

# Modelos ART

## Adaptive Resonance Theory

Germano C. Vasconcelos  
Centro de Informática - UFPE

# Introdução

---



- Capacidade de Generalização
  - Resposta satisfatória ao universo de padrões
  - Problema: Conhecimento modificado ao longo do tempo
    - Desempenho da rede é prejudicado
  - Solução: adaptar aos novos padrões ⇒ Plasticidade
- Re-treinar a rede com novos padrões
  - Perda de informação previamente definida ⇒ Estabilidade
  - Solução: re-treinar a rede com todos os padrões
- Dilema da Plasticidade-Estabilidade
  - Aprender novas informações sem perder informações previamente aprendidas



# Introdução

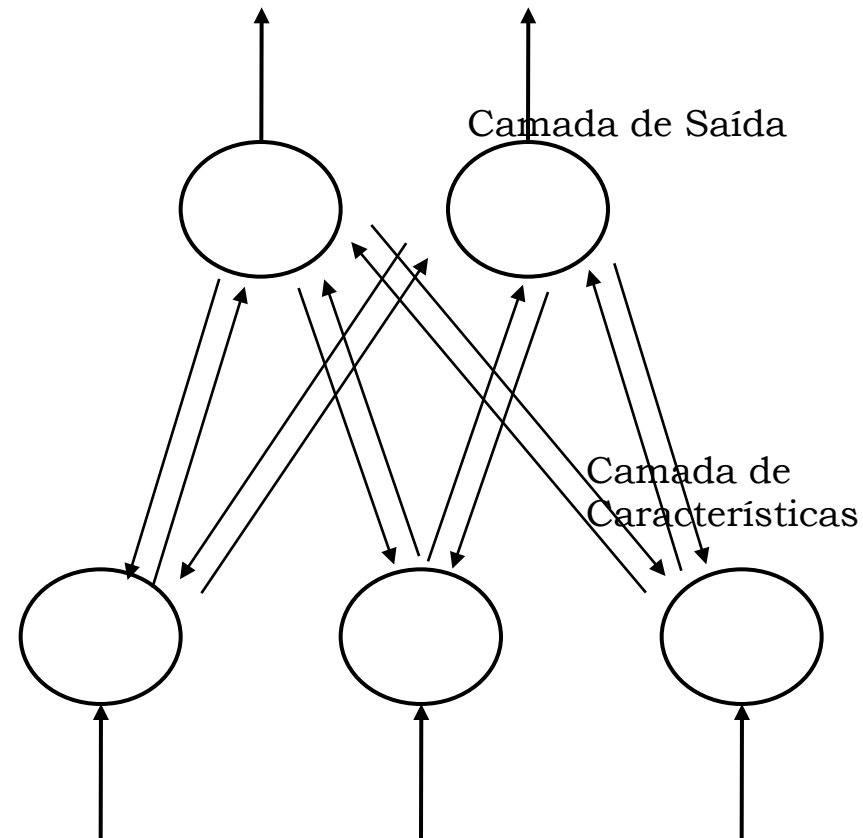
---



- Maioria dos modelos de Redes Neurais não resolvem o dilema plasticidade-estabilidade
  - Treinamento de uma rede MLP com um novo padrão pode ter como efeito colateral a perda das informações aprendidas anteriormente
- Redes ART propõem solução para este dilema
  - Rede Neural Incremental
  - Aprendizado Competitivo e não supervisionado
  - Baseada na teoria da ressonância adaptativa

# Teoria da Ressonância Adaptativa (ART)

- Carpenter and Grossberg (1976)
- Inspirado em estudos sobre detectores de características biológicos
- Algoritmo de adaptação On-line
- Rede recorrente
- Camada de saída competitiva
- Aplicações de segmentação de dados
- Dilema Estabilidade-Plasticidade



# Modelos ART



## Teoria da Ressonância Adaptativa

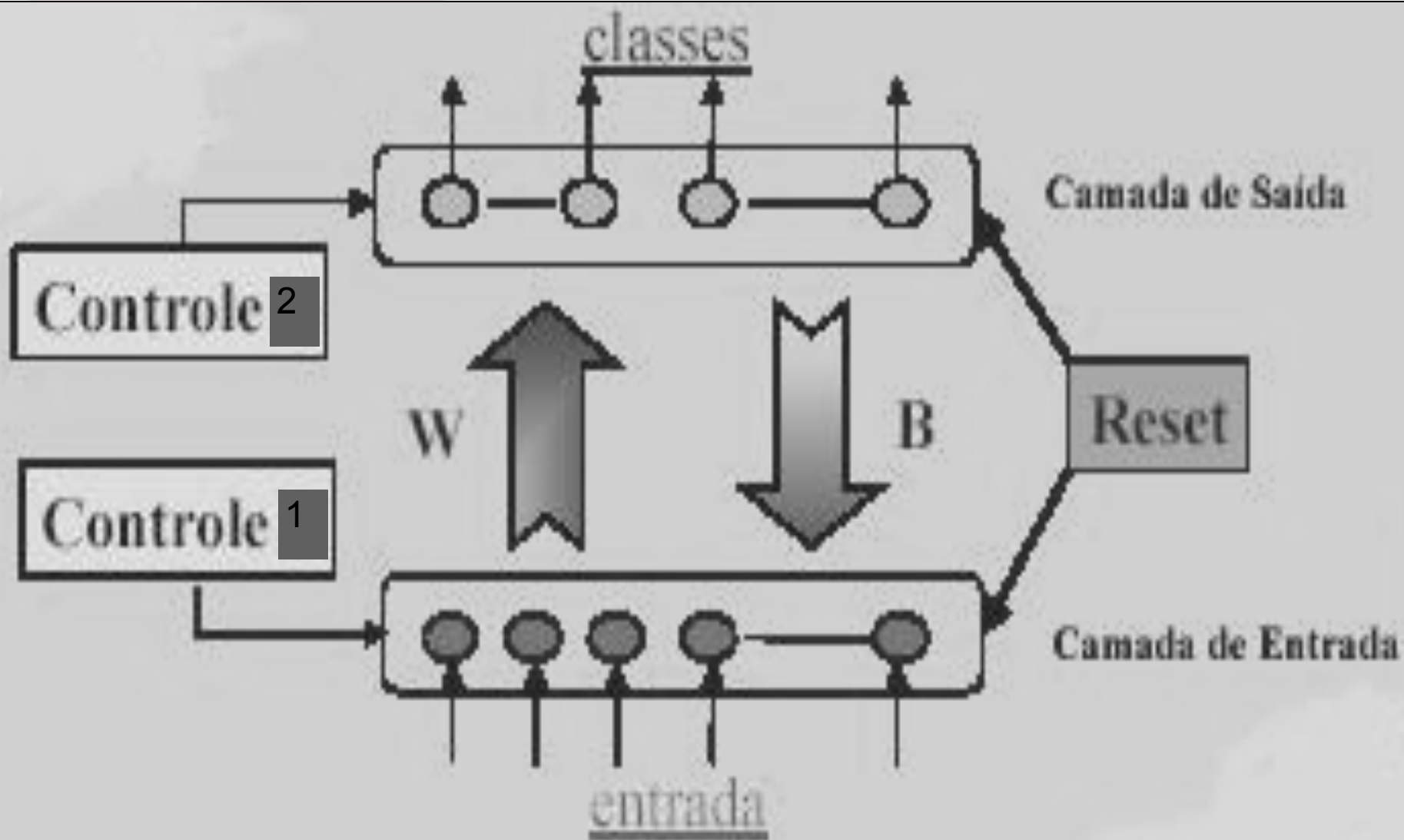
- Rede neural auto-organizável para reconhecimento de padrões baseada em dados biológicos e comportamentais
- Ressonância adaptativa ocorre quando padrões de atividade nas camadas de entrada e de saída se reforçam mutuamente

## Família ART

- ART-1: Entradas e saídas binárias
- ART-2: Entradas e saídas analógicas
- ART-3: Inclui neurotransmissores
- Fuzzy ART: ART + lógica fuzzy
- ARTMAP: supervisionado
- Fuzzy-ARTMAP: combinação dos dois modelos



# Arquitetura da rede ART



# Arquitetura ART

---



- Rede com duas camadas com funções diferentes
  - Entrada: processa dados de entrada
  - Saída: agrupa os padrões em clusters
- Conexões
  - Feedforward ( $W$ ): valores reais
  - Feedback (pesos  $b$ ): valores binários (protótipo dos padrões)
  - Laterais (inibidores laterais): camada de saída
    - Não adaptáveis e bipolares ( $-1$ : neurônios distintos;  $+1$ : mesmo neurônio)
- Reset
  - Similaridade mínima aceitável entre protótipo e entrada



# Arquitetura ART

---



- Unidades de Controle:
- Controle 1:
  - Fluxo de dados para a camada de entrada
  - 1: se uma entrada válida for apresentada a rede
  - 0: se qualquer nó na camada de saída está ativo
- Controle 2:
  - Determina fluxo de dados para a camada de saída
  - Habilita ou desabilita nós na camada de reconhecimento
  - 0: após falha de um teste de vigilância (desabilita os nós da camada de reconhecimento e re-seta seus estados de ativação para 0)
  - 1: se uma entrada válida for apresentada à rede





# Treinamento da Rede ART

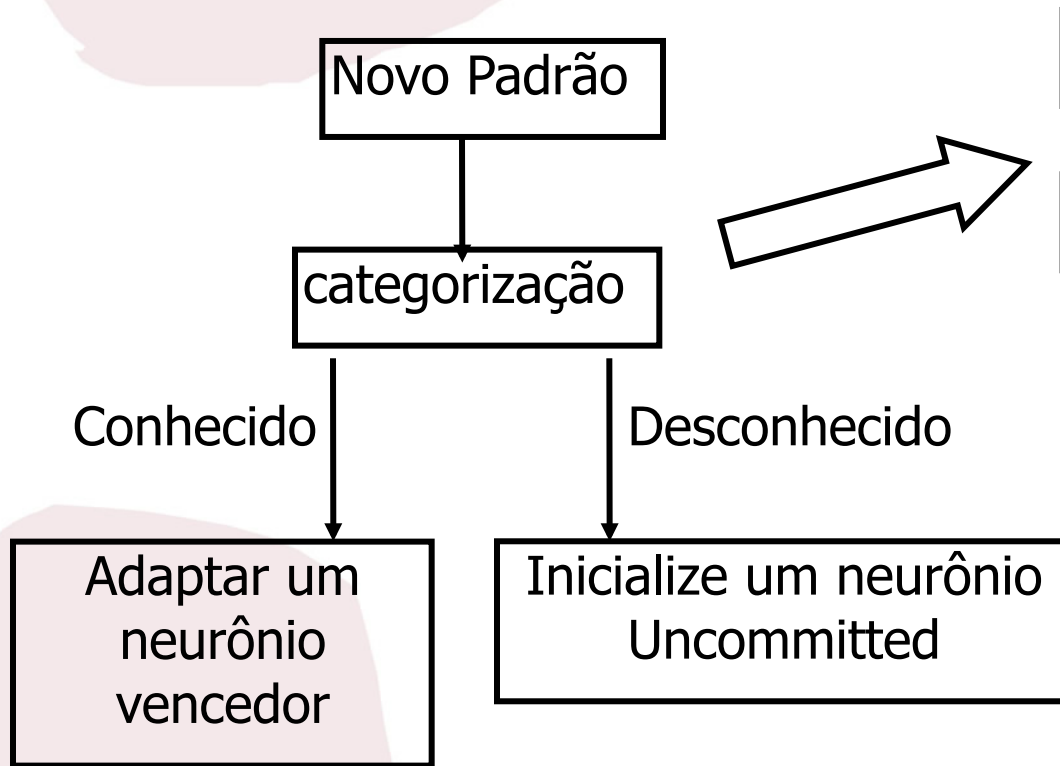
---



- Não-supervisionado
- Processo de treinamento
  - Vetor de entrada continuamente modificado passa para frente e para trás (ressoa) entre as camadas em um processo cíclico
- Competitivo
- Tipos:
  - Rápido
  - Lento
- Inicialização dos pesos e da vigilância
- Sub-fases
  - Reconhecimento
  - Comparação
  - Busca



# Esquema do Algoritmo de Treinamento



Reconhecimento

Comparação

- Padrão de entrada casa com exemplares armazenados
- Se suficientemente semelhante, padrão de entrada junta-se ao cluster de maior semelhança e os pesos são adaptados
- Caso contrário, um novo cluster é criado com o novo padrão como exemplar

# Treinamento da Rede ART - Reconhecimento

---

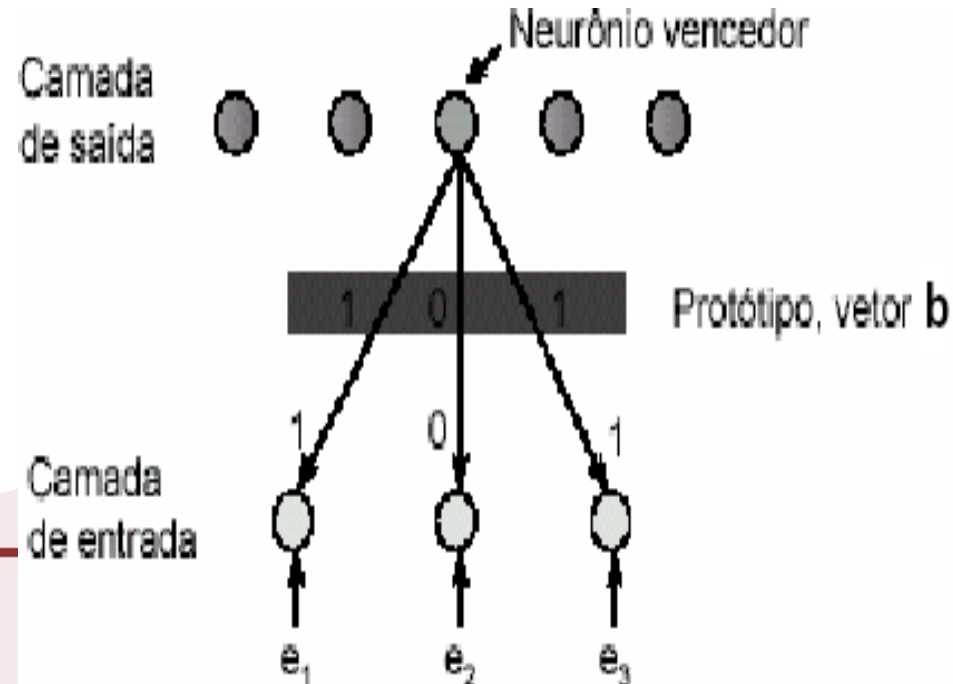


- Cada nó da camada de entrada possui três entradas
  - Componente do vetor de entrada
  - Sinal de feedback da camada de saída
  - Sinal de controle C1
- Fluxo de dados na camada de entrada é controlado pela regra dos “dois-terços”
  - = 1 Se duas das três entradas estão ativas
  - = 0 Caso contrário
- Cada neurônio da camada de saída representa um cluster
  - Vetor de pesos  $W$  é o centro do cluster



# Treinamento da Rede ART - Reconhecimento

- Vetor de entrada é comparado com cada cluster
  - Cluster com maior resposta (mais semelhante ao vetor de entrada) é selecionado (peso  $W$  e resultado dos dois-terços)
  - Inibição lateral faz com que apenas o nó com maior resposta fique ativo
- Nó vencedor da competição passa protótipo (vetor  $b$ ) de seu cluster para a camada de comparação



# Treinamento da Rede ART - Comparação

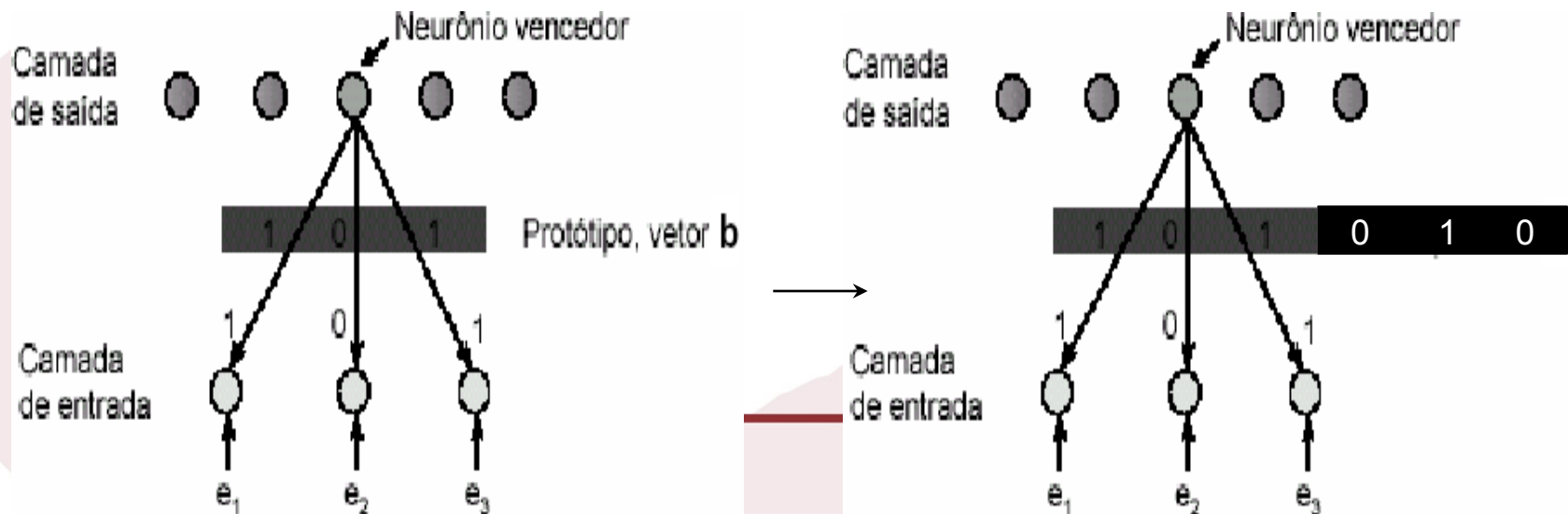


- Dois vetores são apresentados à camada de entrada e passados ao mecanismo de reset
  - Vetor de entrada (X)
  - Vetor de comparação (Z)
    - Vetor de entrada (X), Vetor de protótipo (T) e Sinal de controle C1
    - C1 Igual a zero: Operação AND entre a entrada e o protótipo
- Reset: Semelhança entre o vetor de entrada e o vetor de comparação (Z) utilizando o limiar de vigilância
  - $S = \sum z_{ij} / \sum x_i$ , onde  $z_{ij} = t_{ij} x_i$
  - Se  $S \geq \rho$ . Clusterização terminou e o cluster é indicado pelo nó ativo na camada de saída.
  - Senão  $S < \rho$ . Padrão não casa com o protótipo: fase de busca



# Problema na Computação da Similaridade

- Depende do número de 1s no vetor de entrada  $X$ 
  - Ex:  $X=[1, 0, 0]$  e  $X=[1,1,1]$ . Mesmo que distintos um mesmo número de bits de um protótipo  $T$  (ex:  $[0,0,1]$ ) na camada de saída, os dois vetores produziram valores diferentes para  $S$
  - Solução: duplicar o vetor a ser representado com seu complemento



# Treinamento da Rede ART – Busca

---



## ■ Objetivo

- Encontrar novo cluster 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
  - C2 é setado para zero
- Vetor de entrada é reaplicado à rede se existirem outros nós. Caso contrário outro nó é alocado na camada de saída (novo cluster)

# Algoritmo de Treinamento



- Inicializar todas as conexões  $W(0) = 1/(1+n)$  e  $B(0) = 1$  (onde  $n$ =número de entradas);
- Repita
  - Para cada padrão de treinamento ( $X$ )
  - Entrar na fase de reconhecimento
  - Comparar protótipo do vencedor com a entrada
  - Se comparação  $>$  valor de vigilância
    - Então Atualizar pesos  $W(t+1) = LZ / (L-1 + \sum z)$  e  $B(t+1) = B(t) \cdot X(t)$  do neurônio vencedor
    - Senão Desabilitar neurônio (Busca)
      - Se ainda existir neurônio não analisado
      - Então Voltar
    - Senão Alocar novo neurônio à camada de saída
- Até conjunto de protótipos não mudar





# Algoritmo de Treinamento: Exemplo



- $X1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0]$
- $X2 = [1 \ 0 \ 0 \ 1 \ 1 \ 1]$
- Com a adaptação, considerando  $L=2$ , ou seja:  $W(t+1) = 2Z / (2 - 1 + \sum z_k)$   
(lembrando  $W(t+1) = LZ / (L-1 + \sum z_k)$ )
- $W1$  é definido como  $W1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0]$ , e
- $W2$  é definido como  $W2 = [0.4 \ 0 \ 0 \ 0.4 \ 0.4 \ 0.4]$
  
- Sem escalar, a atualização produziria:
  - $W1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0]$
  - $W2 = [1 \ 0 \ 0 \ 1 \ 1 \ 1]$
  
  - Se  $X1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0]$  fosse apresentado como entrada, a unidade representada por  $W2$  poderia vencer o processo, gerando um problema

# Algoritmo de treinamento

---



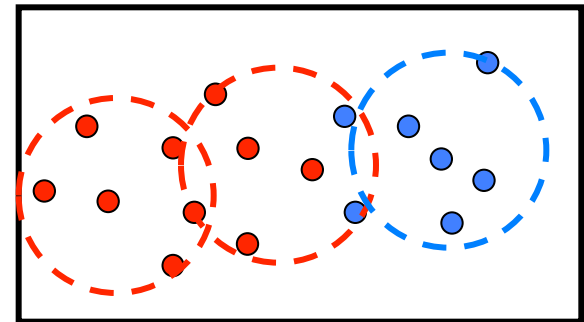
- Rede pode reentrar no modo de treinamento a qualquer tempo
  - Permite incorporar novos dados
  - Rede pode aprender em ambiente dinâmico
  - Resolve dilema estabilidade-plasticidade
  
- Existem dois esquemas de treinamento
  - Treinamento rápido e lento



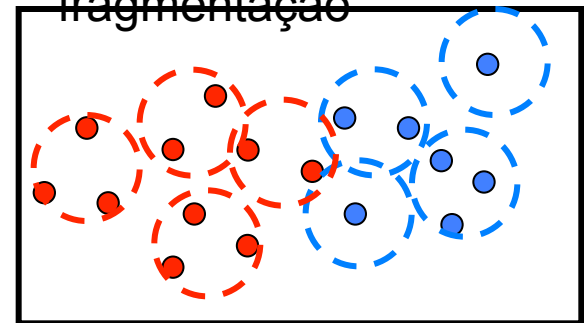
# Treinamento

- ART é muito sensível a variações nos parâmetros durante treinamento
- Mais crítico é o parâmetro de vigilância ( $\rho$ )
- Controla resolução do processo de clusterização
  - Granularidade
- Valor baixo ( $< 0.4$ )
  - Resolução baixa: cria poucos clusters iniciais
- Valor alto
  - Resolução alta: pequenas variações no valor de entrada levarão a criação de novos clusters
- Valor depende do contexto

$\rho$  pequeno, imprecisão



$\rho$  grande,  
fragmentação



# Parâmetro de Vigilância

---



- Solução ótima: variar parâmetro durante o treinamento
  - Valor inicial baixo define clusters
  - Valor seguinte mais alto otimiza clusterização
- Semelhante à técnica utilizada para definição de raio de vizinhanças na redes de Kohonen
- Parâmetro pode ser modificado para punir rede por classificações erradas (em treinamento supervisionado)



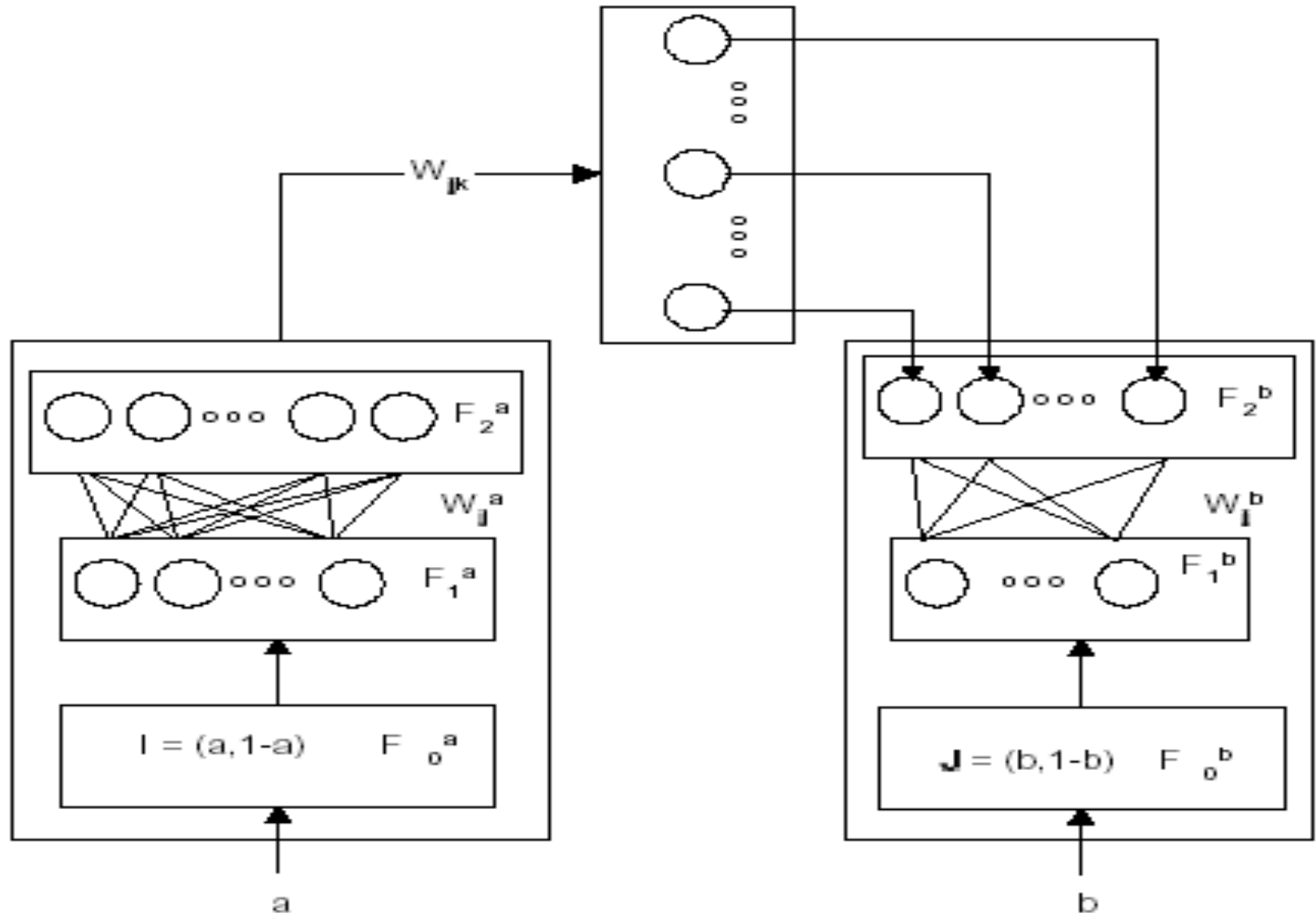
# Redes ARTMAP



- Dois módulos de ART
    - Entrada
    - Saída
  - Módulo de MapField
  - Aprendizado supervisionado
  - Funcionamento:
    - Entradas e saídas desejadas associadas aos módulos ART
    - Associações das saídas no módulo Map Field
  - Número de neurônios no MapField: igual ao de ARTb
- Todos os neurônios de ARTa estão conectados a todos neurônios no Map Field
  - Inicializado com 0
  - Peso = 1, houver ligação entre o vencedor de ARTa e do Map Field
  - Ligação MapField  $\Rightarrow$  ARTb
  - associação 1-para-1
  - Reconhecimento:
    - Escolher o neurônio vencedor em Map Field e verificar associação com ARTb



# Arquitetura do ARTMAP



# Redes ARTMAP

---



- Processo de treinamento baseado em similaridade
  - Diferente de outros tipos de rede: MLP
- Vantagens:
  - Fácil extração de conhecimento
  - Não tem um esquecimento catastrófico
  - Aprendizado rápido
- Desvantagens:
  - Proliferação de categorias
  - Classificação errônea



# Extensões do ARTMAP

---



- Modificações:
  - Na estrutura da rede
  - Na fase de treinamento
  - Na fase de reconhecimento
- ARTMAP-IC
  - Mesma estrutura e aprendizagem
  - Método de distribuição de categorias
    - O número de vencedores: heurística ( $\text{NumNeuA}/\text{NumClasses}$ )
  - Contagem de frequência
    - Frequência de ativação de um neurônio durante o treinamento





# Extensões do ARTMAP - RePART



- Mecanismos adicionais para melhorar o desempenho do ARTMAP
- Vigilância Variável
  - Importância do parâmetro de vigilância
  - Solução: ajuste dinâmico da vigilância
  - Leva em consideração os seguintes parâmetros
    - Freqüência e média (inversa) de ativação de um neurônio
    - Número de neurônios associados à classe correspondente

$$Vig_i(t) = Vig_i(t - 1) + \Delta RFA + \Delta RIAA$$

$$RIA A_i(t) = \frac{T_i(t)}{c_i}$$

$$RFA_i(t) = \frac{c_i}{\sum_{j \in k} c_j} * (1.001 - \frac{N_k}{TN_k})$$



# Extensões do ARTMAP - RePART



- Método de distribuição de categorias :
  - Conjunto de vencedores
  - Conjunto de perdedores
- Mecanismo de recompensa e punição
  - Recompense o conjunto de vencedores

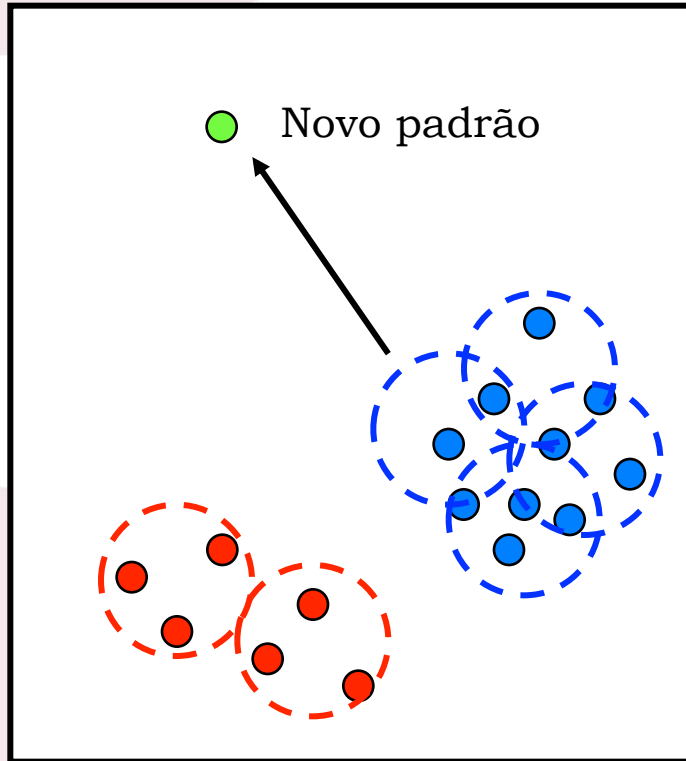
$$R_j = \left( T_j * \frac{NumWin}{NumWin + k_j^w} \right)^2,$$

- Puna o conjunto de perdedores

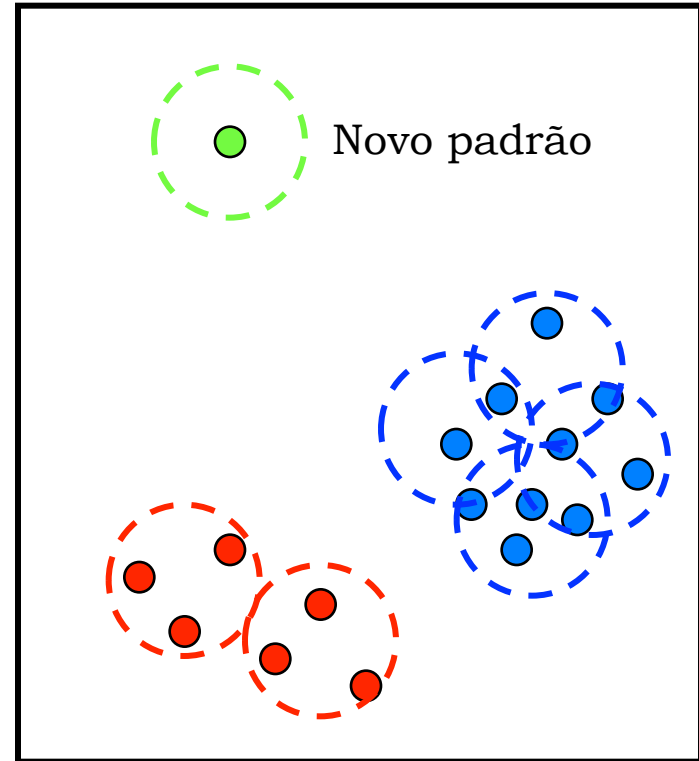
$$P_j = \left( (W - l_j) * \frac{NumLos + k_j^l}{NumLos} \right)^2,$$

# Plasticidade SOM vs. Plasticidade ART

Mapeamento no SOM



Mapeamento no ART



No SOM, dado um novo padrão, um nó existente e sua vizinhança são adaptados em direção ao padrão. Conhecimento prévio é parcialmente esquecido.

# Modelos ART - Aplicações

---



- Processamento de linguagem natural
- Segmentação de documentos
- Recuperação de documentos
- Segmentação de imagens
- Reconhecimento de caracteres
- Reconhecimento de faces
- Detecção de anomalias
- Segmentação de usuários e consumidores
- Mineração de dados
  - Particionamento do conjunto de dados
  - Detecção de clusters emergentes
- Associação de condição-ação (indução de regras)



# Reconhecimento de Faces



# Exercício:

(a) Como uma rede ART construiria uma representação dos padrões de forma não supervisionada para este problema? Considere 2 (dois) valores diferentes para o parâmetro de vigilância (0.6 e 0.3).

Nome	Febre	Enjôo	Manchas
João	S	S	Peq
Pedro	N	N	Grd
Maria	S	S	Grd
José	N	S	Grd

(b) Mostre como a rede ART responderia para outros 2 casos para os dois valores do parâmetro de vigilância