MAC0460 - Introdução ao aprendizado de máquina

Back-propagation 1

Felipe Salvatore
https://felipessalvatore.github.io/

Nina S. T. Hirata
https://www.ime.usp.br/~nina/

April 23, 2018

IME-USP: Institute of Mathematics and Statistics, University of São Paulo

Revisão: regressão logística

O problema de classificação

Em vários casos a função desconhecida $f : \mathbb{R}^d \to \mathbb{R}$ que queremos aproximar é uma **distribuição de probabilidade**.

Temos um vetor $\boldsymbol{x}$ e queremos saber a qual das classes $k_1, \ldots, k_n$ ele pertence. Um modo de formular esse problema como um problema de apreendizado supervisionado é coletar um conjunto de dados $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)$ onde $y_i \in \{k_1, \ldots, k_n\}$ e tentar estimar $p(y|\boldsymbol{x})$ por meio de uma família de modelos $p(y|\boldsymbol{x}; \boldsymbol{\theta})$.

Classificação com duas classes

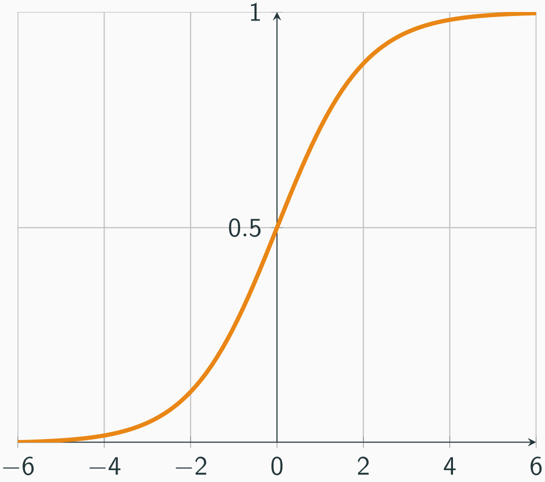
Quando y é uma variável binária definimos o modelo $p(y|\boldsymbol{x};\boldsymbol{\theta})$ do seguinte modo:

$$\begin{aligned}\hat{y} &= p(y=1|\boldsymbol{x};\boldsymbol{\theta})\\ &= h(\boldsymbol{x};\boldsymbol{\theta})\\ &= \sigma(z)\end{aligned}$$

em que

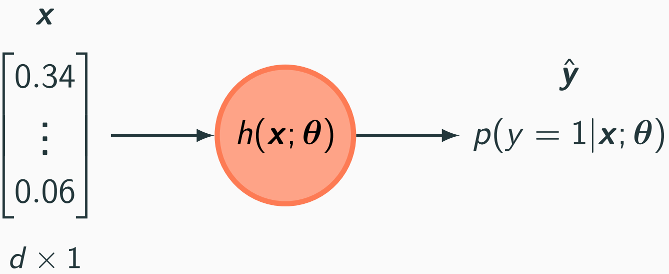
$$z = \boldsymbol{w}^\top \boldsymbol{x} + b$$

Revisão: função sigmoide



$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Classificação



Classificação para várias classes

E quando y é uma variável com n valores definimos $p(y|\boldsymbol{x};\boldsymbol{\theta})$ do seguinte modo:

$$
\begin{aligned}
\hat{\boldsymbol{y}} &= p(y|\boldsymbol{x};\boldsymbol{\theta}) \\
&= h(\boldsymbol{x};\boldsymbol{\theta}) \\
&= softmax(\boldsymbol{z})
\end{aligned}
$$

em que

$$\boldsymbol{z} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$$

Revisão: função softmax

$$\begin{bmatrix} 3.82 \\ 5.35 \\ 1.44 \\ -1.26 \\ 2.71 \\ 1.98 \end{bmatrix} \xrightarrow{\text{softmax}} \begin{bmatrix} 0.16115195 \\ 0.74422819 \\ 0.01491471 \\ 0.00100235 \\ 0.05310907 \\ 0.02559374 \end{bmatrix}$$

$$softmax(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_j e^{\boldsymbol{x}_j}}$$

$$
\begin{array}{ccccc}
\boldsymbol{x} & & & & \hat{\boldsymbol{y}} \\
\begin{bmatrix} 0.34 \\ \vdots \\ 0.06 \end{bmatrix} & \longrightarrow & h(\boldsymbol{x};\boldsymbol{\theta}) & \longrightarrow & \begin{bmatrix} p(y = 1|\boldsymbol{x};\boldsymbol{\theta}) \\ \vdots \\ p(y = n|\boldsymbol{x};\boldsymbol{\theta}) \end{bmatrix} \\
d \times 1 & & & & n \times 1
\end{array}
$$

Princípio da máxima verossimilhança

Os parâmetros $\boldsymbol{\theta}$ vão ser adaptados de modo que $p(y|\boldsymbol{x};\boldsymbol{\theta})$ seja a distribuição mais adequada para os dados

$$(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(N)}, y^{(N)})$$

Classificação

A função que queremos maximizar é

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{x},y\sim p_{data}} \log p(y|\boldsymbol{x};\boldsymbol{\theta}) \\ &= \frac{1}{N}\sum_{i=1}^{N} \log p(y^{(i)}|\boldsymbol{x}^{(i)};\boldsymbol{\theta})\end{aligned}$$

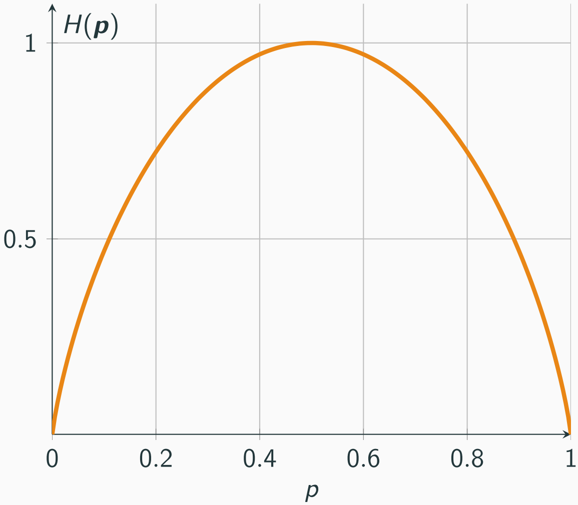
Revisão: entropia

$$\begin{array}{cc} \boldsymbol{p} & \boldsymbol{q} \\ \begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix} & \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \end{array}$$

$$H(\boldsymbol{p}) = 0.72 \qquad H(\boldsymbol{q}) = 1$$

$$H(\boldsymbol{p}) = \textstyle\sum_i \boldsymbol{p}_i \log \frac{1}{\boldsymbol{p}_i}$$

Revisão: entropia



$$\boldsymbol{p}$$

$$\begin{bmatrix} p \\ 1-p \end{bmatrix}$$

Revisão: divergência Kullback-Leibler

$$\boldsymbol{p} = \begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix} \qquad \boldsymbol{q} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \qquad \boldsymbol{p}' = \begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix} \qquad \boldsymbol{q}' = \begin{bmatrix} 0.88 \\ 0.12 \end{bmatrix}$$

$$D_{KL}(\boldsymbol{p}||\boldsymbol{q}) = 0.28 \qquad D_{KL}(\boldsymbol{p}'||\boldsymbol{q}') = 0.04$$

$$D_{KL}(\boldsymbol{p}||\boldsymbol{q}) = \sum_i \boldsymbol{p}_i \log \frac{\boldsymbol{p}_i}{\boldsymbol{q}_i}$$

Revisão: entropia cruzada

$$
\begin{aligned}
CE(\boldsymbol{p}, \boldsymbol{q}) &= H(\boldsymbol{p}) + D_{KL}(\boldsymbol{p}||\boldsymbol{q}) \\
&= -\sum_i \boldsymbol{p}_i \log(\boldsymbol{q}_i)
\end{aligned}
$$

$$
\arg\min_{\boldsymbol{q}} CE(\boldsymbol{p}, \boldsymbol{q}) = \arg\min_{\boldsymbol{q}} D_{KL}(\boldsymbol{p}, \boldsymbol{q})
$$

Entropia cruzada e verossimilhança

Assumindo que $\boldsymbol{y}$ é one-hot temos que:

$$
\begin{aligned}
L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}^{(i)}) &= CE(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}^{(i)}) \\
&= -\sum_{k=1}^{n} \boldsymbol{y}_k^{(i)} \log p(y = k | \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \\
&= -\log p(y^{(i)} | \boldsymbol{x}^{(i)}; \boldsymbol{\theta})
\end{aligned}
$$

Entropia cruzada e verossimilhança

E a função que queremos minimizar é

$$\begin{aligned} J(\boldsymbol{\theta}) &= \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}^{(i)}) \\ &= -\frac{1}{N}\sum_{i=1}^{N} \log p(y^{(i)}|\boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \\ &= -\mathcal{L}(\boldsymbol{\theta}) \end{aligned}$$

$$\arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

Treinando um modelo

- $\hat{\boldsymbol{y}} = f(\boldsymbol{x}; \boldsymbol{\theta})$
- $J(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} L(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}^{(i)})$
- algum algoritmo de otimização (e.g., **SGD**):

$$\boldsymbol{\theta}^{novo} \leftarrow \boldsymbol{\theta}^{velho} - \eta \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

Vamos ver como computar $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ de modo eficiente para uma função arbitrária J.

Grafo de computação (caso escalar)

Grafo de computação

Considere os seguintes conjuntos de funções:

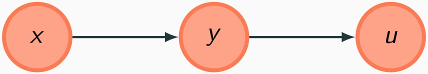
- $OP_1 = \{\lambda x. -x, \lambda x.x^2, \lambda x.e^x, \lambda x.log(x), \lambda x.x\}$
- $OP_2 = \{\lambda xy.x + y, \lambda xy.x * y, \lambda xy.\frac{x}{y}\}$
- $OP = OP_1 \cup OP_2$

Grafo de computação

Um grafo de computação definido em OP $\mathcal{G} = (\mathcal{V}, \mathcal{E}_1, \mathcal{E}_2)$ é um grafo acíclico dirigido (DAG) tal que cada elemento $u \in \mathcal{V}$ indica uma variável, se $(x, y) \in \mathcal{E}_1$ então $f(x) = y$ onde $f \in OP_1 \cup \{g(x, \alpha) | \alpha \in \mathbb{R}, g \in OP_2\}$, e se $(x, y) \in \mathcal{E}_2$ então $f(x) = y$ onde $f \in \{g(\alpha, x) | \alpha \in \mathbb{R}, g \in OP_2\}$.

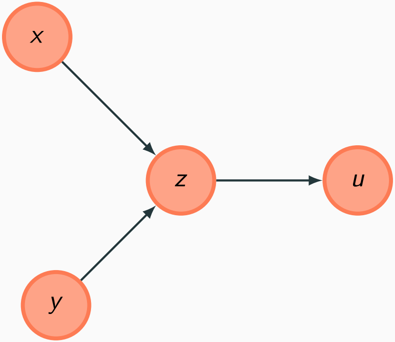
- $Pa(x) = \{y \in \mathcal{V} | (y, x) \in \mathcal{E}_1 \cup \mathcal{E}_2\}$.
- $S(x) = \{y \in \mathcal{V} | (x, y) \in \mathcal{E}_1 \cup \mathcal{E}_2\}$.

Grafo de computação



- $y = x^2$
- $u = e^y$

Grafo de computação



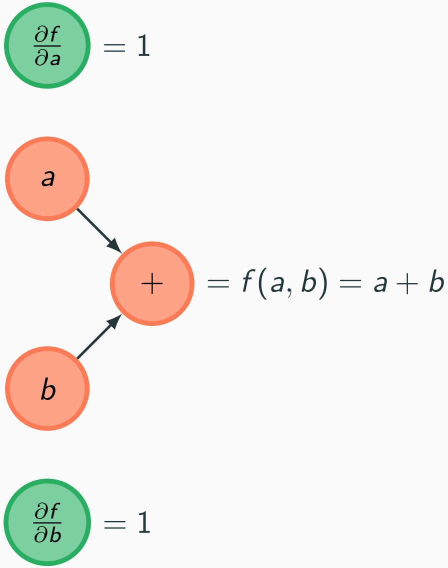
- $z = x + y$
- $u = \log(z)$

Queremos representar uma função L por um grafo definido em OP pois as derivadas parciais das funções de OP são simples de calcular. E com a a regra da cadeia podemos combinar as derivadas das funções locais para obter a derivada parcial de L com respeito a quaisquer parâmetros.

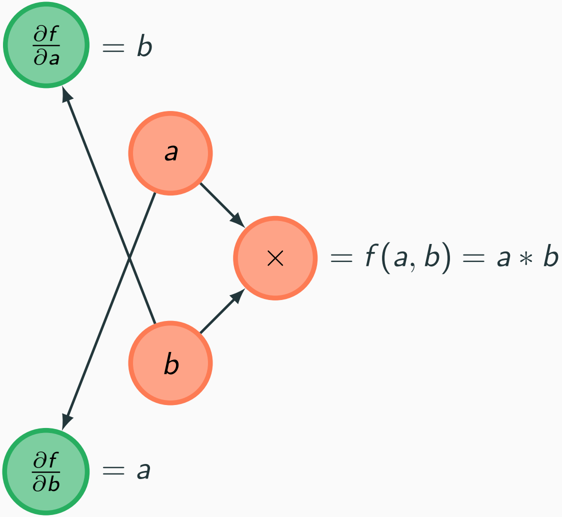
Grafo de computação

Como todas as funções em OP são diferenciáveis, podemos extender $\mathcal{G}$ em $\mathcal{G}'$ adicionando todas as derivadas parciais dos filhos em relação aos pais junto com as respectivas dependências.

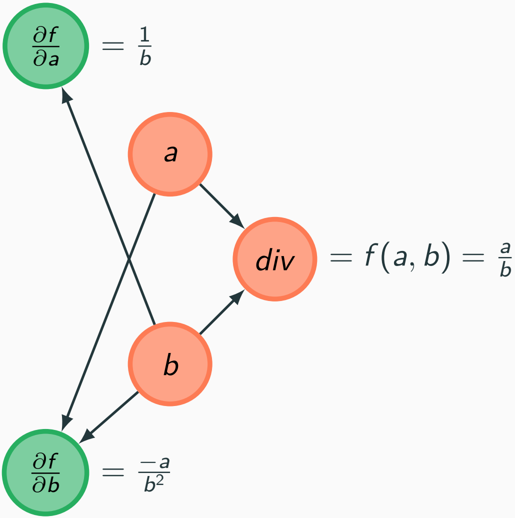
Extendendo o grafo de operações básicas: soma



Extendendo o grafo de operações básicas: multiplicação



Extendendo o grafo de operações básicas: divisão

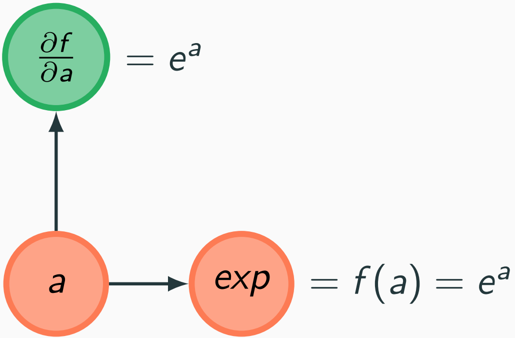


Extendendo o grafo de operações básicas: negativo

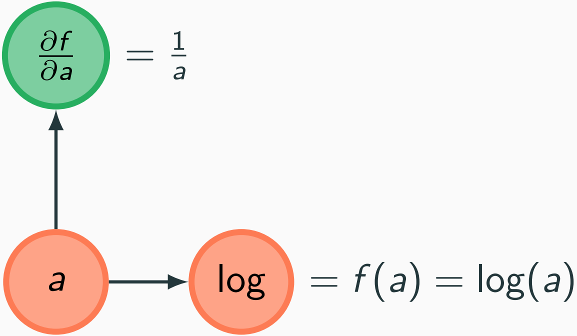
$$\frac{\partial f}{\partial a} = -1$$

$$a \longrightarrow *-1 = f(a) = -a$$

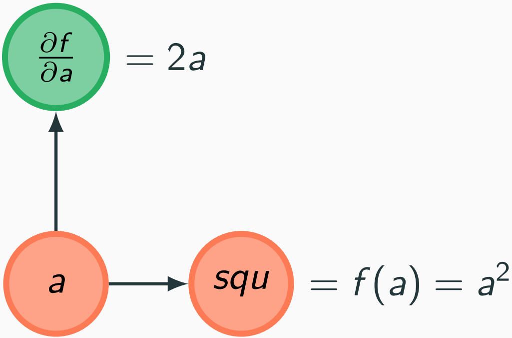
Extendendo o grafo de operações básicas: exponenciação



Extendendo o grafo de operações básicas: logarítimo

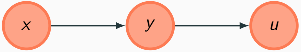


Extendendo o grafo de operações básicas: ao quadrado



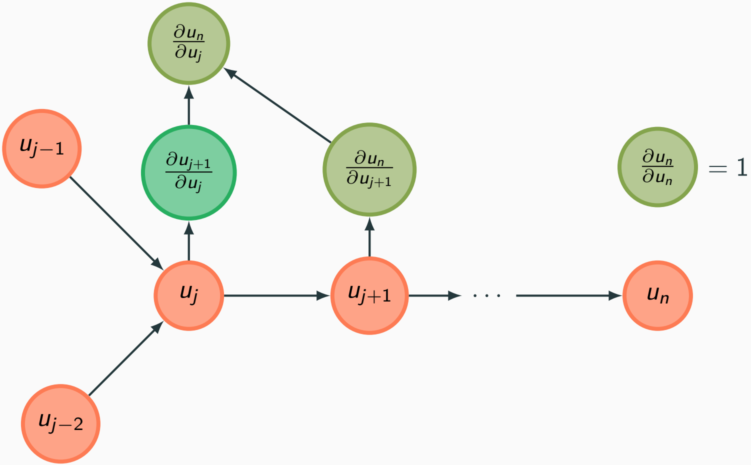
Regra da cadeia

- $f : \mathbb{R} \to \mathbb{R}$, $g : \mathbb{R} \to \mathbb{R}$.
- $y = g(x)$
- $u = f(g(x)) = f(y)$



$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial y}\frac{\partial y}{\partial x}$$

Aplicando a regra da cadeia



- $\frac{\partial u_n}{\partial u_j} = \frac{\partial u_n}{\partial u_{j+1}} \frac{\partial u_{j+1}}{\partial u_j}$

Exemplo 1: regressão linear

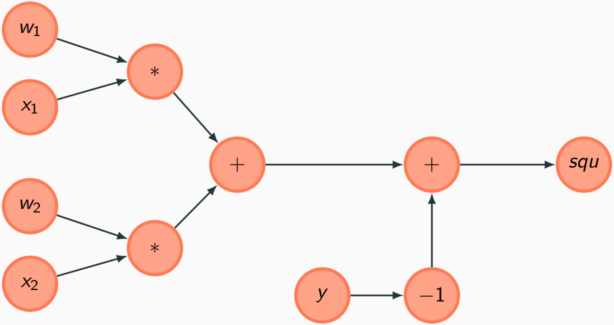
$$
\begin{aligned}
J(\boldsymbol{w}) &= \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{y}_i) \\
&= \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \\
&= \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2
\end{aligned}
$$

Simplificação

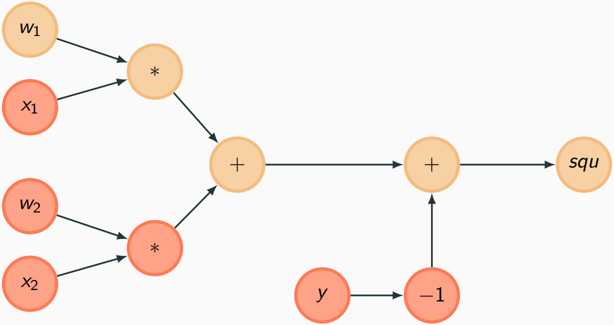
- $\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$

- $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

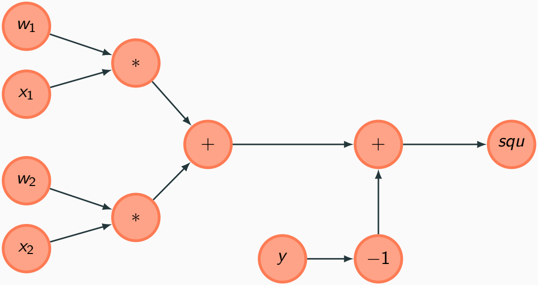
Grafo de $L(\hat{y}, y)$



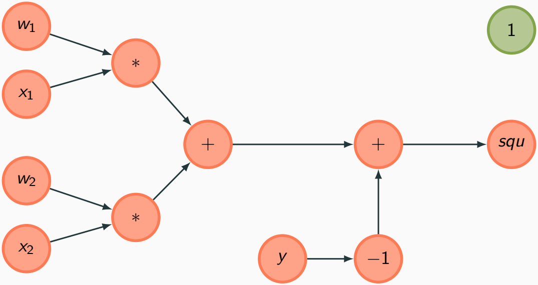
Caminho de w_1



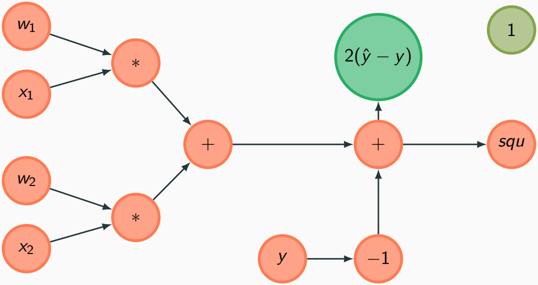
Derivade de L em relação a w_1



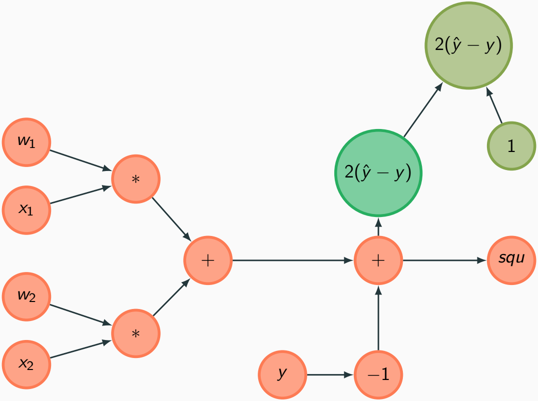
Derivade de L em relação a w_1



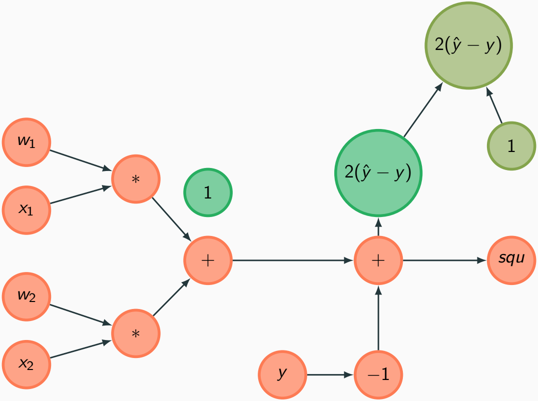
Derivade de L em relação a w_1



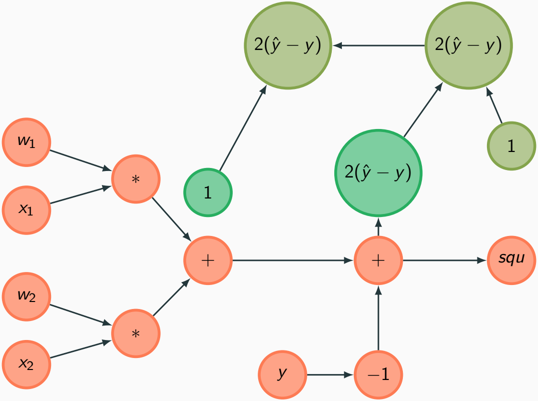
Derivade de L em relação a w_1



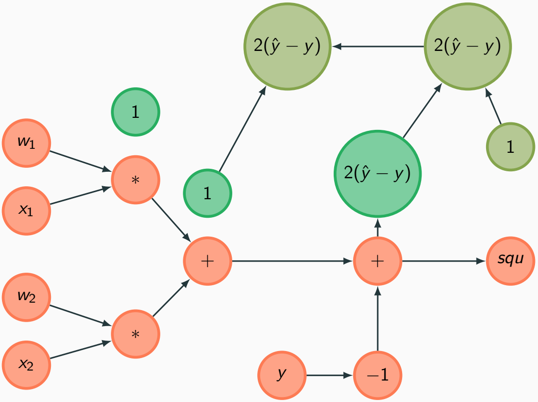
Derivade de L em relação a w_1



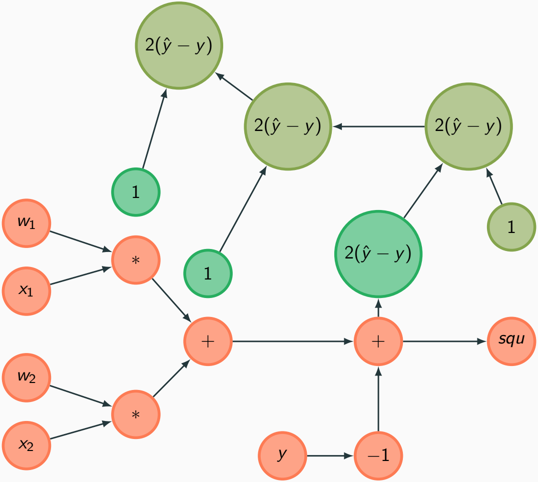
Derivade de L em relação a w_1



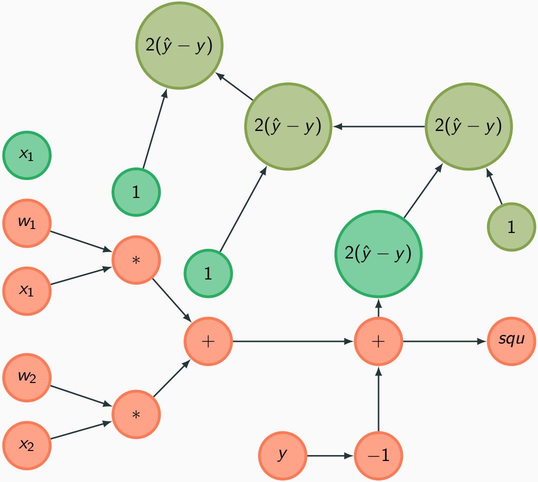
Derivade de L em relação a w_1



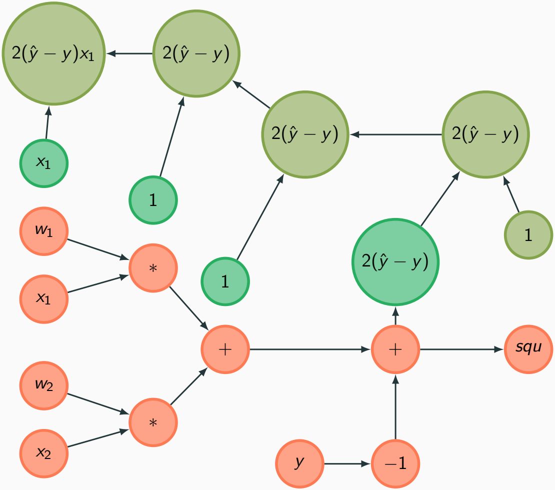
Derivade de L em relação a w_1



Derivade de L em relação a w_1



Derivade de L em relação a w_1

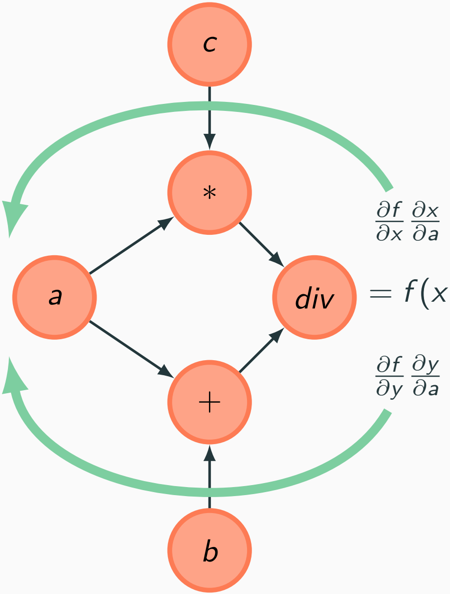


Regra da cadeia para várias variáveis

- $z = f(x, y)$
- $x = f_1(a)$.
- $y = f_2(a)$

$$\frac{\partial z}{\partial a} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial a} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial a}$$

Exemplo



$$= f(x, y) = \frac{x}{y} = \frac{a * c}{a + b}$$

Exemplo 2: regressão logística

$$\hat{\boldsymbol{y}} = softmax(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$$

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = CE(\boldsymbol{y}, \hat{\boldsymbol{y}})$$

$$L(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_i \boldsymbol{y}_i \log \left(\frac{exp(\sum_k \boldsymbol{W}_{i,k}\boldsymbol{x}_k + \boldsymbol{b}_i)}{\sum_j exp(\sum_k \boldsymbol{W}_{j,k}\boldsymbol{x}_k + \boldsymbol{b}_j)} \right)$$

Simplificação

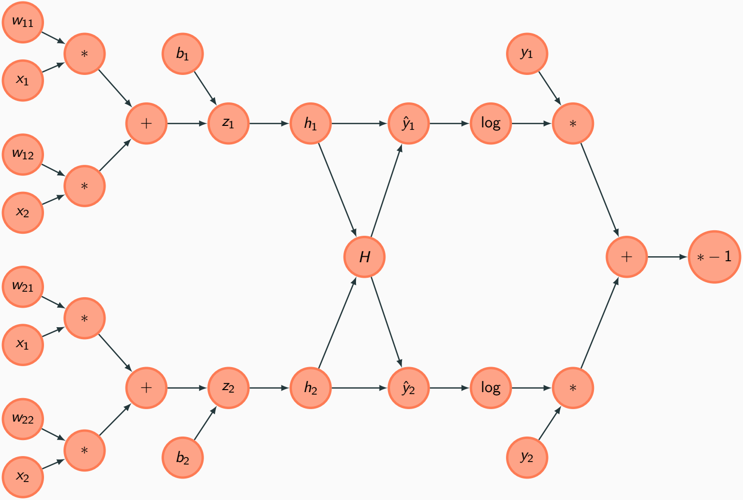
- $\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$

- $\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} exp(z_1) \\ exp(z_2) \end{bmatrix}$

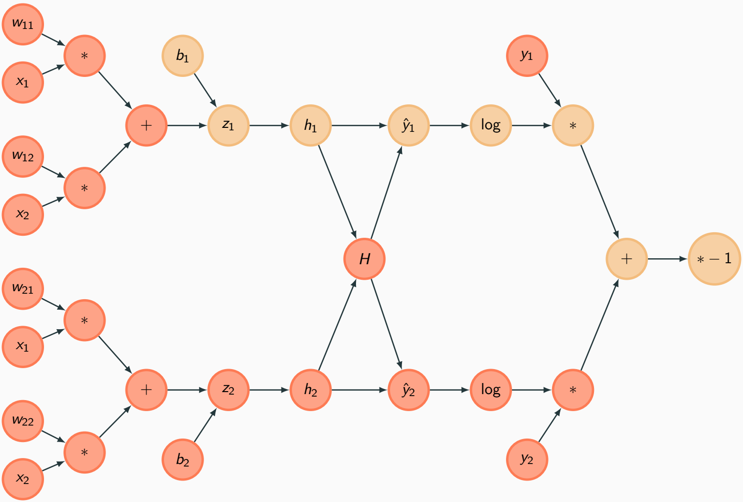
- $H = h_1 + h_2$

- $\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} = \begin{bmatrix} \frac{h_1}{H} \\ \frac{h_2}{H} \end{bmatrix}$

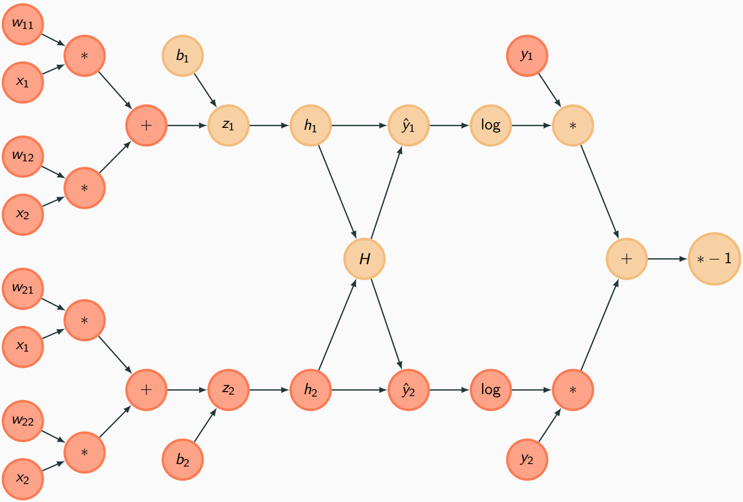
Grafo de $L(\hat{y}, y)$



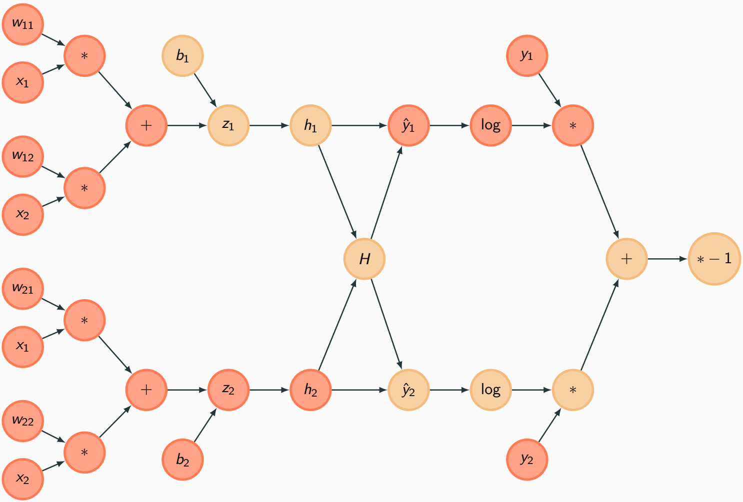
Caminho de b_1: 1



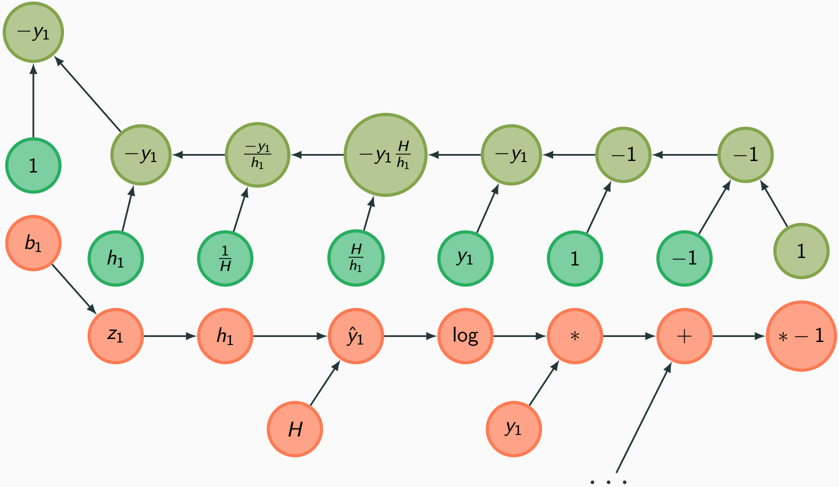
Caminho de b_1: 2



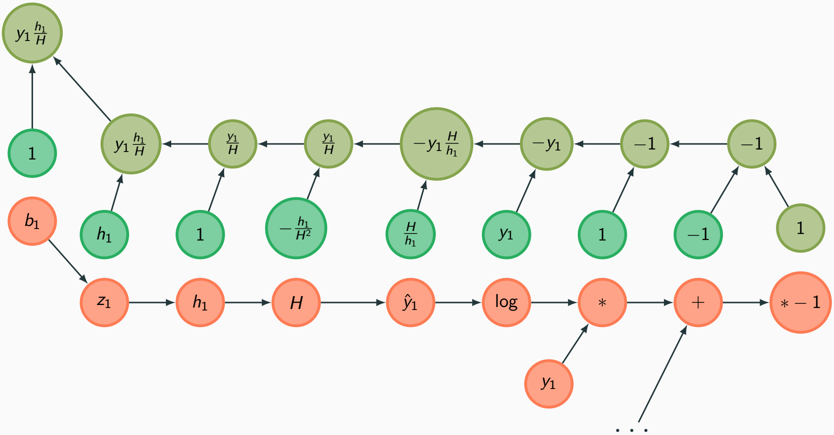
Caminho de b_1: 3



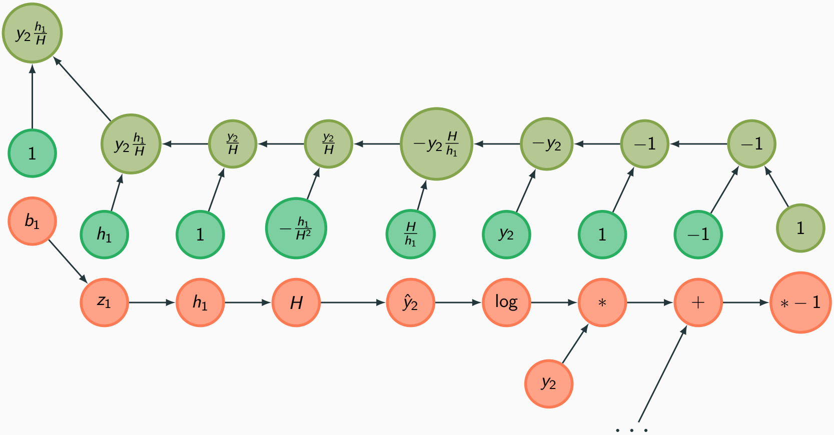
Derivada parcial de L com respeito a b_1: 1



Derivada parcial de L com respeito a b_1: 2



Derivada parcial de L com respeito a b_1: 3



Derivada parcial de L com respeito a b_1

$$\frac{\partial L}{\partial b_1} = -y_1 + y_1 \frac{h_1}{H} + y_2 \frac{h_1}{H}$$

Derivada parcial de L com respeito a b_1

$$\begin{aligned}
\frac{\partial L}{\partial b_1} &= -y_1 + y_1 \frac{h_1}{H} + y_2 \frac{h_1}{H} \\
&= y_1 \left(\frac{h_1}{H} - 1 \right) + y_2 \left(\frac{h_1}{H} - 0 \right)
\end{aligned}$$

Derivada parcial de L com respeito a b_1

$$\begin{aligned}\frac{\partial L}{\partial b_1} &= -y_1 + y_1\frac{h_1}{H} + y_2\frac{h_1}{H} \\ &= y_1\left(\frac{h_1}{H} - 1\right) + y_2\left(\frac{h_1}{H} - 0\right) \\ &= y_1\left(\hat{y}_1 - 1\right) + y_2\left(\hat{y}_1 - 0\right)\end{aligned}$$

Derivada parcial de L com respeito a b_1

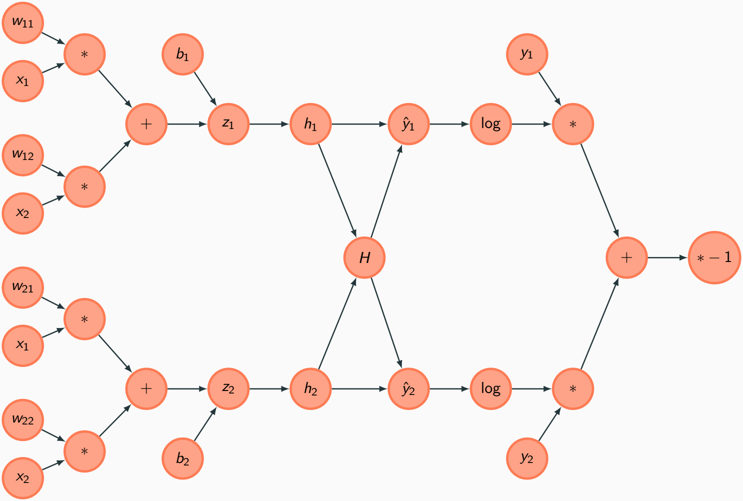
$$\begin{aligned}
\frac{\partial L}{\partial b_1} &= -y_1 + y_1 \frac{h_1}{H} + y_2 \frac{h_1}{H} \\
&= y_1 \left(\frac{h_1}{H} - 1 \right) + y_2 \left(\frac{h_1}{H} - 0 \right) \\
&= y_1 \left(\hat{y}_1 - 1 \right) + y_2 \left(\hat{y}_1 - 0 \right) \\
&= \hat{y}_1 - y_1 \quad \text{(quando } \; y \; \text{ é um vetor one-hot)}
\end{aligned}$$

Exemplo

$$\boldsymbol{W} = \begin{bmatrix} 0.65 & 1.19 \\ 0.69 & -0.92 \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} 0.2 \\ 0.7 \end{bmatrix}$$

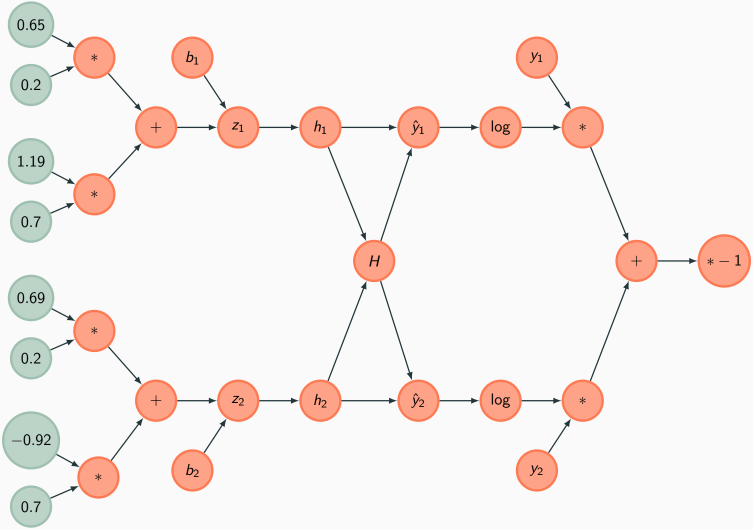
$$\boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Forward

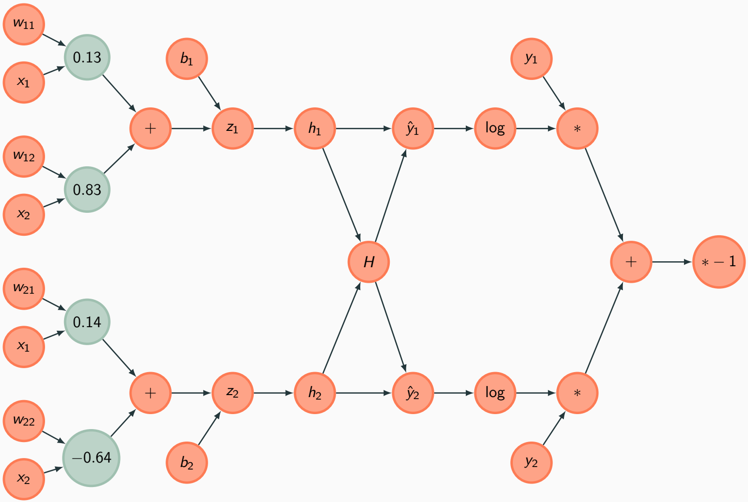


w_{11}
x_1
w_{12}
x_2
w_{21}
x_1
w_{22}
x_2
$*$
$*$
$*$
$*$
$+$
$+$
b_1
b_2
z_1
z_2
h_1
h_2
H
$\hat{y}_1$
$\hat{y}_2$
log
log
y_1
y_2
$*$
$*$
$+$
$* - 1$

Forward

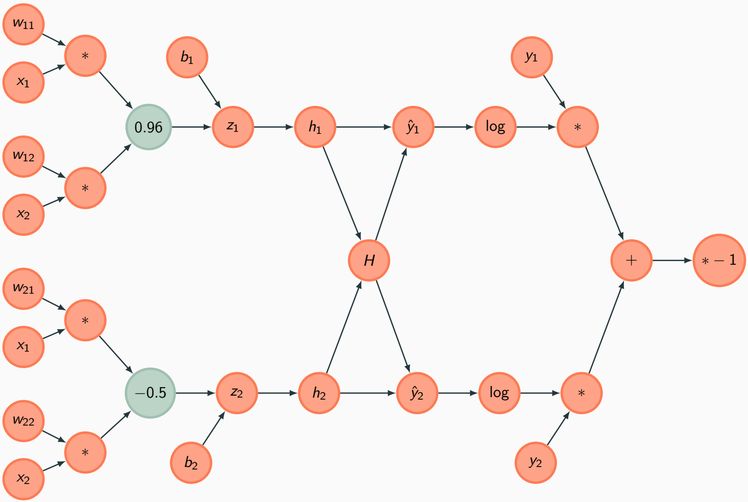

0.65
0.2
1.19
0.7
0.69
0.2
−0.92
0.7
*
+
b_1
b_2
z_1
z_2
h_1
h_2
H
$\hat{y}_1$
$\hat{y}_2$
log
y_1
y_2
$* - 1$

Forward



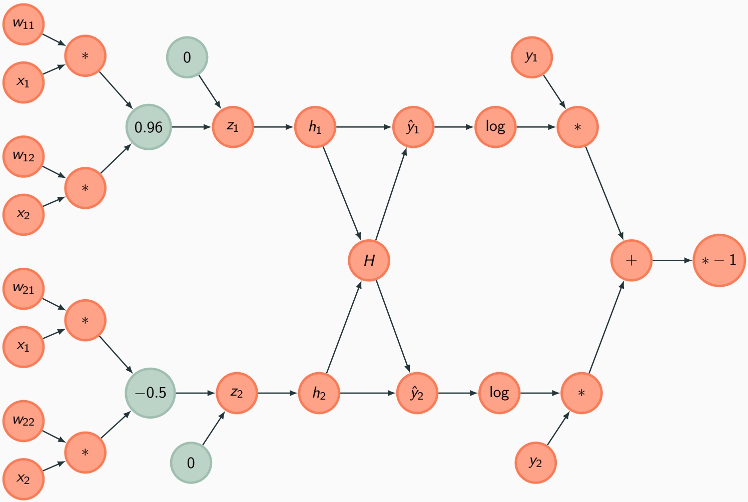
w_{11}
x_1
0.13
w_{12}
x_2
0.83
w_{21}
x_1
0.14
w_{22}
x_2
−0.64
+
+
b_1
b_2
z_1
z_2
h_1
h_2
H
$\hat{y}_1$
$\hat{y}_2$
log
log
y_1
y_2
$*$
$*$
+
$*-1$

Forward

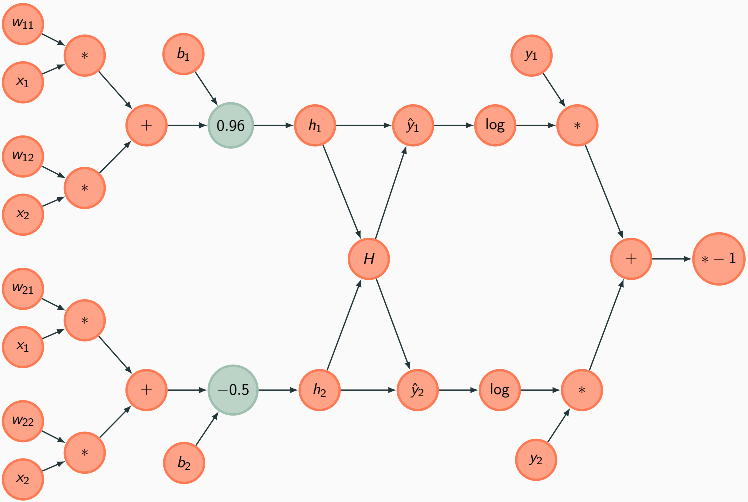


w_{11}
x_1
$*$
w_{12}
x_2
$*$
0.96
b_1
z_1
h_1
$\hat{y}_1$
log
y_1
$*$
H
$+$
$* - 1$
w_{21}
x_1
$*$
w_{22}
x_2
$*$
-0.5
b_2
z_2
h_2
$\hat{y}_2$
log
y_2
$*$

Forward

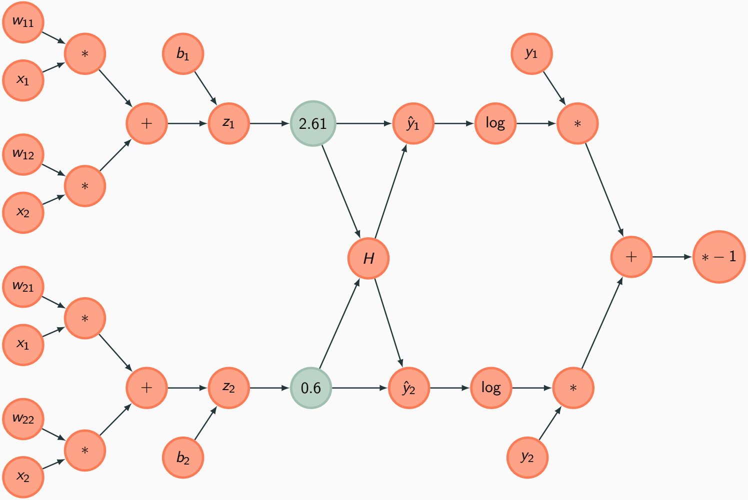

w_{11}
x_1
$*$
w_{12}
x_2
$*$
0.96
0
z_1
h_1
$\hat{y}_1$
log
y_1
$*$
w_{21}
x_1
$*$
w_{22}
x_2
$*$
-0.5
0
z_2
h_2
$\hat{y}_2$
log
y_2
$*$
H
$+$
$*-1$

Forward



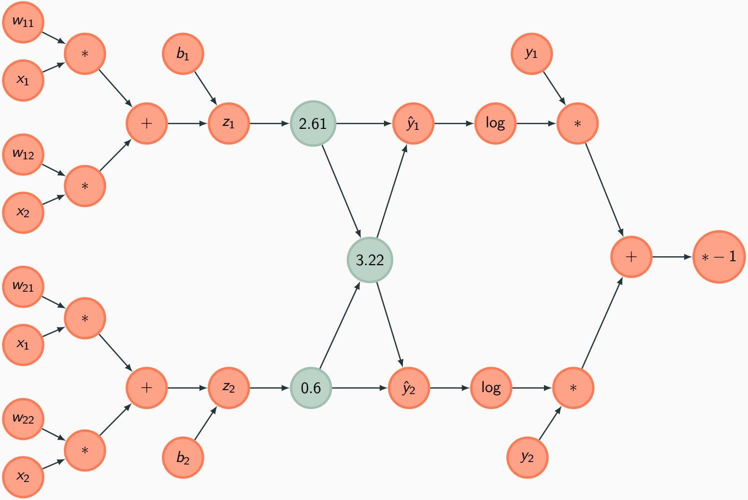
w_{11}
x_1
$*$
w_{12}
x_2
$*$
$+$
b_1
0.96
h_1
$\hat{y}_1$
log
y_1
$*$
H
$+$
$* - 1$
w_{21}
x_1
$*$
w_{22}
x_2
$*$
$+$
-0.5
b_2
h_2
$\hat{y}_2$
log
$*$
y_2

Forward



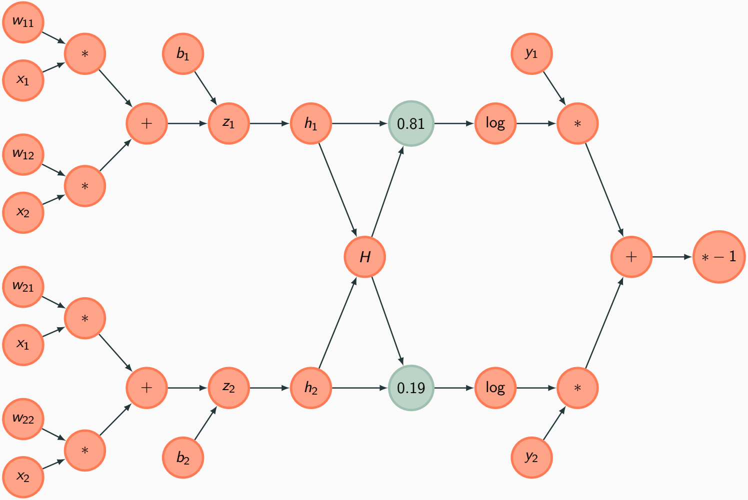
w_{11}
x_1
$*$
w_{12}
x_2
$*$
$+$
b_1
z_1
2.61
$\hat{y}_1$
log
y_1
$*$
H
$+$
$* - 1$
w_{21}
x_1
$*$
w_{22}
x_2
$*$
$+$
z_2
b_2
0.6
$\hat{y}_2$
log
$*$
y_2

Forward



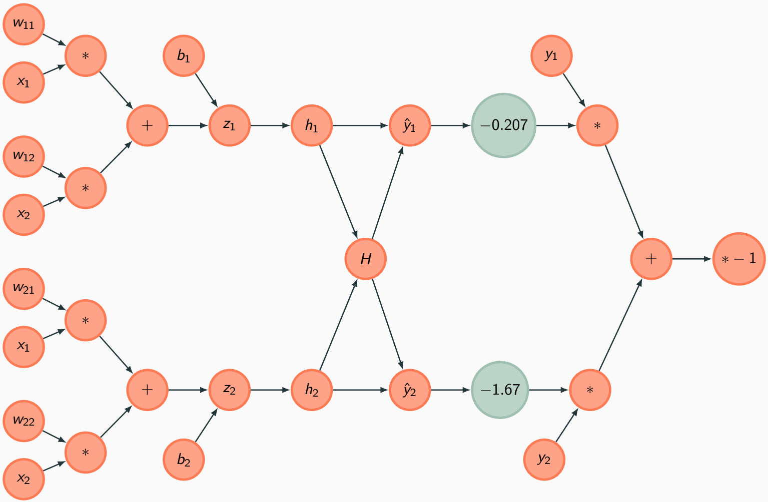
w_{11}
x_1
w_{12}
x_2
w_{21}
x_1
w_{22}
x_2
$*$
$*$
$*$
$*$
$+$
$+$
b_1
b_2
z_1
z_2
2.61
3.22
0.6
$\hat{y}_1$
$\hat{y}_2$
log
log
y_1
y_2
$*$
$*$
$+$
$* - 1$

Forward



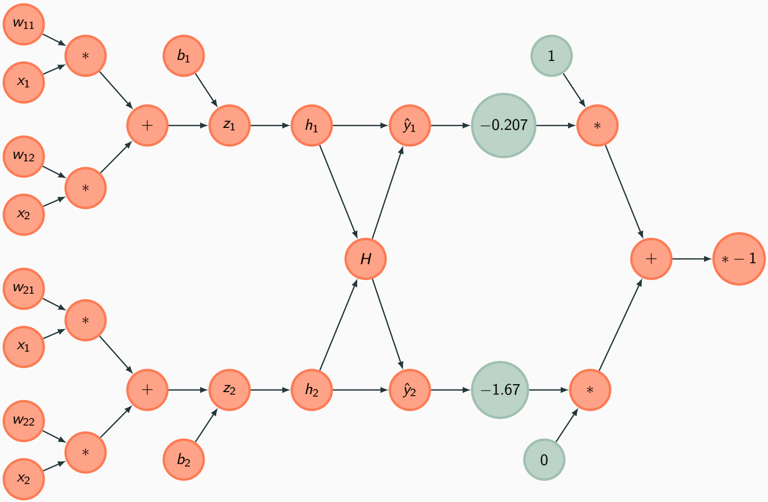
w_{11}
x_1
w_{12}
x_2
w_{21}
x_1
w_{22}
x_2
$*$
$*$
$*$
$*$
$+$
$+$
b_1
b_2
z_1
z_2
h_1
h_2
H
0.81
0.19
log
log
y_1
y_2
$*$
$*$
$+$
$* - 1$

Forward



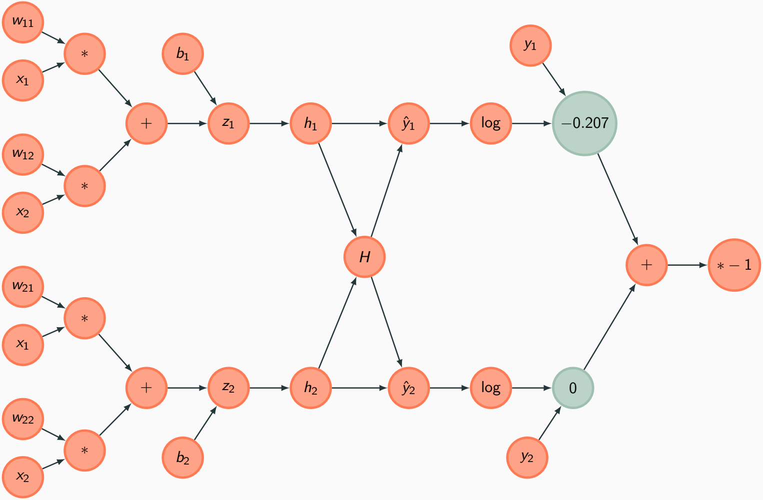
w_{11}
x_1
$*$
w_{12}
x_2
$*$
$+$
b_1
z_1
h_1
$\hat{y}_1$
-0.207
y_1
$*$
H
$+$
$*-1$
w_{21}
x_1
$*$
w_{22}
x_2
$*$
$+$
b_2
z_2
h_2
$\hat{y}_2$
-1.67
y_2
$*$

Forward

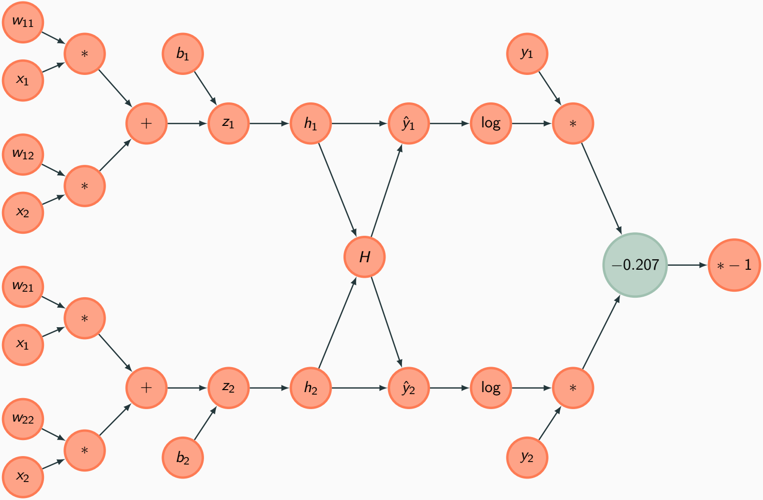


w_{11}
x_1
$*$
b_1
w_{12}
x_2
$*$
$+$
z_1
h_1
$\hat{y}_1$
-0.207
1
$*$
H
$+$
$* - 1$
w_{21}
x_1
$*$
w_{22}
x_2
$*$
$+$
z_2
b_2
h_2
$\hat{y}_2$
-1.67
$*$
0

Forward

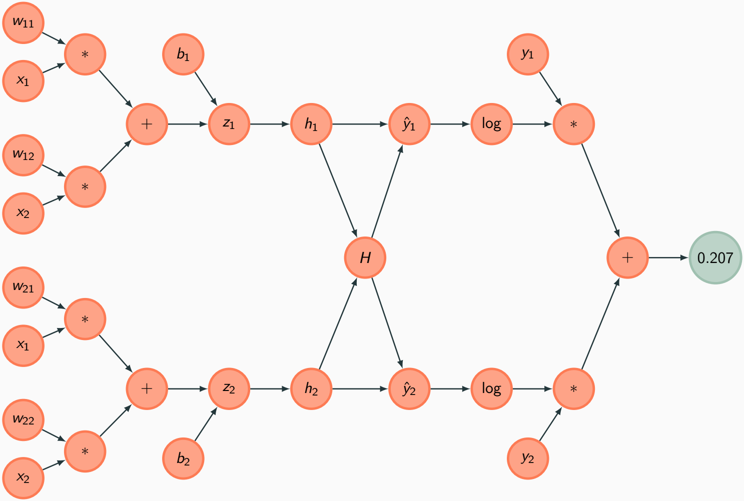

w_{11}
x_1
$*$
w_{12}
x_2
$*$
$+$
b_1
z_1
h_1
$\hat{y}_1$
log
y_1
-0.207
H
w_{21}
x_1
$*$
w_{22}
x_2
$*$
$+$
b_2
z_2
h_2
$\hat{y}_2$
log
y_2
0
$+$
$* - 1$

Forward



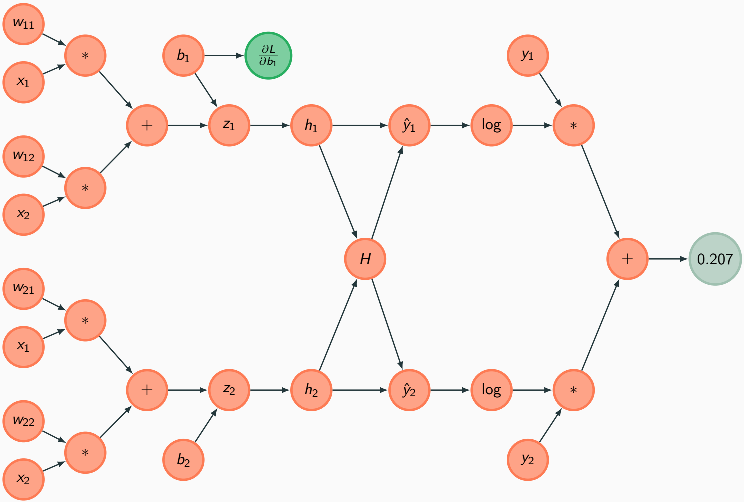
w_{11}
x_1
$*$
w_{12}
x_2
$*$
$+$
b_1
z_1
h_1
$\hat{y}_1$
log
y_1
$*$
H
-0.207
$* - 1$
w_{21}
x_1
$*$
w_{22}
x_2
$*$
$+$
b_2
z_2
h_2
$\hat{y}_2$
log
y_2
$*$

Forward



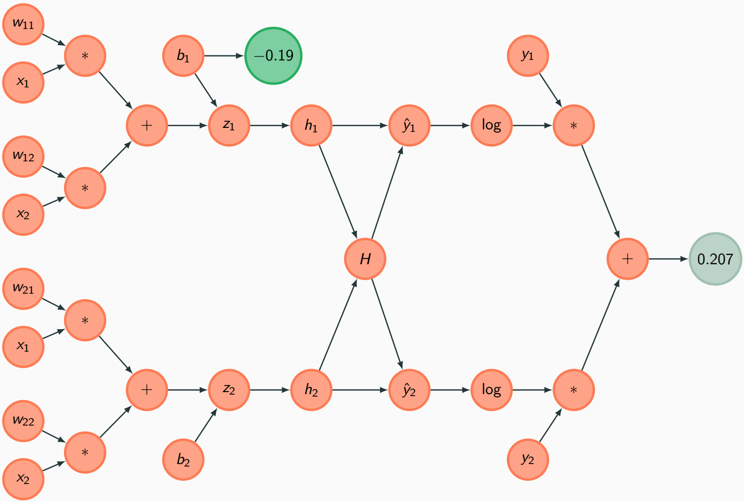
w_{11}
x_1
w_{12}
x_2
w_{21}
x_1
w_{22}
x_2
$*$
$*$
$*$
$*$
$+$
$+$
b_1
b_2
z_1
z_2
h_1
h_2
H
$\hat{y}_1$
$\hat{y}_2$
log
log
y_1
y_2
$*$
$*$
$+$
0.207

Forward



w_{11}
x_1
w_{12}
x_2
w_{21}
x_1
w_{22}
x_2
$*$
$*$
$*$
$*$
$+$
$+$
b_1
$\frac{\partial L}{\partial b_1}$
b_2
z_1
z_2
h_1
h_2
H
$\hat{y}_1$
$\hat{y}_2$
log
log
y_1
y_2
$*$
$*$
$+$
0.207

Backward



w_{11}
x_1
$*$
b_1
-0.19
y_1
$+$
z_1
h_1
$\hat{y}_1$
log
$*$
w_{12}
x_2
$*$
H
$+$
0.207
w_{21}
x_1
$*$
$+$
z_2
h_2
$\hat{y}_2$
log
$*$
w_{22}
x_2
$*$
b_2
y_2

Algoritmo de back-propagation (caso escalar)

Algorithm 1 Back-propagation (scalar case)

1: **Require:** Computational graph $\mathcal{G} = (\{u_1, \ldots, u_n\}, \mathcal{E}_1, \mathcal{E}_2)$, where u_n is a leaf node.
2: Initialize *grad_table*, a data structure that will store the derivatives that have been computed (at the end $grad\_table[u_i] = \frac{\partial u_n}{\partial u_i}$).
3: $grad\_table[u_n] \leftarrow 1$
4: **for** $j = n - 1$ down to 1 **do**
5: $\quad grad\_table[u_j] \leftarrow \sum_{u_i \in S(u_j)} grad\_table[u_i] \frac{\partial u_i}{\partial u_j}$
6: **end for**
7: **return** *grad_table*

Referências I

I. Goodfellow, Y. Bengio, and A. Courville.
***Deep Learning*.**
MIT Press, 2017.