

Redes Neurais Artificiais

Clássica Flipcard Revista Mosaico Menu Lateral Fotografia Linha Do Tempo



Simulação do comporta...

Cursos online gratuitos ...



IBM e Computação Cog...

OCR e Javascript - Res...

Rede neural não tão arti...

Perceptron - Uma...

1



Simulação de es...

2

Video Aulas de Inteligên...

Download Ebooks de R...



Implementação de XOR...

Identificação de bovinos

RNA e o Setor de Trans...

Deteção de defeitos e...

O primeiro model...

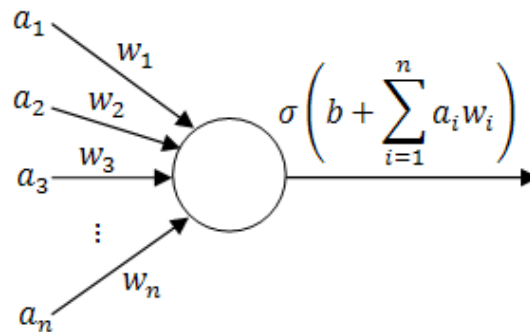
1

Perceptron - Uma breve explicação

O perceptron é o tipo mais básico de [rede neural](http://redesneuraisartificiais.blogspot.com/2010/10/o-primeiro-modelo-de-um-neuronio-criado.html) [http://redesneuraisartificiais.blogspot.com/2010/10/o-primeiro-modelo-de-um-neuronio-criado.html]. O primeiro modelo de uma rede neural artificial demonstrou matematicamente a possibilidade de simulação do aprendizado cerebral. Este modelo simulava um neurônio com entradas e pesos que, ajustados teriam o "poder" de aprender a se comportar de determinada forma. Entretanto, este modelo ainda era muito fraco e era preciso que se conhecessem os pesos para se encontrar determinada saída, era preciso criar uma forma de gerar os pesos algoritmicamente.

Um algoritmo de "ajuste automático" destes pesos foi criado em 1957 por Frank Rosenblatt e funciona da seguinte forma: o usuário fornece as entradas e as saídas esperadas e os pesos recebem valores iniciais (geralmente zero), as entradas são multiplicadas aos pesos e depois tudo é somado e avaliado com um valor de limite, caso esta soma seja maior que este valor, o resultado é 1, caso contrário, -1.

A imagem abaixo representa de forma fiel um Perceptron, que nada mais é que um neurônio de MCP:



[<http://blog.zabarauskas.com/img/perceptron.gif>]

Os pesos (w_i) são multiplicados com as entradas (a_i) e a soma dos dados é aplicada a uma função σ .
Tema Visualizações dinâmicas. Tecnologia do [Blogger](#).

geralmente uma destas três:

- *Função sigmóide*: $\sigma(x) = \frac{1}{1+\exp(-x)}$,
- *Tangente Hiperbólica*: $\sigma(x) = \tanh(x)$,
- *Linear simples*: $\sigma(x) = x$

Se o resultado de x for positivo, o resultado é 1, caso contrário, -1.

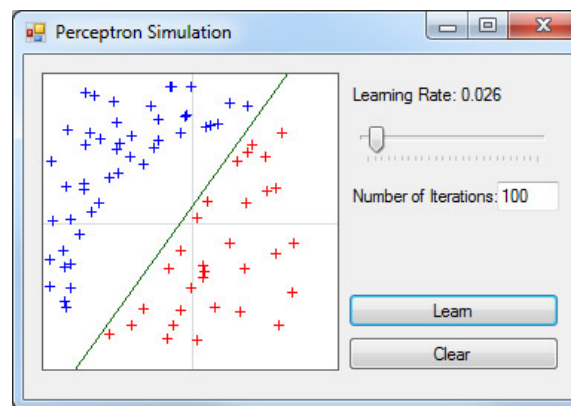
O perceptron, então é ajustado de acordo com a fórmula abaixo:

$$w_i(t+1) = w_i(t) + \alpha(d_j - y_j(t))x_{j,i}$$

[<http://upload.wikimedia.org/math/a/0/4/a04f856c6b62711012dcbf0ac6c918b6.png>]

$w_i(t+1)$ é o novo peso, que é ajustado levando em conta: o peso anterior e o erro. O erro é calculado pela diferença entre o valor esperado (d_j) e o valor encontrado (y_j), multiplicado pela entrada associada àquele peso e α , que é um valor chamado de taxa de aprendizado. A taxa de aprendizado, geralmente com um valor muito pequeno (como 0,000001) ajuda o perceptron a não "fugir" muito da solução ideal.

Este tipo de rede neural é muito utilizado para classificar dados em dois conjuntos (1 ou -1), uma representação bastante utilizada é um gráfico com uma linha divisória, separando os tipos de dados diferentes.



[http://www.codeproject.com/KB/recipes/single_layer_perceptron/PerceptronForm.jpg]

Tema Visualizações dinâmicas. Tecnologia do [Blogger](#).

A inclinação da reta na imagem acima depende da taxa de aprendizagem.
Portanto, é possível encontrar maior ou menor precisão de acordo com o ajuste da mesma.

Neste [site \[http://www.codeproject.com/KB/recipes/single_layer_perceptron.aspx\]](http://www.codeproject.com/KB/recipes/single_layer_perceptron.aspx) , é possível encontrar uma implementação deste perceptron em C#

Postado há 13th June 2011 por Anonymous

Marcadores: [clojure](#), [implementação](#), [lisp](#), [or](#), [perceptron](#)

1 Visualizar comentários



Alex Sandro Batista Pereira 29 de setembro de 2017 02:43

melhor explicação que achei em 3 dias de pesquisa obrigado ..rs

[Responder](#)

Digite seu comentário...

Comentar como:

Publicar

Visualizar

