Universidade de Brasília

DEPARTAMENTO DE ESTATÍSTICA

9 de abril de 2024

Lista 1: Ajustando uma RNA "no braço".

Prof. Guilherme Rodrigues

Redes Neurais Profundas

Tópicos em Estatística 1

(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.**

Considere um processo gerador de dados da forma

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

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(1.2024)
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:

$$\begin{aligned} h_1 &= \phi(x_1 w_1 + x_2 w_3 + b_1) = \phi(a_1) \\ h_2 &= \phi(x_1 w_2 + x_2 w_4 + b_2) = \phi(a_2) \\ \hat{y} &= h_1 w_5 + h_2 w_6 + b_3, \end{aligned}$$

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

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

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

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

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

Uma representação gráfica da rede está apresentada na Figura 2.

Em notação matricial, a rede neural pode ser descrita por

$$\begin{aligned} \boldsymbol{a} &= \boldsymbol{W}^{(1)\top} \boldsymbol{x} + \boldsymbol{b}^{(1)} \\ \boldsymbol{h} &= \phi(\boldsymbol{a}) \\ \hat{y} &= \boldsymbol{W}^{(2)\top} \boldsymbol{h} + b_3, \end{aligned}$$

onde

$$\boldsymbol{W}^{(1)} = \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix}, \quad \boldsymbol{W}^{(2)} = \begin{pmatrix} w_5 \\ w_6 \end{pmatrix}, \quad \boldsymbol{b}^{(1)} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \boldsymbol{h} = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}, \quad \boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}.$$

Considerando as informações acima, responda os itens a seguir.

a) Crie uma função computacional para calcular o valor previso da variável resposta $\hat{y} = f(\boldsymbol{x}; \boldsymbol{\theta})$ em função de $\boldsymbol{x}$ e $\boldsymbol{\theta}$. Use a função para calcular $\hat{y}$ para $\boldsymbol{\theta} = (0.1, \ldots, 0.1)$ e $\boldsymbol{x} = (1, 1)$. Dica: veja o Algoritmo 6.3 do livro Deep Learning.

b) Crie uma rotina computacional para calcular a função de custo $J(\boldsymbol{\theta})$. 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 o custo da rede **no conjunto de teste** quando $\boldsymbol{\theta} = (0.1, \ldots, 0.1)$?

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

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

d) Crie uma função computacional que receba como entrada o vetor $\boldsymbol{\theta}$, uma matrix design (x) e as respectivas observações (y) e forneça, como saída, o gradiente definido no item c). Apresente o resultado da função aplicada sobre o **banco de treinamento**, quando $\boldsymbol{\theta} = (0.1, \ldots, 0.1)$. Atenção: implemente o algoritmo *back-propagation* (Algoritmo 6.4 do livro Deep Learning) para evitar realizar a mesma operação múltiplas vezes.

e) Aplique o método do gradiente para encontrar os parâmetros que minimizam a função de custo no **banco de validação**. Inicie o algoritmo no ponto $\boldsymbol{\theta} = (0, \ldots, 0)$, use taxa de aprendizagem $\epsilon = 0.1$ e rode o algoritmo por 100 iterações. Reporte o menor custo obtido e indique em qual iteração ele foi observado. Apresente também o vetor de pesos estimado 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 interação do processo de otimização. Comente os resultados.

g) Calcule os valores previstos ($\hat{y}_i$) e os resíduos ($y_i - \hat{y}_i$) da rede no conjunto de teste e represente-os graficamente em função de x_1 e x_2. Dica: tome como base o código usado para a visualização da superfície $(E(Y|X_1, X_2), X_1, X_2)$. Altere o gradiente de cores e, se necessário, use pontos semi-transparentes. Analise o desempenho da rede nas diferentes regiões do plano. Há locais onde o modelo é claramente viesado ou menos acurado?

h) Faça um gráfico do valor observado (y_i) em função do valor esperado ($\hat{y}_i = E(Y_i|x_{1i}, x_{2i})$) para cada observação do conjunto de teste. Interprete o resultado.

i) Para cada $k = 1, \ldots, 300$, recalcule o gradiente obtido no item d) usando apenas as k-primeiras observações do banco de dados original. Novamente, use $\boldsymbol{\theta} = (0.1, \ldots, 0.1)$. Apresente um gráfico com o valor do primeiro elemento do gradiente (isso é, a derivada parcial $\frac{\partial J}{\partial w_1}$) em função do número de amostras k. Como referência, adicione uma linha horizontal vermelha indicando o valor obtido em d). Em seguida, use a função `microbenchmark` para comparar o tempo de cálculo do gradiente para $k = 300$ e $k = 100000$. Explique de que maneira os resultados dessa análise podem ser usados para acelerar a execução do item e).

j) Ajuste sobre o conjunto de treinamento um modelo linear normal (**modelo linear 1**)

$$Y_i \sim N(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i}, \sigma)$$

usando a função `lm` do pacote `R` (ou outra equivalente). Em seguida, inclua na lista de covariáveis termos quadráticos e de interação linear. Isso é, assuma que no **modelo linear 2**,

$$E(Y|x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2.$$

Compare o erro quadrático médio no conjunto de teste dos dois modelos lineares acima com o da rede neural ajustada anteriormente. Qual dos 3 modelos você usaria para previsão? Justifique sua resposta.

k) Para cada modelo ajustado (os dois lineares e a rede neural), descreva o efeito no valor esperado da variável resposta causado por um aumento de uma unidade da covariável x_1?

l) Novamente, para cada um dos 3 modelos em estudo, calcule o percentual de vezes que o intervalo de confiança de 95% (para uma nova observação!) capturou o valor de y_i. Considere apenas os dados do conjunto de teste. No caso da rede neural, assuma que, aproximadamente, $\frac{y_i - \hat{y}}{\hat{\sigma}} \sim N(0,1)$, onde $\hat{\sigma}$ representa a raiz do erro quadrático médio da rede. Comente os resultados. Dica: para os modelos lineares, use a função `predict(mod, interval="prediction")`.

m) Para o **modelo linear 1**, faça um gráfico de disperção entre x_1 e x_2, onde cada ponto correponde a uma observação do conjunto de teste. Identifique os pontos que estavam contidos nos respectivos intervalos de confianças utiliizando a cor verde. Para os demais pontos, use vermelho. Comente o resultado.

```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("$X_2$"))
```

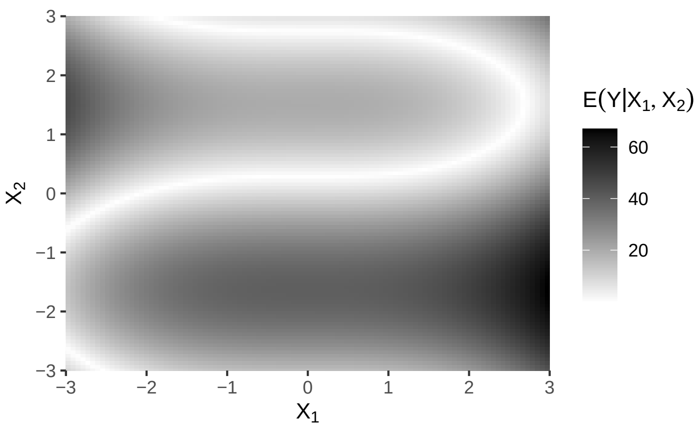

Figura 1: Gráfico da superfície do valor esperado da variável resposta Y em função das variáveis de entrada X_1 e X_2.

```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 otput
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}", paste0("w_", 1:4),
          "b_1", "b_2", "w_5", "w_6", "b_3")
)
walk(transpose(aux), ~ text(.$x, .$y,
                            TeX(str_c("$", .$rotulo, "$")), cex=.8))
```

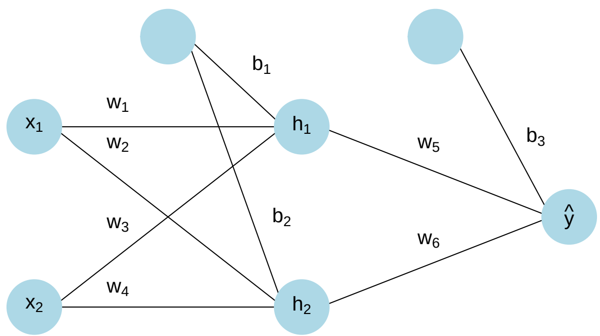

Figura 2: Arquitetura da rede neural artificial. Adotamos função de ativação sigmoide e linear nas camadas escondidas e de saída, respectivamente.