

Previsão de Carga Elétrica para o Horizonte de Curto Prazo via Redes Neurais Artificiais Utilizando C++ e Python

Maick Eugênio* Fernando A. Assis*
Luiz Carlos Nascimento*

* *Universidade Federal de São João del-Rei (UFSJ), MG, (e-mails: maick2803@hotmail.com, fernandoassis@ufs.edu.br, nascimentolc@ufs.edu.br).*

Abstract: The technological and social advances currently observed around the world imply the demand for ever better levels of efficiency and quality in the supply of electricity. In order to ensure the safe and economical operation of power systems, the future prediction study of the state of the load plays a fundamental role. With this motivation, in this work is applied a widespread method in the literature for daily load forecasting in the short term horizon, the Multilayer Perceptron method (MLP), via Backpropagation Algorithm (BP) for training. The implementation of the forecasting technique is performed for performance evaluation of two different programming languages, C++ and Python. Tool validation and analysis of the results are based on the application of three different databases, one of which is obtained from current information provided by the Brazilian independent system operator (Operator of the National Electricity System - ONS).

Resumo: Os avanços tecnológico e social observados atualmente em todo o mundo implicam na exigência de níveis cada vez melhores de eficiência e de qualidade no fornecimento da energia elétrica. Neste sentido, para garantir a operação segura e econômica dos sistemas elétricos de potência, o estudo de previsão futura do estado das cargas elétricas desempenha um papel de fundamental importância. Com essa motivação, nesse trabalho é aplicado um método difundido na literatura para previsão diária de cargas elétricas no horizonte de curto prazo, o método *Perceptron* Multicamadas (MLP), via Algoritmo de Retropropagação (BP) para treinamento. A implementação da técnica de previsão é realizada para avaliação do desempenho de duas diferentes linguagens de programação, C++ e *Python*. A validação da ferramenta e a análise dos resultados se baseiam na aplicação de três diferentes bases de dados, sendo um delas obtida a partir de informações atuais disponibilizadas pelo Operador Nacional do Sistema (ONS).

Keywords: Load Forecasting; Artificial Neural Networks; Multilayer Perceptron; Backpropagation; C++; Python; Electric Power System.

Palavras-chaves: Previsão de Carga; Redes Neurais Artificiais; *Multilayer Perceptron*; *Backpropagation*; C++; *Python*; Sistema Elétrico de Potência.

1. INTRODUÇÃO

Juntamente com os avanços tecnológico e social que vêm ocorrendo mundialmente nos últimos anos, busca-se cada vez mais por eficiência e qualidade no fornecimento de energia elétrica. Para que isso seja possível, e para que os sistemas elétricos de potência (SEPs) possam manter sua operação segura e econômica, é imprescindível o conhecimento futuro do comportamento das cargas elétricas, o que torna possível a realização de diversos estudos inerentes a estes sistemas. Estudos estes que vão desde o planejamento da expansão até a programação da geração e a tomada de decisões durante a operação em tempo real.

De forma geral, o estudo de previsão de carga em sistemas elétricos tem por objetivo estimar valores e comportamentos de cargas para desde minutos até anos a frente, podendo, dessa forma, ser tratado em diferentes

horizontes. Estes horizontes são divididos, basicamente, em curtíssimo, curto, médio e longo prazo (Zor et al. (2017), Kuster et al. (2017), Hammad et al. (2020)). Cada um desses intervalos abrange áreas distintas de estudo e exige a observância de fatores específicos de influência do comportamento da carga (Fahad and Arbab (2014)). Como fatores de influência podem ser citados os fatores climáticos (e.g., temperatura, velocidade do vento e precipitação), os temporais (e.g., dias da semana, feriados e estações do ano), os econômicos e demográficos (e.g., produto interno bruto e crescimento populacional) e os fatores aleatórios relacionados, por exemplo, à eventos de grande repercussão (e.g., eventos esportivos e programas de televisão de amplo interesse).

Devido às características diversas inerentes aos diferentes SEPs e aos horizontes de análise, os estudos de previsão de cargas elétricas podem ser tratados por diferentes

abordagens e empregar diferentes técnicas de solução. São inúmeros os métodos para previsão que se baseiam nas informações do comportamento da carga ocorrido no passado e na observância de fatores indicativos (Singh et al. (2012)). Basicamente, essas técnicas podem ser agrupadas em métodos estatísticos, tais como regressão linear múltipla e séries temporais estocásticas (Huang and Shih (2003)), ou métodos de inteligência artificial, tais como redes neurais artificiais e algoritmos genéticos (Zor et al. (2017)). Pode-se, inclusive, combinar dois métodos diferentes, formando um método híbrido (Lu et al. (2004)). Até o momento, pode-se dizer que não há um método definido ou algoritmo que seja considerado superior para ser utilizado em qualquer estudo de previsão de carga.

Neste sentido, realiza-se no presente trabalho um estudo de previsão de carga de curto prazo (24 horas à frente) a fim de investigar o desempenho computacional de duas diferentes linguagens de programação, C++ e *Python*. Com base nas classificações e distinções discutidas sobre o tema, um método não-linear baseado em Redes Neurais Artificiais (RNAs) é escolhido para realização do estudo. A arquitetura da RNA utilizada é a *Perceptron Multicamadas (Multilayer Perceptron - MLP)*, com o Algoritmo de Retropropagação (*Backpropagation - BP*) para treinamento da rede (Haykin (1999)). A fim de atingir o objetivo proposto, no emprego da técnica de previsão são realizadas duas análises distintas: inicialmente, para validação da ferramenta empregada, é realizada uma análise a respeito do desempenho de previsão obtido (verificado por meio de erros de previsão); em seguida, é verificado o desempenho de processamento computacional dos programas desenvolvidos em C++ e *Python*. Neste ponto, cabe destacar que os programas foram implementados de forma idêntica para as duas linguagens. Três diferentes bases de dados, sendo uma delas obtida a partir de informações atuais disponibilizadas pelo Operador Nacional do Sistema (ONS), são utilizadas para realização do estudo.

2. METODOLOGIA PROPOSTA PARA PREVISÃO DE CARGA

De forma geral, neste trabalho é utilizado o método *Perceptron Multicamadas (MLP)*, de RNAs para realização do estudo de previsão de cargas elétricas no curto prazo. A escolha deste método se deve à sua flexibilidade e popularidade de aplicação para estudos de previsão (Hammad et al., 2020). O Algoritmo *Backpropagation (BP)* é empregado para o treinamento da rede. Inferências sobre a qualidade do método são realizadas por meio de índices de desempenho, relacionados aos erros de previsão (Média Percentual Absoluta do Erro, ou *Mean Absolute Percentage Error (MAPE)*, e o Erro Máximo (EM) (Srinivasan et al. (1998))).

Nesta seção é apresentada a metodologia utilizada no presente trabalho, exibindo, primeiramente, a construção básica da rede empregada (MLP) e do algoritmo de treinamento (BP). Em seguida, é descrito o desenvolvimento computacional empregado.

2.1 Método Perceptron Multicamadas (MLP)

De forma básica, as RNAs modelam os neurônios biológicos encontrados em cérebros humanos. Com base no

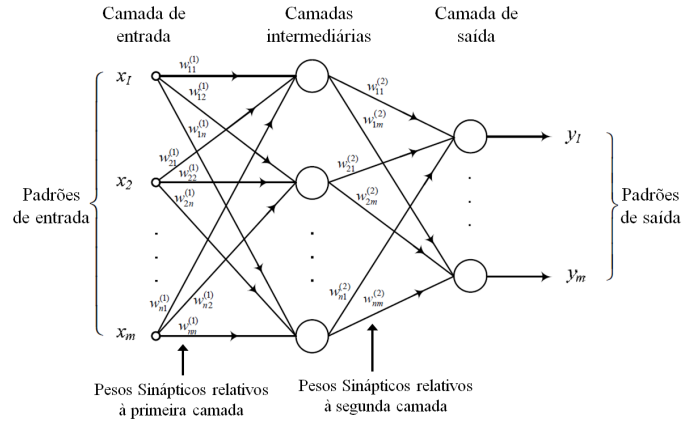


Figura 1. *Perceptron Multicamadas - MLP* (Adaptado de Altran (2010)).

neurônio biológico, pode-se modelar um neurônio artificial dividindo-o, basicamente, em três partes (Haykin (1999)):

- (1) **Elos de entrada** (x_m): responsável pela entrada do sinal no neurônio;
- (2) **Somador**: responsável por somar os sinais de entrada já multiplicados pelos pesos sinápticos;
- (3) **Função de ativação**: responsável por restringir a amplitude da saída de um neurônio, com restrição a um intervalo unitário fechado, $[0,1]$ ou $[-1,1]$, caso esta função seja não-linear. Pode ser do tipo: relé, sigmoide, gaussiana, etc.

A MLP é um tipo de rede neural e pode ser modelada de acordo com a Figura 1, apresentando três tipos básicos de camadas: a de entrada, as intermediárias, também chamadas de camadas ocultas, e a de saída. Cada um dos três tipos de camadas tem sua função específica (Haykin (1999), Altran (2010)). A camada de saída, com base nos dados recebidos na camada de entrada, produz o padrão da resposta da rede. As camadas ocultas são responsáveis por extrair as características do sistema, de acordo com os dados de entrada. Caso existam conexões certas entre as unidades de neurônios da camada de entrada, e uma quantidade suficiente de neurônios nas camadas intermediárias, é possível encontrar o mapeamento correto da entrada para a saída.

Basicamente, um neurônio artificial k pode ser matematicamente modelado como (Haykin (1999)):

$$v_k = \sum_{j=0}^m w_{kj} x_j, \quad (1)$$

em que v_k representa o campo induzido (sinal presente na saída do somador) do neurônio k , w_{kj} é o peso sináptico do neurônio k , e x_j é o sinal de entrada. Neste modelo, a consideração de um *bias* é feita para quando $j = 0$.

Por sua vez, a saída y_k do neurônio k é dada por:

$$y_k = \varphi(v_k), \quad (2)$$

em que $\varphi(\cdot)$ é sua função de ativação.

2.2 Algoritmo de Retropropagação (BP)

O treinamento da rede pelo algoritmo de Retropropagação pode ser realizado seguindo os procedimentos a seguir (Altran (2010)):

- (1) Apresentação de um padrão de entrada X à rede, o qual fornece uma saída Y ;
- (2) Cálculo do erro (diferença entre o valor desejado e a saída obtida) para cada saída;
- (3) Determinação do erro retropropagado pela rede, o qual está associado à derivada parcial do erro quadrático de cada elemento relacionado aos pesos;
- (4) Ajuste dos pesos de cada elemento;
- (5) Finalmente, apresenta-se um novo padrão à rede e repete-se o processo até que haja a convergência da rede, ou seja, o erro seja menor do que a tolerância estabelecida.

No início do processo de treinamento, pode-se adotar pequenos valores aleatórios para os pesos sinápticos iniciais, considerando o intervalo $[-1,1]$. Após o treinamento, seguindo os passos descritos, a rede poderá operar como previsora de cargas elétricas. Nesta fase, ela operará apenas no modo progressivo (*feedforward*). Ou seja, um vetor de entrada é apresentado na sua camada de entrada, os dados então são processados nas camadas ocultas e, finalmente, os resultados serão apresentados na camada de saída, sem a presença da realimentação.

O treinamento do método MLP via Algoritmo BP baseia-se na minimização do erro. Este erro, na saída do neurônio k , pode ser dado por:

$$\epsilon_k = d_k - y_k, \quad (3)$$

sendo ϵ_k o sinal de erro, d_k o valor desejado na saída e y_k o valor obtido na saída (Haykin (1999)).

Como o interesse está na soma dos erros quadráticos (ϵ^2), pode-se expressar esta equação por:

$$\epsilon^2 = \sum_{k=1}^{ns} \epsilon_k^2, \quad (4)$$

em que ns é o número de neurônios da última camada da rede (Widrow and Lehr (1990)).

A adaptação dos pesos sinápticos pode então ser realizada pelo método do gradiente descendente (Widrow and Lehr (1990)). Para tanto, expressa-se estes ajustes como (Altran (2010)):

$$V_i(h+1) = V_i(h) - \theta_i(h), \quad (5)$$

sendo: h o índice da iteração; $V_i(h)$ o vetor atual de pesos do neurônio i , formado por $[w_{0i}, w_{1i}, w_{2i}, \dots, w_{ni}]^T$; $V_i(h+1)$ o vetor de pesos do neurônio i após o ajuste de pesos; $\theta_i(h) = \gamma [\nabla_i(h)]$; γ o parâmetro de controle da estabilidade, ou taxa de treinamento (determina o comprimento de θ); e $\nabla_i(h)$ é o gradiente do erro quadrático relacionado aos pesos do neurônio i .

O sinal negativo em (5) indica a direção inversa ao gradiente, justificando o nome gradiente descendente. Essa equação pode ainda ser reescrita como:

$$V_i(h+1) = V_i(h) + 2\gamma\beta_i X_i, \quad (6)$$

sendo X_i o vetor de entrada. Neste caso, $\beta_i = \sigma_i \epsilon_i$, caso o neurônio esteja na última camada, ou $\beta_i = \sigma_i \sum_{j \in Q(i)} w_{ij} \beta_j$, caso o neurônio esteja nas camadas anteriores à última. Por fim, σ_i é a derivada da função de ativação e $Q(i)$ contém os índices da camada seguinte.

Dependendo do valor da taxa de treinamento γ , a velocidade com que se chega à solução se altera. Pode-se mover mais lentamente em direção à solução, caso γ seja pequeno,

ou pode-se ocasionar uma instabilidade no sistema, caso γ seja demasiadamente grande. Em Widrow and Lehr (1990) foi proposta uma simples forma de garantir a estabilidade do sistema ao aplicar o fator de momento, tal estratégia foi implementada no presente trabalho. Em relação a estes estados, três caminhos diferentes podem ser tomados em direção à solução, sendo: a) taxa de treinamento pequena, onde o sistema será lento e demorará a chegar em um resultado; b) taxa de treinamento grande, onde o sistema será rápido, mas poderá perder sua estabilidade e c) taxa de treinamento grande, porém com a aplicação do fator de momento, onde ele será rápido e estável. Para modelar este comportamento, a adição do fator de momento à relação (6) resulta em:

$$V_{ij}(h+1) = V_{ij}(h) + \Delta V_{ij}(h), \quad (7)$$

sendo $\Delta V_{ij}(h) = 2\gamma(1-\eta)\beta_j X_i + \eta\Delta V_{ij}(h-1)$ e η a constante de momento, para $0 \leq \eta \leq 1$.

A ativação do neurônio é feita por uma função de ativação (e.g., relé, linear, sigmoide e gaussiana). Estas funções devem ser diferenciáveis, pois o algoritmo utiliza sua derivada, e devem ser limitadas, pois elas têm por objetivo restringir a amplitude do sinal de saída do somador do neurônio (Widrow and Lehr (1990); Hippert et al. (2001)). No presente trabalho é utilizada a função de ativação sigmoide, sendo esta a função mais comumente aplicada no emprego de redes neurais artificiais.

2.3 Representação dos Dados de Entrada da RNA

Dois tipos de dados de entrada da rede MLP são considerados para realização do estudo de previsão de cargas neste trabalho: i) dados exógenos, referentes ao dia da semana e à hora que se pretende realizar a previsão; e ii) dados correspondentes à série histórica de carga elétrica.

Basicamente, a representação binária dos dados de tempo (tanto de dia quanto de hora) é realizada utilizando a notação -1 e $+1$. A adoção de tal representação busca garantir uma convergência mais rápida para o processo de treinamento (Altran (2010)). Já os dados de carga são normalizados de acordo com um valor base definido (e.g., o valor de carga pico observado na série). Portanto, para obtenção da carga prevista pela rede em MW, é preciso multiplicar o valor de saída verificado na rede pelo valor base de carga considerado.

O vetor X de entrada considerado neste trabalho possui, portanto, dimensão 13 (i.e., a entrada é um vetor com 13 posições), sendo preenchido de acordo com as seguintes informações: dia da semana (3 posições do vetor); é ou não final de semana (1 posição no vetor); hora da carga a ser prevista (5 posições do vetor); carga da hora atual (1 posição do vetor); e cargas horárias passadas (cargas das 3 horas anteriores à atual, ocupando, portanto, as últimas 3 posições do vetor de entrada).

Para exemplificar o método de entrada de dados utilizado, considere uma situação hipotética em que se deseja prever a carga de uma quinta-feira, às 17h, onde a carga da hora atual seja de 60.000MW e as cargas das 3 horas anteriores sejam de 55.000MW, 50.000MW e 45.000MW. Considerando um valor base de 60.000MW para normalizar os dados de carga, o vetor convertido ao formato de entrada da rede é preenchido com os seguintes 13 elementos:

$$X(17h) = [+1; -1; -1; -1; +1; -1; -1; -1; +1; 1,00; 0,75; 0,82; 0,92] \quad (8)$$

sendo: $[+1; -1; -1]$ para indicar "Quinta-feira", $[-1]$ para indicar que não é final de semana, $[+1; -1; -1; -1; +1]$ para indicar 17h, $[1,00]$ para 60.000MW normalizado de carga da hora atual, $[0,75; 0,82; 0,92]$ para indicar os valores normalizados de carga para 45.000MW, 50.000 MW e 55.000 MW.

Nesta situação hipotética, se a saída da rede é dada por $Y(17h) = 1,08$, a carga prevista para as 17h é, portanto, de 64.800 MW, dado o valor base definido (60.000 MW).

2.4 Avaliação do Desempenho de Previsão

A análise de erros de previsão permite indicar se o método é viável ou não para ser utilizado em previsões de cargas elétricas. Para análise do método empregado, utiliza-se neste trabalho os índices de desempenho de Média Percentual Absoluta do Erro (MAPE) e de Erro Máximo (EM) (Srinivasan et al. (1998)) em cada estudo de caso realizado. É importante destacar que ambas as técnicas utilizam os dados da carga real do sistema e da carga prevista pela rede. O cálculo destes índices é dado como:

$$MAPE_{(\%)} = \frac{1}{N} \left\{ \sum_{h=1}^N \frac{|L(h) - \underline{L}(h)|}{L(h)} \right\} * 100 \quad (9)$$

$$EM_{(\%)} = \max_{h \in H_d} \left\{ \frac{|L(h) - \underline{L}(h)|}{L(h)} \right\} * 100 \quad (10)$$

em que $L(h)$ representa a carga real referente à hora h ; $\underline{L}(h)$ é a carga prevista pelo sistema referente à hora h ; N é o número de horas da curva de carga; e H_d representa o intervalo de horas ($1 \leq h \leq 24$).

2.5 Avaliação do Desempenho Computacional

Para a solução do problema de previsão de cargas elétricas com o horizonte de curto prazo são implementados neste trabalho dois programas utilizando o método MLP com treinamento via BP, um desenvolvido utilizando a linguagem C++ e o outro a linguagem *Python*. A utilização de duas linguagens de programação se justifica na pretensão de se comparar, de forma simples, os tempos de execução de cada programa, permitindo que seja apontada qual linguagem apresenta menor tempo computacional no treinamento da rede. Neste ponto, cabe destacar que os dois programas são implementados de forma idêntica em relação às técnicas de previsão empregadas, tanto em relação às estruturas da rede neural, quanto em relação ao ajuste de parâmetros, permitindo, assim, que a comparação direta seja realizada. Inclusive, foram utilizados os mesmos pesos iniciais (gerados aleatoriamente) para os dois programas.

3. ESTUDOS DE CASO

Nesta seção são apresentados os resultados obtidos na previsão de cargas para três diferentes estudos (três bases de dados), seguindo a metodologia apresentada na seção anterior. Primeiramente, é realizada uma análise dos erros MAPE e EM obtidos na previsão de carga no curto prazo (análise do desempenho de previsão), a fim de validar a ferramenta empregada no estudo. Em seguida,

é feita uma comparação entre os tempos de treinamento da rede para as rotinas implementadas nas linguagens de programação C++ e *Python* (análise do desempenho computacional). Salienta-se que, na primeira etapa, de validação da ferramenta, espera-se resultados iguais para as duas linguagens de programação utilizadas, uma vez que a mesma metodologia é implementada nos dois casos.

3.1 Bases de Dados

Para efetuar os estudos com a metodologia proposta neste trabalho, foram utilizadas três diferentes bases de dados, aqui denominadas BD-I, BD-II e BD-III. A primeira (BD-I) foi obtida de Altran (2010). A BD-II foi extraída de Lopes (2005). Por fim, para compor a BD-III, foram utilizadas informações de carga horária obtidas diretamente do site do Operador Nacional do Sistema (ONS), órgão responsável pela operação do sistema brasileiro (ONS (2019)).

Os parâmetros necessários ao treinamento da rede e à previsão de cargas, listados na Tabela 1, foram ajustados empiricamente por meio de vários testes realizados para cada base de dados. Destaca-se, que alterações nesses valores podem acarretar em mudanças no desempenho dos programas. Contudo, não foi o objetivo deste trabalho encontrar o ajuste ótimo de tais valores, apenas garantir uma ferramenta com bom desempenho para avaliação do desempenho em termos das linguagens de programação. Destaca-se, ainda, que estes parâmetros são diferentes dos utilizados nas referências fontes das bases de dados e que cada entrada da rede é composta somente por nós, e não por neurônios.

Tabela 1. Parâmetros da rede.

Item	BD-I	BD-II	BD-III
Vetores Padrão	360	480	360
Camadas	3	3	3
Neurônios por Camada	13-28-1	13-30-1	13-30-1
Tolerância	$5,09 \times 10^{-4}$	$4,97 \times 10^{-4}$	$7,69 \times 10^{-4}$
γ de Treinamento	0,29500	0,30100	0,50000
Termo Momento (η)	0,00371	0,20200	0,85000
λ da Função Sigmoid	0,95005	0,99970	1,20000

3.2 Análise do Desempenho de Previsão

BD-I: Em Altran (2010), fonte dessa base de dados, foram realizados três diferentes estudos de previsão de carga, sendo que no presente trabalho é abordado somente o primeiro deles. No estudo considerado, para o treinamento da rede foram considerados dados de carga de 15 dias, compreendidos entre 15/10/2007 e 29/10/2007, resultando em 360 vetores padrão (24 vetores por dia). A carga do 16º dia, 30/10/2007, conhecida (Altran (2010)), é utilizada para verificação do desempenho final da previsão realizada pela RNA implementada.

A Figura 2 apresenta o resultado da previsão de carga do dia 30/10/2007 para esta base de dados, para cada uma das 24 horas do 16º dia de BD-I. Neste gráfico, a curva com linha contínua representa a carga real (Altran (2010)) e a curva de linha tracejada representa a carga obtida na saída da rede empregada (carga prevista). Por meio deste gráfico, verifica-se um bom resultado de previsão, comprovado pelos baixos níveis de erro MAPE e EM conforme Tabela 2 (resultados provenientes do programa

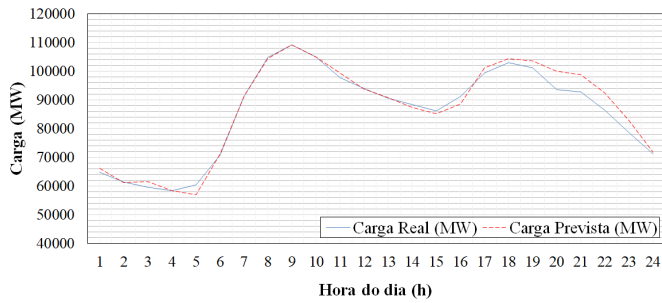


Figura 2. Curva de previsão para BD-I - 30/10/2007.

Tabela 2. Erros obtidos ao utilizar a BD-I.

Método	MAPE (%)	EM (%)
Presente trabalho - MLP via BP	2,07	7,07
Ref. - MLP	1,91	5,86

implementado em C++). De forma geral, menores erros de previsão são observados nas primeiras horas do dia. Este fato pode ser explicado pela propagação do erro ocorrida ao utilizar os próprios valores cargas previstas nas primeiras horas para estimar as cargas seguintes. Além disso, uma maior variabilidade da carga no final do dia (horário de ponta) pode prejudicar a atuação do previsor.

Na Tabela 2 é indicada a sigla do método utilizado no trabalho fonte de BD-I (MLP), para termos de comparação (maiores detalhes em Altran (2010)). Verifica-se que os erros verificados estão na mesma ordem do trabalho referência, validando a implementação realizada no presente trabalho. A diferença entre os índices MAPE dos dois resultados, por exemplo, foi de 0,16 pontos percentuais.

BD-II: Como segunda base de dados, é considerado um dos estudos de previsão de cargas realizado por Lopes (2005), onde foram considerados dados de carga de 21 dias para o treinamento de uma rede neural, compreendidos entre os dias 08/07/1998 e 28/07/1998, resultando em 504 vetores padrão. A carga do 22º dia, 29/07/1998, é então prevista. Porém, nessa base de dados não são apresentadas as informações de carga para o 22º dia, o que impossibilita o cálculo dos erros de previsão no presente trabalho. Sendo assim, definiu-se utilizar no estudo apenas os dados dos 20 primeiros dias (480 vetores padrão) e prever a carga do 21º dia (28/07/1998). Neste caso, não é possível realizar uma comparação direta dos erros de previsão obtidos no presente trabalho com aqueles apresentados pela referência fonte da base de dados. Contudo, apenas para termos de validação da ferramenta, é possível realizar uma análise em relação aos níveis de erro de previsão.

A Figura 3 apresenta o resultado da previsão de carga do dia 28/07/1998 (21º dia) para a BD-II, obtido no presente trabalho. No gráfico, a curva contínua representa a carga real e a curva tracejada representa a carga prevista pela rede. Os erros MAPE e EM obtidos via programa implementado em C++ ao realizar esta previsão podem ser vistos na Tabela 3. Nesta tabela, os erros calculados por cinco diferentes métodos aplicados em Lopes (2005), para previsão da carga do dia 29/07/1998 (22º dia), são também apresentados (mais detalhes podem ser encontrados em Lopes (2005)). Verifica-se valores reduzidos de erros, na mesma ordem daqueles encontrados pela referência, o

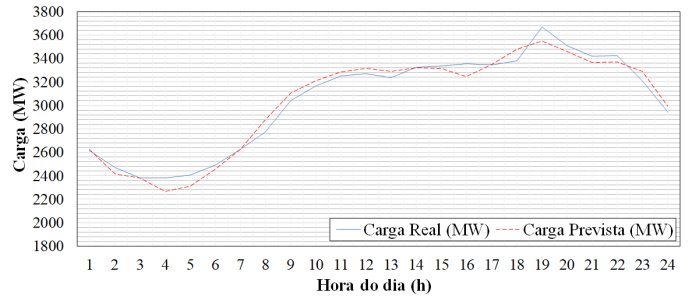


Figura 3. Curva de previsão para BD-II - 29/07/1998.

Tabela 3. Erros obtidos ao utilizar a BD-II.

Método	MAPE (%)	EM (%)
Presente trabalho - MLP via BP	1,73	4,74
Ref. - RP Convencional (M)	2,03	5,62
Ref. - RP Convencional (SM)	1,80	6,20
Ref. - RP Nebuloso	1,35	3,31
Ref. - RACN	0,97	2,98
Ref. - ART&ARTMAP Nebulosa	1,49	3,38

que sinaliza resultados satisfatórios para validação da ferramenta empregada na previsão.

BD-III: Essa última base de dados é constituída por dados retirados do site do ONS (ONS (2019)). Nele, foram obtidos para treinamento da rede os dados de carga de 15 dias, compreendidos entre os dias 17/10/2019 e 31/10/2019, referentes às regiões sudeste e centro-oeste do país. Desse modo, a curva de carga do 16º dia (01/11/2019) foi prevista pela rede neural implementada.

A Figura 4 apresenta o resultado da previsão de carga do dia 01/11/2019 para a BD-III. No gráfico apresentado nesta figura, a curva contínua representa a carga real verificada do acordo com os dados do ONS e a curva tracejada representa a carga obtida na saída da rede neural implementada. Através dessa imagem fica visualmente claro que a previsão de cargas apresentou bom resultado, sendo que o comportamento da carga real pôde ser previsto com baixos níveis erros. Percebe-se que o erro se eleva nas últimas horas do dia, fato que pode ser explicado, mais uma vez, pela utilização de valores previstos na estimação de cargas futuras e pela maior dificuldade de se prever os valores horários de carga nesse período.

Os erros MAPE e EM obtidos para BD-III, via programa em C++, podem ser vistos na Tabela 4. Neste caso, não há resultados para comparação dos erros, contudo, os erros verificados estão dentro dos mesmo níveis daqueles encontrados para as demais bases de dados consideradas (I e II), sendo, portanto, considerados satisfatórios.

3.3 Análise do Desempenho Computacional

Após análise de erros para validação da ferramenta empregada, é realizada uma comparação entre os tempos necessários para treinamento da rede com o emprego de duas diferentes linguagens de programação: C++ e *Phyton*. Salienta-se que o tempo relativo ao cálculo da previsão não é considerado, uma vez que este pode ser considerado insignificante quando comparado ao tempo de treinamento.

Os programas utilizados foram executados em um computador equipado com processador Intel Core i7-5500U, 2.40

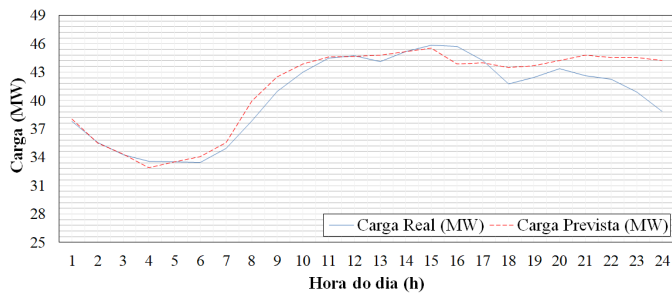


Figura 4. Curva de previsão para BD-III - 01/11/2019.

Tabela 4. Erros obtidos ao utilizar a BD-III.

Método	MAPE (%)	EM (%)
Presente trabalho - MLP via BP	2,70	13,76

GHz, com 8 GB de memória RAM, sistema operacional Windows 10/64 bits. Em relação às linguagens C++ e *Python*, os programas foram implementados seguindo o mesmo método e sem utilizar nenhuma função interna própria.

Para análise do desempenho computacional, foram realizados diversos treinamentos da rede neural para cada programa em cada base de dados, considerando pesos aleatórios de igual valor nos treinamentos. Os tempos médios de treinamento obtidos estão expostos na Tabela 5.

Tabela 5. Tempos médios de treinamento.

Base de Dados	Tempo C++ (s)	Tempo <i>Python</i> (s)
BD-I	6,78	813,25
BD-II	3,76	485,91
BD-III	0,59	72,84

Por meio da Tabela 5, pode-se perceber que os tempos médios de treinamento do programa implementado em *Python* foram consideravelmente maiores do que os tempos médios verificados com o programa implementado em C++. De forma geral e superficial, tal diferença pode ser explicada pelo fato da linguagem C++ ser considerada uma linguagem de “mais baixo nível” se comparada à linguagem *Python*. Isso implica, neste caso, que um menor tempo de desenvolvimento deve ser dispensado para implementação do programa em *Python*, mas em contrapartida, um tempo maior de execução é necessário.

4. CONCLUSÃO

Neste trabalho é implementado o método *Perceptron* Multicamadas (*Multilayer Perceptron* - MLP), com treinamento via Algoritmo de Retropropagação (*Backpropagation* - BP), para a previsão de cargas elétricas no curto prazo (24 horas a frente). A técnica de previsão é implementada em duas diferentes linguagens de programação, C++ e *Python*. Os desempenhos apresentados nos dois casos são comparados. Análises de erros de previsão para três diferentes estudos de caso são utilizados para validação da ferramenta implementada. Através dos resultados obtidos, pôde ser verificado que o método apresentou desempenho satisfatório.

Em relação desempenho computacional dos programas, foi possível concluir que a linguagem C++ apresentou tempos médios de treinamento da rede consideravelmente menores do que os tempos gastos pelo programa implementado

em *Python*. Entretanto, é importante deixar claro neste ponto que o tempo de execução de um programa não deve ser o único critério para escolha de uma linguagem de programação. Outros parâmetros devem ser considerados, como a bibliografia disponível (recursos de apoio) e, principalmente, o tempo necessário de desenvolvimento. Esse último critério, inclusive, é uma das principais características positivas do *Python*, sendo esta uma linguagem desenvolvida, dentre outros objetivos, com o intuito de facilitar e acelerar o tempo de criação das rotinas.

Deve-se destacar, por fim, que não foi realizado um estudo aprofundado para o ajuste dos parâmetros utilizados para a rede e, ainda assim, foi possível encontrar resultados satisfatórios em todos os estudos, fato que demonstra a qualidade do método empregado.

REFERÊNCIAS

- Altran, A.B. (2010). *Sistema Inteligente Para Previsão de Carga Multinodal em Sistemas Elétricos de Potência*. Ph.D. thesis, UNESP, Fac. de Eng. de Ilha Solteira.
- Fahad, M.U. and Arbab, N. (2014). Factor affecting short term load forecasting. *J. Clean Energy Technol.*
- Hammad, M., B.Jereb, Rosi, B., and Dragan, D. (2020). Methods and models for electric load forecasting: A comprehensive review. *Logistics Sustainable Transport.*
- Haykin, S. (1999). *Redes Neurais: Princípios e Prática*. Bookman.
- Hippert, H., Pedreira, C., and Souza, R. (2001). Neural networks for short-term load forecasting: A review and evaluation. *IEEE Trans. Power Syst.*
- Huang, S.J. and Shih, K.R. (2003). Short-term load forecasting via arma model identification including non-gaussian process considerations. *IEEE Trans. Power Syst.*
- Kuster, C., Rezgui, Y., and Mourshed, M. (2017). Electrical load forecasting models: A critical systematic review. *Sustainable Cities and Society.*
- Lopes, M.L.M. (2005). *Desenvolvimento de Redes Neurais Para Previsão de Cargas Elétricas de Sistemas de Energia Elétrica*. Ph.D. thesis, UNESP, Faculdade de Engenharia de Ilha Solteira.
- Lu, J.C., Niu, D.X., and Jia, Z.Y. (2004). A study of short-term load forecasting based on arima-ann. *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*.
- ONS (2019). Curva de carga horária. URL http://www.ons.org.br/Paginas/resultados-da-operacao/historico-da-operacao/curva_carga_horaria.aspx. [Online; acesso em Novembro-2019].
- Singh, A.K., Ibraheem, Khatoon, S., Muazzam, M., and Chaturvedi, D.K. (2012). Load forecasting techniques and methodologies: A review. *2012 2nd International Conference on Power, Control and Embedded Systems.*
- Srinivasan, D., Tan, S., Chang, C., and Chan, E. (1998). Pratical implementation of a hybrid fuzzy network for one-day-load forecasting. *IEEE P Gener Transm D.*
- Widrow, B. and Lehr, M. (1990). 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation. *Proceedings of the IEEE.*
- Zor, K., Timur, O., and Teke, A. (2017). A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting. *2017 6th International Youth Conference on Energy (IYCE)*.