



Algoritmo de treinamento do Perceptron

```
Inicializar o vetor de pesos com valores aleatórios;  
Inicializar a taxa de aprendizado;  
Repita  
    Erro ← “não existe”;  
    Para cada par de treinamento {x(k),d(k)} faça  
         $u \leftarrow x(k)^T \cdot w$ ;  
         $y \leftarrow \text{Sinal}(u)$ ;  
        Se  $(d(k) \neq y)$  então  
             $w_i \leftarrow w_{i-1} + \eta \cdot (d(k) - y) \cdot x(k)$   
            Erro ← “existe”;  
        fim_se;  
    fim_para;  
Até Erro = “não existe”;
```

Algoritmo de treinamento do Adaline

```
Inicializar o vetor de pesos com valores aleatórios;  
Inicializar a taxa de aprendizado;  
EQM_ant ← INF;  
EQM_atual ← 1;  
Enquanto  $|EQM\_atual - EQM\_ant| > \text{Erro}$   
    EQM_ant ← EQM_atual;  
    Para cada par de treinamento {x(k),d(k)} Faça  
         $u \leftarrow x(k)^T * w$ ;  
         $w_i \leftarrow w_{i-1} + \eta \cdot (d(k) - u) \cdot x(k)$ ;  
    Fim_para;  
    EQM_atual ← EQM;  
Fim_enquanto;
```

Algoritmo de teste do Perceptron e Adaline

```
Apresentar padrão 'x' a ser reconhecido;  
 $u \leftarrow x(k)^T * w$ ;  
 $y \leftarrow \text{Sinal}(u)$ ;  
Se  $(y = 1)$  então  
     $x \in \text{“Classe 1”}$ ;  
Senão  
     $x \in \text{“Classe 2”}$ ;  
Fim_se;
```

Algoritmo do Erro Quadrático Médio

```
EQM  $\leftarrow$  0;  
Para cada par de treinamento  $\{x(k), d(k)\}$  faça  
     $u \leftarrow x(k)^T \cdot w$ ;  
    EQM  $\leftarrow$  EQM +  $(d(k) - u)^2$ ;  
fim_para;  
EQM  $\leftarrow$  EQM / k;
```