no corpo do texto ou como anexo.**

(E) **O aluno deverá enviar o trabalho até a data especificada na plataforma *Microsoft Teams*.**

(F) **O trabalho será avaliado considerando o nível de qualidade do relatório, o que inclui a precisão das respostas, a pertinência das soluções encontradas, a formatação adotada, dentre outros aspectos correlatos.**

(G) **Escreva seu código com esmero, evitando operações redundantes, comentando os resultados e usando as melhores práticas em programação.**

(H) **O uso de Modelos de Linguagem de Grande Escala (LLMs), como ChatGPT, Gemini ou equivalentes, é permitido exclusivamente como ferramenta de apoio para organização de ideias, revisão textual, esclarecimento de conceitos e sugestões de escrita. O uso indiscriminado pode ser caracterizado como plágio acadêmico.**

Considere um processo gerador de dados da forma

$$\begin{aligned} Y &\sim N(\mu, \sigma^2 = 1) \\ \mu &= |X_1^3 - 30\text{sen}(X_2) + 10| \\ X_j &\sim \text{Uniforme}(-3, 3), \quad j = 1, 2. \end{aligned}$$

Neste modelo (que iremos considerar como o "**modelo real**"), a esperança condicional de Y é dada por $E(Y|X_1, X_2) = |X_1^3 - 30\text{sen}(X_2) + 10|$. A superfície tridimensional $(E(Y|X_1, X_2), X_1, X_2)$ está representada em duas dimensões cartesianas na Figura 1.

O código a seguir simula $m = 100.000$ observações desse processo.

```
### Gerando dados "observados"
set.seed(22025)
m.obs <- 100000
dados <- tibble(x1.obs=runif(m.obs, -3, 3),
                x2.obs=runif(m.obs, -3, 3)) %>%
  mutate(mu=abs(x1.obs^3 - 30*sin(x2.obs) + 10),
         y=rnorm(m.obs, mean=mu, sd=1))
```

Nesta lista estamos interessados em estimar o modelo acima usando uma rede neural simples, ajustada sobre os dados simulados. Precisamente, queremos construir uma rede neural com apenas uma camada escondida contendo dois neurônios.

Matematicamente, a rede é descrita pelas seguintes equações:

$$\begin{aligned} f_{0,1} &= x_1 w1 + x_2 w2 + b_1 \\ f_{0,2} &= x_1 w3 + x_2 w4 + b_2 \\ h_{1,1} &= a(f_{0,1}) \\ h_{1,2} &= a(f_{0,2}) \\ f_{1,1} &= h_{1,1} w_5 + h_{1,2} w_6 + b_3 \\ \hat{y} &= h_{2,1} = f_{1,1}, \end{aligned}$$

onde $a(x) = \frac{1}{1+e^{-x}}$ representa a função de ativação logística (sigmoide).

Adotaremos como função de perda o erro quadrático médio, expresso por

$$J(\boldsymbol{\phi}) = \frac{1}{m} \sum_{i=1}^{m} L(f(x_{1i}, x_{2i}; \boldsymbol{\phi}), y_i) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2,$$

onde x_{ji} representa a j-ésima covariável (*feature*) da i-ésima observação, $\boldsymbol{\phi} = (w_1, \ldots, w_6, b_1, b_2, b_3)$ é o vetor de pesos e viéses (parâmetros) e, pela definição da rede,

$$f(x_{1i}, x_{2i}; \boldsymbol{\phi}) = \hat{y}_i = a(x_{1i} w_1 + x_{2i} w_3 + b_1) w_5 + a(x_{1i} w_2 + x_{2i} w_4 + b_2) w_6 + b_3.$$

Uma representação gráfica da rede está apresentada na Figura 2.

Em notação matricial, a rede neural pode ser descrita por

$$\begin{aligned} \boldsymbol{f}_0 &= \boldsymbol{\Omega}_0 \boldsymbol{x} + \boldsymbol{\beta}_0 \\ \boldsymbol{h}_1 &= \boldsymbol{a}(\boldsymbol{f}_0) \\ f_1 &= \boldsymbol{\Omega}_1 \boldsymbol{h}_1 + \boldsymbol{\beta}_1 \\ \hat{y} &= h_2 = f_1, \end{aligned}$$

onde

$$\boldsymbol{x} = \boldsymbol{h}_0 = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \ \boldsymbol{\Omega}_0 = \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix}, \ \boldsymbol{\beta}_0 = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \ \boldsymbol{f}_0 = \begin{pmatrix} f_{0,1} \\ f_{0,2} \end{pmatrix}, \ \boldsymbol{h}_1 = \begin{pmatrix} h_{1,1} \\ h_{1,2} \end{pmatrix}, \ \boldsymbol{\Omega}_1 = \begin{pmatrix} w_5 & w_6 \end{pmatrix}, \ \beta_1 = b_3 \quad \text{e}$$

$$\boldsymbol{\Phi} = \{\boldsymbol{\Omega} = \{\boldsymbol{\Omega}_0, \boldsymbol{\Omega}_1\}, \boldsymbol{\beta} = \{\boldsymbol{\beta}_0, \beta_1\}\}.$$

```r
### Figura 1: Gerando o gráfico da superfície
n <- 100
x1 <- seq(-3, 3, length.out=n)
x2 <- seq(-3, 3, length.out=n)
dados.grid <- as_tibble(expand.grid(x1, x2)) %>%
  rename_all(~ c("x1", "x2")) %>%
  mutate(mu=abs(x1^3 - 30*sin(x2) + 10))

ggplot(dados.grid, aes(x=x1, y=x2)) +
  geom_point(aes(colour=mu), size=2, shape=15) +
  coord_cartesian(expand=F) +
  scale_colour_gradient(low="white",
                        high="black",
                        name=TeX("$E(Y|X_1, X_2)$")) +
  xlab(TeX("$X_1$")) + ylab(TeX("$X_2$"))
```

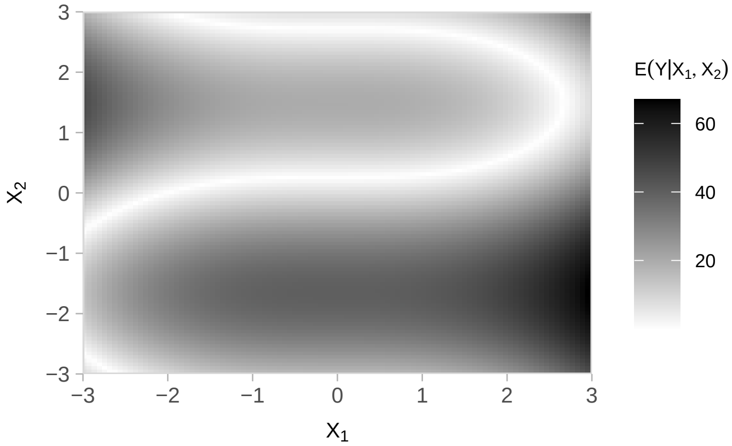

Figura 1: Gráfico da superfície do valor esperado da variável resposta Y em função das variáveis de entrada X_1 e X_2.

```r
### Figura 2: Arquitetura da RNA.
par(mar=c(0, 0, 0, 0))
wts_in <- rep(1, 9)
struct <- c(2, 2, 1) # dois inputs, dois neurônios escondidos e um output
plotnet(wts_in, struct = struct,
       x_names="", y_names="",
       node_labs=F, rel_rsc=.7)
aux <- list(
 x=c(-.8, -.8, 0, 0, .8, rep(-.55, 4), -.12, -0.06, .38, .38, .7),
 y=c(.73, .28, .73, .28, .5, .78, .68, .48, .32, .88, .5, .68, .44, .7),
 rotulo=c("x_1", "x_2", "h_1", "h_2", "\\hat{y}", "w_1", "w_3", "w_2", "w_4",
          "b_1", "b_2", "w_5", "w_6", "b_3")
)
walk(transpose(aux), ~ text(.$x, .$y,
                            TeX(str_c("$", .$rotulo, "$")), cex=.8))
```

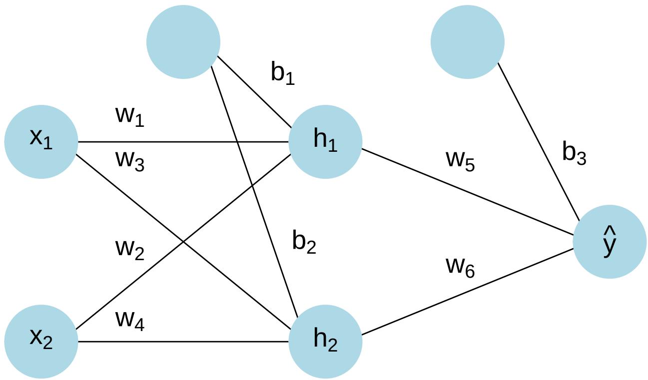

Figura 2: Arquitetura da rede neural artificial. Adotamos função de ativação sigmoide e linear nas camadas escondidas e de saída, respectivamente.

Considerando as informações acima, responda os itens a seguir.

a) Crie uma função computacional para calcular o valor previsto da variável resposta $\hat{y} = f(\boldsymbol{x}; \boldsymbol{\phi})$ em função de $\boldsymbol{x}$ e $\boldsymbol{\Phi}$. Use a função para calcular $\hat{y}$ para

$$\boldsymbol{\Phi}^\star = \left\{ \boldsymbol{\Omega} = \left\{ \boldsymbol{\Omega}_0 = \begin{pmatrix} 0.1 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}, \boldsymbol{\Omega}_1 = \begin{pmatrix} 0.1 & 0.1 \end{pmatrix} \right\}, \boldsymbol{\beta} = \left\{ \boldsymbol{\beta}_0 = \begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}, \boldsymbol{\beta}_1 = 0.1 \right\} \right\} \text{ e } \boldsymbol{x} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

b) Crie uma função computacional para calcular a função de perda $J(\boldsymbol{\phi})$. Em seguida, divida o conjunto de dados observados de modo que as **primeiras** 80.000 amostras componham o conjunto de **treinamento**, as próximas 10.000 o de **validação**, e as **últimas** 10.000 o de **teste**. Qual é a perda da rede **no conjunto de teste** quando $\boldsymbol{\phi} = \boldsymbol{\Phi}^\star$?

c) Use a regra da cadeia para encontrar expressões algébricas para o vetor gradiente

$$\nabla_{\boldsymbol{\phi}} J(\boldsymbol{\phi}) = \left(\frac{\partial J}{\partial w_1}, \ldots, \frac{\partial J}{\partial b_3} \right).$$

d) Crie uma função computacional que receba como entrada a lista $\boldsymbol{\Phi}$, uma matrix design ($\boldsymbol{x}$) e as respectivas observações ($\boldsymbol{y}$) e forneça, como saída, o gradiente definido no item c). Apresente o resultado da função aplicada sobre o **banco de treinamento**, quando $\boldsymbol{\Phi} = \boldsymbol{\Phi}^\star$. Atenção: implemente o algoritmo *back-propagation* para evitar realizar a mesma operação múltiplas vezes.

e) Aplique o método do gradiente para encontrar os parâmetros que minimizam a função de perda no **conjunto de treino**. Inicie o algoritmo no ponto

$$\boldsymbol{\Phi}^0 = \left\{ \boldsymbol{\Omega} = \left\{ \boldsymbol{\Omega}_0 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \boldsymbol{\Omega}_1 = \begin{pmatrix} 0 & 0 \end{pmatrix} \right\}, \boldsymbol{\beta} = \left\{ \boldsymbol{\beta}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \boldsymbol{\beta}_1 = 0 \right\} \right\},$$

use taxa de aprendizagem $\epsilon = 0.1$ e rode o algoritmo por 100 iterações. Para cada uma delas, calcule a perda no **conjunto de validação**. Apresente a lista de parâmetros estimados (isso é, aqueles que geraram a menor perda na validação), indique em qual iteração eles foram observados e comente o resultado.

f) Apresente o gráfico do custo no conjunto de treinamento e no de validação (uma linha para cada) em função do número da iteração do processo de otimização. Comente os resultados.