

Modelagem técnica-comercial de peças automotivas com rede neural artificial

Technical-commercial modeling of automotive parts with artificial neural network

DOI: 10.34140/bjbv3n1-026

Recebimento dos originais: 20/11//2020

Aceitação para publicação: 20/12/2020

Vinicius Pimentel Martins

Bacharel em Engenharia de Produção pela Universidade Estadual do Norte Fluminense - UENF

Instituição: Universidade Federal Fluminense – UFF

Endereço: Rua Carlos Gomes 85, Jardim Amália, Volta Redonda – RJ, Brasil

E-mail: eng.vpmartins@gmail.com

Renato Figueira Guedes Esteves

Bacharel em Engenharia de Produção pela Universidade Federal Fluminense - UFF

Instituição: Universidade Federal Fluminense – UFF

Endereço: Rua Fernão de Magalhães, 360 - apto 202, Jardim Amália, Volta Redonda - RJ, Brasil

E-mail: renatofge@gmail.com

Matheus Barbosa Cardozo

Bacharel em Engenharia Química pela Universidade Federal do Rio Grande - UFRGS

Instituição: Universidade Federal Fluminense – UFF

Endereço: Rua 164 A, 68, apto 31, Volta Redonda – RJ, BRasil

E-mail: matheusbc1993@gmail.com

Sebastião José de Andrade

Bacharel em Engenharia Mecânica pela UniFOA Volta Redonda - UNIFOA

Instituição: Universidade Federal Fluminense – UFF

Endereço: Rua vinte e seis, 544 Vila Rica, Volta Redonda, RJ, Brasil

E-mail: sebastiaoandrade@outlook.com

Francisco Luiz Marzinotto Júnior

Bacharel em Relações Internacionais pela Universidade Federal Rural do Rio de Janeiro - UFRRJ

Instituição: Universidade Federal do Rio de Janeiro - UFRJ

Endereço: Rua São Paulo, 283. Bairro N. S. do Carmo, Frutal, MG, Brasil

E-mail: franciscomarzinotto@gmail.com

RESUMO

Em mercados cada vez mais competitivos, o fator produtivo interno tem ganhado grande destaque e preocupação por parte das empresas. Para o setor automotivo, é de suma importância realizar previsões de custo e preço de peças, pois tais informações podem balizar de forma vantajosa muitas negociações comerciais e reduzir custos desnecessários. Contudo, devido à complexidade de mensuração, um dos principais problemas que a empresa que cedeu os dados enfrenta na predição é a dificuldade em modelar os dados de preços para precificação de seus produtos. Assim, o objetivo do trabalho é desenvolver e descrever a aplicação de uma Rede Neural Artificial para modelar os

dados de preço de uma montadora de caminhões e ônibus, a fim de obter vantagem competitiva na negociação dos preços. Foram utilizados dados técnicos e comerciais de uma amostra de 100 tubos de borracha da empresa. Na camada de entrada da rede, foram utilizados dados como peso, comprimento, quantidade de dobras e mudanças de diâmetro, e na variável de saída, o preço da peça. Os resultados da correlação entre as variáveis mostraram alta capacidade preditiva do modelo. Concluiu-se que o algoritmo gerado pela RNA alcançou estatísticas satisfatórias para validá-lo como uma ferramenta comercial útil para suportar as negociações de uma montadora de caminhões e ônibus.

Palavras-chaves: Rede Neural Artificial, Indústria Automotiva, Modelagem Linear, Algoritmo Levenberg-Marquardt.

ABSTRACT

In increasingly competitive markets, the internal production factor has gained great prominence and concern from companies. For the automotive sector, it is of utmost importance to make cost and price forecasts of parts, because such information can advantageously guide many commercial negotiations and reduce unnecessary costs. However, due to the complexity of measurement, one of the main problems that the company that provided the data faces in predicting is the difficulty in modeling the price data for pricing its products. Thus, the objective of the work is to develop and describe the application of an Artificial Neural Network to model the price data of a truck and bus manufacturer in order to obtain a competitive advantage in price negotiation. Technical and commercial data from a sample of 100 rubber tubes from the company were used. In the entrance layer of the network, data such as weight, length, amount of folds and diameter changes were used, and in the exit variable, the price of the part. The results of the correlation between the variables showed high predictive capacity of the model. It was concluded that the algorithm generated by RNA achieved satisfactory statistics to validate it as a useful commercial tool to support the negotiations of a truck and bus manufacturer.

Keywords: Artificial Neural Network, Automotive Industry, Linear Modeling, Levenberg-Marquardt Algorithm.

1 INTRODUÇÃO

O processo de precificação de produtos necessita de grande esforço analítico para a real compreensão de suas variáveis diretas. Componentes do preço final de um produto são comumente avaliados conforme os custos de equipamentos, mão de obra, matéria-prima e especificidades do produto acabado, como comprimento, peso, categoria e funcionalidades.

Com o mercado atingindo níveis de competitividade cada vez mais altos, a preocupação com a redução de desperdícios e custos desnecessários ao longo da cadeia produtiva é fator crítico para a sobrevivência empresarial. Empresas como as do ramo automotivo se destacam nos investimentos em tecnologias de otimização de processos. Por necessitarem de informações confiáveis em suas diferentes interfaces, o setor comercial demanda parte das preocupações com a precisão dos dados. Estudos mostram que informações refinadas podem melhorar os processos de negociação, bem

como os seus resultados finais (ESSA; DEKKER; GROOT, 2018). Naturalmente, com boas negociações, as indústrias conseguem prosperar em seus mercados.

Apesar do constante avanço entre os mercados internacionais nos últimos anos, as indústrias de bens de consumo apresentam uma maior concentração de seu *market share* no mercado interno (COŞAR et al., 2018). Tal constatação se confirma na indústria automotiva brasileira. No ano de 2019, o único item que possuía perspectivas negativas para o segmento era a balança comercial. (AUTODATA, 2019).

Com a expectativa de recuperação econômica no Brasil, as montadoras se dedicam a encontrar soluções para manter os custos de seus produtos sob controle. A análise dos atributos e dos processos produtivos de seus componentes é uma saída para o controle eficaz dos custos (BOCK; PÜTZ, 2017), principalmente para aqueles que são passíveis de negociação. Isso se dá devido ao suporte às transações comerciais com informações mensuráveis e relevantes para compradores e vendedores.

Entretanto, devido à alta complexidade de mensuração, um dos principais problemas que a empresa que cedeu os dados para o presente estudo enfrenta é a dificuldade em modelar os dados de preços para precificação de seus produtos de forma mais efetiva. Assim, faz-se necessária a aplicação de ferramentas de auxílio à tomada de decisão direcionadas para a resolução de problemas avançados.

Diferentes setores produtivos buscam recursos que venham a auxiliar suas decisões estratégicas, permitindo uma melhor tratativa nos dados e explicações mais fiéis à realidade corporativa. Em destaque entre os modelos com maior crescente em relevância e pesquisas da atualidade, as Redes Neurais Artificiais (RNA) vêm sendo cada vez mais utilizadas em função de seu desempenho na resolução de problemas complexos a partir do reconhecimento de padrões (MARTINELLI, 1999).

Assim, o objetivo do presente trabalho é desenvolver e descrever a aplicação de uma Rede Neural Artificial para modelar os dados de preço de uma amostra de 100 tubos de borracha de uma montadora de caminhões e ônibus, correlacionados com informações coletadas de seus respectivos desenhos técnicos, a fim de obter vantagem competitiva na negociação dos preços dos tubos. Ao final, serão analisados os resultados do *output* da rede, com ênfase na capacidade preditiva do modelo gerado e nos ganhos estratégicos que a ferramenta pode oferecer à empresa envolvida no estudo.

2 REFERENCIAL TEÓRICO

2.1 REDES NEURAIS ARTIFICIAS (RNA)

2.1.1 Definição

Redes Neurais Artificiais são modelos auto adaptativos que usam como base argumentos sobre um problema conhecido e buscam a partir de exemplos, aprender relações que possam servir de solução (ZHANG; PATUWO; HU, 1998). Com foco no desenvolvimento de modelos matemáticos baseados no comportamento cerebral, esta área tem atraído muita atenção de diversos tipos de profissionais, como neurocientistas, físicos, engenheiros computacionais, psicólogos e filósofos, todos com o mesmo propósito, o de simular um modelo cerebral que compreenda e resolva problemas de cunho real. Apesar da grande gama de problemas propostos com modelos de RNA, todos tratam o neurônio como unidade básica de processamento, cuja caracterização é dada pelo seu nível de atividade, ou seja, seu estado de polarização, seus dados de entrada (sinapses) e suas conexões de saída (axônios).

2.1.2 Funcionamento

Cada parte do conjunto é escrito numericamente, onde pesos, chamados de forças sinápticas, transcrevem a intensidade com que a variável exerce no nível de ativação da unidade. A forma como esse modelo se expressa, pode ser explicitado na Equação 1 abaixo, onde a soma linear produz um valor de ativação para a unidade i no tempo t :

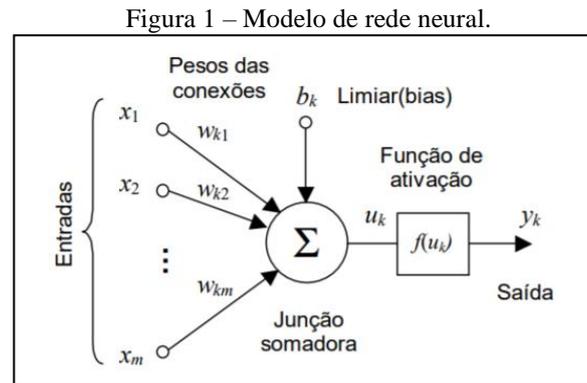
$$n_i(t) = \sum_j w_{ij} x_j(t) + \beta_i \quad (1)$$

Onde W_{ij} representa a força da conexão entre a unidade i e j , β_i representa o *bias*, ou erros associados ao modelo, e x_{ij} o valor de saída para cada j (RUMELHART; WIDROW; LEHR, 1994).

No neurônio artificial, os dados de entrada são recebidos através de canais nas sinapses, que permitem a entrada e saída de íons. A partir da integração dos sinais de entrada, um potencial de membrana determina se haverá sinal de saída, chamado de potencial de ação. Este potencial libera a informação se o potencial de membrana for superior a um determinado limiar, ou *threshold*, na literatura, e que posteriormente com a junção somadora irá processar a informação (HAYKIN, 1999). Assim, as principais partes do neurônio artificial são:

- Sinapses com seus pesos associados;
- Junção somadora;
- Função de ativação.

A Figura 1 apresenta o modelo descrito por Haykin (1999):



Fonte: Adaptado de Haykin (1999).

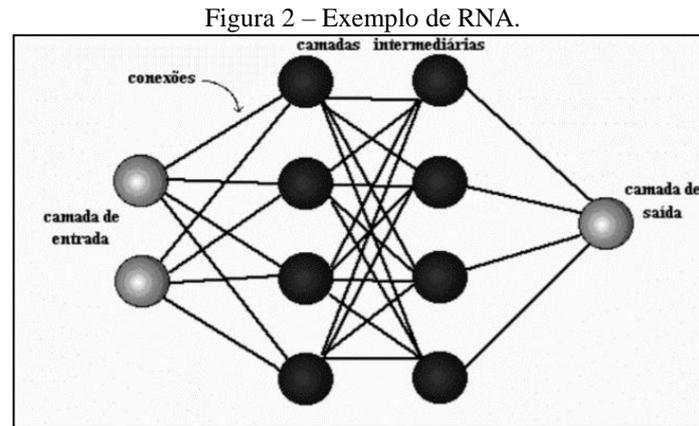
2.1.3 Treinamento da rede de aprendizagem

Neurônios artificiais componentes da rede, com sua capacidade de aprendizagem e “treinamento”, processamento de informações e análise de erros, recebem um *input* de informações, alteram seus parâmetros algorítmicos para diminuir a diferença entre valores previstos e valores reais, e emitem um *output* de informação (DE BARCELOS *et al.*, 2008; revisado em RAUBER, 2005). Os pesos das conexões quando colocados em treinamento, se ajustam de acordo com os padrões a eles apresentados, fazendo com que o processo de aprendizagem ocorra através de exemplos. A sistemática necessária para executar adequadamente o processo passa pelas diversas camadas da rede, diferindo restrições impostas pelo algoritmo dos dados reais imputados (DARTORA, 2003).

Ainda de acordo com Dartora (2003), as camadas são usualmente classificadas em 3 grupos:

- Camada de entrada: apresenta padrões à rede;
- Camadas intermediárias ou ocultas: processa os dados através das características do algoritmo;
- Camada de saída: apresenta o resultado final.

A representação da rede neural artificial contemplando suas camadas é ilustrada na Figura 2.



Fonte: Dartora (2003).

Matematicamente a saída do neurônio k pode ser descrita pela Equação 2:

$$y_k = f(u_k) = f\left(\sum_{j=1}^m w_{kj}x_j + b_k\right) \quad (2)$$

Considerando x_0 um sinal de entrada cujo valor é 1 e peso associado $w_{k0} = b_k$. Segundo Tubb (1993), diversos são os benefícios de se usar modelos de Redes Neurais Artificiais, dentre eles:

- Robustez: tratam a falha gradualmente, oferecendo respostas adequadas ao longo do tempo;
- Aprendizado contínuo: sua capacidade de aprender com a experiência aperfeiçoa seus resultados e se adapta em novos cenários;
- Processamento acelerado: unidades de processamento operam em paralelo, permitindo velocidades maiores de resposta se comparado à métodos tradicionais.

Desde 1980, diversos modelos de RNA foram propostos, mas possivelmente o modelo mais influente seja o *multi-layer Perceptron* (MLP) ou *Perceptron* de multicamada (ZHANG; PATUWO; HU, 1998). O tipo de problema a ser solucionado normalmente é o responsável pela definição das restrições em relação aos modelos de algoritmos de redes possíveis (BRAGA; CARVALHO; LUDERMIR, 2003). Tal modelo que baseia no aprendizado baseado no algoritmo de retro propagação, é o modelo de RNA mais utilizado em áreas com foco na definição de padrões, previsão de séries temporais e processamento de fala (TÁPIA, 2000).

2.2 ALGORITMO LEVENBERG–MARQUARDT

Considerado um método melhorado do algoritmo de *Gauss-Newton*, visto que este não garante a convergência do modelo, o algoritmo *Levenber-Marquardt* limita o tamanho de δ para prevenir que quando se está muito além do ponto de mínimo, δ seja muito grande. Para isto, uma

função mérito é definida com um fator de amortecimento λ , que regula a dimensão de δ , como mostra a Equação 3:

$$\varphi = \phi + \lambda |\delta|^2 \quad (3)$$

Assim, ϕ divide os mesmos pontos de mínimos que φ , apenas respeitando a igualdade $\delta=0$, além de ser proporcional ao quadrado da dimensão de δ , sendo este minimizado ao longo do processo com φ . Conseqüentemente, o tamanho do vetor de mudança fica impedido de crescer, sendo este o problema principal do Algoritmo de *Gauss-Newton*. Desta forma, a expressão para o vetor gradiente de ϕ pode ser representada pelas Equações 4, 5, 6 e 7.

$$\frac{\partial \varphi}{\partial x_j} = \frac{\partial \phi}{\partial x_j} + \frac{\partial}{\partial x_j} \lambda |\delta|^2 = 2 \left(\sum_{i=1}^m \frac{\partial \epsilon_i}{\partial x_j} + \lambda \delta_j \right) \quad (4)$$

$$\therefore \nabla \varphi = 2(J^T \varepsilon + \lambda \delta) \quad (5)$$

$$\therefore \nabla \varphi = 2(J^T (\varepsilon_0 + J\delta) + \lambda \delta) = 0 \quad (6)$$

$$\therefore \delta = -\frac{1}{2}(J^T J + \lambda I)^{-1} G_0 \quad (7)$$

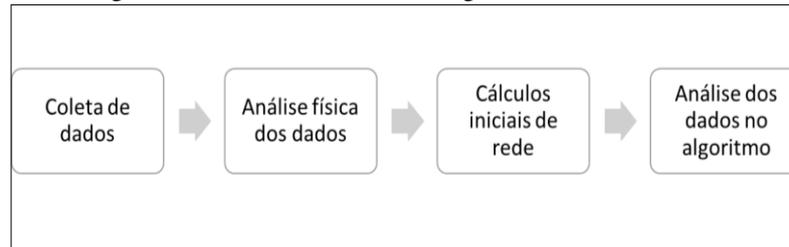
Assim, o Algoritmo de *Levenberg-Marquardt* pode ser entendido como um algoritmo robusto e com rápida convergência (KOH; CHEONG, 2018). Por possuir tais características, foi o algoritmo escolhido para aplicação na rede neural do presente projeto.

3 METODOLOGIA

Este trabalho é de natureza aplicada, pois sua finalidade é gerar conhecimento para resolver um problema específico. O objetivo da pesquisa pode ser classificado como exploratório, pois constrói e expõem hipóteses sobre o problema. Sua abordagem é quantitativa, utilizando medidas padronizadas e sistemáticas, além de estabelecer estatísticas dos dados. Com relação ao método, este constitui como estudo de caso, pois trata de um caso único, gerando descoberta de um determinado contexto (NASCIMENTO, 2016).

Para entender a forma com que as diversas variáveis que compõem a precificação do produto afetam o valor final de tubos flexíveis, dividiu-se o trabalho em quatro etapas, conforme Figura 3, onde o *software* MATLAB® foi utilizado na execução do estudo.

Figura 3 – Procedimento metodológico utilizado no estudo.



Fonte: Autores (2020).

Na primeira etapa foram coletados 100 desenhos técnicos junto à empresa automotiva em estudo. Estes foram verificados conforme sua aderência à proposta de modelagem comercial via RNA. Desta forma, foram definidas as principais variáveis a serem estudadas, onde foram levados em consideração a qualidade dos dados e a relevância da variável em relação ao produto. Na terceira etapa foi desenhada a estrutura da rede neural, realizando os procedimentos iniciais de verificação do algoritmo para rodar o código e simular os diferentes cenários. Por fim, foram realizados os testes estatísticos, onde foi avaliada a aderência do modelo de previsão dos resultados para sua possível validação.

4 DESENVOLVIMENTO

4.1 MODELAGEM DA REDE NEURAL ARTIFICIAL

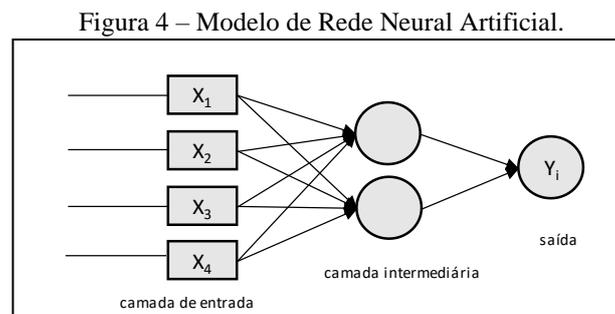
Com o objetivo de obter vantagem competitiva na negociação dos preços dos tubos, a rede neural foi utilizada para modelar e verificar a correlação entre as variáveis de entrada (atributos físicos e de fabricação) com a variável de saída (preço da peça). Desta forma, foi possível verificar se as variáveis possuíam certa correlação e quais entradas exerciam maior influência sobre a saída. Dentre os parâmetros utilizados, ou seja, as variáveis de entrada, foram considerados os seguintes itens:

- Peso;
- Comprimento;
- Quantidade de dobras;
- Quantidade de mudanças de diâmetro.

Para determinar o número de neurônios na camada intermediária, levou-se em conta a média geométrica de n variáveis de entrada em relação a m variáveis de saída, cujo valor é dado na Equação 8:

$$\text{neurônios na camada intermediária} = (n.m)^{1/2} \quad (8)$$

Assim, com 4 variáveis de entrada e 1 variável de saída, o número de neurônios na camada intermediária tem valor igual a 2, conforme pode ser visto na Figura 4 abaixo que representa o modelo de RNA adotado no presente estudo.



A próxima etapa levou em consideração a quantidade de parâmetros da rede. Para se obter uma análise confiável, era necessário no mínimo 81 vetores, porém foram utilizados um total de 100 vetores. O código para o rodar o algoritmo de *Levenberg-Marquardt* leva em conta entradas normalizadas, calculadas pela Equação 9:

$$XN_i = (x_i - x_{imin}) / (x_{imax} - x_{imin}) \quad (9)$$

Logo, todos os valores da base de dados se ajustarem em valores entre 0 e 1. Em seguida, a base de dados normalizada foi convertida em um arquivo de texto e imputada no código da rede neural, executada com o auxílio do *software* MATLAB®. O código-fonte se encontra no Anexo I do presente trabalho.

4.2 TESTE DOS DADOS PARA O MODELO COM TUBOS DE BORRACHA

Neste modelo, foi utilizada uma amostra tubos de características variadas em relação a peso (entre 226 e 833 gramas), comprimento (entre 304 e 1091 milímetros), quantidade de dobras (entre 1 e 5) e quantidade de mudanças de diâmetro do tubo (entre 0 e 4). Ao todo, foram utilizados 100 desenhos de peças, que viraram dados para as variáveis de entrada.

Para as mesmas peças da amostra, foram coletados os valores de preço de compra dos pedidos mais recentes de cada item. Os valores foram consultados e extraídos com o auxílio do banco de dados da empresa envolvida.

4.3 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Os resultados apresentados pelo algoritmo mostraram que foram necessárias 132 iterações até se chegar ao ponto ótimo. O treinamento apresentou $R=0,996$ para os 4 estados analisados, onde foram usados como *target* os seguintes *outputs*, presentes na Tabela 1.

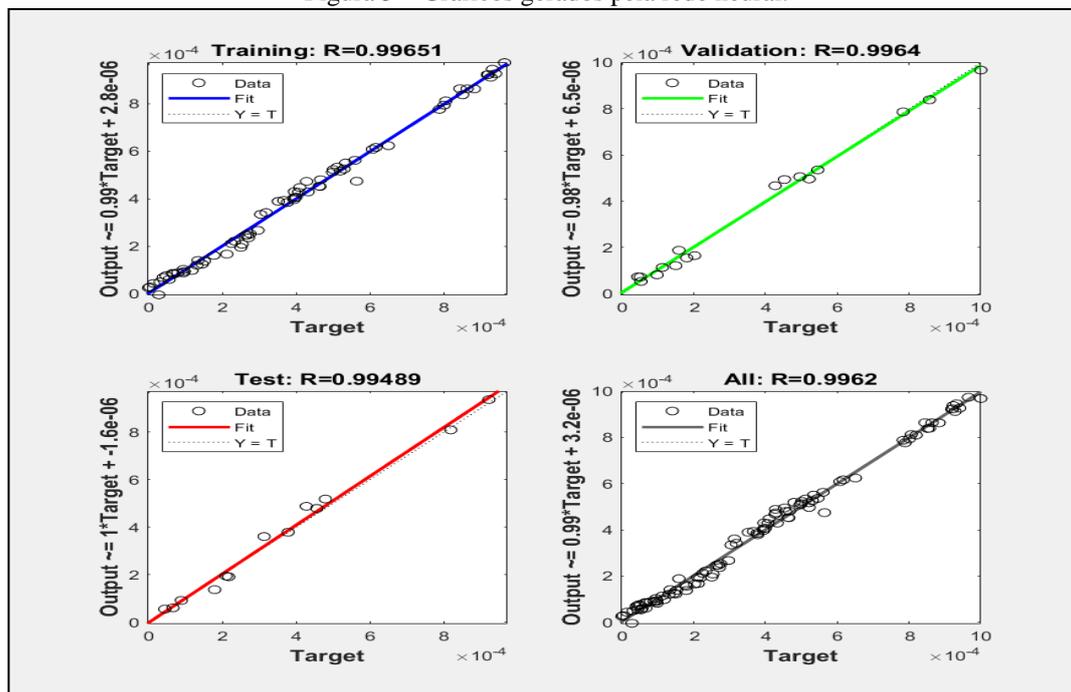
Tabela 1 – *Output* das iterações da rede neural.

Análise	Target	R
1	$0,99*Target + 2,8e-06$	0,99651
2	$0,98*Target + 6,5e-06$	0,9964
3	$1,00*Target + 1,6e-06$	0,99489
4	$0,99*Target + 3,2e-06$	0,9962

Fonte: Autores (2020).

Foi possível verificar um R (coeficiente de correlação linear) com valor próximo a 1, o que demonstra uma correlação quase perfeita positiva entre as variáveis. Esta propriedade é observada através da Figura 5, na qual verificam-se os gráficos gerados pelo MATLAB® ao final da execução das iterações da rede neural.

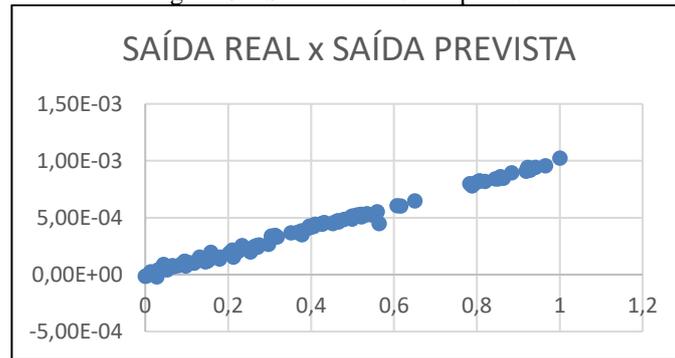
Figura 5 – Gráficos gerados pela rede neural.



Fonte: Autores (2020).

Os *outputs* produzidos, ou seja, o Y_e que o algoritmo gerou, foi comparado com a saída real para verificação do modelo, obtendo os seguintes resultados plotados no gráfico apresentado na Figura 6.

Figura 6 – Saída real x saída prevista.



Fonte: Autores (2020).

A partir dos dados usados na plotagem do gráfico, foi elaborada a Tabela 2 com as informações específicas da Regressão Linear entre as saídas.

Tabela 2 – Resultados estatísticos.

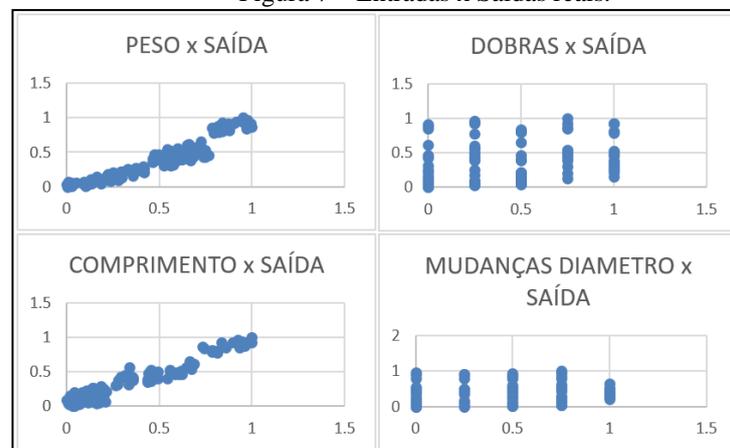
Estatística de regressão	R
R múltiplo	0,996717
R-Quadrado	0,993445
R-Quadrado ajustado	0,993377
Erro padrão	2,35e-05
Observações	99

Fonte: Autores (2020).

Foi verificada, assim, a forte correlação entre os dados reais e os previstos pelo algoritmo após o processo de treinamento. O modelo atingiu a marca de *R-Quadrado* de 0,993445.

Não obstante, foi feita uma análise da correlação entre as variáveis de entrada com as saídas reais para se verificar se elas já explicavam o modelo. Os dados desta última análise são apresentados na Figura 7, com a combinação dos quatro gráficos.

Figura 7 – Entradas x Saídas reais.



Fonte: Autores (2020).

Como visto na Figura 7, a variável peso e comprimento possuem relação direta com o preço final da peça, excluindo as demais variáveis do resultado final.

5 CONSIDERAÇÕES FINAIS

Concluiu-se que o modelo consegue explicar de forma coerente os dados encontrados em situações reais e prevê com grande precisão saídas futuras. Pela presença constante de erros inerentes ao processo, os passos seguidos ao longo de toda a análise de rede se mostram de fundamental importância na assertividade dos resultados finais.

Deve-se também analisar a força da correlação das variáveis de entrada para que não influenciem ou até expliquem o modelo todo por si só, ou seja, caracterizando relações diretas entre seus *inputs* e *outputs*. Desta forma, o modelo apresentará maior robustez e complexidade.

É comum poucas variáveis impactarem de forma conclusiva no *output* produzido pela rede, impedindo a justificativa para o uso de redes neurais de solução de problemas de alta complexidade.

Para trabalhos futuros, sugere-se uma maior coleta de dados para se realizar a amostragem, além da quantificação de um número maior de variáveis. Este procedimento permitirá obter uma melhor abrangência do nicho do produto inserido e o ranqueamento de variáveis significativas e desprezáveis de consideração.

REFERÊNCIAS

AUTODATA. *Guia Automotivo Brasileiro*. Ano 2. São Paulo, 2019.

BOCK, S.; PÜTZ, M. Implementing Value Engineering based on a multidimensional quality-oriented control calculus within a Target Costing and Target Pricing approach. *International Journal of Production Economics*, v. 183, p. 146–158, 2017.

BRAGA, A. P.; CARVALHO, A. C. P. L. F.; LUDERMIR, T. B. Redes neurais artificiais. In: *Sistemas Inteligentes: Fundamentos e Aplicações*. 1ª ed. Barueri: Manole, 2003.

COŞAR, A. K.; GRIECO, P. L. E.; LI, S.; TINTELNOT, F. What drives home market advantage? *Journal of International Economics*. v. 110, p. 135–150, 2018.

DARTORA, G. A. *REDES NEURAS ARTIFICIAIS: APLICAÇÃO NA TIPIFICAÇÃO DE CARCAÇA DE SUÍNOS*. 105f. Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-Graduação em Ciências da Computação, Universidade Federal de Santa Catarina - UFSC, Florianópolis, 2003.

DE BARCELOS TRONTO, I. F.; DA SILVA, J. D. S.; SANT'ANNA, N. An investigation of artificial neural networks based prediction systems in software project management. *Journal of Systems and Software*, v. 81, n. 3, p. 356–367, 2008.

ESSA, S. A. G.; DEKKER, H. C.; GROOT, T. L. C. M. Your gain my pain? The effects of accounting information in uncertain negotiations. *Management Accounting Research*, v. 41, n. April, p. 20–42, 2018.

HAYKIN, S. S. *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, Nova Jersey: Prentice Hall, 1999.

KOH, J. M.; CHEONG, K. H. Journal of Electron Spectroscopy and Automated electron-optical system optimization through switching Levenberg – Marquardt algorithms. *Journal of Electron Spectroscopy and Related Phenomena*, v. 227, n. March, p. 31–39, 2018.

MARTINELLI, E. *Extração de conhecimento de redes neurais artificiais*. 1999. 113f. Dissertação (Mestrado em Ciências da Computação) – Curso de Pós-graduação em Ciências da Computação, Universidade Federal de São Paulo, São Paulo, 1999.

NASCIMENTO, F. P. *Metodologia da Pesquisa Científica: teoria e prática – como elaborar TCC*. Brasília: Thesaurus, 2016.

RUMELHART, D. E.; WIDROW, B.; LEHR, M. A. The Basic Ideas in Neural Networks. *Models of the brain and evolution*, v. s/n, n. s/n, 1994.

TÁPIA, M. *Redes neurais artificiais: uma aplicação na previsão de preços de ovos*. 122f. Dissertação (Mestrado em Ciências da Computação) - Programa de Pós-Graduação em Ciências da Computação, Universidade Federal de Santa Catarina - UFSC, Florianópolis, 2000.

TUBB, N. R. A Development Path to Success in Neural Computing, *Expert Systems Applications*, v. 9, n. 5. p. 5-9, 1993.

ZHANG, G.; PATUWO, B. E.; HU, M. Y. Forecasting with artificial neural networks : The state of the art. *International Journal of Forecasting*, v. 14, p. 35–62, 1998.

ANEXO I – PROGRAMAÇÃO UTILIZADA NO TRABALHO

```
clear;
M=load('Base.txt');
    %M=INT(1000*(((X-Xmin)/(Xmax-Xmin))*0.9+0.05))
    %=INT(1000*(((A1-$A$184)/($A$185-$A$184))*0.9+0.05))
M=0.001*M';
    %transposta os dados
X=M(1:4, 1:100); %valores de entrada da posicao 1 ate a 140, valor de treinamento
Y=M(5:5, 1:100); %valores de saida da posicao 1 ate a 140, valor de treinamento
net.numLayers=3; % sem considerar a entrada (camada intermediaria e saida)
% 4=(entrada + saida)/2, numero de neuronios
% 1=numero de saidas
% logsig(sigmoidal) e purelin(linear) sao funcoes de ativacao (funcao de transferencia da camada)
% trainlm(Levenberg-Maguardt) algoritmo de minimizacao (funcao de treinamento)
net=newff(minmax(X),[4, 1],{'logsig','purelin'},'trainlm');
%divide a rede e pega os dados randomizado de forma automatica
net.divideFcn = 'dividerand';
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 20/100;
%net.divideParam.testRatio = 15/100;
net.trainParam.epochs=500; %periodo de treinamento, numero de iteracoes a serem realizadas
net.trainParam.show=25; %periodo a cada mostra de resultado
net.trainParam.goal=1e-25; %convergencia aceitavel, minimo erro tolerado
net.initFcn='initlay'; %configura demais layers para iniciar de acordo com a funcao initlay
net.performFcn='sse';
net.trainParam.min_grad=1e-100; %performance minima do gradiente
net.trainParam.mu_max=1e+100;
net=init(net); %inicia a rede neural
%executa o treinamento
[net,tr]=train(net,X,Y);
Ye=sim(net,X);
plot(Y,Ye,'r.o');
Yve=sim(net,Xv);
plot(Yv,Yve,'b.o');
%resultado dos pesos e bias
W1=net.LW{1};
W2=net.LW{2};
B1=net.b{1};
B2=net.b(2);
```