



Introdução Inteligência Computacional

Profa. Dra. Ana Paula Abrantes de Castro e Shiguemori

anapaula.acs@ifsp.edu.br

Instituto Federal de Educação, Ciência e Tecnologia – IFSP

Dr. Elcio Hideiti Shiguemori

elcio@ieav.cta.br

Instituto de Estudos Avançados - IEAv

Curso: Análise e Desenvolvimento de Sistemas
4º. Semestre - IIC14 – 4 aulas Semanais

A stylized, glowing blue neuron is the central focus, with its cell body and branching dendrites and axons. The neuron is set against a dark blue background filled with a complex network of faint, glowing blue lines representing neural connections. Several bright yellow and orange nodes are scattered throughout the network, particularly along the axons and at junctions, suggesting points of high activity or signal transmission. The overall aesthetic is futuristic and scientific.

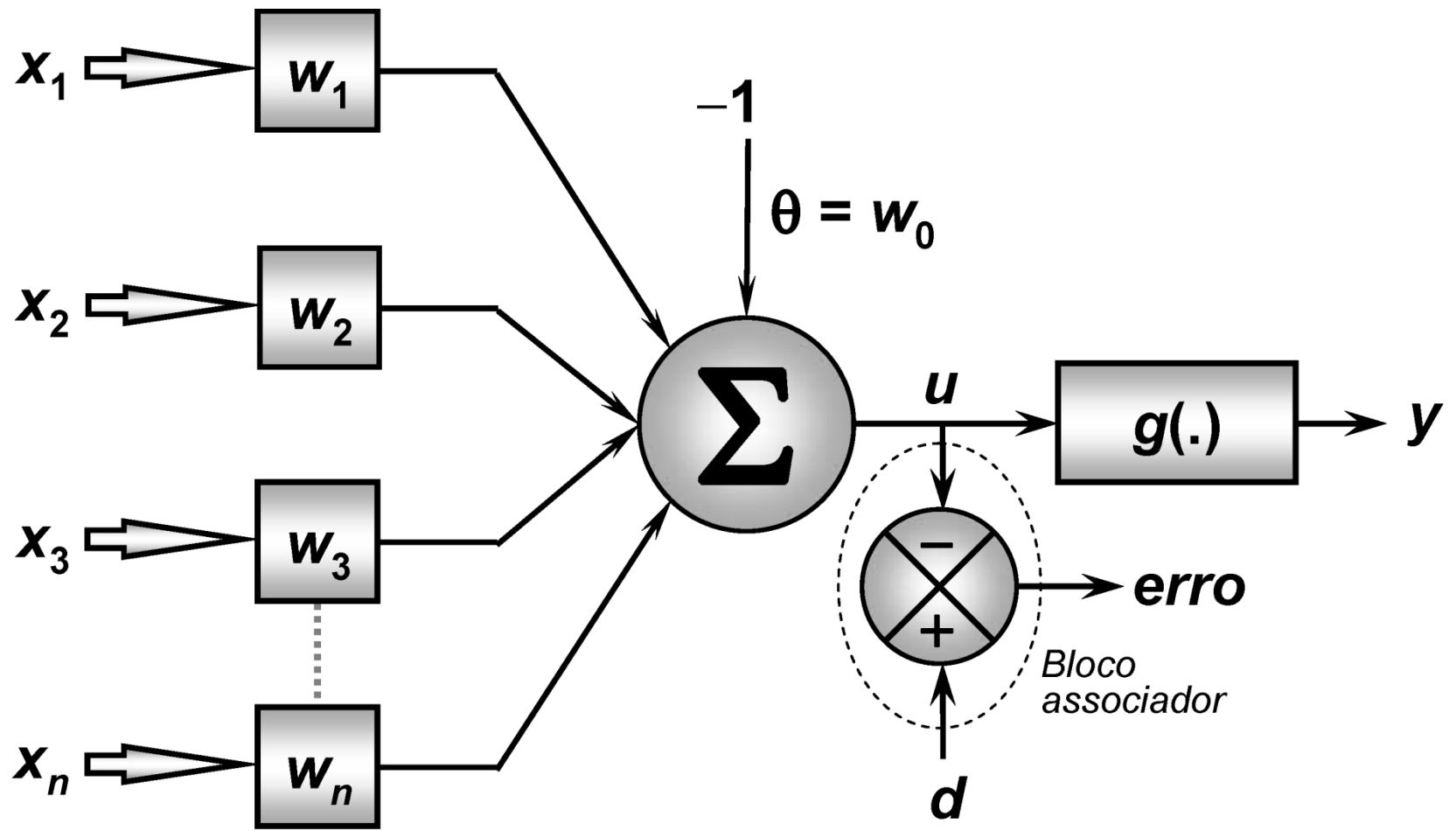
Adaline

- Introdução
- Modelo Adaline
- Bias
- Treinamento
- Interpretação Geométrica
- Exemplos
- Limitações

A Introdução

- A Rede Neural Perceptron, em seus ajustes de peso não é considerada a distância entre a saída e a resposta desejada;
- A Rede Neural Adaline (ADaptive LINEar) foi desenvolvida em 1960 (Widrow e Ho) e sua principal função era na área de processamento digital de sinais;
- Utiliza o ajuste de erro pela Regra Delta;
- Uma Adaline de várias camadas é chamada Madaline;
- É de arquitetura feedforward;
- Soluciona problemas de regressão;

A Exemplo – Classificação 2 Classes



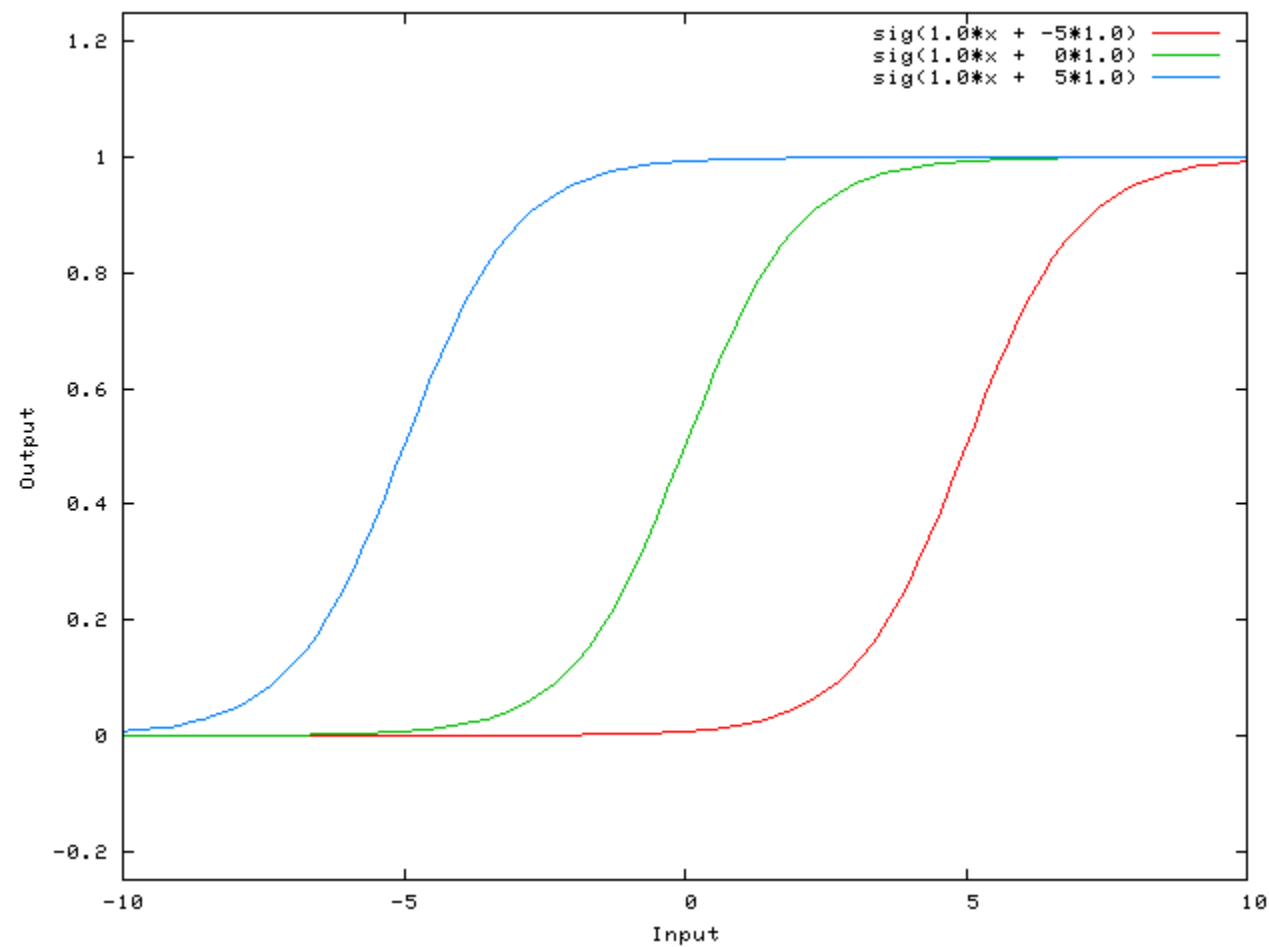
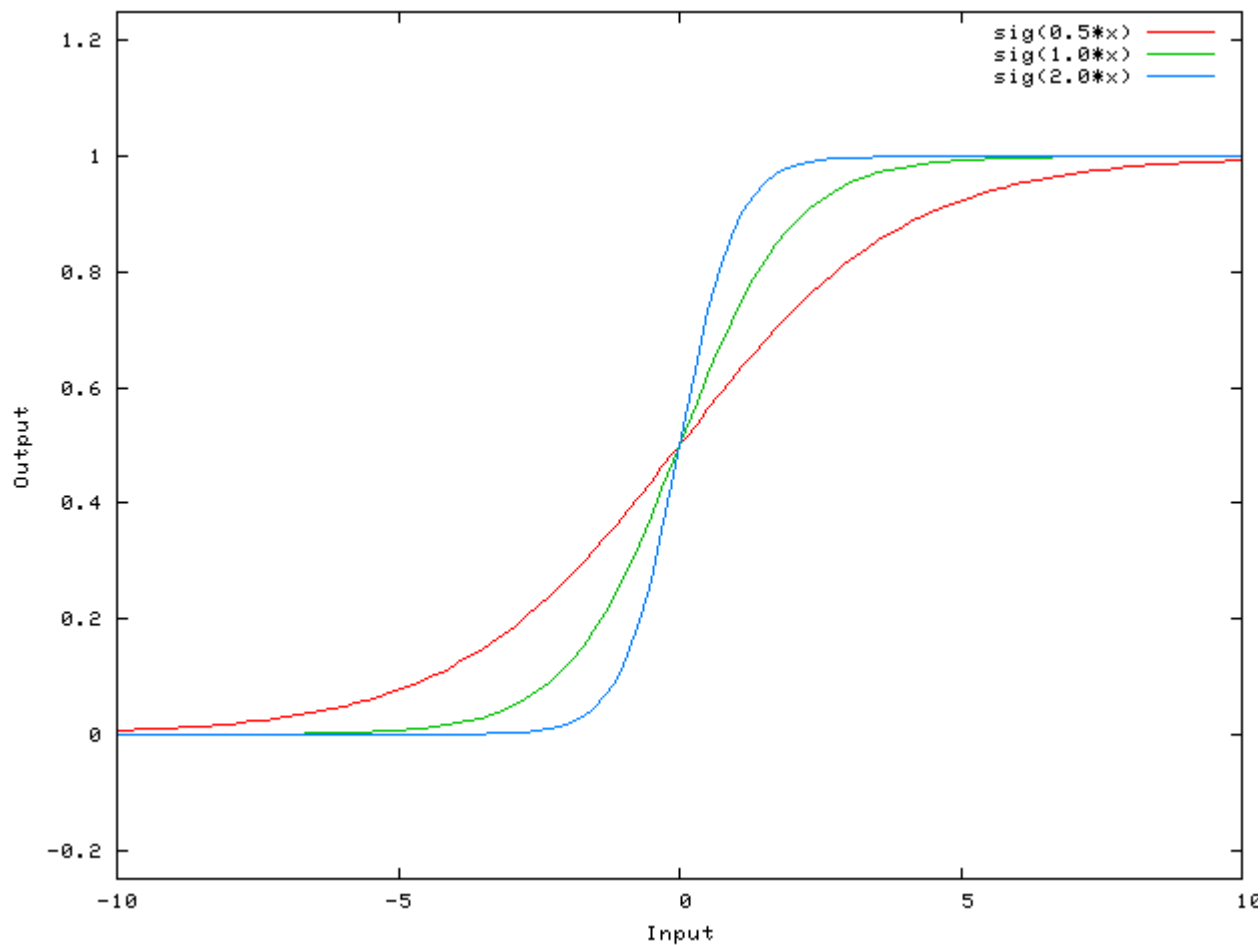
A Bias (limiar) θ

- O bias (θ), também chamado de viés ou limiar tem a mesma função do intercepto para os modelos de regressão. [representa o ponto em que a reta regressora corta um dos eixos]
- Se uma RNA não faz o uso do bias em uma dada camada não será possível produzir uma saída diferente de 0 quando os atributos forem 0;
- O bias promove flexibilidade ao ajuste da RNA;
- Os valores atribuídos ao bias são normalmente -1 ou 1;
- O peso (w_0) deve ser associado ao bias e ajustado como os outros.



Bias (limiar) θ

- Funções sigmoídes sem (esquerda) e com bias = 1 (direita)



Rede Neural Adaline

- ▶ Uma das diferenças com relação à Perceptron é o bloco de verificação de erro, para Perceptron: erro = $d - y$, já para a Adaline erro = $d - u$.
- ▶ Assim, para ajustes dos pesos serão utilizados os valores obtidos antes da função de ativação $g(\cdot)$.
- ▶ O treinamento é baseado na Regra Delta que tem como núcleo o Gradiente Descendente ou Mínimos Quadrados. Assim espera-se encontrar os pesos e limiar que minimizam a diferença entre d e u em função do Erro Quadrático Médio (EQM).

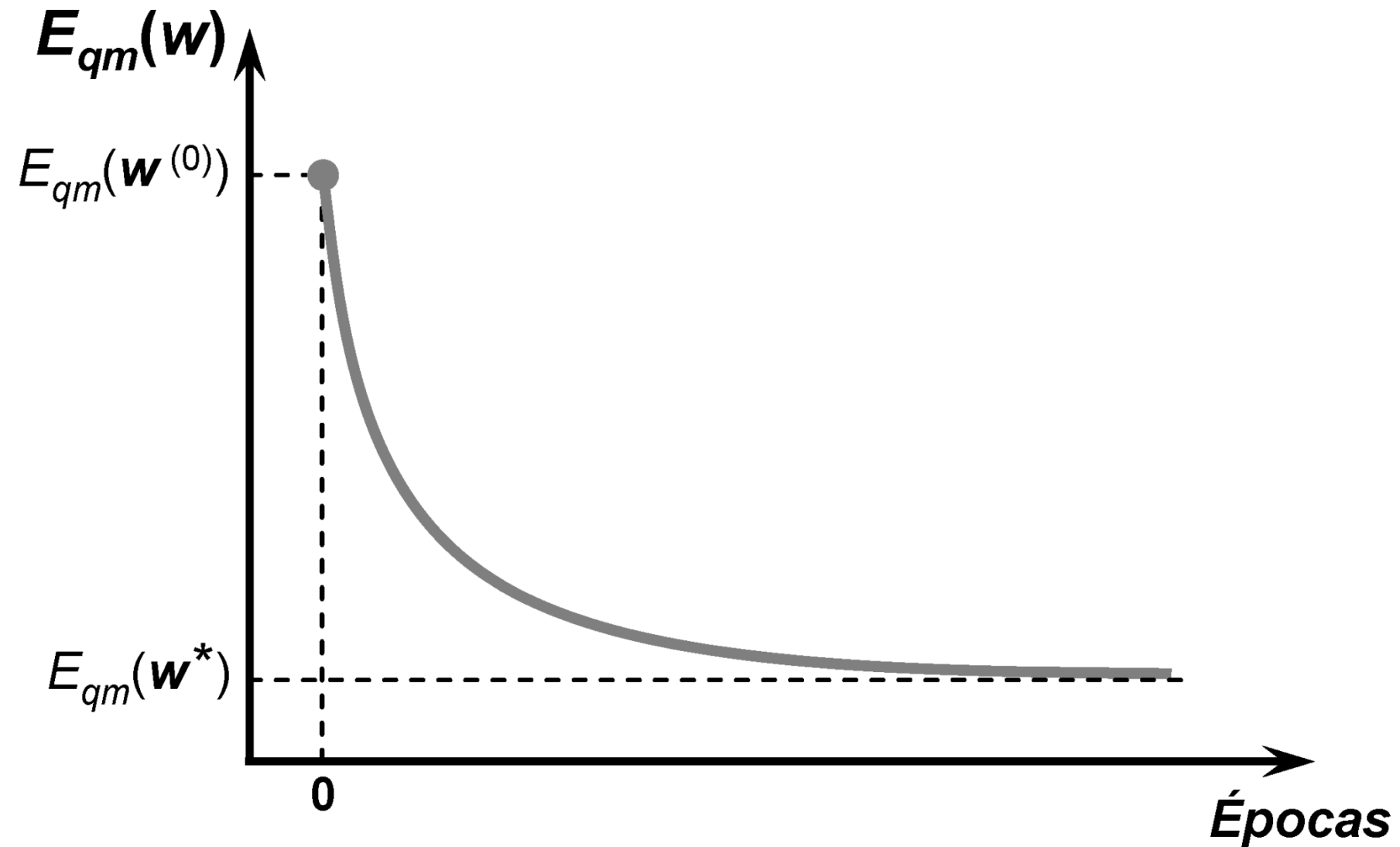
$$E_{qm}(w) = \frac{1}{p} \sum_{k=1}^p (d^{(k)} - u)^2$$

Início {Algoritmo EQM}

- <1> Obter a quantidade de padrões de treinamento $\{p\}$;
- <2> Iniciar a variável E_{qm} com valor zero $\{E_{qm} \leftarrow 0\}$;
- <3> Para todas as amostras de treinamento $\{\mathbf{x}^{(k)}, d^{(k)}\}$, fazer:
 - <3.1> $u \leftarrow \mathbf{w}^T \cdot \mathbf{x}^{(k)}$;
 - <3.2> $E_{qm} \leftarrow E_{qm} + (d^{(k)} - u)^2$;
- <4> $E_{qm} \leftarrow \frac{E_{qm}}{p}$;

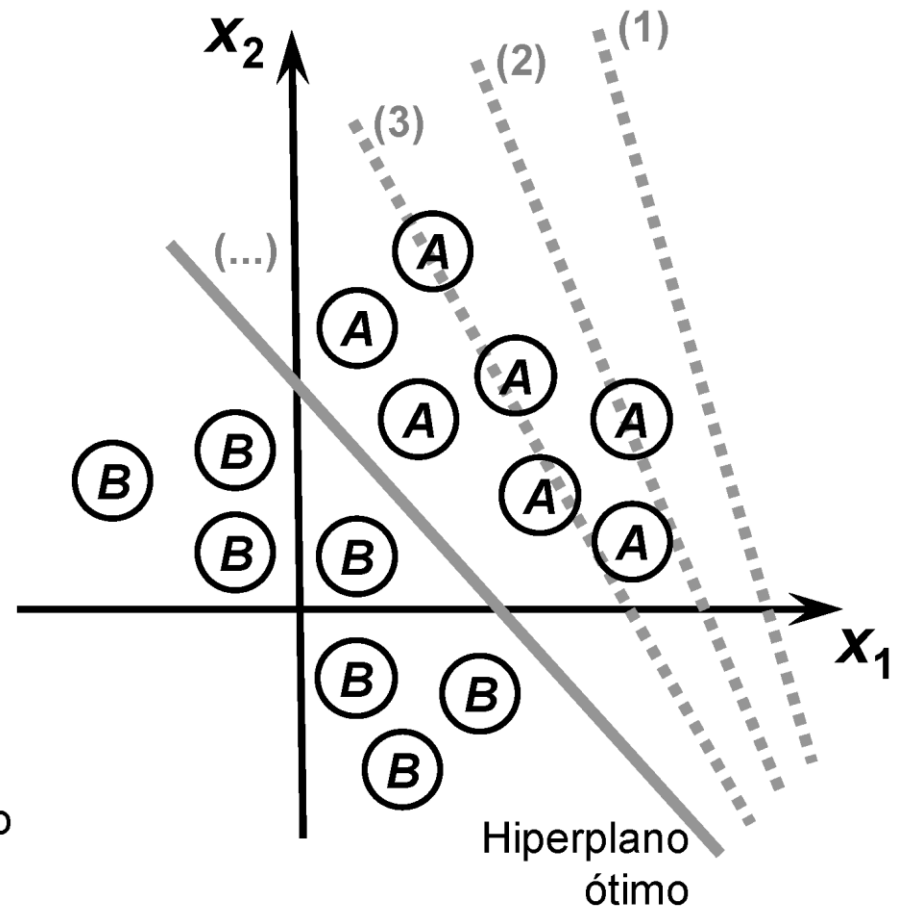
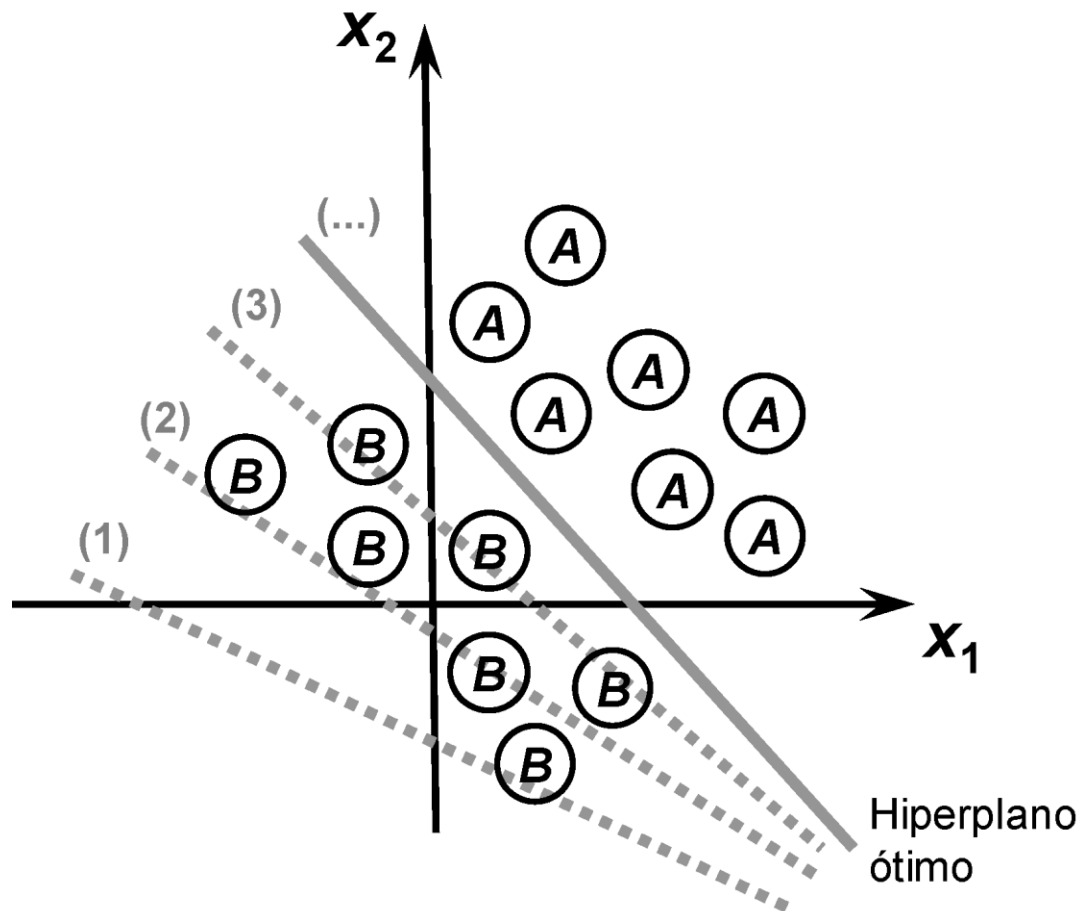
Fim {Algoritmo EQM}

Rede Neural Adaline



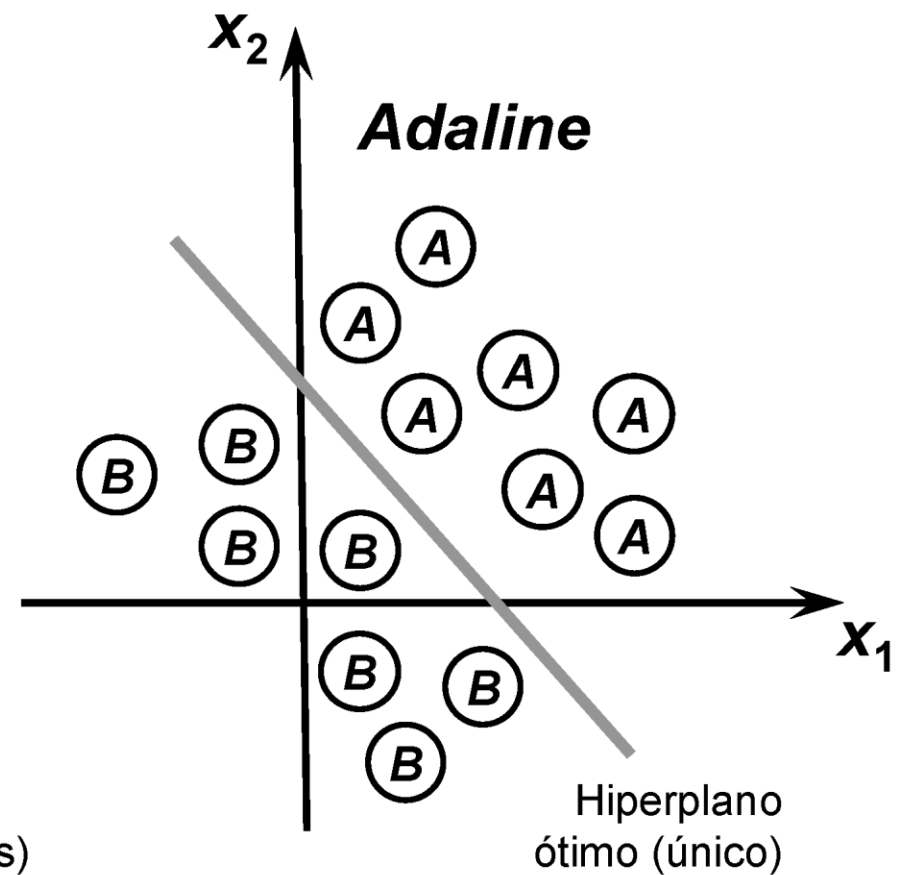
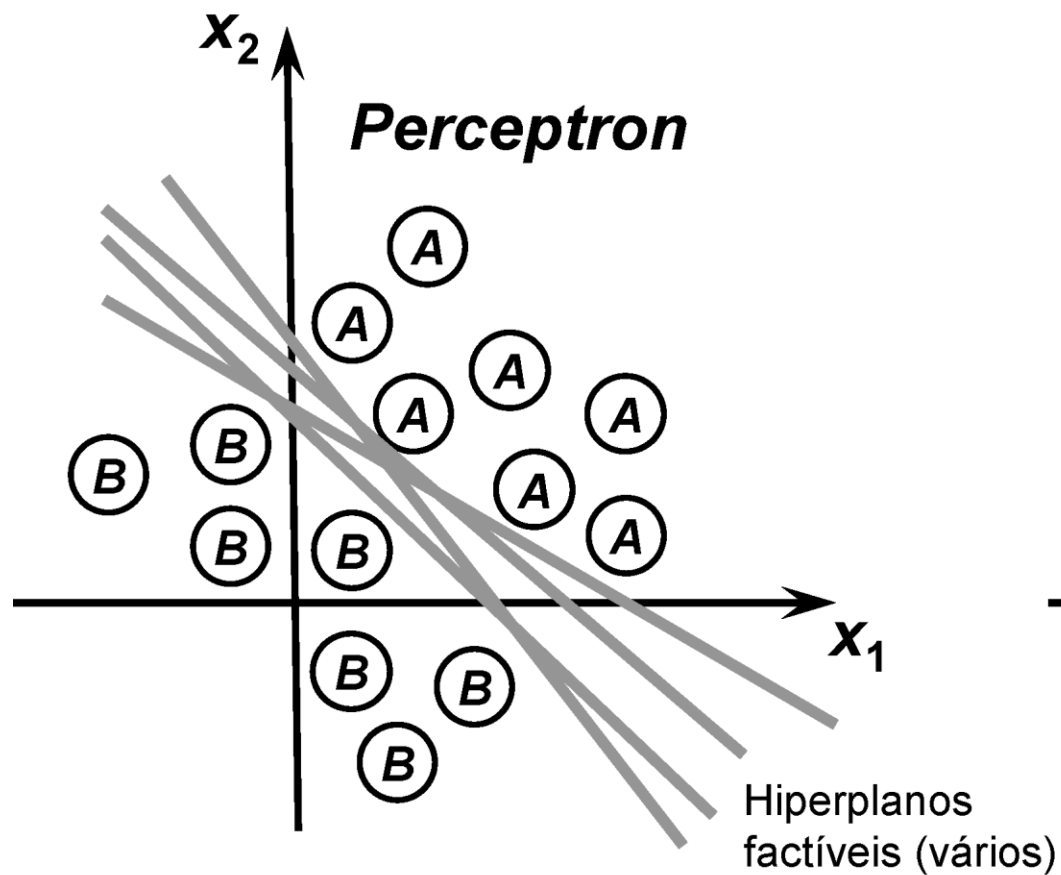
Rede Neural Adaline

- Épocas de treinamento para encontrar o melhor Hiperplano.



Rede Neural Adaline

► Comparação entre a Perceptron e Adaline





Rede Neural Adaline

Início {Algoritmo *Adaline* – Fase de Treinamento}

- <1> Obter o conjunto de amostras de treinamento $\{ \mathbf{x}^{(k)} \}$;
 - <2> Associar a saída desejada $\{ d^{(k)} \}$ para cada amostra obtida;
 - <3> Iniciar o vetor \mathbf{w} com valores aleatórios pequenos;
 - <4> Especificar taxa de aprendizagem $\{ \eta \}$ e precisão requerida $\{ \varepsilon \}$;
 - <5> Iniciar o contador de número de épocas $\{ \text{época} \leftarrow 0 \}$;
 - <6> Repetir as instruções:
 - <6.1> $E_{qm}^{anterior} \leftarrow E_{qm}(\mathbf{w})$;
 - <6.2> Para todas as amostras de treinamento $\{ \mathbf{x}^{(k)}, d^{(k)} \}$, fazer:
 - <6.2.1> $u \leftarrow \mathbf{w}^T \cdot \mathbf{x}^{(k)}$;
 - <6.2.2> $\mathbf{w} \leftarrow \mathbf{w} + \eta \cdot (d^{(k)} - u) \cdot \mathbf{x}^{(k)}$;
 - <6.3> $\text{época} \leftarrow \text{época} + 1$;
 - <6.4> $E_{qm}^{atual} \leftarrow E_{qm}(\mathbf{w})$;
- Até que: $|E_{qm}^{atual} - E_{qm}^{anterior}| \leq \varepsilon$

Fim {Algoritmo *Adaline* – Fase de Treinamento}

Início {Algoritmo *Adaline* – Fase de Operação}

- <1> Obter uma amostra a ser classificada $\{ \mathbf{x} \}$;
- <2> Utilizar o vetor \mathbf{w} ajustado durante o treinamento;
- <3> Executar as seguintes instruções:
 - <3.1> $u \leftarrow \mathbf{w}^T \cdot \mathbf{x}$;
 - <3.2> $y \leftarrow \text{sinal}(u)$;
 - <3.3> Se $y = -1$
 - <3.3.1> Então: amostra $\mathbf{x} \in \{\text{Classe } A\}$
 - <3.4> Se $y = 1$
 - <3.4.1> Então: amostra $\mathbf{x} \in \{\text{Classe } B\}$

Fim {Algoritmo *Adaline* – Fase de Operação}



Dúvidas?



Referências Bibliográficas

- José Demísio Simões da Silva – Notas de Aula
- Ana Paula A. C. Shiguemori – Notas de Aula
- Elcio Hideiti Shiguemori – Notas de Aula