



Fundação Universidade Federal do ABC

Pró reitoria de pesquisa

Av. dos Estados, 5001, Santa Terezinha, Santo André/SP, CEP 09210-580

Bloco L, 3ºAndar, Fone (11) 3356-7617

iniciacao@ufabc.edu.br

Relatório Final de Iniciação Científica referente
ao Edital Nº 01/2019

Nome do aluno: Carolina Rosa Villegas

Assinatura do aluno:

Nome do orientador: Bruno Guzzo da Silva

Assinatura do orientador:

Título do projeto: Perda de carga em leito fixo de partículas de xisto betuminoso: Modelagem via Redes Neurais Artificiais

Palavras-chave do projeto: rede neural artificial; modelagem de processos; queda de pressão; xisto; leito fixo

Área do conhecimento do projeto: Engenharia de Processos

Bolsista: Sim. Modalidade PIC

São Paulo

14 de agosto de 2020

Sumário

1 Resumo.....	3
2 Introdução	4
3 Fundamentação teórica	5
3.1 Redes Neurais Artificiais.....	5
3.2 Sistemas Particulados.....	6
3.3 Xisto.....	6
4 Metodologia	7
4.1 Materiais e Métodos	7
4.2 Etapas da pesquisa	9
5 Resultados e discussão dos resultados	10
5.1 Treinamento e Validação utilizando a função tangente hiperbólica.....	10
5.2 Treinamento, Validação e Teste utilizando a função tangente hiperbólica.....	14
5.3 Treinamento, Validação e Teste utilizando a função sigmoidal.....	20
6 Conclusões e perspectivas de trabalhos futuros.....	25
Referências	26
APÊNDICE A - Código utilizado para treinamento e validação das RNAs no <i>software</i> MATLAB.....	27
APÊNDICE B - Código utilizado para treinamento, validação e teste das RNAs no <i>software</i> MATLAB.....	29

1 Resumo

A determinação da queda de pressão através de leitos de sólidos é fundamental para o projeto de colunas empacotadas, além de ser necessária para o dimensionamento de acessórios como compressores e bombas. Neste projeto várias topologias de Redes Neurais Artificiais (RNAs) para a previsão da queda de pressão em leito fixo de partículas de xisto betuminoso foram investigadas. Nas modelagens foram utilizadas RNAs do tipo *feedforward* formadas por 3 camadas. A camada de entrada é formada pelas variáveis independentes do processo, sendo três: porosidade do leito, velocidade do fluido e razão entre diâmetro da coluna e diâmetro das partículas. Nos neurônios da camada oculta foram investigados os desempenhos das funções de ativação tangente hiperbólica e sigmoideal. Para o treinamento das RNAs utilizou-se o algoritmo de Levenberg-Marquardt com regularização bayesiana. O coeficiente de determinação (R^2), a raiz do erro quadrático médio (RMSE), o coeficiente angular e linear da reta obtida em cada simulação foram analisados para comparar os desempenhos das RNAs. Os resultados mostraram que uma RNA com 21 neurônios ocultos apresentou os resultados mais satisfatórios ($R^2 > 0,99$), o que indica que o modelo neural foi uma ferramenta promissora para prever as condições operacionais de leitos fixos, como uma alternativa às equações tradicionais. Além disso, através das simulações, foi possível concluir que a melhor função utilizada foi a sigmoideal uma vez que as RNAs formadas por 18 neurônios ocultos já apresentavam o R^2 maior que 0,99 e o RMSE do teste menor que 210 Pa/m enquanto que para a função tangente hiperbólica isso acontecia para 19 neurônios ocultos em diante.

2 Introdução

Colunas empacotadas de sólidos (leitos fixos, fluidizados e pneumáticos) são amplamente utilizadas na indústria química, sendo que a gama de aplicações inclui a secagem de grãos, a filtração, os reatores químicos, os adsorvedores, entre outros. Para o projeto e operação de tais equipamentos, é necessária uma boa compreensão da perda de energia (queda de pressão) que acompanha o fluxo de fluidos através das colunas compactadas de sólidos. Por exemplo, a descrição correta da queda de pressão em leitos empacotados é de fundamental importância para o projeto de reatores de leito fixo, uma vez que determina os requisitos de energia das bombas e compressores. A queda de pressão nos leitos empacotados depende, principalmente, das características dos equipamentos, dos sólidos e do fluido. Os principais fatores a serem considerados são: diâmetro da coluna; porosidade média e altura do leito; velocidade, viscosidade e massa específica do fluido; e tamanho, massa específica e forma das partículas [ERGUN, 1952; EISFELD e SCHNITZLEIN, 2001; CHOI et al., 2008].

O xisto betuminoso é uma rocha sedimentar contendo material orgânico sólido, espalhado por uma matriz mineral, que é convertido em óleo e gás por meio de aquecimento (pirólise), normalmente realizada em retorta de leito fixo. O óleo de xisto possui as características do petróleo, podendo ser refinado para produzir os derivados do mesmo. A modelagem de processos de xisto demanda o conhecimento das propriedades do mesmo. Várias são as formas de contato gás-sólido que podem ser utilizadas para o processamento de xisto. Entre elas estão o leito fixo, o leito em jorro, o leito fluidizado e o leito em arraste, cada uma adequada a um tamanho de partícula [OJA e SUUBERG, 2013].

O modelo mais utilizado para a descrição da queda de pressão em leitos fixos é a equação de Ergun, que expressa a queda de pressão como a soma das perdas de energia causadas por perdas simultâneas de energia inercial e viscosa [ERGUN, 1952]. No entanto, se o diâmetro do leito não for grande em comparação com o diâmetro das partículas, a queda de pressão calculada pela equação de Ergun apresenta erros em função dos efeitos de parede [WINTERBERG e TSOTSAS, 2000; EISFELD e SCHNITZLEIN, 2001; DI FELICE e GIBILARO, 2004; CHOI et al., 2008; CHENG, 2011].

Uma ferramenta de modelagem alternativa e promissora, para a determinação da queda de pressão, é a Rede Neural Artificial (RNA). As RNAs surgiram como uma inspiração na estrutura neural de organismos inteligentes. Essa ferramenta computacional foi criada como um modelo de resolução de problemas complexos por meio da simulação do cérebro humano, inclusive em seu comportamento, ou seja, aprendendo, errando e fazendo descobertas. Uma RNA é uma coleção massivamente paralela de pequenas unidades de processamento, que computam determinadas funções matemáticas, cujas interligações formam a maior parte da inteligência da RNA. As RNAs, uma subespecialidade da Inteligência Artificial, são sem dúvida uma “poderosa” ferramenta matemática não convencional, que pode ser aplicada para modelagem de processos industriais. É uma abordagem alternativa à forma fenomenológica de resolver problemas, principalmente para o ajuste não linear e multivariável de dados [HIMMELBLAU, 2008].

O elevado número de trabalhos publicados que aplicaram as RNAs reflete a potencialidade desta ferramenta de modelagem empírica em diversos processos da engenharia [HIMMELBLAU, 2008]. Algumas aplicações em que as RNAs foram usadas foram: para modelar um secador de leito fluidizado [SATISH e SETTY, 2005]; para prever a queda de pressão em um leito fixo rotativo [LASHKARBOLOOKI et al., 2012]; para modelar uma coluna compactada visando calcular a sua queda de pressão [PANDHARIPANDE e SINGH, 2012]; para prever a velocidade mínima em leitos de jorro [ZHONG et al., 2013]; e para prever a queda de pressão em um leito fluidizado [KRISHNA et al., 2017]. No presente trabalho, várias topologias de RNAs para a previsão da queda de pressão em leito fixo de partículas de xisto betuminoso foram investigadas visando obter um modelo baseado em Inteligência Artificial para a descrição da fluidodinâmica do escoamento do ar em um leito fixo.

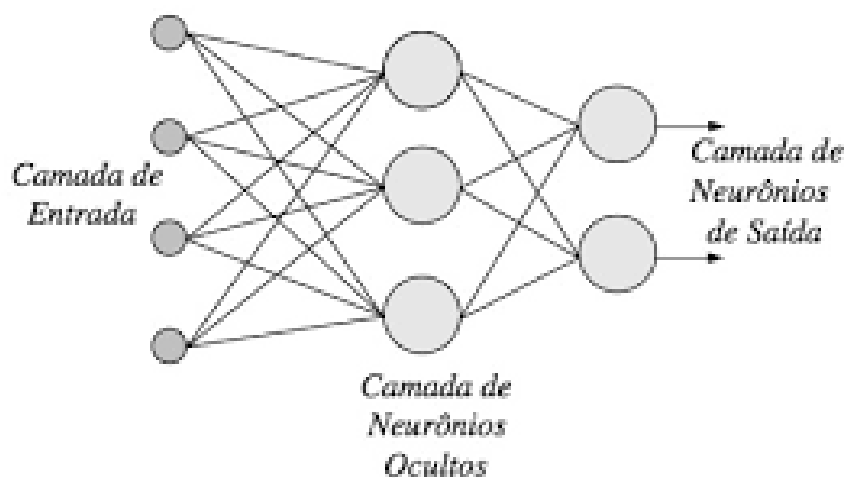
3 Fundamentação teórica

3.1 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) surgiram no final da década de 1980 como uma forma de computação caracterizada por sistemas que remetem a estrutura do cérebro humano. As RNAs são sistemas paralelos distribuídos compostos por unidades de processamento simples (neurônios artificiais) que calculam determinadas funções matemáticas (normalmente não-lineares). Tais unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos essas conexões estão associadas a pesos, os quais armazenam o conhecimento adquirido pelo modelo e servem para ponderar a entrada recebida por cada neurônio da rede [BRAGA et al., 2007]. Nas RNAs o procedimento usual na solução de problemas passa inicialmente por uma fase de aprendizagem, em que um conjunto de exemplos é apresentado para a rede, a qual extrai as características necessárias para representar a informação fornecida. Essas características são utilizadas posteriormente para gerar respostas para o problema [BRAGA et al., 2007].

As RNAs, do ponto de vista fisiológico, ainda diferem bastante das redes biológicas. É importante, contudo, observar as similaridades entre esses dois tipos de sistemas, tanto para que se possa entender melhor o sistema nervoso quanto para buscar ideias e inspirações para a pesquisa em neurocomputação. Como características comuns, podemos citar que os dois sistemas são baseados em unidades de computação paralela e distribuída que se comunicam por meio de conexões sinápticas, possuem detectores de características, redundância e modularização das conexões [BRAGA et al., 2007]. Independentemente da função escolhida, neurônios individuais possuem capacidade computacional limitada. No entanto, um conjunto de neurônios artificiais conectados na forma de uma rede (neural) é capaz de resolver problemas de complexidade elevada [BRAGA et al., 2007]. Existem várias configurações possíveis de neurônios artificiais conectados na forma de RNAs, porém a utilizada neste trabalho foi a rede *feedforward* de três camadas (Figura 1).

Figura 1 - Rede neural *feedforward* de três camadas.



Fonte: Cintra (2018, p.11)

As redes neurais podem ser aplicadas nos mais diversos setores, tais como: setor financeiro; elétrico; automação e controle; modelagem de sistemas industriais; bioinformática; comércio eletrônico; e telecomunicações [BRAGA et al., 2007].

3.2 Sistemas Particulados

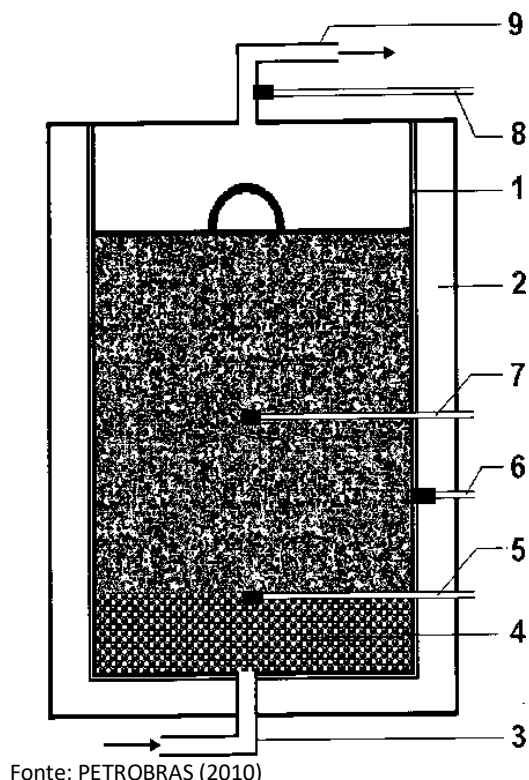
Nos sistemas particulados existe, necessariamente, a preocupação relativa ao entendimento fenomenológico da interação sólido-fluido e sólido-sólido envolvendo ou não o efeito de dispositivos fluidomecânicos. Os sistemas particulados estão, portanto, associados às operações de transporte, mistura, separação e modificação de tamanho de sólidos; escoamento de fluidos através de leitos fixos e móveis de partículas; centrifugação sólido-líquido; agitação de mistura sólido-líquido; separação de partículas por ciclones, hidrociclones, sedimentação, filtração; transporte pneumático e hidráulico de sólidos [CREMASCO, 2018]. Na indústria química, a pirólise é composta por um reator pirolítico (Figura 2) que tem como um dos seus constituintes um leito fixo no qual ocorre a interação entre partículas de xisto e gás. Tal leito é fixo uma vez que a matriz sólida é indeformável e as partículas estão constantemente em contato.

3.3 Xisto

O xisto é uma rocha sedimentar que possui quantidades significativas de matéria orgânica (óleo). Essa rocha pode ser utilizada diretamente como combustível para geração de calor e energia elétrica ou processada para produzir combustíveis líquidos ou materiais. O material orgânico contido na matriz mineral do xisto é denominado de querogênio ou betume, e é resultante da fossilização de matéria orgânica, principalmente algas, sedimentadas em ambientes aquáticos, tais como: lagos, lagoas, pântanos etc., que, com a ação do tempo, temperatura e pressão, foram transformadas em hidrocarbonetos de estruturas macromoleculares complexas. A existência de determinados elementos na matriz mineral do xisto, principalmente para aqueles com baixos teores de material orgânico, faz com que sua exploração se viabilize economicamente, uma vez que, após o processamento da rocha para a extração do óleo, ainda podem ser produzidos vários subprodutos como: enxofre, amônia, cimento, tijolos ou mesmo um combustível sólido de baixo poder calorífico para geração termoelétrica [DOS SANTOS, 2010].

O xisto, na indústria, passa pelo processo da pirólise que consiste no aquecimento em um ambiente com pouco ou nenhum oxigênio da matéria orgânica com o objetivo de transformá-la em óleo e gás. No reator pirolítico (Figura 2) a matéria orgânica passa pela etapa de secagem liberando produtos gasosos e logo após passa pela zona de pirólise onde ocorre reações químicas como fusão, volatilização e oxidação que retiram vários subprodutos. Por último, o processo é finalizado na zona de resfriamento [LEAL, 2010].

Figura 2 - Representação esquemática de um possível reator de pirólise.



Na Figura 2 encontra-se um modelo de reator pirolítico que possui os seguintes componentes: um vaso (1) externamente envolvido por um isolamento térmico (2), onde este vaso (1) possui a capacidade de processar xisto minerado, cru e em partículas; apresenta no fundo uma entrada (3) para gás inerte sob pressão superatmosférica aquecido a uma temperatura na faixa entre 480 °C a 705 °C; gás inerte quente que é injetado através de um leito de esferas (4) dispostas no fundo do vaso (1), as quais têm a finalidade de difundir o gás quente de maneira uniforme pelo interior deste vaso (1); apresenta medidores de temperatura localizados em diferentes pontos do reator: um primeiro medidor (5) localizado no topo do leito de esferas (4), com a finalidade de monitorar a temperatura de entrada do gás inerte; um segundo medidor (6) localizado na parede interna do vaso (1), com a finalidade de medir a temperatura da parede do vaso (1), um terceiro medidor de temperatura (7) localizado no meio do leito reacional, com a finalidade de medir a temperatura efetiva da pirólise e um quarto medidor de temperatura (8) localizado na saída (9) dos gases reacionais, com a finalidade de medir a temperatura no topo do vaso (1); adicionalmente o vaso (1) é equipado com medidores de fluxo não mostrados na Figura 2, que têm a finalidade de tornar possível a avaliação da influência destas variáveis no desempenho do processo [PETROBRAS, 2010].

4 Metodologia

4.1 Materiais e Métodos

Para realizar as modelagens e simulações, foi utilizado como ferramenta computacional o *software* MATLAB (*Neural Network Toolbox*).

Em todas as modelagens investigadas foi utilizada uma RNA do tipo *feedforward* (**newff** no MATLAB) formada por três camadas: uma camada de entrada; uma camada oculta; e uma camada de saída (Figura 3). Na camada oculta, as funções de ativação dos neurônios foram as funções tangente

hiperbólica (**tansig** no MATLAB) e sigmoidal (**logsig** no MATLAB). Já a função de ativação utilizada no neurônio da camada de saída foi a função linear (**purelin** no MATLAB).

A função **tansig** é definida de acordo com a Equação 4.1:

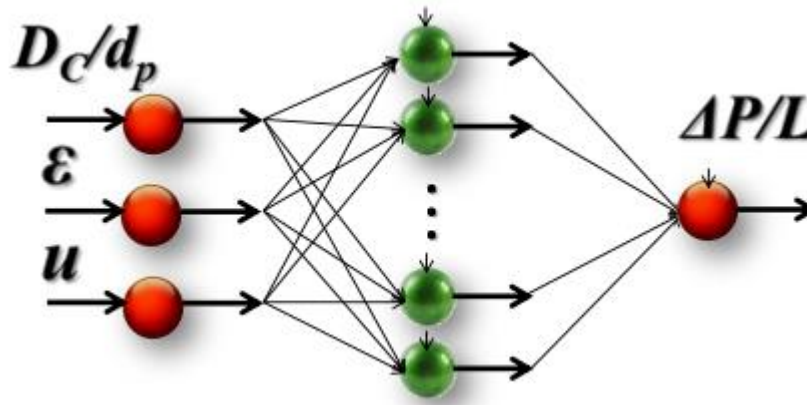
$$f(x) = (1 - e^{-x}) / (1 + e^{-x}) \quad (4.1)$$

Já a função **logsig** é definida de acordo com a Equação 4.2:

$$f(x) = 1 / (1 + e^{-x}) \quad (4.2)$$

x = soma das entradas dos neurônios ocultos (Figura 4).

Figura 3 – Topologia da rede neural utilizada.



Fonte: da Silva (2020)

Legenda:

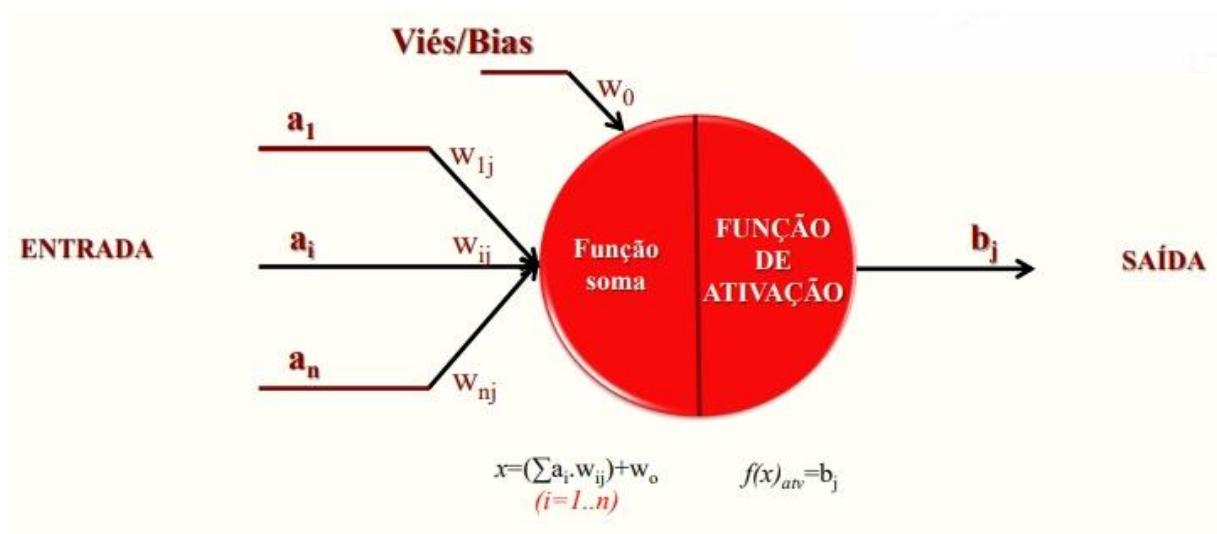
D_c/d_p a razão entre o diâmetro da coluna (D_c) e o diâmetro da partícula (d_p);

ϵ a porosidade do leito;

u a velocidade superficial;

$\Delta P/L$ a perda de carga no leito.

Figura 4 – Função de ativação nos neurônios ocultos.



Fonte: da Silva (2020)

Legenda:

a_1 , a_i e a_n os neurônios de entrada;

w_0 , w_{1j} , w_{ij} e w_{nj} os vieses ou bias;

b_j o neurônio de saída.

O programa desenvolvido (APÊNDICE A e APÊNDICE B) apresentou, de forma geral, as seguintes etapas: carregamento do banco de dados, normalização dos dados, divisão dos dados (treinamento e verificação), definição da arquitetura da RNA, treinamento e verificação da RNA.

O banco de dados foi organizado para que garantisse um mapeamento para qualquer par entrada/saída que fosse introduzido na RNA e que fizesse parte de seu domínio. Assim, a partir de dados experimentais da literatura [FOLTIN, 2013] foi organizado um banco de dados constituído por experimentos que foram utilizados na etapa de treinamento da RNA e por experimentos que foram utilizados na etapa de verificação da RNA. O banco de dados apresentou os valores máximos e mínimos das variáveis operacionais na etapa de treinamento. O banco de dados foi normalizado, procurando-se desta maneira eliminar eventuais problemas relacionados às diferentes magnitudes dos dados. Para cada variável foram assumidos os valores +1 para o seu maior valor real e -1 para o seu menor valor real. Esta normalização foi realizada no MATLAB utilizando o comando **premnmx** no APÊNDICE A e o comando **mapminmax** no APÊNDICE B. Os dados de saída da RNA foram desnormalizados utilizando o comando **postmnmx**.

Na literatura utilizada como banco de dados [FOLTIN, 2013], foram realizados experimentos com 12 faixas granulométricas diferentes e para cada faixa era modificado o tamanho da coluna de ar. As faixas granulométricas possuíam diâmetros médios de 0,256 mm até 4,38 mm e os diâmetros das colunas utilizadas foram: 0,0173 m; 0,0246 m; 0,0354 m; e 0,0470 m.

Para a etapa de treinamento da RNA, foi utilizado o algoritmo de treinamento de Levenberg-Marquardt com regularização bayesiana (função **trainbr** no MATLAB), pois esta técnica evita o sobreajuste de dados. A raiz do erro quadrático médio (RMSE) de cada modelagem foi calculada, tanto para a etapa de treinamento como para a etapa de verificação da rede. Após a etapa de treinamento, foi realizada a etapa de verificação da rede, sendo apresentados novos dados de entrada, obtendo-se assim, por simulação, os dados de saída desejados. O valor determinado através do modelo neural foi então comparado ao valor real através da análise dos gráficos de dispersão (dados reais de verificação versus preditos pelo modelo). Quando os dados se apresentaram na forma de uma reta coincidente com a diagonal, em que o coeficiente linear se aproximava de zero e o coeficiente angular e o R^2 fossem próximos de um, a configuração do modelo neural foi considerada adequada. Estes parâmetros, juntamente com a raiz do erro médio quadrático (RMSE) foram utilizados para definir a topologia de RNA mais adequada para representar o processo.

O RMSE foi definido de acordo com a Equação 4.3:

$$RMSE = \sqrt{(\sum e_a^2)/N} \quad (4.3)$$

e_a = valor real – valor simulado;

$a = 1, 2, 3, \dots, N$;

N = número de pontos experimentais.

Em cada topologia de RNA, o número de neurônios da camada oculta foi determinado buscando-se um ajuste eficaz do modelo neural aos dados de verificação. Deste modo, durante as modelagens o número de neurônios ocultos variou de 3 a 30 e para cada número fixo de neurônios foram realizados vários treinamentos até encontrar o melhor ajuste da RNA aos dados de verificação.

4.2 Etapas da pesquisa

Primeiramente foi realizada uma revisão bibliográfica abordando os seguintes temas: RNAs; xisto; leito fixo de sólidos; e sistemas particulados. Após a revisão, iniciou-se o treinamento da RNA a partir de um banco de dados normalizado no MATLAB e foi escolhida a topologia de RNA do tipo *feedforward* na qual o número de neurônios da camada oculta foi alterado de forma a buscar um ajuste eficaz aos dados de verificação. Em seguida, ocorreu a etapa de treinamento da RNA na qual foi calculada a raiz do erro quadrático médio (RMSE) de cada modelagem. Por fim, foi realizada a etapa de validação da rede, a partir de novos dados de entrada oriundos do banco de dados para que o valor determinado através do modelo neural fosse comparado ao valor real.

A etapa seguinte foi focada na determinação das influências da função de ativação dos neurônios ocultos na modelagem por RNAs e no cálculo do RMSE para cada simulação, sendo alternadas as funções *tansig* e *logsig*. Com isso, cada RNA passou pelo processo de treinamento, validação e teste.

5 Resultados e discussão dos resultados

5.1 Treinamento e Validação utilizando a função tangente hiperbólica

As Tabelas 1, 2 e 3 apresentam os resultados obtidos do treinamento e da validação das RNAs com diferentes quantidades de neurônios ocultos e com o código do APÊNDICE A sendo o SSE do treinamento, em todas as simulações deste trabalho, definido de acordo com a Equação 5.1:

$$SSE = MSE \times 342 \quad (5.1)$$

TABELA 1 – RESULTADOS DE TREINAMENTO E VALIDAÇÃO DA RNA COM NEURÔNIOS OCULTOS ENTRE 3 E 9

Treinamento					Validação		
Topologia da RNA	Número de parâmetros originais	Número de parâmetros efetivos	Número de iterações	SSE	RMSE (Pa/m)	R ²	Equação da reta (Y=ax+b)
3x3x1	16	14,6	75	22,6	1,3282x10 ³	0,70156	Y=0,98x+43
3x4x1	21	19	134	21,3	1,2919x10 ³	0,71748	Y=0,99x-3,5
3x5x1	26	23,5	96	18	1,2178x10 ³	0,74931	Y=0,97x+83
3x6x1	31	28,5	145	16,3	1,1704x10 ³	0,76876	Y=0,97x+95
3x7x1	36	30,9	160	12,6	1,0440x10 ³	0,81659	Y=0,97x+60
3x8x1	41	35,6	568	11,1	975,4384	0,83999	Y=0,97x+56
3x9x1	46	38	835	11,1	977,6453	0,83929	Y=0,97x+57

TABELA 2 – RESULTADOS DE TREINAMENTO E VALIDAÇÃO DA RNA COM NEURÔNIOS OCULTOS ENTRE 10 E 21

Treinamento					Validação		
Topologia da RNA	Número de parâmetros originais	Número de parâmetros efetivos	Número de iterações	SSE	RMSE (Pa/m)	R ²	Equação da reta (Y=ax+b)
3x10x1	51	46,8	325	8,77	884,2917	0,86859	Y=0,97x+41
3x11x1	56	48,9	490	8,96	872,4611	0,87165	Y=0,97x+62
3x12x1	61	52,1	374	7,95	835,9622	0,88247	Y=0,98x+31
3x13x1	66	53,8	356	8,63	881,8279	0,86948	Y=0,97x+56
3x14x1	71	63,4	720	6,73	773,6033	0,89925	Y=0,98x+59
3x15x1	76	66,9	485	6,44	778,2709	0,89931	Y=0,96x+80
3x16x1	81	75,9	1167	1,19	314,1776	0,98327	Y=x-14
3x17x1	86	81,3	886	0,977	304,4038	0,98434	Y=0,99x+13
3x18x1	91	85,6	985	0,549	226,95	0,99128	Y=x+3,5
3x19x1	96	90,5	2546	0,342	182,4567	0,99437	Y=x-12
3x20x1	101	94,5	627	0,278	161,4335	0,99558	Y=x-9,8
3x21x1	106	100	1798	0,228	141,2201	0,99662	Y=x+0,11

TABELA 3 – RESULTADOS DE TREINAMENTO E VALIDAÇÃO DA RNA COM NEURÔNIOS OCULTOS ENTRE 22 E 30

Treinamento					Validação		
Topologia da RNA	Número de parâmetros originais	Número de parâmetros efetivos	Número de iterações	SSE	RMSE (Pa/m)	R ²	Equação da reta (Y=ax+b)
3x22x1	111	103	1208	0,273	167,4541	0,99525	Y=x+8,1
3x23x1	116	107	1263	0,279	159,0560	0,99570	Y=x+4,3
3x24x1	121	112	955	0,234	164,1478	0,99544	Y=x-16
3x25x1	126	116	1129	0,190	140,5495	0,99666	Y=x-7,3
3x26x1	131	121	1319	0,210	152,6068	0,99606	Y=x+11
3x27x1	136	128	1787	0,232	170,4439	0,99509	Y=x+15
3x28x1	141	131	2006	0,183	149,9224	0,99620	Y=x+0,58
3x29x1	146	131	1290	0,213	155,3040	0,99590	Y=x+1,7
3x30x1	151	136	1453	0,185	161,2088	0,99558	Y=x+5,2

Nas Tabelas 1, 2 e 3 observa-se que, de forma geral, conforme o número de neurônios ocultos foi aumentando o valor do RMSE da validação foi diminuindo, assim como o valor do SSE do treinamento. Além disso, o valor do R² aproximou-se de 1, o coeficiente angular (a) obtido na equação da reta (Y=ax+b) tendeu a 1, o coeficiente linear (b) diminuiu, e graficamente geraram-se pontos de dispersão mais próximos da reta ideal.

As topologias que mais se adequaram ao resultado real foram aquelas com 18 ou mais neurônios ocultos. Estas obtiveram o coeficiente angular e o R² próximos de 1, sendo que com 5 neurônios ocultos (Figura 5) a simulação obteve dados muito dispersos do ideal e com 21 neurônios ocultos (Figura 6) a simulação obteve resultados próximos dos realizados em laboratório.

Figura 5 – Resultados no MATLAB da RNA com 5 neurônios ocultos.

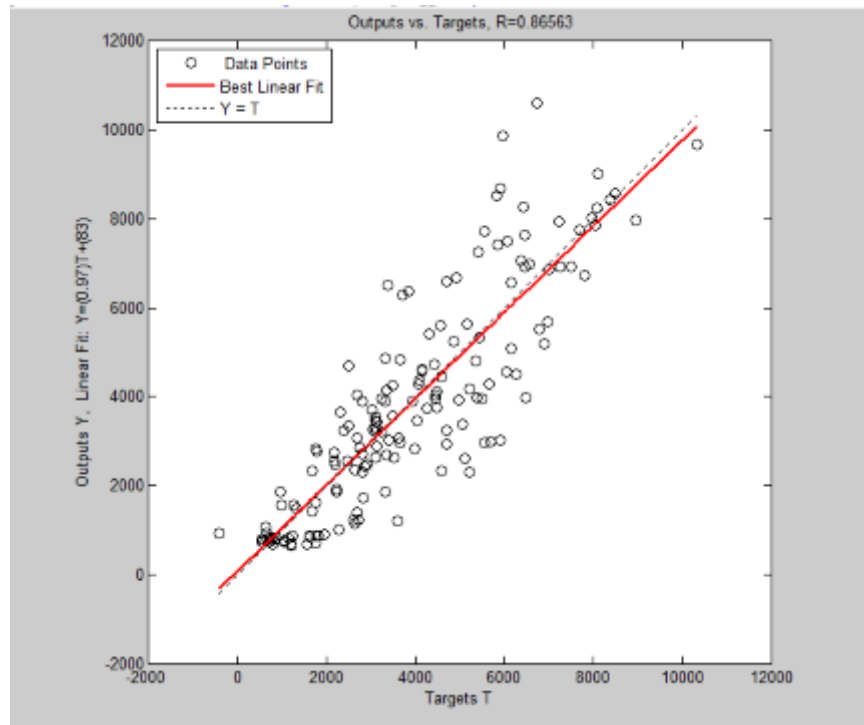
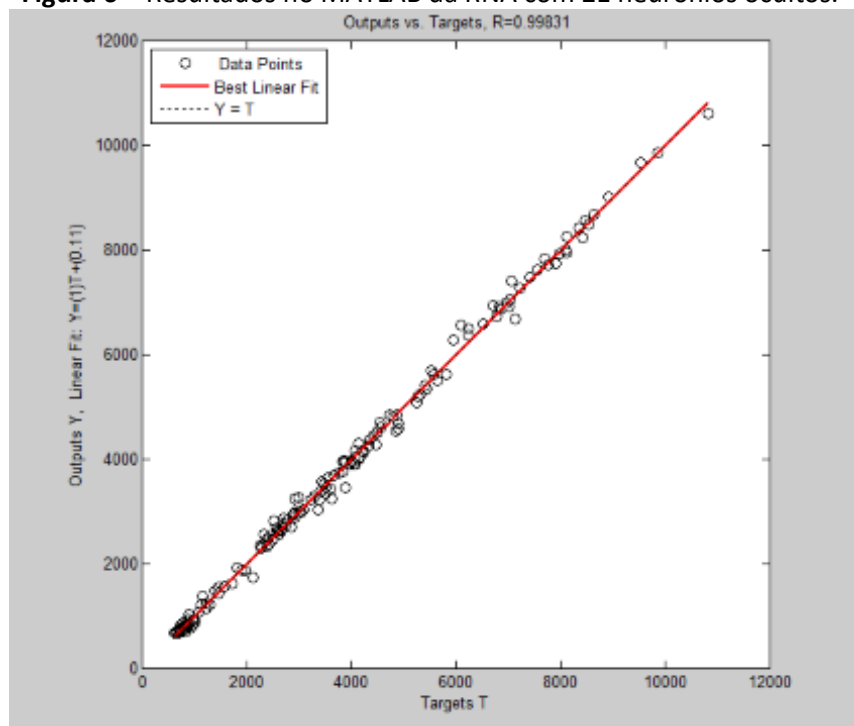


Figura 6 – Resultados no MATLAB da RNA com 21 neurônios ocultos.



5.2 Treinamento, Validação e Teste utilizando a função tangente hiperbólica

Após as simulações do capítulo 5.1, foi realizado o treinamento, validação e teste das RNAs utilizando a função *tansig*. As Tabelas 4, 5 e 6 apresentam os resultados obtidos com diferentes quantidades de neurônios ocultos e com o código do APÊNDICE B.

TABELA 4 – RESULTADOS DE TREINAMENTO E VALIDAÇÃO DA RNA COM NEURÔNIOS OCULTOS ENTRE 3 E 13

Treinamento					Validação		
Topologia da RNA	Número de parâmetros originais	Número de parâmetros efetivos	Número de iterações	SSE	RMSE (Pa/m)	R ²	Equação da reta (Y=ax+b)
3x3x1	16	14,6	65	22,6404	1,4187x10 ³	0,6905	Y=0,98x-1,2
3x4x1	21	19	147	21,3408	1,3799x10 ³	0,7071	Y=0,99x-42
3x5x1	26	23,5	105	17,9892	1,2663x10 ³	0,7530	Y=0,99x-27
3x6x1	31	26,4	265	17,8524	1,2738x10 ³	0,7509	Y=0,98x+3,1
3x7x1	36	30,9	150	12,5856	1,0413x10 ³	0,8338	Y=0,98x+0,92
3x8x1	41	35,8	249	14,9796	1,1608x10 ³	0,7923	Y=0,99x-0,24
3x9x1	46	40,6	300	9,7128	933,4611	0,8659	Y=x-33
3x10x1	51	45,8	374	8,6526	888,2393	0,8783	Y=x-19
3x11x1	56	49,2	358	9,3366	905,6684	0,8738	Y=x-61
3x12x1	61	54,1	453	7,5240	826,7558	0,8959	Y=0,97x+55
3x13x1	66	55,6	603	8,2080	869,7312	0,8836	Y=x-32

TABELA 5 – RESULTADOS DE TREINAMENTO E VALIDAÇÃO DA RNA COM NEURÔNIOS OCULTOS ENTRE 14 E 26

Treinamento					Validação		
Topologia da RNA	Número de parâmetros originais	Número de parâmetros efetivos	Número de iterações	SSE	RMSE (Pa/m)	R ²	Equação da reta ($Y=ax+b$)
3x14x1	71	59,2	304	7,6950	836,8478	0,8924	$Y=0,99x+3,7$
3x15x1	76	66,3	543	6,6348	830,4261	0,8941	$Y=0,98x+71$
3x16x1	81	75,7	1030	1,0328	286,0676	0,9874	$Y=x-12$
3x17x1	86	78,5	729	1,7989	405,0944	0,9747	$Y=x-8,3$
3x18x1	91	85,8	686	0,7490	277,1578	0,9883	$Y=0,99x+18$
3x19x1	96	90,4	1190	0,3249	177,8500	0,9952	$Y=0,99x+18$
3x20x1	101	92,2	1553	0,3865	196,2423	0,9941	$Y=x-5,4$
3x21x1	106	101	840	0,3016	155,9849	0,9963	$Y=x+4,2$
3x22x1	111	105	1619	0,2596	151,2507	0,9966	$Y=0,99x+30$
3x23x1	116	106	2621	0,2319	135,5226	0,9972	$Y=0,99x+20$
3x24x1	121	113	1551	0,1895	130,6532	0,9974	$Y=0,99x+12$
3x25x1	126	117	1131	0,2575	152,7751	0,9966	$Y=0,99x+29$
3x26x1	131	118	1115	0,2329	144,4015	0,9968	$Y=x+13$

TABELA 6 – RESULTADOS DE TREINAMENTO E VALIDAÇÃO DA RNA COM NEURÔNIOS OCULTOS ENTRE 27 E 30

Treinamento					Validação		
Topologia da RNA	Número de parâmetros originais	Número de parâmetros efetivos	Número de iterações	SSE	RMSE (Pa/m)	R ²	Equação da reta (Y=ax+b)
3x27x1	136	124	1856	0,1761	137,3269	0,9972	Y=0,99x+29
3x28x1	141	127	1463	0,1748	135,7849	0,9972	Y=x-7,5
3x29x1	146	132	1687	0,2387	145,7568	0,9969	Y=0,99x+36
3x30x1	151	134	2256	0,2021	143,4922	0,9969	Y=0,99x+18

Em consonância com os resultados discutidos anteriormente, nas Tabelas 4, 5 e 6 observa-se que, de forma geral, conforme o número de neurônios ocultos foi aumentando o valor do RMSE da validação foi diminuindo, assim como o valor do SSE do treinamento. Além disso, o valor do R² aproximou-se de 1, o coeficiente angular (a) obtido na equação da reta (Y=ax+b) tendeu a 1, o coeficiente linear (b) diminuiu, e graficamente geraram-se pontos de dispersão mais próximos da reta ideal (Figura 7). A topologia que mais se adequou ao resultado real foi aquela com 24 neurônios ocultos. Assim como no caso anterior, esta obteve o coeficiente angular e o R² mais próximos de 1.

Nas Tabelas 7, 8, 9 e 10, são mostrados os pesos e os vieses entre neurônios de entrada, neurônios ocultos e neurônio de saída para a RNA com 24 neurônios ocultos.

TABELA 7 - PESOS ENTRE NEURÔNIOS OCULTOS DE 1 A 6 (RNA 3x24x1)

Pesos entre os neurônios da camada de entrada (I) e os neurônios da camada oculta (H)				Pesos entre os neurônios da camada oculta (H) e o neurônio da camada de saída (O)
	I1	I2	I3	O
H1	-0,6632	-2,9126	2,9663	-5,9320
H2	-4,6333	-9,8275	0,8074	5,3819
H3	12,9436	17,3029	8,1892	9,1348
H4	1,9643	5,1938	-0,1177	18,1853
H5	-2,4689	15,7996	-0,1452	-5,0396

H6	-1,2788	-0,5745	-4,4717	-0,0722
-----------	---------	---------	---------	---------

TABELA 8 - PESOS ENTRE NEURÔNIOS OCULTOS DE 7 A 24 (RNA 3x24x1)

Pesos entre os neurônios da camada de entrada (I) e os neurônios da camada oculta (H)				Pesos entre os neurônios da camada oculta (H) e o neurônio da camada de saída (O)
	I1	I2	I3	O
H7	-10,3786	8,5773	-0,6835	4,3843
H8	-7,1520	7,8499	1,1435	2,3803
H9	14,5118	-7,5175	-0,7183	-4,9059
H10	15,1575	-17,8917	-0,9584	-1,9255
H11	-9,2197	0,0759	3,4934	-9,0292
H12	-2,9035	0,5108	-0,5374	-9,4255
H13	-21,2164	6,6050	0,7139	-5,1902
H14	3,6827	-1,7645	1,6732	1,2967
H15	5,0489	1,5704	-0,9720	3,7200
H16	6,8385	9,4755	6,9770	-15,2379
H17	-3,5782	-3,2423	3,4572	13,1776
H18	8,6058	-2,5318	-0,4618	9,3720
H19	5,9511	10,4981	-0,3840	-10,2836
H20	-4,9113	6,2502	-0,4975	-4,8075
H21	-6,2747	-2,0135	-5,3323	-4,6686
H22	-6,0254	-10,1836	8,5375	12,0857
H23	6,4762	-0,6968	0,1789	-11,1082
H24	11,1387	-6,7022	-0,7196	5,5679

TABELA 9 - VIÉS DOS NEURÔNIOS OCULTOS (H) DE 1 A 3 (RNA 3x24x1)

Quantidade de neurônios ocultos	Viés
H1	5,7724
H2	2,3650

H3	-11,1112
-----------	----------

TABELA 10 - VIÉS DOS NEURÔNIOS OCULTOS (H) ENTRE 4 E 24 E DO NEURÔNIO DE SAÍDA (O) (RNA 3x24x1)

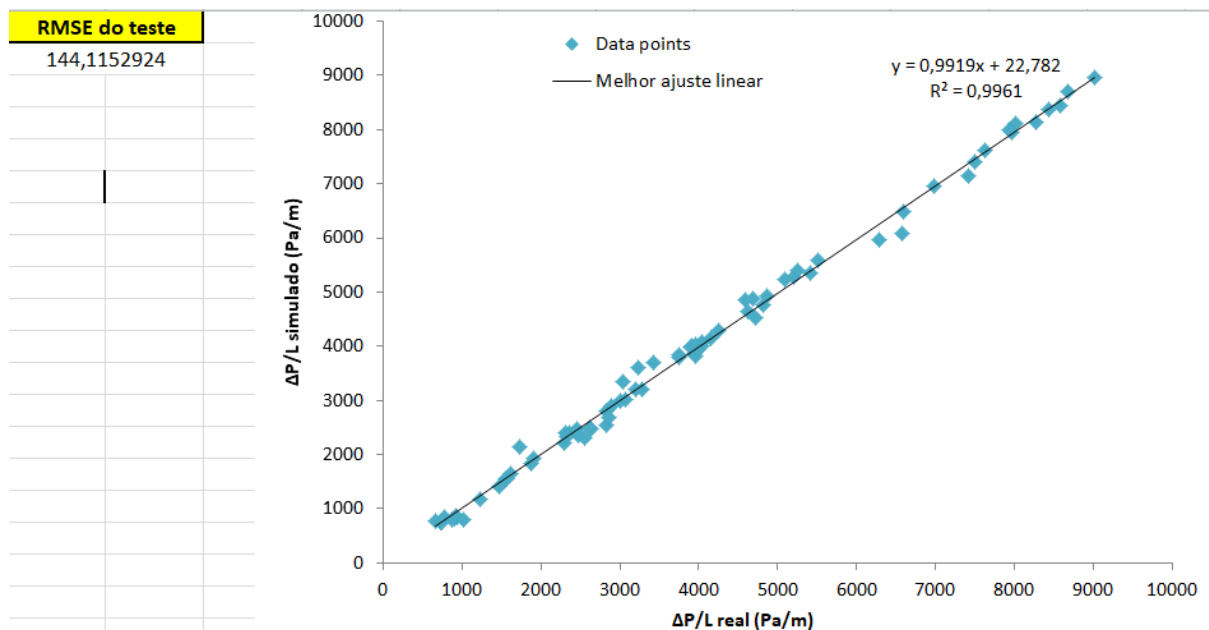
Quantidade de neurônios ocultos	Viés
H4	0,6391
H5	-3,1645
H6	-1,5509
H7	-6,8649
H8	-7,4525
H9	7,8767
H10	14,4377
H11	3,8638
H12	-1,8691
H13	-14,8729
H14	2,0049
H15	6,2601
H16	-4,2893
H17	5,8417
H18	3,6080
H19	2,6959
H20	-2,2743
H21	1,8033
H22	20,0614
H23	4,7194
H24	11,1028
O	-11,1489

Figura 7 – Resultados no MATLAB da RNA com 24 neurônios ocultos.



A seguir, são apresentados os resultados simulados pelo *software* MATLAB e os dados de teste. Estes dados de teste não foram apresentados na etapa de treinamento e validação da RNA. Os resultados indicam que a RNA apresentou um resultado satisfatório para prever a queda de pressão durante o fluxo de ar na coluna de leito fixo de xisto, pois a equação da reta do teste (Figura 8) se aproximou da equação da reta simulada (Figura 7), assim como o R^2 (Figuras 7 e 8) e o valor do RMSE (Tabela 5 e Figura 8).

Figura 8 – Resultados no Excel com 24 neurônios ocultos.



5.3 Treinamento, Validação e Teste utilizando a função sigmoidal

Na última etapa, foi realizado o treinamento, validação e teste das RNAs utilizando a função *logsig*. As Tabelas 11 e 12 apresentam os resultados obtidos com diferentes quantidades de neurônios ocultos.

TABELA 11 – RESULTADOS DE TREINAMENTO E VALIDAÇÃO DA RNA COM NEURÔNIOS OCULTOS ENTRE 3 E 23

Treinamento					Validação		
Topologia da RNA	Número de parâmetros originais	Número de parâmetros efetivos	Número de iterações	SSE	RMSE (Pa/m)	R ²	Equação da reta (Y=ax+b)
3x3x1	16	14,6	68	22,7088	1,4206x10 ³	0,6896	Y=0,98x-0,48
3x4x1	21	19,2	85	21,5118	1,3799x10 ³	0,7080	Y=0,98x-16
3x5x1	26	22,6	208	20,9646	1,3486x10 ³	0,7204	Y=0,98x-6,7
3x6x1	31	28,1	127	18,1602	1,2451x10 ³	0,7622	Y=0,99x-42
3x7x1	36	30,2	162	12,7908	1,0439x10 ³	0,8322	Y=0,99x-19
3x8x1	41	37	193	13,9536	1,0677x10 ³	0,8242	Y=0,99x+8,6
3x9x1	46	41,6	603	10,2942	954,9860	0,8594	Y=0,99x+21
3x10x1	51	43,8	259	9,8154	960,1102	0,8584	Y=0,99x-5,4
3x11x1	56	49,6	364	8,5842	862,5122	0,8854	Y=0,99x+2,5
3x12x1	61	54,4	744	7,6950	838,5993	0,8917	Y=x-32
3x13x1	66	58,4	447	7,1820	820,3567	0,8967	Y=0,99x-9,5

3x14x1	71	60,5	252	7,8660	854,5223	0,8879	$Y=0,99x-1,4$
3x15x1	76	65,7	589	6,2928	814,4785	0,8990	$Y=0,97x+50$
3x16x1	81	73,6	496	1,6929	391,2532	0,9765	$Y=x-33$
3x17x1	86	80,4	812	0,9850	328,0787	0,9834	$Y=x-12$
3x18x1	91	85,4	1811	0,5438	202,3558	0,9937	$Y=x-26$
3x19x1	96	90,2	2451	0,4685	177,0529	0,9953	$Y=0,99x+29$
3x20x1	101	94	1059	0,2880	156,6728	0,9962	$Y=x+9,5$
3x21x1	106	97,9	1068	0,2678	149,2299	0,9966	$Y=0,99x+17$
3x22x1	111	105	1578	0,3246	169,2803	0,9956	$Y=x+15$
3x23x1	116	108	1130	0,2178	187,0310	0,9946	$Y=x-4,9$

TABELA 12 – RESULTADOS DE TREINAMENTO E VALIDAÇÃO DA RNA COM NEURÔNIOS OCULTOS ENTRE 24 E 30

Treinamento					Validação		
Topologia da RNA	Número de parâmetros originais	Número de parâmetros efetivos	Número de iterações	SSE	RMSE (Pa/m)	R ²	Equação da reta ($Y=ax+b$)
3x24x1	121	111	1540	0,2326	158,0019	0,9964	$Y=0,99x+27$
3x25x1	126	114	1567	0,2189	146,9407	0,9968	$Y=0,99x+29$
3x26x1	131	119	907	0,2568	146,1822	0,9968	$Y=0,99x+33$
3x27x1	136	120	1691	0,1833	135,5125	0,9972	$Y=0,99x+11$
3x28x1	141	130	1342	0,2261	146,9950	0,9967	$Y=0,99x+24$
3x29x1	146	129	1773	0,1772	131,3969	0,9973	$Y=x+2,5$
3x30x1	151	134	1946	0,1878	137,4919	0,9974	$Y=0,99x+12$

Através das Tabelas 11 e 12 e pelos mesmos critérios adotados anteriormente, observa-se que a topologia que mais se adequou ao resultado real foi aquela com 29 neurônios ocultos. Assim como no caso anterior, esta obteve o coeficiente angular e o R² mais próximos de 1. Na Figura 9 é possível observar que o gráfico gerado a partir de 29 neurônios se aproxima da reta ideal.

Nas Tabelas 13, 14, 15 e 16, são mostrados os pesos e os vieses entre neurônios de entrada, neurônios ocultos e neurônio de saída para a RNA com 29 neurônios ocultos.

TABELA 13 - PESOS ENTRE NEURÔNIOS OCULTOS DE 1 A 8 (RNA 3x29x1)

Pesos entre os neurônios da camada de entrada (I) e os neurônios da camada oculta (H)				Pesos entre os neurônios da camada oculta (H) e o neurônio da camada de saída (O)
	I1	I2	I3	O
H1	-6,4828	-15,2284	1,3132	11,0016
H2	15,1303	13,9546	9,0077	-10,5844
H3	7,3047	-13,9332	-2,2317	-7,6632
H4	1,8373	32,2221	-1,3593	14,0480
H5	3,6953	-2,0349	-2,0758	-8,1586
H6	-1,5051	-7,9290	6,1386	11,2109
H7	-28,4938	-2,9739	-19,1044	18,7951
H8	11,8227	-3,2518	-1,7286	6,5706

TABELA 14 - PESOS ENTRE NEURÔNIOS OCULTOS DE 9 A 29 (RNA 3x29x1)

Pesos entre os neurônios da camada de entrada (I) e os neurônios da camada oculta (H)				Pesos entre os neurônios da camada oculta (H) e o neurônio da camada de saída (O)
	I1	I2	I3	O
H9	-27,8252	-20,7951	-18,3828	-21,3460
H10	3,1932	-10,3060	6,1647	-9,0637
H11	14,2972	-10,8186	-1,2432	16,6959
H12	1,6337	-18,0401	-0,5535	-7,0437
H13	-4,0122	-4,0416	-1,8576	6,8844
H14	-6,1953	9,8447	-6,3652	-13,8485
H15	-5,1227	11,8197	2,4372	-8,2458
H16	-1,4687	12,0092	-0,4379	12,4336
H17	7,2195	1,4910	5,2157	11,3554
H18	-11,0148	-17,4529	-2,7615	-10,8887
H19	1,5862	-15,8077	-1,3669	-7,2146
H20	18,4631	-9,5616	-2,1216	-13,5218
H21	8,1397	5,8351	2,3442	-17,3397
H22	15,1864	-10,6628	-2,7606	9,0281

H23	-10,7900	-1,0678	-3,0704	-4,8415
H24	-13,0643	20,4957	2,0844	7,5936
H25	-2,2498	8,1637	2,3061	9,0718
H26	7,5386	-3,2265	5,6022	-4,6846
H27	9,3281	15,8735	-4,3549	-8,0730
H28	-11,0344	9,5236	-6,8527	4,9219
H29	7,4118	-13,7325	0,1656	18,5677

TABELA 15 - VIÉS DOS NEURÔNIOS OCULTOS (H) ENTRE 1 E 27 (RNA 3x29x1)

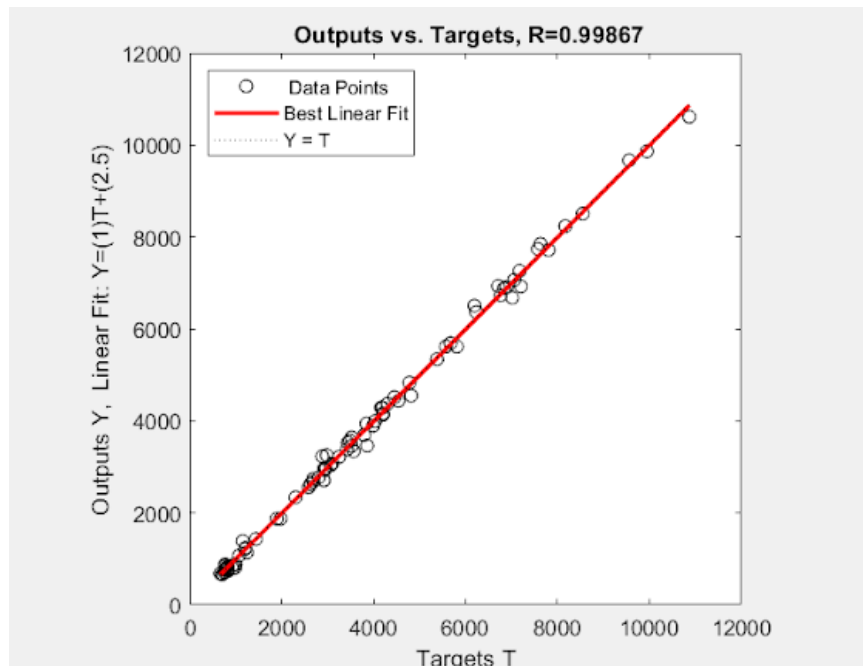
Quantidade de neurônios ocultos	Viés
H1	-2,0839
H2	-6,8317
H3	11,0296
H4	-3,4744
H5	2,9656
H6	12,9584
H7	-4,4889
H8	10,3227
H9	9,1818
H10	1,2155
H11	14,3675
H12	-0,6728
H13	-4,3994
H14	-4,1255
H15	-0,4924
H16	-7,5477

H17	1,0184
H18	-1,0295
H19	4,1188
H20	15,2667
H21	4,2271
H22	11,7834
H23	-8,4128
H24	-12,8620
H25	0,8722
H26	0,8135
H27	-7,2420

TABELA 16 - VIÉS DOS NEURÔNIOS OCULTOS (H) ENTRE 28 E 29 E DO NEURÔNIO DE SAÍDA (O) (RNA 3x29x1)

Quantidade de neurônios ocultos	Viés
H28	-8,6955
H29	8,6696
O	-10,2040

Figura 9 – Resultados no MATLAB da RNA com 29 neurônios ocultos.



A seguir, são apresentados os resultados simulados pelo *software* MATLAB e os dados de teste. Estes dados de teste não foram apresentados na etapa de treinamento e validação da RNA. Os resultados indicam que a RNA apresentou um resultado satisfatório para prever a queda de pressão durante o fluxo de ar na coluna de leito fixo de xisto, pois a equação da reta do teste (Figura 9) se aproximou da equação da reta simulada (Figura 10), assim como o R^2 (Figuras 9 e 10) e o valor do RMSE (Tabela 12 e Figura 10). No entanto, ao comparar as duas funções de ativação, percebe-se que com a função tangente hiperbólica a RNA treinada e validada foi mais eficiente para prever os dados de teste uma vez que os valores do RMSE, do R^2 e da equação da reta foram mais próximos entre si.

Na Tabela 17 encontra-se os valores comparados e seus erros relativos. É possível perceber que os erros relativos da função tangente hiperbólica são mais baixos que os da função sigmoidal.

Figura 10 – Resultados no Excel com 29 neurônios ocultos.

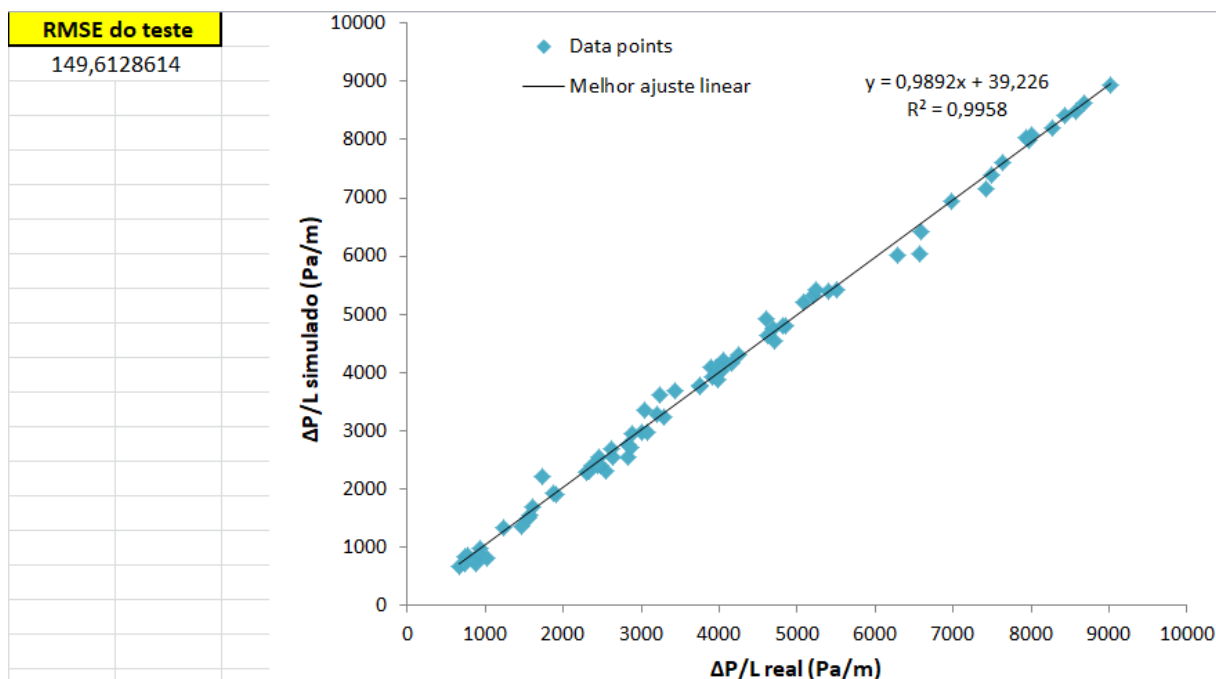


TABELA 17 – VALORES COMPARADOS DAS MELHORES RNAs E SEUS ERROS RELATIVOS

	TANGENTE HIPERBÓLICA			SIGMOIDAL		
	VALOR VALIDADO	VALOR DO TESTE	ERRO RELATIVO	VALOR VALIDADO	VALOR DO TESTE	ERRO RELATIVO
RMSE	130,6532	144,1152924	9,34%	131,3969	149,6128614	12,18%
EQUAÇÃO DA RETA	$Y=0,99x+12$	$Y=0,9919x+22,782$	—	$Y=x+2,5$	$Y=0,9892x+39,226$	—
R ²	0,9974	0,9961	0,13%	0,9973	0,9958	0,15%

6 Conclusões e perspectivas de trabalhos futuros

Para uma simulação com a função tangente hiperbólica, 24 neurônios resultaram em condições ótimas e para a função sigmoideal 29 neurônios aproximaram as simulações dos experimentos feitos em laboratório. Ademais, comparando as duas funções, a sigmoideal apresentou melhores resultados, pois chegou a um R^2 maior que 0,99 e a um RMSE da validação menor que 210 Pa/m mais rápido do que a função tangente hiperbólica. A função sigmoideal foi efetiva para 18 ou mais neurônios ocultos enquanto a tangente hiperbólica para 19 ou mais neurônios ocultos.

Em trabalhos futuros, sugere-se realizar simulações a partir do mesmo banco de dados para outro intervalo de neurônios ocultos e utilizar outra topologia da RNA em vez da *feedforward*.

Referências

BRAGA, A.P.; CARVALHO, A. P. F. L.; LUDERMIR, T. B.; *Redes Neurais Artificiais*. Editora LTC, 2007.

- CHENG, N.; *Wall effect on pressure drop in packed beds*. Powder Technol, p.261-266, 2011.
- CHOI, Y. S.; KIM, S. J.; KIM D.; *A Semi-empirical Correlation for Pressure Drop in Packed Beds of Spherical- Particles*. Transp. Porous Med., p. 133-149, 2008.
- CINTRA, R.S.C.; *Introdução à Neurocomputação*. INPE, 2018.
- CREMASCO, M. A.; *Operações unitárias em sistemas particulados e fluidomecânicos*. 3. Ed., 2018.
- DA SILVA, B. G.; *Modelagem de Processos Industriais por Redes Neurais Artificiais*. 2 figuras, 2020.
- DI FELICE, R.; GIBILARO, L. G.; *Wall effects for the pressure drop in fixed beds*. Chem. Eng. Sci., p. 3037-3040, 2004.
- DOS SANTOS, M. M.; *Xisto: Um estudo de viabilidade econômica para o Brasil.*, p.10-14, 2010.
- EISFELD, B.; SCHNITZLEIN, K.; *The influence of confining walls on the pressure drop in packed beds*. Chem. Eng. Sci., p. 4321-4329, 2001.
- ERGUN, S.; *Fluid Flow Through Packed Columns*. Chem. Eng. Prog., p. 89-94, 1952.
- FOLTIN, J.P.; *Avaliação da perda de carga em leito fixo de partículas irregulares utilizando xisto betuminoso, analisando a modelagem matemática através do efeito de parede e porosidade*. Universidade Federal do Paraná (UFPR), 2013.
- HIMMELBLAU, D. M.; *Accounts of experiences in the application of artificial neural networks in chemical engineering*. Ind. Eng. Chem. Res., p. 5782-5796, 2008.
- KRISHNA, M. S. V.; BEGUM, K. M. M. S.; ANANTHARAMAN, N.; *Hydrodynamic studies in fluidized bed with internals and modeling using ANN and ANFIS*. Powder Technol., p. 37-45, 2017.
- LASHKARBOLOOKI, M.; VAFERI, B.; MOWLA, D.; *Using Artificial Neural Network to Predict the Pressure Drop in a Rotating Packed Bed*. Sep. Sci. Technol., p. 2450-2459, 2012.
- LEAL, A.; *Pirólise - Reações Químicas*. Disponível em: <<https://www.infoescola.com/reacoes-quimicas/pirolise/>>. Acesso em: 14 jul. 2020, 17:26.
- OJA, V.; SUUBERG, E. M.; *Oil Shale Processing, Chemistry and Technology*. In: Malhotra, R. (Ed.). *Fossil Energy*, Springer, New York, p. 99–148, 2013.
- PANDHARIPANDE, S. L.; SINGH, A.; *Estimation of pressure drop of packed column using artificial neural network*. International Journal of Advanced Engineering Research and Studies, p. 01-03, 2012.
- PETROLEO BRASILEIRO S/A - PETROBRAS. *Modelo de reator para teste de pirólise de xisto*. Documento MU 8803173-0 U2, 28 set. 2010. Disponível em: <<https://www.escavador.com/patentes/318903/modelo-de-reator-para-teste-de-pirolise-de-xisto>>. Acesso em: 14 jul. 2020, 17:35.

SATISH, S.; SETTY, Y. P.; *Modeling Of A Continuous Fluidized Bed Dryer Using Artificial Neural Networks*. International Communications In Heat And Mass Transfer, p.539-547, 2005.

WINTERBERG, M.; TSOTSAS E.; *Impact of tube-to-particle-diameter ratio on pressure drop in packed beds*. AIChE Journal 46, p. 1084-1088, 2000.

ZHONG, W.; CHEN, X.; GRACE, J. R.; EPSTEIN, N.; JIN, B.; *Intelligent prediction of minimum spouting velocity of spouted bed by back propagation neural network*. Powder Technol., p. 197-203, 2013.

APÊNDICE A - Código utilizado para treinamento e validação das RNAs no software MATLAB

```
%%%%%%%%%%%%BANCO DE DADOS:
```

```
%limpa as variáveis do histórico:
```

```
clear;
```

```
%limpa o prompt:
```

```
clc;
```

```
%carregamento do arquivo de dados:
```

```
M=load('dados.dat');
```

```
%definição das variáveis de entrada e saída:
```

```
entrada=M(:,1:3); %colunas de 1 a 3
```

```
saida=M(:,4); %coluna 4
```

```
%matriz transposta:
```

```
entrada=entrada';
```

```
saida=saida';
```

```
%normalização dos dados entre -1 e 1:
```

```
[entradan,minentrada,maxentrada,saidan,minsaida,maxsaida]=premnmx(entrada,saida);
```

```
%divisão dos dados (treino e verificação):
```

```
%%treino:
```

```

entradatreina=entradan(:,1:342); %reserva as 342 linhas de um arquivo de 500 linhas

saidatreina=saidan(:,1:342);

%%verificação:

entradaverifica=entradan(:,343:500); %reserva os últimas linhas de um arquivo de 500 linhas

saidaverifica=saidan(:,343:500);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%ARQUITETURA DA REDE:

%rede feedforward:

net=newff(minmax(entradatreina),[4,1],{'tansig','purelin'},'trainbr');

%4 é o número de neurônios ocultos iniciais, 1 é o neurônio da saída

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%TREINAMENTO DA RNA:

%definição do número máximo de iterações (épocas):

net.trainParam.epochs=10000;

%define o objetivo de erro quadrático (MSE):

net.trainParam.goal=0.000001;

%inicialização dos pesos da rede aleatoriamente (chuta valores iniciais para os pesos e bias):

net=init(net);

%treinamento:

[net,tr]=train(net,entradatreina,saidatreina);

%determinados os pesos e bias da rede e arquivados temporariamente em "net"

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%VERIFICAÇÃO DA RNA:

%simulação da variável de saída a partir das entradas do conjunto de verificação;

Z=sim(net,entradaverifica);

%desnormalização da variável de saída;

S=postmnmx(Z,minsaida,maxsaida); %simulada

R=postmnmx(saidaverifica,minsaida,maxsaida); %real

```

```

%cálculo do RMSE:

erro=R-S

msev=mse(erro)

RMSEv=(msev)^0.5

%grafico das saídas (real x simulada):

figure(1);

postreg(R(1,:),S(1,:));

regress(R(:,1), S(:,1))

%Salvar a rede criada e os parâmetros ajustados no arquivo "rede":

save 'rede' net;

%Para visualizar os pesos e bias finais:

net.IW{1}

%pesos entre camadas entrada/intermediária

net.LW{2}

%pesos entre camadas intermediária/saída

net.b{1}

%bias da camada intermediária

net.b{2}

%bias da camada de saída

%Para visualizar o RMSE:

RMSEv

```

APÊNDICE B - Código utilizado para treinamento, validação e teste das RNAs no *software* MATLAB

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%BANCO DE DADOS:

%limpa as variáveis do histórico:

clear;

```

```

%limpa o prompt:

clc;

%carregamento do arquivo de dados (500 linhas e 4 colunas):

M=load('banco.dat');

%definição das variáveis de entrada e saída:

entrada=M(:,1:3); %colunas de 1 a 3

saida=M(:,4); %coluna 4

%matriz transposta (matriz 500 colunas e 4 linhas), colunas viram linhas pois o matlab ler linha e não
coluna:

entrada=entrada'; % agora tem 500 colunas e 3 linhas

saida=saida'; % 500 colunas e 1 linha

%normalização dos dados entre -1 e 1 (normaliza por linha):

[entradan,ps]=mapminmax(entrada);

[saidan,ts]=mapminmax(saida);

%divisão dos dados já normalizados (treino, validação e teste):

%%treino:

entradatreina=entradan(:,1:342); %reserva as 342 colunas de um arquivo de 500 colunas e 3 linhas

saidatreina=saidan(:,1:342); %matriz 342 colunas e 1 linha

%%validação:

entradavalida=entradan(:,343:424); %matriz 82 colunas e 3 linhas

saidavalida=saidan(:,343:424); %matriz 82 colunas e 1 linha

%%teste/simulação final:

entradateste=entradan(:,425:500); %%matriz 76 colunas e 3 linhas

saidateste=saidan(:,425:500); %matriz 76 colunas e 1 linha

%agora tenho 6 matrizes normalizadas: entradatreina, saidatreina, entradavalida, saidavalida,
entradateste, saidateste

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%ARQUITETURA DA REDE:

%rede feedforward (RNA totalmente conectada e com alimentação direta):

```

```

net=newff(minmax(entradatreina),[X, 1],{'logsig' 'purelin'},'trainbr');

% minmax ler as linhas do banco de dados entradatreina (3 linhas=3 neurônios), pode ser “entradan”
tb

%X é o número de neurônios ocultos iniciais, 1 é o neurônio da saída

%%%%%%%%%%TREINAMENTO DA RNA:

%definição do número máximo de iterações (épocas):

net.trainParam.epochs=10000;

% define o objetivo de erro quadrático (MSE):

net.trainParam.goal=0.000001;

%inicialização dos pesos da rede aleatoriamente (chuta valores iniciais para os pesos e bias):

net=init(net);

%treinamento:

[net,tr]=train(net,entradatreina,saidatreina);

%determinados os pesos e bias da rede e arquivados temporariamente em “net”

%%%%%%%%%%VALIDAÇÃO DA RNA:

%simulação da variável de saída a partir das entradas do conjunto de validação;

Z=sim(net,entradavalida);

%desnormalização da variável de saída;

minsaida=min(saida(end,:));

maxsaida=max(saida(end,:));

S=postmnmx(Z,minsaida,maxsaida); %simulada

R=postmnmx(saidavalida,minsaida,maxsaida); %real

%cálculo do RMSE:

erro=R-S

msev=mse(erro)

RMSEv=(msev)^0.5

%grafico das saídas (real x simulada):

figure(1);

```



```

postreg(R(1,:),S(1,:));
regress(R(:,1), S(:,1))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%TESTE DA RNA:

%simulação da variável de saída a partir das entradas do conjunto de teste;
T=sim(net,entradataeste);

%desnormalização da variável de saída;
minsaida=min(saida(end,:));
maxsaida=max(saida(end,:));

SS=postmnmx(T,minsaida,maxsaida); %simulada com o teste
RR=postmnmx(saidateste,minsaida,maxsaida); %real do teste

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PESOS E BIAS/VIÉS:

%Salvar a rede criada e os parâmetros ajustados no arquivo "rede":
save 'rede' net;

%Para visualizar os pesos e bias finais:
net.IW{1} %pesos entre camadas entrada/intermediária
net.LW{2} %pesos entre camadas intermediária/saída
net.b{1} %bias da camada intermediária
net.b{2} %bias da camada de saída

%Para visualizar o RMSE:
RMSEv %validação

```