

Gabriel Augusto N. S. Costa

**Redes Neurais Artificiais como Técnica de Regularização
de Problemas Inversos em Transferência de Calor**

Orientadora: Dra. Juliana de Oliveira

Trabalho de Conclusão de Curso
apresentado à Faculdade de Ciências e
Letras da Universidade Estadual Paulista
“Júlio de Mesquita Filho” – Câmpus de
Assis, para obtenção do grau de Bacharel
em Engenharia Biotecnológica

**Assis
2022**

RESUMO

Em processos industriais térmicos a temperatura é captada por sensores que oferecem os sinais térmicos com ruídos intrínsecos ao processo, por isso técnicas para atenuar os efeitos nocivos dos ruídos em um sinal são consideradas ferramentas de grande importância. Assim, este projeto de pesquisa tem por objetivo atestar a viabilidade de Redes Neurais Artificiais (RNA's) como técnica de regularização de sinais oriundos de um problema inverso mal posto. Diversas metodologias para processo de desenvolvimento de RNA's são testadas, desde o tratamento de dados de entrada, formato de apresentação dos sinais, arquitetura das redes, entre outros. Modelos aproximadores de função eficientes e suas peculiaridades foram testadas e analisadas, baseando-se tanto no padrão das curvas tanto em parâmetros termodinâmicos. Foi observado que a maneira com a qual relacionamos as informações entregues aos modelos, e conseqüentemente, como os dados de entrada são transformados tem forte influência sobre o desempenho da regularização e aplicabilidade da rede.

Palavras-chave: Problemas inversos, processos industriais térmicos, redes neurais artificiais, regularização.

ABSTRACT

In thermal industrial processes, the temperature is captured by sensors that provide thermal signals with noise intrinsic to the process, so techniques to attenuate the harmful effects of noise in a signal are considered tools of great importance. Thus, this research project aims to attest the viability of Artificial Neural Networks (ANN's) as a technique for regularizing signals from a poorly posed inverse problem. Several methodologies for the ANN development process are tested, from the treatment of input data, signal presentation format, network architecture, among others. Efficient function approximation models and their peculiarities were tested and analyzed, based both on the pattern of the curves and on thermodynamic parameters. It was observed that both the way the information gets fed to the network and consequently the way the input data is preprocessed have major influence on regularization performance and model applicability.

Keywords: Inverse problems, industrial thermal processes, artificial neural network, regularization.

SUMÁRIO

1. INTRODUÇÃO	4
2. OBJETIVOS.....	5
3. MATERIAL E MÉTODOS	6
3.1 Formulação do Problema Inverso	6
3.2 Banco de Dados dos sinais e análise de parâmetros	8
3.3.1. Aplicação de uma RNA para o problema inverso	14
3.3.2. Regressão Ponto a Ponto.....	15
3.3.3. Regressão Multivariável	17
3.3.4. Previsão de Série Temporal	17
3.3.5. Ferramentas e implementação em código	19
4. RESULTADOS E DISCUSSÃO	20
4.1. Desempenho dos Modelos	20
4.1.1. Regressão ponto a ponto	21
4.1.2. Regressão Multivariável	25
4.1.3. Previsão da Série Temporal	29
5. CONCLUSÃO	35
6. REFERÊNCIAS.....	36

1. INTRODUÇÃO

Em 1943 o psiquiatra e neuroanatomista Warren Sturgis McCulloch e o matemático Walter Pitts desenvolveram o primeiro modelo matemático de um neurônio artificial, seguindo a lei do “tudo ou nada”, inspirado em um neurônio biológico (MCCULLOCH; PITTS, 1943). Assim, surgiram as Redes Neurais Artificiais (RNA's), definidas como modelos computacionais inspirados no sistema nervoso de seres vivos (DA SILVA *et al.*, 2010).

Estas máquinas, RNA's, projetadas para modelar a forma como o cérebro realiza uma tarefa (HAYKIN, 1999) possuem características que as tornaram populares entre os cientistas da área de Inteligência Artificial (IA), como adaptação por experiência, capacidade de aprendizado, habilidade de generalização, organização de dados, tolerância às falhas, não linearidade dos modelos, armazenamento distribuído, facilidade de prototipagem (HAYKIN, 1999 e DA SILVA *et al.*, 2010).

As RNA's podem ser usadas nas mais diversas aplicações, entre elas, aproximação de funções, controle de processos, reconhecimento e classificação de padrões, agrupamentos de dados, sistemas de previsão, otimização de sistemas e memórias associativas (HAYKIN, 1999 e DA SILVA *et al.*, 2010). Estas aplicações contemplam diferentes áreas do conhecimento, como militar, medicina, química, biologia, finanças, economia, ecologia, etologia, energia, robótica, farmácia, acústica, engenharias, ciências, processamento de sinais e imagens, indústrias alimentícias, automotivas e aeroespaciais, etc (DAYHOFF; DELEO, 2001, DA SILVA *et al.*, 2010).

Com esta vasta área de aplicações práticas e computadores com maior processamento, as RNA's têm obtido avanços consideráveis em pesquisa e consequente progresso nos projetos de uma RNA, que englobam o desenvolvimento em arquiteturas de redes, processos de treinamento, tipos de redes, topologias, algoritmos de treinamento, entre outros parâmetros (HAYKIN, 1999 e DA SILVA *et al.*, 2010).

As RNA's têm por tarefa aprender sobre uma informação real através de um modelo suficientemente consistente com a realidade da aplicação desejada. Desta maneira, os dados apresentados as RNA's são medidas reais obtidas por sensores projetados para sondar o ambiente no qual a RNA irá trabalhar. Tais medidas são, normalmente, ruidosas, sendo sujeitas a erros devido a ruído do sensor e imperfeições do sistema (HAYKIN, 1999). Assim, quanto melhor os dados apresentados as RNA's, melhor será o seu desempenho. Outra classe de problemas que também são dependentes da qualidade dos dados apresentados, e tem se destacado nas mais diversas áreas do conhecimento e com variadas aplicações práticas, é a de Problemas Inversos (PI's).

Os PI's, os quais unem a análise matemática dos problemas aos dados experimentais, estão em contínua ascensão e são extremamente sensíveis a erros experimentais e numéricos a ponto de comprometer completamente os resultados obtidos. Os PI's são intrinsecamente mal condicionados e por causa desta instabilidade, sua solução é extremamente complexa, motivo este de constantes estudos na área. Assim sendo, os PI's devem ser tratados de forma a minimizar a introdução de erros que podem corromper os resultados (SILVA NETO; MOURA NETO, 2000).

Matematicamente, os PI's pertencem à classe de problemas mal-postos. No início do século passado (1902) o matemático francês Jacques Hadamard definiu um problema bem-posto como sendo aquele que cumpre três condições: existência de solução, unicidade da solução e regularidade, ou seja, a solução tem uma dependência contínua (suave) com os dados de entrada (HADAMARD, 1902). Assim, o problema é dito mal-posto se alguma das condições de Hadamard não é satisfeita.

Em 1977 os matemáticos Tikhonov e Arsenin apresentaram uma técnica, chamada de regularização, como maneira de trabalhar com os PI's, na qual procuraram contornar as dificuldades associadas ao problema da não dependência suave da solução aos dados de entrada (TIKHONOV; ARSENIN, 1977). Desde então, muitos métodos para a solução de PI's foram desenvolvidos. Em 1990 Poggio e Girosé aplicaram RNA's na regularização de Tikhonov (POGGIO; GIROSE, 1990).

Métodos numéricos para a solução de aplicações envolvendo problemas inversos lineares, como, a regularização de Tikhonov citada acima, já estão relativamente estabelecidos. No entanto, quando comparados a uma abordagem dinâmica utilizando RNA's, estas têm se mostrado mais rápidas e estáveis, o que acarretou no desenvolvimento de uma variedade de algoritmos adaptados para a solução desses problemas mal-postos lineares e não lineares (BRAGA *et al.*, 2016).

E desde então RNA's têm se mostrado bastante efetivas quando usadas como uma ferramenta de alisamento e redução de ruído. O principal objetivo do projeto de uma rede para aproximação de sinais é uma boa aproximação no seu mapeamento entrada-saída. Além de se desenvolver uma aproximação convencional, procura-se também conferir à rede a capacidade de generalização, para que a mesma possa operar com as variedades morfológicas de sinais similares (BADRI, 2010).

2. OBJETIVOS

Dentro deste contexto, este trabalho tem por objetivo averiguar o desempenho de diferentes métodos dentro da aplicação de uma RNA no problema inverso proposto. As redes desenvolvidas devem ser capazes de apresentar como resultado um aproximador de

funções o qual se comporta como um alisador de sinais específicos, tendo como dados de entrada sinais térmicos obtidos de um modelo matemático inverso no qual o processo térmico é afetado pelo acúmulo térmico, convecção e radiação dos materiais envolvidos. A finalidade é suavizar os ruídos inerentes a esses sinais os quais são amplificados devido ao modelo inverso se comportar como um filtro passa-alta.

Como objetivos específicos, este trabalho pretende determinar os melhores parâmetros para as RNA's propostas, de tal forma que se obtenham as arquiteturas das redes mais adequadas para cada método empregado, e consequentes processos de treinamento, tipos de redes, topologias, algoritmos de treinamento, entre outros parâmetros.

3. MATERIAL E MÉTODOS

3.1 Formulação do Problema Inverso

A formulação do problema inverso está baseada em Oliveira *et al.* (2006). Considere uma sonda térmica encapsulada imersa num escoamento bifásico reagente como indicado na figura 1. A temperatura real do processo, T_{proc} , representa a temperatura local instantânea, já a temperatura indicada pela sonda térmica, T_{ind} , corresponde a temperatura do escoamento.

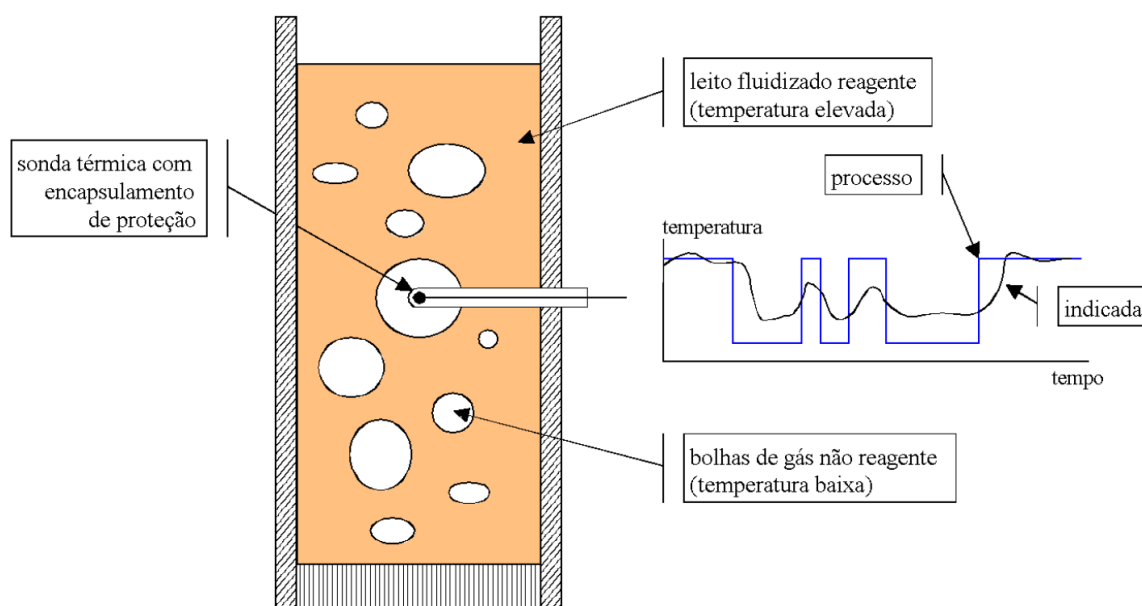


Figura 1: Delineamento da formulação do problema (Adaptado: Oliveira *et al.* (2006)).

A diferença entre a T_{proc} e T_{ind} está associada ao acúmulo térmico, devido ao revestimento do sensor, o que provoca distorções, e às razões de transferência de calor, devido à convecção e radiação, ocasionando atrasos na medida do sinal.

O acúmulo térmico é caracterizado pela massa M (kg) do encapsulamento e pelo calor específico C (J/kgK). As propriedades das transferências de calor convectiva e radiativa através da área A (m²) são calculadas, respectivamente, pelo coeficiente de convecção h (W/m²K), pela emissividade ε (adimensional) e pela constante de Stefan-Boltzmann σ a qual possui o valor 5.67×10^{-8} W/m²K⁴. A condução de calor através do cabo do sensor foi desconsiderada, devido à área do fio ser desprezível. Admitindo que o meio radiativo envolva completamente o sensor, a equação que relaciona T_{proc} e T_{ind} pode ser escrita como segue:

$$MC \frac{dT_{ind}}{dt} - hA(T_{proc} - T_{ind}) - \varepsilon \sigma A(T_{\infty}^4 - T_{ind}^4) = 0 \quad (1)$$

onde T_{∞} denota a temperatura na qual ocorre a transferência radiativa.

A equação (1) pode ser escrita em termos mais apropriados dividindo ambos os lados por hA e reorganizando as potências de T_{∞} e T_{ind} tem-se:

$$\tau \frac{dT_{ind}}{dt} - (T_{proc} - T_{ind}) - \gamma(T_{\infty} - T_{ind}) = 0 \quad (2)$$

onde

$$\tau = \frac{MC}{hA} \quad (3)$$

$$\gamma \cong \frac{4\varepsilon\sigma}{h} \left(\frac{T_{\infty} + T_{ind}}{2} \right)^3 \quad (4)$$

Na equação (3), τ representa a constante de tempo da sonda, isto é, o aumento de temperatura causado pelo acúmulo de calor sobre o calor transferido por convecção. O coeficiente de radiação γ , da equação (4), avalia a intensidade de transferência de calor radiativo em comparação com a transferência de calor convectivo.

Assim a equação (2) expressa a relação entre a T_{proc} (estímulo) e a T_{ind} (resposta). O Problema Direto (PD) pode ser resolvido de maneira imediata, calculando a saída (T_{ind}) da conhecida entrada (T_{proc}). Entretanto, resolver o PI correspondente, ou seja, determinar a entrada (T_{proc}) a partir da saída (T_{ind}) é certamente uma tarefa difícil por causa de sua natureza intrínseca mal condicionada.

De acordo com Oliveira et al. (2006), a equação (2) pode ser discretizada no tempo através do método de diferenças finitas. Definindo um passo no tempo Δt e um esquema de discretização atrasado com índices n e $n-1$ indicando que a variável se refere nos tempo $t = n\Delta t$ e $t = (n - 1)\Delta t$, respectivamente. É possível obter:

$$\frac{\tau_n}{\Delta t}(T_{ind,n} - T_{ind,n-1}) - (T_{proc,n} - T_{ind,n}) - \gamma_n(T_\infty - T_{ind,n}) = 0 \quad (5)$$

Assim os PD's e PI's são expressos, respectivamente, como:

$$T_{ind,n} = \frac{1}{\frac{\tau_n}{\Delta t} + 1 + \gamma_n} \left(T_{proc,n} + \gamma_n T_\infty + \frac{\tau_n}{\Delta t} T_{ind,n-1} \right) \quad (6)$$

$$T_{proc,n} = T_{rec,n} = \frac{\tau_n}{\Delta t} (T_{ind,n} - T_{ind,n-1}) + T_{ind,n} - \gamma_n (T_\infty - T_{ind,n}) \quad (7)$$

A T_{proc} obtida pela equação (7) deveria ser exata, mas devido ao mau condicionamento dos PI's e aos pequenos erros de medida, a derivada da equação (2) se comporta como um filtro passa-alta, atenuando os ruídos de baixa frequência e ampliando os de alta frequência. Desta forma a T_{proc} obtida pela equação (7) será denominada de temperatura reconstruída, T_{rec} .

3.2 Banco de Dados dos sinais e análise de parâmetros

Os sinais foram simulados e processados dentro da plataforma LabView, no Laboratório de Matemática Computacional (LabMaC) da UNESP, campus de Assis.

A geração dos sinais simulados é programada a fim de representar o comportamento e correlações dos parâmetros da equação (7), ou seja, simular o aferimento de temperatura de um processo dentro dos problemas propostos com determinada variabilidade, para conservar o poder preditivo dos métodos e conferir ao mesmo a capacidade de generalização.

Ao todo foram simulados 25000 sinais de temperatura de cinco tipos diferentes (Quadrado, Seno, Triangular, Serrilhado e Padrão), cada sinal é constituído por 1024 pontos e 9 colunas, dentre eles estão a temperatura real do processo (variável alvo), T_{proc} , e 8 parâmetros/features (derivados da equação (7) do problema), figura 2.

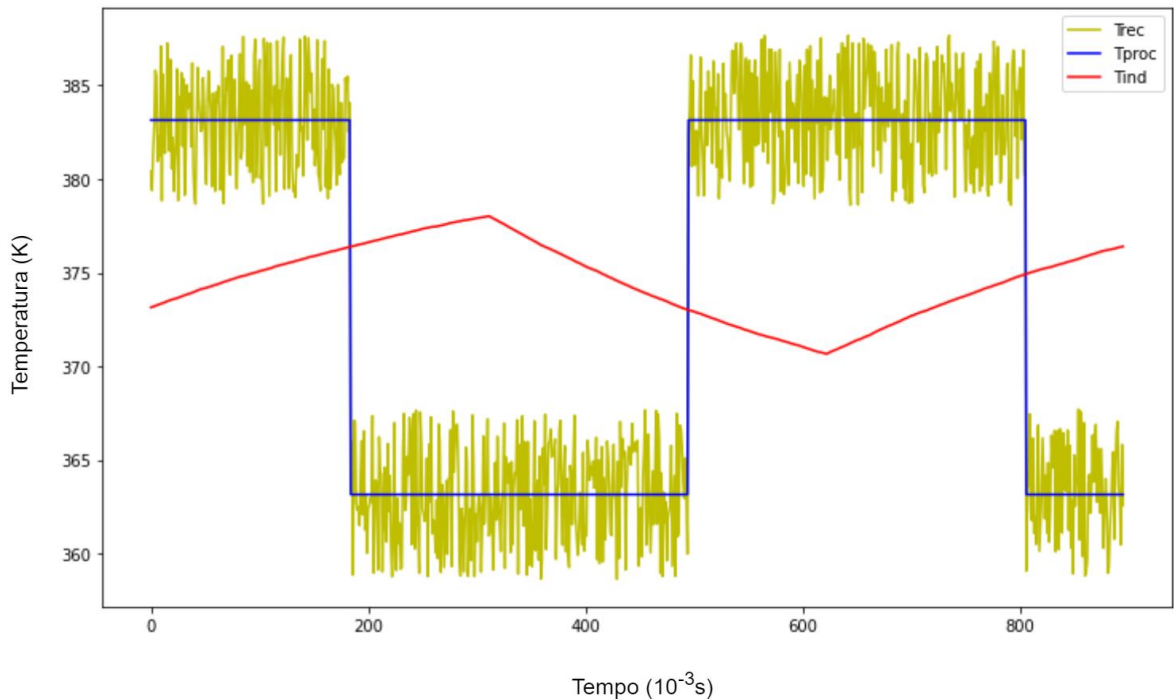


Figura 2: Exemplo de sinal simulado quadrado (Fonte: próprio autor).

Dentre os 25000 sinais constituintes do banco, além do tipo, os mesmos variam no coeficiente de convecção h , por consequência em τ , equação (3), já que estão intrinsecamente correlacionados, assim como em amplitude e frequência.

Ambas as amplitudes e frequências variam de 10 a 100, a um passo de 10, a T_{ind} é obtida através da equação do problema direto (6) cujo ruído foi acrescentado sobre proporção de 0,01. A T_{rec} é obtida através da equação (7) do problema inverso. O coeficiente de convecção é um parâmetro que de certa forma pode ser considerado um problema inverso dentro do problema inverso e varia de 50 W/m²K a 25000 W/m²K, a um passo de 50, a fim de contemplarmos uma gama variada de possíveis meios para o modelo.

Conhecer como os parâmetros estão correlacionados, e como os mesmos afetam tanto o atraso da T_{ind} quanto o desempenho da T_{rec} é fundamental na tomada de decisão sobre as estratégias e modelos a serem testados. Portanto, uma análise exploratória prévia dos dados é uma importante fonte de informações.

A correlação mais influente entre as variáveis se dá em h , uma vez que, como já descrito na formulação do problema, este interfere na transferência de calor por convecção e radiação, influenciando diretamente no acúmulo térmico e, portanto, no atraso do sinal. Massa M e o calor específico C foram fixados, pois são propriedades físicas da sonda de captação do sinal, no caso um termopar tipo K.

Observa-se na figura 3 que a relação entre o h e o desempenho da T_{rec} é inversamente proporcional, quanto maior o coeficiente de convecção, menor é o atraso do

signal, portanto os ruídos exacerbados pela natureza mal posta do problema inverso serão menores. Coeficientes de convecção maiores significam trocas térmicas mais eficientes e acúmulos térmicos menores figura 4.

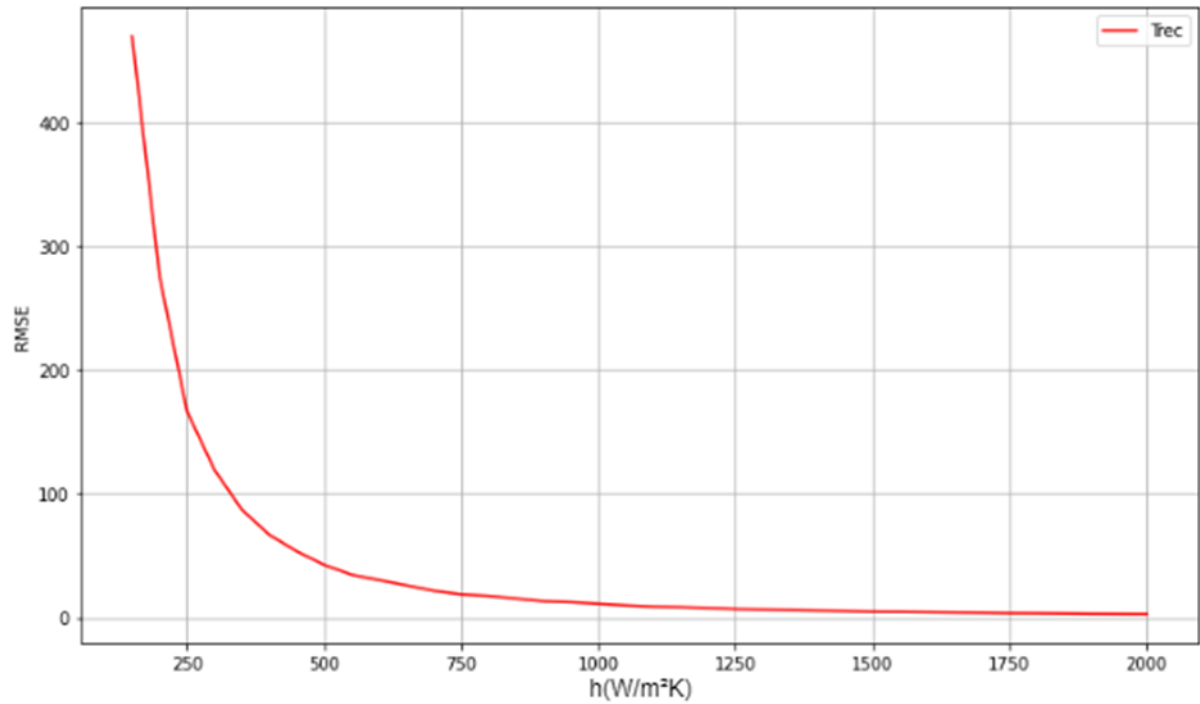


Figura 3: Relação entre h e o RMSE da T_{rec} (Fonte: próprio autor).

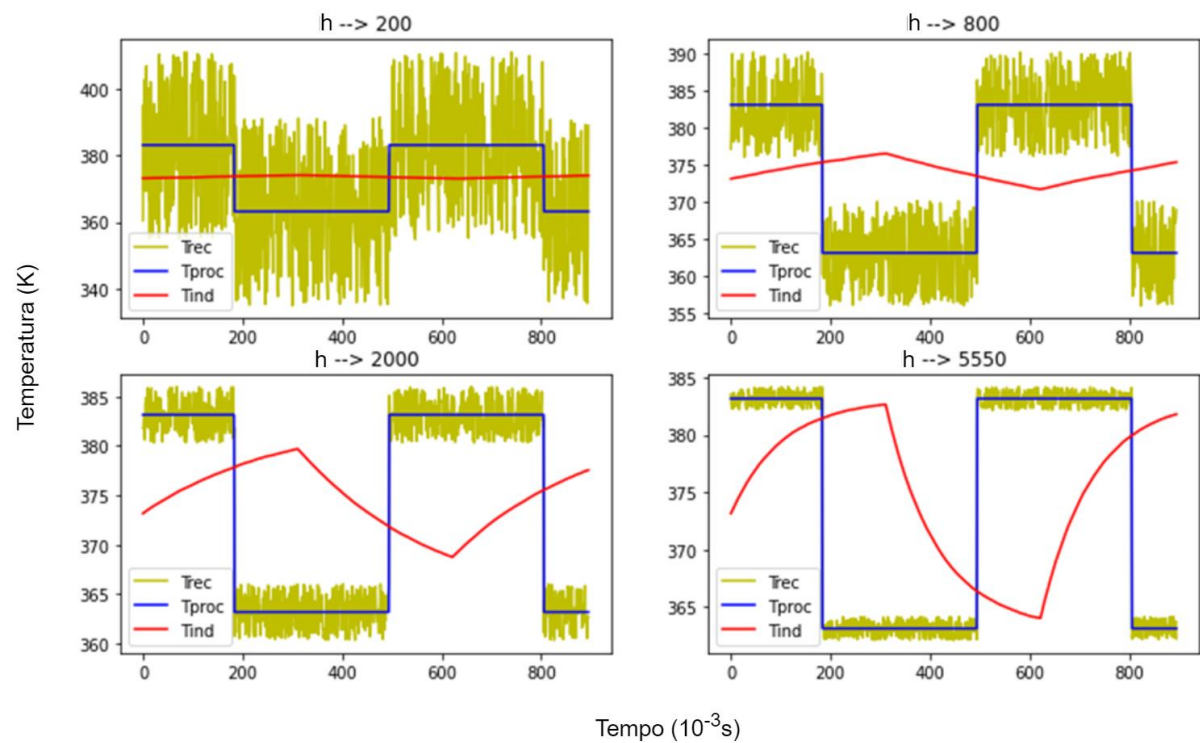


Figura 4: Efeitos do coeficiente de calor na T_{ind} e T_{rec} (Fonte: próprio autor).

O modo como o tratamento posterior do banco de dados de treinamento ocorre depende do método a ser utilizado, diferentes estratégias necessitam que os dados de entrada sejam estruturados de maneiras diferentes. Em seguida será discutido as diferentes estratégias e algoritmos utilizados na construção das soluções e modelos.

3.3 Métodos e *Pipelines*

O modo como as RNA's são capazes de aprender e solucionar problemas se dá pela sua capacidade de “absorver” a relação entre parâmetros do fenômeno a ser modelado. Como todas as relações termodinâmicas do problema proposto já estão definidas e modeladas, a possibilidade de um banco de dados simulado se torna a opção mais viável, portanto, justifica-se a escolha dos algoritmos de aprendizados supervisionados, já que existe a disponibilidade do desenvolvimento de uma grande quantidade de exemplos entrada/resposta. Na etapa de treinamento do aprendizado supervisionado, isso ocorre através do fornecimento abundante de exemplos (entrada/*input*) e seus respectivos resultados esperados (saída/*output*), a fim de se detectar padrões subjacentes entre os parâmetros propostos. O algoritmo então tenta extrapolar essas relações aprendidas na apresentação dos exemplos para dados nunca antes vistos pela rede, isso ocorre durante a etapa de testes, a qual tem o papel de aferir a capacidade de generalização da rede, característica fundamental uma vez que garante a eficácia da sua aplicação em problemas reais subsequentes (DA SILVA *et al.*, 2010).

As RNA's são conhecidas por proporcionarem soluções robustas e tolerantes a variações nas condições iniciais do problema, isso se deve ao fato da sua arquitetura ser constituída por vários nódulos/neurônios altamente interconectados. Cada nódulo conecta-se a outro e possui um peso e um limite associados, caso o seu cálculo individual ultrapassar esse limite, o neurônio é então ativado e a informação passa para a próxima camada da rede (HAYKIN, 1999).

Quando se refere ao número de camadas da rede, é convenção excluir a camada de entrada e saída, uma vez que estas compõem os dados externos e não contém parâmetros a serem otimizados, conta-se então, apenas as intermediárias, também apelidadas de camadas ocultas. A rede recebe os dados de entrada na primeira camada, a informação vai sendo processada pelos nódulos em cada camada, até ser finalmente expelida em forma de resposta (DA SILVA *et al.*, 2010). Na figura 5 pode-se observar uma rede *feedforward* de 2 camadas ocultas. Pode-se pensar em cada neurônio como um pequeno modelo individual de regressão linear cuja fórmula está representada Figura 6.

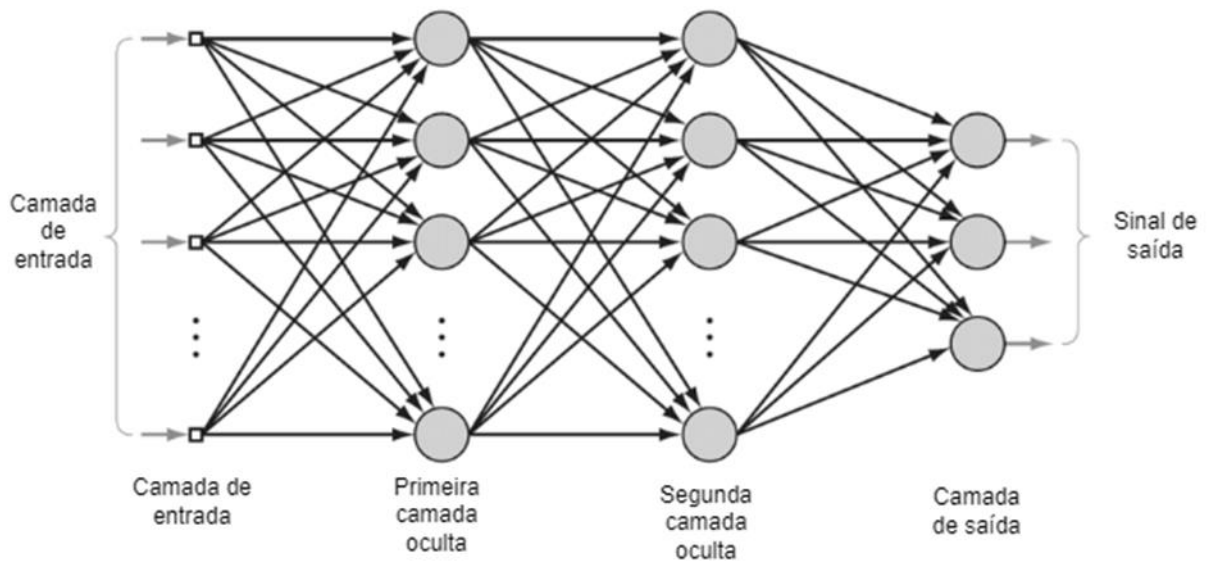


Figura 5: Exemplo de RNA com 2 camadas ocultas (Adaptado: HAYKIN, 1999).

Na figura 6, os valores $(x_1, x_2, x_3, \dots, x_n)$ representam o vetor de entrada da rede, ou seja, os dados os quais constituem nossos exemplos. O vetor $(w_1, w_2, w_3, \dots, w_n)$ constitui os pesos com os quais os dados de entrada são multiplicados. Esses pesos ajudam a aferir a importância de cada parâmetro de entrada atribuído à rede, pesos maiores significam influência maior na saída do neurônio e consequentemente na informação que é passada para as camadas adjacentes. Os produtos das multiplicações são então somados ao bias (b) e em seguida passam pela função de ativação, a qual determina a forma final da informação a ser transmitida adiante. Esse fluxo de dados unidirecional camada por camada é o que caracteriza a rede como uma rede *feedforward* (HAYKIN, 1999).

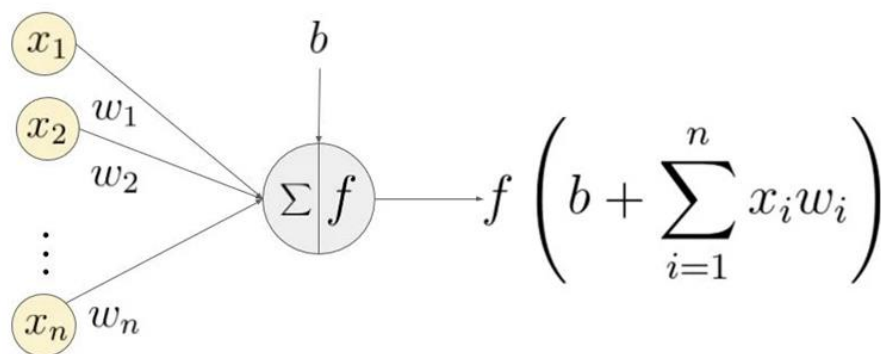


Figura 6: Cálculo de um neurônio artificial (SHARMA, 2017).

O cálculo do neurônio individual transforma os dados de entrada de forma linear (soma das multiplicações), a função de ativação tem o papel fundamental de transformar esse sinal não linearmente, é essa transformação a qual confere à rede a capacidade de

assimilar relações complexas não lineares entre as entradas e saídas. Ao longo do tempo diversas funções de ativação foram estudadas a fim de obtermos um treinamento robusto e de forma rápida. Alguns exemplos de funções populares na literatura são exemplificados na figura 7.

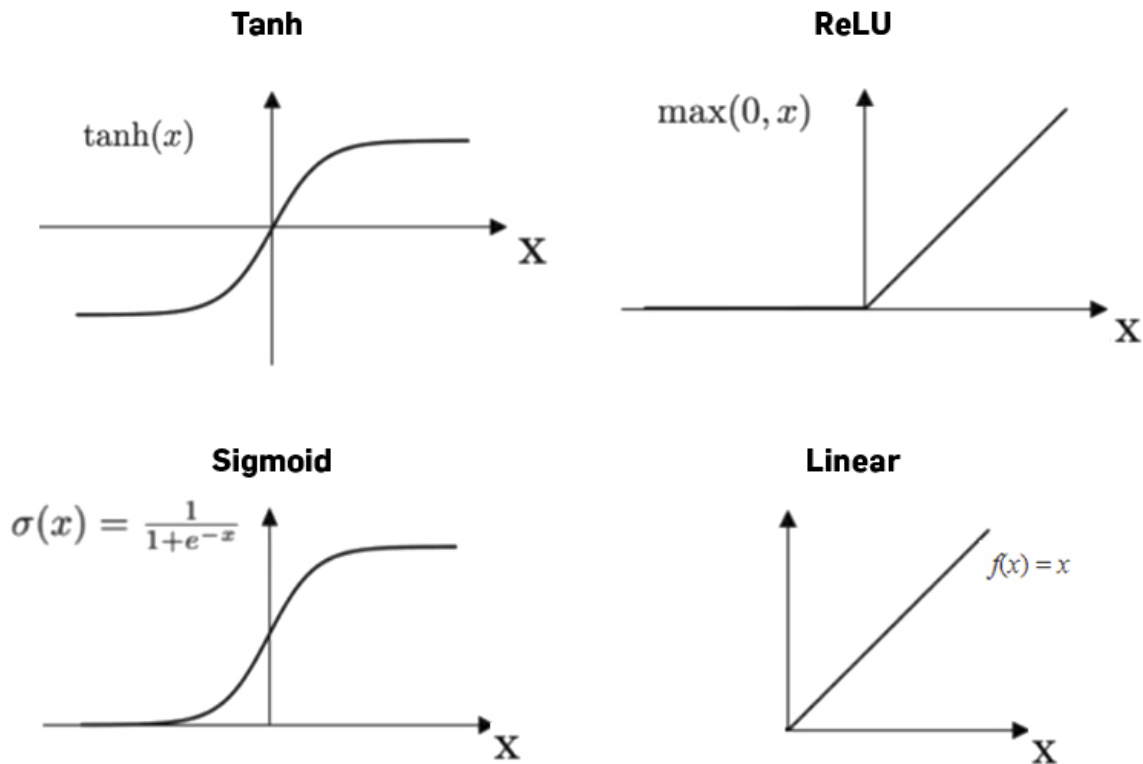


Figura 7: Exemplos de funções de ativação (ACTIVATION, 2022).

A escolha adequada da função de ativação é fundamental para o treinamento eficaz da rede, uma vez que elas definem a imagem da função da saída dos neurônios em cada camada, portanto é preciso levar em consideração a natureza dos dados utilizados, a fim de não restringir ou limitar o fluxo da informação no aprendizado.

Funções de ativação lineares, também chamadas de funções identidade, são representadas na forma $f(x) = x$. Basicamente a entrada é passada adiante sem transformação, em um fluxo desimpedido (IBM CLOUD EDUCATION, 2020). Esse tipo de função é muito usada na última camada em problemas de regressão, onde a rede se propõe a prever um valor numérico dentro de uma escala contínua, como por exemplo, temperatura. Funções de ativação não lineares por sua vez, tem a capacidade de distinguir dados os quais não são separáveis linearmente, e em sua maioria, controlam e restringem a imagem da saída da rede, por esse motivo, são muito empregadas em problemas de classificação, onde a saída da rede tem caráter probabilístico e é interpretada categoricamente (SHARMA, 2017).

3.3.1. Aplicação de uma RNA para o problema inverso

A forma como uma RNA aprende é através do cálculo da função erro. Seguindo o fluxo da informação das camadas, a rede produz uma saída, aqui chamada de temperatura da rede, T_{rede} a cada iteração n , obtém-se então a função do erro instantâneo (referente àquela iteração n específica):

$$e_j(n) = T_{proc,j}(n) - T_{rede,j}(n) \quad (8)$$

O erro particular da iteração n é representado pela diferença entre a resposta da rede, T_{rede} , e o valor desejado, T_{proc} . A partir dessa relação pode-se descrever a função do erro como um todo, também chamada de função “loss” ou função de custo:

$$E = \frac{1}{2} \sum_{j=1}^m e_j^2 \quad (9)$$

Essa função é dada pelo cálculo a partir da soma dos quadrados dos erros instantâneos. A aprendizagem ocorre, pois a função tem como parâmetros os próprios pesos dos neurônios da rede. Portanto, a rede é capaz de obter uma solução através do ajuste dos pesos e bias em seus nódulos na direção de minimizar a função custo, esses ajustes são proporcionais às derivadas da mesma (DA SILVA *et al.*, 2010). O processo iterativo de determinar os mínimos locais por meio das derivadas parciais da função custo é chamado de gradiente descendente (DA SILVA *et al.*, 2010).

Como descrito por Borges e Menezes em seu trabalho de aplicação de RNA's do tipo Hopfield para a solução de PI's em cinética química, a solução para o problema inverso pode ser obtida através da escolha de uma função custo apropriada (BORGES; MENEZES, 2017).

Nota-se que T_{rede} é a própria saída da RNA, um produto dos cálculos sinápticos do processo de treinamento, o qual é intrinsecamente dependente de T_{ind} (entrada da rede).

A RNA, como descrita, tem, portanto, a capacidade de minimizar a função custo até que o critério de parada seja alcançado, este pode ser totalmente arbitrário e manual ou pré-estabelecido. Dessa forma, mais de uma solução para o problema podem ser obtidas, fazendo-se necessário a análise cuidadosa dos resultados e a escolha da melhor solução produzida pela rede. A solução indicada pode ser descrita como uma solução regularizada para o problema inverso (BORGES; MENEZES, 2017).

Como se pode observar, a estrutura de uma RNA pode variar bastante na quantidade de camadas, na quantidade de neurônios em cada uma delas e no tipo de função de ativação utilizada em cada saída. O formato do vetor de entrada, ou seja, a

maneira como tratar e alimentar a informação da primeira camada da rede, também deve ser aferido cuidadosamente. O pré-processamento dos dados do vetor de entrada está intrinsecamente ligado à forma como o problema é considerado, e tem influência muito significativa sobre a maneira com que a RNA se propõe a operar (CRONE; KOURENTZES, 2007). Em seguida se discute maneiras com que o tratamento prévio dos dados foi efetuado e suas implicações, para cada método empregado.

3.3.2. Regressão Ponto a Ponto

O algoritmo da RNA funciona através do fornecimento de um banco de dados contendo “exemplos” e suas respectivas “respostas”, o algoritmo é capaz de mapear as relações não lineares entre as variáveis de entrada e saída, funcionando como um aproximador de funções. Isso é possível através do cálculo do erro entre as previsões e resultados esperados, o papel da rede é minimizar a função do erro através da modificação dos pesos nas conexões de seus neurônios a cada iteração.

A maneira mais prática de abordar o problema inverso proposto é considerá-lo um problema de regressão, onde se admite cada passo do sinal como um exemplo ou ponto dos dados, e configura-se a última camada (*output*) da rede para gerar valores contínuos de temperatura dentro de uma escala. Neste método, a função a ser aproximada, portanto, é a própria equação (7), onde n representa cada iteração do treinamento e possui 1024 temperaturas vezes a quantidade de sinais.

Neste método T_{proc} é considerada a variável alvo e resposta do exemplo. Os parâmetros a serem entregues na entrada da rede são selecionados dentre os residentes do lado direito da equação (7) ($T_{ind,n}$, $T_{ind,n-1}$, T_{∞} , γ_n e h). Os parâmetros específicos referentes ao equipamento, como M (massa), C (Calor específico) e A (Área) descritos na equação (3) não foram entregues à rede, já que são propriedades físicas fixas da sonda de captação do sinal, figura 8.

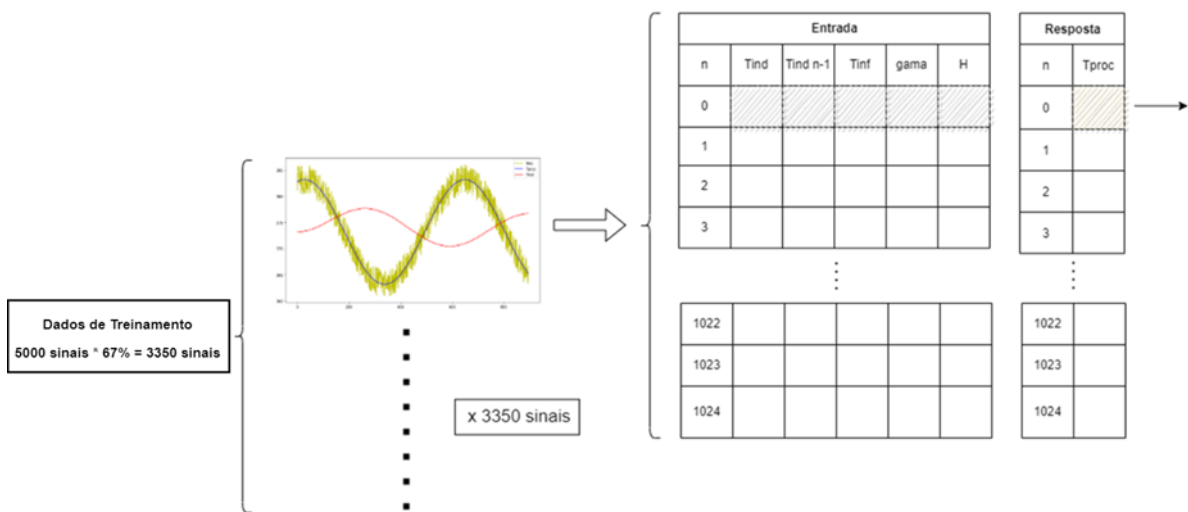


Figura 8: Diagrama do tratamento dos dados de treinamento (Fonte: próprio autor).

Uma rede foi treinada para cada tipo de sinal, dentre esses, 67% dos sinais foram destinados à etapa de treinamento e 33% à etapa de testes. Nesse método, cada passo de tempo da sonda é transformado em um *data point* a ser entregue à rede. A matriz dos vetores dos dados de entrada, portanto, tem o formato 5 x (5000 sinais x 67% x 1024 pontos). Seguindo a mesma lógica, o vetor dos dados alvo tem o formato 1 x (5000 sinais x 67% x 1024 pontos). Dessa forma a rede possui 5 nódulos na camada de entrada e 1 nódulo na última camada, figura 9.

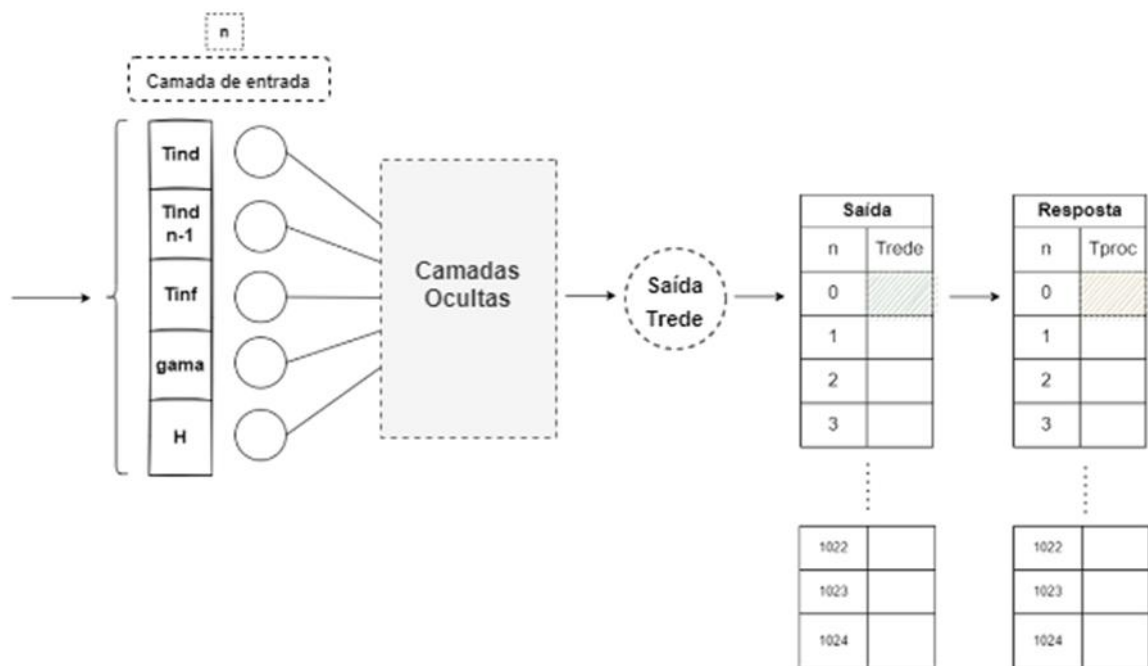


Figura 9: Diagrama da entrada de dados ponto a ponto (Fonte: próprio autor).

3.3.3. Regressão Multivariável

A regressão multivariável apresenta outra possível maneira de estruturar o vetor de entrada, atribuindo a rede 1024 nódulos na primeira camada, de maneira a contemplar todos os pontos do sinal. Dessa forma, considerou-se o sinal completo e seu respectivo acúmulo térmico h como um *data point*, figura 10. Com o sinal inteiro, atribuiu-se a RNA um papel de pós-processamento da curva, uma vez que é necessário que todos os pontos do sinal estejam disponíveis. Essa forma de tratamento do vetor de entrada confere maior robustez no poder regularizador da rede, já que a mesma tem acesso a mais medida de temperatura por iteração do treinamento.

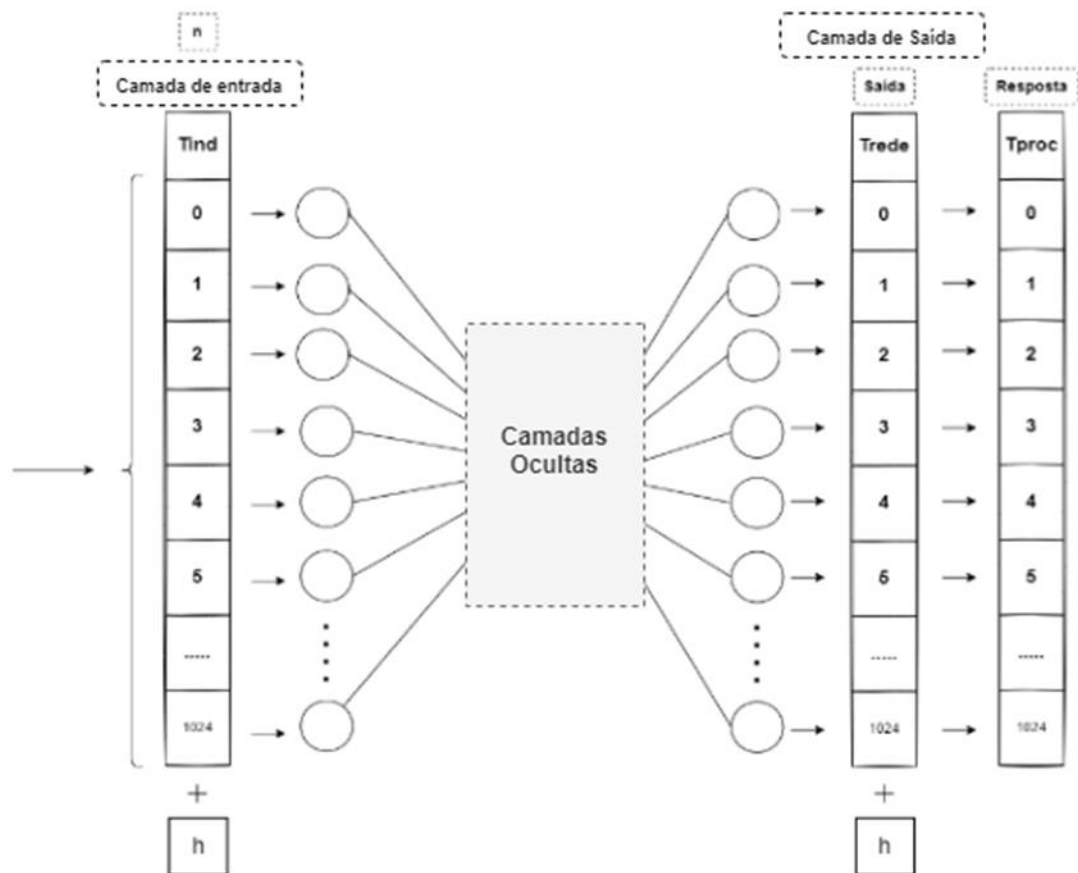


Figura 10: Diagrama da entrada de dados do sinal completo (Fonte: próprio autor).

3.3.4. Previsão de Série Temporal

Com a relação mal posta do problema, pode-se considerar o sinal como uma série temporal ao invés de uma coleção de pontos e observações individuais as quais compõem o mesmo. Esse ponto de vista permite aplicar uma série de conceitos relacionados à previsão de valores futuros baseados em um histórico de valores passados.

Na literatura encontram-se diversas aplicações para métodos de previsão de séries

temporais relacionados ao processamento de sinais. Os métodos tradicionais incluem filtro dos mínimos quadrados, filtro Wiener, entre outros filtros adaptativos. Recentemente uma série de estudos tem demonstrado que as RNA são capazes de oferecer modelos mais robustos e resultados mais satisfatórios para esses problemas. A ideia principal é aproveitar a capacidade de atenuação do ruído da rede como demonstrado por Sharma e Prakarsha (2022) em seu estudo com sinais ECG (Eletrocardiograma), ao passo que a mesma exerce sua capacidade preditiva sobre a curva do sinal.

O objetivo da rede neste método é prever a $T_{proc,n+1}$, baseado no histórico anterior de n registros de T_{ind} . A atenuação do ruído, o qual ocorre na recuperação do sinal inicial do problema inverso, se dá então, como subproduto da saída entregue pela rede. Os dados são apresentados à rede na forma de uma janela móvel de tamanho n assim como valor do acúmulo térmico h referente ao sinal, portanto o vetor de entrada tem o formato de $((1024 - n) \text{ janelas}) \times (5000 \text{ sinais} \times 67\%)$, como descrito nas figuras 11 e 12.

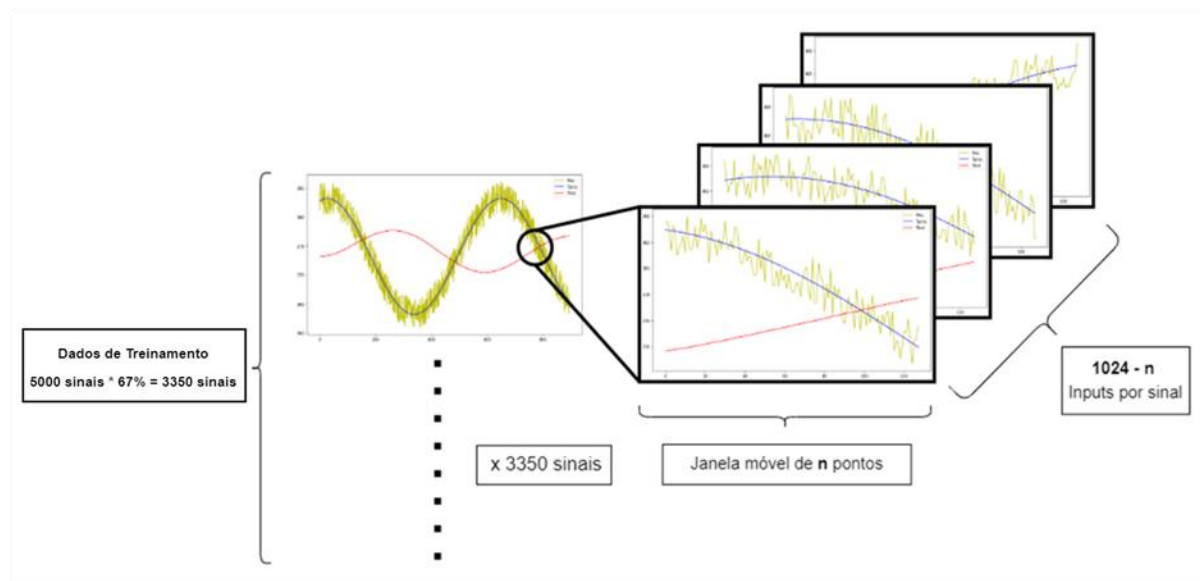


Figura 11: Diagrama da entrada de dados (Fonte: próprio autor).

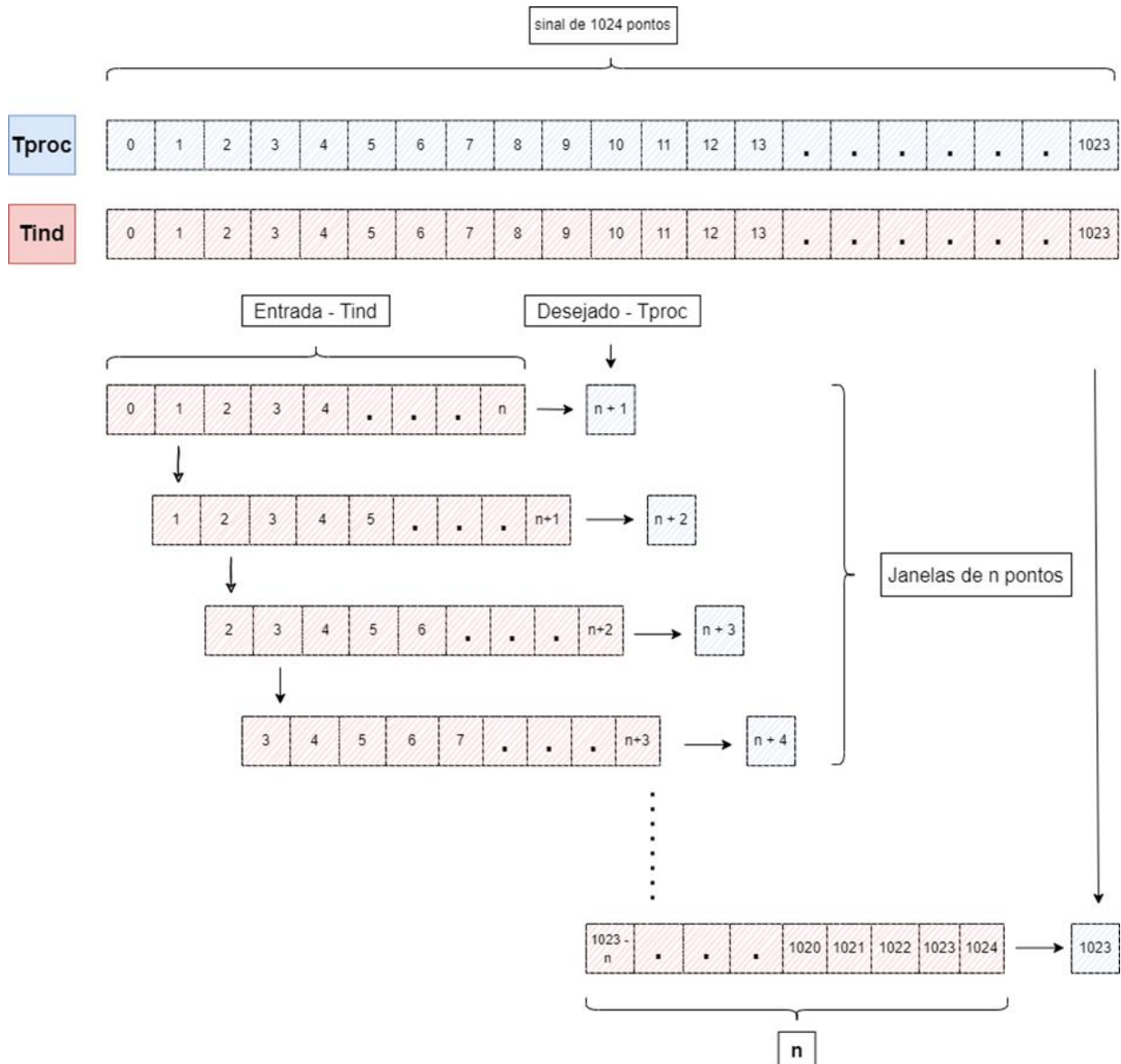


Figura 12: Diagrama da criação do vetor de entrada por janela móvel (Fonte: próprio autor).

3.3.5. Ferramentas e implementação em código

O tratamento dos dados e treinamento das redes foram feitos em Python (PYTHON SOFTWARE FOUNDATION, 2020). O pré-processamento dos bancos de treinamentos foi realizado através das bibliotecas Pandas (MCKINNEY, 2010), Numpy (GULLI, 2020) e Sklearn (PEDREGOSA, 2011). Os gráficos foram elaborados através da biblioteca de visualização de dados Matplotlib (HUNTER, 2007). O desenvolvimento e avaliação dos modelos foram gerados pela biblioteca e API de deep learning Keras (GERON, 2019), a qual se baseia e roda em cima da biblioteca de código Tensorflow (ABADI, 2016), especializada no desenvolvimento e treinamento otimizado de modelos de Aprendizado de Máquina.

4. RESULTADOS E DISCUSSÃO

4.1. Desempenho dos Modelos

A discussão e interpretação do desempenho de cada método/modelo é feita tendo como controle o desempenho da própria T_{rec} , ou seja, o produto mal posto recuperado do problema inverso, a qual a rede se propõe a regularizar. A tabela a seguir demonstra um ranking de desempenho dos modelos propostos, na forma da Raiz Quadrada do Erro Médio (RMSE) e o coeficiente de determinação R^2 , tabelas 1 e 2.

Tabela 1: Desempenho das redes.

Método	RMSE	R^2	RMSE recuperado	R^2 recuperado
Regressão	5,65215	0,98840	7,42493	0,96190
Ponto a Ponto				
Regressão	4,38939	0,98781	15,04529	0,92948
Multivariável				
Previsão de Série	5,37020	0,99587	9,66575	0,97145
Temporal				

Tabela 2: Desempenho das redes com $h < 5000$.

Método	RMSE	R^2	RMSE recuperado	R^2 recuperado
Regressão	26,13432	0,94656	35,06975	0,81988
Ponto a Ponto				
Regressão	15,11135	0,94631	72,84745	0,65560
Multivariável				
Previsão de Série	18,94954	0,98422	46,56405	0,86254
Temporal				

O desempenho de cada método é calculado através da média do RMSE e R^2 das redes para cada tipo de sinal utilizado. Foi calculada também, a média do desempenho para valores baixos de h , ou seja, sinais com expressivos atrasos na T_{ind} e ruídos exacerbados na T_{rec} , devidos aos acúmulos térmicos elevados como descrito na formulação do problema.

A tabela 2 apresenta os dados com a maioria dos sinais com valores de h acima de 5000 apresentarem boas regularizações pela própria temperatura reconstruída através do equacionamento inverso (T_{rec}). Portanto os resultados específicos para os valores baixos de

h destacam melhor a vantagem das soluções apresentadas pelas redes.

Diversas combinações de hiperparâmetros das redes são testadas, entre esses destacam-se o número de neurônios, camadas, taxa de aprendizado (indica a intensidade de atualização dos pesos a cada época) e tamanho da batelada (quantidade de exemplos a serem computados a cada época), as combinações as quais ofereceram resultados melhores e mais consistentes foram selecionadas.

O algoritmo de otimização utilizado para os modelos foi o Adam, já que atualmente é considerado a escolha mais robusta dentre os otimizadores adaptativos. Tal algoritmo é uma extensão do método de aprendizado gradiente descendente, e serve para acelerar o processo de treinamento ao manter uma taxa de aprendizado adaptativa e específica a cada neurônio. Dentre outros benefícios do algoritmo podemos citar a sua eficiência computacional e o pouco uso de memória, além de oferecer um bom desempenho em problemas com muitos parâmetros e grande quantidade de dados (KINGMA; BA, 2017).

Os resultados das redes estão expostos visualmente, em conjunto com os sinais da T_{rec} , T_{ind} e T_{proc} . Os resultados são separados em intervalos de h , com valores baixos, médios e altos, sendo $h < 500$, $500 < h < 5000$ e $h > 5000$, respectivamente.

4.1.1. Regressão ponto a ponto

Foi selecionada uma rede de 2 camadas ocultas, a primeira camada contendo 40 neurônios, a segunda 10 neurônios e a camada de saída retornando apenas uma medida de temperatura (um neurônio).

Os melhores resultados foram obtidos quando os dados de treinamento foram normalizados, entre 0 e 1. Esse método de escalonamento dos dados é muito útil para algoritmos de otimização como o de gradiente descendente, usado no processo de aprendizagem das redes. O escalonamento é particularmente útil também para ocasiões em que as ordens de grandeza entre os parâmetros de entrada são fundamentalmente diferentes, como é o caso do método de regressão ponto a ponto, figuras 13, 14 e 15.

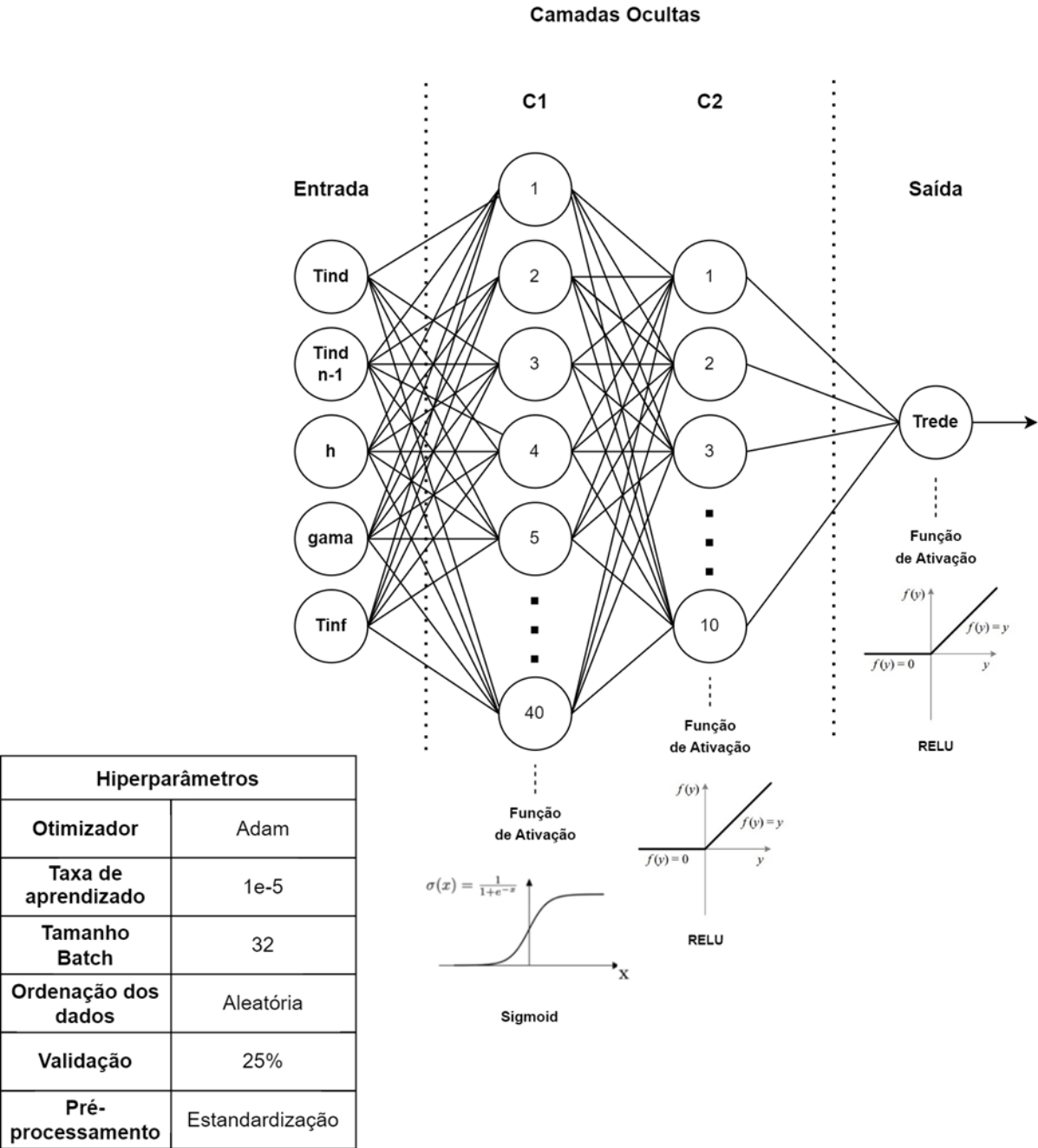


Figura 13: Hiperparâmetros do modelo de regressão ponto a ponto

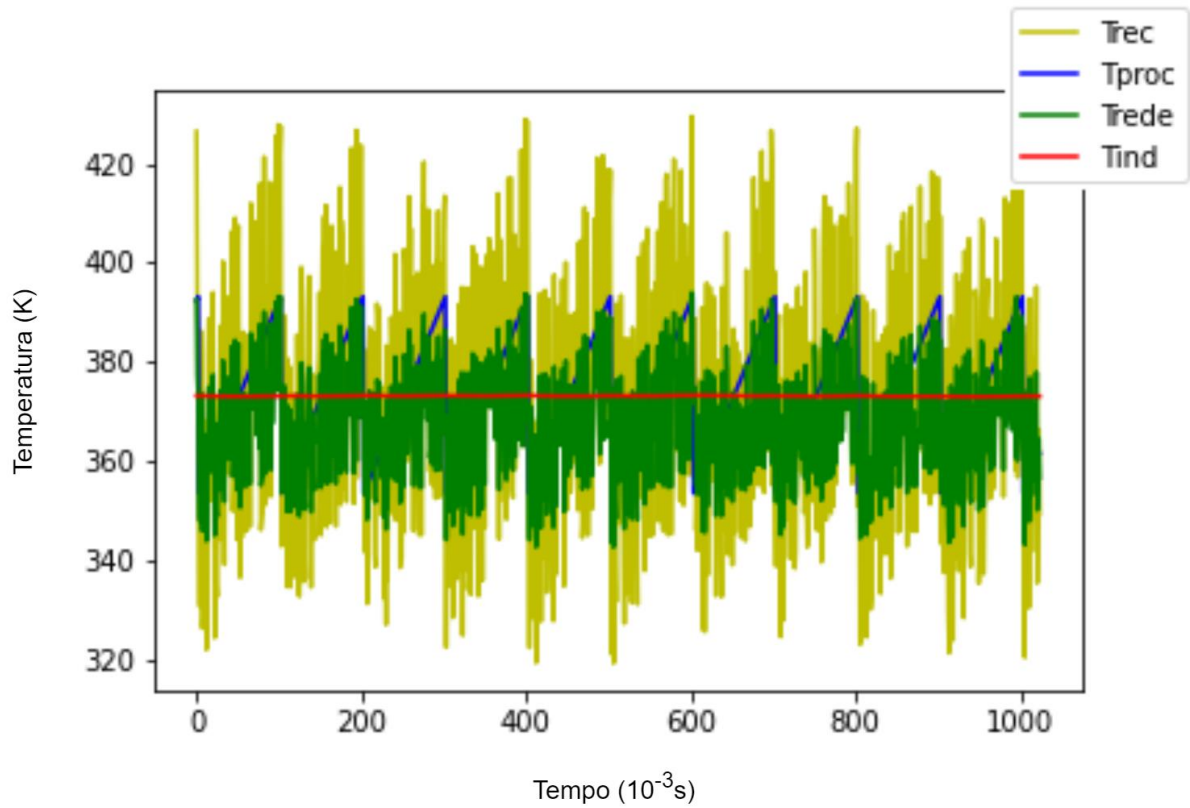


Figura 14: Exemplo de um sinal de h baixo e do tipo serrilhado, regularizado pela rede a partir do método de regressão ponto a ponto.

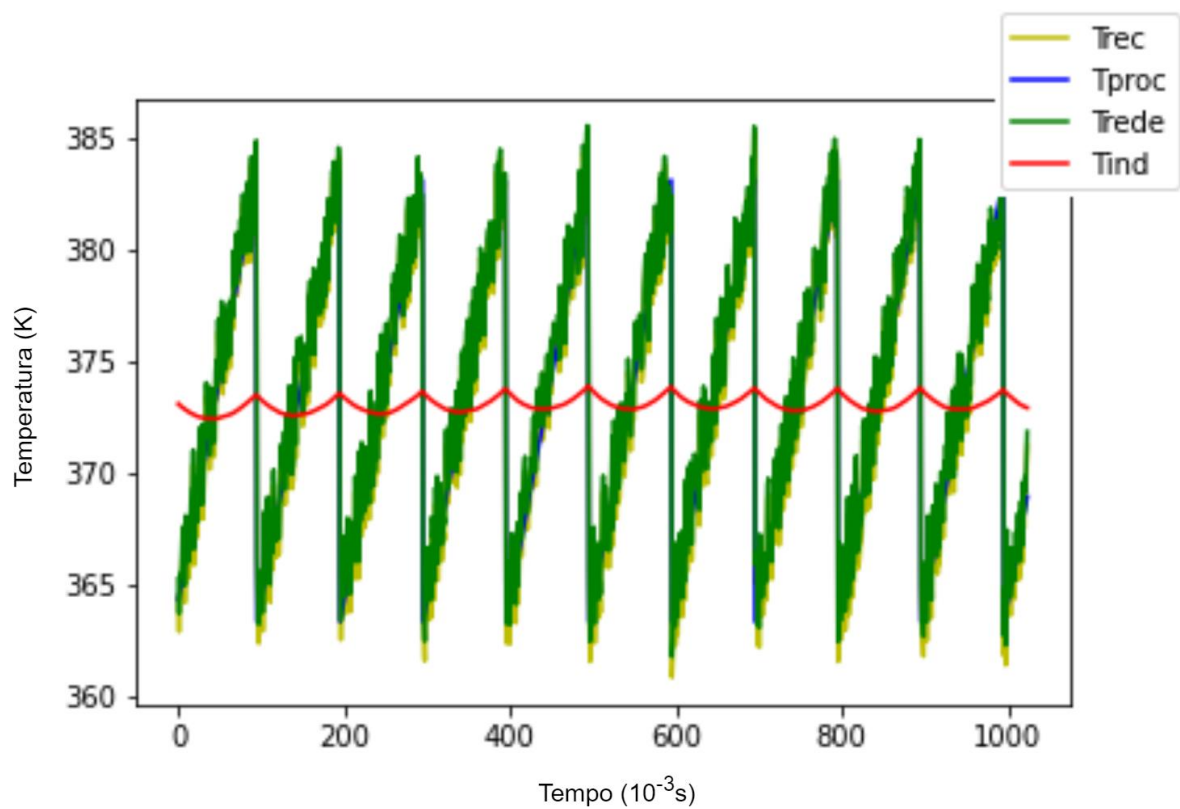


Figura 15: Exemplo de um sinal de h médio e do tipo serrilhado, regularizado pela rede a partir do método de regressão ponto a ponto.

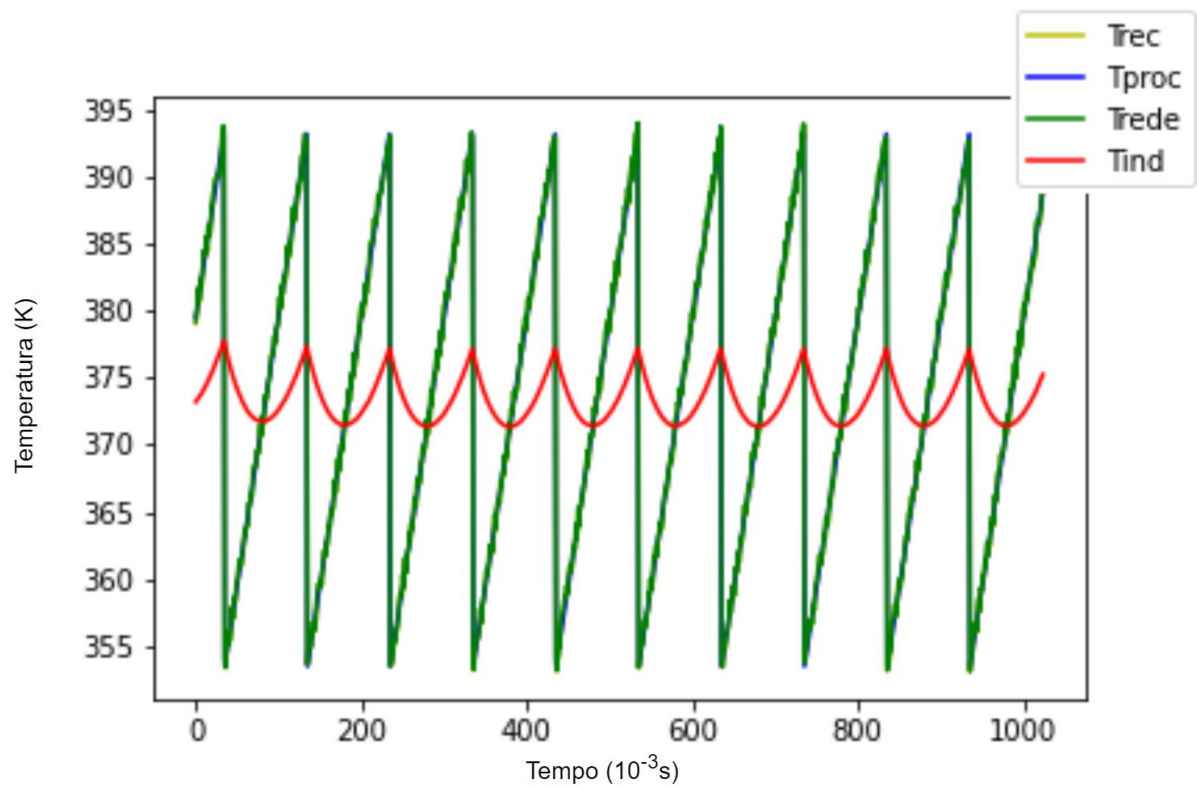


Figura 16: Exemplo de um sinal de h alto e do tipo Serrilhado, regularizado pela rede a partir do método de regressão ponto a ponto

O método de regressão ponto a ponto, como descrito nas tabelas 1 e 2 dos resultados, apesar de se apresentar como uma boa solução regularizadora quando comparada com a reconstrução T_{rec} , se demonstra inferior aos métodos de regressão multivariável e previsão de série temporal. O fato de o *data point* da entrada de dados da rede consistir apenas da medida de temperatura de um dos dois únicos passos de tempo da sonda, $T_{ind,n}$ e $T_{ind,n-1}$, a torna extremamente suscetível aos ruídos elevados do sinal. Observa-se esse fenômeno, analisando graficamente como a magnitude do ruído afeta os sinais entregues a rede ponto a ponto, principalmente em valores de h baixos figura 16.

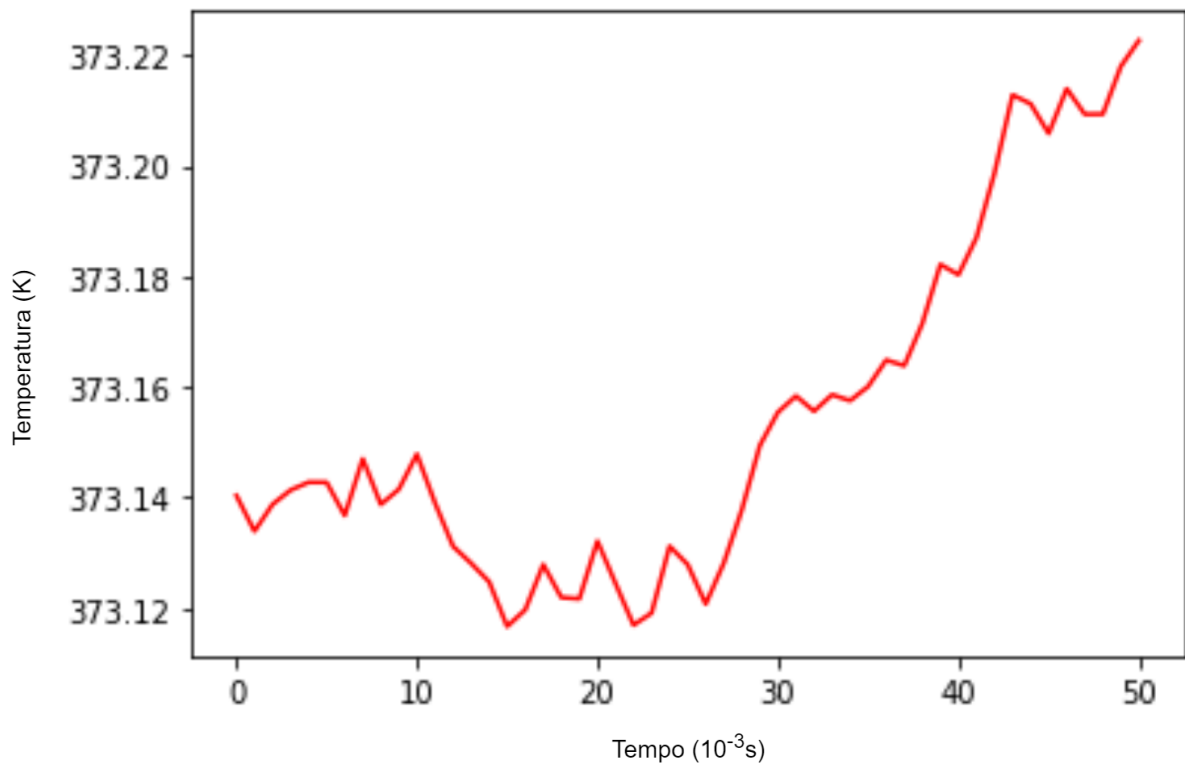


Figura 16: Intervalo de 50 medidas da T_{ind} de um sinal serrilhado de h baixo.

Observa-se na figura 16, o tamanho da deformação sofrida pela T_{ind} no nível ponto a ponto das medidas. Portanto é natural que o aprendizado e resposta da rede sejam afetados pela volatilidade causada pelo ruído natural da temperatura, uma vez que os valores das entradas da rede podem variar consideravelmente em um intervalo de apenas uma iteração dentro da época, atrasando a atualização dos pesos e dificultando o direcionamento correto do gradiente no processo de treinamento. A influência desse fenômeno na suavidade da resposta da rede ficará evidente na discussão do método de previsão de série temporal.

4.1.2. Regressão Multivariável

Foi selecionada uma rede de 2 camadas ocultas, a primeira camada contendo 1700 neurônios, a segunda 1360 neurônios e a camada de saída retornando um sinal inteiro, ou seja, 1024 neurônios, um para cada passo do sinal. É importante ressaltar que a escolha da quantidade de neurônios em cada camada depende intrinsecamente do formato dos dados de entrada e saída, neste caso, a rede possui 1024 neurônios de entrada e saída respectivamente, portanto são necessários neurônios suficientes para que a rede possa assimilar os padrões não lineares entre todos esses parâmetros, sem prejudicar a sua capacidade de generalização para novos sinais, figura 17. Os resultados do método são apresentados nas figuras 18, 19 e 20.

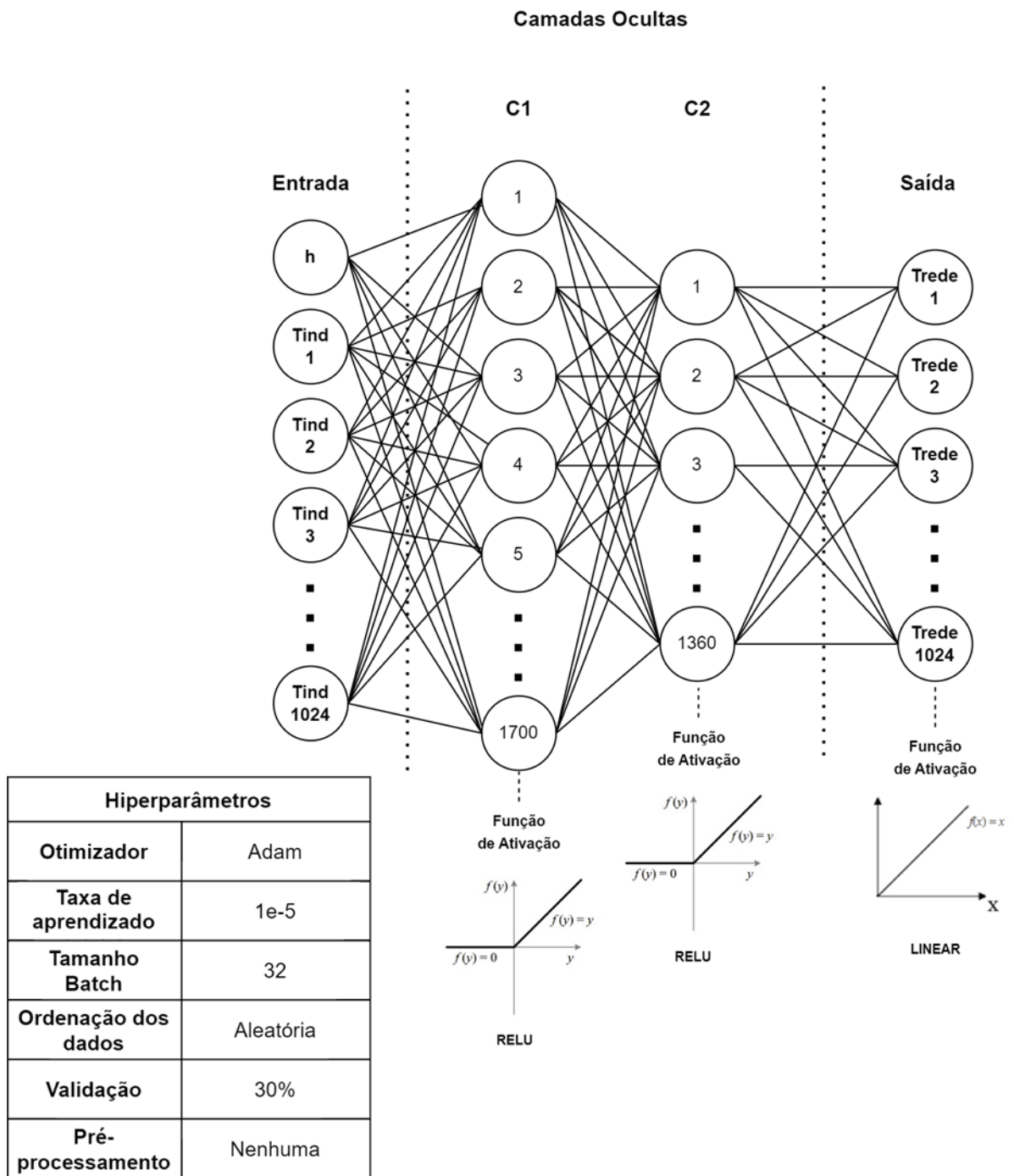


Figura 17: Hiperparâmetros do modelo multivariável

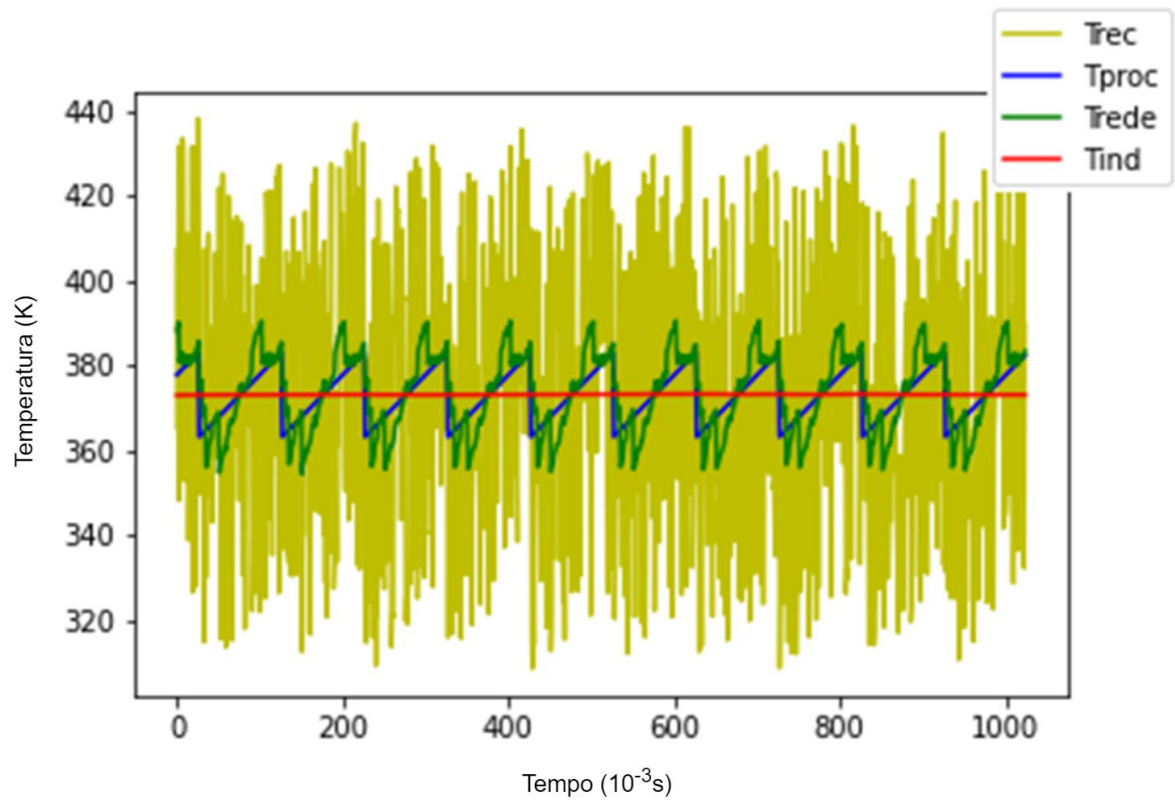


Figura 18: Exemplo de um sinal de h baixo e do tipo serrilhado, regularizado pela rede partir do método de regressão multivariável.

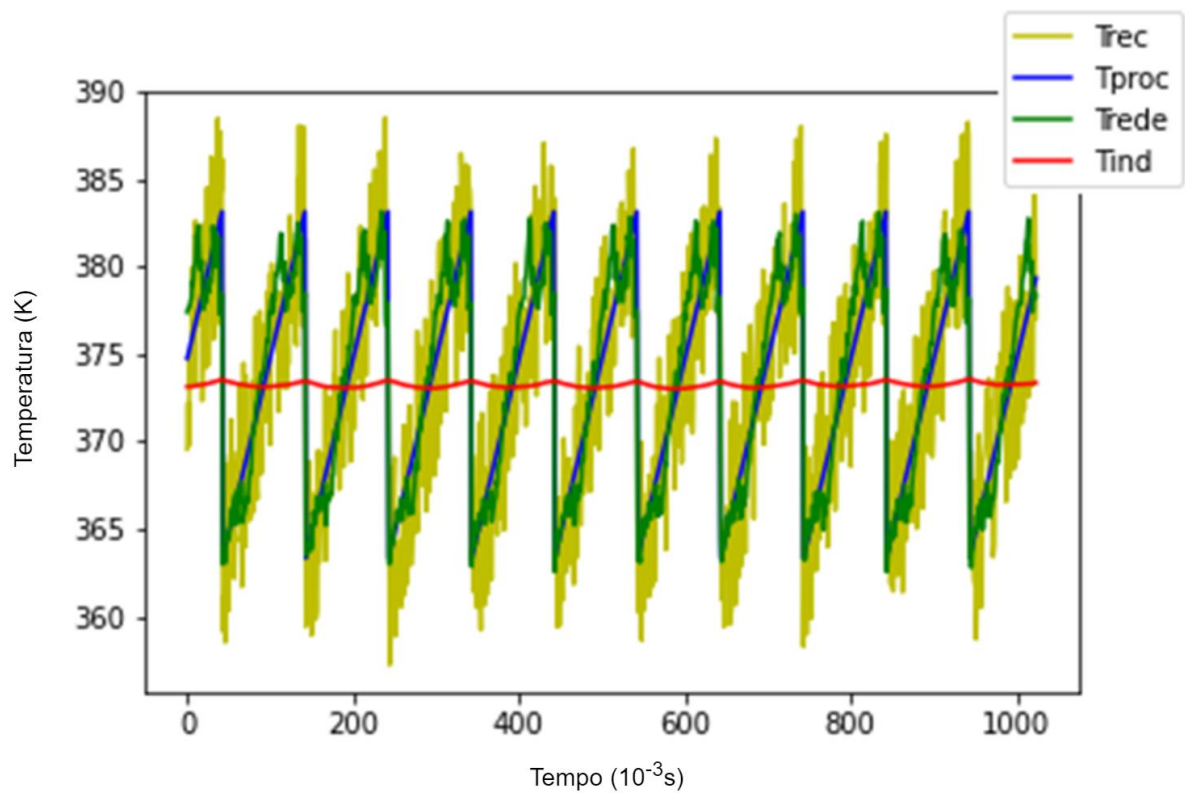


Figura 19: Exemplo de um sinal h médio e do tipo Serrilhado, regularizado pela rede a partir do método de regressão multivariável.

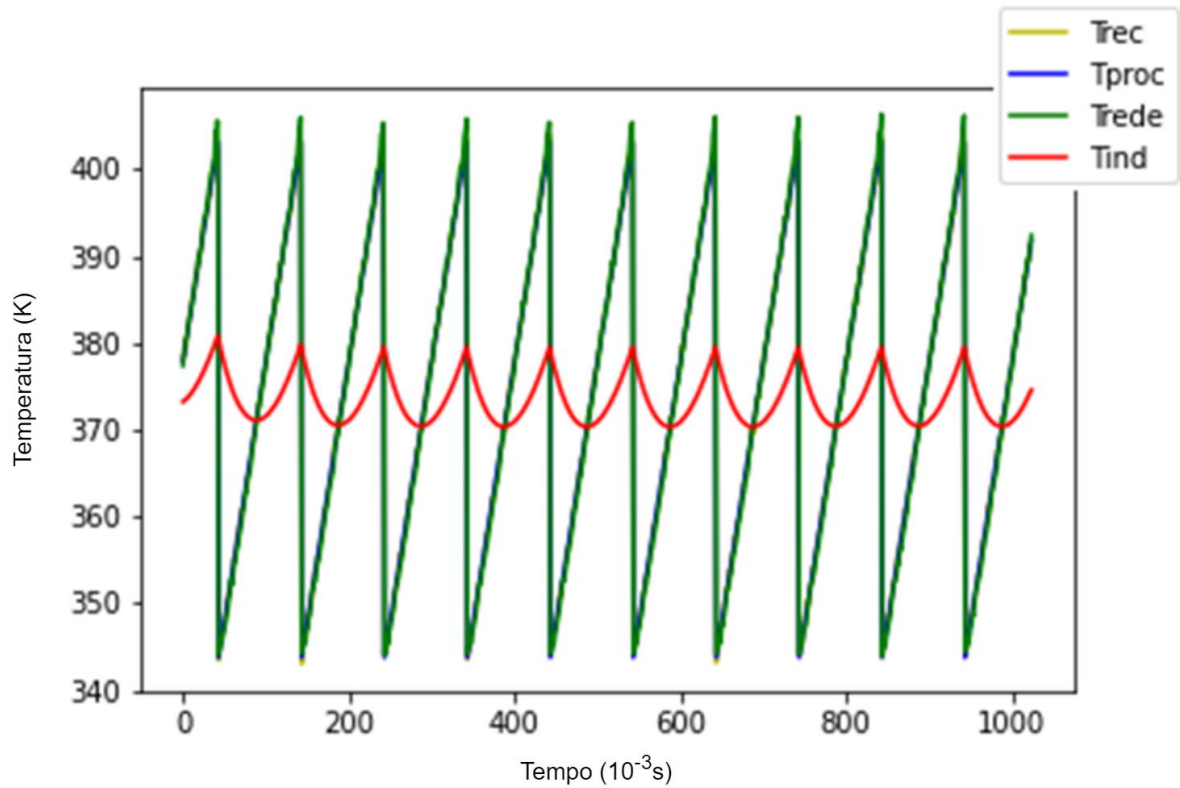


Figura 20: Exemplo de um sinal de h alto e do tipo serrilhado, regularizado pela rede a partir do método de regressão multivariável.

Nota-se que este método apresenta a melhor solução regularizada dentre os demais, pois o sinal se caracteriza como uma espécie de imagem fixa, um conjunto de coordenadas as quais compõem a curva total. Esta maneira de tratar os dados de entrada atribui à rede o máximo de informação possível dentro de um único *data point*, ou seja, a “memória” de todas as medidas de temperatura, passadas e futuras do sinal, conferindo à rede uma vantagem significativa no processo de aprendizado em relação aos demais métodos.

Esse tratamento, porém, proporciona desvantagens em outros aspectos da solução. Ao utilizar todos os 1024 pontos do sinal na entrada, inviabiliza-se a possível aplicação da solução em tempo real, uma vez que todos os pontos do sinal precisam estar disponíveis para alimentar a rede, por esse motivo, essa estratégia se caracteriza como uma técnica de regularização a partir do pós-processamento do sinal.

Outro ponto importante a ser considerado é o fato de que a solução passa a ser altamente dependente da morfologia do sinal, ou seja, durante o processo de aprendizado, partes dos padrões assimilados pela rede são pertencentes ao próprio formato do sinal em si (quadrado, seno, triangular, serrilhado e padrão), não somente relacionados às relações da equação (7), portanto torna-se imprescindível que os dados coletados para a solução do problema sejam representativos o suficiente do processo em que o problema inverso está instalado, para evitar que a rede crie “vícios” de sobreajuste (*overfitting*) e assimile padrões

indesejados no treinamento.

A figura 20 apresenta um exemplo de erro de sobreajuste da curva durante o processo de seleção e ajuste da rede para o método de regressão multivariável, observa-se que a T_{rede} está sob influência tendenciosa da amplitude dos sinais de treinamento, a rede neste caso, “decorou” características específicas a esses sinais e as extrapolou de maneira equivocada. Isso pode ocorrer com diversas características da curva como tipo, amplitude e frequência. Assim, o método de regressão multivariável está particularmente mais suscetível a esse fenômeno.

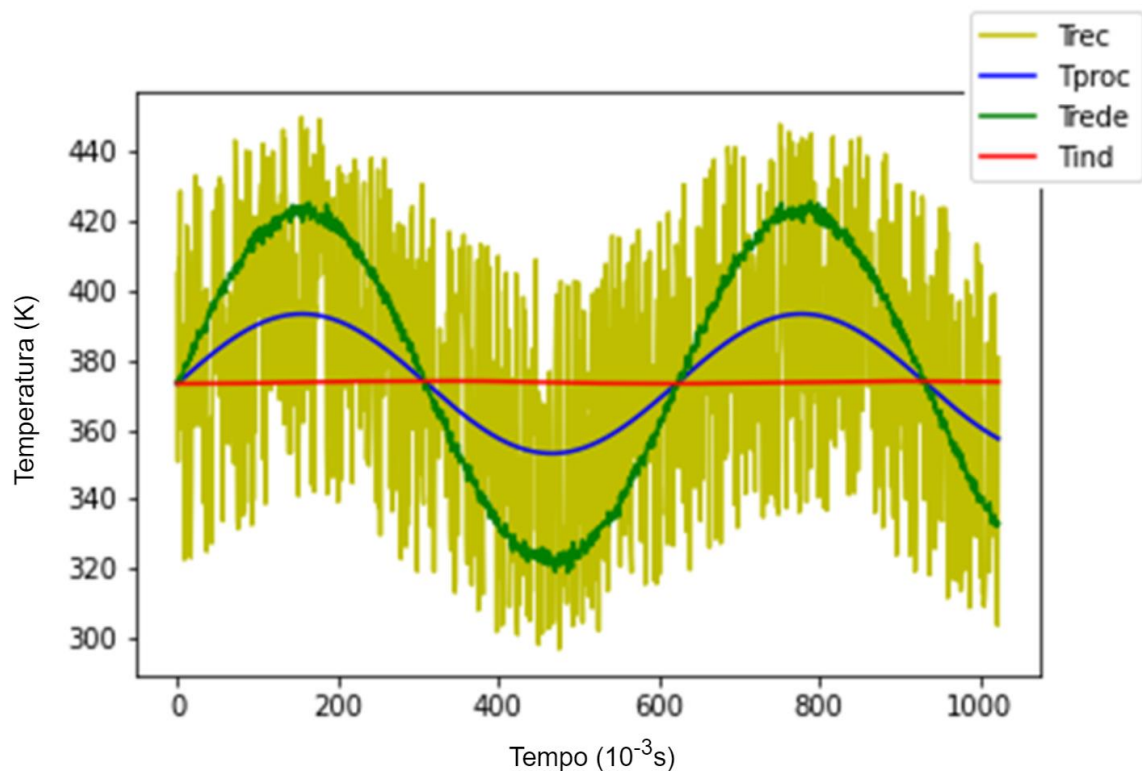


Figura 21: Exemplo de *overfit* em sinal do tipo seno no método de regressão multivariável.

4.1.3. Previsão da Série Temporal

Para o método de previsão da série temporal foi selecionada uma rede de 2 camadas ocultas, a primeira camada contendo 64 neurônios, a segunda 16 neurônios. A saída da rede se dá na forma da previsão do valor da temperatura no passo $n+1$, sendo n o tamanho da janela móvel utilizada. Dessa forma a camada de saída da rede terá apenas um neurônio, figura 22.

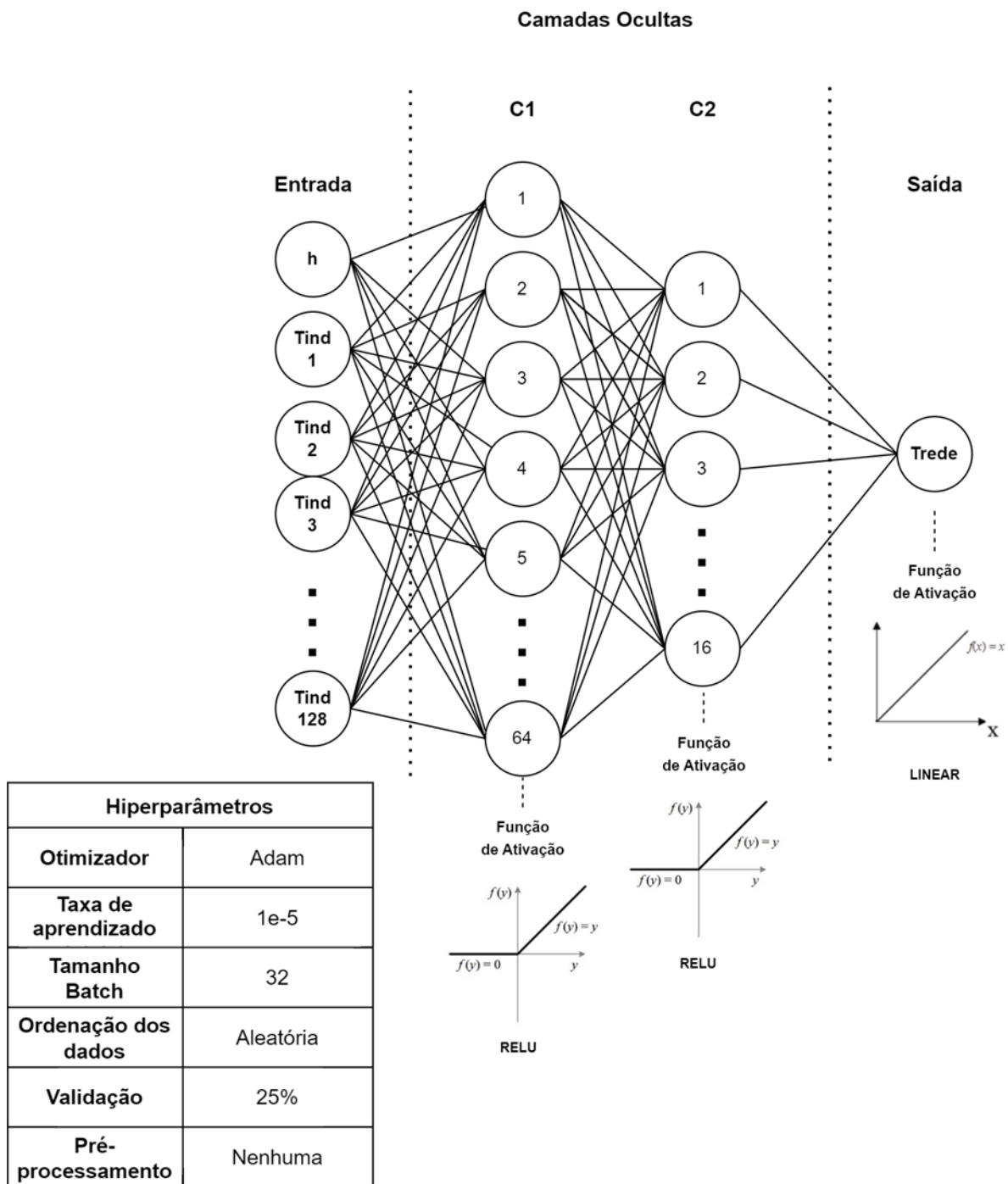


Figura 22: Hiperparâmetros do método de previsão de série temporal.

Esse método soluciona o problema de regularização através da previsão de uma série temporal. Portanto os dados de entrada passam a ter características específicas a esse formato, como tendência, sazonalidade, periodicidade, estacionaridade e ruído. A extração e entendimento desses componentes é a base para o tratamento dos parâmetros de entrada (FERNANDEZ, 2020). No caso do método de previsão da série temporal, a rede possui “memória” dos n pontos anteriores aquele que se quer determinar, assim como a intensidade do acúmulo térmico (valor de h), portanto podemos aferir que não deve ser

grande o suficiente para que o *data point* dentro dos dados de treinamento seja capaz de transmitir as informações essenciais da série temporal para a solução do problema, não obstante, sem sujeitar a rede as mesmas limitações do método de regressão multivariável.

O tamanho de n o qual proporcionou os melhores resultados foi de 128 pontos, a análise dos dados demonstrou que tal valor é suficiente, mesmo sob efeito de distorção do ruído, para transmitir a uma noção de tendência local da janela (temperatura em crescimento ou decrescimento) ao mesmo tempo em que omite estrategicamente características intrínsecas ao formato do sinal, como periodicidade, as quais prejudicam a capacidade de generalização da rede. Nas figuras 21, 22 e 23 estão expostas representações gráficas da análise de diferentes valores de n .

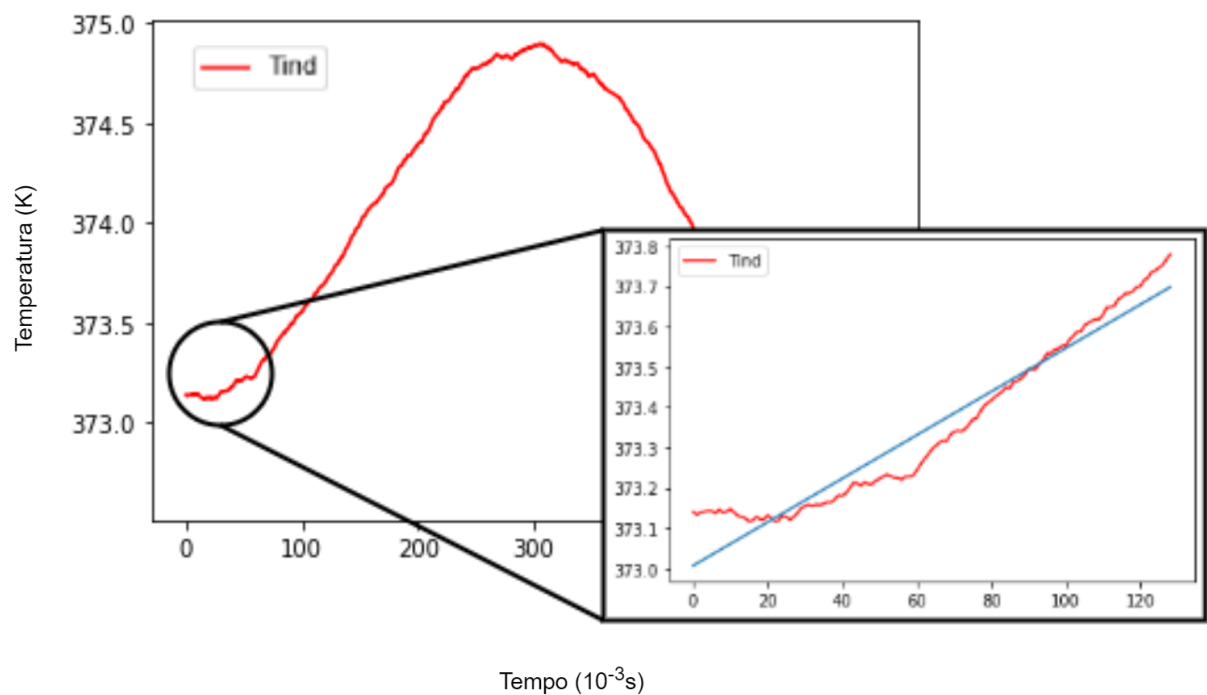


Figura 22: Representação da tendência local da janela de 128 pontos tirada da T_{ind} de um ciclo do sinal seno.

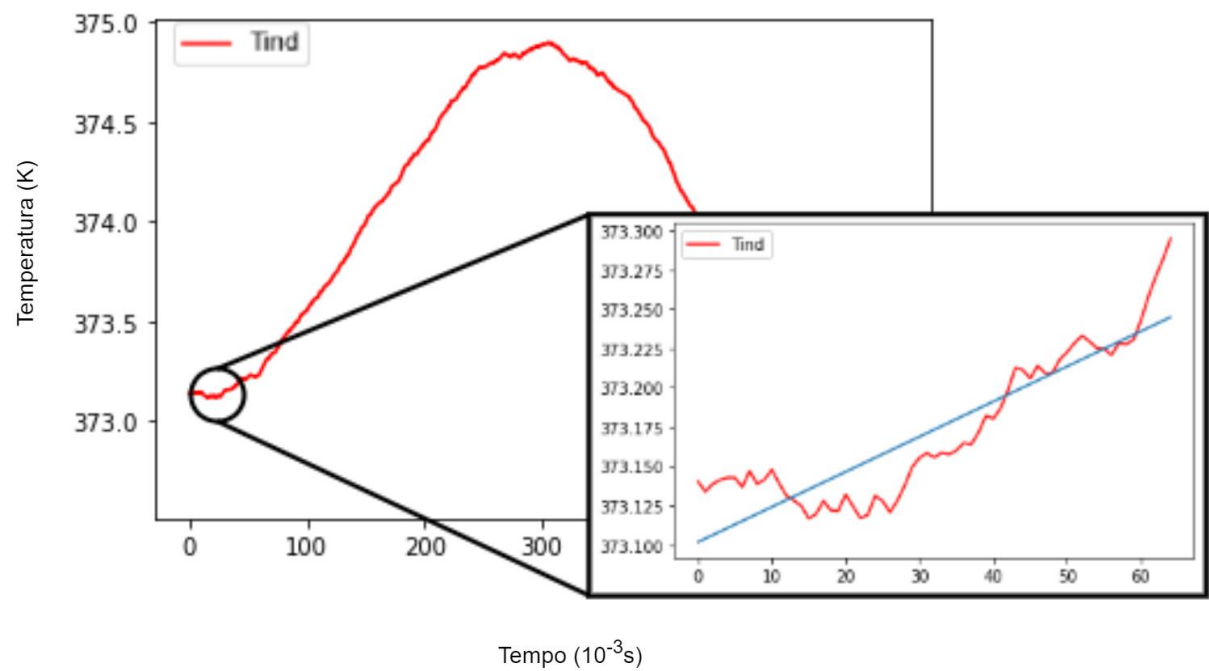


Figura 23: Representação da tendência local da janela de 64 pontos tirada da T_{ind} de um ciclo do sinal seno.

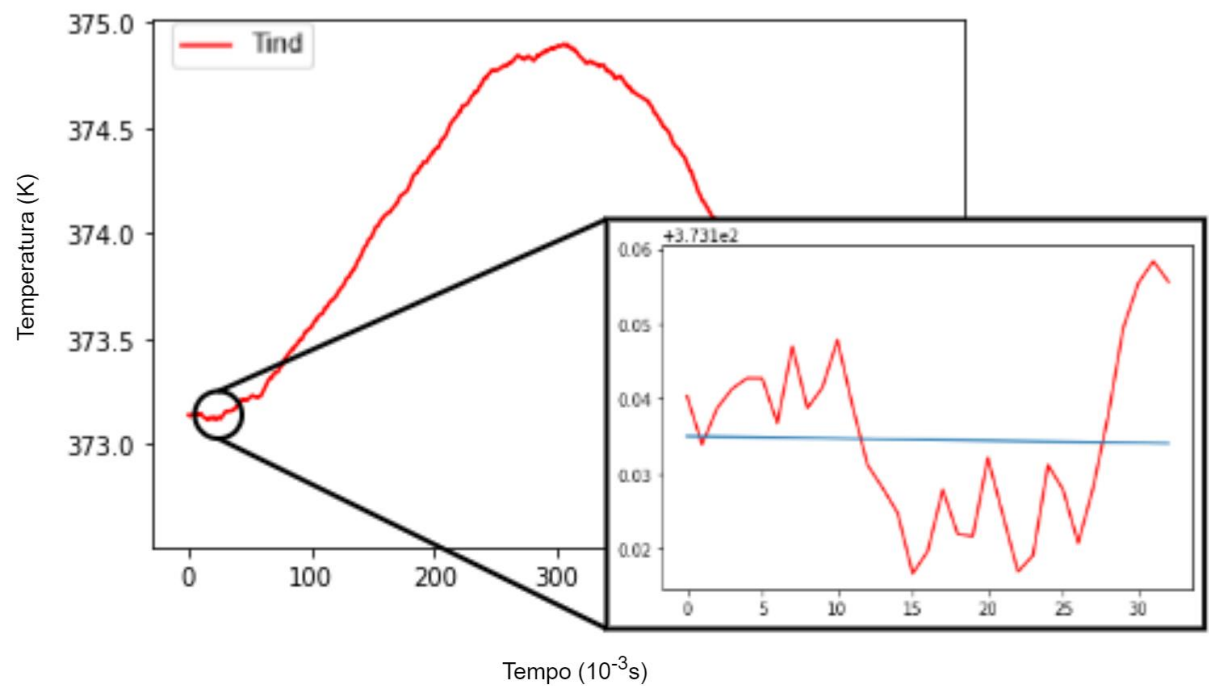


Figura 24: Representação da tendência local de uma janela de 32 pontos tirada da T_{ind} de um ciclo do sinal seno.

Observa-se, na figura 23, que janelas muito pequenas sofrem do mesmo problema do método ponto a ponto, deformações ruidosas na T_{ind} descaracterizam excessivamente os parâmetros de entrada.

Os resultados obtidos pela rede de previsão da série temporal são apresentados nas figuras 24, 25 e 26. Nota-se também nos resultados que a curva de temperatura gerada pela rede é bem mais suave do que a gerada pela rede no método ponto a ponto, a qual também possui apenas um neurônio na camada de saída. Esse fenômeno se deve ao fato de que a janela de memória conferida à rede concede ao método robustez em relação ao ruído.

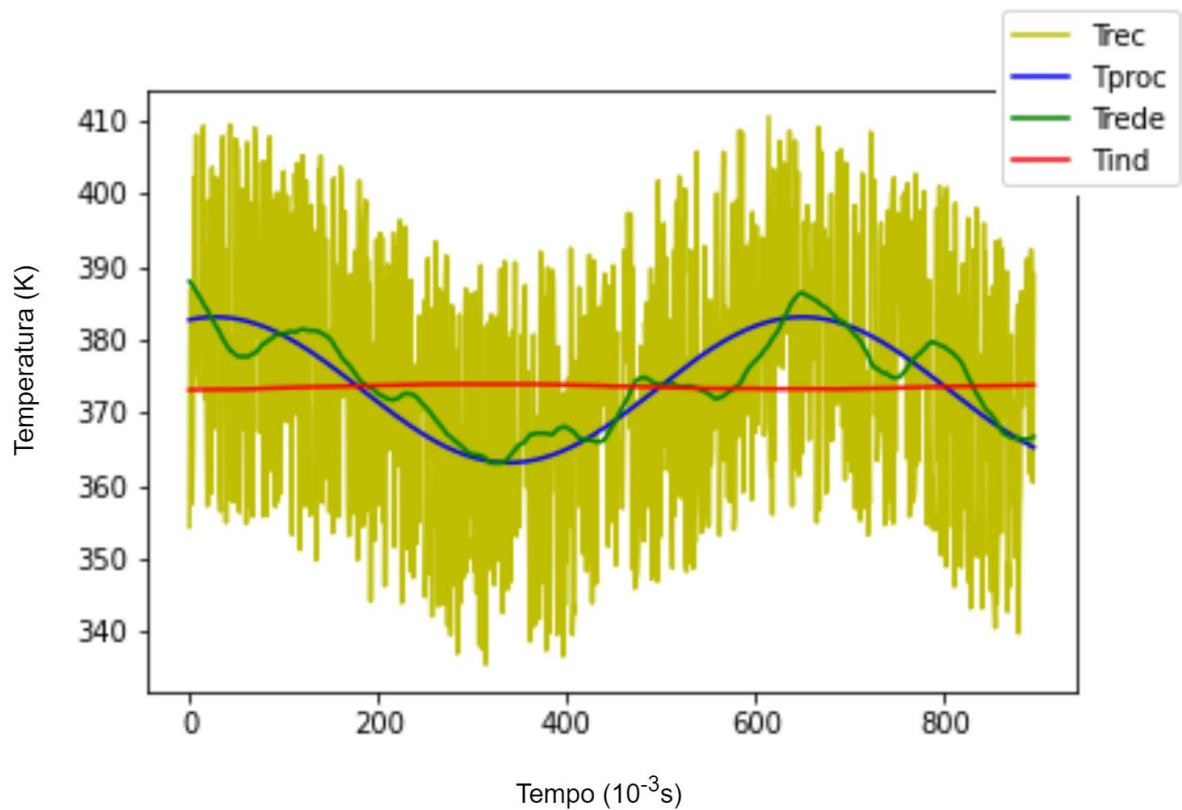


Figura 25: Exemplo de um sinal de h baixo e do tipo Seno, regularizado pela rede a partir do método de previsão da série temporal.

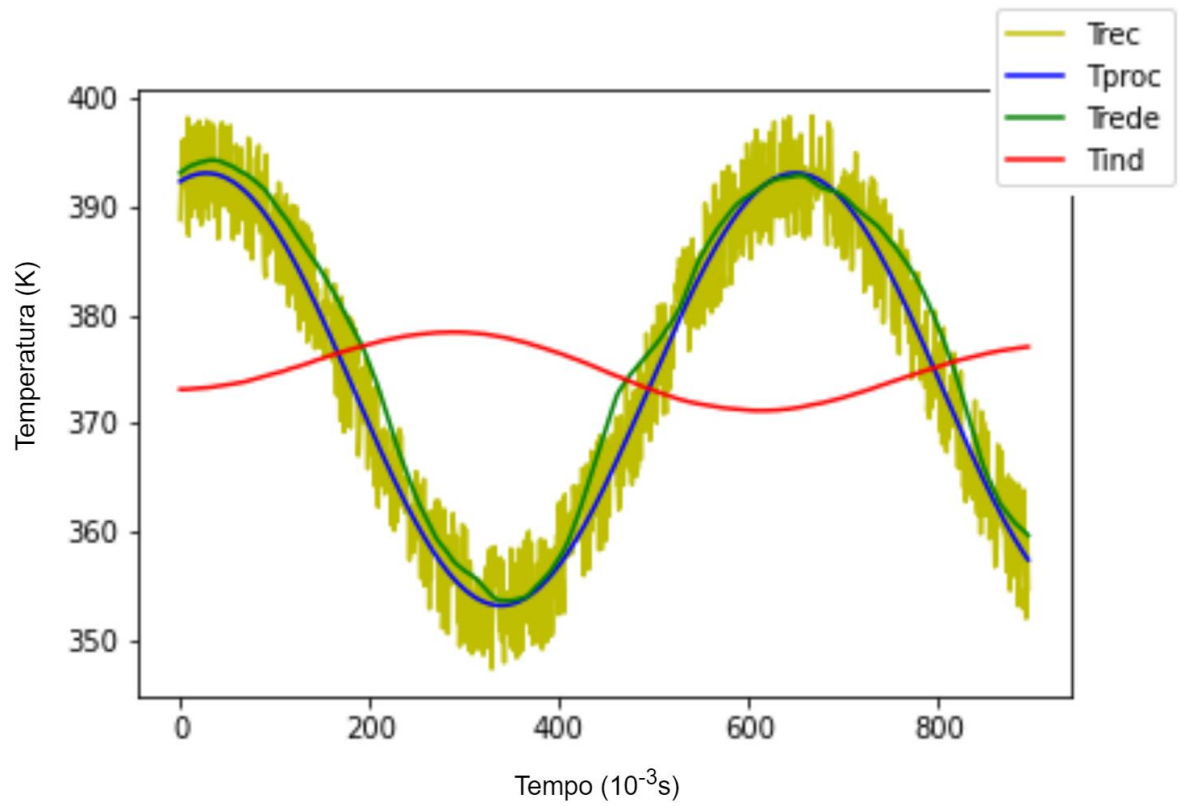


Figura 26: Exemplo de um sinal de h médio e do tipo Seno, regularizado pela rede a partir do método de previsão da série temporal.

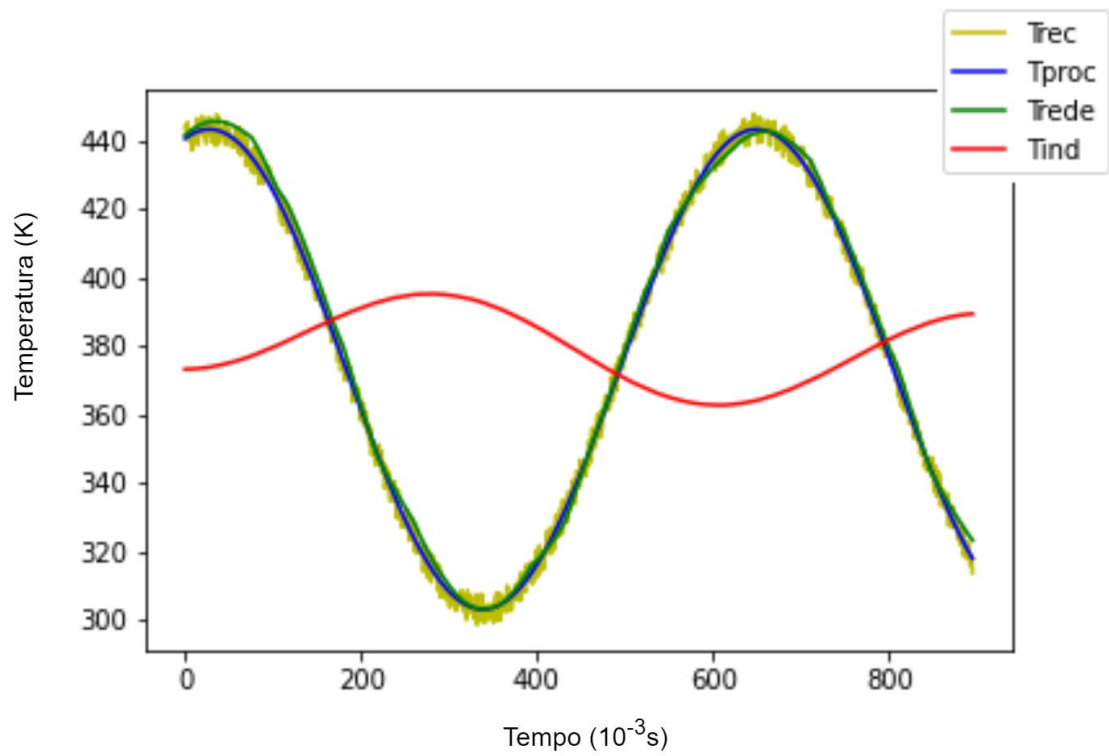


Figura 27: Exemplo de um sinal de h alto e do tipo Seno, regularizado pela rede a partir do método de previsão da série temporal.

5. CONCLUSÃO

Este trabalho apresentou três modelos de redes, regressão ponto a ponto, regressão multivariável e previsão de série temporal, em que foram capazes de regularizar um sinal simulado proveniente de problemas inversos em transferência de calor. A partir dos resultados obtidos, o método de regressão multivariável se sobressaiu sobre o método de previsão da série temporal, o qual se apresenta levemente superior ao método de regressão ponto a ponto. É importante ressaltar, porém, que cada uma das metodologias possuem suas vantagens e limitações.

O método de regressão multivariável, por conferir à rede a maior quantidade de memória por iteração, gera a regularização mais precisa, porém limita-se ao inviabilizar a aplicação da solução em processos em tempo real, assim como apresenta uma solução altamente especializada quanto à morfologia do sinal, por esse motivo, demanda a existência de um banco de dados de treinamento extremamente robusto, contendo grande número de sinais, de forma a contemplar todo o espaço matemático abrangido pelo problema.

O método de regressão ponto a ponto, por sua vez, entrega a rede mais flexível e generalizadora dentre as três implementadas e treinadas, altamente independente da morfologia ou tipo do sinal, ao custo de se tornar também altamente sensível às deformações ruidosas da temperatura, penalizando portanto a sua capacidade regularizadora como um todo.

Já o método de previsão da série temporal, se propõe como uma solução equilibrada dentre os métodos de regressão ponto a ponto e regressão multivariável. O método de previsão da série temporal se utiliza da capacidade de atenuação de ruído da rede enquanto exerce sua capacidade preditiva sobre o problema de previsão. A memória parcial, ou seja, a janela móvel de n pontos entregue a rede não sacrifica a possibilidade de aplicação da solução em tempo real, ao mesmo tempo em que exerce um aproveitamento bastante eficiente dos dados, já que gera (tamanho do sinal - n) exemplos por sinal dentro do banco de treinamento. Essa janela proporciona também, a transmissão de informações cruciais sobre o sinal, que em conjunto com o parâmetro do acúmulo térmico h , permite com que a rede contextualize o *datapoint* em relação ao sinal completo de maneira bastante eficiente.

Este trabalho verificou a viabilidade e explorou as peculiaridades da aplicação das redes neurais para o problema experimental mal posto. É importante ressaltar que está fora do escopo do mesmo o desenvolvimento de uma RNA específica, pois cada processo demanda uma coleta de dados especializada, a qual seja representativa do problema a ser solucionado. Assim, como evidenciado por este estudo, diversos outros estudos na literatura já vem demonstrando a eficácia do uso das RNA's para essa classe de problemas.

Atualmente, arquiteturas de rede bem mais robustas do que o simples *perceptron* de arquitetura *feedforward* já foram desenvolvidas para lidar com os mesmos, como as redes recorrentes, redes de base radial, redes de longa memória de curto prazo, entre outras.

Futuramente, o direcionamento do trabalho deve ser feito para a coleta de dados experimentais, onde poderá ser explicitada as dependências e limitações dos modelos propostos em relação às morfologias dos processos a serem monitorados em si, ou seja, aos formatos dos sinais de temperatura. Outra maneira de alcançar o mesmo objetivo seria através da criação de um banco de dados contendo exemplos de sinais com características mais “orgânicas” e menos previsíveis, para dificultar cada vez mais o processo de absorção de padrões específicos ao formato do sinal, e dessa forma, agregar robustez e generalidade ao modelo proposto.

6. REFERÊNCIAS

ABADI, M. et al. **Tensorflow: A system for large-scale machine learning**. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), p. 265-283, 2016.

ACTIVATION Function - AI Wiki. Disponível em: <https://machine-learning.paperspace.com/wiki/activation-function>. Acesso em: 15 set. 2022.

BADRI, L. **Development of Neural Networks for Noise Reduction**. Lubna Badri Faculty of Engineering, Philadelphia University, Jordan. The International Arab Journal of Information Technology, v. 7, jul. 2010.

BORGES, E.; MENEZES, D. C. **Resolução de problemas inversos em cinética química usando redes neurais artificiais**. The Journal of Engineering and Exact Sciences, v. 3, n. 3, 9 maio 2017.

BRAGA, J. P. et al. **ILL-POSED INVERSE PROBLEMS IN CHEMISTRY**. Química Nova, 2016.

CRONE, S.; KOURENTZES, N. **Input variable selection for time series prediction with neural networks- an evaluation of visual, autocorrelation and spectral analysis for varying seasonality**. 2007. Disponível em: https://kourentzes.com/forecasting/wp-content/uploads/2014/04/ESTSP07_Crone_Kourentzes.pdf. Acesso em: 15 set. 2022.

DA SILVA, I.N. et al. **Redes neurais artificiais para engenharia e ciências aplicadas: curso prático**. Artliber Editora Ltda, São Paulo, 2010.

DAYHOFF, J. E.; DELEO, J. M. **Artificial Neural Networks: Opening the Black Box**. Cancer supplement, v. 91, n. 8, 15 abr. 2001.

FERNANDEZ, J. **Time-Series Forecasting Based on Trend and Seasonal components**. 2020. Disponível em: <https://towardsdatascience.com/time-series-forecasting-based-on-the-trend-and-seasonal-components-26b92866e548>. Acesso em: 01 nov. 2022.

GERON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. O'Reilly Media, 2019.

GULLI, A.; PAL, S. **Deep learning with Keras**. Packt Publishing Ltd., 2020.

HARRIS, C.R. *et al.* **Array programming with NumPy**. Nature, 585, p.357-362, 2020.

HADAMARD, J. **Sur les problèmes aux dérivées partielles et leur signification physique**. Princeton University Bulletin, p. 49-52, 1902.

HAYKIN, S. **Neural networks: a comprehensive foundation**. Prentice Hall, 1999.

HUNTER, J. D. **Matplotlib: A 2D graphics environment**. Computing in science & engineering, v. 9, n. 3, p.90-95, 2007.

IBM CLOUD EDUCATION. **O que são Redes Neurais?** 2020. Disponível em: <https://www.ibm.com/br-pt/cloud/learn/neural-networks>. Acesso em: 23 nov. 2022.

KINGMA, D. P.; BA, J. A.: **A Method for Stochastic Optimization**. In: 3rd International Conference for Learning Representations, San Diego, 2015. 2017. Disponível em: <https://arxiv.org/abs/1412.6980>. Acesso em: 23 nov. 2022.

MCCULLOCH, W.; PITTS, W. **A logical calculus of ideas immanent in nervous activity**. Bulletin of Mathematical Biophysics, v. 5, p. 115-133, 1943.

MCKINNEY, W. *et al.* **Data structures for statistical computing in python**. In Proceedings of the 9th Python in Science Conference, v. 445, p. 51-56, 2010.

OLIVEIRA, J. *et al.* **Inverse measurement method for detecting bubbles in a fluidized bed reactor—toward the development of an intelligent temperature sensor**. Powder Technology, v. 169, n. 3, p. 123-135, 2006.

PEDREGOSA, F. *et al.* **Scikit-learn: Machine learning in Python**. The Journal of Machine Learning Research, v. 12, p. 2825-2830, 2011.

POGGIO, T.; GIROSI, F. **Networks for approximation and learning**. Proceedings of the IEEE, v. 78, p. 1481-1497, 1990.

PRAKARSHA, K. R.; SHARMA, G. **Time series signal forecasting using artificial neural networks: An application on ECG signal**. Biomedical Signal Processing and Control, v. 76, p. 103705, jul. 2022.

PYTHON SOFTWARE FOUNDATION. **Python Language Site: Documentation**. 2020. Disponível em: <https://docs.python.org/3/>. Acesso em: 06 de nov. de 2020.

SHARMA, A. **New Understanding Activation Functions in Deep Learning**. 2017. Disponível em: <https://learnopencv.com/understanding-activation-functions-in-deep-learning/>. Acesso em: 15 set. 2022.

SILVA NETO, A. J.; MOURA NETO, F. D. **Problemas inversos em engenharia**. In: Congresso Brasileiro de Engenharia e Ciências Térmicas, 8., 2000, Porto Alegre. Mercofrio 2000 [recurso eletrônico]... Porto Alegre: UFRGS. 1 CD-ROM. 2000.

TIKHONOV, A.N.; ARSENIN, V.Y. **Solutions of ill-posed problems**. Washington; New York, Winston, 1977.