FACULDADE DE TECNOLOGIA, CIÊNCIAS E EDUCAÇÃO

Graduação

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Utilização de Redes Neurais Artificiais para previsão do número de pacientes em um plantão hospitalar

José Isidoro de Oliveira Júnior
Adinovam Henriques de Macedo Pimenta (Orientador)

RESUMO

O número de pessoas atendidas em hospitais no país cresce a cada dia devido ao número de hospitais fechados e a diminuição do número de médicos em diversas regiões do país. Este fato acarreta em uma sobrecarga do número de pacientes nos hospitais. Dentre diversas consequências deste fato está a dificuldade em se escalar o número de enfermeiros em cada plantão, já que devido a vários fatores diferentes o número de pessoas pode aumentar muito ou diminuir significativamente. Considerando este cenário, o presente trabalho de pesquisa apresenta uma rede neural artificial que, baseando-se em dados de atendimentos de um período pré-determinado, é capaz de prever com uma margem de erro aceitável o número de pacientes que estará presente em determinado plantão de determinado dia do ano. Os resultados deste trabalho podem ser aplicados diretamente nas escalas de plantão de hospitais para que o número de enfermeiros seja suficiente para atender todas as pessoas e também para que o custo total das escalas seja proporcional ao número de enfermeiros realmente necessários.

Palavras-chave: Redes Neurais Artificiais. Modelo de previsão. Sistemas de apoio à tomada de decisão.

ABSTRACT

The number of people assisted in hospitals in the country grows every day due to the number of closed hospitals and the decrease in the number of doctors in various regions of the country. This fact results in an overload of the number of patients in the hospitals. Among several consequences of this fact is the difficulty in scaling the number of nurses in each shift, since due to several

different factors the number of people can increase significantly or decrease significantly. Considering this scenario, the present study presents an artificial neural network that, based on data from a predetermined period of time, is able to predict with an acceptable margin of error the number of patients that will be present in determining on-call certain day of the year. The results of this study can be applied directly to hospital attendance scales so that the number of nurses is sufficient to attend to all people and also so that the total cost of the scales is proportional to the number of nurses actually needed.

Keywords: Artificial Neural Networks, Forecasting Models, Decision Support Systems.

Introdução

A saúde é uma área em constante problema no país e, apesar de existirem alguns esforços no sentido de melhorar os resultados deste setor, há ainda muito o que fazer. A falta de profissionais na área é um problema que não é recente e precisa ser solucionado o mais rápido possível. Em notícia de maio de 2018[1] foi constatado que o problema de falta de enfermeiros é um problema mundial e que carece de soluções rápidas e eficientes. Houve recentemente um programa adotado no país, o Mais Médicos, que foi uma tentativa de solução para outro problema da saúde (a falta de médicos em regiões críticas) e atualmente passa por atualização em seu regulamento[2].

Considerando a realidade dos hospitais, existe um número fixo e pequeno de diversos profissionais que precisa ser cuidadosamente distribuído entre os turnos de plantões. Existem casos em que são alocados muitos profissionais e o plantão acontece com um número mínimo de ocorrências. Há também casos em que são alocados poucos profissionais e o plantão fica sobrecarregado. Este aumento no número de pessoas que chegam em hospitais é uma consequência direta do número de leitos fechados, que está em aproximadamente 6 leitos por dia no país[3].

Este problema reflete em uma necessidade maior de se alocar o número correto de enfermeiros para um determinado plantão para que os

[1] https://www.fenf.unicamp.br/pt-br/noticia/no-mundo-todo-falta-de-profissionais-enfermeiros-e- um-problema-grave-de-saude

[2] https://exame.abril.com.br/brasil/cuba-abandona-programa-mais-medicos-no-brasil-apos-ameacas-de-bolsonaro/

[3]https://noticias.r7.com/saude/hospitais-do-pais-fecham-seis-leitos-por-dia-aponta-estudo- 19102018

pacientes possam ser atendidos sem problemas e também que o custo seja o mais razoável possível.

O objetivo principal deste trabalho de pesquisa, levando em conta o problema apresentado, é o desenvolvimento de um sistema de auxílio à tomada de decisão para a escolha do número de enfermeiros plantonistas de um hospital em determinado dia e período. O sistema é composto por uma rede neural artificial que foi treinada utilizando dados de plantões de um hospital no período dos últimos dois anos considerando o número de atendimentos em cada período do dia (manhã, tarde e noite).

Os resultados obtidos demonstram que o software é capaz de auxiliar na previsão com erro aceitável para os dados que foram selecionados para treinamento e teste da rede neural. O sistema será colocado em prática nos próximos planejamentos de plantões do setor e sua eficácia em situações futuras poderá ser verificada.

Este artigo apresenta um referencial teórico sobre os temas relacionados ao trabalho na seção 2, os materiais e métodos utilizados para o desenvolvimento do trabalho na seção 3 seguidos pelos resultados e análise na seção 4 e as considerações finais.

1 Redes Neurais Artificias

Um neurônio artificial *k* é composto por:

- Sinapses (entradas), em que cada entrada x_j é um valor numérico do tipo binário, inteiro ou real;
- *Bias,* que é uma entrada especial com valor constante igual a *+1*;
- Os pesos w_{kj}, representam a importância de cada entrada x_j associada ao respectivo peso. Cada peso é um valor numérico real e o treinamento do neurônio é justamente a busca do melhor valor de cada peso;
- A junção aditiva, que soma todos os produtos obtidos entre a entrada x_j e o peso w_{kj} correspondente;
- A função de ativação $f(u_k)$, que transforma o resultado numérico do somatório, podendo saturá-lo entre dois valores limites. Esta função

pode ser: degrau, sigmoidal, linear, semi-linear, tangente hiperbólica, etc.

A Figura 1 apresenta um diagrama de neurônio *Perceptron* com todas as suas principais características. As operações em um neurônio artificial se resumem em: sinais (valores), que são apresentados às entradas (x_1 à x_m) e, em seguida, cada sinal é multiplicado por um peso que indica sua influência na saída *y* do neurônio.

Em seguida é feita, então, a soma ponderada dos sinais de entrada que produz um nível de atividade que será utilizado como entrada de uma função de ativação. A função de ativação transforma o valor de entrada, limitando o valor da saída entre máximo e mínimo e introduzindo não linearidade ao modelo.

O *bias* tem o papel de aumentar ou diminuir a influência do valor das entradas, sendo possível considerar o *bias* como uma entrada de valor constante 1, multiplicado por um peso igual ao valor do Bias.

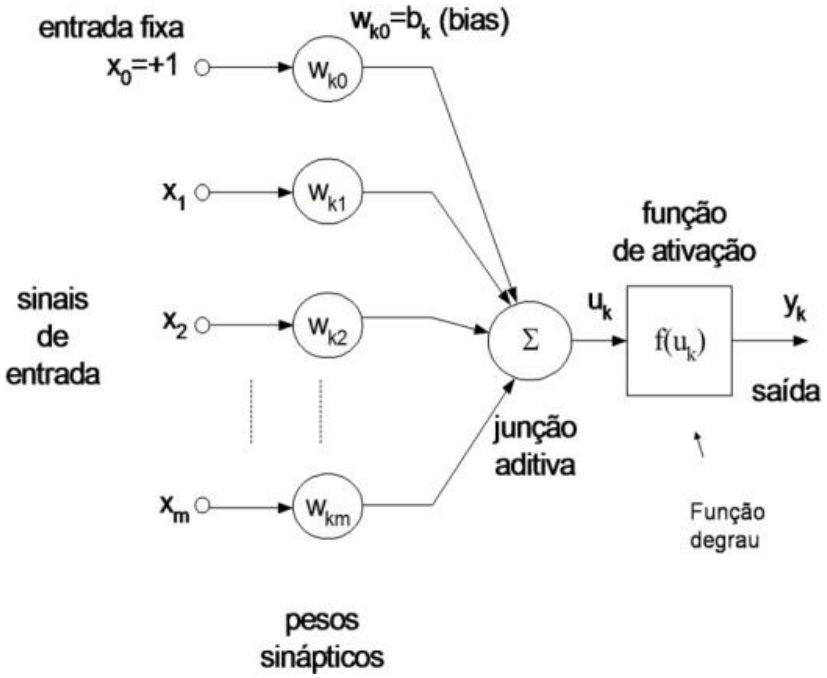

Figura 1. Exemplo de neurônio perceptron
Fonte: Elaborado pelo autor

Dentre as funções de ativação é possível citar a função de ativação degrau que possui o formato apresentado na Figura 2. Neste formato, quando o sinal de entrada atinge um determinado limiar, é o neurônio é disparado e

muda a sua saída para 1.

O neurônio artificial como apresentado é simplesmente um separador linear, ou seja, ele cria uma divisão linear em um plano 2D que irá dividir os dados em dois grupos distintos (HAYKIN, 2007). Caso o problema não seja linearmente separável este neurônio não poderá resolver o problema. A associação destes neurônios em rede de apenas uma camada também não soluciona este problema sendo necessário criar uma rede neural artificial com múltiplas camadas.

Dentre as redes neurais com múltiplas camadas utilizadas para a solução de diversos problemas é possível destacar as redes *Multi-Layer Perceptron.* Estas redes são redes que possuem a camada de entrada, uma ou mais camadas escondidas e as camadas de saída de dados. Este tipo de rede neural permite que problemas mais complexos sejam resolvidos. Porém, por outro lado, também necessita de um algoritmo de treinamento mais complexo.

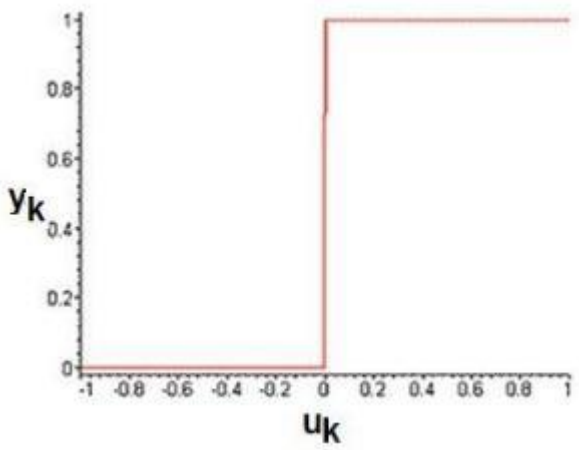

Figura 2. Exemplo de função degrau
Fonte: Elaborado pelo autor

O neurônio perceptron foi desenvolvido por Rosenblatt nos anos 50. Trata-se de um neurônio artifical que possui as mesmas características do neurônio apresentado na Figura 1. Porém, o perceptron tem um algoritmo de treinamento: a regra delta. Basicamente, a regra delta é um algoritmo de treinamento que atualiza os pesos dos neurônios considerando que o valor novo do peso é a soma do peso anterior a uma multiplicação entre: a taxa de aprendizado definida para o neurônio (um valor entre 0 e 1) e o erro que o neurônio apresenta entre a saída desejada e a saída obtida.

Considerando esta maneira de se calcular o novo peso é possível dizer que o aprendizado que ocorre na rede é um aprendizado supervisionado, ou seja, a rede possui um par entrada-saída que diz o que é esperado da rede e o erro é utilizado diretamente para cálculo do novo valor do peso da estrutura.

Este algoritmo funciona apenas para o cálculo do erro de neurônios em que a saída esperada é conhecida para uma determinada entrada. Portanto, com a regra delta, o melhor que se pode fazer é a organização de uma rede de camada simples que agrupa classificadores lineares. A mudança na arquitetura da rede para que a rede possua diversas camadas, conhecida como MLP (*Multi Layer Perceptron*) é exemplificada na Figura 3.

Uma rede MLP possui então as camadas de entrada, as camadas escondidas e a camada de saída. Os neurônios normalmente são representados de forma diferente na camada de entrada por serem considerados apenas receptores dos valores de entrada. O treinamento deste tipo de rede utiliza o conceito de retro propagação do erro, ou seja, o erro é calculado nas saídas e é então propagado às camadas anteriores da rede para que os pesos sejam atualizados (RUMELHART; MCCLELLAND, 1986). Este procedimento é necessário pois não é possível criar um par entrada-saída para um neurônio da camada escondida levando em conta que não se sabe qual a saída esperada de um neurônio nesta posição da rede.

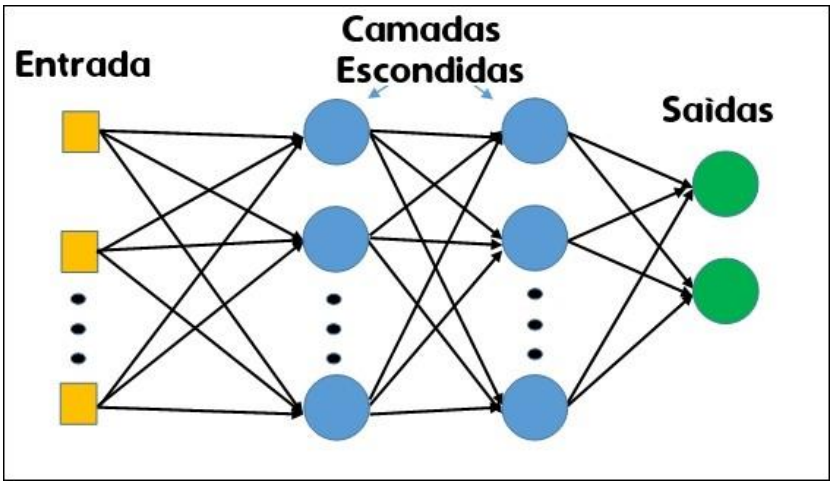

Figura 3. Exemplo de rede MLP.
Fonte: Elaborado pelo autor.

No caso do *backpropagation*, algoritmo que implementa a retro propagação do erro, os pesos novos da camada de saída da rede são calculados exatamente como na regra delta, porém a atualização dos pesos

das camadas escondidas é uma equação que considera que o peso novo é a soma do peso antigo mais a multiplicação de: uma taxa de aprendizado α; a saída do neurônio; e um termo que é calculado baseado no valor da derivada da função de ativação para a entrada do neurônio. Utilizar o gradiente para o cálculo do novo valor do peso faz com que o algoritmo tenha tendência a convergir para determinada solução. Para um detalhamento maior do algoritmo de treinamento *backpropagation* consultar Rumelhart e McClelland (1986).

O algoritmo *backpropagation* é um algoritmo que reconhecidamente é capaz de treinar uma MLP. Porém, existe o problema do tempo de convergência. O cálculo do gradiente e a forma como é utilizado para ajuste dos pesos contribui para que o tempo de convergência do algoritmo seja demasiado elevado em alguns casos. Uma alternativa interessante é a utilização de algumas variações do algoritmo em casos específicos. Um dos algoritmos que atinge bons resultados é o algoritmo Rprop (*Resilient Backpropagation)* (HAYKIN, 2007). Este algoritmo, ao invés de utilizar o valor da derivada para atualizar os pesos da rede, utiliza o sinal da derivada (positivo ou negativo) e uma taxa de atualização. Neste caso é possível ajustar a taxa de atualização independentemente da função de ativação. Esta vantagem pode se tornar um problema ao se configurar o algoritmo.

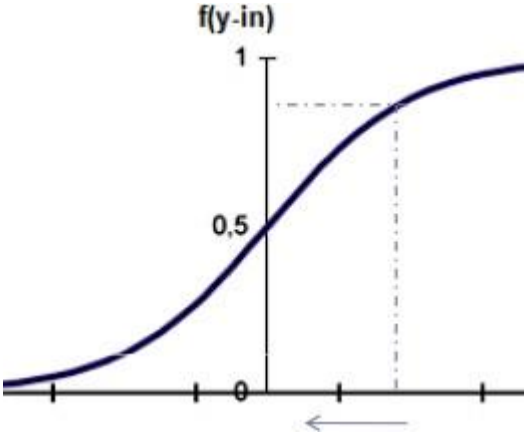

Figura 4. Exemplo de função sigmoidal simétrica utilizada em MLPs.
Fonte: Elaborado pelo autor.

Para esta abordagem, o gradiente só pode ser calculado a partir de uma função de ativação diferenciável. Uma grande quantidade de problemas pode ser resolvida com a utilização da função sigmoidal simétrica (Figura 4).

Segundo Braga, Carvalho e Ludermir (2000) é possível que uma rede MLP de três camadas escondidas consiga resolver problemas de qualquer complexidade. Considerando este fato, o presente trabalho de pesquisa irá utilizar uma rede MLP para fazer a sugestão do número de enfermeiros necessários para determinado plantão de atendimento de um hospital com base nos dados dos números de atendimentos realizados por um período de 2 anos.

Foi utilizado um framework para tratar os dados e configurar o classificador e ele será apresentado a seguir.

2 Materiais e Métodos

Este trabalho de pesquisa tem por objetivo construir um sistema de auxílio à decisão para a escolha do número de enfermeiros plantonistas para determinado período (dia e horário). Os dados necessários para a realização do trabalho foram obtidos da Santa Casa de Misericórdia de Pirassununga-SP. O hospital é um importante centro de tratamento para a cidade e passou por dificuldades financeiras recentemente[4]. Apesar da boa estrutura de hospital, existe um problema constante de filas para atendimento. Este problema, em partes, pode ser causado pelo número insuficiente de profissionais atendendo em determinados horários.

O setor de TI da entidade liberou o acesso aos dados de números de atendimento no plantão em três diferentes períodos (manhã, tarde e noite) durante todos os dias de outubro de 2016 a outubro de 2018. Estes dados em forma de planilha serão utilizados em um ambiente de programação para serem tratados e, então, utilizados para treinamento e validação da rede neural.

O ambiente de programação utilizado para o desenvolvimento do trabalho é o framework WEKA. O ambiente WEKA (sigla para *Waitako Environment for Knowledge Analysis)* é desenvolvido em linguagem JAVA e amplamente utilizado por possuir diversos algoritmos de aprendizado de máquina previamente implementados e disponíveis para configuração e

[4] https://g1.globo.com/sp/sao-carlos-regiao/noticia/2018/09/21/com-salarios-atrasados-medicos-da-santa-casa-de-pirassununga-sp-reduzem-os-atendimentos.ghtml

utilização.

Outra parte fundamental do ambiente é a parte de tratamento de dados. O software trabalha com os dados em um formato .arff, que indica as classes de dados e os respectivos valores. Este formato não é tão simples de ser gerado a partir de um conjunto de dados, porém o software possui a opção de se trabalhar com o formato .csv (*comma separated value)*, que é facilmente obtido por softwares de planilhas, ou por meio da conversão do formato .csv para o formato .arff. Neste trabalho optou-se por exportar os dados para o formato .csv.

Dependendo da linguagem adotada no software de exportação é preciso cuidado com o arquivo gerado. Neste trabalho os dados foram gerados com casas decimais separadas por vírgula. Para que os dados pudessem ser lidos diretamente pelo WEKA foi necessário inicialmente substituir todas as vírgulas por ponto (pois no sistema brasileiro a separação das casas decimais é feita por vírgula e no software WEKA é considerado o padrão internacional). Em seguida foi feita a substituição dos pontos-e-vírgulas (caractere de separação) por vírgulas.

Este trabalho utiliza como estrutura de previsão uma rede neural do tipo MLP. A instalação padrão do software WEKA permite que sejam utilizados alguns algoritmos básicos de agrupamento, classificação e pré-processamento de dados. Porém, para a realização deste trabalho foi necessário instalar o pacote MultilayerPerceptron [5] [7] para a configuração, treinamento e validação da rede. Diversos outros pacotes também estão disponíveis para o ambiente e podem ser utilizados para mineração de dados. A arquitetura final da rede foi desenvolvida de acordo com os resultados dos erros de previsão.

O conjunto de dados obtido para o sistema possui intervalo de outubro de 2016 a outubro de 2018. A rede neural então foi treinada e validada utilizando a técnica de *n-fold cross validation,* também conhecida como validação cruzada. Esta técnica é uma técnica de reamostragem de dados que permite avaliar modelos de aprendizado de máquina em um espaço limitado de amostras. O único parâmetro do modelo é o parâmetro *n* que se refere ao

[5] http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/MultilayerPerceptron.html

número de grupos que serão obtidos a partir da divisão dos dados. Assim que um número é escolhido o método deve ser referido a partir do número. Para este trabalho foram utilizados 10 grupos. Sendo assim, utilizamos o *10-fold cross validation.*

Este método é popular por ser simples e produzir resultados menos enviezados e menos otimistas com relação aos dados. O procedimento para o método pode ser dividido em 4 passos básicos:

1. Inicialmente o banco de dados é reordenado de forma aleatória;
2. Em seguida os dados são divididos em *n* grupos;
3. Para cada grupo:
    1. Um grupo é considerado o conjunto de teste;
    2. Os grupos restantes são encarados como conjunto de treinamento;
    3. O modelo é treinado com o conjunto de treinamento e testado em seguida com o conjunto de testes;
    4. É feito um ranking do resultado e descarte este modelo.
4. Por fim, é feita a combinação dos resultados para então obter a real nota de avaliação do modelo.

É importante salientar que assim que determinado dado é colocado em um grupo ele dever pertencer apenas àquele grupo por todo o procedimento da validação cruzada. Isso garante que um determinado dado é utilizado para teste 1 vez e para treinamento *n-1* vezes. Diversos testes empíricos demonstraram que os valores *5* e *10* para *n* são os que geram resultados menos afetados por *bias e* variância dos dados. Outra questão importante é balancear os grupos com relação ao número e diversidade das amostras.

3 Resultados e Análise

As seções anteriores descreveram os aspectos básicos da estrutura utilizada para este trabalho, bem como as técnicas e o ambiente em que o trabalho foi desenvolvido. Nesta seção serão apresentados os resultados da criação de um modelo de MLP capaz de auxiliar na previsão do número de atendimentos do hospital escolhido a fim de melhorar a distribuição de profissionais no plantão.

Algumas características básicas da rede neural devem ser definidas a fim de se obter um treinamento eficiente. A primeira parte do trabalho é decidir como será feita a modelagem do problema. Os dados estavam modelados considerando o dia do ano, o período (manhã, tarde ou noite) e o número de atendimentos em determinado dia e período. Para que a rede fosse capaz de considerar todos os dados sem repetições foi preciso modelar os dados de entrada em pares entrada/saída de dados. Os valores de número de atendimentos continuam numéricos, mas são divididos por 100, os períodos da semana foram mapeados em 0,2 (representando o período da manhã), 0,5 (representando o período da tarde) e 0,7 (representando o período da noite).

Esta divisão foi feita pelo fato da rede lidar melhor com números entre 0 e 1, devido a natureza de seus pesos serem normalmente iniciados aleatoriamente entre -1 e 1. Os valores da data foram concatenados e divididos por 100000, sendo que o dia 25/12/16 foi interpretado pela rede como 0,251216. Esta modelagem de dados se mostrou bastante eficaz considerando os resultados obtidos.

Sendo assim, é possível modelar a rede com 3 neurônios de entrada sendo o primeiro neurônio responsável por receber o número de atendimentos, o segundo neurônio responsável por receber o período do dia e o terceiro neurônio responsável por representar a data. A rede foi construída com duas camadas intermediárias com 10 neurônios cada uma. A saída da rede deveria ser mapeada para que fosse interpretada corretamente. Sendo assim, foram criados 3 possíveis saídas para a rede em um determinado intervalo:

- O intervalo [-1; -0,5] indica um número grande de profissionais;
- O intervalo (-0.5; 0,5] indica um número médio de profissionais; e
- O intervalo (0,5; 1] indica um número grande de profissionais.

Desta forma, a saída compreende o intervalo [-1; 1], permitindo a utilização de uma função sigmoidal simétrica inclusive no neurônio de saída da rede.

Após a modelagem dos dados é necessário escolher o algoritmo de treinamento da rede e também os parâmetros do algoritmos. O algoritmo de treinamento do pacote utilizado no weka é o *backpropagation.* Muitos de seus parâmetros foram utilizados em seu valor padrão. Porém, alguns parâmetros foram alterados empiricamente de forma a obter o melhor modelo.

A taxa de aprendizado foi alterada para 0,5 (o padrão era 0,3) buscando acelerar o aprendizado da rede. O momentum do algoritmo que representa o quanto o algoritmo avança em sua convergência foi de 0,2 para 0,4 aumentando assim a velocidade com que o algoritmo converge para uma solução. A rede foi configurada para que as conexões fossem feitas automaticamente da forma *fully- connected,* ou seja, a rede possui todos os neurônios conectados com todas as conexões possíveis.

O critério de parada foi escolhido para um erro médio quadrático não maior que 0,2 na classificação de amostras do conjunto de teste, erro este que tornaria a classificação aceitável. O cálculo do erro médio quadrático foi feito de

acordo com a equação 1, em que *n* é o número de amostras, Y_i é o vetor de resultados obtidos e $\hat{Y}$ é o vetor de resultados esperados. Este valor de 0,2 foi escolhido empiricamente e apresentou resultados satisfatórios.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2. \tag{1}$$

O decaimento automático da taxa de aprendizagem também foi desativado. Esta foi a configuração que obteve os melhores resultados da rede, porém não foi a única testada. A variação de parâmetros e os respectivos erros dos testes iniciais são apresentados na Tabela 1. O tempo de treinamento da rede com todo o conjunto de dados não ultrapassou as duas horas.

Tabela 1. Testes iniciais para a configuração do classificador

Rede	Camadas	Neurônios Escondidos	Taxa de Aprendizado	Momentum	Erro Quadrático
1	2	6	0,4	0,2	74,6584
2	2	10	0,4	0,2	12,5147
3	2	50	0,4	0,2	35,7958
4	3	10	0,4	0,2	48,6235
5	3	50	0,4	0,2	39,8356

Fonte: Elaborado pelo próprio autor

Considerando os primeiros resultados é possível concluir que a rede

que obteve melhor resultado é a rede 2 com duas camadas escondidas de 10 neurônios cada. Como o erro encontrado não se aproximou do critério de parada o treinamento foi encerrado com 5000 épocas.

Após este resultado foram feitas algumas alterações nos parâmetros da rede para aproximar o erro obtido do erro desejado. Uma delas foi ativar o decaimento automático da taxa de aprendizado que inicia-se em 0,5 e é modificada ao longo do treinamento terminando em 0,2. Além disso, o *momentum* do algoritmo foi modificado para 0,3. Estas mudanças foram responsáveis por fazer com que o erro final do treinamento fosse de 0,19845, atingindo assim o objetivo inicial. Com relação ao número de instâncias classificadas corretamente do conjunto de testes, a rede final classificou corretamente cerca de 80% das instâncias, indicando que o resultado obtido é válido e se aproxima do que era desejado.

A partir dos resultados obtidos é possível perceber que a abordagem escolhida para solucionar o problema foi satisfatória, e que os dados disponibilizados permitiram que fosse realizada a abordagem, apesar de serem simples. A técnica do *10-fold cross-validation* resultou em uma rede bem treinada. Os dados analisados demonstram que em alguns períodos há uma concentração maior de atendimentos que podem ou não ser recorrentes. A rede neural, apesar de sua capacidade de generalização, pode não ser capaz de direcionar corretamente a decisão em dias em que um evento inesperado gerou um aumento de pessoas na cidade e que precisavam de atendimento.

Considerações Finais

Este trabalho de pesquisa apresentou a elaboração de um classificador baseado em redes neurais MLP para o auxílio da previsão do número de funcionários a serem alocados para que os atendimentos no plantão da Santa Casa de Pirassununga fossem realizados de forma mais adequada.

A técnica utilizada se mostrou eficiente e os resultados demonstram que a estrutura escolhida para a solução do problema foi capaz de atingir resultados satisfatórios. Os resultados ainda precisam ser testados em situações reais diferentes das apresentadas, sendo este o principal trabalho

futuro. Além disso, podem ser utilizados outros classificadores e seus resultados podem ser avaliados de forma comparativa.

Referências

BRAGA, A. de P.; CARVALHO, A. P. L..; LUDERMIR, T. B. **Redes Neurais Artificiais:** teorias e aplicações. 2. ed. [S.l.]: LTC, 2000.

HAYKIN, S. **Redes Neurais:** princípios e práticas. 2. ed. [S.l.]: Bookman, 2007.

RUMELHART, D. E., McCLELAND, J. L. (Eds.). **Parallel Distributed Processing:** explorations in the microstructure of cognition. Foundations. MIT Press, Cambridge, MA, USA, 1986. V. 1.

VALENÇA, M. **Fundamentos das redes Neurais.** 2. ed. Pernambuco: Livro Rápido, 2009.

WITTEN, I. H.; FRANK, E.; HALL, M. A. Data Mining: Practical machine learning tools and techniques. 3. ed. [S.l.[: Morgan Kaufmann, 2011.