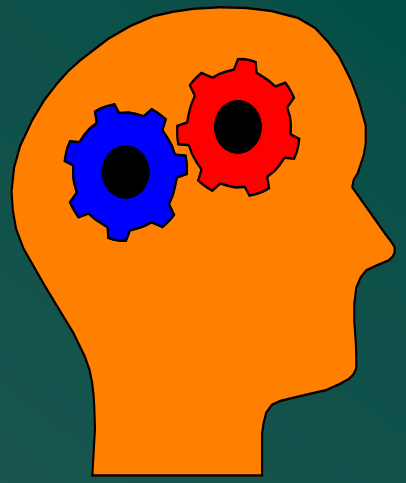
Redes Neurais Artificiais

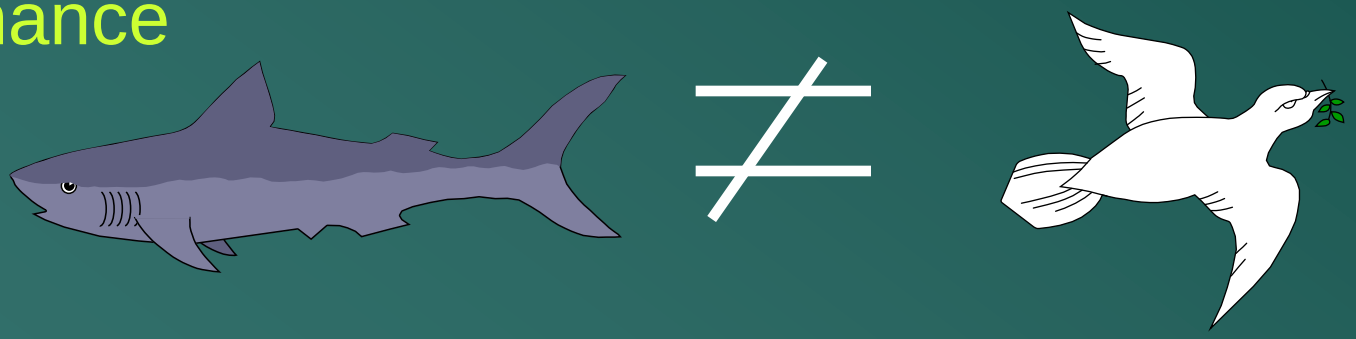


Profa. Teresa Ludermir

Sistemas Inteligentes

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



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

O que são Redes Neurais Artificiais

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

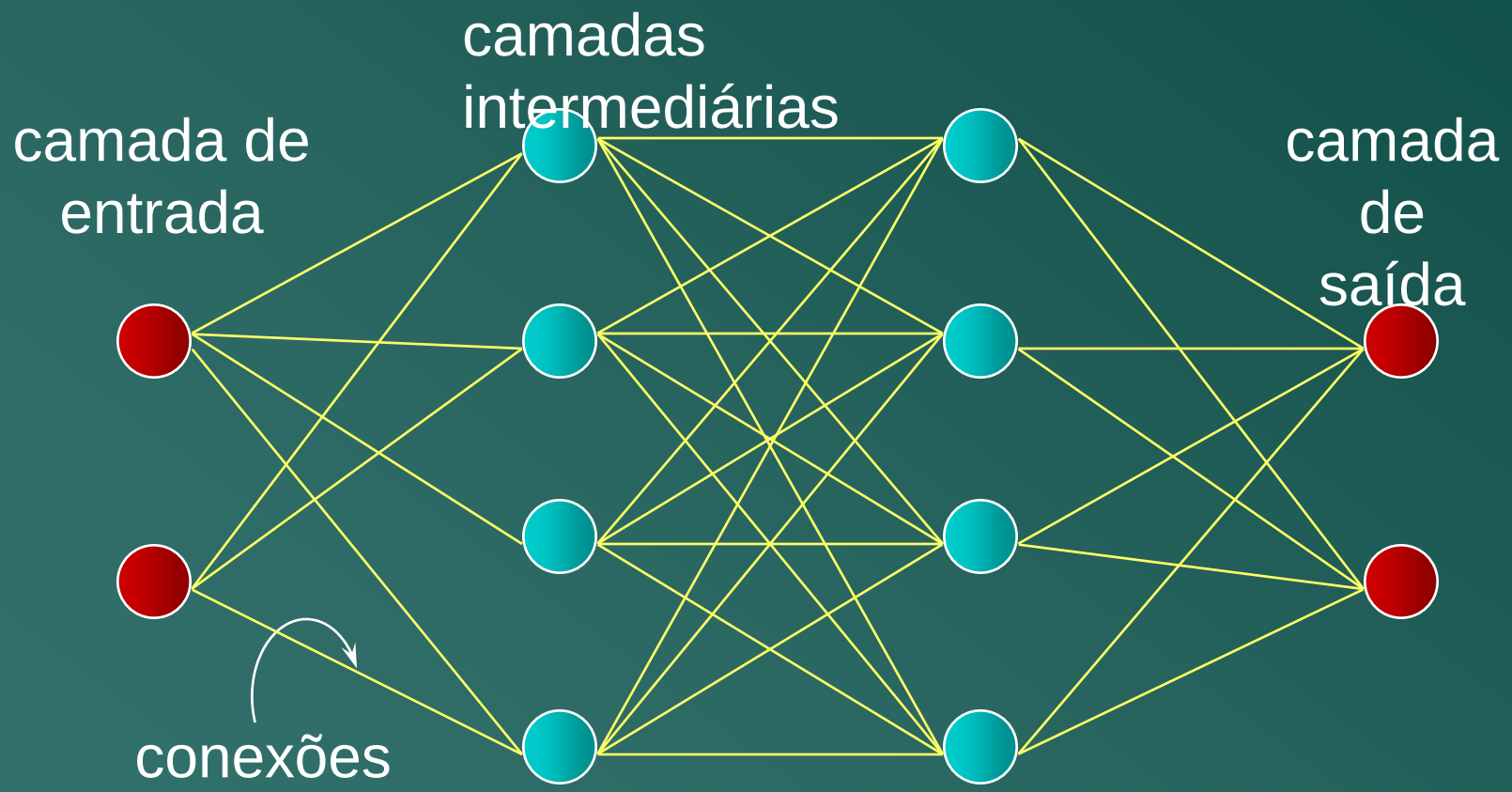
O que são Redes Neurais Artificiais

- RNA: estruturas distribuídas formadas por grande número de unidades de processamento conectadas entre si
- Multi-disciplinaridade : Ciência da Computação, Matemática, Física, Engenharias, Psicologia, Biologia, Lingüística, Filosofia, etc

O que são Redes Neurais Artificiais

- 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

Redes Neurais Artificiais



Características das RNAs

- Aprendem através de exemplos
  - Inferência estatística não paramétrica
- Adaptabilidade
- Capacidade de generalização
- Tolerância a falhas
- Implementação rápida

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-endereçavel 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 simbólica	Problemas de percepção
Ambiente operacional	bem definido muito restrito	pouco definido não restrito

Potenciais áreas de aplicação das RNAs

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

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

Clustering/categorização

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

Aproximação de funções

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

Previsão

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

Otimização

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

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

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

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

Década de 40 : O começo

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

1950-1960: Anos de euforia

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

Década de 70: Pouca atividade

- (1969) Minsky & Papert analisam Perceptron e mostram suas limitações
  - Não poderiam aprender a resolver problemas simples como o OU-exclusivo
  - Causou grande repercussão

Década de 70: Pouca atividade

- (1971) Aleksander propõe redes Booleanas
- (1972) Kohonen e Anderson trabalham com RNA associativas
- (1975) Grossberg desenvolve a Teoria da Ressonância Adaptiva (redes ART)

Década de 80: 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*
  - Werbos (1974)

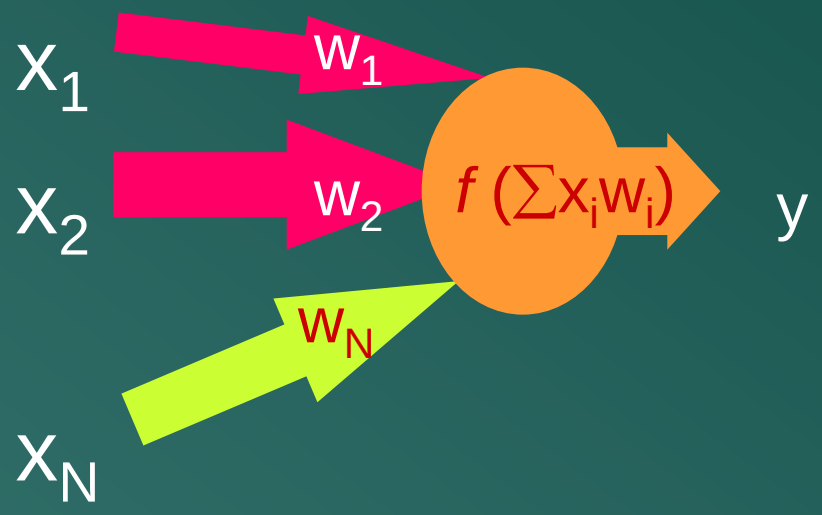
Conceitos básicos

- 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

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

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



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

Funções de ativação

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

Funções de ativação

- Função de ativação
  - Atualiza estado de ativação
    - a(t + 1) = F [a(t), u(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)
  - $a(t+1) = \dfrac{(1 - e^{-\lambda u(t)})}{(1 + e^{-\lambda u(t)})}$ (tangente hiperbólica)

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

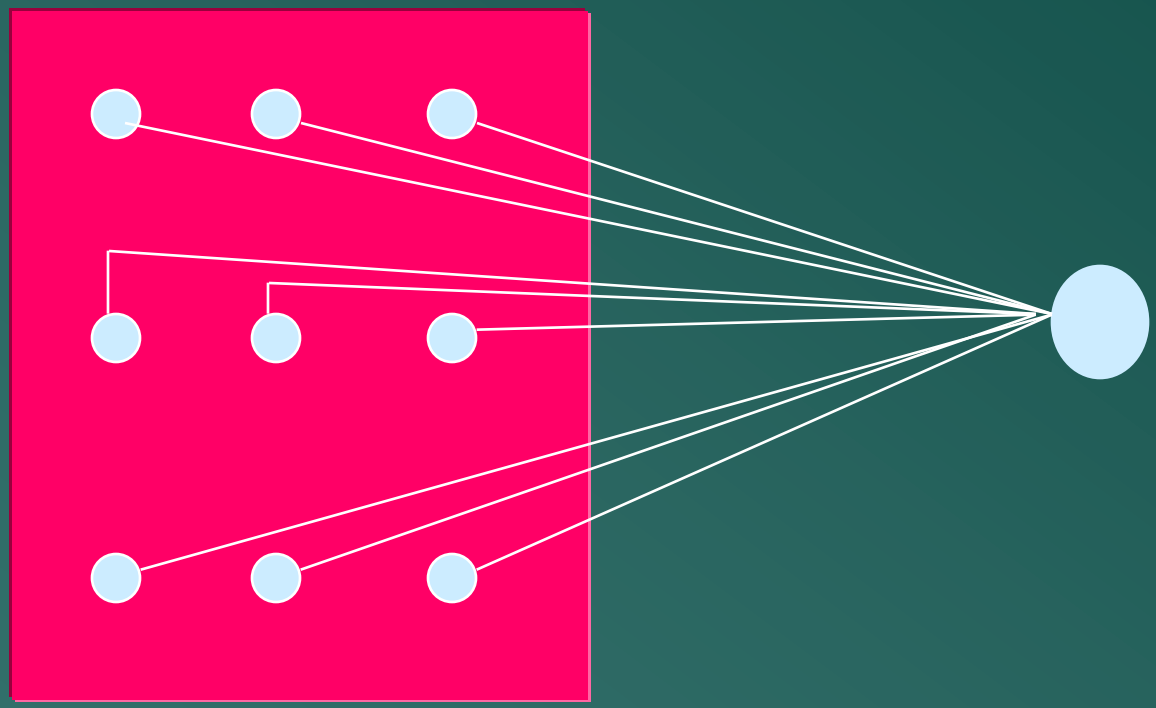
Conexões

- 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

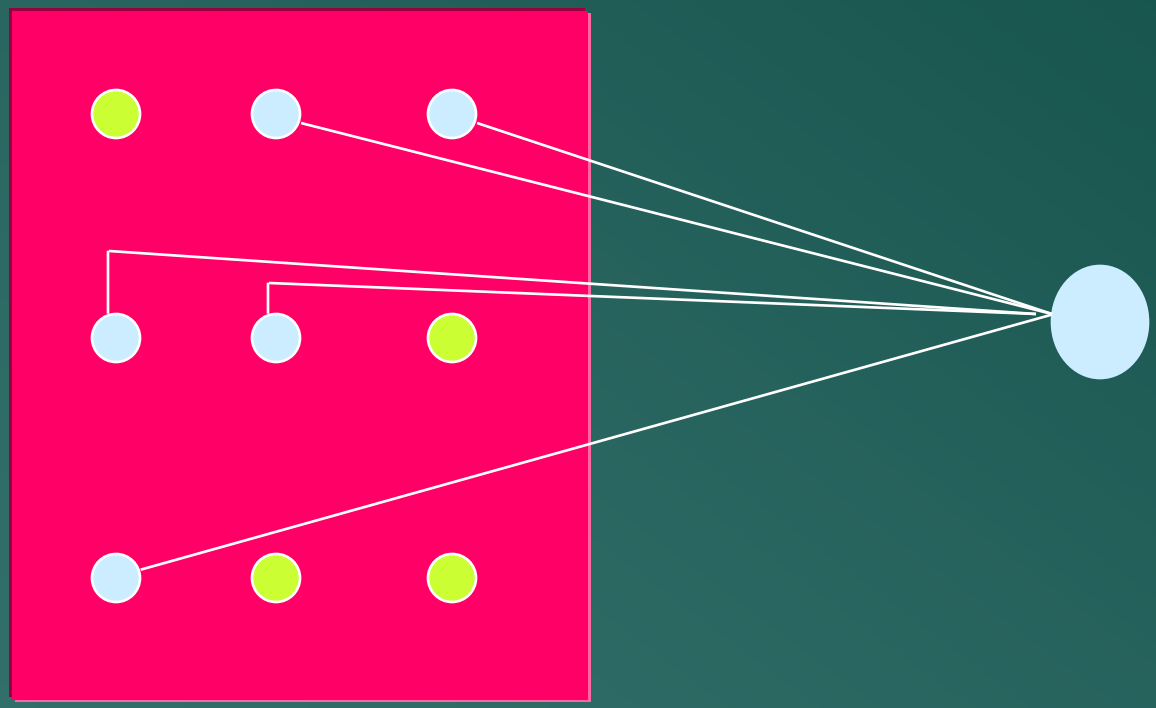
Topologia

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

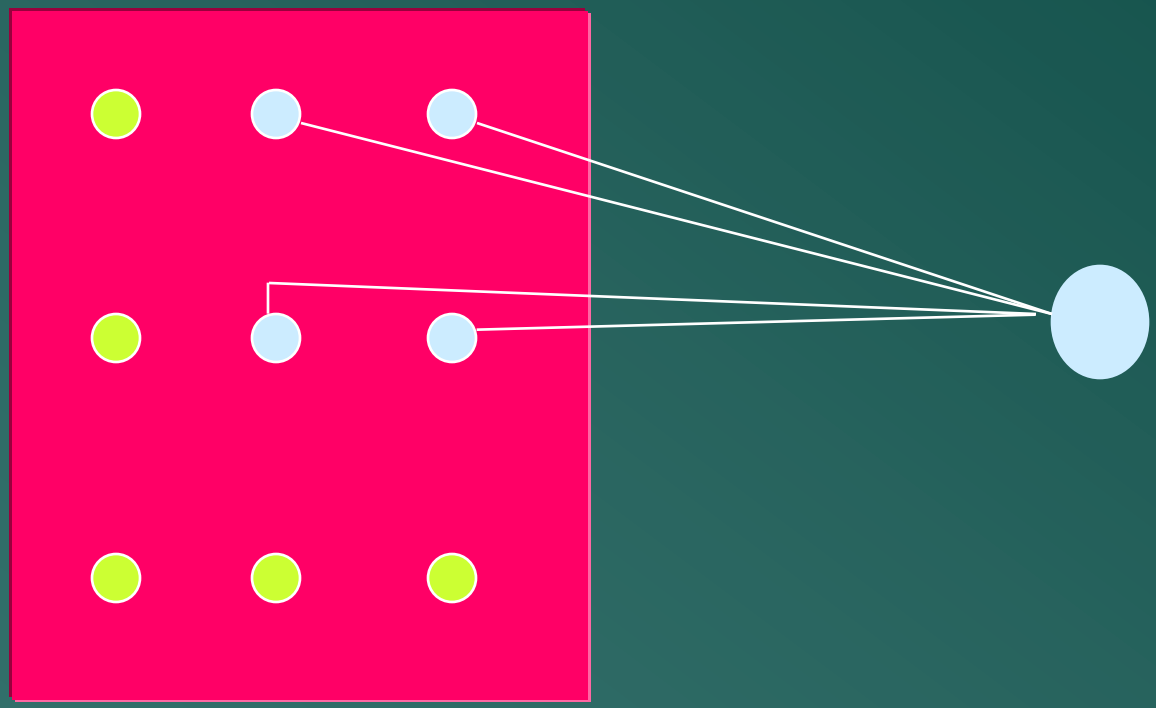
Completamente conectada



Parcialmente conectada



Localmente conectada

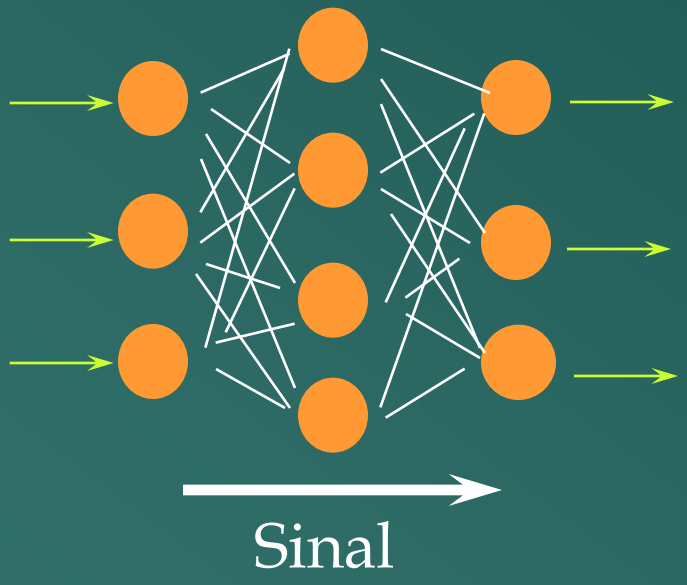


Topologia

- 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

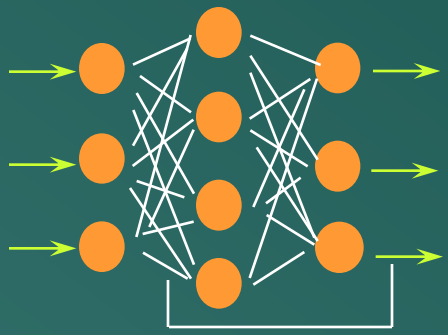


OU

- Tipo mais comum

Redes recorrentes

- Possuem conexões ligando saída da rede a sua entrada



- Podem lembrar entradas passadas e, conseqüentemente, processar seqüência 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

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

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

Aprendizado

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

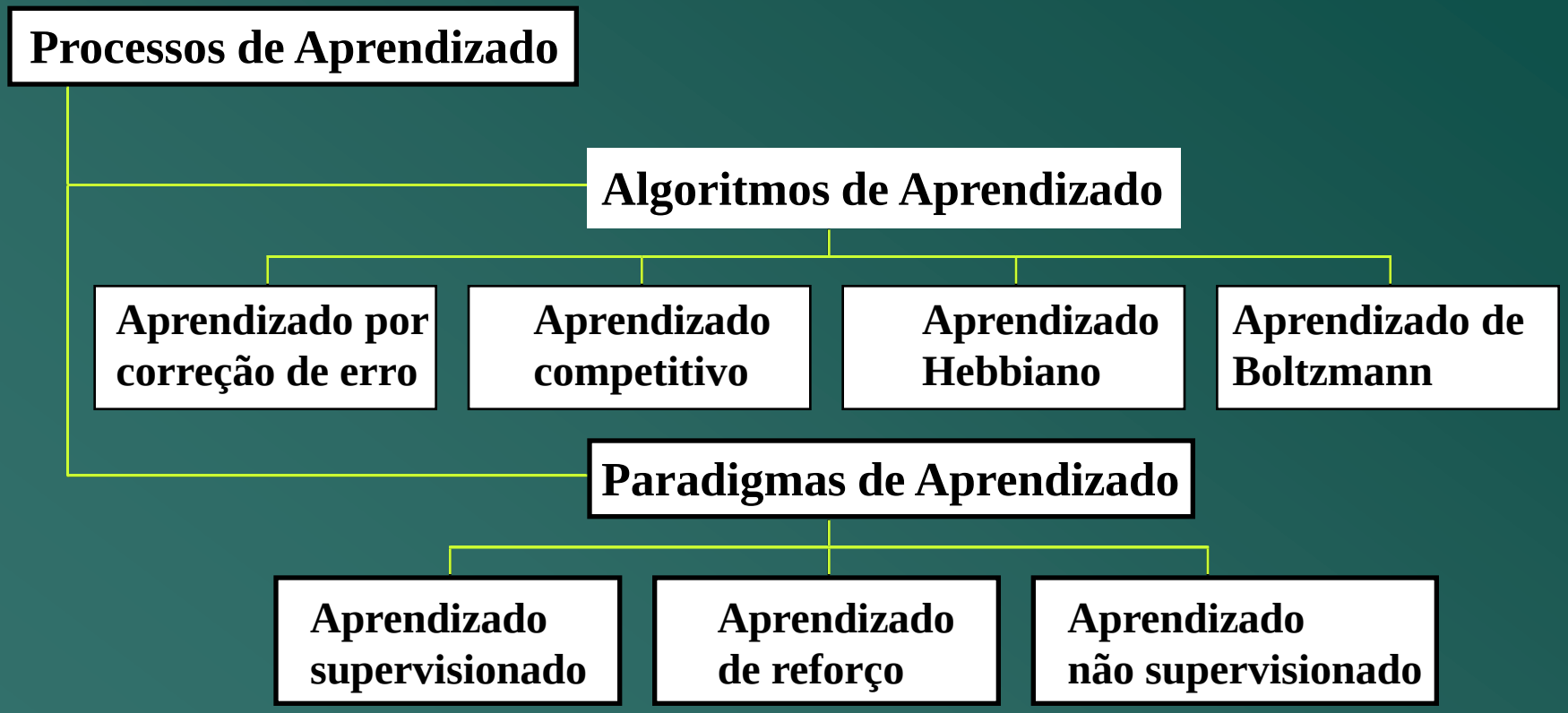
Aprendizado

- 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

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

Aprendizado



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
  - $c(t) = -1/2 \sum e^2_k(t)$
  - Minimização de c(t) utiliza método de gradiente descendente
  - Aprendizado atinge solução estável quando os pesos não precisam mudar muito

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

Aprendizado por correção de erro

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

Aprendizado competitivo

- Von der Marlsburg (1973)
- Neurônios competem entre si para ser ativado
  - 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)
  - $\sum w^2_{ik}(t) = 1$ (para cada neurônio)
- Modelo matemático:
  - $\Delta w_{ik}(t) = \eta (x_i(t) - w_{ik}(t))$ (vencedor)

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 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

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 Heteroassociação	 Associar x com x Associar x com y	 Não supervisionado Supervisionado
Classificação de padrões	Categorias definidas Cluster, deteção de características	Supervisionado Não supervisionado
Previsão	Série temporal	Supervisionado
Controle	Corpo humano	Reforço
Localização	Localização de alvo	Não supervisionado