

UNIVERSIDADE SANTA CECÍLIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
MESTRADO EM ENGENHARIA MECÂNICA

MARCOS CARNEIRO RODRIGUES

**REDE NEURAL ARTIFICIAL PARA CONSISTENTE APLICADA NA
CRIAÇÃO DE UM MODELO COMPUTACIONAL PARA ESTIMATIVA
DE EMISSÕES DE ÓXIDOS DE NITROGÊNIO (NOX) EM SISTEMAS DE
COMBUSTÃO INDUSTRIAL**

SANTOS/SP

2024

MARCOS CARNEIRO RODRIGUES

**REDE NEURAL ARTIFICIAL PARA CONSISTENTE APLICADA NA
CRIAÇÃO DE UM MODELO COMPUTACIONAL PARA ESTIMATIVA
DE EMISSÕES DE ÓXIDOS DE NITROGÊNIO (NOX) EM SISTEMAS DE
COMBUSTÃO INDUSTRIAL**

Dissertação apresentada a Universidade Santa Cecília como parte dos requisitos para obtenção de título de mestre no Programa de Pós-Graduação em Engenharia Mecânica, sob a orientação do Prof. Dr. João Inácio da Silva Filho

SANTOS/SP

2024

511.31
R611r

Rodrigues, Marcos Carneiro.

Rede neural artificial paraconsistente aplicada na criação de um modelo computacional para estimativa de emissões de óxidos de nitrogênio (nox) em sistemas de combustão industrial / Marcos Carneiro Rodrigues.

2024.

76 f.

Orientador: Dr. João Inacio da Silva Filho.

Dissertação (Mestrado) - Universidade Santa Cecília, Programa de pós-graduação em Mestrado em Engenharia Mecânica, Santos, SP, 2024.

1. Lógica Paraconsistente. 2. Redes Neurais Artificiais. 3. Emissões Industriais. 4. Método de Levenberg-Marquardt. I. Silva Filho, João Inacio da. II. Rede neural artificial paraconsistente aplicada na criação de um modelo computacional para estimativa de emissões de óxidos de nitrogênio (nox) em sistemas de combustão industrial.

DEDICATÓRIA

Dedico este trabalho à minha esposa Juliana, pela imensa parceria e amor demonstrado durante todo esse tempo em que estamos casados.

AGRADECIMENTOS

Eu aprecio e agradeço profundamente por este trabalho:

Jeová, fonte de toda sabedoria e poder, por conduzir e salvar minha vida e iluminar meu caminho em todos os momentos.

À minha amada esposa Juliana, por seu amor inabalável, paciência inesgotável e apoio incondicional durante esta jornada desafiadora. Sem ela este trabalho nunca seria possível.

Aos meus pais, cujo amor, sacrifício formaram uma base sólida sobre a qual construí meus sonhos e vitórias.

Ao meu estimado mentor, Dr. João Inácio, cuja sábia orientação, dedicação inabalável e incentivo contínuo foram essenciais para o meu crescimento acadêmico e pessoal.

Ao meu amigo Josué Lima da Silva, pela amizade genuína, palavras de incentivo e apoio nos momentos mais desafiadores deste trabalho.

Para meu amigo Rubens Lacerda de Sá, na qual sua camaradagem, apoio mútuo e risadas compartilhadas levaram alegria e leveza aos dias mais intensos.

À Universidade Santa Cecília (Unisantia), por proporcionar o ambiente acadêmico propício para meu crescimento intelectual e profissional, bem como pelo suporte de valor incalculável dos servidores durante o período do mestrado

Aos colegas da Petrobras, cuja colaboração e experiências compartilhadas foram inestimáveis durante este processo.

A todos vocês, minha mais profunda gratidão. Este trabalho é uma homenagem ao apoio generoso e à inspiração que vocês proporcionaram. Que nossos laços de amizade e amor continuem a fortalecer-se e a enriquecer nossas vidas.

RESUMO

As emissões industriais, uma preocupação significativa no contexto ambiental atual, referem-se aos resíduos liberados durante os processos de produção e fabricação em diversas indústrias, como as indústrias química, petroquímica e metalúrgica, entre outras. As emissões dos óxidos de nitrogênio (NOx) são uma família de compostos químicos que incluem óxido nítrico (NO) e dióxido de nitrogênio (NO₂), que são formados principalmente durante a combustão de combustíveis fósseis, como carvão, petróleo e gás natural, usados em motores de veículos, instalações industriais e outros processos de queima. Neste trabalho, para estimar as intensidades das emissões dos óxidos de nitrogênio (NOx) em um forno industrial do tipo Cilindro Vertical de 6 queimadores, são utilizadas técnicas com algoritmos fundamentados em Lógica Paraconsistente (LP), que é um ramo da lógica que lida com situações em que há contradições e inconsistências nos dados. Ao contrário da lógica clássica, que assume que todas as proposições são, ou verdadeiras ou falsas, a lógica paraconsistente permite lidar com dados de informações incompletas como os gerados pelas emissões industriais de NOx que podem ser simultaneamente consideradas verdadeiras e falsas, ou cuja veracidade não pode ser determinada de forma totalmente conclusiva. Para aumentar o índice de eficiência na detecção e estimativa das emissões dos óxidos de nitrogênio (NOx) foram formadas estruturas de algoritmos paraconsistentes compondo modelos de Redes Neurais Artificiais onde neurônios artificiais paraconsistentes processam e transmitem informações através de conexões ponderadas. Na estrutura computacional construída nesta pesquisa foi utilizado uma nova abordagem para a função aditiva dos Neurônios em uma Rede Neural Artificial Paraconsistente (RNAP) onde se utilizou o método de Levenberg-Marquardt para otimizar o treinamento através do processo iterativo de ajuste dos parâmetros e minimizar a função de erro. Ao final, o desempenho da RNAP na detecção das intensidades das emissões dos óxidos de nitrogênio (NOx) foi comparado com o de uma rede Neural Artificial com função de ativação log-sigmoid (RNA Log-Sig). Os resultados mostraram que a RNAP apresentou parâmetros de desempenho superiores em relação a RNA de ativação logística, destacando a eficácia da abordagem proposta para a criação de estruturas computacionais que lidam com problemas complexos de modelagem e previsão, como as emissões industriais de NOx.

Palavras-chave: Lógica Paraconsistente, Redes Neurais Artificiais, Emissões industriais, Método de Levenberg-Marquardt.

ABSTRACT

Industrial emissions, a significant concern in the current environmental context, refer to waste released during production and manufacturing processes in various industries, such as the chemical, petrochemical and metallurgical industries, among others. Nitrogen oxide (NO_x) emissions are a family of chemical compounds that include nitric oxide (NO) and nitrogen dioxide (NO₂), which are formed primarily during the combustion of fossil fuels, such as coal, oil and natural gas, used in vehicle engines, industrial installations and other burning processes. In this work, to detect and estimate the intensities of nitrogen oxides (NO_x) emissions in an industrial oven of the Vertical Cylinder type with 6 burners, techniques with algorithms based on Paraconsistent Logic (PL), which is a branch of logic that deals with situations in which there are contradictions and inconsistencies in the data. Unlike classical logic, which assumes that all propositions are either true or false, paraconsistent logic allows us to deal with incomplete information data such as those generated by industrial NO_x emissions that can be simultaneously considered true and false, or whose veracity cannot be determined completely conclusively. To increase the efficiency rate in detecting and estimating nitriding oxide (NO_x) emissions, paraconsistent algorithmic structures were formed by composing Artificial Neural Network models where paraconsistent artificial neurons process and transmit information through weighted connections. The computational structure built in this research used a new approach to the additive function of Paraconsistent Artificial Neural Network (PANNet) Neurons where the Levenberg-Marquardt method was used to improve training through the iterative process of adjusting parameters and minimizing the error function. Finally, the performance of PANNet in detecting the intensities of nitrogen oxide (NO_x) emissions was compared with that of Artificial Neural Network with a log-sigmoid activation function (ANN Log-Sig). The results show that PANNet included superior performance parameters compared to logistics activation ANN Log-Sig, highlighting the effectiveness of the proposed approach for creating computational frameworks that deal with modeling and forecasting problems such as industrial NO_x emissions.

Keywords: Paraconsistent Logic, Artificial Neural Networks, Industrial Emissions, Levenberg-Marquardt Method.

LISTA DE FIGURAS

Figura 1. Componentes de um neurônio. (a) Representação do neurônio humano, (b) modelo do neurônio artificial.	19
Figura 2. Diagrama de Hasse com ε_T , constituído a partir do par GC,GCT.....	24
Figura 3. Semicírculos contendo o conjunto de pontos " ε " _ " τ ", para D=1.	25
Figura 4. Projeção de GCR caso D não estivesse limitado a 1.	26
Figura 5. Intervalo de Certeza para o ponto " ε " _ " τ " representado.	27
Figura 6. Célula Neural Artificial Paraconsistente de aprendizagem (CNAPap).	30
Figura 7. Comportamento de aprendizagem para (a) CNAPap, (b) CNAPapctx	31
Figura 8. Neurônio Artificial Paraconsistente (NAP) composto de CNAPps.....	32
Figura 9. Neurônio Artificial Paraconsistente (NAP) com CNAPs de saída linear μ_E em camadas intermediárias.....	32
Figura 10. Forno de destilação atmosférica de petróleo	35
Figura 11. Célula neural paraconsistente com entradas adaptadas.....	39
Figura 12. Comportamento de $\partial\mu_{ER}/\partial\mu_E$, considerando a função por partes montada a partir das equações (47) E (48).....	43
Figura 13- Comparação entre as derivadas de " μ " _ " ER ", com relação a μ_E μ : (a) entrada ímpar. (b) entrada par.	43
Figura 14. Arquitetura da RNAP utilizada.	45
Figura 15. Comparação do Comportamento de Redes Neurais com Funções de Ativação Log-Sigmoide e RNAP.....	47
Figura 16. Resultados de desempenho da RNAP: (a) comparação do MSE; (b) histograma de erros.	48
Figura 17. Validação da Redes.	50
Figura 18. Comparação da RNAP e alvo em sequência temporal.....	50
Figura 19. Comparação da saída para diferentes valores de entrada. (a):Log-sig. (b): RNAP.....	53
Figura 20. Relação entre distribuição de Espécies químicas minoritárias em relação a razão Combustível/Ar.	54

LISTA DE TABELAS

Tabela 1 – Correlação entre dados normalizados de um forno industrial	37
Tabela 2 - Valores de MSE para diferentes tamanhos da camada oculta	49
Tabela 3 - Pesos finais de treinamento da RNAP: (a) Entrada-Oculta. (b) Oculta-saída	51
Tabela 4 - Relação da influência das entradas propagadas pela RNAP	52

LISTA DE ABREVIATURAS E SIGLAS

NO_x	- Óxidos de Nitrogênio
CO₂	- Dióxido de Carbono
CH₄	- Metano
VOC	- <i>Volatile Organic Compounds</i> (Compostos Orgânicos Voláteis)
CO	- Monóxido de Carbono
TCA	- Temperatura de chama adiabática
SP	- Setpoint (valor de referência)
T-NOX	- <i>Thermal Nox</i> (Nox Térmico)
LPA2v	- Lógica Paraconsistente Anotada de dois valores
QUPC	- Quadrado Unitário de Plano Cartesiano
μ	- Grau de Evidência Favorável
λ	- Grau de Evidência Desfavorável
G_c	- Grau de Certeza
G_{CT}	- Grau de Contradição
G_{CR}	- Grau de Certeza Real
T	- Estado Lógico Contraditório
⊥	- Estado Lógico Paracompleto
μ_E	- Grau de Certeza Resultante
μ_{ER}	- Grau de Certeza Resultante Real
μ_{ECT}	- Grau de Contradição Resultante
Φ	- Intervalo de Certeza
ALGO-NAP	- Algoritmo Nó de Análise Paraconsistente
NAP	- Neurônio Artificial Paraconsistente
RNAP	- Rede de Análise Paraconsistente
CNAP	- Célula Neural Artificial Paraconsistente
CNAP_p	- Célula Neural Artificial Paraconsistente Padrão
CNAP_{ap}	- Célula Neural Artificial de Aprendizagem
CNAP_{apctx}	- Célula Neural Artificial Paraconsistente de Aprendizagem com Extração do Efeito da Contradição
V_{scc}	- Valor Superior de Controle de certeza
V_{icc}	- Valor Inferior de Controle de certeza
V_{sccT}	- Valor Superior de Controle de Contradição
V_{icct}	- Valor Inferior de Controle de Contradição
F_A	- Fator de Aprendizagem

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	Importância.....	13
1.2	Problemática	14
1.3	Fundamentação teórica.....	15
1.3.1	Sistemas de Combustão e Fornos Industriais	15
1.3.2	Redes Neurais.....	19
1.1.1	Método de Levenberg Marquadt	20
1.1.2	Lógica Paraconsistente	21
1.1.2.1	Lógica Paraconsistente Anotada com anotação de dois Valores...22	
1.1.2.2	Redes de Análise Paraconsistente (RAP) e Redes Neurais Artificiais Paraconsistentes (RNAP)	28
1.2	Justificativa	33
1.3	Objetivo	34
1.3.1	Objetivo geral.....	34
1.3.2	Objetivos específicos	34
2	MATERIAIS E MÉTODOS.....	35
2.1	Materiais	35
2.2	Métodos.....	37
2.2.1	Estrutura no Neurônio Artificial Paraconsistente Proposto.....	37
2.2.2	Estrutura da RNAP	43
2.2.3	Método de Levenberg-Marquadt associado a RNAP proposta	44
3	RESULTADOS E DISCUSSÃO.....	46
4	CONCLUSÕES.....	56
	REFERÊNCIAS.....	58
	APENDICE A: DEMONSTRAÇÃO DE μECT PARA N ENTRADAS.....	61
	APENDICE B: CÉLULA NEURAL DE APRENDIZAGEM COM MÉTODO DE LEVENBERG MARQUARDT- ALGORÍTMO EM MATLAB.....	63

1 INTRODUÇÃO

1.1 Importância

Processos industriais que envolvem aquecimento e combustão são complexos e exigem o aumento da temperatura de determinados materiais ou substâncias para alcançar o objetivo desejado. Esses processos, geralmente realizados em ambientes controlados como fornos, reatores ou caldeiras, demandam uma grande quantidade de energia, comumente fornecida pela queima de combustíveis fósseis. Como resultado, há emissão de dióxido de carbono (CO_2), um dos principais gases de efeito estufa, além de uma série de poluentes adicionais, como óxidos de nitrogênio (NO_x), partículas finas ($\text{PM}_{2.5}$ e PM_{10}) (SAMET *et al.*, 2000), compostos orgânicos voláteis (VOCs) e carbono negro, subprodutos comuns da combustão a altas temperaturas. Outro aspecto importante é a emissão de monóxido de carbono (CO), um gás tóxico perigoso em altas concentrações, liberado em casos de combustão incompleta (GODISH, 2014). A variedade desses emissores pode causar impactos significativos à saúde humana e ao meio ambiente, reforçando a importância do controle e da minimização dessas emissões nos processos industriais.

Fornos industriais desempenham um papel vital em diversos processos químicos, sendo amplamente aplicados em diferentes setores industriais. Eles são projetados para operar com uma variedade de combustíveis, como sólidos, líquidos e gasosos, conforme as necessidades específicas do processo. O princípio fundamental desses fornos baseia-se na teoria dos sistemas de combustão, onde uma chama é produzida pela interação entre três elementos essenciais: combustível, agente oxidante (comburente) e uma fonte inicial de energia. Esse processo de combustão é cuidadosamente controlado para assegurar eficiência, segurança e cumprimento das normas ambientais, o que é essencial para o desempenho adequado dos processos industriais (SEINFELD; PANDIS, 2016).

O objetivo principal deste trabalho é desenvolver uma metodologia que possa estimar a presença de NO_x com base nas emissões geradas durante os processos industriais de aquecimento, a partir dos insumos utilizados. Esta

metodologia se baseia na lógica paraconsistente, um ramo da lógica matemática que, diferente da lógica clássica, permite a manipulação e a convivência de contradições em um mesmo sistema de inferência. Espera-se que esta abordagem contribua para o desenvolvimento de estratégias mais eficientes e sustentáveis para a gestão de emissões em processos industriais de combustão.

1.2 Problemática

A comunidade global está cada vez mais preocupada com os impactos da poluição do ar na saúde humana (PEPPLOW *et al.*, 2019). A poluição do ar está presente em vários lugares, como áreas industriais, urbanas e agrícolas. Emissões de diferentes tipos de poluentes podem ter efeitos adversos na saúde humana e no meio ambiente, tornando essencial a adoção de técnicas de otimização na produção para controlar e reduzir essas emissões (YU, 2016). A qualidade do ar cada vez mais precária tem gerado diretamente riscos à saúde dos moradores urbanos e, em uma área altamente poluída, as emissões produzidas durante a combustão de combustíveis fósseis podem ser um perigoso fator de transmissão de doenças (LI, 2015).

Os fornos industriais desempenham um papel crítico em uma variedade de processos, necessitando operar com a máxima eficiência para garantir a transferência ideal de calor gerado à carga em processamento. A qualidade de tal sistema de combustão pode ser indicada pela quantidade de ar presente: uma concentração equilibrada de oxigênio é crucial para a efetividade da combustão. Se o oxigênio se torna escasso, a combustão incompleta pode surgir, levando a um uso de energia ineficiente e à produção de gases tóxicos. Por outro lado, um excesso de oxigênio na câmara de combustão pode resultar na diminuição da Temperatura de Chama Adiabática (TCA), limitando a eficiência térmica do processo (KUTLUMUKHAMEDOV, 2021).

A conformidade com os controles regulatórios exige manter a emissão dos gases de combustão em um valor específico, conhecido como set-point (SP). Para alcançar esse objetivo, ajustes na vazão de combustível são frequentemente necessários. Contudo, essa prática pode aumentar tanto os custos operacionais do forno quanto a quantidade de gases emitidos pela combustão, que precisam ser gerenciados para minimizar o impacto ambiental

e assegurar a eficiência do processo conforme os requisitos regulatórios (STANIER; FARD, 2015).

Identificar, classificar e controlar adequadamente a temperatura dos gases produzidos é fundamental tanto para a integridade estrutural do forno quanto para a proteção do meio ambiente. Uma temperatura de saída de gases muito baixa pode provocar a condensação de vapores de água, potencialmente causando problemas de corrosão nas estruturas metálicas do forno. No entanto, temperaturas excessivamente altas podem favorecer a formação de Óxidos de Nitrogênio térmico (T-NO_x), cuja presença é determinada por fatores como temperatura, concentração de O₂ e tempo de residência (SMITH, 2019). Estudos experimentais mostraram que o dióxido de nitrogênio pode permanecer nos pulmões por períodos prolongados. A doença pulmonar obstrutiva crônica pode ser relacionada a inalação de NO₂, causando fadiga, falta de ar, enfisema, sensação de aperto no peito, tosse e dificuldade para respirar (SAKI, 2020)

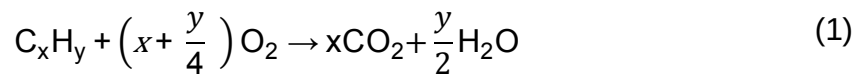
Este trabalho visa abordar a estimativa da presença de NO_x com base nas emissões geradas em fornos industriais. Ao direcionar o foco para a identificação da produção de NO_x, será possível obter informações valiosas sobre o desempenho da combustão quanto as emissões. Esses dados poderão contribuir para o desenvolvimento de um controle mais preciso e adaptável, resultando em uma operação mais segura e sustentável dos fornos industriais.

1.3 Fundamentação teórica

1.3.1 Sistemas de Combustão e Fornos Industriais

Um sistema de combustão é um processo complexo que envolve a oxidação de um agente combustível, geralmente um hidrocarboneto, por um agente comburente, que é o gás oxigênio. A reação química exotérmica resulta na liberação de uma quantidade significativa de energia na forma de plasma. Quando um hidrocarboneto é queimado, os principais produtos resultantes são o CO₂ e a água (H₂O). Essa reação química, representada como Equação (1), retrata de forma simplificada os mecanismos que ocorrem em um sistema de combustão (WESTBROOK & DRYER 1981). Na Equação (1) são descritas as transformações químicas em uma etapa que ocorre em nível molecular durante

esse processo representado na forma quase global, onde o agente combustível é consumido e convertido em produtos estáveis.



É importante ressaltar que, embora a Equação (1) forneça uma representação simplificada da reação de combustão, na prática, há uma diversidade de compostos químicos e intermediários presentes que influenciam o processo global. Entre eles estão radicais livres altamente reativos, como OH, HO₂ e CH₃, que desempenham um papel crucial na progressão das reações de combustão, afetando tanto a velocidade quanto a eficiência do processo (SMITH, 2019). No entanto, a Equação (1) serve como um ponto de partida para que os aspectos fundamentais desse fenômeno sejam mais claros.

A tecnologia presente em sistemas de combustão permite diversas formas de abordagem, na qual a onda de propagação pode ocorrer em diversas velocidades. Quando a onda de propagação ocorre em velocidades supersônicas, segundo Turns (2013), diz-se que houve detonação. Por outro lado, em baixas velocidades, ocorre a deflagração da onda.

A chama, por definição, é "uma propagação autossustentada, em velocidades subsônicas, de uma zona de combustão localizada". Em fornos industriais, a função principal é fornecer calor ao produto no processo, o que se dá por meio da geração de chama. Nesse contexto, a maior temperatura do sistema de combustão ocorre na chama gerada pela combustão, conhecida como Temperatura de Chama Adiabática (TCA) (TURNS, 2013).

Para otimizar o desempenho dos fornos e aumentar a TCA, o pré-aquecimento do ar é uma estratégia amplamente adotada. Esse procedimento fornece energia adicional para acelerar as reações de combustão e reduz as perdas térmicas da câmara, uma vez que o ar pré-aquecido necessita de menos combustível para atingir a temperatura desejada. Essa prática não apenas diminui os custos operacionais, mas também reduz as emissões de CO₂, já que menos combustível é necessário. O ar é aquecido ao entrar em contato com os gases de exaustão, aproveitando o calor gerado no próprio processo, o que

contribui para maior eficiência energética e sustentabilidade nas operações industriais (WESTBROOK & DRYER, 1981; TURNS, 2013).

A Equação (1) ilustra o que ocorre quando há combustível e comburente misturados em proporção estequiométrica. Embora a Equação leve em conta apenas os reagentes do processo de oxidação do combustível, o O₂ presente é fornecido por meio do ar do ambiente, que contém em maior proporção o gás Nitrogênio (N₂). A operação de um forno industrial é realizada considerando vazões elevadas de carga de processo, portanto é infactível que exista tal mistura de forma ideal, ocorrendo então excesso do combustível ou do comburente na reação. Quando o combustível está presente em excesso, denomina-se combustão rica, onde Monóxido de Carbono (CO) produzido em reações intermediárias não será consumido para a geração de CO₂ na reação finalizadora. O excesso de combustível em proporções críticas leva à formação de partículas carbonáceas que são liberadas como fumaça ou depositadas em superfícies próximas. Além disso, ocorrerá um aumento das taxas de combustível não queimado na reação presente nos gases de combustão, podendo inundar a câmara de combustão e ocasionando um cenário de risco de segurança de processo operacional (TURNS, 2013).

Uma solução adotada em indústrias que utilizam fornos industriais é operar com excesso de O₂, de modo a garantir a combustão completa dos produtos intermediários. A TCA máxima que um combustível consegue alcançar é obtido quando o excesso de O₂ está próximo dos valores estequiométricos. Nessa condição, as emissões de CO são reduzidas devido a sua oxidação, gerando CO₂, enquanto se obtém uma temperatura de chama que permitirá maior transferência de calor para o processo industrial. Em contraponto, altas temperaturas de chama leva a formação de Óxido Nítrico (NO), segundo o mecanismo de Zeldovich que determina as Equações (2) e (3) (WESTBROOK & DRYER, 1981) (TURNS, 2013).



E em sua forma estendida, mostrada na reação (4)



A Equação (4) mostra que embora haja uma situação de proporção estequiométrica entre combustível e oxidante, os radicais livres presentes (no caso a hidroxila) promovem a formação de NO. Segundo Turns (2013), a formação do NO₂ ocorre de forma minoritária em regiões pós chama, pois ocorrerá a destruição (ou decomposição) do NO₂ através da Equação (5) e (6)



Em regiões de baixa temperatura, a formação de NO₂ se torna facilitada, devida a propensão da formação da espécie HO₂, que participa da formação de NO₂ na Equação (7). Isso ocorre principalmente quando moléculas de NO sofrem difusão ou são advectadas para dentro de regiões ricas em HO₂ (TURNS, 2013).



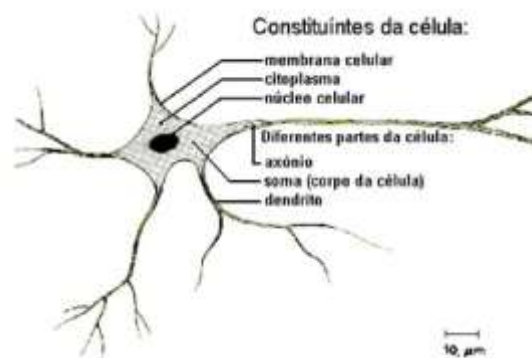
Aumentar o excesso de %O₂ irá auxiliar a combustão completa, como também mitigará a formação do NO_x térmico, uma vez que a temperatura de chama adiabática irá diminuir (PARK, 2012). Porém, o excesso de O₂ quando elevado sem o devido cuidado, irá influenciar na temperatura da zona de radiação do forno. É comum controlar a temperatura de saída do forno através do consumo de gás combustível. As válvulas de controle irão requerer maior consumo de combustível, afetando na eficiência energética do forno, além de produzir ainda mais emissões, mesmo em combustão completa (TURNS, 2013).

No Trabalho de Khodabandeh *et. al* (2016), elevar o excesso de %O₂ acima de 18% levou a uma perda de eficiência do forno, como também um acréscimo de Emissões de CO₂ e H₂O.

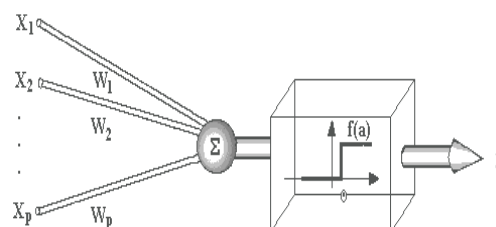
A caracterização precisa da qualidade das emissões com base nas concentrações de O_2 , consumo de combustível, temperaturas e pressão da câmara de combustão é fundamental. Esses dados permitem implementar estratégias avançadas de classificação, utilizando aprendizado de máquina. Sendo assim, é possível identificar padrões, distinguir tipos de emissões e tomar medidas direcionadas para reduzir os impactos ambientais, auxiliando em uma gestão mais eficiente e sustentável das emissões industriais e contribuindo para a preservação ambiental e práticas responsáveis (KHODABANDEH *et al.*, 2016).

1.3.2 Redes Neurais

Agarwal (2018) define que as redes neurais foram originalmente desenvolvidas com o propósito de simular o sistema nervoso humano para tarefas de aprendizado de máquina, buscando replicar o funcionamento das unidades computacionais de maneira semelhante aos neurônios humanos. A Figura 1 mostra a representação do neurônio e seu modelo artificial.



(a)



(b)

Figura 1. Componentes de um neurônio. (a) Representação do neurônio humano, (b) modelo do neurônio artificial.

Fonte: (Martins, 2015)

O objetivo central das redes neurais era criar inteligência artificial através da construção de máquinas cuja arquitetura emulasse os processos de cálculo do sistema nervoso humano. No entanto, essa é uma tarefa altamente complexa, uma vez que o poder de processamento dos computadores atuais é apenas uma fração minúscula do poder computacional do cérebro humano (AGARWAL, 2018).

O perceptron é um modelo de rede neural artificial desenvolvido Rosenblatt (1950) que simula o funcionamento de um neurônio biológico, realizando a multiplicação das entradas por pesos, somando esses produtos e ativando com base em um limiar. Sendo capaz de resolver problemas de classificação binária, o perceptron é limitado, e só consegue lidar apenas com problemas linearmente separáveis. Embora tenha limitações, o conceito do perceptron foi fundamental para o desenvolvimento de redes neurais mais avançadas e complexas, como as redes MLP (*Mutlilayer Perceptron* – Perceptron de multicamadas) e CNNs (*Convulutional Neural Network*- Redes Neurais Convolucionais), que são usadas em uma variedade de aplicações de aprendizado de máquina e processamento de dados (AGARWAL, 2018).

1.1.1 Método de Levenberg Marquadt

Toushmalani (2013) descreve o método de Levenberg Marquadt como uma aproximação do método de Newton, utilizado para treinamento de redes neurais, onde o erro está associado a uma expressão contendo derivadas de segunda ordem. A atualização dos pesos se dá pela Equação (8)

$$W_K = W_{K-1} + \frac{J_K^T E_k}{J_K^T J_K + \eta \Omega_K} \quad (8)$$

Onde,

K: época atual de treinamento;

E_K : Matriz de Erros

J_K : Matriz jacobiana dos Erros

η : Parâmetro de regularização

Ω_K : Matriz diagonal de J_K

O produto $\mu\Omega_K$ determina o quanto os pesos são ajustados de forma a minimizar o erro. Se os pesos atualizados aumentam o MSE (*Mean Square Error* - Erro Médio Quadrático), deve-se aumentar o valor de R, multiplicando por um fator β , geralmente na ordem de 10, e em seguida repetir o cálculo de atualização de pesos. Caso o erro seja diminuído com os novos pesos aplicados, estes substituem os pesos anteriores, e o valor de R é dividido por β (KHAN E SAHAI, 2012). O algoritmo é interrompido caso alcance alguns critérios de parada, que podem ser

1. ε_1 : $MSE < MSE_{ACEITÁVEL}$
2. ε_2 : $K_N > K_{NMÁXIMO}$, sendo K_N o número de iterações em que o erro não é reduzido
3. ε_3 : $R > R_{MÁXIMO}$

1.1.2 Lógica Paraconsistente

A lógica clássica, possui fundamentos estritamente binários e portanto somente admite proposições como verdadeiras ou falsas. Com estes princípios que consideram apenas dois estados lógicos “Falso” ou “Verdadeiro”, a lógica clássica desempenhou um papel fundamental no desenvolvimento do pensamento lógico ao longo da história. No entanto, essa abordagem apresenta limitações quando se trata de lidar com incertezas e informações ambíguas. A estrutura binária da lógica clássica, em que uma proposição é categorizada exclusivamente como verdadeira ou falsa, não é adequada para situações em que existem graus de incerteza ou quando as informações são contraditórias ou parcialmente ambíguas. Questões complexas do mundo real, como a tomada de decisões em condições de incerteza e a modelagem de sistemas com múltiplas variáveis, requerem abordagens mais flexíveis que possam lidar com nuances, inconsistências e contradições parciais nas evidências disponíveis. Tudo isto levou ao surgimento de outras formas de lógica, denominadas de não clássicas, entre estas; as lógicas paraconsistentes, que buscam superar essas limitações e oferecer um quadro mais adequado para a análise de situações com incertezas e informações ambíguas (DA SILVA FILHO, 2008).

Para exemplificar, pode-se descrever a diferença entre o ponto de vista da lógica clássica e da lógica paraconsistente na proposição "Os animais são racionais ou irracionais". No contexto da lógica clássica, a proposição seria avaliada de acordo com a lei do terceiro excluído, sendo verdadeira ou falsa de forma definitiva (DA SILVA FILHO, 2008). Por exemplo, se considera-se a definição tradicional de racionalidade, onde apenas os seres humanos são racionais devido à sua capacidade de raciocínio lógico e linguagem complexa, então a proposição seria falsa. Os animais seriam classificados como irracionais nessa perspectiva. No entanto, essa visão restrita pode ser questionada quando observamos certas habilidades cognitivas em animais, como a capacidade de resolver problemas, aprender com a experiência e demonstrar inteligência em situações específicas. Nesse sentido, a proposição também poderia ser considerada verdadeira, pois alguns animais exibem traços de racionalidade em certos aspectos. Por outro lado, a lógica paraconsistente oferece uma abordagem mais flexível e não-exclusiva para lidar com a proposição. Ela reconhece que um sistema pode conter informações conflitantes sem necessariamente ser considerado falso.

1.1.2.1 Lógica Paraconsistente Anotada com anotação de dois Valores

Segundo Da Silva Filho (2012), a Lógica Paraconsistente LP é capaz de contornar o princípio clássico da não contradição e assim realizar tratamento de sinais contraditórios.

A Lógica Paraconsistente Anotada com anotação de dois valores (LPA2v) é uma extensão da LP e, portanto pertence à família das Lógicas não clássicas. Sendo uma lógica proposicional e evidencial, um sinal lógico paraconsistente da LPA2v é representado por uma proposição P acompanhada de anotação (μ, λ) tal que: $P_{(\mu, \lambda)}$. A anotação sempre é composta dos graus de evidência favorável (μ) e desfavorável (λ) , ambos contidos no intervalo $0 \leq (\mu, \lambda) \leq 1$, e pertencentes ao conjunto dos números reais $\mathbb{R}$ (DA SILVA FILHO, 2008).

Na análise da informação paraconsistente a anotação (μ, λ) atribui conotação lógica à proposição P sustentado pelos valores dos graus de evidência favorável (μ) e desfavorável (λ) .

A LPA2v pode ser associada a um diagrama de Hasse (Reticulado de 4 vértices) conforme mostra a Figura 2, onde são representados os estados lógicos extremos V - Verdadeiro, T - Inconsistente, F - Falso e $\perp$ - Paracompleto.

Uma análise matemática no Reticulado associado à LPA2v permite que, a partir dos valores dos graus de evidência favorável (μ) e desfavorável (λ) da anotação, sejam deduzidos valores de graus de Certeza (G_C) e de graus de contradição G_{CT} . A Equação (9) permite o cálculo de grau de certeza (DA SILVA FILHO, 2012):

$$G_C = \mu - \lambda \quad (9)$$

Sendo $-1 \leq G_C \leq 1$, um alto valor de μ e baixo λ irá levar G_C para o estado lógico verdadeiro (V), onde $G_C = 1$. Caso contrário, o estado de falsidade (F) será alcançado, com $G_C = -1$ (DA SILVA FILHO, 2012).

O valor do grau de contradição G_{CT} é dado pela Equação (10):

$$G_{CT} = \mu + \lambda - 1 \quad (10)$$

Quando duas fontes de evidência favoráveis e desfavoráveis possuem alto valor, um grau de contradição G_{CT} determinará o estado lógico Contraditório (T), com $G_{CT} = 1$. No entanto, a ausência de evidência favorável e desfavorável simultaneamente levará a proposição a um estado lógico paracompleto ($\perp$), onde $G_C = 0$ e $G_{CT} = -1$.

A partir dos valores de G_C e G_{CT} , é possível representa-los no diagrama de Hasse, como na Figura 2, onde os valores de G_C estão dispostos no eixo horizontal, enquanto os valores de G_{CT} se situam no eixo vertical. O encontro destes dois graus forma um estado lógico paraconsistente, tal que:

$$\varepsilon_T = (G_C, G_{CT})$$

O Segmento D representa a distância do ponto $\varepsilon_T(G_C, G_{CT})$ à extremidade lógica mais próxima, ou seja, Verdadeiro (V) ou Falso (F).

Existe um conjunto de infinitos elementos ε_T , equidistantes a (V) e (F), limitados por $G_C, G_{CT} \in [-1, 1]$ que podem ser representar Geometricamente no

Diagrama de Hasse pela equação da circunferência na forma canônica com centro em (1,0) ou (-1,0), delimitada pelo Quadrado Unitário de Plano Cartesiano QUPC, como mostram as Equações (12) e (13).

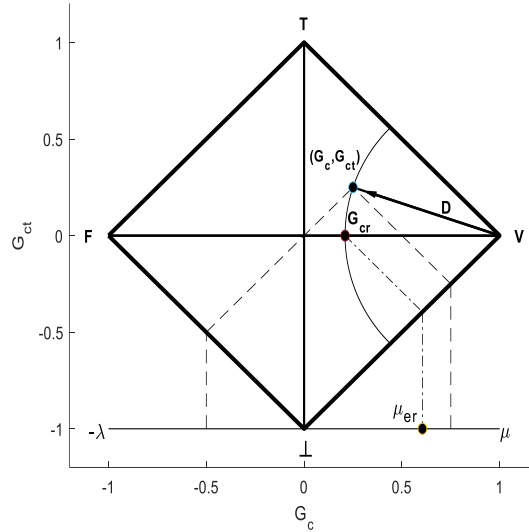


Figura 2. Diagrama de Hasse com ε_r , constituído a partir do par G_c, G_{ct} .
Fonte: (da SILVA FILHO, 2009 adaptado pelo Autor, 2023).

Na Figura 3, os semicírculos de raio unitário representam os valores da notação $\varepsilon_r(G_c, G_{ct})$ para o máximo valor de $D=1$, limitados ao QUPC.

$$\frac{(1-G_c)^2}{D^2} + \frac{G_{ct}^2}{D^2} = 1 \quad \text{se } G_c > 0 \quad (11)$$

$$\frac{(G_c - (-1))^2}{D^2} + \frac{G_{ct}^2}{D^2} = 1 \quad \text{se } G_c < 0 \quad (12)$$

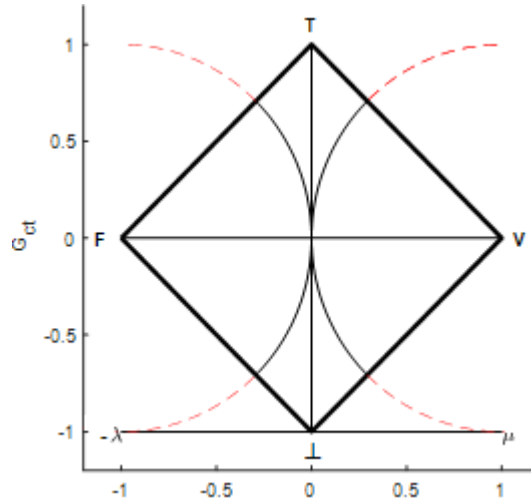


Figura 3. Semicírculos contendo o conjunto de pontos "ε" _ "τ", para D=1.

Denomina-se G_{CR} a notação ε_{τ_i} com distância D do valor lógico V ou F, cujo Grau de contradição seja nulo. Atribuindo $G_{CT}=0$ em (11) e (12), tem-se as Equações (13) a (16)

$$(1-G_{CR})^2=D^2 \quad \text{se } G_C>0 \quad (13)$$

$$(G_{CR}+1)^2=D^2 \quad \text{se } G_C<0 \quad (14)$$

$$G_{CR}=1-D \quad \text{se } G_C>0 \quad (15)$$

$$G_{CR}=D-1 \quad \text{se } G_C<0 \quad (16)$$

Embora as Equações (12) e (13) limitem o valor de D entre 0 e $\sqrt{2}$, quando este ultrapassar o valor unitário, a notação $\varepsilon_{\tau}(G_{CR},0)$ terá o sinal oposto de G_C . Isso significaria que para valores contraditórios elevados, extrair seus efeitos levaria a negação da extremidade lógica mais próxima. Por exemplo, tomando $P_{\mu,\lambda}=(0,9469;0,8469)$, cujos valores do grau de certeza e contradição sejam, respectivamente, $G_C=0,1$ e $G_{CT}=0,7937$. Então, de acordo com a Equação (12), o valor de D será 1,2 e o valor do grau de certeza real $G_{CR}=-0,2$, como mostra a Figura 4. Além da inversão do valor lógico, G_{CR} agora teria um valor absoluto maior de G_C . Adota -se portanto o limite unitário para D, definido pela Equação (9).

$$D=\sqrt{(1-G_C)^2+G_{CT}^2} \quad 0 \leq D \leq 1 \quad (17)$$

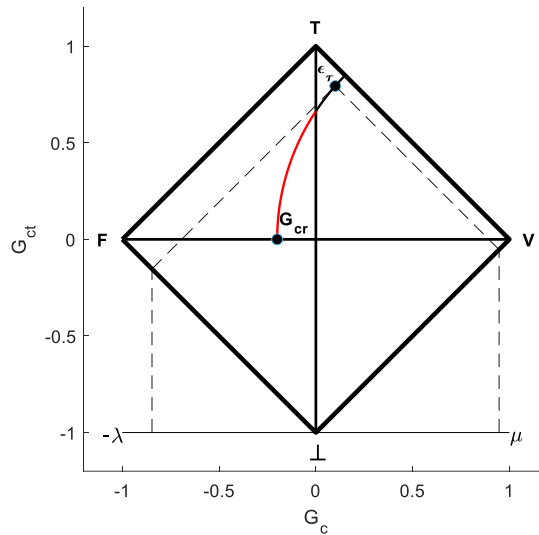


Figura 4. Projeção de GCR caso D não estivesse limitado a 1.

A Figura 4 mostra como G_{CR} se comporta com o valor de μ variando e com λ fixo. Na prática, as Equações (15) e (16) mostram que para $D > 1$, o valor de G_{CR} será sempre nulo quando extraído os efeitos de contradição. Isso é o mesmo que dizer que a proposição lógica é indefinida e nada se pode dizer, pois os efeitos de contradição são suficientes para invalidar qualquer certeza sobre o proposto. Nota-se também que a notação G_{cr} pode ser determinado por um par conjugado $(G_C, +G_{CT})$ e $(G_C, -G_{CT})$, indicando que a informação possui um tratamento de extração do efeito de contradição, sem dizer se as informações tendem à paracompleteza ou contradição. É necessário então que G_{CR} seja expresso em conjunto outra variável que expresse com a natureza de G_{CT} .

Define-se como intervalo de certeza (ϕ), o intervalo ao qual G_C pode variar para o mesmo valor de G_{CT} . Geometricamente, o valor de ϕ será o ponto G_C que forma o estado lógico paraconsistente $\epsilon_T(G_C, G_{CT})$ na fronteira do QUPC, conforme a Figura 6 (DE CARVALHO JUNIOR, 2021).

1.1.2.2 Redes de Análise Paraconsistente (RAP) e Redes Neurais Artificiais Paraconsistentes (RNAP)

Os valores de G_C , G_{CT} , G_{CR} e $\varphi(\pm)$ estão contidos no intervalo fechado $[-1,1]$. É interessante que esses valores sejam normalizados de acordo a entrada, pois podem ser utilizados em conjunto com outras fontes de informação para criação de Redes de Análise Paraconsistente (RAP). Deve-se estabelecer então novos parâmetros que sejam normalizados no mesmo intervalo que μ e λ . Define-se então que a saída de uma RAP deve conter os valores do grau de certeza resultante (μ_E), grau de contradição resultante (μ_{ECT}), grau de certeza resultante real (μ_{ER}) e intervalo de evidência resultante real (φ_E) (DA SILVA FILHO, 2012).

$$\mu_E = \frac{G_C + 1}{2} = \frac{\mu - \lambda + 1}{2} \quad (20)$$

$$\mu_{ECT} = \frac{G_{CT} + 1}{2} = \frac{\mu + \lambda}{2} \quad (21)$$

$$\mu_{ER} = \frac{G_{CR} + 1}{2} \quad (22)$$

$$\varphi_E = 1 - |2\mu_{ECT} - 1| \quad (23)$$

As Equações (20) e (21) estabelecem uma relação linear de μ_E e μ_{ECT} com as fontes de evidência μ e λ , porém o fato da condição de dependência de G_{CR} com o sinal de G_C , tal como a existência da condição de saturação de D no valor unitário, causa não linearidades no parâmetro G_{CR} , como mostra a Figura 4 (DE CARVALHO JUNIOR, 2021).

Define-se como Célula Neural Artificial Paraconsistente Padrão (CNAPp) um elemento capaz de fornecer saída μ_e e um intervalo de evidência resultante, a partir de uma entrada formada por um par (μ, λ) de evidência favorável e desfavorável. A diferença entre um Algoritmo do tipo NAP e o de uma CNAP se dá pelo fato do segundo ser delimitado por fatores externos que controlam os níveis de contradição e certeza (DA SILVA FILHO, 2008).

As Equações (24) a (29) postulam a relação da saída de acordo com os fatores

$$V_{SCC} = \frac{1+F_{TC}}{2} \quad (24)$$

$$V_{ICC} = \frac{1-F_{TC}}{2} \quad (25)$$

$$V_{SCCT} = \frac{1+F_{TCT}}{2} \quad (26)$$

$$V_{ICCT} = \frac{1-F_{TCT}}{2} \quad (27)$$

$$V_{IV} = \frac{1+F_D}{2} \quad (28)$$

$$V_{IF} = \frac{1-F_D}{2} \quad (29)$$

O valor superior de Controle de certeza (V_{SCC}) indica o limite de G_c para que a proposição não seja indefinida. Seu valor é dependente do fator de tolerância a certeza (F_{TC}). Pode se estabelecer também o valor inferior de controle de certeza (V_{ICC}) que levará a proposição a indefinição, caso as fontes de evidência sejam o suficiente para satisfazer o critério da Equação (25). Analogamente, tem-se os valores superior e inferior de controle de contradição (V_{SCCT} e V_{ICCT} , respectivamente) dependentes do fator de tolerância a contradição. Ainda na CNAPP, pode se estabelecer os valores limites de verdade (V_{IV}) e falsidade (V_{IF}), que levam a proposição ao verdadeiro ou falso, de acordo com o fator de decisão F_D (DE CARVALHO JUNIOR, 2021).

O fator de aprendizagem F_A é capaz de transformar uma CNAPP em uma Célula Neural Artificial de Aprendizagem (CNAPap), representada na Figura 6. As Equações (31) e (32) estabelecem a relação entre saída e entradas, em função de F_A (DA SILVA FILHO, 2008). Para que a saída “aprenda” o valor da entrada, a mesma deve ser comparada através da realimentação, onde λ receberá o valor de saída complementado.

$$\mu_{E_{K+1}} = \frac{\{\mu_K - \lambda_K \cdot F_A\} + 1}{2} \quad (30)$$

$$\lambda_K = 1 - \mu_{E_K} \quad (31)$$

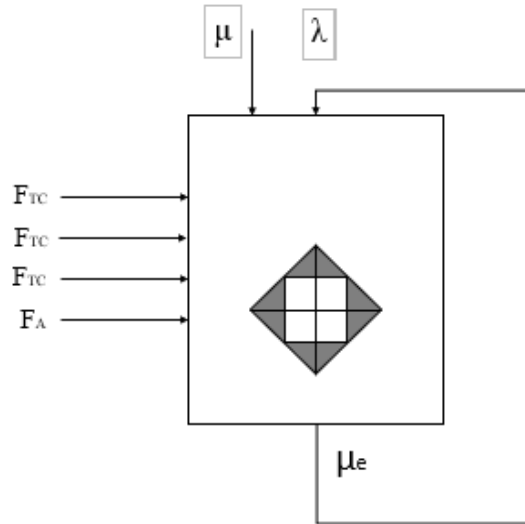


Figura 6. Célula Neural Artificial Paraconsistente de aprendizagem (CNAPap).

A Figura 7 (a) mostra que, para $F_A \neq 1$, a saída produzirá uma diferença em regime permanente. A fim de se conseguir erro nulo, é proposto que a saída da CNAPp produza um resultado livre do efeito da contradição. Tem-se então a Célula Neural Artificial Paraconsistente de Aprendizagem com Extração do Efeito da Contradição (CNAPapctx), onde o Fator de aprendizagem fará interação não somente com λ , mas também com a entrada μ , sendo que a saída será μ_{ECT} ao invés de μ_E , como mostra a Equação 24 (MINICZ *et al.*, 2014).

A Figura 8(b) mostra a saída em função da entrada para diferentes valores de F_A para uma CNAPapctx. Percebe-se que, como estabelecido por Minicz *et al.* (2014), a CNAPapctx terá aprendizagem natural em $0 < F_A < 1$, sendo que em $F_A = 0$ não haverá aprendizado, e em $F_A = 1$ a saída se comportará como uma função degrau.

$$\mu_{ECT_{K+1}} = \frac{\mu_k + \{\mu_k - \lambda_k\} F_A + 1}{2} \quad (32)$$

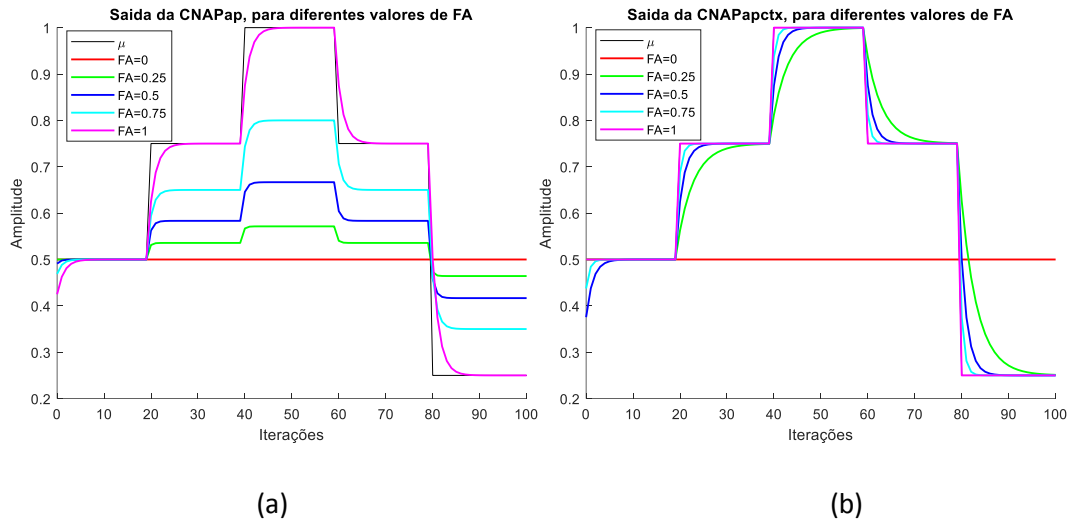


Figura 7. Comportamento de aprendizagem para (a) CNAPap , (b) CNAPapctx
 Fonte: (De CARVALHO JUNIOR, 2021)

As CNAPP, CNAPap e CNAPapctx, quando combinadas, são capazes de formar as Redes Neurais Artificiais Paraconsistentes (RNAP).

Segundo De Carvalho Junior (2021), a RNAP não é capaz de aprender sozinha, mas é ajustada a limiares de transição definidos por F_{CT} , F_{TCT} , e F_d . É proposto então um neurônio artificial paraconsistente, onde os pesos iniciais são atualizados de acordo em função do Erro médio Quadrático (*Mean Square Error -MSE*). A Figura 8 ilustra a Célula Neural Paraconsistente concebida por De Carvalho Junior (2021).

Na célula neural paraconsistente, os pesos de entrada são tratados utilizando CNAPP de acordo com o número de entradas da célula. Em De Carvalho Junior (2021) é proposto que seja utilizado sempre entradas da ordem geométrica de 2 (2, 4, 8, ...) fazendo com que as entradas não utilizadas fiquem em zero. Considerando o Neurônio Paraconsistente com 4, pode-se chegar à Equação (34) (DE CARVALHO JUNIOR, 2021)

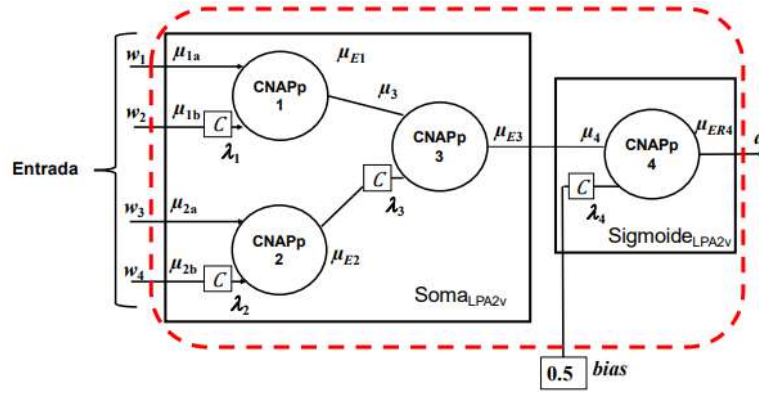


Figura 8. Neurônio Artificial Paraconsistente (NAP) composto de CNAPps.

Fonte: (De CARVALHO JUNIOR, 2021)

$$\mu_{E3} = \frac{1}{4}(\mu_1 - \lambda_1 - \mu_2 + \lambda_2) + \frac{1}{2} \quad (33)$$

E, estendendo para 8 entradas, conforme a Equação 26 (DE CARVALHO JUNIOR, 2021)

$$\mu_{E7} = \frac{1}{8}(\mu_1 - \lambda_1 - \mu_2 + \lambda_2 - \mu_3 + \lambda_3 + \mu_4 - \lambda_4) + \frac{1}{2} \quad (34)$$

Como mostra a Figura 09, De Carvalho Junior (2021) utiliza CNAPs com saída linear μ_E em camadas intermediárias como elementos auxiliares de ligação entre camadas. No decorrer deste trabalho, será abordada uma estratégia diferente da apresentada por De Carvalho Junior (2021).

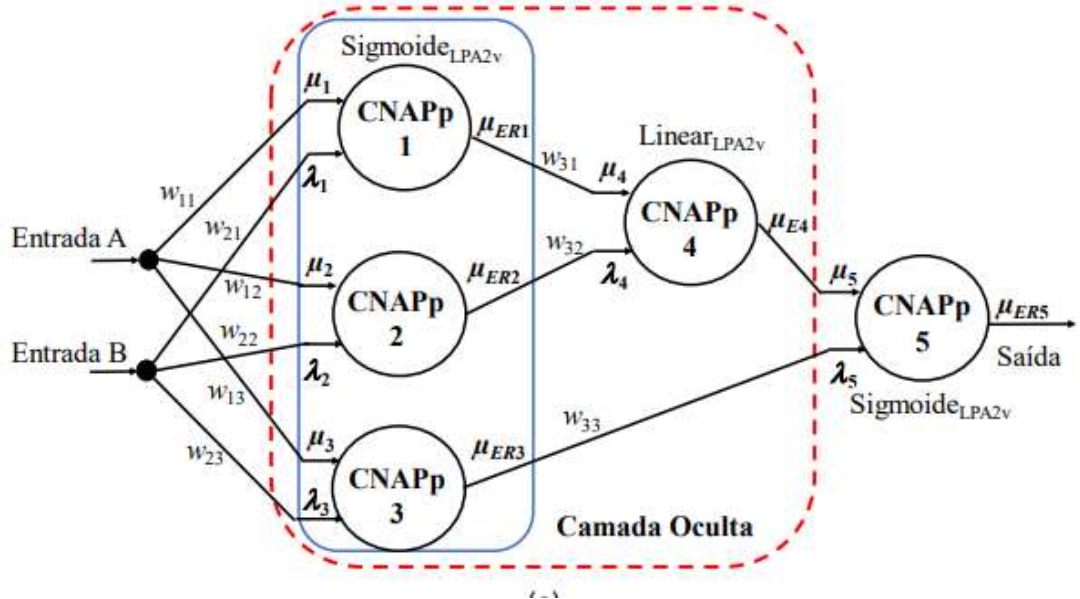


Figura 9. Neurônio Artificial Paraconsistente (NAP) com CNAPs de saída linear μ_E em camadas intermediárias.

Fonte: (De CARVALHO JUNIOR, 2021).

O presente trabalho tem como foco a utilização dos conceitos da lógica paraconsistente para criar modelos computacionais dedicados a identificação da qualidade da chama de um forno de combustão, considerando a emissão de agentes poluentes. A lógica paraconsistente, por sua flexibilidade em lidar com incertezas, informações ambíguas e contradições parciais, oferece uma abordagem adequada para a análise de incertezas envolvendo o processo de combustão de um forno. Espera-se que a utilização da lógica paraconsistente proporcione resultados promissores e contribua para avanços na área de combustão em processos industriais.

1.2 Justificativa

A crescente preocupação a respeito das mudanças climáticas leva a discussões que envolvem a comunidade científica e conjunto de autoridades políticas, pois, segundo Bloomfield (2022, p.278), todos os governantes estão pressionados a aumentar suas metas e ações. Em 2021, foi realizada a COP26 (*Conference of Parts* – Conferência das Partes), onde o Brasil se comprometeu a reduzir 50% dos gases de Efeito Estufa (GEE). Paralelamente, a presença de Compostos Nitrogenados (NO_x) é questão de saúde pública. De acordo com Wang (2022), NO_x e CO_2 são os principais poluentes da emissão de gases, onde NO_x afeta seriamente a atmosfera e a saúde humana. Segundo Hill (2000), a principal fonte de emissão de NO_x são os sistemas de combustão.

A necessidade de se controlar as emissões levam ao surgimento de publicações a respeito do tema. Pavlak (2019), em seu trabalho, realizou previsões de NO_x em uma usina termelétrica utilizando Redes Neurais com função de ativação Sigmoide. Tuttle (2019) desenvolveu um sistema de redução de emissões de uma usina termelétrica a carvão utilizando redes com saída do tipo tangente hiperbólica (TANH), com otimização pelo método de enxame.

Sendo assim, as constantes pressões à mitigação das emissões de gases levam a necessidade do desenvolvimento, otimização de processos e equipamentos que gerem cada vez menos agentes responsáveis por mudanças climáticas e outros impactos ambientais.

1.3 Objetivo

1.3.1 Objetivo geral

O objetivo geral dessa pesquisa é implementar um modelo computacional de sistema dedicado à estimativa da intensidade de emissões de NO_x em um forno industrial com base em uma estrutura configurada com Redes Neurais Artificiais utilizando algoritmos fundamentados em LPA2v.

1.3.2 Objetivos específicos

De forma a se alcançar o objetivo geral do trabalho, serão adotados os seguintes objetivos específicos:

- a) Aquisição de dados de um sistema real e processos de pré-processamento para normalização de valores;
- b) Construção de uma estrutura computacional RNAP funcionando com algoritmos fundamentados em LPA2v capaz de relacionar os parâmetros de entrada com a emissão de NOX.
- c) Elaboração do Algoritmo de Levenberg-Marquadt adaptado a estrutura computacional RNAP funcionando com algoritmos fundamentados em LPA2v.
- d) Efetuação de testes comparativos entre RNA convencional e a RNA paraconsistente construída.

2 MATERIAIS E MÉTODOS

2.1 Materiais

Os dados utilizados nesta pesquisa foram obtidos de um forno industrial do tipo Cilindro Vertical com Convecção, alimentado através de 6 queimadores, com pré-aquecimento do primário do ar através da troca térmica com os gases de combustão gerados.

A Figura 10 mostra um típico forno de destilação de petróleo semelhante ao utilizado neste trabalho.

O combustível utilizado é composto por uma mistura de gases gerados em diferentes unidades de processo de refino, com predominância da fração de metano (CH_4).



Figura 10. Forno de destilação atmosférica de petróleo.

Fonte: (Combustol Metalpo, 2023) ¹

¹ Forno industrial com câmara cilíndrica vertical para aquecimento de Petróleo, com alimentação a gás combustível. Disponível em <<https://www.metalpo.com.br/fornos-fornos-petroquimicos/>> acesso em: novembro de 2023.

Os dados foram coletados considerando a operação em regime estacionário.

Os dados utilizados foram obtidos em um período de amostragem de 5 minutos de um forno industrial com operação em regime permanente. Foi escolhido um intervalo com variação brusca de processo, para análise de como a RNAP irá se comportar frente a essa mudança. Embora estivesse disponível outros parâmetros candidatos a entrada de dados, a escolha da vazão de ar e combustível foi realizada por se tratar de variáveis com maior independência entre si. A saída de CO₂ foi descartada pois, quando se realizou levantamento dos dados, percebeu-se através da matriz de correlação que o mesmo tinha correlação unitária com o combustível, portanto, se tratava de um sensor virtual ou calculados. O conjunto de dados passou a ser então, 2 variáveis de entrada e uma de saída, cada uma com 360 amostras, totalizando 1080 pontos de amostragem.

No processo de pré-processamento os dados adquiridos, representados nas Equações por $\mathbf{D}$, foram arranjados matricialmente de forma que cada amostra ficasse disposta em linhas e as variáveis, em colunas. Valores nulos e negativos foram removidos. Após o processo de adequação os dados foram então normalizados de acordo com a Equação 35. Assim, todo universo de discurso ficou limitado ao intervalo fechado $[0,1]$, adequando-se aos limites dos graus de evidencia da LPA2v μ e λ .

$$d_{i,j} = \frac{(D_{i,j} - \min(\mathbf{D}_j))}{\max(\mathbf{D}_j) - \min(\mathbf{D}_j)} \quad (35)$$

Considerando que dados provenientes de processos industriais com ambiente severo podem carregar diversas inconsistências ocasionadas por defeito ou mesmo por manutenções preventivas e corretivas, foi realizada uma prévia análise de correlação que encontrasse melhor intervalo de explicação. A Tabela 1 mostra a correlação de todas as variáveis entre si, que foram consideradas na pesquisa.

Tabela 1 – Correlação entre dados normalizados de um forno industrial.

	A	B	C	D	E	F	G
A	1	0.95	0.95	0.88	0.92	0.78	0.96
B	0.95	1	0.96	0.90	0.92	0.85	0.99
C	0.95	0.96	1	0.94	0.94	0.89	0.97
D	0.88	0.90	0.94	1	0.96	0.98	0.94
E	0.92	0.92	0.94	0.96	1	0.91	0.96
F	0.78	0.85	0.89	0.98	0.91	1	0.89
G	0.96	0.99	0.97	0.94	0.96	0.89	1

- A. Vazão de Ar para queima
- B. Vazão de Combustível
- C. Temperatura de saída na Chaminé
- D. Temperatura de Ar quente para queima
- E. Temperatura de Gases de combustão (quente)
- F. Temperatura de Gases de combustão (frio)
- G. Produção de NO_x

Considerando que há forte correlação das variáveis com a produção de NO_x (G), foram então escolhidas variáveis de entrada da RNAP a Vazão de Ar de entrada utilizada na queima (A) e a vazão de combustível (B).

A produção de NO_x foi então selecionada como parâmetro de saída.

2.2 Métodos

2.2.1 Estrutura no Neurônio Artificial Paraconsistente (NAP) Proposto

A estratégia abordada na pesquisa foi a de utilizar apenas neurônios com saída não linear. Com isso nos Neurônios Artificiais Paraconsistentes (NAPs) não será necessária a utilização de CNAPs com saída linear μ_E em camadas intermediárias. Para que isso seja possível, altera-se o número de entradas dos neurônios de camada posterior, conforme mostrado a seguir.

Considerando o número total de neurônios da camada anterior (n_{l-1}) e o número de portas (n_p) dos neurônios da camada posterior (n_l), a Equação 36 estabelece que:

$$n_{p;l} = 2^{\lceil \log_2(n_{l-1}) \rceil} \quad (36)$$

A estrutura interna da RNAP terá Neurônios Artificiais Paraconsistentes (NAPs) com entradas λ complementadas. Desta forma, tem-se que uma função linear μ_{E_3} de um NAP de 4 entradas, passará a ser definida, de acordo com a Equação 37.

$$\begin{aligned}
 \mu_{e_3} &= \frac{\mu_3 - \lambda_3 + 1}{2} \\
 \mu_3 &= \mu_{e_1} = \frac{\mu_1 - \lambda_1 + 1}{2} \\
 \lambda_3 &= 1 - \mu_{e_2} = 1 - \frac{\mu_2 - \lambda_2 + 1}{2} \\
 \mu_{e_3} &= \frac{\left(\frac{\mu_3 - \lambda_3 + 1}{2} + \frac{\mu_2 - \lambda_2 + 1}{2}\right)}{2} = \frac{1}{4} [(\mu_1 + \mu_1) - (\lambda_1 + \lambda_2)] + \frac{1}{2}
 \end{aligned} \tag{37}$$

Estendendo o conceito para um NAP de 8 entradas, tem-se através da Equação 38:

$$\begin{aligned}
 \mu_{e_5} &= \frac{\mu_5 - \lambda_5 + 1}{2} = \frac{1}{4} [(\mu_1 + \mu_1) - (\lambda_1 + \lambda_2)] + \frac{1}{2} \\
 \mu_{e_6} &= \frac{\mu_6 - \lambda_6 + 1}{2} = \frac{1}{4} [(\mu_3 + \mu_4) - (\lambda_3 + \lambda_4)] + \frac{1}{2} \\
 \mu_{e_7} &= \frac{\mu_7 - \lambda_7 + 1}{2} \\
 \mu_{E7} &= \frac{\left(\left\{\frac{1}{4} [(\mu_1 + \mu_1) - (\lambda_1 + \lambda_2)] + \frac{1}{2}\right\} + \left\{\frac{1}{4} [(\mu_3 + \mu_4) - (\lambda_3 + \lambda_4)] + \frac{1}{2}\right\}\right)}{2} \\
 \mu_{E7} &= \left\{\frac{1}{8} [(\mu_1 + \mu_2 + \mu_3 + \mu_4) - (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4)] + \frac{1}{2}\right\}
 \end{aligned} \tag{38}$$

Sendo μ_j e λ_j respectivamente, a entrada correspondente a um grau de evidência favorável e desfavorável, define-se como μ_i qualquer entrada do Neurônio Artificial Paraconsistente, considerando a paridade, conforme a Figura 11. Portanto, o novo neurônio NAP irá receber os graus de evidencia favorável nas entradas impares e desfavoráveis nas entradas pares. A Equação 39 define que,

$$\mu_i = \begin{cases} \mu_j, & \text{se } i \text{ é ímpar} \\ \lambda_j, & \text{se } i \text{ é par} \end{cases} \quad (39)$$

$$(40)$$

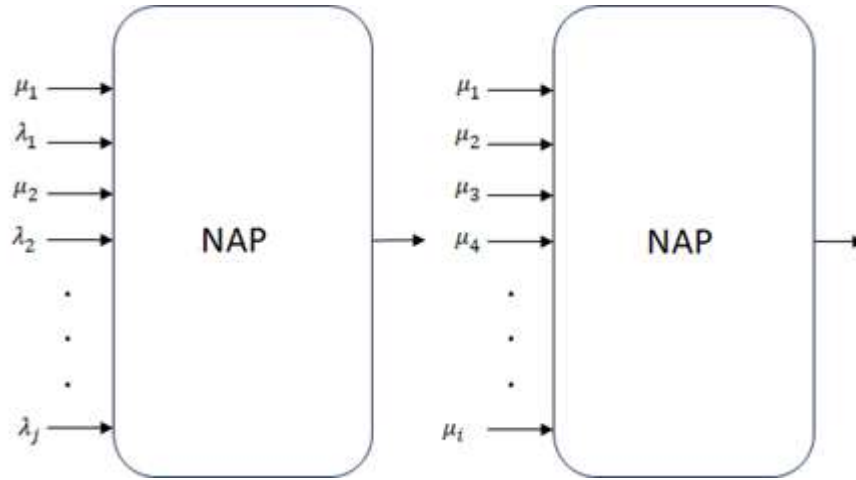


Figura 11. Neurônios Artificiais Paraconsistentes com entradas adaptadas.

A Equação (41) define então a saída linear $\mu_{E_{N-1}}$, sendo N o número de portas do NAP.

$$\mu_{E_{N-1}} = \frac{1}{2N} \sum_{j=1}^{n_p} (-1)^{i+1} (\mu_i) + \frac{1}{2} \quad (41)$$

Da Equação (34) proposta em De Carvalho Junior (2021), que se assemelha ao grau de contradição resultante da arquitetura aqui apresentada, considerou-se um neurônio genérico de N entradas. Dessa forma pode-se estabelecer o seguinte algoritmo.

1. Conte o número de entradas
2. Da entrada 1 até N/2, faz-se a diferença entre as somas das entradas ímpares e entradas pares
3. Da entrada N/2+1 até N, faz-se a diferença entre as somas das entradas pares e entradas ímpares
4. Soma-se o passo 2 e 3
5. Faz-se a divisão por N

A Equação (42) estabelece $\mu_{ECT_{N-1}}$, para N entradas.

O Apêndice A mostra o desenvolvimento do Neurônio Artificial Paraconsistente de 16 entradas, onde prova-se a Equação (42).

$$\mu_{ECT_{N-1}} = \frac{1}{N} \left[\sum_{i=1}^{\frac{N}{2}} (-1)^{i+1} \mu_i + \sum_{i=\frac{N}{2}+1}^N (-1)^i \mu_i \right] + 0,5 \quad (42)$$

Pode-se perceber pela Equação (41) que todos os elementos de entrada são uniformemente ponderados pelo neurônio paraconsistente, diferentemente do que acontece quando há o uso de CNAPs lineares intermediários entre camadas, representado pela Figura 09, onde o valor de saída μ_{ER5} terá maior sensibilidade para μ_{ER3} em relação a μ_{ER1} e μ_{ER2} . Quanto mais entradas o Neurônio Artificial Paraconsistente possuir, o valor da parcela linear ficará mais próximo de 0,5, ou seja,

$$\lim_{n_p \rightarrow \infty} \mu_E = 0,5 \text{ (indefinição)} \quad (43)$$

Propõe-se então a saída do Neurônio Artificial Paraconsistente como não linear e para isso, uma saída μ_{ER} é utilizada como a função de ativação. O valor λ_B deve ser de 0,5 para que o valor de μ_{ER} se comporte como uma função sigmoide em função da saída linear μ_E , como é observado na Figura 5 da seção 1.3.4.1. Segundo as Equações (20), (21), (43) e (44), a saída do Neurônio Artificial Paraconsistente utilizado será então:

$$\mu_{ER} = \frac{G_{CR}+1}{2} \quad (44)$$

Para $\mu_E > 0,5$

$$\begin{aligned} D &= \sqrt{2\mu_E^2 - 4\mu_E + \frac{5}{2}} \\ \mu_{ER} &= \frac{1-D+1}{2} \\ \mu_{ER} &= 1 - \frac{\sqrt{2\mu_E^2 - 4\mu_E + \frac{5}{2}}}{2} \end{aligned} \quad (45)$$

Para $\mu_E < 0,5$

$$D = \sqrt{\left(\mu_E + \frac{1}{2}\right)^2 + \left(\mu_E - \frac{1}{2}\right)^2}$$

$$D = \sqrt{2\mu_E^2 + \frac{1}{2}}$$

$$\mu_{ER} = \frac{D-1+1}{2}$$

$$\mu_{ER} = \frac{\sqrt{2\mu_E^2 + \frac{1}{2}}}{2} \quad (46)$$

Para $\mu_E = 0,5$, $\mu_{ER} = 0,5$ (indefinição).

A função de saída μ_{ERnp} é formada por uma função por partes, como mostrado nas Equações (44) e (45). Deve-se encontrar as derivadas parciais da função de saída em relação as entradas μ_j . A derivada de $\mu_{E_{np-1}}$ em relação da μ_j será, de acordo com a regra da cadeia demonstrada na Equação 46.

$$\frac{\partial \mu_{ER}}{\partial \mu_j} = \frac{\partial \mu_{E_{np-1}}}{\partial \mu_j} \cdot \frac{\partial \mu_{ERnp}}{\partial \mu_{E_{np-1}}} \quad (47)$$

As derivadas parciais $\partial \mu_{ERnp} / \partial \mu_{E_{np-1}}$ são calculadas considerando μ_{ERnp} obtido através das Equações (47) e (48).

Para $\mu_{E_{np-1}} > 0.5$

$$\begin{aligned} \frac{\partial \mu_{ER}}{\partial \mu_E} &= \frac{\partial \left(1 - \sqrt{\frac{2\mu_{E_{np-1}}^2 - 4\mu_{E_{np-1}} + 5}{2}} \right)}{\partial \mu_E} \\ \frac{\partial \mu_{ER}}{\partial \mu_E} &= - \frac{(\mu_{E_{np-1}} - 1)}{\sqrt{2\mu_{E_{np-1}}^2 - 4\mu_{E_{np-1}} + 5}} \\ \frac{\partial \mu_{ER}}{\partial \mu_E} &= \frac{(1 - \mu_{E_{np-1}}) \sqrt{2\mu_{E_{np-1}}^2 - 4\mu_{E_{np-1}} + 5}}{2\mu_{E_{np-1}}^2 - 4\mu_{E_{np-1}} + 5} \end{aligned} \quad (48)$$

Para $\mu_{E_{np-1}} < 0.5$

$$\frac{\partial \mu_{ER}}{\partial \mu_{E_{np-1}}} = \frac{\partial \left(\frac{\sqrt{2\mu_E^2 + 1}}{2} \right)}{\partial \mu_E}$$

$$\frac{\partial \mu_{ER}}{\partial \mu_E} = \frac{\mu_{E_{np-1}} \sqrt{2\mu_{E_{np-1}}^2 + 1}}{2\mu_{E_{np-1}}^2 + \frac{1}{2}} \quad (49)$$

Analisando a continuidade de $\partial \mu_{ER} / \partial \mu_{E_{np-1}}$, para $\mu_{E_{np-1}} \rightarrow 0,5$ nas Equações (48) e (49)

$$\lim_{\mu_E \rightarrow 0,5^+} \frac{(1-\mu_{E_{np-1}}) \sqrt{2\mu_{E_{np-1}}^2 - 4\mu_{E_{np-1}} + \frac{5}{2}}}{2\mu_{E_{np-1}}^2 - 4\mu_{E_{np-1}} + \frac{5}{2}} = 0,5$$

$$\lim_{\mu_E \rightarrow 0,5^-} \frac{\mu_E \sqrt{2\mu_{E_{np-1}}^2 + 1}}{2\mu_{E_{np-1}}^2 + \frac{1}{2}} = 0,5$$

Percebe-se que os critérios de continuidade são válidos para $\partial \mu_{ER} / \partial \mu_E$ em todo intervalo fechado de $0 \leq \mu_E \leq 1$. Pode-se então generalizar as Equações das derivadas parciais de μ_{ER} em função das entradas de um neurônio paraconsistente com n_p entradas, de acordo com a Equação (50), onde $(-1)^{j+1}$ define a alternância de sinal conforme a paridade i , que representa o índice de entrada do neurônio.

$$\frac{\partial \mu_{ER}}{\partial \mu_i} = (-1)^{i+1} \begin{cases} \frac{(1-\mu_E) \sqrt{2\mu_E^2 - 4\mu_E + \frac{5}{2}}}{2N \left(2\mu_E^2 - 4\mu_E + \frac{5}{2} \right)} & \mu_{E_{N-1}} > 0,5 \\ \frac{\mu_E \sqrt{2\mu_E^2 + 1}}{N^* (4\mu_E^2 + 1)} & \mu_{E_{N-1}} < 0,5 \\ 0.5 & \mu_{E_{N-1}} = 0,5 \end{cases} \quad (50)$$

Utilizando as Equações anteriores foram efetuados os primeiros ensaios, com resultados mostrados nas figuras a seguir. A Figura 12 mostra que $\partial\mu_{ER}/\partial\mu_E$ apresenta uma curva suave em todo intervalo [0,1] semelhante a derivada de uma função Log-sigmoid. As Figuras 13 (a) e (b) ilustram o comportamento de $\partial\mu_{ER}/\partial\mu$ para as entradas pares e ímpares do neurônio paraconsistente.

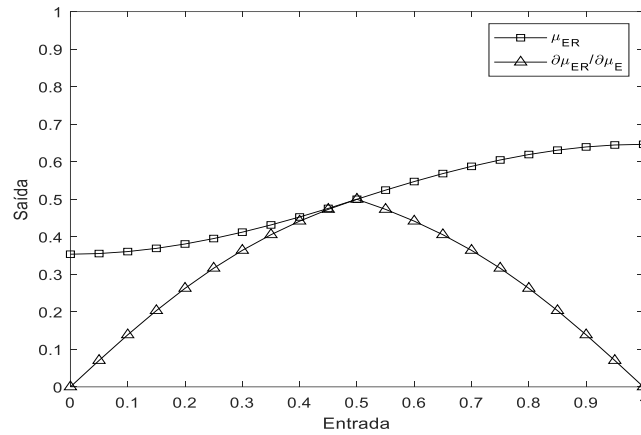


Figura 12. Comportamento de $\partial\mu_{ER}/\partial\mu_E$, considerando a função por partes montada a partir das Equações (47) E (48).

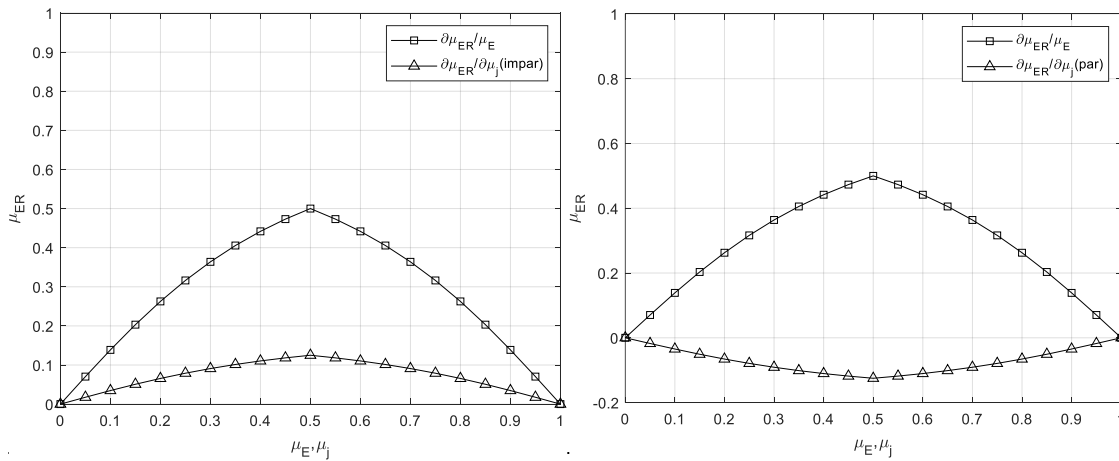


Figura 13- Comparação entre as derivadas de " μ_{ER} ", com relação a μ_E : (a) entrada ímpar. (b) entrada par.

2.2.2 Estrutura da RNAP

Considerando que os neurônios de uma RNAP exigem que as entradas estejam normalizadas no intervalo [0,1], então estabeleceu-se que o algoritmo descreverá uma entrada μ_{ji} de um neurônio N_i qualquer que receba uma saída

$A_{j, \ell-1}$ de um neurônio N_j camada anterior combinado com um peso W_{ij} . Esta descrição estará de acordo com a Equação (51).

$$\mu_{ji,\ell} = \max[0, \min(1, W_{ij,\ell} * A_{j,\ell-1})] \quad (51)$$

Portanto, o algoritmo de propagação direta fica definido de acordo com a sequência a seguir:

- 1- Determine o número total de camadas L
- 2- Determine o número de neurônios para cada camada ℓ ;
- 3- Determine o número de portas para os neurônios N_i ;
- 4- Estabeleça o critério de saturação das entradas de N_i ;
- 5- Calcule $\mu_{E,\ell}$ e $\mu_{ER,\ell}$ e $\partial\mu_{ER,\ell}$
- 6- Repita o passo 2 ao 5 até o neurônio da camada de saída.

2.2.3 Método de Levenberg-Marquadt associado a RNAP proposta

Para o treinamento da estrutura da RNAP considerou-se o que foi exposto na literatura onde a LPA2v foi utilizada em RN. Em trabalhos anteriores utilizou-se o método *Backpropagation* para treinamento da RNAP onde, ao treinar a rede para aprendizagem de uma porta lógica XOR (*Exclusive OR*), os pesos eram atualizados de modo a reproduzir uma saída predefinida. Neste trabalho, foram utilizados os dados coletados de um forno industrial, contendo 360 pontos de amostragem. Estes dados foram recombinaados de forma aleatória, de forma a evitar viés de sequência durante o treinamento. Sendo assim o Algoritmo de treinamento foi o de Levenberg-Marquadt, que permite treinar todo conjunto de dados por época de treinamento. A sequência de construção do algoritmo utilizado nesta pesquisa se dá por:

- 1- Estabeleça os critérios de parada
- 2- Inicie o parâmetro R
- 3- Para a n -ésima entrada, realize a propagação direta para calcular a n -ésima saída y e estabeleça o vetor de colunas Y
- 4- Estabeleça o valor de MSE
- 5- Encontre as derivadas parciais $\partial y / \partial W$ e estabeleça o vetor de linhas P
- 6- Estabeleça a matriz Jacobiana (J), a matriz Hessiana H e a função ômega
- 7- Estabeleça o Vetor de Linhas P_{K+1}
- 8- Calcule o novo valor de MSE e compare com o antigo
 - 8.1 Se $MSE_{K+1} \geq MSE_K$, aumente o valor de R e retorne ao passo 5
 - 8.2 Se $MSE_{K+1} < MSE_K$, atualize os pesos, regresse o valor de R e retorne ao passo 3
 - 8.3 Se ϵ_1 , ϵ_2 ou ϵ_3 , pare

A Figura 14 mostra como se dá então a arquitetura da RNAP proposta, onde X_1 será a vazão de combustível, X_2 será a vazão de ar. X_1 e X_2 se combinam para criar uma terceira entrada X_1X_2 , de modo a acrescentar não linearidade na rede.

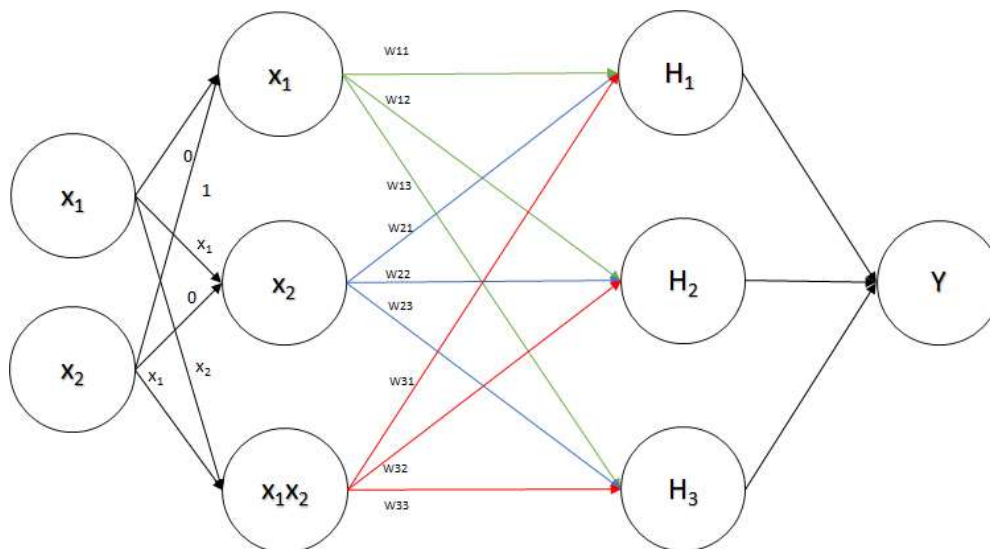


Figura 14. Arquitetura da RNAP utilizada.

Para uma análise comparativa foram efetuados testes utilizando dois tipos de redes onde a primeira configuração de rede utiliza a função log-sigmoide, semelhante a utilizada no trabalho de Pavlak (2019), sendo comparada com a rede com estrutura cuja configuração incorpora a Célula RNAP. Quanto ao número de camadas e neurônios, ambas as redes possuem a mesma estrutura, diferindo apenas quanto a função aditiva e de ativação.

3 RESULTADOS E DISCUSSÃO

Apresenta-se a seguir os resultados obtidos com as simulações efetuadas nos dois tipos de redes que foram configuradas para os testes de simulação.

A Figura 15 apresenta a comparação do desempenho de duas redes neurais distintas em relação às funções de ativação utilizadas.

A primeira rede utiliza a célula RNAP, enquanto a segunda incorpora a função Log-Sig. É perceptível que ambas as redes conseguem acompanhar variações bruscas em relação ao alvo.

A visualização dessa característica se dá graças a aleatoriedade dos dados. Percebe-se porém melhor ajuste da RNAP em relação a Rede Log-Sig.

A Figura 16a apresenta a evolução do MSE ao longo do treinamento para ambas as redes neurais. Notavelmente, a rede neural com função log-sigmoide iniciou o treinamento com um MSE inicialmente menor em comparação com a rede paraconsistente. No entanto, ao longo das épocas, observamos uma inversão na tendência.

A RNAP superou a log-sigmoide, atingindo um MSE significativamente menor ao final do treinamento. Em aproximadamente 15 épocas, a rede estabilizou o MSE.

Este comportamento sugere que, inicialmente, a RNAP teve mais dificuldades em se ajustar aos dados. No entanto, sua capacidade de superar esse início desfavorável e atingir um MSE menor ao final do treinamento sugere uma notável capacidade de adaptação.

A Figura 16b compara o histograma de erros da RNA log-sigmoide com a RNAP. Pode-se perceber que a RNAP manteve o erro concentrado próximo a região de erro nulo, enquanto a RNA log-sigmoide teve seus erros mais distribuídos na região de erro negativo. Percebe-se que a RNAP possui característica próxima da distribuição normal.

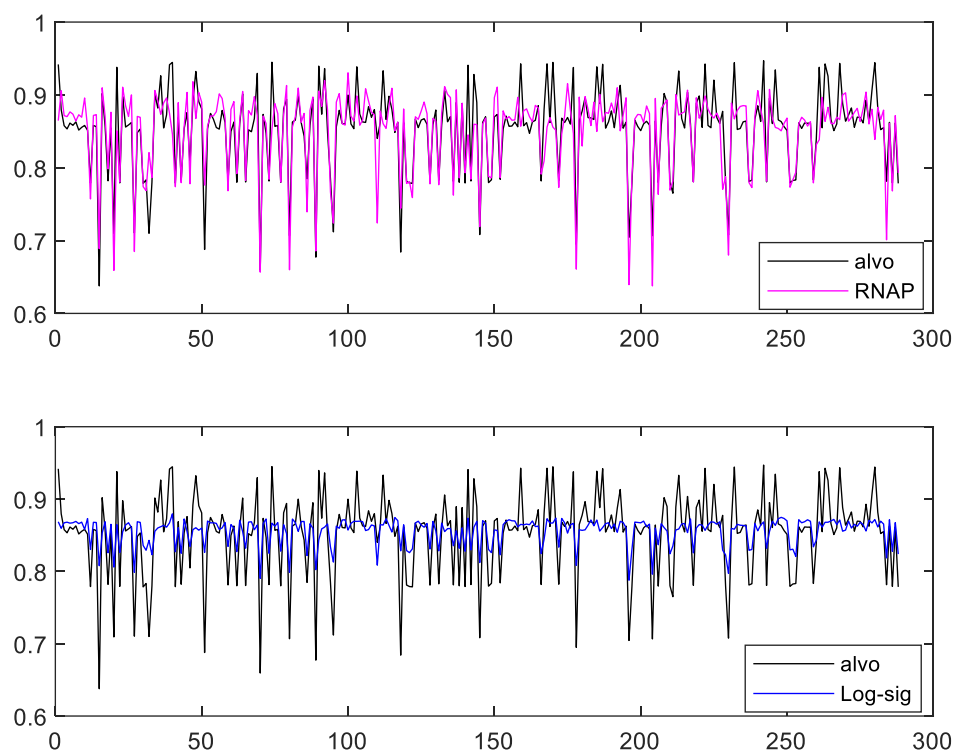
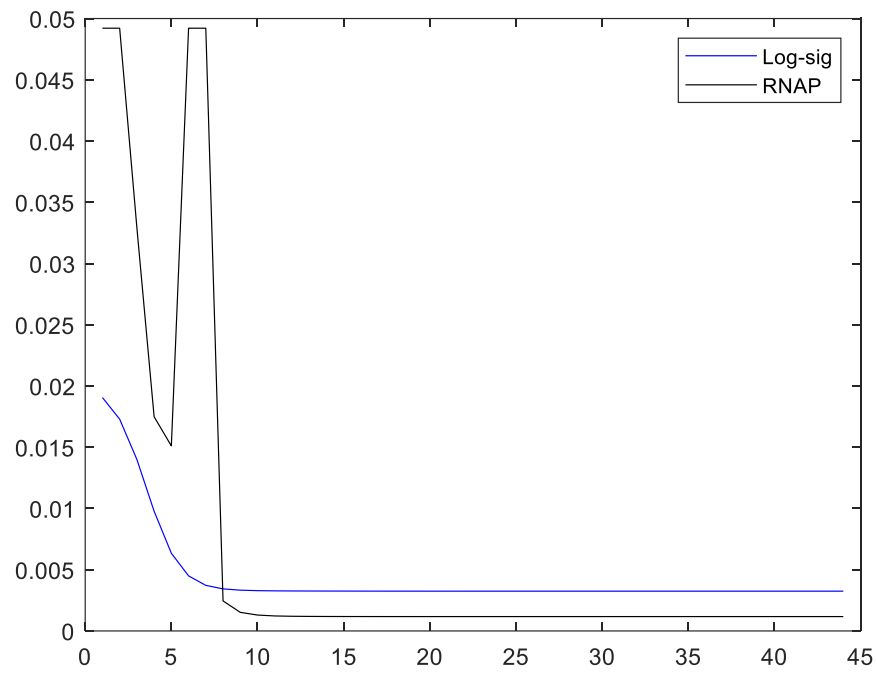
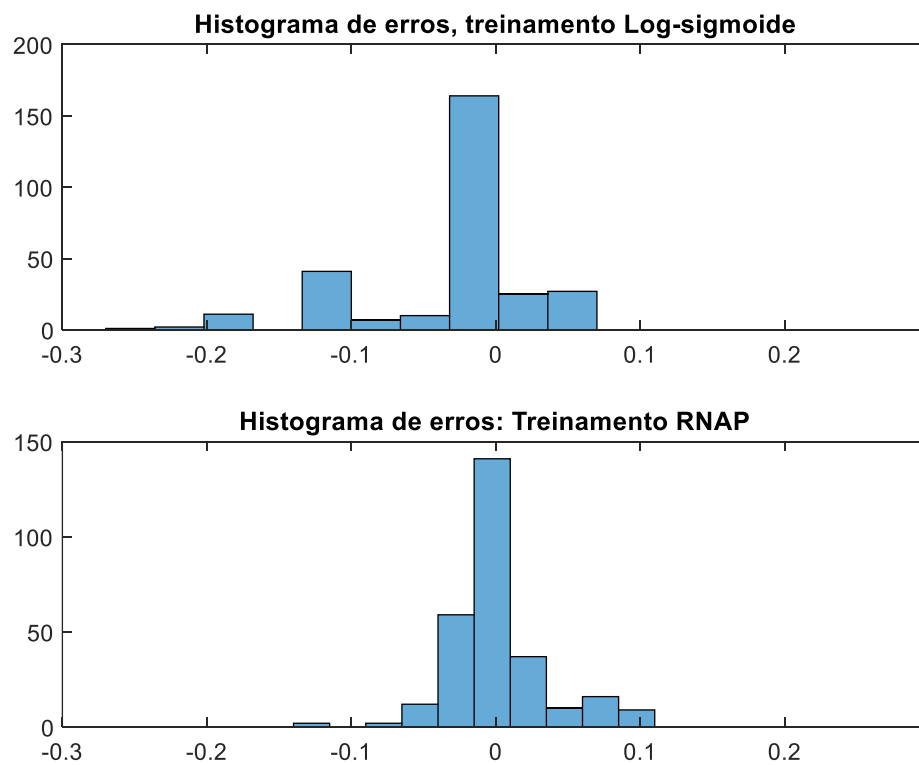


Figura 15. Comparação do Comportamento de Redes Neurais com Funções de Ativação Log-Sigmoide e RNAP.



(a)



(b)

Figura 16. Resultados de desempenho da RNAP: (a) comparação do MSE; (b) histograma de erros.

A Figura 17 compara os valores da saída com a RNAP, alimentada por entradas independentes do treinamento, equivalentes a 20% do conjunto total de dados. Para valores distribuídos de modo aleatório, a RNAP foi capaz de acompanhar as mudanças de valor da saída real.

Na Figura 18, os dados são rearranjados em ordem temporal, e a RNAP é novamente testada. Percebeu-se um desvio entre a saída temporal e a rede RNAP de aproximadamente $\pm 0,05$. No geral percebe-se que a RNAP tende a manter o erro estável para saídas cujo valor não varia bruscamente, porém é capaz de corrigir variações bruscas.

A Tabela 2 apresenta o comportamento do MSE para diferente número de neurônios da camada oculta. Estimar o número de neurônios ideais para camada oculta é uma tarefa complexa que requer muitas vezes experiência do desenvolvedor (GOMEZ, 2009) Considerando a rede com 2 entradas e uma saída, a arquitetura da RNAP tende a não aprender quando se acrescenta mais que 5 neurônios na camada oculta.

A camada de saída tenderá a apresentar um neurônio com demasiadas entradas, que sempre aproximará cada vez mais o valor de μ_E a 0,5, de acordo com a Equação (41).

Tabela 2 - Valores de MSE para diferentes tamanhos da camada oculta.

	2 neurônios	3 neurônios	4 neurônios	5 neurônios
Mínimo (10^{-3})	0,96	0,96	1,30	0,98
Máximo (10^{-2})	2,95	4,60	3,63	4,84
Média (10^{-2})	0,82	1,84	1,87	1,08
Desvio Padrão (10^{-3})	10,8	1,11	1,72	1,10

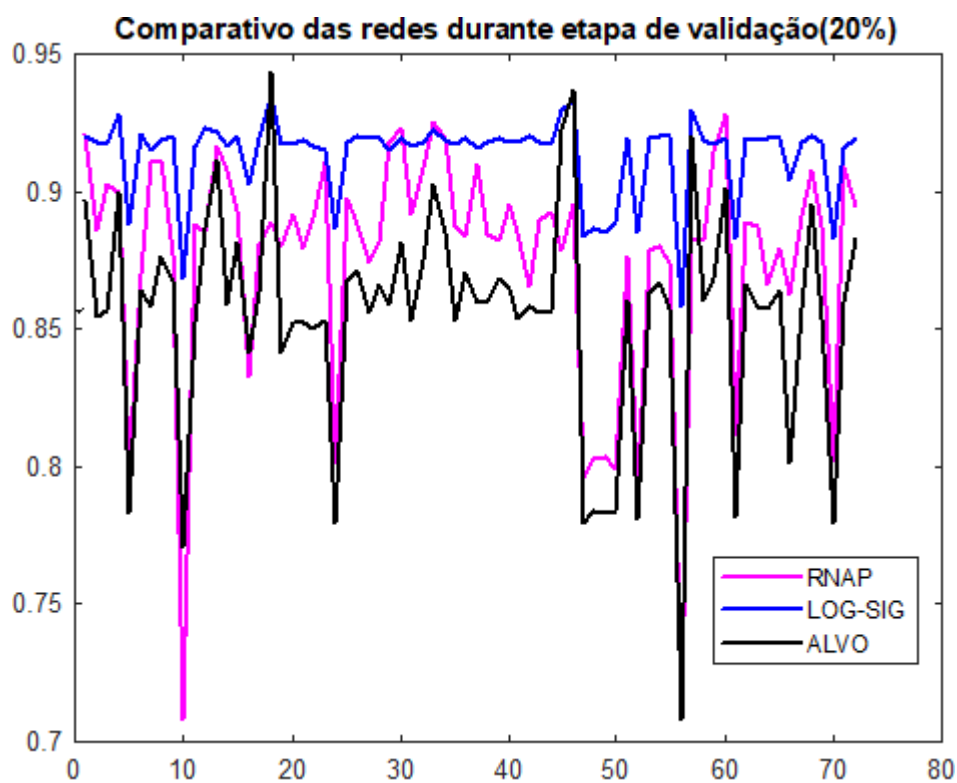


Figura 17. Validação da Redes.

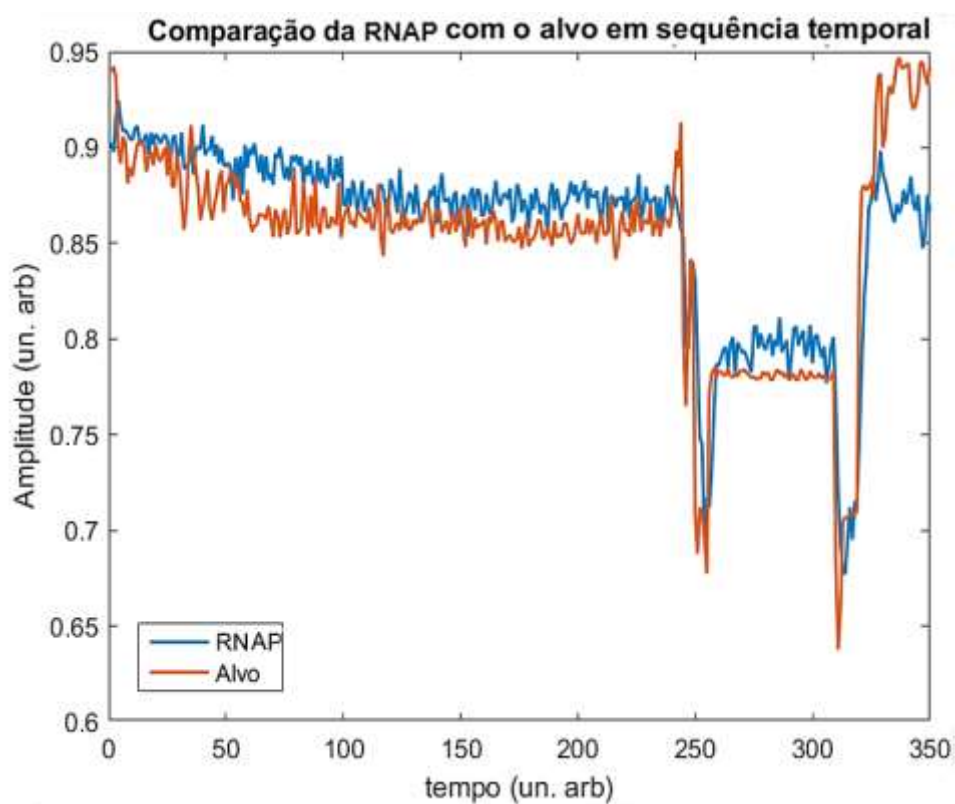


Figura 18. Comparação da RNAP e alvo em sequência temporal.

As emissões industriais são referidas aos subprodutos liberados durante os processos de produção e fabricação em indústrias. De fato, o mecanismo de Zeldovich apresentado na seção 1.3.1 indica que a TCA influencia na formação do T-NO_x. A Tabela 3 representa os valores finais dos pesos da RNAP. As Figuras 19 (a) e (b) ilustram o teste de propagação direta com entradas hipotéticas, de modo a se construir superfícies equipotenciais para análise comportamental. A Figura 19 (a) demonstrou que a Rede Log-sigmoide constrói superfícies em que ocorre variação brusca de valores. Além do mais, tanto para a variável X_1 fixa quanto para X_2 , a curva será crescente, o que demonstraria que ambas as variáveis de combustível e vazão de ar são favoráveis para produção de NO_x. O fato de não haver cruzamento entre superfícies indica que a Rede Log-sig não acusou interação entre as variáveis.

Na Figura 19 (b), mostra-se comportamento crescente para X_1 e decrescente para X_2 . O teste de propagação direta realizado com os pesos aplicados à rede RNAP, comparando o comportamento da Figura 19 (b) com a Tabela 3 e 4, mostram que não se pode confirmar o a natureza da rede apenas observando os valores dos pesos. A entrada X_2 , que equivale a vazão de ar no problema apresentado, teve pesos altos para as camadas ocultas 2 e 3, sendo uma entrada par (negativa). A propagação, porém, da camada oculta para a camada de saída faz com que a maior influência de X_2 seja positiva em relação a saída, conforme mostrado na 2ª linha da 5ª coluna da Tabela 5, que mostra como os pesos se propagam pela rede. Já a entrada X_1 , que é a vazão de combustível, esteve relacionada as entradas ímpares, mas teve seus pesos influenciados negativamente tanto pela camada seguinte, quanto pela entrada combinada X_1X_2 .

Tabela 3 - Pesos finais de treinamento da RNAP: (a) Entrada-Oculta.

	H_1	H_2	H_3
X_1	0,5806	0,2989	0,7137
X_2	0,3605	0,7185	0,9953
X_1X_2	0,0837	0.46820	0.0938

Tabela 4 - Pesos finais de treinamento da RNAP: Oculta-saída.

	Y_1
H_1	0,7252
H_2	0,9120
H_3	0,0864

Tabela 5 - Relação da influência das entradas propagadas pela RNAP

	H₁	H₂	H₃	H₁Y₁	H₂Y₁	H₃Y₁
X₁	(+)0,5806	(+)0,2989	(+)0,7137	(+)0,4211	(-)0,2725	(+)0,0616
X₂	(-)0,3605	(-)0,7185	(-)0,9953	(-)0,2614	(+)0,6553	(-)0,0860
X₁X₂	(+)0,0837	(+)0,46820	(+)0,0938	(+)0,0607	(-)0,4270	(+)0,0081

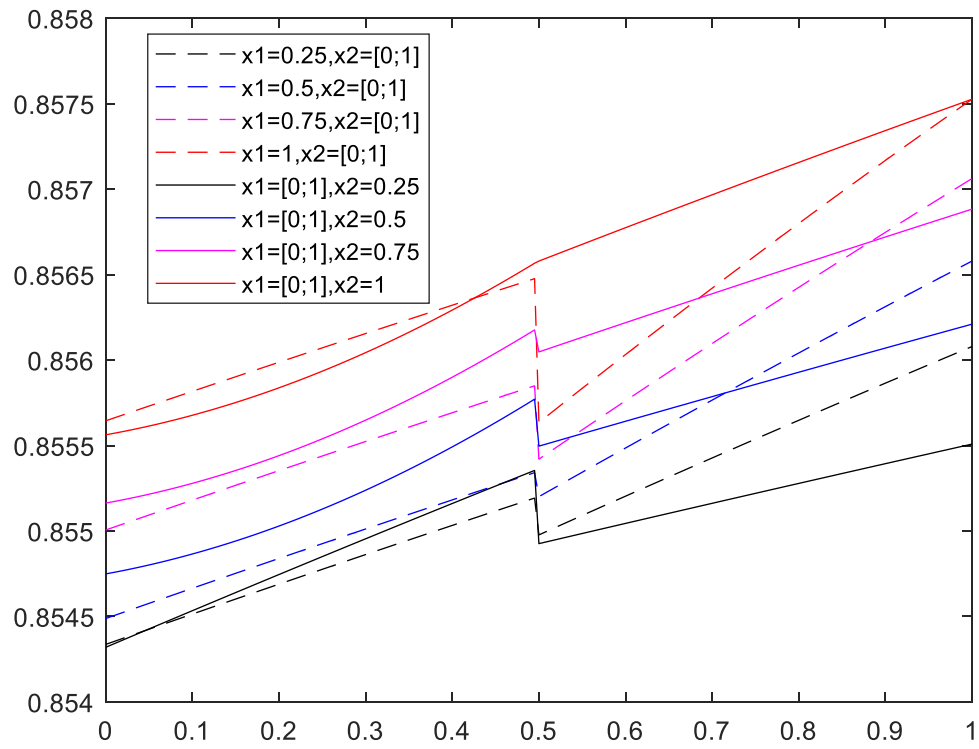
As emissões industriais são referidas aos subprodutos liberados durante os processos de produção e fabricação em indústrias. De fato, o mecanismo de Zeldovich apresentado na seção 1.3.1 indica que a TCA influencia na formação do T-NO_x. Uma vez que o excesso de O₂ diminui a temperatura, é de se esperar que haja uma diminuição da produção de T-NO_x à medida que o de O₂ aumente. Porém, X₂ representa a vazão de ar, que contém aproximadamente 79% de N₂ disponível, e aumentar esse valor contribui para a formação de NO_x através de sua decomposição.

Percebe-se na Figura 19 (b) que existe uma estabilização da produção de NO_x à medida que X₂ cresce, devido à queda do valor de TCA devido a entrada de Ar.

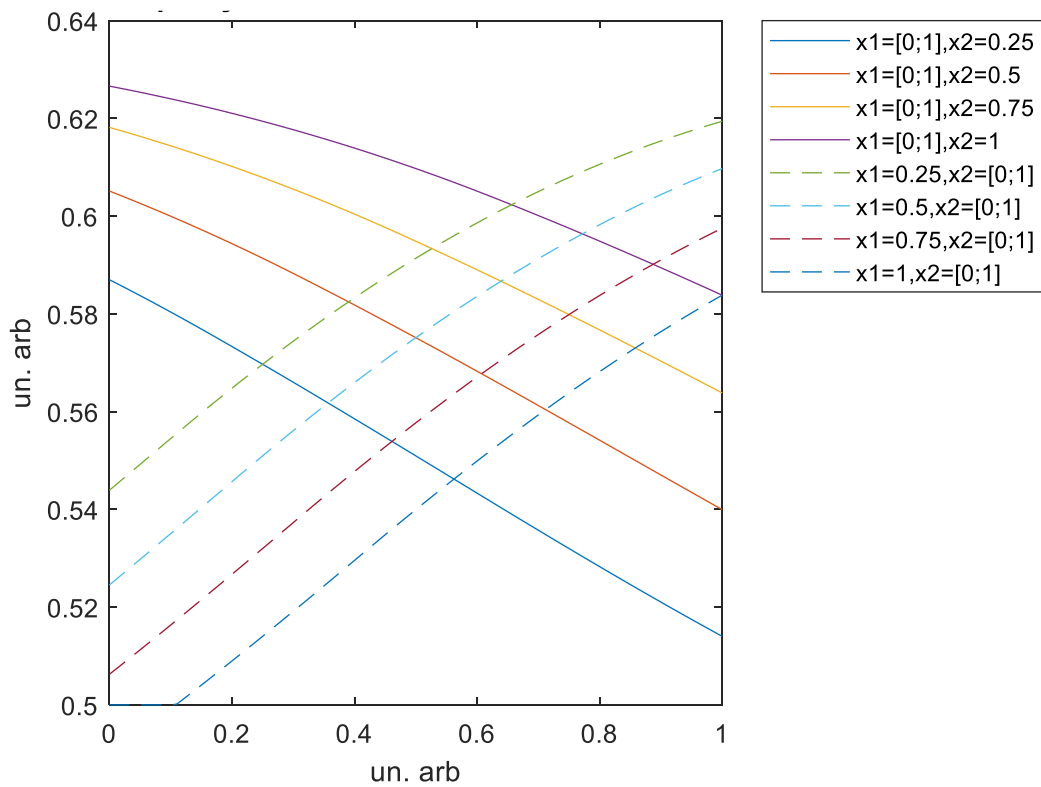
Em relação ao combustível, apesar de ser um aliado ao aumento da TCA, se colocado em excesso, acarretará a formação de outros gases (CO, CO₂). Aumentar seu valor irá utilizar o oxigênio para produção desses gases, que irá criar um déficit para a formação de NO_x.

Na Figura 19 (b), o acréscimo de combustível faz com que a formação de NO_x diminua de forma suave, devido ao aumento da TCA. À medida que a mistura se torna rica, a combustão se torna incompleta, e a TCA começará a diminuir, assim como haverá a formação de CO. Tal comportamento pode ser observado na acentuação das curvas de nível quando a partir de certo ponto de crescimento de X₁.

Pode se comparar o observado na Figura 19 (b) com a distribuição de espécies minoritárias conforme se aumenta a riqueza da mistura de combustão, conforme Figura 20 (TURNS, 2013), onde se representa a formação de NO.



(a)



(b)

Figura 19. Comparação da saída para diferentes valores de entrada. (a):RNA Log-sig. (b): RNAP.

A Figura 19 (b) Mostra que a não linearidade provocada pelo terceiro neurônio de entrada não é capaz de gerar pontos de encontro nas curvas de nível mostradas. A RNAP treinada não foi capaz então de demonstrar traços de sinergismo entre Combustível e Ar, para produção de NO_x . Tal resultado pode ser previsto ao analisar os pesos de X_1X_2 na tabela 4, onde verifica-se que seus valores estão bem abaixo em relação aos demais, apesar de estarem na mesma ordem de grandeza. Isso indicia que a natureza não linear das reações de produção de NO_x foram detectadas, embora não sejam capazes de demonstrar sinergismo.

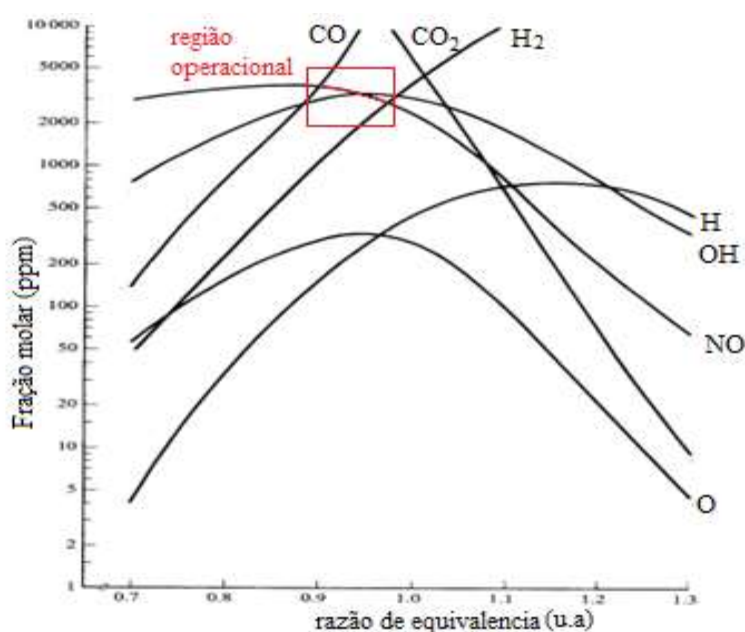


Figura 20. Relação entre distribuição de Espécies químicas minoritárias em relação a razão Combustível/Ar.

Fonte: (TURNS, 2013. Adaptado)

A estrutura computacional construída com a função de saída LPA2v-sigmoide foi capaz de reproduzir não linearidades ao passo que o único viés utilizado no neurônio serviu para manter sua continuidade.

A função “aditiva” foi capaz de admitir operações de subtração, possibilitando que os pesos sempre se mantenham no intervalo $[0,1]$. Com esta forma de estrutura computacional houve também um ajuste na obtenção da função aditiva μ_E , onde as CNAPs internas recebem o grau de evidência desfavorável de forma complementada. Deste modo, foi possível estabelecer

uma fórmula geral para qualquer número de entradas presentes, preservando sempre o modelo para entradas 2^N .

Verificou-se que com a nova função aditiva associada ao viés de continuidade, a derivada da função de ativação não possui mais descontinuidades, sendo também diferenciável para todo intervalo $[0,1]$. E assim, a possibilidade de se derivar a função de saída em 2º ordem permite que se possa trabalhar com matrizes hessianas, possibilitando então a abordagem do método LM para aprendizado da RNAP.

4 CONCLUSÕES

As emissões industriais são referidas aos subprodutos liberados durante os processos de produção e fabricação em indústrias. As emissões dos óxidos de nitrogênio (NOx) que ocorrem mais notadamente em indústrias química, petroquímica e metalúrgica, tornou-se uma preocupação significativa no contexto da poluição ambiental. Portanto, a cada dia torna-se mais importante as investigações que visam o encontro de modelos computacionais mais eficientes que possam assim trazer novas técnicas da detecção do NOx oferecendo condições as indústrias otimizar seus processos com diminuição das emissões. Considerando este problema, o presente trabalho apresentou um uma modelagem computacional capaz de estimar as intensidades das emissões dos óxidos de nitrogênio (NOx) em um forno industrial do tipo Cilindro Vertical de 6 queimadores. O modelo utilizado no tratamento dos dados envolveu configurações de Redes Neurais Artificiais construída com algoritmos fundamentados em Lógica Paraconsistente, que é um ramo da lógica que lida com situações em que há contradições e inconsistências nas informações. Ao contrário da lógica clássica, que assume que todas as proposições são; ou verdadeiras ou falsas, a lógica paraconsistente permite lidar com informações que podem ser simultaneamente verdadeiras e falsas, ou cuja veracidade não pode ser determinada de forma conclusiva. Para o estudo comportamentais das emissões dos óxidos de nitrogênio (NOx) foram formadas estruturas de algoritmos paraconsistentes compondo modelos de RNAP utilizando neurônios que tem seus pesos entre sinapses ajustados conforme vão aprendendo a detectar a saída, a partir da entrada de dados. De maneira geral verificou-se que a o modelo paraconsistente construído com a RNAP mostrou significativo aumento do índice de eficiência na tarefa de responder as variações mostradas nas emissões do NOx. Além disso, entre as características benéficas apresentadas destaca-se a sua capacidade em fazer com que a rede aprendesse a saída a partir de um conjunto de dados, ao invés de aprender uma única saída a partir de um conjunto de entradas. Isso permitiu uma melhor generalização da função alvo a partir do conjunto de dados de entrada. Comparando o desempenho do modelo construído com a RNAP com o construído com a Rede convencional do tipo log-sigmoide, também submetida ao método LM, verificou-se que apesar da RNAP apresentar um início mais desfavorável, ao longo do aprendizado os seus resultados mostraram-se melhores. Os valores obtidos quando comparados mostraram que o modelo construído com a RNAP apresentou o seu Erro Médio Quadrático - MSE menor que a Log-Sigmoide, o que se conclui que a estrutura construída com os algoritmos RNAP se mostrou mais adequado para sustentar arquiteturas computacionais em sistemas de detecção de NOx, sendo portanto, essencial para o aprimoramento de otimização de fornos industriais e assim colaborar para uma diminuição das emissões industriais desse tipo.

4.1 Trabalhos futuros

Nesta pesquisa a RNAP apresentada permitiu acompanhar o comportamento da produção de NOx a partir da alimentação de combustível e

ar. Para trabalhos futuros, sugere-se utilizar estas mesmas características para investigar a utilização de μ_{ECT} como função aditiva para a RNAP por ativação LPA2v-Sigmoide.

Por outro lado, verifica-se que neste trabalho as curvas equipotenciais foram levantadas considerando um forno industrial em regime de operação, onde sabe-se que por questões econômicas, as indústrias trabalham com excesso de O_2 , ou seja, com uma mistura levemente pobre, o que levou a comparação da Figura 19 (b) com a Figura 20. No futuro, seguindo esta mesma linha de pesquisa, pode-se então estudar o comportamento da rede para um experimento controlado que permita levantar a curva completa de emissão de NO_x , como mostra a Figura 20.

Considera-se também que em fornos industriais, a variável econômica mais relevante é a carga do produto aquecido, que interfere diretamente nos índices de produção. Desse modo, na constante busca por implementar sistemas de controle avançado, sugere-se como trabalho futuro um sistema de predição de NO_x a partir da carga de produção de um forno industrial.

Em relação as variáveis, percebeu-se também que a vazão de ar, por ser rica em N_2 , é uma variável favorável para a produção de NO_x , enquanto o aumento de combustível acaba por reduzir a produção do gás em discussão. Portanto, para trabalhos futuros, pode-se investigar a influência de outras variáveis, como a Pressão interna da câmara de combustão, que influencia no tempo de residência dos gases e, por consequência, a produção de T- NO_x .

REFERÊNCIAS

- AGARWAL, S. J. *Neural Networks and Deep Learning: A Textbook*. 1. ed. New York: Springer, 2018.
- BLOOMFIELD, J.; STEWARD, F. **Strategies for Climate Change Post COP26**. *The Political Quarterly*, v. 93. p. 278-287.
- DA SILVA FILHO J.I. *Treatment of Uncertainties with Algorithms of the Paraconsistent Annotated Logic*. **Journal of Intelligent Learning Systems and Applications**, Vol. 4 No. 2, 2012, pp. 144-153.
- DA SILVA FILHO, J.I.; ABE, J.M.; LAMBERT-TORRES, G. **Inteligência Artificial com as Redes de Análises Paraconsistentes** – Rio de Janeiro: LTC, 2008
- DE CARVALHO JUNIOR, A. **Identificação e Controle de Sistemas Dinâmicos com Rede Neural Paraconsistente**. 2021. 196p. Tese (Doutorado) – Programa de Engenharia Elétrica, Escola politécnica, Universidade de São Paulo, 2021.
- GODISH, Thad. *Air Quality*. 5. ed. Boca Raton: CRC Press, 2014.
- HILL, S. C; SMOOT, D. *Modeling of nitrogen oxides formation and destruction in combustion systems*. **Progress in Energy and Combustion Science**. v.26. p. 417-458
- KHAN,K; SAHAI, A. *A Comparison of BA, GA, PSO, BP and LM for Training Feed forward Neural Networks in e-Learning Context*. **I.J. Intelligent Systems and Applications**, 2012, 7, 23-29.
- KHODABANDEH, E; POURRAMEZAN, M; PAKRAVAN, M. H.. *Effects of excess air and preheating on the flow pattern and efficiency of the radiative section of a fired heater*. **Applied Thermal Engineering**, 105, 537–548. 2016
- KUTLUMUKHAMEDOV, A.R; SKIBA, D.V; BAKIROV, F.G. *Combined Method for Prediction of Carbon Monoxide Emission from Gas Turbine Combustion Chambers*. 2021.
- LI, Y. X; ZHAO, F. J; YU D.D. *Effect of nitrogen limitation on cell growth, lipid accumulation and gene expression in Chlorella sorokiniana*. **Brazilian Archives of Biology and Technology**, 2015. 58(3), 462–467.

MARTINS, W.; MEIRELES, V.; MELO, F.; NALINI, L.; NALINI, G. *Estilos de aprendizagem em educação a distância*. 2015

MILLER, J. A.; BOWMAN, C. T. *Mechanism and modeling of nitrogen chemistry in combustion*. **Progress in Energy and Combustion Science**, 15(4), 287–338. 1989

MINICZ, M.; MATUCK, G.; TASINAFFO, P.; DA SILVA FILHO, J.I.. Célula Neural Artificial Paraconsistente de Aprendizagem por Extração do Efeito da Contradição. *Seleção Documental*, 2014.

PARK, C; WON, S; KIM, C; CHOI, Y. *Effect of mixing CO₂ with natural gas–hydrogen blends on combustion in heavy-duty spark ignition engine*. **Fuel** 102 (2012). 299–304.

PAULO, K.C.S. **Identificação de padrões de sinais acústicos com base em classificação paraconsistente**. 2016. Tese (Doutorado em Ciências) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2016.

PAVLAK, L.P. **Previsão de Emissões de NO_x de uma Usina Termelétrica a Carvão por meio de Redes Neurais Artificiais**. 2019. Trabalho de Conclusão de Curso (Graduação) – Universidade Federal do Rio Grande do Sul, Escola de Engenharia, Curso de Engenharia Mecânica, Porto Alegre, BR-SR, 2016

PEPPLOW, L.A; SILVA, V.L; BETINI, R.C; PEREIRA, T.C.G. *Evaluation of Global Heating Reduction Potential with the Replacement of Electricity Supplied by the Local Concessionaire Via Solar Renewable Source*. **Brazilian Archives of Biology and Technology**. 2019. 62(spe), e19190003.

SAKI, H; *et al.* Study of relationship between nitrogen dioxide and chronic obstructive pulmonary disease in Bushehr, Iran, **Clinical Epidemiology and Global Health**, V.8, 2020, pp 446-449

SAMET, Jonathan M.; DOMINICI, Francesca; CURRIERO, Frank C.; COURSAK, Ivan; ZEGGER, Scott L. Fine Particulate Air Pollution and Mortality in 20 U.S. Cities, 1987–1994. **New England Journal of Medicine**, Boston, v. 343, n. 24, p. 1742-1749, 2000.

SMITH, Gregory P. *Combustion and Emission Control for Industrial Furnaces*. Oxford: Elsevier, 2019.

TOUSHMALANI, R. *Comparison result of inversion of gravity data of a fault by particle swarm optimization and Levenberg-Marquardt methods*. **SpringerPlus** **2**, 462 (2013).

URNS, S.R. *An Introduction to Combustion: Concepts and Applications*. 3. ed. New York: McGraw-Hill, 2013.

TUTTLE, J.F. Sustainable NO_x emission reduction at a coal-fired power station through the use of online neural network modeling and particle swarm optimization. **Control Engineering Practice**. v. 93, 2019 n.104167 p.1

WANG, S. *Numerical simulation of combustion process in different heating furnaces*. **Applied Thermal Engineering**, v. 201 A, 2022. n.117736. p.1

WESTBROOK, C.K., & DRYER, F.L. (1981). Simplified Reaction Mechanisms for the Oxidation of Hydrocarbon Fuels in Flames. *Combustion Science and Technology*, 27(1–2), 31–43. <https://doi.org/10.1080/00102208108946970>

Xu, S. *et al.* (2015). Data cleaning in the process industries. **Reviews in Chemical Engineering**, 31(5). 2015

YU, C.; HUI, S. *Research on Value Assessment and Compensation for Health Hazards of Urban Air Pollution-A Case Study of Urumqi*. **Brazilian Archives of Biology and Technology**. 2016. 59(spe), e16160503.

ZHENHAI, D. *Varied Slope Rules Self Adaptive Fuzzy Control Algorithm of Crude Oil Heating*. **International congress on educational information technology**. 2010. pp: 122 – 124, China September, 2010.

APENDICE A: DEMONSTRAÇÃO DE μ_{ECT} PARA N ENTRADAS

Para 2 valores

$$\mu_{ECT} = \frac{G_{CT} + 1}{2}$$

Para 4 Valores

$$\mu_{E1} = \frac{G_C + 1}{2}$$

$$\mu_{E2} = \frac{G_C + 1}{2}$$

$$\mu_{ECT3} = \frac{\mu_{E1} + (1 - \mu_{E2}) - 1}{2} = \frac{\mu_{E1} - \mu_{E2}}{2} + 0,5$$

$$\mu_{ECT3} = \frac{1}{2} \left(\frac{\mu_1 - \mu_2 + 1 - \mu_3 + \mu_4 - 1}{2} \right) = \frac{(\mu_1 - \mu_2 - \mu_3 + \mu_4)}{4} + 0,5$$

Para 8 Valores

$$\mu_{E5} = \frac{1}{4}(\mu_1 + \mu_3 - \mu_2 - \mu_4) + 0,5$$

$$\mu_{E6} = \frac{1}{4}(\mu_5 + \mu_7 - \mu_6 - \mu_8) + 0,5$$

$$\mu_{ECT7} = \frac{(\mu_{E5} - \mu_{E6})}{2}$$

$$\mu_{ECT7} = \frac{1}{2} \left(\frac{1}{4}(\mu_1 + \mu_3 - \mu_2 - \mu_4) + 0,5 - \frac{1}{4}(\mu_5 + \mu_7 - \mu_6 - \mu_8) - 0,5 \right) + 0,5$$

$$\mu_{ECT7} = \frac{1}{8}(\mu_1 + \mu_3 - \mu_2 - \mu_4 - \mu_5 - \mu_7 + \mu_6 + \mu_8) + 0,5$$

Prova: para 16 entradas

Proposição

$$\mu_{ECT} = \frac{(\mu_1 + \mu_3 + \mu_5 + \mu_7 - \mu_2 - \mu_4 - \mu_6 - \mu_8)}{16} + \frac{(\mu_{10} + \mu_{12} + \mu_{14} + \mu_{16} - \mu_9 - \mu_{11} - \mu_{13} - \mu_{15})}{16} + 0,5$$

Prova:

$$\mu_{E1} = \frac{\mu_1 - \mu_2}{2} + 0,5$$

$$\mu_{E2} = \frac{\mu_3 - \mu_4}{2} + 0,5$$

$$\mu_{E3} = \frac{\mu_5 - \mu_6}{2} + 0,5$$

$$\mu_{E4} = \frac{\mu_7 - \mu_8}{2} + 0,5$$

$$\mu_{E5} = \frac{\mu_9 - \mu_{10}}{2} + 0,5$$

$$\mu_{E6} = \frac{\mu_{11} - \mu_{12}}{2} + 0,5$$

$$\mu_{E7} = \frac{\mu_{13} - \mu_{14}}{2} + 0,5$$

$$\mu_{E8} = \frac{\mu_{15} - \mu_{16}}{2} + 0,5$$

$$\mu_{E9} = \frac{\mu_{E1} - (1 - \mu_{E2}) + 1}{2} = \frac{\mu_1 + \mu_3 - \mu_2 - \mu_4}{4} + 0,5$$

$$\mu_{E10} = \frac{\mu_{E3} - (1 - \mu_{E4}) + 1}{2} = \frac{\mu_5 + \mu_7 - \mu_6 - \mu_8}{4} + 0,5$$

$$\mu_{E11} = \frac{\mu_{E5} - (1 - \mu_{E6}) + 1}{2} = \frac{\mu_9 + \mu_{11} - \mu_{10} - \mu_{12}}{4} + 0,5$$

$$\mu_{E12} = \frac{\mu_{E3} - (1 - \mu_{E4}) + 1}{2} = \frac{\mu_{13} + \mu_{15} - \mu_{14} - \mu_{16}}{4} + 0,5$$

$$\mu_{E12} = \frac{\mu_{E3} - (1 - \mu_{E4}) + 1}{2} = \frac{\mu_{13} + \mu_{15} - \mu_{14} - \mu_{16}}{4} + 0,5$$

$$\mu_{E13} = \frac{\mu_{E9} - (1 - \mu_{E10}) + 1}{2} = \frac{\mu_1 + \mu_3 + \mu_5 + \mu_7 - \mu_2 - \mu_4 - \mu_6 - \mu_8}{8} + 0,5$$

$$\mu_{E14} = \frac{\mu_9 + \mu_{11} + \mu_{13} + \mu_{15} - \mu_{10} - \mu_{12} - \mu_{14} - \mu_{16}}{8} + 0,5$$

$$\mu_{ECT15} = \frac{\mu_{E13} - \mu_{E14}}{2} + 0,5$$

$$\mu_{ECT15} =$$

$$\frac{(\mu_1 + \mu_3 + \mu_5 + \mu_7 - \mu_2 - \mu_4 - \mu_6 - \mu_8)}{16}$$

$$- \frac{(\mu_9 + \mu_{11} + \mu_{13} + \mu_{15} - \mu_{10} - \mu_{12} - \mu_{14} - \mu_{16})}{16} + 0,5$$

APENDICE B: CÉLULA NEURAL DE APRENDIZAGEM COM MÉTODO DE LEVENBERG MARQUARDT- ALGORÍTMO EM MATLAB

```
% PARACONSISTENT NEURAL NETWORK WITH LEVENBERG MARQUADT METHOD

% AUTHOR: MARCOS CARNEIRO RODRIGUES, JOÃO INACIO DA SILVA FILHO

% Laboratory of Applied Paraconsistent Logic, Santa Cecilia
University-UNISANTA,

% Oswaldo Cruz Street 288,

% Santos 11045-907, SP, Brazil


% This algorithm trains a new neural network, called the Paraconsistent
% Neural Network (PNN) (CARVALHO JUNIOR, 2021), using the
% Levenberg-Marquardt learning algorithm. The functions 'CNAP_nn' and
% 'dCNAP_nn' are used to find, respectively, the resultant certain
degree,

% normalized real certain degree and the resulting degree of
certainty, the

% normalized actual degree of certainty, and their partial
derivatives.


% for this example, we train a dataset containing 360 samples of
normalized

% and randomized data, with two inputs and a single output. the
hidden

% layer has 3 neurons with 2 inputs each. for use this model, we
suggest

% edit the lines 22 to 28.

%%

clc, close all, clear all;


%% Data settings
data=table2array(readtable('test.xlsx')); % read the sample
data=data(1:360,:);
array_rand=randperm(size(data,1));
rand_data=data(array_rand,:);
entrada=rand_data([1:324],[1:2]);
entrada(:,3)=entrada(:,1).*entrada(:,2);
alvo=rand_data(1:324,3);
%% Levenberg Marquadt Critéria
```

```

eta=0.001; % ajust parameter
fail=0;
max_fail=6;
MSE_FAIL=0; % Initialize the MSE Criteria

nInputNeuron =3;      %Define
nHiddenNeuron = 3;      % Número de neurônios na camada oculta
nOutputNeuron = 1;      % Número de neurônios na camada de saída

% PESOS ALEATÓRIOS W(i,j,l)i: n° deneuronios da camada l, j:n° de
neuronios
%da camada l-1
W1=rand(nInputNeuron,1);
W2=rand(nHiddenNeuron,nInputNeuron);
W3=rand(nOutputNeuron,nHiddenNeuron);

WRNP_2=W2; %Hidden Weights (ramdom)
WRNP_3=W3; %Output Weights (ramdom)

epoch=1000; % define the number of train epochs

%% TEST RNP
%ymin=1e+10;  %[eliminar]
%ymax=-1e+10; %[eliminar]
for k=1:epoch % Epoch loop
    E_RNP=0; % Reset the Error Vector for each starting epoch
    WRNP_3_ANT=WRNP_3; % save previous weights
    WRNP_2_ANT=WRNP_2; % save previous weights
    WRNP_3_V=reshape(WRNP_3,1,numel(WRNP_3)); %reshape the matrix
Weights in a single vector
    WRNP_2_V=reshape(WRNP_2,1,numel(WRNP_2)); %reshape the matrix
Weights in a single vector
    WRNP=[[WRNP_2_V],[WRNP_3_V]]; %reshape the matrix
Weights in a single vector
    HIST_WRNP(k,:)=WRNP; % create a Weighs
history Matrix

```

```

for m=1:length(alvo) % sample error scan loop
    Z0=entrada(m,:)' ; % transpose input vector
    t=alvo(m); % return the m-row output
    %% FORWARD PROPAGATION
    % Input Layer

    S_RNP_1=max(0,min(1,Z0));
    Z_RNP_1=S_RNP_1;

    % Hidden Layer

    for i=1:3 % Set MI for first input
        if WRNP_2(:,1)<0
            MI_RNP_2=abs(WRNP_2(:,1)).*(1-Z_RNP_1(1,:));
        else
            MI_RNP_2=WRNP_2(:,1).*(Z_RNP_1(1,:));
        end
    end

    for i=1:3 % Set MI for second input
        if WRNP_2(:,2)<0
            LDA_RNP_2=abs(WRNP_2(:,2)).*(1-Z_RNP_1(2,:));
        else
            LDA_RNP_2=WRNP_2(:,2).*(Z_RNP_1(2,:));
        end
    end

    for i=1:3 % Calculate MI,Mier and dMier
        [mie_2(i,:),mier_2(i,.)]=CNAP_nn([MI_RNP_2(i,:),LDA_RNP_2(i,.)]);
        dmier_2(i,.)=dCNAP_nn(ones(2,1),mie_2(i,.) );
    end
    Z_RNP_2=mier_2;
    dZ_RNP_2=dmier_2;

    % Output Layer

```



```

        if WRNP_3(i)<0 %same as line 69
            S_RNP_3=abs(WRNP_3)*(1-Z_RNP_2);
        else
            S_RNP_3=WRNP_3*(Z_RNP_2);
        end

        [mie_3,mier_3]=CNAP_nn(S_RNP_3); % same as Line 78
        dmier_3=dCNAP_nn(S_RNP_3,mie_3); % same as Line 79
        Z_RNP_3=mier_3;
        dZ_RNP_3=dmier_3;

        %Delta evaluating

        D_RNP_3=dZ_RNP_3;
        D_RNP_2=WRNP_3'*D_RNP_3.*dZ_RNP_2;

        %% Jacobian Matrix
        GW_RNP_3=D_RNP_3*Z_RNP_2';
        GW_RNP_2=D_RNP_2*Z_RNP_1';
        GW_RNP_3_V=reshape(GW_RNP_3,1,numel(GW_RNP_3));
        GW_RNP_2_V=reshape(GW_RNP_2,1,numel(GW_RNP_2));
        PRNP=[[GW_RNP_2_V],[GW_RNP_3_V]];
        yp(m,:)=Z_RNP_3;
        JRNP(m,:)=PRNP;

    end

    yp=(yp-min(yp))*(max(alvo)-min(alvo))/(0.001+abs(max(yp)-min(yp)))+min(alvo); %normalize the output with respect to the
    target (resizing). 0.001 ensures that there is no division by zero

    E_RNP=alvo-yp; %Error Vector

    HRNP=JRNP'*JRNP; % Get the Hessian Matrix

    [mhrnp,nhrnp]=size(HRNP); % get HRNP dimensions

    IHRNP=diag(HRNP).*eye(mhrnp,nhrnp); % create an diagonal matrix

    MSE_ANT_RNP=E_RNP'*E_RNP; % Calculate the MSE

    omega_RNP=eta*IHRNP;

    grnp=JRNP'*E_RNP;

```

```

    dWRNP=inv(HRNP+omega_RNP)*grnp; % Calculate the LM increment to
weight
    WRNP=WRNP-dWRNP';
    p2r=numel(GW_RNP_2_V);
    WRNP_2= reshape(WRNP(1:p2r),[size(WRNP_2)]);
    WRNP_3= reshape(WRNP(p2r+1:end),[size(WRNP_3)]); % Reshape the
weight Vector to Original matrix

```

```

%% MSE TEST

```

```

% Redo Forward Propagation for comparison

```

```

for m=1:length(alvo)

```

```

    Z0=entrada(m,:)';

```

```

    t=alvo(m);

```

```

% FORWARD PROPAGATION

```

```

% Input Layer

```

```

S_RNP_1=max(0,min(1,Z0));

```

```

Z_RNP_1=S_RNP_1;

```

```

% Hidden Layer 1

```

```

for i=1:3

```

```

    if WRNP_2(:,1)<0

```

```

        MI_RNP_2=abs(WRNP_2(:,1)).*(1-Z_RNP_1(1,:));

```

```

    else

```

```

        MI_RNP_2=WRNP_2(:,1).*(Z_RNP_1(1,:));

```

```

    end

```

```

end

```

```

for i=1:3

```

```

    if WRNP_2(:,2)<0

```

```

        LDA_RNP_2=abs(WRNP_2(:,2)).*(1-Z_RNP_1(2,:));

```

```

    else

```

```

        LDA_RNP_2=WRNP_2(:,2).*(Z_RNP_1(2,:));

```

```

    end

```

```

end

```

```

for i=1:3

```

```

[mie_2(i,:),mier_2(i,:)]=CNAP_nn([MI_RNP_2(i,:),LDA_RNP_2(i,:)]);
    dmier_2(i,:)=dCNAP_nn(ones(2,1),mie_2(i,:));
end
Z_RNP_2=mier_2;
dZ_RNP_2=dmier_2;
% Output Layer
    if WRNP_3(i)<0
        S_RNP_3=abs(WRNP_3)*(1-Z_RNP_2);
    else
        S_RNP_3=WRNP_3*(Z_RNP_2);
    end
[mie_3,mier_3]=CNAP_nn(S_RNP_3);
dmier_3=dCNAP_nn(S_RNP_3,mie_3);
Z_RNP_3=mier_3;
dZ_RNP_3=dmier_3;
yp(m,:)=Z_RNP_3;

end

yp=(yp-min(yp))*(max(alvo)-min(alvo))/(0.001+abs(max(yp)-
min(yp)))+min(alvo);
E_RNP=alvo-yp;
MSE_RNP=E_RNP'*E_RNP;

if MSE_RNP<1e-4 % first LM stop Criteria
    break;
end

if MSE_RNP>=MSE_ANT_RNP
    if MSE_RNP>MSE_FAIL
        fail=fail+1;
    end
    MSE_FAIL=MSE_RNP;
    WRNP_2=WRNP_2_ANT;
    WRNP_3=WRNP_3_ANT;
    eta=2*eta;

```

```

else
    eta=.5*eta;
end
HIST_MSE_RNP(k)=MSE_RNP/m;

HIST_MSE_ANT_RNP(k)=MSE_ANT_RNP/m;
if log10(eta)>10    % second LM stop Criteria (a)
    break;
end
if log10(eta)<-10   % second LM stop Criteria (b)
    break;
end
if fail>max_fail    % third LM stop Criteria
    break;
end
end
if(mod(k,100)==0)
k=k
MSE_RNP=MSE_RNP
end

end

plot(1:k,HIST_MSE_RNP(1:k),'-k');
hold on;
title('Epoch x Mean Square Error ');
xlabel ('Epoch (a.u)');
ylabel('MSE (a.u)');
legend('PNN');
figure()

xlabel('sample');
ylabel('value(a.u)');
plot(alvo,'-k');
hold on
plot(yp);

```

```

title('Target and PNN Comparison');
legend('target','PNN','Position',[.2 .15 .2 .1]);
xlabel('sample');
ylabel('value(a.u)');

%% Validate RNP
valid_input=rand_data([325:360],[1:2]);
valid_input(:,3)=rand_data([325:360],3);
valid_target=rand_data([325:360],3);
    for m=1:length(valid_target)
        Z0=valid_input(m,:)';
        t=valid_target(m);

        % FORWARD PROPAGATION

        % Input Layer
        S_RNP_1=max(0,min(1,Z0));
        Z_RNP_1=S_RNP_1;

        %Hidden Layer
        for i=1:3
            if WRNP_2(:,1)<0
                MI_RNP_2=abs(WRNP_2(:,1)).*(1-Z_RNP_1(1,:));
            else
                MI_RNP_2=WRNP_2(:,1).*(Z_RNP_1(1,:));
            end
        end
        for i=1:3
            if WRNP_2(:,2)<0
                LDA_RNP_2=abs(WRNP_2(:,2)).*(1-Z_RNP_1(2,:));
            else
                LDA_RNP_2=WRNP_2(:,2).*(Z_RNP_1(2,:));
            end
        end
        for i=1:3
            [mie_2(i,:),mier_2(i,:)]=CNAP_nn([MI_RNP_2(i,:),LDA_RNP_2(i,:)]);

```

```

        dmier_2(i,:)=dCNAP_nn(ones(2,1),mie_2(i,:));
    end
    Z_RNP_2=mier_2;
    dZ_RNP_2=dmier_2;

    %Output Layer
    if WRNP_3(i)<0
        S_RNP_3=abs(WRNP_3)*(1-Z_RNP_2);
    else
        S_RNP_3=WRNP_3*(Z_RNP_2);
    end
    [mie_3,mier_3]=CNAP_nn(S_RNP_3);
    dmier_3=dCNAP_nn(S_RNP_3,mie_3);
    Z_RNP_3=mier_3;
    dZ_RNP_3=dmier_3;
    v_yp(m,:)=Z_RNP_3;

end

v_yp=(v_yp-min(v_yp))*(max(valid_target)-min(valid_target))/(0.001+abs(max(v_yp)-min(v_yp)))+min(valid_target);
v_E_RNP=valid_target-v_yp;

MSE_RNP=MSE_RNP/324
v_MSE_RNP=v_E_RNP'*v_E_RNP/(360-325)
r_MSE=v_MSE_RNP/MSE_RNP
figure()
plot(v_yp, 'LineWidth',1.3);
hold on
plot(valid_target, 'LineWidth',1.3)
title('Validate stage comparison');
legend('PNN','Target','Position',[.2 .15 .2 .1]);
xlabel('sample');
ylabel('value(a.u)');

figure()
histogram(E_RNP,10)

```

```

xlim([-0.3 0.3])
xlabel('sample');
ylabel('value(a.u)');
title('Error Histogram: Trained PNN')
legend('PNN');
%% testing RNP with not randomize inputs

valid_input=data(:,[1:2]);
valid_input(:,3)=valid_input(:,1).*valid_input(:,2)
valid_target=data(:,3);
for m=1:length(valid_target)
    Z0=valid_input(m,:)';
    t=valid_target(m);

    % FORWARD PROPAGATION
    %CAMADA 1
    S_RNP_1=max(0,min(1,Z0));
    Z_RNP_1=S_RNP_1;
    %CAMADA 2
    for i=1:3
        if WRNP_2(:,1)<0
            MI_RNP_2=abs(WRNP_2(:,1)).*(1-Z_RNP_1(1,:));
        else
            MI_RNP_2=WRNP_2(:,1).*(Z_RNP_1(1,:));
        end
    end
    for i=1:3
        if WRNP_2(:,2)<0
            LDA_RNP_2=abs(WRNP_2(:,2)).*(1-Z_RNP_1(2,:));
        else
            LDA_RNP_2=WRNP_2(:,2).*(Z_RNP_1(2,:));
        end
    end
    for i=1:3
        [mie_2(i,:),mier_2(i,:)]=CNAP_nn([MI_RNP_2(i,:),LDA_RNP_2(i,:)]);
    end
end

```

```

        dmier_2(i,:)=dCNAP_nn(ones(2,1),mie_2(i,:));
    end
    Z_RNP_2=mier_2;
    dZ_RNP_2=dmier_2;
    %CAMADA 3
    if WRNP_3(i)<0
        S_RNP_3=abs(WRNP_3)*(1-Z_RNP_2);
    else
        S_RNP_3=WRNP_3*(Z_RNP_2);
    end
    [mie_3,mier_3]=CNAP_nn(S_RNP_3);
    dmier_3=dCNAP_nn(S_RNP_3,mie_3);
    Z_RNP_3=mier_3;
    dZ_RNP_3=dmier_3;
    v_yp(m,:)=Z_RNP_3;
    %    E_RNP(m,:)=t-yp;
    %    y(m,:)=yp;
end

v_yp=(v_yp-min(v_yp))*(max(valid_target)-
min(valid_target))/(0.001+abs(max(v_yp)-min(v_yp)))+min(valid_target);

figure

plot (data(1:360,3))
hold on
plot (v_yp)
xlim([1 360])
title('Target and PNN Comparison with time-series samples');
xlabel('sample');
ylabel('value(a.u)');
legend('target','PNN','Position',[.2 .15 .2 .1]);

function [mie_n,mier_n,miect_n]=CNAP_nn (arg1)
%dbstop if error;
soma_impair=0;
soma_par=0;
n_args=numel(arg1);

```



```

% Finds the next power of 2 greater than or equal to the number of
arguments

p=ceil(log2(n_args));

% Calculates the number of zeros that need to be added
n_zeros=2^p-n_args;

% Pad with zeros until the next power of 2
args = [arg1, zeros(1, n_zeros)];

%MIE
for i=1:length(args)
    if mod(i,2)==1
        soma_impar=soma_impar+args(i);
    else
        soma_par=soma_par+args(i);
    end
    n=i;
end
mie_n=1/(n)*(soma_impar-soma_par)+0.5;
GC=mie_n-0.5;
GCT=mie_n-0.5;

%MIER
D=((1-abs(GC))^2+GCT^2)^0.5;
if D>1
    D=1;
    GCR=0;
else
    if GC>0
        GCR=1-D;
    else
        if GC<0
            GCR=D-1;
        else
            GCR=0;
        end
    end
end

```

```

        end
    end
end
mier_n = (GCR+1)/2;
end

```

```

function y=dCNAP_nn (mie)
%dbstop if error;

```

```

    if mie>0.5

        y=(1-mie)*(2*mie^2-4*mie+5/2)^0.5/(2*mie^2-4*mie+5/2);

    else if mie<0.5
        y=mie*(2*mie^2+0.5)^0.5/(2*mie^2+0.5);
    else
        y=0.5;
    end
end
end

```