Introdução a Inteligência Artificial

Redes Neurais Artificiais

PROFA. LETICIA T. M. ZOBY

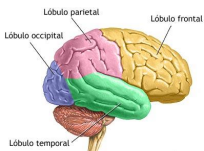
LETICIA.ZOBY@IESB.EDU.BR

Introdução

Na busca pela construção de máquinas inteligentes, ou com comportamento inteligente, um modelo que ocorre naturalmente é o do cérebro humano.

A partir dessas motivações, o desenvolvimento das Redes Neurais Artificiais (RNAs) tomou como inspiração a estrutura e o funcionamento do sistema nervoso, com o objetivo de simular a capacidade de aprendizado do cérebro humano na aquisição de conhecimento.

No cérebro, o **comportamento inteligente** é uma propriedade emergente de um grande número de **unidades simples** (ao contrário do que acontece com regras e algoritmos simbólicos).

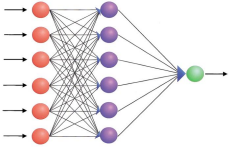


Introdução

**Redes Neurais** podem ser consideradas um paradigma diferente de computação.

Inspirado na **arquitetura paralela** do cérebro humano.

◦ Elementos de processamento simples.

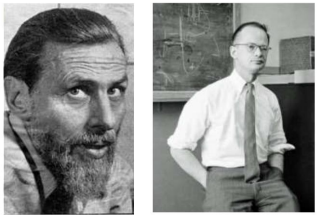
◦ Grande grau de interconexões. 

◦ Interação adaptativa entre os elementos.

Introdução

Os estudos pioneiros na área foram realizados por McCulloch e Pitts.

Em 1943, eles propuseram um modelo matemático de neurônio artificial, a unidade lógica com limiar (LTU, do inglês Logic Threshold Unit), que podia executar funções lógicas simples.



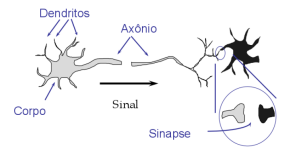
Warren McCulloch Walter Pitts

Introdução

**Estrutura de um Neurônio:**

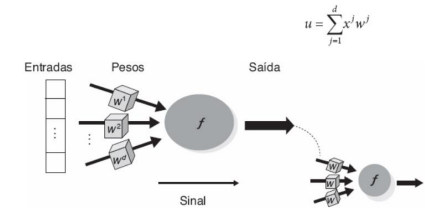
◦ O **sistema nervoso**, do qual faz parte o cérebro, é um conjunto complexo de células que determinam o funcionamento e o comportamento dos seres vivos.

◦ A unidade fundamental do sistema nervoso é a célula nervosa, o **neurônio**, que se distingue das outras células por apresentar excitabilidade, que lhe permite responder a estímulos externos e internos. Isso possibilita a transmissão de impulsos nervosos a outros neurônios e a células musculares e glandulares.

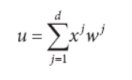
◦ O principal bloco de construção do cérebro é o neurônio. Os principais componentes de um neurônio são: **dendritos, corpo celular e axônio**.

Introdução

**Elementos Básicos de um Neurônio Artificial**

◦ Uma RNA é, portanto, caracterizada por dois aspectos básicos: arquitetura e aprendizado. 

Modelo simplificado de neurônio artificial (Haykin 1999 apud )

Introdução 

**Elementos Básicos de um Neurônio Artificial**

**Entradas:**

◦ As entradas podem ser as saídas de outros neurônios, entradas extrenas, um **bias** ou qualquer combinação destes elementos.

◦ Bias (Tendência) é um erro sistemático diferentemente de erro aleatório. Um ou mais componentes do erro sistemático podem contribuir para a tendência.

**Somador:**

◦ O somatório de todas estas entradas, multiplicadas por suas respectivas forças de conexão sináptica (os pesos)

**Saída:**

◦ A saída de um neurônio é definida por meio da aplicação de uma função de ativação à entrada total

Os neurônios podem apresentar conexões de entrada negativas (wj < 0) ou positivas (wj > 0). Um valor de peso igual a zero equivale à ausência da conexão associada.

Introdução

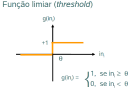
**Elementos Básicos de um Neurônio Artificial**

**Função de Ativação:**

◦ Ou função restritiva já que restringe (limita) o intervalo permissível de amplitude do sinal de saída a um valor finito.

◦ Tipicamente, o intervalo normalizado da amplitude da saída de um neurônio é escrito como o intervalo unitário fechado [0,1] ou alternativamente [-1.1].

◦ Tipos básicos dessas funções: **linear, limiar, sigmoidal, tangente hiperbólica, gaussiana e linear retificada (ReLU).**

****

Introdução

**Elementos Básicos de um Neurônio Artificial**

**Função de Ativação:**

◦ O uso da **função linear** identidade implica retornar como saída o valor de u.

◦ Na **função limiar**, empregada no modelo de neurônio artificial de McCulloch e Pitts (1943), o valor do limiar define quando o resultado da função limiar será igual a 1 ou 0 (alternativamente, pode-se empregar o valor –1). Quando a soma das entradas recebidas ultrapassa o limiar estabelecido, o neurônio torna-se ativo (saída +1). Quanto maior o valor do limiar, maior tem que ser o valor da entrada total para que o valor de saída do neurônio seja igual a 1.

◦ A **função sigmoidal** representa uma aproximação contínua e diferenciável da função limiar. A sua saída é um valor no intervalo aberto (0, 1), podendo apresentar diferentes inclinações.



Introdução

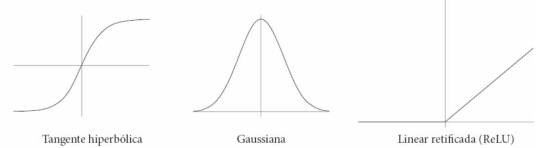
**Elementos Básicos de um Neurônio Artificial**

**Função de Ativação:**

◦ A função **tangente hiperbólica** é uma variação da função sigmoidal que utiliza o intervalo aberto (–1, +1) para o valor de saída.

◦ Outra função utilizada com frequência, também contínua e diferenciável, é a **função gaussiana**.

◦ Mais recentemente, com a popularização das redes profundas, passou a ser cada vez mais utilizada a função linear retificada, também conhecida como **ReLU** (do inglês Rectified Linear Unit). Essa função retorna 0 se recebe um valor negativo ou o próprio valor, no caso contrário. Junto com suas variações, ela tem apresentado bons resultados em várias aplicações.



Introdução

APLICAÇÕES

Quase todas as tarefas de aprendizagem: classificação, previsão numérica e mesmo reconhecimento não supervisionado de padrões

Programas de reconhecimento de voz e escrita

Automação de dispositivos inteligentes

Modelos sofisticados de padrões climáticos

Análise de desempenho em esportes

Detecção de Spams em e-mails

Aplicações na Bolsa de Valores e Finanças

Introdução APLICAÇÕES

Google

◦ Alpha Go

◦ Photos

Netflix

Spotify

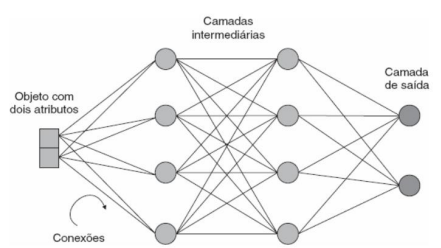
Robótica

Chatbots

►Exemplo

Introdução

**Arquitetura**

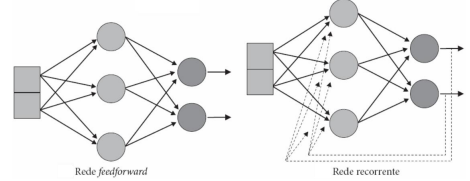
◦ Em uma RNA, os neurônios podem estar dispostos em uma ou mais camadas.

Introdução

**Arquitetura**

◦ As RNAs podem apresentar ou não conexões de retroalimentação, ou feedback.

◦ O número de camadas, o número de neurônios em cada camada e a presença ou não de conexões de retropropagação definem a topologia de uma RNA



Processos de Aprendizagem

◦ A propriedade primordial para uma RNA é a sua habilidade de aprender a partir de seu ambiente e de melhorar o seu desempenho através da aprendizagem.

◦ Definição de aprendizagem no contexto de RNA, segundo Haykin:

Aprendizagem é um processo pela qual os parâmetros livres

de uma rede neural são adaptados através de um processo

de estimulação pelo ambiente no qual a rede está inserida. O

tipo de aprendizagem é determinado pela maneira pela qual

a modificação de parâmetros ocorre.

Processos de Aprendizagem

◦ A definição do processo de aprendizagem implica a seguinte sequência de eventos: ◦ A rede neural é estimulada por um ambiente

◦ A rede neural sofre modifições nos seus parâmetros livres como resultado desta simulação ◦ A rede neural responde de uma maneira nova ao ambiente, devido às modificações ocorridas na estrutura interna

◦ Um conjunto de regras bem definidas para a solução de um problema de aprendizado é chamado de algoritmo de aprendizado.

◦ Há vários algoritmos de aprendizado. Cada um dos algoritmos oferece vantagens próprias. ◦ Basicamente, algoritmos de aprendizado diferem na forma como os pesos sinápticos são ajustados. ◦ Um outro fator é a maneira como a RNA se relaciona com seu ambiente.

Processos de Aprendizagem

Processos de Aprendizagem:

◦ Aprendizagem por Correção de Erro

◦ Aprendizagem Hebbiana

◦ Aprendizagem Competitiva

◦ Aprendizagem de Boltzmann

Pesquisa!!!!

Tipos de Aprendizagem

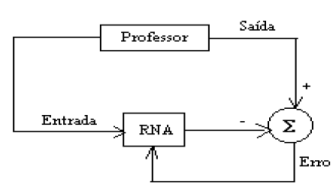
A propriedade mais importante das redes neurais é a habilidade de aprender de seu ambiente e com isso melhorar seu desempenho. Isso é feito através de um processo iterativo de ajustes aplicado a seus pesos denominado treinamento.

Denomina-se algoritmo de aprendizado a um conjunto de regras bem definidas para a solução de um problema de aprendizado

Formas de Aprendizado

Aprendizagem Supervisionada

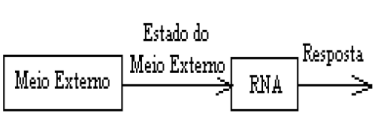
◦ a saída desejada e conhecida e informada para que a rede compare com a saída processada ◦ se houver erro, a rede tenta corrigir este erro ate que a mesma forneça uma saída igual a saída desejada



Formas de Aprendizado

Aprendizagem Não Supervisionada

◦ a saída desejada e obtida através de entradas repetitivas até a rede reter o conhecimento ◦ não existe saída informada para comparação

◦ deve existir redundâncias nos dados para a rede encontrar padrões ou características dos dados

Vantagens e Limitações

Vantagens:

• solução naturalmente paralela

• robusta, tolerante a falhas

• permite a integração de informações oriundas de fontes ou tipos diferentes • sistema adaptativo, capaz de aprender

• mostra certo grau de autonomia no aprendizado

• performance muito rápida no reconhecimento.

Limitações:

• ainda muito difícil explicar seu comportamento, por causa da falta de transparência • soluções não escalam bem... computacionalmente cara para problemas de maior porte • ainda muito distante da realidade biológica

Modelos

Formas mais básicas de **aprendizado** em Redes Neurais:

• **Perceptron**: Algoritmo para aprendizagem de redes neurais simples (uma camada) desenvolvido nos anos 50.

• **Backpropagation**: Algoritmo mais complexo para aprendizagem de redes neurais de múltiplas camadas desenvolvido nos anos 80.



Neurônio de McCulloch-Pitts(1943)

Aprendizagem de Perceptron

Usa-se um conjunto de **exemplos de treinamento** que dão a saída desejada para uma unidade, dado um conjunto de entradas.

O objetivo é **aprender pesos** sinápticos de tal forma que a unidade de saída produza a saída correta pra cada exemplo.

O algoritmo faz atualizações iterativamente até chegar aos **pesos corretos**.

Aprendizagem de Perceptron

**Unidade de Limite (*Threshold Linear*)**

A rede Perceptron é treinada por um algoritmo supervisionado de correção de erro e usa a função de ativação do tipo limiar. Durante o seu treinamento, para um objeto xρ, os pesos são ajustados de acordo

com a Equação: onde,



é o peso da j-ésima conexão de entrada no instante de tempo t,

η é uma taxa de aprendizado,

é o valor do j-ésimo atributo do vetor de entrada xi, 

é a saída produzida pela rede no instante de tempo t e yi é a saída desejada para a rede (o rótulo de xi).

Perceptron

**Unidade de Limite (*Threshold Linear*)**

◦ Se esses sinais forem superiores a aproximadamente 50mV (limiar do disparo), seguem pelo axônio. Caso contrário, são bloqueados e não prosseguem (são considerados irrelevantes). ◦ Se o sinal for superior a certo **limite** (*threshold*), vai em frente; caso contrário é bloqueado e não segue.

◦ Cada condutor, está associado um **peso** pelo qual o sinal é multiplicado.

◦ A memória são os pesos.

Perceptron



Rede de Perceptrons Valores de Saída

Sinais de entrada

Camada de Saída

Pesos Ajustaveis

Camada de Entrada

Treinando um Neurônio Operador AndThreshold = 0.2

| A | B | Saída |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

A

Learning Rate = 0.1

0.3

T=0.2

B-0.1

| A | B | Somatório | Saída | Erro |
| --- | --- | --- | --- | --- |
| 0 | 0 | (0\*0.3)+(0\*-0.1) = 0 | 0 | 0 |
| 0 | 1 | (0\*0.3)+(1\*-0.1) = -0.1 | 0 | 0 |
| 1 | 0 | (1\*0.3)+(0\*-0.1) = 0.3 | 1 | -1 |
| 1 | 1 | (1\*0.3)+(1\*-0.1) = 0.2 | 1 | 0 |

Treinando um Neurônio Operador AndThreshold = 0.2

| A | B | Saída |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

A

Learning Rate = 0.1

0.2

T=0.2

B0.1

| A | B | Somatório | Saída | Erro |
| --- | --- | --- | --- | --- |
| 0 | 0 | (0\*0.2)+(0\*-0.1) = 0 | 0 | 0 |
| 0 | 1 | (0\*0.2)+(1\*-0.1) = -0.1 | 0 | 0 |
| 1 | 0 | (1\*0.2)+(0\*-0.1) = 0.2 | 1 | -1 |
| 1 | 1 | (1\*0.2)+(1\*-0.1) = 0.1 | 0 | -1 |

Treinando um Neurônio Operador AndThreshold = 0.2

| A | B | Saída |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

A

Learning Rate = 0.1

0.1

T=0.2

B0.1

| A | B | Somatório | Saída | Erro |
| --- | --- | --- | --- | --- |
| 0 | 0 | (0\*0.1)+(0\*0.1) = 0 | 0 | 0 |
| 0 | 1 | (0\*0.1)+(1\*0.1) = 0.1 | 0 | 0 |
| 1 | 0 | (1\*0.1)+(0\*0.1) = 0.1 | 0 | 0 |
| 1 | 1 | (1\*0.1)+(1\*0.1) = 0.2 | 1 | 0 |

Limitações

Um único Perceptron consegue resolver somente funções linearmente separáveis. 

Em funções não linearmente separáveis o perceptron não consegue gerar um hiperplano para

separar os dados.



Redes Multicamadas

Perceptrons expressam somente superfícies de decisão linear.

Entretanto, é possível combinar vários perceptrons lineares para gerar superfícies de decisão mais complexas.

Dessa forma podemos, por exemplo, gerar uma superfícies de classificação para o operador XOR.

Para resolver problemas não linearmente separáveis utilizando RNAs, a alternativa mais emprega é adicionar uma ou mais camadas intermediárias.

As redes do tipo Perceptron multicamadas (MLP, do inglês *multilayer Perceptron*) apresentam uma ou mais camadas intermediárias de neurônios e uma camada de saída.

Operador XOR

Operador XOR

| A | B | Saída |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

X1

-0.5

1h11

1

OR

1

o

X2

1

-1.5

h2

-1-0.5

AND

Redes Multicamadas

Camadas Ocultas

Camada de Entrada

Camada de Saída

Redes Multicamadas

**Unidades lineares** são capazes gerar **funções lineares**, dessa forma função de uma rede multicamada também será linear.

Entretanto, existem muitas funções que **não podem ser modeladas por funções lineares**.

Por esse motivo é necessário utilizar uma **outra função de ativação**.

Redes Multicamadas

Funções de ativação mais comuns: ◦ Sigmoidal:

⎛

*n*

⎞

1

= = ⋅ +∑ ⋅ *y f h w w x p*

⎜

0

1 ;

⎟ =

*i i* − *h*

⎝

= 1

+

*e*

⎠ 

*p*

*i*

1

◦ Radial (Gausiana):

2

*h n*

1

⎛

⎞

−

= = ∑ ⋅ =

2

σ

*y f h x w w e*

⎜

⎝

*i*

2

=

1

( ) ; *i i*

0

⎟ =

⎠

σ

πσ

2

2 

Backpropagation

Um obstáculo que havia para utilizar redes multicamadas era a ausência de um algoritmo para o treinamento dessas redes, o que foi transposto com a proposta de um algoritmo baseado em **gradiente descendente** denominado ***back-propagation*** (Rumelhart et al., 1986).

Para que esse algoritmo seja utilizado, a função de ativação precisa ser contínua, diferenciável e, de preferência, não decrescente. A função de ativação do tipo **sigmoidal** obedece a esses requisitos.

Descida do Gradiente

• A **descida do gradiente** busca 

determinar um vetor de pesos que

minimiza o erro.

• Começando com um vetor inicial de

**pesos arbitrário** e modificando-o

repetidamente em pequenos passos.

A cada passo, o vetor de pesos é alterado na direção que produz a maior queda ao longo da superfície de erro.

Backpropagation

O ajuste dos pesos de uma rede MLP pelo algoritmo backpropagation:



onde,

representa o peso entre um neurônio l e o j-ésimo atributo de entrada ou a saída do j-ésimo neurônio da camada anterior, 

δlindica o erro associado ao l-ésimo neurônio e xjindica a entrada recebida por esse neurônio (o j-ésimo atributo de entrada ou a saída do j-ésimo neurônio da camada anterior).

Backpropagation



Backpropagation

O Backpropagation **não é um algoritmo ótimo** e não garante sempre a melhor resposta.

O algoritmo de descida do gradiente pode ficar preso em um erro **mínimo local**.

É possível refazer o treinamento variando os valores iniciais dos pesos.

Backpropagation é o algoritmo de aprendizagem mais comum, porém existem muitos outros.

Referencia

◦ CARVALHO, A. C. P. de L. F. et al. Inteligência Artificial - Uma

Abordagem de Aprendizado de Máquina, Grupo Gen, 2021.

◦ Haykin, S. Redes Neurais, Bookman 2008.

