

PREDIÇÃO DE CASOS DE DENGUE UTILIZANDO REDES NEURAIS ARTIFICIAIS

Fernando Eduardo Rezende Mattioli¹; Rogério Bernardes Andrade²; Emanuel Tobias Estevez³

¹ Universidade Federal de Uberlândia – UFU, Uberlândia (MG), Brasil
^{1,2,3} Faculdade de Talentos Humanos - FACTHUS, Uberaba (MG), Brasil

fernando.mattioli@facthus.edu.br, rbandrade@facthus.edu.br, emanuel_tobias_@hotmail.com

RESUMO: O presente artigo teve como objetivo apresentar um modelo de previsão de casos dengue através da aplicação de treinamento de rede neural do tipo Perceptron. A rede neural foi alimentada com os dados oferecidos pelos órgãos governamentais, SES (Secretaria de Estado de Saúde) e o SINAM (Sistema de Informação de Agravos de Notificação). A referida rede foi desenvolvida em linguagem *Python*, sob o sistema operacional Linux. Como resultado, observou-se que a rede neural resultante foi muito próxima ao resultado obtido por uma série real que serviu como parâmetro de comparação para verificarmos o desempenho da rede neural desenvolvida. Concluiu-se, portanto, que a rede desenvolvida no presente trabalho poderá servir como método de previsão que poderá sugerir atos futuros de prevenção contra a dengue.

PALAVRAS CHAVE: Predição; Redes neurais; *Python*; Dengue

DENGUE CASES PREDICTION USING ARTIFICIAL NEURAL NETWORKS

ABSTRACT: This paper aims to present a prediction model of dengue cases by applying neural network training of perceptron type. The neural network was fed with data provided by governmental agencies, SES (Secretary of State for health) and SINAM (Information System of reportable diseases). This network was developed in Python language under the Linux operating system. As a result, it was observed that the resulting neural network was very close to the result obtained by a real series which served as the comparison parameter to verify the performance of the neural network developed. Therefore, it was concluded that the network developed in this study can serve as prediction method which may suggest future acts of prevention against dengue fever.

KEYWORDS: Prediction; Neural networks; Python; Dengue fever

INTRODUÇÃO

A dengue é uma doença febril infecciosa cujos sintomas sobrevivem repetidamente e se caracteriza por fortes dores na cabeça, olhos, músculos e articulações, inflamação de garganta, sintoma catarral e as vezes erupções cutâneas e inchaços doloridas. É causada por um vírus filtrável, transmitido por duas espécies de mosquitos, o *Aedes Aegypti* e o *Aedes Albopictus*; também chamado de febre da dengue. (MICHAELIS, 2004, p.364).

Com o grande aumento dos casos de dengue ao longo dos anos em todo o Brasil, surgiu a necessidade de desenvolvimento ferramentas e atividades de combate mais eficientes a essa doença. A criação de novas estratégias e principalmente um controle detalhado das informações requerem ferramentas informatizadas.

Objetivando um combate contra a dengue de forma mais eficaz, esse trabalho acadêmico apresenta um sistema de predição de casos da doença para uma política de controle eficaz, utilizando redes neurais artificiais na linguagem de programação *Python*.

MATERIAIS

Para o estudo dos casos confirmados e alimentação da rede neural, foram utilizados dados de dois órgãos governamentais. O primeiro é a SES (Secretaria

Estadual de Saúde), contendo dados de 1990 a 1998. O segundo é o SINAM (Sistema de Informação de Agravos de Notificação), contendo dados de 1999 a 2014. Os órgãos fazem o levantamento dos dados por estado, que são somados e produzem os índices nacionais de confirmação de casos de dengue.

Foram utilizadas neste estudo ferramentas de software livre, que flexibilizaram o desenvolvimento tornando assim o trabalho portátil a qualquer dispositivo atual. Foram utilizados como softwares de plataforma e desenvolvimento o sistema operacional GNU/Linux (distribuição Ubuntu versão 14.04 LTS 64-bits), a linguagem de programação *Python* (versão 2.7, já nativa dos sistemas operacionais baseados em Unix), biblioteca Matplotlib para criação dos gráficos, e a biblioteca PyBrain (versão 0.3.3) para configuração da rede neural artificial. Os testes foram executados em um laptop CCE WIN T33B com processador i3 - 330M, 2.13GHz e 3MB de cache, 6GB de memória RAM DDR3, SSHD de 500GB.

GNU/LINUX

O GNU/Linux é um sistema operacional que auxilia a interação entre o ser humano e o computador. Grande parte das aplicações do GNU/Linux inclusive seu kernel que apareceu pela primeira vez em 1991 pela iniciativa do estudante Finlandês Linus Torvalds, foram

herdadas do GNU Project, que é um imenso conjunto de aplicações de software livre escritos originalmente para serem utilizadas no sistema operacional Unix. Por esse motivo, é comum atribuir o nome GNU/Linux com o sistema operacional, Linux.

O GNU/Linux foi anunciado como substituto do Minix, sistema operacional do tipo Unix-like, gratuito e de código fonte aberto. TANENBAUM et al.(2010, p. 508) diz que “um estudante finlandês chamado Linus Torvalds decidiu escrever um outro clone do Unix, chamado Linux, o qual seria um sistema de produção completo com muitas características que faltavam (intencionalmente) no Minix”.

PYTHON

Python é uma linguagem de programação interpretada, portátil, orientada a objetos, de tipagem dinâmica. No passado essas linguagens eram usadas apenas para automatizar tarefas usando scripts. O principal fator de aumento do uso das linguagens dinâmicas foi o crescimento da internet e a necessidade de ferramentas simples e de grande produtividade.

Segundo BORGES (2010, p. 14) “a linguagem foi criada em 1990 por Guido Van Rossum, no Instituto Nacional de Pesquisa para Matemática e Ciência da Computação da Holanda (CWI) e tinha originalmente foco em usuários como físicos e engenheiros. O Python foi concebido a partir de outra linguagem existente, chamada ABC. Hoje, a linguagem é bem aceita na indústria por empresas de alta tecnologia”.

REDES NEURAS ARTIFICIAIS

As redes neurais artificiais (RNAs) são algoritmos de processamento de dados baseado na mente humana que armazenam conhecimento após experimentos. As RNAs visam solucionar problemas de classificação e predição com base em modelos de aprendizagem baseados em conjuntos de dados de treinamento.

Segundo HAYKIN (2001, p. 28), uma rede neural é um processador maciçamente e paralelamente distribuído, constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental é torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:

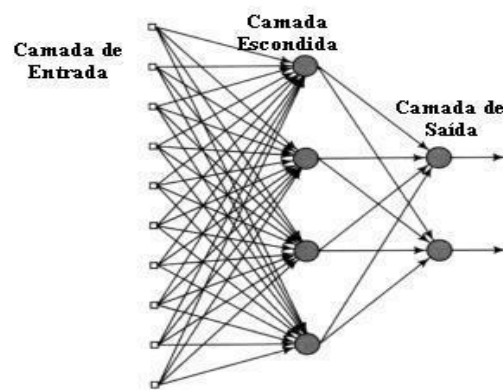
1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.
2. Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

Esse trabalho acadêmico utiliza uma rede neural artificial do tipo Perceptron de múltiplas camadas (*multilayer perceptron* ou MLP). Esse tipo de rede é a mais indicada para previsão de dados, já que seu treinamento pode ser supervisionado com algoritmos de *backpropagation* (retropropagação). HAYKIN et al. (2001, p. 183) afirmam que “os Perceptrons de múltiplas camadas têm sido aplicados com sucesso para resolver diversos problemas difíceis, através do seu treinamento de

forma supervisionada com um algoritmo de retropropagação de erro. O aprendizado com retropropagação é fixado em duas etapas através de diferentes camadas da rede:

1. Propagação: Uma etapa para frente. Um vetor de entrada é atribuído aos nós sensoriais da rede se propagando camada por camada produzindo as saídas.
2. Retropropagação: A partir do erro calculado, são realizados ajustes nos nós da rede e da camada de saída em direção à camada de entrada. A Figura 1 apresenta um exemplo de rede neural Perceptron de múltiplas camadas.

Figura 1: Exemplo de Rede Neural Perceptron de Múltiplas Camadas. Camadas de entrada: os dados são apresentados a rede. Camadas escondidas: a maior parte do processamento é feita através de conexões. - Camada de saída: responsável pela apresentação do resultado final.



Fonte: <http://www.visgraf.impa.br/Courses/ip00/proj/texta/meto1.html>

PYBRAIN

PyBrain é uma biblioteca modular para Python. Seu objetivo é oferecer algoritmos flexíveis, fáceis de serem utilizados, mas ainda poderosos para as tarefas de aprendizado de máquina. A biblioteca contém algoritmos para redes neurais, reforço de aprendizagem e a combinação dos dois, aprendizagem não supervisionada e de evolução para a resolução da maioria dos problemas atuais.

MÉTODOS

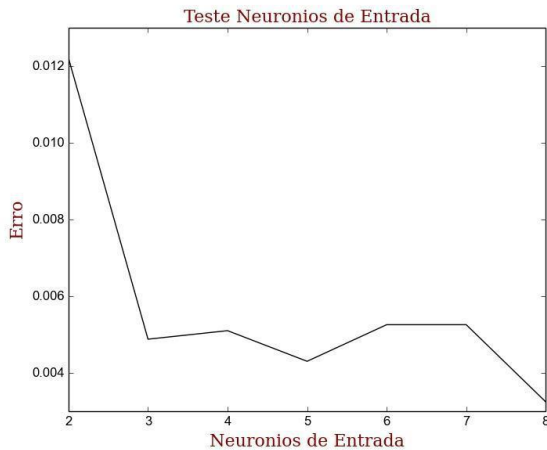
Os dados de treinamento foram coletados em dois órgãos governamentais SES e SINAM, e estão disponíveis no Portal da Saúde em (SINAM/SES, 2016). Os dados coletados foram normalizados, dividindo-se todos os valores pelo maior valor coletado.

Com o objetivo de se encontrar os melhores parâmetros de configuração da RNA, foram realizados testes com diferentes configurações. Para cada configuração foram realizadas 10 repetições, calculando-se o erro médio e o tempo médio de execução.

RESULTADOS E DISCUSSÃO

O Gráfico 1 apresenta o resultado dos experimentos realizados para avaliação do impacto do número de neurônios de entrada no desempenho da RNA.

Gráfico 1: Teste Neurônios de Entrada.

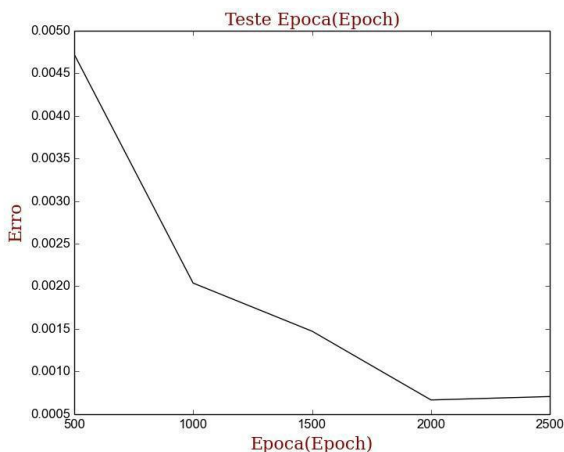


Com dois neurônios na camada de entrada, a taxa de erro ficou em $1,2 \times 10^{-2}$. O menor tempo de execução foi dezoito segundos e oito milésimos. Foram utilizados vinte e três dados de entrada. Treinando a rede neural com quatro, cinco, seis e sete neurônios, não se obteve grande diferença comparando com três neurônios de entrada onde a taxa de erro obtida é de $4,9 \times 10^{-3}$. O menor tempo de execução foi dezoito segundos e um milésimo, utilizando-se vinte e dois dados de entrada.

Observando o Gráfico 1 a melhor configuração na camada de entrada foi de oito neurônios com uma taxa de erro de $3,2 \times 10^{-3}$. O menor tempo de execução foi de quinze segundos e setenta e um milésimos, para dezessete dados de entrada. Devido à quantidade reduzida de dados de entrada disponíveis para os testes, não foram realizados experimentos com mais de 8 neurônios na camada de entrada.

O segundo teste foi feito variando-se a quantidade de épocas de treinamento. Os resultados desse teste são apresentados no Gráfico 2.

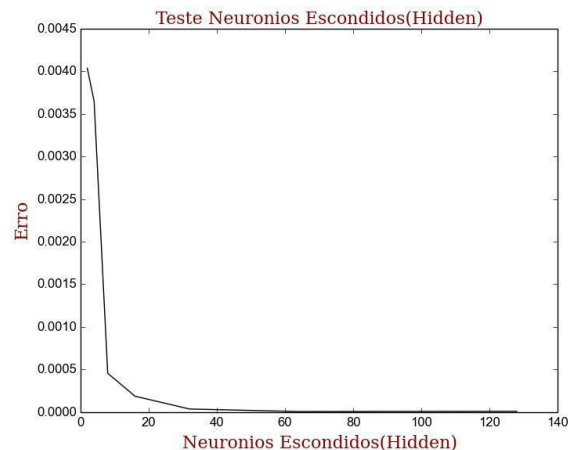
Gráfico 2: Teste de Épocas.



O treinamento começou com 500 épocas com uma taxa de erro de $4,7 \times 10^{-3}$, com tempo de execução de oito segundos e seis milésimos. Os melhores resultados foram obtidos com 2000 épocas, com uma taxa de erro de $6,6 \times 10^{-4}$ e tempo de execução 29 segundos e 83 milésimos. Com adição de 500 épocas para cada novo treinamento (até o limite de 2500 épocas) observa-se elevação da taxa de erro para $7,05 \times 10^{-4}$ com tempo de execução de 38 segundos e vinte e seis milésimos.

O Gráfico 3 apresenta a influência do número de neurônios na camada escondida no desempenho da rede.

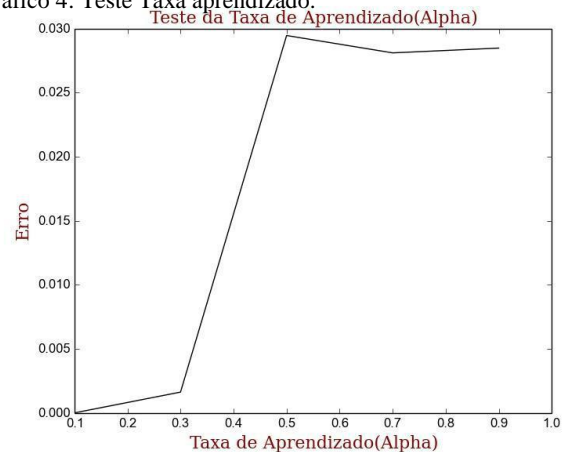
Gráfico 3: Teste Neurônios Ocultos



Com dois neurônios na camada oculta, foi observada uma taxa de erro de $4,03 \times 10^{-3}$, com o tempo de execução de 29 segundos e 83 milésimos. O número de neurônios na camada escondida foi então dobrado, até o valor limite de 128. O melhor resultado encontrado, para 64 neurônios na camada escondida, apresentou uma taxa de erro de $6,6 \times 10^{-6}$, com tempo médio de execução de 30 segundos e 87 milésimos. Com 128 neurônios a rede neural apresentou um aumento na taxa de erro chegando a $9,2 \times 10^{-6}$.

Finalmente, o Gráfico 4 apresenta a influência da taxa de aprendizagem no desempenho da rede.

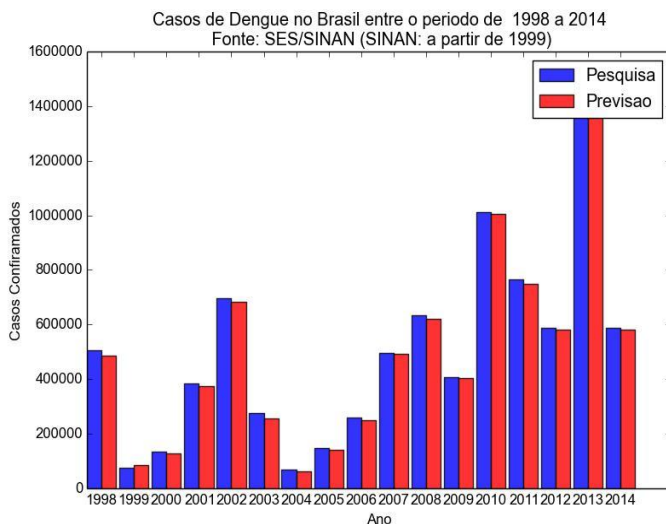
Gráfico 4: Teste Taxa de Aprendizagem.



Para a taxa de aprendizado, foi definido o valor inicial de 0,1, apresentando um erro de $6,6 \times 10^{-6}$ e tempo de execução de 30 segundos e 87 milésimos. Neste experimento, as alterações realizadas na taxa de aprendizado não resultaram em diminuição no erro da rede. A primeira alteração – para o valor de 0,3 - apresentou aumento na taxa de erro para $1,6 \times 10^{-3}$ com tempo de execução de 32 segundos e 97 milésimos. Aumentando este parâmetro de 0,2 em 0,2 até o valor limite de 0,9, a taxa de erro aumentou como se observa no Gráfico 4.

Os testes realizados possibilitaram a definição do modelo de rede neural artificial Perceptron de múltiplas camadas para a previsão dos casos de dengue. Ficou claro que esta etapa foi a mais importante desse trabalho acadêmico, que resultou na seguinte configuração: oito neurônios na camada de entrada, 64 neurônios na camada oculta, um neurônio de saída, taxa de aprendizado de 0,1 e 2000 épocas de treinamento. Esta configuração resulta em uma taxa de erro de $6,6 \times 10^{-6}$. A comparação entre os dados reais e os dados previstos pela rede, considerando-se apenas os dados de treinamento, pode ser visualizada no Gráfico 5.

Gráfico 5: Resultados da Rede Neural.



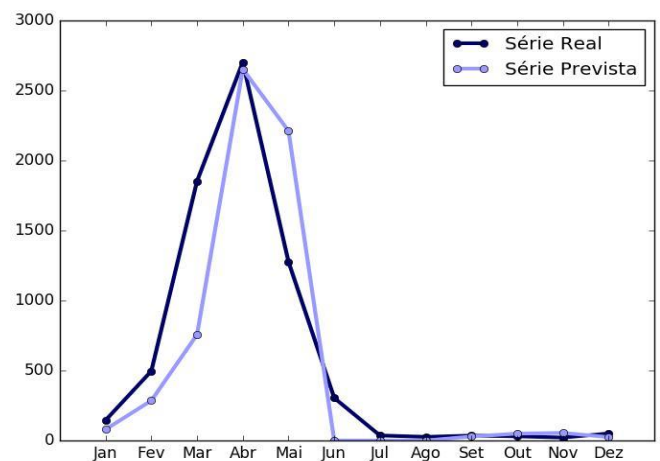
Com o objetivo de se verificar o desempenho da rede na predição de dados desconhecidos, foram utilizados os dados do Quadro 1. A rede foi treinada com os dados de 1998 até 2010, sendo os dados de 2011, 2012 e 2013 utilizados para teste da predição. O Gráfico 6 apresenta o resultado da predição para o ano de 2013.

Quadro 1: Dados para de treinamento e previsão.

Casos de dengue confirmados por mês em moradores de Campinas													
	Jan	Feb	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	
1998	237	331	562	187	32	11	6	4	7	7	5	8	1.397
1999	7	12	27	49	8	3	1	3	3			4	117
2000	6	11	21	15	8	4	4	3		2	2	5	81
2001	32	38	160	223	136	21	13	10	2	2	5	86	728
2002	224	364	348	266	156	50	8	9	6	4	7	22	1.464
2003	90	91	125	76	28	7	2			1		3	423
2004	9	8	6	3	1							3	30
2005	5	7	8	38	29	17	8	2	2			3	119
2006	8	24	187	292	129	31	7	16	10	17	10	11	742
2007	169	922	3.213	4.207	2.364	300	67	17	35	49	57	42	11.442
2008	40	37	72	79	21	10	8	9	2	8	6	14	306
2009	17	29	53	40	25	16	2	3	2	3	3	7	200
2010	65	249	626	942	630	84	11	10	5	5	8	12	2.647
2011	68	288	658	1.202	714	133	26	11	13	23	22	20	3.178
2012	49	53	152	352	205	88	18	12	8	6	11	25	979
2013	145	496	1.853	2.703	1.277	304	36	26	35	30	21	50	6.976
2014	261	1.656	7.388	19.886	10.195	1.262	54						40.702

Fonte: SINAN - DEVISA/Campinas
Obs: dados sujeitos à alterações.

Gráfico 6: Previsão de Casos De Dengue em Campinas-SP.



CONCLUSÃO

Com os experimentos realizados, foi possível projetar uma rede neural artificial do tipo Perceptron de múltiplas camadas capaz de prever casos de dengue a partir de dados históricos. A rede projetada foi capaz de detectar as tendências de aumento e redução de casos, inclusive para dados desconhecidos.

REFERÊNCIAS

Dicionário **MICHAELIS**. 1998-2009 Editora Melhoramentos Ltda. Disponível em: <<http://goo.gl/verto7>>. Acesso em: 10/05/2016.

HAYKIN, S.; LUGER, G. **Redes Neurais Artificiais Principios e Praticas**. Porto Alegre. 2. ed. Porto Alegre, Bookman, 2001.

LUIZ EDUARDO BORGES.; **Python para Desenvolvedores**. Rio de Janeiro: Edição do Autor, 2010, página 14. ISBN 978-85-90-94-51-1-6. p. 11-14.

Portal da Saúde – Disponível em: <<http://portalsaude.saude.gov.br/index.php/situacao-epidemiologica-dados-dengue>> Acesso em: 03/04/2016.

Pybrain – Disponível em: <<http://pybrain.org/docs/>> Acesso em: 02/06/2016. Python – Disponível em: <<https://docs.python.org/2/>> Acesso em: 23/04/2016.

SINAM/SES

<<http://portalsaude.saude.gov.br/images/pdf/2015/julho/29/Dengue-at---2014.pdf>> Acesso em: 03/04/2016.

TANENBAUM, ANDREW S.; **Sistemas Operacionais Modernos** – 3º edição. Pearson ISBN: 9788576052371. Capítulo 10, Disponível em: <<http://factus.bv3.digitalpages.com.br/users/publications/9788576052371/pages/447>>. Acesso em: 10/05/2016.