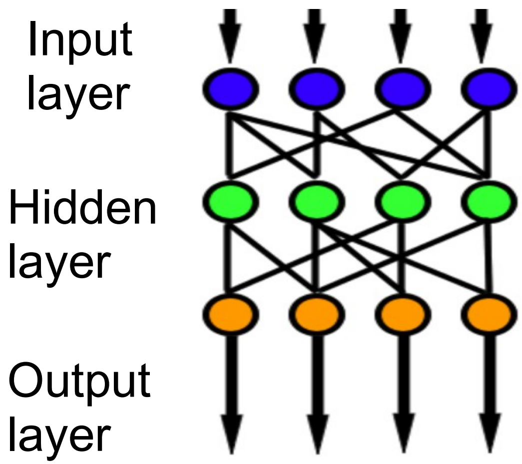
Feed Forward NNs and Perceptron Definition

https://en.wikipedia.org/wiki/Feedforward_neural_network

A feedforward neural network whose connections between the units do not form a cycle

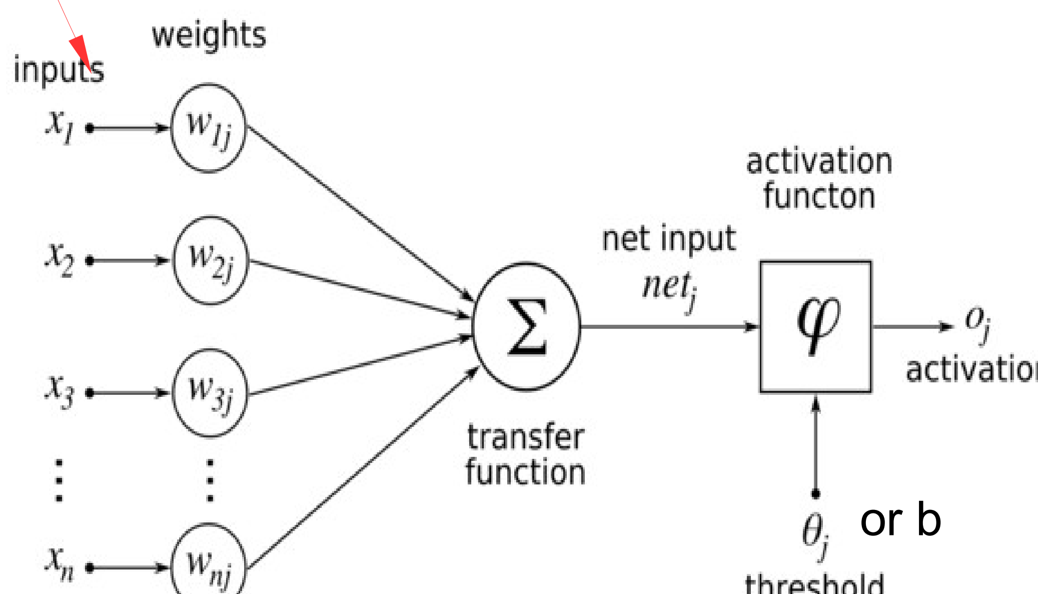


The Simplest form:
The perceptron is a network for supervised learning, a binary classifier, e.g., maps its input x to an output f (x)

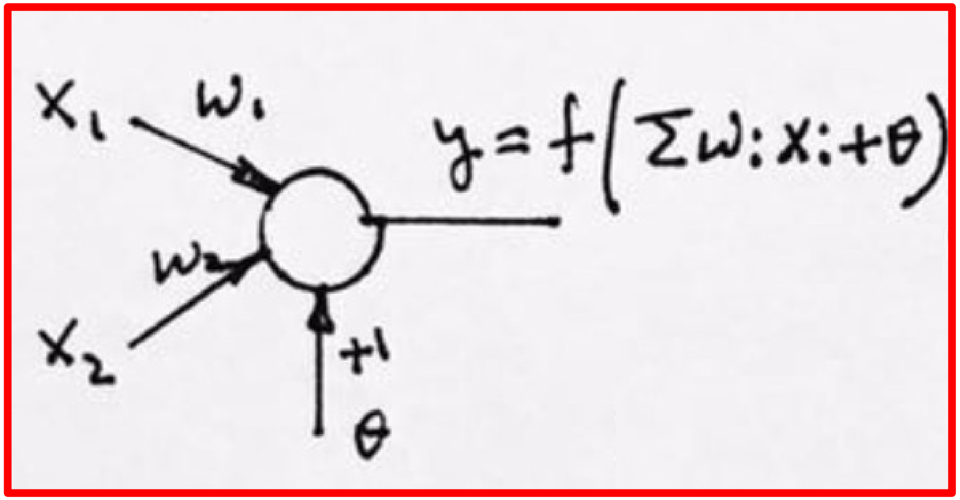
$$f(x) = \begin{cases} 1 & \text{if } \boxed{w \cdot x + b} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$\boxed{w \cdot x}$ is the dot product $\sum_{i=1}^{m} w_i x_i$

1. number of inputs: n;
2. threshold phi (or denoted as b), or bias, does not affect the orientation of the decision function;
3. transfer functions can be step function or sigmoid



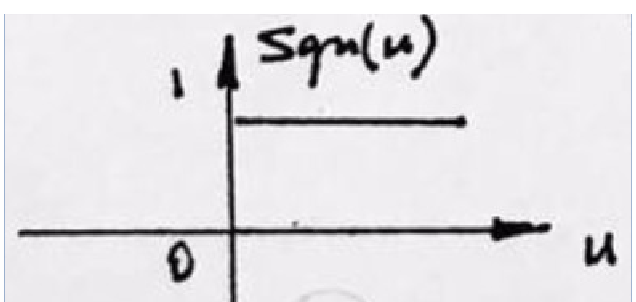
Feed Forward NN Example (1/3)



$$\boxed{y=f\left(\sum w_i x_i+\theta\right)} \quad \ldots (1)$$

Where

$$\boxed{y=f(\cdot) \text{ as } \mathrm{sgn}(\cdot)} \quad \ldots (2)$$



Given 2 traffic signs, train neural network to detect them based on moments (features)

$$M_{ij}=\sum_x\sum_y x^i y^j I(x,y)$$

Suppose moments of 2 traffic signs are computed for i=0, j=2, and i=2, j=0, shown below, and their values are plotted in the figure below

From stop sign group (3 stop sign samples)

From right turn group (3 right turns)

C_1: (1,2), (1,3), (2,3)

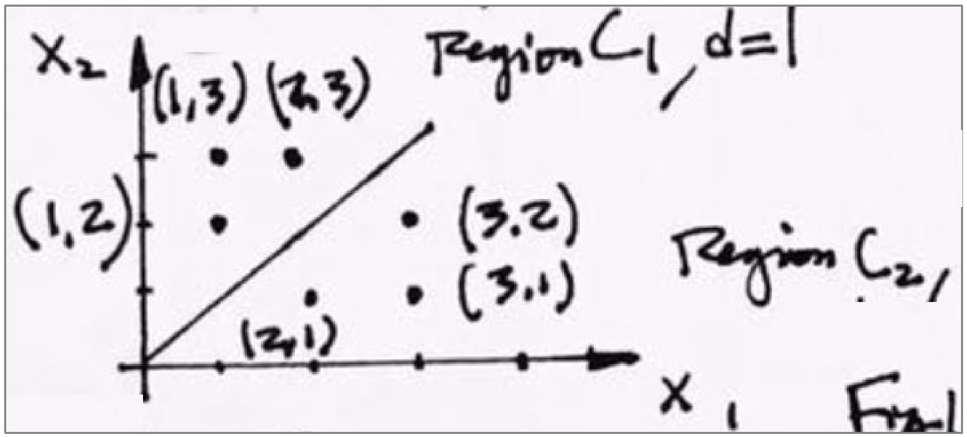
C_2: (2,1), (3,1), (3,2)

This is supervised learning for feed forward NNs

Feed Forward NN Example (2/3)

$C_1: (1,2), (1,3), (2,3)$

$C_2: (2,1), (3,1), (3,2)$



The training algorithm:

$$\boxed{w_i^+ = w_i^- + \eta(d-y)x_i} \quad \ldots (3)$$

where

η is gain.

d : desired the output.

y: actual output.

First, initialize

$$w_i = 0.5, \text{ for } i=1,2$$

and

$$\theta = 0.5$$

From the given data set, choose (1,2) from C_1, for w_1:

$$w_1^+ = w_1^- + \eta(1-y)x_1 \Big|_{x_1=1}$$
$$= 0.5 + \eta(1-y)\cdot 1$$

Then

$$y = f\left(\sum_{i=1}^{2} w_i x_i + \theta\right)$$
$$= f(w_1x_1 + w_2x_2 + \theta)$$
$$= f(0.5\times 1 + 0.5\times 2 + 0.5) = f(2)$$
$$= \mathrm{sgn}(2) = 1$$

hence,

$$w_1^+ = w_1^- + \eta(1-1)x_1 = w_1^- = 0.5$$

See my handout, continue this process

Feed Forward NN Example (3/3)

Step by step from my handout, till w_i converged as

$$\boxed{\begin{cases} w_1^+ = -0.4 \\ w_2^+ = -0.1 \end{cases}}$$

The convergence is achieve when w_i now longer updates and exhausted the input data

Now the NN is trained. We can conduct test (deployment)

Suppose from Stop sign we have (4,2), use the trained NN we have

$$y = f\left(\sum_{i=1}^{2} w_i x_i + \theta\right)$$

$$= f\left(w_1 x_1 + w_2 x_2 + \theta\right) =$$

$$f\left(-0.4 \times 4 - 0.1 \times 2 + 0.5\right) = f(-1.3) = 0$$

therefore, (4,2) belongs to C_2.

(which is correct | see the figure | illustration)

see Fig.

Appendix A: Raw Moments

The "raw moment" of order (p + q) for image f(x,y) is defined as:

$$M_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,dx\,dy \qquad (1)$$

For the discrete function, we have:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y) \qquad (2)$$

Note: image I(x,y) can be binary image or gray scale images. But we start the discussion from the binary images first.

We can treat image intensity as its probability density function

$$\sum_x \sum_y I(x,y) \qquad (3)$$

Reference: Robot Vision, by BPK, Horn, Chapter 3, pp. 46-64

Appendix B

Example On Simple Pattern Recognition

Given two binary images, derived from two objects, T and O, design a technique to identify them

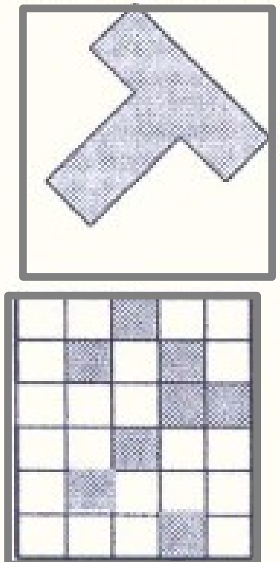


Fig1(a),(b)

Example: Computation of
(1) Area (size);
(2) X-bar;
(3) Y-bar;
(4) Orientation, theta angle
(5) Perimeter of an object

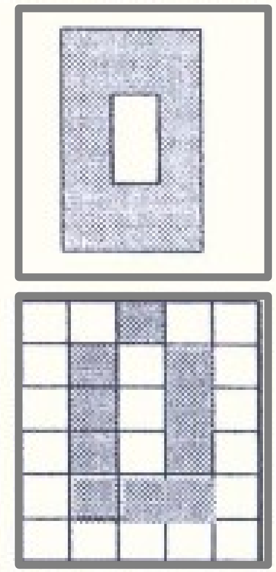


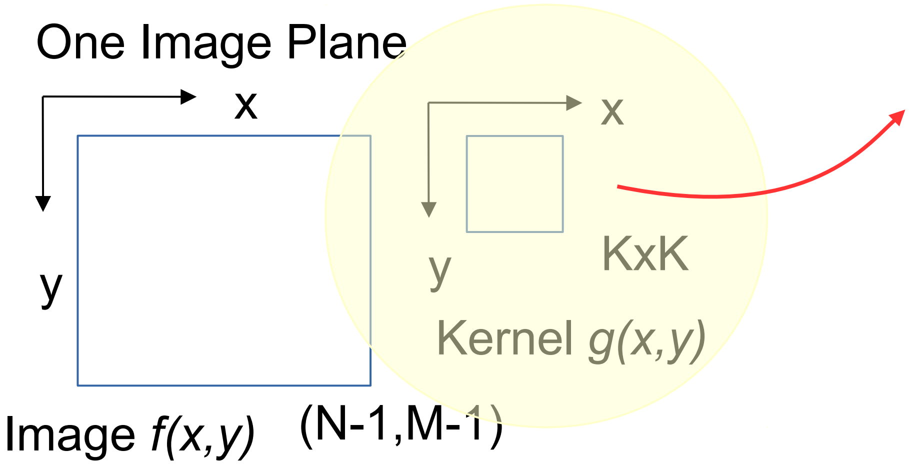
Fig2(a),(b)

Good continuation or noise? What to do with this noise?

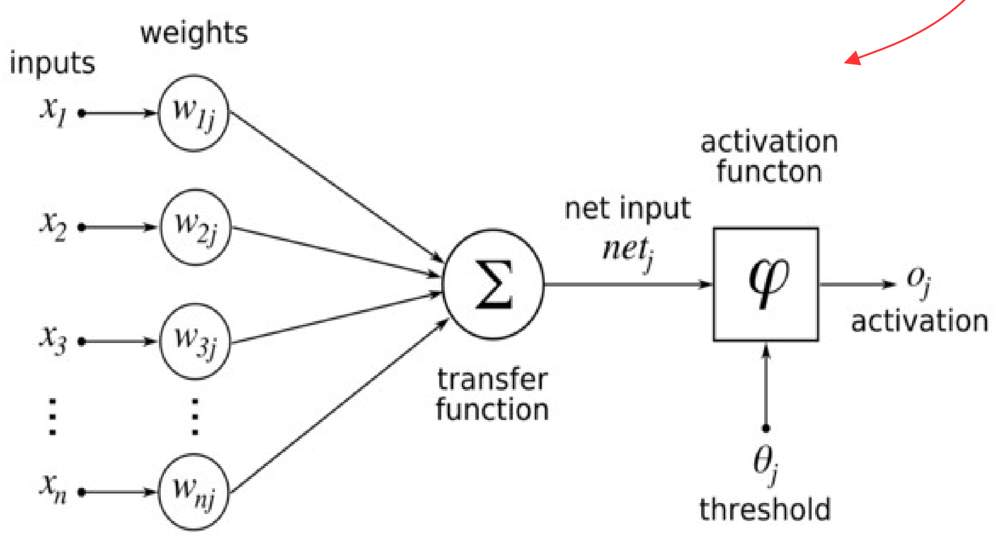
Feature Vector		Size	X-bar	Y-bar	Orientation	Perimeter	
V_1(v1,..., v5)	T	v11	v12	v13	v14	v15	From Fig1(b)
V_2(v1,..., v5)	L	v21	v22	v23	v24	v25	From Fig2(b)

Harry Li, PhD. SJSU

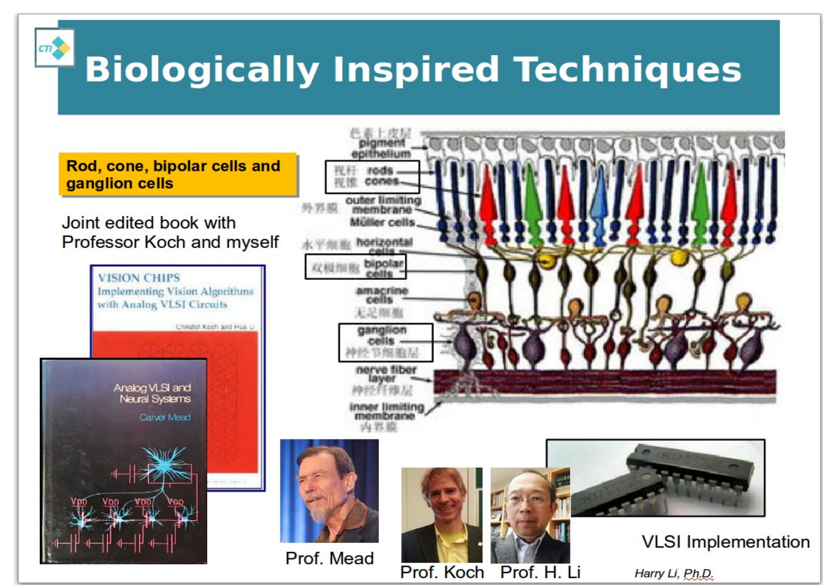
Kernel Coefficients to Neural Nets



w_11	w_12	w_13
w_21	w_22	w_23
w_31	w_32	w_33



Neural Nets: Biological Inspirations



Harry Li, Ph.D.