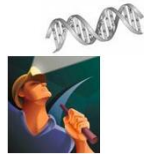


Mineração de Dados em Biologia Molecular

Redes Neurais Artificiais

Docente: André C. P. L. F. de Carvalho
PAE: Victor Hugo Barella



Principais tópicos

- Técnicas geométricas
- Análise discriminante
- Redes neurais
- Arquitetura e aprendizado de redes neurais
- Rede perceptron
- Rede MLP

0 0 0 3 3 1 1 1 2
0 2 2 2 2 2 3 3 3
2 4 4 4 4 5 5 5
6 6 7 7 7 8 8 8
9 9 9 9 9 9 9

© André de Carvalho - ICMC/USP

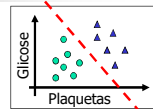
2

Discriminante linear

- Busca modelo que melhor se ajuste aos dados

■ Representação matemática

- Dois atributos preditivos
 - Fronteira de decisão = reta (hiperplano)
- Classificação ou regressão
- Função de hipótese
 - Combinação linear dos atributos preditivos
 - Soma ponderada
 - Como definir valores dos pesos?



$$y = ax + b$$

$$g = ap + b$$

$$g = -2p + 15$$

Função de classificação:

$$classe(x) = \begin{cases} +1 & \text{se } g + 2p - 15 \geq 0 \\ -1 & \text{se } g + 2p - 15 < 0 \end{cases}$$

$$f(x) = w_0 + w_1x_1$$

$$\hat{f}(x) = w_0 + w_1x_1 + w_2x_2 + \dots$$

© André de Carvalho - ICMC/USP

3

Redes Neurais

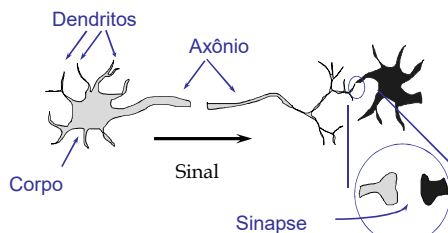
- Sistemas distribuídos inspirados no cérebro humano
 - Compostas por várias unidades de processamento ("neurônios")
 - Interligadas por um grande número de conexões ("sinapses")
- Eficientes em várias aplicações

© André de Carvalho - ICMC/USP

4

Neurônio natural

- Um neurônio simplificado:

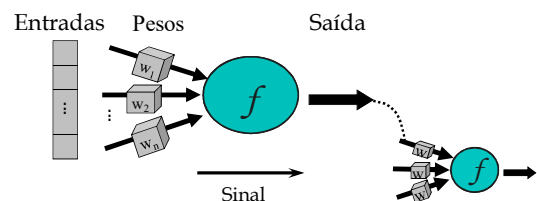


© André de Carvalho - ICMC/USP

5

Neurônio artificial

- Modelo de um neurônio abstrato



© André de Carvalho - ICMC/USP

6

Conceitos básicos

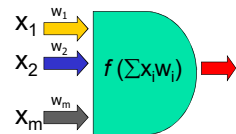
- Principais aspectos das RNA
 - Arquitetura
 - Unidades de processamento (neurônios)
 - Conexões
 - Topologia
 - Aprendizado
 - Algoritmos
 - Paradigmas

© André de Carvalho - ICMC/USP

7

Unidades de processamento

- Funcionamento
 - Recebe entradas de conjunto de unidades A
 - Aplica função sobre entradas
 - Envia resultado para conjunto de unidades B
- Entrada total

$$u = \sum_{i=1}^m x_i w_i$$


© André de Carvalho - ICMC/USP

8

Conexões

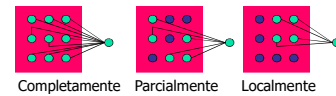
- Definem como neurônios estão interligados
- Codificam conhecimento da rede
- Tipos de conexões:
 - Excitatória: ($w_{ik}(t) > 0$)
 - Inibitória: ($w_{ik}(t) < 0$)

© André de Carvalho - ICMC/USP

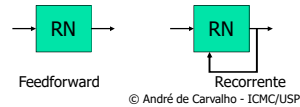
9

Topologia

- Número de camadas
- Cobertura das conexões



- Arranjo das conexões



© André de Carvalho - ICMC/USP

10

Algoritmo de aprendizado

- Conjunto de regras que define como ajustar os parâmetros da rede
- Principais formas de ajuste
 - Correção de erro
 - Hebbiano
 - Competitivo
 - Termodinâmico (Boltzmann)

© André de Carvalho - ICMC/USP

11

Paradigma de aprendizado

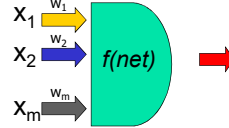
- Define informações externas que a rede recebe durante seu aprendizado
 - Principais abordagens
 - Supervisionado
 - Não supervisionado
 - Semi-supervisionado
 - Reforço
 - Híbrido

© André de Carvalho - ICMC/USP

12

Perceptron

- Primeira rede - Rosembat, 1958
 - Modelo de neurônio de McCulloch-Pitts
- Treinamento
 - Supervisionado
 - Correção de erro
 - $w_i(t) = w_i(t-1) + \Delta w_i$
 - $\Delta w_i = \eta x_i \delta$
 - $\Delta w_i = \eta x_i (\gamma - f(x))$
- Teorema de convergência



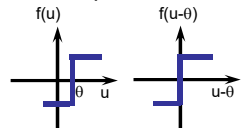
© André de Carvalho - ICMC/USP 13

Perceptron

- Resposta / saída da rede
 - Aplica função de ativação limiar sobre soma total de entrada recebida por um neurônio

$$u = \sum_{i=1}^m x_i w_i$$

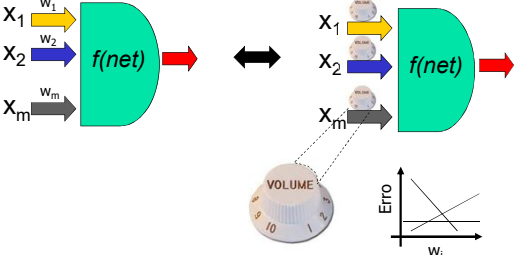
$$f(u) = \begin{cases} +1 & \text{if } u \geq \theta \\ -1 & \text{if } u < \theta \end{cases}$$

$$net = \sum_{i=0}^m x_i w_i$$


$f(u) = \text{sigal}(u - \theta)$
 $f(\text{net}) = f(u - \theta)$

© André de Carvalho - ICMC/USP 14

Treinamento



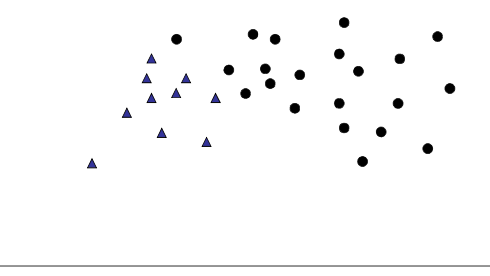
© André de Carvalho - ICMC/USP 15

Algoritmo de treinamento

- 1 Iniciar peso de cada conexão com o valor 0
- 2 Repita
 - Para cada par de treinamento (X, y)
 - Calcular a saída $f(X)$
 - Se $(y \neq f(X))$
 - Então
 - Atualizar pesos do neurônio
- Até condição de parada

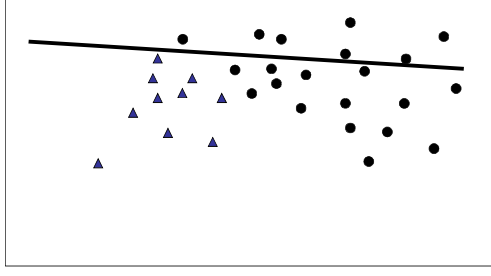
© André de Carvalho - ICMC/USP 16

Treinamento modificando fronteiras

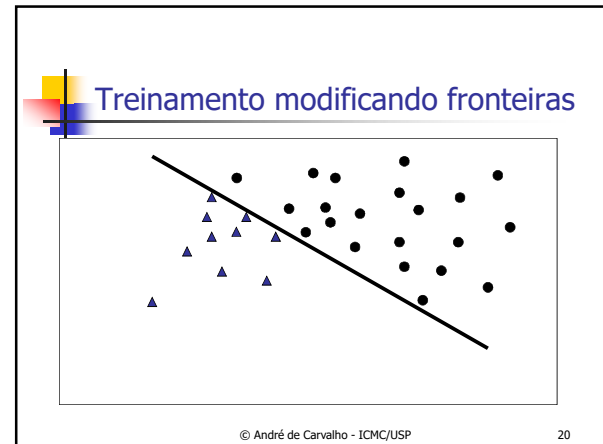
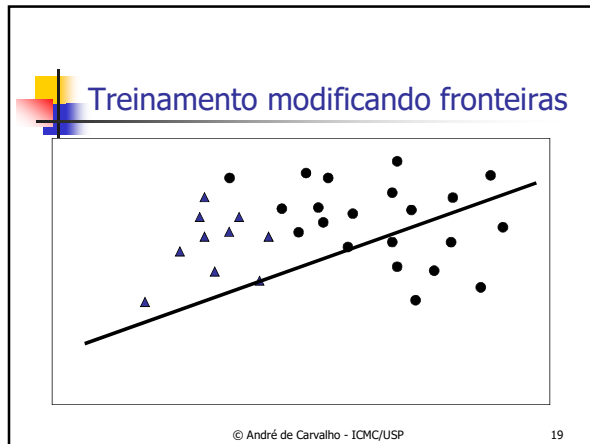


© André de Carvalho - ICMC/USP 17

Treinamento modificando fronteiras



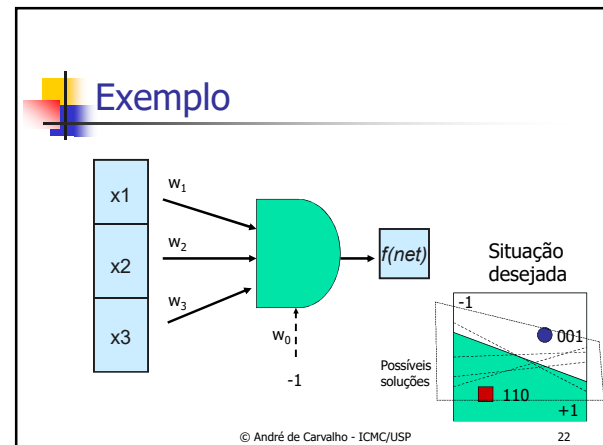
© André de Carvalho - ICMC/USP 18



Exemplo

- Dada uma rede Perceptron com:
 - Três entradas, pesos $w_1 = 0.4$, $w_2 = -0.6$ e $w_3 = 0.6$, e limiar (-viés) $\theta = 0.5$:
 - Ensinar a rede com os exemplos (001, -1) e (110, +1)
 - Utilizar taxa de aprendizado $\eta = 0.4$
 - Definir a classe dos exemplos: 111, 000, 100 e 011

© André de Carvalho - ICMC/USP 21



Exemplo (treinamento)

Treinar a rede

$x_0, w_0(\theta), w_1, w_2, w_3 = -1, 0.5, 0.4, -0.6, 0.6$

a.1) Para o exemplo 001 ($y = -1$)

Passo 1: definir a saída da rede ($\sum xw$)

$$u - \theta = -1(0.5) + 0(0.4) + 0(-0.6) + 1(0.6) = 0.1$$

$$f(\text{net}) = +1 \text{ (uma vez } 0.1 \geq 0)$$

Passo 2: atualizar pesos ($y \neq f(\text{net})$)

$$w_0 = 0.5 + 0.4(-1 - (+1)) = 1.3$$

$$w_1 = 0.4 + 0.4(0(-1 - (+1))) = 0.4$$

$$w_2 = -0.6 + 0.4(0(-1 - (+1))) = -0.6$$

$$w_3 = 0.6 + 0.4(1(-1 - (+1))) = -0.2$$

$$w_i(t) = w_i(t-1) + \eta x_i(y - f(x))$$

© André de Carvalho - ICMC/USP 23

Exemplo (teste)

- Utilizar a rede treinada para classificar os exemplos 111, 000, 100 e 011
- Pesos aprendidos: 0.4, 1.2, 0.2 e -0.2

b.1) Para o exemplo 111

$$u - \theta = -1(0.4) + 1(1.2) + 1(0.2) + 1(-0.2) = 0.6$$

$$f(\text{net}) = +1 \text{ (porque } 0.6 \geq 0) \Rightarrow \text{classe } +1$$

b.2) Para o exemplo 000

$$u - \theta = -1(0.4) + 0(1.2) + 0(0.2) + 0(-0.2) = -0.4$$

$$f(\text{net}) = -1 \text{ (porque } -0.4 < 0) \Rightarrow \text{classe } -1$$

© André de Carvalho - ICMC/USP 24

Exercício

- Seja o seguinte cadastro de pacientes:

Nome	Febre	Enjôo	Manchas	Dores	Diagnóstico
João	sim	sim	pequenas	sim	doente
Pedro	não	não	grandes	não	saudável
Maria	não	sim	pequenas	não	saudável
José	sim	sim	grandes	sim	doente
Ana	sim	não	pequenas	sim	saudável
Leila	não	não	grandes	sim	doente

© André de Carvalho - ICMC/USP

25

Exercício

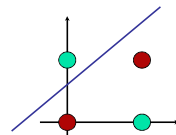
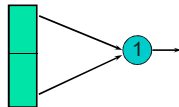
- Ensinar uma rede do tipo Perceptron a distinguir:
 - Pacientes potencialmente saudáveis
 - Pacientes potencialmente doentes
- Testar a rede para novos casos
 - (Luis, não, não, pequenas, sim)
 - (Laura, sim, sim, grandes, sim)

© André de Carvalho - ICMC/USP

26

Problemas com Perceptron

0, 0 → 0
 0, 1 → 1
 1, 0 → 1
 1, 1 → 0



© André de Carvalho - ICMC/USP

27

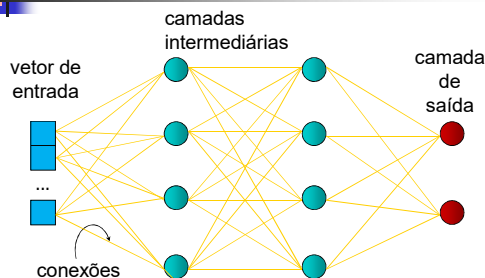
Rede Multi-Layer Perceptron

- Arquitetura de RNA mais utilizada
 - Uma ou mais camadas intermediárias de neurônios
- Funcionalidade (teórica)
 - Uma camada intermediária: qualquer função contínua ou Booleana
 - Duas camadas intermediárias: qualquer função
- Originalmente treinada com o algoritmo *backpropagation*

© André de Carvalho - ICMC/USP

28

MLP e *backpropagation*

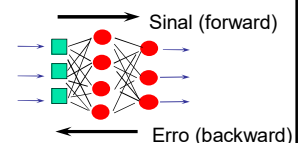


© André de Carvalho - ICMC/USP

29

Backpropagation

- Treina a rede com pares entrada-saída
 - Cada vetor de entrada é associado a uma saída desejada
- Treinamento em duas fases, cada uma percorrendo a rede em um sentido
 - Fase forward
 - Fase backward



© André de Carvalho - ICMC/USP

30

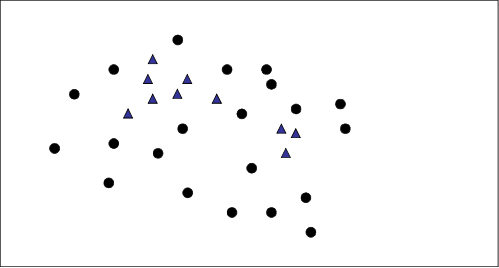
Treinamento

```

Iniciar todas as conexões com valores aleatórios  $\in [a,b]$ 
Repetir
  erro = 0;
  Para cada par de treinamento  $(X, y)$ 
    Para cada camada  $k := 1$  a  $N$ 
      Para cada neurônio  $j := 1$  a  $M_k$ 
        Calcular a saída  $f_j(\text{net})$ 
      Se  $k = N$ 
        Calcular soma dos erros de seus neurônios;
  Se erro  $> \epsilon$ 
    Para cada camada  $k := N$  a  $1$ 
      Para cada neurônio  $j := 1$  a  $M_k$ 
        Atualizar pesos;
  Até erro  $< \epsilon$  (ou número máximo de ciclos)
  
```

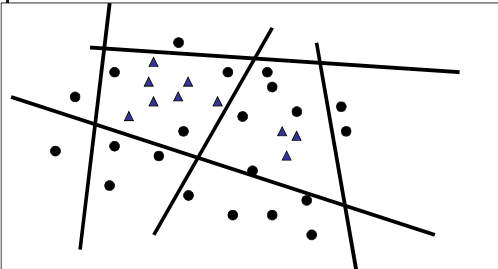
© André de Carvalho - ICMC/USP 31

Treinamento modificando fronteiras



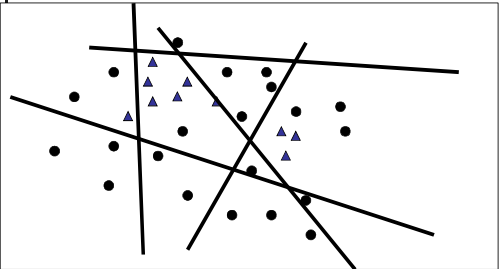
© André de Carvalho - ICMC/USP 32

Treinamento modificando fronteiras



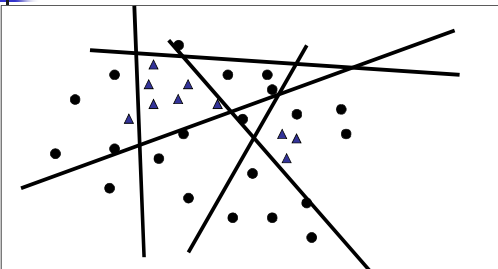
© André de Carvalho - ICMC/USP 33

Treinamento modificando fronteiras



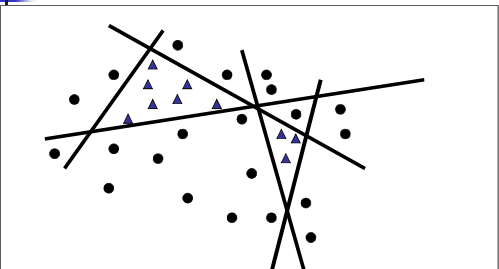
© André de Carvalho - ICMC/USP 34

Treinamento modificando fronteiras



© André de Carvalho - ICMC/USP 35

Treinamento modificando fronteiras



© André de Carvalho - ICMC/USP 36

MLPs como classificadores

© André de Carvalho - ICMC/USP

37

Regiões convexas

© André de Carvalho - ICMC/USP

38

Combinações de regiões convexas

© André de Carvalho - ICMC/USP

39

Combinações de regiões convexas

- Encontrar fronteiras de decisão que separem os dados abaixo:

© André de Carvalho - ICMC/USP

40

Combinações de regiões convexas

- Encontrar fronteiras de decisão que separem os dados abaixo:

© André de Carvalho - ICMC/USP

41

Exercício

- Quantas camadas e pelo menos quantos neurônios em cada camada tem a rede que divide o espaço de entradas das formas abaixo:

■ classe 1
 □ classe 2

□ classe 1
 ■ classe 2
 ■ classe 3

© André de Carvalho - ICMC/USP

42

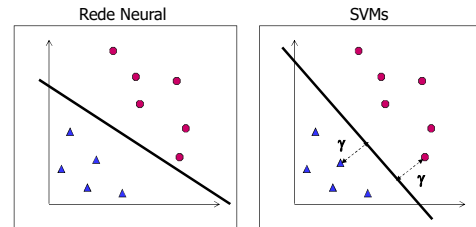
Teoria de Aprendizado Estatístico

- Difícil garantir que função induzida representa a função verdadeira
 - Modelo apresenta boa generalização
- TAE estabelece princípios para obter modelo com boa generalização
 - Vapnik e Chervonenkis em 1968
 - Busca função com menor erro e complexidade
 - Máquinas de Vetores de Suporte (SVMs)

© André de Carvalho - ICMC/USP

43

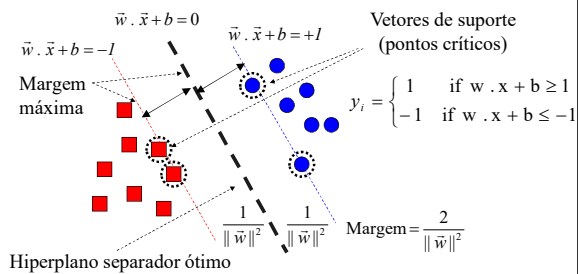
Máquinas de Vetores de Suporte (SVMs)



© André de Carvalho - ICMC/USP

44

Maximização das margens

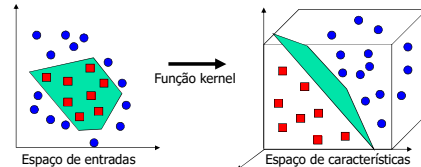


© André de Carvalho - ICMC/USP

45

Problemas não lineares

- Generalização de SVMs para problemas não lineares
 - Mapeamento de dados de entrada para um espaço de maior dimensão utilizando funções *kernel*



© André de Carvalho - ICMC/USP

46

Redes neurais profundas (RNP)

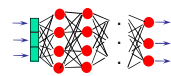
- Redes neurais MLP em geral têm 1 ou 2 camadas intermediárias
 - Redes neurais rasas (*shallow*)
- Poucas camadas:
 - Difícil extrair função que represente os dados
- Muitas camadas:
 - Uso de *backpropagation* leva a soluções ruins
 - Problema de atribuição de erro

© André de Carvalho - ICMC/USP

47

RNs profundas

- RNs rasas
 - Características extraídas manualmente (por especialistas) ou por técnicas de extração
- RN profundas
 - Características extraídas hierarquicamente por algoritmos de aprendizado
 - Não supervisionado
 - Pode usar dados não rotulados
 - Semi-supervisionado

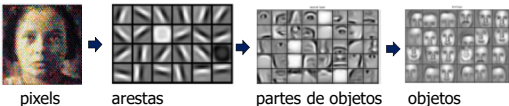


© André de Carvalho - ICMC/USP

48

RNs profundas

- Extração de características
 - Inicialmente características simples
 - Nível crescente de abstração
 - Cada camada aplica transformação não linear às características recebidas da camada anterior



pixels arestas partes de objetos objetos

© André de Carvalho - ICMC/USP 49

Principais RNs profundas

- Redes neurais profundas (RNP)
- Redes credais profundas (RCP)
- Redes autocodificadoras profundas (RAP)
- Redes neurais convolucionais profundas (RNCP)


© André de Carvalho - ICMC/USP 50

Conclusão

- Redes Neurais
 - Sistema nervoso
 - Muito utilizadas em problemas reais
 - Várias arquiteturas e algoritmos
 - Magia negra
 - Caixa preta

© André de Carvalho - ICMC/USP 51

Perguntas?



© André de Carvalho - ICMC/USP 52