

ESTUDOS DAS IMPLICAÇÕES DO TEOREMA DE BAYES NA COMPUTAÇÃO NATURAL

STUDIES OF THE IMPLICATIONS OF THE BAYES THEOREM IN NATURAL COMPUTING

ESTUDIOS DE LAS IMPLICACIONES DEL TEOREMA DE BAYES EN LA COMPUTACIÓN NATURAL

Vinicius Alexandre de Oliveira Zevarex¹
Carlos Henrique da Silva Santos²

Resumo: Aprendizado de Máquina é um ramo da Inteligência Artificial, cujo principal objetivo é desenvolver sistemas que adquiriram conhecimentos automaticamente e os utilizem na resolução de problemas de classificação. Para tal, essa área demanda conhecimentos em matemática e estatística. Este artigo apresenta resultados dos estudos de duas técnicas de aprendizado de máquina, Naive Bayes e Redes Neurais Artificiais (RNA), expondo um levantamento bibliográfico sobre cada uma e testes com o algoritmo Naive Bayes para classificação de idiomas de textos e com um RNA treinada com dados de algoritmos de otimização.

Palavras-chave: Teorema de Bayes. Aprendizado de Máquina. Classificação. Redes Neurais Artificiais.

Abstract: Machine Learning is a branch of Artificial Intelligence, that the main objective is to develop system that learning automatically for to be applied on the classify problems. For this, this area requires mathematical and statistics knowledge. This article present results of two machine learning technical studies, Naive Bayes and Artificial Neural Network (ANN), exposing a biblical survey about each technical and tests with the algorithm Naive Bayes for language text classify and with a ANN trained with data of optimization algorithm.

Keywords: Bayes' Theorem. Machine Learning. Classify. Artificial Neural Networks.

Resumen: Aprendizaje de Máquina es una área de la Inteligencia Artificial, cuyo principal objetivo es desarrollar sistemas que adquirieron conocimientos automáticamente y utilizar como base en la resolución de problemas de clasificación. Para eso, esa área demanda conocimientos en matemática y estadística. Este artículo presenta resultados de los estudios de dos técnicas de aprendizaje de máquina, Naive Bayes y Redes Neuronales Artificiales (RNA), exponiendo un levantamiento bibliográfico sobre cada una y ensayos con el algoritmo Naive Bayes para clasificación de idiomas de textos y con una RNA entrenado con datos de algoritmos de optimización.

Palabras-clave: Teorema de Bayes. Aprendizaje de Máquina. Clasificación. Redes Neuronales Artificiales.

Envio 07/10/2019

Revisão 10/10/2019

Aceite 17/06/2020

¹Aluno do Curso Técnico em Informática Integrado ao Ensino médio. IFSP - Câmpus Itapetininga. E-mail: vinicius.zevarex2002@gmail.com, <http://lattes.cnpq.br/3467313059899689>.

²Professor do curso Técnico em Informática Integrado ao Ensino Médio. IFSP - Câmpus Itapetininga. E-mail: santos.chs@gmail.com, <http://lattes.cnpq.br/2704773293662530>

Introdução

De acordo com Monard et al. (2003), o Aprendizado de Máquina é um ramo da Inteligência Artificial que tem como objetivo desenvolver técnicas computacionais para a construção de sistemas capazes de adquirir conhecimento de forma automática e de tomar decisões utilizando como base experiências acumuladas na solução bem sucedida de problemas anteriores.

Assim, este artigo tem como intuito apresentar resultados de uma pesquisa de natureza exploratória e de cunho matemático-computacional que baseou-se no estudo de duas técnicas de aprendizado de máquina, o classificador Naive Bayes, construído a partir da inferência Bayesiana, e as Redes Neurais Artificiais que se inspiram no processo de aprendizado do cérebro humano para fundamentar o seu desenvolvimento. Esse projeto de pesquisa foi dividido em duas etapas principais, cada uma concentrou-se em uma técnica de aprendizado de máquina diferente. A primeira etapa a técnica estudada o Naive Bayes, e a segunda focou-se em Redes Neurais Artificiais. Para esses estudos foram realizadas pesquisas e levantamento bibliográficos nas bases de dados do Google Acadêmico e Scielo, com os códigos desenvolvidos neste trabalho sendo disponibilizados na plataforma GitHub³.

Nas seções a seguir são apresentados os resultados de estudos sobre o Teorema de Bayes e sua aplicação na classificação de textos de acordo com seu idioma, com um código disponibilizado por (LUKASZ,2018) que foi modificado para a classificação de instruções de sistemas operacionais, Windows ou Linux. Posteriormente, são expostos os resultados do desenvolvimento de uma RNA Perceptron de múltiplas camadas em Scilab⁴ para substituir o método numérico de um algoritmo de otimização, encontrando um valor *fitness* dado uma solução de uma determinada função, assim diminuindo o custo de processamento.

Teorema de Bayes

Esse Teorema surge com Thomas Bayes que nasceu em 1702 em Londres. Ele foi um importante matemático e reverendo da igreja presbiteriana, tendo estudado Teologia na

³ GitHub - plataforma de hospedagem de código-fonte com controle de versão disponível em: <https://github.com/>

⁴ Scilab – Software científico para computação numérica disponível em <https://www.scilab.org/>

Universidade de Edimburgo. Em 1731 assumiu a paróquia de Tunbridge Wells localizada em Kent, um condado situado no sudeste da Inglaterra a 58 km de Londres, onde morreu no ano de 1761. (PENA,2003).

Bayes publicou somente dois livros durante sua vida, o primeiro chamado “Benevolência divina” (1731) e o segundo “Uma introdução a doutrina dos fluxions”. O primeiro foi encontrado na Inglaterra e inicialmente foi considerado de autor anônimo, entretanto hoje o livro é creditado a ele. No segundo livro, ele defendia Isaac Newton das críticas feitas por George Berkeley, um filósofo irlandês da época. (ARA-SOUZA, 2010).

Dois anos após a morte de Thomas Bayes, o filósofo Richard Price (1723-1791) encontrou um artigo em meio aos papéis de Bayes com o nome “Ensaio buscando resolver um problema na doutrina das probabilidades”. Nesse artigo estava a demonstração do famoso Teorema de Bayes. Price acreditava que aquele artigo também tentava elucidar a existência de Deus (PENA,2003). A Figura 1 apresenta a única ilustração existente de Thomas Bayes:

Figura 1 - Única ilustração existente de Thomas Bayes



Fonte: Pena(2006, p. 24)

O raciocínio de Thomas Bayes apresentado em seu teorema tem como ponto de partida o conhecimento a priori do evento $P(A)$ e da probabilidade condicional $P(B|A)$ para calcular a probabilidade a posteriori $P(A|B)$. Para chegar no Teorema de Bayes, seguem-se as seguintes etapas expostas em Pena(2006, p. 28):

1. A probabilidade conjunta entre um evento A e um evento B é dada por:

$$P(A \cap B) = P(B)P(A|B) \quad (1)$$

2. Essa probabilidade conjunta também pode ser representada por:

$$P(B \cap A) = P(A)P(A|B) \quad (2)$$

3. Combinando as equações (1) e (2) tem-se:

$$P(B)P(A|B) = P(A)P(B|A) \quad (3)$$

4. Reorganizando a equação (3) e tem-se:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \quad (4)$$

5. Geralmente $P(B)$ não é conhecido, então deve-se utilizar uma formulação alternativa que é dada por (5):

$$P(B) = P(A \cap B) + P(A^c \cap B) \quad (5)$$

Em que

a. $P(A^c)$ é a probabilidade complementar de A, isto é, probabilidade de A não ocorrer.

6. Combinando as equações (2) e (5), tem-se (6):

$$P(B) = [P(A)P(B|A)] + [P(A^c)P(B|A)] \quad (6)$$

7. Logo, o Teorema de Bayes é descrito por (7):

$$P(A|B) = \frac{P(A)P(B|A)}{[P(A)P(B|A)] + [P(A^c)P(B|A^c)]} \quad (7)$$

8. Quando o espaço do evento é dado em termos de $P(A_j)$ e $P(B|A_j)$, então, pela lei da probabilidade total, para calcular a probabilidade de $P(B)$ é dada p

$$P(B) = \sum_j P(B|A_j)P(A_j) \quad (8)$$

9. Por fim, o Teorema de Bayes também é descrito por:

$$P(A_j|B) = \frac{P(B|A_j)P(A_j)}{\sum_j P(B|A_j)P(A_j)} \quad (9)$$

$P(A)$ e $P(A^c)$ compõem a probabilidade a priori, $P(B|A)$ e $P(B|A^c)$ a probabilidade condicional e $P(A|B)$ a probabilidade a posteriori.

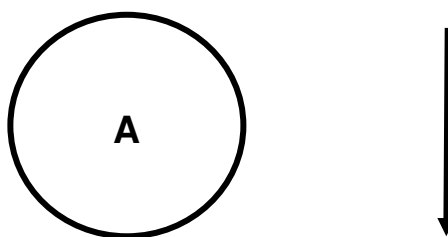
Redes Bayesianas

As Redes Bayesianas são técnicas que descrevem determinados problemas de forma probabilística e auxiliam na tomada de decisão. Ela consiste em uma combinação dos princípios da teoria de grafos, teoria de probabilidades, Ciência da Computação e Estatística (BEN-GAL, 2007 apud ARA-SOUZA, 2010).

Uma Rede Bayesiana representa um conjunto de variáveis $X = \{x_1, x_2, x_3, \dots, x_n\}$, em que cada uma dessas variáveis é representada por um nó, conforme mostrado na Figura 2. Nesse caso, as variáveis possuem dependências entre si que são representadas por arcos cíclicos direcionados, isto é, se existe um arco ligando $x_i \rightarrow x_j$, então x_i é pai de x_j (POLASTRO, 2012).

Cada nó da estrutura de uma Rede Bayesiana possui uma distribuição de probabilidade condicional $P(x_i|\text{pais}(x_i))$, que representa a probabilidade de x_i condicionada a seus pais x (POLASTRO, 2012). Os nós que não possui filhos são chamados de folhas, são eles que representam o final da Rede, já os nós que não possuem pai são chamados de raízes, pois representam o início da Rede. A Figura 2 apresenta os dois elementos que constituem uma Rede Bayesiana.

Figura 2 - Ilustração dos elementos que compõem uma rede bayesiana



Disponível em: Ara-Souza (2010).

As relações entre as variáveis de uma Rede Bayesiana são regidas pela propriedade de Markov, a qual diz que não existem dependências diretas entre as variáveis que não estão

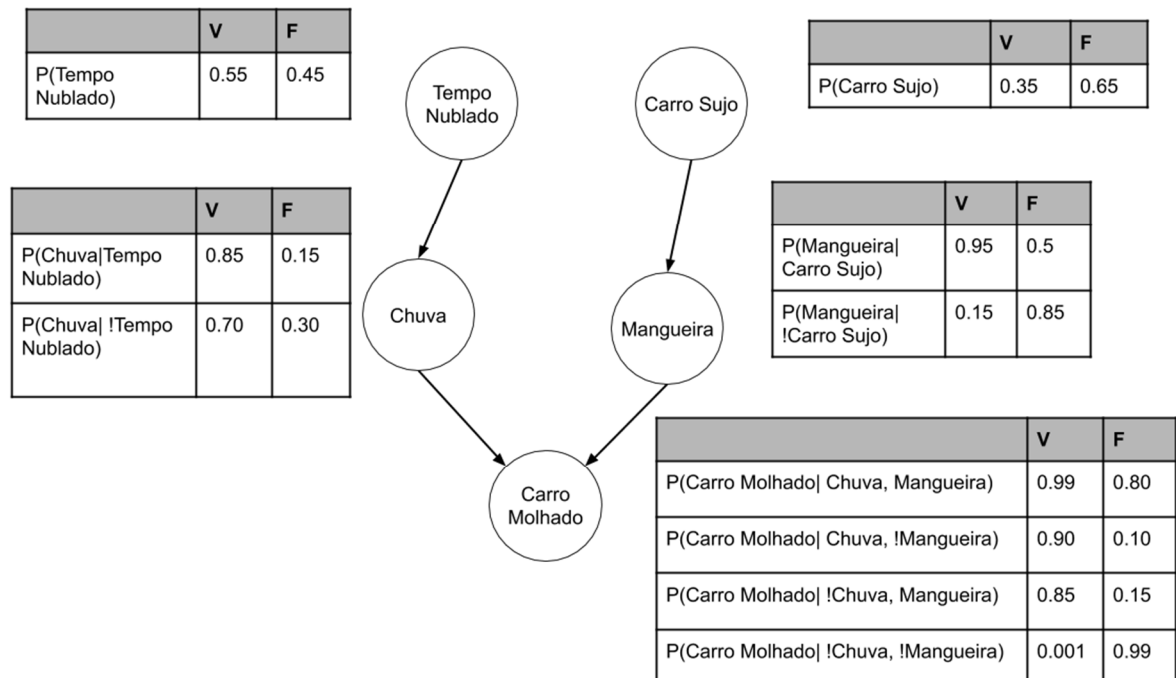
explícitas pela representação orientadas dos arcos, isto é, cada variável possui dependência direta apenas com sua(s) variável (eis) pai (ARA-SOUZA, 2010).

Essa propriedade permite calcular a distribuição de probabilidade conjunta de todas as variáveis x_1, \dots, x_n de uma Rede Bayesiana utilizando a equação (8), em que $P(x_1, x_2, \dots, x_n)$ representa a probabilidade conjunta de todas ou algumas variáveis da Rede e $P(x_i | \text{pais}(x_i))$ a probabilidade de uma determinada variável x_i condicionada a sua(s) variável(eis) pai(s):

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{pais}(x_i)) \quad (8)$$

A Figura 3, apresenta um exemplo de um problema representado em uma Rede Bayesiana. Este exemplo possui 5 variáveis, sendo as variáveis Tempo Nublado e Carro sujo raízes e a variável Carro Molhado folha.

Figura 3 - Estrutura em Grafo de uma Rede Bayesiana Simples

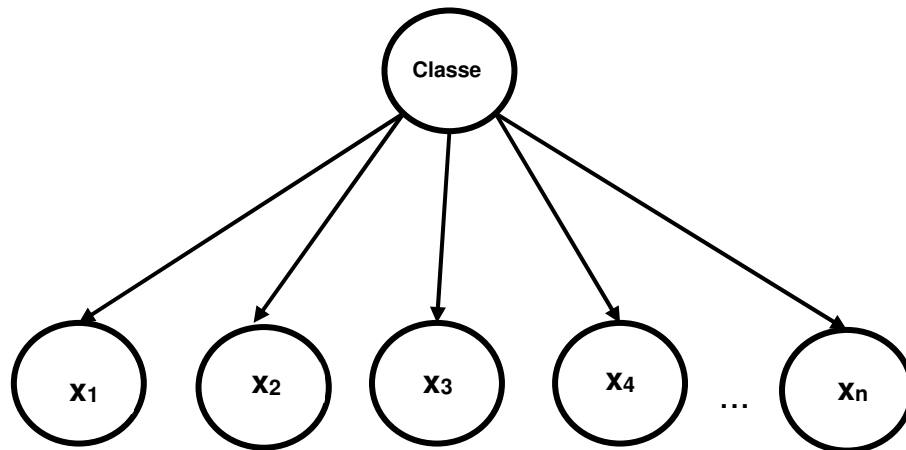


Fonte: Seffrin (2013, apud Millán et al. 2010)

Naive Bayes

O Naive Bayes é um classificador do aprendizado de máquina supervisionado, ele consiste em uma Rede Bayesiana de estrutura simples, trazendo maior destreza em aplicá-lo em problemas de classificação (SANTOS,2007). Esse classificador é chamado de ingênuo (*Naive*), pois sua estrutura assume que as variáveis são independentes entre si, dado o valor do atributo classe, em outras palavras, todas as variáveis folhas possuem somente relação de dependência com a variável classe, considerada variável raiz, como mostrado na Figura 4:

Figura 4 - Estrutura em grafo do classificador Naive Bayes



Fonte: Santos (2007)

Na Figura 4 as variáveis $x_1, x_2, x_3, \dots, x_n$ são as novas instâncias que são folhas da Rede. Na prática, a classificação com o Naive Bayes tem como ponto de partida uma base com um conjunto de dados já rotulados que são utilizados como alicerce na classificação de novas instâncias. Quando o classificador se depara com um novo conjunto de dados de rótulos desconhecidos, esses novos dados são comparados com os que já são conhecidos.

Assim, com base nos dados já conhecidos, foi implementada uma Rede Bayesiana, representada na Figura 4 para cada classe. Por fim, calcula-se a probabilidade de todas as possíveis classes e é selecionada como rótulo da nova instância a que possui maior probabilidade (PARDO et al, 2002). Para calcular tais probabilidades utiliza-se a equação (9):

$$P(Classe|a_1, \dots, a_n) = \prod_{i=1}^n P(a_i|classe)P(Classe) \quad (9)$$

Teste do classificador Naive Bayes em reconhecimento de idiomas de textos

Visando entender o funcionamento do Naive Bayes na prática, realizou-se testes com um algoritmo implementado em PHP disponibilizado por usuário da plataforma GitHub em Naive Bayes Classifier for PHP (2018), cujo a aplicação é a classificação de textos de acordo com seu idioma. No algoritmo identificou-se duas estruturas principais, sendo a de treinamento e a de classificação.

A estrutura de treinamento tem por finalidade ensinar o algoritmo a reconhecer determinados idiomas, para tal é fornecido como entrada textos com seus respectivos idiomas. Esses textos são divididos em palavras e organiza-se um vocabulário, separando-as por idiomas e calculando seu grau de frequência na base de dados para utilizar nos cálculos probabilísticos.

Na estrutura de classificação, é fornecido uma frase ou um documento de texto como entrada para o algoritmo que é fragmentado em palavras para sua análise. Com essas palavras, organiza-se uma Rede Bayesiana, com a estrutura da Figura 4, para cada idioma, sendo o idioma a variável raiz e as palavras as folhas.

Com os números obtidos na função de treinamento, é calculado a probabilidade conjunta de cada Rede Bayesiana, utilizando a equação (9), a qual é escolhida como rótulo a classe com maior probabilidade. As Figura 9 apresenta as entradas das funções de treinamento e a Figura 10 as entradas e as saídas da função de classificação.

Figura 9 - Chamada da função de treinamento

```
//Fornecendo documentos para o treinamento do classificador,  
$nbc->train(new FileDataSource('polish.txt'), 'polish'); //Ch  
dizendo qual a classe do mesmo  
$nbc->train(new FileDataSource('english.txt'), 'english');  
$nbc->train(new FileDataSource('japanese.txt'), 'japanese');  
$nbc->train(new FileDataSource('german.txt'), 'german');
```

Fonte: Autoria própria

Figura 10 - Dados de entrada e classificação feita pelo algoritmo.

```

echo $nbc->classify("This color is very bright.") . "<br>";
echo $nbc->classify("人生の意味は何ですか?") . "<br>";
echo $nbc->classify("Byłem ostatni w kolejce") . "<br>";
echo $nbc->classify("Ich mag es wirklich, am Bundesinstitut zu studieren") . "<br>";

```

localhost/PHPNaiveBayes x

localhost/PHPNaiveBayesClassifier-master_IDIOMAS/examples/test.php

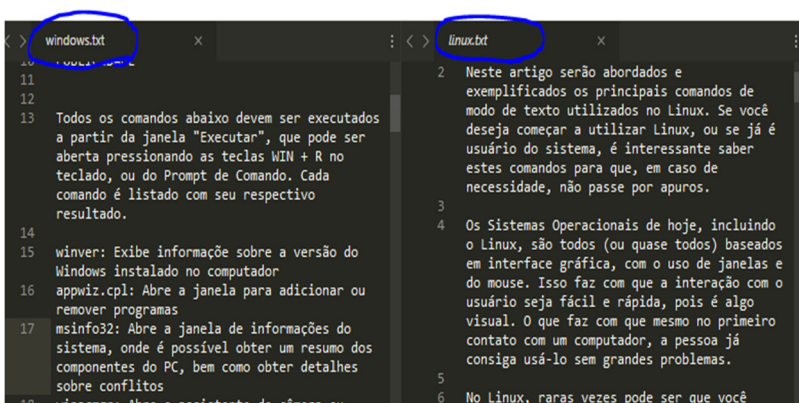
english
japanese
polish
german

Fonte: Autoria própria

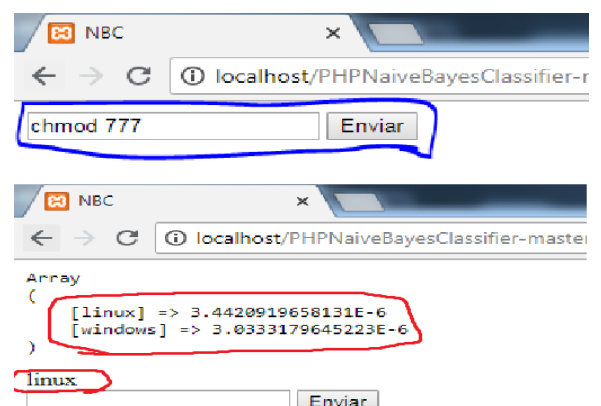
Após o entendimento da estrutura do algoritmo, realizaram-se testes para analisar o desempenho do algoritmo, classificando comandos de acordo com seu Sistema Operacional de origem, prompt Windows ou Terminal Linux. Para isso, na estrutura de treinamento forneceu-se textos encontrados na Internet sobre esses dois sistemas operacionais, que foram utilizados para aumentar o vocabulário do algoritmo com palavras relacionadas a esses Sistemas Operacionais (Figura 11a). No teste estabeleceu-se como entrada o comando “chmod 777”, que foi dividido em palavras e analisado a frequência delas no vocabulário de cada classe conhecida pelo algoritmo, assim calculou-se as probabilidades para a classe “Linux” e “Windows”, classificando corretamente a entrada pertencente a primeira.

66

Figura 11 - Dados de entrada e classificação feita pelo algoritmo



(a)



(b)

A complexidade de processamento das Redes Neurais Biológicas e Estrutura Morfológica do neurônio

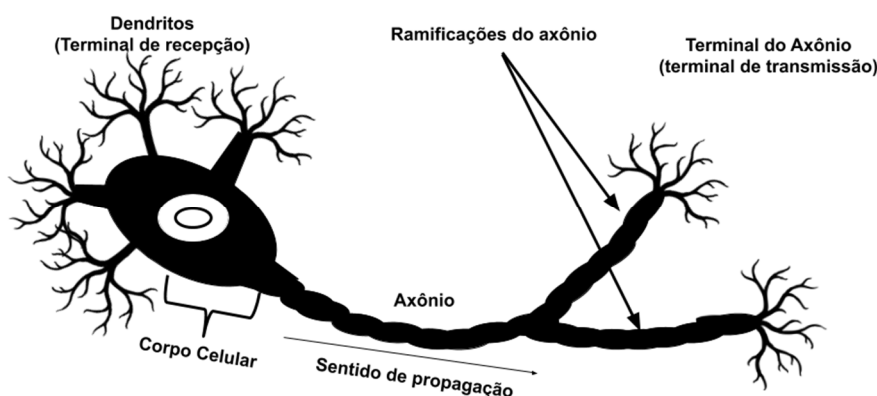
Segundo Silva (2003), o cérebro humano pode ser comparado a um computador extremamente complexo, não-linear e paralelo, pois ele tem a capacidade de organizar sua estrutura neural para realizar seus processamentos de forma muito eficiente.

O cérebro biológico é constituído por um conjunto de inúmeras células nervosas, denominadas neurônios. Acredita-se que em um único cérebro existem aproximadamente 15 bilhões destas células, o neurônio destaca-se das outras células pela sua capacidade de processar informações, sendo que o cérebro organiza essas células para realizar esse processamento com maior eficiência (SANTANA, 2012).

A estrutura morfológica do neurônio é composta por quatro elementos, os dendritos, corpo celular, axônio e terminais do axônio. Os dendritos são prolongamentos em forma de ramificações responsáveis por receber impulsos elétricos, o corpo celular presta auxílio metabólico à toda célula, o axônio é um prolongamento único responsável pelo transporte dos sinais dos dendritos e os terminais do axônio enviam sinais para os outros neurônios (SANTANA, 2012). A figura 5 é uma ilustração da estrutura morfológica do neurônio biológico:

67

Figura 5 - Estrutura do Neurônio Biológico



Fonte: Santos (2007)

Quando um neurônio biológico recebe um estímulo ocorre a despolarização da membrana, ocasionando a inversão de cargas, em que no lado externo tem-se cargas negativas

e no lado interno positivos. Então ocorre a propagação de um impulso elétrico em cada direção do ponto despolarizado, que se mantém até chegar às extremidades da fibra. O sentido do impulso nervoso é dos dendritos, corpo celular e axônio, até chegar à região conectada com outro neurônio. Esse impulso elétrico que percorre a rede neural pode excitar ou inibir outros neurônios, a eficiência das transmissões de pulsos elétricos podem ser modificadas, essas mudanças de acordo com os sinais externos representa o processo de aprendizagem. (SILVA,2003)

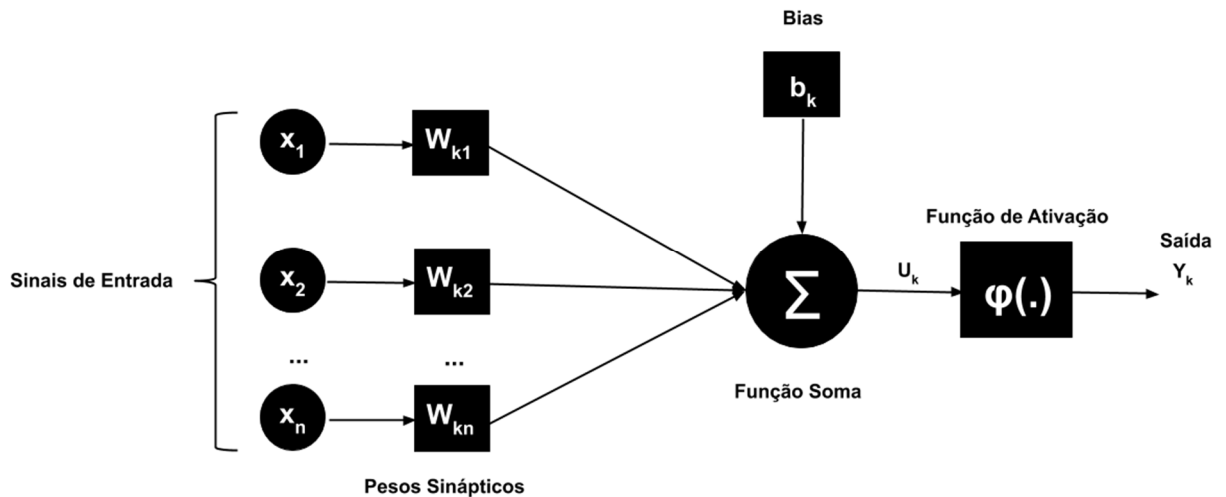
Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA), se trata de uma técnica de aprendizagem de máquina que utiliza um modelo matemático inspirado na complexidade do cérebro e estrutura neural dos seres humanos, ela tenta modelar a maneira em que o cérebro aprende e realiza determinadas tarefas. Conforme Santana(2012), as RNAs são aplicadas em diversos contextos, sendo eles em problemas de classificação, regressão, aproximação de funções e reconhecimento de padrões.

A unidade básica de uma Rede Neural Artificial são os neurônios matemáticos, eles trabalham em conjunto no processo de aprendizagem, cada uma dessa unidade apresenta um comportamento específico de entrada e saída, determinado pela sua função de ativação, conexões com os outros neurônios e pelas entradas externas. Segundo Silva(2005), os três elementos básicos de um neurônio artificial são: os conjunto de conexões entre os neurônios, o combinador linear e a função de ativação.

O conjunto de conexões entre os neurônios também são conhecidas como sinapses e é responsável por propagar e ponderar os valores de entrada entre os neurônios, assim, uma sinapse j conectada a um neurônio k recebe uma entrada (x_j) que é multiplicada pelo peso (W_{kj}) . Por definição, o combinador linear se trata de uma expressão construída a partir de um conjunto de elementos multiplicados por uma constante, sendo ele responsável por realizar a somatória dos sinais de entrada (elementos) ponderados pelos pesos sinápticos do neurônio (constantes) com o bias (b_k) , cujo valor é fixo para cada neurônio. Função de ativação tem como objetivo limitar e determinar a saída do neurônio baseando-se resultado da somatória do combinador linear. A Figura 6 é uma representação gráfica da estrutura de um Neurônio Artificial:

Figura 6 - Representação do Neurônio Artificial



Fonte: Silva(2005)

A saída de um neurônio pode ser definida, matematicamente, pela seguinte equação:

$$y_k = \varphi\left(\sum_{j=1}^n w_{kj}x_j + b_k\right) \quad (10)$$

Em que,

n - números de entrada;

y_k - saída de um neurônio k ;

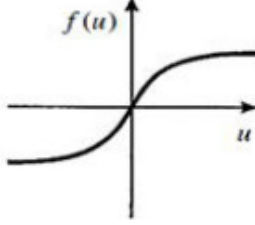
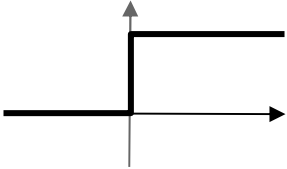
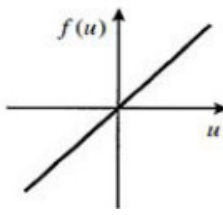
φ - função de ativação;

w_{kj} - pesos sinápticos de uma sinapse j que conecta um neurônio k ;

b_k - bias de um neurônio k .

Existem diversos tipos funções de ativação, em que as mais difundidas, segundo Santana(2012), são a Sigmóide, Limiar e a Linear, apresentadas no quadro 1:

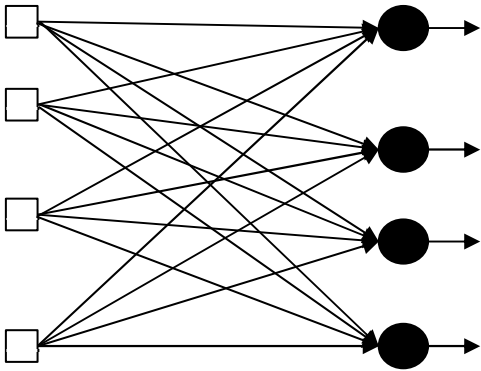
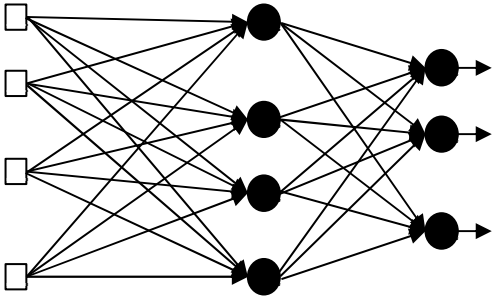
Quadro 1 -Funções de ativação para RNAs mais difundidas

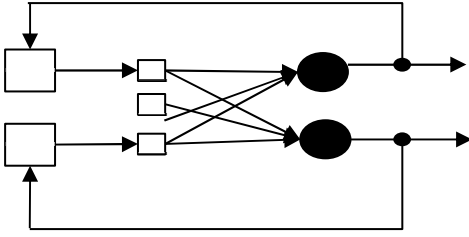
Funções de ativação		
Nome	Fórmula	Gráficos
Sigmóide	$f(u) = \frac{1}{1+\exp(-a \cdot u)}$ Ou $f(u) = \tanh(u)$	
Limiar	$f(u) = \begin{cases} 1 & \rightarrow u \geq 0 \\ 0 & \rightarrow u < 0 \end{cases}$	
Linear	$f(u) = au + b$	

Arquitetura de Redes Neurais

O que define uma arquitetura de uma RNA é a organização dos neurônios, os tipos de arquiteturas estão inteiramente ligados com o algoritmo de aprendizagem utilizado para o treinamento e ao problema no qual se deseja solucionar (SILVA,2003). As Redes Neurais são representadas por grafos direcionados, semelhantemente às Redes Bayesianas, em cada nó representa um neurônio e as arestas representam as entradas ou as saídas. Segundo Haynkin (2009, apud SANTANA, 2012), as classes de arquitetura mais difundidas são: Redes *feedforward* de única camada, Redes *feedforward* de múltiplas camadas e Redes recorrentes, apresentadas no Quadro 2.

Quadro 2 - Arquiteturas de RNAs mais difundidas

Arquitetura	Representação	Descrição
<p>Redes feedforward de única camada</p>	 <p>Fonte: Silva(2005)</p>	<p>Essa arquitetura possui somente uma camada que processa as entradas e realiza as saídas. O termo <i>feedforward</i> significa que a rede possui somente um sentido de propagação dos sinais de entrada, não havendo ciclos no processamento.</p>
<p>Redes feedforward de múltiplas camadas</p>	 <p>Fonte: Silva(2005)</p>	<p>Diferente da arquitetura de única camada, a de múltiplas camadas possui uma ou mais camadas de neurônios escondida que é responsável por fazer um processamento antes de propagar os sinais para a camada de saída.</p>

<p>Redes recorrentes</p>	 <p>Fonte: Silva(2005)</p>	<p>Uma arquitetura é denominada recorrente, quando possui ao menos um laço responsável por realimentar a saída de neurônios para os outros neurônios.</p>
------------------------------	--	---

Fonte: Autoria própria

Algoritmo de Treinamento *Backpropagation*

Para o treinamento de uma Rede Neural Artificial, é necessário montar um conjunto de dados já rotulados do problema que se deseja resolver, sendo com o estímulos que a RNA recebe desses dados ocorre o ajuste dos pesos sinápticos (SANTANA,2012). Em suma, o algoritmo de treinamento *Backpropagation* é uma técnica de treinamento do tipo supervisionada, em que se fornece entradas para a RNA, calcula-se o erro e faz-se um caminho inverso ajustando os pesos.

A primeira etapa, chamada de *forward*, consiste em iniciar os pesos sinápticos aleatoriamente. Em seguida, fornece-se à RNA um vetor de entrada com suas respectivas saídas desejadas, que propaga-se por todas as camadas, gerando uma saída que é comparada com a desejada para calcular uma taxa de erro. (BARCA et al., 2005), (FERREIRA et al, 2018)

Na segunda etapa, chamada de *backward*, é feito o sentido inverso da RNA, isto é, a taxa de erro se propaga pela rede no sentido: camadas de saída, camadas intermediárias e camadas de entrada. Com esse estímulo realizado pela taxa de erro, realiza-se o ajuste dos pesos sinápticos para que a saída da RNA seja o mais próxima possível do desejado. Essas duas etapas são realizadas repetidamente até que a taxa de erro seja inexistente ou o número máximo de repetições seja atingido. (BARCA et al., 2005)

Rede Neural Perceptron de Múltiplas Camadas para indução de valor *Fitness* de um conjunto de soluções de determinada função em um algoritmo de otimização

Depois dos levantamento bibliográfico sobre a teoria por trás das Redes Neurais Artificiais, desenvolveu-se uma RNA Perceptron de Múltiplas Camadas em Scilab. O objetivo da RNA é substituir o método numérico para encontrar valores *fitness* em um algoritmo de otimização, isto é, dado um vetor de soluções a RNA deve induzir qual será o melhor valor desse conjunto de soluções. Para isso, utilizou-se dados gerados pelo algoritmo de otimização *Cuckoo Search*, apresentado em (SILVA-SANTOS et al, 2020). A entrada é um vetor de valores que representam as soluções de uma determinada função e a saída é o melhor valor encontrado em meio essas soluções. A arquitetura da RNA é 2-4-1 e a técnica de treinamento utilizada foi o *BackPropagation*. O Quadro 3 apresenta as funções objetivo, os gráficos da otimização do algoritmo de otimização e os gráficos de média de erro por iteração no treinamento, tomando-se como base práticas de testes apresentados em (BRIANEZE et al, 2007), (BRIANEZE et al, 2009) e (SANTOS, 2010).

Quadro 3 - Treinamento da RNA com dados de algoritmo de otimização Cuckoo Search

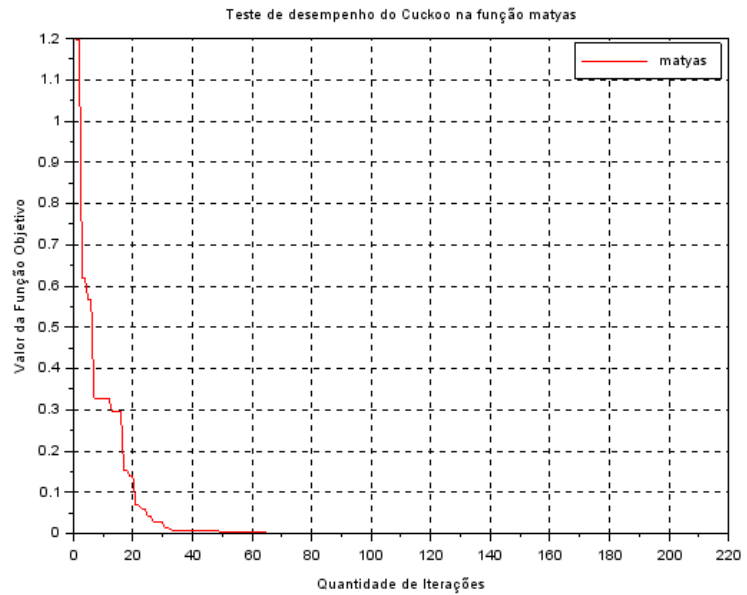
<p>Função Ackley</p>	<p>Definida Por:</p>	$f(x) = -a \cdot \exp \left(-b \sqrt{\frac{1}{d} \cdot \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \cdot \sum_{i=1}^d \cos(cx_i) \right) + a$ <p>+ exp (1) Fonte: (SURJANOVIC; BINGHAM, 2017)</p>
	<p>Gráfico de otimização do Algoritmo Cuckoo:</p>	<p>Teste de desempenho do Cuckoo na função ackley</p>
	<p>Gráfico de Média dos Erros da RNA por Iteração no treinamento</p>	
<p>Função</p>	<p>Definida Por:</p>	



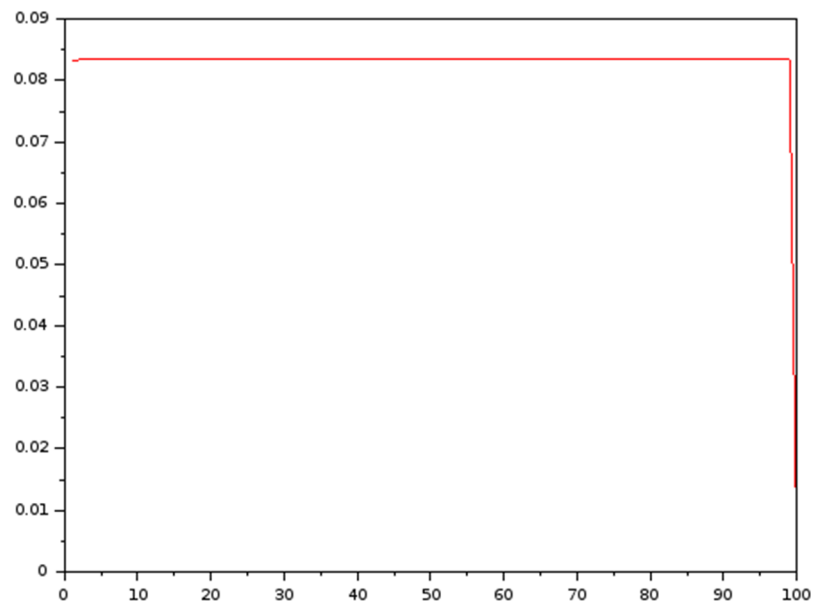
<p>Zakharov</p>		$f(x) = \sum_{i=1}^d x_i^2 + \left(\sum_{i=1}^d 0.5 \cdot ix_i \right)^2 + \left(\sum_{i=1}^d 0.5 \cdot ix_i \right)^4$ <p>Fonte: (SURJANOVIC; BINGHAM, 2017)</p>
	<p>Gráfico de otimização do Algoritmo Cuckoo</p>	<p>Teste de desempenho do Cuckoo na função zakharov</p>
	<p>Gráfico de Média dos Erros da RNA por Iteração no treinamento</p>	
<p>Função Matyas</p>	<p>Definida Por:</p>	$f(x) = 0.26 \cdot (x_1^2 + x_2^2) - 0.48 \cdot x_1 x_2$ <p>Fonte: (SURJANOVIC; BINGHAM, 2017)</p>



**Gráfico de
otimização do
Algoritmo
Cuckoo**



**Gráfico de
Média dos
Erros da RNA
por Iteração no
treinamento**



Fonte: Autoria própria

Considerações finais

Este artigo tem como intuito apresentar os estudos realizados sobre a implicação do teorema de Bayes na Computação Natural, cujo os objetivos são: fazer um levantamento bibliográfico sobre o Teorema de Bayes e sua associação nas técnicas de aprendizado de máquina, realizar estudos sobre a técnica de Redes Neurais Artificiais e realizar comparações entre essas duas técnicas.

Os dois conceitos de aprendizado de máquinas estudados nesse projeto de pesquisa, apesar de utilizarem técnicas matemáticas muito díspares, pôde se identificar um padrão entre eles. Ambos, são classificadores de aprendizado supervisionado, entretanto a maneira como esses dois algoritmos exercem o método indutivo são extremamente diferentes, enquanto um utiliza de métodos de probabilidade e inferência bayesiana, o outro se inspira na estrutura das Redes Neurais do Cérebro Biológico.

Percebeu-se nos testes realizados, que os algoritmo inspirado na inferência Bayesiana consegue ter uma maior precisão quando possui uma maior quantidade de dados de treinamento, já o *Perceptron* Multicamadas depende da qualidade dos dados, isto é, se o número de amostras for grande, mas forem repetitivas a RNA não terá um bom treinamento.

Para atender a essas demandas, esse grupo vem trabalhando em técnicas alternativas para potencializar a qualidade de dados e aprimorar o treinamento das redes neurais, desde a concepção de estudos quanto ao controle da população inicial (SILVA, SILVA-SANTOS, 2019) e quanto em modelos híbridos de heurísticas (FERRAZ JR., SILVA-SANTOS, GONÇALVES, 2020).

Agradecimentos

Os autores agradecem o IFSP - Câmpus Itapetininga pelo apoio financeiro (PIBIFSP/2017) e o Aluno João Victor Ferreira de Moraes pelos dados disponibilizados. Os autores também agradecem o IFSP pelo suporte financeiro via edital PRP-226/2016.

Referências

ARA-SOUZA, Anderson Luiz. **REDES BAYESIANAS: UMA INTRODUÇÃO APLICADA A CREDIT SCORING**. São Carlos, 2010.

BARCA, Maria Carolina Stockler; SILVEIRA, Tiago Redondo de Siqueira; MAGINI, Marcio. Treinamento de redes neurais artificiais: o algoritmo Backpropagation. **IX Encontro Latino Americano de Iniciação Científica, V Encontro Latino Americano de Pós-Graduação- Universidade do Vale do Paraíba, Anais. Jacareí, 2005.**

BRIANEZE, Juliano Rodrigues; DA SILVA SANTOS, Carlos Henrique; HERNANDEZ-FIGUEROA, Hugo Enrique. Evolutionary algorithms applied to microstrip antennas design. 2007.

BRIANEZE, Juliano Rodrigues; DA SILVA-SANTOS, Carlos Henrique; HERNÁNDEZ-FIGUEROA, Hugo Enrique. Multiobjective evolutionary algorithms applied to microstrip antennas design. **Ingeniare. Revista chilena de ingeniería**, v. 17, n. 3, p. 288-298, 2009.

FERRAZ JR., Wilton Moreira; SILVA-SANTOS, Carlos Henrique; GONÇALVES, Marcos Sérgio, Filtros ópticos otimizados por algoritmos genéticos associados à lógica nebulosa, XIII Encontro Acadêmico de Modelagem Computacional Laboratório Nacional de Computação Científica – LNCC/MCTIC (EAMC-2020), Petrópolis-RJ, 2020, Disponível em: <http://www.eamc.lncc.br/PastEditions/Proceedings/Proceedings_EAMC2020.pdf>, Acessado em: 07 de Maio de 2020.

FERREIRA, Adriano da Silva et al. Towards an integrated evolutionary strategy and artificial neural network computational tool for designing photonic coupler devices. **Applied Soft Computing**, v. 65, p. 1-11, 2018.

NAIVE Bayes Classifier for PHP. Disponível em: <https://github.com/uhho/PHPNaiveBayesClassifier>. Acesso em: 14 ago. 2018.

MILLÁN, Eva; LOBODA, Tomasz; PÉREZ-DE-LA-CRUZ, Jose Luis. Bayesian networks for student model engineering. **Computers & Education**, v. 55, n. 4, p. 1663-1683, 2010.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, v. 1, n. 1, p. 32, 2003.

PARDO, Thiago Alexandre Salgueiro; NUNES, Maria das Graças Volpe. Aprendizado Bayesiano Aplicado ao Processamento de Línguas Naturais. **Série de Relatórios Técnicos do Instituto de Ciências Matemáticas e de Computação-ICMC, Universidade de São Paulo**, n. 180, 2002.

PENA, Sérgio Danilo. Thomas Bayes O "cara"! **Ciência Hoje**, Minas Gerais, v. 38, p.24-29, jun. 2006

POLASTRO, Rodrigo Bellizia. **Lógica Probabilística Baseada em Redes Bayesianas Relacionais com inferência em Primeira Ordem**. 2012. 105 f. Tese (Doutorado) - Curso de Engenharia Mecânica, Escola Politécnica da Universidade de São Paulo, São Paulo, 2012.

SANTANA, Alan Felipe. **TREINAMENTO DE REDES NEURAS ARTIFICIAIS UTILIZANDO ALGORITMOS GENÉTICOS EM PLATAFORMA DISTRIBUÍDA**. 2012. 123 f. Monografia (Especialização) - Curso de Sistemas de Informação, Universidade Federal de Lavras, Lavras, 2012.

SANTOS, Carlos Henrique da Silva et al. Computação bio-inspirada e paralela para a análise de estruturas metamateriais em microondas e fotônica. 2010.

SANTOS, Edimilson Batista dos. **A Ordenação das Variáveis no Processo de Otimização de Classificadores Bayesianos: Uma Abordagem Evolutiva**. 2007. 114 f. Dissertação (Mestrado) - Curso de Ciência da Computação A, Universidade Federal de São Carlos, São Carlos, 2007.

SEFFRIN, Henrique M.; RUBI, Geiseane; JAQUES, Patricia. Uma Rede Bayesiana aplicada a Modelagem do Conhecimento Algebrico do Aprendiz. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2013. p. 597.

SILVA-SANTOS, C. H. et al. Purification of naphthalene by zone refining: Mathematical modelling and optimization by swarm intelligence-based techniques. **Separation and Purification Technology**, v. 234, p. 116089, 2020.

SILVA, Denis Tavares da; SILVA-SANTOS, Carlos Henrique. Controlando população inicial de algoritmos genéticos para a otimização de funções. **Revista Brasileira de Iniciação Científica**, v. 6, n. 2, p. 68-91, 2019.

SILVA, Juliana Mendes N. Redes Neurais Artificiais: Rede Hopfield e Redes Estocásticas. 2003.

SOUZA, Marcos Castro de. **Python - Implementação da rede neural Perceptron**. Disponível em: <<https://gist.github.com/marcoscastro/491bd5837815fe11181dce6c50f457ee>>. Acesso em: 30 ago. 2018.