

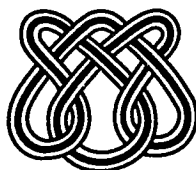
UNIVERSIDADE DE SÃO PAULO

NOTAS DE AULA DE REDES NEURAIS

ANDRÉ C.P.L.F. DE CARVALHO

Nº 32

NOTAS DIDÁTICAS



Instituto de Ciências Matemáticas de São Carlos

Instituto de Ciências Matemáticas de São Carlos

ISSN - 0103-2577

NOTAS DE AULA DE REDES NEURAIIS

ANDRÉ C.P.L.F. DE CARVALHO

Nº 32

NOTAS DIDÁTICAS DO ICMSC

**São Carlos
Ago./1998**

Apresentação

Estas notas apresentam os principais tópicos cobertos pela disciplina de Redes Neurais, tanto na graduação como na pós-graduação. Este material foi compilado a partir das notas de aula das disciplinas de Introdução a Redes Neurais (SCE-211), para alunos de graduação, e Redes Neurais (SCE-5809), para alunos de pós-graduação. Estas notas devem ser utilizadas como material auxiliar pelos alunos durante seus estudos. Elas devem ser utilizadas conjuntamente com os livros indicados, ou seja, elas não substituem os livros recomendados.

Gostaria de agradecer aos alunos do programa PAE Elson Félix de Mendes Filho, Paulo Sérgio Prampeiro e Ernesto Cuadro Vargas pelo auxílio na confecção destas notas. Gostaria de agradecer também aos alunos que cursaram as disciplinas Introdução a Redes Neurais e Redes Neurais que, com suas críticas e sugestões, contribuíram para o aprimoramento deste material.

Principais Tópicos

1 Fundamentos

- 1.1 Introdução
- 1.2 O que são redes neurais
- 1.3 Redes neurais *versus* computadores convencionais
- 1.4 Aplicações
- 1.5 Histórico
- 1.6 Conclusão

2 Sistema nervoso

- 2.1 Introdução
- 2.2 Neurônios
- 2.3 Sinapses
- 2.4 Cérebro
- 2.5 Sistema visual

3 Reconhecimento de padrões

- 3.1 Introdução
- 3.2 Abordagens
- 3.3 Extração de características
- 3.4 Classificação estatística

4 Conceitos básicos

- 4.1 Introdução
- 4.2 Estrutura geral das RNA
- 4.3 Conexões
- 4.4 Topologias

5 Aprendizado

- 5.1 Introdução
- 5.2 Algoritmos de aprendizado
- 5.3 Paradigmas de aprendizado

6 Perceptrons

- 6.1 Introdução
- 6.2 Modelos perceptron
- 6.3 Modelos adaline
- 6.4 Limitações

7 Redes MLP

- 7.1 Introdução
- 7.2 Treinamento
- 7.3 Generalização
- 7.4 Variações

8 Redes RBF

8.1 Introdução

8.2 Arquitetura

8.3 Treinamento

8.4 RBF *versus* MLP

9 Modelo de Hopfield

9.1 Introdução

9.2 Armazenamento

9.3 Recuperação

9.4 Variações

10 Redes Self-organising

10.1 Introdução

10.2 Princípios

10.3 Redes de Kohonen

10.4 Arquitetura

10.5 Aprendizado

10.6 Teste

11 Redes ART

11.1 Introdução

11.2 Família ART

11.3 Arquitetura

11.4 Treinamento

11.5 Teste

12 Algoritmos Genéticos

12.1 Introdução

12.2 Conceitos básicos

12.3 Codificação

12.4 Função de aptidão

12.5 Operadores Genéticos

12.6 Convergência




Redes Neurais

Prof. Dr. André C. P. L. F de Carvalho
LABIC - ICMSC - USP

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 1



Tópicos do curso

- Introdução
- Sistema nervoso
- Aprendizado
- Modelos de Redes Neurais Artificiais
- Algoritmos Genéticos
- Sistemas Híbridos Inteligentes

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 2



Introdução

- Por que Redes Neurais?
- O que são Redes Neurais
- Redes Neurais X Computadores Convencionais
- Aplicações
- Histórico
- Conclusão

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 3



Por que Redes Neurais?

- Utilizar máquinas efetivamente para resolver problemas simples (humanos)
- Exemplo1: distinguir padrões visuais
 - utilizando exemplos e feedback
 - Maior experiência permite melhorar a performance



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 4



Por que Redes Neurais?

- Exemplo 2: otimizar tempo de trabalho (maximizar recurso)

Pode-se derivar regras, mas elas não refletem o processo de pensamento utilizado



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 5



Por que Redes Neurais?

- Computadores convencionais são eficientes em várias áreas, mas têm desapontado em outras
 - Sistema visual humano
 - Reconhece rosto familiar em ambiente estranho (100-200 ms)
 - Sonar de morcegos
 - Reconhece alvos (distância e velocidade)
 - Cérebro do tamanho de uma ameixa

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 6



Por que Redes Neurais?

- Desenvolvimento de RNA começou com desejo de entender o cérebro e copiar seu poder
- Sucessos iniciais de RNA, foram encobertos
 - Progresso em computação digital
 - Promessas exageradas sobre capacidade dos primeiros modelos

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

7



O que são Redes Neurais

- Redes Neurais Artificiais (RNA) são novos modelos de computação com propriedades particulares
 - Capacidade de se adaptar ou aprender
 - Generalizar
 - Agrupar ou organizar dados

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

8



O que são Redes Neurais

- RNA: estruturas distribuídas formadas por grande número de unidades de processamento conectadas entre si
- RNA são pesquisadas em várias disciplinas: Ciência da Computação, Matemática, Física, Engenharias, Psicologia, Biologia, Linguística e Filosofia

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

9



O que são Redes Neurais

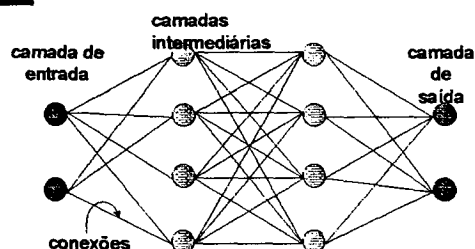
- Modelos inspirados no cérebro humano
 - Compostas por várias unidades de processamento ("neurônios")
 - Interligadas por um grande número de conexões ("sinapses")
- Eficientes onde métodos tradicionais têm se mostrado inadequados

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

10



Redes Neurais Artificiais



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

11



Características das RNA

- Aprendem através de exemplos
 - Inferência estatística não paramétrica
- Adaptabilidade
 - Dilema plasticidade X estabilidade
- Capacidade de generalização
- Tolerância a falhas
- Implantação rápida

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

12



Von Neumann X Sistema neural

- Computadores modernos superam seres humanos em diversos domínios
 - Computação numérica
 - Manipulação de símbolos (arquivos)
- Existem problemas complexos onde seres humanos são mais eficientes que o mais potente dos computadores
 - Reconhecimento de faces em uma multidão
 - Linguagem natural

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

13



Von Neumann X Sistema neural

- Qual a causa desta diferença de desempenho?
 - Arquitetura de um sistema neural biológico é completamente diferente da arquitetura Von Neumann
 - Diferença afeta o tipo de função que cada modelo computacional pode realizar melhor

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

14



Von Neumann X Sistema neural

	Computador Von Neumann	Sistema neural biológico
Processador	complexo alta velocidade um ou poucos	simples baixa velocidade grande número
Memória	Separada do processador localizada não-endereçável pelo conteúdo	integrada com processador distribuída endereçável pelo conteúdo
Computação	centralizada seqüencial programas armazenados	distribuída paralela aprendizado
Confiabilidade	muito vulnerável	robusto
Adequação	manipulações num. e simbólicas	Problemas de percepção
Ambiente operacional	bem definido muito restrito	pouco definido não restrito



Potenciais áreas de aplicação

- Classificação de padrões
- Clustering/categorização
- Aproximação de funções
- Previsão
- Otimização
- Memória endereçável pelo conteúdo
- Controle

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

16



Classificação de padrões

- Tarefa: atribuir um padrão de entrada a uma das várias classes pré-definidas
 - Entradas representadas por vetores de características
- Exemplos de aplicações
 - Reconhecimento de caracteres
 - Reconhecimento de voz
 - Análise de crédito

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

17



Clustering/categorização

- Tarefa: explorar semelhanças entre padrões e agrupar padrões parecidos
 - Também conhecido como aprendizado não supervisionado
 - Classes não são conhecidas
- Exemplos de aplicações
 - Garimpagem de dados
 - Compressão de dados

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

18



Aproximação de funções

- Tarefa: encontrar uma estimativa f' de uma função desconhecida f
 - Conhece conjunto de pares de entrada-saída $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$
- Exemplos
 - Problemas de modelagem científica e de engenharia

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

19



Previsão

- Tarefa: dado um conjunto de exemplos $\{(y(t_1), (y(t_2), \dots, (y(t_n)\},$ prever o exemplo $(y(t_{n+1}))$ no instante de tempo t_{n+1}
- Exemplos
 - Previsão do tempo
 - Previsão de falências
 - Previsão de ações na bolsa
 - Previsão de desgaste de peças

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

20



Otimização

- Tarefa: encontrar solução que satisfaça um conjunto de restrições tal que uma função objetivo seja maximizada ou minimizada
- Exemplos
 - Problema de corte
 - Problema do caixeiro viajante (NP completo)
 - Problema do container

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

21



Memória endereçável pelo conteúdo

- Tarefa: recuperar itens utilizando eles mesmos como endereços
 - Recupera item correto mesmo que a entrada seja parcial ou distorcida
- Exemplos
 - Bases de Dados
 - Sistemas Multimídia

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

22



Controle

- Tarefa: gerar entrada de controle para que sistema siga trajetória especificada por modelo de referência
 - Modelo definido por conjunto de tuplas $\{x(t), y(t)\}$
- Exemplo
 - Controle de processos químicos
 - Controle de robôs

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

23



RNA na indústria

- RNA são muito utilizadas em fábricas no exterior
 - Especialmente quando instalação rápida ou grande velocidade de operação é requerida
 - Ex. Birdseye fish fingers são inspecionados visualmente por uma RNA
 - Caixas imperfeitas são rejeitadas para exportação

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

24



RNA no comércio

- Tem havido um grande número de pesquisas sobre a utilização de RNA para previsão financeira
 - Tendências sugerem expansão destas aplicações
 - Ex. Cartão de crédito VISA utiliza RNA para liberação de cartões e detecção de fraudes

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

25



RNA na medicina

- RNA têm sido utilizadas para
 - Diagnóstico de doenças
 - Rede treinada para, dado um conjunto de sintomas, diagnosticar a doença e sugerir melhor tratamento
 - Reconhecimento de imagens médicas
 - Reconhecimento de fraturas
 - Reconhecimento de anomalias no coração

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

26



História das RNA

- Inter-relação entre
 - Investigação do comportamento e estrutura do sistema nervoso através de experimentação e modelagem biológica
 - Desenvolvimento de modelos matemáticos e suas aplicações para a solução de vários problemas práticos
- Simulação e implementação destes modelos

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

27



A origem

- (300 A. C.) Aristóteles escreveu: *de todos os animais, o homem, proporcionalmente, tem o maior cérebro*
- (1700) Descartes acreditava que mente e cérebro eram entidades separadas
- (1911) Ramon e Cajal introduzem a idéia de neurônios como estruturas básicas do cérebro

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

28



1940: O começo

- (1943) McCulloch & Pitts
 - Desenvolvem modelo matemático de RNA
 - Combinação de vários neurônios em sistemas neurais produz um elevado poder computacional
 - Neurônios executam funções lógicas simples
 - Cada neurônio pode executar uma função diferente
 - Qualquer saída que possa ser representada por uma combinação de funções lógicas pode ser gerada por uma rede de neurônios

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

29



1940: O começo

- (1943) McCulloch & Pitts (continuação)
 - Provam, teoricamente, que qualquer função matemática ou lógica pode ser implementada utilizando unidades soma de produtos
- (1949) Hebb desenvolve algoritmo para treinar RNA (aprendizado Hebbiano)
 - Se dois neurônios estão simultaneamente ativos, a conexão entre eles deve ser reforçada

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

30



1950-1960: Os anos dourados

- (1958) Von Neumann mostra interesse em modelagem do cérebro (RNA)
 - The Computer and the Brain, Yale University Press
- (1959) Roseblatt implementa primeira RNA, a rede Perceptron
 - Ajuste iterativo de pesos
 - Prova teorema da convergência

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

31



1950-1960: Os anos dourados

- (1969) Minsky & Papert analisam Perceptron e mostram suas limitações
 - Não poderiam aprender a resolver simples problemas lógicos (XOR)
 - Por muito tempo foi assumido que este trabalho desacreditava RNA, embora não fosse esta a intenção

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

32



1970: Os anos quietos

- (1971) Aleksander propõe redes Booleanas
- (1972) Kohonen trabalha com RNA associativas
- (1975) Grossberg desenvolve a Teoria da Ressonância Adaptiva (redes ART)
- (1977) Anderson desenvolve rede associativa Brain-State-in-a-Box

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

33



1980: A segunda onda

- (1982) Hopfield mostra que Redes Neurais podem ser tratadas como sistemas dinâmicos
- (1986) Hinton, Rumelhart e Williams, propõem algoritmo de aprendizagem para redes multi-camadas
 - *Parallel Distributed Processing*
 - Bryson e Ho (1959), Werbos (1974), Parker (1985) e Le Cun (1985)

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

34



Problemas das RNAs

- Definição dos parâmetros
 - Magia negra
- Extração de conhecimento
 - Caixa preta
- Existem vários tipos de RNA diferentes
 - Cada tipo tem propósitos diferentes
 - Alguns tipos são mais adequados para resolver classes particulares de problemas

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

35



Conclusão

- Introdução
- Comparação com computação tradicional
- Aplicações
- Histórico
- Problemas
- A seguir: sistema nervoso

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

36



Reconhecimento de Padrões



Prof. Dr. André C. P. L. F de Carvalho
LABIC - ICMSC - USP

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 1



Reconhecimento de padrões

- Introdução
- Abordagens
- Extração de características
- Classificação estatística
 - Não paramétrica
 - Paramétrica

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 2



Reconhecimento de Padrões (RP)

- Área de pesquisa estabelecida
 - Surgiu na Engenharia Elétrica
 - Tornou-se popular com o barateamento da captura automática de sinais e imagens
- RNA e RP
 - Base matemática semelhante
 - Atacam mesmos problemas
 - Várias aplicações de RNA são RP

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 3



O que é Rec. de Padrões?

- RP é uma técnica de tomada de decisões a partir de padrões complexos de informações
- Objetivo: lidar com tarefas realizadas por seres humanos
- Função: classificação de padrões de entrada em categorias ou classes

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 4



Exemplos de RP

- Leitura (impressos e manuscritos)
- Imagens
- Vozes
- Tendências da bolsa de valores
- Decisão do próximo movimento em um jogo de xadrez

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 5



Reconhecimento de Padrões

- Natureza dos dados define tipo de Reconhecimento de Padrões
 - Estruturais
 - Utiliza informação qualitativa sobre as classes
 - Prova de teoremas
 - Sintáticas
 - Informação sobre as classes é fornecida por uma gramática
 - Estatísticas (numéricas)

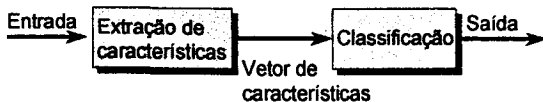
8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 6



Reconhecimento de Padrões

- Etapas de um sistema de RP:

- Extração de características
- Classificação



8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

7



Reconhecimento de Padrões

- Extração de características

- Medidas do padrão de entrada
 - Formam vetor de características
- Caracterizam tipo de entrada
 - Exemplo: padrões E e F (número de traços horizontais)
 - E: três traços horizontais
 - F: dois traços horizontais
- Raramente é trivial

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

8



Reconhecimento de Padrões

- Vetor de características

- Composto por várias medidas
- Dimensão do vetor = número de medidas extraídas
- Exemplo: classificar entre bailarinas e halterofilistas
 - Características: peso e altura
 - Esportista = ponto no espaço (p,a)

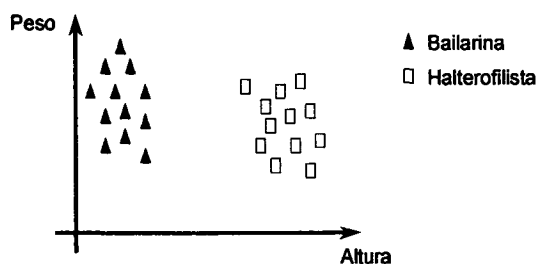
8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

9



Reconhecimento de Padrões



8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

10



Reconhecimento de Padrões

- Classificação

- Recebe vetores de características
- Define classe para cada entrada
 - Probabilidade, função discriminante e Medidas de distância
- Possíveis classificações
 - Classificação correta
 - Rejeição
 - Erro

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

11



Abordagem estatística

- Conhecimento das classes

- Supervisionada
- Não supervisionada
- Estimar função de densidade de probabilidade
 - Não paramétricos
 - Paramétricos
 - Semi-paramétricos
 - Redes Neurais

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

12



Abordagem estatística

● Entre principais técnicas de classificação de padrões

■ Classificação Bayesiana

- Técnica poderosa e amplamente utilizada
- Baseada em teoria de probabilidades
- Utiliza medidas dos padrões (vetor de características) para estimar probabilidade de um padrão pertencer a uma classe particular



Abordagem estatística

● Seja C_i , ($i = 1, 2, \dots, N$) a lista de possíveis classes

- A cada classe é associada a probabilidade de um padrão pertencer a ela : $P(C_i)$
 - $0 \leq P(C_i) \leq 1$

● Padrão X pertence a classe C_i se:

- $P(C_i/X) > P(C_k/X)$ para $k = 1, 2, \dots, n$ ($i \neq k$)
- Melhor estimativa que existe
- Difícil definir $P(C_i/X)$



Abordagem estatística

● Definição de $P(C_i/X)$

■ Regra de Bayes

$$P(C_i/X) = \frac{P(X/C_i) P(C_i)}{\sum P(X/C_k) P(C_k)}$$

■ Técnicas para definição de $P(X/C_i)$

- Paramétricas
- Não paramétricas
- Semi-paramétricas



Técnicas paramétricas

● Assumem um modelo de distribuição de probabilidade dos dados

- Contém um número de parâmetros que são otimizados para encaixar no modelo escolhido
- Distribuição escolhida pode não modelar a verdadeira

● Principais técnicas

- Inferência Bayesiana
- Máxima probabilidade



Técnicas paramétricas

● Inferência Bayesiana

- São utilizados modelos para $P(X/C_i)$ baseados em uma distribuição padrão
- Performance depende de quanto os dados reais se encaixam na distribuição escolhida



Técnicas não paramétricas

● Não assumem nenhum modelo de distribuição dos dados

● Forma da função densidade é definida pelos dados

- Número de parâmetros cresce com o tamanho do conjunto de dados



Técnicas não paramétricas

- Técnicas mais comuns
 - K-vizinhos mais próximos
 - Utiliza métodos de comparação
 - Medidas de distância
 - Classificação linear
 - Utiliza fronteiras de decisão
 - Histogramas

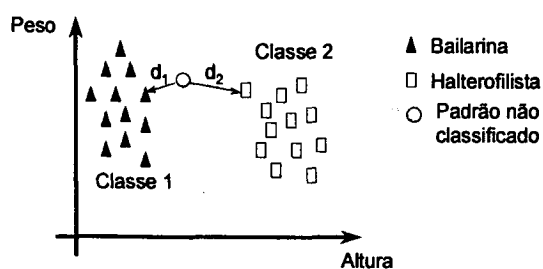


K-vizinhos mais próximos

- Utiliza métodos de comparação
 - Distância mais curta entre padrão desconhecido e os k-padrões mais próximos de cada classe
 - Atribui padrão desconhecido à classe que parece estar mais próxima
 - Função discriminante
 - $f(x) = \text{mais_próximo}_{\text{classe1}} - \text{mais_próximo}_{\text{classe2}}$



K-vizinhos mais próximos

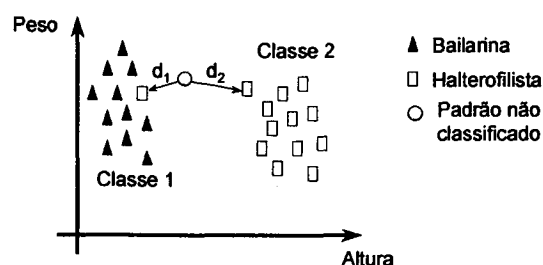


K-vizinhos mais próximos

- Aplicação restrita
 - Necessário encontrar uma medida de distância confiável
 - Padrão "fora de esquadro"
 - Utilizar várias medidas de distância com vários exemplos de cada classe
 - K-vizinhos mais próximos (k é o número de exemplos utilizados de cada classe)



K-vizinhos mais próximos



K-vizinhos mais próximos

- Medidas de distância
 - Calculam similaridade entre padrões no espaço geométrico
 - Medidas mais utilizadas:
 - Distância de Hamming
 - Distância Euclidiana
 - Distância bloco-cidade (Manhattan)
 - Distância quadrada




Sistema Nervoso

Prof. André C. P. L. F de Carvalho
LABIC - ICMSC - USP

24/04/98 Redes Neurais - André Ponco de Leon F. de Carvalho - LABIC/USP 1



Redes Neurais Naturais

- Sistema nervoso
- Neurônios
- Sinapses
- Cérebro
- Sistema visual

24/04/98 Redes Neurais - André Ponco de Leon F. de Carvalho - LABIC/USP 2



Introdução

- Sistema nervoso
 - Conjunto de células extremamente complexo que tem papel essencial na determinação do funcionamento e comportamento dos seres vivos
 - Unidade fundamental: célula nervosa (neurônio)
 - Distingue-se das outras células por apresentar excitabilidade
 - Possibilita transmissão de impulsos nervosos a outros neurônios e a células musculares e granulares

24/04/98 Redes Neurais - André Ponco de Leon F. de Carvalho - LABIC/USP 3



Sistema nervoso

- Sistema nervoso divide-se em
 - Sistema nervoso periférico (SNP)
 - Sistema nervoso central (SNC)
- Sistema nervoso periférico (SNP)
 - Nervos aferentes ou sensitivos transmitem ao SNC sinais vindos da periferia do corpo
 - Nervos eferentes ou motores transmitem aos órgãos ordens dadas pelo SNC

24/04/98 Redes Neurais - André Ponco de Leon F. de Carvalho - LABIC/USP 4



Sistema nervoso

- Sistema nervoso central (SNC)
 - Constituído por:
 - Medula, contida na coluna vertical
 - Encéfalo, massa de tecido nervoso contida na caixa craniana
 - Forma-se nas primeiras fases da vida embrionária de um tubo oco (tubo neural)
 - Durante desenvolvimento do feto, transforma-se em um conjunto de cinco vesículas

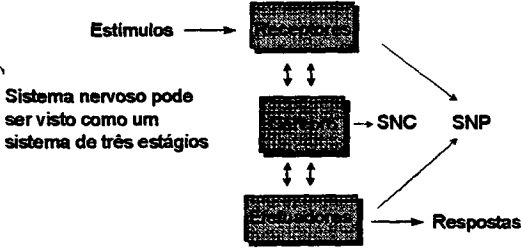
24/04/98 Redes Neurais - André Ponco de Leon F. de Carvalho - LABIC/USP 5



Sistema nervoso

Estímulos →

Sistema nervoso pode ser visto como um sistema de três estágios



Respostas

24/04/98 Redes Neurais - André Ponco de Leon F. de Carvalho - LABIC/USP 6



Cérebro

- Funciona de forma inteiramente diferente dos computadores convencionais
 - Neurônios são 100 mil a 1 milhão de vezes mais lentos que portas lógicas de silício
 - Lentidão compensada por grande número de neurônios massivamente conectados
 - Para certas operações, muito mais rápido que computadores convencionais
 - Visão, audição, controle e previsão

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

7



Cérebro

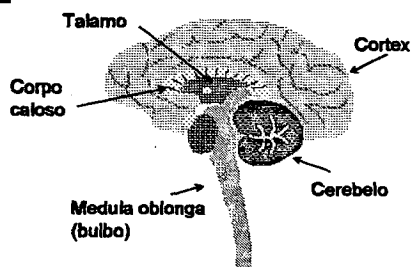
- Cérebro tem estrutura e capacidade de construir suas próprias regras através da experiência
 - Um milhão de sinapses por segundo são desenvolvidas nos dois primeiros anos
- Cérebro é composto por várias regiões especializadas

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

8



Cérebro



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

9



Cérebro

- Regiões do cérebro
 - Tálamo: o centro de transmissão de vias sensoriais mais importante
 - Medula oblonga (bulbo): onde maioria dos nervos cranianos entra e sai do encéfalo
 - Cortex: Principal área de processamento
 - Áreas motoras: produzem respostas motoras quando estimuladas eletricamente
 - Áreas sensitivas: Quando estimuladas, provocam efeitos sensoriais

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

10



Cérebro

- Regiões do cérebro (continuação)
 - Cerebelo: Coordenação de atividades motoras e manutenção da postura
 - Dados vêm de duas fontes principais
 - Área auditiva do bulbo
 - Sistema de receptores existentes nos músculos e tendões
 - Corpo caloso: permite a união dos recursos de memória e aprendizado dos dois hemisférios

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

11



Cérebro

- Hemisférios cerebrais
 - Simétricos
 - Maioria das funções do lado direito do corpo são controladas pelo lado esquerdo e vice-versa
 - Dividido em quatro lobos:
 - Frontal
 - Parietal
 - Occipital
 - Temporal

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

12



Medidas de distância

- Distância de Hamming
 - Medida mais básica
 - $D_H(X, Y) = \sum_{i=1}^N |x_i - y_i|$
 - Vetores binários
 - $|x_i - y_i| \Leftrightarrow x_i \text{ XOR } y_i$ (ou-exclusivo)

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

25



Medidas de distância

- Distância Euclidiana
 - Uma das medidas mais comuns
 - $D_E(X, Y) = \sqrt{\sum_{i=1}^N (x_i - y_i)^2}$
 - Exemplo bidimensional
 - Equivalente ao Teorema de Pitágoras
 - Vetores Binários: $D_E(X, Y) = \sqrt{D_H}$
 - Muito utilizada por ser simples de calcular

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

26



Medidas de distância

- Distância bloco-cidade (Manhattan)
 - Versão simplificada da distância Euclidiana
 - $D_{bc}(X, Y) = \sum_{i=1}^N |x_i - y_i|$
 - Cálculo mais rápido que distância Euclidiana
 - Equivale a D_H para vetores não binários
 - Menor tempo de cálculo, às custas de uma menor precisão

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

27



Medidas de distância

- Distância quadrática
 - $D_q(X, Y) = \text{MAX}(|x_i - y_i|)$
 - Simplifica ainda mais Distância Euclidiana
 - Maior simplificação e menor precisão

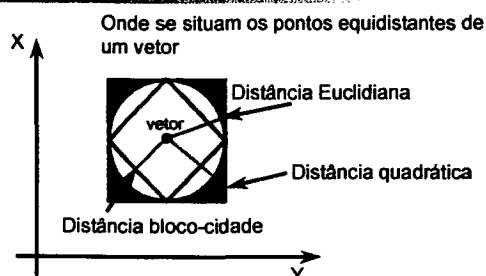
8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

28



Medidas de distância



8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

29



Classificadores lineares

- Técnica muito encontrada em RNA
- Função discriminante utiliza fronteiras de decisão (linha reta)
 - Base para maioria das técnicas de RP

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

30



Classificadores lineares

● Funções discriminantes

- Define fronteiras de decisão no espaço de padrões
 - Número de fronteiras possíveis pode ser infinito
- Quanto mais simples, melhor
- Problemas:
 - Como implementar fronteira de decisão
 - Como definir posição da fronteira de decisão

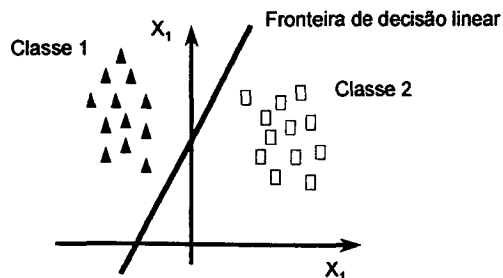
8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

31



Classificadores lineares



8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

32



Classificadores lineares

● Fronteira de decisão define função discriminante da forma:

$$f(x) = ax + b \text{ (função linear)}$$

- a : coeficiente angular
 - Fornece inclinação da reta
- b : coeficiente linear
 - Fornece o ponto onde a reta corta o eixo $f(x)$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

33



Classificadores lineares

● Fronteira de decisão define função discriminante da forma:

$$f(x) = \sum w_i x_i - \theta \text{ (função linear)}$$

- x_i : i -ésimo componente de um vetor de entrada
- w_i : i -ésimo componente de um vetor de pesos
- θ : threshold (ou limiar)
- Valor de $f(x)$ pode ser positivo ou negativo
 - se $f(x) > 0 \Rightarrow x \in \text{classe A}$
 - se $f(x) < 0 \Rightarrow x \in \text{classe B}$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

34



Classificadores lineares

● Supor problema com duas classes

- $f(x) = x_1 w_1 + x_2 w_2 - \theta = 0$
- $x_2 = -x_1 w_1 / w_2 - \theta / w_2$ ($y = ax + b$)
- Equação de reta
 - w_1 / w_2 : coeficiente angular
 - Fornece inclinação da reta
 - θ / w_2 : coeficiente linear
 - Fornece o ponto onde a reta corta o eixo $f(x)$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

35



Classificadores lineares

● Se forem conhecidos os valores corretos dos pesos (w_1 e w_2) e bias (θ)

- Processo de discriminação pode ser executado
 - Posição da fronteira de decisão pode ser definida
- Problema: Definir pesos (w_1 e w_2) e bias (θ)
 - Não é trivial
 - Método iterativo de tentativa e erro que modifica valores de acordo com uma função de erro $e = d - y$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

36



Classificadores lineares

- Definição dos valores não é fácil
 - Sejam as entradas vetores de n bits
 - 2^n possíveis vetores
 - 2^{2^n} mapeamentos diferentes
 - Pequena fração é linearmente separável
 - Classificador linear pode realizar apenas os linearmente separáveis
 - Problemas linearmente separáveis são aqueles que podem ser satisfeitos utilizando um hiperplano como superfície de decisão

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

37



Classificadores lineares

- Classificadores lineares podem separar mais que duas classes
 - Várias fronteiras de decisão
- Problemas não linearmente separáveis
 - Pré-processamento pode transformar dados em linearmente separáveis
 - Transformação não-linear dos dados originais
 - Pode ser um processo lento

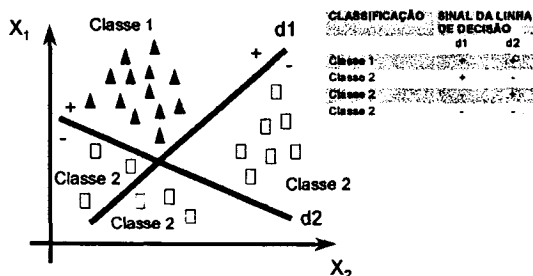
8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

38



Classificadores lineares



8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

39



Conclusão

- Reconhecimento de padrões
- Técnicas
 - Estruturais
 - Sintáticas
 - Estatísticas
 - Paramétricas
 - Não paramétricas

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

40




Sistema Nervoso

Prof. André C. P. L. F de Carvalho
LABIC - ICMSC - USP

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 1



Redes Neurais Naturais

- Sistema nervoso
- Neurônios
- Sinapses
- Cérebro
- Sistema visual

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 2



Introdução

- Sistema nervoso
 - Conjunto de células extremamente complexo que tem papel essencial na determinação do funcionamento e comportamento dos seres vivos
 - Unidade fundamental: célula nervosa (neurônio)
 - Distingue-se das outras células por apresentar excitabilidade
 - Possibilita transmissão de impulsos nervosos a outros neurônios e a células musculares e glandulares

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 3



Sistema nervoso

- Sistema nervoso divide-se em
 - Sistema nervoso periférico (SNP)
 - Sistema nervoso central (SNC)
- Sistema nervoso periférico (SNP)
 - Nervos aferentes ou sensitivos transmitem ao SNC sinais vindos da periferia do corpo
 - Nervos eferentes ou motores transmitem aos órgãos ordens dadas pelo SNC

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 4



Sistema nervoso

- Sistema nervoso central (SNC)
 - Constituído por:
 - Medula, contida na coluna vertical
 - Encéfalo, massa de tecido nervoso contida na caixa craniana
 - Forma-se nas primeiras fases da vida embrionária de um tubo oco (tubo neural)
 - Durante desenvolvimento do feto, transforma-se em um conjunto de cinco vesículas

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 5



Sistema nervoso

Estímulos →

Sistema nervoso pode ser visto como um sistema de três estágios

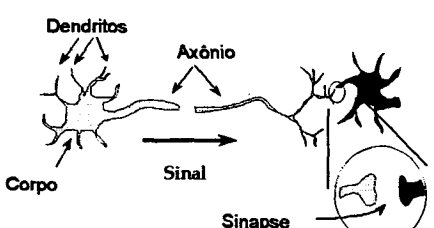
```

graph TD
    Est[Estímulos] --> SNP[SNP]
    SNP --> SNC[SNC]
    SNC --> SNP
    SNC --> Res[Respostas]
  
```

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 6

Neurônio natural

- Um neurônio simplificado:



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP 19

Processamento de informações

- Processo eletro-químico
 - Transmissão de impulsos nervosos dentro do neurônio é um processo de natureza elétrica
 - Sinais viajam através dos axônios em forma de potenciais de ação (potenciais elétricos breves)
 - Transmissão sináptica se faz por um mecanismo de natureza química
 - Através da ação de neurotransmissores

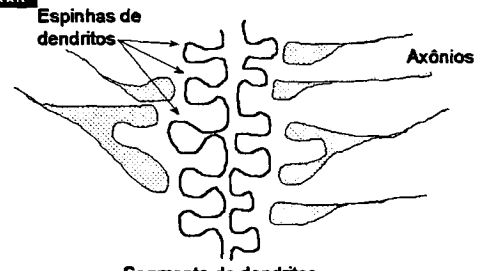
24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP 20

Processamento de informações

- Transmissão de impulsos
 - Dentro de um neurônio as mensagens fluem dos dendritos para o axônio
 - Disparo do axônio depende da soma algébrica de dos impulsos recebidos
 - Neurônio dispara quando a soma supera um dado valor (limiar de ação)

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP 21

Sinapses



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP 22

Sinapses

- Região onde dois neurônios entram em contato, e através da qual os impulsos nervosos são transmitidos de neurônio para neurônio
- Axônio → dendritos
- Efeito
 - Excitatório: estimulam ação do neurônio
 - Inibitório: efeito contrário

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP 23

Sinapses

- Cada pulso que ocorre em uma sinapse inicia liberação de neurotransmissores
 - Substâncias químicas
 - Viajam através dos cleft de sinapses
 - Recebidos nos locais receptores pós-sinápticos dos dendritos
 - Modificam o potencial da membrana dos dendritos (potencial pós-sináptico, PPS)

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP 24



Sinapses

- **Potencial pós sináptico**
 - Aumenta polarização na membrana pós sináptica (hiperpolariza)
 - Inibe geração de pulsos pelo neurônios aferente
 - Reduz polarização na membrana pós sináptica (depolariza)
 - Estimula geração de pulsos pelo neurônios aferente

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

25



Sinapses

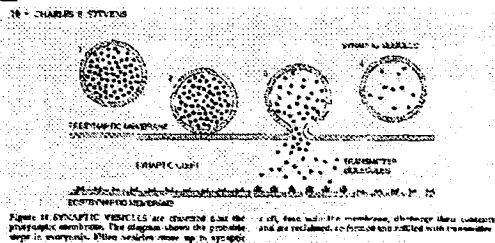
- **Potencial pós-sináptico (continuação)**
 - Tamanho e tipo de PSP produzido depende de:
 - Geometria da sinapse
 - Tipo de neurotransmissor
- Foram descobertos cerca de 50 neurotransmissores diferentes
 - Cada um produz um efeito diferente

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

26



Sinapses

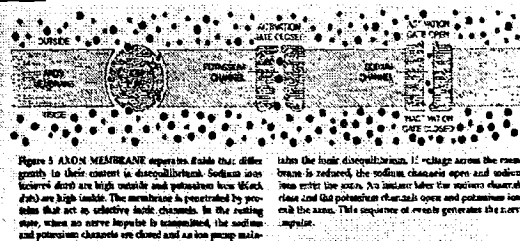


24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

27



Sinapses



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

28



Sinapses

- **Raramente é necessário ultrapassar três sinapses no interior do cérebro**
 - Com quatro sinapses, praticamente qualquer neurônio em local pode se comunicar com qualquer outro
 - Percepção, maioria das recuperações de memória, boa parte do processamento de linguagens e raciocínio intuitivo ocorrem em não mais que 100 passos seriais

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

29



Conexões do sistema nervoso

- **Alto grau de precisão e especificidade**
- **Estabelecimento das conexões**
 - Nativismo: conexões são definidas geneticamente
 - Empirismo: conexões são determinadas pela experiência
 - Combinação das duas correntes

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

30



Características adicionais

- Seres humanos desenvolvem células nervosas até os dezoito meses
 - Conexões são estabelecidas até os três anos
- Seres humanos perdem 10^3 células por dia
 - Aos 70 anos 2,5% de células a menos que aos 10 anos
 - Plasticidade no cérebro adulto
 - Criação de novas sinapses
 - Modificação das sinapses existentes

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

31



Sistema visual

- Na Grécia antiga, acreditava-se que visão era uma espécie de tato
- Fortemente relacionado ao estudo do cérebro
 - Cortex visual é uma das regiões mais estudadas
- Componentes básicos
 - Retina
 - Nervo ótico

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

32



Sistema visual

- Sistema hierárquico
 - Retina
 - Fotoreceptores
 - Bastonetes
 - Cones
 - Células ganglionares
 - Núcleo geniculado lateral
 - Cortex visual



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

33



Retina

- Fotoreceptores: traduzem sinal luminoso em estímulos eletroquímicos
 - Bastonetes
 - Especializados em visão noturna
 - Baixa resolução espacial
 - Visão monocromática
 - Cones
 - Dedicados ao processamento de imagens claras
 - Amostram o sinal luminoso com maior resolução espacial
 - Indispensáveis para visão colorida

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

34



Retina

- Retina possui duas regiões especiais
 - Ponto cego
 - Local onde o nervo ótico deixa a retina
 - Axônio das células ganglionares que constituem a última camada neural da retina
 - Fóvea
 - Possui apenas cones
 - Corresponde a 2-5 graus do campo de visão
 - Onde ocorre visão de alta acuidade

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

35



Retina

- Principal função da retina é detecção (realce) das bordas dos objetos em imagens
 - Representação da imagem original por seus elementos de borda corresponde a uma primeira simplificação da imagem a ser analisada
 - Qualquer imagem pode ser recuperada a partir de sua representação em termos de elementos de bordas
 - Esta representação leva a uma economia: Embora existam 126 milhões de receptores no olho, 900.000 axônios são suficientes para levar o estímulo visual até o cortex

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

36



Retina

- Células ganglionares da retina apresentam campos receptivos
 - on-center
 - Centro excitatório, redondeza inibitória
 - off-center
 - Centro inibitório, redondeza excitatória
 - Movimento

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABCUSP

37



Sistema visual

- Nervo ótico liga olho ao núcleo geniculado lateral (LNG) passando pelo quiasma ótico
 - Conexão unidirecional
- LNG
 - Primeiro ponto onde as fibras do nervo ótico realizam conexões sinápticas
 - Campos receptivos semelhantes ao da retina
 - Matém relação entre elementos vizinhos

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABCUSP

38



Cortex Visual

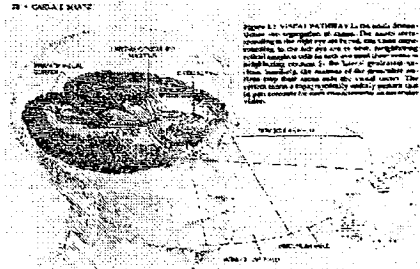
- LNG se conecta com cortex visual primário por conexões bidirecionais
- Situado na região occipital do cérebro
 - Cortex visual primário ou cortex estriado (região V1)
 - Cortex médio-temporal (regiões V2, V3, V4 e V5)
- Formado por seis camadas

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABCUSP

39



Sistema visual

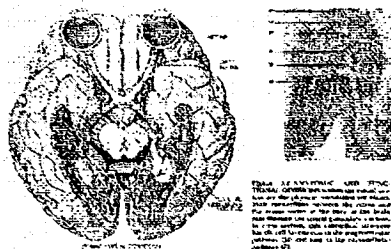


24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABCUSP

40



Sistema visual



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABCUSP

41



Redes Neurais Artificiais

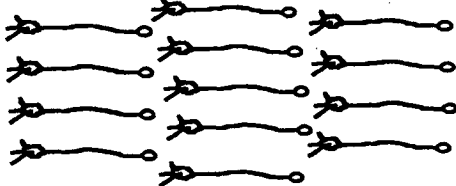
- Redes Neurais Artificiais (RNAs) são tentativas de produzir sistemas de aprendizado biologicamente realísticos
 - São baseadas em modelos abstratos de como pensamos que o cérebro (e os neurônios) funcionam
 - RNAs aprendem por exemplo
 - RNA = arquitetura + aprendizado

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABCUSP

42

Redes Neurais Artificiais

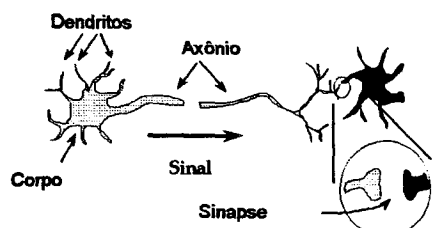
- Simulam a arquitetura do cérebro através de processamento paralelo



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 43

Neurônio natural

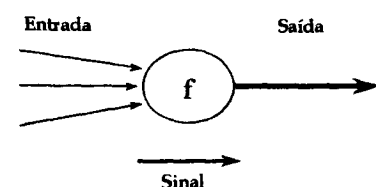
- Um neurônio simplificado:



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 44

Neurônio artificial

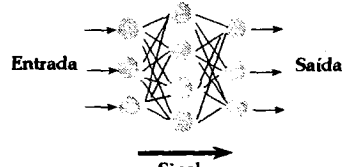
- Modelo de um neurônio (nó) abstrato



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 45

Redes Neurais Artificiais

- Estrutura das RNAs
 - Nós equivalem aos neurônios
 - Connexões são semelhantes as sinapses



24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 46

Conclusão

- Conhecer o funcionamento e estrutura do sistema nervoso é fundamental para o projeto de Redes Neurais Artificiais
 - Cérebro
 - Neurônio
 - Sinapses
- Próxima aula: Fundamentos de Redes Neurais Artificiais

24/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 47



Conceitos Básicos



Prof. André C. P. L. F de Carvalho
LABIC - ICMSC - USP

27/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

1



Conceitos básicos

- Estrutura geral das RNA
 - Unidades de processamento n_i (nós)
 - Estado de ativação a_i
 - Função de ativação F_i
 - Função de saída f_i
 - Conexões w_{ij}
 - Topologia

27/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

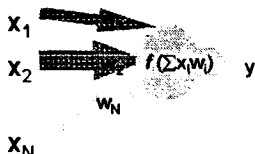
2



Unidades de processamento

- Função: receber entradas de conjunto de unidades A, computar função sobre entradas e enviar resultado para conjunto de unidades B
- Entrada total

$$u = \sum_{j=1}^N x_j w_j$$



27/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

3



Unidades de processamento

- Representação
 - Local: unidades representam objetos bem definidos (Ex. letras, palavras, faces, ...)
 - Distribuída: unidades representam elementos abstratos
- Localização das unidades
 - Intermediária (escondida)
 - Saída

27/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

4



Unidades de processamento

- Estado de ativação
 - Representa o estado dos neurônios da rede
 - Pode assumir valores
 - Binários (0 e 1)
 - Bipolares (-1 e +1)
 - Reais
 - Definido através de funções de ativação

27/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

5



Funções de ativação

- Processa conjunto de entradas recebidas e o transforma em estado de ativação
- Funções de ativação típicas envolvem:
 - Adições
 - Comparações
 - Transformações matemáticas

27/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

6



Funções de ativação

● Função de ativação

■ Atualiza estado de ativação

- $a(t+1) = F[a(t), u(t)]$
- $a(t+1) = F[a(t)]$
- $a(t+1) = F[u(t)]$

■ Atualização

- Síncrona (mais comum)
- Assíncrona



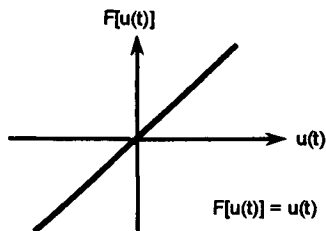
Funções de ativação

● Funções de ativação mais comuns

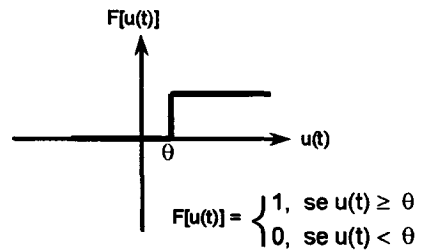
- $a(t+1) = u(t)$ (linear)
- $a(t+1) = \begin{cases} 1, & \text{se } u(t) \geq \theta \\ 0, & \text{se } u(t) < \theta \end{cases}$ (threshold ou limiar)
- $a(t+1) = 1/(1 + e^{-\lambda u(t)})$ (sigmoid logística)
- $a(t+1) = \frac{(1 - e^{-\lambda u(t)})}{(1 + e^{-\lambda u(t)})}$ (tangente hiperbólica)



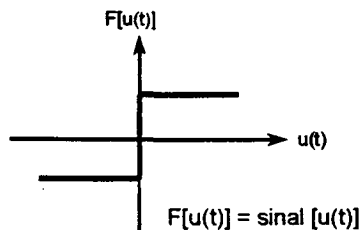
Função linear



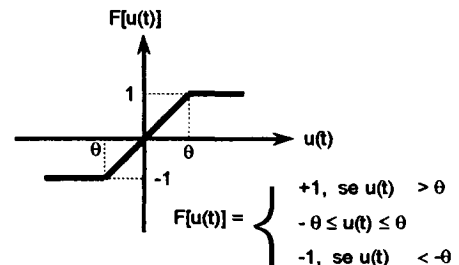
Função threshold unipolar



Função threshold bipolar

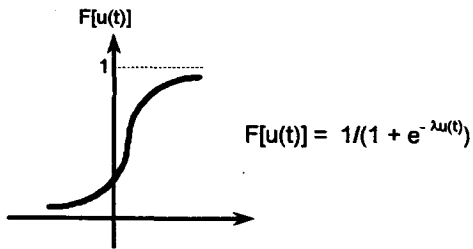


Função piecewise linear

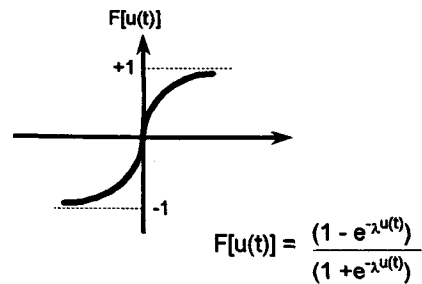




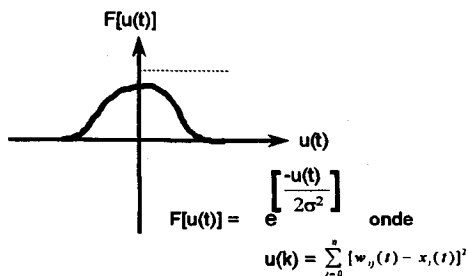
Função sigmoid logística (unipolar)



Função sigmoid bipolar



Função Gaussiana



Funções de saída

● Função de saída

- Transforma estado de ativação de uma unidade em seu sinal de saída

$$y_i(t) = f_i(a_i(t))$$

- Geralmente é uma função identidade



Valores de entrada e saída

- Sinais de entrada e saída de uma RNA geralmente são números reais
 - Números devem estar dentro de um intervalo
 - Tipicamente entre -1 e +1 ou 0 e 1
 - Codificação realizada pelo projetista da rede
- Técnica de codificação mais simples é a binária
 - Número restrito de aplicações



Conexões

- Definem como neurônios estão interligados
 - Nós são conectados entre si através de conexões específicas
- Codificam conhecimento da rede
 - Uma conexão geralmente tem um valor de ponderamento ou **peso** associada a ela



Conexões

- Tipos de conexões ($w_{ik}(t)$)
 - Excitatória: ($w_{ik}(t) > 0$)
 - Inibitória: ($w_{ik}(t) < 0$)
 - Conexão inexistente: ($w_{ik}(t) = 0$)
- Número de conexões de um nó
 - Fan-in: número de conexões de entrada
 - Fan-out: número de conexões de saída

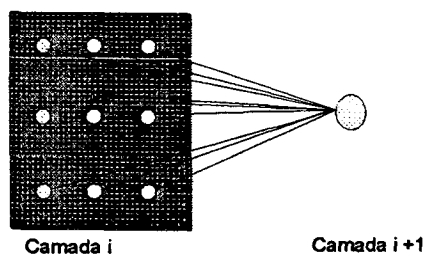


Topologia

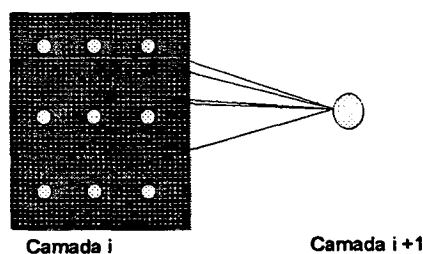
- Número de camadas
 - Uma camada (Ex Perceptron, Adaline)
 - Multi-camadas (Ex MLP)
 - Completamente conectada
 - Parcialmente conectada
 - Localmente conectada



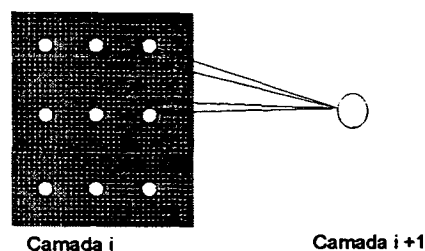
Completamente conectada



Parcialmente conectada



Parcialmente conectada



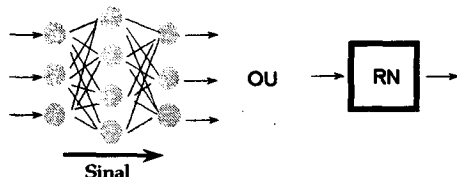
Topologia

- Arranjo das conexões
 - Redes feedforward
 - Não existem loops de conexões
 - Redes recorrentes
 - Conexões apresentam loops
 - Mais utilizadas em sistemas dinâmicos
 - Lattices
 - Matriz n-dimensional de neurônios



Redes feedforward

- Sinais seguem em uma única direção

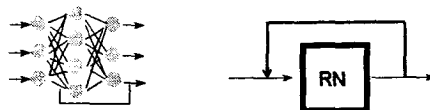


- Tipo mais comum



Redes recorrentes

- Possuem conexões ligando saída da rede a sua entrada



- Podem lembrar entradas passadas e, conseqüentemente, processar sequência de informações (no tempo ou espaço)



Projeto de Redes Neurais

- Projeto de sistemas convencionais
 - Formular modelo matemático a partir de observações do ambiente
 - Validar o modelo com dados reais
 - Construir o sistema utilizando o modelo
- Projeto de Redes Neurais
 - Baseado apenas nos dados
 - Exemplos para treinar uma rede devem ter padrões positivos e negativos



Conjunto de dados

- Tamanho depende da complexidade dos dados
 - Quanto maior a complexidade, maior a quantidade necessária
 - Pré-processamento dos dados
 - Dados numéricos
 - Presença de valores em todos os campos
 - Balanceamento entre classes



Pré-processamento dos dados

- Conversão de valores simbólicos para valores numéricos
 - Quando os valores não apresentam ordenação
 - Vetores de valores binários, cada um com um elemento igual a 1 e os demais iguais a 0
 - Tamanho do vetor igual ao número de valores diferentes
 - Quando os valores apresentam ordenação
 - Codificar cada valor por um número real
 - Codificar cada valor por um vetor binário, utilizando código cinza



Pré-processamento dos dados

- Estimativa de valores ausentes
 - Média de todos os valores do mesmo campo
 - Média entre anterior e posterior
 - Criação de um novo valor
- Normalização de valores numéricos
 - Normalizar cada campo individualmente
 - Assegurar que todos os valores de um dado campo estejam dentro de um intervalo (Ex. [0.0..., 1.0])



Criação de conjuntos de dados

- Dividir o conjunto de dados em:
 - Subconjunto de treinamento (50% dos dados)
 - Necessidade de manter uniformidade pode reduzir proporção
 - Subconjunto de validação (25% dos dados)
 - Subconjunto de teste (25% dos dados)
 - Dividir dados aleatoriamente entre os conjuntos
 - Conjuntos devem ser disjuntos
- Repetir o processo acima três vezes
 - Criando três conjuntos, cada um com subconjuntos de treinamento, validação e teste

27/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

31



Projeto da rede

- Escolha do modelo
- Selecionar arquitetura adequada para a rede
 - Número de camadas
 - Número de nós da camada de entrada igual ao de pixels (campos) do vetor de entrada
 - Pré-processamento pode aumentar ou diminuir número de campos

27/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

32



Conclusão

- Conceitos básicos
- Unidades de processamento
- Arquitetura
- Inclusão de conhecimento no projeto de Redes Neurais
- Projeto de Redes Neurais
- A seguir: primeiras Redes Neurais

27/04/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

33




Aprendizado

Prof. André C. P. L. F de Carvalho
LABIC - ICMSC - USP

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 1



Aprendizado

- Capacidade de aprender a partir de seu ambiente e melhorar sua performance com o tempo
- Parâmetros livres de uma RNA são adaptados através de estímulos fornecidos pelo ambiente
 - Processo iterativo de ajustes aplicado a sinapses e thresholds
 - Idealmente, a RNA sabe mais sobre seu ambiente após cada iteração

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 2



Aprendizado

- RNA deve produzir para cada conjunto de entradas apresentado o conjunto de saídas desejado
 - $w_{ik}(t+1) = w_{ik}(t) + \Delta w_{ik}(t)$

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 3



Aprendizado

- Mecanismos de aprendizado
 - Modificação de pesos ($\Delta w_{ij}(t)$) associados às conexões
 - Armazenamento de novos valores em conteúdos de memória
 - Acréscimo e/ou eliminação de conexões/neurônios

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 4



Aprendizado

- Algoritmos de aprendizado
 - Conjunto de regras bem definidas para a solução de um problema de aprendizado
 - Grande variedade
 - Cada um com suas vantagens
 - Diferem na maneira como ajuste $\Delta w_{ik}(t)$ é realizado
- Paradigmas de aprendizado
 - Diferem na maneira como RNA se relaciona com seu ambiente

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 5



Aprendizado

```

graph TD
    A[Processos de Aprendizado] --> B[Algoritmos de Aprendizado]
    B --> C[Aprendizado por correção de erro]
    B --> D[Aprendizado competitivo]
    B --> E[Aprendizado Hebbiano]
    B --> F[Aprendizado de Boltzmann]
    C --> G[Paradigmas de Aprendizado]
    D --> G
    E --> G
    F --> G
    G --> H[Aprendizado supervisionado]
    G --> I[Aprendizado de reforço]
    G --> J[Aprendizado não supervisionado]
  
```

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 6

Aprendizado por correção de erro

- Regra Delta (Widrow e Hoff 1960)
- Erro: $e_k(t) = d_k(t) - y_k(t)$
- Minimizar função de custo baseada em $e_k(t)$
- Função de custo
 - $c(t) = -1/2 \sum e_k^2(t)$
 - Minimização de $c(t)$ utiliza método de gradiente descendente
 - Aprendizado atinge solução estável quando os pesos não precisam mudar muito

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

7

Aprendizado por correção de erro

- Após seleção da função de custo, aprendizado se torna um problema de otimização
 - RNA é otimizada pela minimização de $c(t)$ com respeito aos pesos da rede
- Modelo matemático:
 - $\Delta w_{ik}(t) = \eta e_k(t) x_i(t)$

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

8

Aprendizado por correção de erro

- Superfície de erro: superfície multi-dimensional representando gráfico da função de custos X peso
 - Unidades lineares: superfície é uma função quadrática dos pesos (mínimo global único)
 - Unidades não-lineares: superfície tem mínimos locais e mínimo(s) global(is)
- Começando em um ponto qualquer da superfície mover em direção a um mínimo global

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

9

Aprendizado por correção de erro

- Taxa de aprendizado (η)
 - $0 < \eta < 1$
 - Taxas pequenas
 - Média das entradas anteriores
 - Estimativas estáveis de peso
 - Aprendizado lento
 - Taxas grandes
 - Aprendizado rápido
 - Captação de mudanças no processo
 - Instabilidade
 - Taxas variáveis

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

10

Aprendizado Hebbiano

- Regra mais antiga e famosa (Hebb 1949)
- Dois neurônios estão simultaneamente ativos, a conexão entre eles deve ser fortalecida
- Regra modificada (sinapse Hebbiana)
 - Dois neurônios são ativados sincronamente, força da sinapse deve ser aumentada
 - Dois neurônios são ativados assincronamente, força da sinapse deve ser reduzida
 - Sinapse anti-Hebbiana
 - Sinapse não Hebbiana

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

11

Aprendizado Hebbiano

- Propriedades da sinapse Hebbiana
 - Dependência do tempo
 - Local
 - Interativo
 - Correlacional
- Depressão sináptica: enfraquecimento de uma sinapse com o passar do tempo

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

12



Aprendizado Hebbiano

● Modelo matemático:

- $\Delta w_{ik}(t) = f(y_k(t)x_i(t))$
- $\Delta w_{ik}(t) = \eta y_k(t)x_i(t)$
- $\Delta w_{ik}(t) = \eta y_k(t)x_i(t) - \alpha y_k(t)w_{ik}(t)$



Aprendizado competitivo

- Von der Marlsburg (1973)
- Neurônios competem entre si para ser ativo
 - Apenas um neurônio se torna ativo
- Adequado para descobrir características estatisticamente salientes
 - Podem classificar conjuntos de entradas



Aprendizado competitivo

● Elementos básicos

- Conjunto de neurônios iguais (exceto por alguns pesos randomicamente distribuídos)
- Limite no poder de cada neurônio
- Mecanismo que permita neurônios competirem pelo direito de responder a um dado subconjunto de entradas (*Winner-takes-all*)
- Neurônios individuais se especializam em conjuntos de padrões semelhantes



Aprendizado competitivo

● Aprendizado competitivo mais simples

- Uma camada de neurônios completamente conectados a entrada (excitatória)
- Conexões laterais entre neurônios (inibitórias)
- $\sum w_{ik}^2(t) = 1$ (para cada neurônio)

● Modelo matemático:

- $\Delta w_{ik}(t) = \eta (x_i(t) - w_{ik}(t))$ (vencedor)



Aprendizado de Boltzmann

- Algoritmo estocástico
- Baseado em termodinâmica e teoria da informação
- Arquitetura recorrente
- Neurônios com estados bipolares
- Caracterizado por uma função de energia
 - Valor determinado pelos estados dos neurônios
 - $E = -1/2 \sum_i \sum_j w_{ij} s_i s_j$ ($i \neq j$)



Aprendizado de Boltzmann

● Operação

- Escolher um neurônio n_i aleatoriamente
- Alterar o estado de n_i com probabilidade
 - $P = \frac{1}{1 + \exp(-\Delta E_i / T)}$
 - ΔE_i : Mudança de energia resultante
 - T : Pseudo temperatura
 - Regra aplicada até rede atingir equilíbrio termal



Aprendizado de Boltzmann

- Tipos de neurônios
 - Visíveis
 - Escondidos
- Modos de operação
 - Preso (neurônios visíveis)
 - Livre



Aprendizado de Boltzmann

- Modelo matemático
 - $w_{ik}(t) = \eta(\rho_{ik}^+ - \rho_{ik}^-)$
 - ρ_{ik}^+ : Correlação positiva entre neurônios n_i e n_k
 - ρ_{ik}^- : Correlação negativa entre neurônios n_i e n_k
 - Correlações variam de -1 a +1



Aprendizado supervisionado

- Professor externo
 - Possui conhecimento sobre ambiente
 - Representado por conjunto de pares (x, d)
 - Geralmente, a rede não possui informações prévias sobre ambiente
 - Parâmetros da rede são ajustados por (x, d)
 - Rede procura emular professor



Aprendizado supervisionado

- Problema de atribuição de crédito
 - Atribuir crédito ou culpa pelo resultado à cada uma das decisões internas que contribuíram para ele
- Para rede melhorar desempenho, o custo deve mover para ponto de mínimo na superfície de erro
 - Utiliza informação sobre gradiente da superfície para os parâmetros atuais da rede
 - Gradiente: vetor que aponta na direção da descida mais íngreme



Aprendizado supervisionado

- Formas de aprendizado supervisionado
 - Offline (estático)
 - Módulo externo para aprendizado
 - Rede é congelada após o treinamento
 - Online (dinâmico)
 - Rede nunca para de ser treinada
 - Aprendizado auto-contido
- Desvantagem: dependência do professor



Aprendizado por reforço

- Crítico externo
 - Processo de tentativa e erro
 - Procura maximizar sinal de reforço
- Se ação tomada por sistema é seguida por estado satisfatório, sistema é fortalecido, caso contrário, sistema é enfraquecido (lei de Thorndike)



Aprendizado por reforço

- Tipos de reforço
 - Positivo = recompensa
 - Negativo = punição
 - Nulo



Aprendizado por reforço

- Formas de aprendizado por reforço
 - Não associativo
 - Sistema deve selecionar ação ótima
 - Reforço é única entrada recebida
 - Associativo
 - Aprender mapeamento estímulo-ação
 - Informação adicional é fornecida



Aprendizado por reforço

- Conflito identificação-controle
 - Desejo de utilizar conhecimento disponível sobre mérito de ações tomadas pelo sistema
 - Desejo de adquirir mais conhecimento sobre conseqüências de ações para selecionar melhor no futuro
 - Quando parar?



Supervisionado X Reforço

Aprendizado supervisionado	Aprendizado por reforço
Professor	Crítico
Sistema de feedback instrutivo	Sistema de feedback estimativo
É dito o que fazer	Faz as coisas e ver o que acontece
Mais rápido	mais lento



Aprendizado não supervisionado

- Não tem crítico ou professor externo
- Extração de características estatisticamente relevantes
 - Cria classes automaticamente
- Aprendizado não supervisionado X supervisionado
 - Problema de escala
 - Rede modular



Tarefas de aprendizado



Escolha do procedimento de aprendizado é influenciada pela tarefa a ser realizada pela rede

Principais tarefas de aprendizado

Tarefa	Exemplo	Paradigma
Aproximação	Dado pares (x, d) , aproximar função	Supervisionado
Associação		
Autoassociação	Associar x com x	Não supervisionado
Heteroassociação	Associar x com y	Supervisionado
Classificação de padrões	Categorias definidas	Supervisionado
	Cluster, detecção de características	Não supervisionado
Previsão	Série temporal	Supervisionado
Controle	Corpo humano	Reforço
Localização	Localização de alvo	Não supervisionado

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 31

Adaptação e aprendizado

- Dimensões do processo de aprendizado
 - Espaço
 - Tempo
- Ambientes de operação
 - Estacionário
 - Não estacionário

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 32

Ambientes de operação

- Estacionário: Características estatísticas não mudam com o tempo
 - Aprendizado supervisionado
 - Parâmetros da rede são congelados após treinamento
 - Sistema de aprendizado utiliza memória para lembrar e explorar experiências passadas

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 33

Ambientes de operação

- Não estacionário: Características estatísticas mudam com o tempo
 - Rede deve adaptar seus parâmetros em tempo real
 - Sistema adaptativo responde a toda entrada diferente como a uma nova entrada
 - Quando operando em um ambiente não estacionário, quanto mais adaptativo o sistema, melhor sua performance

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 34

Ambientes de operação

- Aprendizado de estrutura temporal
 - Treinamento contínuo com exemplos ordenados pelo tempo (Previsão)
 - Extensão de métodos tradicionais de aprendizado supervisionado

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 35

Conclusão

- Aprendizado
- Paradigmas
- Algoritmos
- Escolha

01/06/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP 36



Perceptrons



Prof. Dr. André C. P. L. F de Carvalho
LABIC - ICMSC - USP

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

1



Perceptrons

- Desenvolvido por Rosembat, 1958
- Rede mais simples para classificação de padrões linearmente separáveis
- Utiliza modelo de McCulloch-Pitts como neurônio

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

2



Perceptrons

- Estado de ativação

- 1 = ativo
- -1 = inativo

- Função de ativação

$$a_i(t+1) = f(u_i(t))$$

$$a_i(t+1) = \begin{cases} -1, & \text{se } u_i(t) < 0 \\ +1, & \text{se } u_i(t) \geq 0 \end{cases} \quad u = \sum_{j=1}^N x_j w_j - \theta$$

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

3



Perceptrons

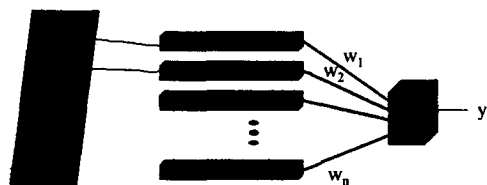
- Função de saída = função identidade
- Duas camadas
 - Camada de pré-processamento
 - M máscaras fixas para extração de características
 - Podem implementar qualquer função, mas pesos são fixos
 - Camada de discriminação
 - Uma unidade de saída para discriminar padrões de entrada
 - Pesos determinados através de aprendizado

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

4



Perceptrons



8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

5



Perceptrons

- Treinamento
 - Supervisionado
 - Correção de erro
 - $\Delta w_{ij} = \eta x_i (d_j - y_j)$ ($d \neq y$)
 - $\Delta w_{ij} = 0$ ($d = y$)
- Teorema de convergência: se é possível classificar um conjunto de entradas, uma rede Perceptron fará a classificação

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

6

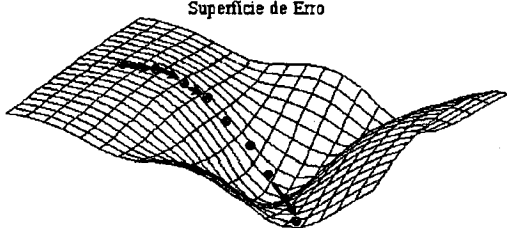
Algoritmo de treinamento

- 1) Iniciar todas as conexões com $w_{ij} = 0$;
- 2) Repita
 - Para cada par de treinamento (X, d)
 - Calcular a saída y
 - Se $(d \neq y)$
 - Então
 - Atualizar pesos dos neurônios
 - Até o erro ser aceitável

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP 7

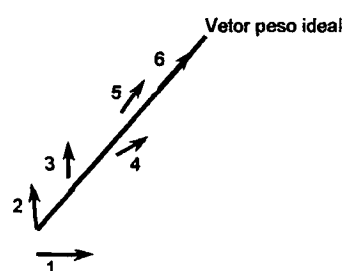
Treinamento

Superfície de Erro



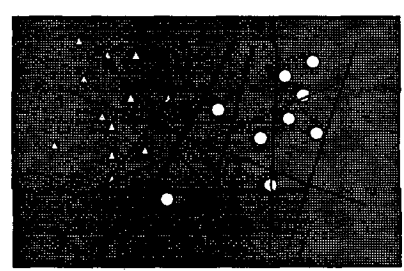
8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP 8

Treinamento



8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP 9

Treinamento



8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP 10

Algoritmo de teste

- 1) Apresentar padrão X a ser reconhecido
- 2) Calcular a saída y
- 3) Se $(d \neq y)$
- Então
- $X \in \text{classe } 0$
- Senão
- $X \in \text{classe } 1$

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP 11

Exemplo

Dada uma rede do tipo Perceptron formada por um com três terminais de entrada, utilizando pesos iniciais $w_0 = 0.4$, $w_1 = -0.6$ e $w_2 = 0.6$, limiar $\theta = 0.5$ e uma taxa de aprendizado $= 0.4$, responda os itens abaixo:

- a) Ensinar a rede a gerar a saída -1 para o padrão 001 e a saída $+1$ para os padrões 110
- b) A que classe pertencem os padrões 111, 000, 100 e 011?

8/12/98 Sistemas Inteligentes - André Ponce de Leon F. de Carvalho - LABICUSP 12



Exemplo 1: resposta a

a) Treinar a rede

a.1) Para o padrão 001 (d = -1)

Passo 1: definir a saída da rede

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

$$y = u = +1 \text{ (uma vez } 0.1 \geq 0)$$

Passo 2: atualizar os pesos

$$w_0 = 0.4 + 0.4(0)(-1 - (+1)) = 0.4$$

$$w_1 = -0.6 + 0.4(0)(-1 - (+1)) = -0.6$$

$$w_2 = 0.6 + 0.4(1)(-1 - (+1)) = -0.2$$

$$w_3 = 0.5 + 0.4(-1)(-1 - (+1)) = 1.3$$



Exemplo 1: resposta a

a) Treinar a rede

a.2) Para o padrão 110 (d = 1)

Passo 1: definir a saída da rede

$$u = 1(0.4) + 1(-0.6) + 0(-0.2) - 1(1.3) = -1.5$$

$$y = u = -1 \text{ (uma vez } -1.5 < 0)$$

Passo 2: atualizar pesos

$$w_0 = 0.4 + 0.4(1)(1 - (-1)) = 1.2$$

$$w_1 = -0.6 + 0.4(1)(1 - (-1)) = 0.2$$

$$w_2 = -0.2 + 0.4(0)(1 - (-1)) = -0.2$$

$$w_3 = 1.3 + 0.4(-1)(1 - (-1)) = 0.5$$



Exemplo 1: resposta a

a) Treinar a rede

a.3) Para o padrão 001 (d = -1)

Passo 1: definir a saída da rede

$$u = 0(1.2) + 0(0.2) + 1(-0.2) - 1(0.5) = -0.7$$

$$y = u = -1 \text{ (uma vez } -0.7 < 0)$$

Passo 2: atualizar pesos

Como d = y, os pesos não precisam ser modificados



Exemplo 1: resposta a

a) Treinar a rede

a.4) Para o padrão 110 (d = 1)

Passo 1: definir a saída da rede

$$u = 1(1.2) + 1(0.2) + 0(-0.2) - 1(0.5) = 0.9$$

$$y = u = 1 \text{ (uma vez } 0.9 \geq 0)$$

Passo 2: atualizar pesos

Como d = y, os pesos não precisam ser



Exemplo 1: resposta b

b) Testar a rede

b.1) Para o padrão 111

$$u = 1(1.2) + 1(0.2) + 1(-0.2) - 1(0.5) = 0.7$$

$$y = u = 1 \text{ (porque } 0.7 \geq 0) \Rightarrow \text{classe 1}$$

b.2) Para o padrão 000

$$u = 0(1.2) + 0(0.2) + 0(-0.2) - 1(0.5) = -0.5$$

$$y = u = -1 \text{ (porque } -0.5 < 0) \Rightarrow \text{classe 0}$$



Exemplo 1: resposta b

b) Testar a rede

b.3) Para o padrão 100

$$u = 1(1.2) + 0(0.2) + 0(-0.2) + 1(-0.5) = 0.7$$

$$y = u = 1 \text{ (porque } 0.7 \geq 0) \Rightarrow \text{classe 1}$$

b.4) Para o padrão 011

$$u = 0(1.2) + 1(0.2) + 1(-0.2) - 1(0.5) = -0.5$$

$$y = u = -1 \text{ (porque } -0.5 < 0) \Rightarrow \text{classe 0}$$



Madaline

- Trata algumas funções não linearmente separáveis
 - Cada Adaline pode estar associado a uma reta
- Multicamadas
 - Primeira camada
 - Adaptativa
 - Várias redes Adaline
 - Segunda camada
 - Fixa
 - Funções AND, OR, maioria

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

25



Conclusão

- Reconhecimento de Padrões
- Perceptrons
 - Não leva distâncias em consideração
- Adaline
 - Classes linearmente separáveis
- Madaline

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

26



Conclusão

- Redes de uma camada mapeiam entradas semelhantes em saídas semelhantes
 - Similaridade determinada pelo *overlap* dos padrões
 - Impedem o sistema de aprender certos mapeamentos
 - Estrutura de similaridade das entradas e das saídas são diferentes
 - Exemplos: paridade, ou-exclusivo

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

27



Conclusão

- Nestes casos, a rede precisa criar uma representação interna das entradas
 - Necessário utilizar uma ou mais camadas intermediárias
 - Como treinar redes com mais de uma camada?
 - Aprendizado competitivo
 - Assumir uma representação interna razoável
 - Desenvolver um algoritmo de aprendizado capaz de aprender uma representação interna adequada

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

28



Redes Neurais do tipo MLP



Prof. Dr. André C. P. L. F de Carvalho
LABIC - ICMSC - USP

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 1



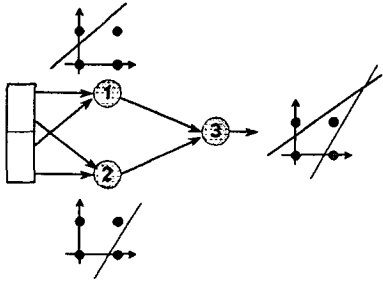
Introdução

- Redes de uma camada resolvem apenas problemas linearmente separáveis
- Solução: utilizar mais de uma camada
 - Camada 1: uma rede Perceptron para cada grupo de entradas linearmente separáveis
 - Camada 2: uma rede combina as saídas das redes da primeira camada, produzindo a classificação final

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 2



Introdução



8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 3



Introdução

- Treinamento da rede
 - Treinar cada rede independentemente
 - Saber como dividir o problema em sub-problemas
 - Nem sempre é possível
 - Treinar a rede toda
 - Qual o erro dos neurônios da camada intermediária?
 - Função threshold leva ao problema de atribuição de crédito
 - Usar função de ativação linear?

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 4



Introdução

- Função de ativação linear
 - Cada camada computa uma função linear
 - Composição de funções lineares é uma função linear
 - Sempre vai existir uma rede com uma camada equivalente uma rede multicamadas com funções de ativação lineares

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 5



Introdução

- Função de ativação para redes multicamadas
 - Não deve ser linear
 - Deve informar os erros para as camadas inferiores da rede
 - Função sigmóid
 - Função piecewise (linear threshold)

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 6



Rede Multi-Layer Perceptron

- Arquitetura de RNA mais utilizada
- Possuem uma ou mais camadas intermediárias de nós
 - Geralmente utiliza função de ativação sigmóid logística

8/12/98 Sistemas Inteligentes - André Ponce de Leon F. de Carvalho - LABIC/USP

7



Treinamento de redes MLP

- Grande variedade de Algoritmos
 - Geralmente supervisionados
 - Estáticos
 - Não alteram estrutura da rede
 - Backpropagation, Função de Base Radial
 - Construtivos
 - Alteram estrutura da rede
 - Upstar, Cascade Correlation

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

8



Treinamento de redes MLP

- Treinamento estático
 - MLPs com formatos e tamanhos diferentes podem utilizar mesma regra de aprendizado
 - Topologias diferentes podem resolver o mesmo problema
 - Regra mais utilizada: backpropagation

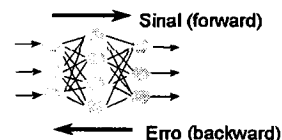
8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

9



Backpropagation

- Rede é treinada com pares entrada-saída
- Cada entrada de treinamento está associada a uma saída desejada
- Treinamento em duas fases, cada uma percorrendo a rede em um sentido
 - Fase forward
 - Fase backward

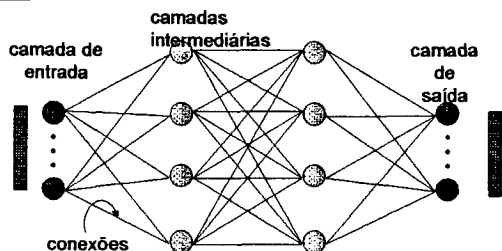


8/12/98 Sistemas Inteligentes - André Ponce de Leon F. de Carvalho - LABIC/USP

10



Rede MLP

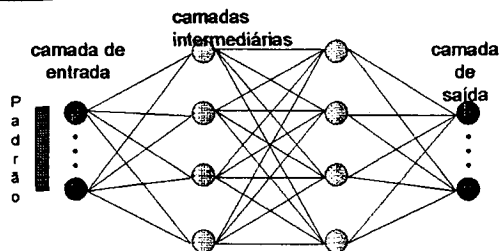


8/12/98 Sistemas Inteligentes - André Ponce de Leon F. de Carvalho - LABIC/USP

11

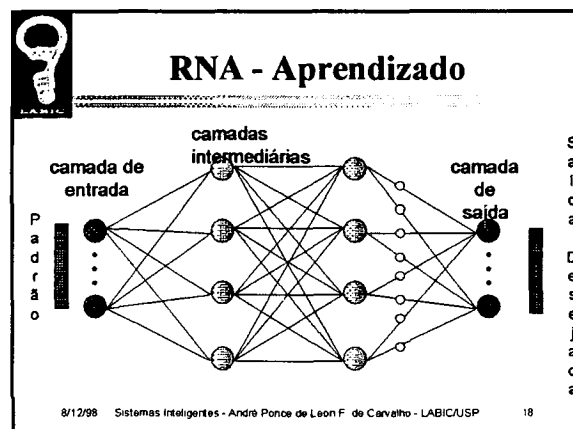
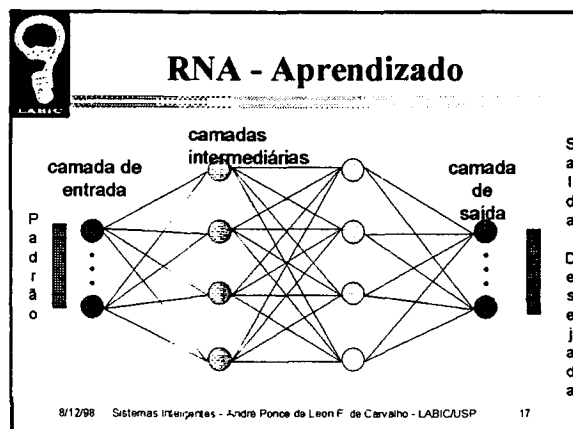
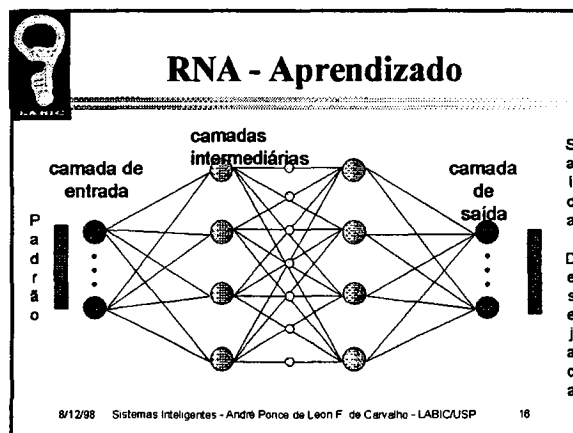
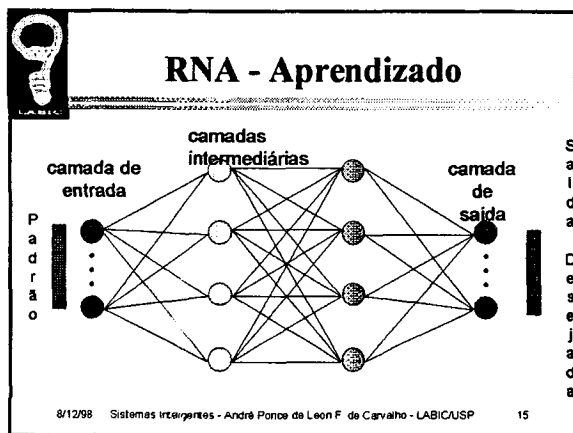
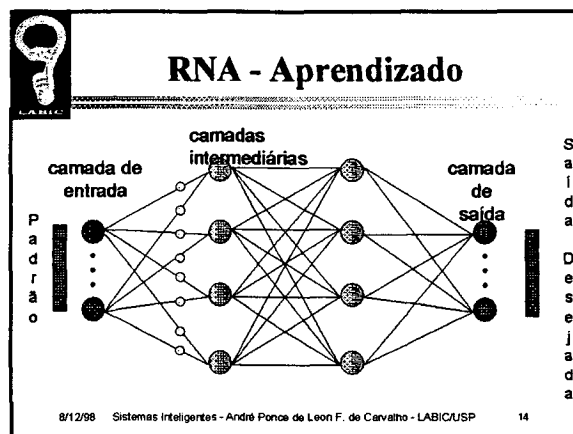
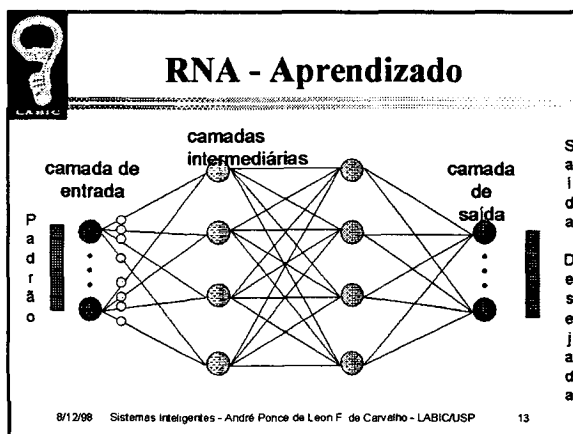


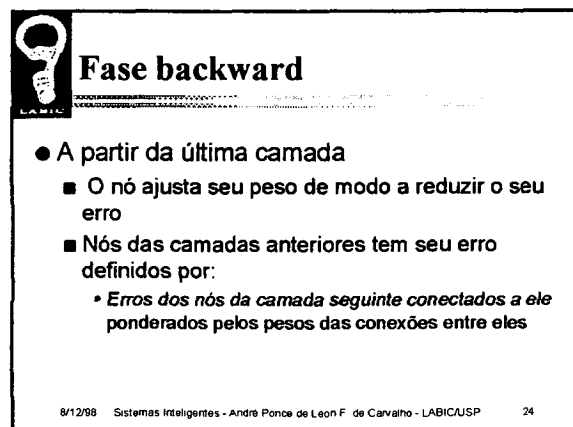
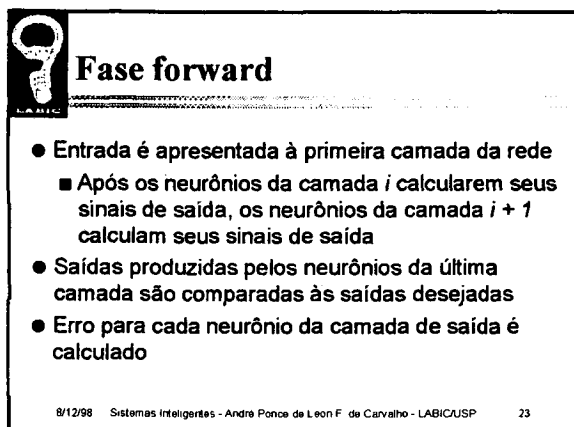
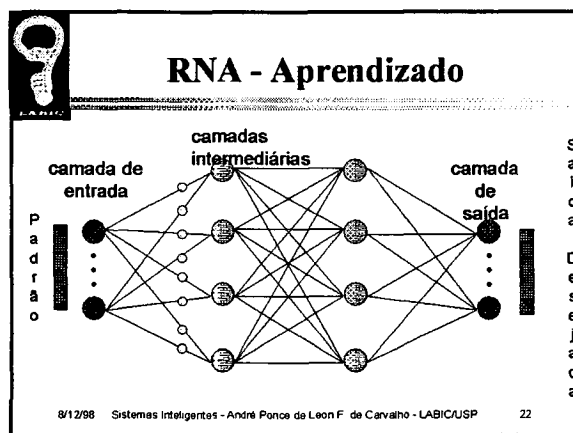
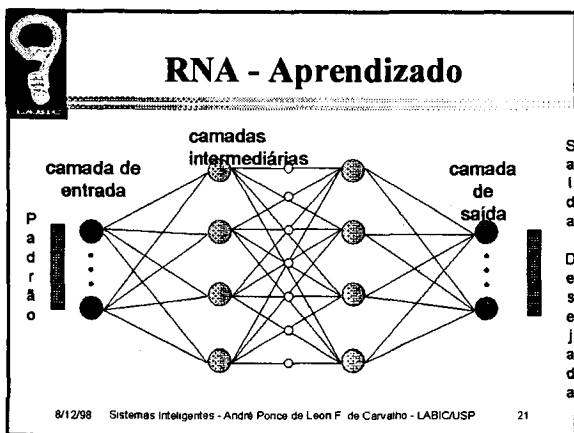
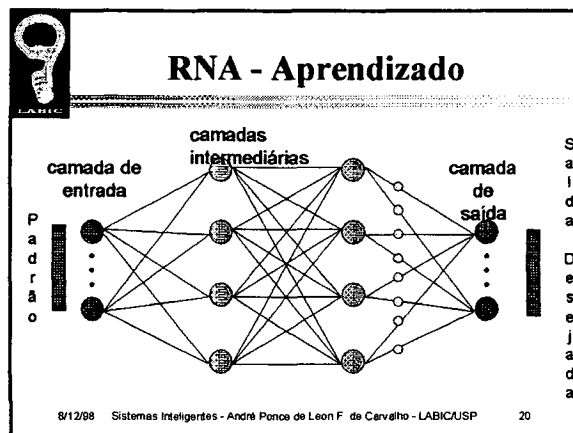
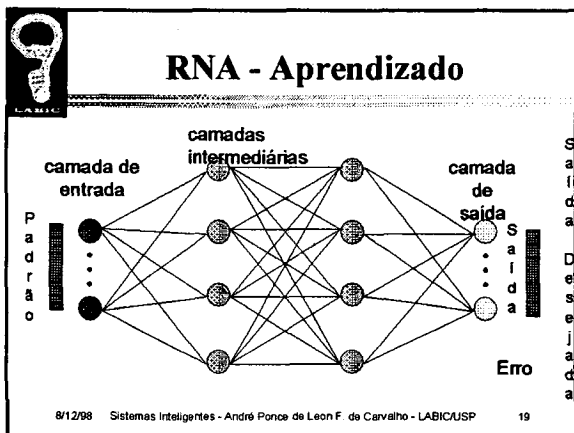
Aprendizado



8/12/98 Sistemas Inteligentes - André Ponce de Leon F. de Carvalho - LABIC/USP

12







Backpropagation

- Treina redes MLP produzindo representações internas necessárias para nós intermediários
- Supor que cada combinação de pesos e thresholds corresponda a um ponto em uma superfície de solução
 - Solução = pontos mais baixos da superfície
 - Procura minimizar erro ajustando pesos e thresholds para que eles correspondam aos pontos mais baixos da superfície
 - método do gradiente descendente

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

25



Backpropagation

- Gradiente de uma função está na direção e sentido onde a função tem taxa de variação máxima
 - Garantido de achar uma solução para superfícies simples
- Backpropagation fornece aproximação da trajetória no espaço de peso computado pelo método do gradiente descendente

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

26



Backpropagation

- Processamento
 - Forward (teste)
 - Backward (treinamento)
- Estados de ativação
 - 1 (+1) = ativo
 - 0 (-1) = inativo

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

27



Backpropagation

- Função de ativação
 - Não linear
 - Diferenciável, contínua e, geralmente, não decrescente
 - Sigmoidal
 - $a_i(t+1) = 1/(1 + e^{-u_i(t)})$ (sigmoidal logística)
 - $a_i(t+1) = \frac{(1 - e^{-u_i(t)})}{(1 + e^{-u_i(t)})}$ (tang. hiperbólica)
 - Áreas cinza com tonalidades variadas próximas às fronteiras de decisão

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

28



Backpropagation

- Função de saída
 - Função de identidade $y_i = a_i$
- Treinamento
 - Supervisionado
 - $\Delta w_{ij} = \eta x_j y_j \delta_j$
 - $\Delta w_{ij} = \eta x_j y_j (1 - y_j) \delta_j$
 - $\delta_j = (d_j - y_j)$ (Última camada)
 - $\delta_j = \sum w_{jk} \delta_k$ (Camadas intermediárias)
 - Treinamento não é garantido de convergir

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

29



Treinamento

```

1) Inicializar todas as conexões com valores aleatórios
2) Repita
    erro = 0
    Para cada par de treinamento (X, d)
        Para cada camada k := 1 a N
            Para cada neurônio J := 1 a Mk
                Calcular a saída yjk
        Se erro > ε
            Então Para cada camada k := N a 1
                Para cada neurônio J := 1 a Mk
                    Atualizar pesos
    Até erro < ε
  
```

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

30



Teste

- 1) Apresentar padrão X a ser reconhecido
- 2) Para cada camada $k := 1$ a N
 Para cada neurônio $J := 1$ a M_k
 Calcular a saída y_{kj}
 Comparar saída y_{kj} com d_{cj} para cada classe c
 Classificar padrão como pertencente a classe
 cuja saída desejada é mais próxima da saída
 produzida

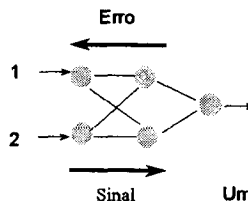
8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

31



Exemplo de rede

- Rede que aprende a função:



Ent. 1	Ent. 2	Saída
V	V	F
V	F	V
F	V	V
F	F	F

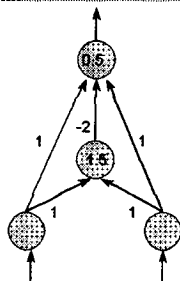
Um sinal verdadeiro é codificado para 1 e um falso para 0

8/12/98 Sistemas Inteligentes - André Ponce de Leon F. de Carvalho - LABIC/USP

32



Possível solução

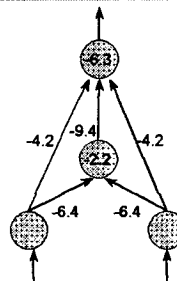


8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

33



Solução após treinamento

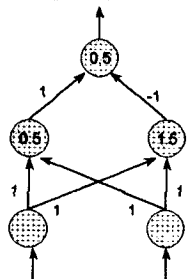


8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

34



Utilizando uma rede diferente



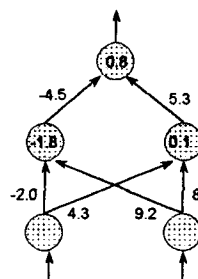
Rede sem conexões
entrada-saída

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

35



Problemas



Rede estável que
não funciona

Mínimo local

Ocorre em 1% das
vezes para problemas
do tipo ou-exclusivo

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

36



Visualização do comportamento da rede

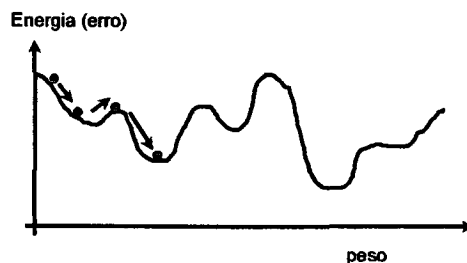
- Rede computa uma função de erro ou energia
 - $E_p = -\frac{1}{2} \sum (d_n - o_n)^2$
 - Quantidade que, para um dado padrão p , a saída produzida difere da saída desejada
 - Rede é garantida de encontrar solução se a superfície de energia for simples

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

37



Visualização do comportamento da rede



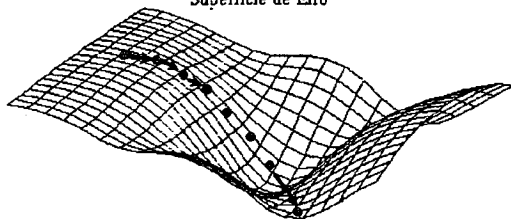
8/12/98 Sistemas Inteligentes - André Ponce de Leon F. de Carvalho LABIC/USP

38



Visualização do comportamento da rede

Superfície de Erro

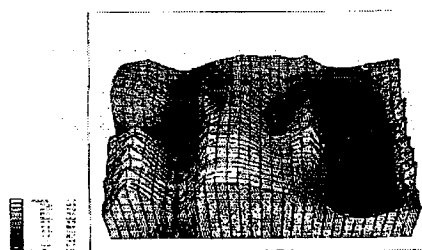


8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

39



Visualização do comportamento da rede



8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

40



MLPs como classificadores

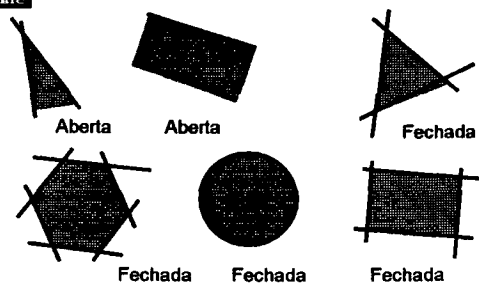
- Função implementada por cada neurônio é formada pela combinação das funções implementadas por neurônios da camada anterior
 - Camada 1: linhas retas no espaço de decisão
 - Camada 2: regiões convexas
 - Número de lados = número de unidades na camada 1
 - Camada 3: Combinações de figuras convexas, produzindo formatos abstratos

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

41



Regiões convexas



8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

42

Combinções de regiões convexas

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 43

Exercício

- Quantas camadas com quantos neurônios em cada camada possui a rede que divide o espaço de entradas da forma abaixo:

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 44

MLPs como classificadores

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 45

MLPs como classificadores

- Unidades intermediárias
 - Detetores de características
 - Geram uma codificação interna da entrada
 - Dado um número grande o suficiente de unidades intermediárias, é possível formar representações internas de qualquer padrão de entrada

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 46

Unidades intermediárias

- Número de camadas intermediárias necessárias
 - 1 camada: suficiente para aproximar qualquer função contínua ou Booleana
 - 2 camadas: suficiente para aproximar qualquer função
 - 3 ou mais camadas: pode facilitar o treinamento da rede
 - Cada vez que o erro é propagado para a camada anterior, ele se torna menos útil

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 47

Unidades intermediárias

- Número de neurônios nas camadas intermediárias
 - Em geral não é conhecido
 - Utilizar função do número de entradas e saídas
 - Não funciona
 - Número de pesos vezes dez é menor que o número de exemplos
 - Apenas reduz overfitting
 - Se o número de exemplos for muito maior que o número de pesos, overfitting é improvável, mas pode ocorrer underfitting

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP 48



Unidades intermediárias

● Número de neurônios nas camadas intermediárias (cont.)

- Depende de:
 - Número de exemplos de treinamento
 - Quantidade de ruído
 - Complexidade da função a ser aprendida
 - Distribuição estatística

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

49



Unidades intermediárias

● Número de neurônios nas camadas intermediárias (cont.)

- Existem problemas com uma entrada e uma saída que precisam de milhares de unidades e vice-versa
- Pode crescer exponencialmente com o número de entradas
- Solução neural eficiente: aquela onde o número de unidades cresce apenas polinomialmente com o número de entradas

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

50



Generalização

- Classificação correta de padrões não utilizados no treinamento ou com ruído
- Ocorre através da detecção de características relevantes do padrão de entrada
- Padrões desconhecidos são atribuídos a classes cujos padrões apresentam características semelhantes
- Tolerância a falhas

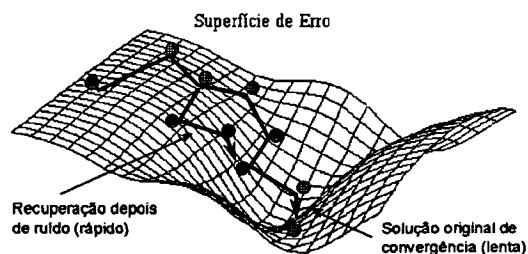
8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

51



Generalização



8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

52



Dificuldades de aprendizado

- Backpropagation é muito lento em superfícies complexas
 - Considerar efeitos de segunda ordem para gradiente descendente
- Mínimos locais: solução estável que não fornece saída correta
 - Taxa de aprendizado decrescente
 - Adicionar nós intermediários
 - Utilizar momentum
 - Adicionar ruído

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

53



Dificuldades de aprendizado

- Overfitting
 - Depois de um certo ponto do treinamento, a rede piora ao invés de melhorar
 - Memoriza padrões de treinamento, incluindo suas peculiaridades (piora generalização)
 - Alternativas
 - Encerrar treinamento cedo
 - Reduzir pesos

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

54



Atualização dos pesos

- Ciclo
 - Apresentação de todos os exemplos de treinamento durante o aprendizado
 - Exemplos devem ser apresentados em ordem aleatória
- Abordagens para atualização dos pesos
 - Por padrão (online)
 - Por ciclo (batch)

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

55



Atualização dos pesos

- Por padrão
 - Pesos atualizados após apresentação de cada padrão
 - Estável se taxa de aprendizado e momentum forem pequenos (reduzir progressivamente as taxas)
 - Altas taxas $\rightarrow$ rede instável
 - Mais rápida, principalmente se o conjunto de treinamento for grande e redundante
 - Requer menos memória

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

56



Atualização dos pesos

- Por ciclo
 - Pesos atualizados depois que todos os padrões de treinamento forem apresentados
 - Geralmente mais estável
 - Pode ser lento se o conjunto de treinamento for grande e redundante
 - Estimativa mais precisa do vetor gradiente
- Método depende da aplicação

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

57



Atualização dos pesos

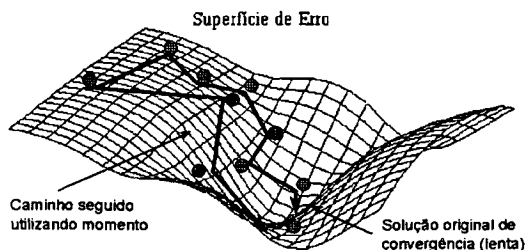
- Momentum
 - $\Delta w_{ij}(t+1) = \eta x_i y_j (1 - y_j) \delta_j + \alpha (w_{ij}(t) - w_{ij}(t-1))$
 - Aumenta velocidade de aprendizado evitando perigo de instabilidade
 - Pode acelerar treinamento em regiões muito planas da superfície de erro
 - Suprime oscilação de pesos em vales e ravinas

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

58



Momento



8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

59



Dicas para melhorias

- Projeto de uma RNA utilizando backpropagation é mais uma arte que uma ciência
 - Envolve inúmeros fatores
 - Resultado da experiência do projetista
- Utilizar função sigmoidal assimétrica (tangente hiperbólica)
 - Aprendizado mais rápido (em geral)
 - Igual a função logística com bias e re-escalada

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

60



Dicas para melhorias

- Resposta desejada deve estar $[-a + \varepsilon, a - \varepsilon]$
 - a = valor máximo da função de ativação
- Inicialização dos pesos e thresholds deve ser uniformemente distribuído dentro de um intervalo pequeno
 - Reduz probabilidade dos neurônios saturarem
 - Intervalos muito pequenos podem tornar treinamento lento
 - Geralmente utiliza-se $(-2.4/\text{fan_in}, + 2.4/\text{fan_in})$

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

61



Dicas para melhorias

- Taxa de aprendizado não deve, preferencialmente, ser a mesma para todos os neurônios
 - Geralmente, últimas camadas têm gradiente maior que camadas iniciais
 - Taxa de aprendizado deve ser menor para neurônios das últimas camadas
 - Neurônios com muitas entradas, devem ter taxas de aprendizado menores

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

62



Dicas para melhorias

- Utilizar, sempre que possível, modo padrão
 - Atualização *on-line*
 - Classificação de padrões envolvendo base de dados grande e redundante
 - Paralelização ineficiente
- Incluir conhecimento a priori
- Utilizar técnicas de pruning
 - Elimina conexões não essenciais
- Empregar cross-correlation

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

63



Aplicações

- Gerar sons a partir de textos
- Reconhecimento de padrões
- Venda de passagens aéreas
- Filtragem de ruído de eletrocardiograma
- Previsão de séries temporais

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

64



Conclusão

- Necessidade de redes multicamadas
 - Redes de uma camada não resolvem problemas não linearmente separáveis
 - Problema: treinamento de redes multicamadas
 - Backpropagation
 - Overfitting
 - Momento

8/12/98 Sistemas Inteligentes - André Ponce de Leon F. de Carvalho - LABICUSP

65



Redes Neurais do tipo RBF



Prof. Dr. André C. P. L. F. de Carvalho
LABIC - ICMSC - USP

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

1



Introdução

- Maioria das RNAs multicamadas computa uma função não-linear do produto escalar dos vetores de entrada e de pesos
- Existem RNAs multicamada cuja função de ativação de um neurônio da camada intermediária é função da distância entre os vetores de peso e de entrada

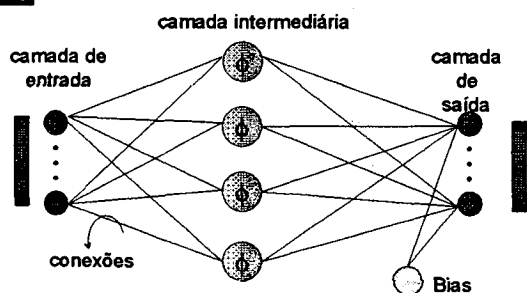
■ Redes RBF

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

2



Rede RBF



8/12/98 Sistemas Inteligentes - André Ponce de Leon F. de Carvalho - LABICAUSP

3



Redes RBF

- Evolução das redes MLP
- Redes de duas camadas
 - Primeira camada
 - Utiliza funções de ativação não lineares (funções de base radial)
 - Segunda camada
 - Utiliza funções de ativação lineares
 - Semelhante ao madaline

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

4



Redes RBF

● Redes RBF X MLP

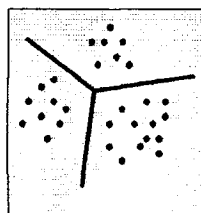
- MLP utiliza hiperplanos para particionar espaço de entradas (camada intermediária)
 - Definidos por funções da forma $f(\sum w_i x_i)$
- RBF utiliza hiperelipsóides para particionar o espaço de entradas (camada intermediária)
 - Definidos por funções da forma $\phi(\|x_i - \mu_i\|)$
 - Distância do vetor de entrada a um centro

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

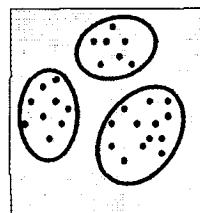
5



Fronteiras de decisão



MLP



RBF

8/12/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

6



Redes RBF

- Cada neurônio da camada intermediária computa uma função base radial
 - Centro
 - Protótipo para o cluster
 - Largura
 - Área de influência de cada cluster
- Podem ser bem mais rápidas que redes MLP

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

7



Redes RBF

- Entrada total
 - $u = \|x_i - \mu_i\|$ (camada intermediária)
 - $u = \sum w_i \phi_i(\|x_i - \mu_i\|)$ (camada de saída)
- Medida de distância
 - Geralmente utiliza distância Euclidiana
 - $\|x_i - \mu_i\| = \sqrt{\sum_{l=1}^N (x_{il} - y_{il})^2}$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

8



Redes RBF

- Processamento
 - Forward
- Estados de ativação
 - 1 (+1) = ativo
 - 0 (-1) = inativo

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

9



Redes RBF

- Função de ativação
 - Não linear
 - Valor aumento ou diminui com relação à distância a um ponto central
 - Funções típicas:
 - Função Gaussiana
- $$\phi(v) = \exp\left(-\frac{v^2}{2\sigma^2}\right)$$

$$v = \|x - \mu\|$$

$$x: \text{vetor de entrada}$$

$$\mu: \text{centro da função radial}$$

$$\sigma: \text{largura da função radial}$$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

10



Redes RBF

- Função de ativação
 - Funções típicas
 - Função Multiquadrática
- $$\phi(v) = \sqrt{v^2 + \sigma^2}$$
- Função Thin-Plate-Spline
- $$\phi(v) = v^2 \log(v)$$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

11



Redes RBF

- Escolha da função depende de:
 - Nível de conhecimento sobre os dados
 - Funções devem cobrir pontos uniformemente distribuídos do espaço de entradas
 - Conhecimento da estrutura geral das entradas
 - Levar a estrutura em consideração na escolha das funções

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

12



Redes RBF

- Função de saída
 - Função de identidade $y_i = a_i$
- Treinamento
 - Dois estágios
 - Primeiro estágio: camada intermediária
 - Segundo estágio: camada de saída



Primeiro estágio

- Determina parâmetros das funções de base radial
 - Número, centros e larguras das funções radiais
 - Não supervisionado
- Definição dos centros
 - Existem várias abordagens
 - Seleção aleatória dos centros
 - Clustering
 - K-means
 - Kohonen



Definição dos centros

- Seleção aleatória de centros (μ)
 - Seleciona centros aleatoriamente a partir dos padrões de treinamento
 - Assume que padrões estão distribuídos de uma maneira representativa para o problema
 - Padrões são capazes de capturar as principais características do problema
 - Problema: distribuição dos dados pode não ser representativa



Definição dos centros

- K-means clustering
 - Centros são colocados no meio de agrupamentos de vetores de entrada (clusters)
 - Utiliza aprendizado não-supervisionado
 - Divide os vetores de entrada em K conjuntos disjuntos S_j
 - Cada conjunto S_j tem N_j vetores
 - Objetivo: minimizar distâncias entre vetores de S_j e seu centro



Definição dos centros

- Heurísticas para definir larguras (raios) das funções radiais (σ)
 - Todas as largura iguais a média sobre todas as distâncias Euclidianas entre o centro de cada unidade N_i e o centro da unidade N_j mais próxima

$$\sigma = \frac{1}{m} \sum_{i=1}^m ||\mu_i - \mu_j|| \quad \mu_j \text{ é o centro mais próximo de } \mu_i$$



Definição das larguras

- Heurísticas para definir larguras (raios) das funções radiais (σ) - continuação
 - Atribuir a cada unidade uma largura diferente baseada na distância do seu centro ao centro da unidade mais próxima

$$\sigma = \alpha ||\mu_i - \mu_j|| \quad \begin{array}{l} \mu_j \text{ é o centro mais} \\ \text{próximo de } \mu_i \\ 1.0 < \alpha < 1.5 \end{array}$$



Definição das larguras

- Heurísticas para definir larguras (raios) das funções radiais (σ) - continuação
 - Atribuir a cada σ_j a distância média de seu centro aos N vetores de entrada mais próximos
 - Atribuir a σ_j um valor constante (geralmente 1)
- Função da primeira camada é transformar conjunto de dados não-linearmente separáveis em linearmente separáveis



Segundo estágio

- Determina pesos da camada de saída
 - Recebe vetores linearmente separáveis
 - Supervisionado
 - Define classe dos vetores de entrada
- Métodos para ajustar pesos
 - Decomposição em valores singulares
 - (M)adeline



Observações

- Sejam M o número de funções base e N o número de vetores de entrada:
 - $M \ll N$ ($M = N$ pode levar a overfitting)
 - Número de funções de base deve ser determinada pela complexidade dos dados
 - Número de funções base radial = número de clusters



Observações

- Redes RBF possuem a propriedade de *melhor aproximação*
 - No conjunto de funções de aproximação existe uma que tem o erro de aproximação mínimo para qualquer função a ser aproximada



Aplicações

- Reconhecimento de caracteres
- Reconhecimento de alvos
- Análise de crédito
- Processamento de sinais
- Processamento de voz



Conclusão

- Redes RBF
 - Alternativa a MLP
 - Funções de base radial
 - Função de ativação leva em conta distância entre vetores de entrada e protótipos
 - Aprendizado em dois estágio
 - Supervisionado
 - Não supervisionado



Redes Self-Organising

Prof. Dr. André C. P. L. F de Carvalho
LABIC - ICMSC - USP

19/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

1



Redes Self-organising

- Em várias aplicações é desejável que a rede faça sua própria classificação dos dados de treinamento
- Para isso é necessário que:
 - Padrões pertencentes a mesma classe possuam o mesmo valor para um número de características
 - Rede consiga identificar estas características
 - Critério para agrupar os dados

19/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

2



Redes Self-organising

- Existem problemas que:
 - Não utilizam resposta desejada
 - Não recebem punição/recompensa
 - Única informação fornecida está no conjunto de padrões de entrada

19/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

3



Redes Self-organising

- Propósito de um algoritmo self-organising
 - Descobrir padrões ou características significativas nos dados de entrada
 - Sem um professor
 - Algoritmo apresenta um conjunto de regras de natureza local

19/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

4



Redes Self-organising

- Princípios de self-organization
 - Modificações em pesos de conexões tendem a amplificar estes pesos
 - Limitações de recursos leva a competição entre sinapses e a seleção das mais aptas em prejuízo das outras
 - Modificações nos pesos tendem a cooperar

19/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

5



Redes Self-organising

- Redes self-organising
 - Definem os parâmetros da rede por si próprias, sem auxílio externo
 - Descobrem padrões significativos ou características nos dados de entrada
 - Rede forma sua própria classificação dos dados de treinamento
 - Padrões de uma mesma classe compartilham características em comum

19/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

6



Redes Self-organising

- **Aprendizado não supervisionado funciona apenas quando existe redundância na entrada**
 - Redundância fornece conhecimento
 - Ausência de redundância
 - Impossível encontrar padrões ou características nos dados
 - Dados seriam semelhantes a ruídos aleatórios
- **Aprendizado não supervisionado**
 - Aprendizado Hebbiano
 - Aprendizado competitivo

19/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

7



Redes Self-Organising

- **Competitivo**
 - Neurônios competem entre si pelo direito de atualizar seus pesos
 - Tarefa
 - Classificação
 - Extração de características (Compressão de dados)
 - Formação de clusters (agrupamentos)
 - Exemplos
 - ART, Kohonen

19/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

8



Redes Self-Organising

- **Hebbiano**
 - Utilizam procedimento baseado na regra de Hebb para atualizar os pesos
 - Tarefas
 - Extração de características
 - Análise de dados
 - Memória autoassociativa
 - Exemplos
 - Hopfield, PCA, Linsker

19/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

9



Redes Self-organising

- **Mais semelhante às estruturas neurobiológicas que redes supervisionadas**
- **Estrutura de sistemas self-organising**
 - Camada de entrada (fonte) e camada de saída (representação)
 - Conexões feedforward da entrada para a saída
 - Conexões laterais entre neurônios da camada de saída
 - Pode ter várias camadas, onde o processo de self-organização segue de camada em camada

19/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

10



Redes de Kohonen

- **Determinadas áreas do cérebro são responsáveis por funções específicas**
 - Fala
 - Visão
 - Controle de movimentos

19/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

11



Redes de Kohonen

- **Cada área pode conter sub-áreas**
 - Cada sub-área mapeia internamente respostas do órgão sensorial representado por ela
 - **Cortex auditivo:** de acordo com a resposta a diferentes frequências sonoras
 - **Cortex visual:** de acordo com características visuais primitivas
 - Intensidade de luz
 - Orientação e curvatura de linhas
- **Neurônios espacialmente ordenados**

19/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

12



Redes de Kohonen

- Utilizam algoritmo de aprendizado baseado em conceitos de auto-organização biologicamente plausíveis
- Baseadas no mapeamento realizado pelo cérebro Cérebro
 - Permite representação de dados n-dimensionais em um espaço m-dimensional ($m \ll n$)
 - Utiliza técnica de quantização de vetores para comprimir dados dos vetores de entrada

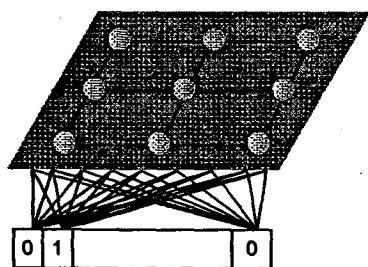


Características básicas

- Arquitetura
 - Uma camada bi-direcional
 - Grade plana (reticulado)
 - Cada neurônio
 - Recebe todas as entradas e gera saída
 - Está conectado aos seus vizinhos (feedback)
 - Funciona como classificador de características
 - Pode ser utilizada uma hierarquia de camadas



Rede de Kohonen



Características básicas

- Estados de ativação
 - $[0, N]$
- Função de ativação
 - $d_j = \sum (x_i - w_{ij})^2$
 - Baseada em distância Euclidiana
- Função de saída
 - Função identidade



Características básicas

- Treinamento
 - Não supervisionado
 - Organiza neurônios em vizinhanças locais
 - Neurônios competem entre si
 - Apenas neurônio vencedor e seus vizinhos atualizam seus pesos



Características básicas

- Treinamento
 - Atualização dos pesos
 - Atualiza neurônio vencedor e seus vizinhos dentro de um certo raio
 - Raio e taxa de aprendizado são decrementados durante treinamento
 - $w_{ij}(t+1) = \eta(t)(x_i(t) - w_{ij}(t))$ ($x_i \in$ vizinhança do vencedor)
 - Cria regiões que respondem a um grupo de entradas semelhantes



Algoritmo de Kohonen

1. Iniciar conexões com pequenos valores aleatórios;
2. Definir raio e taxa de aprendizado iniciais
3. Repita
 - Para cada padrão de treinamento x
 - Para cada neurônio n_j
 - Calcular a saída d_j ;
 - Selecionar neurônio n_k com menor d_k ;
 - Atualizar pesos de n_k e seus vizinhos;
 - Reduzir taxa de aprendizado
 - Reduzir raio
 - Até raio $<$ raio_mínimo



Características básicas

● Observações

- Encontra a unidade mais parecida com o padrão de entrada
- Aumenta sua semelhança e a de seus vizinhos com o padrão de entrada
- Forma mapa topográfico
 - Neurônios topologicamente próximos respondem de forma semelhante a entradas semelhantes



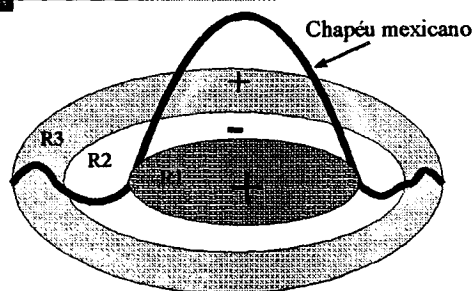
Características básicas

● Justificativa biológica

- Córtex cerebral
 - Neurônios ativos apresentam conexões mais fortes para neurônios fisicamente mais próximos
 - A partir de uma certa distância, conexões se tornam inibitórias (chapeu mexicano)
 - Parte da razão para mapeamento topológico no cérebro



Características básicas



Características básicas

- Rede de Kohonen modela córtex
 - Redes interconectadas localmente
 - Adaptação restrita aos neurônios vencedor e seus vizinhos
- Aspectos centrais do treinamento de Kohonen
 - Processo de adaptação dos pesos
 - Conceito de vizinhança dos nós



Adaptação dos pesos

● Processo em dois estágios

● Estágio 1

- Criação de uma ordenação topológica sobre mapa de nós aleatoriamente orientados

● Estágio 2

- Ajuste dos nós de cada sub-área (classe) para os padrões de entrada



Adaptação dos pesos

● Estágio 1

- Busca agrupar os nós do mapa topológico de modo a refletir as diferentes classes
- Rede descobre quantas classes deve identificar e suas posições relativas no mapa
 - mapeamento grosseiro
- Ocorrem grandes mudanças nos pesos
 - Taxa de aprendizado alta ($\eta(t) > 0.5$)



Adaptação dos pesos

● Estágio 2

- Taxa de aprendizado decrescente
- Requer 100 a 1000 apresentações a mais que no primeiro estágio
- Normalização dos pesos
 - Não considerar magnitude, apenas orientação do vetor peso
 - Reduz magnitude dos pesos para 1



Adaptação dos pesos

● Inicialização dos pesos

- Aleatória
 - Muitos nós com vetores de pesos muito diferentes dos padrões de entrada
 - Não haverá nós utilizáveis para separar classes adequadamente
 - Não convergência ou ciclos muito lentos



Adaptação dos pesos

● Inicialização dos pesos (continuação)

- Inicializar todos os pesos com mesmo valor
 - Tomar padrões de treinamento inicialmente semelhantes
 - Adicionar ruído aos vetores de entrada nos primeiros estágios de treinamento



Adaptação dos pesos

● Inicialização dos pesos (continuação)

- Utilizar um threshold para cada nó (consciência)
 - Nós regularmente selecionados têm seu threshold aumentado
 - Reduz sua chance de ser selecionado
 - Permite utilização de nós redundantes
- Reduzir vizinhança durante treinamento
 - Solução utilizada por Kohonen



Vizinhanças

- Define quantos nós em torno do nó vencedor terão seus pesos ajustados
- Tamanho modificado dinamicamente durante treinamento
 - Inicialmente grande (ex. todos os nós)
 - Reduzido progressivamente até limite pré-definido
 - Taxa de redução é função linear do número de ciclos

Treino de vizinhança localizada

The effect of changing the neighborhood is to change the shape of the receptive field. All the nodes within the receptive field are updated with the same weights. A more localized receptive field results in a more precise representation of the input pattern. The receptive field is defined by the neighborhood of nodes in the input space. The receptive field is defined by the neighborhood of nodes in the input space. The receptive field is defined by the neighborhood of nodes in the input space.

19/05/98 Redes Neurais - André Ponca de Leon F. de Carvalho - LABICAUSP 31

Alteração dos pesos

The effect of changing the weights is to change the shape of the receptive field. All the nodes within the receptive field are updated with the same weights. A more localized receptive field results in a more precise representation of the input pattern. The receptive field is defined by the neighborhood of nodes in the input space. The receptive field is defined by the neighborhood of nodes in the input space. The receptive field is defined by the neighborhood of nodes in the input space.

19/05/98 Redes Neurais - André Ponca de Leon F. de Carvalho - LABICAUSP 32

Vizinhanças

- Pode ter diferentes formatos
 - Quadrado
 - Hexágono
 - Círculo
- Treinamento de rede de Kohonen é afetado por:
 - Taxa de aprendizado
 - Taxa de redução de vizinhança
 - Formato da região de vizinhança

19/05/98 Redes Neurais - André Ponca de Leon F. de Carvalho - LABICAUSP 33

Vizinhanças

- Após treinamento, rede forma agrupamentos
 - Grupos devem ser rotulados para indicar classe que representam
 - Permite classificação de padrões desconhecidos

19/05/98 Redes Neurais - André Ponca de Leon F. de Carvalho - LABICAUSP 34

Redes de Kohonen

- Pode ser necessário incluir novos padrões em uma rede já treinada
 - Melhorar performance de certos agrupamentos
- Vetor de quantização de aprendizado (LVQ)
 - Técnica de aprendizado supervisionado
 - Ajusta mapa de características para melhorar sua performance em circunstâncias modificáveis

19/05/98 Redes Neurais - André Ponca de Leon F. de Carvalho - LABICAUSP 35

Algoritmo LVQ

1. Selecionar vetores de treinamento com classificação conhecida
2. Definir raio e taxa de aprendizado iniciais
3. Repita
 - Para cada padrão de treinamento com classificação conhecida
 - Para cada neurônio n_i
 - Calcular a saída d_i
 - Selecionar neurônio n_j com menor d_i
 - Atualizar pesos de n_j e seus vizinhos
 - Reduzir taxa de aprendizado

Até erro < erro_mínimo

19/05/98 Redes Neurais - André Ponca de Leon F. de Carvalho - LABICAUSP 36



Algoritmo LVQ

- Utiliza para cada entrada a saída desejada
- Compara saída produzida com saída desejada
- Não mexe no raio da vizinhança
- Atualização dos pesos para nó vencedor
 - $w_{ij}(t+1) = \eta(t)(x_i(t) - w_{ij}(t))$ (correta)
 - $w_{ij}(t+1) = -\eta(t)(x_i(t) - w_{ij}(t))$ (incorreta)

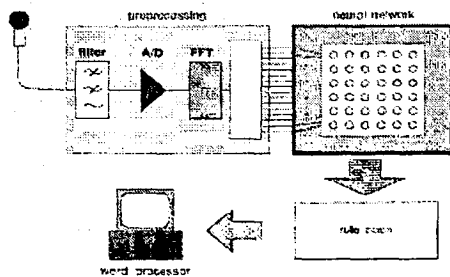


Aplicações

- Datilógrafo fonético
 - Conversão de fala em texto datilografado
 - Sistema híbrido
 - Rede utilizada para classificar fonemas
 - Mesmo fonema pode apresentar variações (orador, contexto onde palavra é utilizada)
 - Utiliza LVQ



Datilógrafo fonético



Aplicações

- Classificação de sinais de radar
- Controle de braços de robôs
- Segmentação de textura
- Modelamento do cérebro
- Tratamento de água



Conclusão

- Redes Self-organising
- Aprendizado não-supervisionado
 - Hebbiano
 - Competitivo
- Rede de Kohonen



Redes ART



Prof. Dr. André C. P. L. F de Carvalho
LABIC - ICMSC - USP

18/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

1



Redes ART

● Grossberg

- Contribuiu com várias teorias para o estudo de redes neurais, cobrindo principalmente
 - Psicologia humana
 - Neurobiologia
- Trabalho caracterizado por uma atenção rigorosa a precisão e detalhes matemáticos



Redes ART

● Grossberg (continuação)

- Objetivo a longo prazo: desenvolvimento de uma base teórica e metodologia matemática para unir as diversas sub-áreas englobadas pelo estudo de sistemas neurobiológicos



18/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

3



Redes ART

● Grossberg propôs modelos para

- Cognição
- Controle motor
- Visão
- Percepção
- Sistemas self-organising



Redes ART

● Uma das principais preocupações de Grossberg foi obter estabilidade em sistemas *self-organising*

- Sistemas complexos, como o cérebro, com arquitetura massivamente paralela e modular devem ter meios de manter estabilidade em todos os níveis
- Investigou dilema estabilidade-plasticidade

18/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

5



Redes ART

● Dilema estabilidade-plasticidade

- Aprender novas informações sem perder informações previamente aprendidas
- Maioria dos modelos de Redes Neurais não resolvem o dilema estabilidade-plasticidade
 - Treinamento de uma rede MLP com um novo padrão pode ter como efeito colateral a perda das informações aprendidas anteriormente
- Redes ART propõem uma solução para este dilema

18/05/98 Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

6



Redes ART

● Teoria da Ressonância Adaptativa

- Desenvolvida para modelar uma rede neural *self-organising* para reconhecimento de padrões, baseada em dados biológicos e comportamentais - ART
- Ressonância adaptativa ocorre quando padrões de atividade nas camadas de entrada e de saída se reforçam mutuamente



Redes ART

● Características de ART

- Arquitetura massivamente paralela
- Capacidade de aprender novas informações sem perder informações aprendidas previamente
- Sensível ao contexto: a rede distingue informações irrelevantes de informações apresentadas repetidamente



Redes ART

● Família ART

- Criada por Grossberg e Carpenter 1982
- Utiliza modelo de McCulloch-Pitts como neurônio
- Rede self-organising
 - Aprendizado competitivo
- Baseada na teoria da ressonância adaptativa



Redes ART

● Família ART

- ART-1: Entradas e saídas binárias
- ART-2: entradas e saídas binárias/análogicas
- ART-3: Inclui neurotransmissores
- Fuzzy ART: ART + lógica fuzzy
- ARTMAP: supervisionado
- Fuzzy-ARTMAP: combinação dos dois anteriores



Arquitetura da rede ART1

● Rede com duas camadas

- Camadas têm funções diferentes
 - Camada de saída/reconhecimento
 - Camada de entrada/comparação

● Mecanismos externos

- Sinais de controle lógico para cada camada
 - Direcionam o fluxo de informações dentro da rede
- Circuito de reset
 - Entre camada de entrada e camada de saída
 - Utiliza threshold de vigilância



Arquitetura da rede ART1

● Conexões

- Feedforward
- Laterais
- Feedback

● Conexões feedforward

- Dos neurônios da camada de entrada para os neurônios da camada de saída
- Vetor real de pesos W
 - Modificados durante treinamento



Arquitetura da rede ART1

● Conexões laterais

- De cada neurônio da camada de saída para todos os neurônios da mesma camada
- Vetor bipolar
 - Fixo
 - Excita neurônio de origem (reforço positivo)
 - Inibe demais neurônios (inibição lateral)



Arquitetura da rede ART1

● Conexões feedback

- Dos neurônios da camada de saída para os neurônios da camada de entrada
- Vetor binário de pesos T
 - Modificados durante treinamento



Arquitetura da rede ART1

- Cada nó da camada de entrada tem três entradas
 - Componente e_i do vetor de entrada
 - Sinal de feedback da camada de saída
 - Sinal de controle C_1
- Fluxo de dados na camada de entrada é controlado pela regra dos "dois-terços"

$$x_i = \begin{cases} 1 & \text{Se duas das três entradas estão ativas} \\ 0 & \text{Caso contrário} \end{cases}$$



Sinais de controle

● Controle 1 (C_1)

- Determina fluxo de dados para a camada de entrada

$$C_1 = \begin{cases} 1 & \text{se uma entrada válida for apresentada a rede} \\ 0 & \text{se qualquer nó na camada de reconhecimento está ativo} \end{cases}$$



Sinais de controle

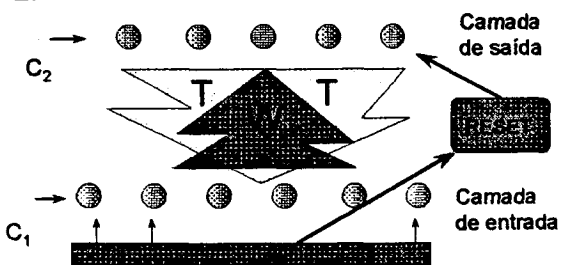
● Controle 2 (C_2)

- Determina fluxo de dados para a camada de saída
- Habilita ou desabilita nós na camada de reconhecimento

$$C_2 = \begin{cases} 1 & \text{se uma entrada válida for apresentada a rede} \\ 0 & \text{após falha de um teste de vigilância (desabilita os nós da camada de reconhecimento e reseta seus estados de ativação para 0)} \end{cases}$$

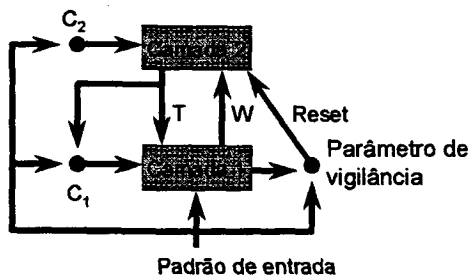


Rede ART1





Rede ART1



Rede ART1

Estado de ativação

- 1 = ativo
- 0 = inativo

Função de ativação

$$a_i(t+1) = f(u_i(t))$$

$$a_i(t+1) = \begin{cases} 1, & \text{se } \forall j, u_i(t) > u_j(t) \\ 0, & \text{caso contrário} \end{cases} \quad u = \sum_{j=1}^N x_j w_j$$



Rede ART1

Treinamento

- Não supervisionado
- Competitivo
- Apenas neurônio vencedor atualiza seus pesos
 - $T_i(t+1) = Z_i = T_i(t) X_i$
 - $w_{ij}(t+1) = LZ_i / (L - 1 + \sum Z_i)$
 - $w_{ij}(t+1) = T_i(t) X_i / (0.5 + \sum T_i(t) X_i)$



Operação da rede ART1

Existem várias fases para aprendizado e classificação

- Inicialização
- Treinamento
 - Fase de reconhecimento
 - Fase de comparação
 - Fase de busca



Operação da rede ART1

Processo de treinamento:

- Vetor de entrada continuamente modificado passa para frente para trás (ressoa) entre as camadas em um processo cíclico

Inicialização

- ART possui mais parâmetros para inicializar que outras redes



Fase de inicialização

Vetor de pesos W

$$w_i = \frac{1}{1+n} \quad (n = \text{número de nós de entrada})$$

Vetor de pesos T

$$t_i = 1 \quad (\text{todo nó de saída é conectado a todo nó de entrada})$$

Threshold de vigilância

$$0 < \rho < 1$$



Fase de reconhecimento

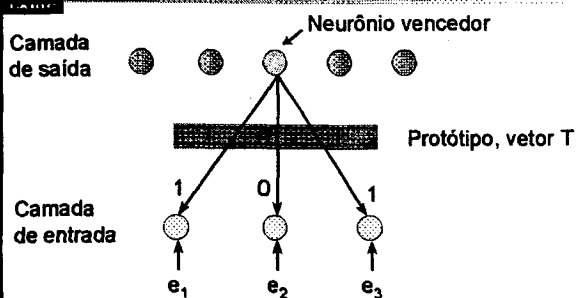
- Cada neurônio da camada de saída representa uma classe
 - Vetor de pesos W é o centro do cluster para a classe
- Vetor de entrada é comparado com cada classe
 - Classe com maior resposta (peso mais semelhante ao vetor de entrada) é selecionado
 - Inibição lateral faz com que apenas o nó com maior resposta fique ativo
 - Nó vencedor da competição passa protótipo (vetor T) de sua classe para a camada de comparação

18/05/98 Redes Neurais - André Ponca de Leon F. de Carvalho - LABICAUSP

25



Fase de reconhecimento



18/05/98 Redes Neurais - André Ponca de Leon F. de Carvalho - LABICAUSP

26



Fase de comparação

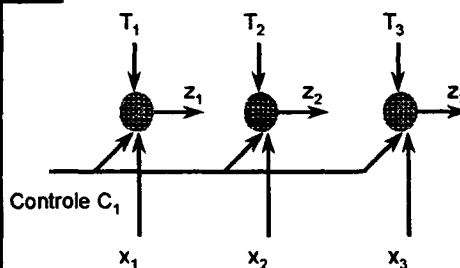
- Dois vetores são apresentados à camada de entrada e passados ao mecanismo de reset
 - Vetor de entrada (e)
 - Vetor de comparação (Z)
 - Cada nó tem três entradas
 - Vetor de entrada
 - Vetor de feedback
 - Sinal de controle C_1 (igual a zero na fase de comparação)

18/05/98 Redes Neurais - André Ponca de Leon F. de Carvalho - LABICAUSP

27



Fase de comparação



18/05/98 Redes Neurais - André Ponca de Leon F. de Carvalho - LABICAUSP

28



Fase de comparação

- Circuito reset
 - Testa semelhança entre o vetor de entrada e o vetor de comparação utilizando o threshold de vigilância

$$S = \frac{\sum_i x_i}{\sum_i x_i}$$

S

$\left\{ \begin{array}{l} > \rho \Rightarrow \text{Classificação terminou e a classe é indicada pelo nó ativo na camada de saída} \\ \leq \rho \Rightarrow \text{Padrão não casa com o protótipo do nó vencedor. Re-entrar na fase de busca} \end{array} \right.$

18/05/98 Redes Neurais - André Ponca de Leon F. de Carvalho - LABICAUSP

29



Fase de busca

- Objetivo
 - Encontrar nova classe na camada de saída para o padrão de entrada atual
- Passos
 - Nó de saída atualmente ativo é desabilitado (saída igual a zero)
 - Nó não entra mais na competição
 - C_2 é setado para zero
 - Vetor de entrada é re-aplicado a rede

18/05/98 Redes Neurais - André Ponca de Leon F. de Carvalho - LABICAUSP

30



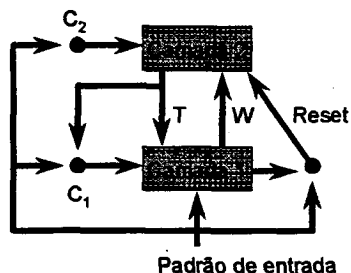
Fase de busca

● Passos (continuação)

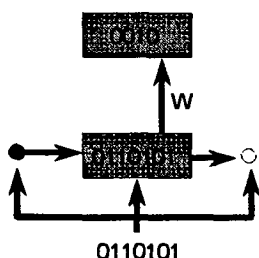
- Novo neurônio vence a competição
 - Entra na fase de comparação
- Processo se repete desabilitando nós até encontrar nó que case com a entrada dentro dos limites do parâmetro de vigilância
- Se tal nó não for encontrado, a rede declara o vetor de entrada desconhecido e aloca um novo nó de saída (nova classe) para ele



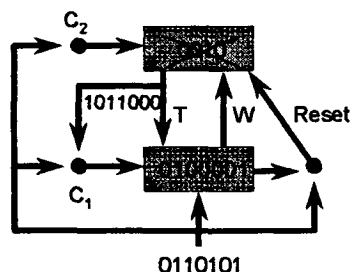
Exemplo



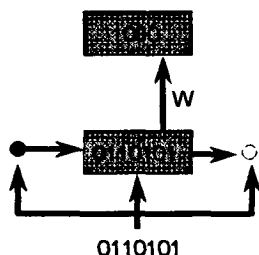
Exemplo



Exemplo



Exemplo



Algoritmo de treinamento

- 1 Inicializar todas as conexões W e T;
- 2 Repita
 - 2.1 Para cada padrão de treinamento (X)
 - 2.2 Definir neurônio vencedor
 - 2.3 Comparar protótipo do vencedor com a entrada
 - 2.4 Se comparação > valor de vigilância
 - Então Atualizar pesos W e T do neurônio vencedor
 - Senão Desabilitar neurônio
 - Se ainda existir neurônio não analisado
 - Então Voltar a 2.2
 - Senão Alocar novo neurônio à camada de saída
- Até conjunto de protótipos não mudar



Treinamento

- Rede pode re-entrar no modo de treinamento a qualquer tempo
 - Permite incorporar novos dados
 - Rede pode aprender em ambiente dinâmico
 - Resolve dilema estabilidade-plasticidade
- Existem dois esquemas de treinamento
 - Treinamento rápido
 - Treinamento lento



Treinamento

- Treinamento rápido
 - Pesos nas conexões feedforward definem seus valores ótimos em poucos ciclos
 - Maioria das vezes em um único ciclo
 - Mais simples de implementar
 - Menos preciso
 - Método mais adotado



Treinamento

- Treinamento lento
 - Pesos são adaptados lentamente, utilizando vários ciclos
 - Pesos são treinados para representar a média estatística dos dados de entrada para cada classe
 - Mais eficiente para encontrar características salientes dos padrões de entrada que determinam as classificações



Treinamento

- ART é muito sensível a variações nos seus parâmetros durante treinamento
 - Parâmetro mais crítico é o parâmetro de vigilância
- Parâmetro de vigilância
 - Controla resolução do processo de classificação



Treinamento

- Parâmetro de vigilância
 - Valor baixo (< 0.4)
 - Resolução baixa $\Rightarrow$ cria poucas classes iniciais
 - Valor alto
 - Resolução alta $\Rightarrow$ pequenas variações no valor de entrada levarão a criação de novas classes
 - Valor depende do contexto
 - Distinguir entre rosto masculino e feminino
 - Reconhecimento de espécies de passáros



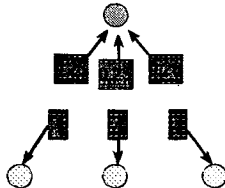
Treinamento

- Parâmetro de vigilância (continuação)
 - Solução ótima: variar parâmetro de vigilância durante o treinamento
 - Valor inicial baixo define classes
 - Valor seguinte mais alto otimiza classificação
 - Semelhante a técnica utilizada para definição de raio de vizinhanças para redes de Kohonen
 - Parâmetro pode ser modificado para punir rede por classificações erradas



Exemplo

- Treinar uma rede ART para agrupar os padrões 10011, 00110 e 11000



Aplicações

- Reconhecimento de padrões
 - Reconhecimento de caracteres
 - Reconhecimento automático de alvos
- Agrupamento de padrões
- Controle de braço de robô
- Análise de dados de questionário
- Controle de processos químicos



Conclusão

- Redes ART
 - Família de redes
 - Teoria da ressonância adaptativa
- ART1
 - Aprendizado competitivo
 - Rede construtiva
 - Parâmetro de vigilância
- Aplicações



Modelo de Hopfield



Prof. Dr. André C. P. L. F de Carvalho
LABIC - ICMSC - USP



Redes de Hopfield

- Associações fazem parte de nosso dia-a-dia
 - Ex. Verde $\Rightarrow$ árvore
 - Informações sensoriais (estímulos) podem produzir várias associações em cascata
 - Inf A $\Rightarrow$ inf B $\Rightarrow$... $\Rightarrow$ inf $\Rightarrow$ X
 - Ex. Verde $\Rightarrow$ árvore $\Rightarrow$ tombo $\Rightarrow$ dor



Redes de Hopfield

- Associações são realizadas com certa facilidade, por seres humanos
 - Estímulos incompletos
 - Estímulos parcialmente incorretos
 - Estímulos com ruídos



Memória endereçável pelo conteúdo



Redes de Hopfield

- Associações realizadas por computadores convencionais são bastante limitadas
 - Banco de dados
 - Estímulo = chave
 - Entrada deve ser específica e completa
 - Ex. Catálogo de biblioteca



Redes de Hopfield

- Desenvolvida por Hopfield em 1982
- Redes recorrentes que podem ser utilizadas como memórias associativas não lineares
 - Memórias endereçáveis pelo conteúdo
- Função primária:
 - Recuperar um padrão armazenado na memória em resposta a apresentação de uma versão incompleta ou com ruído daquele padrão

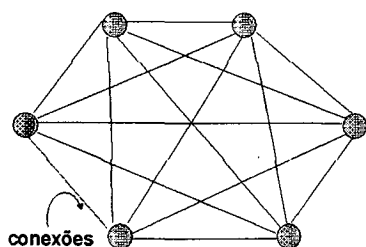


Contribuições de Hopfield

- Contribuiu para o ressurgimento do interesse na área
- Desenvolveu o uso de uma função de energia
- Estabeleceu o isomorfismo entre redes recorrentes e modelos utilizados na física
 - Incentivou a participação de físicos
 - Impulsionou a área de Redes Neurais



Rede de Hopfield



Características básicas

- Arquitetura
 - Recorrente
 - Completamente conectada
 - Simétrica: $w_{ij} = w_{ji}$ ($w_{ii} = 0$)
 - Não existem nós de entrada ou de saída
 - Todos os nós têm o mesmo papel
- Estado de ativação
 - Binário (0, 1)
 - Bipolar (-1, +1)



Características básicas

- Saída = estado de ativação
- Função de ativação
 - $a_i(t+1) = \begin{cases} +1, & \text{se } v_i(t) > 0 \\ -1, & \text{se } v_i(t) < 0 \\ a_i(t), & \text{se } v_i(t) = 0 \end{cases}$
 - OU
 - $a_i(t+1) = 1/(1 + e^{-kv_i(t)})$



Características básicas

- Características
 - Vetor de entrada aplicado a todos os nós de uma vez
 - $O(t) = X(t+1)$
- Operações
 - Fase de armazenamento
 - Regra do produto externo
 - Generalização da regra de Hebb
 - Fase de recuperação
 - Gradiente descendente

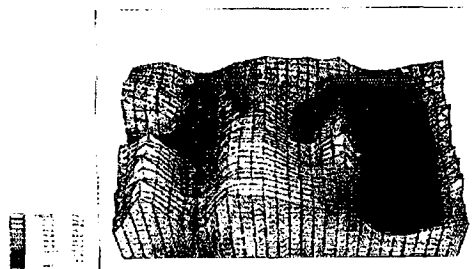


Superfícies de energia

- Comportamento do modelo de Hopfield pode ser explicado através de superfícies de energia
- Superfícies de energia
 - Fase de armazenamento forma vales na superfície
 - Vales representam padrões armazenados na rede
 - Padrão desconhecido é um ponto qualquer da superfície
 - Quando a rede itera em direção a uma solução, o ponto move-se em direção a um vale
 - Solução: ponto estaciona em região de menor energia



Superfícies de energia





Superfícies de energia

● Função de energia

- Define energia em um dado ponto da superfície
- Valor proporcional ao erro
- Depende dos pesos e da entrada atual
- $E = -\frac{1}{2} \sum_i \sum_j w_{ij} x_i x_j + \sum_i x_i \theta_i$
- Geralmente $\theta_i = 0$

● Gradiente descendente pode ser utilizado para atingir os locais mais baixos (vales)

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICA/USP

13



Armazenamento padrões

● Para armazenar um padrão P, é necessário minimizar o valor da função de energia para P

- Encontrar expressão para definir os pesos que levam a um valor mínimo para a função de energia

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} x_i x_j$$

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij}^* x_i x_j - \frac{1}{2} \sum_i \sum_j w_{ij}^p x_i x_j$$

$$E = E_{\text{todos menos P}} + E_P$$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICA/USP

14



Armazenamento de padrões

● Minimização do valor da função de energia para P

- É necessário:

$$E_P = -\frac{1}{2} \sum_i \sum_j w_{ij}^p x_i x_j \quad \text{minimizar}$$

$$\sum_i \sum_j w_{ij}^p x_i x_j \quad \text{maximizar}$$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICA/USP

15



Armazenamento de padrões

● Minimização do valor da função de energia para P

- Sabendo que:

$$x_i = +1 \text{ ou } x_i = -1 \Rightarrow x_i^2 > 0$$

- Se o termo a ser maximizado depender de x_i^2 , ele será positivo

$$\sum_i \sum_j w_{ij}^p x_i x_j = \sum_i \sum_j x_i^2 x_j^2 w_{ij}^p = x_i x_j$$

$$w_{ij} = \sum_p w_{ij}^p = \sum_p x_i^p x_j^p$$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICA/USP

16



Recuperação de padrões

- Padrões são recuperados calculando o gradiente descendente a partir do padrão de entrada
- Os padrões de treinamento estão armazenados em locais de mínima energia
- Mostrar que a energia é reduzida durante a recuperação

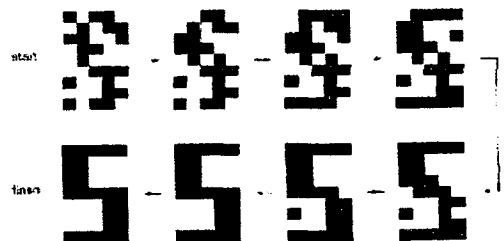
8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICA/USP

17



Recuperação de padrões



8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICA/USP

18



Recuperando padrões

Seja o_k' o novo estado do neurônio k

Se ($o_k' = o_k$)

Então: a energia não muda

Senão: $o_k' = -o_k$

$$E' - E = -\frac{1}{2} \sum_j \sum_{j \neq k} w_{kj} o_j' o_j + \frac{1}{2} \sum_j \sum_{j \neq k} w_{kj} o_j o_j$$

$$E' - E = -\frac{1}{2} o_k' \sum_j w_{kj} o_j + \frac{1}{2} o_k \sum_j w_{kj} o_j$$

$$E' - E = o_k \sum_j w_{kj} o_j = o_k o_k' = -o_k^2$$



Atualização dos estados

● Atualização síncrona

- Todos os neurônios são atualizados simultaneamente
- Duplica memória
- Determinística
- Mais apropriada para simulação



Atualização dos estados

● Atualização assíncrona

- Neurônios a serem atualizados são definidos randomicamente a cada instante de tempo t
 - Mudança no estado de um neurônio afeta estado do sistema e pode afetar mudança do próximo neurônio
 - Ordem de atualização dos nós afeta comportamento da rede
- Probabilística
- Mais apropriada implementação em hardware



Exemplo

Construir uma rede de Hopfield para armazenar os padrões -1-111, 1-1-11, 11-1-1 e 1-11-1 e definir a resposta da rede para o padrão 111-1.

a) Definir a rede:

Número de neurônios: 4

$$\begin{aligned} w_{01} &= (-1) \cdot (-1) + 1 \cdot (-1) + 1 \cdot 1 + 1 \cdot (-1) = 0 \\ w_{02} &= (-1) \cdot 1 + 1 \cdot (-1) + 1 \cdot (-1) + 1 \cdot 1 = -2 \\ w_{03} &= (-1) \cdot 1 + 1 \cdot (-1) + 1 \cdot (-1) + 1 \cdot 1 = -2 \\ w_{12} &= (-1) \cdot 1 + (-1) \cdot (-1) + 1 \cdot (-1) + (-1) \cdot 1 = -2 \\ w_{13} &= (-1) \cdot 1 + (-1) \cdot 1 + 1 \cdot (-1) + (-1) \cdot (-1) = -2 \\ w_{23} &= 1 \cdot 1 + (-1) \cdot 1 + (-1) \cdot (-1) + 1 \cdot (-1) = 0 \end{aligned}$$



Exemplo

b) Definir a resposta da rede para o padrão 111-1

$$\begin{aligned} o_0(1) &= [1 \cdot 0 + 1 \cdot (-2) + (-1) \cdot (-2)] = f[0] = 1 \\ o_1(1) &= [1 \cdot 0 + 1 \cdot (-2) + (-1) \cdot (-2)] = f[0] = 1 \\ o_2(1) &= [1 \cdot (-2) + 1 \cdot (-2) + (-1) \cdot 0] = f[-4] = -1 \\ o_3(1) &= [1 \cdot (-2) + 1 \cdot (-2) + 1 \cdot 0] = f[-4] = -1 \end{aligned}$$

$$\begin{aligned} o_0(2) &= [1 \cdot 0 + (-1) \cdot (-2) + (-1) \cdot (-2)] = f[4] = 1 \\ o_1(2) &= [1 \cdot 0 + (-1) \cdot (-2) + (-1) \cdot (-2)] = f[4] = 1 \\ o_2(2) &= [1 \cdot (-2) + 1 \cdot (-2) + (-1) \cdot 0] = f[-4] = -1 \\ o_3(2) &= [1 \cdot (-2) + 1 \cdot (-2) + (-1) \cdot 0] = f[-4] = -1 \end{aligned}$$



Observações

- Padrões parecidos provocam interferências
- Limite do número de padrões que podem ser armazenados sem interferências
 - Hopfield sugeriu que, para padrões aleatórios, o limite teórico era $P = 0.15N$ (1982)
 - Amit e outros mostraram que o limite era, na verdade, $P = 0.138N$ (1985)
 - McEliece e outros provaram que $P < N/4 \ln N$



Observações

- Modificações propostas para melhorar capacidade
 - Capacidade de memória depende de como os pesos das conexões são especificados
 - Uso da matriz pseudo-inversa de Kohonen aumenta capacidade para $P = 0.5N$ (1973)
 - Algoritmo spectral de Pancha torna $P = N$ (1993)
 - Gardner mostrou que capacidade de memória é função do tamanho da base de atração
 - Quando tamanho da base tende a 0, capacidade máxima tende a $P = 2N$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

25



Observações

- Ocorrência de mínimos locais
- Redes de Hopfield não são modelos realísticos do sistema biológico
- Principal limitação: falta de neurônios escondidos
 - Aprendem representações internas dos padrões de treinamento
 - Melhoram performance da rede

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

26



Máquinas de Boltzmann

- Equivalência formal ao trabalho de Boltzmann em termodinâmica estatística
- Utiliza função de probabilidade para atualizar estado dos neurônios da rede
 - Permite escapar de mínimos locais
 - Simulated annealing
 - Temperatura decrescente
 - Equilíbrio térmico

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

27



Máquinas de Boltzmann

- Arquitetura semelhante a rede de Hopfield
- Principais diferenças entre Maq. de Boltzmann e redes de Hopfield
 - Neurônios visíveis e intermediários
 - Função de ativação estocástica
 - Podem ser treinadas de forma supervisionada

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

28



Máquinas de Boltzmann

- Soluções equivalem aos mínimos das funções de energia
- Máquina de Boltzmann pode ser utilizada como memória endereçável pelo conteúdo
 - Minimiza função de energia para padrões armazenados

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

29



Algoritmo de treinamento

- 1 Definir pesos e thresholds uniformemente distribuídos no intervalo $[-a, a]$
- 2 Para cada par de treinamento (X, d) realizar simulated annealing para uma sequência decrescente de temperaturas $T_0, T_1, \dots, T_{\text{final}}$
 - 2.1 Associar elementos de X a neurônios de entrada e elementos de d a neurônios de saída
 - 2.2 Deixar a rede atingir equilíbrio térmico
 - 2.3 Incrementar peso entre cada dois neurônios ativos
 - 2.4 Associar cada elemento do vetor de entrada a um neurônio de entrada
 - 2.5 Deixar a rede atingir equilíbrio térmico
 - 2.6 Decrementar peso entre cada dois neurônios ativos

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICAUSP

30



Máquina de Boltzman

- Rede é garantida de achar mínimo global da função de energia
 - Esquema de congelamento do processo de aprendizado é suficientemente lento
 - Resfriamento é lento demais, não tendo valor prático



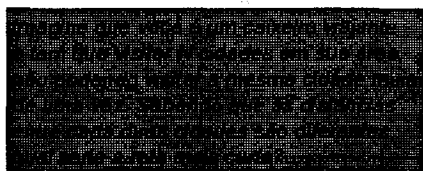
Satisfação de restrições

- Empecilhos a serem enfrentadas para solução de um problema
 - Expressar restrições a serem satisfeitas em termos de uma função de energia apropriada
 - Satisfazer restrições = encontrar solução que minimiza energia
 - Ex. Minimizar custo de transporte de mercadorias
 - Custo é proporcional a distância
 - Função de energia grande quando distâncias são grandes e vice-versa
 - Minimização da função = minimização dos custos

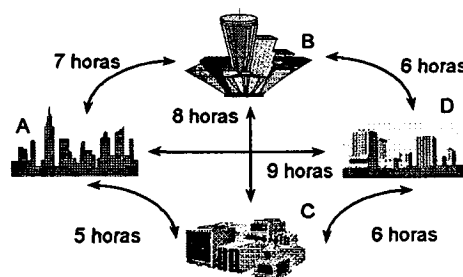


Problema do caixeiro viajante

- Estudado por diferentes pesquisadores utilizando as mais diversas técnicas
- XOR para problemas de satisfação de restrições
- Problema:



Problema do caixeiro viajante



Problema do caixeiro viajante

- É muito difícil encontrar a melhor solução
 - Tempo para resolver o problema cresce exponencialmente com o número de cidades
 - Solução boa é suficiente
- Restrições do problema
 - Cada cidade deve ser visitada uma única vez
 - Distância viajada deve ser a mais curta possível



Problema do caixeiro viajante

- Solução do problema
 - Minimizar função de energia = produzir solução que otimiza restrições
 - Construir representação do problema para a rede
 - Solução: lista de cidades a ser visitada em certa ordem
 - Especificar cidade e posição onde ela é visitada



Problema do caixeiro viajante

● Solução do problema (continuação)

- Se existem N cidades cada uma ocorre em uma das N posições
 - N x N Neurons
 - Possível solução para 4 cidades

Cidade / posição	1	2	3	4
A	1	0	0	0
B	0	0	1	0
C	0	0	0	1
D	0	1	0	0

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

37



Problema do caixeiro viajante

● Solução do problema (continuação)

- Construir função de energia onde mínimo = boa solução
- Produzir caminhos curtos mais válidos

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

38



Problema do caixeiro viajante

$$E = \sum \sum A_{xi} V_{xj} + \sum \sum B_{yj} V_{xi} + C(\sum V_{xi} - n)^2 + 1/2 D \sum \sum d_{xy} V_{xi} (V_{y,j+1} V_{y,j-1})$$

- V_{xi} : saídas dos neurônios
- X, Y : cidades
- i, j : posição que aparece nas visitas
- d_{xy} : distância entre cidades x e y (peso)
- n : número de elementos da matriz
- A, B, C: constantes

Termo 1 = 0 se cada linha de cidade x contém apenas um valor igual a '1'
 Termo 2 = 0 se cada coluna de posição contém apenas um valor igual a '1'
 Termo 3 = 0 se existem n entradas na matriz com valor igual a '1'
 Termo 4 = distância percorrida nas visitas

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

39



Problema do caixeiro viajante

● Solução do problema (continuação)

- Energia mede a distância percorrida em uma sequência válida de visitas
- Dificuldade: encontrar conj. adequado de constantes

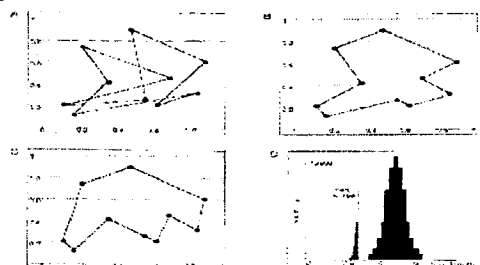
8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

40



Problema do caixeiro viajante



8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

41



Aplicações

- Problemas de otimização
 - Satisfação de restrições
 - Problema do caixeiro viajante
 - Modelagem de fenômenos físicos
- Memoração endereçável pelo conteúdo
 - Armazenamento de imagens

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

42



Conclusão

- Modelos de memória associativa

- Hopfield

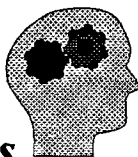
- Paralelo com mecânica estatística
 - Rede completamente conectada
 - Não existem, explicitamente, neurônios de entrada e saída
 - Não existem neurônios escondidos

- Máquina de Boltzmann

- Neurônios de entrada, escondidos e de saída



Algoritmos Genéticos



Prof. Dr. André C. P. L. F de Carvalho
LABIC - ICMSC - USP



Principais Tópicos

- Introdução
- Conceitos básicos
- Codificação
- Função de aptidão
- Operadores Genéticos
- Convergência



Introdução

- Algoritmos genéticos (AGs) são métodos adaptativos que podem ser utilizadas para resolver problemas de busca e otimização
 - São baseados nos processos genéticos de organismos biológicos
 - Depois de várias gerações populações naturais evoluem de acordo com os princípios de seleção natural e sobrevivência dos mais aptos (*Charles Darwin, A Origem das Espécies*)



Introdução

- Algoritmos Genéticos são capazes de “evoluir” soluções para problemas do mundo real
 - Problemas devem ser adequadamente codificados
 - Deve haver uma forma de avaliar as soluções apresentadas



Introdução

- Desenvolvido por John Holland e sua equipe (popularizado por David Goldberg)
- Objetivos:
 - Abstrair e explicar rigorosamente os processos adaptativos dos sistemas naturais
 - Desenvolver sistemas artificiais que conservam mecanismos importantes dos sistemas naturais



Evolução natural

- Na natureza
 - Indivíduos e populações competem entre si por recursos
 - Alimento
 - Água
 - Abrigo
 - Membros da mesma espécie frequentemente competem para atrair um par



Evolução natural

● Natureza (continuação)

- Indivíduos mais bem sucedidos na sobrevivência e atração de um par terão, relativamente, mais descendentes
- Indivíduos mal sucedidos geram poucos ou nenhum descendente
- Genes dos indivíduos mais adaptados se espalham para um número cada vez maior de indivíduos, geração a geração



Evolução natural

● Natureza (continuação)

- Combinação de boas características de diferentes ancestrais podem produzir indivíduos ainda mais aptos
- Espécies evoluem para se tomarem cada vez mais adaptadas aos seus ambiente
- Genética é a ciência da hereditariedade
 - Estuda mecanismo de transmissão de caracteres de uma espécie, de uma geração para outra



Evolução natural

● Provavelmente todos os seres vivos vieram de um mesmo ancestral

- Forte evidência vem dos caracteres compartilhados por espécies que, a primeira vista, parecem não ter nenhuma relação
 - Ex. Asa de pássaros X antebraço dos seres humanos



Evolução natural

● Asa X antebraço

- Morfologia geral
 - Virtualmente idêntica relação de ossos e músculos
 - Ancestrais comuns compartilhavam estes elementos
- Morfologia específica
 - com penas X sem penas
 - sem dedos X com dedos
- Função
 - vôo X manipulação de objetos



Fatos históricos

● 1809 Lamarck publica *Philosophie Zoologique*

- Seres vivos modificam-se através do tempo
 - Lei do uso e desuso
 - Lei da transmissão das características adquiridas

● 1859 Charles Darwin publica o livro *The origin of Species*

- Mecanismos de transformação das espécies



Fatos históricos

● 1865 Mendel apresenta experimentos em cruzamento genético de ervilhas

- Pai da genética

● 1976 Holland definiu os princípios básicos de Algoritmos Genéticos



Algoritmos genéticos

- AGs se baseiam na evolução natural
 - Trabalham com uma população de indivíduos, cada um representando uma possível solução para um dado problema
 - A cada indivíduo é associado um escore de aptidão, que mede o quão boa é a solução
 - Indivíduos mais aptos têm mais oportunidades de serem reproduzidos
 - Produzindo descendentes cada vez mais aptos

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

13



Algoritmos genéticos

- São algoritmos de busca e otimização baseados em mecanismos de evolução natural e genética
- Combinam
 - *Sobrevivência dos mais aptos entre estruturas de string*
 - Troca de informação estruturada, ainda que randômica

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

14



Algoritmos genéticos

- Busca acontece em várias gerações
 - A cada geração, um novo conjunto de criaturas artificiais (strings) é criado utilizando partes das criaturas mais aptas da geração anterior
 - Ocasionalmente, uma parte nova pode ser criada
 - Exploram eficientemente informação passada para especular sobre novos pontos de busca com esperada melhoria na performance

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

15



Princípios básicos

- Codificação
- Função de aptidão
- Reprodução
- Convergência

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

16



Conceitos básicos

- Indivíduo
 - Possível solução para um dado problema
 - Também chamado de cromossomo, string ou solução
 - Conjunto de indivíduos forma uma população
 - Codificado na forma de um vetor de características

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

17



Exemplo: preparo de biscoitos

- Otimizar quantidades de açúcar e farinha de trigo para preparar biscoitos
- Passos
 - Criar população inicial
 - *Codificar strings ou cromossomos*
 - Definir função de aptidão
 - Reprodução

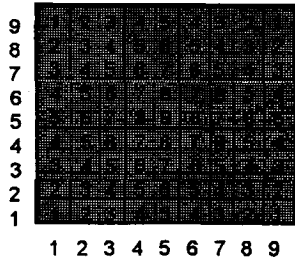
8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

18



Mapa de qualidade do preparo



Codificação

- Solução potencial para um problema é definida por um conjunto de parâmetros (genes)
- Parâmetros são combinados para formar strings de valores (cromossomos)
- Strings podem ser vetores
 - Reais
 - Binários
 - Inteiros



Codificação

- Genótipo
 - Conjunto de parâmetros representado por um cromossomo
 - Contém informação necessária para construir um organismo
 - Fenótipo codificado
- Fenótipo
 - Produto da iteração de todos os genes
 - Aptidão de um indivíduo depende da performance de seu fenótipo (inferido do genótipo usando função de aptidão)



Codificação

- Tradicionalmente, os indivíduos são representados por vetores binários
 - 1 = presença
 - 0 = ausência
 - Esta representação é independente do problema
 - Os operadores padrão podem ser utilizados



Codificação

- Representações em níveis abstratos mais altos
 - Facilitam sua utilização em determinados domínios, onde essa transformação "fenótipo - genótipo" é muito complexa
 - Necessitam de operadores específicos



Função de aptidão

- Mede o grau de aptidão de um indivíduo
 - Retorna um valor de aptidão numérico proporcional a utilidade ou habilidade do indivíduo
 - Aptidão é a probabilidade do indivíduo sobreviver para a próxima geração
 - Pode envolver uma (otimização de função) ou mais medidas (otimização combinatorial)
 - Ex. projeto de ponte
 - Custo, tempo de construção, capacidade máxima



Função de aptidão

- Cada aplicação requer sua própria função de aptidão
 - Utilizada para definir índice de aptidão
- Métodos para calcular índice de aptidão
 - Padrão
 - Baseada em rank
 - Rank-espaco

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

25



Função de aptidão padrão

- Utiliza apenas informação sobre "qualidade do cromossomo"

$$f_i = \frac{q_i}{\sum_j q_j} \quad q = \text{qualidade do cromossomo}$$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

26



Exemplo

Cromossomo Qual. Rank Apt. padrão

1 4	4	1	0.4
3 1	3	2	0.3
1 2	2	3	0.2
1 1	1	4	0.1

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

27



Aptidão baseada em rank

- Escolha infeliz da escala de medida de qualidade pode prejudicar reprodução
- Aptidão baseada em rank utiliza medida de qualidade apenas para definir um "ranking" de cromossomos por aptidão
- Utiliza algoritmo para selecionar candidato por rank

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

28



Seleção de candidato por rank

1. Classificar os N indivíduos por qualidade
2. Seja a probabilidade de selecionar o i-ésimo candidato dado que os primeiros i-1 candidatos não tenham sido selecionados igual p, exceto para o último candidato (selecionado se nenhum outro foi)
3. Selecionar o candidato utilizando as probabilidades computadas

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

29



Exemplo

- Seja $p = 0.667$

Crom.	Qual.	Rank	Apt. padrão	Apt. rank
1 4	4	1	0.4	0.667
3 1	3	2	0.3	0.222
1 2	2	3	0.2	0.074
1 1	1	4	0.1	0.025

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICUSP

30



Aptidão rank-espço

- Medidas de aptidão anteriores ignoram diversidade
 - Mede o quão diferente são os genes dos cromossomos de uma população
- Princípio da diversidade
 - Ser diferente pode ser tão bom quanto ser apto

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICA/USP

31



Aptidão rank-espço

- medida de diversidade

$$D_i = \sum_j \frac{1}{d_{ij}^2}$$

- Aptidão rank-espço

$$R_i = f_i + D_i$$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICA/USP

32



Aptidão rank-espço

1. Selecionar indivíduo com maior aptidão baseada em rank
2. Enquanto a população não estiver completa
 - Selecionar indivíduo com maior aptidão rank-espço (diversidade é medida com relação aos indivíduos já selecionados)
 - Em casa de empate, desempatar com rank ou diversidade)

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICA/USP

33



Exemplo: preparo de biscoitos

- População inicial
 - 14, 11, 12, 11
- Codificação
 - Quantidade de farinha de trigo e de açúcar
- Função de aptidão

$$f_i = \frac{q_i}{\sum_j q_j} \quad q = \text{qualidade da combinação}$$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICA/USP

34



Seleção

- Escolhe preferencialmente, embora não exclusivamente, indivíduos com maiores notas de aptidão
 - Procura de manter a diversidade da população.
- Método da Roleta escolhe indivíduos para fazer parte da próxima geração através de um sorteio

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABICA/USP

35



Seleção

- Cada indivíduo da população é representado na roleta proporcionalmente ao seu índice de aptidão
- Elitismo pode ser utilizado para que os melhores indivíduos não desapareçam da população pela manipulação dos operadores genéticos

8/12/98

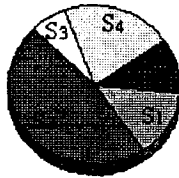
Redes Neurais - André Ponce de Leon F. de Carvalho - LABICA/USP

36



Método da Roleta

Indivíduo S_i	Aptidão $f(S_i)$	Aptidão Relativa
S_1 10110	2.23	0.14
S_2 11000	7.27	0.47
S_3 11110	1.05	0.07
S_4 01001	3.35	0.21
S_5 00110	1.69	0.11



Reprodução

● Fase de reprodução

- N indivíduos da população são selecionados aleatoriamente (favorecendo os mais aptos)
- Pares de indivíduos selecionados são combinados
- São produzidos descendentes (próxima geração)
 - Indivíduos mais aptos serão provavelmente selecionado várias vezes
 - Indivíduos menos aptos podem nem ser selecionados



Reprodução

● Fase de reprodução

- Cromossomos da próxima geração são produzidos utilizando operadores genéticos
 - Elitismo
 - Crossover
 - Um ponto
 - Mais de um ponto
 - Uniforme
 - Mutação



Crossover

- Recombinação de características dos pais durante a reprodução
- Permite que as próximas gerações herdem essas características
- Funcionamento
 - Escolhe dois indivíduos e divide seus cromossomos em uma posição aleatória

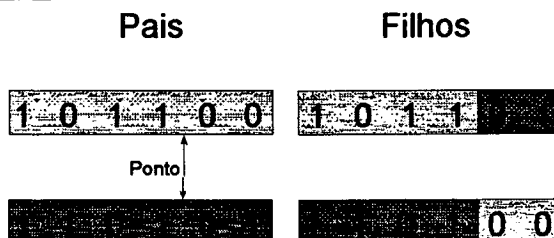


Crossover

- É o operador genético predominante
 - A taxa de crossover deve ser maior que a taxa de mutação
 - Taxa de crossover: $0.6 < P_c < 1.0$
 - Caso crossover não seja aplicado, descendentes são iguais aos pais
- É a operação mais importante para exploração rápida do espaço de busca



Exemplo de Crossover





Mutação

- Introdução e manutenção da diversidade genética
 - Aplicado a cada indivíduo após crossover
- Altera *randomicamente* um ou mais componentes de uma estrutura escolhida
- Assegura que a probabilidade de se chegar a qualquer ponto do espaço de busca nunca será zero
- Operador genético secundário
 - taxa de mutação pequena $P_m \cong 0.001$

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

43



Exemplo de Mutação

Antes da Mutação: 1 1 1 0 0

1	1	1	0	0
---	---	---	---	---

Depois da Mutação: 1 1 0 0 0

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

44



Convergência

- Se o GA estiver corretamente implementado, a população evolui em gerações sucessivas
- Aptidão do melhor indivíduo e do indivíduo médio aumentam em direção a um ótimo global

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

45



Convergência

- Convergência é o progressão em direção à uma uniformidade crescente
 - Um gene converge quando 95% da população compartilha o mesmo valor
 - A população converge quando todos os genes tiverem convergido

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

46



Escolha de parâmetros

- Escolhidos de acordo com o problema
 - Quantos cromossomos em uma população
 - Poucos $\Rightarrow$ efeito pequeno do crossover
 - Muitos $\Rightarrow$ aumenta tempo de computação
 - Taxa de mutação
 - Baixa $\Rightarrow$ mudanças lentas
 - Alta $\Rightarrow$ traços desejados não são mantidos (caos)

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

47



Escolha de parâmetros

- Outros parâmetros
 - Quantos indivíduos selecionados para reprodução?
 - Quantos pontos de crossover?
 - Critério para medir aptidão?
- Manter limites no tamanho da população e complexidade da análise
 - Algoritmo pode se tornar ineficiente

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

48



Um Algoritmo Genético

1. Escolher população inicial de cromossomo
2. Enquanto não convergir
 - Avaliar cada cromossomo da população
 - Se convergiu
 - Então: Sair do loop
 - Senão: Selecionar indivíduos mais aptos
 - Criar novos cromossomos aplicando mutação e crossover
 - Voltar para o passo 2

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

49



Comparação com outras técnicas

- Várias técnicas têm sido propostas para resolver problemas de busca e otimização
 - Todas assumem que o problema é definido por uma função de aptidão que precisa ser maximizada
 - Várias técnicas são aplicadas apenas a domínios específicos (ex. programação dinâmica)

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

50



Exemplo

- Encontrar de x para o qual a função $f(x) = x^2 - 3x + 4$ assume o valor mínimo
 - Assumir que $x \in [-10, +10]$
 - Codificar x como vetor binário
 - Criar uma população inicial com 4 indivíduos
 - Utilizando mutação em apenas um gene da população e crossover, definir o valor mínimo após no máximo 10 gerações

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

51



Aplicações

- Otimização de função numérica
- Otimização combinatorial
 - Problema do caixeiro viajante
 - Problema de empacotamento
 - Alocação de recursos (job shop scheduling)
- Projetos
 - Projeto de pontes
- Aprendizado de Máquina
 - Jogos

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

52



Conclusão

- Conceitos básicos
- Evolução Natural
- Algoritmos genéticos
 - Codificação
 - Função de aptidão
 - Operadores Genéticos
 - Reprodução

8/12/98

Redes Neurais - André Ponce de Leon F. de Carvalho - LABIC/USP

53