International University for Science & Technology (IUST)
Department of Computer &Informatics Engineering
Neural Networks unit (4)

Adaline and MAdaline

Neural Networks

Dr. Ali Mayya

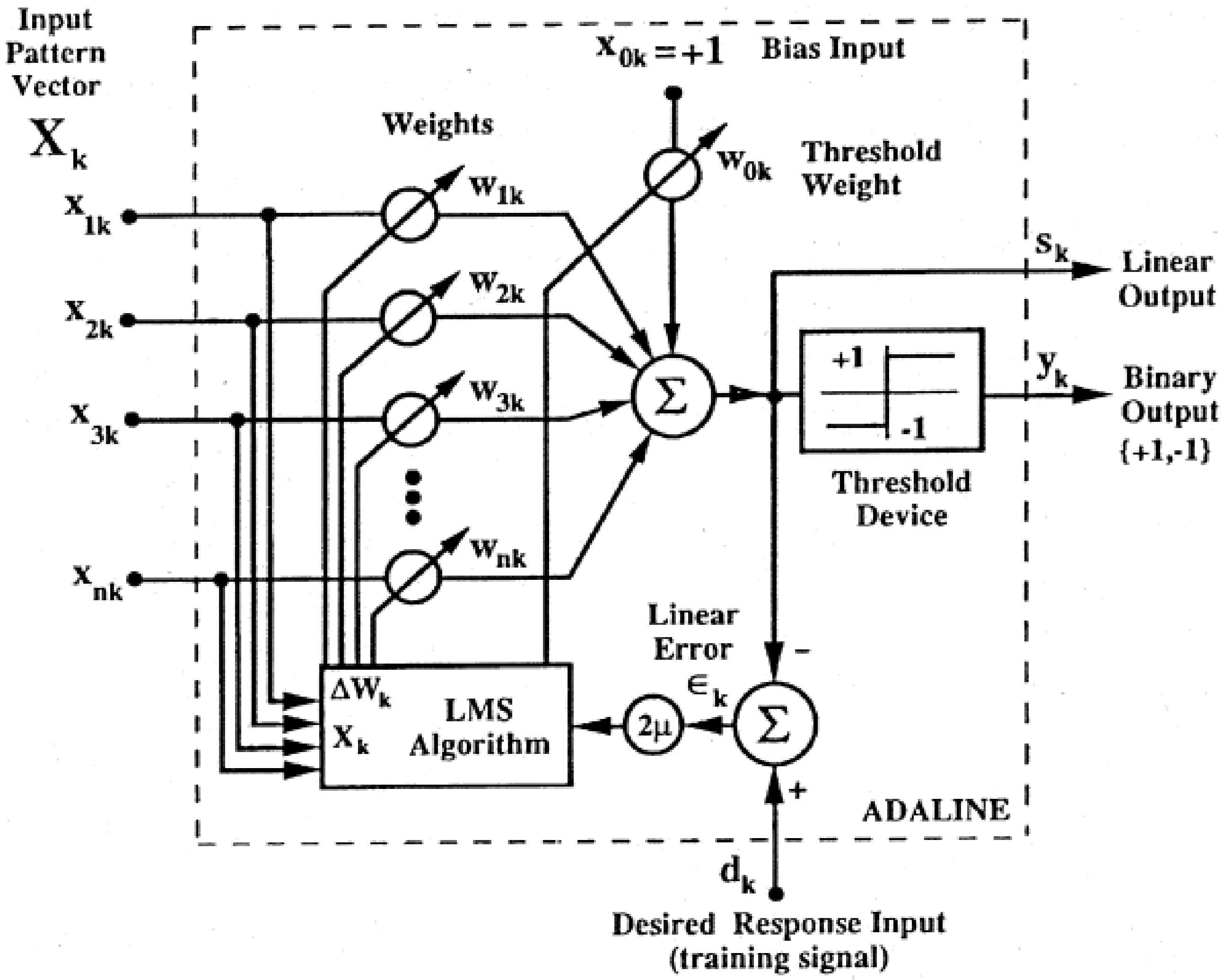
Adaline Architecture

By Widrow and Hoff (1960)

- **Ada**ptive **Line**ar Neuron for signal processing
- The same architecture of simple network
- Learning method: **delta rule** (another way of error driven), also called Widrow-Hoff learning rule
- The delta: ***t – y_in***
  - NOT ***t – y*** because ***y = f(y_in)*** is not differentiable
- Learning algorithm: same as Perceptron learning except in Step 5:

  b := b + α * (t – y_in)

  wi := wi + α * xi * (t – y_in)



Adaline Architecture –Learning Steps- Method1 (Sequential mode)

Sequential model: Update weights after each training pattern (as in Perceptron)

Step 1: Initialize weight not zero but small random values are used. Set learning rate α.

Step 2: While the stopping condition is False do steps 3 to 7.

Step 3: For each training set perform steps 4 to 6.

Step 4: Set xi for (i=1 to n).

Step 5: Compute net input to output unit:

$$y_{in} = \sum w_i x_i + b$$

Here, b is the bias and n is the total number of neurons.

Step 6: Update the weights and bias for i=1 to n ***(Delta rule)***

$$w_i(new) = w_i(old) + \alpha\,(t - y_{in})x_i$$
$$b_i(new) = b_i(old) + \alpha\,(t - y_{in})$$

note: $\mathbf{y_{in}}$ is denoted as **net** in previous lectures

And calculate the error: $(t - y_{in})^2$

Step7: stop training when the stopping condition is satisfied (for example, reaching the required number of epochs, or getting an accumulative error value less than the specified tolerance value).

Accumulative error formula is the mean squared error (***MSE***) : $\mathrm{MSE} = \frac{1}{P}\sum_{i=1}^{P}(t - y_{in})^2$

Adaline Architecture –Learning Steps- Method2 (Batch mode)

change wi at the end of each epoch. Within an epoch, cumulate $\alpha(t(p)-y\_in(p))x_i$ for every pattern **(x(p), t(p))**

Method 2 is slower but may provide slightly better results (because Method 1 may be sensitive to the sample ordering)

Notes:

E monotonically decreases until the system reaches a state with (local) minimum E (a small change of any wi will cause E to increase).

At a local minimum E state, $\partial E / \partial w_i = 0 \;\; \forall i$, but E is not guaranteed to be zero

Adaline Architecture –Learning Steps- Method2 (Batch mode)

- **Derivation of the delta rule**
  - Error for all P samples: mean square error

$$E = \frac{1}{P}\sum_{p=1}^{P}(t(p) - y\_in(p))^2$$

    - E is a function of W = {w1, ... wn}
  - Learning takes **gradient descent** approach to reduce E by modify W
    - the gradient of E: $\nabla E = (\frac{\partial E}{\partial w_1}, \ldots\ldots \frac{\partial E}{\partial w_n})$

$$\Delta w_i \propto -\frac{\partial E}{\partial w_i}$$

$$\frac{\partial E}{\partial w_i} = [\frac{2}{P}\sum_{p=1}^{P}(t(p) - y\_in(p))]\frac{\partial}{\partial w_i}(t(p) - y\_in(p)$$

$$= -[\frac{2}{P}\sum_{p=1}^{P}(t(p) - y\_in(p))]x_i$$

    - There for $\Delta w_i \propto -\frac{\partial E}{\partial w_i} = [\frac{2}{P}\sum_{1}^{P}(t(p) - y\_in(p))]x_i$

Adaline Architecture –Learning Steps- Method1 (Example)

OR gate: Consider all weights are initialized to 0.1, and bias weight =0.1, learning rate=0.1.

Tolerance = 0.1

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

1. Compute net (yin)

$y_{in} = \sum w_i x_i + b$ = 1*0.1 + 1*0.1 + 0.1 = 0.3 → e=(t-yin) = 1-0.3=0.7

2. $Update\ Weights$:

$$w_i(new) = w_i(old) + \alpha\,(t - y_{in})x_i$$
$$w_1(new) = 0.1 + 0.1 * 0.7 * 1 = 0.17$$
$$w_2(new) = 0.1 + 0.1 * 0.7 * 1 = 0.17$$
$$b(new) = 0.1 + 0.1 * 0.7 = 0.17$$

3. Compute error²

$(E_{w1})^2 = (t - y_{in})^2 = 0.49$

Round1 (Epoch1) of training

x_1	x_2	t	y_{in}	$(t-y_{in})$	Δw_1	Δw_2	Δb	w_1 (0.1)	w_2 (0.1)	b (0.1)	$error^2$
1	1	1	0.3	0.7	0.07	0.07	0.07	0.17	0.17	0.17	0.49
1	-1	1	0.17	0.83	0.083	-0.083	0.083	0.253	0.087	0.253	0.69
-1	1	1	0.087	0.913	-0.0913	0.0913	0.0913	0.1617	0.1783	0.3443	0.83
-1	-1	-1	0.0043	-1.0043	0.1004	0.1004	-0.1004	0.2621	0.2787	0.2439	1.01

Adaline Architecture –Learning Steps- Method1 (Example)

OR gate: Consider all weights are initialized to 0.1, and bias weight =0.1, learning rate=0.1.

Tolerance = 0.1

x_1	x_2	t	y_{in}	$(t-y_{in})$	Δw_1	Δw_2	Δb	w_1 (0.1)	w_2 (0.1)	b (0.1)	$error^2$
1	1	1	0.3	0.7	0.07	0.07	0.07	0.17	0.17	0.17	0.49
1	-1	1	0.17	0.83	0.083	-0.083	0.083	0.253	0.087	0.253	0.69
-1	1	1	0.087	0.913	-0.0913	0.0913	0.0913	0.1617	0.1783	0.3443	0.83
-1	-1	-1	0.0043	-1.0043	0.1004	0.1004	-0.1004	0.2621	0.2787	0.2439	1.01

$$\mathrm{MSE} = \frac{1}{P}\sum_{i=1}^{P}(t - y_{in})^2 = \frac{1}{4} * (0.49 + 0.69 + 0.83 + 1.01) = 0.755$$

MSE > 0.1 → continue training

Adaline Architecture –Learning Steps- Method1 (Example)

Round 2 (Epoch2)

Input X1	Input x2	Target t	yin	Δw1	Δw2	Δb	Updated w1 (0.26213)	Updated w2 (0.27873)	Updated b (0.24387)	$Error^2$
1	1	1	0.78473	0.021527	0.021527	0.021527	0.283657	0.300257	0.265397	0.04641
1	-1	1	0.248797	0.0751203	-0.0751203	0.0751203	0.3587773	0.2251367	0.3405173	0.56431
-1	1	1	0.2068767	-0.07931233	0.07931233	0.07931233	0.27946497	0.30444903	0.41982963	0.62904
-1	-1	-1	-0.16408437	0.083591563	0.083591563	-0.083591563	0.363056533	0.388040593	0.336238067	0.69876

$$\mathrm{MSE}_{\mathrm{epoch}\ 2} = \frac{1}{4}(0.04635 + 0.56431 + 0.62904 + 0.69876)$$

$$\mathrm{MSE}_{\mathrm{epoch}\ 2} = \frac{1}{4} \times (1.93846) = 0.484615$$

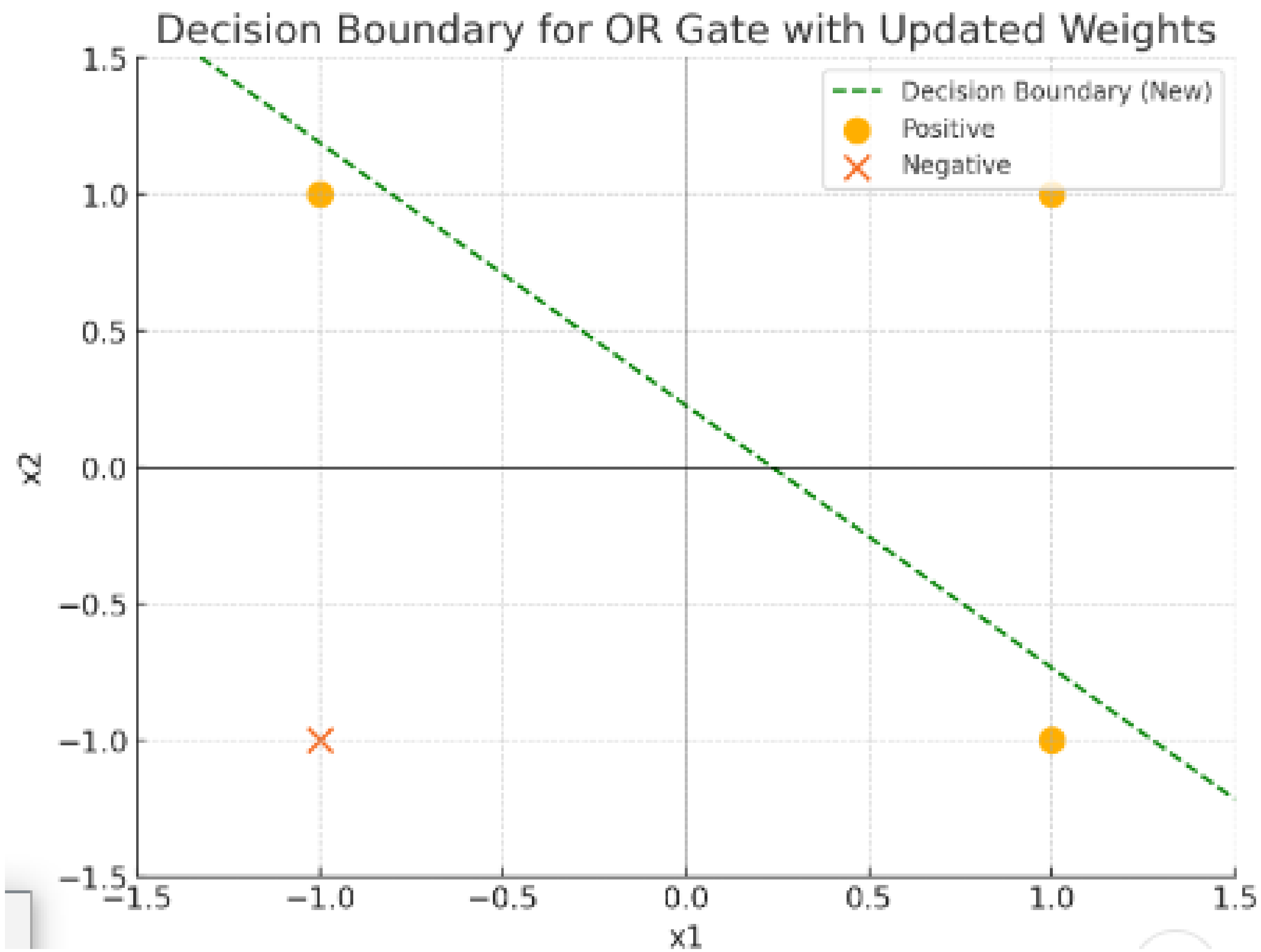
MSE > 0.1 → continue training

Adaline Architecture –Learning Steps- Method1 (Example1)

Let's check the decision boundary with the final weights and biases

W1 (new)	W2 (new)	B (new)
0.51095	0.53995	-0.12095

Decision line equation is:
$w_1*x_1 + w_2*x_2 + b = 0$ →
0.5109 x1 + 0.5399 x2 -0.1209 = 0



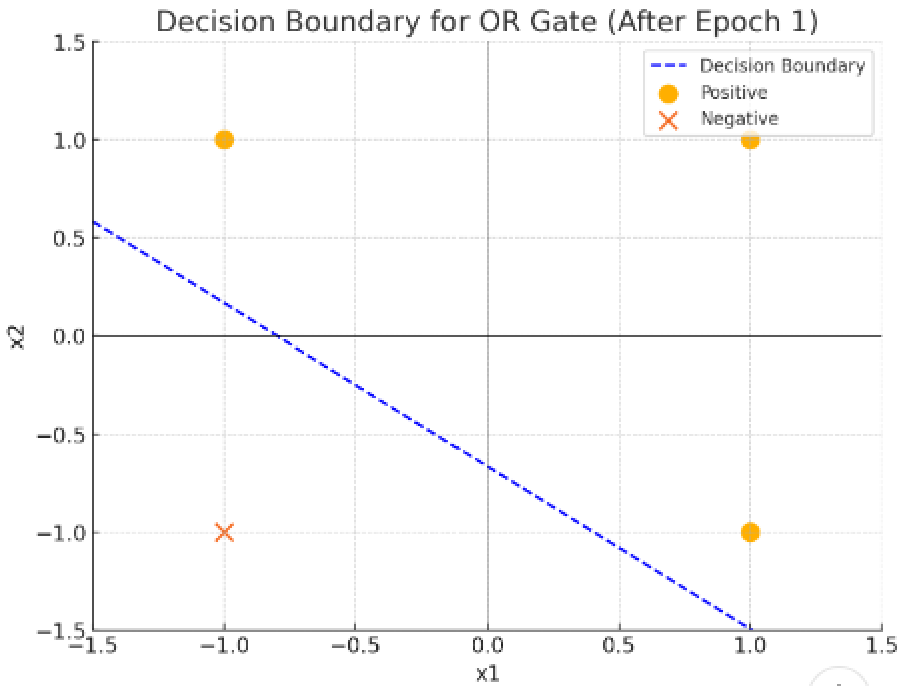
Adaline Architecture –Learning Steps- Method1 (Example2)

OR gate in ADALINE with initial weights of w1=0.2, w2=0.3, Bias=0.1 and learning rate of 0.2

X1	X2	T	Yin	Error	Δw1	Δw2	Δb	W1	W2	b
1	1	1	0.6	0.4	0.08	0.08	0.08	0.28	0.38	0.18
1	-1	1	0.08	0.92	0.184	-0.184	0.184	0.464	0.196	0.364
-1	1	1	0.096	0.904	-0.1808	0.1808	0.1808	0.2832	0.3768	0.5448
-1	-1	-1	-0.1152	-0.8848	0.17696	0.17696	-0.17696	0.46016	0.55376	0.36784

- w1=0.46016
- w2=0.55376
- b=0.36784

$$0 = 0.46016 \cdot x_1 + 0.55376 \cdot x_2 + 0.36784$$



Multiple Adaline (Madaline) Architecture

MADALINE: Multiple adaline neurons

- Multi-layer network consists of many Adaline modules
- Suitable to solve non-linear problems
- Uses the Madaline Rule1 algorithm (MR1) for learning
- Only the weights for the hidden Adalines are adjusted; the weights for the output unit are fixed.

Multiple Adaline (Madaline) MR1 training algorithm

MADALINE: Multiple adaline neurons

Compute net input to each Adaline unit:

$$Z\text{-}in_1 = b_1 + x_1 w_{11} + x_2 w_{21}$$

$$Z\text{-}in_2 = b_2 + x_1 w_{12} + x_2 w_{22}$$

Determine output of each hidden unit:

$$Z_1 = f(Z\text{-}in_1)$$

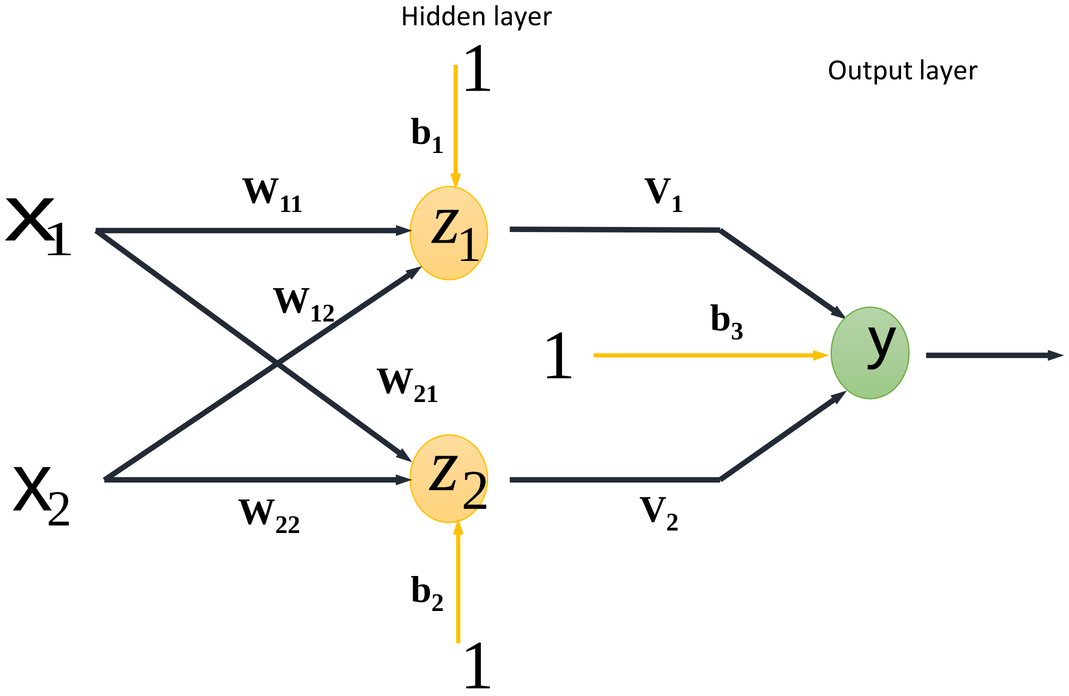
$$Z_2 = f(Z\text{-}in_2)$$

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

Determine output of net:

$$y\text{-}in = b_3 + Z_1 v_1 + Z_2 v_2$$

$$y = f(y\text{-}in)$$



Multiple Adaline (Madaline) MR1 training algorithm

MADALINE: Multiple adaline neurons

Determine error and update weights according to the following:

if $t = y$, no weight updates are performed.

if $t \neq y$, then:

if $t = 1$, then: update weights on Z_j, the unit whose net input is closest to 0

$$b_J(new) = b_J(old) + \alpha(1 - Z\text{-}in_J)$$

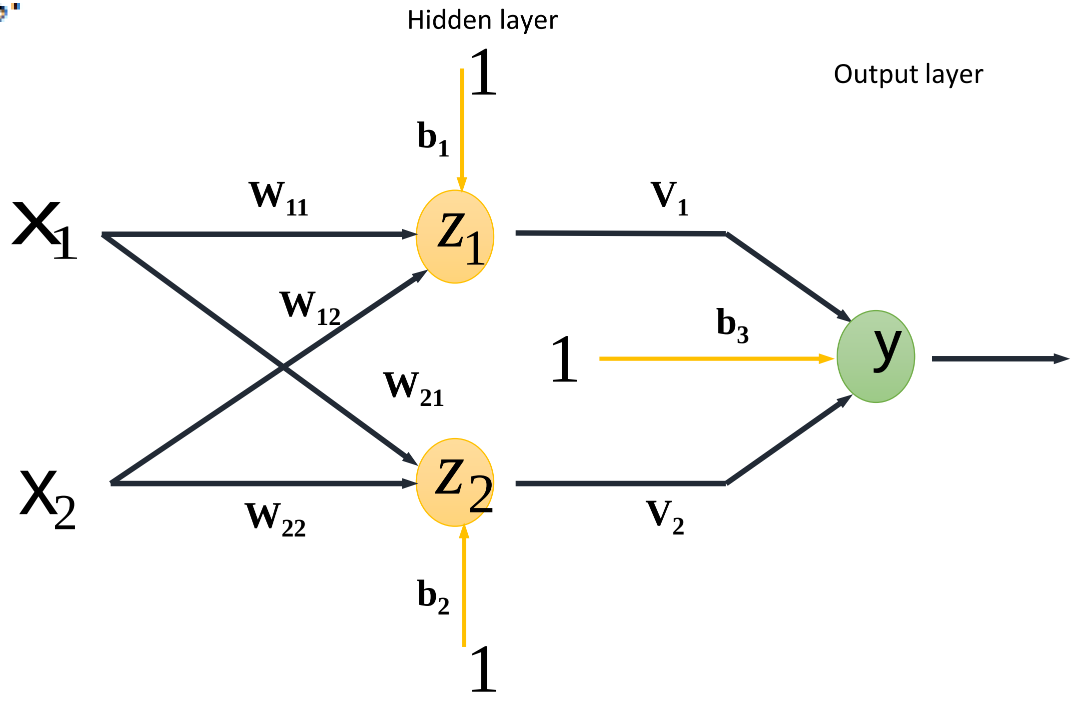
$$w_{iJ}(new) = w_{iJ}(old) + \alpha(1 - Z\text{-}in_J)x_i$$

if $t = -1$, then: update weights on all units Z_k that have positive net input ($Z\text{-}in_k > 0$):

$$b_k(new) = b_k(old) + \alpha(-1 - Z\text{-}in_k)$$

$$w_{ik}(new) = w_{ik}(old) + \alpha(-1 - Z\text{-}in_k)x_i$$

If weight changes have stopped (or reached an acceptable level), , then stop; otherwise continue.



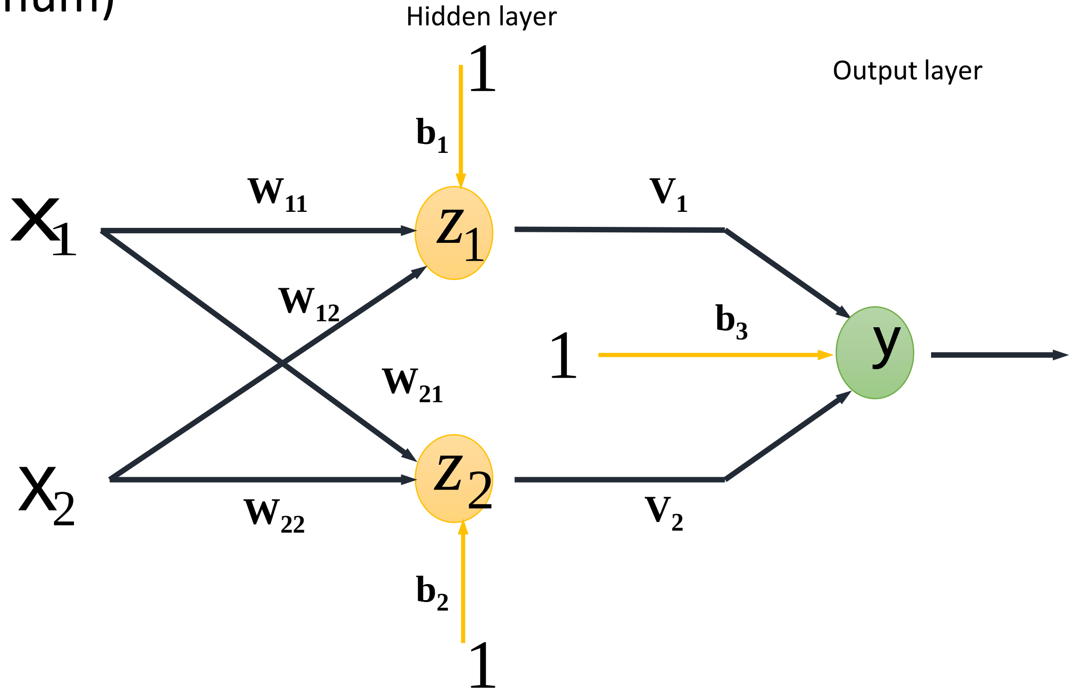
Multiple Adaline (Madaline) Example (XOR)

W11=0.05, W12=0.2, W21=0.1, W22=0.2, b1=0.3, b2=0.15, V1=V2=b3=0.5, learning rate=0.5

X_1	X_2	t
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Note: activation function is hardlims (signum)

$$a = hard\lim s\,(n) = \begin{cases} 1 & \text{if a} \geq 0 \\ -1 & \text{if a} \prec 0 \end{cases}$$



Solution: For first sample (1,1) →

$$Z_1 = \text{hardlims}(Z_{in1}) = \text{hardlims}(w_{11} * x_1 + w_{12} * x_2 + b_1)$$
$$= \text{hardlims}(0.05 + 0.2 + 0.3) = \text{hardlims}(0.55) = 1$$

$$Z_2 = \text{hardlims}(Z_{in2}) = \text{hardlims}(w_{21} * x_1 + w_{22} * x_2 + b_2)$$
$$= \text{hardlims}(0.1 + 0.2 + 0.15) = \text{hardlims}(0.45) = 1$$

$$y = \text{hardlims}(y_{in}) = \text{hardlims}(v_1 * z_1 + v_2 * z_2 + b_3)$$
$$= \text{hardlims}(0.5 + 0.5 + 0.5) = \text{hardlims}(1.5) = 1$$

t$\neq y$ and (t=-1) →

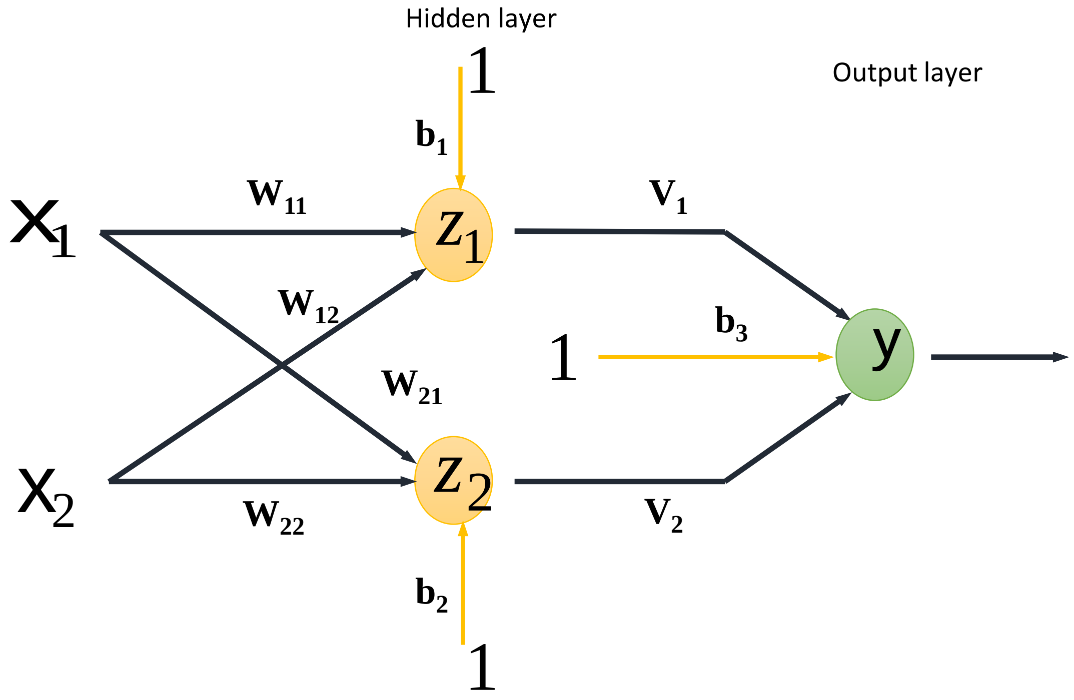
update weights of only positive Zins (Zin1 and Zin2 are positives) → Update W11,W12, W21,W22, b1, b2

Multiple Adaline (Madaline) Example (XOR)

W11=0.05, W12=0.2, W21=0.1, W22=0.2, b1=0.3, b2=0.15, V1=V2=b3=0.5, learning rate=0.5

X_1	X_2	t
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Solution: t$\neq y$ → update weights (t=-1)



$$\Delta w_{11} = lr*(t - z_{in1})*x_1 = 0.5*(-1-0.55)*1 = -0.775$$

$$w_{11}(new) = w_{11}(old) + \Delta w_{11} = 0.05 - 0.775 = -0.725$$

$$\Delta w_{12} = lr*(t - z_{in1})*x_2 = 0.5*(-1-0.55)*1 = -0.775$$

$$w_{12}(new) = w_{12}(old) + \Delta w_{12} = 0.2 - 0.775 = -0.575$$

$$\Delta w_{21} = lr*(t - z_{in2})*x_1 = 0.5*(-1-0.45)*1 = -0.725$$

$$w_{21}(new) = w_{21}(old) + \Delta w_{21} = 0.1 - 0.725 = -0.625$$

$$\Delta w_{22} = lr*(t - z_{in2})*x_2 = 0.5*(-1-0.45)*1 = -0.725$$

$$w_{22}(new) = w_{22}(old) + \Delta w_{22} = 0.2 - 0.725 = -0.525$$

$$\Delta b_1 = lr*(t - z_{in1}) = 0.5*(-1-0.55)*1 = -0.775$$

$$b_1(new) = b_1(old) + \Delta b_1 = 0.3 - 0.775 = -0.475$$

$$\Delta b_2 = lr*(t - z_{in2}) = 0.5*(-1-0.45)*1 = -0.725$$

$$b_2(new) = b_2(old) + \Delta b_2 = 0.15 - 0.725 = -0.575$$

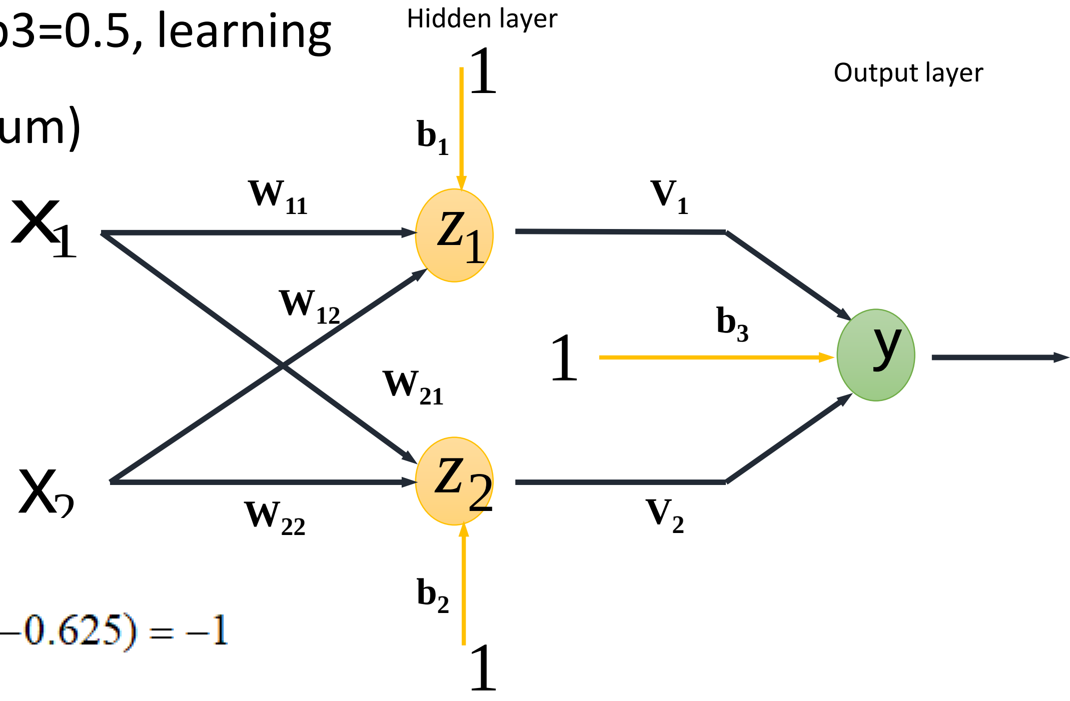
Multiple Adaline (Madaline) Example (XOR)

W11=0.05, W12=0.2, W21=0.1, W22=0.2, b1=0.3, b2=0.15, V1=V2=b3=0.5, learning

X_1	X_2	t
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Note: activation function is hardlims (signum)

$$a = hard\lim s(n) = \begin{cases} 1 & \text{if } a \geq 0 \\ -1 & \text{if } a \prec 0 \end{cases}$$



Solution: For second sample (1,-1) →

$$Z_1 = \text{hardlims}(Z_{in1}) = \text{hardlims}(w_{11(new)} * x_1 + w_{12(new)} * x_2 + b_{1(new)})$$

$$= \text{hardlims}((-0.725*1) + (-0.757*(-1)) - 0.475) = \text{hardlims}(-0.625) = -1$$

$$Z_2 = \text{hardlims}(Z_{in2}) = \text{hardlims}(w_{21(new)} * x_1 + w_{22(new)} * x_2 + b_{2(new)})$$

$$= \text{hardlims}((-0.625*1) + (-0.525*(-1)) - 0.575) = \text{hardlims}(-0.675) = -1$$

$$y = \text{hardlims}(y_{in}) = \text{hardlims}(v_1 * z_1 + v_2 * z_2 + b_3)$$

$$= \text{hardlims}(-0.5 - 0.5 + 0.5) = \text{hardlims}(-0.5) = -1$$

t≠ y and (t=1) →

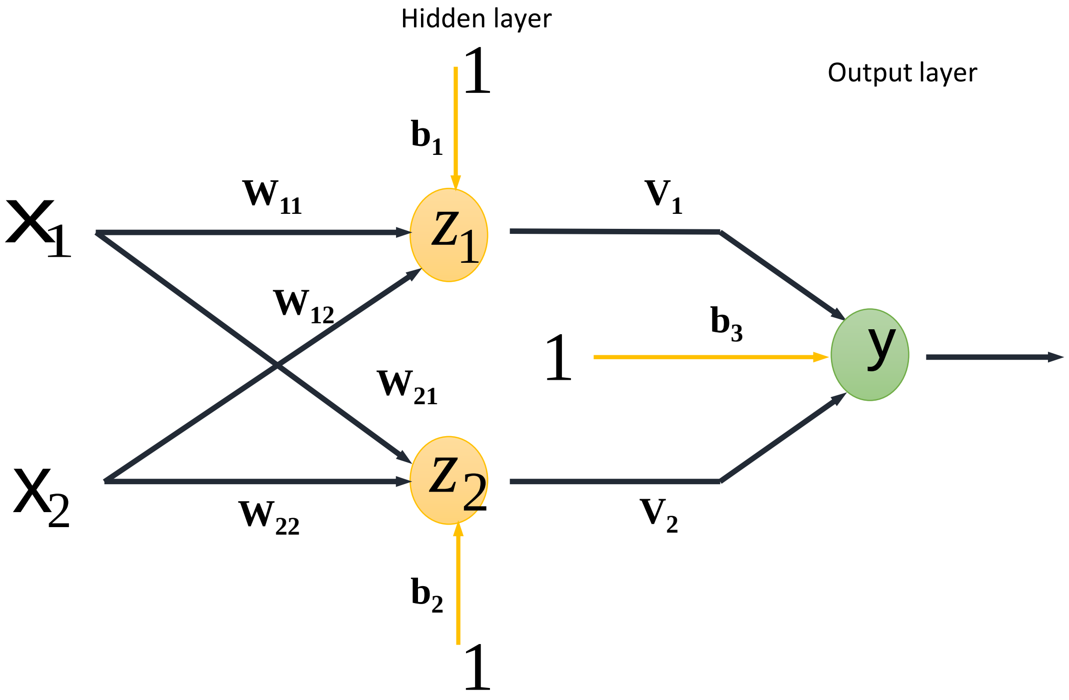
update weights of Z whose Zin is closest to 0 and here Z1 have the lowest absolute zin value closest to 0 → update only W11, W12, b1

Multiple Adaline (Madaline) Example (XOR)

W11=0.05, W12=0.2, W21=0.1, W22=0.2, b1=0.3, b2=0.15, V1=V2=b3=0.5, learning rate=0.5

X_1	X_2	t
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Solution:



$$\Delta w_{11(new)} = lr * (t - z_{in1}) * x_1 = 0.5 * (1 + 0.625) * 1 = 0.8125$$

$$w_{11}(new) = w_{11}(old) + \Delta w_{11} = -0.725 + 0.8125 = 0.0875$$

$$\Delta w_{12} = lr * (t - z_{in1}) * x_2 = 0.5 * (1 + 0.625) * (-1) = -0.8125$$

$$w_{12}(new) = w_{12}(old) + \Delta w_{12} = -0.575 - 0.8125 = -0.3875$$

$$\Delta b_1 = lr * (t - z_{in1}) = 0.5 * (1 + 0.625) * 1 = 0.8125$$

$$b_1(new) = b_1(old) + \Delta b_1 = -0.475 + 0.8125 = 0.3375$$

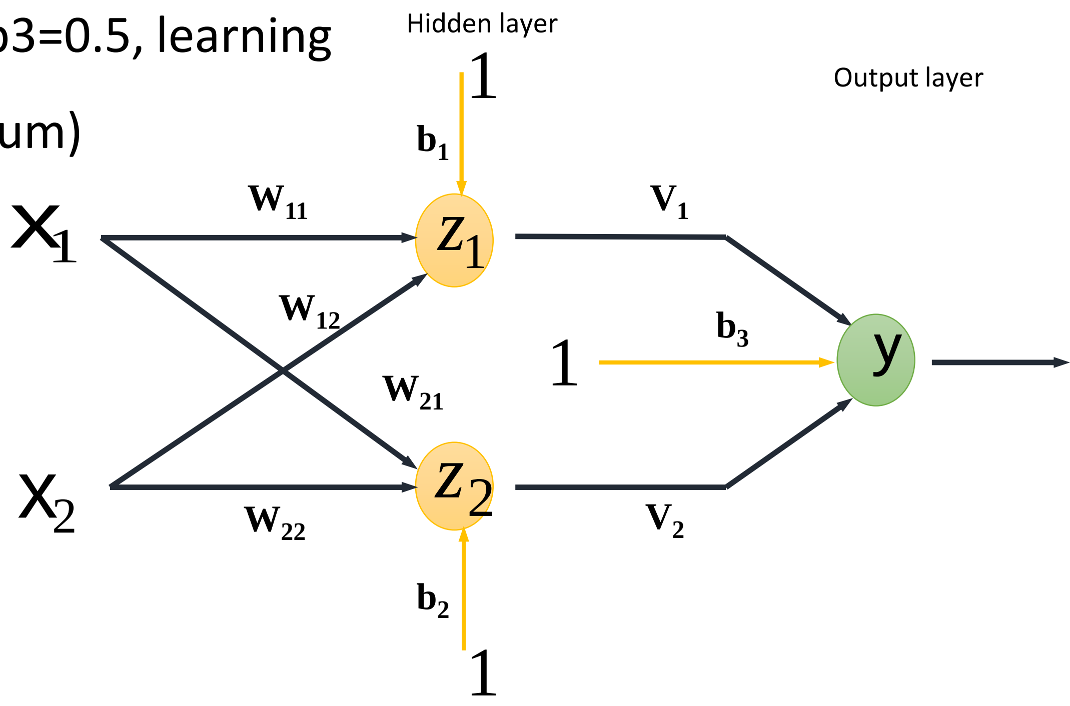
Multiple Adaline (Madaline) Example (XOR)

W11=0.05, W12=0.2, W21=0.1, W22=0.2, b1=0.3, b2=0.15, V1=V2=b3=0.5, learning

X_1	X_2	t
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Note: activation function is hardlims (signum)

$$a = hard\lim s(n) = \begin{cases} 1 & \text{if } a \geq 0 \\ -1 & \text{if } a \prec 0 \end{cases}$$



Solution: For the third sample (-1,1) →

$$Z_1 = \text{hardlims}(Z_{in1}) = \text{hardlims}(w_{11(new)} * x_1 + w_{12(new)} * x_2 + b_{1(new)})$$

$$= \text{hardlims}((0.0875*(-1)) + (-1.3875*(1)) + 0.3375)$$

$$= \text{hardlims}(-1.1375) = -1$$

$$Z_2 = \text{hardlims}(Z_{in2}) = \text{hardlims}(w_{21(new)} * x_1 + w_{22(new)} * x_2 + b_{2(new)})$$

$$= \text{hardlims}((-0.625*(-1)) + (-0.525*(1)) - 0.575)$$

$$= \text{hardlims}(-0.475) = -1$$

$$y = \text{hardlims}(y_{in}) = \text{hardlims}(v_1 * z_1 + v_2 * z_2 + b_3)$$

$$= \text{hardlims}(-0.5 - 0.5 + 0.5) = \text{hardlims}(-0.5) = -1$$

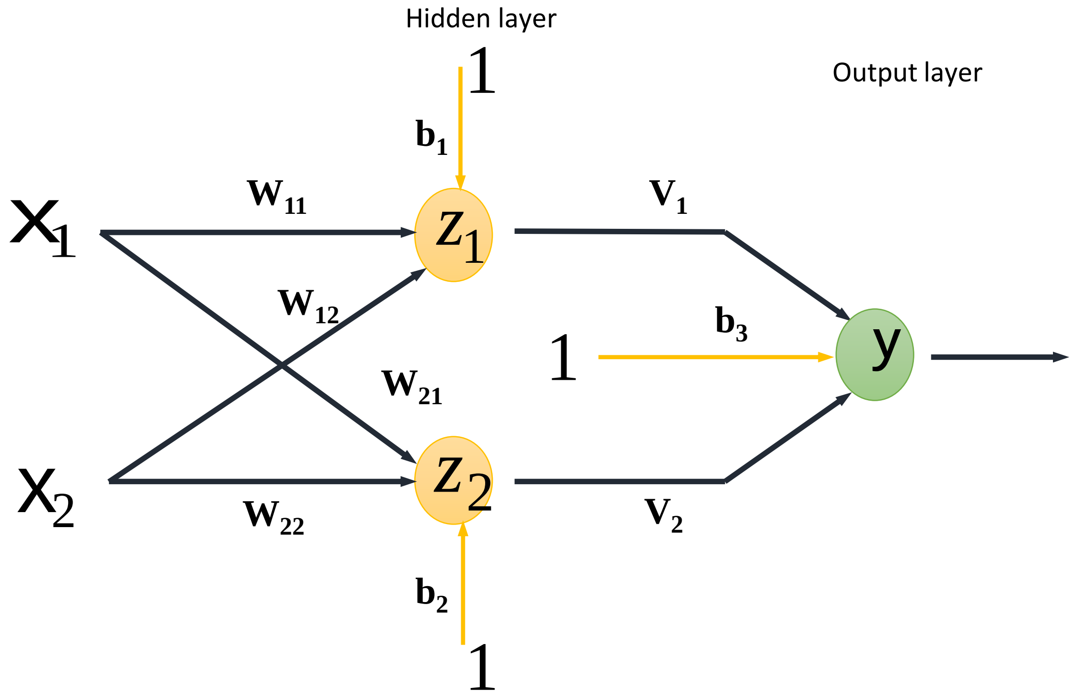
t$\neq y$ and (t=1) → update weights of Z whose Zin is closest to 0 and here Z2 have the lowest absolute zin value closest to 0 → update only W21, W22, b2

Multiple Adaline (Madaline) Example (XOR)

W11=0.05, W12=0.2, W21=0.1, W22=0.2, b1=0.3, b2=0.15, V1=V2=b3=0.5, learning rate=0.5

X_1	X_2	t
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Solution:



$$\Delta w_{21} = lr*(t - z_{in2})*x_1 = 0.5*(1+0.475)*(-1) = -0.7375$$

$$w_{21}(new) = w_{21}(old) + \Delta w_{21} = -0.625 - 0.7375 = -1.3625$$

$$\Delta w_{22} = lr*(t - z_{in2})*x_2 = 0.5*(1+0.475)*1 = 0.7375$$

$$w_{22}(new) = w_{22}(old) + \Delta w_{22} = -0.525 + 0.7375 = 0.2125$$

$$\Delta b_2 = lr*(t - z_{in2}) = 0.5*(1+0.475)*1 = 0.7375$$

$$b_2(new) = b_2(old) + \Delta b_2 = -0.575 + 0.7375 = 0.1625$$

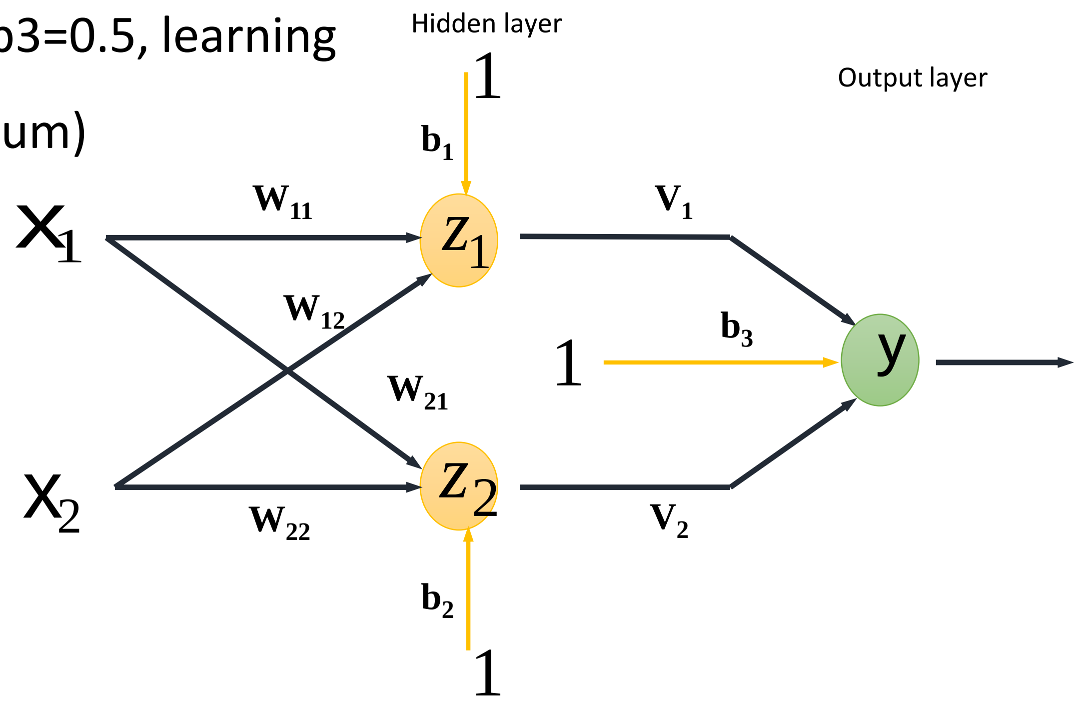
Multiple Adaline (Madaline) Example (XOR)

W11=0.05, W12=0.2, W21=0.1, W22=0.2, b1=0.3, b2=0.15, V1=V2=b3=0.5, learning

X_1	X_2	t
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Note: activation function is hardlims (signum)

$$a = hard\lim s(n) = \begin{cases} 1 & \text{if } a \geq 0 \\ -1 & \text{if } a \prec 0 \end{cases}$$



Solution: For the third sample (-1,-1) →

$$Z_1 = \text{hardlims}(Z_{in1}) = \text{hardlims}(w_{11(new)} * x_1 + w_{12(new)} * x_2 + b_{1(new)})$$

$$= \text{hardlims}((0.0875 * (-1)) + (-1.3875 * (-1)) + 0.3375)$$

$$= \text{hardlims}(1.6375) = 1$$

$$Z_2 = \text{hardlims}(Z_{in2}) = \text{hardlims}(w_{21(new)} * x_1 + w_{22(new)} * x_2 + b_{2(new)})$$

$$= \text{hardlims}((-1.3625 * (-1)) + (0.2125 * (-1)) + 0.1625)$$

$$= \text{hardlims}(1.3125) = 1$$

$$y = \text{hardlims}(y_{in}) = \text{hardlims}(v_1 * z_1 + v_2 * z_2 + b_3)$$

$$= \text{hardlims}(0.5 + 0.5 + 0.5) = \text{hardlims}(1.5) = 1$$

t$\neq y$ and (t=-1) →

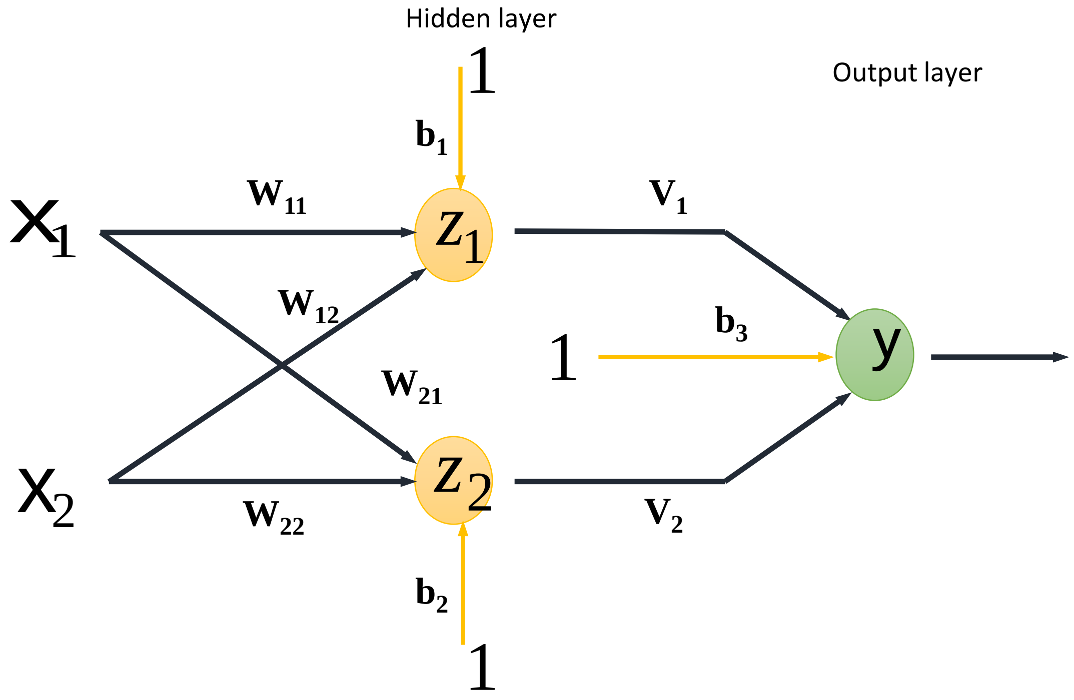
update weights of only positive Zins (Zin1 and Zin2 are positives) → Update W11,W12, W21,W22, b1, b2

Multiple Adaline (Madaline) Example (XOR)

W11=0.05, W12=0.2, W21=0.1, W22=0.2, b1=0.3, b2=0.15, V1=V2=b3=0.5, learning rate=0.5

X_1	X_2	t
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Solution:



$$w_{new} = w_{old} + \Delta w$$

$$\begin{bmatrix} w_{11new} & w_{12new} \\ w_{21new} & w_{22new} \end{bmatrix} = \begin{bmatrix} w_{11old} & w_{12old} \\ w_{21old} & w_{22old} \end{bmatrix} + \begin{bmatrix} \Delta w_{11} & \Delta w_{12} \\ \Delta w_{21} & \Delta w_{22} \end{bmatrix}$$

$$\begin{bmatrix} w_{11new} & w_{12new} \\ w_{21new} & w_{22new} \end{bmatrix} = \begin{bmatrix} 0.0875 & -1.3875 \\ -1.3625 & 0.2125 \end{bmatrix} + \begin{bmatrix} 1.319 & 1.3217 \\ 1.1555 & 1.1565 \end{bmatrix}$$

$$\begin{bmatrix} w_{11new} & w_{12new} \\ w_{21new} & w_{22new} \end{bmatrix} = \begin{bmatrix} 1.4065 & -0.0685 \\ -0.207 & 1.369 \end{bmatrix}$$

$$b_{new} = b_{old} + \Delta b$$

$$\begin{bmatrix} b_{1new} \\ b_{2new} \end{bmatrix} = \begin{bmatrix} b_{1old} \\ b_{2old} \end{bmatrix} + \begin{bmatrix} \Delta b_1 \\ \Delta b_2 \end{bmatrix}$$

$$\begin{bmatrix} b_{1new} \\ b_{2new} \end{bmatrix} = \begin{bmatrix} 0.3375 \\ 0.1625 \end{bmatrix} + \begin{bmatrix} -1.319 \\ -1.1565 \end{bmatrix}$$

$$\begin{bmatrix} b_{1new} \\ b_{2new} \end{bmatrix} = \begin{bmatrix} -0.9815 \\ -0.994 \end{bmatrix}$$

Multiple Adaline (Madaline) Example (XOR)

W11=0.05, W12=0.2, W21=0.1, W22=0.2, b1=0.3, b2=0.15, V1=V2=b3=0.5, learning rate=0.5

X_1	X_2	t
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Weights and biases at the end of epoch 1

Inputs				t	Zin1	Zin2	W11	W21	b1	W12	W22	b2	Z1	Z2	Yin	y
	x1	x2	b													
Epoch1	1	1	1	-1	0.55	0.45	-0.725	-0.575	-0.475	-0.625	-0.525	-0.575	1	1	1.5	1
	1	-1	1	1	-0.625	-0.675	0.0875	-1.3875	0.3375	-0.625	-0.525	-0.575	-1	-1	-0.5	-1
	-1	1	1	-1	-1.1375	-0.475	0.0875	-1.3875	0.3375	-1.3625	0.2125	0.1625	-1	-1	-0.5	-1
	-1	-1	1	-1	1.6375	1.3125	1.4065	-0.0685	-0.9815	-0.207	1.369	-0.994	1	1	1.5	1

Multiple Adaline (Madaline) Example (XOR)

W11=0.05, W12=0.2, W21=0.1, W22=0.2, b1=0.3, b2=0.15, V1=V2=b3=0.5, learning rate=0.5

Weights and biases at the end of epoch 2

Inputs				t	Zin1	Zin2	W11	W21	b1	W12	W22	b2	Z1	Z2	Yin	y
	x1	x2	b													
Epoch2	1	1	1	−1	0.3565	0.168	0.7285	−0.7465	−1.6595	−0.791	−0.207	−1.578	1	1	1.5	1
	1	−1	1	1	−0.1845	−3.154	1.3205	−1.339	−1.068	−0.791	0.785	−1.578	−1	−1	−0.5	−1
	−1	1	1	−1	−3.728	−0.002	1.3205	−1.339	−1.068	−1.292	0.785	−1.077	−1	−1	−0.5	−1
	−1	−1	1	−1	−1.0495	−1.071	1.3205	−1.339	−1.068	−1.292	1.286	−1.077	−1	−1	−0.5	−1

Weights and biases at the end of epoch 3

Inputs				t	Zin1	Zin2	W11	W21	b1	W12	W22	b2	Z1	Z2	Yin	y
	x1	x2	b													
Epoch3	1	1	1	−1	−1.0865	−1.083	1.3205	−1.339	−1.068	−1.292	1.286	−1.077	−1	−1	−0.5	−1
	1	−1	1	1	1.5951	−3.655	1.3205	−1.339	−1.068	−1.292	1.286	−1.077	1	−1	0.5	1
	−1	1	1	−1	−3.728	1.501	1.3205	−1.339	−1.068	−1.292	1.286	−1.077	−1	1	0.5	1
	−1	−1	1	−1	−1.0495	−1.071	1.3205	−1.339	−1.068	−1.292	1.286	−1.077	−1	−1	−0.5	−1

Multiple Adaline (Madaline) Example (XOR)

Let's draw the two decision lines:

$$Z_{in1} = 0 \quad \Rightarrow w_{11} * x_1 + w_{12} * x_2 + b_1 = 0$$

$$x_2 = -\frac{w_{11}}{w_{12}} x_1 - \frac{b_1}{w_{12}} = 0.9862 x_1 - 0.798 \qquad L_1$$

$$Z_{in2} = 0 \quad \Rightarrow w_{21} * x_1 + w_{22} * x_2 + b_2 = 0$$

$$x_2 = -\frac{w_{21}}{w_{22}} x_1 - \frac{b_2}{w_{22}} = 1.0047 x_1 + 0.837 \qquad L_2$$