Lecture Series on

# Intelligent Control

Lecture 9
**Artificial Neural Networks**
**Associative Learning**

Kwang Y. Lee
Professor of Electrical & Computer Engineering
Baylor University
Waco, TX 76706, USA
Kwang_Y_Lee@baylor.edu

1

## Associative Learning

An association is any link between a system's input and output such that when a pattern A is presented to the system it will respond with pattern B.

When two patterns are linked by an association, the input pattern is often referred to as stimulus. Likewise, the output pattern is referred to as the response.

Associations are so fundamental that they formed the foundation of the behaviorist school of psychology. This branch of psychology attempted to explain much of animal and human behavior by using associations and rules for learning associations.

2

## Associative Learning

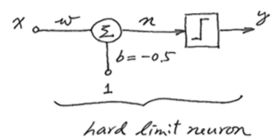
One of the earliest influences on the behavorist school of psychology was the classic experiment of Ivan Pavlov, in which he trained a dog to salivate at the sound of a bell, by ringing the bell whenever food was presented. This is an example of what is now called classical conditioning.

It was to provide a biological explanation for some of this behavior that led Donald Hebb to his postulate in The Organization of Behavior: The Hebbian learning.

3

## Associative Learning

Simple Associative Network




$x = \begin{cases} 1, & \text{stimulus} \\ 0, & \text{no stimulus} \end{cases}$

$y = \begin{cases} 1, & \text{response} \\ 0, & \text{no response} \end{cases}$

The presence of an association between the stimulus $x = 1$, and the response $y = 1$ is dictated by the value of $w$.

$$y = \text{hardlimit}(wx + b) = \text{hardlimit}(wx - 0.5)$$

The network will respond to the stimulus only if $w$ is greater than $-b = 0.5$.

4

## Associative Learning

Two types of inputs:

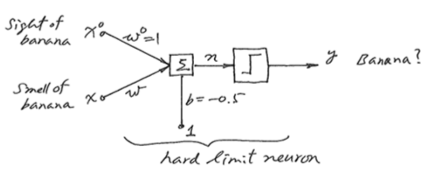
Unconditioned Stimulus : food — Unconditioned response: salivation

Conditioned stimulus : bell — Conditioned response: salivation

Banana Associator:

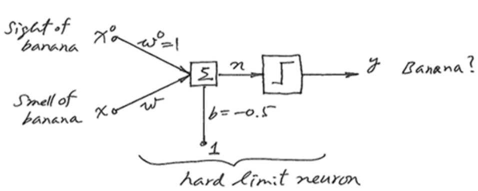
Unconditioned stimulus (banana shape) : $x^0$

Conditioned stimulus (banana smell) : $x$




5

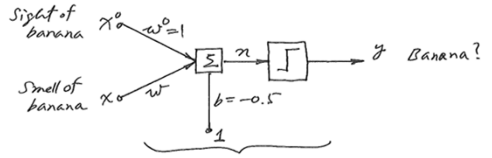
## Associative Learning




$x^0 = \begin{cases} 1, & \text{shape detected} \\ 0, & \text{" not detected} \end{cases}$ $x = \begin{cases} 1, & \text{smell detected} \\ 0, & \text{" not detected} \end{cases}$

6

## Associative Learning




First, the network associates the shape of banana, but not the smell, with a response indicating the fruit is a banana.

$$y = \text{hardlimit}\,(w^0 x^0 + w x + b)$$

$w=0$, $\quad w^0(1) + b > 0 \;\Rightarrow\; w^0 > -b = 0.5$

Assign: $\quad w^0 = 1\,,\; w = 0$

$\Rightarrow \quad y = \text{hardlim}\,(x^0 - 0.5)$

7

## Associative Learning

<u>Unsupervised Hebb Rule</u>:

When should an association be learned? It is generally accepted that both animals and humans tend to associate things that occur <u>simultaneously</u>.

Paraphrasing Hebb:

If a banana smell stimulus occurs simultaneously with a banana concept response (activated by some other stimulus such as the sight of a banana shape), the network should strengthen the connection between them so that later it can activate its banana concept in response to the banana smell alone.

8

## Associative Learning

Following the Hebb rule:

$$w_{ji}(p) = w_{ji}(p-1) + \alpha\, y_j(p)\, x_i(p)$$

$w_{ji}$ : weight between a neuron's input $x_i$ and output $y_j$

$\alpha$ : learning rate : dictates how many times a stimulus and response must occur together before an association is made.

In the banana associator, if we make

$$w(1) + b > 0 \;\Rightarrow\; w > -b = 0.5$$

an association is made, regardless of the value of $x^0$.

9

## Associative Learning

Note: The above rule uses only signals available within the layer containing the weights being updated. Rules that satisfy this condition are called local learning rules.

This is in contrast to the backpropagation rule, in which the sensitivity (delta) must be propagated back from the final layer.

10

## Associative Learning

In vector form:

$$W(p) = W(p-1) + \alpha\, y(p)\, x^T(p)$$

Learning is performed in response to a series of inputs presented in time (the training sequence):

$$x(1),\ x(2),\ \cdots,\ x(P)$$

11

## Associative Learning

Returning to the banana associator, with initial weights

$$w^0 = 1, \quad w(0) = 0.$$

Smell sensor works reliably, but the shape sensor operate only intermittently (on even time steps).

Thus, the training sequence will consist of the repititions of the following two sets of inputs:

$$\{x^0(1) = 0,\ x(1) = 1\},\ \{x^0(2) = 1,\ x(2) = 1\},\ \cdots$$

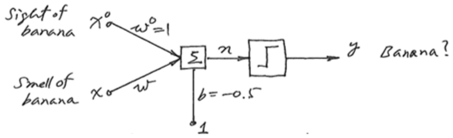
$w^0$ will remain constant

$w$ will be updated at each iteration, using the unsupervised Hebb rule with a learning rate of 1:

$$w(p) = w(p-1) + y(p)\, x(p).$$

12

## Associative Learning




$$w(p) = w(p-1) + y(p)\,x(p).$$

The output of the first iteration ($p=1$) is

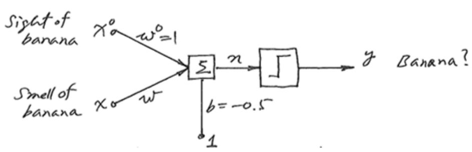
$$y(1) = \text{hardlim}(w^0 x^0 + w(0)\,x(1) - 0.5)$$
$$= \text{hardlim}((1)(0) + (0)(1) - 0.5)$$
$$= 0 \quad \text{(no response)}$$

The smell alone did not generate a response. Without a response, the Hebb rule does not alter $w$:

$$w(1) = w(0) + y(1)\,x(1) = 0 + (0)(1) = 0$$

13

## Associative Learning




In the second iteration ($p=2$), both the banana's shape and smell are detected and the network respond:

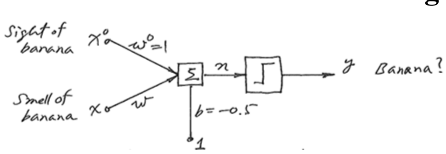
$$y(2) = \text{hardlim}(w^0 x^0(2) + w(1)\,x(2) - 0.5)$$
$$= \text{hardlim}((1)(1) + (0)(1) - 0.5)$$
$$= 1 \quad \text{(banana)}$$

The weight changed is

$$w(2) = w(1) + y(2)\,x(2) = 0 + (1)(1) = 1$$

14

## Associative Learning




When the sight detector fails again, in the third iteration, the network responds anyway.

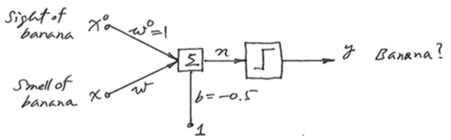
$$y(3) = \text{hardlim}(w^0 x^0(3) + w(2)\,x(3) - 0.5)$$
$$= \text{hardlim}((1)(0) + (1)(1) - 0.5)$$
$$= 1 \quad \text{(banana!)}$$

$$w(3) = w(2) + y(3)\,x(3) = 1 + (1)(1) = 2$$

From now on, the network is capable of responding to bananas that are detected by either by sight or smell.

15

## Associative Learning




NOTE: 1. If we continue to present inputs and update $w$, the weight $w$ will become arbitrarily large. This is at odds with the biological systems that inspired the Hebb rule. Synapses cannot grow without bound.

2. There is no mechanism for weights to decrease. If the inputs or outputs experience any noise, every weight will grow (however slowly) until the network responds to any stimulus.

16

## Associative Learning

Hebb Rule with Decay:

One way to improve the Hebb rule is by adding a weight decay term:

$$W(p) = W(p-1) + \alpha\, y(p)\, x^T(p) - \gamma\, W(p-1)$$

$$= (1-\gamma)\, W(p-1) + \alpha\, y(p)\, x^T(p),$$

where, $\gamma$, the decay rate, is a positive constant less than one. As $\gamma$ approaches zero, the learning law becomes the standard rule. As $\gamma$ approaches one, the learning law quickly forgets old inputs and remembers only the most recent patterns.

This keeps the weight matrix from growing without bound.

17

## Associative Learning

The maximum weight value $w_{ji}^{max}$ is determined by $\gamma$. To maximize learning, set both $x_j$ and $y_i$ to a value of 1 for all $p$ and solve for the steady state weight (i.e., when both new and old weights are equal):

$$w_{ji} = (1-\gamma)\, w_{ji} + \alpha\, y_j\, x_i$$

$$w_{ji} = (1-\gamma)\, w_{ji} + \alpha$$

$$w_{ji} = \frac{\alpha}{\gamma}$$

18

## Associative Learning

Return to the banana associator. Use a decay rate $\gamma$ of 0.1

The 1st iteration, with only the smell stimulus present, is the same:

$$y(1) = 0 \quad (\text{no response}), \quad w(1) = 0.$$

The 2nd iteration also produce the identical results:

$$y(2) = 1 \quad (\text{banana}), \quad w(2) = 1.$$

The 3rd iteration. The network has learned to respond to smell, and the weight continues to increase. However, this time the weight increases by only 0.9, instead of 1.0:

$$w(3) = w(2) + y(3)\,x(3) - 0.1\,w(2)$$
$$= 1 + (1)(1) - 0.1\,(1) = 1.9$$

19

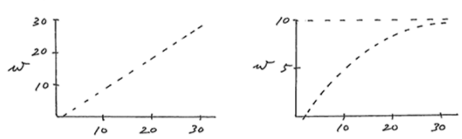
## Associative Learning

The decay term limits the weight's value, so that no matter how often the association is reinforced, $w$ will never increase beyond $w^{max}$

$$w^{max} = \frac{\alpha}{\gamma} = \frac{1}{0.1} = 10$$

The new rule also ensures that associations learned by the network will not be artifacts of noise.

Any small random increas will soon decay away.




Response without decay — Response with decay

20

## Associative Learning

NOTE: The environment must be counted on to occasionally present all stimuli that have associations. Without reinforcement, association will decay away:

For example, if $y_j = 0$,

$$w_{ji}(p) = (1-\gamma)\, w_{ji}(p-1)$$

If $\gamma = 0.1$, this reduces to

$$w_{ji}(p) = 0.9\, w_{ji}(p-1)$$

Therefore, $w_{ji}$ will be decreased by 10% at each presentation for which $y_j = 0$.

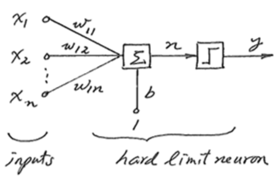
Any association that was previously learned will eventually be lost!

21

## Associative Learning

Simple Recognition Network

**Instar** : Neuron with a vector input.




The input/output expression for the instar is

$$y = \text{hardlim}(Wx + b) = \text{hardlim}({}_1w^T x + b)$$

22

## Associative Learning

The input/output expression for the instar is

$$y = \text{hardlim}(Wx + b) = \text{hardlim}({}_1w^T x + b)$$

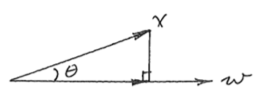
The instar will be active whenever the inner product between the weight vector and the input vector is greater than or equal to $-b$:

$${}_1w^T x \geq -b .$$

or

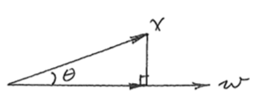
$${}_1w^T x = \| {}_1w \| \, \| x \| \cos\theta \geq -b .$$

where $\theta$ is the angle between two vectors.




23

## Associative Learning




$${}_1w^T x = \| {}_1w \| \, \| x \| \cos\theta \geq -b .$$

where $\theta$ is the angle between two vectors.

If $x$ and ${}_1w$ have the same length, then the inner product will be largest when $x = {}_1w$. Therefore, the instar will be active when $x$ is "close" to ${}_1w$.
By setting the bias $b$ appropriately, we can select how close the input vector must be to the weight vector in order to activate the instar.

24

## Associative Learning

If we set

$$b = -\|{}_1w\| \|x\|,$$

then the instar will only be active when $x$ points in exactly the same direction as ${}_1w$ ($\theta = 0$). Thus, we will have a neuron that recognizes only the pattern ${}_1w$.

If we would like the instar to respond to any pattern near ${}_1w$ ($\theta$ small), then we can increase $b$ to some value larger than $-\|{}_1w\| \|x\|$.

25

## Associative Learning

<u>Instar Rule</u>: One problem of the Hebb rule with decay was that it required stimuli to be repeated or associations would be lost.

A better rule might allow weight decay only when the instar is active ($y \neq 0$).

Consider the Hebb rule:

$$w_{ji}(p) = w_{ji}(p-1) + \alpha\, y_j(p)\, x_i(p)$$

Adding a decay term,

$$w_{ji}(p) = w_{ji}(p-1) + \alpha\, y_j(p)\, x_i(p) - \gamma\, y_j(p)\, w_{ji}^{old}$$

We can simplify by setting $\gamma$ equal to $\alpha$ (so new weight values are learned at the same rate old values decay),

$$w_{ji}(p) = w_{ji}(p-1) + \alpha\, y_j(p) \left( x_i(p) - w_{ji}^{old} \right)$$

26

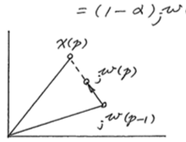
## Associative Learning

In vector form,

$${}_jw(p) = {}_jw(p-1) + \alpha\, y_j(p) \left( x(p) - {}_jw(p-1) \right)$$

Consider the case when the instar is active ($y_j = 1$).

$${}_jw(p) = {}_jw(p-1) + \alpha \left( x(p) - {}_jw(p-1) \right)$$
$$= (1-\alpha)\, {}_jw(p-1) + \alpha\, x(p).$$




<u>NOTE</u>: When the instar is active, the weight vector is moved toward the input vector along a line between the old weight vector and the input vector.

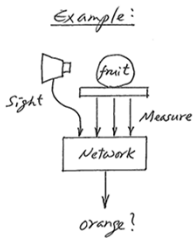
The distance the weight vector moves depends on the value of $\alpha$.

If the input vectors are normalized, then ${}_jw$ will also be normalized once it has learned a particular vector $x$.

27

## Associative Learning

Example:




Inputs:

visual input (orange) : unconditioned stimulus

Measured inputs : conditioned stimulus.

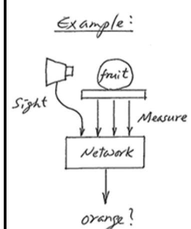
$$x^0 = \begin{cases} 1, & \text{orange detected visually} \\ 0, & \text{orange not detected.} \end{cases}$$

$$x = \begin{bmatrix} \text{shape} \\ \text{texture} \\ \text{weight} \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}, \quad \|x\| = \sqrt{3}$$

28

## Associative Learning

Example:

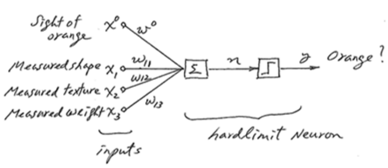



The output of the network is

$$y = \text{hardlim}(w^0 x^0 + Wx + b)$$

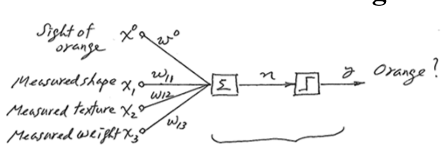
The bias : $b = -\|w\|\,\|x\| = -\|x\|^2 = -3$

Set bias as $b = -2$.




29

## Associative Learning




We need the network to have a constant association between the sight of an orange and its response, so $w^0$ will be set greater value than $-b$. But initially, the network should not respond to any combination of fruit measurements:

$$w^0 = 3 \qquad W(0) = {}_1w^T(0) = [\,0 \quad 0 \quad 0\,]$$

Use the instar rule to update the weights with $\alpha = 1$.

$${}_1w(p) = {}_1w(p-1) + y(p)\left(x(p) - {}_1w(p-1)\right)$$

30

Training sequence: repeated presentation of an orange.
Assume - visual system only operates correctly on even time steps. (due to a fault in its construction).

$$\left\{x^0(1)=0,\ x(1)=\begin{bmatrix}1\\-1\\-1\end{bmatrix}\right\},\ \left\{x^0(2)=1,\ x(2)=\begin{bmatrix}1\\-1\\-1\end{bmatrix}\right\},\ \ldots$$

1st: $y(1) = \text{hardlim}(w^0 p(1) + W x(1) - 2)$

$$= \text{hardlim}\left((3)(0) + [0\ 0\ 0]\begin{bmatrix}1\\-1\\-1\end{bmatrix} - 2\right) = 0 \quad \text{(no response)}$$

$${}_1w(1) = {}_1w(0) + y(1)\,(x(1) - {}_1w(0)) = 0 \quad \text{(not adjusted)}$$

2nd: $y(2) = \text{hardlim}(w^0 x(2) + W x(2) - 2)$

$$= \text{hardlim}\left((3)(1) + [0\ 0\ 0]\begin{bmatrix}1\\-1\\-1\end{bmatrix} - 2\right) = 1 \quad \text{(orange)}$$

$${}_1w(2) = {}_1w(1) + y(2)\,(x(2) - {}_1w(1))$$

$$= \begin{bmatrix}0\\0\\0\end{bmatrix} + 1\left(\begin{bmatrix}1\\-1\\-1\end{bmatrix} - \begin{bmatrix}0\\0\\0\end{bmatrix}\right) = \begin{bmatrix}1\\-1\\-1\end{bmatrix}$$

31

## Associative Learning

3rd: $y(3) = \text{hardlim}(w^0 x^0(3) - W x(3) - 2)$

$$= \text{hardlim}\left((3)(0) + [1\ -1\ -1]\begin{bmatrix}1\\-1\\-1\end{bmatrix} - 2\right) = 1 \quad \text{(orange)}$$

$${}_1w(3) = {}_1w(2) + y(3)\,(x(3) - {}_1w(2))$$

$$= \begin{bmatrix}1\\-1\\-1\end{bmatrix} + 1\left(\begin{bmatrix}1\\-1\\-1\end{bmatrix} - \begin{bmatrix}1\\-1\\-1\end{bmatrix}\right) = \begin{bmatrix}1\\-1\\-1\end{bmatrix}$$

Having completely learned the measurements, the weights stop changing. The network has learned to recognise an orange by its measurements, even when its visual detection system fails.

32

## Associative Learning

Kohonen Rule:

$${}_iw(p) = {}_iw(p-1) + \alpha\,(x(p) - {}_iw(p-1)), \quad \text{for } i \in I(p).$$

Unlike the instar rule, learning is not proportional to the neuron's output $y_i(p)$. Instead, learning occurs when the neuron's index $i$ is a member of the set $I(p)$.

Kohonen rule can be made equivalent to the instar rule by defining $I(p)$ as the set of all $i$ such that $y_i(p) = 1$.

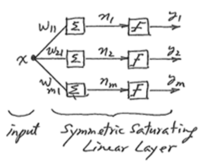
The advantage of the Kohonen rule is that it can also be used with other definitions. It is useful for training networks such as the self-organizing feature map.

33

## Associative Learning

Simple Recall Network:

Outstar Network: Has scalar input and a vector output.




The input/output association is

$y = satlins(Wx)$

: recall a vector containing values of $-1$ or $1$.

If $x=1$, stimulus, then

$y = satlins(Wx) = satlins(w \cdot 1) = w$

: $w$ is an output vector associated with input $x=1$.

Outstar: Column of the weight matrix is set to the desired vector.

Instar: Row of the weight matrix is set to the desired vector.

34

Outstar Rule:

Weight decay term of the Hebb rule:

proportional to the output of the network, $y_j$ : instar

" to the input of the network, $x_i$ : outstar

$$w_{ji}(p) = w_{ji}(p-1) + \alpha\, y_j(p)\, x_i(p) - \gamma\, x_i(p)\, w_{ji}(p-1)$$

Setting $\gamma = \alpha$,

$$w_{ji}(p) = w_{ji}(p-1) + \alpha\left(y_j(p) - w_{ji}(p-1)\right) x_i(p)$$

The outstar rule has properties complimentary to the instar rule. Learning occurs whenever $x_i$ is nonzero (instead of $y_j$). When learning occurs, column $w_j$ moves toward the output vector.

In vector form,

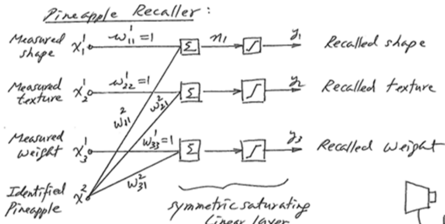
$$w_j(p) = w_j(p-1) + \alpha\left(y(p) - w_j(p-1)\right) x_i(p)$$

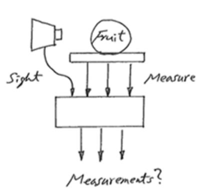
where $w_j$ is the $j$th column of the matrix $W$.

35

## Associative Learning

Pineapple Recaller:

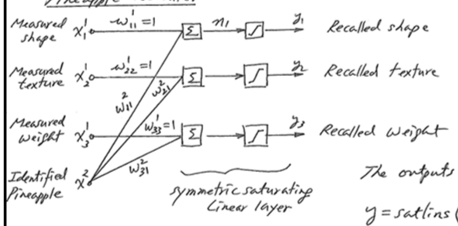



36

## Associative Learning

*Pineapple Recaller:*




The outputs of the network:

$$y = \text{satlins}(W^0 x^0 + W x)$$

$$W^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad W(0) = 0, \text{ initial}$$

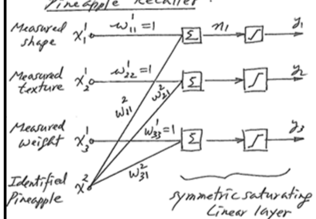
$x^0$ : unconditioned stimulus

$x$ : conditioned stimulus

37

## Associative Learning

*Pineapple Recaller:*




The outputs of the network:

$$y = \text{satlins}(W^0 x^0 + W x)$$

$$W^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad W(0) = 0, \text{ initial}$$

$x^0$ : unconditioned stimulus

$x$ : conditioned stimulus

$$x^0 = \begin{bmatrix} \text{shape} \\ \text{texture} \\ \text{weight} \end{bmatrix} \qquad x = \begin{cases} 1, & \text{if a pineapple can be seen} \\ 0, & \text{otherwise} \end{cases}$$

$$x^{\text{pineapple}} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix},$$ measured values will only be available on even iterations.

38

## Associative Learning

$$x^0 = \begin{bmatrix} \text{shape} \\ \text{texture} \\ \text{weight} \end{bmatrix} \qquad x = \begin{cases} 1, & \text{if a pineapple can be seen} \\ 0, & \text{otherwise} \end{cases}$$

$$x^{\text{pineapple}} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix},$$ measured values will only be available on even iterations.

Training sequence:

$$\left\{ x^0(1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, x(1) = 1 \right\}, \left\{ x^0(2) = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}, x(2) = 1 \right\}, \cdots$$

1st: $y(1) = \text{satlins}(W^0 x^0(1) + W x(1))$

$$= \text{satlins}\left( \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} 1 \right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \text{ no response}$$

39

## Associative Learning

$$x^0 = \begin{bmatrix} \text{shape} \\ \text{texture} \\ \text{weight} \end{bmatrix} \qquad x = \begin{cases} 1, & \text{if a pineapple can be seen} \\ 0, & \text{otherwise} \end{cases}$$

$$x^{pineapple} = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix},$$ measured values will only be available on even iterations.

Training sequence:

$$\left\{ x^0(1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, x(1) = 1 \right\}, \left\{ x^0(2) = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}, x(2) = 1 \right\}, \ldots$$

1st: $$y(1) = \text{satlins}\left( W^0 x^0(1) + W\, x(1) \right)$$

$$= \text{satlins}\left( \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} 1 \right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$ no response

$$w_j(p) = w_j(p-1) + \left( y(p) - w_j(p-1) \right) x(p)$$

$$w_1(1) = w_1(0) + \left( y(1) - w_1(0) \right) x(1)$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \left( \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) 1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$ no changes

40

## Associative Learning

2nd: $$y(2) = \text{satlins}\left( \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} 1 \right) = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$$ measurements given

$$w_1(2) = w_1(1) + \left( y(2) - w_1(1) \right) x(2)$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \left( \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right) 1 = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$$

3rd: Measurements are unavailable,

$$y(3) = \text{satlins}\left( \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} 1 \right) = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$$ measurements recalled.

$$w_1(3) = w_1(2) + \left( y(2) - w_1(2) \right) y(2)$$

$$= \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} + \left( \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} - \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \right) 1 = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix}$$

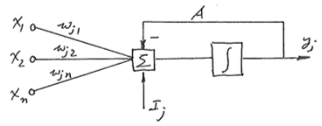
From now on, the weights will no longer change values unless a pineapple is seen with different measurements.

41

## Grossberg Learning

Mathematically formulate Hebb's law in continuous-time.

$$\frac{\partial y_j(t)}{dt} = -A\, y_j(t) + I_j(t) + \sum_{i=1}^{n} w_{ji}\, \underset{\substack{\parallel \\ x_i}}{y_i(t)}$$




$y_j(t)$ : output of the neuron

$I_j(t)$ : external input to the neuron

$w_{ji}(t)$ : weight connecting the output of ith neuron to the input of the jth neuron

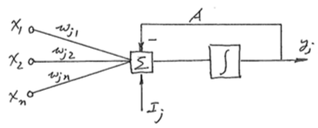
$A$ : positive constant contrlling the decay of the output in the absence of any other inputs.

42

## Grossberg Learning

Mathematically formulate Hebb's law in continuous-time.

$$\frac{\partial y_j(t)}{dt} = -A\, y_j(t) + I_j(t) + \sum_{i=1}^{n} w_{ji}\, y_i(t), \quad y_i(t) = x_i$$



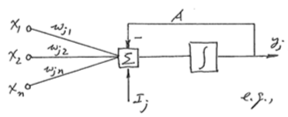

In Pavlov's dog:
- $I_j$ : plate of food
- $x_i$ : bell ringing
- $y_j$ : dog is salivating
- $A$ : dog will eventually discontinue salivating if it is not sufficiently stimulated by either the plate of food or the bell.

43

## Grossberg Learning

Mathematically formulate Hebb's law in continuous-time.

$$\frac{\partial y_j(t)}{dt} = -A\, y_j(t) + I_j(t) + \sum_{i=1}^{n} w_{ji}\, y_i(t), \quad y_i(t) = x_i$$




e.g., $e^{-At} = e^{-t/T}$, $T = 10^{-2} \Rightarrow A = 100$

$e^{-Ft} = e^{-t/T}$, $T = 1 \Rightarrow F = 1$

Learning of the network:

$$\frac{\partial w_{ji}(t)}{\partial t} = -F\, w_{ji}(t) + G\, y_j(t)\, y_i(t)$$

where $F$ : forgetting rate (much smaller than $A$)

$G$ : learning rate in Hebb rule.

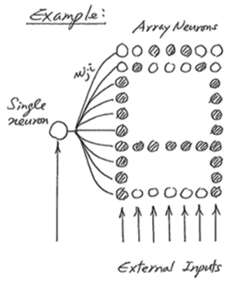
Time constants for $A$ : in the order of milliseconds

" $F$ : " seconds

44

## Grossberg Learning

<u>Example:</u>




Training:

External inputs are used to stimulate the single neuron and to stimulate the array with a pattern (A, for example)

The array neurons will adjust their output $y_j(t)$ and weight $w_{ji}(t)$ from the single neuron.

Array neurons which are not stimulated by an external input (O in the figure) have a small output and their weight slowly decay.

$\Rightarrow$ Array neurons do not respond to the single neuron's stimulation.

45

## Grossberg Learning

On the otherhand, array neurons stimulated by an external input have large outputs and increase the strength of the weight, learning to respond to the stimulus of the single neuron.

If the above external stimulations are repeated a sufficient number of times, the array neurons eventually learn to produce the pattern when only the single neuron stimulate them.

Network is trained to respond to the conditioned stimulus in the absence of the unconditioned stimulus.

By adding additional single neurons, and retraining the array neurons to respond with a different pattern (say, B), the network can be made useful.

46