UNIVERSIDADE DE SÃO PAULO

ESCOLA DE ENGENHARIA DE LORENA

LUCAS DA SILVA ABREU

Estimação da acidez de óleos vegetais via nariz eletrônico através de uma rede neural feed-forward

Lorena

2019

LUCAS DA SILVA ABREU

Estimação da acidez de óleos vegetais via nariz eletrônico através de uma rede neural feed-forward

Projeto de pesquisa apresentado à Escola de Engenharia de Lorena da Universidade de São Paulo como requisito parcial para conclusão do Curso de Graduação em Engenharia Química.

Orientador: Prof. Dr. Adriano Francisco Siqueira

Versão Original

Lorena

2019

Ficha catalográfica elaborada pelo Sistema Automatizado
da Escola de Engenharia de Lorena,
com os dados fornecidos pelo(a) autor(a)

Abreu, Lucas da Silva
Estimação da acidez de óleos vegetais via nariz eletrônico através de uma rede neural feed-forward / Lucas da Silva Abreu; orientador Adriano Francisco Siqueira. - Lorena, 2019.
66 p.

Monografia apresentada como requisito parcial para a conclusão de Graduação do Curso de Engenharia Química - Escola de Engenharia de Lorena da Universidade de São Paulo. 2019

1. Redes neurais. 2. Estimação de propriedades físico-químicas. 3. Modelagem estatística. 4. Nariz eletrônico. I. Título. II. Siqueira, Adriano Francisco, orient.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais Omar Iran Abreu e Sônia Maria da Silva por terem sempre me mostrado e motivado a importância e beleza dos estudos não só para o desenvolvimento pessoal como para o profissional. Agradeço também ao corpo docente da EEL–USP ao longo desses 7 anos de aprendizado nos mais diversos campos, em especial ao professor Adriano Siqueira que foi quem me mostrou pela primeira vez a beleza por trás da matemática e suas inúmeras aplicações. Também agradeço a todos os colegas da faculdade que estiveram presente ao longo desse trajeto oferecendo suporte nos mais diversos momentos, em especial a Melina Murgel, Gabriel Borges, Henrique Oya, Lucas Barbosa, Inocêncio Costa, Catherine Gazolla, Ana Paula Sanches e Ramon Lucena por todo suporte direto na faculdade e também a amigos fora da faculdade, Rodrigo Oliveira, Ana Rita e Mateus Spessotto que sempre ofereceu ajudas em situações difíceis e ao João Victor que foi quem me ajudou de forma mais consistente quando comecei a me interessar mais por programação.

RESUMO

ABREU, L. S. **Estimação da acidez de óleos vegetais via nariz eletrônico através de uma rede neural feed-forward.**2019. Trabalho de Conclusão de Curso – Escola de Engenharia de Lorena, Universidade de São Paulo, Lorena, 2019.

A modelagem estatística tem sido de grande valia nos mais variados campos de conhecimento, como análise de fraudes, métodos de classificação e mais recentemente aplicações como identificação de imagens. Em tais aplicações, um modelo que tem ganhado bastante atenção nos últimos tempos são as redes neurais artificias. As redes neurais têm sido utilizadas como sistemas de classificação em narizes eletrônicos para identificação das substâncias analisadas com o nariz.

O presente trabalho propôs-se a modelar os dados coletados com auxílio de um nariz eletrônico utilizando uma rede neural do tipo *feed-forward* para estimação da acidez de óleos vegetais. A pequena quantidade de amostras presentes disponíveis para treino foi uma dificuldade encontrada no processo de treinamento da rede de forma que este obteve um coeficiente de correlação linear no conjunto de validação máximo de 0,28.

Uma regressão linear multivariada foi desenvolvida de modo a permitir uma comparação de desempenhos. A regressão obteve na tarefa em questão valores superiores ao da rede neural: 0,9664 de coeficiente de correlação linear no conjunto de treino e 0,9150 no de teste.

Palavras-chave: Redes neurais. Estimação de propriedades físico-químicas. Modelagem estatística. Nariz eletrônico.

ABSTRACT

ABREU, L. S. **Estimation of vegetable oils' acidity with a electronic nose through a feed-forward neural network.**2019. Trabalho de Conclusão de Curso – Escola de Engenharia de Lorena, Universidade de São Paulo, Lorena, 2019.

Statistical modelling has been of great value in several knowledge fields, such as fraud analysis, clustering methods and lately on identification of images. In those applications, a model that has been given special attention lately are artificial neural networks. They have been applied as classifications systems in electronic noses for compound identification.

The current work aimed at modelling a feed-forward neural network using data read by an electronic nose for estimation of vegetable oil's acidity. The small number of samples available proposed itself as a challenge during the training of the network resulting in a linear correlation coefficient in the validation set of 0,28.

A multivariate regression line was developed in order to establish a comparison between the model's performance. The regression line performance was much better than the neural network one, achieving the value of 0,9664 for the linear correlation coefficient in the training set and 0,9150 in the validation set.

Keywords: Artificial Neural Networks. Estimation of physico-chemical properties. Statistical modelling. Electronic Nose.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Adam	Adaptive Moment Estimation
AOAC	Association of Analytical Chemists
BP	Backpropagation
CVNN	Complex Valued Neural Networks
DNN	Deep Neural Networks
FPE	Estados precursores que ainda gerarão sinal
FPEr	Estados precursores que ainda gerarão sinal levando-se em conta Tr (momento de remoção de substâncias voláteis do sensor)
ICA	Independent Component Analysis
LDA	Linear Discriminant Analysis
LSM	Liquid State Machine
MAE	Mean Absolute Error
PCA	Principal Component Analysis
PE	Estado Precursor
PEr	Estado Precursor levando-se em conta Tr (momento de remoção de substâncias voláteis do sensor)
PE’	Estado Gerador de Sinal
PE’r	Estado Gerador de Sinal levando-se em conta Tr (momento de remoção de substâncias voláteis do sensor)
R^2	Coeficiente de correlação linear
ReLU	Rectified Linear Unit
RMSprop	Root Mean Squared Prop
SOM	Self Organizing Maps
SGD	Stochastic Gradient Descent
Tanh	Tangente hiperbólica

LISTA DE SÍMBOLOS

a	Parâmetro a da modelagem da equação estocástica
a	Parâmetro a da modelagem do sinal do nariz eletrônico
A	Platô atingido pelo sinal do nariz eletrônico
b	*Bias*
b	Parâmetro b da modelagem da equação estocástica
b	Platô horizontal da modelagem do sinal do nariz eletrônico
c	Parâmetro c da modelagem da equação estocástica
ϵ	Hiper parâmetro do algoritmo *Adam* para controle da atualização da matriz de pesos
e	Logaritmo natural
E	Média móvel do segundo momento do vetor gradiente
E	Função erro médio quadrado
$f(u)$	Função de ativação genérica de uma rede neural
g	Vetor gradiente
H	Função de Heaviside
k	Parâmetro da modelagem do sinal do nariz eletrônico
k	Parâmetro k da modelagem da equação estocástica
k_1	Constante de adsorção
k_2	Constante de dessorção
k_3	Constante de conversão de estado precursor em estado gerador de sinal
k_4	Constante de liberação da molécula pós geração de sinal
k_e	Constante de equilíbrio entre os processos de adsorção e dessorção

k_s Constante de proporcionalidade ente taxa de variação do sinal e ocupação de sítios livres

L Fração de sítios do nariz eletrônico não ocupados

m Primeiro momento do vetor gradiente

$\hat{m}$ Primeiro momento corrigido do vetor gradiente

M Número de moléculas livres

n Número de substâncias ligadas aos sítios ativos do nariz eletrônico

N Número total de sítios ativos do nariz eletrônico

p Parâmetro p da modelagem da equação estocástica

S Sinal lido pelo nariz eletrônico

Sr Sinal lido pelo nariz eletrônico levando-se em conta Tr

t' Tempo transcorrido desde o início da medição levando-se em conta Tr

Tr Tempo quando ocorre a remoção de substâncias voláteis do sensor

Q Quadrado da variância do sinal do nariz eletrônico divido pelo tempo

u Resultado do somatório do produto da matriz de pesos de uma rede neural com sua entrada acrescido do bias

v Segundo momento do vetor gradiente

$\hat{v}$ Segundo momento corrigido do vetor gradiente

w Matriz de peso

W Distribuição de Wiener

y Variável resposta

$\hat{y}$ Variável resposta calculada pela rede

X Medida da variabilidade do sinal do nariz eletrônico

LETRAS GREGRAS

α	Taxa de aprendizado
β_1	Taxa de decaimento exponencial do primeiro momento do vetor gradiente
β_2	Taxa de decaimento exponencial do segundo momento do vetor gradiente
ΔR	Diferença de resistência registrada por um nariz eletrônico
∂	Derivada parcial
σ	Função sigmoide
Σ	Operador somatório
θ	Matriz de pesos

SOBRESCRITO

I	Época de treinamento
t	Época de treinamento
t	Tempo de exposição do nariz eletrônico à substância

SUBESCRITO

i	Posição de uma camada dentro da rede
j	Posição de um neurônio dentro da rede na camada i -1
k	Posição de um neurônio dentro da rede na camada i
t	Época de treinamento

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A partir do advento das máquinas e de suas aplicações em tarefas onde seguia-se uma linha de execução sequencial e em ambientes conhecidos, o homem começou a imaginar a criação de máquinas autônomas que pudessem operar também em ambientes desconhecidos.

Logo, a criação de máquinas autônomas traria então os benefícios das máquinas, como precisão numérica e velocidade de execução, para junto de características de processamento do cérebro humano. Características como interpolação, paralelismo, generalizações, tolerância a erros e o mais desejado, a capacidade de aprendizado, o que possibilitaria às máquinas a aprendizagem contínua mesmo em ambientes totalmente novos (MAO, 1996).

Esses ambientes novos exigiriam das máquinas uma elasticidade no processamento da informação que tornaria o processo, assim como no cérebro, não sequencial a fim de conseguir compreender o novo meio e conseguir atuar de formar consistente. Nesse sentido os modelos de redes neurais artificiais aplicam uma estrutura análoga à usada no funcionamento do cérebro (MAO, 1996), tendo-se então como componente básico de uma rede neural artificial o neurônio, que se dispõe em camadas interligadas sucessivamente, estando cada neurônio conectado com pelo menos outro neurônio da rede.

De acordo com SVOZIL; KVASNICKA; POSPÍCHAL (1997) o funcionamento básico de uma rede neural se dá da seguinte forma:

1. Recebimento por um neurônio de dados de entrada, podendo esses dados serem variáveis de entrada da rede ou respostas emitidas por outros neurônios*;*
2. Processamento dos dados, que consiste na multiplicação matricial dos dados de entrada com uma matriz de pesos e adição de um *bias*. Onde o valor de cada peso da matriz reflete a influência de cada dado na resposta emitida pelo neurônio e o *bias* pode ser interpretado como um segundo ajuste para um melhor resultado de saída;
3. Aplicação do resultado obtido no passo 2 em uma função denominada função de ativação, o que confere às redes a capacidade de modelar processos não lineares;
4. Disparo do resultado da função de ativação proveniente de um neurônio para outros neurônios da rede ou como variável de saída da rede neural.

E no que tange à forma como as redes refinam seus algoritmos, de forma a aumentarem sua precisão, tem-se três grupos: treinamento supervisionado, treinamento não supervisionado e treinamento por reforço. O treinamento supervisionado se dá através da comparação do valor de saída da rede para um dado conjunto de dados com o valor real.

No treinamento supervisionado a rede realiza cálculos para reduzir a diferença entre o valor de saída da rede e o valor real para o valor mais próximo possível de zero. Já nos outros dois modelos, respectivamente, a aprendizagem se dá através da detecção de padrões nos dados de entrada e através de estímulos a rede dependendo da resposta que ela obtém para um conjunto de dados (PRIETO *et al*., 2016).

Segundo CROSS; HARRISON; KENNEDY (1995) ao contrário dos computadores convencionais, as redes neurais artificiais têm seu poder computacional proveniente da densidade e da complexidade das conexões entre as camadas de neurônios, ao ponto de que uma rede neural artificial com pelo menos uma camada aproxima de forma satisfatória qualquer outro modelo (DEBAO, 1993), isto é, modelos que fazem aproximações de funções, classificadores de padrões e *clusterizadores*, que é o agrupamento de dados (GORGENS *et al*., 2009).

Como reportado por PRIETO *et al*. (2016), as redes neurais possuem aplicações nos mais variados campos, como por exemplo:

- Na medicina através da classificação de imagens biomédicas;
- Na química com a modelagem de processos na área de química analítica;
- Na genética com a modelagem de genomas, como da *Drosophila Melanogaster*;
- Na meteorologia com a predição do clima e classificação de nuvens.

Focando no âmbito das engenharias, temos na engenharia ambiental a determinação da qualidade de águas potáveis (SALARI *et al*., 2018) e na engenharia química na análise de falhas em processos químicos como nos trabalhos de WU e ZHAO (2018) e também no trabalho de ZHANG e ZHAO (2017).

Ainda no âmbito da engenharia química, uma aplicação das redes neurais seria a identificação de propriedades de substâncias químicas, puras ou não, baseadas em um conjunto de dados de entrada que é utilizado para treinamento da rede neural. Nesse caso, o conjunto de dados é obtido a partir de leituras de um nariz eletrônico através da passagem de vapores da substância sob análise pelo equipamento.

O primeiro esboço de um nariz eletrônico foi com Persaud & Dood em 1982 como sendo um sistema que pudesse realizar leituras de amostras e então identificar a

substância em questão de forma mais rápida do que por métodos usuais (MAJCHRZAK *et al*., 2018).

Segundo GHASEMI-VARNAMKHASTI *et al*. (2018), o nariz eletrônico tem uma estrutura próxima da do nariz humano compreendendo assim uma sequência de sensores com especificidades parciais a alguns compostos e um sistema de detecção apto a identificar desde odores simples até mais complexos.

O funcionamento se dá pela passagem dos aromas pela sequência de sensores de forma a gerar um sinal elétrico associado à substância de análise. Esse sinal elétrico depende da variação da resistência elétrica dos sensores quando uma molécula adsorve nos sensores. Uma vez gerado o sinal, ele é lido por um software, pré tratado, visando eliminar ruídos e usado como entrada em um software de reconhecimento que tem como função identificar a substância que está sob análise (GHASEMI-VARNAMKHASTI *et al*., 2018).

1.2 JUSTIFICATIVA

Um dos problemas enfrentados por modelos utilizando nariz eletrônico é a perda de validade do modelo quando se utiliza um volume grande de amostras para validação devido ao excesso de ruído gerado por algumas substâncias. Uma solução foi obtida por SIQUEIRA *et al*. (2018) que fez uso do ruído gerado pelas leituras como variável de entrada no modelo por ele descrito.

O nariz eletrônico apresenta-se especialmente útil na aferição da qualidade de odores quaisquer emitidos por períodos prolongados, tanto porque a exposição repetida a odores tende a enviesar avaliações humanas sobre a qualidade do odor emitido, quanto porque os odores podem ser tóxicos (LISBOA; PAGÉ; GUY, 2009).

Dentre outras aplicações de destaque para o nariz eletrônico, segundo LISBOA; PAGÉ; GUY (2009), temos:

- Aferição de níveis de glicose em pacientes diabéticos, entre outras patologias como a tuberculose;
- Monitoramento de processos de cozimento;
- Controle de qualidade de fermentados como queijos e cervejas;
- Análises de água e esgoto.

Tendo em vista as aplicações do nariz eletrônico apresentadas, uma outra possível aplicação, tendo em vista a solução e resultados obtidos por SIQUEIRA *et al*. (2018), é o

uso dos parâmetros de seu modelo em uma rede neural para estimação da acidez de óleos vegetais, o que possibilitaria análises mais rápidas em procedimentos rotineiros de laboratório, por exemplo.

1.3 OBJETIVOS

1.3.1 Objetivos gerais

Desenvolver uma rede neural que receba os sinais lidos por um nariz eletrônico para estimação da acidez de óleos vegetais.

1.3.2 Objetivos específicos

O presente trabalho tem como objetivo a estimação da acidez de óleos vegetais através de uma rede neural *feed-forward* treinada utilizando o algoritmo de *backpropagation* e dados obtidos através de leituras realizadas com um nariz eletrônico.

2 REVISÃO DA LITERATURA

2.1 REDES NEURAIS

2.1.1 Histórico

O desenvolvimento das redes neurais pode ser dividido historicamente em quatro maiores momentos (PRIETO *et al*., 2016):

1. Entre 1940 e 1950 com McCulloch e Pits, que propuseram o primeiro modelo formal de um neurônio. Esse modelo considerava a existência do neurônio e de sua memória associativa, pensada justamente no modelo dos neurônios e suas interações entre si.

   Algumas descobertas à época sobre as redes neurais biológicas influenciaram grandemente no desenvolvimento das redes, como a ideia proposta por Hodgkin e Hulexy e pelo psicólogo Hebb. Hodgkin e Hulexy propuseram equações sinápticas em 1942 e em 1949, segundo Hebb, os neurônios guardavam informações de suas sinapses e faziam uso delas para novos aprendizados;
2. Entre 1960 e 1970 com o desenvolvimento dos algoritmos de aprendizagem tanto para redes de uma única camada como para redes recorrentes. Dentre os métodos, temos o método dos mínimos quadrados, implementação de memórias associativas, entre outros.

   Minsky e Paper, em 1967, publicaram um livro no qual afirmavam que algumas tarefas exigiriam interconexão entre os neurônios. Essa interconexão, na época, era ainda impossível devido à falta de equações que se adequassem a esse modelo, fazendo com que o estudo de redes neurais estagnasse durante alguns anos;
3. Entre 1980 e 1990 com um ressurgimento do interesse pelas aplicações com redes neurais, estudos mais focados sobre a auto-organização das redes e expansão para redes com mais de duas camadas ocultas foram desenvolvidos. Também caracterizado pela aplicação de métodos Bayesianos e Gaussianos às redes neurais, o que trouxe a aprendizagem de máquinas mais próxima da teoria de probabilidade.

   Uma série de topologias e técnicas novas foram descobertas no período, como por exemplo:

   - modelos do tipo *self-organizing maps* (SOM), que buscam padrões mais implícitos em conjuntos de dados;

- *indepedent component analysis* (ICA), que foi inicialmente aplicado no problema da separação cega de fontes em 1985 por Herault, Jutten e Anns;
- o algoritmo de *back-propagation*, que sanava os problemas citados por Minsky e Paper em 1967.

A aplicação dos métodos Bayesianos e Gaussianos nos modelos de redes neurais vigentes à época trouxe esse campo mais próximo à teoria de probabilidade culminando com a criação de um campo denominado *Statistical Machine Learning*;

4. De 2000 até o presente com a tentativa de melhorar a performance dos modelos através de técnicas mais avançadas de otimização.

   Modelos que ganharam uma maior atenção no período foram as *Complex-Valued Networks* (CVNN) e as *Deep Neural Networks* (DNN). CVNNs são redes neurais nas quais qualquer variável pode assumir valores complexos, sendo especialmente útil em modelagens que envolvam ondas eletromagnéticas.

   As DNNs são compostas de camadas ocultas dispostas em série, aumentando a capacidade de processamento da rede neural, sendo útil especialmente em tarefas mais complexas como processamento de linguagem natural e visão computacional. Porém um limitante no uso de DNNs é o alto custo computacional necessário para treinamento e teste, de forma que métodos para amenizar esse custo tem sido alvo de bastante interesse.

Em 2002, Maass apresentou o conceito de *Liquid state machine* (LSM), que sanava o problema de processamento em tempo real de variáveis. Nesse modelo, o neurônio recebe uma série de dados de entrada conseguindo transformar um estado transiente em uma resposta estável através do aprendizado da noção de igualdade por cada neurônio. Aplicações de LSMs resultaram em modelagens ainda mais próximas do real funcionamento de sistemas biológicos, a saber as sinapses.

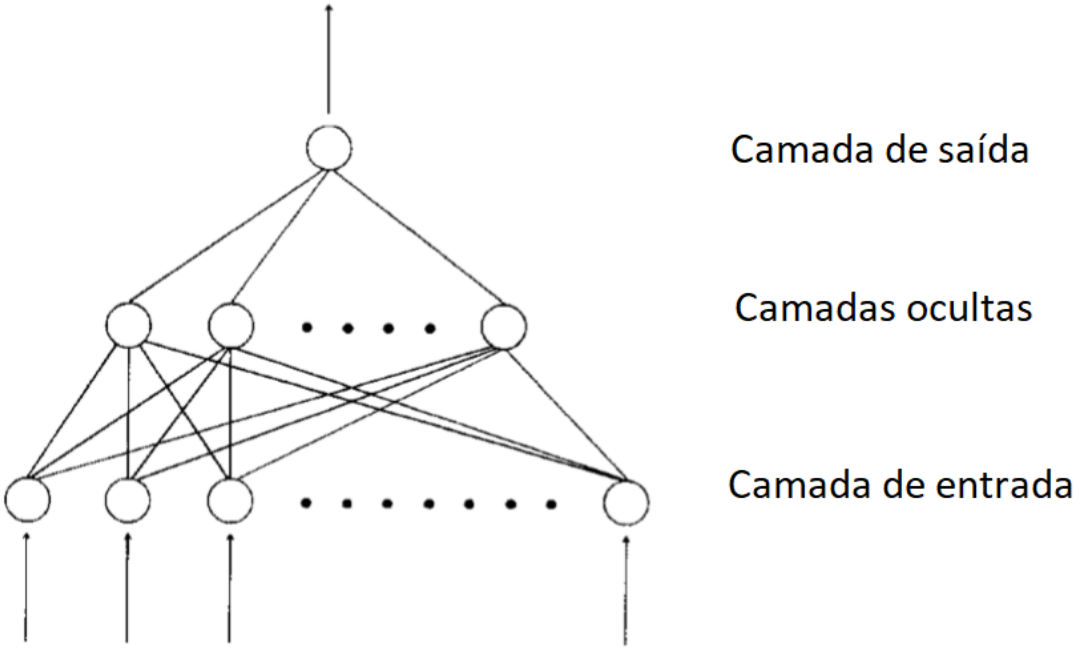
Um dos tópicos de maior interesse da quarta fase foi o melhor dimensionamento do tamanho da rede de forma a obter o melhor desempenho com o menor custo computacional.

2.1.2 Estrutura de redes neurais do tipo *feed-forward*

O componente básico de uma rede neural é conhecido como neurônio. Cada neurônio recebe variáveis de entrada, processa-as e emite uma resposta de saída. Em uma rede neural, cada neurônio está conectado com pelo menos um outro neurônio da rede de forma que a resposta de saída de um neurônio é comumente utilizada como variável de entrada de outro.

Segundo BRAGA; CARVALHO; LUDERMIR (2000) definem-se três tipos de camada em uma estrutura de rede neural: a camada de entrada, a camada de saída e camadas que não a de entrada e a de saída, denominadas ocultas. A camada de entrada é responsável pelo envio sem quaisquer modificações dos sinais de entrada para a rede neural; a de saída pelo envio da(s) variável(is) resposta(s) da rede e as camadas ocultas pelo tratamento e envio dos sinais provenientes de camadas anteriores. O esquema de uma rede neural *feed-forward* pode ser encontrado na Figura 1.

Figura 1 – Modelo de rede neural *feed-forward*



Fonte: Adaptada de SVOZIL; KVASNICKA; POSPÍCHAL (1997)

Segundo HAYKIN (2001), três componentes básicos encontrados em um neurônio são:

1. Conjunto de pesos que faz menção ao peso de cada variável de entrada sobre a variável de saída. A notação de cada peso, $w_{kj,}$ constitui-se de dois números kj: k é o índice do neurônio na camada e j a posição do neurônio na camada anterior;
2. Um operador somador de todos sinais recebidos por cada neurônio;
3. Uma função denominada função de ativação que permite que a rede modele problemas que não são linearmente separáveis. O aspecto mais importante que a função de ativação de escolha deve ter é que ela seja diferenciável.

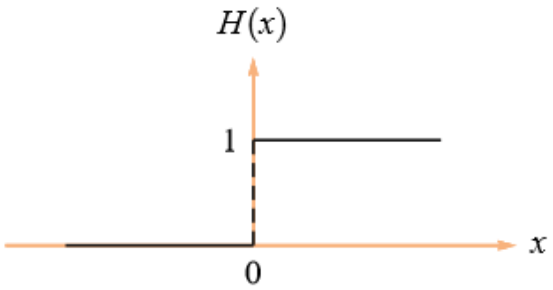
Uma outra notação utilizada para os pesos é a seguinte w_{kj}^{l}, onde l é o índice da camada na qual o neurônio está; k é o índice do neurônio na camada $l+1$ e k é o índice do neurônio na camada l (SVOZIL; KVASNICKA; POSPÍCHAL, 1997).

Quatro funções de ativação conhecidas são (HAYKIN, 2001):

- Função de limiar/Heaviside: foi a função das primeiras redes neurais apresentadas por McCulloch e Pitts (1943) e simboliza um modelo “tudo ou nada”. A representação gráfica da função de Heaviside pode ser vista na Figura 2, sendo matematicamente definida como:

$$H(x) = \begin{cases} 0 \ \ se\ x < 0 \\ 1\ se\ x > 0 \end{cases} \quad (1)$$

Figura 2 – Função de Heaviside

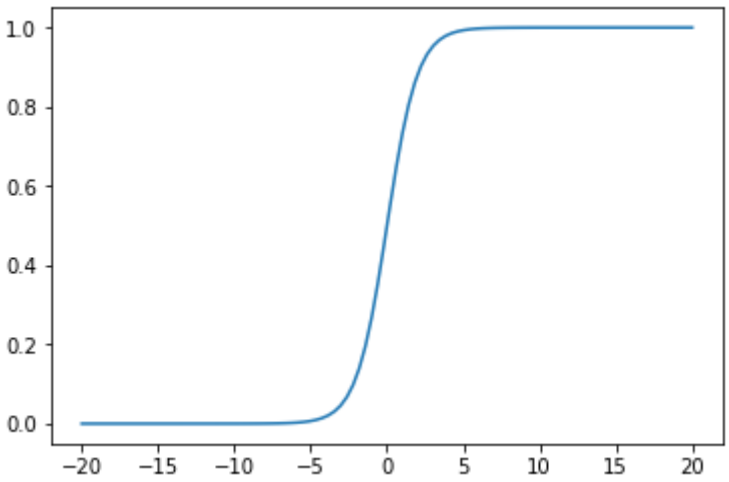


Fonte: SALIH (2015)

- A função sigmoide: apresenta-se como uma função intermediária entre os modelos lineares e não lineares e com valores sempre positivos. A representação gráfica da função sigmoide pode ser vista na Figura 3, sendo matematicamente descrita como:

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

Figura 3 – Função Sigmoide

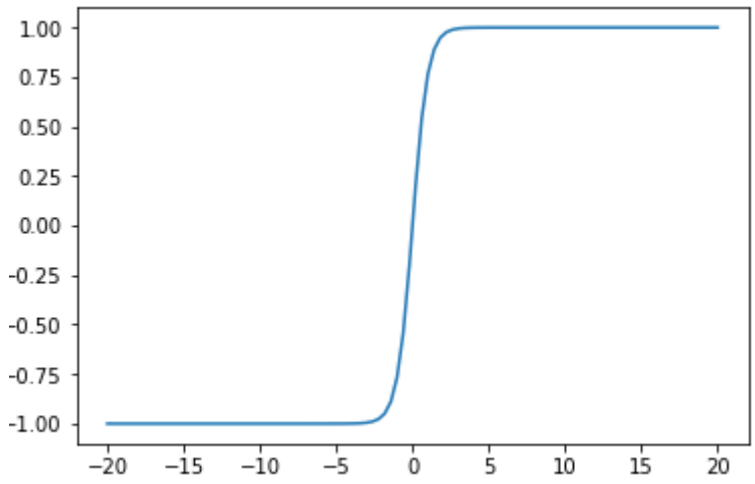


Fonte: Próprio Autor

- A função tangente hiperbólica (Tanh) preserva a forma da função sigmoide ao mesmo tempo que permite um intervalo de valores de saída maior. A representação gráfica da função se encontra na Figura 4:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (3)$$

Figura 4 – Função tangente hiperbólica

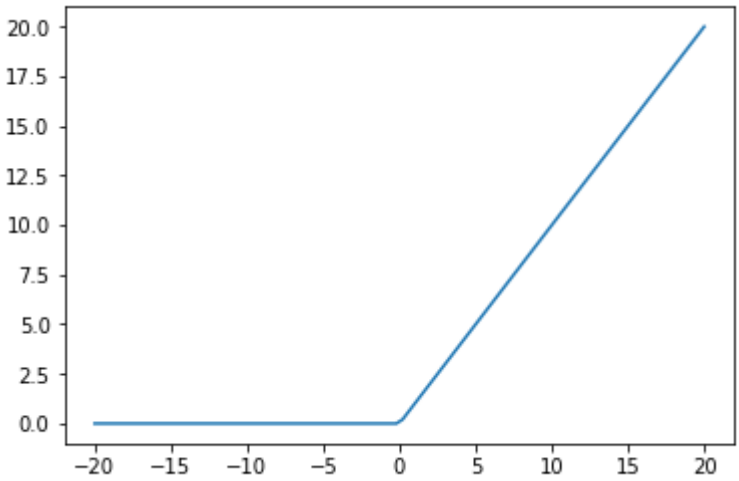


Fonte: Próprio Autor

- A função ReLU (*Rectified Linear Unit*) é uma função que mapeia a identidade no intervalo positivo das abscissas e 0 no intervalo negativo:

$$ReLU(x) = \max(0, x) \quad (4)$$

Figura 5 – Função *ReLU*



Fonte: Próprio Autor

Tem-se também a adição de um termo *b*, conhecido como *bias*, que permite uma maior flexibilidade para o neurônio para o ajuste dos dados. Logo, um neurônio que recebe

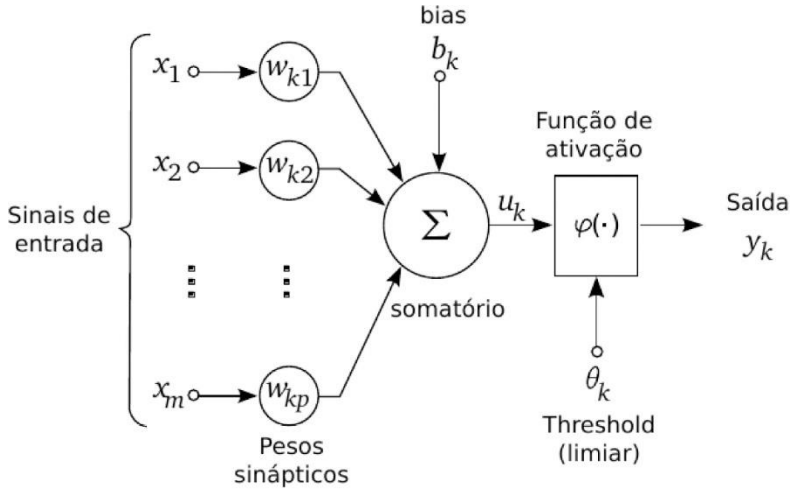
uma entrada x_j pode ser descrito, de acordo com HAYKIN (2001), de acordo com a Equação 5 e 6 abaixo, onde $f(u_k)$ simboliza uma função de ativação qualquer:

$$\widehat{y_k} = f(u_k) \quad (5)$$

$$u_k = \sum_{j=1}^{m}(w_{kj} * x_j + b_k) \quad (6)$$

Na Figura 6 está o fluxograma de uma rede neural idealizado por McCulloch e Pitts, que foram os primeiros a idealizarem e projetarem as redes neurais.

Figura 6 – Modelo de neurônio de McCulloch e Pitts



Fonte: HAYKIN (2001)

O processo de cálculo de uma rede neural inicia-se com as variáveis de entrada no começo da rede neural. Os cálculos da Equação 6 são feitos camada a camada, começando da esquerda e indo até a direita, neurônio a neurônio, até que os neurônios da última camada computem a saída da rede neural. Esse processo é conhecido como alimentação positiva ou *feed-forward*.

2.1.3 Algoritmos de treinamento e métodos de otimização

O treinamento das redes neurais pode ser dividido em duas categorias (BRAGA; CARVALHO; LUDERMIR, 2000):

- Supervisionado: são modelos de rede neural nos quais a variável de saída para um determinado conjunto de dados de entrada é previamente fornecida;
- Não supervisionado: modelos onde uma correlação entre os dados de entrada é buscada pela rede neural sem valores de saída previamente conhecidos.

Após a realização dos cálculos de *feed-forward* pela rede neural parte-se para o processo de verificação da precisão da rede. A precisão obtida no primeiro cálculo da rede

neural é comumente baixa, fazendo-se necessário a modificação dos únicos parâmetros variáveis da rede: a matriz de pesos e os *bias* associados a cada neurônio.

O procedimento mais comum utilizado nesse ajuste dos parâmetros da rede neural é o método de *Backpropagation* (BP). De acordo com BRAGA; CARVALHO; LUDERMIR (2000) o método de *Backpropagation* é uma generalização do método dos mínimos quadrados, fazendo neste caso uso do erro quadrado médio como indicador de performance.

Esse método, sendo classificado como um método iterativo, utiliza os valores designados para as matrizes de pesos e de *bias* na iteração anterior para definir os novos valores desses parâmetros em uma próxima iteração. Esses novos valores são calculados de forma que os pesos e bias convirjam para pontos de mínimo da função erro (Equação 7) (SVOZIL; KVASNICKA; POSPÍCHAL, 1997).

$$E = \frac{1}{2} * (y - \hat{y})^2 \tag{7}$$

Na Equação 7 o termo y é o valor da variável de saída para um dado conjunto de dados de treino e $\hat{y}$ é o valor predito pela rede para o mesmo conjunto de dados de treino.

A busca do ponto de mínimo da função de erro é feita através de métodos de otimização que utilizam o vetor gradiente do erro da saída em função da matriz de pesos e *bias* da rede. Dentre os métodos de otimização, podemos citar o gradiente descendente e métodos mais recentes como o *Adam* e *RPROP*.

As Equações 8 até 11 representam a dedução do algoritmo de *Backpropagation* fazendo uso do algoritmo de otimização do gradiente descendente. A notação das equações é (SVOZIL; KVASNICKA; POSPÍCHAL, 1997):

- w_{kj}^{i} representa o peso de uma conexão entre dois neurônios *k* e *j*;
- *α* simboliza o parâmetro denominado taxa de aprendizado;
- E é a função erro médio quadrado;
- b_{kj}^{i} é o *bias* da camada oculta *i;*
- $\widehat{y_k}$ é a variável de saída da rede neural (Equação 6);
- u_k é a somatória das entradas em um neurônio multiplicadas pelos respectivos pesos com o *bias* da camada *k*.

$$w_{kj}^{i+1} = w_{kj}^{i} - \alpha * \left(\frac{\partial E}{\partial w_{kj}}\right)^{(i)} \tag{8}$$

$$b_k^{i+1} = b_k^i - \alpha * (\frac{\partial E}{\partial b_k})^{(i)} \quad (9)$$

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial \widehat{y_k}} * \frac{\partial \widehat{y_k}}{\partial w_{kj}} = \frac{\partial E}{\partial \widehat{y_k}} * \frac{\partial f(u_k)}{\partial w_{kj}}$$

$$= \frac{\partial E}{\partial \widehat{y_k}} * \frac{\partial f(u_k)}{\partial u_k} * \frac{\partial u_k}{\partial w_{kj}}$$

$$= \frac{\partial E}{\partial \widehat{y_k}} * \frac{\partial f(u_k)}{\partial u_k} * \frac{\partial \sum_{j=1}^{m}(w_{kj} * x_j + b_k)}{\partial w_{kj}}$$

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial x_k} * \frac{\partial f(u_k)}{\partial u_k} * x_j \quad (10)$$

$$\frac{\partial E}{\partial b_k} = \frac{\partial E}{\partial \widehat{y_k}} * \frac{\partial \widehat{y_k}}{\partial b_k} = \frac{\partial E}{\partial \widehat{y_k}} * \frac{\partial f(u_k)}{\partial u_k} \quad (11)$$

Uma vez que a rede neural realiza os cálculos de *Backpropagation* em um conjunto de dados, processo denominado época, os valores da matriz de pesos e *bias* são atualizados para que então a rede realize uma nova passagem dos dados pela rede.

Os novos valores obtidos são então comparados com os valores esperados. Se esses novos valores não tiverem atingido um erro mínimo esperado, o algoritmo é executado novamente (SVOZIL; KVASNICKA; POSPÍCHAL, 1997).

Para que um modelo tenha um desempenho adequado deve-se alimentar o modelo com um conjunto de dados representativo (costumeiramente utiliza-se cerca de 60% a 70% do conjunto de dados total). Esse conjunto garantiria que o modelo funcione bem tanto para casos de interpolação quanto para casos de extrapolação, sendo o último o caso de maior dificuldade de tratamento (SVOZIL; KVASNICKA; POSPÍCHAL, 1997).

O processo de *Backpropagation* pode se dar de duas formas (BRAGA; CARVALHO; LUDERMIR, 2000):

- Local ou *on-line*: a atualização dos pesos e *bias* é feita após a apresentação de cada conjunto de dados à rede. Apresenta-se à rede um conjunto de dados, a rede neural faz os cálculos de *feed-forward* e *backpropagation* e atualiza prontamente os pesos antes de realizar o cálculo em cima de um segundo conjunto de dados de treinamento.

  Este método requer menos memória para os cálculos uma vez que os pesos são atualizados a cada época e evita que o algoritmo pare em mínimos locais;

- Em lote ou *off-line*: a atualização dos pesos e *bias* é feita após a apresentação de todos os conjuntos de dados à rede neural.
  Este método é preferível no sentido de que oferece uma melhor estimativa do vetor gradiente, já que possui um espaço amostral maior do que no método local.

Um método misto entre os métodos local e em lote é o método dos mini lotes, que consiste em fazer a atualização dos pesos após a época de *k* conjunto de dados, sendo *k* menor que o número total de conjunto de dados disponível para treino de uma rede neural. Neste caso a atualização dos pesos é uma média entre os valores de atualização obtidos para cada conjunto de dados pertencentes a *k* (PARK *et al*., 2018).

Outros algoritmos de otimização que receberam atenção devido ao pequeno número de épocas necessário para convergência da função erro, mas que também tem recebido certa crítica devido à seus ganhos reais são (WILSON *et al*., 2017):

- *Adam* (*Adaptive moment estimation*): ao contrário do método do gradiente descendente, o parâmetro *α* varia ao longo do processo de aprendizado da rede. A cada época de treinamento registra-se o valor da função erro e seu gradiente (g_t), os pesos então são atualizados através da média móvel do gradiente (m_t) e do quadrado do gradiente (v_t) que tem seus valores controlados por dois hiper parâmetros, β_1, $\beta_2 \in [0,1)$ e pela época t (KINGMA; BA, 2014).

$$m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * g_t \quad (12)$$

$$v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * g_t^2 \quad (13)$$

Mas como m_t e v_t são estimadores enviesados, tem-se um fator de correção como demonstrado nas Equações 14 e 15, onde t é a época.

$$\widehat{m} = \frac{m_t}{1-\beta_1^t} \quad (14)$$

$$\hat{v} = \frac{v_t}{1-\beta_2^t} \quad (15)$$

Sendo os pesos atualizados pela Equação 16 abaixo, onde θ é a matriz de parâmetros da rede (pesos e *bias*) e ϵ um hiper parâmetro do algoritmo com a finalidade de evitar que a taxa de aprendizado aumente bruscamente em casos onde $\widehat{v_t}$ é próximo de zero:

$$\theta_t = \theta_{t-1} - \alpha * \widehat{m}/(\sqrt{\widehat{v_t}} + \epsilon) \quad (16)$$

- *RMSprop*: também pertencente ao grupo de algoritmos que não possuem a taxa de aprendizado constante. Foi descrito por LYON (2017) e, assim como *Adam* comporta-se como uma média móvel exponencial, tendo como diferença a forma como a matriz de parâmetros da rede é atualizada, como demonstrado nas Equações 17 e 18.

$$E[g_t^2] = \beta * E[g_{t-1}^2] + (1-\beta) * [g_t^2] \quad (17)$$

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{E[g_t^2]}} * g_t \quad (18)$$

A diferença entre os dois métodos reside no fato de que o algoritmo *Adam* tende ser mais estável uma vez que o gradiente atinge uma região de mínimo (HEUSEL *et al*., 2017).

2.1.4 Tamanho de redes neurais

O desempenho de uma rede neural é dependente do número de neurônios dispostos em suas camadas ocultas. Um excesso de neurônios na rede pode causar um problema conhecido como sobre ajuste, que consiste no ajuste muito próximo da rede aos dados de treino utilizados, o que implica em uma baixa capacidade de generalização da rede. Na contramão desse problema tem-se o baixo desempenho da rede no aprendizado quando se faz uso de poucos neurônios (PRIETO *et al*., 2016).

Quanto às camadas ocultas, um aumento no número das camadas tem dois efeitos negativos mais pronunciados sobre a performance das redes neurais. O primeiro é a dificuldade para encontrar o melhor mínimo local da função erro através do método de otimização escolhido (Equação 7) e o segundo é a instabilidade causada no termo vetor gradiente tornando o processo de treinamento mais lento (SVOZIL; KVASNICKA; POSPÍCHAL, 1997).

Entretanto, um número razoavelmente grande de camadas ocultas (DNN) possibilitou o desenvolvimento de redes neurais para tarefas mais complexas, com destaque especial para a classificação de imagens e de texto. Além da aplicação em tarefas complexas, modelos de DNNs tem obtido maior sucesso em tarefas antes realizadas por modelos que já obtinham boas performances, como predições, cálculo de energia de ativação de moléculas, entre outros (LECUN; BENGIO; HINTON, 2015).

A presença de ruídos nos dados pode aumentar o problema de sobre ajuste de uma rede neural (SVOZIL; KVASNICKA; POSPÍCHAL, 1997). Uma abordagem para o

tratamento de ruídos foi feita por SIQUEIRA *et al*. (2018), onde utilizou-se o próprio ruído dos dados como variável de entrada do sistema estudado.

O número de neurônios e o número de camadas ocultas são perguntas iniciais no desenvolvimento de uma rede neural. Métodos desenvolvidos para solucionar o problema do tamanho ideal da rede podem ser divididos em dois tipos (HAYKIN, 2001):

- Crescimento de rede: parte-se uma rede neural com neurônios dispostos em uma ou mais camada(s) oculta(s) e adiciona-se ramificações gradativamente de modo a fazer com que alcance um desempenho satisfatório;
- Método de poda: parte-se de uma rede neural com um número elevado de neurônios que já possui um desempenho satisfatório e neurônios são eliminados gradativamente mantendo o desempenho da rede acima de um limiar desejado.

2.2 NARIZ ELETRÔNICO

Narizes eletrônicos são dispositivos que tentam mimetizar o funcionamento do nariz humano na tarefa de reconhecimento de aromas. Zwaardemaker e Hogewind foram os primeiros a estudar o comportamento elétrico de voláteis. Partindo de medidas feitas sob um fino spray de água, descobriram que substâncias voláteis causavam alterações nas propriedades elétricas do spray de água tornando possível a detecção de componentes voláteis através de métodos sensoriais não clássicos (WILSON; BAIETTO, 2011).

O nariz humano realiza a tarefa de identificação com o auxílio de cerca de 390 receptores olfativos que se conectam a moléculas voláteis causando modificações estruturais nos receptores. Essas modificações fazem com que sinais sejam enviados ao cérebro para identificação (ZHANG *et al*., 2018).

Já o nariz eletrônico faz o processo de identificação através de uma série de sensores que apresentam respostas que dependem das moléculas adsorvidas. Essas respostas são geralmente medidas em função da variação de uma propriedade física do sensor (GHASEMI-VARNAMKHASTI *et al*., 2018).

Os sensores utilizados para identificação são escolhidos de acordo com a aplicação, de forma que o sensor escolhido seja sensível à maior parte possível das substâncias voláteis a serem identificadas. Nesse sentido uma baixa seletividade dos sensores é desejada, permitindo que uma mesma amostra seja detectada por mais de um sensor, incrementando o sinal de determinada substância (WILSON; BAIETTO, 2011).

Após a adsorção da molécula de odor nos receptores do nariz eletrônico, o sinal elétrico gerado é identificado e gravado em um banco de dados para servir como posterior referência para novas análises (ZHANG *et al*., 2018).

Uma seletividade muito baixa de um sensor pode resultar em sinais elétricos para uma quantidade muito grande de moléculas, aumentando o ruído na resposta. Métodos para tratamento desse ruído têm sido desenvolvidos com auxílio de métodos estatísticos como o LDA (*Linear Discriminant Analysis*) e as redes neurais artificiais (ZHANG *et al*., 2018).

Pelo fato de se utilizar de mecanismos químicos e não biológicos na identificação de moléculas, o nariz eletrônico apresenta algumas desvantagens perante o nariz humano como (ZHANG *et al*., 2018):

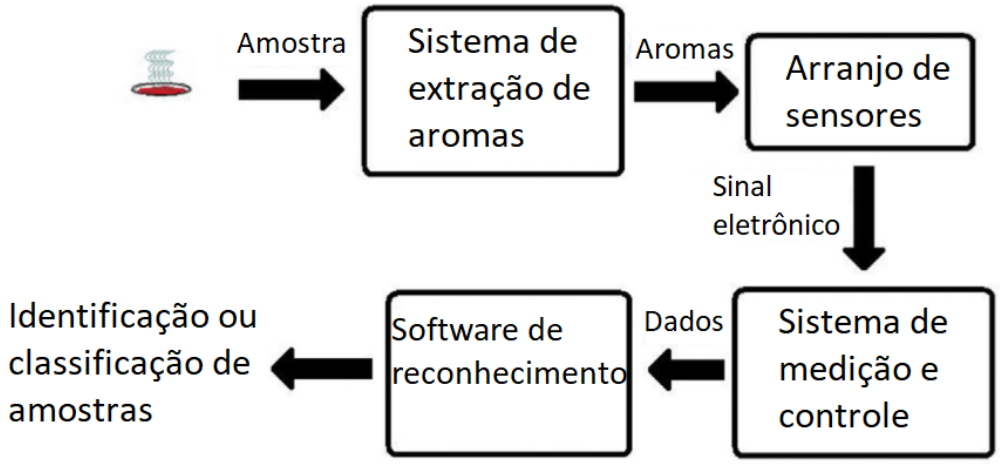
- não detecção de substâncias a níveis abaixo de pbm (partes por bilhão);
- dificuldades em separar misturas.

No que tange às vantagens do nariz eletrônico frente ao nariz humano temos (MAJCHRZAK *et al*., 2018):

- Maior reprodutibilidade de pareceres sobre aromas;
- Análises mais rápidas e menos custosas;
- Ausência de vieses humanos como fadiga ou acomodação à odores.

Um modelo esquemático do processo desde a exibição da amostra até a identificação da substância em questão é apresentado na Figura 7:

Figura 7 – Diagrama de bloco de um nariz eletrônico



Fonte: Adaptada de GHASEMI-VARNAMKHASTI *et al*., (2018)

O desenvolvimento no ramo dos narizes eletrônicos tem focado especialmente na estabilização dos sinais lidos. Soluções como o uso de espectrômetros de massa em narizes eletrônicos têm simbolizado um avanço nesse sentido, uma vez que possuem sensores dotados de maior seletividade (MAJCHRZAK *et al*., 2018).

Dentre as classificações existentes para os narizes eletrônicos temos como principal critério o tipo de sistema de detecção empregado pelo aparelho (MAJCHRZAK *et al*., 2018):

- Semicondutores: são modelos mais econômicos, mas apresentam baixa estabilidade do sinal de saída;
- Sensores eletroquímicos: apresentam um sinal mais estável quando comparado com os semicondutores, especialmente por reduzir interferências como da umidade. São geralmente maiores, o que pode se tornar uma barreira para algumas aplicações;
- Sensores piezoelétricos: são compactos, o que facilita o uso em trabalhos de campo; no entanto, possuem medidas que são replicadas com dificuldade;
- Espectrômetros de massa: possuem alta sensibilidade e um vasto leque de aplicações, porém são mais caros que os outros mencionados até aqui;
- Cromatografia gasosa: permite análises quantitativas e qualitativas, mas demandam um tempo maior para análise das amostras.

A combinação de sensores pode melhorar as análises realizadas com o auxílio do nariz eletrônico, já que cada sensor apresenta uma seletividade específica, aumentando dessa forma a quantidade de informação para detecção de substâncias. Deve-se, no entanto, atentar-se ao número de dimensões, relacionado com o número de sensores,

gerado por análise quando do uso de sensores variados. Um excesso no número de dimensões pode acarretar em um ruído que prejudica uma avaliação mais criteriosa do equipamento (RÖCK; BARSAN; WEIMAR, 2008).

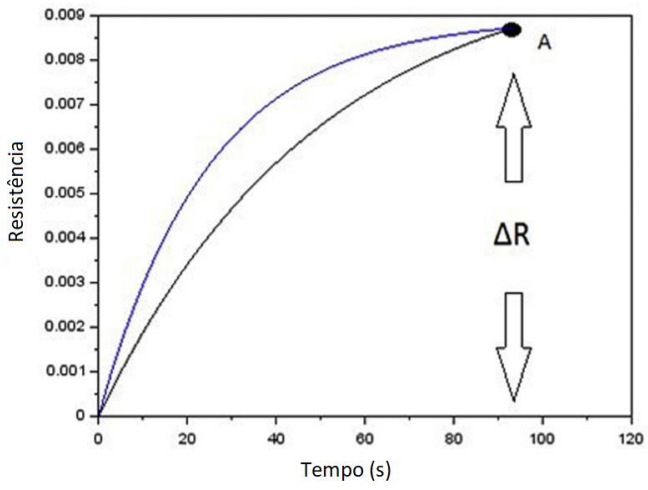
Dentre algumas limitações enfrentadas na identificação de substâncias com narizes eletrônicos tem-se a diminuição no desempenho de alguns modelos quando se faz uso de um grande número de amostras. Essa diminuição, como demonstrado por BOEKER (2014), é devida ao aumento na variabilidade dos dados de entrada.

A variabilidade apresentada nos sinais de um nariz eletrônico depende de vários fatores, em especial da substância em si. A utilização da variabilidade como uma variável de auxílio na identificação de substâncias foi realizado por SIQUEIRA *et al*. (2018) por meio da modelagem do sinal do nariz eletrônico através de uma equação estocástica diferencial.

A melhora na performance quando do uso da variabilidade em problemas de identificação foi reportado por DUTTA *et al*. (2006). Em seu trabalho, constatou-se que um melhor desempenho na identificação de bactérias é obtido quando da adição de um ruído gaussiano artificial ao sinal lido. À essa melhora na identificação quando da adição de um ruído gaussiano em fenômenos não lineares dá-se o nome de ressonância estocástica.

Um exemplo do efeito da vantagem quando do uso de equações estocásticas diferenciais na identificação de compostos através de sinais de um nariz eletrônico pode ser visto na Figura 8:

Figura 8 – Sinal de um sensor para duas substâncias diferentes com mesmo ΔR



Fonte: Adaptada de SIQUEIRA *et al*. (2018)

No processo de identificação, A, medido por ΔR, é o platô comumente utilizado para identificação de uma substância com o uso de um nariz eletrônico. As curvas da Figura 8

são de substâncias diferentes e apesar disso apresentam o mesmo ΔR, o que pode levar a conclusões errôneas sobre a substância em questão.

Os percentuais de amostras classificadas corretamente obtidos por SIQUEIRA *et al*. (2018) com a aplicação da equação estocástica diferencial são mostradas abaixo. Os valores foram obtidos utilizando-se três sensores dentre os 32 disponíveis no nariz eletrônico utilizado. Esses três sensores foram escolhidos aleatoriamente dentre os que mais apresentaram um comportamento típico ao longo das medidas:

- Com redes neurais: 67%;
- Com PCA (utilizando apenas o primeiro componente principal): 58%;
- Com LDA :63,6%;
- Modelo estocástico: 91,6%.

A alta performance obtida pelo modelo estocástico é devida ao fato de o modelo utilizar-se de cinco parâmetros e da própria variabilidade dos dados no processo de identificação de substâncias. O modelo estocástico é, portanto, uma alternativa considerável para caracterização de substâncias não identificáveis por métodos convencionais (SIQUEIRA *et al*., 2018).

3 MATERIAIS E MÉTODOS

3.1 NARIZ ELETRÔNICO

O presente trabalho foi executado com auxílio de dados coletados via nariz eletrônico Cyrano Sciences' Cyranose 320 no Laboratório de Biocatálise – EEL/USP. O nariz eletrônico Cyranose 320 possui 32 sensores a base de polímeros e nanopartículas de carbono em série.

3.1.1 Tempo de exposição

O tempo durante o qual o nariz ficou exposto às amostras foi curto o suficiente (15 segundos), de modo a garantir uma concentração constante de substâncias voláteis ao longo das leituras. O tempo de medição é o período no qual a agulha ficou inserida dentro do frasco até sua retirada (SIQUEIRA *et al*., 2018).

Os dados utilizados como base nesse trabalho foram coletados e pré-tratados por SIQUEIRA *et al*. (2018) como discutido em 3.1.2 Pré-tratamento dos dados.

3.1.2 Pré-tratamento dos dados

Pelo fato de o tempo de sucção do material proposto no modelo ser menor do que o tempo de remoção, os dados obtidos no final do período de sucção são descartados. Esse pré-tratamento nos dados é uma etapa importante no cálculo dos parâmetros da Equação 27 (SIQUEIRA *et al*., 2018).

Na metodologia utilizada por SIQUEIRA *et al*. (2018), apenas a fase de adsorção foi modelada e a linha de base do sinal foi separada da etapa de leitura do sinal através da adsorção de elementos voláteis no início da medição. O critério de rejeição de dados foi com o auxílio do coeficiente de correlação linear (R^2) calculado entre os dados lidos e os valores provenientes da Equação 27.

3.1.3 Modelagem do sinal do nariz eletrônico

A equação estocástica diferencial escrita por SIQUEIRA *et al*. (2018) é apresentada abaixo. Ela foi desenvolvida a partir dos mecanismos de adsorção e dessorção de sensores propostas por LUNDSTRÖM (1996).

O mecanismo proposto por LUNDSTRÖM (1996) tem as seguintes premissas como verdadeiras:

1. As velocidades de adsorção e dessorção são obtidas através de modelos cinéticos de ordem 1;

2. O sinal (*S*) lido pelo nariz eletrônico é proporcional ao número de substâncias voláteis ligadas aos sítios ativos (*n*) dividido pelo número total de sítios ativos (*N*). Logo o sinal (*S*) é *S=n/N*;
3. Considera-se a concentração da substância volátil constante ao longo de todo o processo de medição.

As premissas 1-3 resultam na Equação 19, onde k e b são parâmetros obtidos por meio do ajuste do modelo aos sinais de entrada e do tempo de exposição.

$$S(t) = b * (1 - e^{-k*t}) \quad (19)$$

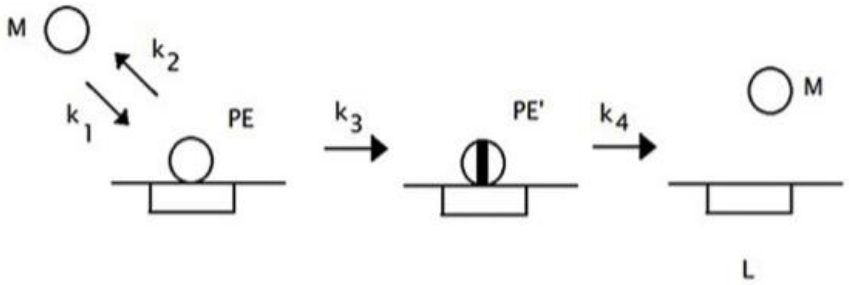
Após um longo período de tempo, o sistema atinge o estado estacionário e o valor da Equação 19 atinge um platô horizontal de valor b.

Uma quarta premissa foi adotada no desenvolvimento devido à identificação de inclinações positivas ou negativas ao invés do platô (SIQUEIRA *et al*., 2018):

4. Criação de um estado precursor (*PE*) antes do estado gerador de sinal (*PE'*).

Portanto, nos dados utilizados nesse trabalho, o sinal *S(t)* é uma função do número de estados precursores ocupados (SIQUEIRA *et al*., 2018). Um esquema para o mecanismo de adsorção e dessorção proposto na Figura 9:

Figura 9 – Modelo para o mecanismo de adsorção e dessorção



Fonte: SIQUEIRA *et al*. (2018)

As premissas de 1 - 4 foram então aplicadas na modelagem do mecanismo de adsorção e dessorção, onde L é a fração de sensores não ocupados (SIQUEIRA *et al*., 2018):

$$L + PE = 1 \quad (20)$$

Assumiu-se que os estados precursores podem ser classificados entre os que já geraram (*PE*) e os que ainda irão gerar sinal (*FPE*) (SIQUEIRA *et al*., 2018):

$$PE = FPE + PE' \quad (21)$$

A partir da Figura 9 obteve-se a Equação diferencial 22, onde k_e é a constante de equilíbrio entre o processo de adsorção e dessorção. Na Equação 21 assumiu-se uma reação de equilíbrio rápida para a criação do estado precursor (*PE)* (SIQUEIRA *et al*., 2018):

$$\frac{dPE}{dt} = k_1 * M * L - k_2 * PE = 0 \rightarrow \frac{k_2}{k_1} = k_e = \frac{M*L}{PE} \quad (22)$$

Isolando *L* e substituindo na Equação 20, tem-se:

$$PE = \frac{1}{1+\frac{k_e}{M}} \quad (23)$$

Com auxílio da Figura 9, deduziu-se a Equação diferencial 16 para o estado gerador de sinal (SIQUEIRA *et al*., 2018):

$$\frac{dPE'}{dt} = k_3 * PE - k_4 * PE' = \frac{k_3}{1+\frac{k_e}{M}} - k_4 * PE' \quad (24)$$

Resolvendo a Equação diferencial 23 para *PE' (0)* =0*:*

$$PE' = \frac{k_3}{k_4*(1+\frac{k_e}{M})} * (1 - e^{-k_4*t}) \quad (25)$$

Por fim, assumiu-se que a taxa de variação do sinal é proporcional à ocupação de sítios livres do estado precursor (SIQUEIRA *et al*., 2018):

$$\frac{dS}{dt} = k_s * FPE = k_s * (PE - PE')$$

$$\frac{dS}{dt} = k_s * \left(\frac{1}{1+\frac{k_e}{M}} - \frac{k_3}{k_4*\left(1+\frac{k_e}{M}\right)} + \frac{k_3*e^{-k_4*t}}{k_4*\left(1+\frac{k_e}{M}\right)} \right) \quad (26)$$

Resolvendo-se a Equação 26 para *S(0)* = 0, tem-se:

$$S(t) = \frac{k_s}{\left(1+\frac{k_e}{M}\right)} * \left(1 - \frac{k_3}{k_4}\right) * t + \frac{k_3*k_s}{(k_4)^2*\left(1+\frac{k_e}{M}\right)} * (1 - e^{-k_4*t}) \quad (27)$$

Denominando o termo que multiplica *t* por *a,* o que multiplica $(1 - e^{-k_4*t})$ por *b* e k_4= *k,* tem-se:

$$S(t) = a * t + b * (1 - e^{-k*t}) \quad (28)$$

Uma modelagem também foi proposta para o período de purga do sistema, momento onde ocorre a remoção das substâncias voláteis do sensor em um tempo *Tr* a partir do início da medição. Para fins de cálculo, transladou-se o tempo para *t=t'-Tr*, onde

t' é o tempo transcorrido desde o início do processo de medição (SIQUEIRA *et al*., 2018). Na escala de tempo *t* os valores de *S, PE, PE'* e *FPE* tornam-se:

S(0) =Sr, PE(0) =PEr, PE'(0) =PE'r, FPE(0) =FPEr

Modelou-se então a variação de *PE* de acordo com as condições acima e através da Equação 29 (SIQUEIRA *et al*., 2018):

$$\frac{dPE}{dt} = -(k_2 + k_3) * PE \tag{29}$$

Sendo a solução da Equação 29 dada pela Equação 30:

$$PE(t) = PEr * e^{-(k_2+k_3)*t} \tag{30}$$

A variação de *PE*' foi modelada de acordo com a Equação 31 e considerando-se a condição inicial *PE'(0)* = *PE'r* (SIQUEIRA *et al*., 2018):

$$\frac{dPE'}{dt} = k_3 * PE - k_4 * PE' \tag{31}$$

A solução da Equação 31 é dada pela Equação 32:

$$PE'(t) = PE'r * e^{-(k_4)*t} + \frac{k_3*PEr*e^{-(k_2+k_3)*t}}{k_4-(k_2+k_3)} - \frac{k_3*PEr*e^{-k_4*t}}{k_4-(k_2+k_3)} \tag{32}$$

E a variação do sinal *S* foi modelada pela Equação 33 (SIQUEIRA *et al*., 2018):

$$S(t) = Sr + D * \left(1 - e^{-(k_2+k_3)*t}\right) - E * (1 - e^{-k_4*t}) \tag{33}$$

Onde *D* e *E* foram definidos como:

$$D = \frac{k_s * PEr * (k_2 + k_3 - k_4 + k_3)}{(k_2 + k_3 - k_4) * (k_2 + k_3)}, \qquad E = \left(PE'r + \frac{k_3 * PEr}{k_2 + k_3 - k_4}\right) * \frac{k_s}{k_4}$$

Portanto a modelagem do sinal lido por um nariz eletrônico, considerando a purga pode ser descrito por (SIQUEIRA *et al*., 2018):

$$S(t) = \begin{cases} a * t + b * e^{-(k_2+k_3)*t}, & se\ t < Tr \\ Sr + D * \left(1 - e^{-(k_2+k_3)*(t-Tr)}\right) - E * \left(1 - e^{-k_4*(t-Tr)}\right), & se\ t > Tr \end{cases} \tag{34}$$

Na Equação 34, o termo *Sr* é dado por:

$$Sr = a * Tr + b * (1 - e^{-k*t_r}) \tag{35}$$

A aplicação de um modelo estocástico no sinal de um nariz eletrônico foi devida ao fato de que a variabilidade apresentada em leituras com nariz eletrônico depende da

própria substância sob análise. Logo, por depender da própria substância, acredita-se que essa variabilidade possa auxiliar na identificação da substância (SIQUEIRA *et al*., 2018).

A equação estocástica diferencial utilizada nesse trabalho para modelagem do sinal do nariz eletrônico foi proposta por (SIQUEIRA *et al*., 2013). Na Equação 36, a, b, c, k e p são os parâmetros que dependem do sinal medido e X_t é a medida da variabilidade do sinal em um tempo qualquer *t* em minutos.

$$dX_t = \left(a + \frac{b*k}{e^{k*t}}\right) * dt + \frac{c}{(t+1)^p} * dW_t \quad (36)$$

O algoritmo utilizado para a obtenção dos parâmetros da Equação 28 é baseado numa estimativa inicial do valor de k e foi descrito por SIQUEIRA *et al*. (2013) tendo como base técnicas para estimativas de equações diferenciais estocásticas e de amostragem (SIQUEIRA *et al*., 2018).

Uma variação apresentada na metodologia descrita por SIQUEIRA *et al*. (2013) está no parâmetro *c*:

$$c = \frac{1}{8} * \sqrt{\frac{\sum Q_i}{\sum \frac{1}{(t_p+1)^p}}} \quad (37)$$

Sendo Q_i calculado através da Equação 38:

$$Q_i = \frac{(\Delta X_i)^2}{\Delta t_i} \quad (38)$$

3.2 ÍNDICE DE ACIDEZ

Os óleos vegetais utilizados como amostras para o nariz eletrônico foram caracterizados quanto ao teor de acidez, valor de peróxido, viscosidade, densidade e cor. As amostras foram coletadas na cidade de Lorena-SP de duas fontes diferentes: residencial e comercial em um total de 4 vezes a cada 20 dias visando obter as amostras em diferentes períodos (SIQUEIRA *et al*., 2018).

As amostras foram filtradas à vácuo, homogeneizadas e estabilizadas em cerca de 23ºC antes de serem submetidas ao nariz eletrônico. Cada uma das 12 amostras foi introduzida em 10 frascos de 45 ml de capacidade com uma capa de borracha, totalizando 120 frascos, sendo 60 de cada fonte, residencial ou comercial. Os frascos ficaram em repouso durante 12h à 23ºC para permitir o equilíbrio liquido vapor, sendo então a agulha

do nariz eletrônico introduzida na região do frasco acima da mistura. A medida de acidez foi realizada de acordo com as normas da *Association of Official Analytical Chemists* (AOAC) (SIQUEIRA *et al*., 2018).

3.3 REDES NEURAIS ARTIFICIAIS

A rede neural foi escrita em linguagem Python, versão 3.7, utilizando a biblioteca Keras (CHOLLET, 2015) e a biblioteca *Tensorflow* (GOOGLE, 2015) como *backend e* tendo como variáveis de entrada os parâmetros do modelo estocástico e como variável de saída a acidez dos óleos vegetais.

3.3.1 Escolha das variáveis independentes

Os dados utilizados para treino da rede representaram 70% do total de dados da base original; e como cada um dos 12 óleos foi amostrado 10 vezes, garantiu-se que uma quantidade igual de leituras de cada óleo estivesse presente na base de treino.

A fim de identificar os sensores a serem utilizados na rede neural, fez-se um filtro utilizando o R^2 de cada sensor, que media o ajuste do modelo estocástico ao sinal do nariz eletrônico. Assim, sensores que não obtiveram pelo menos uma das 120 amostras com R^2 maior ou igual a 0,99 foram descartadas.

Como os parâmetros c e p da Equação 28 modelam a variância do sinal enquanto os outros três a, b e k modelam a média, utilizou-se apenas esses três últimos parâmetros na construção da rede.

Visando escolher o conjunto de variáveis a ser utilizado no modelo, foram construídas duas redes neurais, uma com função de ativação sigmoide e outra com função de ativação tangente hiperbólica.

Para cada uma delas, inicialmente modelou-se cada um dos parâmetros disponíveis por vez e registrou-se seu erro médio absoluto (MAE).

Escolhendo-se então o preditor com o menor valor de MAE de validação, modelou-se novamente uma rede neural para cada um dos parâmetros restantes, sendo que neste caso as variáveis de entrada eram o preditor escolhido anteriormente e cada um dos restantes. Prosseguiu-se dessa forma até que todos os parâmetros fossem adicionados à rede.

Dado o pouco volume de dados, utilizou-se nesta etapa validação cruzada com 3 partições para se garantir uma maior robustez dos dados obtidos.

A configuração dessa rede neural inicial, doravante mencionada como rede simples, foi:

- 1 camada oculta com 5 neurônios;
- Função de ativação identidade na saída;
- Algoritmo de otimização do método de *Backpropagation* sendo o gradiente estocástico descendente;
- Dados normalizados entre [0,1] para a rede com função de ativação sigmoide e dados convertidos em valores z para a rede com função de ativação tangente hiperbólica;
- 200 épocas.

Somou-se então a ordem em que cada variável foi adicionada as duas redes neurais a fim de gerar um ranking de importância de cada variável ao modelo.

O embasamento por trás de tal mecanismo para escolha de variáveis é o de que, mesmo utilizando redes neurais com diferentes funções de ativação, e consequentemente intervalos de dados normalizados diferentes, tais variáveis foram identificadas pela rede como as que mais aumentaram a capacidade preditiva da rede.

3.3.2 Ajuste dos parâmetros

Após a seleção de variáveis, converteu-se os valores de entrada e saída da rede para valores z e então verificou-se como alterações na função de ativação e no otimizador utilizado no método de *Backpropagation* influenciavam o R^2 de teste da rede, para que esses dois melhores parâmetros pudessem ser definidos.

As funções de ativação testadas neste trabalho foram: sigmoide, tangente hiperbólica e *ReLU*. Já os algoritmos de otimização foram: Gradiente Descendente Estocástico, *RMSprop e Adam*. Todos as análises utilizaram validação cruzada com 5 partições.

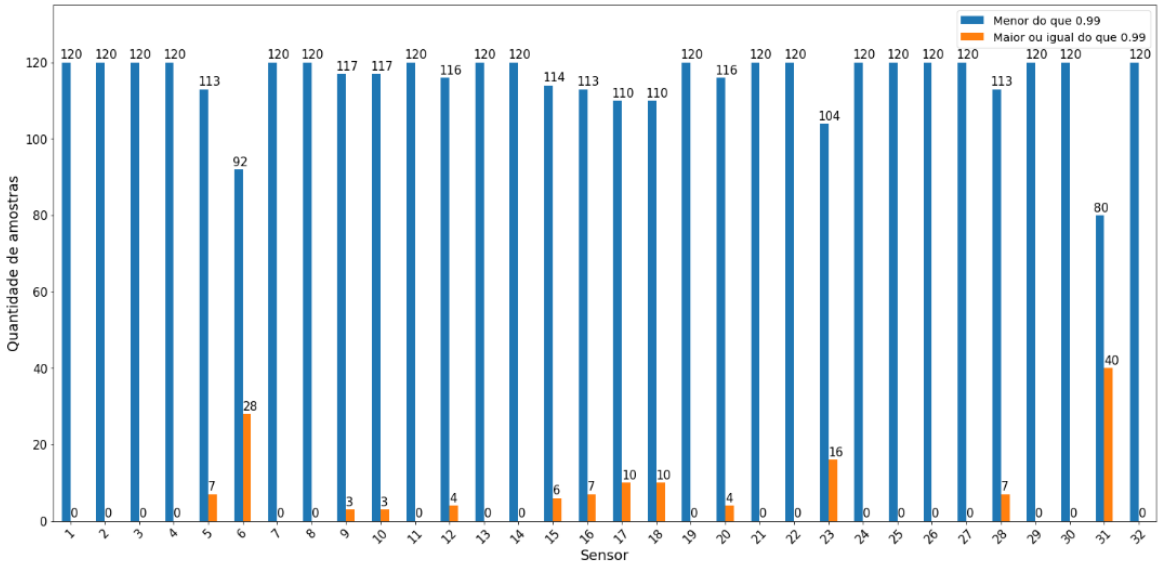
Partindo então do melhor conjunto de função de ativação e otimizador, variou-se o número de neurônios na rede e ponderou-se a necessidade da adição de camadas ocultas adicionais em caso de baixo desempenho da rede na predição.

4 RESULTADOS E DISCUSSÃO

4.1 ESCOLHA DE SENSORES

Para verificar quais sensores obtiveram pelo menos uma amostra com R^2 maior ou igual a 0,99, contou-se quantos deles respeitaram esse critério. O resultado é exibido na Figura 10 abaixo.

Figura 10 – Contagem do número de amostras por sensor segundo o critério R^2>=0,99

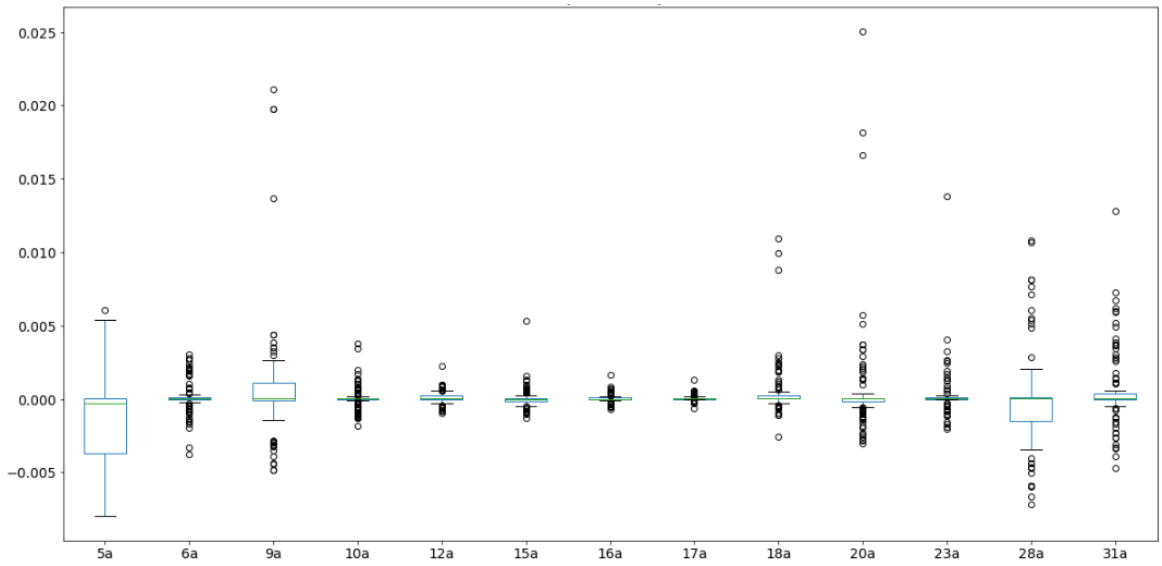


Fonte: Próprio Autor

Um total de 13 sensores passaram por esse critério de aderência dos dados ao modelo, sendo que o sensor 31 foi o que melhor se ajustou ao modelo, com um total de 40 amostras com um R^2 maior ou igual a 0,99. Seguido pelo sensor 6 com 28 amostras e pelo sensor 23 com 16.

Buscando analisar o comportamento dos parâmetros, fez-se um *boxplot* de cada parâmetro, como mostrado nas figuras abaixo para os parâmetros a, b e k a fim de observar como os valores de cada um dos parâmetros se comportam.

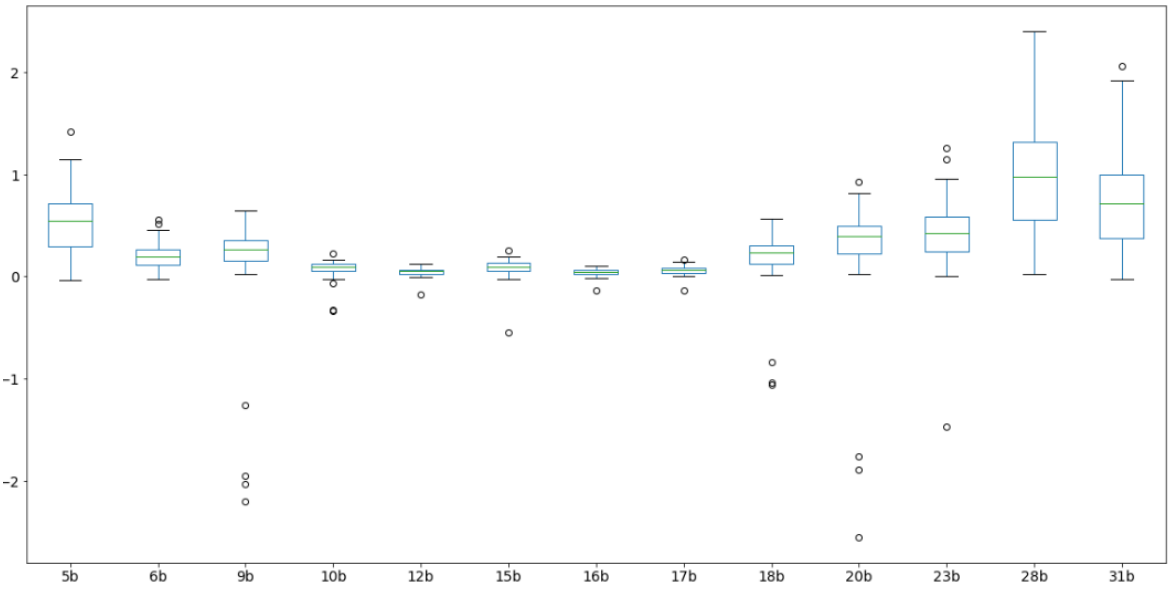
Figura 11 – *Boxplot* do parâmetro a para os sensores selecionados



Fonte: Próprio Autor

Pela análise da Figura 11 percebe-se que o parâmetro a obteve distribuições em sua grande maioria com muitos *outliers*, à exceção desse parâmetro para o sensor 5 que obteve apenas um *outlier*.

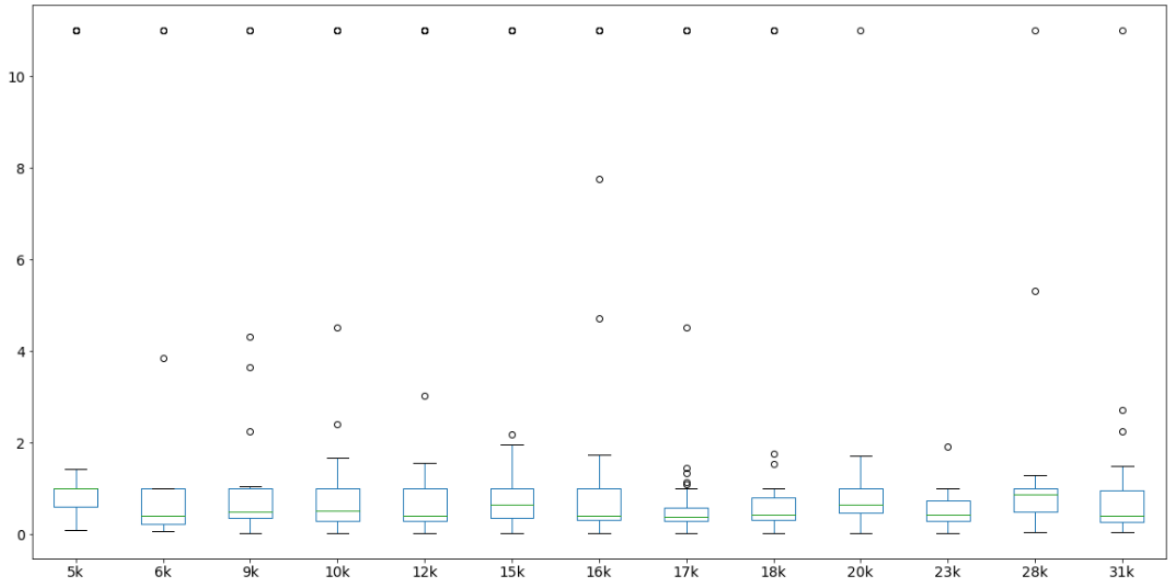
Figura 12 – *Boxplot* do parâmetr b o para os sensores selecionados



Fonte: Próprio Autor

Analisando-se então a Figura 12 para o parâmetro b quase todos os sensores obtiveram distribuições bem comportadas, em especial os obtidos no sensor 28, 5, 12, 16 e 31.

Figura 13 – *Boxplot* do parâmetro k para os sensores selecionados



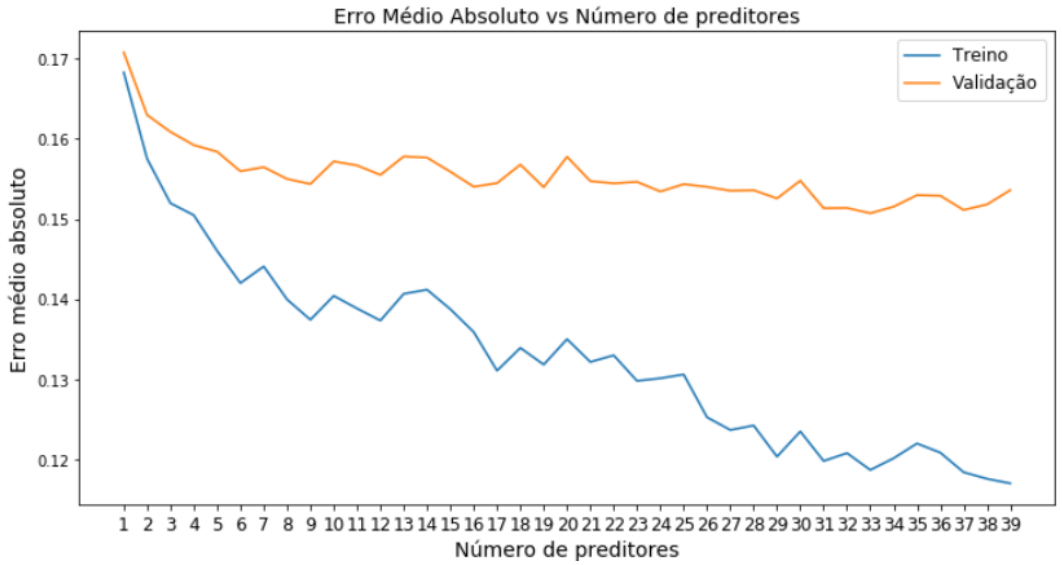
Fonte: Próprio Autor

E para o parâmetro k, analisando-se a figura acima, o número de outliers por sensor foi parecido com os obtidos para o parâmetro b, sendo os sensores mais comportados os de número 5, 20 e 23.

4.2 ESCOLHA DOS PARÂMETROS

Utilizou-se então a rede neural simples com função de ativação sigmoide e a abordagem descrita em 3.3.1. Os dados são exibidos na Figura 14.

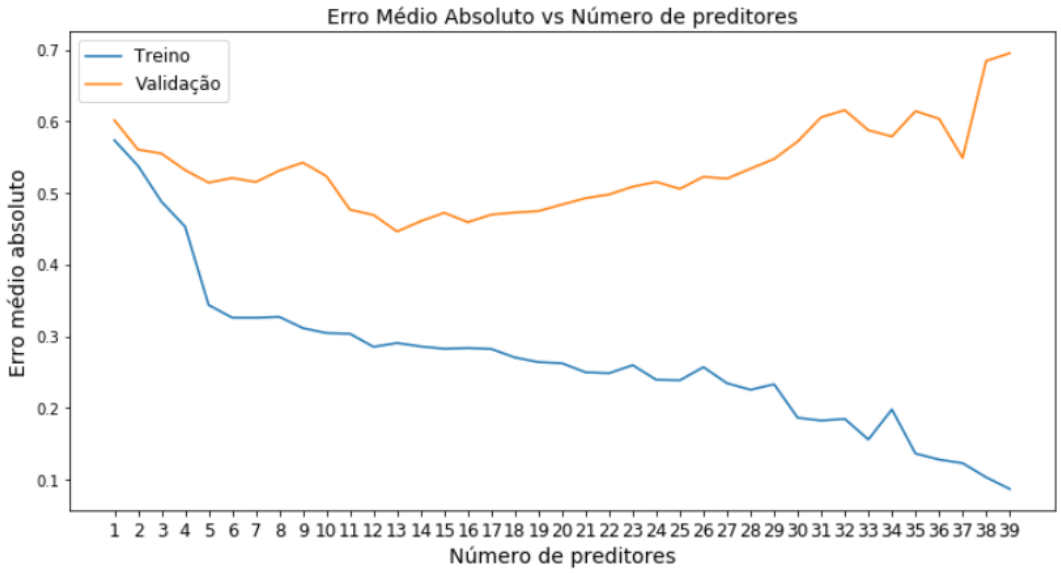
Figura 14 – Erro médio absoluto vs número de preditores com função de ativação sigmoide



Fonte: Próprio Autor

A Figura 15 exibe os resultados obtidos pela mesma abordagem da figura anterior, mas utilizando como função de ativação a função tangente hiperbólica.

Figura 15 – Erro médio absoluto vs número de preditores com função de ativação tangente hiperbólica



Fonte: Próprio Autor

As curvas são as tipicamente obtidas quando métricas de precisão de um modelo são plotadas contra seu grau crescente de complexidade, sendo o grau de complexidade aqui representado pelo número de preditores.

Nestes casos após um determinado grau de complexidade a métrica de precisão no conjunto de dados de treino cai com o aumento da complexidade, ao passo que a mesma métrica no conjunto de validação atinge um equilíbrio ou então aumenta novamente. Essa inversão/estabilização é interpretada como o ponto onde o modelo começa a sobreajustar os dados de treinamento.

Verificando-se então a ordem de entrada de cada variável em seu respectivo modelo e somando-se esses índices. Os resultados obtidos apenas para primeiras dez variáveis com a menor soma das ordens de entrada são exibidos na Tabela 1.

Tabela 1 – *Ranking* de variáveis de acordo com a entradas nos modelos

Variável	Ordem de entrada sigmoide	Ordem de entrada tangente hiperbólica	Soma das ordens
9a	4	6	10
9b	7	3	10
23b	3	11	14
20k	11	4	15
6a	13	5	18
20a	17	2	19
9k	6	15	31
5b	2	21	23
17k	8	16	24
31a	16	10	26

Fonte: Próprio Autor

Então as cinco primeiras variáveis com a menor soma das ordens de entrada ($9a$, $9b$, $23b$, $20k$, $6a$) foram selecionadas para modelar a rede neural final. Esse número foi escolhido tendo em vista o pouco número de amostras disponíveis para treinamento.

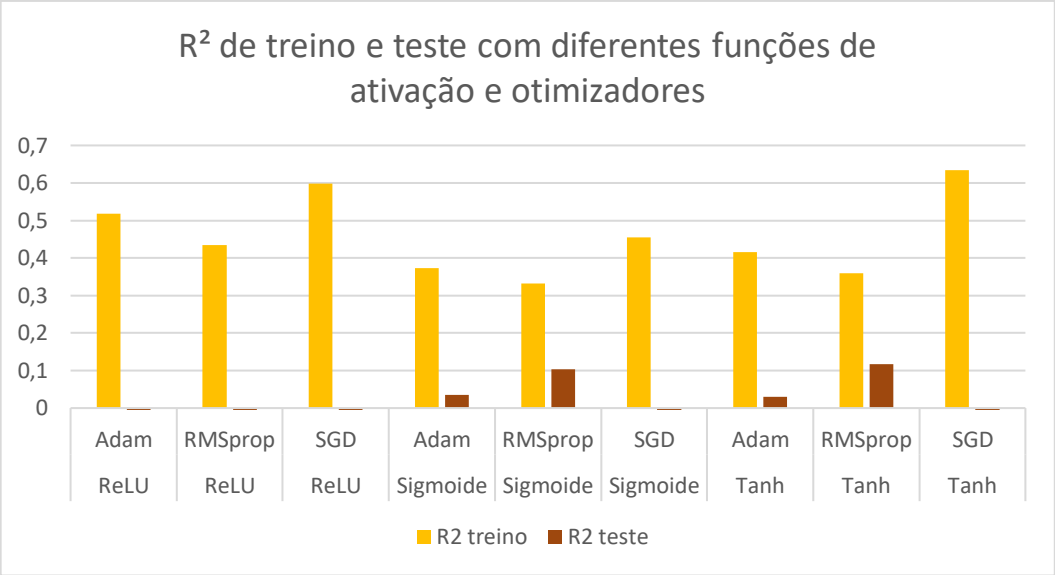
Verificando-se nas Figura 11, Figura 12 e Figura 13 percebe-se que dos parâmetros selecionados, os mais bem comportados são o $20k$, $23b$ e $9b$ por apresentarem poucos *outliers*, ao passo que os parâmetros 9a e 6a possuem valores com uma maior quantidade de *outliers.*

4.3 AJUSTE DOS PARÂMETROS

Tendo em vista o pouco volume de dados disponíveis, utilizou-se validação cruzada em cinco partições na etapa de ajuste dos parâmetros visando uma maior robustez dos resultados obtidos.

Normalizando-se então os dados em valores z e variando os parâmetros descritos em 3.3.2 utilizando-se 200 épocas, obteve-se os resultados exibidos na Figura 16.

Figura 16 – R² de treino e teste variando-se a função de ativação e otimizador



Fonte: Próprio Autor

Por motivos de visualização, valores negativos foram convertidos em zero. Assim, analisando a Figura 16, nota-se que a função *ReLU* exibiu um melhor R² de 0,5977 nos dados de treino quando em conjunto com o otimizador *SGD*. No entanto, no conjunto de dados de teste obteve um desempenho nulo, independente do otimizador utilizado para os dados de teste. Isso pode ser caracterizado como um caso de grande sobre ajuste da rede em relação aos dados de treino.

A função sigmoide apresentou, assim como a função *ReLU*, seu melhor valor de R² de treino (0,4548) quando treinada com o otimizador *SGD* e também um R² de teste nulo. Já quando utilizada com o otimizador *RMSprop*, o R² de treino foi de 0,3314 e o de teste 0,1042.

Assim como a função sigmoide e *ReLU*, os dados obtidos com a função tangente hiperbólica (Tanh) exibiram o melhor R² de treino e um R² de teste nulo quando a rede foi treinada com o *SGD*. E assim como a sigmoide, a função Tanh também obteve o melhor R² de teste(0,1167) quando treinada com o *RMSprop*.

Analisando-se o efeito do otimizador sobre o R² de treino, nota-se que o *SGD* exibe uma maior capacidade de ajuste aos dados, no entanto essa capacidade falha ao ser testada contra novas amostras, ou seja, não permite uma capacidade de generalização resultando no R² de teste obtido de 0.

Dados os baixos valores exibidos na Figura 16, verificou-se como a alteração da função de saída da rede afetaria os resultados. Os resultados, mantendo-se a função de ativação da camada oculta como a *ReLU* e variando a de saída, são exibidos na Tabela 2.

Tabela 2 – R^2 de treino e teste para rede neural com função de ativação *ReLU* na camada oculta e variando a de saída

Função Otimizador	R^2 treino *ReLU*	R^2 treino Sigmoide	R^2 treino Tanh	R^2 teste *ReLU*	R^2 teste Sigmoide	R^2 teste Tanh
Adam	0,4067	0,3168	0,4295	-1,6461	0,0391	-0,0769
RMSprop	0,2838	0,2568	0,3905	-0,4124	0,0179	0,0049
SGD	0,4455	0,2996	0,4649	-1,37701	0,0035	-0,2784

Fonte: Próprio Autor

Comparando-se os dados acima com os valores obtidos para a função *ReLU* da Figura 16, percebe-se que nenhuma das funções utilizadas na saída obteve um R^2 de treino melhor, independente do otimizador. Já no que tange aos resultados de teste, todas obtiveram uma ligeira melhora comparada com os valores referência , sendo que o melhor desempenho é atribuído ao uso da função sigmoide na camada de saída.

O fato de o R^2 de treino ter sido menor com um ganho simultâneo no R^2 de teste (máximo de 0,0391 vs -0,0315) pode ser associado a uma leve melhora na capacidade de generalização da rede, no entanto, os baixos valores ainda caracterizam um sub ajuste dos da rede aos dados.

Os resultados mantendo-se a função de ativação da camada oculta como a sigmoide e variando a de saída são exibidos na Tabela 3.

Tabela 3 – R^2 de treino e teste para rede neural com função de ativação sigmoide na camada oculta e variando a de saída

Função Otimizador	R^2 treino *ReLU*	R^2 treino Sigmoide	R^2 treino Tanh	R^2 teste *ReLU*	R^2 teste Sigmoide	R^2 teste Tanh
Adam	0,2177	0,2109	0,3479	-0,1138	0,0399	0,0689
RMSprop	0,2249	0,1752	0,3232	0,0137	0,0484	0,0951
SGD	0,2233	0,1839	0,3552	-0,0034	0,0068	0,0307

Fonte: Próprio Autor

Comparando-se então os resultados obtidos para a função sigmoide da Figura 16, nenhuma melhora significativa no desempenho para os valores de treino foi observada. Já para os dados de teste, apenas quando se modelou a rede ou com o *RMSprop* ou com a função de saída como a *ReLU* não se obteve melhora perante os dados referência.

Neste caso, o R^2 de treino das combinações foi menor quando comparado com a rede original e os R^2 de teste foram, em média, ligeiramente maiores, sofrendo do mesmo efeito que a rede com função da camada oculta *ReLU* que é o sub ajuste.

Por fim, os resultados mantendo-se a função de ativação da camada oculta como a tangente hiperbólica e variando a de saída são exibidos na Tabela 4.

Tabela 4 – R^2 de treino e teste para rede neural com função de ativação tangente hiperbólica na camada oculta e variando a de saída

	R^2 treino			**R^2 teste**		
Função Otimizador	***ReLU***	**Sigmoide**	**Tanh**	***ReLU***	**Sigmoide**	**Tanh**
Adam	0,3326	0,2891	0,3752	-0,2075	-0,0285	0,06460
RMSprop	0,2767	0,2589	0,3426	-0,0790	-0,0005	0,1069
SGD	0,5370	0,2799	0,4293	-0,6988	-0,0176	0,1051

Fonte: Próprio Autor

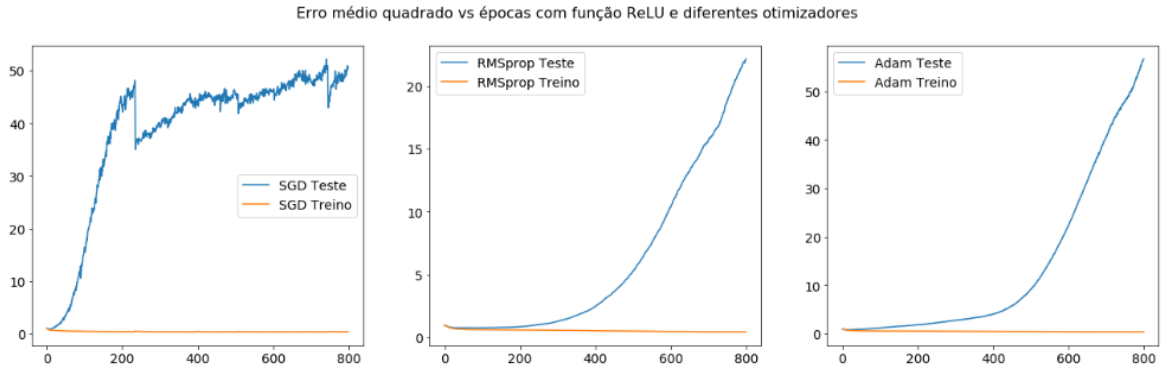
Neste caso, o R^2 de treino dos dados referência (Figura 16) foi menor para todos os casos analisados. Já para os resultados no conjunto de teste, o uso da função *ReLU* foi acompanhado de uma queda no desempenho em todos os casos, e o uso da sigmoide apresentou melhora apenas com o otimizador *SGD*.

Apresentando grande melhoria nesta etapa, foi o uso da função tangente hiperbólica que apesar do decréscimo no R^2 de treino foi acompanhado de um acréscimo no R^2 de teste, sendo este aumento mais pronunciado no caso do otimizador *SGD* (-0,3193 para 0,1051). O valor de R^2 de teste neste caso, dobrou para o otimizador *Adam* e permaneceu praticamente constante para o *RMSprop*.

Como os resultados obtidos para combinações foram baixos, verificou-se a hipótese de sobre ajuste da rede com função de saída linear. Os gráficos foram feitos com 800 épocas de treinamento, visando permitir uma melhor visualização de todo o treinamento.

Colocou-se então em um gráfico a curva do erro médio quadrado da função *ReLU* e os otimizadores utilizados. O resultado é exibido na Figura 17.

Figura 17 – Erro médio quadrado vs épocas para função de ativação ReLU com diversos otimizadores



Fonte: Próprio Autor

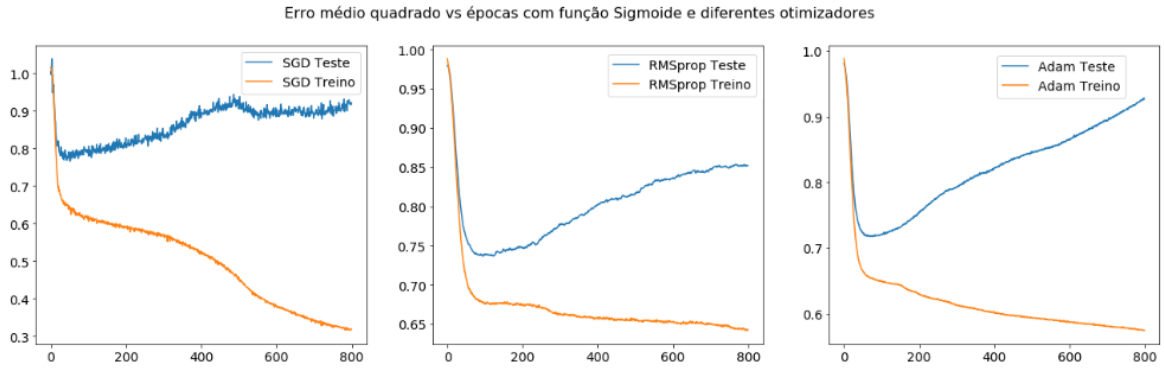
Percebe-se pela análise da Figura 17, que quando do treino da rede com a função *ReLU* o erro médio quadrado decresce muito lentamente, independente do otimizador, o que significa que a rede lentamente vai se adaptando cada vez melhor ao conjunto de dados de treino, ao passo que o erro médio quadrado do conjunto de teste cresce devido à esse mesmo ajuste mencionado.

Neste caso, uma parada precoce do treinamento não solucionaria o problema, visto que o erro médio não decresce de forma consistente em nenhuma etapa do treinamento.

O desempenho da função *ReLU* nessas condições pode ser explicada pelo fato de que quando esse tipo de função é aplicada em redes neurais de poucas camadas, denominadas redes rasas, um número maior de neurônios é necessário para que a rede consiga encontrar mínimos mais precisos da função erro (ECKLE; SCHMIDT-HIEBER, 2019).

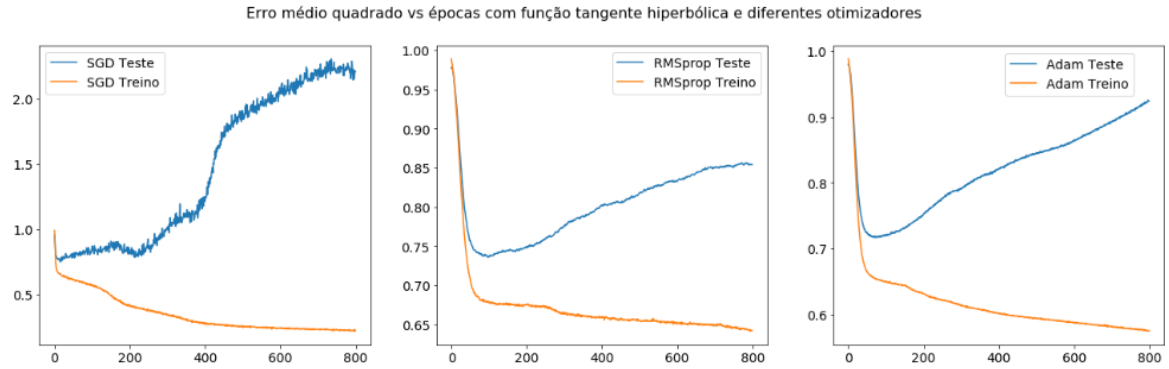
A mesma curva exibida para a função de ativação *ReLU* foi feita para as outras duas funções de ativação sigmoide e tangente hiperbólica nas Figura 18 e Figura 19, respectivamente.

Figura 18 – Erro médio quadrado vs épocas para função de ativação sigmoide com diversos otimizadores



Fonte: Próprio Autor

Figura 19 – Erro médio quadrado vs épocas para função de ativação tanh com diversos otimizadores

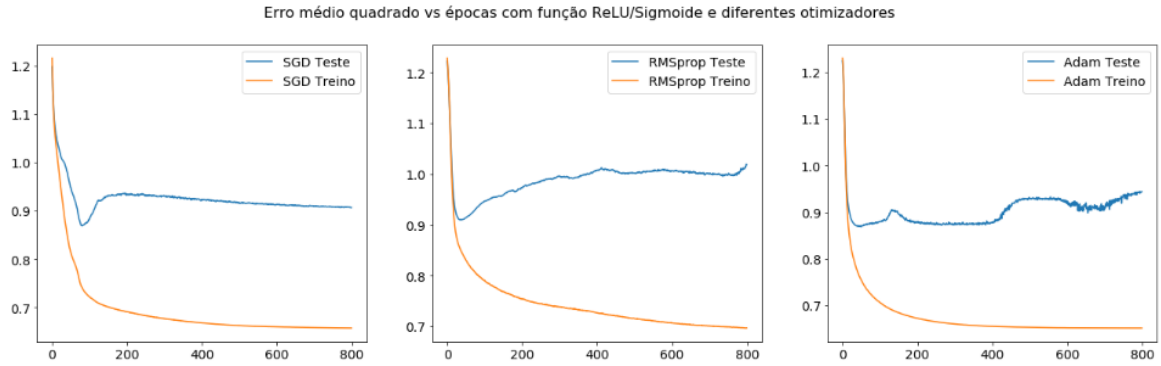


Fonte: Próprio Autor

Analisando a Figura 18, percebe-se que antes das 200 épocas as funções sigmoide e tangente hiperbólica, independente do otimizador, haviam atingido um erro mínimo quadrado e esse aumentou novamente até que o treinamento terminasse. O mesmo comportamento é evidenciado nas Figura 19. Esse comportamento é característico de processos de sobre ajuste.

A mesma análise da hipótese de sobre ajuste foi conduzida para as redes onde a função de saída foi variada. Alguns casos mais pronunciados para cada par função de ativação na camada oculta e de saída são mostrados nas figuras abaixo.

Figura 20 – Erro médio quadrado vs épocas para função de ativação *ReLU* na camada oculta e sigmoide na saída com diversos otimizadores

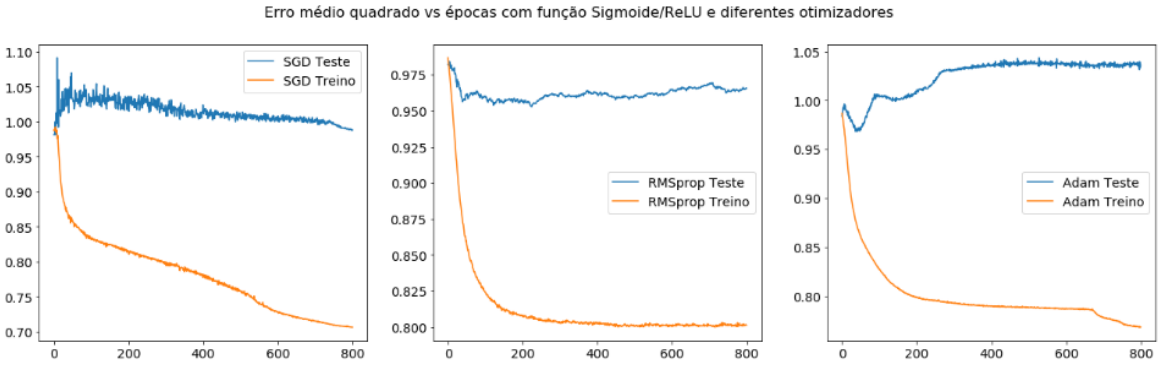


Fonte: Próprio Autor

Analisando-se a Figura 20 em conjunto com os dados da Tabela 2, nota-se que no caso da rede com o par de funções *ReLU* – sigmoide a rede exibe sinais de sobre ajuste para todos os otimizadores, exceto o *SGD,* sendo a forma menos pronunciada quando do uso do otimizador *Adam*.

Entretanto, como evidenciado na Tabela 2, ainda que a rede sofra tal processo, seus baixos valores de R^2 as colocam também na condição de ou não ter complexidade suficiente para capturar as relações nos dados ou não ser a função que melhor representa a função verdadeira.

Figura 21 – Erro médio quadrado vs épocas para função de ativação sigmoide na camada oculta e ReLU na saída com diversos otimizadores

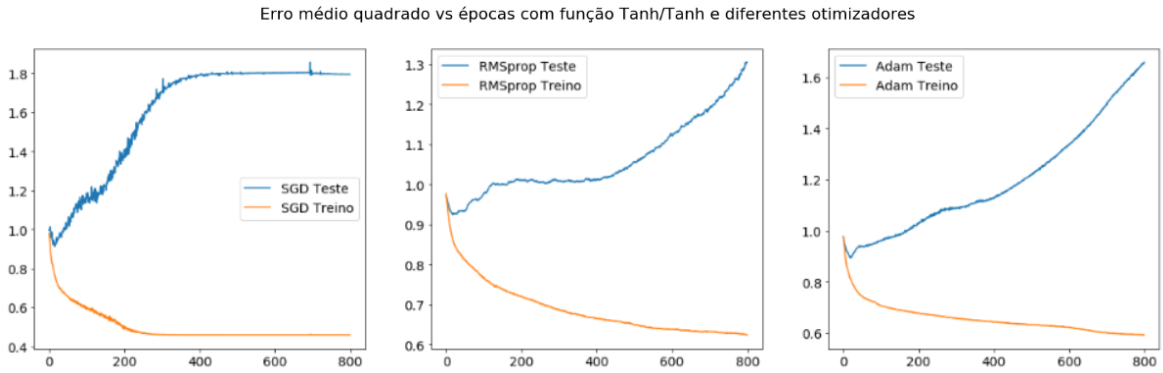


Fonte: Próprio Autor

Analisando a Figura 21, nota-se que quando a sigmoide é usada em conjunto com a *ReLU* , a rede exibe sinais de sobre ajuste leve para o otimizador *RMSprop* e mais intenso para o *Adam*. Já com o otimizador *SGD* nota-se uma queda consistente e lenta do erro médio dos dados de teste, ainda que de forma muito ruidosa, dificultando assim determinar um ponto ótimo de parada do treinamento.

Essa diferença no número de épocas necessárias para determinação do mínimo quando comparado o otimizador *SGD* com o *RMSprop* e o *Adam* pode ser explicado pela diferença no valor da taxa de aprendizado. Estes possuem uma taxa de aprendizado que é atualizada a cada época de treinamento, enquanto que naquela a é fixa ao longo de todo o treinamento, refletindo então no número de épocas necessário para convergência (RUDER, 2016).

Figura 22 – Erro médio quadrado vs épocas para função de ativação tangente hiperbólica na camada oculta e na saída com diversos otimizadores



Fonte: Próprio Autor

Pela Figura 22 percebe-se que em todos um casos um número alto de épocas causa um sobre ajuste da rede aos dados quando do treinamento da rede com função de ativação tangente hiperbólica na camada oculta e na de saída, e que o mínimo local no conjunto de teste é encontrado rapidamente para então subir de forma consistente até o final do treinamento.

Tendo confirmado então a hipótese de sobre ajuste da rede aos dados de treino, modelou-se a rede com um mecanismo denominado *Early Stopping*. Tal mecanismo interrompe o processo de treinamento da rede quando uma métrica de interesse deixa de seguir um comportamento desejado. Em modelos preditivos, utilizam-se métricas como erro quadrado médio (PRECHELT, 2012).

Nos resultados daqui para frente utilizou-se então o erro quadrado médio como métrica para determinar quando o treinamento deveria ser interrompido. O critério de parada foi um aumento por 10 épocas consecutivas do erro nos dados de teste.

Além disso, dada a baixa performance da rede, mesmo em casos onde o sobre ajuste foi pequeno estudou-se também o efeito do aumento do número de neurônios da camada oculta sobre o R^2 de treino e teste. O número de neurônios foi variado no intervalo [5,25].

Os resultados obtidos para redes que foram modeladas com a função de ativação *ReLU* na camada oculta são exibidos na Tabela 5.

Tabela 5 – R^2 de treino e teste com função tangente *ReLU* na camada oculta e variando função de saída, otimizador e neurônios

Função	Otimizador	Épocas	# neurônios	R^2 treino	R^2 teste
Linear	*Adam*	44	5	0,4205	0,0747
	RMSprop	16	5	0,3548	0,1499
	SGD	37	12	0,4775	-0,2819
ReLU	*Adam*	27	5	0,2829	-0,0045
	RMSprop	28	16	0,2169	0,0454
	SGD	11	20	0,2969	-0,0551
Sigmoide	*Adam*	103	7	0,2758	0,0795
	RMSprop	73	9	0,2158	0,0606
	SGD	96	24	0,2130	0,0369
Tangente hiperbólica	*Adam*	31	7	0,3809	0,1294
	RMSprop	54	6	0,3430	0,1694
	SGD	31	7	0,3972	0,1178

Fonte: Próprio Autor

Analisando os valores da tabela acima com os obtidos anteriormente para redes que tinham a função *ReLU* na camada oculta (Figura 16 e Tabela 2), nota-se que em todos os casos foram obtidos desempenhos superiores aos anteriores.

Quando a função da camada de saída foi ou a *ReLU* ou a sigmoide, nota-se que um aumento significativo da complexidade da rede foi necessário para que desempenho maiores fossem obtidos.

De destaque, nota-se a um ganho de mais de dez vezes quando a rede foi modelada com a tangente hiperbólica na saída sem um ganho excessivo na complexidade da rede, indicando assim que nesta estrutura o *Early Stopping* teve um efeito positivo pronunciado na capacidade de generalização da rede.

Os resultados obtidos quando para a rede modelada com função de ativação da camada oculta como sigmoide são exibidos na Tabela 6.

Tabela 6 – R^2 de treino e teste com função sigmoide na camada oculta e variando função de saída, otimizador e neurônios

Função	Otimizador	Épocas	# neurônios	R^2 treino	R^2 teste
Linear	*Adam*	63	10	0,3378	0,2898
	RMSprop	79	7	0,2981	0,2838
	SGD	73	5	0,3745	0,2628
ReLU	*Adam*	54	25	0,2229	0,0862
	RMSprop	23	23	0,1556	0,0556
	SGD	55	6	0,2279	0,0551
Sigmoide	*Adam*	142	22	0,2259	0,0952
	RMSprop	129	25	0,1542	0,0481
	SGD	161	5	0,1595	0,0560
Tangente hiperbólica	*Adam*	67	12	0,3266	0,2986
	RMSprop	63	15	0,3015	0,2857
	SGD	51	11	0,3280	0,2908

Fonte: Próprio Autor

Comparando-se agora os valores da tabela acima com os da Figura 16 e da Tabela 3, nota-se que quando a função de saída utilizada foi a linear ou a tangente hiperbólica um desempenho muito maior em relação ao R^2 de teste foi obtido.

Já quando as funções de saída foram a *ReLU* e a própria sigmoide, nota-se um ganho grande na complexidade da rede sem um ganho no desempenho no conjunto de teste que justifique o ganho de complexidade.

Os resultados obtidos quando para a rede modelada com função de ativação da camada oculta como tangente hiperbólica são exibidos na Tabela 7.

Tabela 7 – R^2 de treino e teste com função tangente hiperbólica na camada oculta e variando função de saída, otimizador e neurônios

Função	Otimizador	Épocas	# neurônios	R^2 treino	R^2 teste
Linear	*Adam*	38	5	0,3833	0,2672
	RMSprop	38	5	0,3314	0,2616
	SGD	26	17	0,3985	0,2286
ReLU	*Adam*	68	8	0,2896	0,0970
	RMSprop	58	17	0,2403	0,0854
	SGD	30	8	0,3180	-0,0033
Sigmoide	*Adam*	90	5	0,2518	0,1025
	RMSprop	71	5	0,2160	0,0524
	SGD	67	19	0,2330	0,0915
Tangente hiperbólica	*Adam*	36	5	0,3372	0,2879
	RMSprop	41	5	0,3186	0,2726
	SGD	20	24	0,3383	0,2583

Fonte: Próprio Autor

Por último, analisando-se os dados acima com os da Figura 16 e Tabela 4, percebe-se que os dados se comportaram de forma análoga aos das duas tabelas anteriores, onde ganhos mais pronunciados foram exibidos quando a função de saída da rede foi a linear ou tangente hiperbólica, com ganhos de complexidade variável.

Essa diferença quando do uso das funções de saída pode ser explicada pelo fato de que a função sigmoide e a função *ReLU* possuem características que dificultam o processo de treinamento.

A função sigmoide possui um intervalo de saída pequeno e comprime os resultados de sua derivada em um espaço muito curto do eixo x, fazendo com que o gradiente facilmente atinja zero no momento do treinamento, o que impede um treinamento maior da rede. A função tangente hiperbólica por sua vez, possui um intervalo de saída maior e um gradiente menos concentrado.

Já a função *ReLU*, apesar de muito semelhante com a linear no intervalo positivos das abscissas, sofre do problema de gradientes iguais a zero quando a entrada é negativa, dificultando assim o refinamento dos pesos e bias da rede.

Percebe-se então que a função de saída da rede tem influência significativa no desempenho da rede neural, sendo as funções lineares e tangente hiperbólica, neste caso, as que favorecem resultados maiores.

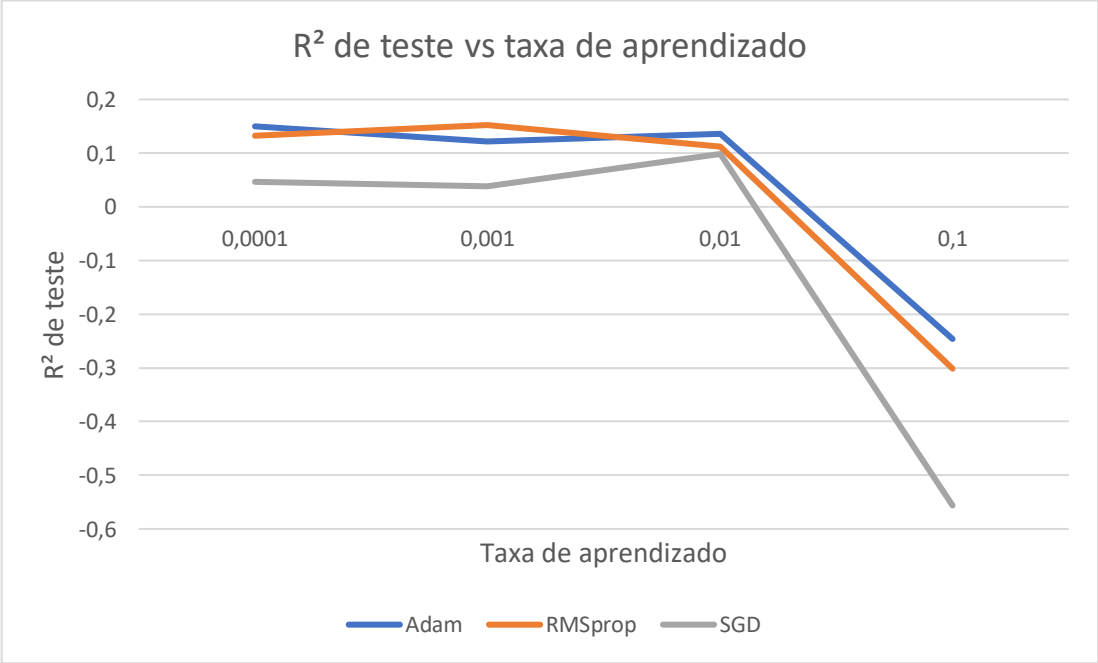
Apesar de as redes com a função tangente hiperbólica na saída terem obtido um bom desempenho quando comparadas as com outras funções de ativação, tais redes estariam com uma restrição quando utilizadas na predição devido ao fato de os valores estarem normalizados em valores z e a tangente hiperbólica ter sua saída no intervalo [-1,1].

Logo, os estudos feitos com funções de ativação que não a linear na camada de saída foram realizadas com o intuito de analisar diferentes estruturas da rede, mas não possuem efeito prático nesse trabalho e serão portanto descartadas deste ponto em diante.

Tendo-se então aplicado o mecanismo de *Early Stopping* para diminuir a intensidade de sobre ajuste da rede aos dados, verificou-se o efeito da diminuição da taxa de aprendizado da rede, uma vez que o mecanismo citado opera em cima da função erro da rede que por sua vez depende da taxa de aprendizado da mesma.

Verificou-se então o efeito de quatro valores da taxa de aprendizado: 0,0001, 0,001, 0,01, 0,1. Os resultados foram compilados como a média para as três funções de ativação pelo fato de terem exibido curvas muito semelhantes e são exibidos nas Figura 23.

Figura 23 – R² de teste vs taxa de aprendizado para diferentes otimizadores



Fonte: Próprio Autor

Pela figura acima percebe-se que com o *Adam* e o *RMSprop* obteve-se um desempenho inferior no R² de teste quando se aumentou a taxa de aprendizado do otimizador, sendo essa queda pronunciada quando se elevou a taxa de aprendizado de 0,01 para 0,1. Essa degradação pode ser explicada pelo fato de que quando tal parâmetro

é muito alto, o otimizador atualiza os pesos tão grandemente que pontos de mínimo da função erro não são detectados em próximos processos de atualização.

Nota-se também que o otimizador *SGD* exibe uma curva diferente dos outros nos três primeiros pontos analisados, ou seja, ele apresenta uma melhora no R^2 de teste com o aumento da taxa de aprendizado nesse intervalo. Isso pode ser explicado pelo fato de que o *SGD*, quando utilizado com taxas de aprendizado muito baixas movimenta-se muito lentamente em direção ao mínimo da função erro (RUDER, 2016).

Além disso o *SGD* é conhecido por uma curva de erro muito ruidosa devido a atualização a cada amostra apresentada ao modelo, isso aliado a uma baixa taxa de aprendizado e ao mecanismo de *Early Stopping* pode ter causado uma parada precoce do algoritmo.

Dado o baixo desempenho geral da rede, variou-se também o método de estimação inicial da matriz de pesos e *bias*. Foram testados os métodos com extração da distribuição normal com média 0 e desvio padrão 0,05 e da distribuição uniforme no intervalo [-0,05; 0,05]. Os resultados são exibidos na Tabela 8.

Tabela 8 – R^2 de teste para diversos otimizadores variando-se o iniciador de parâmetros

		Otimizador		
Função	**Iniciador**	***Adam***	***RMSprop***	***SGD***
ReLU	Uniforme	0,0799	0,1592	0,0032
ReLU	Normal	0,1219	0,1259	-0,0726
Sigmoide	Uniforme	0,1003	0,1114	-0,0065
Sigmoide	Normal	0,0708	0,1131	-0,0739
Tanh	Uniforme	0,1854	0,1863	0,1164
Tanh	Normal	0,1890	0,1788	0,1882

Fonte: Próprio Autor

Percebe-se que o modo com que a matriz de parâmetros da rede é iniciada influencia de forma significativa como o algoritmo *SGD* desempenhou e em menor grau os outros dois otimizadores.

A função *ReLU* e a função sigmoide exibiram melhor desempenho quando os parâmetros foram iniciados através da extração de valores da distribuição uniforme, já para a função tangente hiperbólica, o modo de inicialização dos parâmetros não afetou o desempenho.

Buscando então a melhor combinação dos parâmetros variou-se o número de neurônios na camada oculta entre 5 e 25 com incrementos de 5, as taxas de aprendizado e iniciadores demonstrados visando encontrar o melhor conjunto de parâmetros. Os cinco melhores modelos são exibidos na Tabela 9.

Tabela 9 – R^2 de treino e teste variando-se a função de ativação, otimizador, neurônios, taxa de aprendizado e iniciador

Função	**Otimizador**	**Épocas**	**Neurônios**	**Taxa de aprendizado**	**Iniciador**	**R^2 treino**	**R^2 teste**
ReLU	*SGD*	44	25	0,01	Normal	0,6800	0,2659
Tanh	*Adam*	29	15	0,01	Uniforme	0,4123	0,2453
Tanh	*RMSprop*	30	15	0,01	Normal	0,3609	0,2431
Tanh	*RMSprop*	30	10	0,01	Normal	0,3640	0,2318
ReLU	*Adam*	30	25	0,0001	Uniforme	0,4680	0,2281

Fonte: Próprio Autor

O melhor R^2 de teste obtido foi de 0,2659 para a função *ReLU* com o otimizador *SGD* com taxa de aprendizado de 0,01 e iniciador normal e 25 neurônios. Nota-se, no entanto, que o segundo e terceiro modelos obtiveram performance análoga ao melhor e sem um número excessivo de neurônios, o que pode ser uma dificuldade no processo de convergência devido ao pouco número de amostras disponíveis para treinamento.

Como baixos valores de R^2 de teste foram obtidos, argumentou-se que a rede não era dotada de complexidade suficiente para captar a função que verdadeiramente regia os dados.

Tendo em vista que os melhores resultados foram obtidos para redes com funções de ativação na primeira camada como sendo a *ReLU* ou a tangente hiperbólica e de que a inicialização de pesos para estas funções foi melhor quando se usou o método de extração da normal, seguiu-se com uma modelagem da rede com duas camadas dotada das características mencionadas. Os cinco melhores modelos obtidos são exibidos na Tabela 10.

Tabela 10 – R^2 de teste para rede neural de duas camadas

Função Camada 1	Função Camada 2	Neurônios Camada 1	Neurônios Camada 2	Otimizador	Épocas	R^2 teste
Tanh	*ReLU*	8	8	*Adam*	45	0,3034
Tanh	*ReLU*	7	13	*Adam*	55	0,2898
Tanh	Tanh	20	25	*Adam*	56	0,2867
Tanh	*ReLU*	10	8	*RMSprop*	57	0,2861
Tanh	*ReLU*	8	9	*Adam*	48	0,2860

Fonte: Próprio Autor

Nota-se que uma melhoria de cerca de 14% comparando os melhores modelos foi observada quando se modelou a rede com duas camadas. No entanto, esse incremento no R^2 de teste é pequeno quando comparado com o ganho de complexidade no treinamento da rede, tendo em vista que quanto mais parâmetros a serem determinados, mais amostras são necessárias para um processo de treinamento completo.

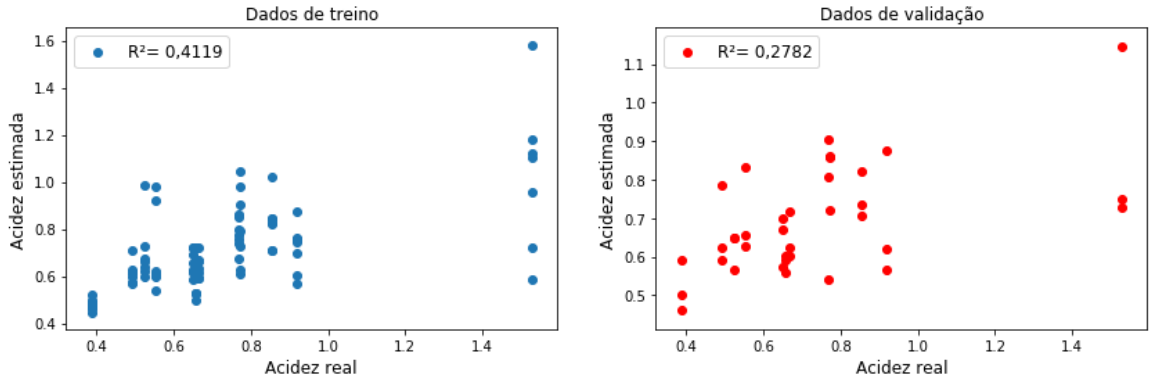
Pelo motivo acima não foi considerado um aumento no número de camadas da rede e parou-se aqui o crescimento da rede. O melhor modelo obtido então foi o com a tangente hiperbólica na primeira camada de 8 neurônios, *ReLU* na segunda camada de 8 neurônios, otimizador *Adam* com um R^2 de treino de 0,3918 e de teste de 0,3034.

Comparou-se então o desempenho da rede neural com a regressão linear, sendo que o método de seleção das variáveis para o modelo linear foi o método *Stepwise*.

Na modelagem da regressão linear considerou-se uma abordagem na qual a média dos valores das variáveis preditoras por variável dependente era utilizada aos invés das 10 leituras recomendadas pelo fabricante do nariz eletrônico.

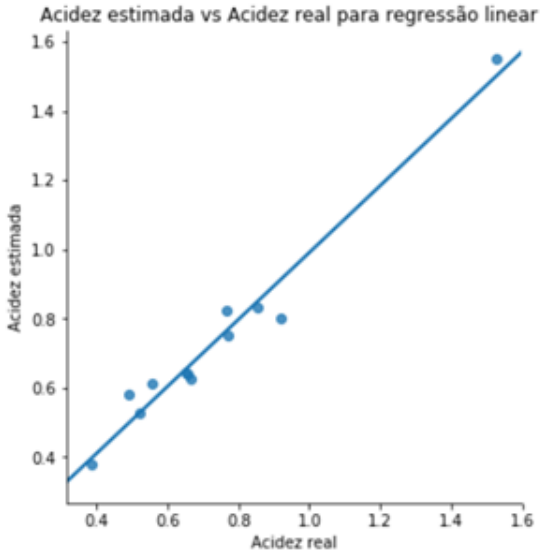
Os resultados obtidos para a rede neural e para a regressão linear são exibidos nas Figuras 24 e 25.

Figura 24 – R² de treino e teste para a rede neural final



Fonte: Próprio Autor

Figura 25 – Acidez estimada vs Acidez real para a regressão linear multivariada



Fonte: Próprio Autor

Para a regressão linear os R² obtidos foram de 0,9664 e 0,915 para os conjuntos de treino e validação, respectivamente, utilizando-se as variáveis independentes M2, P6 K19 e intercepto. Neste caso M é calculado como A + B*K.

Nota-se então que a regressão linear obteve um desempenho muito superior ao da rede neural obtida. Essa diferença na performance pode ser explicada pela dificuldade de obter-se boas estimativas dos coeficientes da rede neural visto o grande número de parâmetros a serem treinados versus o número de amostras disponíveis para tal.

O número de amostras disponíveis para treino de uma rede neural então impõe-se como um problema na aplicação do trabalho em questão, visto que o número de parâmetros treináveis era muito superior ao de amostras. Na rede elaborada, por exemplo,

tem-se 40, 64 e 8 elementos pesos na primeira camada oculta, segunda camada oculta e camada de saída, mais 8, 8 e 5 elementos *bias*.

Logo, ainda que a rede tivesse obtido um desempenho minimante satisfatório, tal valor deveria ser contestado ainda assim, visto que o número de graus de liberdade seria negativo, o que acarretaria em problemas quanto à confiabilidade do uso do modelo em novas amostras.

O número de graus liberdade fornece informação acerca da confiança na variabilidade dos parâmetros estimados de um dado modelo. É comumente definido como a diferença entre o número de amostras disponíveis menos o número de parâmetros ou número de relações necessárias para obtenção de tais parâmetros de um modelo de escolha (ZOURNAZI, 2017).

5 CONCLUSÃO

Nesse trabalho verificou-se a capacidade de redes neurais em estimar a acidez de óleos vegetais tendo como variável independente os parâmetros do algoritmo estocástico desenvolvido por SIQUEIRA *et al*., (2018) para modelagem do sinal lido por um nariz eletrônico.

Os resultados obtidos indicam que as redes neurais não são uma boa escolha de algoritmo para este objetivo, o que pode ser atribuído ao baixo número de amostras disponíveis para treinamento.

Quanto a isso, dois pontos são de ressalva: a importância de um número razoável de amostras para treinamento adequado da rede de forma a obter uma melhor estimativa dos coeficientes das matrizes de peso e *bias*; um número razoável de graus de liberdade de uma rede neural com bom desempenho tendo em vista que o número de parâmetros treináveis de uma rede cresce muito rapidamente.

Deve-se então atentar a esses dois fatores quando da escolha de uma rede neural como modelo para determinada tarefa, especialmente dado o teorema de Kolmogorov (KŮRKOVÁ, 1992) que diz que uma rede neural de duas camadas ocultas e número adequado de neurônios consegue aproximar qualquer função contínua.

Outro fator que pode ter contribuído para um baixo desempenho da rede foi a escolha das variáveis independentes. Outros métodos de escolha de variáveis como o de importância da variável FISHER; RUDIN; DOMINICI (2018), no entanto, não foram considerados neste trabalho.

Tem-se como hipótese também a falta de complexidade da rede, no entanto, devido à baixa quantidade de amostras para treino, um ganho progressivo da complexidade dos modelos não foi possível.

Algoritmos que necessitam de um menor número de amostras para estimação dos parâmetros, como a regressão linear, exibiram desempenho muito superior ao da rede, podendo essa grande diferença de desempenho ser explicada pela melhor determinação da regressão de seus parâmetros mesmo com uma baixa quantidade de amostras disponíveis.

REFERÊNCIAS

BOEKER, P. On “Electronic Nose” methodology. **Sensors and Actuators, B: Chemical**, v. 204, p. 2–17, 2014. Disponível em: <http://dx.doi.org/10.1016/j.snb.2014.07.087> Acesso em 13/09/2018.

BRAGA, A. P.; CARVALHO, A. C. P. L. F.; LUDERMIR, T. B. Redes neurais artificiais: teoria e aplicações. p. 262, 2000.

CHOLLET, F and others, 2015. Keras. Disponível em: <https://github.com/fchollet/kreas>.

CROSS, S. S.; HARRISON, R. F.; KENNEDY, R. L. Introduction to neural networks. **The Lancet**, v. 346, n. 8982, p. 1075–1079, 1995. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0140673695917462> Acesso em 08/09/2018.

DEBAO, C. Degree of approximation by superpositions of a sigmoidal function. **Approximation Theory and its Applications**, v. 9, n. 3, p. 17–28, 1993.

DUTTA, R.; DAS, A.; STOCKS, N. G.; MORGAN, D. Stochastic resonance-based electronic nose: A novel way to classify bacteria. **Sensors and Actuators, B: Chemical**, v. 115, n. 1, p. 17–27, 2006.

ECKLE, K.; SCHMIDT-HIEBER, J. A comparison of deep networks with ReLU activation function and linear spline-type methods. **Neural Networks**, v. 110, p. 232–242, 2019. Disponível em: <https://doi.org/10.1016/j.neunet.2018.11.005> Acesso em 10/10/2019.

FISHER, A.; RUDIN, C.; DOMINICI, F. All Models are Wrong, but Many are Useful: Learning a Variable’s Importance by Studying an Entire Class of Prediction Models Simultaneously. n. Vi, 2018. Disponível em: <http://arxiv.org/abs/1801.01489> Acesso em 20/10/2019.

GHASEMI-VARNAMKHASTI, M.; APETREI, C.; LOZANO, J.; ANYOGU, A. Potential use of electronic noses, electronic tongues and biosensors as multisensor systems for spoilage examination in foods. **Trends in Food Science & Technology**, v. 80, n. August 2017, p. 71–92, 2018. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0924224417305575> Acesso em 13/09/2018.

GORGENS, E. B.; LEITE, H. G.; SANTOS, H. do N.; GLERIANI, J. M. Estimação do volume de árvores utilizando redes neurais artificiais. **Revista Árvore**, v. 33, n. 6, p. 1141–1147, 2009. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-67622009000600016&lng=pt&tlng=pt> Acesso em 08/09/2018.

GOOGLE. TensorFlow. 2015. Disponível em : <https://www.tensorflow.org>

HAYKIN, S. Redes neurais: princípios e prática. **Bookman**, p. 900, 2001.

HEUSEL, M.; RAMSAUER, H.; UNTERTHINER, T.; NESSLER, B.; HOCHREITER, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. **Advances in Neural Information Processing Systems**, v. 2017-Decem, n. Nips, p. 6627–6638, 2017.

KINGMA, D. P.; BA, J. Adam: A Method for Stochastic Optimization. p. 1–15, 2014. Disponível em: <http://arxiv.org/abs/1412.6980> Acesso em 15/09/2019.

KŮRKOVÁ, V. Kolmogorov's theorem and multilayer neural networks. **Neural Networks**, v. 5, n. 3, p. 501–506, 1992.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, v. 521, n. 7553, p. 436–444, 2015.

LISBOA, H.; PAGÉ, T.; GUY, C. Aplicações do nariz eletrônico nas indústrias e na gestão de odores. **Estudos Tecnológicos em Engenharia**, v. 5, n. 2, p. 195–211, 2009. Disponível em: <http://revistas.unisinos.br/index.php/estudos_tecnologicos/article/view/4975> Acesso em 20/09/2018.

LUNDSTRÖM, I. Approaches and mechanisms to solid state based sensing. **Sensors and Actuators, B: Chemical**, v. 35, n. 1–3, p. 11–19, 1996.

LYON, R. F.; LYON, R. F. Neural Networks for Machine Learning. **Human and Machine Hearing**, p. 419–440, 2017.

MAJCHRZAK, T.; WOJNOWSKI, W.; DYMERSKI, T.; GĘBICKI, J.; NAMIEŚNIK, J. Electronic noses in classification and quality control of edible oils: A review. **Food Chemistry**, v. 246, n. June 2017, p. 192–201, 2018.

MAO, J. Why artificial neural networks? 1996.

PARK, J. S.; KIM, H. G.; KIM, D. G.; YU, I. J.; LEE, H. K. Paired mini-batch training: A new deep network training for image forensics and steganalysis. **Signal Processing: Image Communication**, v. 67, n. March, p. 132–139, 2018. Disponível em: <https://doi.org/10.1016/j.image.2018.04.015> Acesso em 02/10/2018.

PRECHELT, L. Early stopping - But when? **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 7700 LECTU, p. 53–67, 2012.

PRIETO, A.; PRIETO, B.; ORTIGOSA, E. M.; ROS, E.; PELAYO, F.; ORTEGA, J.; ROJAS,

I. Neural networks: An overview of early research, current frameworks and new challenges. **Neurocomputing**, v. 214, p. 242–268, 2016. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2016.06.014> Acesso em 09/09/2018.

RÖCK, F.; BARSAN, N.; WEIMAR, U. Electronic nose: Current status and future trends. **Chemical Reviews**, v. 108, n. 2, p. 705–725, 2008.

RUDER, S. An overview of gradient descent optimization algorithms. p. 1–14, 2016. Disponível em: <http://arxiv.org/abs/1609.04747> Acesso em 15/09/2019.

SALARI, M.; SALAMI SHAHID, E.; AFZALI, S. H.; EHTESHAMI, M.; CONTI, G. O.; DERAKHSHAN, Z.; SHEIBANI, S. N. Quality assessment and artificial neural networks modeling for characterization of chemical and physical parameters of potable water. **Food and Chemical Toxicology**, v. 118, n. April, p. 212–219, 2018. Disponível em: <https://doi.org/10.1016/j.fct.2018.04.036> Acesso em 15/09/2018.

SALIH, A. Delta Function and Heaviside Function Heaviside step function. n. February, p. 1–5, 2015.

SIQUEIRA, A. F.; GUIMARÃES, O. L. C.; FILHO, H. J. I.; GIORDANI, D. S. Modeling the Photocatalytic Process of Variation in Chemical Oxygen Demand via Stochastic Differential Equations. p. 1–8, 2013.

SIQUEIRA, A. F.; MELO, M. P.; GIORDANI, D. S.; GALHARDI, D. R. V.; SANTOS, B. B.; BATISTA, P. S.; FERREIRA, A. L. G. Stochastic modeling of the transient regime of an electronic nose for waste cooking oil classification. **Journal of Food Engineering**, v. 221, p. 114–123, 2018. Disponível em: <https://doi.org/10.1016/j.jfoodeng.2017.10.003> Acesso em 15/08/2018.

SVOZIL, D.; KVASNICKA, V.; POSPÍCHAL, J. Introduction to multi-layer feed-forward neural networks. **Chemometrics and Intelligent Laboratory Systems**, v. 39, p. 43–62, 1997.

WILSON, A. C.; ROELOFS, R.; STERN, M.; SREBRO, N.; RECHT, B. The marginal value of adaptive gradient methods in machine learning. **Advances in Neural Information Processing Systems**, v. 2017-Decem, n. Nips, p. 4149–4159, 2017.

WILSON, A. D.; BAIETTO, M. Advances in electronic-nose technologies developed for biomedical applications. **Sensors**, v. 11, n. 1, p. 1105–1176, 2011.

WU, H.; ZHAO, J. Deep convolutional neural network model based chemical process fault diagnosis. **Computers and Chemical Engineering**, v. 115, p. 185–197, 2018. Disponível em: <https://doi.org/10.1016/j.compchemeng.2018.04.009> Acesso em 13/09/2018.

ZHANG, X.; CHENG, J.; WU, L.; MEI, Y.; JAFFREZIC-RENAULT, N.; GUO, Z. An overview of an artificial nose system. **Talanta**, v. 184, n. January, p. 93–102, 2018.

ZHANG, Z.; ZHAO, J. A deep belief network based fault diagnosis model for complex chemical processes. **Computers and Chemical Engineering**, v. 107, p. 395–407, 2017. Disponível em: <https://doi.org/10.1016/j.compchemeng.2017.02.041> Acesso em 13/09/2018.

ZOURNAZI, M. Introduction to the Article degrees of freedom. **Alternative Approaches in Conflict Resolution**, v. 31, n. 4, p. 149–152, 2017.