



UNIVERSIDADE FEDERAL DE SERGIPE - UFS
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA – CCET
NÚCLEO DE PÓS-GRADUAÇÃO EM FÍSICA – NPGFI

DISSERTAÇÃO

**APLICAÇÃO DE REDES NEURAIS DO TIPO FUNÇÃO DE BASE RADIAL
SIMPLES COM POUCOS NEURÔNIOS NA PREVISÃO DA TEMPERATURA DE
TRANSIÇÃO VÍTREA DE BORATOS DE METAIS ALCALINOS**

LEONARDO DOS SANTOS VITÓRIA

SÃO CRISTÓVÃO – SERGIPE
JANEIRO – 2022



UNIVERSIDADE FEDERAL DE SERGIPE - UFS
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA – CCET
NÚCLEO DE PÓS-GRADUAÇÃO EM FÍSICA – NPGFI

LEONARDO DOS SANTOS VITÓRIA

**APLICAÇÃO DE REDES NEURAIS DO TIPO FUNÇÃO DE BASE RADIAL
SIMPLES COM POUCOS NEURÔNIOS NA PREVISÃO DA TEMPERATURA DE
TRANSIÇÃO VÍTREA DE BORATOS DE METAIS ALCALINOS**

Dissertação apresentada ao Núcleo de Pós-Graduação em Física da Universidade Federal de Sergipe como requisito parcial para obtenção do título de Mestre em Física.

Orientadora: Profa. Dra. Susana de Souza Lalic

Coorientador: Prof. Dr. Marcio Luis Ferreira Nascimento

SÃO CRISTÓVÃO – SERGIPE
JANEIRO – 2022

LEONARDO DOS SANTOS VITÓRIA

**APLICAÇÃO DE REDES NEURAIS DO TIPO FUNÇÃO DE BASE RADIAL
SIMPLES COM POUCOS NEURÔNIOS NA PREVISÃO DA TEMPERATURA DE
TRANSIÇÃO VÍTREA DE BORATOS DE METAIS ALCALINOS**

**Dissertação apresentada como um dos requisitos para a obtenção do grau de
mestre em Física ao Núcleo de Pós-Graduação em Física da Universidade
Federal de Sergipe**

Aprovado em:

São Cristóvão, 31 de JANEIRO de 2022.

BANCA EXAMINADORA

Profa. Dra. Susana de Souza Lalic (Orientadora)

Universidade Federal de Sergipe - UFS

Prof. Dr. Daniel Roberto Cassar

Centro Nacional de Pesquisa em Energia e Materiais - CNPEM

Prof. Dr. Marcos Antonio Couto dos Santos

Universidade Federal de Sergipe - UFS

**SÃO CRISTÓVÃO - SERGIPE
JANEIRO – 2022**

Dedico este trabalho a minha avó Maria Chagas dos Santos (in memoriam), e a todas as vítimas de COVID 19.

AGRADECIMENTOS

A Deus.

À Ilma de Jesus Santos, dona Joséfa e Thainá Andreza, por terem me recebido e acolhido em vossa casa e corações.

À Prof^a. Dr^a. Susana Lalic pelos ensinamentos e confiança à mim depositada.

Ao Prof. Dr. Marcio Nascimento, pelo total apoio e dedicação à mim e ao projeto. Sua ajuda foi essencial.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro durante todo o projeto.

Ao corpo docente da UFS pelo trabalho de excelência prestado.

Ao Prof. Dr. Luciano Casarini pelos ensinamentos fundamentais a respeito de inteligência artificial e Python.

Aos meus colegas Erison Fonseca, John Mattias e Iranildo Fontenele pelo companheirismo nas épocas de estudo.

À Clenis Santos, Ronilson Corrêa, Danieli Vitória, minha família, por terem me apoiado e ajudado durante esta aventura.

Em especial ao pequeno beibu, minha noiva, Daniela Cristina da Silva Rodrigues, que se manteve ao meu lado em todos os momentos da nossa vida. Você me inspira e forma a base à qual se encontra a projeção do meu centro de massa.

“O cientista não estuda a natureza porque ela é útil; ele a estuda porque se deleita nela, e se deleita nela porque ela é bela. Se a natureza não fosse bela, não valeria a pena ser conhecida, e se não valesse a pena ser conhecida, a vida não valeria a pena ser vivida.”

(Henri Poincaré)

VITÓRIA, L. S. **APLICAÇÃO DE REDES NEURAIS DO TIPO FUNÇÃO DE BASE RADIAL SIMPLES COM POUCOS NEURÔNIOS NA PREVISÃO DA TEMPERATURA DE TRANSIÇÃO VÍTEA DE BORATOS DE METAIS ALCALINOS**. São Cristóvão: Universidade Federal de Sergipe, 2022. 123 p. Dissertação (mestrado em Física) – Universidade Federal de Sergipe, 2022.

RESUMO

O interesse em aplicações e usos de vidros existe desde o início da civilização. Tais materiais são apreciados por diversas funções e propriedades dada a extensa possibilidade de feitura de composições distintas. É estimado existir cerca de 10^{62} composições vítreas utilizando-se de 80 elementos químicos em concentrações de até 1%, e que apenas foram explorados cerca de 10^5 , o que permite muitas descobertas. Em vista das muitas possibilidades relacionadas há o objetivo de se descrever propriedades em termos de composições, e entre elas a característica temperatura de transição vítrea T_g . A inteligência artificial proposta por Alan Turing em 1950, que hoje se ramificou em processos computacionais conhecidos como aprendizado de máquina e aprendizado profundo, e ramificado a estes, as redes neurais artificiais, que são tidas como ferramentas altamente preditivas e podem ser empregadas na descrição de diversos sistemas físicos. Entre os diversos tipos de redes neurais artificiais destacam-se as redes de funções de base radial. Esta rede neural é caracterizada por efetuar um treinamento efetivo e veloz devido ao seu particular mecanismo de aprendizado, capaz de transformar sistemas complexos em um simples problema de álgebra linear. Redes neurais complexas podem ser criadas para descrever diversos fenômenos; entretanto, quanto mais complexa a rede, mais sobre ajustada aos dados ela tenderá a ser, e uma maneira de contornar isso pode ser via o emprego de certa quantidade de neurônios em sua construção. Diante destes aspectos, foram aplicadas redes de funções de base radial com apenas dois neurônios (ou pólos) para descrição da temperatura de transição vítrea de sistemas boratos de metais alcalinos. Os resultados mostram que a técnica é altamente preditiva, pois os ajustes alcançaram um valor de ajuste R^2 superior a 90%. Foi possível ainda efetuar ajustes e treinamento da rede incluindo o conhecido fenômeno da anomalia do boro. A função de ativação gaussiana se mostrou superior às duas outras, ditas funções multiquadráticas. A localização dos neurônios nas redes foi um ponto em destaque, pois testes evidenciaram uma melhora da performance dos ajustes em até 5% quando este fora manipulado. O comportamento da T_g dos sistemas alcalino boratos em função da concentração molar x de óxidos é similar e corresponde ao modelo topológico de Mauro, Gupta e Loucks (2009), estabelecido apenas para os sistemas boratos de lítio e sódio. Tal trabalho sugere estender o estudo de outras propriedades físicas utilizando de poucos neurônios e associadas ao estado vítreo.

Palavras-chave: Vidro; Temperatura de Transição Vítrea; Alcalino Borato; Função de Base Radial, Aprendizado de Máquina, Inteligência Artificial.

VITÓRIA, L. S. **APPLICATION OF SIMPLE NEURAL NETWORKS FROM RADIAL BASED FUNCTION WITH FEW NEURONS TO PREDICT THE VITREOUS TRANSITION TEMPERATURE OF ALKALINE BORATES.** São Cristóvão: Universidade Federal de Sergipe, 2022. 123 p. Dissertação (mestrado em Física) - Universidade Federal de Sergipe, 2022.

ABSTRACT

Interest in glass applications and uses has existed since the beginning of civilization. Such materials are appreciated for several functions and properties given the extensive possibility of making different compositions. It is estimated that there are about 10^{52} glasses compositions using 80 elements of the periodic table considering minimum concentrations up to 1%, and only 10^5 have been explored, which allows many discoveries. In view of the many possibilities associated, the objective of this work was to define properties in terms of compositions, and among them, the characteristic glass transition temperature. Artificial intelligence proposed by Alan Turing in 1950, which today branched out into computational processes known as machine and deep learnings, as well as artificial neural networks, that are considered highly predictive tools and can be used in the description of several physical systems. Among different types of artificial neural networks, stands out radial basis functions. This neural network is characterized by performing an effective and fast training due to its particular learning mechanism, able to transform complex systems into a simple linear algebra problem. Complex neural networks can be created to describe different phenomena; however, the more complex the network, more overfitted to the data it will tend to be, and a way to avoid this can be via the use of a certain number of neurons in its construction. Given these aspects, radial basis functions networks with only two neurons (or poles) were applied to describe the glass transition temperature of alkali metal borate systems. The results show that the technique it's highly predictive, as data reached R^2 adjustment value over 90%. It was also possible to carry out adjustments and train the network including the known phenomenon of the boron anomaly. The Gaussian activation function proved to be superior to two others, named multiquadratic functions. The neuron location in the networks was a highlight, as the tests showed an improvement in the performance of the adjustments by up to 5% when manipulated. The T_g behavior of alkaline borate systems as a function of molar concentration x of oxides is similar and corresponds to the topological model by Mauro, Gupta and Loucks (2009), established only for lithium and sodium borate systems. Such work suggests extending this study to other physical properties using a few neurons and associated with the vitreous state.

Keywords: Glass; Glass Transition Temperature; Alkaline Borate; Radial Base Function, Machine Learning, Artificial Intelligence.

LISTA DE FIGURAS

- Figura 1.1** — Galeria de alguns cientistas que efetuaram importantes descobertas com materiais vítreos: Galileu Galilei (1564 - 1642, filósofo e físico italiano), Isaac Newton (1643 - 1727, filósofo, físico e matemático inglês), Mikhail Vasilyevich Lomonosov (1711 - 1765, polímata russo) e Michael Faraday (1791 - 1867, físico e químico inglês).....3
- Figura 2.1** — Principais funções de base radial: gaussiana (esquerda), multi-quadrática e multi-quadrática inversa (direita), relacionadas às [Equações \(2.1\)](#), [\(2.2\)](#) e [\(2.3\)](#), respectivamente.....8
- Figura 2.2** — Arquitetura de uma típica rede neural composta por uma camada de entrada (esquerda), camada oculta e camada de saída (direita). Adaptada de [Haykin \(2007\)](#).....9
- Figura 2.3** — Galeria de alguns cientistas que efetuaram importantes descobertas em redes neurais: Warren Sturgis McCulloch (1898 - 1969, psicólogo americano), Walter Harry Pitts Jr. (1923 - 1969, graduado em tecnologia americano), Frank Rosenblatt (1928 - 1971, psicólogo americano) e David Sydney Broomhead (1950 - 2014, químico inglês).....11
- Figura 2.4** — Esquema de Tikhonov transposto para a situação de uma rede neural.....15
- Figura 2.5** — Exemplo do comportamento de uma função de ajuste obtido via rede RBF, associada a um fenômeno genérico, vinculado aos dados da [Tabela 2.1](#), representada pelos pares de pontos (2,3), (3,6) e (4,5).....25
- Figura 3.1** — Super-resfriamento de um líquido e definição de T_g , conforme [Varshneya e Mauro \(2019\)](#).....30
- Figura 3.2** — Dois anéis hexagonais unidos por um oxigênio entre dois átomos de boro em ângulo θ formando uma estrutura de grupo *boroxol* (*i.e.*, a junção de dois anéis boroxol), muito comum em vidros boratos. Em cada anel é possível observar três unidades BO_3 , conforme [Varshneya e Mauro \(2019\)](#).....32
- Figura 3.3** — Diversos grupos postulados para os sistema de vidros alcalino borato alcalino (onde ● indica boro): (A) um anel boroxól; (B) grupo triborato; (C) grupo di-triborato; (D) grupo metaborato; (E) grupo diborato; (F) grupo pentaborato; (G) grupo

di-pentaborato; e (H) grupo tri-pentaborato, conforme Varshneya e Mauro (2019)	33
Figura 4.1 — Histograma da distribuição de T_g dos sistemas alcalino-boratos analisados.....	38
Figura 5.1 — Ajustes R^2 para os sistemas alcalino boratos: potássio, sódio, lítio, rubídio e céσιο.....	48
Figura 5.2 — Dados experimentais de T_g (pontos) e ajuste RBF gaussiano (linha verde) do sistema borato de lítio.....	49
Figura 5.3 — Dispersão residual referente aos dados experimentais de T_g e erro médio calculado no sistema borato de lítio.....	50
Figura 5.4 — Previsão da rede referente ao cálculo de T_g (em K) vinculado ao sistema borato de lítio.....	51
Figura 5.5 — Dados experimentais de T_g (pontos) e ajuste RBF gaussiano (linha verde) do sistema borato de sódio.....	52
Figura 5.6 — Dispersão residual referente aos dados experimentais de T_g e erro médio calculado no sistema borato de sódio.....	54
Figura 5.7 — Previsão da rede referente ao cálculo de T_g (em K) vinculado ao sistema borato de sódio.....	55
Figura 5.8 — Dados experimentais de T_g (pontos) e ajuste RBF gaussiano (linha verde) do sistema borato de potássio.....	56
Figura 5.9 — Dispersão residual referente aos dados experimentais de T_g e erro médio calculado no sistema borato de potássio.....	57
Figura 5.10 — Previsão da rede referente ao cálculo de T_g (em K) vinculado ao sistema borato de potássio.....	58
Figura 5.11 — Dados experimentais de T_g (pontos) e ajuste RBF gaussiano (linha verde) do sistema borato de rubídio.....	59
Figura 5.12 — Dispersão residual referente aos dados experimentais de T_g e erro médio calculado no sistema borato de rubídio.....	60
Figura 5.13 — Previsão da rede referente ao cálculo de T_g (em K) vinculado ao sistema borato de rubídio.....	62
Figura 5.14 — Dados experimentais de T_g (pontos) e ajuste RBF gaussiano (linha verde) do sistema borato de céσιο.....	63
Figura 5.15 — Dispersão residual referente aos dados experimentais de T_g e erro médio calculado no sistema borato de céσιο.....	64

Figura 5.16 — Previsão da rede referente ao cálculo de T_g (em K) vinculado ao sistema borato de cério.....	66
Figura 5.17 — Ajuste obtido para os cinco sistemas alcalino-boratos analisados.....	67
Figura 5.18 — Comparação entre os modelos topológicos de Mauro, Gupta e Loucks (2009) e modelo de redes de função de base radial dos sistemas borato de lítio e sódio.....	69
Figura 5.19 — Quantidade de estruturas formadoras de redes em função da concentração molar de modificadores, de acordo com Mauro, Gupta e Loucks (2009)	70

LISTA DE TABELAS

Tabela 2.1 — Dados genéricos para fins de exemplificação do processo de aprendizado das redes RBF.....	22
Tabela 2.2 — Dados genéricos para fins de exemplificação do processo de aprendizado das redes RBF.....	25
Tabela 4.1 — Conjunto de bibliotecas e módulos utilizados para o desenvolvimento das redes neurais utilizadas.....	41
Tabela 4.2 — Exemplo de aplicação deste trabalho. Dez amostras de entrada para o treinamento da rede neural associada ao sistema borato de rubídio (x em mol %)	42
Tabela 4.3 — Matriz de coordenadas dos polos (neurônios) obtido via método de aleatoriedade.....	43
Tabela 4.4 — Matriz com a coordenada do Pólo 1 modificada.....	43
Tabela 4.5 — Coeficientes obtidos para o sistema borato de rubídio via teoria das redes RBF.....	44
Tabela 5.1 — Valores dos coeficientes determinados a partir das teorias RBF para cada um dos sistemas boratos usando a função gaussiana.....	46
Tabela 5.2 — Ajuste R^2 para os demais tipos de função de ativação neuronal.....	72

LISTA DE SÍMBOLOS

T_g	Temperatura de transição vítrea
\vec{x}_i	Pólos ou centros das funções de base radial
$\varphi(\ \vec{x} - \vec{x}_i\)$	Funções de base radial
$\ \cdot\ $	Norma vetorial
c	Constante
σ	Desvio padrão
s	Mapa de interpolação
R^α	Espaço R de dimensão α
R^1	Espaço R de dimensão 1
Γ	Hipersuperfície de interpolação
\subset	Contido em
$R^{\alpha+1}$	Espaço R de dimensão $\alpha+1$
$F(\vec{x}_i)$	Função de base radial interpolada
y_i	Instâncias de treinamento da rede neural
w_i	Valores dos pesos da rede neural ou sinapses
N	Número de neurônios
Φ	Matriz de interpolação
W	Matriz de pesos
Y	Matriz de instâncias de treinamento
$E(F)$	Função erro do modelo
λ	Parâmetro de regularização ou multiplicador de Lagrange
D	Operador auto adjunto de regularização
L_2	Espaço de Hilbert
$\langle \rangle$	Produto interno entre funções
$dE(F, h)$	Derivada de Frechét
$D^\#$	Adjunto do operador D
$\delta(\vec{x} - \vec{x}_i)$	Função ou distribuição Delta de Dirac
$G(\vec{x}, \vec{x}_i)$	Função de Green do operador $D^\#D$
G	Matriz de Green
$G(\vec{x}_j, \vec{x}_i)$	Elementos da matriz de Green
G^+	Pseudo inversa da matriz de Green

w_0	Bias
Li	Lítio
Na	Sódio
K	Potássio
Rb	Rubídio
Cs	Césio
SiO ₂	Óxido de sílica
B ₂ O ₃	Óxido de boro
T_s	Temperatura de solidificação
T_m	Temperatura de fusão
°C	Grau Célsius
Q_i	Unidades estruturais de coordenação atômica
M	Elementos alcalinos {Li,Na,K,Rb,Cs}
R_x	Função para cálculo de configurações estruturais
N_4	Número de estruturas tetraédricas
f_{NBO}	Fração de oxigênios não ponteantes
S_c	Entropia do sistema
$T_g(0)$	Temperatura de transição vítrea do borato puro
K	Kelvin
P	Número de polos
C	Número de características preditoras
R^2	Coefficiente de determinação
$T_{\text{fusão}}$	Temperatura de fusão
$N(Q_i)$	Número de quantidade estruturais
T_β	Valor máximo da temperatura de transição vítrea associada

SUMÁRIO

1. INTRODUÇÃO	1
1.1 HIPÓTESE	5
1.2 PROPOSTA	5
1.3 OBJETIVOS	5
1.4 RELEVÂNCIA e JUSTIFICATIVA	6
2. REDES DE FUNÇÕES DE BASE RADIAL	7
2.1 EXEMPLOS	21
2.1.1 Uma Variável e Dois Centros.....	21
2.1.2 Três Variáveis e Dois Centros	24
3. DEFINIÇÃO DE VIDRO	28
3.1 TEMPERATURA DE TRANSIÇÃO VÍTEA	28
3.2 VIDROS BORATOS (BREVE DESCRIÇÃO E PROPRIEDADES).....	30
4. MATERIAIS E MÉTODOS	36
4.1 OBTENÇÃO DOS DADOS (SCIGLASS)	36
4.2 JUPYTER NOTEBOOK	37
4.3 DESENVOLVIMENTO DA REDE NEURAL.....	38
5. RESULTADOS E DISCUSSÕES	45
5.1 ANÁLISE COMPARATIVA DOS AJUSTES	66
5.2 TESTE DAS FUNÇÕES DE ATIVAÇÃO MULTIQUADRÁTICA E MULTIQUADRÁTICA INVERSA	71
5.3 ANÁLISE GERAL DOS RESULTADOS EM FUNÇÃO DOS POLOS.....	72
CONCLUSÕES	74
FUTUROS TRABALHOS	75
REFERÊNCIA SBIBLIOGRÁFICAS	76

1. INTRODUÇÃO

Vidros são materiais conhecidos pelas civilizações há bastante tempo e se encontram fora do equilíbrio termodinâmico, além de serem não-cristalinos, pois sua estrutura não periódica está temporariamente congelada, e apresentam o fenômeno de transição vítrea (VARSHNEYA e MAURO, 2019). Há indícios de que tais materiais são um dos mais antigos fabricados pelo homem, sendo inclusive considerado pelo povo egípcio como precioso quando fabricado de maneira natural. De fato, a civilização egípcia sempre tratou os vidros como símbolos de riqueza e poder, pois são objetos disponíveis em tumbas e mesmo nas vestes dos faraós (COSTA, 2006).

Na obra *Historia Naturalis* do naturalista romano Caio Plínio Segundo (c. 23 - 79 d.C.) escrita no ano 77 da nossa era foi relatado o primeiro processo de fabricação artificial de vidros, atribuído ao povo fenício por volta de 7.000 anos a.C. Tempos depois de usarem lareira na praia observaram que surgia um líquido ao redor que se solidificava rapidamente, resultando em um sólido que se assemelhava a uma pedra frágil, de aspecto liso. A partir desta observação, experimentos foram repetidos até o momento em que foi possível obter um material utilizável, originando assim a arte vidreira que foi difundida pelo continente asiático, tornando-se inclusive, símbolo de nobreza aos artistas que a praticavam (ALVES, GIMENEZ e MAZALI, 2001).

O desenvolvimento na fabricação, aplicação e estudos sobre vidros se estende desde o século XVIII, onde o início da produção de 'vidro cristal' foi realizada, até dias atuais, em que processos de fabricação modernos são utilizados, como o processo sol-gel (ALVES, GIMENEZ e MAZALI, 2001); entretanto, o conhecimento científico a respeito teve grande crescimento a partir do século XVIII, com os estudos de Mikhail Vasilyevich Lomonosov (1711 - 1765, russo), Thomas Griffiths (? - c. 1852, inglês) e Michael Faraday (1791 - 1867, inglês) (MAURO; ZANOTTO, 2014), dentre outros. Vale lembrar que os físico-matemáticos Galileu Galilei (1564 - 1642, italiano) e Isaac Newton (1643 - 1727, inglês) efetuaram importantes descobertas com os primeiros telescópios elaborados por eles mesmos ao dominar a arte de polimento de lentes de aumento (NASCIMENTO, 2015).



Galileo Galilei



Is. Newton



Михаило Ломоносовъ



M Faraday

Figura 1.1 — Galeria de alguns cientistas que efetuaram importantes descobertas com materiais vítreos: Galileu Galilei (1564 - 1642, filósofo e físico italiano), Isaac Newton (1643 - 1727, filósofo, físico e matemático inglês), Mikhail Vasilyevich Lomonosov (1711 - 1765, polímata russo) e Michael Faraday (1791 - 1867, físico e químico inglês).

Por volta desta época, o padre e mestre vidreiro italiano Antonio Neri (1576 - 1614) havia escrito o primeiro tratado sobre fabricação de vidros, *L'Arte Vetraria*, em 1612 (NASCIMENTO, 2015). O conhecimento científico a respeito de vidros tem destaque no cenário atual uma vez que o mesmo se encontra em grande escala de aplicações (VARSHNEYA e MAURO, 2019; MAURO, GUPTA e LOUCKS, 2009), que se estendem, por exemplo, desde janelas residenciais, até em processos mais sofisticados, por exemplo transmissão de informação via fibra óptica, lentes oculares medicinais (BOLT, 2017). Sendo assim é de grande impacto na sociedade global ter conhecimento a respeito destes materiais e suas propriedades (MAURO, 2014; JIANG *et al.*, 2020; XEN, 2008).

Nos últimos 6.000 anos estima-se que 200.000 vidros com composições diferentes tenham sido criados, porém a possibilidade de combinações químicas se estendem a um número bem maior, por volta de 10^{52} (ZANOTTO e COUTINHO, 2004). O desenvolvimento de novas combinações e os estudos de suas propriedades associadas têm sido alvo de muitos pesquisadores; contudo, a tentativa de alcançar estes resultados demanda muito esforço, e em geral, um custo elevado no sentido financeiro e temporal (MAURO, 2018), uma vez que processos de tentativa e erro ainda são comumente utilizados.

Existem propostas de modelagem que se empenham em caracterizar o comportamento das propriedades vítreas com o objetivo de realizar descrições e extrapolações destas grandezas. Pode-se citar como exemplos: teoria funcional de densidade, simulações de dinâmica molecular, teoria de restrições topológicas, algoritmos evolucionários e mesmo *Big Data* (CASSAR, CARVALHO e ZANOTTO, 2018; ZHANG e XU, 2020). Neste sentido, ferramentas novas com potenciais incríveis de poder para previsões e análise técnica surgem regularmente, como o aprendizado de máquina e o aprendizado profundo (MONARD e BARANAUSKAS, 2003; LIU *et al.*, 2021); esta, em particular, consiste num ramo da inteligência artificial proposta pelo matemático inglês Alan Mathison Turing (1912 - 1954) em seu artigo *Computing Machinery and Intelligence*, publicado em 1950 (TURING, 1950). Propostas vêm sendo caracterizadas segundo alguns estudos como ferramentas de altíssimo poder de extrapolação em áreas como ciências dos materiais, medicina, engenharias, etc (BESSA, 2020; PILANIA *et al.*, 2013).

O aprendizado profundo é uma técnica baseada em inteligência artificial na qual um computador pode aprender a realizar determinada tarefa sem ser

explicitamente programado para realizá-la (GÉRON, 2019). Essa técnica está atualmente em destaque devido a era da Internet das Coisas (da sigla em inglês *Internet of Things*, IoT) e a crescente ascensão da ciência de dados e análise de *Big Data*, já que possui ampla escala de aplicabilidade (reconstituição de imagens, detecção de anomalias, detecção de fraudes financeiras etc) e, atualmente, já constitui o quarto paradigma da ciência dos materiais, demonstrando assim grande benefício e relevância (AGRAWAL e CHOUDHARY, 2019).

Redes Neurais Artificiais (RNA) são técnicas de aprendizado profundo que se baseiam e inspiram-se no neurônio biológico, proposto originalmente pelo pai da neurociência moderna, o espanhol Santiago Ramón y Cajal (1852 - 1934) (HAYKIN, 2004). As ligações constituídas por neurônios biológicos são denominadas de redes neurais, onde esta realiza comunicação entre os neurônios constituintes através de pulsos elétricos denominados de sinapses. Por tal descoberta, Santiago foi laureado com o prêmio Nobel de Fisiologia ou Medicina em 1906.

Uma rede neural artificial utiliza uma entidade chamada de neurônio artificial, uma ideia pioneira proposta por Warren Sturgis McCulloch (1898 - 1969, psicólogo) e Walter Harry Pitts, Jr. (1923 - 1969, graduado em tecnologia) no seminal artigo *A Logical Calculus of the Ideas Immanent in Nervous Activity* (MCCULLOCH e PITTS, 1943). Baseado em tal artigo, o neurônio artificial foi considerado como parte de um sistema que permite a um algoritmo relativamente determinar os valores sinápticos das ligações entre neurônios que constituem a rede neural associada (WERBOS, 1990). Tal proposta utiliza rotinas de aprendizagem de máquina que minimizam erros através de processos específicos, tais como: *backpropagation*, *least mean square*, etc, formando assim duas características importantes de redes neurais: a sua estrutura e o seu algoritmo (RAUBER, 2005).

Redes neurais, em sua maioria, utilizam funções não lineares para efetuar o mapeamento dos seus dados de entrada e realizar o seu processo de aprendizado na sua etapa de treinamento para buscar um ajuste adequado em relação aos dados, efetivando extrapolações para novas possibilidades. Neste caso, o esquema de aprendizado consiste em uma otimização não linear de mínimos quadrados restritos, na qual a natureza do ajuste não é mostrada de forma explícita.

A proposta de gerar uma rede neural que realize o procedimento de aprendizagem de forma explícita foi idealizada com base na teoria de interpolações multivariadas, elaborada por David Sydney Broomhead (1950 - 2014, químico inglês)

e David Lowe (c. 1953, físico inglês) em 1988, denominada de Rede de Funções de Base Radial ou RBF. A idealização desta proposta é caracterizada por reduzir processos não lineares em um problema de álgebra linear simples e o processo de aprendizado passa a ser de otimização linear de mínimos quadrados, ou também conhecido como método da pseudo inversa ([BROOMHEAD e LOWE, 1988](#); [ALFANTITIS *et al.*, 2004](#), [MELAGRAKI *et al.* 2005](#)).

1.1 HIPÓTESE

A hipótese da dissertação trata da possibilidade de previsão do comportamento da temperatura de transição vítrea de alguns sistemas utilizando de inteligência artificial, mais especificamente de redes neurais artificiais. Uma questão a ser levantada é: seria possível estabelecer tal rede com poucos neurônios e razoável ajuste?

1.2 PROPOSTA

A proposta deste trabalho visa buscar uma solução de ajuste e extrapolações dos valores de temperatura de transição vítrea T_g de composições binárias de vidros alcalino-boratos utilizando uma Rede de Funções de Base Radial. A intenção é tratar o banco de dados de maneira tal que o processo de aprendizado, estrutura e propriedades destas composições estejam bem definidos e explicados, em que os valores das extrapolações obtidas por tal método sejam contrastados com o modelo topológico proposto por [Mauro, Gupta e Loucks \(2009\)](#) para as composições vítreas de boratos de sódio e lítio.

1.3 OBJETIVOS

Este trabalho teve por objetivo prever a temperatura de transição vítrea de alguns sistemas binários em função da composição utilizando de recursos de inteligência artificial como as redes neurais. Em particular, o propósito do trabalho foi efetuar comparações com modelos estruturais de sistemas binários de boratos de sódio e lítio, disponíveis na literatura.

1.4 RELEVÂNCIA e JUSTIFICATIVA

Historicamente, os vidros têm desempenhado um papel crítico no desenvolvimento da humanidade e continuarão a demandar novas soluções para os desafios globais, desde energia e meio ambiente até saúde e tecnologia da informação / comunicação (NASCIMENTO, 2015; VARSHNEYA e MAURO, 2019). Para atender ao ritmo acelerado de novas e modernas tecnologias é necessária uma abordagem mais sofisticada para projetos de novas composições e aplicações que permitam um desenvolvimento de tais novos materiais, de modo mais rápido, barato e eficiente.

A temperatura de transição vítrea, T_g , consiste na propriedade de excelência que caracteriza o estado vítreo (CASSAR, CARVALHO e ZANOTTO, 2018). Dados de diversos sistemas encontram-se disponíveis em base de dados como o SciGlass 7.9, referentes a medições efetuadas ao longo de quase um século. Apesar desta enorme quantidade de dados, como relevância e justificativa para esta dissertação são destacadas a insuficiente compreensão sobre o conhecimento da T_g em termos da composição de diversos sistemas vítreos, um conhecimento crítico em todas as etapas de elaboração de vidros em qualquer indústria (ZANOTTO e MAURO, 2017). Ferramentas de inteligência artificial como as funções de base radial podem prever esta temperatura característica de sistemas vítreos em função da composição de modo preciso, fácil e rápido, além de ser possível caracterizar e prever outras propriedades.

Tal estudo visa contribuir de modo significativo ao conjunto de conhecimentos já existentes sobre o estado vítreo. Desta maneira, os resultados e análises aqui gerados poderão auxiliar em novos estudos de outros fenômenos e propriedades que necessitam da correta estimativa ou ainda aplicação da T_g por meio de ferramentas de inteligência artificial.

2. REDES DE FUNÇÕES DE BASE RADIAL

A formulação das redes RBF deve-se a [Broomhead e Lowe \(1988\)](#), que têm seus trabalhos associados aos estudos de interpolação e aproximação de funções realizados por [Powell \(1987\)](#). Uma forma de interpretação comumente designada a esta técnica é a *solução de um problema de ajuste de funções em um espaço de grande dimensão*, no qual o processo de aprendizado de máquina trata-se, na verdade, da tarefa de encontrar uma superfície multidimensional adequada aos dados de treinamento, ou seja, do ponto de vista estatístico, que melhor ajusta os valores de entrada da rede ([HAYKIN, 2007](#); [POGGIO e GIROSI, 1990](#)).

Nas redes de funções de base radial há uma região conhecida como camada oculta, que é responsável por fornecer uma base de funções apropriadas para que as soluções obtidas possam ser expandidas como uma combinação linear em termos desta base. Aqui, o termo “apropriado” se refere a funções que são simétricas a um centro ou polo \vec{x}_i e apresentam comportamento monotônico (crescente ou decrescente), e são da forma $\varphi(\vec{x}) = \varphi(\|\vec{x} - \vec{x}_i\|)$. Esta família de funções são ditas de base radial ([BISHOP, 1991](#)). Alguns exemplos destes tipos são as gaussianas ([Equação \(2.1\)](#)), multi-quadráticas do engenheiro civil Rolland Lee Hardy (1920 - 2009) ([Equação \(2.2\)](#)) e multi-quadrática inversa de Hardy ([Equação \(2.3\)](#)). Seus comportamentos podem ser encontrados na [Figura 2.1](#) abaixo ([BOTROS e ATKESON, 1991](#)), sendo c e σ constantes:

$$\varphi(\vec{x}) = e^{-\frac{\|\vec{x} - \vec{x}_i\|^2}{2\sigma^2}} \quad (2.1)$$

$$\varphi(\vec{x}) = \sqrt{1 + (c\|\vec{x} - x_i\|)^2} \quad (2.2)$$

$$\varphi(\vec{x}) = \frac{1}{\sqrt{1 + (c\|\vec{x} - x_i\|)^2}} \quad (2.3)$$

A arquitetura de uma rede de funções de base radial é bastante simples, sua composição se dá por três camadas distintas, que desempenham tarefas diferentes. A primeira camada é conhecida como fonte, também chamada de nó de entrada, pois além de iniciar a propagação de informação para as camadas seguintes, é responsável por conectar a rede ao seu ambiente. A segunda camada, assim como nas redes perceptrons,¹ denomina-se camada *oculta* - como já mencionado; que no

¹ Elaborada em 1958 pelo psicólogo americano Frank Rosenblatt (1928 - 1971).

caso das redes de funções de base radial são únicas. Nessa região ocorre uma transformação não linear através da função de ativação neuronal de cada elemento da camada (neurônios artificiais). E por fim, existe a camada de saída. Nessa região, a rede realiza outro processo de transformação, porém, diferente das redes neurais convencionais, a transformação é linear (KRIESEL, 2007). A Figura 2.2 ilustra tal arquitetura.

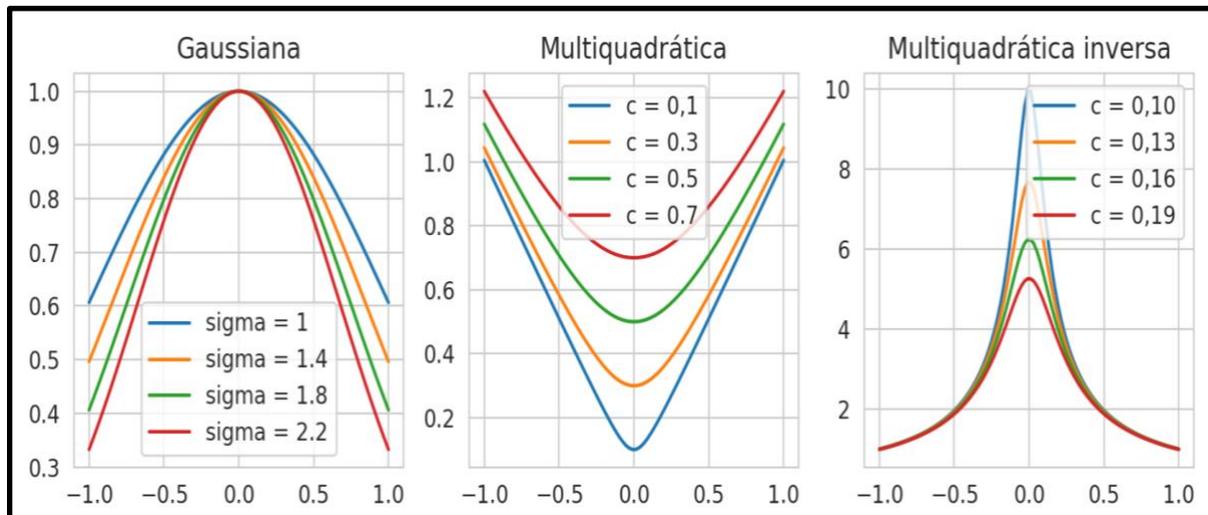


Figura 2.1 — Principais funções de base radial: gaussiana (esquerda), multi-quadrática e multi-quadrática inversa (direita), relacionadas às Equações (2.1), (2.2) e (2.3), respectivamente.

Estas redes têm os padrões de funcionamento mencionados baseados em um conceito global do aprendizado de máquina, chamado de Teorema da Separabilidade, do físico americano Thomas Merrill Cover (1938 - 2012), muito relacionado a processos e métodos de *kernels* não lineares. Em resumo, o objetivo visa simplificar o problema de classificação de padrões realizando uma transformação dos vetores de entrada não linearmente separáveis, para um problema linearmente separável utilizando um mapeamento do espaço oculto (espaço das funções $\varphi(\vec{x})$) em um novo espaço de dimensões apropriadas, ou seja, este mecanismo possibilita realizar um tratamento do espaço vetorial permitindo operações lineares de maneira a simplificar o processo de aprendizado e conseqüentemente a generalização desejada (HOWLETT e LAKHMI, 2013; COVER, 1965; BROOMHEAD e LOWE, 1988).

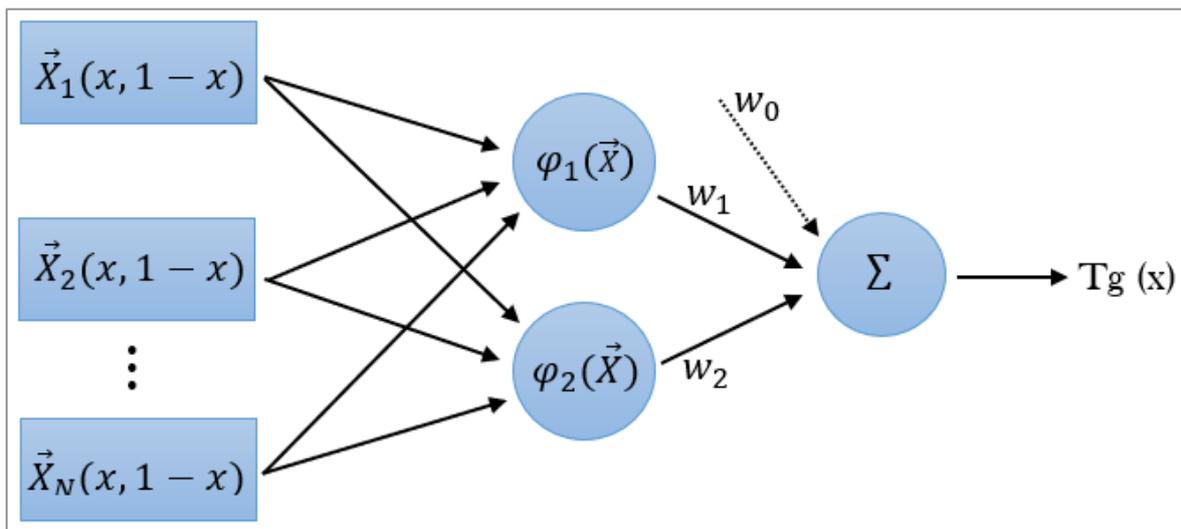


Figura 2.2 — Arquitetura de uma típica rede neural de funções de base radial composta por uma camada de entrada (esquerda), camada oculta e camada de saída (direita). Adaptada de Haykin (2007) para o presente trabalho.

Haykin (2007) considerou que quanto maior a dimensão do espaço oculto melhor será o processo de generalização e, como espera-se que a função de ajuste obtida seja suave, a aproximação desejada estará diretamente ligada a dimensão do espaço de $\varphi(\vec{x})$ (MHASKAR, 1996; NIYOGI e GIROSI, 1996).

Uma forma de entender as consequências do teorema de Cover é mostrado pela Equação (2.4), no qual sua interpretação está associada a uma rede de dimensão α no espaço de entrada, uma camada oculta, e um único neurônio de saída, sendo que s representa o mapa de interpolação. Segundo Haykin (2007), o fato da saída da rede ser unidimensional não ocasiona perda de generalidade.

$$s: R^{\alpha} \rightarrow R^1 \quad (2.4)$$

Para Broomhead e Lowe (1988) o mapa s é uma hipersuperfície $\Gamma \subset R^{\alpha+1}$ até então desconhecida. Para os mesmos, a fase de treinamento tem como principal objetivo ajustar a superfície Γ com base nos dados de entrada da rede e por seguinte, a etapa de generalização, em que o foco é utilizar o processo de interpolação entre os dados e a superfície Γ otimizada no sentido de semelhança à superfície original.



War



Walter Pitts



Frank Rosenblatt



DSB

Figura 2.3 – Galeria de alguns cientistas que efetuaram importantes descobertas em redes neurais: Warren Sturgis McCulloch (1898 - 1969, psicólogo americano), Walter Harry Pitts Jr. (1923 - 1969, graduado em tecnologia americano), Frank Rosenblatt (1928 - 1971, psicólogo americano) e David Sydney Broomhead (1950 - 2014, químico inglês).

O ponto que emerge até então é de fato a forma e o processo de solução para determinação de Γ , e para isto é necessário utilizar o conceito de interpolação exata. De maneira genérica, o problema consiste em interpolar um conjunto de N vetores de entrada $\vec{x}_1, \vec{x}_2, \vec{x}_3, \dots, \vec{x}_N$ de dimensões desprezíveis no âmbito de generalização, como ressalta Haykin (2007). Este processo pode ser condicionado e expresso através da Equação (2.5):

$$F(\vec{x}_i) = y_i, i = 1, 2, 3 \dots N \quad (2.5)$$

Segundo DU e SWAMY (2006), funções de base radial surgem como soluções de uma equação não homogênea através da obtenção da função de Green² de um operador auto adjunto associado ao estabilizador do funcional de custo, proveniente do processo de regularização da rede (mais a frente este processo será mostrado e justificado em detalhes). Essas definições são importantes pois carregam consigo importantes propriedades e aspectos físicos que dizem respeito a forma de solução do problema, e uma delas é que tais soluções são autofunções do operador, que neste caso é auto adjunto, e por isso formam uma base no espaço vetorial; sendo assim toda e qualquer função pode ser escrita como uma combinação linear em termos dos vetores de base (BUTKOV, 1988; DUDLEY, 1994; STAKGOLD, 2000). Como consequência, pode-se definir o problema de interpolação de funções de base radial através da Equação (2.6) (POWELL, 1987; PARK e SANDBERG, 1993):

$$F(\vec{x}) = \sum_{i=1}^N w_i \varphi_i(\|\vec{x} - \vec{x}_i\|). \quad (2.6)$$

onde φ_i são funções de base radial determinadas pela distância (comumente euclidiana, mas não genérica) do vetor \vec{x} em relação ao vetor centróide ou polo \vec{x}_i , onde $\vec{x}_i \in R^\alpha$, N é o número de neurônios na camada oculta da rede. O objetivo agora é resolver as equações simultâneas resultantes da combinação das Equações (2.5) e (2.6) para determinação dos coeficientes da combinação linear w_i , que em forma matricial pode ser representada por:

² Em homenagem ao matemático inglês George Green (1793 - 1841).

$$\begin{pmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1N} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{31} & \varphi_{32} & \cdots & \varphi_{3N} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \quad (2.7)$$

que pode ser reescrita em termos representativos como:

$$\Phi W = Y \quad (2.8)$$

onde Φ é chamada de matriz de interpolação, cujos elementos são $\varphi_{ij}, j, i = 1, 2, \dots, N$. $W = (w_1, w_2, \dots, w_N)^T$ é o vetor peso que representa os coeficientes da expansão da [Equação \(2.6\)](#) que, em termos da rede neural, correspondem às sinapses. O vetor Y equivale às respostas utilizadas na fase de treinamento através das instâncias rotuladas, onde $Y = (y_1, y_2, \dots, y_N)^T$. A solução para os coeficientes da expansão pode ser determinada então por:

$$W = \Phi^{-1}Y \quad (2.9)$$

O ponto importante a se destacar é que a matriz de interpolação Φ deve ter determinante diferente de zero, e ser quadrada para que ocorra a inversão, ou seja, é inversível se e somente se Φ for não singular.

A garantia para não singularidade da matriz de interpolação é trabalhada pelo teorema de Micchelli,³ e para as funções gaussianas, multi-quadráticas e multi-quadráticas inversas de Hardy, representadas pela [Figura 2.1](#), a condição necessária para inversão, é que os centros $\vec{x}_i, i = 1, 2, 3, \dots, N$ associados a estas, devem ser distintos, fazendo com que a condição de interpolação seja validada. Logo, o vetor peso W pode ser determinado através da [Equação \(2.9\)](#) (MICCHELLI, 1986).

Cada uma das funções citadas é robusta para modelagem de sistemas distintos, porém, possuem características diferentes e isso naturalmente reflete nas propriedades da matriz de interpolação que, sem muita surpresa, está associada à função de saída. Por exemplo, para [Haykin \(2010\)](#) e [Powell \(1987\)](#), as funções multi-quadráticas de Hardy possuem uma maior facilidade de obter uma função saída

³ Em homenagem a Charles Anthony Micchelli (n. 1942), matemático americano.

suave do que suas irmãs de base radial, pois crescem ao infinito; isso indica que o conhecimento prévio do sistema a ser estudado garante maior performance no treinamento da rede, que impacta diretamente na generalização.

Até então apresentamos o formalismo necessário para fazer com que a rede neural realize seu treinamento via teoria de interpolação, mas pouco foi escrito sobre a qualidade da generalização. Então, para que se possa ter um melhor entendimento, é interessante retornar a premissa de que o ajuste otimizado da hipersuperfície Γ é aquele que captura todos os pontos de treinamento através de N funções de base radial, ou vetores de base, nos quais, segundo [Haykin \(2010\)](#), quanto maior a dimensão dessa base, (quanto mais funções de base radial forem usadas), melhor será o ajuste através de Γ . Como consequência, tem-se um problema clássico deste ambiente que é o *overfitting* ou sobre ajuste dos dados, que resulta em baixa generalização devido à presença inevitável de ruídos nas amostras fornecidas à rede que acabam por compor a função de saída.

Pode-se observar que até o presente momento desta dissertação há um problema com o processo de generalização da rede que precisa ser resolvido para que o modelo passe a ser válido. Para que esse objetivo seja cumprido, é necessário ter um olhar qualitativo da forma pela qual a rede está “aprendendo” e cuja resposta é: “realizando um mapeamento do espaço de entrada multidimensional para obter uma função saída”. Essa situação leva ao ambiente de modelagem matemática computacional, no qual existem algumas formulações que tratam da análise de classificação de problemas através das condições estabelecidas pelo matemático francês Jacques Salomon Hadamard (1865 - 1963). Estas condições classificam problemas, em geral, de duas maneiras: problemas *mal* postos e problemas *bem* postos. Estas condições são três e podem ser expressas como:

1. **Existência:** Existe solução;
2. **Unicidade:** Existe uma única solução;
3. **Continuidade** ou **Estabilidade:** A solução tem uma dependência contínua dos vetores de entrada.

Um problema é dito bem posto se todas as condições de Hadamard forem satisfeitas. Para qualquer ausência destas condições o problema é dito mal posto. No sentido qualitativo, um problema mal posto significa que um conjunto de dados,

mesmo que extenso, pode não apresentar informação relevante ao mapeamento da hipersuperfície Γ e, conseqüentemente, influenciará na saída desejada de maneira negativa.

Com base nisso, [Haykin \(2010\)](#) formulou o problema do aprendizado das redes de funções de base radial através do mapeamento de uma hipersuperfície em um espaço multidimensional como um problema mal posto. Ou seja, segundo o mesmo, “um problema mal formulado” pelas seguintes razões: não há a garantia de que para qualquer valor do vetor de entrada \vec{x} exista uma única saída distinta y_i . Isso é facilmente entendível, pois para algumas classes de fenômenos físicos um sistema posto sobre as mesmas condições pode apresentar medidas laboratoriais distintas, como por exemplo, a temperatura de transição vítrea; em outras palavras, isso significa que, localmente, para um certo valor de concentração molar de algum elemento, pode haver distintos valores de temperatura de transição. Um outro aspecto que inclusive já foi mencionado anteriormente é a falta de representatividade das amostras de treinamento, ou seja, pode haver dados que não carregam qualquer informação do sistema físico real (pode ser um simples erro) e como consequência disso, a condição de unicidade de Hadamard é facilmente violada.

O interesse então é contornar esses aspectos dúbios através da transformação do problema mal posto em um problema bem posto. A maneira sugerida por [Haykin \(2010\)](#) se dá por meio do processo de regularização idealizado em 1963 pelo matemático russo Andrey Nikolayevich Tikhonov (1906 - 1993) em seu artigo “*On the Solution of Ill-posed Problems and the Method of Regularization*” cuja tradução é: “Sobre a Solução de Problemas Mal Postos e o Método de Regularização”. A ideia central deste trabalho visa determinar uma solução aproximada mais suave e compatível com as amostras fornecidas levando-se em conta uma quantidade de ruído, adicionando um termo perturbativo na função erro do sistema. [Haykin \(2007\)](#) introduziu isso no ambiente das redes de funções de base radial e afirmou que o objetivo local é “*estabilizar a solução por meio de algum funcional não negativo que incorpore informação prévia sobre a solução*” como ilustra a [Figura 2.4](#) abaixo:

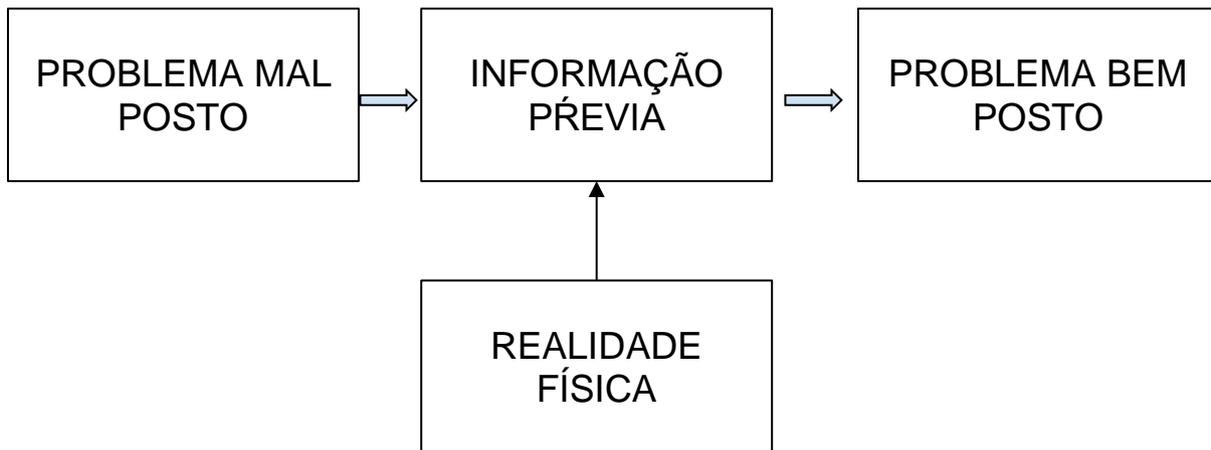


Figura 2.4 – Esquema de Tikhonov transposto para a situação de uma rede neural.

Em situações práticas os processos de aprendizado de máquina e análise de dados lida com um conjunto finito de pares de vetores entrada-saída e isso no geral não consegue de forma exata ou até mesmo, adequada, representar o sistema físico associado. Sendo assim, o problema passa a ser o de minimizar através de um funcional a função de erro associada, que pode ser expressa através da [Equação \(2.10\)](#):

$$E(F) = \frac{1}{2} \sum_{i=1}^N [F(\vec{x}_i) - y_i]^2 + \frac{1}{2} \lambda \|DF\|^2 \quad (2.10)$$

onde o primeiro termo depois da igualdade representa o erro padrão, ou seja, é a medida da distância entre o valor previsto e a instância rotulada fornecida à rede. O segundo termo é chamado de termo regularizador pois está associado ao operador D e este por sua vez é quem carrega a informação prévia sobre o sistema para que haja a transformação para um problema bem posto.

Caso ainda exista confusão sobre o operador D carregar as informações prévias a respeito do sistema, uma nota neste momento talvez seja útil: em problemas de eletromagnetismo, mecânica quântica, mecânica clássica, etc. muitas vezes o sistema estudado está associado a um problema do operador Sturm-Liouville,⁴ onde as auto funções deste operador carregam, de maneira intrínseca, informações a respeito da simetria do sistema; exemplos clássicos consistem na base vetorial

⁴ Em homenagem aos matemáticos franceses Jacques Charles François Sturm (1803 - 1855) e Joseph Liouville (1809 - 1882).

formada pelos polinômios de Adrien-Marie Legendre (1752 - 1833, matemático francês), que estão intimamente ligados a simetria esférica, ou ainda os polinômios de Friedrich Wilhelm Bessel (1784 - 1846, astrônomo alemão), que estão associados a simetria cilíndrica. Ambos são problemas do operador de Sturm-Liouville e este carrega previamente informações a respeito da simetria geométrica do sistema analisado (ARFKEN, 1999) através do operador laplaciano ∇^2 em coordenadas esféricas e cilíndricas respectivamente. Essa mesma contextualização é admitida verdadeira para o operador em questão D , já que nele estão contidas as informações dos pares de vetores entrada-saída da rede. Um outro aspecto deste operador é que ele estabiliza a solução, ou seja, a torna suave indo em favor da condição de continuidade de Hadamard (HAYKIN, 2007).

Ainda sobre a Equação (2.10) o símbolo $\|\cdot\|$ representa a norma associada ao espaço de operações em que DF está contido; este espaço é um conjunto completo *normado*, denominado de espaço de Banach. Mais precisamente Haykin (2010) o denominou de espaço L_2 (um espaço de Hilbert)⁵ pois estas funções são de quadrado integráveis e obedecem a seguinte relação de produto interno:

$$\langle f|g \rangle = \int f(x)g(x)dx \quad (2.11)$$

sendo que vale lembrar que qualquer espaço de Hilbert é um espaço de Banach⁶ (BOLDRINI, 1980).

Neste momento chegamos a um ponto onde o trabalho a ser desenvolvido é o de minimizar a função erro da Equação (2.10). A maneira algébrica comum para esse tipo de tarefa é efetuar a diferenciação do termo a ser minimizado e igualar este mesmo a zero. Deve-se ter cautela neste momento pois como mencionado anteriormente, o operador que estabiliza a função $E(F)$ está em um espaço normado de Banach (ou espaço L_2 de Hilbert se preferível); sendo assim as derivadas apropriadas para esta situação são ditas Derivadas de Fréchet (AGUIAR, FIGUEIREDO e HAVEROTH, 2017).⁷

⁵ Em homenagem ao matemático alemão David Hilbert (1862 - 1943).

⁶ Em homenagem ao matemático polonês Stefan Banach (1892 - 1945).

⁷ Uma curiosidade interessante é que o matemático francês Maurice René Fréchet (1878 - 1973), idealizador de tais derivadas, foi aluno de ninguém menos do que Jacques Salomon Hadamard (1869 - 1963) no ensino secundário na escola Lyceé Buffon em Paris, onde a relação dos dois se prolongou mesmo após Hadamard se tornar professor na Universidade de Bordeaux. Hadamard enviava problemas matemáticos a Fréchet de

Voltando ao objetivo principal, este método seria então o equivalente a realizar uma derivada ordinária de primeira ordem no funcional em questão, mas, ao invés de se estar aproximando uma curva em uma determinada região específica através da tangente associada, busca-se um aproximador local; ou através de uma perspectiva diferente, estamos aplicando o princípio variacional através do funcional em questão que minimiza a [Equação \(2.10\)](#) ([DEBNATH e MIKUSIMKI, 1990](#)).

$$dE(F, h) = \frac{d}{d\beta} [E(F + \beta h(x))]_{\beta=0} \quad (2.12)$$

Seguindo a premissa de que estamos aplicando, o método variacional em nosso problema para minimizar o erro da rede através de funcional para que se obtenha uma função que seja estacionária em seus extremos de maneira similar ao teorema de Fermat em cálculo. É esperado então que isso resulte em uma equação bastante conhecida e de alta aplicabilidade na física, denominada Equação de Euler-Lagrange.⁸ O procedimento para obter tal resultado é a aplicação direta da [Equação \(2.12\)](#) em [\(2.10\)](#). A maneira algébrica e explícita deste procedimento pode ser encontrada em [Haykin \(2010, p. 296 a 299\)](#), e uma maneira alternativa em [Giroi, Jones e Poggio \(1995\)](#), que convergem à um resultado similar como se mostra abaixo:

$$dE(F, h) = D^{\#}DF - \lambda^{-1} \sum_{i=1}^N (F(\vec{x}_i) - y_i) \delta(\vec{x} - \vec{x}_i) = 0$$

$$D^{\#}DF = \lambda^{-1} \sum_{i=1}^N (F(\vec{x}_i) - y_i) \delta(\vec{x} - \vec{x}_i) \quad (2.13)$$

onde $D^{\#}$ é o adjunto do operador diferencial D , $\delta(\vec{x} - \vec{x}_i)$ é a distribuição delta de Dirac, ou função delta de Dirac, λ segundo [Haykin \(2007\)](#) é o parâmetro de regularização, mas aos entusiastas pode-se afirmar que o termo representa o multiplicador de Lagrange associado.

maneira regular e alguns destes assuntos fizeram parte de sua trajetória ao doutorado ([O'CONNOR e ROBERTSON, 2005](#)).

⁸ Em homenagem aos matemáticos Leonhard Euler (1707 - 1783, suíço) e Joseph-Louis Lagrange (1736 - 1813, franco-italiano)

À primeira vista, a [Equação \(2.13\)](#) mostra-se difícil para ser resolvida, mas com um olhar mais atento e obviamente, aos versados em equações diferenciais, estamos em frente à uma expressão similar a equação de Helmholtz⁹ (que inclusive é um problema do operador de Sturm-Liouville) para o caso em que o sistema físico possui fontes perturbativas ao mesmo, como por exemplo, densidade de cargas. É um caso comumente encontrado nos estudos de propagação de sinais, antenas e similares ([DUDLEY, 1994](#)). Para este tipo de equação é comumente empregado uma técnica de integração para solução, como a transformada de Fourier,¹⁰ ou Transformada de Fourier-Bessel, também conhecido como transformada de Hankel,¹¹ que quando utilizado, o trabalho se transforma em encontrar a função de Green associada ao operador auto adjunto $D^\# D$ ([MACHADO, 2004](#)).

Neste momento estamos prestes a resolver o problema de regularização da rede neural. A maneira algébrica de solução neste momento não se torna prioridade pois a interpretação física da [Equação \(2.13\)](#) é suficiente, e uma destas interpretações é que estamos tentando achar uma $F(\vec{x})$ que está vinculada a uma perturbação ou um pulso unitário $\delta(\vec{x} - \vec{x}_i)$; em outras palavras, $F(\vec{x})$ é a resposta do sistema perante uma perturbação. Diante disto, a solução nada mais é do que fazer uma sobreposição de N pulsos unitários dentro do domínio da função, que de maneira discreta e em termos das funções de Green pode ser determinado por:

$$F(\vec{x}) = \sum_{i=1}^N w_i G(\vec{x}, \vec{x}_i) \quad (2.14)$$

onde $w_i = \lambda^{-1}(y_i - F(\vec{x}_i))$, e a função G obedece a seguinte equação homogênea $D^\# D G(\vec{x}, \vec{x}_i) = 0$. Para todo $\vec{x} \neq \vec{x}_i$ a função de Green não apresenta singularidade. Esta classe de funções apresenta diversas propriedades e aplicações em ambientes de Física, Engenharias, Matemática, sendo que as seguintes referências contém exemplos e explicações a respeito: [Dudley \(1990\)](#), [Duffy \(2015\)](#), [Jackson \(1975\)](#), [Arfken \(1999\)](#).

⁹ Em homenagem ao médico e professor de física alemão Hermann Ludwig Ferdinand von Helmholtz (1821 - 1894).

¹⁰ Em homenagem ao matemático francês Jean-Baptiste Joseph Fourier (1768 - 1830).

¹¹ Em homenagem ao matemático alemão Hermann Hankel (1839 - 1873).

O objetivo neste momento é determinar quem são os coeficientes de expansão da [Equação \(2.14\)](#), ou seja, as sinapses da rede. Uma maneira apresentada por [Haykin \(2010\)](#) é fazer com que λ seja suficientemente grande para fazer com que a matriz $(G + \lambda I)$ seja positiva, logo, inversível. Vale lembrar que G é denominada matriz de Green cujos os i,j -ésimos elementos são $G(\vec{x}_j, \vec{x}_i)$ e obedecem a relação de simetria funcional $G(\vec{x}_j, \vec{x}_i) = G(\vec{x}_i, \vec{x}_j)$ e matricial $G^T = G$, sendo I é a matriz identidade de dimensão N por N . Desta maneira a solução para os coeficientes sinápticos são:

$$W = (G + \lambda I)^{-1}Y \quad (2.15)$$

Como nota-se na [Equação \(2.14\)](#) a Equação de Green associada não depende da distância entre os vetores \vec{x} e \vec{x}_i que é uma condição primordial na teoria RBF. A forma para que esta condição seja satisfeita é impor que o operador auto adjunto $D^{\#}D$ seja invariante à translação e à rotação; sendo assim a Função de Green associada ao estabilizador terá dependências da norma da diferença dos vetores \vec{x} e \vec{x}_i . Com isso a solução pode ser representada pela [Equação \(2.16\)](#) ([Haykin, 2007](#)):

$$F(\vec{x}) = \sum_{i=1}^N w_i G(\|\vec{x} - \vec{x}_i\|) \quad (2.16)$$

Considere como exemplo que o estabilizador da [Equação \(2.10\)](#), ou seja, o termo que estabiliza a solução seja dado de uma maneira que o operador auto adjunto associado seja da forma da [Equação \(2.17\)](#):

$$D^{\#}D = \frac{(-1)^k \nabla^{2k}}{k! 2^k} \quad (2.17)$$

cuja Função de Green associada é a gaussiana multivariada, [Equação \(2.18\)](#):

$$G(\|\vec{x} - \vec{x}_i\|) = \exp\left(\frac{-1}{2\sigma_i^2} \|\vec{x} - \vec{x}_i\|^2\right) \quad (2.18)$$

e com isso, uma forma específica da solução através da aplicação é através da combinação entre a [Equação \(2.18\)](#) e [Equação \(2.16\)](#), resultando em:

$$F(\vec{x}) = \sum_{i=1}^N w_i \exp\left(\frac{-1}{2\sigma_i^2} \|\vec{x} - \vec{x}_i\|^2\right) \quad (2.19)$$

A [Equação \(2.19\)](#) representa uma maneira otimizada para o processo de generalização; entretanto, se torna uma quantidade dificilmente aplicável em termos computacionais, pois [Haykin \(2007\)](#) a denomina proibitiva e custosa. O motivo para isso é a dimensão do espaço oculto associado, que nesse caso ficou atrelado ao número de dados utilizados no processo de treinamento pela garantia de que a matriz Φ do processo de interpolação e a matriz G do processo de regularização sejam quadradas; ou seja, para cada instância fornecida à rede existe um vetor x_i associado, que em uma outra representação indica que para cada ponto utilizado para treinamento há um neurônio associado.

Geralmente é evitado utilizar um número muito grande de neurônios em qualquer modelo de rede neural, já que haveria uma maior probabilidade de sobreajuste aos dados ([SRIVASTAVA et al., 2014](#)). Neste sentido, nos encontramos em uma situação em que gostaríamos de encontrar uma solução generalizada no qual o número de neurônios utilizados não necessariamente seja da mesma ordem dos números de instâncias utilizadas no treinamento, e como consequência, determinar a solução para a matriz inversa de G que devido a não dependência do número de neurônios e a quantidade de dados, torna-se singular.

Para a generalização das redes RBF, ou seja, uma solução em termos de uma matriz G genérica e singular, foi mostrado por [Broomhead e Lowe \(1988\)](#) que a condição necessária seria fazer com que o parâmetro de regularização λ da [Equação \(2.10\)](#) tendesse a zero, representando assim um problema bastante conhecido na matemática, denominado de ajuste de funções por mínimos quadrados. O procedimento algébrico destas condições pode ser encontrado em [Haykin \(2010\)](#).

Com o método dos mínimos quadrados, a solução generalizada para os coeficientes sinápticos da rede não se dá mais em função da inversa da matriz de Green de dimensões N por N , mas sim em função da pseudo inversa da matriz de

Green n por m , também conhecida como matriz de Moore-Penrose, que pode ser representada por:

$$G^+ = (G^T G)^{-1} G^T \quad (2.20)$$

desta forma a matriz de pesos pode ser determinada por:

$$W = G^+ Y \quad (2.21)$$

e a função de saída para o caso do operador na [Equação \(2.17\)](#) pode ser representada por:

$$F(\vec{x}) = \sum_{i=1}^N w_i \exp\left(\frac{-1}{2\sigma_i^2} \|\vec{x} - \vec{x}_i\|^2\right) + w_0 \quad (2.22)$$

onde w_0 é o termo de *bias* da rede, e w_i são os i -ésimos elementos da matriz W generalizada.

2.1 EXEMPLOS

A seguir são apresentadas aplicações das redes de funções de base radial para uma variável e dois centros e três variáveis e dois centros.

2.1.1 Uma Variável e Dois Centros

Suponha que estamos de posse de um conjunto de dados da forma exposta pela [Tabela 2.1](#) que está associada a um fenômeno físico genérico, e que o nosso objetivo seja extrair informação deste conjunto para aproximar uma função representativa do fenômeno utilizando a metodologia das redes de funções de base radial. A arquitetura adotada utiliza dois neurônios na camada oculta com função de ativação gaussiana multivariada, e um neurônio na camada de saída. O neurônio um está centrado em $x_1 = 2$ e o neurônio dois está centrado em $x_2 = 4$.

Nº de Instâncias	x	Y
------------------	-----	-----

1	2	3
2	3	6
3	4	5

Tabela 2.1 - Dados genéricos para fins de exemplificação do processo de aprendizado das redes RBF.

Com base nas informações citadas acima podemos utilizar a [Equação \(2.22\)](#) para resolver o problema em termos das sinapses w_i . Ou seja, podemos ajustar função de saída como se mostra abaixo:

$$F(x) = \sum_{i=1}^2 w_i \exp\left(\frac{-1}{2\sigma_i^2} \|x - x_i\|^2\right) + w_0 = y_j \quad (2.23)$$

onde $\sigma_1 = \sigma_2 = 1$ é o desvio padrão dos dados, e y_j são as saídas fornecidas à rede. O objetivo agora é resolver as equações simultâneas resultantes da [Equação \(2.23\)](#), sendo que estas equações são representadas pelo seguinte sistema:

$$\begin{cases} w_1 \exp(-0,5(2-2)^2) + w_2 \exp(-0,5(2-4)^2) + 1w_0 = 3 \\ w_1 \exp(-0,5(3-2)^2) + w_2 \exp(-0,5(3-4)^2) + 1w_0 = 6 \\ w_1 \exp(-0,5(4-2)^2) + w_2 \exp(-0,5(4-4)^2) + 1w_0 = 5 \end{cases} \quad (2.24)$$

que resultam em:

$$\begin{cases} 1,0000w_1 + 0,1353w_2 + 1,000w_0 = 3 \\ 0,6065w_1 + 0,6065w_2 + 1,000w_0 = 6 \\ 0,1353w_1 + 1,0000w_2 + 1,000w_0 = 5 \end{cases} \quad (2.25)$$

que de maneira matricial pode-se representar por:

$$GW = Y, \quad (2.26)$$

onde

$$G = \begin{pmatrix} 1 & 0,1353 & 1 \\ 0,6065 & 0,6065 & 1 \\ 0,1353 & 1 & 1 \end{pmatrix}, W = (w_1 \quad w_2 \quad w_0)^T \text{ e } Y = (3 \quad 6 \quad 5)^T \quad (2.27)$$

G é definida como a matriz de Green associada ao problema, W é a matriz de peso cujos elementos representam as sinapses, e Y são os rótulos das instâncias fornecidas à rede. O trabalho agora é resolver o problema em termos da matriz de pesos definindo seus coeficientes w_i , e para isso basta utilizar as [Equações \(2.20\)](#) e [\(2.21\)](#) para encontrar a matriz pseudo inversa de G e determinar os valores dos coeficientes de expansão. O procedimento para isso é mostrado abaixo:

$$G^T G = \begin{pmatrix} 1 & 0,6065 & 0,1353 \\ 0,1353 & 0,6065 & 1 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0,1353 & 1 \\ 0,6065 & 0,6065 & 1 \\ 0,1353 & 1 & 1 \end{pmatrix}$$

$$G^T G = \begin{pmatrix} 1,3862 & 0,6386 & 1,7419 \\ 0,6386 & 1,3862 & 1,7419 \\ 1,7419 & 1,7419 & 3,000 \end{pmatrix}$$

$$(G^T G)^{-1} = \begin{pmatrix} 248,9582 & 247,6207 & -288,3246 \\ 247,6207 & 248,9592 & -288,3246 \\ -288,3246 & -288,3246 & 335,1485 \end{pmatrix}$$

$$(G^T G)^{-1} G^T = \begin{pmatrix} 248,958 & 247,620 & -288,324 \\ 247,620 & 248,959 & -288,324 \\ -288,324 & -288,324 & 335,148 \end{pmatrix} \begin{pmatrix} 1 & 0,6065 & 0,1353 \\ 0,1353 & 0,6065 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$G^+ = (G^T G)^{-1} G^T = \begin{pmatrix} -5,8546 & 12,8657 & -7,0111 \\ -7,0111 & 12,8657 & -5,8546 \\ 7,8084 & 14,6069 & 7,8085 \end{pmatrix} \quad (2.28)$$

Combinando as [Equações \(2.21\)](#) e [\(2.28\)](#) obtém-se:

$$W = G^+ Y = \begin{pmatrix} -5,8546 & 12,8657 & -7,0111 \\ -7,0111 & 12,8657 & -5,8546 \\ 7,8084 & 14,6069 & 7,8085 \end{pmatrix} \begin{pmatrix} 3 \\ 6 \\ 5 \end{pmatrix} = \begin{pmatrix} 24,5749 \\ 26,8870 \\ -25,2138 \end{pmatrix}. \quad (2.29)$$

Aplicando o resultado da [Equação \(2.29\)](#) em [\(2.23\)](#), conclui-se que:

$$F(x) = 24,5749e^{-\frac{(x-2)^2}{2}} + 26,8870e^{-\frac{(x-4)^2}{2}} - 25,2138 \quad (2.30)$$

A [Equação \(2.30\)](#) representa um ajuste de curva para um fenômeno genérico via teoria de redes de funções de base radial, utilizando dois neurônios cujas coordenadas dos centros são unidimensionais. Seu comportamento pode ser encontrado na [Figura 2.5](#) abaixo:

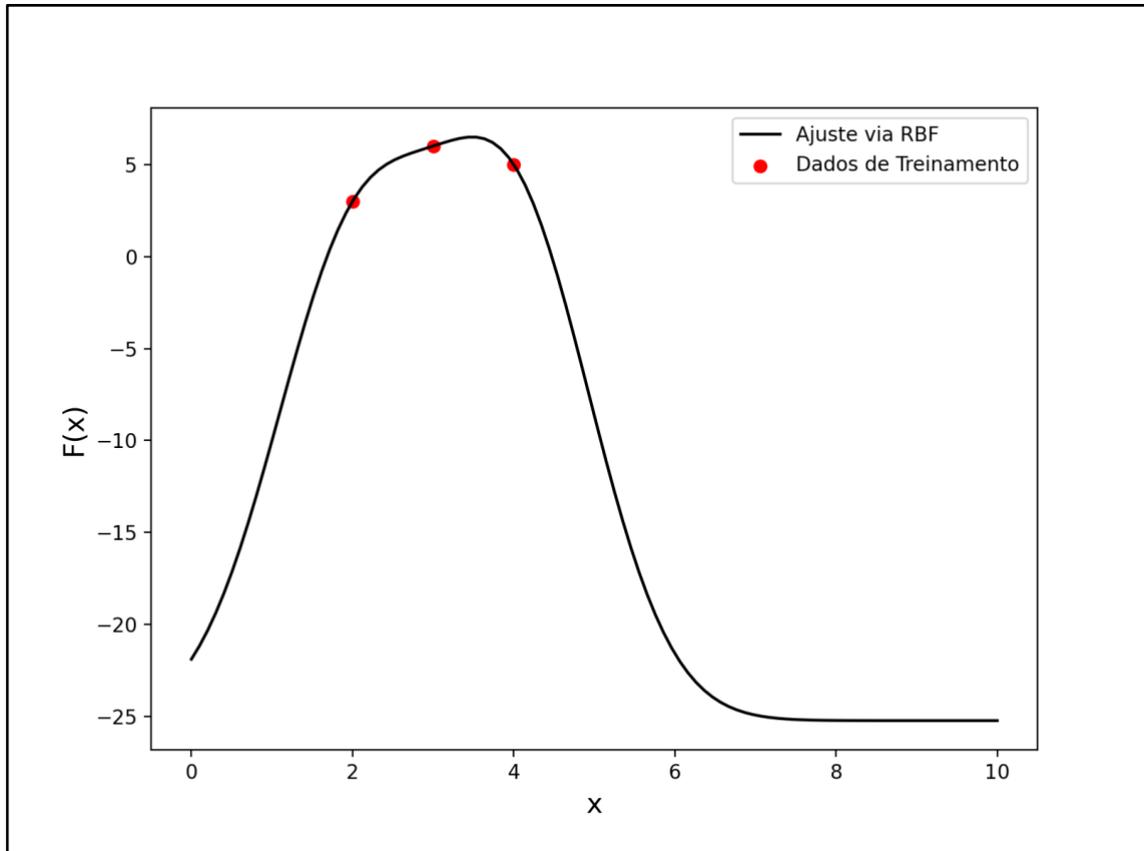


Figura 2.5 – Exemplo do comportamento de uma função de ajuste obtido via rede RBF, associada a um fenômeno genérico, vinculado aos dados da [Tabela 2.1](#), representada pelos pares de pontos (2,3), (3,6) e (4,5).

2.1.2 Três Variáveis e Dois Centros

Vamos considerar agora que estejamos em posse de um novo conjunto de dados de três variáveis e que tenhamos interesse de ajustar uma função correspondente através da teoria de rede de funções de base radial, utilizando apenas dois neurônios na camada oculta. Este conjunto de dados se apresenta como o mostrado na [Tabela 2.2](#) abaixo:

Nº de Instâncias	x	Y
------------------	-----	-----

1	(1,1,1)	2,18
2	(1,1,0)	1,83
3	(0,3,1)	1,31
4	(1,1,2)	2,79
5	(2,1,1)	2,49
6	(2,0,4)	3,95
7	(1,2,0)	2,11

Tabela 2.2 – Dados genéricos para fins de exemplificação do processo de treinamento das redes RBF.

Assim como no exemplo anterior, e com base nas informações citadas acima podemos utilizar a [Equação \(2.22\)](#) para resolver o problema em termos das sinapses w_i , ou seja, podemos treinar a rede para ajustar uma função de saída utilizando a [Equação \(2.31\)](#) abaixo:

$$F(x) = \sum_{i=1}^2 w_i \exp\left(\frac{-1}{2\sigma_i^2} \|x - x_i\|^2\right) + w_0 = y_j \quad (2.31)$$

onde $\sigma_1 = \sigma_2 = 5^{1/2}$ corresponde ao desvio padrão dos dados, e y_j são as saídas fornecidas à rede. As coordenadas dos centros estão localizados em $x_1 = (0,0,0)$ e $x_2 = (2,0,4)$ O objetivo agora é resolver as equações simultâneas resultantes da [Equação \(2.28\)](#), que resulta no seguinte sistema de equações:

$$\begin{cases} w_1 e^{-0,1\|(1,1,1) - (0,0,0)\|^2} + w_2 e^{-0,1\|(1,1,1) - (2,0,4)\|^2} + 1w_0 = 2,18 \\ w_1 e^{-0,1\|(1,1,0) - (0,0,0)\|^2} + w_2 e^{-0,1\|(1,1,0) - (2,0,4)\|^2} + 1w_0 = 1,83 \\ w_1 e^{-0,1\|(0,3,1) - (0,0,0)\|^2} + w_2 e^{-0,1\|(0,3,1) - (2,0,4)\|^2} + 1w_0 = 1,31 \\ w_1 e^{-0,1\|(1,1,2) - (0,0,0)\|^2} + w_2 e^{-0,1\|(1,1,2) - (2,0,4)\|^2} + 1w_0 = 2,79 \\ w_1 e^{-0,1\|(2,1,1) - (0,0,0)\|^2} + w_2 e^{-0,1\|(2,1,1) - (2,0,4)\|^2} + 1w_0 = 2,49 \\ w_1 e^{-0,1\|(2,0,4) - (0,0,0)\|^2} + w_2 e^{-0,1\|(2,0,4) - (2,0,4)\|^2} + 1w_0 = 3,95 \\ w_1 e^{-0,1\|(1,2,0) - (0,0,0)\|^2} + w_2 e^{-0,1\|(1,2,0) - (2,0,4)\|^2} + 1w_0 = 2,11 \end{cases} \quad (2.32)$$

que pode ser reescrito de modo matricial como:

$$GW = Y \quad (2.33)$$

onde:

$$W = (w_1 \quad w_2 \quad w_0)^T$$

$$Y = (2,18 \quad 1,83 \quad 1,31 \quad 2,79 \quad 2,49 \quad 3,95 \quad 2,11)^T$$

$$G = \begin{pmatrix} 0,7408 & 0,3328 & 1 \\ 0,8187 & 0,1652 & 1 \\ 0,3678 & 0,1108 & 1 \\ 0,5488 & 0,5488 & 1 \\ 0,5488 & 0,3678 & 1 \\ 0,1353 & 1 & 1 \\ 0,6065 & 0,1224 & 1 \end{pmatrix} \quad (2.34)$$

A matriz G é singular pois não é quadrada, logo não é inversível. Sendo assim é necessário utilizar a [Equação \(2.17\)](#) para determinar quem são os elementos da matriz de coeficientes W , ou seja:

$$W = G^+Y \quad (2.35)$$

os procedimentos para determinar os elementos da matriz pseudo inversa de G são os mesmos exibidos no exemplo anterior, cujo procedimento e resultado é apresentado como se mostram na [Equação \(2.36\)](#) e [Equação \(2.37\)](#) respectivamente.

$$W = \begin{pmatrix} 1,02 & 1,00 & 1,69 & 0,52 & 0,03 & -0,58 & -0,31 \\ 0,41 & 0,13 & -1,26 & 0,53 & 0 & 0,75 & -0,57 \\ -0,56 & -0,45 & 1,53 & -0,34 & 0,12 & 0,17 & 0,52 \end{pmatrix} \begin{pmatrix} 2,18 \\ 1,83 \\ 1,31 \\ 2,79 \\ 2,49 \\ 3,95 \\ 2,11 \end{pmatrix} \quad (2.36)$$

$$W = (-1,2527 \quad 1,4930 \quad 2,6321)^T \quad (2.37)$$

Combinando a [Equação \(2.37\)](#) com a [Equação \(2.22\)](#) tem-se a função de ajuste para o problema em questão. A mesma pode ser expressa através da [Equação \(2.38\)](#) abaixo:

$$F(\vec{r}_1, \vec{r}_2) = -1,2527e^{-0,1\vec{r}_1^2} + 1,4930e^{0,1\vec{r}_2^2} + 2,2631 \quad (2.38)$$

onde $\vec{r}_1 = \|\vec{x} - \vec{x}_1\|$ e $\vec{r}_2 = \|\vec{x} - \vec{x}_2\|$.

3. DEFINIÇÃO DE VIDRO

Em 2017, Edgar Dutra Zanotto (*n.* 1954) e John Christopher Mauro (*n.* 1975) apresentaram uma definição do estado vítreo que pode ser considerada definitiva. De acordo com estes autores, o vidro consiste num “estado condensado da matéria fora do equilíbrio termodinâmico, não cristalino, que exhibe transição vítrea. A estrutura de vidros é similar ao do líquido super-resfriado (LSR) progenitor e eles relaxam espontaneamente até o estado de um LSR. Seu destino final é a cristalização” (ZANOTTO e MAURO, 2017).

O importante em tal definição é que, além de englobar tudo o que é necessário, esta evita o uso dos termos mais comuns em definições anteriores e já consideradas clássicas, como “sólido” e “amorfo”. Sólidos consistem num estado da matéria que apresenta ordens atômicas de curto, médio e longo alcances além de serem termodinamicamente estáveis abaixo da temperatura de fusão, algo que não ocorre aos vidros. Já o problema com o termo amorfo deve-se a confusão que é feita com materiais ditos não-cristalinos. Vidros são de fato materiais não-cristalinos, porém não são amorfos, uma vez que um material amorfo não apresenta ordem atômica de curto alcance e também não apresenta T_g (GUPTA, 1996).

Vidros de óxidos de sílica (SiO_2) e boro (B_2O_3) são considerados formadores vítreos por simples fusão e resfriamento rápido, conforme apresentado a seguir.

3.1 TEMPERATURA DE TRANSIÇÃO VÍTREA

A temperatura T_g pode ser obtida por diferentes técnicas, entre elas dilatométrica, calorimetria exploratória de varredura e condutividade (VARSHNEYA, 2013). Dependendo da taxa de resfriamento de um líquido pode ocorrer um fenômeno conhecido como transição vítrea, no qual um líquido super-resfriado sofre, com a variação de temperatura, uma mudança mais ou menos brusca nas suas propriedades físicas (volume e densidade) e termodinâmicas (entalpia e calor específico).

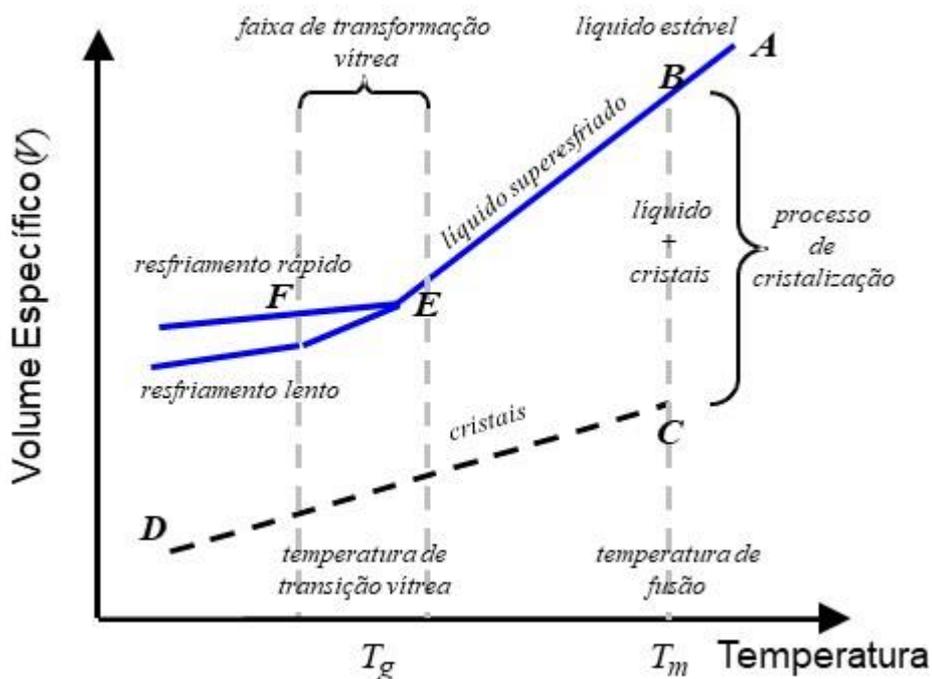


Figura 3.1 — Super-resfriamento de um líquido e definição de T_g , conforme Varshneya e Mauro (2019).

Para explicar este fenômeno é apresentada a [Figura 3.1](#), que representa a variação do volume específico sob resfriamento ([NASCIMENTO, 2004](#); [VARSHNEYA e MAURO, 2019](#)). Partindo do estado líquido no ponto A, observa-se a diminuição constante do volume ao longo da linha AB com o resfriamento. Se os núcleos de fases cristalinas estiverem presentes e a taxa de resfriamento for lenta, a cristalização ocorre na temperatura de solidificação T_s ($\equiv T_m$, ou temperatura de fusão), havendo assim uma redução brusca de volume (linha BC). Em seguida, o sólido se contrai ao longo de CD à medida que o resfriamento prossegue. Entretanto, se a velocidade de resfriamento for suficientemente rápida, o material não cristaliza em T_m . Em vez disso, segue uma trajetória de líquido super-resfriado ao longo da linha BE, extensão de AB, onde se verifica um decréscimo de volume resultante do aumento da viscosidade, o que por sua vez dificulta ainda mais o movimento translacional dos átomos (ou ainda moléculas), diminuindo a intensidade da vibração atômica e torna as relaxações estruturais mais restritas. Em uma certa temperatura T_g , a reta sofre uma mudança significativa na sua inclinação, devido a variação abrupta do coeficiente de expansão térmica e da finalização da relaxação estrutural, e prossegue à linha CD do estado cristalino. T_g é denominada temperatura de transição vítrea e corresponde à

temperatura abaixo da qual o material se comporta como vidro, ou seja, onde ele deixa a forma plástico-viscosa do estado líquido e adquire a rigidez de um sólido.

A localização de T_g varia com a taxa de resfriamento pois quanto mais lento for o resfriamento, maior será o tempo disponível para as reacomodações estruturais e T_g ocorre em F . É, portanto, mais apropriado se falar em T_g como “faixa de transição vítrea” do que como um valor fixo.

3.2 VIDROS BORATOS (BREVE DESCRIÇÃO E PROPRIEDADES)

Vidros boratos apresentam características únicas quando comparados a outros sistemas como silicatos. Devido ao baixo ponto de fusão do B_2O_3 (≈ 475 °C) quando comparado ao do quartzo (≈ 1720 °C), as composições binárias de metais alcalinos de ambos sistemas também apresentam correspondentes baixos pontos de fusão. A simples adição de B_2O_3 tende a diminuir a expansão térmica e aumentar a resistência química de vidros silicatos, sendo a marca comercial Pyrex™, da empresa americana Corning *Inc.*, conhecida por servir de vasilhames que se expandem pouco quando aquecidos e resistirem a diversos ácidos fortes, sendo também quimicamente inertes e, portanto, ideais para produtos farmacêuticos.

Em termos estruturais, num vidro de borato a coordenação de oxigênio em torno de cada B é de apenas 3 e, portanto, a unidade estrutural básica consiste de um triângulo formado de BO_3 , sendo que o átomo de B está ligeiramente acima do plano formado pelos três oxigênios (VARSHNEYA e MAURO, 2019). Dá-se o nome de grupo *boroxol* a estrutura característica de vidros B_2O_3 formados por dois anéis hexagonais de três unidades BO_3 cada (anéis boroxol, conforme Figura 3.2), sendo as distâncias entre átomos de boro calculadas como próximas de 0,273 nm, e as entre boro e oxigênio de 0,137 nm (VARSHNEYA e MAURO, 2019). Outras estruturas de boratos são apresentadas na Figura 3.3.

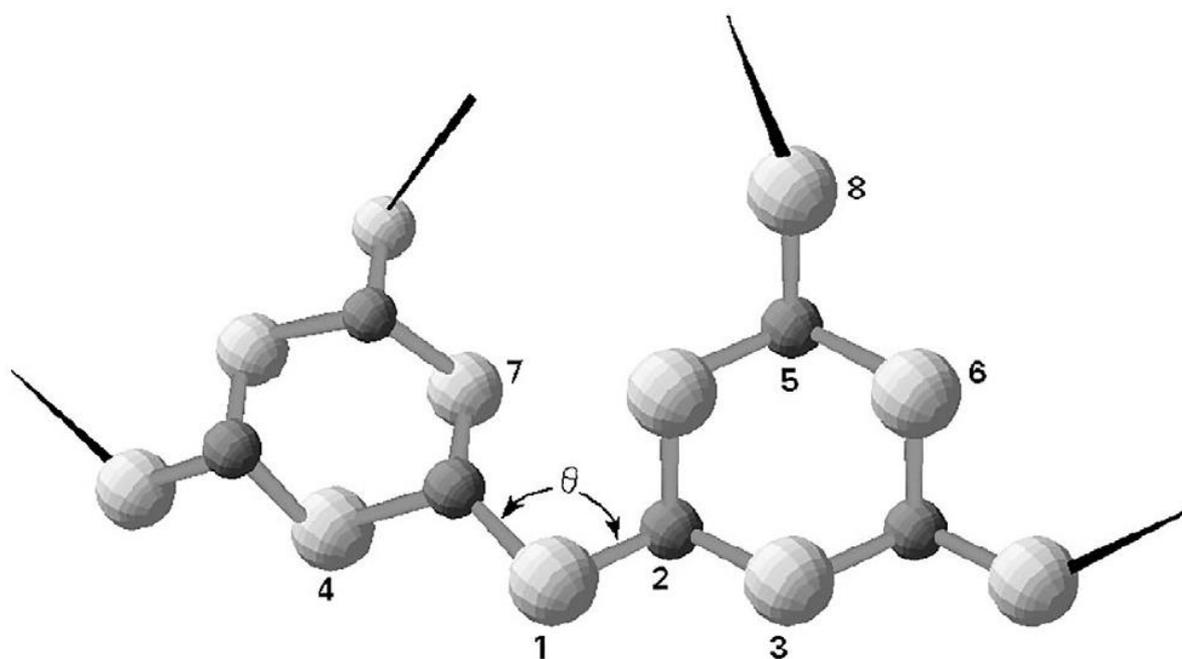


Figura 3.2 — Dois anéis hexagonais unidos por um oxigênio entre dois átomos de boro em ângulo θ formando uma estrutura de grupo *boroxol* (i.e., a junção de dois anéis boroxol), muito comum em vidros boratos. Em cada anel é possível observar três unidades BO_3 , conforme [Varshneya e Mauro \(2019\)](#).

Em vidros a base de silicatos e boratos, os metais alcalinos são considerados modificadores da rede por, em geral, mudarem a temperatura de fusão. Especificamente no caso de silicatos, a adição de óxidos de metais alcalinos tende a diminuir o ponto de fusão do quartzo na pré-mistura da composição de vidros comerciais. Na estrutura vítrea, formam íons monovalentes ocupando sítios intersticiais, unindo-se eventualmente a oxigênios, e desta forma, rompendo a rede dos formadores vítreos. Tais interações entre álcalis e oxigênios permitem a formação de oxigênios não ponteantes (em inglês, *nonbridging oxygens*, *NBO*). A adição de um álcali promove, portanto, o surgimento de um *NBO*. Tal criação reduz a conectividade da rede de formadores vítreos, como o B_2O_3 ([VARSHNEYA e MAURO, 2019](#)), influenciando sobremaneira em propriedades termodinâmicas e cinéticas, tais como T_g , viscosidade, densidade e condutividade, entre outras.

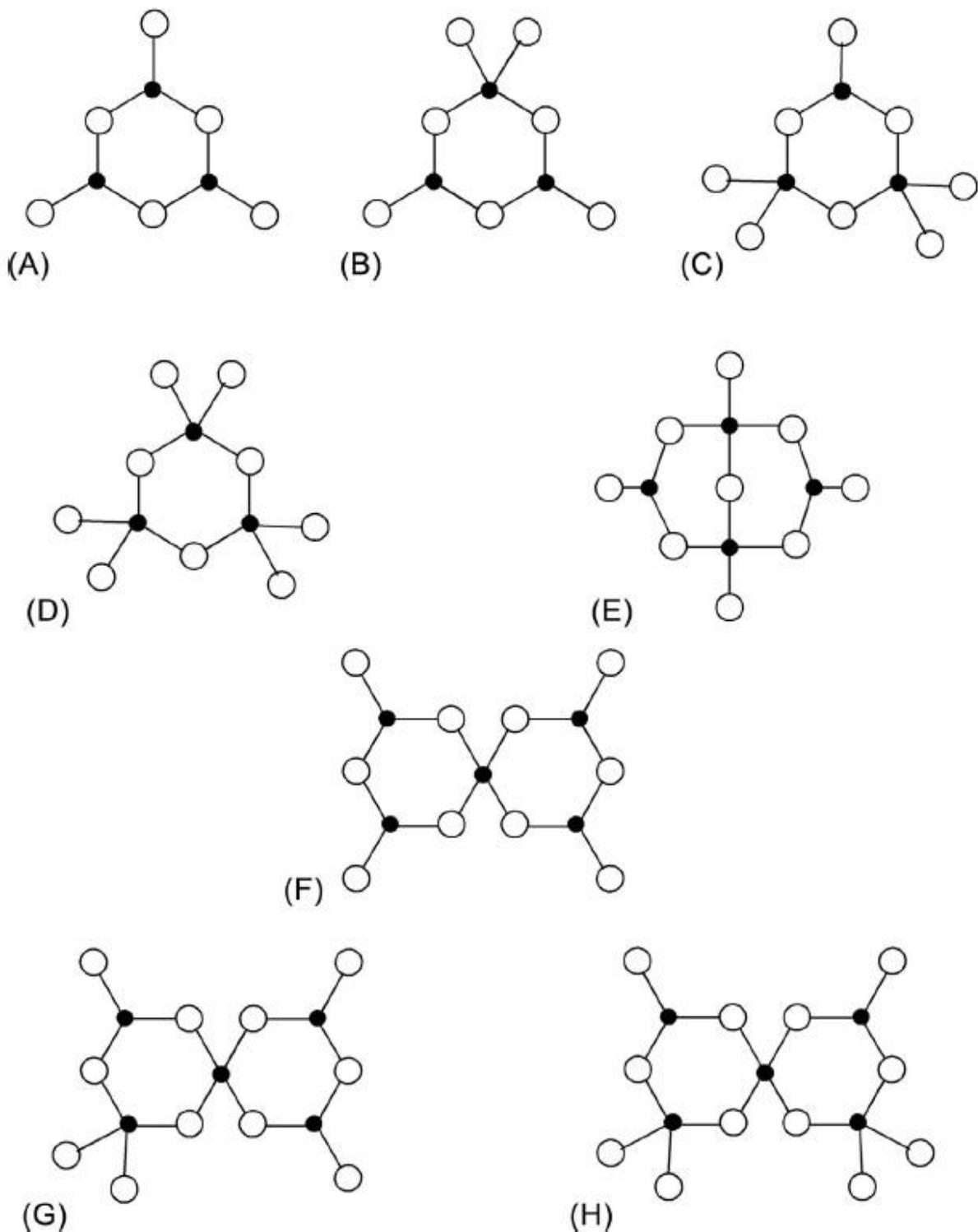
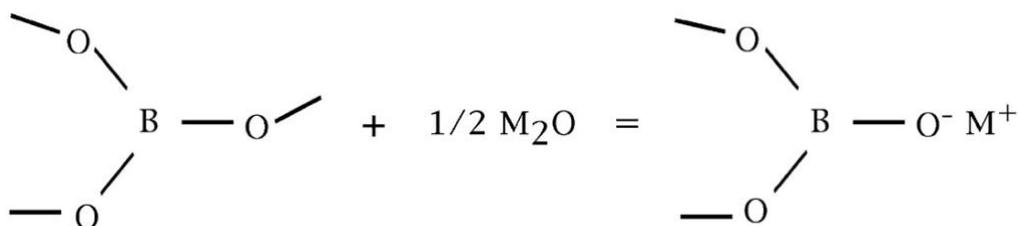


Figura 3.3 – Diversos grupos postulados para os sistema de vidros alcalino borato alcalino (onde • indica boro): (A) um anel boroxol; (B) grupo triborato; (C) grupo di-triborato; (D) grupo metaborato; (E) grupo diborato; (F) grupo pentaborato; (G) grupo di-pentaborato; e (H) grupo tri-pentaborato, conforme [Varshneya e Mauro \(2019\)](#).

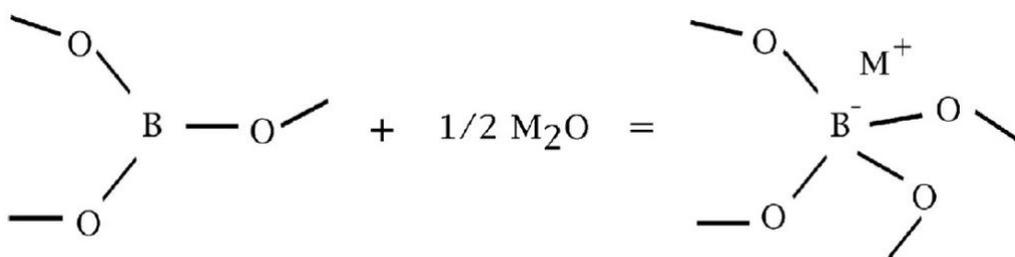
Em termos gerais, a configuração ao redor de cada átomo de boro pode ser expressa como unidades Q_i (com i entre 1 e 4), sendo o íon boron trivalente positivo. Considerando vidros alcalino-boratos de composição $xM_2O \cdot (1-x)B_2O_3$, com $x \leq \frac{1}{2}$ e $M = \{Li, Na, K, Rb, Cs\}$, existem quatro tipos distintos de estruturas formadoras de rede vítrea a serem considerados: i) átomo de boro coordenado a quatro oxigênios ponteantes, Q_4 (i.e., um tetraedro isolado BO_4) - tais estruturas são predominantes em vidros boratos puros (B_2O_3); ii) átomo de boro coordenado a três oxigênios ponteantes, Q_3 (i.e., BO_3) - oxigênios não ponteantes em geral diminuem a rigidez da estruturas dos vidros boratos; iii) boro tri-coordenado a dois oxigênios ponteantes Q_2 (i.e., $-BO_2$); iv) o próprio oxigênio ponteante (B-O). Importante ressaltar, como já descrito, que nenhum dos oxigênios não ponteantes nem íons alcalinos são considerados como formadores de rede (MAURO, GUPTA e LOUCKS, 2009).

Conforme apresentado por Varshneya e Mauro (2019), a introdução de oxigênio a partir de óxidos M_2O numa composição alcalino-borato promove duas situações:

i) Cria um *NBO*, formando a estrutura $BO_{2/2}O^-M^+$ (sendo que o subscrito “/2” é usado para indicar a conexão com 2 cátions M^+), conforme a reação indicada abaixo:



ii) Converte o boro de uma tripla coordenação ("estado B^{3+} ") para uma quádrupla coordenação, conforme a reação indicada abaixo:



Nesta particular formação, grupos tetraédricos $[\text{BO}_{4/2}]^-$, carregados negativamente, têm a carga compensada pela proximidade de um álcali M^+ .

Desta forma, acréscimos de metais alcalinos M aos sistemas binários boratos em estudo, a relação entre metais alcalinos x e $1-x$ boros da forma $x/(1-x) \leq 1/2$, ou ainda $\leq 1/3$, pode levar a uma mudança de coordenação de três para quatro. Oxigênios não ponteantes tendem a se formar em altas concentrações x de álcalis, com $x > 1/3$. Gupta (1986) e Mauro, Gupta e Loucks (2009) determinaram a fração de cada espécie formadora $N(Q^i)$ da rede de alcalino-boratos.

Varshneya e Mauro (2019) apresentaram um procedimento para calcular as configurações estruturais em boratos alcalinos. Primeiro, é necessário calcular a razão $R_x = x/(1-x)$. Se este valor for $\leq 1/2$, o número de estruturas tetraédricas N_4 é igual a R_x e a fração de oxigênios não ponteantes f_{NBO} é nula. Para valores de $R_x > 1/2$, N_4 corresponde a:

$$N_4 = \frac{(3 - R_x)}{5} \quad (2.39)$$

conforme Gupta (1986) e Varshneya e Mauro (2019). Neste caso, a f_{NBO} é calculada a partir dos álcalis restantes.

Como primeira aplicação (VARSHNEYA e MAURO, 2019), considere um vidro de composição $20\text{Na}_2\text{O} \cdot 80\text{B}_2\text{O}_3$ (em mol%). Neste caso, $R_x = x/(1-x) = 20/80 = 0,25$. Como este valor é menor que 50%, $N_4 = R_x$ e $f_{\text{NBO}} = 0$. Por fim, $N_3 + N_4 = 1$, logo $N_3 = 0,75$, e o número de coordenação média do boro neste sistema será $3 \times 0,75 + 4 \times 0,25 = 3,25$.

Enquanto segunda aplicação, considere um vidro de composição $40\text{K}_2\text{O} \cdot 60\text{B}_2\text{O}_3$ (em mol%). Neste caso, $R_x = x/(1-x) = 40/60 = 0,666$. Como este valor excede 50%, $N_4 = (3 - R_x)/5 = 0,467$ e como $N_3 + N_4 = 1$, logo $N_3 = 0,533$.

O número de unidades B_4 é, portanto, $120 \times 0,467 \sim 56$, e como um B_4 requer um íon K^+ , existem 56 K^+ , sobrando 24 K^+ , que criam 24 NBO. No total, são $40 + 180 = 220$ oxigênios. Desta forma, $f_{\text{NBO}} = 24/220 = 0,11$. E o número de coordenação média do boro neste sistema será $3 \times 0,533 + 4 \times 0,467 \sim 3,47$.

Mauro, Gupta e Loucks (2009) propuseram um modelo topológico de predição de T_g em termos da composição x em sistemas binários de borato de lítio e sódio assumindo que o produto da temperatura de transição vítrea e a entropia configuracional de uma composição qualquer seja equivalente a uma composição de referência (*i.e.*, resumidamente, no calor envolvido na reação, de acordo com a Segunda Lei da Termodinâmica):

$$T_g(x)S_c[T_g(x), x] = T_g(x_R)S_c[T_g(x_R), x_R] \quad (2.40)$$

Sem entrar no detalhe da teoria, Mauro, Gupta e Loucks (2009) obtiveram as seguintes previsões do comportamento de T_g em função da composição:

$$T_g(x) = \frac{1}{5} \frac{(5 - 4x)}{1 - 2x} T_g(0), \text{ para } x \leq \frac{1}{3} \quad (2.41)$$

$$T_g(x) = \frac{1}{5} \frac{(31 - 38x)}{8x - 1} T_g(0), \text{ para } \frac{1}{3} < x \leq \frac{1}{2} \quad (2.42)$$

sendo que $T_g(0)$ corresponde ao valor da temperatura de transição vítrea do formador vítreo B_2O_3 , de cerca de 270 °C.

Esta proposta será comparada com a estimativa do comportamento de T_g por meio do uso de redes neurais artificiais do tipo função de base radial com poucos neurônios.

4. MATERIAIS E MÉTODOS

A seguir são apresentados os materiais e métodos utilizados nesta dissertação, envolvendo a obtenção dos dados via banco SciGlass, o uso da ferramenta Jupyter Notebook e o desenvolvimento da rede neural, discutida em capítulos anteriores.

4.1 OBTENÇÃO DOS DADOS (SCIGLASS)

O SciGlass (versão 7.7) é um banco de dados/software sobre vidros, idealizado pelo químico russo Oleg Vsevolodovich Mazurin (1927 - 2021), com base em publicações de periódicos científicos e patentes internacionais. Em sua atual versão, o SciGlass possui o título de maior banco de dados sobre materiais vítreos existentes, contendo mais de 360.000 composições distintas, seguindo pelo INTERGLAD (*International Glass Database*) contendo cerca de 250.000 composições diferentes ([NASCIMENTO e NASCIMENTO, 2021](#)).

O banco de dados conta com basicamente todas as propriedades físicas e químicas comumente analisada no ambiente científico a respeito de vidros, e misturas formadoras de vidros, tais como, viscosidade, análise via espectroscopia do UV ao IV, diagramas de fase binários e ternários, temperatura de transição vítrea, etc. É possível ainda neste ambiente realizar análises preditivas a respeito das propriedades citadas através de métodos computacionais apresentados no software.

Utilizando o referido banco de dados, foram extraídos como amostras um total de 746 composições distintas de vidros óxidos alcalinos boratos (*i.e.*, binários), obtidos via diversas medidas como calorimetria experimental de varredura (DSC) e seus valores de temperatura de transição vítrea associados. Do montante total de amostras, uma quantidade de aproximadamente 14,78% é de sistemas boratos de potássio, 29,35% são do sistema borato de lítio, 39,94% são do sistema borato de sódio, 6,9% são do sistema borato de rubídio, e 8,84% são do sistema borato de cério.

Na [Figura 4.1](#) é possível observar a distribuição da temperatura vítrea associada a cada um dos sistemas isolados em relação a densidade de frequência associado a estas faixas de temperaturas. Como pode ser notado, a maior concentração de dados está relacionada a um intervalo de temperatura de transição vítrea acima de 650 K, muito comum na maioria dos sistemas em questão. Estes

mesmos dados serão utilizados para realizar o treinamento das redes neurais e respectivos testes.

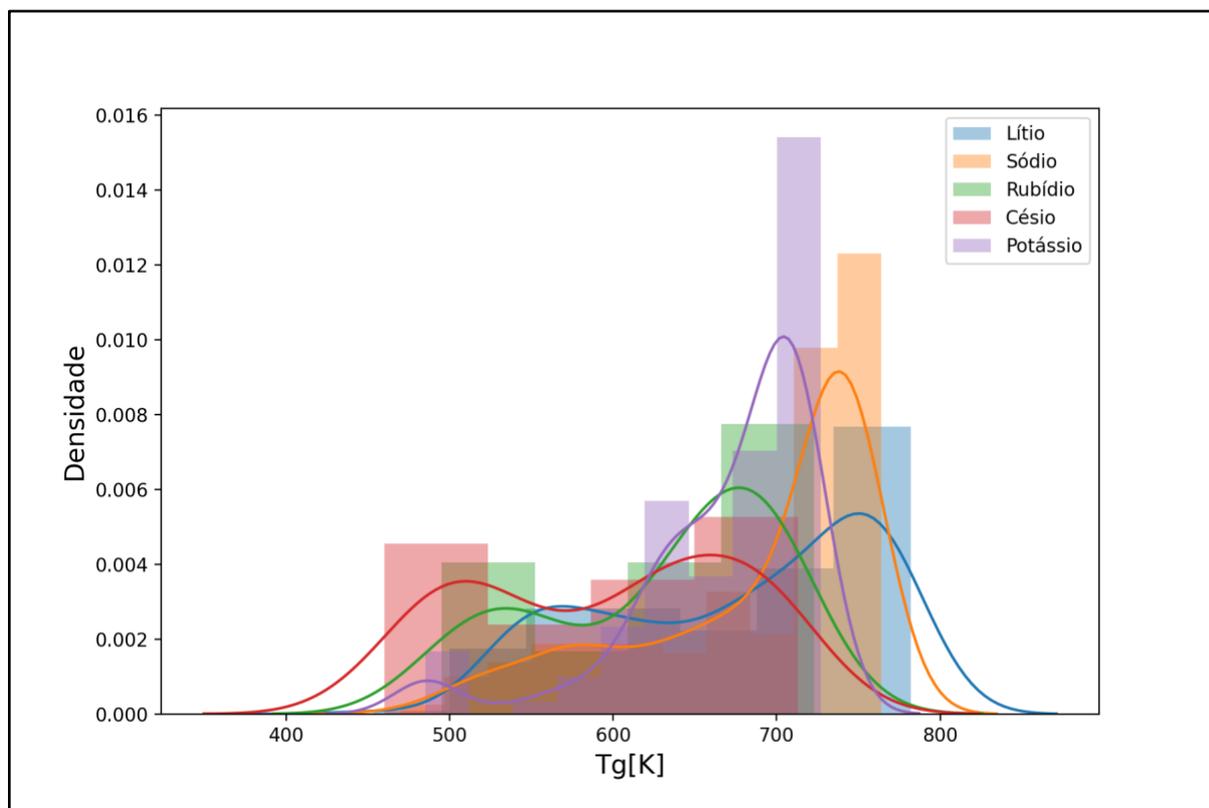


Figura 4.1 — Histograma da distribuição de T_g dos sistemas alcalino-boratos analisados.

4.2 JUPYTER NOTEBOOK

O Jupyter Notebook é um aplicativo *Web* de código aberto que está vinculado a outro pacote de aplicativos chamado ANACONDA, disponível para o Sistema Operacional Linux gratuitamente para download em: www.anaconda.com. O aplicativo em si é um dos mais populares atualmente na era IoT pois seu uso permite a manipulação de documentos, criação de código de programação, ciências de dados, visualização gráfica, visualização de textos narrativos, modelagem estatística, simulação numérica, aprendizado de máquina no geral, dentre outros recursos (SANTOS, 2020).

O projeto Jupyter, desenvolvido e criado por Fernando Pérez (*n.* 1971, físico colombiano) e Brian Ellison Granger (*n.* 1979, cientista da computação) veio a público em 2014, sendo apresentado como um projeto sem fins lucrativos e com o objetivo

de promover a criação de softwares de código aberto nas mais diversas linguagens de programação da época, tais como *R* e Python (JUPYTER, 2015).

A concepção do nome Jupyter é uma homenagem dos criadores aos manuscritos de Galileu que descrevem e registram a descoberta das luas de Júpiter realizadas em aproximadamente 1610 (as descrições a respeito das descobertas das luas de Júpiter se encontram no livro *Sidereus Nuncius*, ou “Mensageiro das Estrelas”), e faz referência às três principais linguagens de programação suportadas, Julia, Python e R (JUPYTER, 2015). Sua interface de utilização é conhecida por ser bastante intuitiva e compreensível.

4.3 DESENVOLVIMENTO DA REDE NEURAL

Utilizando os dados extraídos do banco de dados SciGlass em soma ao ambiente de programação escolhido, que neste caso foi o Jupyter Notebook, determinou-se para a elaboração da rede neural a linguagem de programação Python devido ao enorme uso e acessibilidade da mesma. O hardware de execução consiste em um dispositivo notebook equipado com 4 GB de memória RAM, processador Intel Core I3 com 4 núcleos de frequência máxima de processamento de 2,3 GHz cada, e um dispositivo de armazenamento HD com capacidade máxima de 1Tb.

Ao iniciar na compreensão dos aspectos qualitativos do código de programação em si, dois conceitos que dizem respeito ao Python devem estar definidos: um deles é o de *biblioteca*, que consiste num conjunto de *scripts*, ou sub programas, utilizados para o desenvolvimento de softwares, que tem em sua essência um conjunto de aplicabilidade bem definidos. Como exemplo, pode-se citar que, em uma biblioteca Python relacionada a visualização e criação gráfica, é possível encontrar uma gama de *scripts* que desempenham tarefas independentes, tais como: orientação da imagem, coloração, aplicação de texto narrativo, etc. O destino de aplicação destas bibliotecas está normalmente relacionado a uma maneira de otimizar o processo de criação do código através da diminuição de sua extensão (KELLEHER, 2020).

O outro conceito mencionado diz respeito aos sub programas presentes na biblioteca. Estes por sua vez são definidos em programação como módulos e funcionam como ferramentas para o auxílio em tarefas específicas, tais como: inversão de matrizes, diagonalização de matrizes, operações algébricas, cálculo de

métricas estatísticas, incluindo também os exemplos mencionados no parágrafo anterior (MULLER, 2016).

Voltando ao desenvolvimento do código,¹² a primeira etapa realizada foi elaborar uma biblioteca contendo dois módulos, e a esta biblioteca denominou-se *funções*. O primeiro módulo desenvolvido teve o objetivo de criar as funções de base radial que iriam servir como função de ativação na camada intermediária da rede neural, conforme o mencionado no referencial teórico. O segundo módulo ficou responsável por emitir uma saída ao executor do *script* que menciona qual o tipo de função de base radial escolhida.

Em detalhes, o módulo criado recebe três parâmetros de entrada: o primeiro é um valor entre 1 e 7 que está vinculado ao tipo de função de base radial que se deseja utilizar que, respectivamente, representam as funções de base radial: multi-quadrática, multi-quadrática recíproca, multi-quadrática recíproca inversa, gaussiana, secante hiperbólica, cosseno hiperbólico, e splines de placas finas. O segundo parâmetro consiste no conjunto de dados utilizados \vec{x} (somente características preditoras e em formato numérico). O terceiro parâmetro representa o desvio padrão dos dados apresentados.

O segundo módulo desenvolvido recebe apenas um parâmetro e que é simultâneo ao do primeiro módulo, que quando aplicado, exibe em forma de saída textual a função selecionada no primeiro módulo.

A partir deste momento, entraremos no processo de configuração dos parâmetros da rede neural. Para ir de encontro com a hipótese deste trabalho (estabelecer uma rede neural com poucos neurônios com capacidade preditiva de T_g) foi adotado os seguintes parâmetros: i) todas as redes desenvolvidas tiveram apenas dois neurônios nas suas respectivas camadas intermediárias, onde cada neurônio foi ativado por uma função de base radial gaussiana; ii) as redes tiveram seu treinamento realizado com apenas uma *época* (*epoch*), sem aprendizado por lote, tampouco mini lote, e sem ajuste de números de neurônios via *dropout*, visando simplificar o custo computacional; iii) no código em si, iniciou-se o processo importando todas as bibliotecas necessárias para o desenvolvimento das redes.

¹² Os códigos apresentados nesta dissertação foram embasados no *software* sem registro do matemático, bacharel em ciência e tecnologia e engenheiro aeronáutico brasileiro Eden Denis Ferreira da Silva Lopes Santos (n. 1987), a quem prestamos sinceros agradecimentos.

As bibliotecas e módulos importados que foram utilizados se encontram na [Tabela 4.1](#), onde é possível encontrar seus nomes e suas características de uso:

Bibliotecas e Módulos	Características de uso
Funções	Função de ativação dos neurônios
Numpy	Operações algébricas lineares
Pandas	Manipulação de banco de dados
sklearn.model_selection	Repartição dos bancos de dados
matplotlib.pyplot	Criação de conteúdo visual
seaborn	Criação de conteúdo visual
sklearn.metrics	Métricas estatísticas (R^2)

Tabela 4.1 – Conjunto de bibliotecas e módulos utilizados para o desenvolvimento das redes neurais utilizadas.

Com os módulos já carregados, foi necessário utilizar a biblioteca *pandas* para que fosse possível manipular os dados no ambiente do Jupyter Notebook. Além disso foi aplicado uma função desta biblioteca chamada de *dropna()* para garantir que não houvesse nenhum dado faltando no conjunto apresentado pois poderia ocasionar *bug* durante o treinamento da rede.

O conjunto de dados obtidos possui três colunas, sendo que duas correspondem a valores linearmente dependentes, que são os valores da concentração molar dos modificadores de rede x , e a concentração molar percentual de boro, $1-x$. A terceira coluna é o valor da temperatura de transição vítrea T_g associada. Na [Tabela 4.2](#) é possível observar a estrutura obtida para os onze primeiros pontos utilizados no treinamento do sistema borato de rubídio.

Após a disponibilidade de manipulação do banco de dados, iniciou-se o processo de caracterização das funções de ativação, e para isso, criou-se uma variável que recebe um valor de entrada associado ao tipo de função de base radial que será utilizada como ativação da camada oculta. Como já mencionado, a função escolhida foi a gaussiana, ou seja, o valor que a variável criada recebeu em forma de entrada foi 4. Esta mesma variável será utilizada mais à frente.

Rubídio (x , mol%)	Boro (mol%)	$T_g(K)$
0	100	540
24	76	687
9,0909	90,9091	605
13,0435	86,9565	641
16,6667	83,3333	660
23,0769	76,9231	679
25,9259	74,0741	699
28,5714	71,4286	723
33,3333	66,6667	694
35,4839	64,5161	699
37,5	62,5	667

Tabela 4.2 – Exemplo de aplicação deste trabalho. Dez amostras de entrada para o treinamento da rede neural associada ao sistema borato de rubídio (x em mol %).

Após o processo mencionado no parágrafo anterior, realizaram-se algumas manipulações no conjunto de dados para que a solução do problema em si fosse iniciada apropriadamente. Primeiro, dividiu-se através do módulo `sklearn.model_selection` o conjunto de dados em duas partes, sendo que a primeira parte continha 70% dos dados associados, e que foram utilizados para o treinamento da rede; e os 30% restante foram reservados para o teste da mesma.

Nesta mesma etapa foi introduzida uma *semente* na repartição mencionada, para que fosse obtido o mesmo curso aleatório ao executor do código, possibilitando assim a reprodução fiel das etapas aqui mostradas. Por seguinte foram criadas algumas variáveis para que carregassem algumas informações a respeito dos dados, sendo que as mesmas se associavam respectivamente às seguintes informações: número de instâncias do conjunto de treinamento, número de variáveis preditoras, vetor de variáveis preditoras \vec{x} e matriz com as respostas desejadas utilizadas para o treinamento y .

Após a extração das características do banco de dados anteriormente mencionadas, iniciou-se a configuração das coordenadas dos neurônios. A partir

deste momento é conveniente nos referirmos a neurônios como polos devido à sua contextualização às redes de funções de base radial. O processo de configuração começou criando-se uma condição que deve ser obrigatoriamente satisfeita relacionada ao número de polos (P) utilizados na camada oculta em função do número de características preditoras (C) utilizadas. Esta condição é a seguinte:

$$\text{Se } C = 1 \rightarrow P = C + 1$$

$$\text{Se } C > 1 \rightarrow P = C \tag{4.1}$$

Após isso foi criada uma matriz 2x2 (pois no nosso caso $C > 1$) de elementos nulos com o objetivo de informar a localização dos polos. A estratégia adotada para a alteração dos elementos nulos por vetores de posição dos centros foi a aleatoriedade dentro de um domínio de 0 a 1; ou seja, para cada elemento da referida matriz, foi sorteado um valor dentro do referido domínio e, em sequência, foi acrescentado um termo multiplicativo de escala no valor de 100, fazendo assim com que os vetores de posição fossem da mesma ordem das variáveis preditoras. Na [Tabela 4.3](#) é apresentada a referida matriz de coordenadas.

Neurônios	\vec{i}	\vec{j}
Polo 1	37,454012	95,071431
Polo 2	73,199394	59,865848

Tabela 4.3 – Matriz de coordenadas dos polos (neurônios) obtidos via método de aleatoriedade.

Com a matriz de coordenadas dos polos estabelecida, foi realizado uma pequena alteração nas coordenadas do Polo 1, sendo que o mesmo teve sua localização movida para a origem do sistema de referência; ou seja, seu vetor de localização passou a ser (0,0) como nota-se na [Tabela 4.4](#).

Neurônios	\vec{i}	\vec{j}
Polo 1	0,0	0,0
Polo 2	73,199394	59,865848

Tabela 4.4 – Matriz com a coordenada do Polo 1 modificada.

A partir deste momento, a principal tarefa foi aplicar a [Equação \(2.19\)](#) para obter o conjunto de equações simultâneas associadas. Para este procedimento foi necessário calcular alguns parâmetros com base na matriz de coordenadas obtidas ([Tabela 4.4](#)). Primeiro, calculou-se a distância máxima entre os pólos através da norma euclidiana da diferença entre os vetores de posição associados. O resultado obtido foi de 94,56225; este é um parâmetro importante para estabelecer o desvio padrão σ dos dados, que foi calculado através da [Equação \(4.2\)](#), resultando em:

$$\sigma = \frac{\text{distância máxima}}{\sqrt{2P}} \approx 47,28 \quad (4.2)$$

Com estes dois parâmetros estabelecidos, a aplicação da [Equação \(2.19\)](#) iniciou-se de fato, obtendo-se assim o conjunto de equações mencionados, e conseqüentemente foi possível estabelecer a matriz de Green G associada.

Com a matriz de Green já estabelecida, o passo seguinte passou a ser o cálculo da matriz pseudo inversa de G , possibilitando assim que os coeficientes de expansão da [Equação \(2.19\)](#) fossem encontrados através das [Equações \(2.16\)](#) e [\(2.17\)](#) respectivamente. Estes coeficientes, para o caso do borato de rubídio, se encontram na [Tabela 4.5](#).

Coeficiente	Valor
w_1	3350,94
w_2	-994,929
w_0	415,024

Tabela 4.5 – Coeficientes obtidos para o sistema borato de rubídio via teoria de redes RBF.

Neste momento a especificação das redes de funções de base radial foi finalizada, e o processo de generalização foi iniciado, ou seja, aplicou-se os valores obtidos e expostos na [Tabela 4.5](#) via [Equação \(2.18\)](#) e foram calculadas todas as métricas que serão apresentadas no capítulo seguinte, de resultados e discussões.

Um detalhe importante é que as etapas aqui mencionadas foram elaboradas para estabelecer a rede neural associada ao sistema borato de rubídio (pois contém menos dados, sendo, portanto, uma escolha didática do procedimento apresentado no ANEXO 1); porém, o código utilizado foi criado de maneira universal, ou seja, para a aplicabilidade aos demais sistemas, basta que carregue ao Jupyter Notebook o conjunto de dados de interesse, pois os demais parâmetros são fixos e comuns a todos estes.

5. RESULTADOS E DISCUSSÕES

Foram utilizados 746 instâncias para treinar e testar redes neurais de funções de base radial com ativação gaussiana, com o objetivo de fazer extrapolações e ajustar funções que descrevessem a temperatura de transição vítrea de vidros com composições binárias de sistemas borato e elementos modificadores de rede: Li, Na, K, Rb e Ce. Do número total de instâncias utilizadas, foram 219 do sistema borato de lítio, 298 do sistema borato de sódio, 111 do sistema borato de potássio, 52 do sistema borato de rubídio, e 66 do sistema borato de céσιο. De cada um destes grupos, foram reservados 30% do número de instâncias associados para o processo de teste da rede, enquanto o restante foi utilizado para treino e caracterização dos coeficientes sinápticos.

Os valores sinápticos encontrados na etapa de treinamento, ou seja, os coeficientes da expansão de base radial gaussiana que determina os ajustes para as variáveis de entrada foram definidos utilizando redes de funções de bases radiais individuais (uma rede para cada sistema alcalino borato) com polos localizados a partir dos valores máximos e mínimos associadas a cada uma das distribuições de entrada dos dados de treinamento, e mantendo um dos polos sempre localizado na origem do sistema de referência conforme previsto na teoria. Os valores destes coeficientes são apresentados na [Tabela 5.1](#) abaixo:

Sinapses	Sódio	Lítio	Rubídio	Césio	Potássio
w_1	4928,814284	4757,606621	3350,94	304,913301	3962,267865
w_2	-1439,822214	-1350,348084	-994,929	-973,783830	-1095,682674
w_0	330,495390	331.779586	415,024	470,404873	417,757149

Tabela 5.1 – Valores dos coeficientes determinados a partir da teoria RBF para cada um dos sistemas borato usando a função gaussiana ([Equação \(2.1\)](#)).

A partir da [Tabela 5.1](#) foram definidas as funções que ajustam cada uma das cinco combinações dos sistemas desejados, possibilitando a avaliação dos modelos determinados pelas redes de funções de base radial, utilizando as métricas estatísticas pré determinadas no referencial teórico, estas funções estão dispostas

nas equações a seguir, representando respectivamente os sistemas boratos, de lítio, sódio, potássio, rubídio e céσιο.

$$Tg(\vec{x}) \approx 4757,60e^{-\frac{\|\vec{x}-(0,0)\|^2}{94,56}} - 1350,34e^{-\frac{\|\vec{x}-(73,19,59,86)\|^2}{94,56}} + 331,77 \quad (5.1)$$

$$Tg(\vec{x}) \approx 4928,81e^{-\frac{\|\vec{x}-(0,0)\|^2}{94,56}} - 1439,82e^{-\frac{\|\vec{x}-(73,19,59,86)\|^2}{94,56}} + 330,49 \quad (5.2)$$

$$Tg(\vec{x}) \approx 3962,26e^{-\frac{\|\vec{x}-(0,0)\|^2}{94,56}} - 1095,68e^{-\frac{\|\vec{x}-(70,87,58,87)\|^2}{94,56}} + 417,75 \quad (5.3)$$

$$Tg(\vec{x}) \approx 3350,94e^{-\frac{\|\vec{x}-(0,0)\|^2}{94,56}} - 994,92e^{-\frac{\|\vec{x}-(73,19,59,86)\|^2}{94,56}} + 415,02 \quad (5.4)$$

$$Tg(\vec{x}) \approx 304,91e^{-\frac{\|\vec{x}-(0,0)\|^2}{94,56}} - 973,78e^{-\frac{\|\vec{x}-(73,19,59,86)\|^2}{94,56}} + 470,40 \quad (5.5)$$

O resultado geral da representatividade das funções de ajuste, para os sistemas estudados são mostrados na [Figura 5.1](#), onde é possível observar o teste de determinação realizado em ambos os conjuntos de dados (teste e treino), para cada um dos sistemas explorados. É possível notar da [Tabela 5.1](#) que os valores dos coeficientes w_0 , w_1 e w_2 são equivalentes entre si em termos da ordem de grandezas (com exceção do dado w_1 para o borato de céσιο, que apresentou o menor conjunto de dados).

Não foi observado um aspecto ou tendência de sobreajuste, tampouco sob ajuste, uma vez que os modelos mantiveram um valor de R^2 médio, de 92,42% nos dados de teste. Isto representa uma melhora de 1,85% de fidelidades das amostras já que, na etapa de treinamento, as funções representaram em média 90,56% dos valores de entrada.

Com base nos argumentos anteriores, é considerável que a rede teve uma eficiência de treinamento satisfatória, e, em circunstância disso, as funções de ajustes representaram razoavelmente bem os valores utilizados para as etapas de treinamento e teste.

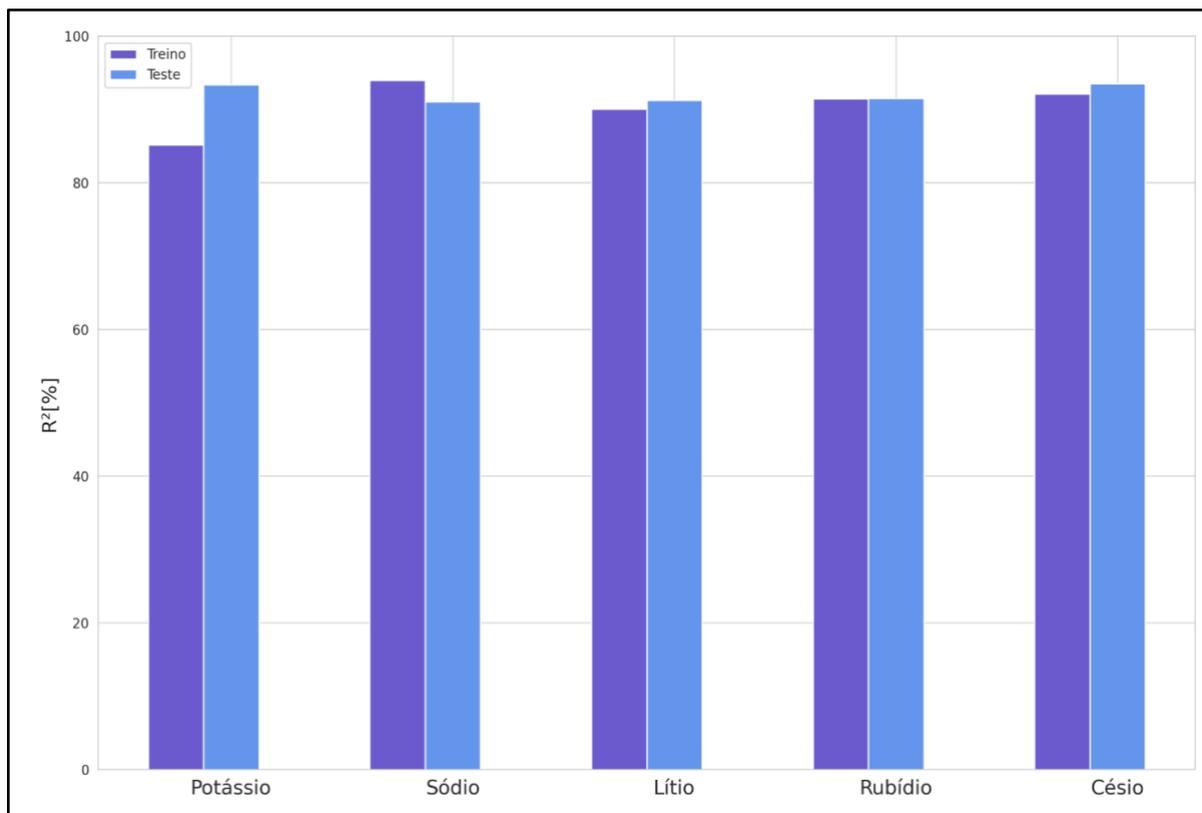


Figura 5.1 — Ajustes R^2 para os sistemas alcalino boratos: potássio, sódio, lítio, rubídio e césio.

O resultado para o sistema borato de lítio é exibido na [Figura 5.2](#), onde estão apresentados a curva de ajuste obtida através da rede neural associada e expressa através da [Equação \(5.1\)](#).

A rede tem uma performance satisfatória, onde os valores sinápticos obtidos por ela adequam-se de forma propícia ao ajuste através da [Equação \(5.1\)](#). Este argumento pode ser confirmado a partir do valor do coeficiente de determinação obtido R^2 , que neste caso foi de 91,6%. O valor deste coeficiente descreve uma boa relação entre a variável predita e seu valor real; entretanto, uma análise mais detalhada foi necessária; por exemplo, o erro médio percentual absoluto calculado através da relação de valores reais e valores de previsão foi de 3,1%, informando um erro de T_g na faixa de 19,6 K. Este valor tem implicações diretas no processo de formação de novos vidros que se baseiam nestas propriedades cinéticas para as suas possíveis aplicações, como as mencionadas no referencial teórico. Dito de outra forma, apesar do modelo se ajustar bem aos dados, existem fatores e limitações a serem considerados que determinarão se o modelo proposto é adequado. Entretanto, do ponto de vista teórico a rede desempenhou sua tarefa de forma eficiente e o ajuste

proposto é vantajoso em sua utilização pois explica a maior parte da distribuição dos dados evidenciado pelo seu valor do coeficiente de determinação encontrado e um erro absoluto médio relativamente baixo.

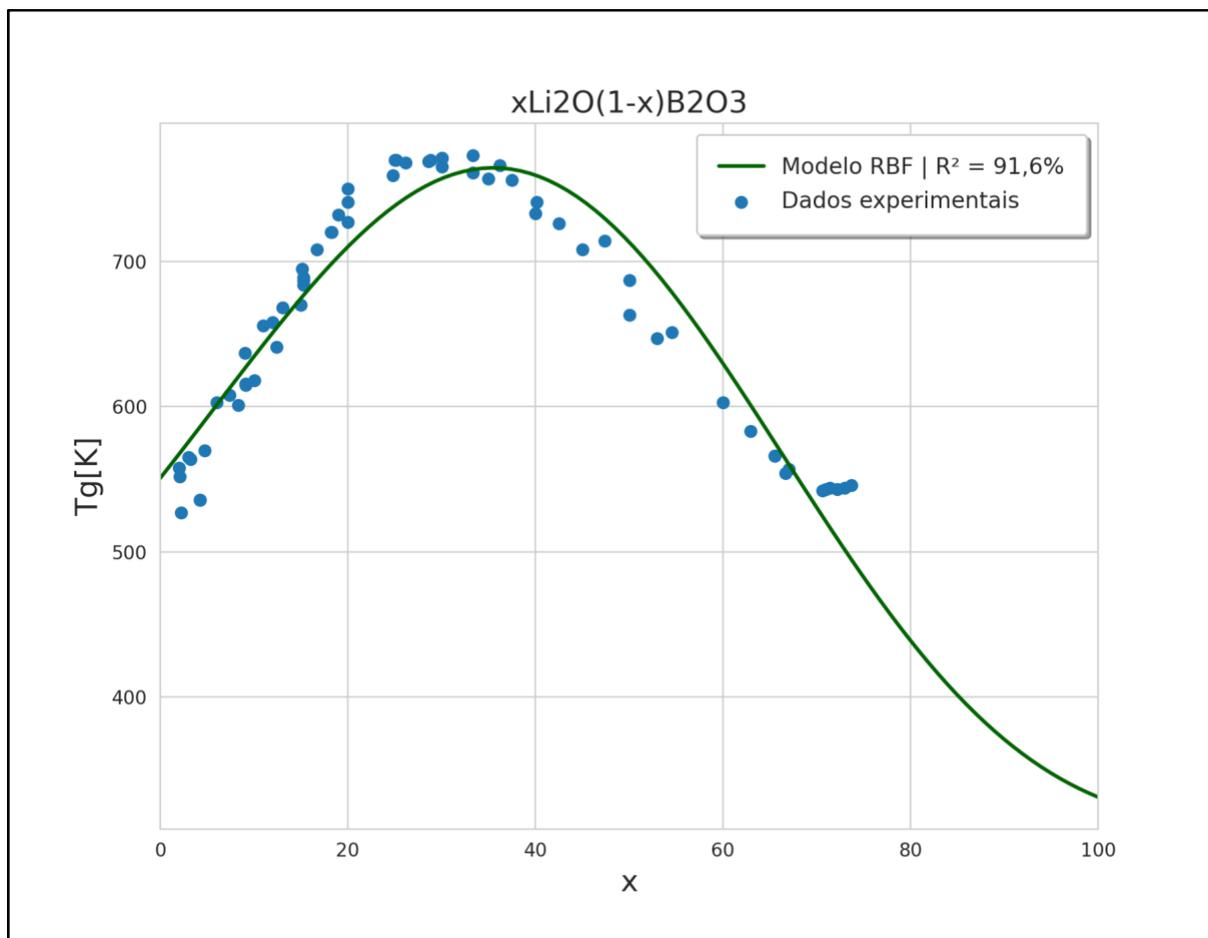


Figura 5.2 — Dados experimentais de T_g (pontos) e ajuste RBF gaussiano (linha verde) do sistema borato de lítio.

A análise da dispersão de erros, ou resíduos dos valores de previsão, encontra-se exibida na [Figura 5.3](#). Nela é possível extrair informações que dizem respeito ao processo qualitativo da rede, analisando possíveis tendências, padrões, e erros de forma mais detalhada, e que precisam ser detectados e resolvidos em uma etapa de aprimoramento quando julgado necessário.

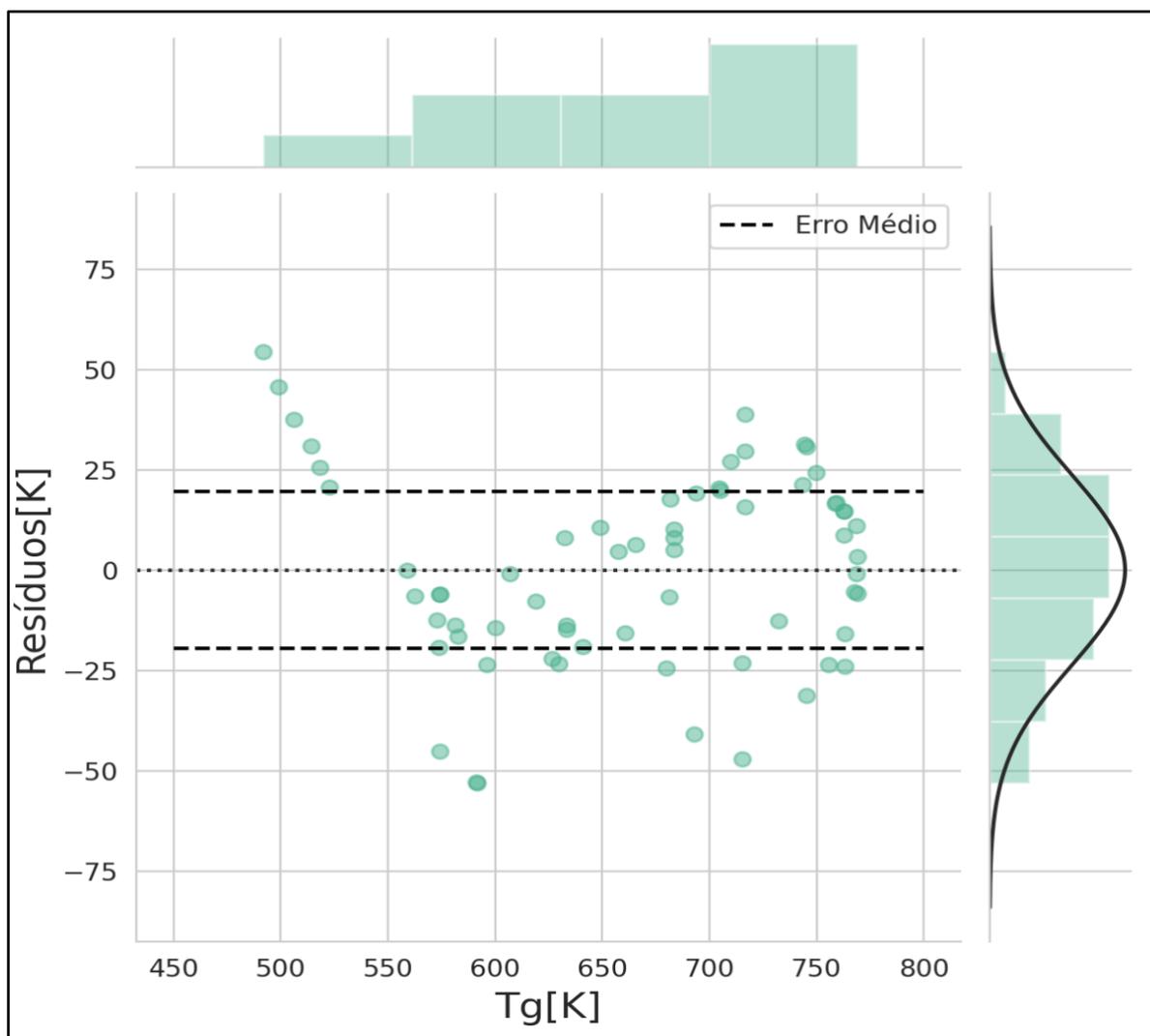


Figura 5.3 — Dispersão residual referente aos dados experimentais de T_g e erro médio calculado no sistema borato de lítio.

Observa-se que a maioria dos pontos se encontram dentro da faixa de erro determinada ($-19,6$ K a $19,6$ K), mais precisamente abrangendo 71% dos valores de previsão. Esta região descreve um intervalo de valores de T_g previstos que se estende de 542 K a 773 K com um erro médio absoluto percentual de 1,8%, caracterizando um valor errôneo para T_g em módulo de 12,3 K. Por estas circunstâncias pode-se considerar que o modelo apresentou resultados satisfatórios, principalmente ao realizar extrapolações para o sistema que estão localizados nesta região. Além do mais, não observou-se padrão ou tendência (sinos, tendências lineares positivas ou negativas) na dispersão de valores como um todo, confirmado a partir do valor da diferença de variâncias associadas a regiões distintas, caracterizando uma dispersão

de dados independentes uns dos outros, podendo ser considerada assim, como uma distribuição do tipo normal (COOK, 1982).

Para valores de T_g abaixo de 500 K, o erro médio absoluto percentual foi de 9,1%, que representa um valor de 35 K. Nesta região a rede não conseguiu capturar as características de distribuição dos valores de entradas e saídas destinadas para o seu treinamento, resultando em baixo poder de generalização. Existem motivos que justificam essa piora de performance para esta área, por exemplo, ao analisar a distribuição dos valores que foram utilizados para a caracterização das sinapses, constatou-se que apenas 0,006% dos dados tinham valores de T_g associados, abaixo de 500 K, ou seja, há uma baixa representatividade de ajuste neste intervalo, inferindo em erros mais elevados.

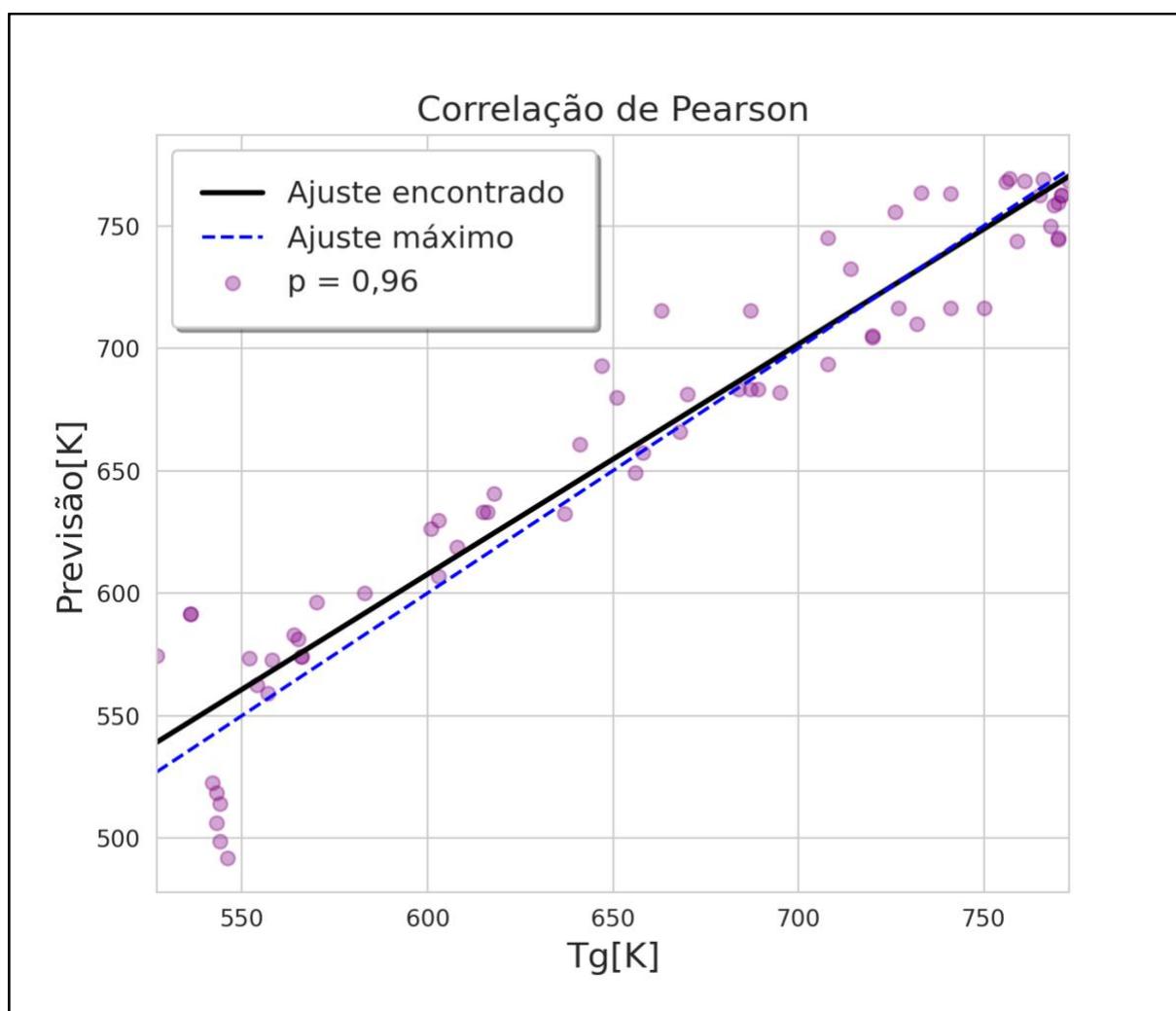


Figura 5.4 — Previsão da rede referente ao cálculo de T_g (em K) vinculado ao sistema borato de lítio.

Ao contrário da dispersão residual, onde espera-se não haver dependência entre as variáveis, o teste de correlação de Pearson (Figura 5.4) entre os valores de controle (valores reais) e variável de ajuste (valores de previsão) deve ter uma dependência linear, que neste caso, espera-se positiva. O valor obtido para este parâmetro de avaliação foi de 0,95 sendo considerada como de alta correlação (CLEOPHAS, 2018), indicando que o modelo tem suas ressalvas pelos argumentos mencionados nos parágrafos anteriores, mas que, em média, apresentou boa generalização.

O resultado para o sistema borato de sódio encontra-se exibido na Figura 5.5, onde estão apresentados a curva de ajuste obtida através da rede neural associada, e expressa através da Equação (5.2) em adição ao valor de determinação obtido a partir do conjunto de teste.

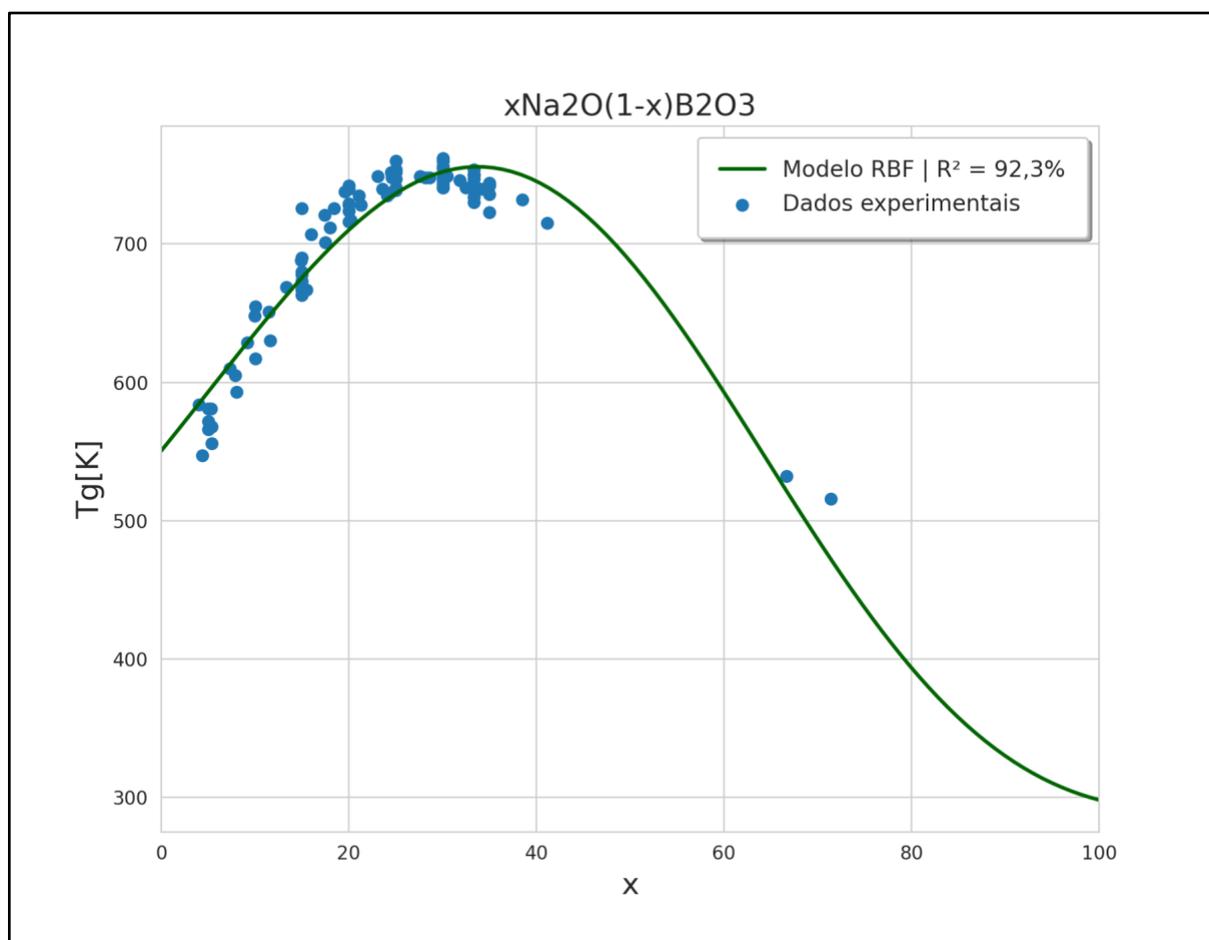


Figura 5.5 — Dados experimentais de T_g (pontos) e ajuste RBF gaussiano (linha verde) do sistema borato de sódio.

Com base na [Figura 5.5](#) é possível observar que os pontos de previsão se adequam bem aos seus valores reais associados, de acordo com o valor do coeficiente de determinação obtido de 92,3%, ou seja, a curva de ajuste ([Equação \(5.1\)](#)) definida com base nos coeficientes sinápticos, determinados pela rede de funções de base radial associada ao sistema borato de sódio, descreveu boa parte dos pontos utilizados para verificar o poder de generalização do modelo.

Percebe-se a partir da [Figura 5.1](#) que a rede associada a este sistema teve uma performance inferior na etapa de teste quando comparada ao seu treinamento (verificado através do valor de R^2); isso certamente poderia indicar que a rede teve problemas de generalização, como sobre ajuste aos dados de teste. No entanto, a curva encontrada ([Equação \(5.2\)](#)) não interceptou todos os pontos da distribuição (algo muito comum em modelos que apresentam problemas no processo de teste ocasionado por *overfitting* ([GÉRON, 2019](#))). Deste modo, conforme a [Figura 5.5](#) indica, este fenômeno possivelmente não esteve presente no resultado obtido. Ressalte-se que o erro médio percentual absoluto associado foi de 2%, que indica na escala original um erro médio em módulo de 14,2 K, sendo considerado aceitável dentro dos parâmetros estabelecidos nos capítulos anteriores; ou seja, a rede comportou-se bem na distribuição de valores de teste, apesar de possuir uma performance inferior ao seu treinamento.

A análise da dispersão dos resíduos associados ao sistema encontra-se exibida na [Figura 5.6](#) para fins de análise do processo de inferência que o modelo proposto ao borato de sódio foi capaz de realizar na sua etapa de teste, possibilitando assim a detecção de problemas associados as técnicas de análise de resíduos, como tendências de subestimação de valores, etc, por parte do ajuste proposto, e seus possíveis métodos de solução quando necessários

A solução proposta obteve no geral um erro médio absoluto percentual de 2,1%, que caracteriza em módulo um erro médio de 14,2 K. Tais valores estão representados pelas linhas tracejadas na [Figura 5.6](#). No entanto, nota-se que a maioria dos pontos da dispersão dos resíduos se encontram dentro desta faixa (mais precisamente 51,1% das previsões obtidas) onde estes valores correspondem ao intervalo de T_g que se estende de 532 K a 762 K, e que se associam a concentrações deste álcali que varia de 0% até aproximadamente 70%, garantindo assim a eficiência do ajuste proposto, já que conseguiu explicar com erro aceitável a maior parte da variabilidade dos valores de teste.

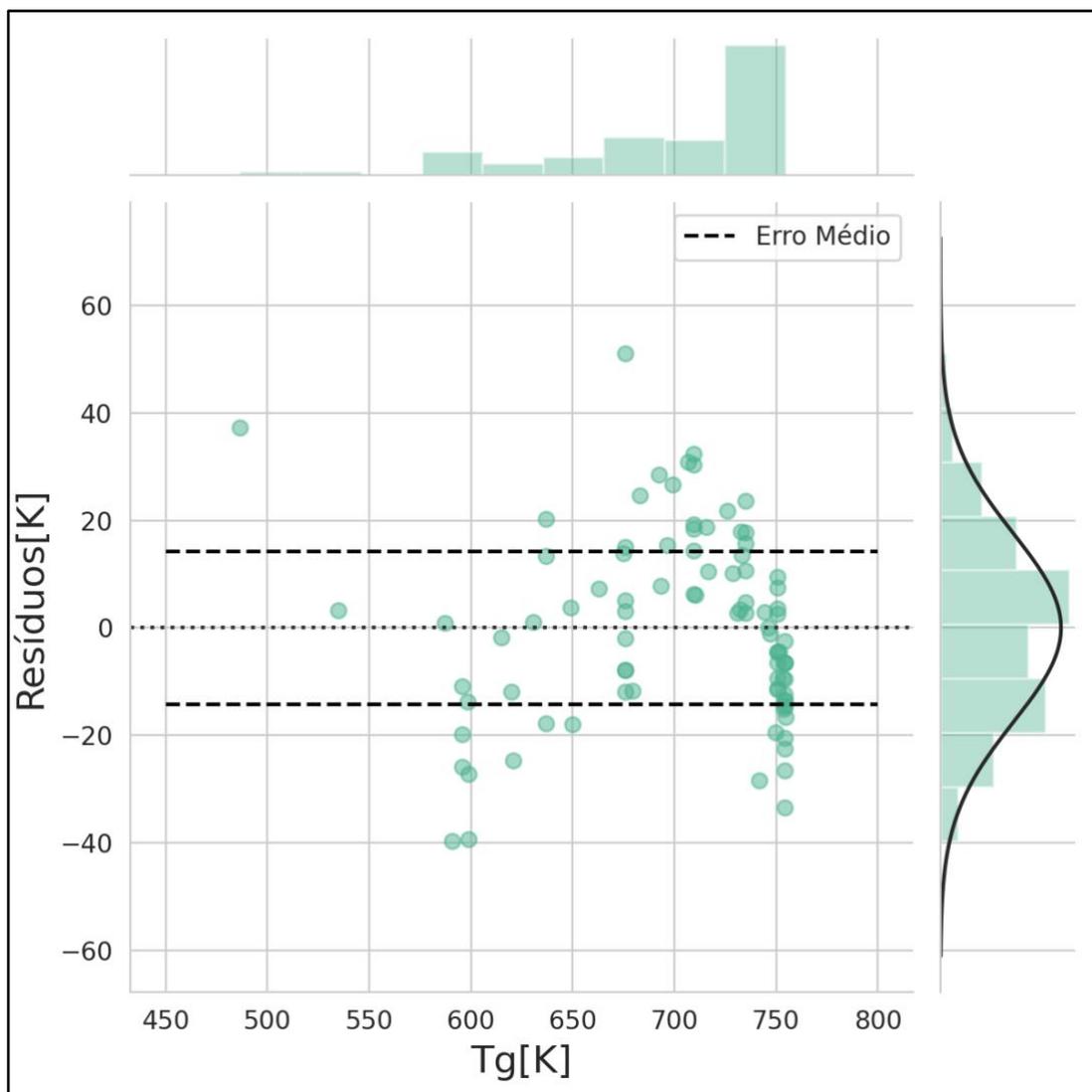


Figura 5.6 — Dispersão residual referente aos dados experimentais de T_g e erro médio calculado no sistema borato de sódio.

A distribuição que se encontra na lateral direita à [Figura 5.6](#) representa os valores dos erros em escala original que foram determinados pelo ajuste; neste caso é notório que a maior barra do histograma se encontra próxima ao valor 0 de resíduo (linha tracejada central) e que tende a diminuir conforme caminha pelo eixo y (positivamente ou negativamente) do sistema de referência. Estes aspectos caracterizam uma distribuição gaussiana, que de maneira ideal deveria estar centrada na origem (caracterizando que a maior partes dos valores possuem erros próximos de 0). Entretanto, neste caso, está levemente deslocada à direita; porém, não revelou qualquer problema no processo de inferência, nem amortecimentos para

superestimar valores de T_g (apesar dos valores de erros obtidos serem em sua maioria positivos), uma vez que não foi detectado aspectos de padrões e tendências na dispersão obtida (GASSER, 1986).

A parte que se encontra acima à Figura 5.6 consiste num histograma que faz referência a frequência de pontos associados aos valores de T_g utilizados para o teste; neste sentido, é notável que a maior parte dos valores previstos representam temperaturas acima de 700K ; estes valores referem-se a 61,1% de todos os pontos utilizados para avaliação do modelo, o que sugere uma maior representatividade definidos pelo ajuste associado (Equação (5.2)) para sistemas borato de sódio que possuem uma concentração acima ou igual a 20% deste álcali; porém, isto não implica em baixa representação da distribuição geral, e sim que existe maior precisão de generalização para estes valores.

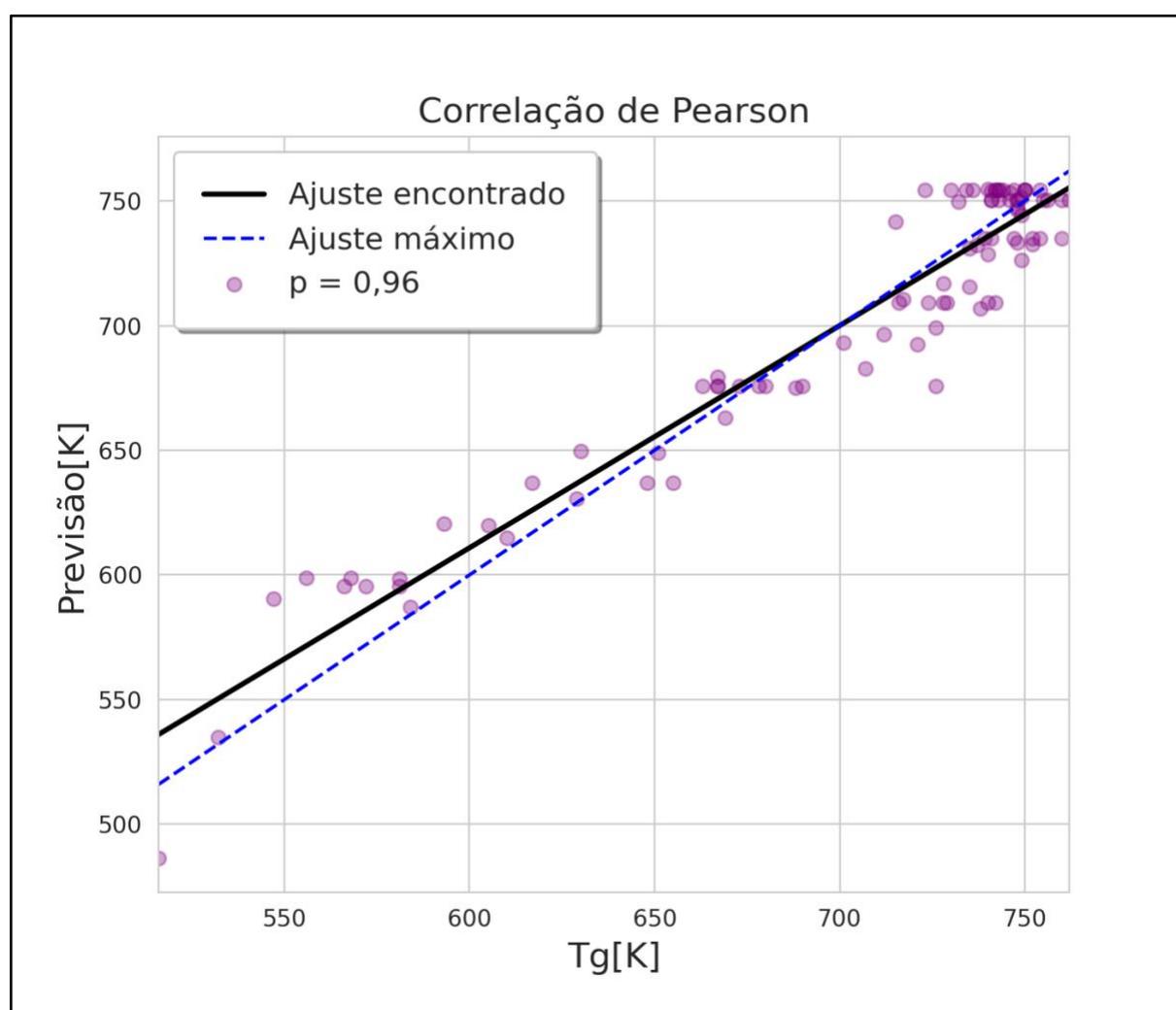


Figura 5.7 — Previsão da rede referente ao cálculo de T_g (em K) vinculado ao sistema borato de sódio.

De forma geral, o desempenho da rede de funções de base radial proposta como solução de ajuste para o sistema em questão, manteve excelentes resultados, uma vez que conseguiu extrair o comportamento da T_g associada à composição, mantendo os seus erros de previsões bem distribuídos, sem indicar regiões com maior influência negativa na análise de inferência; ou seja, não foram detectados regiões anômalas com erro muito elevado em relação ao erro médio obtido; além disto, o modelo obteve um valor de correspondência entre as variáveis preditas e seus respectivos valores reais (obtidas a partir do teste de correlação de Pearson, note [Figura 5.7](#)) de 0,96, sendo este, mais um indicativo que condiz com os já mencionados anteriormente, que dizem respeito ao bom desempenho da rede e da função de ajuste sugerida ([JAYAWEERA, 2018](#)). A linha tracejada em azul indica o modelo ideal, ou seja, variável predita, possui erro nulo.

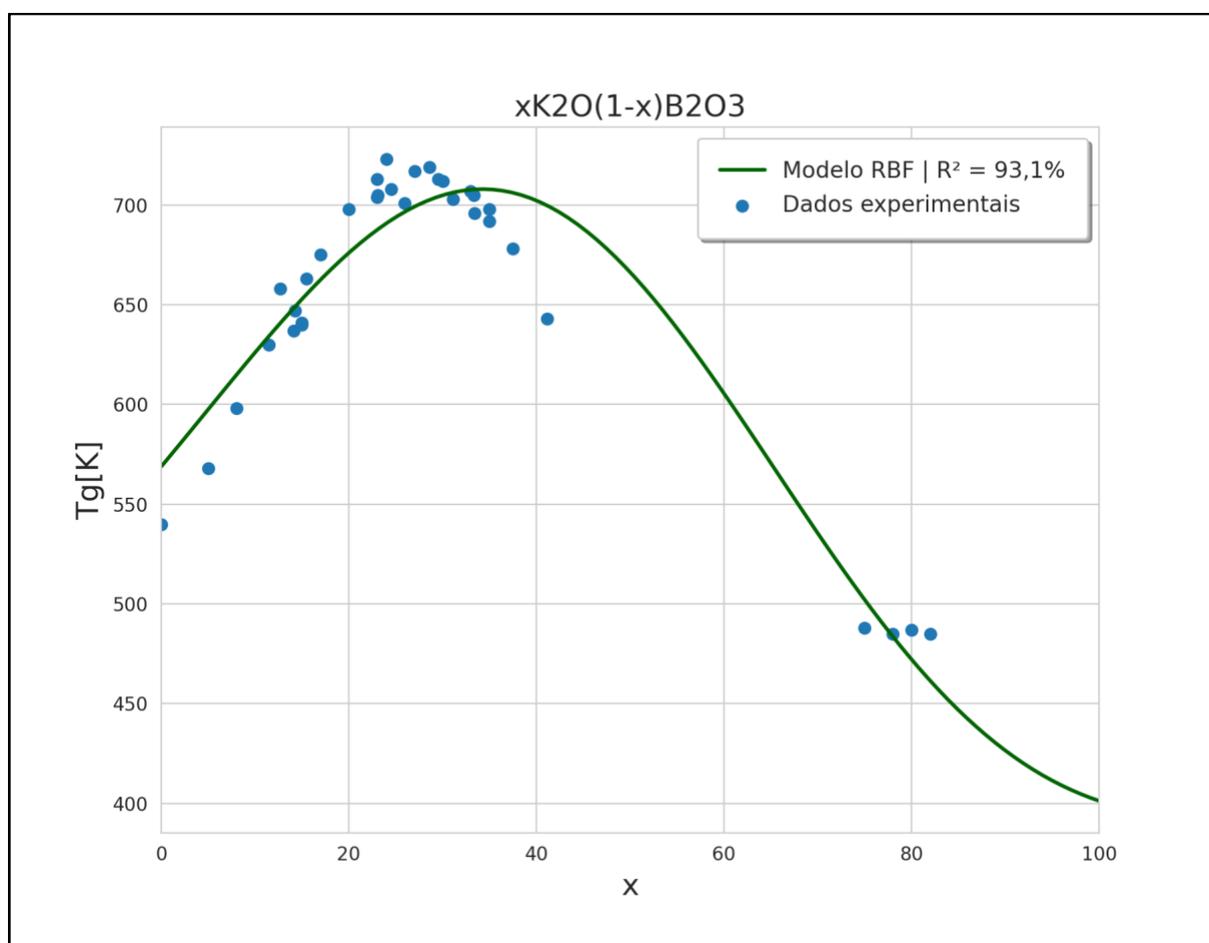


Figura 5.8 — Dados experimentais de T_g (pontos) e ajuste RBF gaussiano (linha verde) do sistema borato de potássio.

O sistema borato de potássio encontra-se descrito na [Figura 5.8](#). Nela, estão apresentados o ajuste, e os valores de T_g experimentais obtidos por meio de extração de dados do banco SciGlass.

A rede apresentou um ajuste bastante razoável pois os parâmetros determinados se adequam bem para o modelo proposto na [Equação \(5.3\)](#), confirmado a partir do valor do coeficiente de determinação obtido R^2 , de 93,1%. Entretanto, os resultados encontrados possuem um erro médio percentual absoluto de aproximadamente 2,5%, e isso traduz-se em um valor de erro médio absoluto de 15,6 K. Tal erro deve ser levado em conta quando utilizado para fins de ajustes bem específicos e de alta precisão, onde erros desta magnitude tem alta relevância no objetivo final; porém, enquanto modelo aproximativo, a rede desempenhou sua tarefa de forma bastante eficiente.

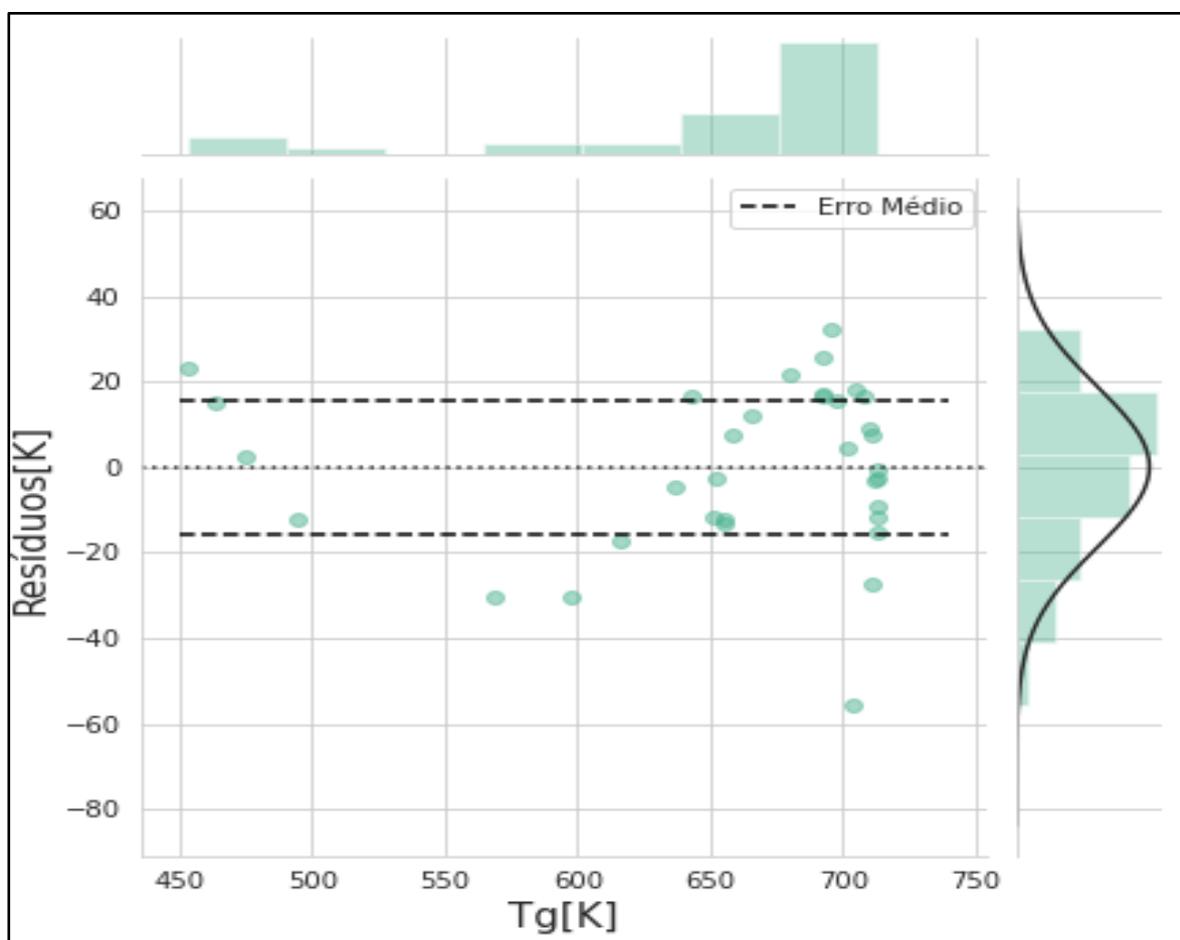


Figura 5.9 — Dispersão residual referente aos dados experimentais de T_g e erro médio calculado no sistema borato de potássio.

Outra análise importante, e de um ponto de vista mais qualitativo trata-se da dispersão residual dos valores previstos pela rede, que se encontram na [Figura 5.9](#). É possível notar que os resíduos em média se encontram dentro do canal de erro médio representado pela linha tracejada; no entanto, constatou-se que a distribuição destes resíduos não é simétrica em torno do centro da distribuição gaussiana (representado na lateral à direita da [Figura 5.9](#)), encontrando-se deslocado à esquerda e apresentando um desvio padrão de 11,6 K. Esta falta de simetria pode ser explicada pelo fato de haver valores considerados *outliers* presentes na distribuição de pontos previstos, que de forma geral tem sua origem explicada por alguns fatores, tais como: imprecisão de medidas laboratoriais, erros de coleta de informações do banco de dados SciGlass, etc ([GÉRON, 2019](#)). Tais pontos em geral trazem informações referente aos dados de entrada designados ao treino mais do que a rede e suas propriedades em si. Dessa forma, é possível considerar que a rede não tende a superestimar ou subestimar qualquer valor de previsão de maneira a amortecer seus valores positivamente ou negativamente de forma tendenciosa.

Um fato considerável é que a maioria dos pontos de previsão para este sistema são representações de um valor de T_g acima de 650 K, como demonstra o histograma da parte superior da imagem, onde o erro absoluto médio associado a esta região foi de 3,6% enquanto para a parte inferior a esse limite foi de 1,8%. Isto é um indicativo de que o ajuste não capturou tão bem as nuances para esta região da distribuição. Os motivos para isso podem estar relacionados a dois parâmetros, de maneira geral: um deles está vinculado aos dados que não são representativos para este limiar, sendo considerada, esta, a causa mais provável; e o outro diz respeito a estrutura da rede (número de polos, distribuição dos pontos de entradas para treinamento, função de ativação, etc.); em suma, trata-se de sua estrutura ([GÉRON, 2019](#)).

A [Figura 5.10](#) representa as previsões feitas pela rede em função dos valores originais de teste. Esta consiste na maneira ideal de visualizar o relacionamento entre tais variáveis, sendo um indicativo inclusive de sob ajuste ou sobreajuste ([WATT, 2020](#)). Neste caso o valor de correlação de Pearson entre as variáveis foi de 0,96 que se traduz em uma alta dependência linear positiva entre os valores, que era o esperado.

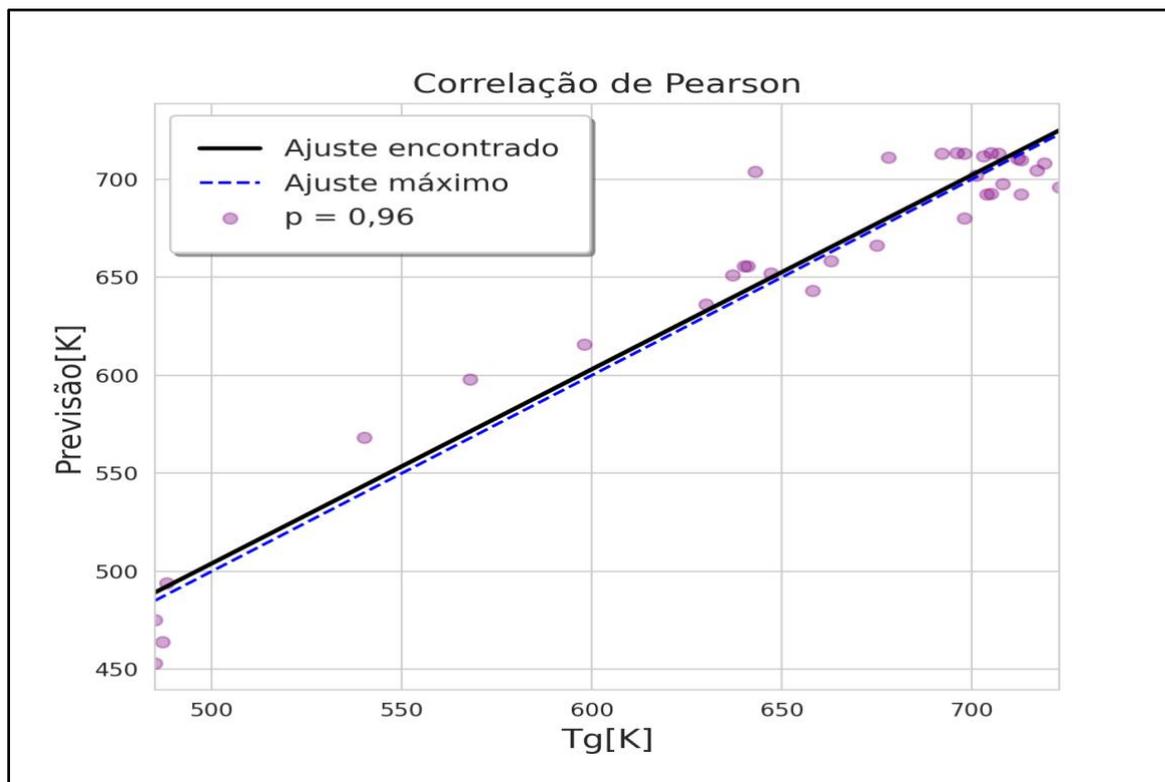


Figura 5.10 – Previsão da rede referente ao cálculo de T_g (em K) vinculado ao sistema borato de potássio.

O sistema borato de rubídio será descrito nos próximos parágrafos e figuras com o objetivo de avaliar a performance do modelo encontrado especificando as métricas determinadas e os aspectos qualitativos analisados.

O modelo encontrado e descrito através da [Equação \(5.4\)](#) se mostrou eficiente para o sistema em questão, uma vez que o ajuste apresentou um valor do coeficiente de determinação R^2 de 90,8% como mostra a [Figura 5.11](#), significando que mais de 90% da distribuição de T_g associada ao sistema e aos dados utilizados para treinamento e teste da rede, conseguiram ser explicados pelo ajuste proposto ([Equação \(5.4\)](#)).

As análises individuais da etapa de treinamento e etapa de teste associadas mostraram uma pequena melhoria no valor do R^2 de 0,07% conforme a [Figura 5.1](#), que apesar de não ser bastante significativo de um ponto de vista voltado ao ajuste proposto, indica que a rede neural elaborada não está tendo problemas de generalização evidenciados por quedas de performances no teste, e que possui bons indicadores de representatividade do sistema ([GÉRON, 2019](#)).

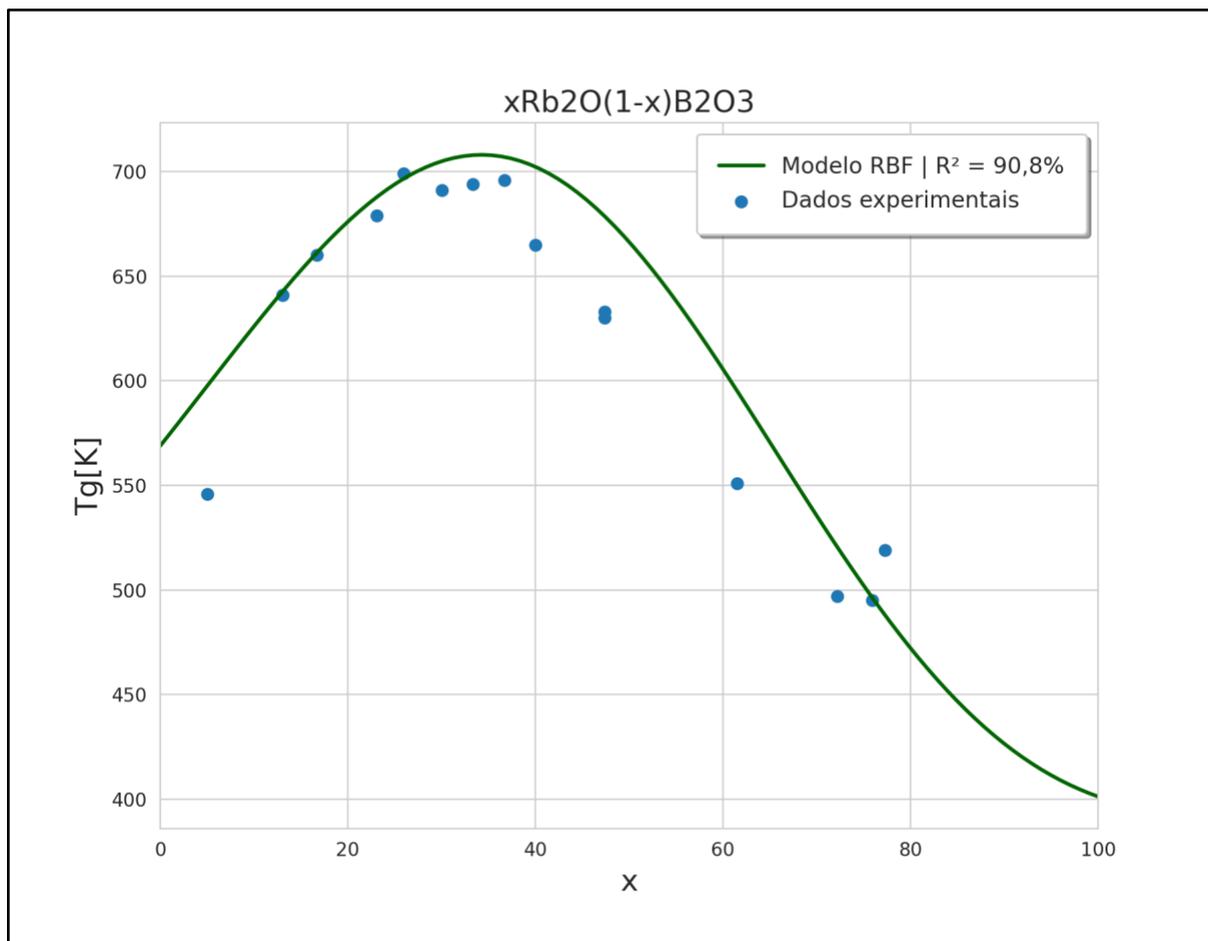


Figura 5.11 — Dados experimentais de T_g (pontos) e ajuste RBF gaussiano (linha verde) do sistema borato de rubídio.

De maneira complementar ao valor do ajuste R^2 , o erro médio absoluto percentual calculado para o modelo foi de 2,8%, que indica um erro médio absoluto e em escala de 16,6 K, revelando uma boa performance, uma vez que apesar de possuir o menor valor de R^2 dos sistemas analisados até então, o seu erro médio absoluto não acompanhou essa leve piora, pois, sistemas com coeficientes de determinação mais elevados obtiveram erros maiores em contraste, como os valores encontrados para o borato de lítio.

Como foi possível notar, é necessário analisar de maneira detalhada as previsões feitas pela rede neural, pois cada métrica revela um aspecto e atributos distintos. A utilização da dispersão residual das previsões permite avaliar de maneira minuciosa de como os erros estão distribuídos, se estes apresentam anomalias estatísticas que precisam ser resolvidas, ou até mesmo, que caracterizam a inutilidade do modelo proposto. A dispersão dos resíduos associados ao sistema se encontra na [Figura 5.12](#).

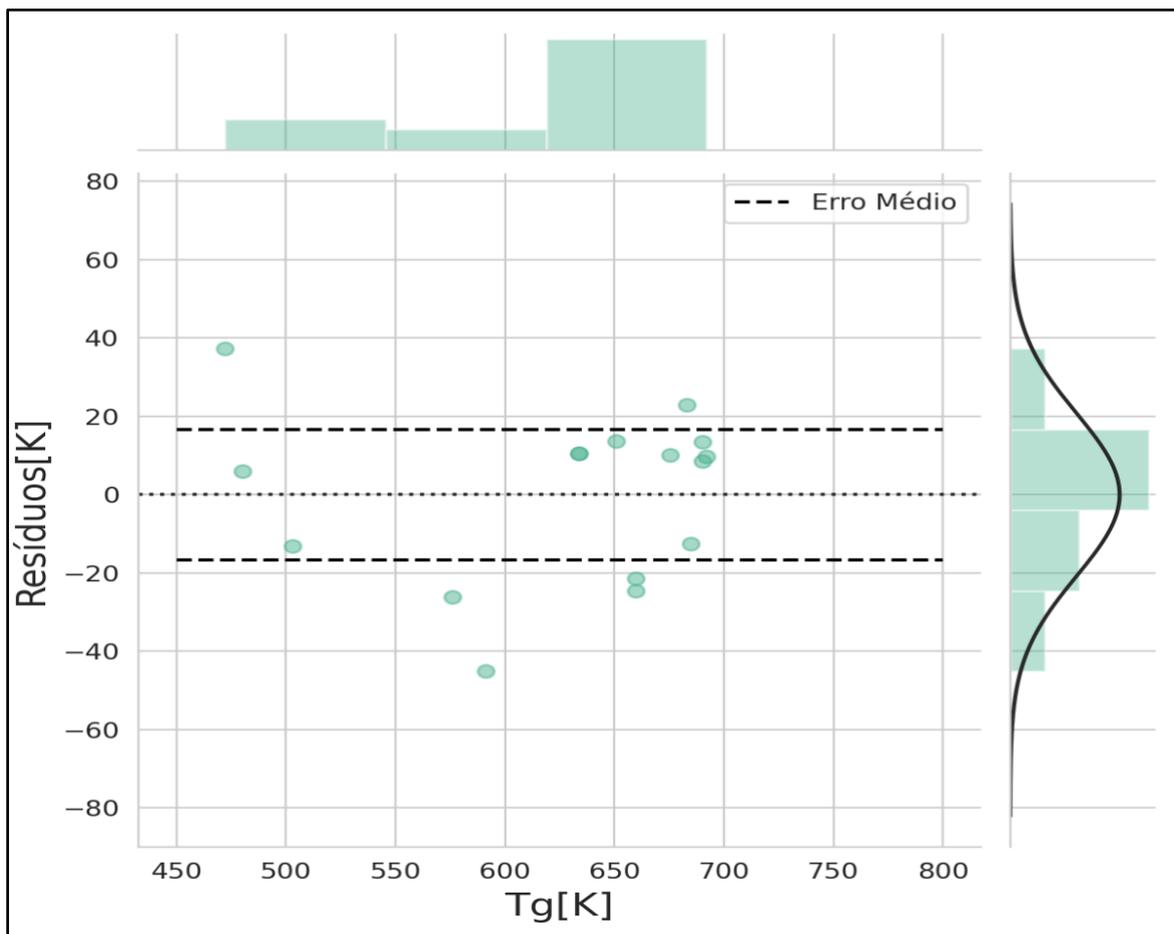


Figura 5.12 — Dispersão residual referente aos dados experimentais de T_g e erro médio calculado no sistema borato de rubídio.

Com base no histograma e na curva de distribuição associada a este, localizada à direita da [Figura 5.12](#), é possível notar que os erros se distribuem de maneira gaussiana, com um pequeno deslocamento no eixo y indicando uma leve tendência de erros em escala positiva; porém, não caracteriza anomalias deste tipo de análise, como padrões lineares, padrões de sinos, que quando encontrados sugerem correções ou até mesmo o descarte do modelo proposto ([STRAUME, 1992](#)). Apesar do indicativo encontrado através do pico da distribuição dos erros, no geral, os resíduos possuem comportamento normal cujos valores ajustados não são superestimados ou subestimados, caracterizando assim, aspectos de eficiência relacionados a função de ajuste e consequentemente a rede neural utilizada ([GÉRON, 2019](#)). Deste mesmo ponto de vista, há a presença de dois valores de T_g que foram previstos pela rede, que apresentam erros próximo ao dobro do erro médio, em módulo, na casa dos 40 K. Assim como nos casos anteriores, esses tipos de problemas possuem uma alta frequência de correlação com problemáticas presentes

no ambiente do aprendizado máquina, como por exemplo, insuficiência de instâncias para treinamento da rede, baixa representatividade dos dados amostrais, alta concentração de ruídos nas amostras, etc. (GÉRON, 2019); estes aspectos contribuem para aparecimento de pontos *outliers* no processo de teste da rede, e por estas razões a presença destes valores podem ser considerados normais, não descaracterizando a eficiência do modelo.

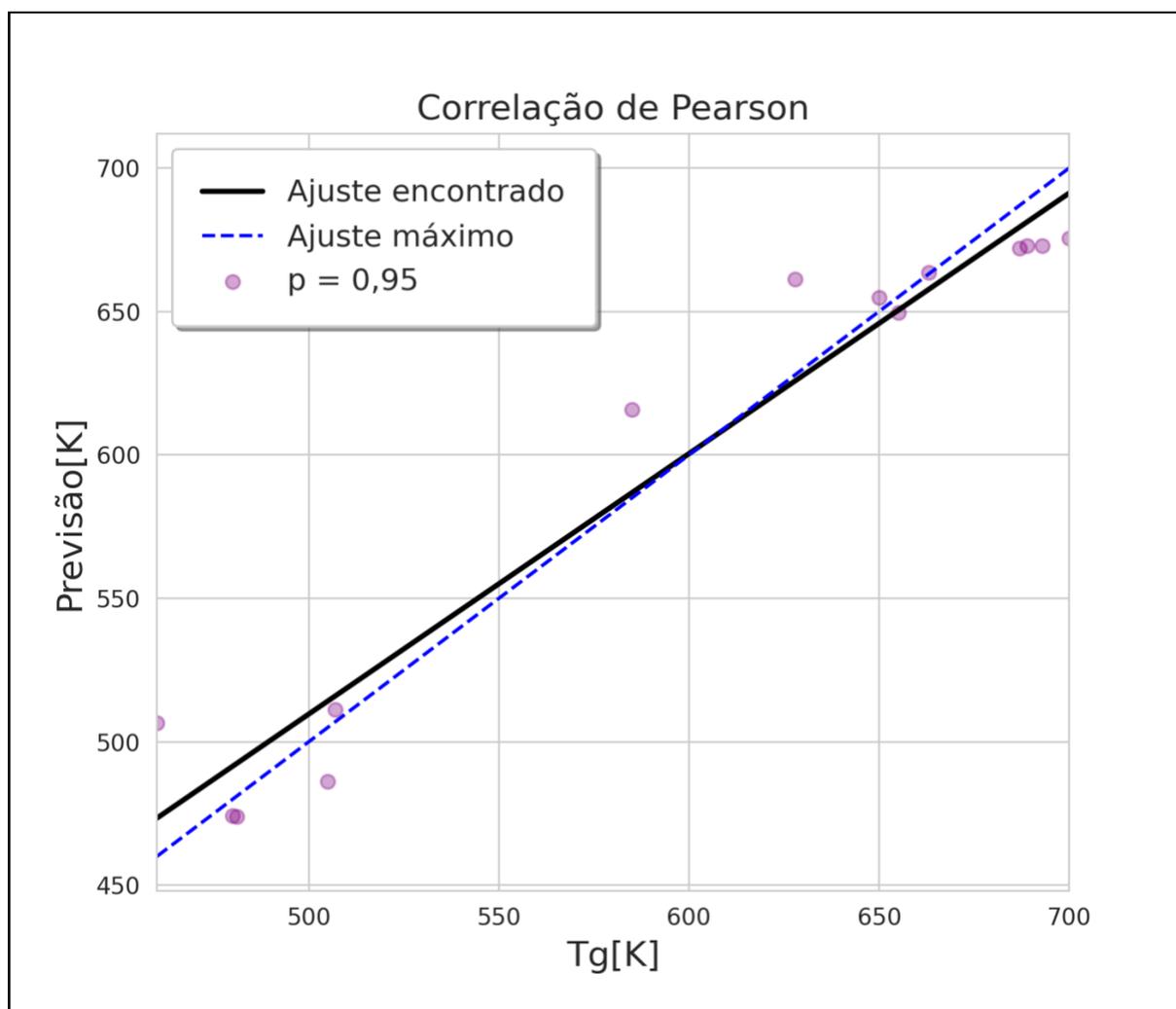


Figura 5.13 — Previsão da rede referente ao cálculo de T_g (em K) vinculado ao sistema borato de rubídio.

Na parte superior da [Figura 5.12](#) se encontra um histograma que indica a quantidade de pontos testados e seus valores de T_g associados. Observa-se que o modelo possui em proporção uma maior quantidade de valores testados para T_g que se estende em um intervalo de 600 K a 700 K, que por sua vez estão associados a uma concentração deste álcali de 5% a 68% como mostra a [Figura 5.11](#). Isso indica

que há uma excelente representatividade do ajuste encontrado, pois o teste realizado varre um grande intervalo de concentrações deste álcali, caracterizando maior representatividade para soluções que possuem estes valores mencionados, sendo complementado inclusive pelos erros associados, que se encontram próximo ao erro médio como mostra a [Figura 5.12](#) (ANSCOMBE, 1963).

Em síntese, a rede neural utilizada realizou sua tarefa de forma eficiente no que diz respeito ao ajuste encontrado, e associados a este, o coeficiente de determinação, erro médio, coeficiente de Pearson ([Figura 5.13](#)).

Por fim, analisamos o sistema borato de céσιο com o objetivo de avaliar a performance do modelo encontrado, especificando as métricas e aspectos qualitativos.

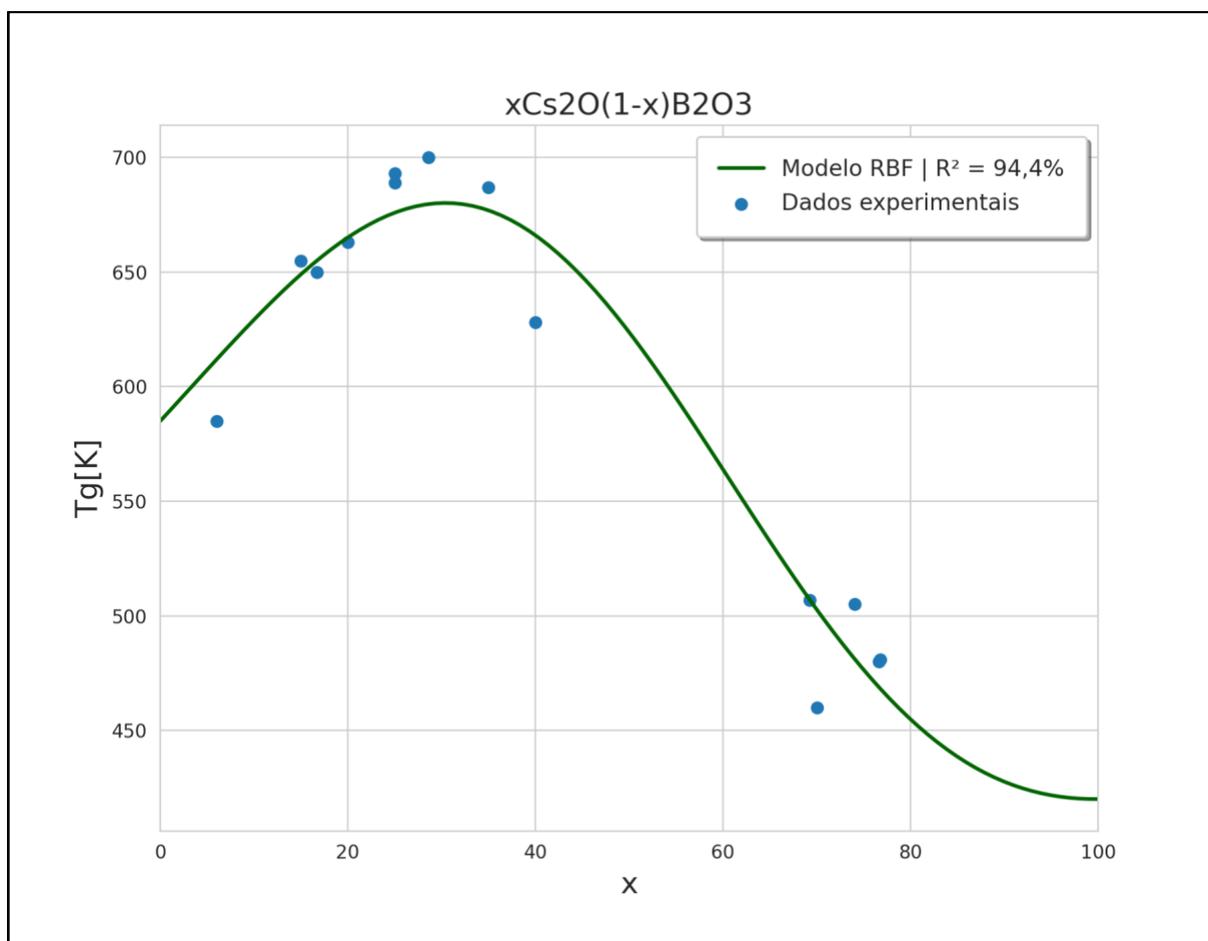


Figura 5.14 — Dados experimentais de T_g (pontos) e ajuste RBF gaussiano (linha verde) do sistema borato de céσιο.

Com base na [Figura 5.14](#) pode-se considerar que foi possível obter um ajuste razoável, já que as extrapolações encontradas através da utilização do conjunto de

teste caracterizam um valor do coeficiente de determinação R^2 de 94,37%, indicando que a função determinada [Equação \(5.5\)](#) é representativa da distribuição de T_g original, e que consegue explicar grande parte do sistema analisado.

O modelo obteve uma melhora na etapa de teste quando comparada ao seu treinamento, e este fato ressalta que a curva de generalização encontrada através dos coeficientes sinápticos não está sobre ajustando os dados de maneira a armazenar pontos de treinamentos e repeti-los em forma de previsão na etapa de teste, caracterizando assim este fenômeno de sobre ajuste ([GÉRON, 2019](#)).

Não somente o valor do coeficiente de determinação caracteriza o modelo como satisfatório, como o erro médio absoluto percentual pode ser considerado um indicativo de boa performance, já que neste caso, foi obtido um valor de 2,9%; ou seja, na escala do sistema estudado e em módulo, a rede apresentou um erro de 15,6 K, mesmo possuindo apenas 14 pontos que constituem os dados de teste reservados para as análises estatísticas, previstas no referencial teórico como avaliadoras deste sistema, informando o grau de “aprendizado” das redes de funções de bases radiais construídas através de arquiteturas simples e aplicadas a sistemas de baixa dimensão, como os binários representados aqui.

A análise gráfica da dispersão residual associadas ao sistema encontra-se exibida na [Figura 5.15](#) como forma de complemento à análise qualitativa do modelo encontrado pela rede e ajustado através da [Equação \(5.4\)](#), permitindo assim uma análise mais minuciosa e detalhada dos erros, onde é possível encontrar aspectos problemáticos e permitem elaborar soluções com base nestes problemas.

Com base no histograma na parte direita da [Figura 5.15](#), é possível notar que existe uma grande quantidade de pontos (em relação ao total de instâncias associadas ao sistema) localizadas próximos ao erro médio absoluto enquanto três pontos se encontram distantes destas margens com erros acima de 30 K em módulo, isso caracteriza uma distribuição gaussiana cujo pico não está localizado próximo ao centro do sistema de referência que indicaria que o menor erro foi atingido.

A presença destes pontos com erros que se aproximam duas vezes o valor do erro médio absoluto pode estar sendo provocado por alguns fatores que na maioria das vezes estão associados a estrutura de dados completa e sua repartição para treinamento e teste, e os parâmetros de arquitetura da rede elaborada ([MARLIN, 2008](#)).

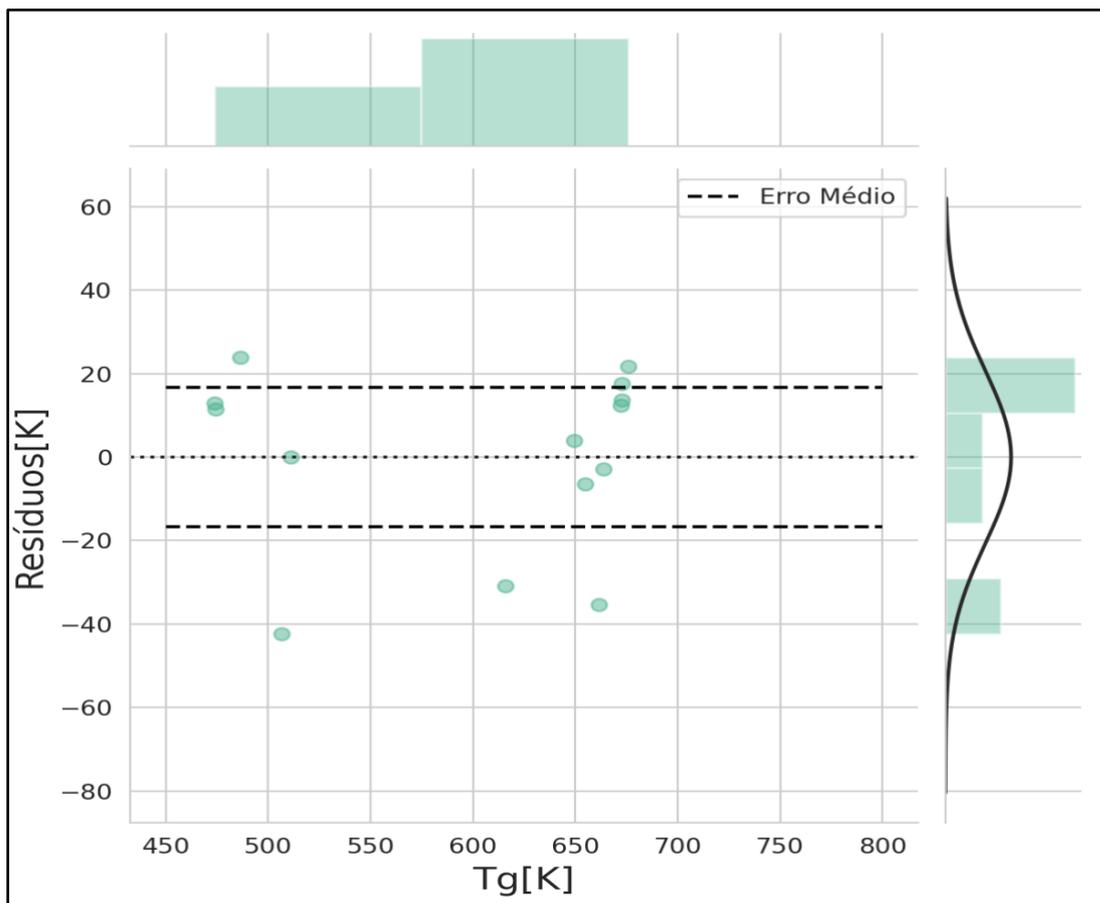


Figura 5.15 — Dispersão residual referente aos dados experimentais de T_g e erro médio calculado no sistema borato de césio.

No que diz respeito a estrutura de dados, o fato da partição ser feita de maneira aleatória promove uma heterogeneidade em ambos os conjuntos, e como consequência há regiões de maior descontinuidade nestas distribuições. Isso implica que os coeficientes determinados através da rede neural não conseguem representar regiões específicas dos dados, uma vez que não puderam capturar ou aprender com os dados de treinamento, ocasionado pela ausência de valores de representatividade destas áreas (GÉRON, 2019).

Um outro fato que pode ter ocasionado estes erros mais elevados estão ligados à arquitetura da rede neural, e de uma maneira mais específica, aos polos associados a esta, já que a localização destes é uma função que depende inteiramente da distribuição dos valores de treino e suas propriedades, conforme explicado no referencial teórico (capítulo 4). De maneira simples, as redes neurais de funções de base radial possuem uma sensibilidade característica aos dados de treinamento, pois

estes impactam diretamente nas posições dos polos e como consequência, a eficácia da rede neural (CHOLLET, 2021; HAYKIN, 2007).

Via de regra, o sistema borato de céσιο obteve bons resultados em todas as métricas avaliadoras propostas, apesar de possuir poucas instâncias para realizar sua validação. O coeficiente de Pearson para este sistema foi de 0,97, o que mostra razoável relação linear entre as variáveis alvo conforme a Figura 5.16, reafirmando os argumentos já mencionados. No entanto um conjunto com maior pontos de representação é necessário para atribuir maior fidelidade aos coeficientes sinápticos encontrados, através de teste com mais pontos, ou um novo treinamento, possibilitando, assim, a utilização mais assertiva do ajuste proposto (Equação (5.5)).

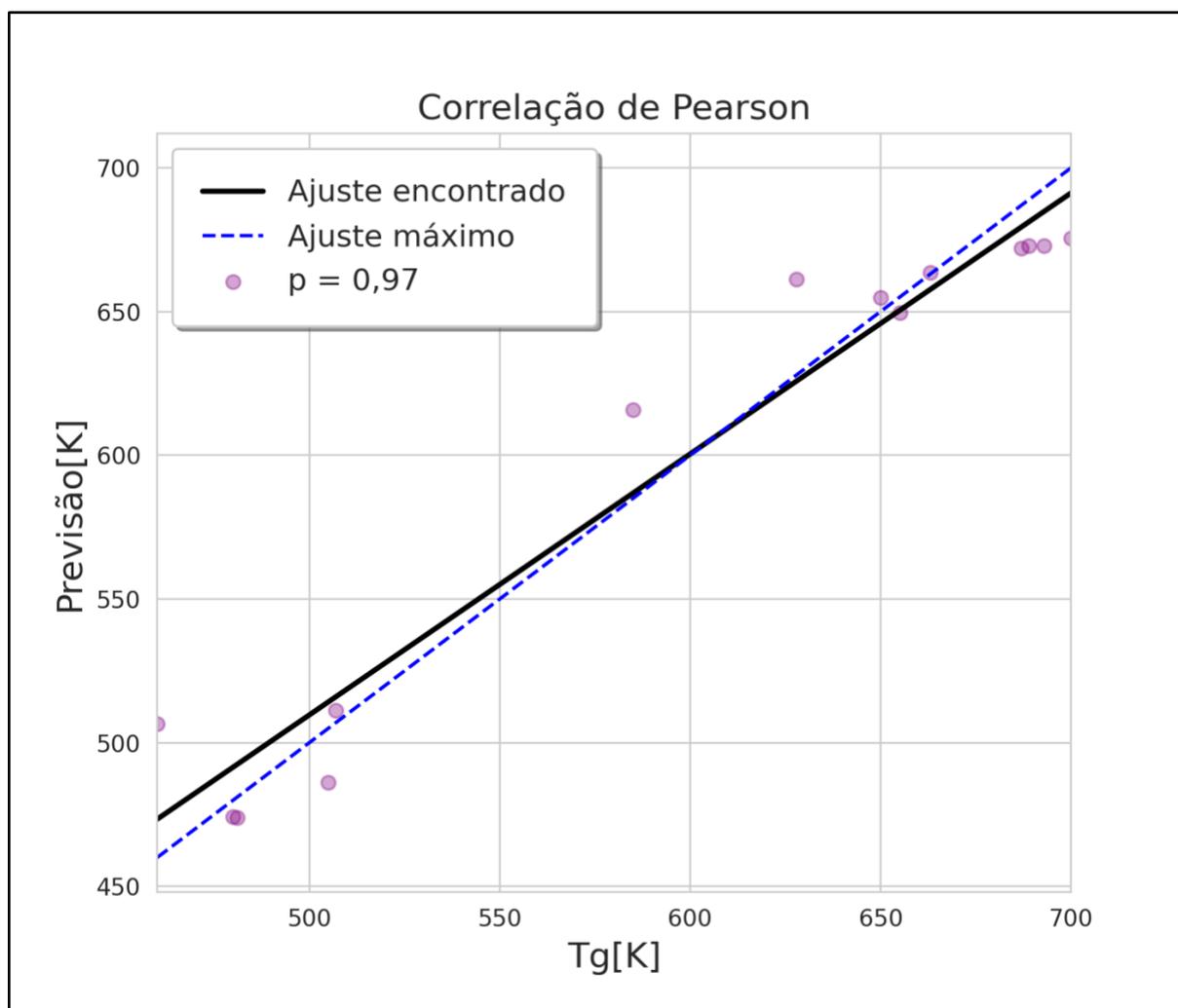


Figura 5.16 — Previsão da rede referente ao cálculo de T_g (em K) vinculado ao sistema borato de céσιο.

5.1 ANÁLISE COMPARATIVA DOS AJUSTES

No geral, nota-se que a variação da T_g em função da composição nos sistemas alcalino boratos comportam-se de modo muito similar, conforme [Figura 5.17](#). Não à toa, os ajustes RBF foram muito próximos, conforme resultados apresentados na [Tabela 5.1](#).

É possível notar a partir das [Figuras 5.2, 5.5, 5.8, 5.11 e 5.14](#) que existe um ponto de inflexão que se encontra em uma região de concentração molar dos modificadores de rede numa faixa de 25% a 33%. Percebe-se também que as funções de ajustes são monotonicamente crescentes em relação ao valor da mesma concentração molar, se estendendo até o ponto onde a derivada de segunda ordem associada tem seu sinal invertido, modificando seu comportamento, como mostra a [Figura 5.17](#).

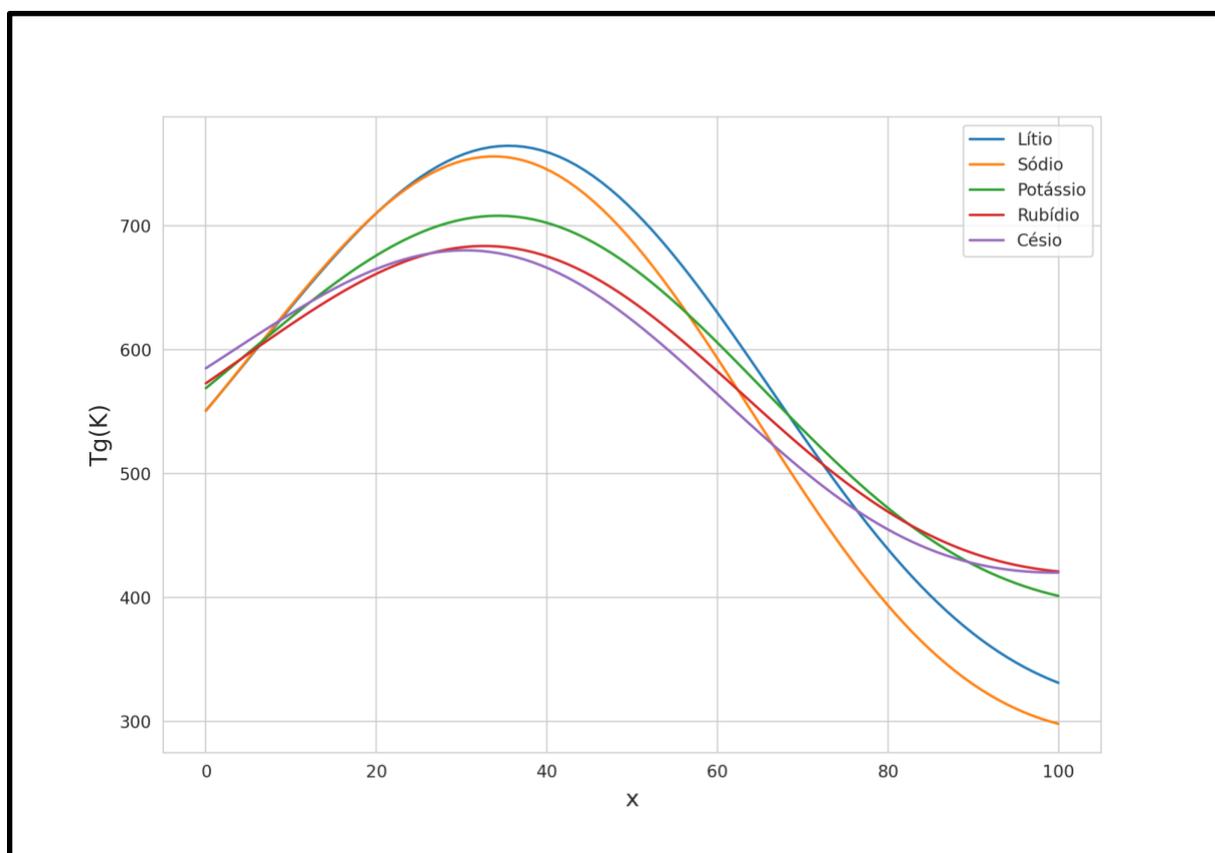


Figura 5.17 — Ajuste obtido para os cinco sistemas alcalino-boratos analisados.

Essa característica das funções obtidas, que descreve a temperatura de transição vítrea para cada sistema isolado, já foi amplamente discutida em trabalhos

que envolvem vidros alcalino boratos binários, e alguns autores, inclusive, atribuem a sistemas mais complexos, como os de composições ternárias, um comportamento aproximado (ZHONG e BRAY, 1988).

Vale destacar que pelo fato do mecanismo de funcionamento das redes aqui apresentadas se da via interpolações de autofunções de base radial de um operador auto adjunto, então para cada termo que será expandido em função desta base existe um ponto de inflexão associado. Isso implica dizer que, com base em outros estudos de caracterização de T_g onde, no geral, o comportamento encontrado para este fenômeno apresenta dois pontos de inflexão (AVRAMOV, VASSILEV e PENKOV, 2004; DECEANNE *et al.*, 2017), então é possível considerar uma relação direta entre o número de polos vinculado e tais pontos de inflexão, uma vez que tais dois polos, em teoria, garantiriam dois pontos de inflexão ao ajuste, sendo esta uma característica da T_g de boratos alcalinos binários, o que faz a escolha de dois neurônios para a rede neural ser considerada adequada aos sistemas em questão (UHLMANN, KOLBECK e WITTE, 1970).

A razão pela qual esses padrões sempre são encontrados nos modelos de alcalino boratos analisados deve-se ao fato do comportamento peculiar destes sistemas, e que se opõe aos sistemas silicatos quando se discute T_g , um problema denominado de anomalia do boro. De maneira simplificada, esse fenômeno ocasiona uma mudança no nível de coordenação atômica, de 3 (estruturas triangulares), para 4 (estruturas tetraédricas), onde surgem os chamados oxigênios não ponteantes, da sigla em inglês NBO, ocasionados devido a adição de metais alcalinos modificadores de rede (conforme apresentado no Capítulo 3). Compreende-se que as grandezas físicas têm alta correlação com essas variações no nível de coordenação associado, e em especial, a T_g (MAJÉRUS *et al.*, 2003; KERNER, 1997). Em linhas gerais, a Figura 5.19 apresenta esta tendência, conforme descrito adiante.

Alguns modelos físicos-matemáticos, com bases em diversas teorias, foram elaborados com o intuito de entender o comportamento da temperatura de transição e demais propriedades físicas destes tipos de materiais, entretanto, para Kojima (2020) o conhecimento científico a respeito ainda não é suficiente. Kauzmann (1948) sugeriu que a temperatura de transição vítrea está intimamente ligada à temperatura de fusão através da relação empírica $T_g \sim \frac{2}{3} T_{fusão}$.

Mauro, Gupta e Loucks (2009), propuseram um modelo topológico de previsão para temperatura de transição vítrea em função da composição molar x ,

especificamente para líquidos alcalinos borato, baseada em um conceito denominado por eles de “*restrições dependentes da temperatura*”. O modelo se aplica a qualquer sistema alcalino borato desde que se tenha conhecimento do maior valor de temperatura de transição vítrea $T\beta$ intrínseca a estes. Os ajustes determinados em suas propostas com $T\beta = 760\text{K}$ e $T\beta=740\text{K}$ para o borato de lítio e borato de sódio respectivamente, se encontra na [Figura 5.18](#).

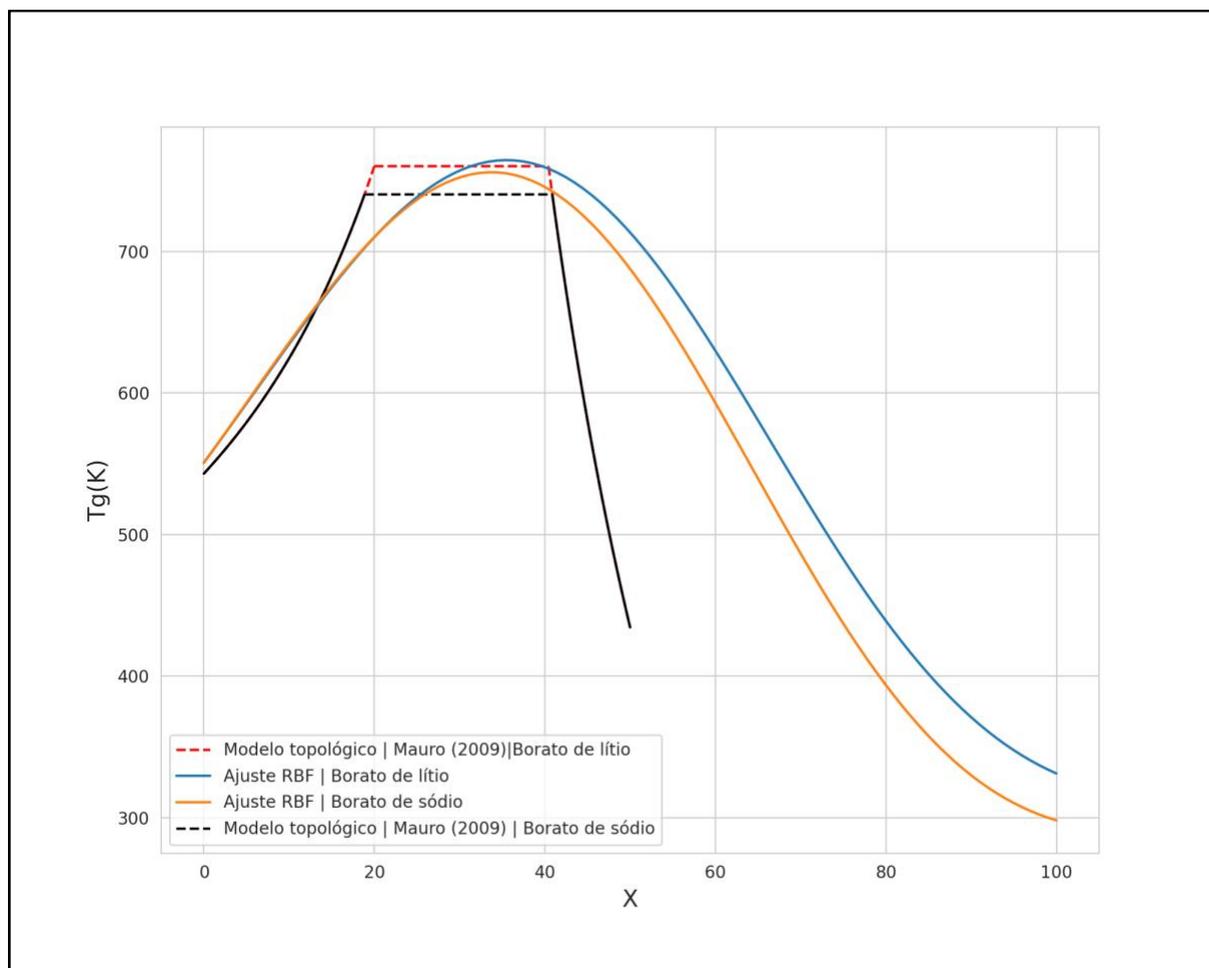


Figura 5.18 – Comparação entre os modelos topológicos de [Mauro, Gupta e Loucks \(2009\)](#) e modelo de redes de função de base radial dos sistemas borato de lítio e sódio.

Como nota-se da [Figura 5.18](#), existe razoável concordância entre o modelo topológico e o modelo de funções de base radial, principalmente no que diz respeito ao intervalo próximo de até 20% de concentração molar alcalina de lítio e sódio, onde o ajuste topológico alcançou valores R^2 de 94% e 89% respectivamente, por meio da base de dados utilizada neste trabalho. Estes resultados estão de acordo com a literatura e representam ótimos ajustes. O patamar teórico previsto entre 20% e 40%

também corresponde aos respectivos ajustes na [Figura 5.17](#), onde se atinge um máximo de forma suave para todos os alcalino-boratos analisados.

O fato interessante que surge do modelo elaborado por [Mauro, Gupta e Loucks \(2009\)](#) é a relação proposta entre os números de quantidades estruturais $N(Q_i)$ devido a adição de modificadores de rede, particularmente para teores de {Li, Na}, e que pode ser estendido para os demais metais alcalinos deste estudo. É nítido que a alteração do nível de coordenação associado para estruturas tetraédricas $N(Q_4)$ e a diminuição da quantidade de átomos de oxigênios ponteantes $N(Q_3)$ descritos sucintamente no item [3.2](#) deste trabalho ocasiona um patamar máximo de temperatura de transição vítrea em cerca de 33% (observe o ponto de intersecção entre $N(Q_4)$ e $N(Q_3)$, e que também é comum no eixo x para $N(Q_2)$), que vai de encontro com os modelos elaborados via redes de funções de base radial apresentados nesta dissertação; isso significa que a rede no geral capturou de maneira independente a presença de tais estruturas do modelo supracitado, bem como seus efeitos, conhecido como anomalia do boro. A [Figura 5.19](#) ilustra bem a quantidade de estruturas $N(Q_i)$ em termos da fração molar dos modificadores.

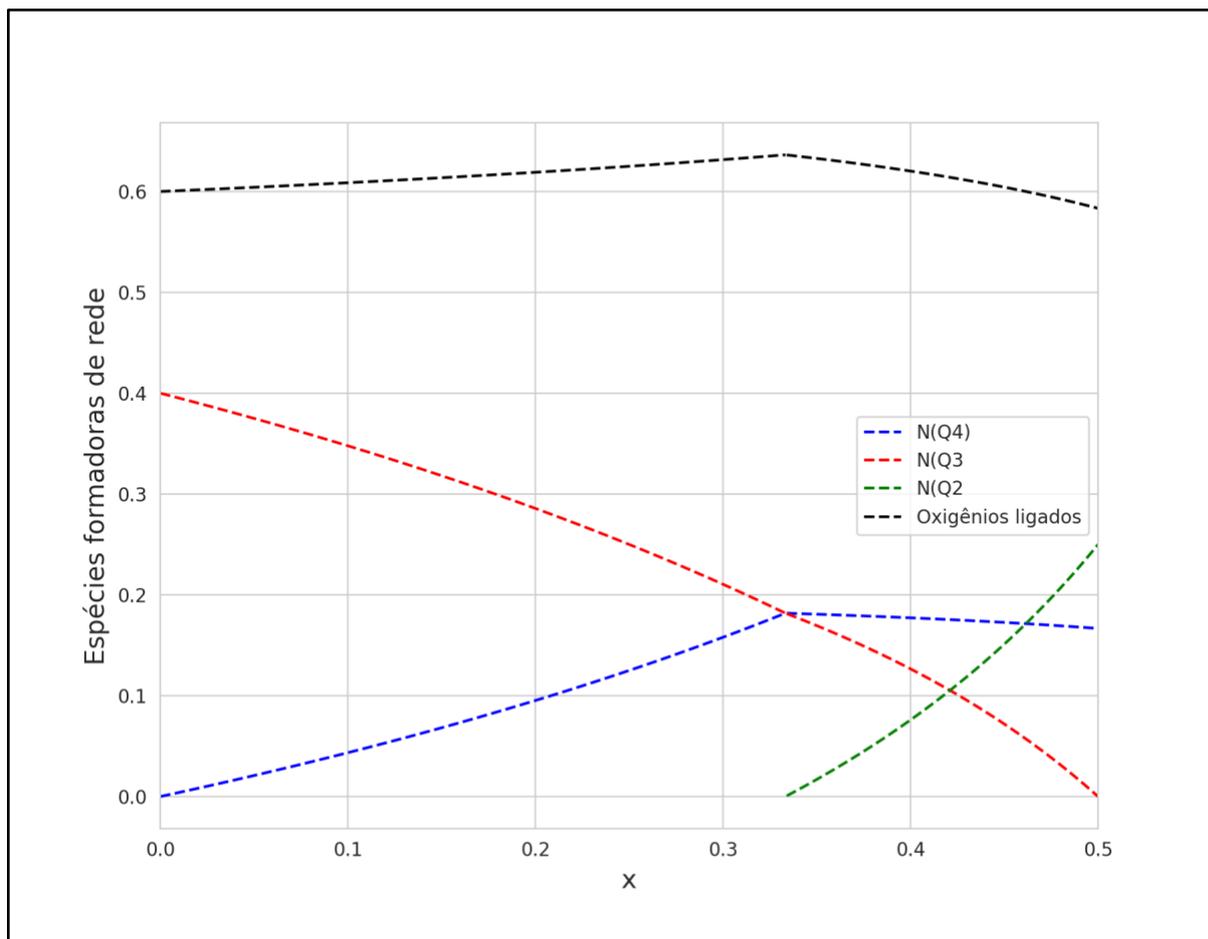


Figura 5.19 — Quantidade de estruturas formadoras de redes em função da concentração molar de modificadores, de acordo com Mauro, Gupta e Loucks (2009).

Desta forma, como o comportamento da T_g dos sistemas alcalino boratos em função da concentração molar x de óxidos em estudo é similar, e este corresponde ao modelo topológico de Mauro, Gupta e Loucks (2009), estabelecido apenas para os sistemas boratos de lítio e sódio, pode-se assumir, enquanto resultado parcial, ser possível propor a extensão do modelo topológico para os demais metais alcalinos, a saber, de potássio, rubídio e cério. Ressalte-se a necessidade de mais estudos a respeito.

Mais recentemente, Cassar, Carvalho e Zanotto (2018) realizaram um trabalho pioneiro na área das ciências dos materiais. Os mesmos desenvolveram uma rede neural do tipo Perceptron multicamadas para realizar previsões de T_g . Utilizando 55.150 dados de T_g de amostras considerando entre 3 e 21 de até 45 elementos distintos em suas composições, retiradas do banco de dados SciGlass, em soma a uma rede neural com duas camadas ocultas contendo 60 e 50 neurônios respectivamente. Obtiveram excelentes resultados com ajuste próximo de $R^2 = 99,8\%$.

Um fato interessante sobre os resultados obtidos por [Cassar, Carvalho e Zanotto \(2018\)](#) diz respeito aos desvios relativos encontrados, que se aproximou bastantes de valores considerados como ruídos pelos autores, presente nas amostras utilizadas, levantando um ponto problemático associado aos dados publicados: os diferentes valores de temperatura para uma mesma composição.

A presença de diferentes valores de certa propriedade para uma mesma composição já é bem discutida e definida nos estudos sobre vidros, porém, quando estes mesmos dados são analisados através de processos de inteligência artificial, existe um problema bastante comum que está diretamente associado a isso, o sobreajuste devido a *outliers*. [Cassar, Carvalho e Zanotto \(2018\)](#) contornaram esse problema por meio do uso de uma função de custo bastante robusta devido a extensão de suas amostras, e como uma forma de compensar a quantidade de neurônios utilizados, diferente da rotina empregada nas redes desenvolvidas neste trabalho, pois com menor quantidades de neurônios, e um número menor de instâncias, a presença de ruídos amostrais foi reduzida, já que é mais fácil de filtrá-los e alimentar a rede com entradas fiéis. O resultado disso encontra-se refletido nos desvios relativos, presentes nos dados utilizados para teste, que em média foi de 3,1%

A construção das redes propostas neste trabalho podem ser consideradas distinta das comumente apresentadas na literatura, pois a utilização das mesmas não se darem apenas por via computacional através do algoritmo, já que é possível realizar a tarefa de previsão ponto a ponto utilizando as funções expostas de ajuste para cada um dos cinco sistemas analisados, sendo esta, mais uma vantagem do recurso adotado.

Por fim, as redes de funções de base radial são ferramentas com poder analítico e descritivo de alta robustez em análise comportamental de T_g de sistemas alcalinos boratos binários, pois foi capaz de ajustar as funções dos cinco sistemas diferentes com métricas adequadas, além de conseguir capturar as nuances da anomalia do boro de maneira independente, utilizando apenas dois polos e indo a favor do modelo estrutural proposto por [Mauro, Gupta e Loucks \(2009\)](#).

5.2 TESTE DAS FUNÇÕES DE ATIVAÇÃO MULTIQUADRÁTICA E MULTIQUADRÁTICA INVERSA

Como forma complementar, foram testadas funções de base radial para os casos em que as funções de ativação fossem as multiquadráticas (Equação (2.2)) e multiquadráticas inversas (Equação (2.3)).

Os resultados obtidos para estes casos em especial revelam uma ligeira inferioridade quando comparada a função de ativação gaussiana (Equação (2.1)); porém, no geral, apresentam números satisfatórios para o caso de apenas dois neurônios. Um fato interessante foi a similaridade dos valores R^2 para estes casos especiais, isso porque, basicamente, alguns coeficientes do ajuste são iguais, e merecem maior aprofundamento.

$\varphi(\vec{x})$	Lítio	Sódio	Potássio	Rubídio	Césio
Gaussiana	91,28	91,04	93,41	91,44	93,54
Multiquadrática	89,66	92,47	89,36	87,78	90,09
Multiquadrática Inversa	89,37	92,47	89,66	87,78	90,09

Tabela 5.2 – Ajuste R^2 para os demais tipos de função de ativação neuronal.

Quanto a leve vantagem assumida pela a ativação gaussiana, há a probabilidade de estar relacionada a sua propriedade de aproximadores de funções universais (ORR, 1996); isso leva a crer que o mapeamento multidimensional através da mesma garante vantagens quando comparados aos testes de suas irmãs de base radial, pois fenômenos físicos naturais geralmente podem ser aproximados por termos gaussianos.

5.3 ANÁLISE GERAL DOS RESULTADOS EM FUNÇÃO DOS POLOS

Em termos dos conceitos a respeito de redes neurais, observa-se que as funções de base radial possuem características especiais em relação aos seus hiperparâmetros, tais como: desvio padrão das amostras, número de polos, e o de maior peso ao resultado final, qual seja - a localização dos polos.

Mais precisamente, os vetores de posição dos polos carregam o papel de informar a distância de cada elemento dos vetores de características ao centro das funções de base radial através da norma associada, e isso está diretamente relacionado ao valor de saída emitido como resposta, uma vez que quanto menor a

norma, maior será o sinal emitido, ou neste caso, maior será a temperatura de transição vítrea.

O polo consiste na característica de maior relevância local, pois o valor máximo da distância entre seus elementos pondera a resposta da rede ao estímulo dos valores de entrada.

Testes preliminares indicaram que uma manipulação deste mecanismo mencionado se concretizou em uma melhora de até 5% da função de ajuste descritora dos sistemas analisados, ou seja, os sistemas obtiveram um ajuste do tipo R^2 de aproximadamente 95%.

Um ponto importante então seria tratar de focalizar na maneira pela qual a rede identifica o melhor ajuste da matriz de coordenadas dos polos. O procedimento aleatório realizado neste trabalho se mostrou bastante eficiente até o nível estabelecido na proposta desta dissertação. Porém, existem técnicas de maior robustez para manipular estes parâmetros, tais como o algoritmo *GridSearchCV*, que realiza diversos testes para diferentes escolhas de hiperparâmetros de redes, ou mesmo o *K-means*, que pode ser utilizado para fazer um mapeamento dos vetores de entradas e realizar agrupamentos (*clusters*), sendo portanto possível indicar os melhores centros das funções de base radial a serem aplicados.

Em resumo, as redes de funções de base radial são ferramentas altamente poderosas e descritivas, e pode-se considerar que seu hiperparâmetro de maior impacto é a localização dos polos, e de maneira intrínseca a este, o valor máximo da distância euclidiana de seus elementos.

CONCLUSÕES

A temperatura de transição vítrea, T_g , é a propriedade de excelência de caracterização do estado vítreo. Neste trabalho foi possível estabelecer uma rede neural com apenas dois polos ("neurônios") preditora do comportamento da T_g de vidros binários do tipo alcalino-boratos considerando até 3 elementos químicos em cada sistema e valores da grandeza variando entre 450 e quase 800 K, validando a hipótese desta dissertação. Este resultado comprova o poder preditivo da técnica, e algo diverso de [Cassar, Carvalho e Zanotto \(2018\)](#), que aplicaram uma relativa extensa rede para prever a T_g de 55.150 composições vítreas multicomponentes a base de óxidos, considerando até 45 elementos químicos e incluindo sistemas binários simples como os analisados neste trabalho.

O comportamento da T_g dos sistemas alcalino boratos em função da concentração molar x de óxidos é similar e corresponde ao modelo topológico de [Mauro, Gupta e Loucks \(2009\)](#), estabelecido apenas para os sistemas boratos de lítio e sódio. Enquanto resultado parcial, é, portanto, possível propor a extensão do modelo topológico para os demais metais alcalinos, a saber, de potássio, rubídio e césio.

A manipulação de apenas dois hiperparâmetros das redes neurais apresentaram características importantes. Primeiramente, a função de ativação gaussiana revelou-se com maior potencial de ajuste dos sistema analisado, já que manteve um valor R^2 acima de 90%, sendo superior às funções de ativação multiquadrática e multiquadrática inversa aplicadas aos mesmos sistemas. A posição dos centros das funções de base radial foi altamente influente na resposta final, pois testes indicaram melhoras no ajuste aos dados de até 5% quando este fora manipulado.

Tais resultados podem ser estendidos para maior número de polos bem como outros sistemas vítreos, visando prever a T_g inclusive de composições ainda não criadas. Outras técnicas de inteligência artificial e aprendizado de máquina podem ser utilizadas para se obter resultados ainda melhores através de um treinamento otimizado das redes neurais por meio de algoritmos como *GridSearchCV* e *K-means* pois é esperado a sua superioridade ao método de aleatoriedade empregado aos centros das funções de base radial.

FUTUROS TRABALHOS

Dar continuidade às análises de redes neurais do tipo base radial com poucos ou mesmo mais neurônios em outros sistemas, como silicatos e germanatos, estendendo o trabalho de [Cassar, Carvalho e Zanotto \(2018\)](#), que apresentaram dados do comportamento de T_g em sódio silicatos e sódio boratos. Pretende-se também analisar estas e outras redes neuronais em outras propriedades físicas, como temperatura *líquidus* de sistemas vítreos ([DREYFUS e DREYFUS, 2003](#)) ou ainda viscosidade, basicamente em função da composição.

Em particular, outras análises de redes neurais poderão ser efetuadas, como por exemplo considerando a quantidade estimada de estruturas Q_i para cada composição em diversos sistemas.

REFERÊNCIAS BIBLIOGRÁFICAS

AGUIAR, R.; FIGUEIREDO, E. B.; HAVEROTH, G. A. Considerações sobre as derivadas de Gâteaux e Fréchet. **C.Q.D. – Revista Eletrônica Paulista de Matemática**, Bauru, v. 9, p. 32-48, jul. 2017. Edição Iniciação Científica.

AGRAWAL, Ankit; CHOUDHARY, Alok. Deep materials informatics: Applications of deep learning in materials science. **MRS Communications**, v. 9, n. 3, p. 779-792, 2019.

AFANTITIS, Antreas et al. Prediction of high weight polymers glass transition temperature using RBF neural networks. **Journal of Molecular Structure: THEOCHEM**, v. 716, n. 1-3, p. 193-198, 2005.

ALVES, Oswaldo Luiz; DE FÁTIMA GIMENEZ, Iara; MAZALI, Italo Odone. A arte de fazer vidro. **Química Nova na Escola Edição Esp**, p. 13-24, 2001.

ANSCOMBE, Francis J.; TUKEY, John W. **The examination and analysis of residuals**. Technometrics, v. 5, n. 2, p. 141-160, 1963.

ARFKEN, George B.; WEBER, Hans J. **Mathematical methods for physicists**. 1999.

AVRAMOV, I.; VASSILEV, Ts; PENKOV, I. The glass transition temperature of silicate and borate glasses. **Journal of non-crystalline solids**, v. 351, n. 6-7, p. 472-476, 2005.

BESSA, Graziela Pentean. Previsão de viscosidade e temperatura liquidus de vidros óxidos via redes neurais artificiais. 2020.

BISHOP, Chris. Improving the generalization properties of radial basis function neural networks. **Neural computation**, v. 3, n. 4, p. 579-588, 1991.

BOLT, Marvin. Glass: The eye of science. **International Journal of Applied Glass Science**, v. 8, n. 1, p. 4-22, 2017.

BOTROS, Sherif M.; ATKESON, Christopher G. Generalization Properties of Radial Basis Functions. In: **NIPS**. 1990. p. 707-713.

BROOMHEAD, David S.; LOWE, David. **Radial basis functions, multi-variable functional interpolation and adaptive networks**. Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.

BUTKOV, Eugene. **Física matemática**. Livros Técnicos e Científicos, 1988.

CASSAR, Daniel R.; DE CARVALHO, Andre CPLF; ZANOTTO, Edgar D. Predicting glass transition temperatures using neural networks. **Acta Materialia**, v. 159, p. 249-256, 2018.

CHEN, Xi; SZTANDERA, Les; CARTWRIGHT, Hugh M. A neural network approach to prediction of glass transition temperature of polymers. **International Journal of Intelligent Systems**, v. 23, n. 1, p. 22-32, 2008.

CHOLLET, Francois. **Deep learning with Python**. Simon and Schuster, 2021.

CLEOPHAS, Ton J.; ZWINDERMAN, Aeilko H. **Bayesian Pearson Correlation Analysis**. In: Modern Bayesian Statistics in Clinical Research. Springer, Cham, 2018. p. 111-118.

COSTA, Herbert Rodrigues do Nascimento. **Aplicação de técnicas de inteligência artificial em processos de fabricação de vidro**. 2006. Tese de Doutorado. Universidade de São Paulo.

COOK, R. Dennis; WEISBERG, Sanford. **Residuals and influence in regression**. New York: Chapman and Hall, 1982.

COVER, Thomas M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. **IEEE Transactions on Electronic Computers**, n. 3, p. 326-334, 1965.

DECEANNE, A. et al. An anomaly in the glass transition width trends of alkali borate glasses at low modifier loadings. **Physics and Chemistry of Glasses-European Journal of Glass Science and Technology Part B**, v. 58, n. 5, p. 187-194, 2017.

DREYFUS, Catherine, DREYFUS, Gérard. A machine learning approach to the estimation of the liquidus temperature of glass-forming oxide blends. **Journal of Non-Crystalline Solids** v. 318, p. 63-78, 2003.

DUDLEY, Donald G. **Mathematical Foundations for Electromagnetic Theory**. New York: IEEE press, 1994.

DU, Ke-Lin; SWAMY, Madisetti NS. **Neural Networks in a Softcomputing framework**. Springer Science & Business Media, 2006.

DUFFY, Dean G. **Green's functions with applications**. Chapman and Hall/CRC, 2015.

GASSER, Theo; SROKA, Lothar; JENNEN-STEINMETZ, Christine. **Residual variance and residual pattern in nonlinear regression**. Biometrika, v. 73, n. 3, p. 625-633, 1986.

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

GIROSI, F.; JONES, M.; POGGIO, T. , Regularization theory and neural networks architectures: **Neural Computation** 7, 219-269. 1995.

GUPTA, P. K. **Proceedings of the XIV International Congress on Glass**, New Delhi, India, p. 1, 1986.

GUPTA, Prabhat K. Non-crystalline solids: glasses and amorphous solids. **Journal of Non-Crystalline Solids**, v. 195, n. 1-2, p. 158-164, 1996.

HAYKIN, Simon; NETWORK, N. A comprehensive foundation. **Neural Networks**, v. 2, n. 2004, p. 41, 2004.

HAYKIN, Simon. **Redes Neurais: Princípios e Prática**. Bookman Editora, 2007.

HAYKIN, Simon. **Neural Networks and Learning Machines**, 3/E. Pearson Education India, 2010.

HOWLETT, Robert J.; JAIN, Lakhmi C. **Radial Basis Function Networks 2: New Advances in Design**. Physica, 2013.

JACKSON, J. David. Electrodynamics. **The Optics Encyclopedia**, 1975.

JAYAWEERA, C. D.; AZIZ, N. **Reliability of principal component analysis and pearson correlation coefficient, for application in artificial neural network model development, for water treatment plants**. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, 2018. p. 012076.

JIANG, Zhuoying et al. A deep neural network for accurate and robust prediction of the glass transition temperature of polyhydroxyalkanoate homo-and copolymers. **Materials**, v. 13, n. 24, p. 5701, 2020.

JUPYTER, 2015. Disponível em: <<https://jupyter.org/>>. Acesso em: 10/10/2021

KELLEHER, John D.; MAC NAMEE, Brian; D'ARCY, Aoife. **Fundamentals of machine learning for predictive data analytics: algorithms**, worked examples, and case studies. MIT press, 2020.

KERNER, Richard; MICOULAUT, Matthieu. On the glass transition temperature in covalent glasses. **Journal of Non-Crystalline Solids**, v. 210, n. 2-3, p. 298-305, 1997.

KOJIMA, Seiji. Mixed-Alkali Effect in Borate Glasses: Thermal, Elastic, and Vibrational Properties. **Solids**, v. 1, n. 1, p. 16-30, 2020.

KRIESEL, David. A brief introduction on neural networks. 2007.

LIU, Han et al. Machine learning for glass science and engineering: A review. **Journal of Non-Crystalline Solids**, v. 557, p. 119419, 2021.

MACHADO, Kleber Daum. **Equações Diferenciais Aplicadas à Física**. Editora UEPG, 2004.

MARLIN, Benjamin. **Missing data problems in machine learning**. 2008. Tese de Doutorado.

MAURO, John C. Decoding the glass genome. **Current Opinion in Solid State and Materials Science**, v. 22, n. 2, p. 58-64, 2018.

MAURO, John C. Grand challenges in glass science. **Frontiers in Materials**, v. 1, p. 20, 2014.

MAURO, John C; GUPTA, Prabhat K. Composition dependence of glass transition temperature and fragility. I. A topological model incorporating temperature-dependent constraints. **The Journal of Chemical Physics**, v. 130, n. 9, p. 094503, 2009.

MAURO, John C; GUPTA, Prabhat K., LOUCKS, Roger J. Composition dependence of glass transition temperature and fragility. II. A topological model of alkali borate liquids. **The Journal of Chemical Physics**, v. 130, n. 9, p. 234503, 2009.

MAURO, John C.; ZANOTTO, Edgar D. Two centuries of glass research: historical trends, current status, and grand challenges for the future. **International Journal of Applied Glass Science**, v. 5, n. 3, p. 313-327, 2014.

MCCULLOCH, Warren S.; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115-133, 1943.

MELAGRAKI, Georgia et al. A novel RBF neural network training methodology to predict toxicity to *Vibrio fischeri*. **Molecular Diversity**, v. 10, n. 2, p. 213-221, 2006.

MHASKAR, Hrushikesh N. Neural networks for optimal approximation of smooth and analytic functions. **Neural computation**, v. 8, n. 1, p. 164-177, 1996.

MICCHELLI, Charles A. Algebraic aspects of interpolation. In: **Proceedings of Symposia in Applied Mathematics**. 1986. p. 81-102.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre Aprendizado de Máquina. **Sistemas Inteligentes - Fundamentos e Aplicações**, v. 1, n. 1, p. 32, 2003.

MÜLLER, Andreas C.; GUIDO, Sarah. **Introduction to machine learning with Python: a guide for data scientists**. " O'Reilly Media, Inc.", 2016.

NASCIMENTO, M. L. F. **Problemas Correntes sobre Nucleação, Crescimento de Cristais e Difusão em Vidros**. Tese Universidade Federal de São Carlos, 377 p., 2004.

NASCIMENTO, M. L. F. Through Glass, the Rise of Modern Science. **Journal of Materials Education** v. 37, p. 137-154, 2015.

NASCIMENTO, Harrison H. S, NASCIMENTO, Marcio L. F. Identifying silica types using viscosity data and principal component analysis. **Journal of Physics and Chemistry of Solids** v. 157, 110177, 2021.

NIYOGI, Partha; GIROSI, Federico. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. **Neural Computation**, v. 8, n. 4, p. 819-842, 1996.

O'CONNOR, J. J.; ROBERTSON E. F. Rene Maurice Fréchet'. MacTutor History of Mathematics Archive. 2005. Disponível em: <<http://www-history.mcs.st-andrews.ac.uk/Biographies/>>

ORR, Mark JL et al. Introduction to radial basis function networks. 1996.

PARK, Jooyoung; SANDBERG, Irwin W. Approximation and radial-basis-function networks. **Neural Computation**, v. 5, n. 2, p. 305-316, 1993.

PILANIA, Ghanshyam et al. Accelerating materials property predictions using machine learning. **Scientific reports**, v. 3, n. 1, p. 1-6, 2013.

POGGIO, Tomaso; GIROSI, Federico. Networks for approximation and learning. **Proceedings of the IEEE**, v. 78, n. 9, p. 1481-1497, 1990.

Powell, M.J.D. 1987. Radial basis functions for multivariable interpolation: a review. In **Algorithms for Approximation**, J.C.Mason & M.G.Cox (eds). Oxford: Clarendon Press.

POWELL, M. J. D. Radial basis functions approximations to polynomials. In: **Proc. 12th Biennial Numerical Analysis Conf.**, 1987. 1987.

RAUBER, Thomas Walter. Redes neurais artificiais. **Universidade Federal do Espírito Santo**, v. 29, 2005.

SANTOS, Alexandre Jones dos. **Avaliação de modelos de aprendizado de máquina para classificação de tumores de câncer de mama**. Universidade Federal do Amapá. 2020.

SRIVASTAVA, Nitish et al. Dropout: a simple way to prevent neural networks from overfitting. **The Journal of Machine Learning Research**, v. 15, n. 1, p. 1929-1958, 2014.

STAKGOLD, Ivar. **Boundary Value Problems of Mathematical Physics: Volume 1**. Society for Industrial and Applied Mathematics, 2000.

STRAUME, Martin; JOHNSON, Michael L. [5] **Analysis of Residuals: Criteria for determining goodness-of-fit**. Methods in enzymology, v. 210, p. 87-105, 1992.

TIKHONOV, Andrei Nikolaevich. On the solution of ill-posed problems and the method of regularization. In: **Doklady Akademii Nauk**. Russian Academy of Sciences, 1963. p. 501-504.

TURING, Alan M. Computing machinery and intelligence. **Mind** v. 59, p. 433–460, 1950.

UHLMANN, Donald R.; KOLBECK, A. G.; DE WITTE, D. L. Heat capacities and thermal behavior of alkali borate glasses. **Journal of Non-Crystalline Solids**, v. 5, n. 5, p. 426-443, 1971.

VARSHNEYA, Arun K. **Fundamentals of Inorganic Glasses**. Elsevier, 2013.

WATT, Jeremy; BORHANI, Reza; KATSAGGELOS, Aggelos K. **Machine learning refined: Foundations, algorithms, and applications**. Cambridge University Press, 2020.

WERBOS, Paul J. Backpropagation through time: what it does and how to do it. **Proceedings of the IEEE**, v. 78, n. 10, p. 1550-1560, 1990.

ZANOTTO, E. D.; COUTINHO, F. A. B. How many non-crystalline solids can be made from all the elements of the periodic table?. **Journal of Non-Crystalline Solids**, v. 347, n. 1-3, p. 285-288, 2004.

ZANOTTO, Edgar D.; MAURO, John C. The glassy state of matter: Its definition and ultimate fate. **Journal of Non-Crystalline Solids**, v. 471, p. 490-495, 2017.

ZHANG, Yun; XU, Xiaojie. Machine learning glass transition temperature of polymers. **Heliyon**, v. 6, n. 10, p. e05055, 2020.

ZHONG, Jianhui et al. Structural modeling of lithium borosilicate glasses via NMR studies. **Journal of non-crystalline solids**, v. 107, n. 1, p. 81-87, 1988.

ANEXO 1: Código Completo em Python

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:

from funcoes import fbr
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import seaborn as sns

# In[2]:

dados_potassio = pd.read_excel("DADOS
LEONARDO/BoroPotássio.xlsx").dropna() [['Potassio (mol%) ', 'Boro (mol%) ', 'T
g (K) ']]
dados_litio = pd.read_excel("DADOS
LEONARDO/BoroLítio.xlsx").dropna() [['Litio (mol%) ', 'Boro (mol%) ', 'Tg (K) '
]
dados_rubidio = pd.read_excel("DADOS
LEONARDO/BoroRubídio.xlsx").dropna() [['Rubidio (mol%) ', 'Boro (mol%) ', 'Tg (
K) ']].iloc[0:52,:]
dados_sodio = pd.read_excel("DADOS
LEONARDO/BoroSódio.xlsx").dropna() [['Sodio (mol%) ', 'Boro (mol%) ', 'Tg (K) '
]
dados_cesio = pd.read_excel("DADOS
LEONARDO/Césio.xlsx").dropna() [['Cesio (mol%) ', 'Boro (mol%) ', 'Tg (K) ']]

# In[3]:

dados_sodio

# In[4]:

num_da_funcao = int(input("Digitar o 'Número da Função' = \n\n"
    "1 - M: Multiquadrática;\n"
    "2 - MR: Multiquadrática Recíproca;\n"
    "3 - MRI: Multiquadrática Recíproca Inversa;\n"))
```

```

        "4 - G: Gaussiana;\n"
        "5 - SH: Secante Hiperbólica;\n"
        "6 - CH: Cosseno Hiperbólico;\n"
        "7 - SPF: Splines de Placas Finas.\n\n"
        "Resposta = ")

print("")
print("Função de Base Radial = " +
      fbr.fbr_descricao(num_da_funcao))

# ### 1 - SINAPSES POTÁSSIO

# In[5]:

treino_potassio, teste_potassio = train_test_split(dados_potassio,
test_size = 0.3, random_state = 42)
num_de_licoes_potassio = len(treino_potassio)
num_de_variaveis_potassio = len(treino_potassio.columns) - 1
variaveis_potassio = treino_potassio[['Potassio (mol%)', 'Boro (mol%)']]
A_potassio = treino_potassio['Tg (K)']
y_exp_potassio = A_potassio.copy()
media_da_functus_potassio = np.mean(y_exp_potassio)

# ### 1.1 - POLOS

# In[6]:

# Coordenada dos Polos:
# num_de_polos = 9Número de polos
np.random.seed(80)
if num_de_variaveis_potassio == 1:
    num_de_polos_potassio = num_de_variaveis_potassio + 1
elif num_de_variaveis_potassio != 1:
    num_de_polos_potassio = num_de_variaveis_potassio

# C: matrix com os vectors de Coordenada dos Polos
C_potassio = np.zeros((num_de_polos_potassio,
num_de_variaveis_potassio),
dtype = float)

# lim_inf: limite inferior
lim_inf_potassio = np.min(np.array(variaveis_potassio))
print("Limite inferior = " + str(round(lim_inf_potassio, 2)))

```

```

# lim_sup: limite superior
lim_sup_potassio = np.max(np.array(variaveis_potassio))
print("Limite superior = " + str(round(lim_sup_potassio, 2)))

# A MATRIZ COORDENADA:
C_potassio = np.random.rand(num_de_polos_potassio,
num_de_variaveis_potassio) * (lim_sup_potassio - lim_inf_potassio) +
lim_inf_potassio

pd.DataFrame(C_potassio)

# In[7]:

"""
É conveniente ter pelo menos um polo na origem 0, pois
isso faz com que o modelo retorne uma Coeficiente de
Acerto (R ** 2) possivelmente maior. Logo:
"""
C_potassio[0] = 0
pd.DataFrame(C_potassio)

# In[8]:

# dist_entre_os_polos: distância entre os polos
dist_entre_os_polos_potassio = np.zeros((num_de_polos_potassio,
num_de_polos_potassio),
dtype = float)
for i in range(0, num_de_polos_potassio, 1):
    for j in range(0, num_de_polos_potassio, 1):
        dist_entre_os_polos_potassio[i, j] =
np.linalg.norm(C_potassio[i] - C_potassio[j])

pd.DataFrame(dist_entre_os_polos_potassio)

# ### 1.2 - RBF VARIÁVEIS

# In[9]:

dps_max_potassio = np.max(dist_entre_os_polos_potassio) #DISTÂNCIA
MÁXIMA
sigma_potassio = dps_max_potassio / np.sqrt(2 * num_de_polos_potassio) #
DESVIO PADRÃO

```

```

# In[10]:

variaveis_potassio = np.array(variaveis_potassio)
# Matrix [R]:
# R: Matrix de Base Radial [R]
R_potassio = np.zeros((num_de_licoes_potassio, num_de_polos_potassio),
                      dtype = float)
for n in range(0, num_de_licoes_potassio, 1):
    for i in range(0, num_de_polos_potassio, 1):
        R_potassio[n, i] = fbr.fbr(num_da_funcao,
                                   variaveis_potassio[n],
                                   C_potassio[i],
                                   sigma_potassio)
pd.DataFrame(R_potassio)

```

```

# In[11]:

# É necessário acrescentar na matrix [R] um coluna com um's, logo:
# vector_unitario: vector de uns
vector_unitario = np.ones((num_de_licoes_potassio, 1),
                           dtype = float)
R_potassio = np.hstack((R_potassio, vector_unitario))
pd.DataFrame(R_potassio)

```

```

# ### 1.3 - OPERAÇÕES MATRICIAIS

```

```

# In[12]:

Rt_potassio = np.transpose(R_potassio)
RtR_potassio = np.matmul(Rt_potassio, R_potassio)
RtRinv_potassio = np.linalg.inv(RtR_potassio)
mpi_potassio = np.matmul(RtRinv_potassio, Rt_potassio)
VS_potassio = np.matmul(mpi_potassio, A_potassio)

```

```

# ### 1.4 - RESULTADOS (TREINO)

```

```

# In[13]:

#RESULTADOS TREINO
y_teo_potassio_treino = np.zeros(len(variaveis_potassio),
                                  dtype = float)
for i in range(0, len(variaveis_potassio), 1):

```

```

    y_teo_potassio_treino[i] = sum(VS_potassio * R_potassio[i])
pd.DataFrame(y_teo_potassio_treino, columns = ['Previsão[K]'])

# In[14]:

from sklearn.metrics import r2_score
r2_score(y_teo_potassio_treino,A_potassio)

# ### 1.5 - RESULTADOS (TESTE)

# In[15]:

variaveis_teste_potassio =
np.array(teste_potassio[['Potassio (mol%)', 'Boro (mol%)']])
R_teste_potassio = np.zeros((num_de_licoes_potassio,
num_de_polos_potassio),
dtype = float)
R_teste_potassio = np.hstack((R_teste_potassio, vector_unitario))
for n in range(0, len(variaveis_teste_potassio), 1):
    for i in range(0, num_de_polos_potassio, 1):
        R_teste_potassio[n, i] = fbr.fbr(num_da_funcao,
variaveis_teste_potassio[n],
C_potassio[i],
sigma_potassio)
pd.DataFrame(R_teste_potassio)

# In[16]:

#Resultados
y_teo_teste_potassio = np.zeros(len(variaveis_teste_potassio),
dtype = float)
for i in range(0, len(variaveis_teste_potassio), 1):
    y_teo_teste_potassio[i] = sum(VS_potassio * R_teste_potassio[i])
pd.DataFrame(y_teo_teste_potassio, columns = ['Previsão[K]'])

# In[17]:

r2_score(y_teo_teste_potassio, teste_potassio['Tg(K)'])

# In[18]:

```

```

pd.DataFrame(VS_potassio, index = ['c1', 'c2', 'Bias'], columns =
['Potássio'])

# ### 2 - SINAPSES SÓDIO

# In[19]:

treino_sodio, teste_sodio = train_test_split(dados_sodio, test_size =
0.3, random_state = 42)
num_de_licoes_sodio = len(treino_sodio)
num_de_variaveis_sodio = len(treino_sodio.columns) - 1
variaveis_sodio = treino_sodio[['Sodio(mol%)', 'Boro(mol%)']]
A_sodio = treino_sodio['Tg(K)']
y_exp_sodio = A_sodio.copy()
media_da_functus_sodio = np.mean(y_exp_sodio)

# ### 2.1 - POLOS

# In[20]:

# Coordenada dos Polos:
# num_de_polos = Número de polos
np.random.seed(80)
if num_de_variaveis_sodio == 1:
    num_de_polos_sodio = num_de_variaveis_sodio + 1
elif num_de_variaveis_sodio != 1:
    num_de_polos_sodio = num_de_variaveis_sodio

# C: matrix com os vectors de Coordenada dos Polos
C_sodio = np.zeros((num_de_polos_sodio, num_de_variaveis_sodio),
dtype = float)

# lim_inf: limite inferior
lim_inf_sodio = np.min(np.array(variaveis_sodio))
print("Limite inferior = " + str(round(lim_inf_sodio, 2)))

# lim_sup: limite superior
lim_sup_sodio = np.max(np.array(variaveis_sodio))
print("Limite superior = " + str(round(lim_sup_sodio, 2)))

# A MATRIZ COORDENADA:
C_sodio = np.random.rand(num_de_polos_sodio, num_de_variaveis_sodio) *
(lim_sup_sodio - lim_inf_sodio) + lim_inf_sodio

```

```

"""
É conveniente ter pelo menos um polo na origem 0, pois
isso faz com que o modelo retorne uma Coeficiente de
Acerto (R ** 2) possivelmente maior. Logo:
"""
C_sodio[0] = 0
pd.DataFrame(C_sodio)

# In[21]:

# dist_entre_os_polos: distância entre os polos
dist_entre_os_polos_sodio = np.zeros((num_de_polos_sodio,
num_de_polos_sodio),
dtype = float)
for i in range(0, num_de_polos_sodio, 1):
    for j in range(0, num_de_polos_sodio, 1):
        dist_entre_os_polos_sodio[i, j] = np.linalg.norm(C_sodio[i]
- C_sodio[j])

pd.DataFrame(dist_entre_os_polos_sodio)

# In[22]:

sigma_potassio

# ### 2.2 RBF VARIÁVEIS

# In[23]:

dps_max_sodio = np.max(dist_entre_os_polos_sodio) #DISTÂNCIA MÁXIMA
sigma_sodio = dps_max_sodio / np.sqrt(2 * num_de_polos_sodio) # DESVIO
PADRÃO

variaveis_sodio = np.array(variaveis_sodio)
# Matrix [R]:
# R: Matrix de Base Radial [R]
R_sodio = np.zeros((num_de_licoes_sodio, num_de_polos_sodio),
dtype = float)
for n in range(0, num_de_licoes_sodio, 1):
    for i in range(0, num_de_polos_sodio, 1):
        R_sodio[n, i] = fbr.fbr(num_da_funcao,
variaveis_sodio[n],
C_sodio[i],

```

```

sigma_sodio)

# É necessário acrescentar na matrix [R] um coluna com um's, logo:
# vector_unitario: vector de uns

vector_unitario = np.ones((num_de_licoes_sodio, 1),
                           dtype = float)
R_sodio = np.hstack((R_sodio, vector_unitario))
pd.DataFrame(R_sodio)

# ### 2.3 OPERAÇÕES MATRICIAIS

# In[24]:

Rt_sodio = np.transpose(R_sodio)
RtR_sodio = np.matmul(Rt_sodio, R_sodio)
RtRinv_sodio = np.linalg.inv(RtR_sodio)
mpi_sodio = np.matmul(RtRinv_sodio, Rt_sodio)
VS_sodio = np.matmul(mpi_sodio, A_sodio)

# ### 2.4 - RESULTADOS TREINO

# In[25]:

#RESULTADOS TREINO
y_teo_sodio_treino = np.zeros(len(variaveis_sodio),
                              dtype = float)
for i in range(0, len(variaveis_sodio), 1):
    y_teo_sodio_treino[i] = sum(VS_sodio * R_sodio[i])
pd.DataFrame(y_teo_sodio_treino, columns = ['Previsão[K]'])

print(r2_score(y_teo_sodio_treino, treino_sodio['Tg(K)'])
pd.DataFrame(VS_sodio, index = ['C1', 'C2', 'Bias'], columns = ['Sódio'])

# ### 2.5 - RESULTADOS TESTE

# In[26]:

variaveis_teste_sodio =
np.array(teste_sodio[['Sodio(mol%)', 'Boro(mol%)']])
R_teste_sodio = np.zeros((num_de_licoes_sodio, num_de_polos_sodio),
                          dtype = float)
R_teste_sodio = np.hstack((R_teste_sodio, vector_unitario))

```

```

for n in range(0, len(variaveis_teste_sodio), 1):
    for i in range(0, num_de_polos_sodio, 1):
        R_teste_sodio[n, i] = fbr.fbr(num_da_funcao,
                                     variaveis_teste_sodio[n],
                                     C_sodio[i],
                                     sigma_sodio)
pd.DataFrame(R_teste_sodio)

# In[ ]:

# In[27]:

#Resultados
y_teo_teste_sodio = np.zeros(len(variaveis_teste_sodio),
                              dtype = float)
for i in range(0, len(variaveis_teste_sodio), 1):
    y_teo_teste_sodio[i] = sum(VS_sodio * R_teste_sodio[i])
pd.DataFrame(y_teo_teste_sodio, columns = ['Previsão[K]'])
r2_score(y_teo_teste_sodio, teste_sodio['Tg(K)'])

# In[28]:

VS_sodio

# ### 3.0 SINAPSES LÍTIO

# In[29]:

treino_litio, teste_litio = train_test_split(dados_litio, test_size =
0.3, random_state = 42)
num_de_licoes_litio = len(treino_litio)
num_de_variaveis_litio = len(treino_litio.columns) - 1
variaveis_litio = treino_litio[['Litio(mol%)', 'Boro(mol%)']]
A_litio = treino_litio['Tg(K)']
y_exp_litio = A_litio.copy()
media_da_functus_litio = np.mean(y_exp_litio)

# In[30]:

dados_litio

# In[ ]:

```

```

# ### 3.1 POLOS

# In[31]:

# Coordenada dos Polos:
# num_de_polos = Número de polos
np.random.seed(80)
if num_de_variaveis_litio == 1:
    num_de_polos_litio = num_de_variaveis_litio + 1
elif num_de_variaveis_litio != 1:
    num_de_polos_litio = num_de_variaveis_litio

# C: matrix com os vectors de Coordenada dos Polos
C_litio = np.zeros((num_de_polos_litio, num_de_variaveis_litio),
                  dtype = float)

# lim_inf: limite inferior
lim_inf_litio = np.min(np.array(variaveis_litio))
print("Limite inferior = " + str(round(lim_inf_litio, 2)))

# lim_sup: limite superior
lim_sup_litio = np.max(np.array(variaveis_litio))
print("Limite superior = " + str(round(lim_sup_litio, 2)))

# A MATRIZ COORDENADA:
C_litio = np.random.rand(num_de_polos_litio, num_de_variaveis_litio) *
(lim_sup_litio - lim_inf_litio) + lim_inf_litio
"""
É conveniente ter pelo menos um polo na origem 0, pois
isso faz com que o modelo retorne uma Coeficiente de
Acerto (R ** 2) possivelmente maior. Logo:
"""
C_litio[0] = 0
pd.DataFrame(C_litio)

# In[32]:

# dist_entre_os_polos: distância entre os polos
dist_entre_os_polos_litio = np.zeros((num_de_polos_litio,
num_de_polos_litio),
                                     dtype = float)
for i in range(0, num_de_polos_litio, 1):
    for j in range(0, num_de_polos_litio, 1):

```

```

        dist_entre_os_polos_litio[i, j] = np.linalg.norm(C_litio[i]
- C_litio[j])

pd.DataFrame(dist_entre_os_polos_litio)

# ### 3.2 RBF VARIÁVEIS

# In[33]:

dps_max_litio = np.max(dist_entre_os_polos_litio) #DISTÂNCIA MÁXIMA
sigma_litio = dps_max_litio / np.sqrt(2 * num_de_polos_litio) # DESVIO
PADRÃO

variaveis_litio = np.array(variaveis_litio)
# Matrix [R]:
# R: Matrix de Base Radial [R]
R_litio = np.zeros((num_de_licoes_litio, num_de_polos_litio),
                    dtype = float)
for n in range(0, num_de_licoes_litio, 1):
    for i in range(0, num_de_polos_litio, 1):
        R_litio[n, i] = fbr.fbr(num_da_funcao,
                                variaveis_litio[n],
                                C_litio[i],
                                sigma_litio)

# É necessário acrescentar na matrix [R] um coluna com um's, logo:
# vector_unitario: vector de uns

vector_unitario = np.ones((num_de_licoes_litio, 1),
                           dtype = float)
R_litio = np.hstack((R_litio, vector_unitario))
pd.DataFrame(R_litio)

# ### 3.3 OPERAÇÕES MATRICIAIS

# In[34]:

Rt_litio = np.transpose(R_litio)
RtR_litio = np.matmul(Rt_litio, R_litio)
RtRinv_litio = np.linalg.inv(RtR_litio)
mpi_litio = np.matmul(RtRinv_litio, Rt_litio)
VS_litio = np.matmul(mpi_litio, A_litio)

```

```
# ### 3.4 RESULTADOS TREINO
```

```
# In[35]:
```

```
y_teo_litio_treino = np.zeros(len(variaveis_litio),
                               dtype = float)
for i in range(0, len(variaveis_litio), 1):
    y_teo_litio_treino[i] = sum(VS_litio * R_litio[i])
pd.DataFrame(y_teo_litio_treino, columns = ['Previsão[K]'])

print(r2_score(y_teo_litio_treino, treino_litio['Tg(K)'])
pd.DataFrame(VS_litio, index = ['C1', 'C2', 'Bias'], columns = ['Lítio'])
```

```
# ### 3.5 RESULTADOS TESTE
```

```
# In[36]:
```

```
variaveis_teste_litio = np.array(teste_litio[['Lítio(mol%)', 'Boro(mol%)']])
R_teste_litio = np.zeros((num_de_licoes_litio, num_de_polos_litio),
                          dtype = float)
R_teste_litio = np.hstack((R_teste_litio, vector_unitario))
for n in range(0, len(variaveis_teste_litio), 1):
    for i in range(0, num_de_polos_litio, 1):
        R_teste_litio[n, i] = fbr.fbr(num_da_funcao,
                                       variaveis_teste_litio[n],
                                       C_litio[i],
                                       sigma_litio)
pd.DataFrame(R_teste_litio)
```

```
# In[ ]:
```

```
# In[37]:
```

```
#Resultados
```

```
y_teo_teste_litio = np.zeros(len(variaveis_teste_litio),
                               dtype = float)
for i in range(0, len(variaveis_teste_litio), 1):
    y_teo_teste_litio[i] = sum(VS_litio * R_teste_litio[i])
pd.DataFrame(y_teo_teste_litio, columns = ['Previsão[K]'])
r2_score(y_teo_teste_litio, teste_litio['Tg(K)'])
```

```

### 4.0 SINAPSES RUBÍDIO

# In[38]:

treino_rubidio, teste_rubidio = train_test_split(dados_rubidio,
test_size = 0.3, random_state = 42)
num_de_licoes_rubidio = len(treino_rubidio)
num_de_variaveis_rubidio = len(treino_rubidio.columns) - 1
variaveis_rubidio = treino_rubidio[['Rubidio(mol%)', 'Boro(mol%)']]
A_rubidio = treino_rubidio['Tg(K)']
y_exp_rubidio = A_rubidio.copy()
media_da_functus_rubidio = np.mean(y_exp_rubidio)

### 4.1 POLOS

# In[39]:

# Coordenada dos Polos:
# num_de_polos = Número de polos
np.random.seed(80)
if num_de_variaveis_rubidio == 1:
    num_de_polos_rubidio = num_de_variaveis_rubidio + 1
elif num_de_variaveis_rubidio != 1:
    num_de_polos_rubidio = num_de_variaveis_rubidio

# C: matrix com os vectors de Coordenada dos Polos
C_rubidio = np.zeros((num_de_polos_rubidio, num_de_variaveis_rubidio),
dtype = float)

# lim_inf: limite inferior
lim_inf_rubidio = np.min(np.array(variaveis_rubidio))
print("Limite inferior = " + str(round(lim_inf_rubidio, 2)))

# lim_sup: limite superior
lim_sup_rubidio = np.max(np.array(variaveis_rubidio))
print("Limite superior = " + str(round(lim_sup_rubidio, 2)))

# A MATRIZ COORDENADA:
C_rubidio = np.random.rand(num_de_polos_rubidio,
num_de_variaveis_rubidio) * (lim_sup_rubidio - lim_inf_rubidio) +
lim_inf_rubidio
"""
É conveniente ter pelo menos um polo na origem 0, pois

```

isso faz com que o modelo retorne uma Coeficiente de Acerto ($R ** 2$) possivelmente maior. Logo:

```

"""
C_rubidio[0] = 0
pd.DataFrame(C_rubidio)

# In[40]:

# dist_entre_os_polos: distância entre os polos
dist_entre_os_polos_rubidio      =      np.zeros((num_de_polos_rubidio,
num_de_polos_rubidio),
                                         dtype = float)

for i in range(0, num_de_polos_rubidio, 1):
    for j in range(0, num_de_polos_rubidio, 1):
        dist_entre_os_polos_rubidio[i,          j]          =
np.linalg.norm(C_rubidio[i] - C_rubidio[j])

pd.DataFrame(dist_entre_os_polos_rubidio)

# ### 4.2 RBF VARIÁVEIS

# In[41]:

dps_max_rubidio = np.max(dist_entre_os_polos_rubidio) #DISTÂNCIA MÁXIMA
sigma_rubidio = dps_max_rubidio / np.sqrt(2 * num_de_polos_rubidio) #
DESVIO PADRÃO

variaveis_rubidio = np.array(variaveis_rubidio)
# Matrix [R]:
# R: Matrix de Base Radial [R]
R_rubidio = np.zeros((num_de_licoes_rubidio, num_de_polos_rubidio),
                     dtype = float)

for n in range(0, num_de_licoes_rubidio, 1):
    for i in range(0, num_de_polos_rubidio, 1):
        R_rubidio[n, i] = fbr.fbr(num_da_funcao,
                                variaveis_rubidio[n],
                                C_rubidio[i],
                                sigma_rubidio)

# É necessário acrescentar na matrix [R] um coluna com um's, logo:
# vector_unitario: vector de uns

vector_unitario = np.ones((num_de_licoes_rubidio, 1),

```

```

dtype = float)
R_rubidio = np.hstack((R_rubidio, vector_unitario))
pd.DataFrame(R_rubidio)

# ### 4.3 OPERAÇÕES MATRICIAIS

# In[42]:

Rt_rubidio = np.transpose(R_rubidio)
RtR_rubidio = np.matmul(Rt_rubidio, R_rubidio)
RtRinv_rubidio = np.linalg.inv(RtR_rubidio)
mpi_rubidio = np.matmul(RtRinv_rubidio, Rt_rubidio)
VS_rubidio = np.matmul(mpi_rubidio, A_rubidio)

# ### 4.4 RESULTADOS TREINO

# In[43]:

y_teo_rubidio_treino = np.zeros(len(variaveis_rubidio),
                                dtype = float)
for i in range(0, len(variaveis_rubidio), 1):
    y_teo_rubidio_treino[i] = sum(VS_rubidio * R_rubidio[i])
pd.DataFrame(y_teo_rubidio_treino, columns = ['Previsão[K]'])

print(r2_score(y_teo_rubidio_treino, treino_rubidio['Tg(K)'])
pd.DataFrame(VS_rubidio, index = ['C1', 'C2', 'Bias'], columns =
['Rubidio'])

# ### 4.5 RESULTADOS TESTE

# In[44]:

variaveis_teste_rubidio =
np.array(teste_rubidio[['Rubidio(mol%)', 'Boro(mol%)']])
R_teste_rubidio = np.zeros((num_de_licoes_rubidio,
num_de_polos_rubidio),
dtype = float)
R_teste_rubidio = np.hstack((R_teste_rubidio, vector_unitario))
for n in range(0, len(variaveis_teste_rubidio), 1):
    for i in range(0, num_de_polos_rubidio, 1):
        R_teste_rubidio[n, i] = fbr.fbr(num_da_funcao,
variaveis_teste_rubidio[n],
C_rubidio[i],

```

```

                                sigma_rubidio)
pd.DataFrame(R_teste_rubidio)

# In[ ]:

# In[45]:

#Resultados
y_teo_teste_rubidio = np.zeros(len(variaveis_teste_rubidio),
                                dtype = float)
for i in range(0, len(variaveis_teste_rubidio), 1):
    y_teo_teste_rubidio[i] = sum(VS_rubidio * R_teste_rubidio[i])
pd.DataFrame(y_teo_teste_rubidio, columns = ['Previsão[K]'])
r2_score(y_teo_teste_rubidio, teste_rubidio['Tg(K)'])

# ### 5 - SINAPSES CÉSIO

# In[46]:

treino_cesio, teste_cesio = train_test_split(dados_cesio, test_size =
0.2, random_state = 20)
num_de_licoes_cesio = len(treino_cesio)
num_de_variaveis_cesio = len(treino_cesio.columns) - 1
variaveis_cesio = treino_cesio[['Cesio(mol%)', 'Boro(mol%)']]
A_cesio = treino_cesio['Tg(K)']
y_exp_cesio = A_cesio.copy()
media_da_functus_cesio = np.mean(y_exp_cesio)

# ### 5.1 - POLOS

# In[47]:

# Coordenada dos Polos:
# num_de_polos = Número de polos
np.random.seed(80)
if num_de_variaveis_cesio == 1:
    num_de_polos_cesio = num_de_variaveis_cesio + 1
elif num_de_variaveis_cesio != 1:
    num_de_polos_cesio = num_de_variaveis_cesio

# C: matrix com os vectors de Coordenada dos Polos
C_rubidio = np.zeros((num_de_polos_cesio, num_de_variaveis_cesio),
                    dtype = float)

```

```

# lim_inf: limite inferior
lim_inf_cesio = np.min(np.array(variaveis_cesio))

print("Limite inferior = " + str(round(lim_inf_cesio, 2)))

# lim_sup: limite superior
lim_sup_cesio = np.max(np.array(variaveis_cesio))

print("Limite superior = " + str(round(lim_sup_cesio, 2)))

# A MATRIZ COORDENADA:
C_cesio = np.random.rand(num_de_polos_cesio, num_de_variaveis_cesio) *
(lim_sup_cesio - lim_inf_cesio) + lim_inf_cesio
"""
É conveniente ter pelo menos um polo na origem 0, pois
isso faz com que o modelo retorne uma Coeficiente de
Acerto (R ** 2) possivelmente maior. Logo:
"""
C_cesio[0] = 0
pd.DataFrame(C_cesio)

# In[48]:

# dist_entre_os_polos: distância entre os polos
dist_entre_os_polos_cesio = np.zeros((num_de_polos_cesio,
num_de_polos_cesio),
dtype = float)
for i in range(0, num_de_polos_cesio, 1):
    for j in range(0, num_de_polos_cesio, 1):
        dist_entre_os_polos_cesio[i, j] = np.linalg.norm(C_cesio[i]
- C_cesio[j])

pd.DataFrame(dist_entre_os_polos_cesio)

# ### 5.2 RBF VARIÁVEIS

# In[49]:

dps_max_cesio = np.max(dist_entre_os_polos_cesio) #DISTÂNCIA MÁXIMA
sigma_cesio = dps_max_cesio / np.sqrt(2 * num_de_polos_cesio) # DESVIO
PADRÃO

```

```

variaveis_cesio = np.array(variaveis_cesio)
# Matrix [R]:
# R: Matrix de Base Radial [R]
R_cesio = np.zeros((num_de_licoes_cesio, num_de_polos_cesio),
                  dtype = float)
for n in range(0, num_de_licoes_cesio, 1):
    for i in range(0, num_de_polos_cesio, 1):
        R_cesio[n, i] = fbr.fbr(num_da_funcao,
                                variaveis_cesio[n],
                                C_cesio[i],
                                sigma_cesio)

# É necessário acrescentar na matrix [R] um coluna com um's, logo:
# vector_unitario: vector de uns

vector_unitario = np.ones((num_de_licoes_cesio, 1),
                           dtype = float)
R_cesio = np.hstack((R_cesio, vector_unitario))
pd.DataFrame(R_cesio)

# ### 4.3 - OPERAÇÕES MATRICIAIS

# In[50]:

Rt_cesio = np.transpose(R_cesio)
RtR_cesio = np.matmul(Rt_cesio, R_cesio)
RtRinv_cesio = np.linalg.inv(RtR_cesio)
mpi_cesio = np.matmul(RtRinv_cesio, Rt_cesio)
VS_cesio = np.matmul(mpi_cesio, A_cesio)

# ### 4.3 RESULTADOS TREINO

# In[51]:

y_teo_cesio_treino = np.zeros(len(variaveis_cesio),
                              dtype = float)
for i in range(0, len(variaveis_cesio), 1):
    y_teo_cesio_treino[i] = sum(VS_cesio * R_cesio[i])
pd.DataFrame(y_teo_cesio_treino, columns = ['Previsão[K]'])

print(r2_score(y_teo_cesio_treino, treino_cesio['Tg(K)']))
pd.DataFrame(VS_cesio, index = ['C1', 'C2', 'Bias'], columns = ['Césio'])

```

```
# ### 4.4 - RESULTADOS TESTE
```

```
# In[52]:
```

```
variaveis_teste_cesio = =
np.array(teste_cesio[['Cesio(mol%)', 'Boro(mol%)']])
R_teste_cesio = np.zeros((num_de_licoes_cesio, num_de_polos_cesio),
                        dtype = float)
R_teste_cesio = np.hstack((R_teste_cesio, vector_unitario))
for n in range(0, len(variaveis_teste_cesio), 1):
    for i in range(0, num_de_polos_cesio, 1):
        R_teste_cesio[n, i] = fbr.fbr(num_da_funcao,
                                     variaveis_teste_cesio[n],
                                     C_cesio[i],
                                     sigma_cesio)
pd.DataFrame(R_teste_cesio)
```

```
# In[53]:
```

```
#Resultados
```

```
y_teo_teste_cesio = np.zeros(len(variaveis_teste_cesio),
                             dtype = float)
for i in range(0, len(variaveis_teste_cesio), 1):
    y_teo_teste_cesio[i] = sum(VS_cesio * R_teste_cesio[i])
pd.DataFrame(y_teo_teste_cesio, columns = ['Previsão[K]'])
r2_score(y_teo_teste_cesio, teste_cesio['Tg(K)'])
```

ANEXO 2: PREVISÕES FEITAS PARA O BORATO DE LÍTIO

Lítio (mol%)	Boro (mol%)	T _g (K)	Previsões	Resíduos
4,2	95,8	536	591,4083274	55,40832739
7,4	92,6	608	618,8029437	10,80294369
15,1	84,9	695	681,857935	13,14206503
15,3	84,7	689	683,3819901	5,618009896
2	98	558	572,6608883	14,66088829
47,36842	52,63158	714	732,2997789	18,29977891
71	29	543	518,3137855	24,68621449
2,1	97,9	552	573,5083705	21,50837046
67	33	557	558,9435849	1,943584925
30	70	765	762,60824	2,391760032
16,66667	83,33333	708	693,5719819	14,42801813
65,51724	34,48276	566	574,145418	8,145418002
18,2	81,8	720	704,4895547	15,51044533
53	47	647	692,6945742	45,6945742
54,54546	45,45454	651	679,8748069	28,87480694
25	75	770	744,4544929	25,54550711
4,22	95,78	536	591,5794897	55,57948968
9	91	637	632,39358	4,606420006
8,3	91,7	601	626,4646531	25,46465307
50	50	687	715,2996142	28,29961419
6	94	603	606,8291416	3,829141646
9,09091	90,90909	616	633,1612401	17,16124011
25,14338	74,85662	770	745,121316	24,87868401
71,42857	28,57143	544	514,0354358	29,9645642
15,3	84,7	684	683,3819901	0,6180098961
15,3	84,7	687	683,3819901	3,618009896
26,2	73,8	768	749,7812518	18,21874815
3	97	565	581,1586519	16,15865186
10	90	618	640,8030247	22,80302467
2,2	97,8	527	574,3564045	47,35640454
33,33	66,67	761	768,3776695	7,377669495
4,7619	95,2381	570	596,2197783	26,21977828
36,22375	63,77625	766	768,9672803	2,967280303
45	55	708	745,0701288	37,07012875
50	50	663	715,2996142	52,29961419
65,51724	34,48276	566	574,145418	8,145418002
20	80	750	716,5117825	33,48821752
20	80	727	716,5117825	10,48821752
40,12	59,88	741	763,0845715	22,08457145

Lítio (mol%)	Boro (mol%)	T_g(K)	Previsões	Resíduos
33,33333	66,66667	773	768,3807476	4,619252444
40	60	733	763,3801456	30,38014563
12,41	87,59	641	660,6626734	19,66267337
30	70	771	762,60824	8,391760032
15	85	670	681,0929415	11,09294152
12	88	658	657,3331868	0,6668132471
18,3	81,7	720	705,1809223	14,81907768
72,22222	27,77778	543	506,1686586	36,83134142
66,66666	33,33334	554	562,3609142	8,360914166
37,5	62,5	756	767,8866948	11,88669477
13,04348	86,95652	668	665,7603184	2,239681567
3,2	96,8	564	582,8636195	18,86361951
35	65	757	769,2301061	12,23010613
70,58823	29,41177	542	522,4425162	19,55748385
42,5	57,5	726	755,7429008	29,74290081
28,81152	71,18848	770	759,2845245	10,71547554
73,68421	26,31579	546	491,8970038	54,10299616
11	89	656	649,123103	6,876897016
62,96296	37,03704	583	600,183624	17,18362396
19	81	732	709,9451287	22,05487126
28,57143	71,42857	769	758,535275	10,46472502
60	40	603	629,6775639	26,67756387
20	80	741	716,5117825	24,48821752
24,81	75,19	759	743,5583987	15,44160128
9,09091	90,90909	615	633,1612401	18,16124011
72,97298	27,02702	544	498,8016568	45,19834318
30	70	771	762,60824	8,391760032

ANEXO 3: PREVISÕES PARA O SISTEMA BORATO DE SÓDIO

Sódio (mol%)	Boro (mol%)	T _g (K)	Previsões
17,5	82,5	701	693,315264
38,5	61,5	732	749,6111531
21,3	78,7	728	716,7749064
11,6	88,4	630	649,7941299
25	75	760	734,9212577
30	70	748	750,5090438
30	70	746	750,5090438
18,4	81,6	726	699,2570511
20	80	729	709,2468946
15,5	84,5	667	679,3705437
9,99865	90,00135	648	636,9720389
9,2	90,8	629	630,4720869
30	70	760	750,5090438
30	70	743	750,5090438
16	84	707	682,9461759
33,33333	66,66667	754	754,452105
5,4	94,6	556	598,9525092
17,4	82,6	721	692,641628
66,66666	33,33334	532	534,7813812
15	85	663	675,7397105
32,5	67,5	741	753,9756538
11,5	88,5	651	649,0028576
5,4	94,6	568	598,9525092
35	65	742	754,3645602
15	85	726	675,7397105
15	85	673	675,7397105
27,6	72,4	749	744,4070451
25	75	747	734,9212577
30	70	748	750,5090438
24,6	75,4	748	733,2118059
7,9	92,1	605	619,7767136
31,8	68,2	746	753,3106252
5	95	581	595,6049607
19,6	80,4	738	706,8216705
30	70	762	750,5090438
30	70	741	750,5090438
23,6	76,4	740	728,661241
7,3	92,7	610	614,8025949
20	80	742	709,2468946

Sódio (mol%)	Boro (mol%)	T _g (K)	Previsões
25	75	741	734,9212577
15	85	690	675,7397105
10	90	655	636,9829729
4,4	95,6	547	590,5820128
20	80	740	709,2468946
71,42857	28,57143	516	486,4941919
13,3	86,7	669	663,0179778
25	75	752	734,9212577
5	95	566	595,6049607
33,33333	66,66667	743	754,452105
30	70	756	750,5090438
35	65	723	754,3645602
33,3	66,7	742	754,439669
25	75	754	734,9212577
8	92	593	620,6036909
28,6	71,4	748	747,2701241
35	65	744	754,3645602
4	96	584	587,2339685
24,5	75,5	752	732,7744273
20	80	724	709,2468946
23,07692	76,92308	749	726,1276781
25	75	739	734,9212577
33,33333	66,66667	730	754,452105
24,4	75,6	737	732,3330755
5,33	94,67	581	598,3668157
15	85	680	675,7397105
20	80	716	709,2468946
20,2	79,8	717	710,4407219
20	80	728	709,2468946
35	65	743	754,3645602
34,2	65,8	740	754,5806343
41,17647	58,82353	715	741,8099397
33,33333	66,66667	734	754,452105
30	70	741	750,5090438
21,1	78,9	735	715,653068
10	90	617	636,9829729
35	65	736	754,3645602
30	70	755	750,5090438
5	95	572	595,6049607
24,1	75,9	735	730,9853474
28,2	71,8	748	746,1789819

Sódio (mol%)	Boro (mol%)	T_g(K)	Previsões
15	85	667	675,7397105
18	82	712	696,6435489
14,9	85,1	688	675,0071549
33,3	66,7	750	754,439669
15	85	678	675,7397105
33,3	66,7	750	754,439669
33,3	66,7	747	754,439669
30,5	69,5	749	751,4430776
15	85	667	675,7397105
33,33333	66,66667	750	754,452105

ANEXO 4: PREVISÕES FEITAS PARA O SISTEMA BORATO DE POTÁSSIO

Potássio (mol%)	Boro (mol%)	T _g (K)	Previsões
75	25	488	494,0006631
8	92	598	615,8773819
23	77	713	692,3251082
30	70	712	710,5006879
23,07692	76,92308	705	692,6045947
41,17647	58,82353	643	703,9778233
35	65	698	713,3065187
37,5	62,5	678	711,2448059
20	80	698	680,2430883
24	76	723	695,831751
78	22	485	475,2818459
80	20	487	463,761543
0	100	540	568,2504965
82	18	485	453,0794168
15,5	84,5	663	658,4418221
35	65	692	713,3065187
12,7	87,3	658	643,2041939
14,1	85,9	637	650,9521939
14,3	85,7	647	652,0389978
25,92592	74,07408	701	701,7758133
24,5	75,5	708	697,4796415
33,4	66,6	696	713,4014209
28,57143	71,42857	719	708,0671953
15	85	640	655,7994147
27	73	717	704,6011823
15	85	641	655,7994147
5	95	568	597,9133178
23	77	704	692,3251082
29,5	70,5	713	709,7284584
33,33333	66,66667	705	713,3848128
17	83	675	666,1291091
33	67	707	713,2772693
31,1	68,9	703	711,8922232
11,5	88,5	630	636,3910279

ANEXO 5: PREVISÕES FEITAS PARA O SISTEMA BORATO DE RUBÍDIO

Rubídio (mol%)	Boro (mol%)	T _g (K)	Previsões
77,27273	22,72727	519	472,1234793
36,70886	63,29114	696	690,2701119
40,012	59,988	665	684,7308435
47,36842	52,63158	633	659,805714
47,36842	52,63158	630	659,805714
23,07692	76,92308	679	675,4321619
72,22222	27,77778	497	503,0417255
5	95	546	590,9624265
13,04348	86,95652	641	633,6990724
75,90362	24,09638	495	480,1312848
61,53846	38,46154	551	575,7700461
33,33333	66,66667	694	692,1317201
30	70	691	690,2503075
25,92592	74,07408	699	683,1983736
13,04348	86,95652	641	633,6990724
16,66667	83,33333	660	650,8843954

ANEXO 6: PREVISÕES FEITAS PARA O SISTEMA BORATO DE CÉSIO

	Boro (mol%)	T_g(K)	Previsões
20	80	663	663,728172
40,012	59,988	628	661,2985167
25	75	689	672,8672265
69,23077	30,76923	507	511,1559966
15	85	655	649,5684723
28,57143	71,42857	700	675,6967972
16,66667	83,33333	650	654,7658418
74,02597	25,97403	505	486,3125366
76,63551	23,36449	480	474,2650021
25	75	693	672,8672265
6	94	585	615,8305109
35	65	687	672,0939514
76,74419	23,25581	481	473,7887372
70,05988	29,94012	460	506,6434365