Campus UFRJ-Macaé
Professor Aloísio Teixeira

Universidade Federal do Rio de Janeiro
Campus Macaé

UFRJ

Trabalho de Conclusão de Curso em Engenharia Mecânica

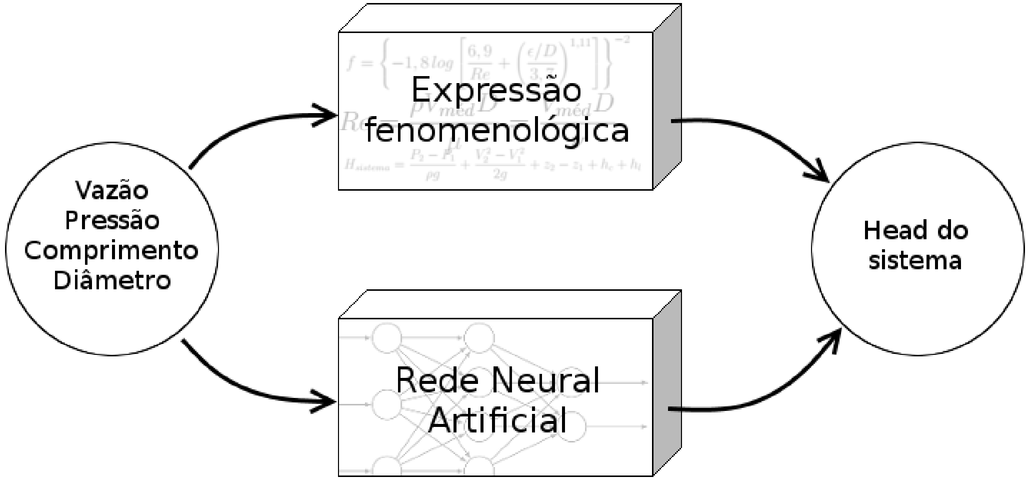
Modelagem de um Sistema Hidráulico Através de Redes Neurais Artificiais

Roberto de Freitas Cabral
Orientador: M. Sc. Marcelo Silva

Macaé (RJ), 07 de dezembro de 2018

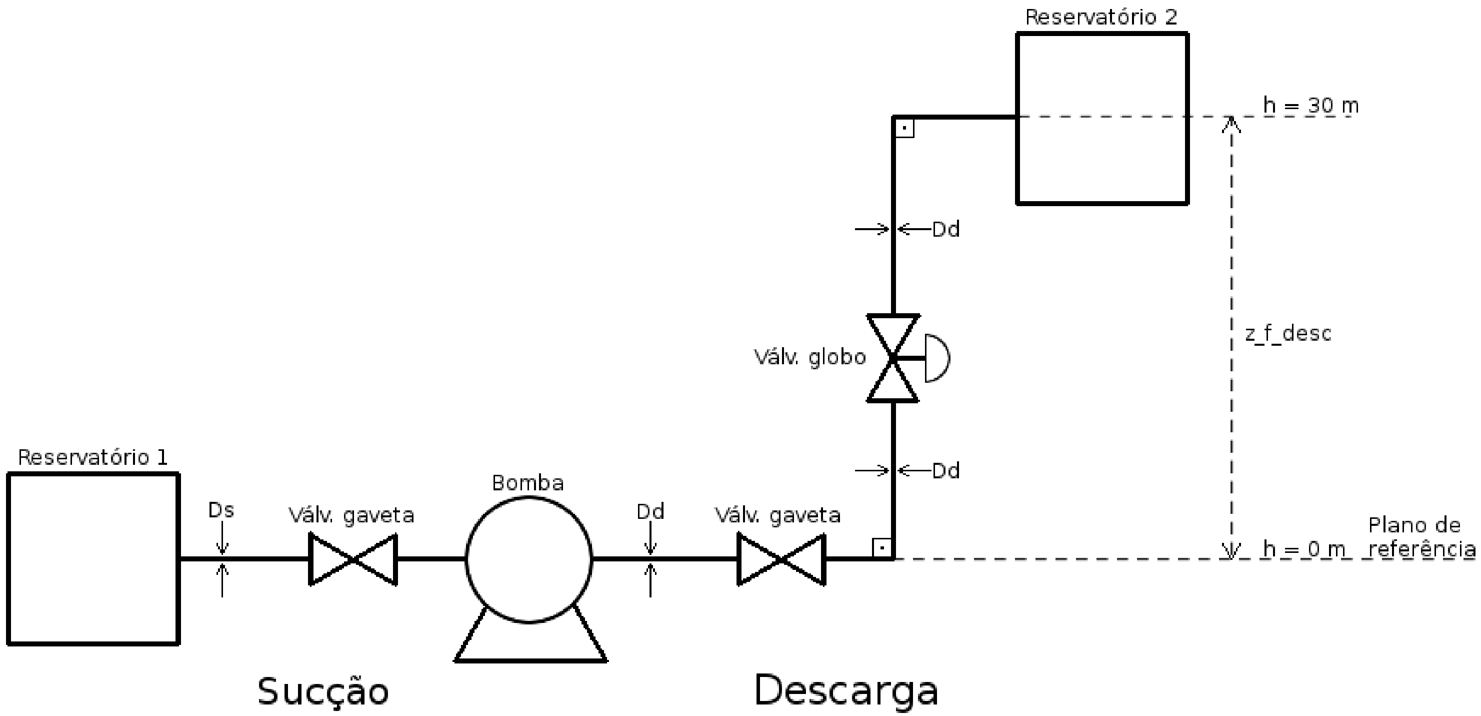
Introdução

- **Objetivo:** modelar um sistema hidráulico por meio de uma RNA e comparar os resultados com os da modelagem tradicional.
- **Motivação:** fornecer uma alternativa à modelagem tradicional que tem o potencial de se beneficiar da maior quantidade de dados disponíveis.

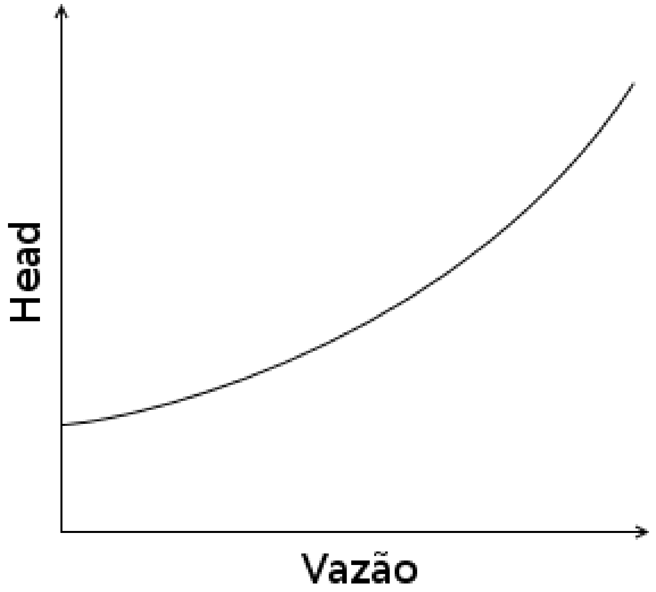


Fonte: Elaborado pelo autor.

Introdução



Fonte: Elaborado pelo autor.



Fonte: Elaborado pelo autor.

A Equação da Energia

$$H_{sis} = \frac{P_2 - P_1}{\rho g} + \frac{V_2^2 - V_1^2}{2g} + z_2 - z_1 + h_c + h_l$$

A Equação da Energia

↳ Head do sistema

$$H_{sis} = \frac{P_2 - P_1}{\rho g} + \frac{V_2^2 - V_1^2}{2g} + z_2 - z_1 + h_c + h_l$$

Head do sistema

$H = Y/g$ $\quad\quad$ Y: Energia específica (em J/kg)

O <u>head do sistema</u> (em m) é a energia que precisa ser adicionada ao fluido para levá-lo de um ponto a outro.

A Equação da Energia

↳ Energia do escoamento

$$H_{sis} = \overbrace{\frac{P_2 - P_1}{\rho g}}^{\text{Energia do escoamento}} + \frac{V_2^2 - V_1^2}{2g} + z_2 - z_1 + h_c + h_l$$

A <u>energia do escoamento</u> (em m) é uma forma de energia mecânica presente devido à pressão do fluido.

A Equação da Energia

↳ Energia cinética

$$H_{sis} = \frac{P_2 - P_1}{\rho g} + \overbrace{\frac{V_2^2 - V_1^2}{2g}}^{\text{Energia cinética}} + z_2 - z_1 + h_c + h_l$$

A <u>energia cinética</u> (em m) é uma forma de energia mecânica presente devido ao movimento relativo das partículas de fluido.

A Equação da Energia

↳ Energia potencial gravitacional

$$H_{sis} = \frac{P_2 - P_1}{\rho g} + \frac{V_2^2 - V_1^2}{2g} + \overbrace{z_2 - z_1}^{\text{Energia potencial gravitacional}} + h_c + h_l$$

A <u>energia potencial gravitacional</u> (em m) é uma forma de energia mecânica presente devido às interações das partículas do fluido com o campo gravitacional local.

A Equação da Energia

↳ Perda de carga contínua - $\boldsymbol{h_c}$

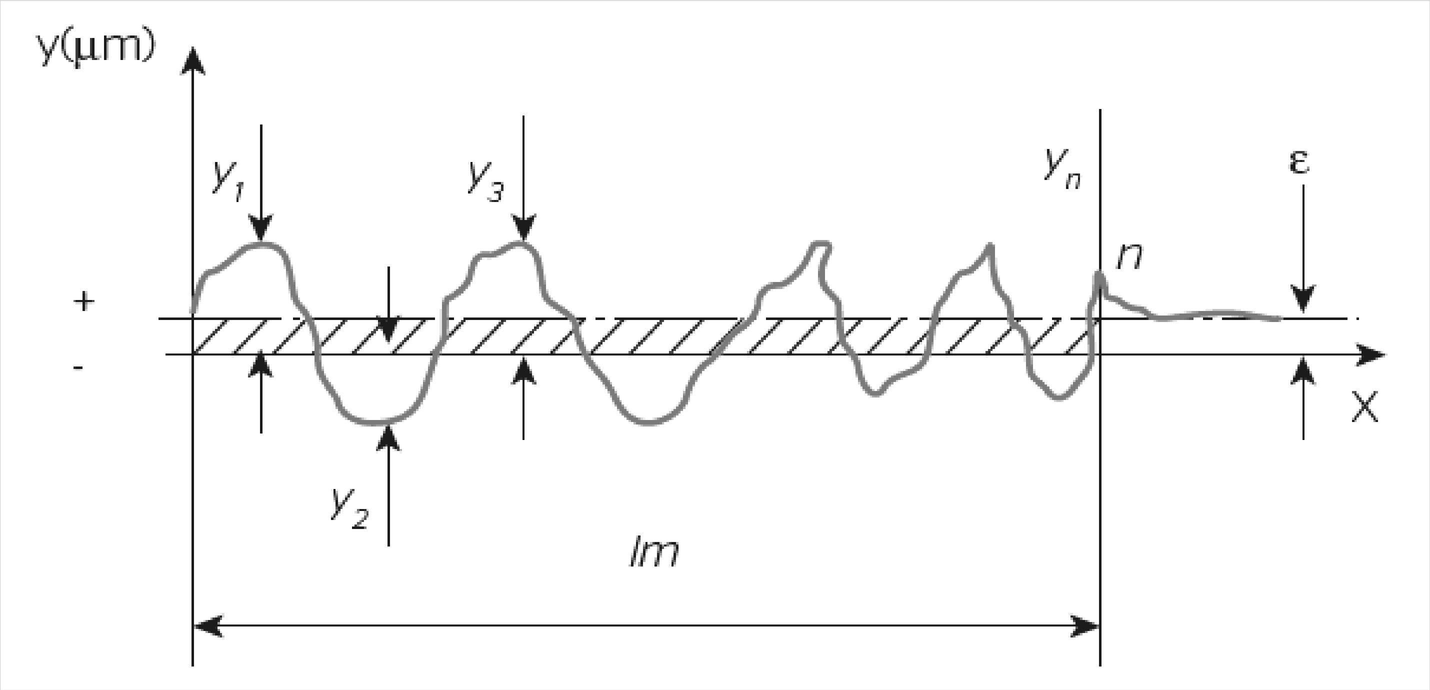
$$H_{sis} = \frac{P_2 - P_1}{\rho g} + \frac{V_2^2 - V_1^2}{2g} + z_2 - z_1 + h_c + h_l$$

Perda de carga contínua

A <u>perda de carga contínua</u> (em m) corresponde à energia do fluido que é dissipada devido ao atrito com as paredes da tubulação ao longo do seu comprimento.

A Equação da Energia

↳ Perda de carga contínua - h_c (cont.)



Fonte: Adaptado de Kellner, Akutsu e Reis (2016, p. 3).

A Equação da Energia

↳ Perda de carga contínua - $\boldsymbol{h_c}$ (cont.)

$$H_{sis} = \frac{P_2 - P_1}{\rho g} + \frac{V_2^2 - V_1^2}{2g} + z_2 - z_1 + h_c + h_l$$

$$h_c = f\,\frac{L}{D}\frac{V_{méd}^2}{2g}$$

Colebrook (1939):

$$\frac{1}{\sqrt{f}} = -2\log\left(\frac{\epsilon/D}{3{,}7} + \frac{2{,}51}{Re\sqrt{f}}\right)$$

Zigrang e Sylvester (1982):

$$\frac{1}{\sqrt{f}} = -2\log\left\{\frac{\epsilon/D}{3{,}7} - \frac{5{,}02}{Re}\log\left[\frac{\epsilon/D}{3{,}7} - \frac{5{,}02}{Re}\log\left(\frac{\epsilon/D}{3{,}7} + \frac{13}{Re}\right)\right]\right\}$$

A Equação da Energia

↳ Perda de carga local - $\boldsymbol{h_l}$

$$H_{sis} = \frac{P_2 - P_1}{\rho g} + \frac{V_2^2 - V_1^2}{2g} + z_2 - z_1 + h_c + h_l$$

Perda de carga local

A perda de carga local (em m) corresponde à energia do fluido que é dissipada devido ao atrito do fluido com obstáculos como válvulas, curvas, cotovelos, etc.

A Equação da Energia

↳ Perda de carga local - $\boldsymbol{h_l}$ (cont.)

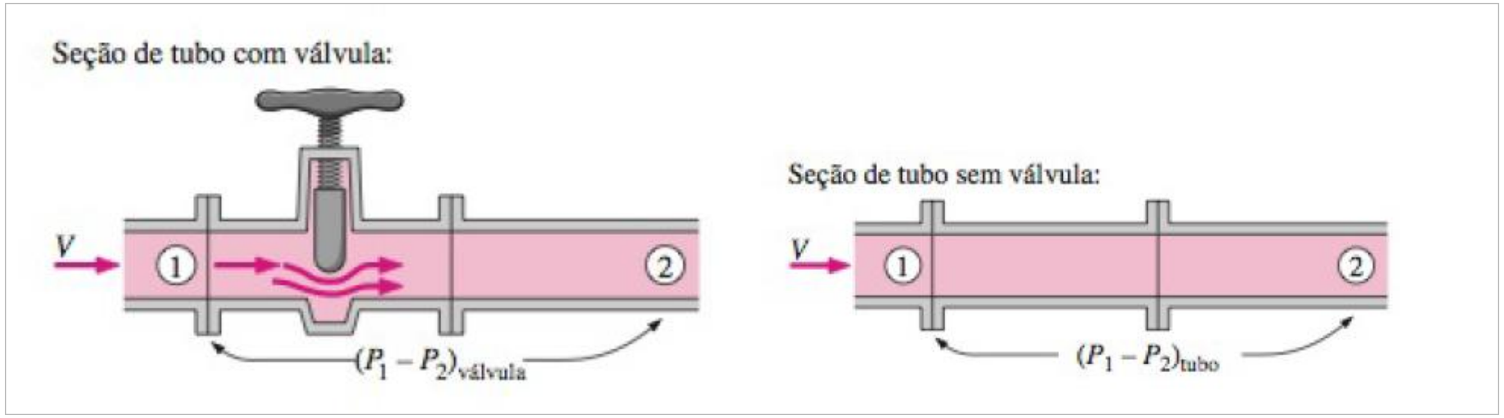
$$H_{sis} = \frac{P_2 - P_1}{\rho g} + \frac{V_2^2 - V_1^2}{2g} + z_2 - z_1 + h_c + h_l$$

$$h_l = K\frac{V_{méd}^2}{2g}$$

$$\Delta P = (P_1 - P_2)_{com} - (P_1 - P_2)_{sem}$$

$$K = \frac{\Delta P}{\frac{1}{2}\rho V^2}$$

Empiricamente obtido



Fonte: Adaptado de Çengel e Cimbala (2012, p. 301)

A Equação da Energia

$$H_{sis} = \overbrace{\frac{P_2 - P_1}{\rho g}}^{\text{Energia do escoamento}} + \overbrace{\frac{V_2^2 - V_1^2}{2g}}^{\text{Energia cinética}} + \overbrace{z_2 - z_1}^{\text{Energia potencial gravitacional}} + h_c + h_l$$

Head do sistema

Perda de carga contínua

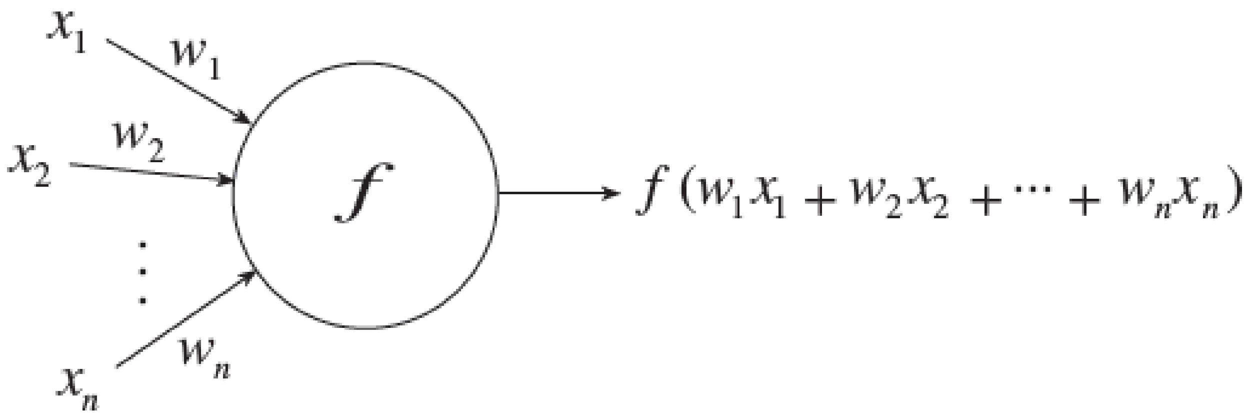
Perda de carga local

Redes Neurais Artificiais

As redes neurais artificiais (RNAs) representam uma forma alternativa de computação.

Neurônio artificial ⇨ componente fundamental da RNA

- Recebe dados
- Processa
- Fornece uma saída



Fonte: Rojas (1996, p. 23).

Função de propagação

A função de propagação é a **primeira etapa** do processamento dos dados de entrada num neurônio artificial.

$$net_j = \sum_{i \in I} (o_i \cdot w_{i,j}) + b_j$$

O função de propagação recebe um **vetor** e fornece um **escalar** conhecido como *entrada de rede,* net_j.

Função de ativação

A ativação de um neurônio, a_j, é uma medida do **quão ativo** um neurônio está num dado momento.

A **função de ativação**, f_{act}, recebe a entrada de rede net_j e fornece um valor de ativação, a_j.

$$a_j = f_{act}(net_j)$$

Continuidade e diferenciabilidade $\Rightarrow$ Propriedades necessárias da função de ativação

Função de ativação (cont.)

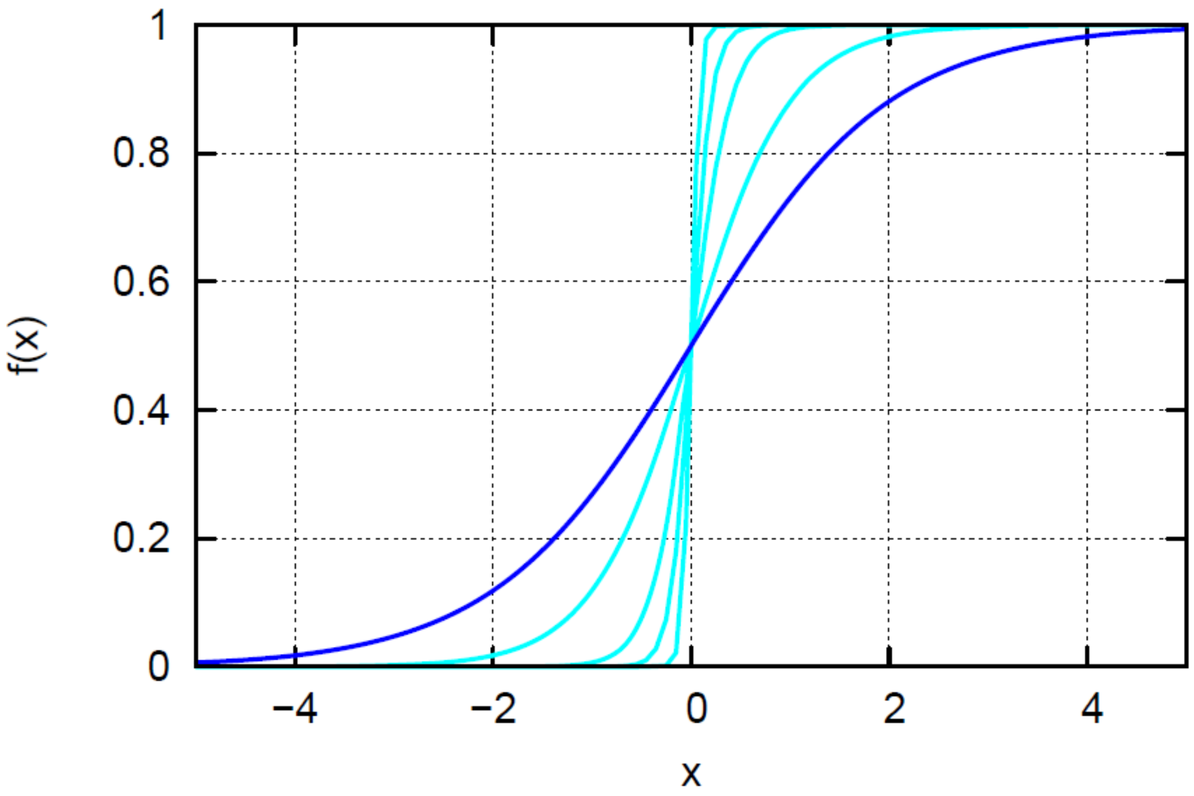
Funções **sigmoides** são geralmente usadas como funções de ativação.

Função logística
(ou função de Fermi)

$$f(x) = \frac{1}{1 + e^{-x/T}}$$

Função tangente
hiperbólica

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$



Fonte: Kriesel (2007, p. 38).

Função de saída

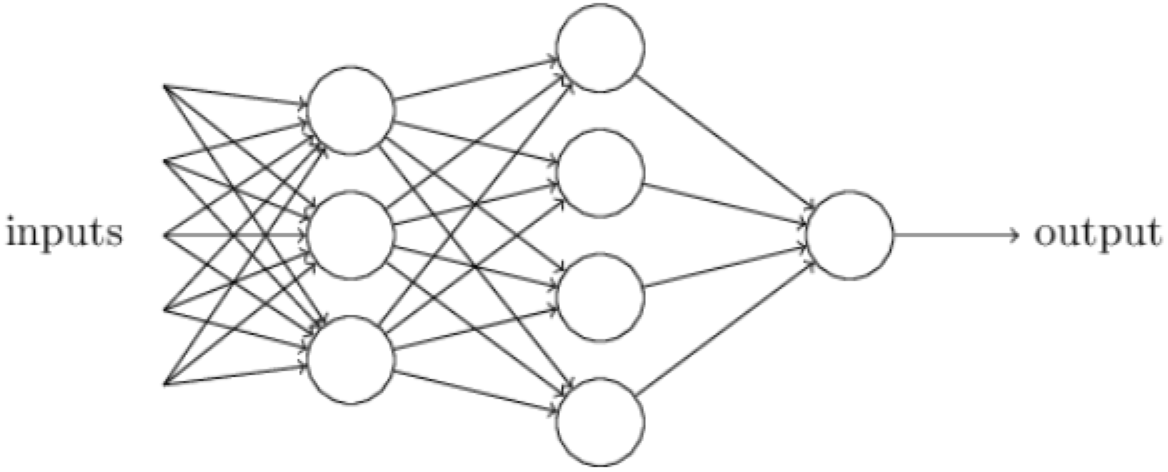
A função de saída, f_{out}, de um neurônio artificial é a **última etapa** do processamento e geralmente é a *identidade*.

$$o_j = f_{out}(a_j) = a_j$$

Na prática, a função de saída como identidade apenas fornece a ativação a_j do neurônio.

Redes neurais artificiais

Zurada (1992, p. 2) diz que uma rede neural artificial é *“uma malha densa de nós de computação e conexões”* que *“operam coletiva e simultaneamente na maioria ou em todos os dados e entradas”*.

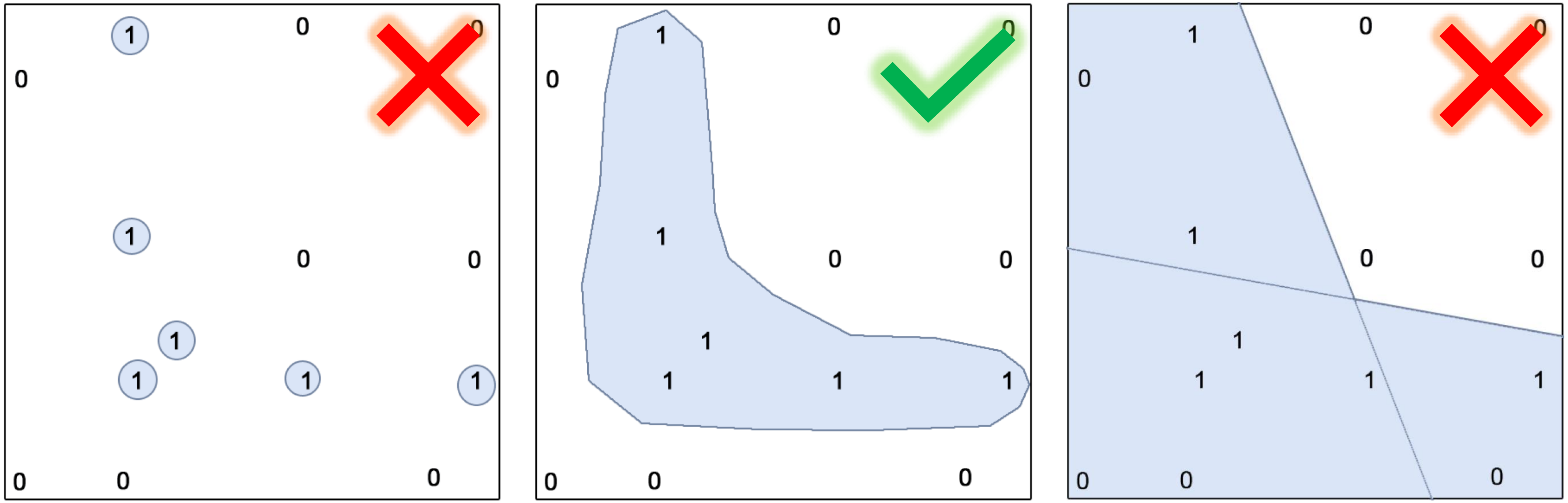


Fonte: Nielsen (2015).

Generalização de uma RNA

Uma boa RNA é capaz de **generalizar**.

Uma certa margem ➪ evita a superespecialização (*overfitting*).



Fonte: Adaptado de Kriesel (2007, p. 56).

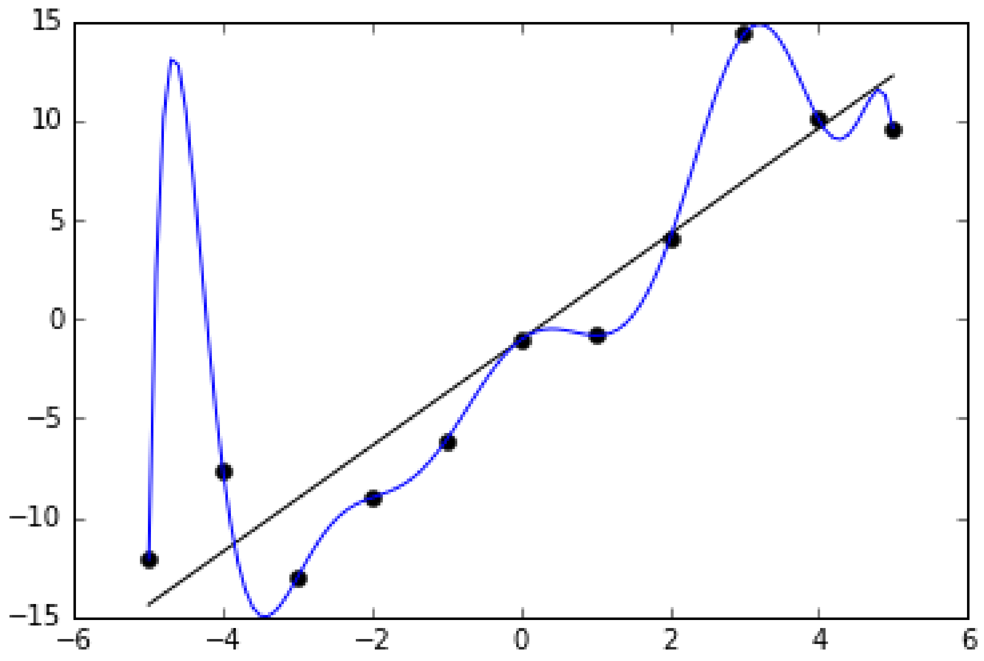
Generalização de uma RNA (cont.)

Ajuste de dados com ruído.

Polinômio ⇨ ajuste perfeito

$$\underbrace{\text{Reta} \Rightarrow \text{melhor generalização}}_{}$$

Melhor capacidade preditiva
(dados desconhecidos)



Fonte: Wikipedia.

Modos de aprendizado das RNAs

- Aprendizado não-supervisionado
- Aprendizado por reforço
- Aprendizado supervisionado

$$P = \{(p,t)_1, \dots, (p,t)_{|P|} \mid p = (p_1, \dots, p_n), t = (t_1, \dots, t_n)\}$$

P : conjunto de treinamento

p : padrão de entrada

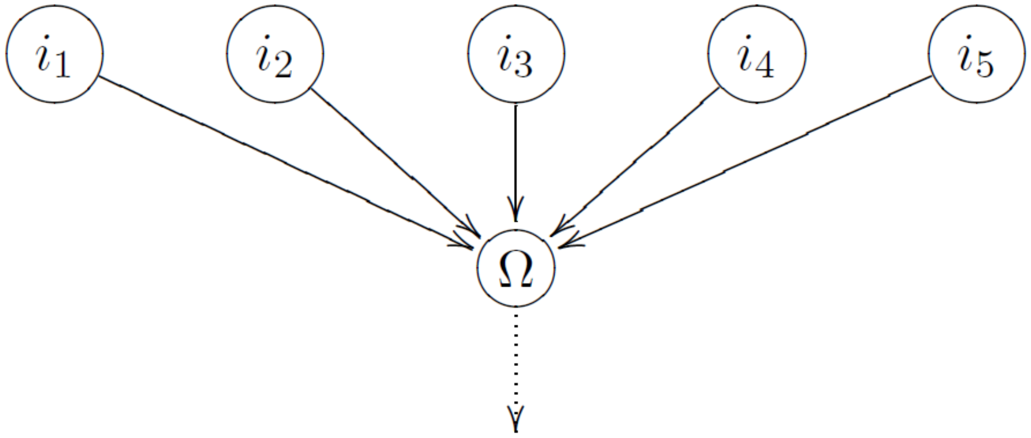
t : alvo

Cada padrão de entrada p tem um alvo t associado.

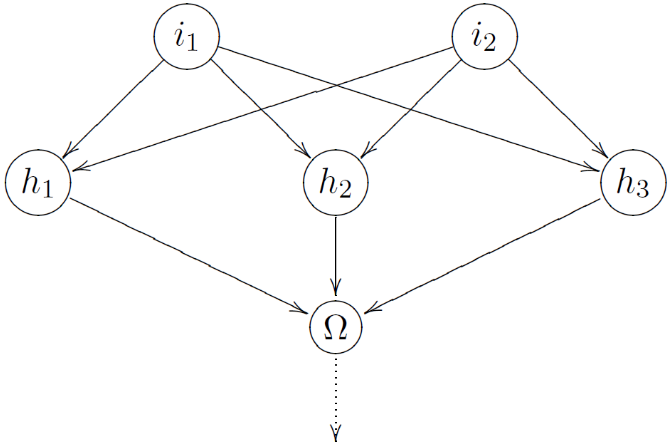
O perceptron

Um perceptron é uma rede neural simples, do tipo *feedforward.*

- Pelo menos uma camada de pesos ajustáveis
- Conexão completa entre camadas
- Primeira camada: camada de entrada (pesos fixos)



Fonte: Kriesel (2007, p. 72).



Fonte: Kriesel (2007, p. 85).

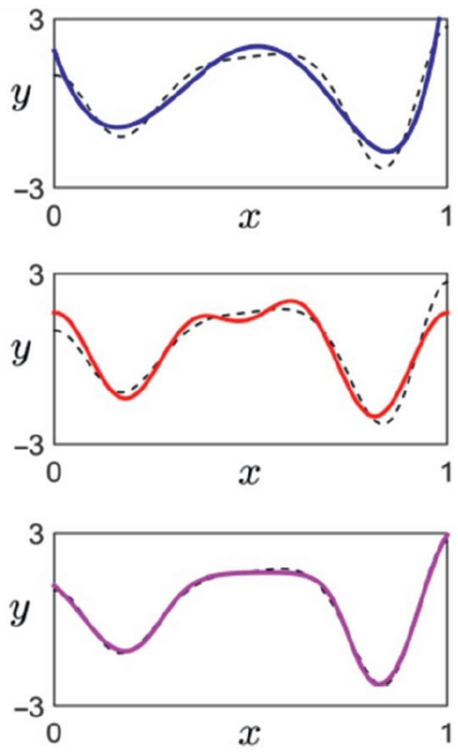
O perceptron (cont.)

Teorema de Cybenko (1989): um perceptron multicamadas é um tipo de **aproximador universal de funções**.

Polinômio

Série de Fourier

Rede Neural Artificial



Fonte: Adaptado de Watt, Borhani e Katsaggelos (2016, p. 138).

A lei de aprendizado de Hebb

Segundo Donald O. Hebb (1949):

> Se dois neurônios artificiais estão fortemente ativos num mesmo instante, então a conexão entre ambos deve se tornar ainda mais significativa.

O peso entre ambos deve ser incrementado ⇨ Rumelhart e McClelland (1986):

$$\Delta w_{i,j} = \eta \cdot h(o_i, w_{i,j}) \cdot g(a_j, t_j)$$

A função erro

Erro ⇨ **diferença** entre os valores fornecidos pela rede e os alvos.

O erro pode ser entendido como uma **função dos pesos** da RNA:

$$Err : W \rightarrow \mathbb{R}$$

Se os pesos variam, o mesmo se dá com os resultados da RNA.

A função erro (cont.)

Erro específico:

$$Err_p(W) = \frac{1}{2}\sum_{\Omega \in O}\left(t_{p,\Omega} - y_{p,\Omega}\right)^2$$

Referente a um padrão de entrada, p

Erro total:

$$Err(W) = \sum_{p \in P} Err_p(W)$$

Referente a todos os padrões de entrada

Combinando...

$$Err(W) = \frac{1}{2}\sum_{p \in P}\sum_{\Omega \in O}\left(t_{p,\Omega} - y_{p,\Omega}\right)^2$$

A Regra Delta

A Regra Delta é um **algoritmo de treinamento (ou de aprendizado)**.

Objetivo ⇨ resposta da rede tender ao alvo.

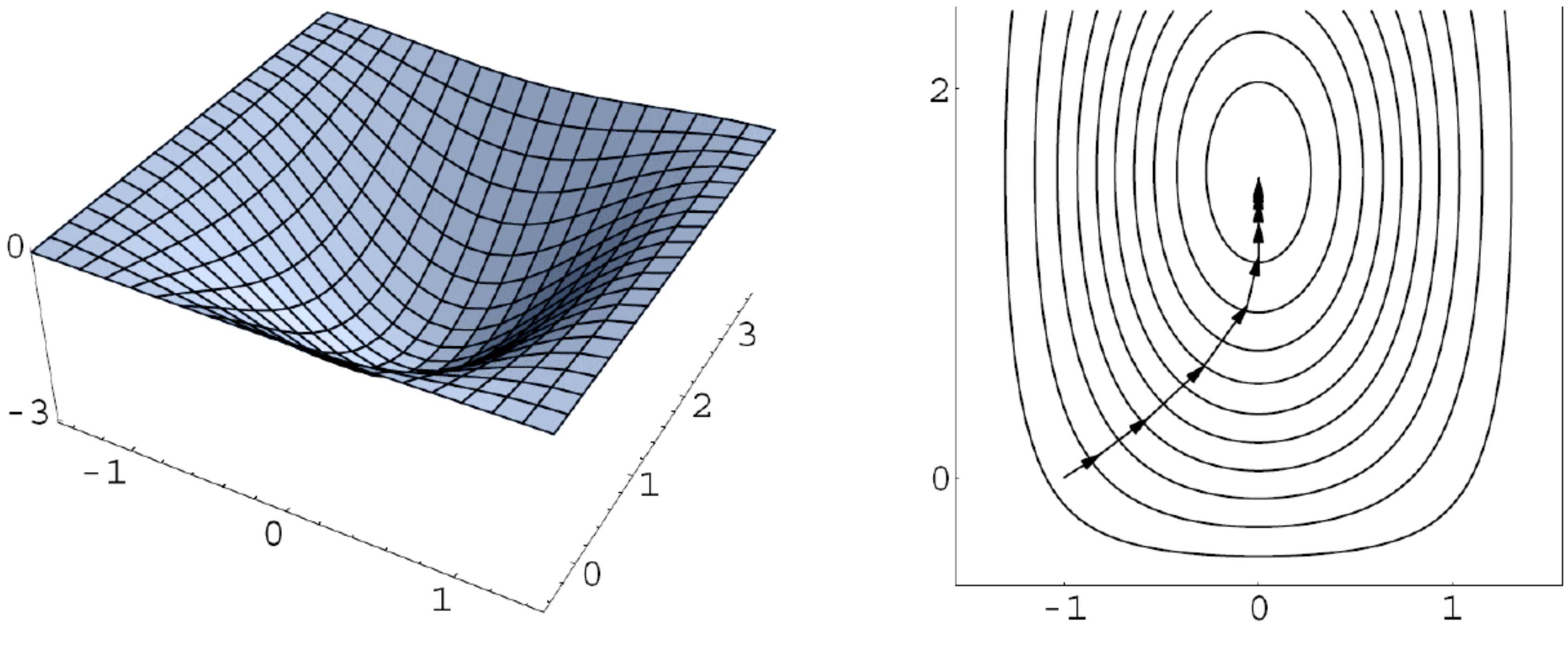
A Regra Delta baseia-se no método do **gradiente descendente** para a minimização da função erro:

$$\Delta W = -\eta \nabla Err(W)$$

A Regra Delta é um **caso específico da Lei de Hebb**.

A Regra Delta (cont.)

$$\Delta W = -\eta \nabla Err(W)$$



Fonte: Kriesel (2007, p. 62).

A Regra Delta

↳ Perceptron de camada única (SLP)

A partir da Regra Delta, Kriesel (2007) fornece a seguinte fórmula para mudança de peso num **perceptron de camada única**:

$$\Delta w_{i,j} = \eta \cdot o_i \cdot (t_\Omega - o_\Omega) = \eta o_i \delta_\Omega$$

η : coeficiente de aprendizado

o_i : saída do neurônio i

t_Ω : saída desejada do neurônio Ω (alvo)

o_Ω : saída efetiva do neurônio Ω

δ_Ω : diferença $t_\Omega - o_\Omega$

A Regra Delta

↳ Perceptron multicamadas (MLP)

A **retropropagação do erro** consiste numa generalização da Regra Delta para um perceptron multicamadas (MLP).

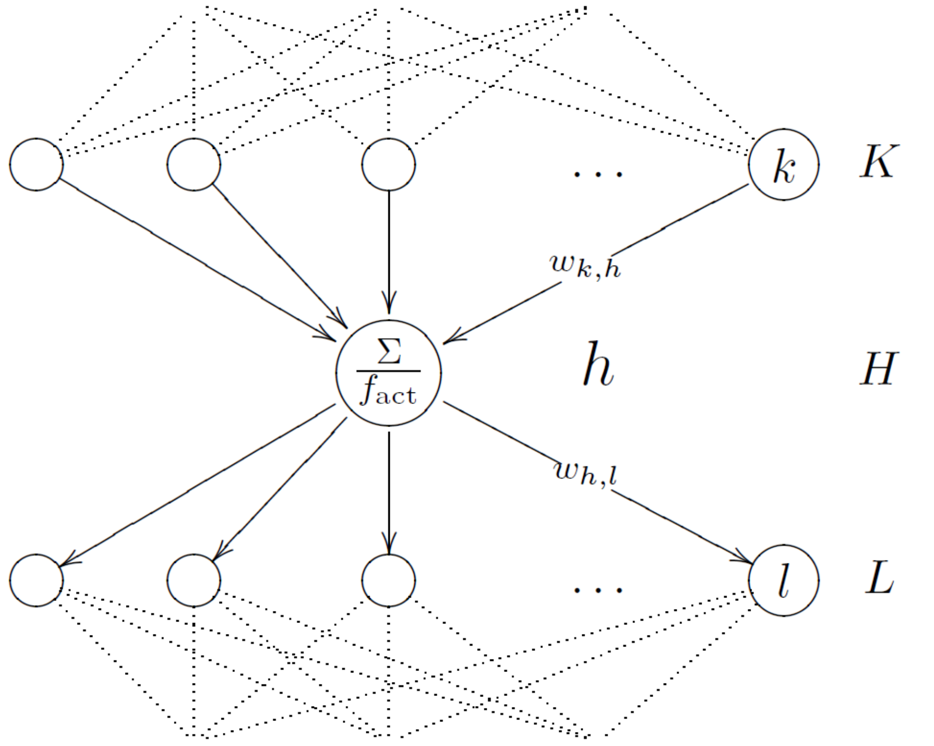
$$\Delta w_{k,h} = \eta o_k \delta_h$$

Se H for a camada de saída:

$$\delta_h = f'_{act}(net_h) \cdot (t_h - y_h)$$

Se H **não** for a camada de saída:

$$\delta_h = f'_{act}(net_h) \cdot \sum_{l \in L} (\delta_l w_{h,l})$$

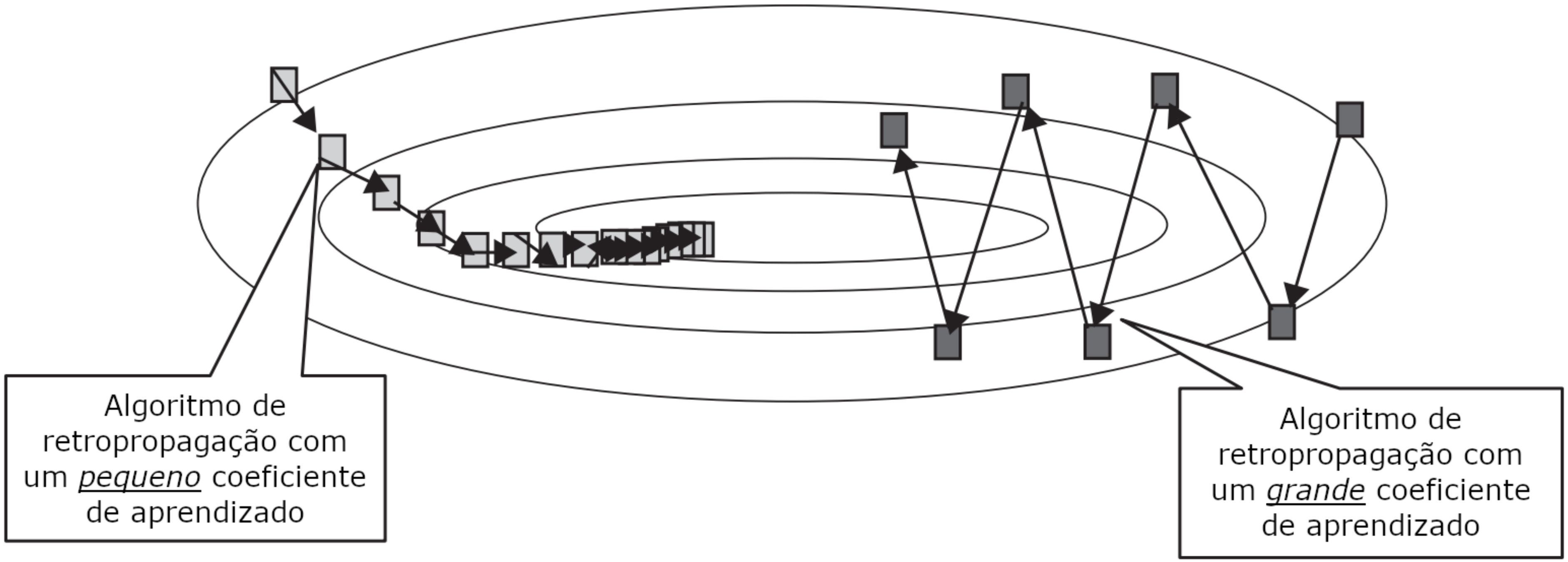


Fonte: Kriesel (2007, p. 87)

O Algoritmo de Levenberg-Marquardt

Regra Delta ⇨ convergência lenta

Método de Gauss-Newton ⇨ necessita de uma superfície quadrática



Fonte: Adaptado de Wilamowski e Irwin (2016, p. 2).

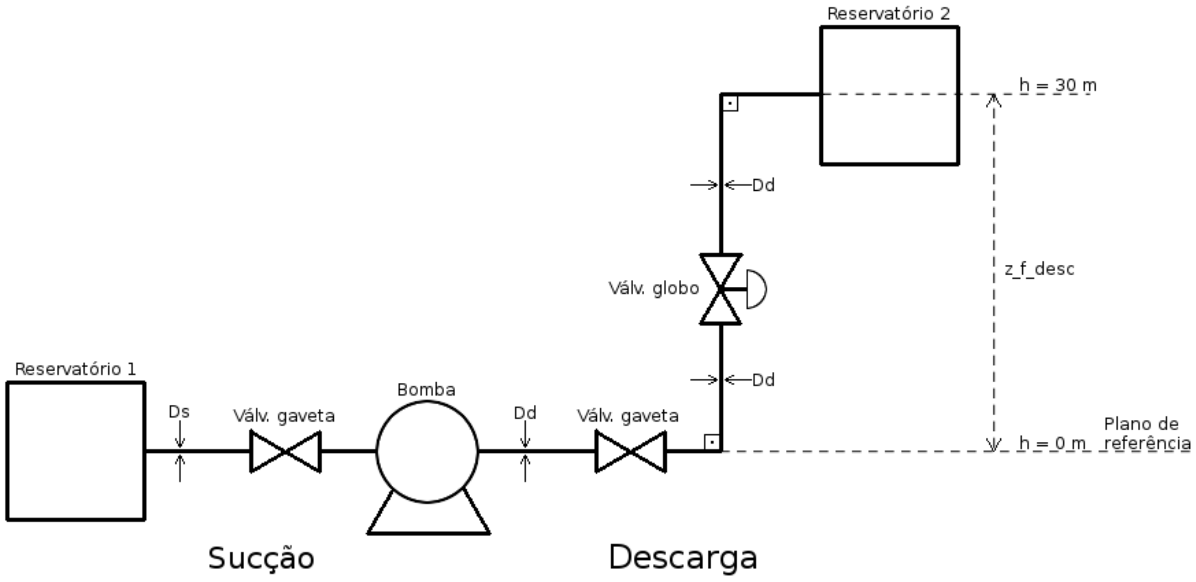
O Algoritmo de Levenberg-Marquardt (cont.)

Algoritmo de Levenberg-Marquardt

- Gradiente descendente
- Algoritmo de Gauss-Newton

$$w_{k+1} = w_k - \left(\boldsymbol{J}_k^T \boldsymbol{J}_k + \mu \boldsymbol{I}\right)^{-1} \boldsymbol{J}_k \boldsymbol{e}_k$$

Contextualização



Fonte: Elaborado pelo autor.

Variáveis e constantes físicas.

Parâmetro	Tipo	Nome	Unidade
Vazão volumétrica	Variável	Q_vazao	m^3/h
Pressão no final da descarga	Variável	pfd	kPa
Diâmetro da tubulação de descarga	Variável	Dd	m
Comprimento da tubulação de descarga	Variável	Ld	m
Pressão no início da sucção	Variável	p0s	kPa
Diâmetro da tubulação de sucção	Variável	Ds	m
Comprimento da tubulação de sucção	Variável	Ls	m
Constante gravitacional	Constante	g	m/s^2
Massa específica da água	Constante	rho	kg/m^3
Viscosidade dinâmica da água a 20 °C	Constante	mu	$Pa.s$
Altura final da descarga	Constante	z_f_desc	m
Rugosidade da tubulação de descarga	Constante	e_desc	mm
Altura inicial da sucção	Constante	z_0_suc	m
Rugosidade da tubulação de sucção	Constante	e_suc	mm

Fonte: Elaborado pelo autor.

Contextualização (cont.)

Valores das constantes.

Constante	Nome	Valor	Unidade
Constante gravitacional	g	9,81	m/s^2
Massa específica da água	rho	1000	kg/m^3
Viscosidade dinâmica da água a 20 °C	mu	0,001	$Pa.s$
Altura final da descarga	z_f_desc	30	m
Rugosidade da tubulação de descarga	e_desc	0,1	mm
Altura inicial da sucção	z_0_suc	0	m
Rugosidade da tubulação de sucção	e_suc	0,1	mm

Fonte: Elaborado pelo autor.

Faixas de valores das variáveis.

Variável	Nome	Faixa	Unidade
Vazão volumétrica	Q_vazao	0 - 100	m^3/h
Pressão no final da descarga	pfd	0 - 101,325	kPa
Diâmetro da tubulação de descarga	Dd	0,1 - 0,3	m
Comprimento da tubulação de descarga	Ld	30 - 60	m
Pressão no início da sucção	p0s	0 - 101,325	kPa
Diâmetro da tubulação de sucção	Ds	0,1 - 0,3	m
Comprimento da tubulação de sucção	Ls	0 - 60	m

Fonte: Elaborado pelo autor.

Rugosidade absoluta: $\varepsilon = 0{,}1\ mm$

Núm. Reynolds crítico: $\mathrm{Re}_{\mathrm{cr}} = 2.300$

Comprimento de entrada: $L \geq 10D$

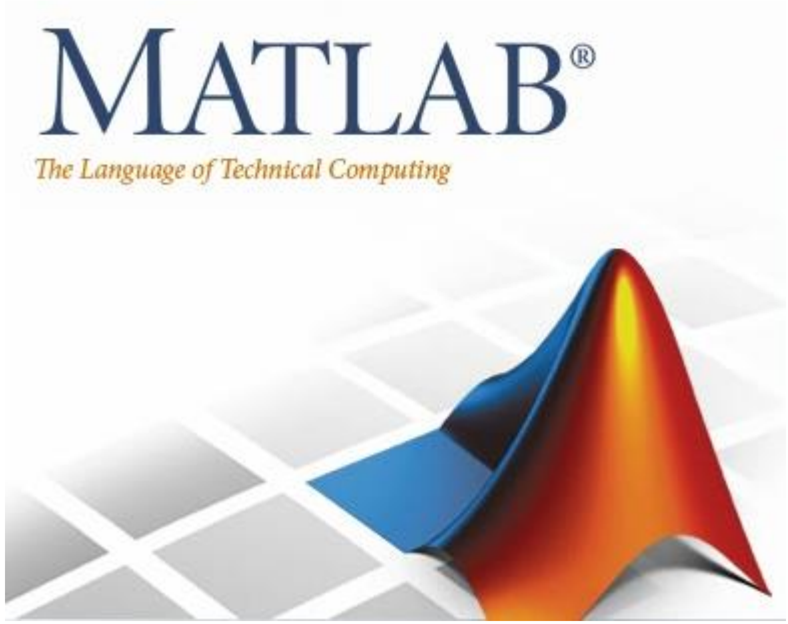
Valores de K para as válvulas globo e gaveta.

Diâmetro (mm)	25	50	100	200	500
Globo	13,00	8,50	6,00	5,80	5,50
Gaveta	0,80	0,35	0,16	0,07	0,03

Fonte: Adaptado de White (2010, p. 397).

Metodologia

- Geração dos padrões de entrada
- Cálculo dos alvos
- Treinamento da RNA
- Comparação
  - Integral
  - Pontual



Fonte: https://medium.com/quick-code/top-tutorials-to-learn-matlab-for-beginners-d19549ecb7b7

Metodologia

↳ Geração dos padrões de entrada

1ª linha	Vazão	Q_vazao
2ª linha	Pressão no final da descarga	pfd
3ª linha	Diâmetro da tubulação de descarga	Dd
4ª linha	Comprimento da tubulação de descarga	Ld
5ª linha	Pressão no início da sucção	p0s
6ª linha	Diâmetro da tubulação de sucção	Ds
7ª linha	Comprimento da tubulação de sucção	Ls

matrizEntradas

7x233280 double

	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
4	30	30	30	30	30	30	30	30	30	30	30
5	0	0	0	0	0	0	0	0	0	0	0
6	0.1000	0.1000	0.1000	0.1000	0.1000	0.1400	0.1400	0.1400	0.1400	0.1400	0.1800
7	12	24	36	48	60	12	24	36	48	60	12

Metodologia

↳ Geração dos padrões de entrada (cont.)

matrizEntradas

7x233280 double

	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
4	30	30	30	30	30	30	30	30	30	30	30
5	0	0	0	0	0	0	0	0	0	0	0
6	0.1000	0.1000	0.1000	0.1000	0.1000	0.1400	0.1400	0.1400	0.1400	0.1400	0.1800
7	12	24	36	48	60	12	24	36	48	60	12

matrizEntradas

7x233280 double

	1	2	3	4	5	6	7	8	9	10	11
1	100	60	20	100	80	60	20	20	80	0	0
2	101.3250	101.3250	20.2650	60.7950	20.2650	60.7950	81.0600	81.0600	81.0600	40.5300	101.3250
3	0.2200	0.1800	0.3000	0.3000	0.2200	0.1800	0.1400	0.3000	0.2200	0.2200	0.2200
4	42	60	36	60	36	36	54	42	54	36	30
5	60.7950	20.2650	101.3250	101.3250	60.7950	60.7950	0	0	60.7950	81.0600	101.3250
6	0.1000	0.2200	0.2200	0.2200	0.2200	0.2600	0.1800	0.1400	0.1800	0.1000	0.1800
7	12	36	60	12	48	36	12	12	24	36	60

233.280 colunas

Metodologia

↳ Cálculo dos alvos

```
% Criação da matrizAlvos, com 1 linha e um número indefinido de colunas.
matrizAlvos = double.empty(1,0);

% O loop abaixo é usado para calcular o head usando-se e cada uma das
% colunas da matrizEntradas como parâmetro de entrada da função
% fn_head_sistema .
for j = 1:tamAmostra
    matrizAlvos(1,j) = fn_head_sistema(matrizEntradas(:,j));
end
```

matrizAlvos

1x233280 double

	1	2	3	4	5	6	7	8	9	10	11
1	35.6521	38.5988	21.7467	25.9655	26.1146	30.2620	38.3751	38.2757	32.3535	25.8685	30

Metodologia

↳ Normalização das entradas e dos alvos

matrizEntradasNorm

7x233280 double

	1	2	3	4	5	6	7	8	9	10	11
1	1	0.2000	-0.6000	1	0.6000	0.2000	-0.6000	-0.6000	0.6000	-1	-1
2	1	1	-0.6000	0.2000	-0.6000	0.2000	0.6000	0.6000	0.6000	-0.2000	1
3	0.2000	-0.2000	1	1	0.2000	-0.2000	-0.6000	1	0.2000	0.2000	0.2000
4	-0.2000	1	-0.6000	1	-0.6000	-0.6000	0.6000	-0.2000	0.6000	-0.6000	-1
5	0.2000	-0.6000	1	1	0.2000	0.2000	-1	-1	0.2000	0.6000	1
6	-1	0.2000	0.2000	0.2000	0.2000	0.6000	-0.2000	-0.6000	-0.2000	-1	-0.2000
7	-1	0	1	-1	0.5000	0	-1	-1	-0.5000	0	1

matrizAlvosNorm

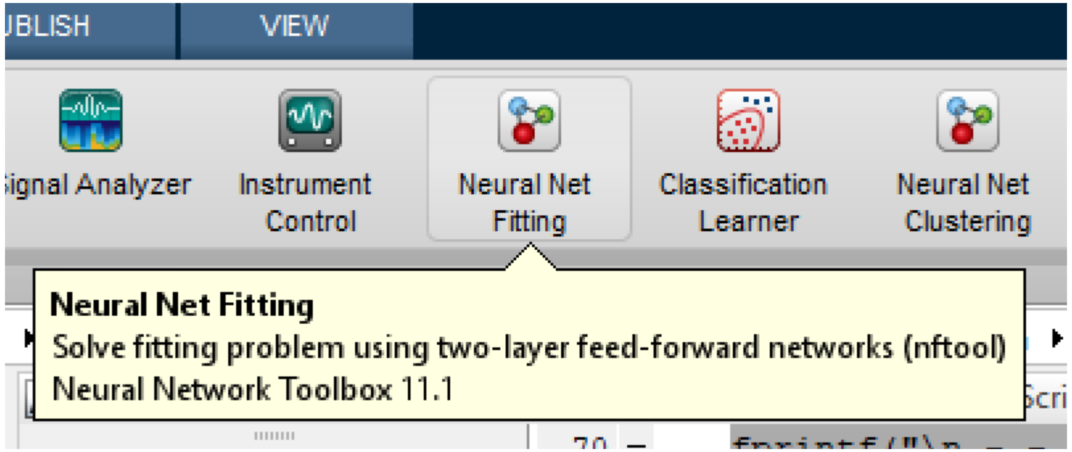
1x233280 double

	1	2	3	4	5	6	7	8	9	10	11
1	-0.2189	-0.0749	-0.8986	-0.6924	-0.6851	-0.4823	-0.0858	-0.0907	-0.3801	-0.6971	-0.4952

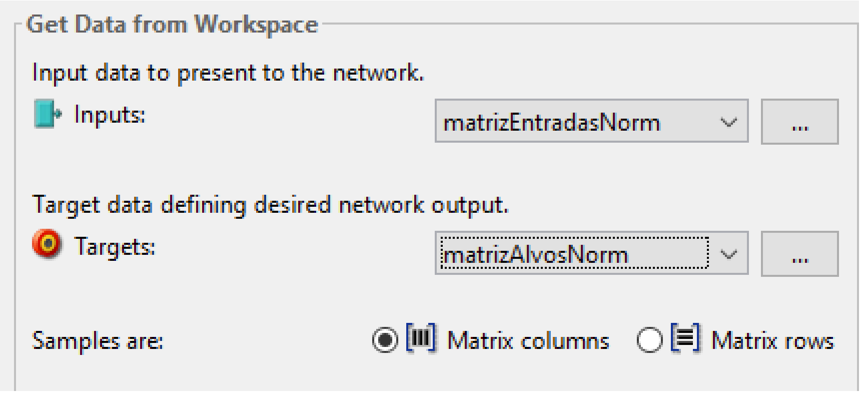
Metodologia

↳ Treinamento da RNA no MATLAB

1. Ferramenta “*Neural Net Fitting*” do MATLAB:



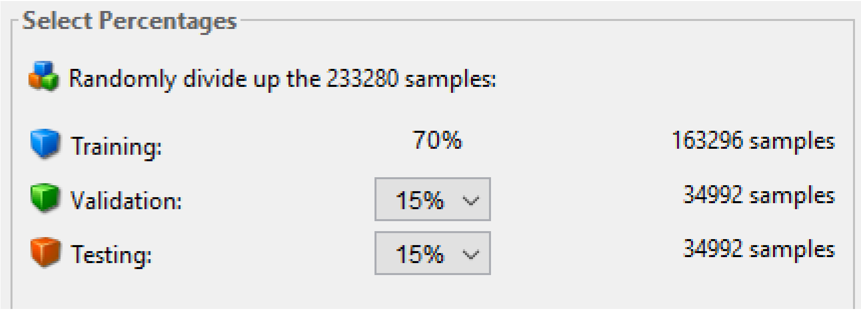
2. Seleção dos inputs e targets:



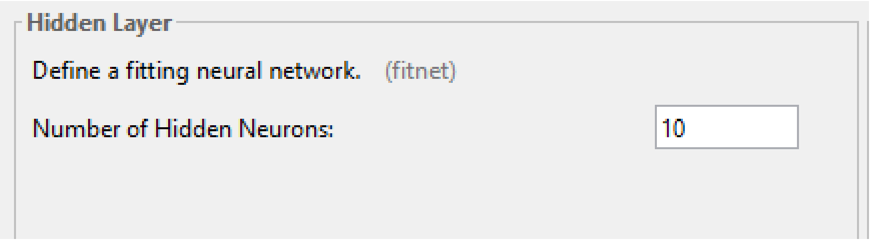
Metodologia

↳ Treinamento da RNA no MATLAB (cont.)

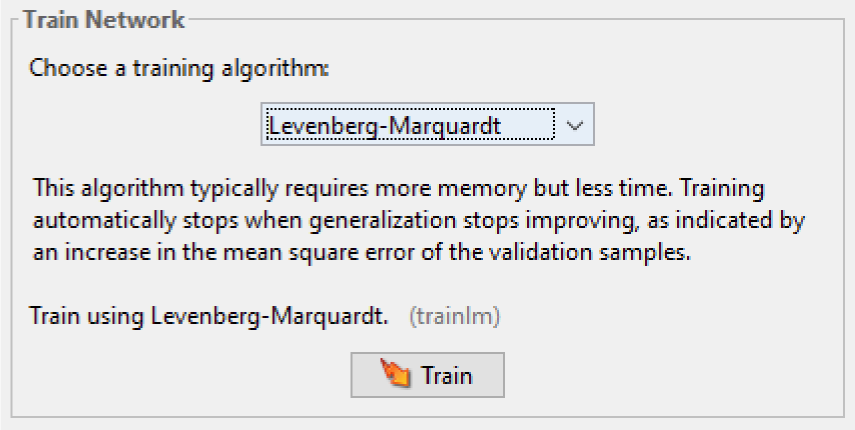
3. Divisão do conjunto dos padrões de entrada:



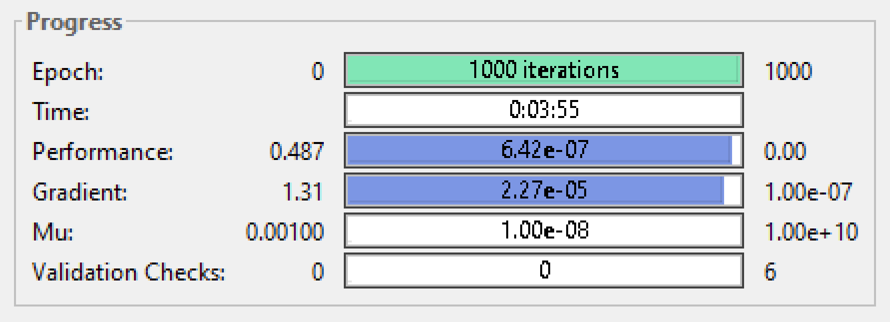
4. Escolha do número de neurônios da camada oculta:



5. Escolha do algoritmo de treinamento:



6. Progresso de treinamento da rede neural artificial:



Metodologia

↳ Treinamento da RNA no MATLAB (cont.)

IW1_1

7x10 double

	1	2	3	4	5	6	7	8	9	10
1	-1.0079	-0.5553	-0.6548	-1.7370	-0.2081	-0.0273	-0.9396	-0.4765	-0.5893	0.8072
2	-1.6075e-04	-1.2118e-05	-5.8891e-04	-0.0063	0.0020	0.0153	0.0058	2.9554e-05	-7.1263e-04	1.0699e-04
3	-8.5609e-04	2.1905	-0.0023	0.0069	-0.1239	-0.0166	-0.7318	2.2745	-0.0029	-2.3679
4	-0.0017	-0.0293	-0.0031	-0.0302	-0.0040	-5.0465e-04	-0.0151	-0.0356	-0.0038	-0.0113
5	2.0707e-04	1.5521e-04	7.1074e-04	0.0066	-0.0021	-0.0154	-0.0061	1.2922e-04	8.6533e-04	-3.3006e-04
6	3.0866	4.5568e-04	3.3740	2.1707	0.0020	2.1751e-04	-0.0038	-8.6042e-05	3.4774	-0.0017
7	0.2168	-2.8639e-04	-0.2274	-0.6727	-0.0020	-2.2624e-04	0.0012	2.2173e-04	-0.2441	0.0013

Pesos das conexões da camada de entrada para a camada oculta.

b1

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	4.6021	3.2736	4.5248	2.2350	0.1945	0.0091	-1.1734	2.9726	4.5657	-3.7312

Vieses da camada oculta.

LW2_1

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	0.3364	-7.6986	-2.8662	-0.0077	-2.2031	16.7454	-0.0709	2.8932	2.1817	-2.4106

Pesos das conexões da camada de entrada para a camada oculta.

$$\boldsymbol{b2 = 2,4699}$$

Metodologia

↳ Treinamento da RNA no MATLAB (cont.)

O MATLAB produz uma função que simula a rede neural artificial:

RNA_head_sistema

RNA_head_sistema recebe como parâmetro um vetor da mesma dimensão do vetor usado no treinamento.

Metodologia

↳ Avaliação da RNA no MATLAB

Obtenção de valores de parâmetros:

matrizEntradas

7x17715610 double

	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
4	30	30	30	30	30	30	30	30	30	30	30
5	0	0	0	0	0	0	0	0	0	0	0
6	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1200
7	6	12	18	24	30	36	42	48	54	60	6

matrizEntradas

7x17715610 double

	1	2	3	4	5	6	7	8	9	10	11
1	100	60	100	10	60	30	0	0	0	80	60
2	10.1325	101.3250	10.1325	101.3250	101.3250	30.3975	60.7950	60.7950	0	10.1325	91.1925
3	0.2600	0.1400	0.2200	0.1400	0.2600	0.1400	0.2800	0.1400	0.1400	0.1800	0.2000
4	48	33	54	45	48	51	48	60	54	54	51
5	81.0600	101.3250	10.1325	0	101.3250	10.1325	10.1325	0	10.1325	40.5300	30.3975
6	0.2200	0.2200	0.1800	0.1200	0.3000	0.2400	0.3000	0.2000	0.1200	0.1000	0.2000
7	42	42	42	6	6	54	30	24	6	60	48

17.715.610 colunas

Metodologia

↳ Avaliação da RNA no MATLAB (cont.)

Obtenção dos valores de *head*:

```
% Criação da matriz que armazenará os valores de saída da expressão
% fenomenológica.
matrizHeads_fenom = double.empty(num_param,0);

% O loop abaixo é usado para calcular o head usando-se cada um dos
% parâmetros da matrizEntradas.
for j = 1:1:tamAmostra
    matrizHeads_fenom(1,j) = fn_head_sistema(matrizEntradas(:,j));
end
```

matrizHeads_fenom

1x17715610 double

	1	2	3	4	5	6	7	8	9	10
1	23.0011	30.7718	30.5592	40.3575	30.0548	32.3062	35.1644	36.1972	28.9671	32.2561

Metodologia

↳ Avaliação da RNA no MATLAB (cont.)

Cálculos dos valores de saída da rede neural artificial:

matrizHeads_RNA

1x17715610 double

	1	2	3	4	5	6	7	8	9	10
1	-0.8366	-0.4563	-0.4673	0.0108	-0.4923	-0.3827	-0.2423	-0.1916	-0.5443	-0.3845

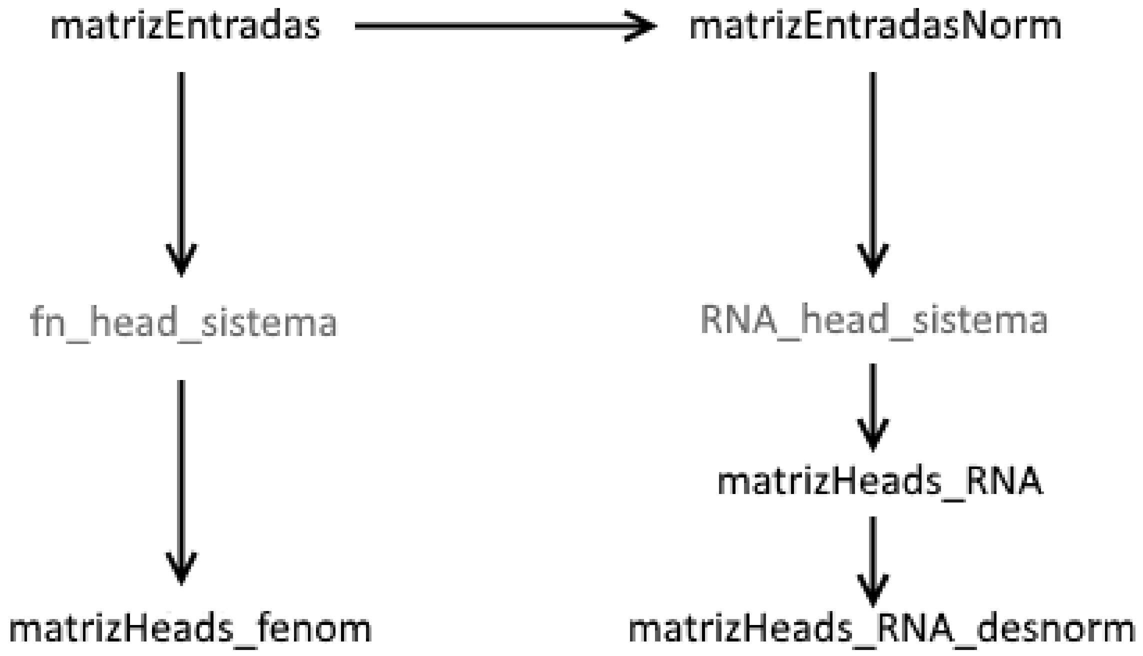
matrizHeads_RNA_desnorm

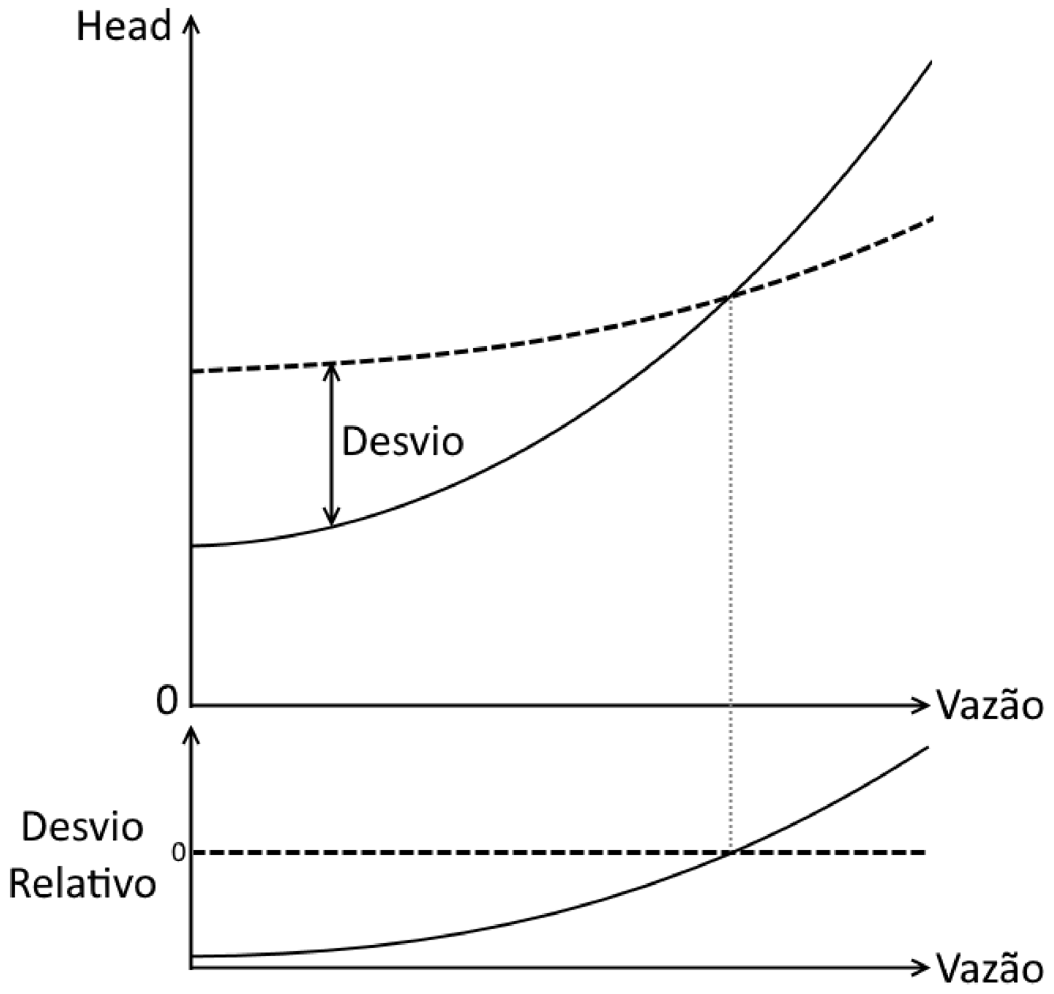
1x17715610 double

	1	2	3	4	5	6	7	8	9	10
1	23.0142	30.7943	30.5692	40.3525	30.0580	32.3001	35.1732	36.2098	28.9950	32.2633

Metodologia

↳ Avaliação da RNA no MATLAB (cont.)





Metodologia

↳ Avaliação da RNA no MATLAB (cont.)

- Avaliação ao longo de todo o conjunto de amostras:

$$desvRel = \frac{|valor - correto|}{correto} \cdot 100\%$$

- Avaliação em conjuntos específicos de valores de parâmetros:

  Graficamente (*head* versus vazão)

$$desvRelPosNeg = \frac{valor - correto}{correto} \cdot 100\%$$

Resultados

matrizHeads_fenom

1x17715610 double

	1	2	3	4	5	6	7	8	9	10
1	23.0011	30.7718	30.5592	40.3575	30.0548	32.3062	35.1644	36.1972	28.9671	32.2561

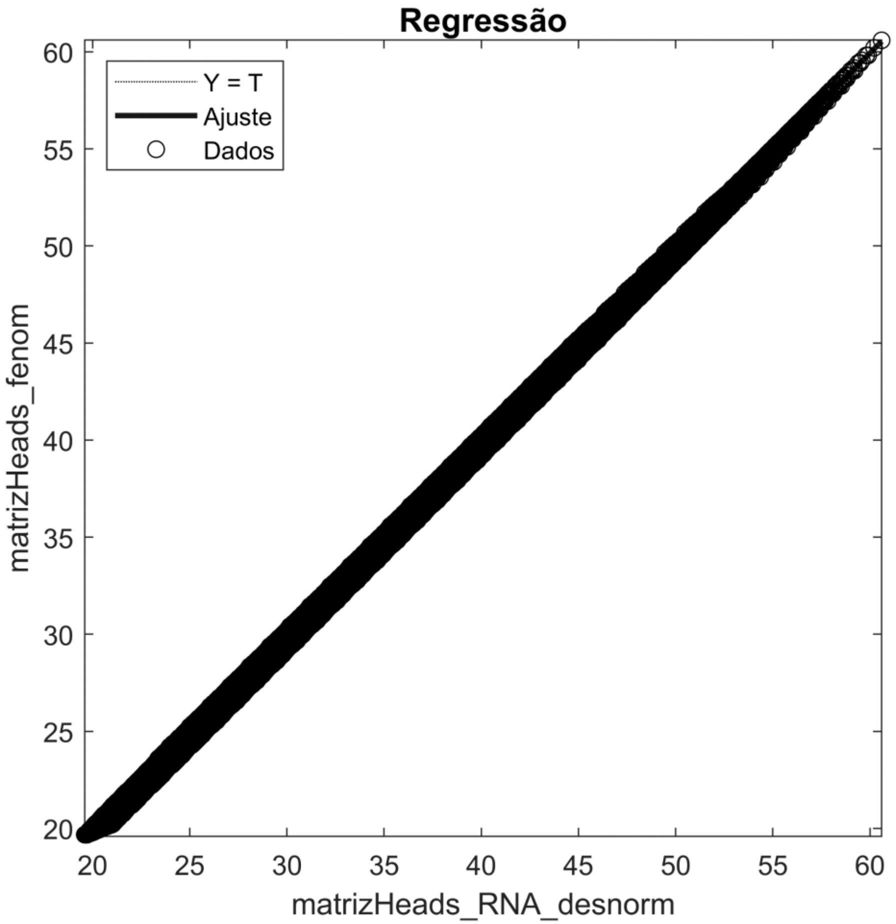
matrizHeads_RNA_desnorm

1x17715610 double

	1	2	3	4	5	6	7	8	9	10
1	23.0142	30.7943	30.5692	40.3525	30.0580	32.3001	35.1732	36.2098	28.9950	32.2633

Resultados (cont.)

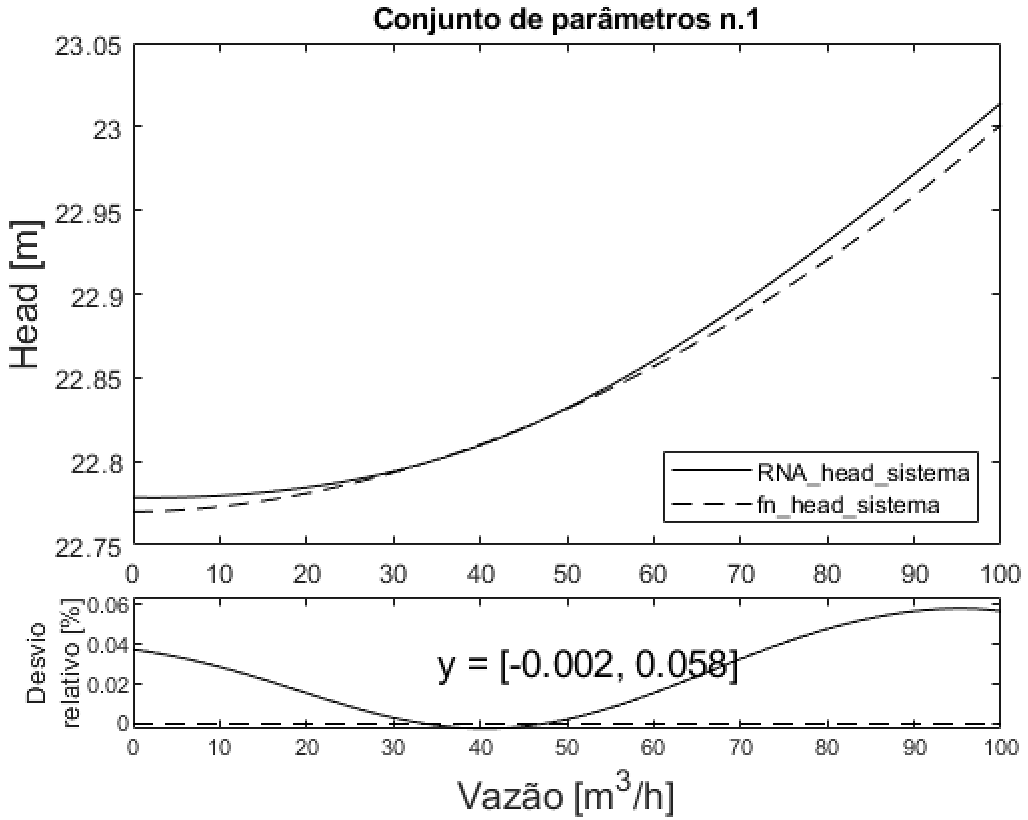
Faixa (%)	Ocorrências	Frequência (%)	Acúmulo (%)
[0 ; 0,1)	13.077.927	73,82	73,82
[0,1 ; 0,2)	2.300.386	12,99	86,81
[0,2 ; 0,3)	863.463	4,87	91,68
[0,3 ; 0,4)	375.649	2,12	93,80
[0,4 ; 0,5)	217.463	1,23	95,03
[0,5 ; 0,6)	171.304	0,97	96,00
[0,6 ; 0,7)	154.379	0,87	96,87
[0,7 ; 0,8)	117.276	0,66	97,53
[0,8 ; 0,9)	85.691	0,48	98,01
[0,9 ; 1,0)	66.312	0,37	98,39
[1,0 ; 1,5)	179.400	1,01	99,40
[1,5 ; 2,0)	67.788	0,38	99,78
[2,0; 4,0]	38.572	0,22	100,00
[0 ; 4,0]	**17.715.610**	**100,00**	**100,00**

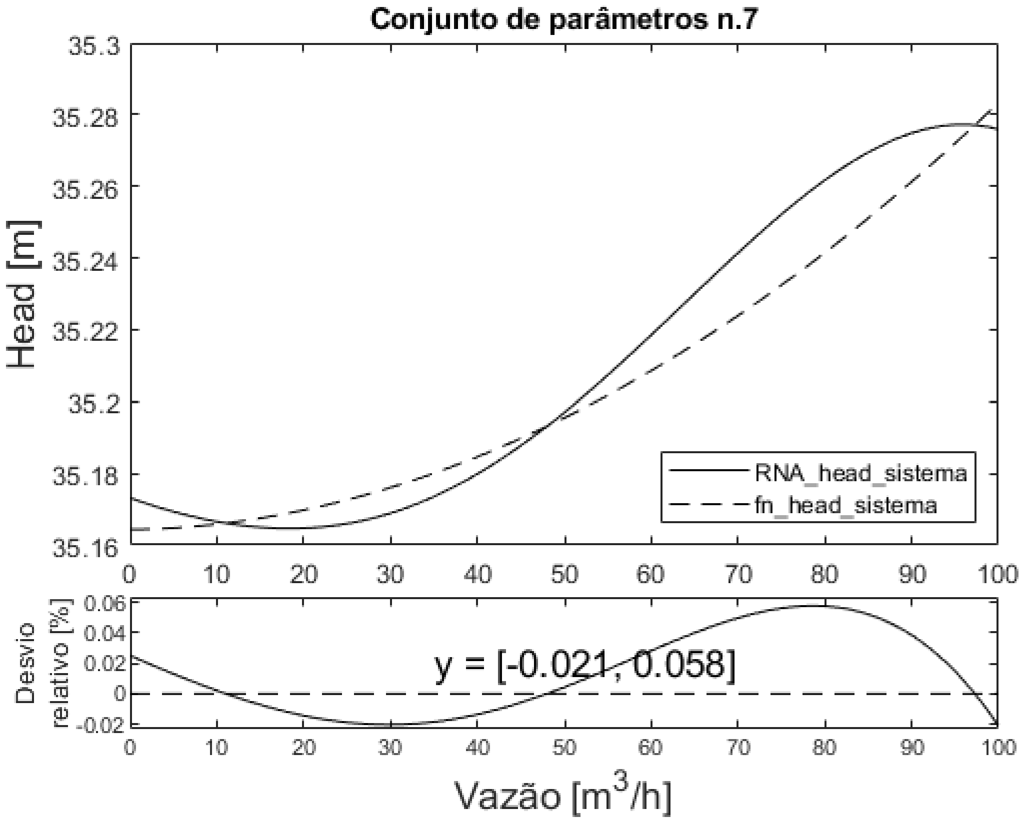


	(%)
Média	0,1161
Desvio padrão	0,2336
Mínimo	$1{,}08 \times 10^{-9}$
Máximo	4,00

Resultados (cont.)

Conj.	1	7
pfd (kPa)	10,1325	60,795
Dd (m)	0,26	0,28
Ld (m)	48	48
p0s (kPa)	81,06	10,1325
Ds (m)	0,22	0,3
Ls (m)	42	30





Conclusão

- A modelagem do sistema hidráulico por meio de uma RNA é possível.
- O treinamento com 163.296 amostras foi suficiente para a RNA computar o head de 17.715.610 conjuntos de valores de parâmetros.
- 98,39% dos resultados situaram-se abaixo de 1,0% de desvio relativo.
- 73,82% dos resultados situaram-se abaixo de 0,1% de desvio relativo.
- Em nenhuma ocorrência o desvio superou os 4,0%.

Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *ABNT 1977* : Elaboração de Projetos de Sistemas de Adução de água para Abastecimento Público. Rio de Janeiro, 1977.

BRUNETTI, F. *Mecânica dos fluidos*. 2. ed. São Paulo: Pearson Prentice Hall, 2008.

COLEBROOK, C. F. Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws. *Journal of the Institution of Civil Engineers*, v. 11, n. 4, p. 133–156, 1939. Disponível em: <https://doi.org/10.1680/ijoti.1939.13150>. Acesso em: 23 out. 2018.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, Springer London, v. 2, n. 4, p. 303–314, dec 1989. Disponível em: <http://dx.doi.org/10.1007/BF02551274>. Acesso em: 23 out. 2018.

ÇENGEL, Y. A.; CIMBALA, J. M. *Mecânica dos fluidos: fundamentos e aplicações*. 3. ed. Porto Alegre: AMGH, 2012.

FOX, R.; PRITCHARD, P.; MCDONALD, A. *Fox and McDonald's Introduction to Fluid Mechanics*. 8. ed. Hoboken, NJ, US: John Wiley & Sons, Inc., 2011.

Referências (cont.)

GENIC, S. et al. A review of explicit approximations of Colebrook's equation. *FME Transactions*, v. 39, n. 2, p. 67–71, 2011. Disponível em: <https://www.mas.bg.ac.rs/_media/istrazivanje/fme/vol39/2/04_mjaric.pdf>. Acesso em: 06 nov. 2018.

HAALAND, S. E. Simple and explicit formulas for the friction factor in turbulent pipe flow. *Journal of Fluids Engineering*, ASME, v. 105, n. 1, p. 89–90, 1983. Disponível em: <http://dx.doi.org/10.1115/1.3240948>. Acesso em: 23 out. 2018.

HALLIDAY, D.; RESNICK, R.; WALKER, J. *Fundamentos de física, volume 2: gravitação, ondas e termodinâmica*. 8. ed. Rio de Janeiro: LTC, 2009. v. 2.

HEBB, D. O. *The organization of behavior: A neuropsychological theory*. New York: Wiley, 1949.

HENN, E. *Máquinas de Fluido*. 2. ed. Santa Maria: Editora da UFSM, 2006.

KELLNER, E.; AKUTSU, J.; REIS, L. Avaliação da rugosidade relativa dos tubos de PVC com vistas ao dimensionamento das redes de distribuição de água. v. 21, 04 2016. Disponível em: <http://www.scielo.br/pdf/esa/v21n2/1809-4457-esa-S1413_41522016141081.pdf>. Acesso em: 06 nov. 2018.

Referências (cont.)

KRIESEL, D. *A Brief Introduction to Neural Networks*. [s.n.], 2007. Disponível em: <http://www.dkriesel.com>. Acesso em: 06 nov. 2018.

LEITHOLD, L. *O Cálculo com Geometria Analítica*. São Paulo: Harbra, 1994.

LEVENBERG, K. A method for the solution of certain non-linear problems in least squares. *Quaterly Journal on Applied Mathematics*, n. 2, p. 164–168, 1944. Disponível em: <https://cs.uwaterloo.ca/~y328yu/classics/levenberg.pdf>. Acesso em: 06 nov. 2018.

MACINTYRE, A. *Bombas e instalações de bombeamento*. Rio de Janeiro: Livros Técnicos e Científicos, 1997.

MARQUARDT, D. W. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, SIAM, v. 11, n. 2, p. 431–441, 1963. Disponível em: <http://dx.doi.org/10.1137/0111030>. Acesso em: 06 nov. 2018.

MARQUES, J.; SOUSA, J. de O. *Hidráulica Urbana. Sistemas de Abastecimento de Água e de Drenagem de Águas Residuais*. 3. ed. Coimbra: Universidade de Coimbra, 2011.

Referências (cont.)

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, v. 5, n. 4, p. 115–133, Dec 1943. ISSN 1522-9602. Disponível em: <https://doi.org/10.1007/BF02478259>. Acesso em: 27 out. 2018.

MOODY, L. Friction factors for pipe flow. *Trans. ASME*, v. 66, n. 8, p. 671–677, 1944.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *SP 330* : NIST Special Publication 330, 2008 Edition. Washington, 2008.

NETTO, J.; ALVAREZ, G. *Manual de hidráulica*. 7. ed. São Paulo: E. Blücher, 1982.

NIELSEN, M. A. *Neural Networks and Deep Learning*. [S.l.]: Determination Press, 2015.

NUSSENZVEIG, H. *Curso de Física Básica, 1: Mecânica*. 4. ed. São Paulo: Edgard Blücher, 2002.

PORTO, R. *Hidráulica Básica*. 2. ed. São Carlos, SP: EESC, 2000.

Referências (cont.)

PRESS, W. H. et al. *Numerical Recipes in C: The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press, 1988.

REYNOLDS, O. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Philosophical Transactions of the Royal Society of London*, The Royal Society, v. 174, p. 935–982, 1883. Disponível em: <http://www.jstor.org/stable/109431>. Acesso em: 25 out. 2018.

REYNOLDS, O. *Papers on Mechanical and Physical Subjects: The sub-mechanics of the universe*. [S.l.]: Cambridge University Press, 1903.

ROJAS, R. *Neural Networks - A Systematic Introduction*. 1. ed. Berlin: Springer-Verlag, 1996.

ROSENBLATT, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. *Psychological Review*, p. 65–386, 1958. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.588.3775>. Acesso em: 27 out. 2018.

Referências (cont.)

ROSSMAN, L. *EPANET 2 - Users Manual*. Washington, D.C., 2000.

RUMELHART, D. E.; MCCLELLAND, J. L. (Ed.). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986.

RUMELHART, D. E.; WIDROW, B.; LEHR, M. The Basic Ideas in Neural Networks. *Commun. ACM*, v. 37, p. 87–92, 03 1994. Disponível em: <http://doi.acm.org/10.1145/175247.175256>. Acesso em: 06 nov. 2018.

STEWART, J. *Cálculo, volume I - Tradução da 6a edição norte-americana.* São Paulo: Cengage Learning Edições Ltda., 2010. v. 1.

SWINGLER, K. *Applying Neural Networks: A Practical Guide*. San Francisco: Academic Press, 1996.

TIPLER, P. A.; MOSCA, G. *Física para cientistas e engenheiros, volume 1: mecânica, oscilações e ondas, termodinâmica*. 6. ed. Rio de Janeiro: LTC, 2009. v. 1.

WATT, J.; BORHANI, R.; KATSAGGELOS, A. K. *Machine Learning Refined: Foundations, Algorithms, and Applications*. Cambridge: Cambridge University Press, 2016.

Referências (cont.)

WERBOS, P. J. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Tese (Doutorado) — Harvard University, 1974.

WHITE, F. *Mecânica dos Fluidos*. 6. ed. Porto Alegre: McGraw Hill Brasil, 2010.

WIDROW, B. Generalization and information storage in networks of adaline neurons. *Self-Organizing Systems*, Spartan, p. 435–461, 1962.

WILAMOWSKI, B.; IRWIN, J. *The Industrial Electronics Handbook - Intelligent Systems*. Boca Raton, FL, US: CRC Press, 2016. (ENGnetBASE 2015).

ZIGRANG, D. J.; SYLVESTER, N. D. Explicit approximations to the solution of Colebrook's friction factor equation. *AIChE Journal*, v. 28, n. 3, p. 514–515, 1982. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/aic.690280323>. Acesso em: 25 out. 2018.

ZURADA, J. *Introduction to Artificial Neural Systems*. St. Paul, MN, USA: West Publishing Co., 1992.

Obrigado!