Aula 4
RNA – Adaline e Regra do Delta

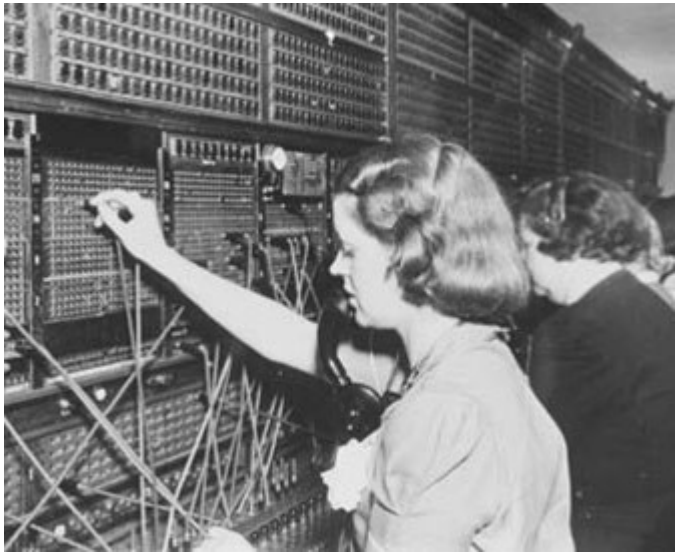
barbon@uel.br

Sumário

1- Introdução

-O ***Adaline*** foi desenvolvido em 1960 e sua principal aplicação era o chaveamento de circuitos telefônicos.

- As principais contribuições do modelo são:
  1) Desenvolvimento do algoritmo de aprendizado “Regra Delta”;
  2) Aplicações de RNAs de forma prática para soluções de sinais analógicos;
  3) Foi a primeira rede neural aplicada na indústria.

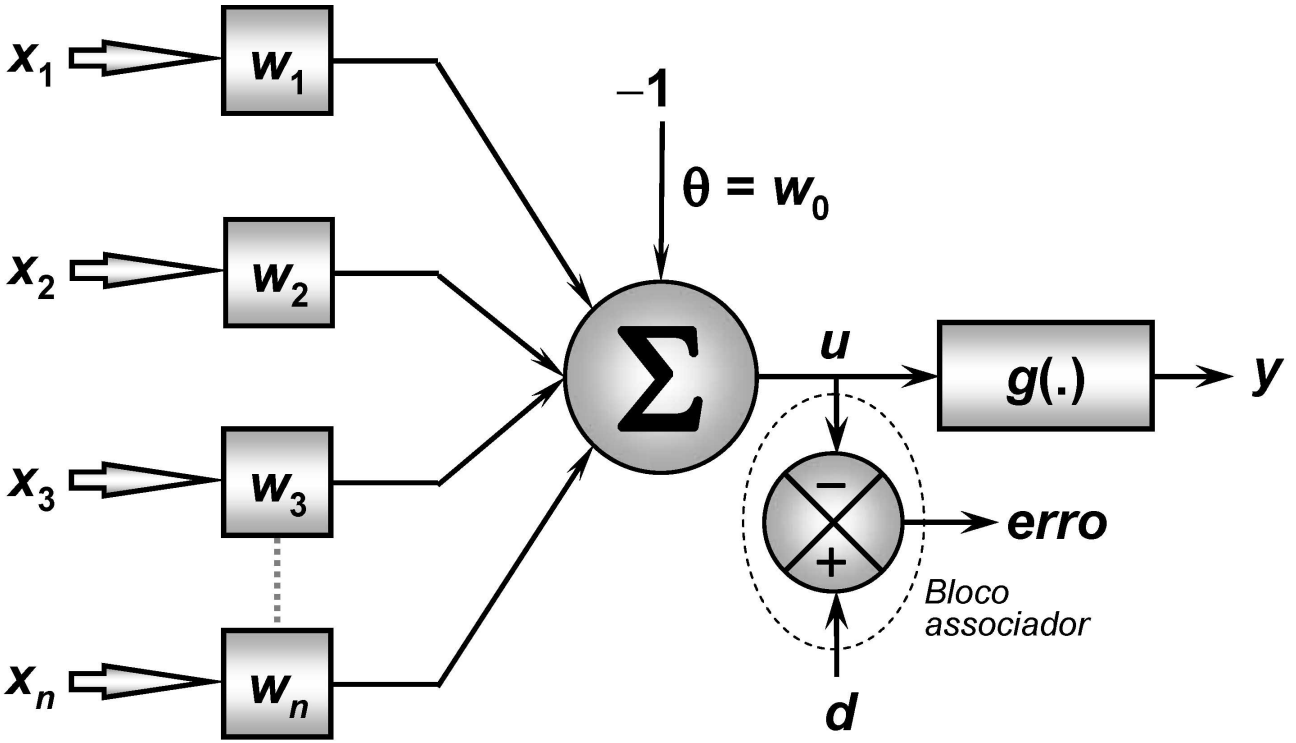


1- Introdução

- Semelhante a ***Perceptron***;

- Vários ***Adalines*** formam um ***Madaline***;

-É de arquitetura ***feedforward***;



2- Funcionamento do Adaline

- Uma das diferenças, quando comparado ao modelo Perceptron é o bloco de verificação do Erro:

$$\boldsymbol{erro = d - u}$$

- O erro (diferença entre potencial de ativação {***u***} e valor desejado {***d***}) será utilizado para realizar ajustes para os pesos $\{w_0, w_1, w_2, \ldots, w_n\}$.

-A teoria matemática sobre convergência do aprendizado da ***Perceptron*** tem o mesmo principio do ***Adaline.***

-**Treinamento:**

- É baseado na Regra Delta, conhecido também como ***LMS*** (least mean square), Gradiente Descendente ou Mínimos Quadrados.

- Assumindo ***p*** amostrar para treinamento, a ideia é encontrar os pesos e limiar que minimizam a diferença entre ***{d}*** e ***{u}***. A função de erro quadrático médio, com relação a ***p*** é definido por:

2- Funcionamento do Adaline

- O peso ótimo (***w****) é obtido:

$$E(w^*) \leq E(w), para \forall w \in \Re^{n+1}$$

- A função de erro quadrático em relação à ***p*** é definida por:

$$E(w) = \frac{1}{2}\sum_{k=1}^{p}(d^{(k)} - (w^T x^{(k)} - \theta))^2$$

- Alternativamente, o valor do $\boldsymbol{w}^{atual}$ pode ser expresso por:

$$w^{atual} = w^{anterior} + \eta\sum_{k=1}^{p}(d^{(k)} - u)x^{(k)}$$

barbon@uel.br

2- Funcionamento do Adaline

- Alternativamente, tem-se:

$$w^{atual} = w^{anterior} + \eta.(d^{(k)} - u).x^{(k)} \qquad k=1,...,p$$

Sendo:

$\boldsymbol{w}$: $[\theta\ w_1\ w_2\ \ldots\ w_n]^T$ é o vetor contendo limiar e pesos

$\boldsymbol{x^{(k)}}$:$[\ -1\ \ x_1^{(k)}\ \ x_2^{(k)}\ \ldots\ \ x_n^{(k)}]^T$ k-ésima amostra de treinamento.

$\boldsymbol{d^{(k)}}$: é o valor desejado para a k-ésima amostra de treinamento.

$\boldsymbol{u}$: é o valor de saída do combinador linear.

$\boldsymbol{\eta}$: é a constante que define a taxa de aprendizado da rede

2- Funcionamento do Adaline

- A função do erro quadrático médio é definido por:

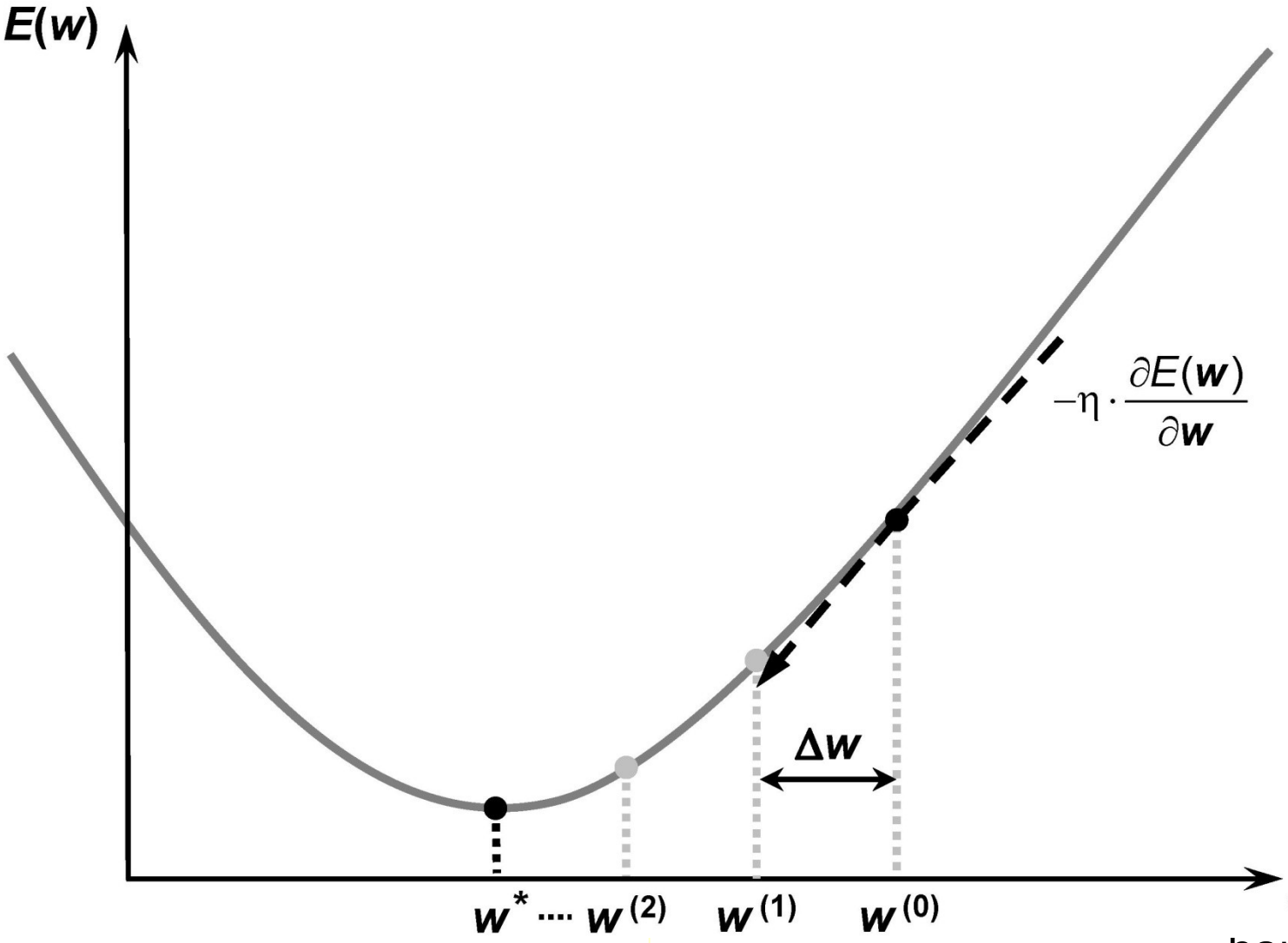
$$E_{qm}(w) = \frac{1}{p}\sum_{k=1}^{p}(d^{(k)} - u)^2$$

- O algoritmo converge quando a diferença entre duas épocas for menor que a precisão imposta (ε).

- $|E_{qm}(w^{atual}) - E_{qm}(w^{anterior})| \leq \varepsilon$

3- Regra Delta

- Interpretação Geométrica da Regra Delta da evolução rumo ao ponto ótimo ***w****



4- Algoritmo Fase de Treinamento

Início {Algoritmo *Adaline* – Fase de Treinamento}

<1> Obter o conjunto de amostras de treinamento { $\boldsymbol{x}^{(k)}$ };

<2> Associar a saída desejada { $d^{(k)}$ } para cada amostra obtida;

<3> Iniciar o vetor $\boldsymbol{w}$ com valores aleatórios pequenos;

<4> Especificar taxa de aprendizagem { η } e precisão requerida { ε };

<5> Iniciar o contador de número de épocas { *época* $\leftarrow 0$ };

<6> Repetir as instruções:

- <6.1> $E_{qm}^{anterior} \leftarrow E_{qm}(\boldsymbol{w})$;
- <6.2> Para todas as amostras de treinamento { $\boldsymbol{x}^{(k)}$, $d^{(k)}$ }, fazer:
  - <6.2.1> $u \leftarrow \boldsymbol{w}^T \cdot \boldsymbol{x}^{(k)}$;
  - <6.2.2> $\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta \cdot (d^{(k)} - u) \cdot \boldsymbol{x}^{(k)}$;
- <6.3> *época* $\leftarrow$ *época* + 1;
- <6.4> $E_{qm}^{atual} \leftarrow E_{qm}(\boldsymbol{w})$;

Até que: $\left| E_{qm}^{atual} - E_{qm}^{anterior} \right| \leq \varepsilon$

Fim {Algoritmo *Adaline* – Fase de Treinamento}

5- Algoritmo Erro Quadrático Médio

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 { $\boldsymbol{x}^{(k)}$, $d^{(k)}$ }, fazer:

<3.1> $u \leftarrow \boldsymbol{w}^T \cdot \boldsymbol{x}^{(k)}$;

<3.2> $E_{qm} \leftarrow E_{qm} + (d^{(k)} - u)^2$;

<4> $E_{qm} \leftarrow \dfrac{E_{qm}}{p}$;

Fim {Algoritmo EQM}

6- Algoritmo Aplicação Adaline

Início {Algoritmo *Adaline* – Fase de Operação}

<1> Obter uma amostra a ser classificada { $\boldsymbol{x}$ };

<2> Utilizar o vetor $\boldsymbol{w}$ ajustado durante o treinamento;

<3> Executar as seguintes instruções:

<3.1> $u \leftarrow \boldsymbol{w}^T \cdot \boldsymbol{x}$;

<3.2> $y \leftarrow \text{sinal}(u)$;

<3.3> Se $y = -1$

<3.3.1> Então: amostra $\boldsymbol{x} \in$ *{Classe A}*

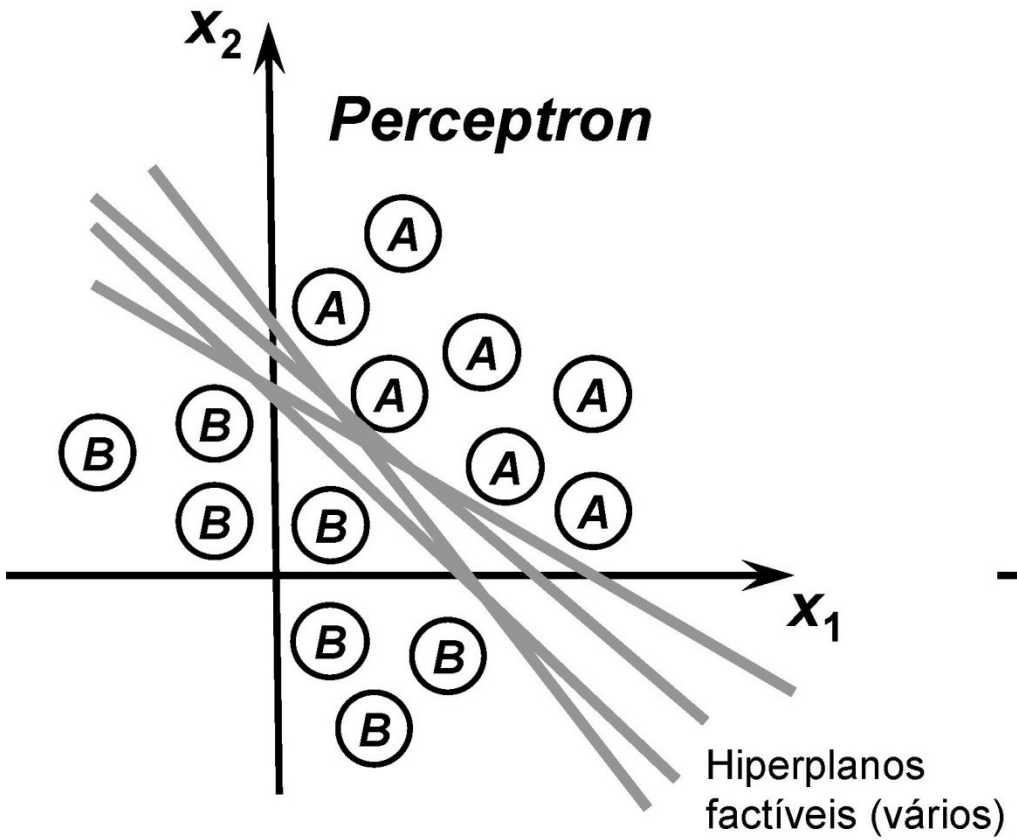
<3.4> Se $y = 1$

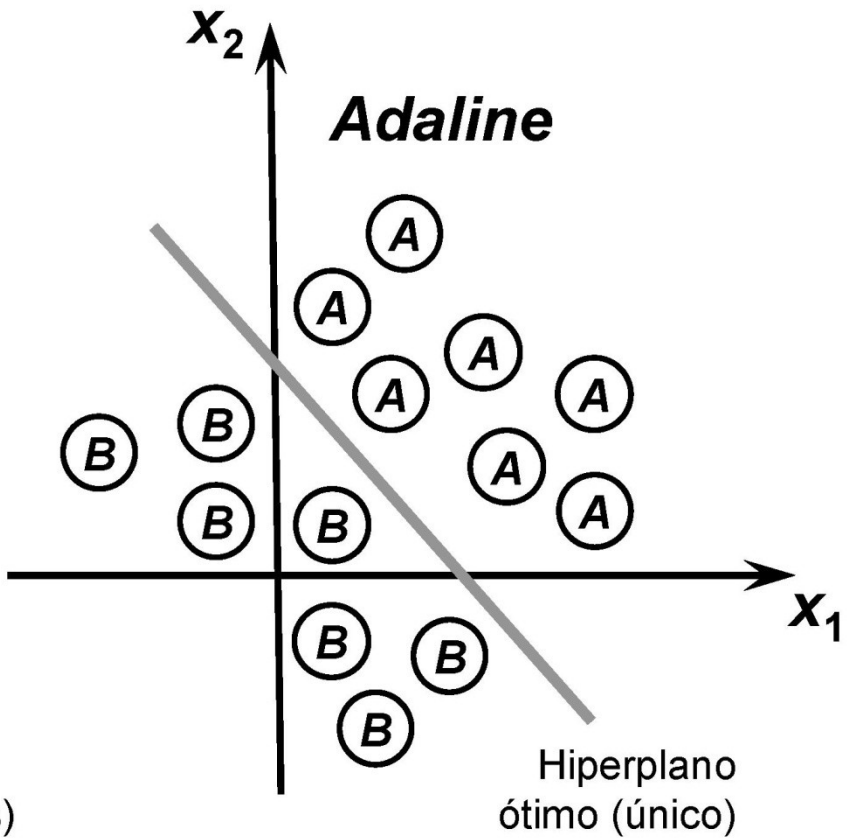
<3.4.1> Então: amostra $\boldsymbol{x} \in$ *{Classe B}*

Fim {Algoritmo *Adaline* – Fase de Operação}

7-Comparação Perceptron e Adaline

- Adaline se torna “imune” à ruídos.





Referências:

Silva, IN da, Danilo Hernane Spatti, and Rogério Andrade Flauzino. "Redes neurais artificiais para engenharia e ciências aplicadas." São Paulo: Artliber (2010).