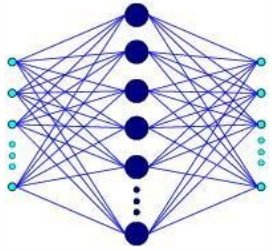
Ciência da Computação

REDE NEURAIS

Semestre: 2010/1

AULA 06



Max Pereira

http://paginas.unisul.br/max.pereira

UNISUL

Aqui seu futuro acontece

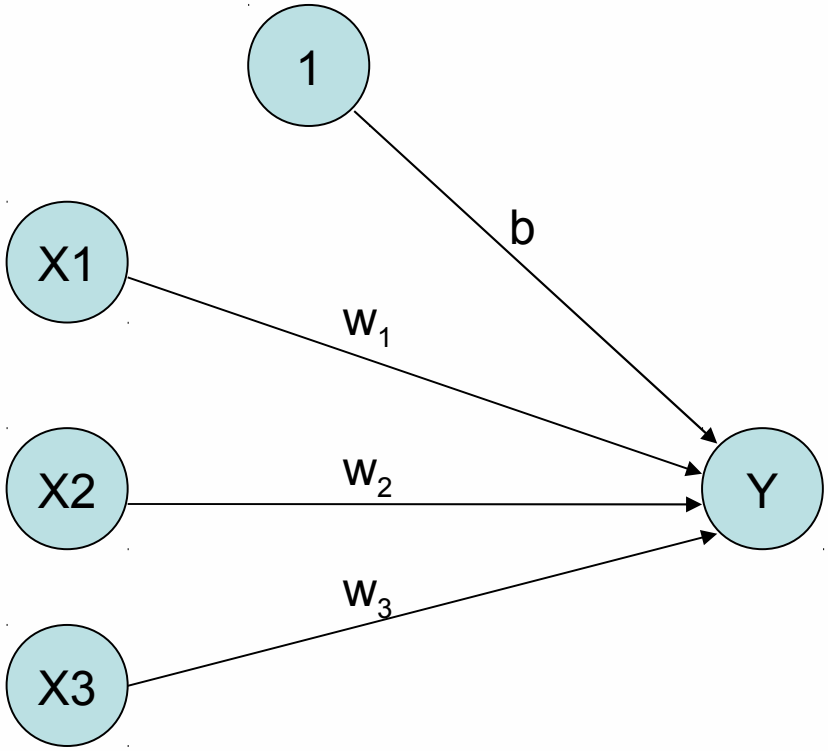
Conteúdo

- Redes Adaline e Madaline
- Exercícios

ADALINE

- Bernard Widrow (1962) na Universidade de Stanford.
- ADAptive LINear Element, mais tarde, ADAptive LInear NEuron.
- A simplicidade do modelo restringe sua relevância ao contexto acadêmico.
- A contribuição importante é o princípio de treinamento conhecido como *Regra Delta*.

ADALINE
Arquitetura



Algoritmo

Passo 1. Inicializar pesos (valores randômicos pequenos são utilizados)
(setar o *bias* igual a zero e definir a taxa de aprendizado α)
Passo 2. Enquanto a condição de parada for falsa, faça os passos 3 até 7.

Passo 3. Para cada par de treinamento *s:t*, faça os passos 4 até 6

Passo 4. Unidades de entrada (valores)

Passo 5. Calcular o valor da unidade de saída

$$rede = b + \sum_i x_i w_i$$

Passo 6. Ajustar pesos e *bias*

$$w_i(new) = w_i(old) + \alpha(t - rede)x_i$$

$$b(new) = b(old) + \alpha(t - rede)$$

Passo 7. Testar condição de parada

Se a maior mudança ocorrida no passo 2 é menor que uma tolerância especificada, então para; caso contrário continua

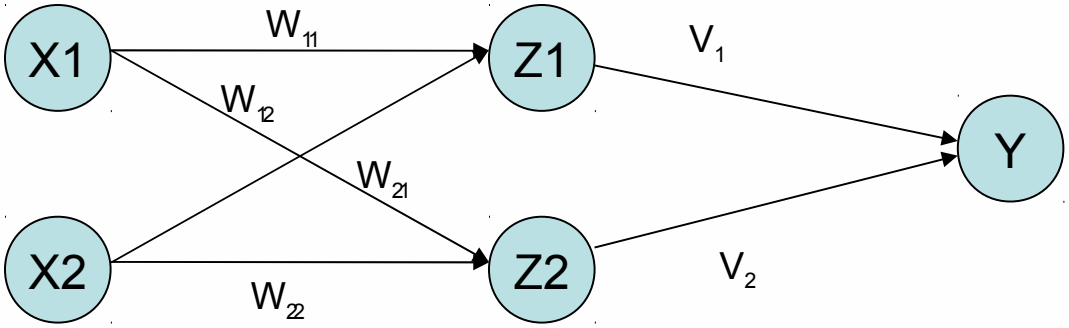
Regra Delta

- A regra delta para ajustar o I-ésimo peso (para cada padrão) é: $\Delta w_i = \alpha(t - y\_in)x_i$

- A regra delta para ajustar o peso da I-ésima unidade de entrada com a J-ésima unidade de saída (para cada padrão) é: $\Delta w_{ij} = \alpha(t_j - y\_in_j)x_i$

MADALINE

- Consiste em uma rede Adaline com múltiplas camadas (Many ADAptive LInear NEurons).



Algoritmo

Training Algorithm for MADALINE (MRI). The activation function for units Z_1, Z_2, and Y is

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0; \\ -1 & \text{if } x < 0. \end{cases}$$

Step 0. Initialize weights:
Weights v_1 and v_2 and the bias b_3 are set as described;
small random values are usually used for ADALINE weights.
Set the learning rate α as in the ADALINE training algorithm (a small value).

Step 1. While stopping condition is false, do Steps 2–8.

Step 2. For each bipolar training pair, **s**:**t**, do Steps 3–7.

Step 3. Set activations of input units:

$$x_i = s_i.$$

Step 4. Compute net input to each hidden ADALINE unit:

$$z\_in_1 = b_1 + x_1 w_{11} + x_2 w_{21},$$

$$z\_in_2 = b_2 + x_1 w_{12} + x_2 w_{22}.$$

Algoritmo

Step 5. Determine output of each hidden ADALINE unit:

$$z_1 = f(z\_in_1),$$

$$z_2 = f(z\_in_2).$$

Step 6. Determine output of net:

$$y\_in = b_3 + z_1 v_1 + z_2 v_2;$$

$$y = f(y\_in).$$

Step 7. Determine error and update weights:
If $t = y$, no weight updates are performed.
Otherwise:
If $t = 1$, then update weights on Z_J, the unit whose net input is closest to 0,

$$b_J(\text{new}) = b_J(\text{old}) + \alpha(1 - z\_in_J),$$

$$w_{iJ}(\text{new}) = w_{iJ}(\text{old}) + \alpha(1 - z\_in_J)x_i;$$

If $t = -1$, then update weights on all units Z_k that have positive net input,

$$b_k(\text{new}) = b_k(\text{old}) + \alpha(-1 - z\_in_k),$$

$$w_{ik}(\text{new}) = w_{ik}(\text{old}) + \alpha(-1 - z\_in_k)x_i.$$

Algoritmo

Step 8. Test stopping condition.
If weight changes have stopped (or reached an acceptable level), or if a specified maximum number of weight update iterations (Step 2) have been performed, then stop; otherwise continue.

Exercícios

Treinar uma rede Madaline para a função lógica XOR, com valores bipolares.

x_1	x_2	t
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Os pesos devem ser inicializados como descritos abaixo, a taxa de aprendizado α é .5

W11	w21	w12	w22	v1	v2
.05	.2	.1	.2	.5	.5

- Apresentar os valores dos pesos após 4 épocas.