



**UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA DE CONTROLE
E AUTOMAÇÃO - CECAU**



GUILHERME BRAGA BRANGIONI

**ANÁLISE E PREVISÃO DE CARGA ELÉTRICA VIA REDES
NEURAS CONVOLUCIONAIS**

**MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E
AUTOMAÇÃO**

Ouro Preto, 2023

GUILHERME BRAGA BRANGIONI

**ANÁLISE E PREVISÃO DE CARGA ELÉTRICA VIA REDES
NEURAS CONVOLUCIONAIS**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. André Almeida Santos, Me.

Coorientador: Prof. Agnaldo José da Rocha Reis, Dr.

**Ouro Preto
Escola de Minas – UFOP
2023**

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

B817a Brangioni, Guilherme Braga.
Análise e previsão de carga elétrica via redes neurais convolucionais.
[manuscrito] / Guilherme Braga Brangioni. - 2023.
40 f.: il.: color., tab., mapa.

Orientador: Me. André Almeida Santos.
Coorientador: Prof. Dr. Agnaldo José da Rocha Reis.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Escola
de Minas. Graduação em Engenharia de Controle e Automação .

1. Redes neurais (Computação). 2. Rede Neural Convolucional
(ConvNet). 3. Análise de séries temporais. 4. Convolutional Neural
Network (CNN). 5. Inteligência artificial. I. Santos, André Almeida. II. Reis,
Agnaldo José da Rocha. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 681.5

Bibliotecário(a) Responsável: Maristela Sanches Lima Mesquita - CRB-1716



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E
AUTOMACAO



FOLHA DE APROVAÇÃO

Guilherme Braga Brangioni

ANÁLISE E PREVISÃO DE CARGA ELÉTRICA VIA REDES NEURAS CONVOLUCIONAIS

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação

Aprovada em 01 de fevereiro de 2023

Membros da banca

Me. André Almeida Santos - Orientador (IHM)
Dr. Agnaldo José da Rocha Reis - Coorientador (Universidade Federal de Ouro Preto)
Dra. Adrielle de Carvalho Santana - Examinadora (Universidade Federal de Ouro Preto)

Agnaldo José da Rocha Reis, coorientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 24/03/2023



Documento assinado eletronicamente por **Agnaldo Jose da Rocha Reis, PROFESSOR DE MAGISTERIO SUPERIOR**, em 28/03/2023, às 18:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0499909** e o código CRC **FF83C7CA**.

Este trabalho é dedicado à minha família, por me proporcionar as condições necessárias para alcançar meus objetivos, e aos amigos, da vida e do curso, que, de alguma maneira, me ajudaram a superar os obstáculos da melhor maneira possível.

AGRADECIMENTOS

A Deus, pela vida e por me proporcionar inteligência e saúde para realizar esse trabalho. Aos meus pais, irmãos e toda minha família, pelo apoio e direcionamento quando precisei, e que diante da minha ausência se mantiveram presentes.

Aos professores, pelo compartilhamento de tantas informações e experiências, que com certeza foram determinantes para que eu apresentasse o melhor desempenho possível no meu processo de formação profissional.

Por último, e não menos importante, a Ouro Preto e a todas as amigadas que construí e vou levar por toda minha vida, por estarem sempre ao meu lado e por deixarem a caminhada menos tortuosa.

*"Sonhos determinam o que você quer. Ações determinam o que você conquista."
(Aldo Novak)*

RESUMO

As redes neurais artificiais têm se expandido e consolidado ao longo do tempo. Dentre suas aplicações, destacam-se a previsão de carga elétrica (*i.e.* demanda energética de um sistema) aplicando-se metodologias de análise de séries temporais e operações de convolução, identificação e classificação comportamental violento ou não violento, demandas industriais, e aplicações de segurança e controle operacional. Neste trabalho, uma série temporal contendo valores de carga elétrica foi avaliada por uma rede neural convolucional, que realizou a previsão da carga dentro de uma janela de tempo. A previsão da carga pode garantir um planejamento eficiente das manutenções nos equipamentos de geração, transmissão e distribuição de energia elétrica, realocação eficiente das cargas do sistema, identificação de obstáculos no cenário energético, além de auxiliar na identificação de melhorias no sistema. A métrica aplicada do RMSE apresentou erro de 0,028% para a previsão da carga elétrica no subsistema elétrico Sudeste/Centro-Oeste de Minas Gerais, o qual demonstra a eficiência do método utilizado.

Palavras-chaves: Redes Neurais. Previsão. Séries Temporais.

ABSTRACT

Artificial Neural Networks are expanding and establishing over time. Between its applications, electric load prediction (*i.e.* electrical demand) using Time Series analysis and convolution operations, violent or non-violent behavior classification and identification, industrial needs, and security and operational control are some applications that stand out. On this thesis, a time series containing electric load values was evaluated by a convolutional neural network, and predicted the load in a specific period. The load prediction can provide an efficient maintenance schedule on generation, transmission and distribution electrical equipments, efficient load allocation, obstacle identification on electrical systems, furthermore, helps on improvements in general. The proposed methodology had an RMSE error close to 0,028% for the electric load prediction on Southeast/Middle West Minas Gerais Subsystem, which shows the method efficiency.

Key-words: Neural Networks. Forecast. Time Series.

LISTA DE ILUSTRAÇÕES

Figura 1 – Neurônios e Redes Neurais Artificiais	15
Figura 2 – Total mensal de passageiros em uma empresa aérea nos EUA entre 1949 e 1960	15
Figura 3 – Consumo de energia elétrica da Região Nordeste do Brasil, na escala original e logarítmica, em MWh, no período de 2004 a 2012	16
Figura 4 – Arquitetura MLP	18
Figura 5 – Modelo matemático de um neurônio	19
Figura 6 – Funções de ativação	19
Figura 7 – Operação de convolução unidimensional	20
Figura 8 – Operação de convolução unidimensional com matrizes	21
Figura 9 – Comparando A) <i>Aprendizado de Máquina</i> e B) <i>Aprendizado Profundo</i>	24
Figura 10 – Python	24
Figura 11 – Jupyter e Spyder	25
Figura 12 – Locais e negócios da CEMIG no Brasil	26
Figura 13 – Dados utilizados organizados em uma série temporal	27
Figura 14 – Decomposição da Série Temporal	28
Figura 15 – Conjunto de treinamento normalizado	29
Figura 16 – Conjunto de teste normalizado	29
Figura 17 – Janela de previsão	29
Figura 18 – Estrutura da Rede Neural	32
Figura 19 – Descida do gradiente	34
Figura 20 – Erro da RNA em função das épocas	35
Figura 21 – Desempenho da RNA	35
Figura 22 – Resultado: Real vs. Previsto	36

LISTA DE TABELAS

Tabela 1 – Resumo da estrutura da Rede Neural	32
Tabela 2 – Parâmetros de treinamento da RNA	33
Tabela 3 – Avaliação do modelo. Método <i>evaluate()</i>	36

LISTA DE ABREVIATURAS E SIGLAS

AF	<i>Activation Function</i>
ANEEL	Agência Nacional de Energia Elétrica
API	<i>Application Programming Interface</i>
CEMIG	Companhia Energética de Minas Gerais
CNN	<i>Convolutional Neural Network</i>
IDE	<i>Integrated Development Environment</i>
LSTM	<i>Long Short-Term Memory</i>
MLP	<i>Multilayer Perceptron</i>
MSE	<i>Mean Squared Error</i>
MWh	megawatt-hora
RNA	Redes Neurais Artificiais
RNN	<i>Recurrent Neural Network</i>
SE/CO	Sudeste/Centro-Oeste
SIN	Sistema Interligado Nacional
TCN	<i>Temporal Convolutional Neural Network</i>
TS	<i>Time Series</i>
UF	Unidade da Federação
UFOP	Universidade Federal de Ouro Preto

LISTA DE SÍMBOLOS

kV	quilovolt
kWh	quilowatt-hora
V	volt
*	Convolução

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivo geral	16
1.2	Objetivos específicos	16
1.3	Motivação	16
1.4	Estrutura do trabalho	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Redes Neurais Artificiais	18
2.1.1	<i>O Neurônio Artificial</i>	18
2.1.2	<i>Redes Neurais Convolucionais</i>	19
2.1.3	<i>LSTM - Long Short Term Memory Neural Networks</i>	22
2.2	Séries Temporais	22
2.2.1	<i>RNA e Séries Temporais</i>	23
2.3	Aprendizado Profundo: <i>Deep Learning</i>	23
2.4	<i>Python</i>	23
3	DESENVOLVIMENTO	26
3.1	Caracterização da área de estudo	26
3.2	Dados	27
3.3	Metodologia	28
3.3.1	<i>Pré-processamento dos dados</i>	28
3.3.2	<i>StandardScaler</i>	30
3.3.3	<i>Variáveis utilizadas</i>	30
3.3.4	<i>Estrutura do modelo da RNA</i>	31
3.3.4.1	Parâmetros de treinamento	32
3.3.4.2	Otimizador - Adam	33
3.3.4.3	Erro - MSE	33
3.3.4.4	Épocas	33
3.3.4.5	Métrica - RMSE	33
3.3.5	<i>Cálculo do erro - Descida do Gradiente</i>	34
4	RESULTADOS	35
5	CONCLUSÃO	37
	REFERÊNCIAS	38

1 INTRODUÇÃO

A energia elétrica, desde sua descoberta até os dias atuais, tem sido indispensável para o desenvolvimento mundial do ponto de vista tecnológico, econômico e também social. Conforme o tempo passou, a eletricidade tornou-se mais necessária no dia a dia das pessoas, o que fez com que a demanda pela energia também aumentasse consideravelmente, sendo importante o desenvolvimento e implantação de fontes de energia mais eficientes.

Do ponto de vista geográfico, a expansão dos países em desenvolvimento depende muito da qualidade e disponibilidade de energia elétrica, visto que ela permite uma aceleração nos processos produtivos. Segundo [Freitas e Silveira \(2015\)](#), é comum relacionar a demanda por energia ao crescimento de um país, pois ela também proporciona uma melhor qualidade de vida à população, melhor qualidade em eletrodomésticos, eletrônicos e outras ferramentas de uso doméstico.

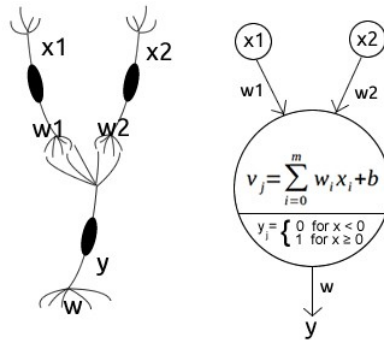
A energia elétrica atualmente funciona como um mercado, sendo vendida e comprada por concessionárias e revendida aos consumidores (residenciais, rurais ou industriais). Entender o comportamento da sociedade e como isso influencia a demanda por energia possibilita o desenvolvimento de métodos de previsão de demanda, fazendo com que esse mercado opere de modo a maximizar seus lucros e reduzir as perdas, sejam financeiras, obtendo o melhor preço pela energia, ou sejam elétricas, garantindo a entrega de energia de qualidade e ininterrupta ao consumidor final ([LOPES, 2005](#)). Essa previsão pode ser aplicada a sistemas de geração de energia (hidrelétricas e termelétricas), transmissão (subestações e linhas de transmissão), distribuição (subestações, linhas de distribuição e redes de distribuição), planejamento de intervenções, segurança do sistema, entre outras.

Dentre as abordagens que surgiram com o desenvolvimento dos métodos computacionais, uma delas foi a utilização de Redes Neurais Artificiais (RNA), que foram inspiradas no cérebro humano ([Figura 1](#)). Elas se organizam em camadas e realizam o processamento da informação por meio de pesos sinápticos e funções de ativação, fazendo a relação entre neurônios de camadas diferentes e sintetizando o resultado. As RNA consolidam-se como uma ferramenta de resolução de problemas complexos, que exigem a modelagem e análise de dados sob aspectos computacionais e estatísticos simultaneamente ([KOVÁCS, 2002](#)).

Anteriormente, as técnicas utilizadas para previsão se baseavam em regressões e interpolações, e nos dias atuais o poder de processamento permite a utilização de operações mais complexas, como a convolução. As análises da curva de carga, segundo [Park et al. \(1991\)](#), trouxeram duas vertentes de estudo. Uma vertente aproxima a curva de carga em um padrão de uma série temporal, que utiliza técnicas de previsão por meio de análise de séries temporais. Já a outra vertente leva em conta que a curva de carga é influenciada principalmente por fatores

climáticos, e encontra relações entre o clima e a demanda de energia. A previsão nesse segundo caso é dada inserindo a relação da previsão do tempo com a demanda energética.

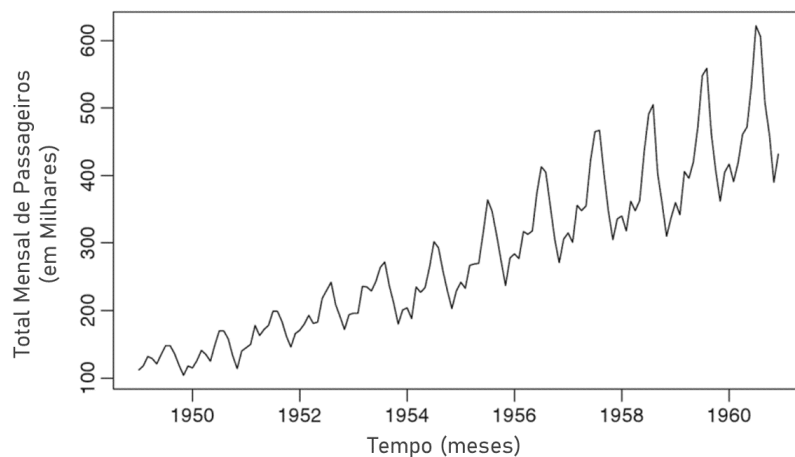
Figura 1 – Neurônios e Redes Neurais Artificiais



Fonte: [Wichmann et al. \(2020\)](#)

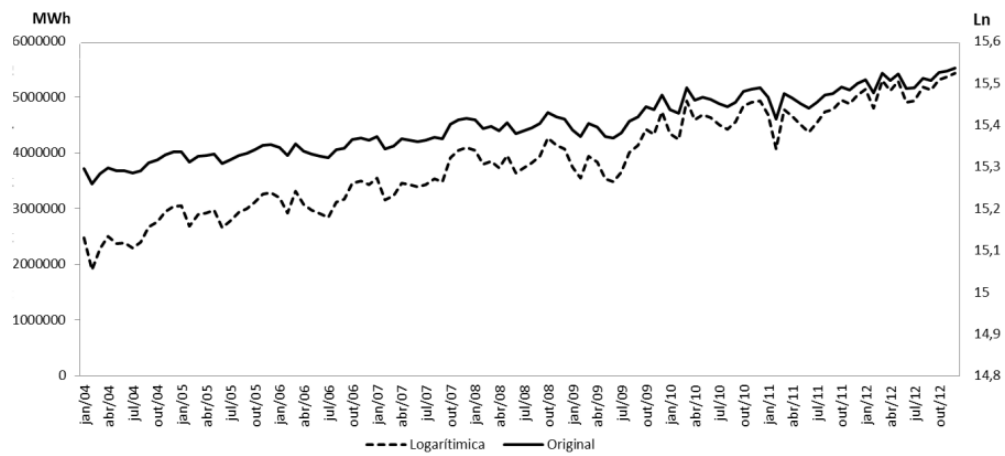
Essa previsão de carga elétrica pode ser realizada a partir da utilização de RNA em conjunto com a análise de séries temporais. Existem no nosso mundo, fenômenos das mais variadas origens (naturais, sociais, econômicas etc.), que ao serem observados e quantizados, permitem perceber uma sequência de dados que segue uma tendência distribuída ao longo do tempo. Essa sequência ordenada recebe o nome de Série Temporal (TS - *Time Series*) ([SOUZA, 1981](#)), exemplificada na [Figura 2](#) e na [Figura 3](#).

Figura 2 – Total mensal de passageiros em uma empresa aérea nos EUA entre 1949 e 1960



Fonte: [Box et al. \(2015\)](#)

Figura 3 – Consumo de energia elétrica da Região Nordeste do Brasil, na escala original e logarítmica, em MWh, no período de 2004 a 2012



Fonte: Junior et al. (2018)

No contexto apresentado anteriormente, foi proposto neste trabalho o desenvolvimento de um previsor baseado em redes neurais para a previsão da carga elétrica num determinado subsistema. Esse previsor foi desenvolvido em duas frentes, ou seja, duas redes neurais diferentes com um propósito em comum, a fim de avaliar a viabilidade da aplicação desses dois tipos de RNA e comparar sua eficiência.

1.1 Objetivo geral

Aplicar a metodologia de Redes Neurais Convolucionais em séries temporais e avaliar sua eficiência para o tipo de problema proposto.

1.2 Objetivos específicos

- Avaliar a utilização de inteligência artificial para fins de previsão de carga no sistema elétrico;
- Coletar e analisar dados do subsistema Sudeste/Centro-Oeste (SE/CO);
- Verificar a aplicabilidade de Redes Neurais Convolucionais Unidimensionais em problemas de séries temporais;

1.3 Motivação

Durante alguma falha ou manutenção da rede elétrica, alguns pontos do sistema ficam prejudicados, pois sustentam cargas maiores do que a esperada, ou até mesmo acima da permitida para determinados equipamentos. Os equipamentos em questão podem sofrer danos, o

que prejudica a disponibilização de energia a médio e longo prazo, sendo talvez necessária a substituição de um equipamento.

A falta da previsão de carga elétrica ou uma previsão ineficaz pode levar a intervenções mal programadas e perda ou sobrecarga de equipamentos, resultando em um ônus para as empresas, já que sua remuneração é baseada no tempo que os equipamentos e a energia estão disponíveis para utilização.

1.4 Estrutura do trabalho

O presente trabalho está estruturado em 5 capítulos. No capítulo 1 - Introdução, encontra-se a contextualização do assunto, objetivo geral e objetivos específicos e a motivação para a realização deste estudo. No capítulo 2 - Revisão da Literatura, encontra-se a revisão da literatura referente ao tema abordado. No capítulo 3 - Desenvolvimento, estão a caracterização da área de estudo, descritas as metodologias utilizadas desde a organização dos dados até o resultado final da rede neural. Já no capítulo 4 - Resultados, estão dispostos os resultados obtidos e sua análise. Por fim, no capítulo 5 - Conclusão, estão as conclusões obtidas e as sugestões para trabalhos futuros.

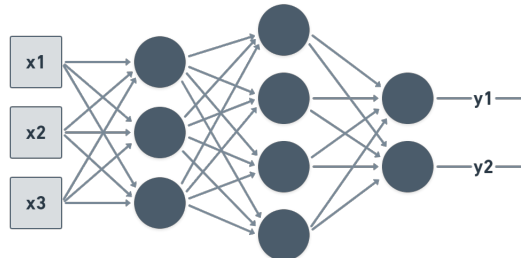
2 FUNDAMENTAÇÃO TEÓRICA

2.1 Redes Neurais Artificiais

Uma Rede Neural Artificial (RNA) é um modelo matemático inspirado no funcionamento de um conjunto de neurônios biológicos. Uma das arquiteturas de RNA utilizada atualmente é conhecida como Perceptron de Múltiplas Camadas (*MLP - Multilayer Perceptron*), cujos neurônios das suas camadas são conectados entre si.

As RNA do tipo MLP têm uma estrutura como a da [Figura 4](#), sendo x_1 , x_2 e x_3 as entradas, os círculos representam os neurônios de cada camada, as setas representam as sinapses ou as relações entre os neurônios, e y_1 e y_2 as saídas. A primeira camada de neurônios é chamada de camada de entrada (ou *input layer*), responsável por receber os dados amostrados. A partir da segunda camada até a penúltima camada são as chamadas camadas ocultas (ou *hidden layers*), onde ocorre todo o processamento dos dados. A última camada é chamada de camada de saída (ou *output layer*), responsável por computar os resultados ([GONÇALVES; SORTE, 2014](#)).

Figura 4 – Arquitetura MLP



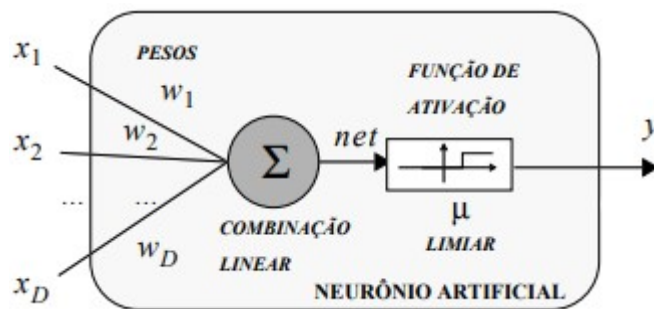
Fonte: Elaborado pelo autor (2022)

2.1.1 O Neurônio Artificial

O modelo de neurônio artificial foi inspirado no comportamento de um neurônio real. A [Figura 5](#) mostra a representação de um neurônio artificial. Os dados são inseridos na camada de entrada, representadas por x_1 , x_2 e x_D , e são multiplicadas por pesos sinápticos w_1 , w_2 e w_D , que ponderam a influência de cada entrada x . O resultado do processamento dos dados consiste em uma combinação linear dos dados de entrada, net , e deve respeitar um limiar μ . A representação matemática do processo está descrita na [Equação 2.1](#).

$$net = w_1x_1 + w_2x_2 + \dots + w_Dx_D = \sum_{j=1}^D w_jx_j - \mu \quad (2.1)$$

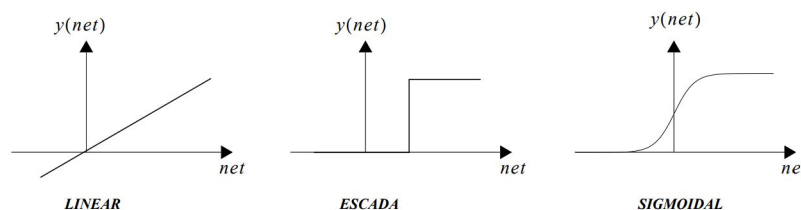
Figura 5 – Modelo matemático de um neurônio



Fonte: Rauber (2005)

O resultado na saída y é dado a partir da aplicação de uma função de ativação, que pode ser uma função Escada (ou Degrau - *Step*), Rampa ou Linear, *Sigmoidal*, *ReLU*, *Leaky ReLU*, *Softmax*, *Tanh* dentre outros (RAUBER, 2005), e a utilização de cada uma delas varia de acordo com a aplicação da RNA. A Figura 6 mostra exemplos dessas funções.

Figura 6 – Funções de ativação



Fonte: Rauber (2005)

Uma rede neural sem as funções de ativação das camadas se torna um modelo de regressão linear. As funções de ativação são as responsáveis pela capacidade de aprendizado da rede neural, possibilitando a execução de tarefas mais complexas.

2.1.2 Redes Neurais Convolucionais

Como alternativa à MLP, foram desenvolvidas as Redes Neurais Convolucionais (*CNN* - *Convolutional Neural Networks*), ou *ConvNets*, que são um tipo especializado de RNA para processamento de dados cuja topologia é conhecida, como por exemplo a base de dados MNIST e, no caso desse trabalho, séries temporais, podendo ser interpretadas como grades unidimensionais com amostras em intervalos de tempo regulares.

O nome Redes Neurais Convolucionais indica que a RNA emprega a operação matemática chamada de *convolução* (GOODFELLOW et al., 2016), um tipo especializado de operação linear, que utiliza filtros (*kernels*) para realizar as operações e identificar características específicas dos dados. As CNN são redes neurais simples que usam a convolução no lugar de multiplicação de matrizes em pelo menos uma de suas camadas, e a principal diferença para as MLP é o

compartilhamento de pesos sinápticos, permitindo um treinamento relativamente mais simples, já que a quantidade de pesos é menor.

A operação de convolução é uma operação matemática realizada a partir de duas funções (ou sinais), $x(t)$ e $h(t)$, com o objetivo de gerar uma terceira função, $y(t)$, que é o somatório do produto entre as funções, dentro da região de sobreposição de $x(t)$ e $h(t)$, à medida em que uma função se desloca sobre a outra. A função $h(t)$ representa a resposta ou comportamento de um sistema quando submetido ao sinal de entrada $x(t)$. A convolução é definida pela [Equação 2.2](#), cujo operador $*$ é o símbolo que representa a operação de convolução entre as duas funções, e a [Equação 2.3](#) é chamada de integral da convolução. Lê-se que a função $y(t)$ é a convolução entre $x(t)$ e $h(t)$, e a variável de integração é alterada para τ (letra grega *Tau*).

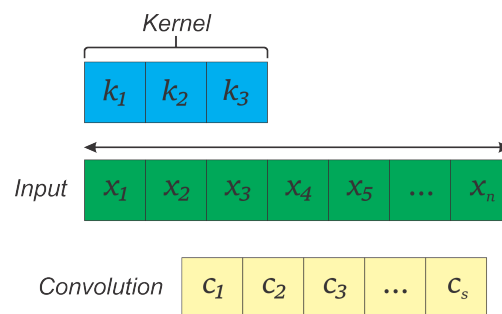
$$y(t) = x(t) * h(t) \quad (2.2)$$

$$y(t) = \int_{-\infty}^{+\infty} x(\tau) \cdot h(t - \tau) \cdot d\tau \quad (2.3)$$

A convolução é uma operação linear, portanto obedece aos princípios de comutação, distribuição, associação e superposição ([SOVIERZOSKI, 2011](#)).

A [Figura 7](#) mostra uma operação unidimensional de convolução onde k_1 a k_3 representam o *Kernel*, x_1 a x_n são dados de entrada, e c_1 a c_s são os mapas de características extraídos após a operação de convolução mostrada pela [Equação 2.4](#).

Figura 7 – Operação de convolução unidimensional



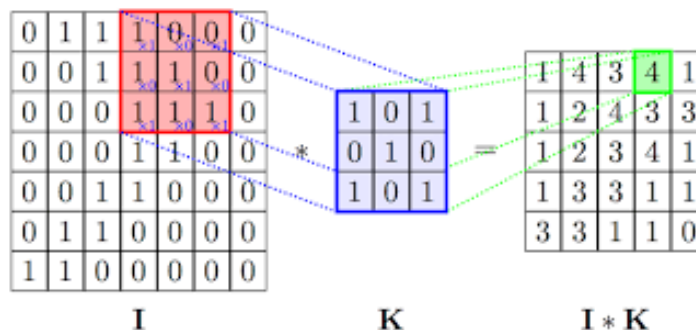
Fonte: Elaborado pelo autor (2023)

$$\begin{aligned} c_1 &= k_1x_1 + k_2x_2 + k_3x_3 \\ c_2 &= k_1x_2 + k_2x_3 + k_3x_4 \\ c_s &= k_1x_{p-2} + k_2x_{p-1} + k_3x_p \end{aligned} \quad (2.4)$$

Outro exemplo é a operação de convolução pela multiplicação de duas matrizes I (matriz I - dados de entrada) e K (matriz K - *Kernel*), de forma que a matriz resultante $I * K$ tem

dimensão menor que a matriz I obedecendo o *padding* definido para o *kernel*, como exemplificado na Figura 8¹.

Figura 8 – Operação de convolução unidimensional com matrizes



Fonte: laptrinhx.com/

Uma CNN possui estruturas padronizadas, chamadas camadas convolucionais, compostas de neurônios, responsáveis por filtrar uma parte específica dos dados (RODRIGUES, 2020). Cada neurônio se conecta a uma parte dos dados e a cada conexão é atribuído um peso.

Como dito anteriormente, apenas algumas entradas são conectadas a um neurônio, e os pesos dessas conexões são compartilhados durante o treinamento da rede, o que diminui a quantidade de parâmetros que devem ser aprendidos e, portanto, otimizando o tempo de treinamento via *backpropagation*.

Logo após a camada de convolução, utiliza-se uma camada de agrupamento ou junção, denominada de Camada de *Pooling*. Essa camada tem a função de diminuir a complexidade dos dados a fim de agilizar o treinamento. Essa camada de *pooling* agrupa um conjunto de dados e seleciona um dado específico para representar aquele conjunto, que geralmente é selecionado pela função de máximo (VARGAS; PAES; VASCONCELOS, 2016). Um exemplo de arquitetura que aplica a camada de convolução seguida de *pooling* é a arquitetura GoogLeNet disponível em Szegedy et al. (2015), que apresenta cinco camadas de convolução seguidas sempre de camadas de *pooling*.

As camadas que se sucedem à esse conjunto convolução-*pooling* são camadas totalmente conectadas como as usadas nas MLP, responsáveis por traçar linhas de definição, que serão utilizadas por camadas classificadoras. Essas são camadas mais simples, mas não deixam de ser importantes para a eficácia e resultados da rede.

As CNN têm diversas aplicações, dentre elas, está o reconhecimento de imagens e vídeos. Um exemplo clássico é a utilização desse tipo de rede para o reconhecimento de dígitos da base de dados *MNIST*, disponível em *THE MNIST DATABASE*².

¹ <https://laptrinhx.com/implementing-kernel-filter-convolutional-in-your-own-1511826395/>

² <http://yann.lecun.com/exdb/mnist/>

2.1.3 LSTM - Long Short Term Memory Neural Networks

As redes neurais recorrentes do tipo LSTM (*Long Short Term Memory*) são utilizadas para lidar com problemas com alta dependência do tempo, ou seja, a ordem dos dados influencia no comportamento e no resultado da rede (GRAVES; FERNÁNDEZ; SCHMIDHUBER, 2005).

As camadas ocultas das LSTM consistem em blocos recorrentes interconectados (Blocos de Memória), que armazenam os valores das entradas anteriores para prever melhor o próximo valor. Essa leitura de dados pode ser feita em uma ou mais direções (*Forward*, ou *Backward*, ou *Forward & Backward* simultaneamente), dependendo da aplicação ou complexidade dos dados.

2.2 Séries Temporais

Uma série temporal pode ser descrita como sendo um conjunto de observações discretas, ou seja, dados obtidos ao realizar amostragens em períodos equidistantes entre si, e apresentam dependências entre os elementos anteriores. São várias as situações em que os dados aparecem como séries temporais, como por exemplo a quantidade de produtos entregues por determinada fábrica a cada mês, quantidade de chuva diária, número de acidentes automobilísticos semanais, dentre outros (BOX et al., 2015).

A análise de séries temporais é uma área dentro da estatística bem definida, sendo que certos dados são ignorados, como, por exemplo, aqueles que não possuem correlação ou dados muito discrepantes (*outliers*), que causariam prejuízos à interpretação dos dados e no resultado final.

Ao analisar uma série temporal, percebe-se que seus dados seguem uma **tendência**, que norteia a evolução dos dados ao longo do tempo, seja ela de crescimento ou decréscimo. Percebe-se, também, que os dados apresentam **sazonalidade**, ou seja, em intervalos de tempo regulares, por exemplo, nas estações do ano, os dados apresentam um comportamento semelhante a períodos anteriores. Quando observadas a tendência e a sazonalidade, há uma parcela dos dados remanescentes que não influencia nessa análise, e recebe o nome de **Resíduo**. O processo de identificar esses três componentes é denominado **Decomposição** de Séries Temporais (ATWAN, 2022).

Segundo Bowerman e O'Connell (1979), a análise e previsão de séries temporais tem várias aplicações, como:

- Planejamento econômico;
- Previsão de Vendas;
- Controle de estoque;
- Planejamento e controle da produção;

- Avaliação de estratégias econômicas alternativas;
- Previsão orçamentária;
- Gestão de risco financeiro;
- Avaliação de modelos.

Isso permite que a análise de séries temporais via redes neurais artificiais sejam amplamente utilizadas em quase todas as áreas de planejamento e previsão, pois oferecem uma gama de ferramentas para esse propósito.

2.2.1 RNA e Séries Temporais

Os problemas de previsão envolvendo redes neurais estão cada dia mais presentes. Os resultados obtidos por [Zhang, Patuwo e Hu \(1998\)](#), mostram que, utilizando redes neurais, a previsão de séries temporais pode ser realizada com alta precisão e assertividade devido à sazonalidade dos dados.

Uma curva de carga é entendida como uma série temporal com periodicidade conhecida, sendo possível prevê-la de acordo com a estação do ano, clima e horário. Para a análise de séries temporais, o trabalho de [Park et al. \(1991\)](#) propõe diversos métodos, e aplica redes neurais para a previsão de uma série temporal que representa carga elétrica.

2.3 Aprendizado Profundo: *Deep Learning*

Tal qual [Marumo \(2018\)](#), *Deep Learning* (Aprendizado Profundo) é uma parte do *Machine Learning* (Aprendizado de Máquina) que faz uso de algoritmos para aprender níveis de abstração diferentes, utilizando várias camadas não lineares. Isso que torna possível o reconhecimento de padrões dos dados de entrada durante o próprio processo de aprendizado, enquanto em técnicas tradicionais de aprendizado de máquina a identificação de padrões é realizada manualmente por profissionais especializados.

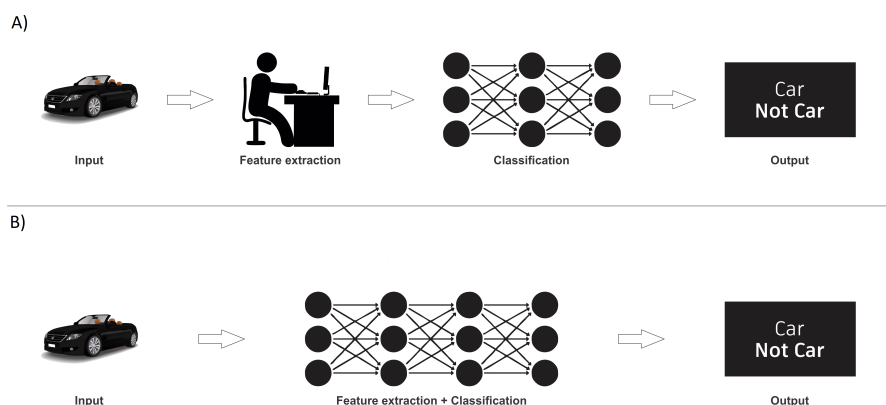
A [Figura 9](#) retrata essa diferença na implementação de redes neurais baseadas em A) *Aprendizado de Máquina* e B) *Aprendizado Profundo*.

Visto que no *deep learning* é possível a extração de características mais abstratas se comparadas ao aprendizado de máquina tradicional, a evolução do *deep learning* está condicionada também ao poder computacional disponível ([HOSAKI; RIBEIRO, 2021](#)).

2.4 *Python*

A linguagem de Programação utilizada neste trabalho foi a linguagem *Python* ([Figura 10](#)). Essa linguagem foi escolhida por ser uma linguagem relativamente simples, de fácil aprendizado,

Figura 9 – Comparando A) *Aprendizado de Máquina* e B) *Aprendizado Profundo*



Fonte: Elaborado pelo autor (2022)

alto nível, e utilizada para diversos fins. Outro ponto importante é que existem diversas bibliotecas em *Python* para aplicação em *Machine Learning*, facilitando o tratamento e visualização dos dados, e organização do código.

Figura 10 – Python



Fonte: <https://www.python.org/>

Além disso, existem interfaces *web* em *Python* que permitem o compartilhamento dos projetos facilmente, como o *Jupyter Notebook* (Figura 11a) e o *Spyder* (Figura 11b), que, além de permitir a execução do código linha por linha (*debug*) para avaliar se o que está ocorrendo realmente é o esperado, apresentam interfaces amigáveis e já comuns para quem trabalha com IDEs de programação.

Assim, *Python* é uma linguagem simples, fácil de se aprender e aplicar, e com muitas funcionalidades que a permite ser aplicada em diversas áreas, como *Machine Learning*, *marketing*, vendas, e outras mais.

Figura 11 – Jupyter e Spyder



(a) Jupyter



(b) Spyder

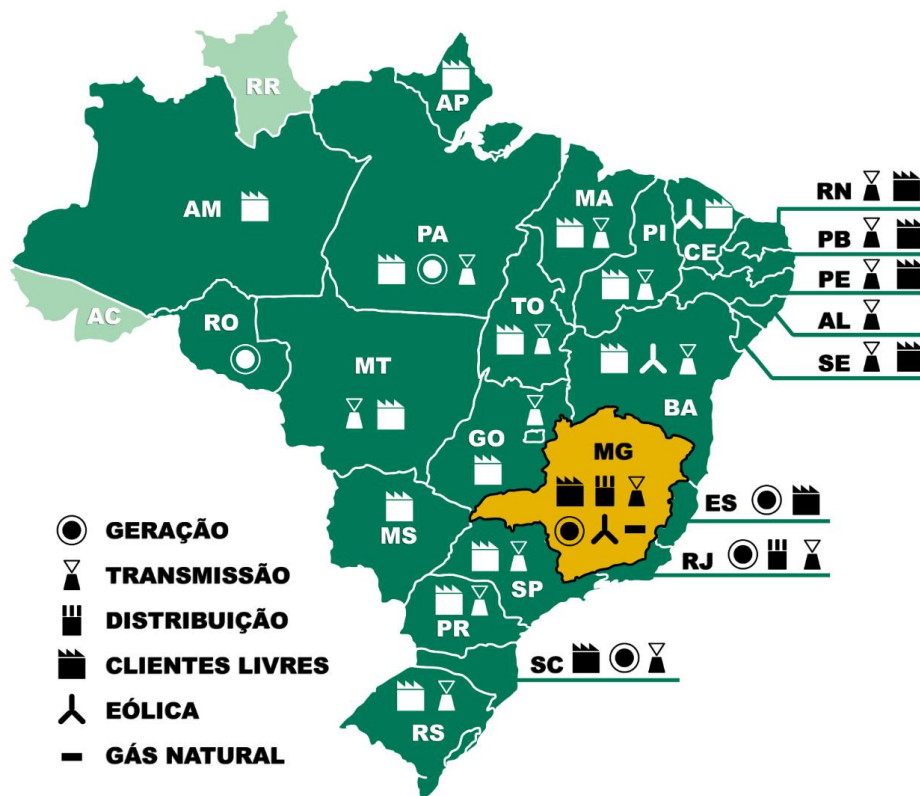
Fonte: a) <https://jupyter.org/> e b) <https://www.spyder-ide.org/>

3 DESENVOLVIMENTO

3.1 Caracterização da área de estudo

O objeto de estudo desse trabalho é a curva de carga do SIN, mais especificamente a região SE/CO, sistema no qual o estado de Minas Gerais está presente e no qual a CEMIG é a principal concessionária de energia, envolvendo geração, transmissão e distribuição da energia elétrica, e outros serviços, conforme mapa da [Figura 12](#).

Figura 12 – Locais e negócios da CEMIG no Brasil



Fonte: novoportal.cemig.com.br/quem-somos/

A CEMIG é a sexta maior geradora de energia elétrica do país. Possui 89 usinas em operação, cuja capacidade instalada é de 5,9GW, contando com todas as empresas subsidiárias integrais, controladas e coligadas. Além disso, opera a segunda maior rede de transmissão de energia, cuja extensão é de quase 10000km, responsável pelo transporte de energia elétrica desde as usinas até os consumidores. Essa rede de transmissão também transporta a energia comprada da Usina de Itaipú. No campo da distribuição, a CEMIG ocupa a primeira posição no sistema brasileiro em extensão de rede, atendendo aproximadamente 96% do estado de Minas Gerais. Todos esses e outros dados podem ser consultados no site da CEMIG¹.

¹ <https://novoportal.cemig.com.br/quem-somos/>

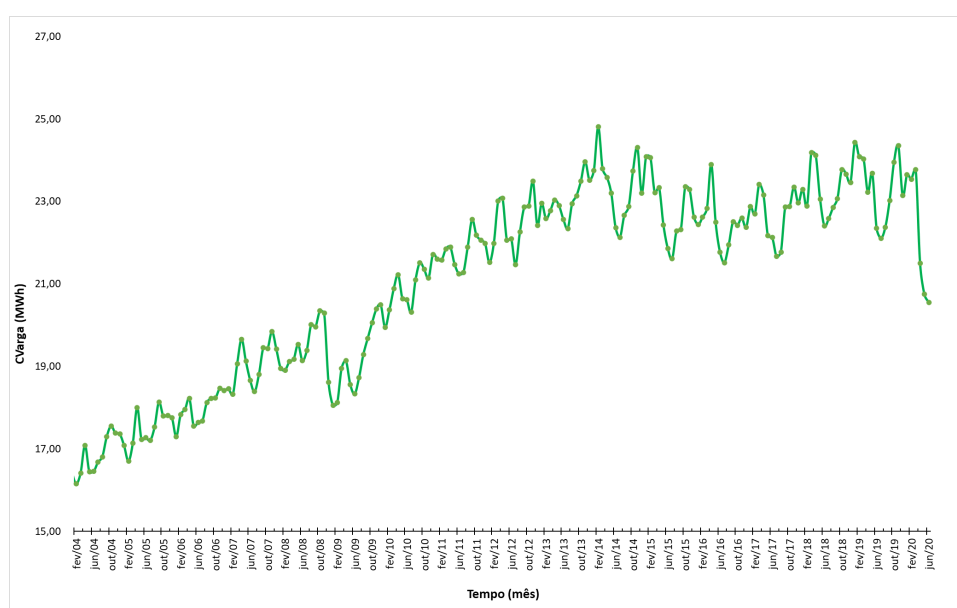
Os dados são disponibilizados pela ONS e ANEEL, por meio da Empresa de Pesquisa Energética - EPE. Essa base de dados contém diversas informações como consumo total por região, por tipo de consumidores, por UF, quantidade de consumidores divididos por tipo, dentre outros, e estão especificados na Seção 3.2 - Dados.

3.2 Dados

Os dados utilizados estão disponíveis para consulta no site da EPE ([EPE, 2022](#)), que disponibiliza o histórico mensal, a partir de 2004, do consumo de energia elétrica em níveis nacional, regional e por subsistemas, segmentado pelas classes residencial, industrial, comercial e outros (rural, serviço público e iluminação pública). Os dados são atualizados pela Superintendência de Estudos Econômicos e Energéticos – SEE da Diretoria de Estudos Econômico-Energéticos e Ambientais – DEA. Os dados utilizados para este trabalho vão de Janeiro de 2004 até junho de 2020, porém é possível acessar os dados do ano corrente de acordo com a data de acesso.

Os dados são apresentados em MWh, subdivididos por região: Norte, Nordeste, Sudeste, Sul e Centro-Oeste, ou subsistema elétrico: Sistemas Isolados, Norte, Nordeste, Sudeste/Centro-Oeste e Sul. Para este trabalho, os dados a serem considerados são referentes ao subsistema elétrico SE/CO, e podem ser visualizados na [Figura 13](#), na forma de uma série temporal, pois as amostras foram feitas mês a mês, sendo que a cada mês, a carga do sistema segue uma tendência e depende dos meses anteriores. Vale ressaltar que todos os dados devem ser organizados e pré-processados, ou seja, normalizados, de modo que a RNA trabalhe com números em escala semelhante, sem haver distorção de valores nem perda de informação ([AZURE, 2022](#)).

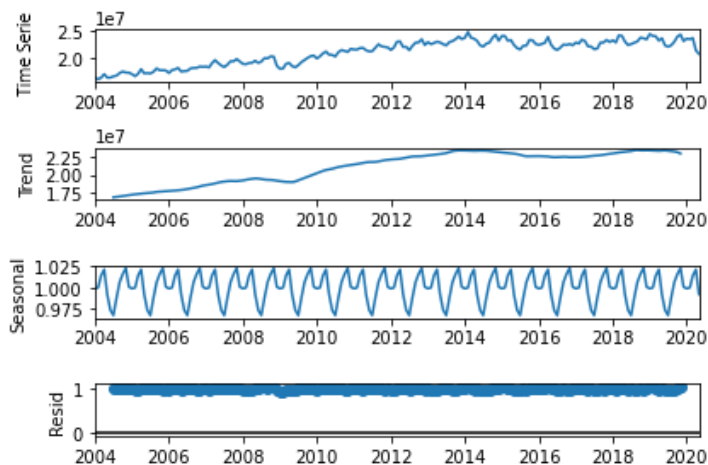
Figura 13 – Dados utilizados organizados em uma série temporal



Fonte: [EPE \(2022\)](#)

A seguir, a [Figura 14](#) apresenta a decomposição dessa série temporal, ou seja, sua tendência - *Trend*, Sazonalidade - *Seasonal* e resíduo - *Resid*.

Figura 14 – Decomposição da Série Temporal



Fonte: Elaborado pelo autor (2022)

3.3 Metodologia

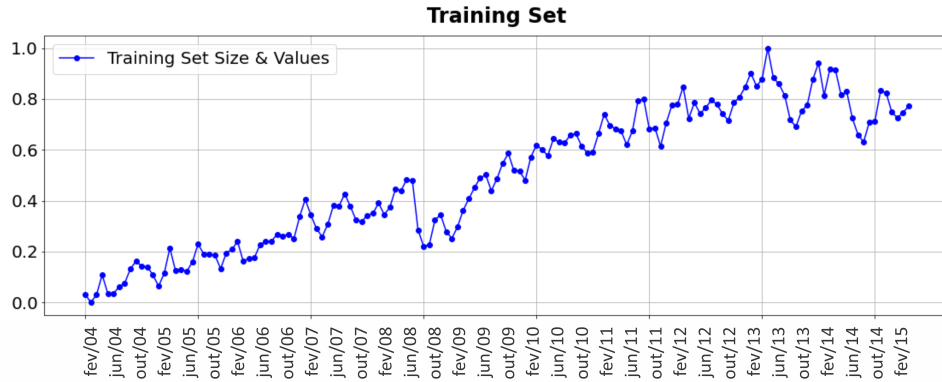
Existem pesquisas importantes na área das CNN que as utilizam para reconhecimento de padrões e dados organizados como Séries Temporais. Um desses estudos, intitulado *1D Convolutional Neural Networks for Time Series Modeling* desenvolvido e apresentado por [Janos e Roach \(2018\)](#), foi submetido ao Congresso *PyData*, em 2018, na cidade de Los Angeles, e teve como objetivo compartilhar técnicas aplicadas utilizando ferramentas da linguagem de programação *Python*. A apresentação desse trabalho pode ser assistida no *YouTube*².

3.3.1 Pré-processamento dos dados

Os dados foram organizados em uma tabela, onde na primeira coluna estão as datas, formatada como ano-mês ('*aaaa-mm*'), e na segunda coluna está a carga em MWh. O *script* da RNA realiza a leitura do arquivo e armazena o conteúdo em uma variável, de modo que esse conjunto de dados é separado, sendo um conjunto de Treinamento ([Figura 15](#)), com 70% dos dados, e outro conjunto de Teste ([Figura 16](#)), com 30% dos dados, e armazenado em variáveis do tipo "lista". Em seguida, aplica-se a classe *StandardScaler* e o método *fit_transform* proveniente dessa classe, para normalizar e padronizar os dados a partir de sua média e variância. Durante o processamento da rede, ela utiliza uma janela para realizar a previsão de um valor subsequente a essa janela, como exemplificado na [Figura 17](#). A janela total utilizada foi de 25 elementos, sendo 24 desses utilizados pela rede como histórico (*Input Width*) para realizar a previsão do 25º elemento, ajustar os parâmetros e seguir com as previsões.

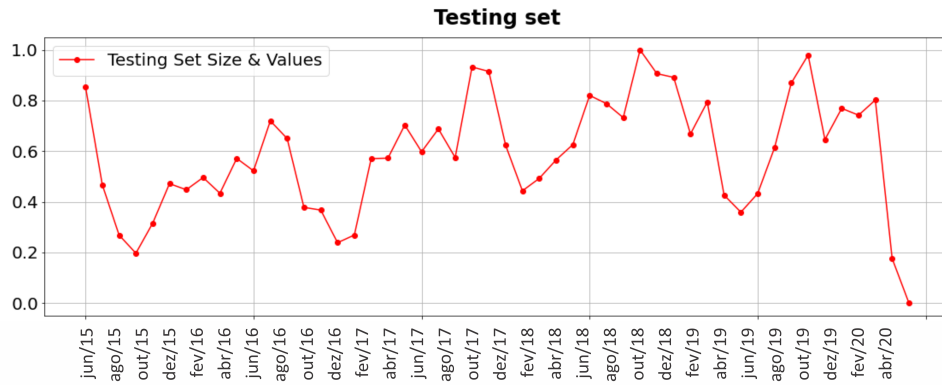
² <https://www.youtube.com/watch?v=nMkqWxMjWzg&t=492s>

Figura 15 – Conjunto de treinamento normalizado



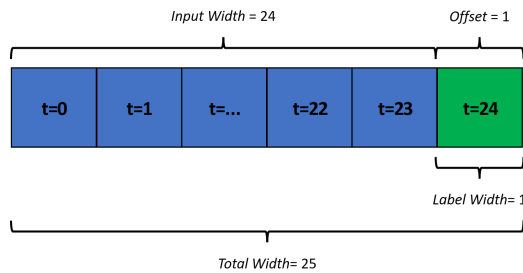
Fonte: Elaborado pelo autor (2022)

Figura 16 – Conjunto de teste normalizado



Fonte: Elaborado pelo autor (2022)

Figura 17 – Janela de previsão



Fonte: Elaborado pelo autor (2022)

Onde:

Input Width: Tamanho do histórico utilizado.

Offset: Tamanho do passo da janela (no caso, 1 em 1).

Label Width: Ou *Output Width*, é a quantidade de valores que serão previstos utilizando o histórico.

Total Width: Tamanho total da janela.

3.3.2 *StandardScaler*

A normalização tem como objetivo utilizar uma escala comum a todos os dados, para evitar (i) distorção da diferença entre eles, pois são valores cuja diferença entre menor e o maior valor é alta, e (ii) erros ao combiná-los.

O *StandardScaler*, como já apresentado anteriormente, padroniza os dados, utilizando a média e o desvio padrão. Supondo uma amostra dos dados x , o valor normalizado é calculado conforme a [Equação 3.1](#), também conhecida como *z-score*.

$$z = \frac{(x - u)}{s} \quad (3.1)$$

Sendo que z representa o valor normalizado, x o valor amostrado, u a média dos valores do conjunto e s o desvio padrão dos dados.

3.3.3 Variáveis utilizadas

As variáveis do processo são cruciais para um bom entendimento do funcionamento da RNA, e quando as variáveis são escolhidas com atenção, ajudam na manutenção e entendimento do código. A seguir estão listadas as principais variáveis utilizadas no processo:

df: Todos os dados (em MWh) da carga elétrica em todos os meses, a partir de 2004 até o mês 5 (maio) do ano de 2020.

dataset_train: Parte da base de dados separada para ser utilizada como entrada. Separada na forma de uma tupla de listas, cujo tamanho é de (197, 1).

datelist_train: Lista das datas, contendo *strings* no formato 'aaaa-mm'.

X_train: variável que recebe os dados já pre-processados e normalizados pelo *StandardScaler*, prontos para utilização da RNA.

y_train: variável que irá receber os valores das saídas da RNA, valores estes ainda normalizados.

history: Armazena as informações de execução e acurácia da RNA.

error_final: Armazena todos os valores do erro durante o treinamento da RNA.

y_pred_future: variável que irá de fato armazenar o valor da previsão da RNA.

Durante a manipulação de dados e processamento da rede, foram criadas cópias idênticas das variáveis anteriores, para manter a base de dados e as variáveis de processamento inalteradas, sendo uma escolha não obrigatória para a realização trabalho.

3.3.4 Estrutura do modelo da RNA

A rede neural criada utiliza o *TensorFlow* (ABADI et al., 2015) como o principal *framework*, e com a interface da API (*Application Programming Interface*) do *Keras*³. Ela foi construída seguindo o modelo *Sequential*⁴ (Sequencial), e estruturada da seguinte maneira:

Camada de entrada: Camada de Convolução de uma dimensão, que aplica a função de ativação *ReLU* e utiliza 32 *kernels*, vetores de 3 colunas e 1 linha, já que se trata de um conjunto de dados unidimensional, não representando uma imagem ou vídeo.

Camada de pooling: Camada responsável por diminuir progressivamente a dimensionalidade dos dados, e, assim, reduzir a quantidade de parâmetros a serem processados, ou seja, reduz a possibilidade de *overfitting*. Esse processo pode ser observado mais facilmente quando são utilizados imagens e vídeos, não sendo necessária nessa aplicação.

Camada de Dropout: Camada responsável por ignorar aleatoriamente parte dos neurônios, a fim de evitar o *overfitting* da rede.

Camadas Bidirecionais: Camadas que realizam a leitura dos dados nos dois sentidos, ou seja, uma das camadas utiliza o início dos dados para o final (*Forward*) e a outra utiliza-os do final para o início (*Backward*), com 32 neurônios em cada, o que permite que a camada de saída utilize valores futuros e passados simultaneamente.

Camada de Saída: Camada com 1 neurônio, aplicada a função de ativação *Linear*, cujo objetivo é apenas informar o valor previsto.

Existe a possibilidade de utilizar uma camada do tipo *Lambda*, que, assim como na lógica da linguagem *Python*, têm o objetivo de permitir ao programador incluir sua própria função ou operação no processamento da rede neural, a fim de realizar uma operação nos dados ou nas estruturas da rede que julgue necessária para o desenvolvimento do trabalho. Porém, é recomendado utilizar essa camada somente em experimentos e operações simples.

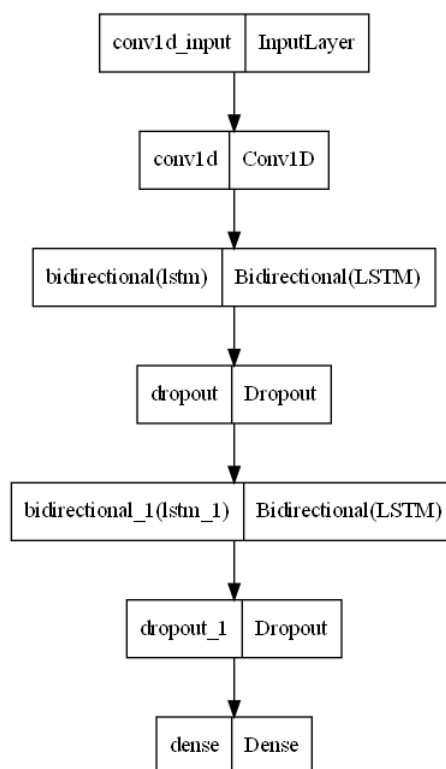
Em resumo, a estrutura da rede está representada na [Figura 18](#) e a quantidade de parâmetros treinados estão na [Tabela 1](#), cujos valores foram dados a partir da função *summary()* do modelo Sequencial.

Essa é uma estrutura simples para esse tipo rede, no entanto, é possível acrescentar mais camadas de convolução e mais camadas ocultas. A quantidade de camadas a serem usadas pode

³ Disponível em <https://keras.io/>

⁴ Disponível em <https://keras.io/api/models/sequential/>

Figura 18 – Estrutura da Rede Neural



Fonte: Elaborado pelo autor (2023)

Tabela 1 – Resumo da estrutura da Rede Neural

<i>Layer</i>	<i>Output Shape</i>	<i>Parameters</i>
Conv1D	(None, None, 32)	128
MaxPooling1D	(None, None, 32)	0
Bidirectional	(None, None, 64)	16640
Bidirectional	(None, 64)	24832
Dense	(None, 1)	65
<i>Total Parameters: 41665</i>		
<i>Trainable Parameters: 41665</i>		
<i>Non-trainable Parameters: 0</i>		

ser alterada conforme os resultados obtidos. Na etapa de construção da RNA, foram feitos vários ajustes dos parâmetros, como funções de ativação, quantidade de neurônios, tamanho do *kernel*, função de cálculo do erro, dentre outros, a fim de diminuir o erro das previsões.

3.3.4.1 Parâmetros de treinamento

Alguns dos parâmetros citados anteriormente, e que devem ser observados, são o otimizador, as métricas, número de épocas de treinamento e como o erro vai ser tratado durante a execução do treinamento. Esses parâmetros estão dispostos na [Tabela 2](#) com seus respectivos valores.

Tabela 2 – Parâmetros de treinamento da RNA

<i>Parameter</i>	<i>Value</i>
<i>Optimizer</i>	Adam
<i>Metrics</i>	RMSE
<i>Error</i>	MSE
<i>Epochs</i>	208

3.3.4.2 Otimizador - Adam

O otimizador *Adam*⁵ é um método estocástico de descida do gradiente, detalhada no item 3.3.5, que se fundamenta na estimativa de momentos de primeira e segunda ordem, que basicamente são a tendência (*math expectation*) e a dispersão (*dispersion* ou *variation*) dos dados. De acordo com Kingma e Ba (2014), é um algoritmo simples, computacionalmente eficiente, apresenta pouca demanda de memória, e é otimizado para problemas de aprendizado de máquina que utilizam grandes bases de dados e parâmetros.

3.3.4.3 Erro - MSE

O erro foi calculado pela metodologia do *Mean Squared Error* (*MSE*, ou Erro Quadrático Médio), que mostra o quão perto uma regressão linear está de um conjunto de dados. O cálculo é feito a partir da distância entre o dado real e o calculado (erro), esse valor então é elevado ao quadrado, a fim de remover qualquer valor negativo, além de dar mais peso a erros maiores. Por fim, é chamado de Médio pois utiliza-se a média do conjunto de erro. Quanto menor o MSE, melhor é a previsão.

3.3.4.4 Épocas

Para a definição do valor de 500 épocas, foram feitos testes com 200, 300, 400, 500 e 600 épocas, e para evitar eventuais treinamentos muito longos, 500 épocas foram suficientes para atingir um resultado satisfatório. Foi aplicada uma função do *Keras*, chamada de *EarlyStopping*⁶, que interrompe o treinamento da rede quando não há alteração significativa de um parâmetro, nesse caso o erro, dentro de um determinado número de épocas (parâmetro *patience* da função). O valor obtido pelo *EarlyStopping* foi de 208 épocas, sendo apenas informativo.

3.3.4.5 Métrica - RMSE

As métricas são funções que calculam o desempenho da rede neural, ou seja, o quão próximo do esperado essa rede desempenhou. São calculadas durante o processo de *compile* (compilação) da RNA, cujas funções são semelhantes às utilizadas no cálculo do erro.

⁵ <https://keras.io/api/optimizers/adam/>

⁶ Disponível em https://keras.io/api/callbacks/early_stopping/

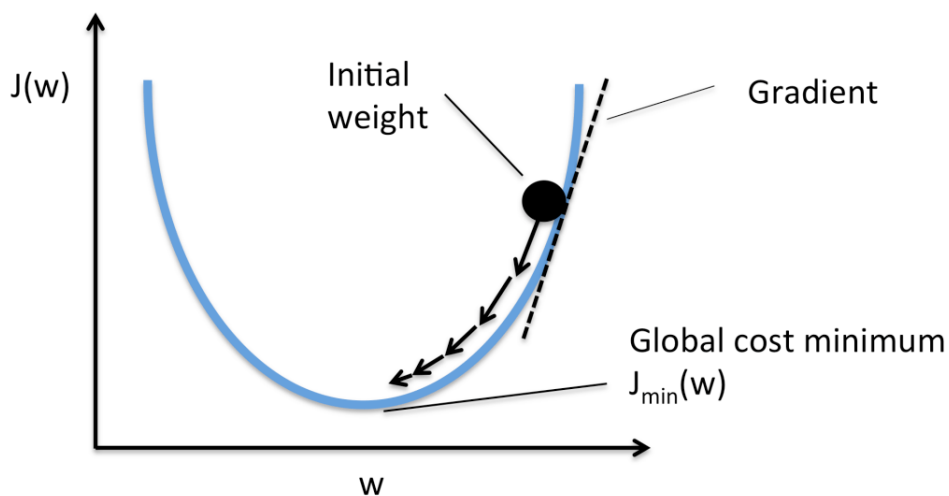
A métrica utilizada foi o *RMSE*, que calcula a raiz quadrada do quadrado da diferença entre os valores reais e os previstos.

3.3.5 Cálculo do erro - Descida do Gradiente

Imagine-se soltando uma massa (círculo preto) em uma rampa cuja forma se parece com a letra "U", conforme a [Figura 19](#), onde w são os valores dos pesos e $J(w)$ é o erro associado a um peso. À medida que essa massa se movimenta, sai de uma posição alta, passa por uma posição baixa, volta para uma posição alta, depois para uma baixa, e continua assim até perder toda energia e chegar no ponto J_{min} , chamado mínimo local.

Este método é aplicado internamente ao cálculo do erro pelo MSE, e se chama Descida do Gradiente (*Gradient Descending*), que basicamente inicia os pesos e *bias* da rede em valores muito altos, gerando erros maiores, e que vão reduzindo à medida do tempo, mas intercalando entre valores de erro altos e baixos. Espera-se que esses valores atinjam um valor mínimo local, que pode ser o valor mínimo global, sendo assim o valor ótimo para esses parâmetros.

Figura 19 – Descida do gradiente



Disponível em: [A Blog on Science](#)

4 RESULTADOS

A escolha da utilização de redes neurais para esse trabalho, como já citado em capítulos anteriores, foi pela capacidade de resolver problemas complexos com estruturas utilizadas em outras aplicações, e ainda haver muito espaço para novos caminhos e metodologias.

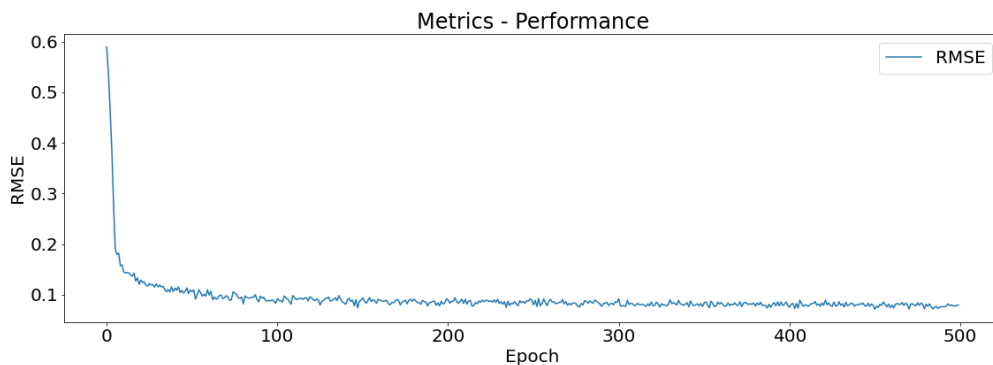
Depois de testes e ajustes, realizando o treinamento e teste com 500 épocas, para uma janela de 24 meses para a previsão do 25º, o resultado obtido na [Figura 20](#) apresenta o erro quando calculado pela função MSE, e a [Figura 21](#) mostra a métrica, ou seja, o desempenho da rede.

Figura 20 – Erro da RNA em função das épocas



Fonte: Elaborado pelo autor (2023)

Figura 21 – Desempenho da RNA

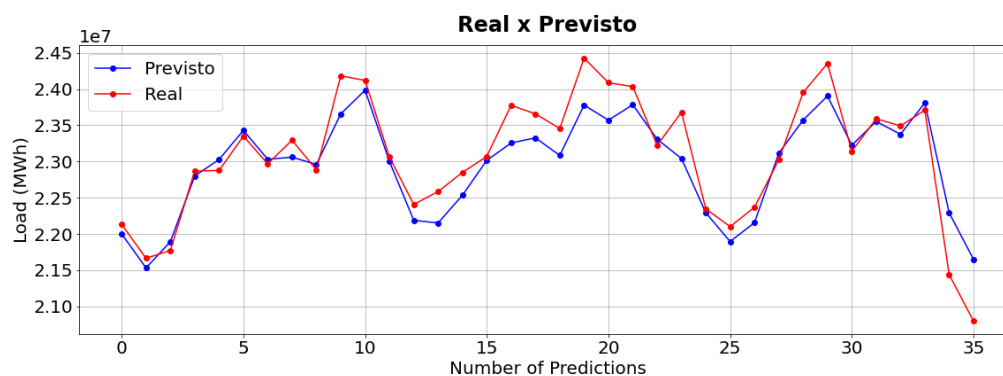


Fonte: Elaborado pelo autor (2023)

A [Figura 20](#) e a [Figura 21](#) mostram que 50 épocas, aproximadamente, já seriam suficientes, mas de acordo com os parâmetros informados para o método do *EarlyStopping*, o resultado ainda poderia ser otimizado.

Nota-se a eficiência da rede neural em solucionar esse tipo de problema, apresentado no resultado *Real vs. Previsto* na Figura 22 e na avaliação da RNA, a partir do método *evaluate* do próprio modelo, na Tabela 3, visto que os valores previstos seguem a tendência e a amplitude dos valores reais.

Figura 22 – Resultado: Real vs. Previsto



Fonte: Elaborado pelo autor (2023)

Para fins de avaliação, quanto menor o *RMSE*, melhor foi o desempenho da RNA (REZENDE, 2018).

Tabela 3 – Avaliação do modelo. Método *evaluate()*

<i>Loss</i>	<i>RMSE</i>
0.04211492836475372	0.20521922409534454

As redes neurais convolucionais unidimensionais são mais comumente utilizadas em sinais de áudio, e mostrou-se eficiente em problemas envolvendo séries temporais.

Nesse caso, mais especificamente citando a motivação pela realização desse estudo, a previsão auxiliaria na programação de manutenções preventivas e corretivas com mais eficácia, sendo possível a realocação de carga, ou seja, consumidores, entre outras linhas de transmissão e demais equipamentos do SIN, diminuindo o prejuízo financeiro tanto de concessionárias quanto de pessoas físicas ou jurídicas que dependem diretamente do fornecimento de energia para realizar suas atividades.

Para que a previsão da carga elétrica seja possível, as etapas de coleta e pré-processamento de dados, criação da rede neural convolucional, treinamento, ajustes de parâmetros da rede, devem ser realizadas com muita atenção, cuidado e sensibilidade. Qualquer tratamento indevido dos dados, ou parâmetros ajustados erroneamente, pode levar ao *overfitting* ou *underfitting*, que proporciona menor desempenho e acurácia, sendo necessário reavaliar dados e parâmetros.

5 CONCLUSÃO

Redes Neurais, sejam elas convolucionais, recorrentes, LSTM, *perceptron* ou outro tipo, são ferramentas com potencial imensurável, cujos principais gargalos estão na obtenção e organização dos dados, *tunning* de parâmetros, hardware disponível para a implementação, e capacidade do programador em identificar e tratar os eventuais erros.

As CNN são redes neurais eficientes para muitas aplicações, principalmente se o problema envolver imagens e vídeos. Apesar disso, a utilização de CNN em dados unidimensionais é pertinente, como para a previsão de carga elétrica, sendo essa apresentada como uma série temporal.

Nesse trabalho, os parâmetros utilizados para treinamento e estrutura da rede são os principais pontos a serem avaliados. Realizar mais testes com diferentes funções de ativação, inicialização de pesos, função de erro ou número de neurônios podem afetar diretamente o resultado obtido, como foi mostrado anteriormente.

Existem maneiras mais eficientes de avaliar o comportamento da RNA com diferentes parâmetros, sem a necessidade de alterá-los manualmente. Uma dessas maneiras é utilizar a técnica do *Grid Search* (LAVALLE; BRANICKY; LINDEMANN, 2004), que faz testes na RNA utilizando um *grid* de parâmetros previamente definidos.

No entanto, mesmo sem utilizar parâmetros ótimos, a rede neural convolucional construída nesse trabalho apresentou resultados satisfatórios, podendo ser aplicada em situações em que os dados se apresentam como séries temporais.

É importante dizer que a linguagem *Python* simplifica o desenvolvimento desse tipo de aplicação com redes neurais, como no caso das convoluções mandatórias para as CNN serem aplicadas diretamente na camada de convolução, sem a necessidade de realizar cálculos complexos.

Futuramente, pode-se realizar um estudo comparativo entre a eficiência de Redes Neurais Convolucionais e, por exemplo, Redes Neurais Recorrentes (*RNN - Recurrent Neural Network*) ou outro tipo de RNA. Também é possível reduzir o espaço de amostragem, para estudar a variação de carga a cada dia, e verificar qual dia com menor carga para facilitar a programação de manutenções nos equipamentos do sistema.

REFERÊNCIAS

- ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: <<http://tensorflow.org/>>. Citado na página 31.
- ATWAN, T. *Time Series Analysis with Python Cookbook: Practical Recipes for Exploratory Data Analysis, Data Preparation, Forecasting, and Model Evaluation*. Packt Publishing, Limited, 2022. ISBN 9781801075541. Disponível em: <<https://books.google.com.br/books?id=yVU7zwEACAAJ>>. Citado na página 22.
- AZURE, M. *Componente Normalizar Dados*. 2022. Url<https://docs.microsoft.com/pt-br/azure/machine-learning/component-reference/normalize-data>. Citado na página 27.
- BOWERMAN, B. L.; O'CONNELL, R. T. *Time series and forecasting*. [S.l.]: Duxbury Press North Scituate, MA, 1979. Citado na página 22.
- BOX, G. E. et al. *Time series analysis: forecasting and control*. [S.l.]: John Wiley & Sons, 2015. Citado 2 vezes nas páginas 15 e 22.
- EPE. *Consumo Mensal de Energia Elétrica por Classe (regiões e subsistemas)*. 2022. Url<https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/consumo-de-energia-eletrica>. Citado na página 27.
- FREITAS, G. de; SILVEIRA, S. d. F. R. Programa luz para todos: uma representação da teoria do programa por meio do modelo lógico. *Planejamento e Políticas Públicas*, 2015. n. 45, 2015. Citado na página 14.
- GONÇALVES, G. S.; SORTE, L. X. B. Análise de métricas para otimização de topologias da rede neural artificial–perceptron de múltiplas camadas–aplicada a classificação de padrões. 2014. 2014. Citado na página 18.
- GOODFELLOW, I. et al. *Deep learning*. [S.l.]: MIT press Cambridge, 2016. Citado na página 19.
- GRAVES, A.; FERNÁNDEZ, S.; SCHMIDHUBER, J. Bidirectional lstm networks for improved phoneme classification and recognition. In: SPRINGER. *International conference on artificial neural networks*. [S.l.], 2005. p. 799–804. Citado na página 22.
- HOSAKI, G. Y. G. Y.; RIBEIRO, D. F. Deep learning: ensinando a aprender. 2021. 275, 2021. Citado na página 23.
- JANOS, N.; ROACH, J. *1D Convolutional Neural Networks for Time Series Modeling*. 2018. Disponível em: <<https://pydata.org/la2018/schedule/presentation/14/>>. Citado na página 28.
- JUNIOR, L. M. C. et al. Previsão do consumo de eletricidade no nordeste brasileiro. *Engevista*, 2018. v. 20, n. 3, p. 408–423, 2018. Citado na página 16.
- KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2014. Citado na página 33.

- KOVÁCS, Z. L. *Redes neurais artificiais*. [S.l.]: Editora Livraria da Física, 2002. Citado na página 14.
- LAVALLE, S. M.; BRANICKY, M. S.; LINDEMANN, S. R. On the relationship between classical grid search and probabilistic roadmaps. *The International Journal of Robotics Research*, 2004. SAGE Publications, v. 23, n. 7-8, p. 673–692, 2004. Citado na página 37.
- LOPES, M. L. M. Desenvolvimento de redes neurais para previsão de cargas elétricas de sistemas de energia elétrica. 2005. Universidade Estadual Paulista (UNESP), 2005. Citado na página 14.
- MARUMO, F. S. *Deep Learning para classificação de Fake News por sumarização de texto*. 2018. Citado na página 23.
- Park, D. C. et al. Electric load forecasting using an artificial neural network. *IEEE Transactions on Power Systems*, 1991. v. 6, n. 2, p. 442–449, 1991. Citado 2 vezes nas páginas 14 e 23.
- RAUBER, T. W. *Redes neurais artificiais*. Universidade Federal do Espírito Santo, 2005. p. 29, 2005. Citado na página 19.
- REZENDE, T. *RMSE ou MAE? Como avaliar meu modelo de machine learning?* 2018. Disponível em: <<https://www.linkedin.com/pulse/rmse-ou-mae-como-avaliar-meu-modelo-de-machine-learning-rezende/>>. Citado na página 36.
- RODRIGUES, J. d. S. S. R. Aplicação de redes neurais convolucionais e processamento digital de imagens para classificação do estado dos olhos e avaliação de sonolência. 2020. 2020. Citado na página 21.
- SOUZA, R. Metodologias para a análise e previsão de séries temporais univariadas e multivariadas. *Brazilian Review of Econometrics*, 1981. v. 1, n. 2, p. 78–105, 1981. Citado na página 15.
- SOVIERZOSKI, M. A. Convolução de sinais: Definição, propriedades e ferramentas. *Revista Ilha Digital*, 2011. v. 2, p. 81–95, 2011. Citado na página 20.
- SZEGEDY, C. et al. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 1–9. Citado na página 21.
- VARGAS, A. C. G.; PAES, A.; VASCONCELOS, C. N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: *Proceedings of the xxix conference on graphics, patterns and images*. [S.l.: s.n.], 2016. v. 1, n. 4. Citado na página 21.
- WICHMANN, P. et al. Extracting supply chain maps from news articles using deep neural networks. *International Journal of Production Research*, 2020. Taylor & Francis, v. 58, n. 17, p. 5320–5336, 2020. Disponível em: <<https://doi.org/10.1080/00207543.2020.1720925>>. Citado na página 15.
- ZHANG, G.; PATUWO, B. E.; HU, M. Y. Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, 1998. Elsevier, v. 14, n. 1, p. 35–62, 1998. Citado na página 23.