Introduction to Neural Networks

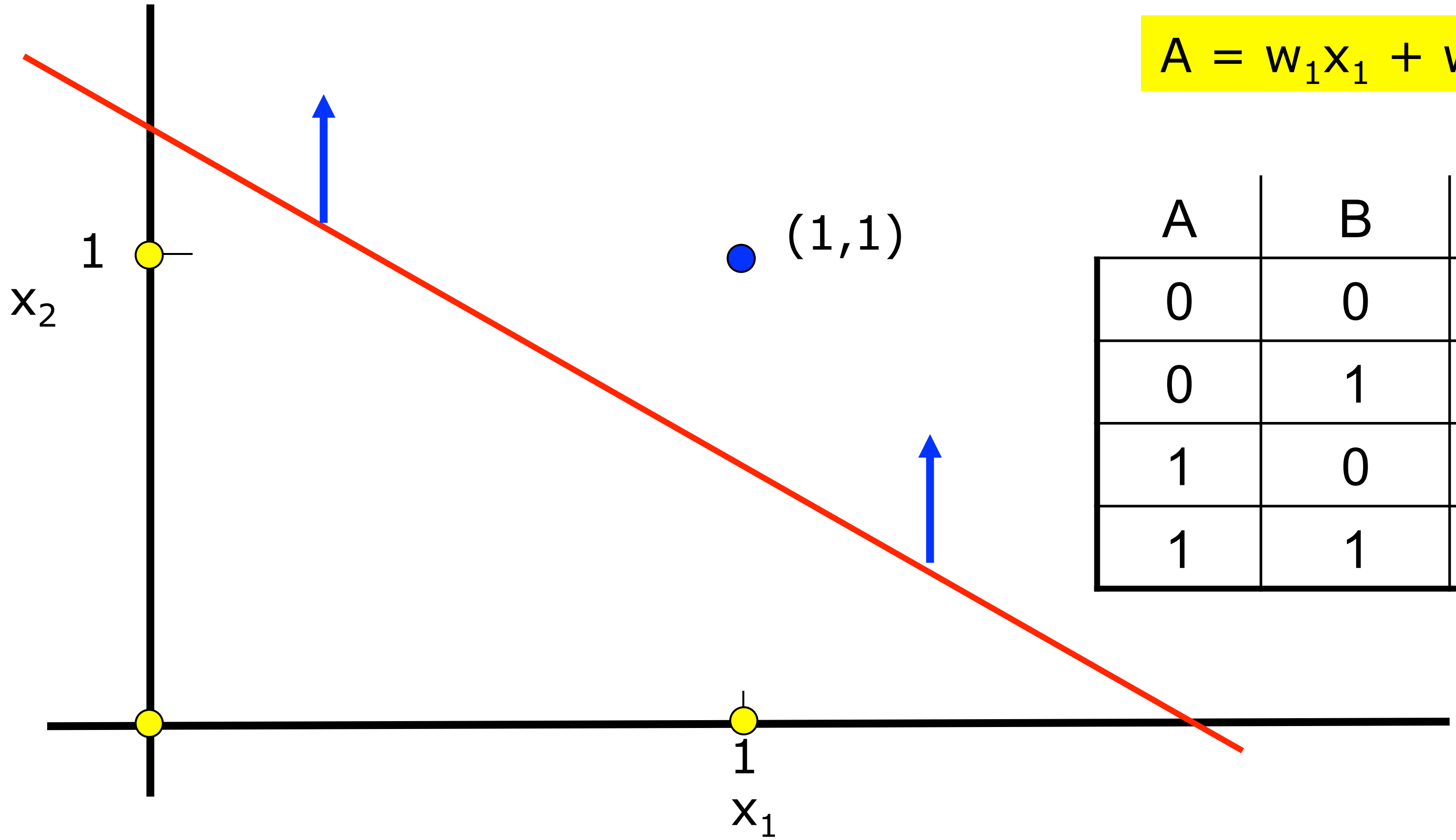
Johns Hopkins University
Engineering for Professionals Program
605-447/625-438
Dr. Mark Fleischer

Module 3.2: Perceptrons and Logic

This Sub-Module Covers …

- How Perceptrons can model logic statements.
- How Perceptron networks can model compound statements.
- Limitations on Perceptrons: The XOR problem.
- Second Order Perceptrons and the XOR problem.

A New Angle to Perceptrons



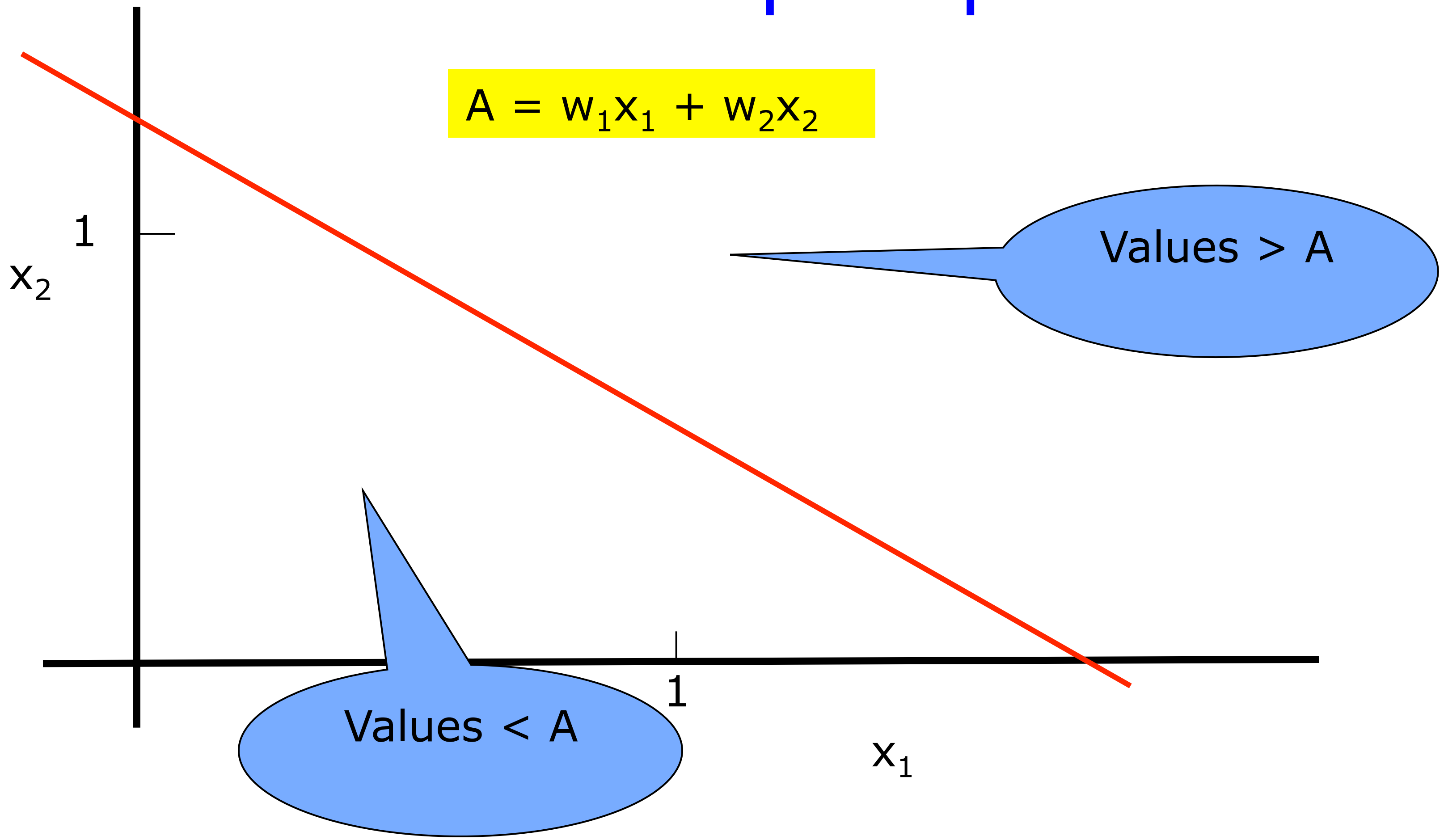
$A = w_1x_1 + w_2x_2$

A	B	A∧B
0	0	0
0	1	0
1	0	0
1	1	1

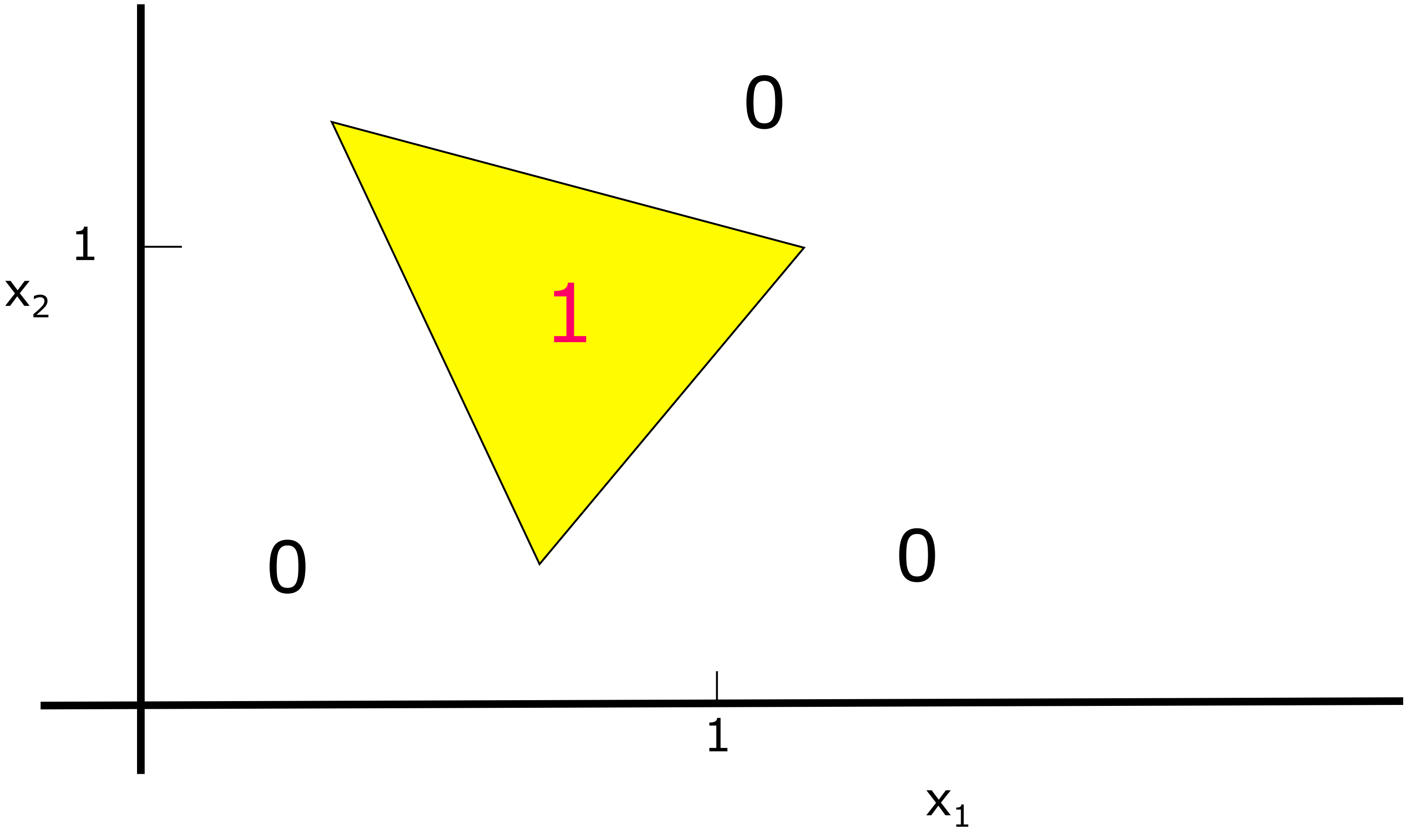
Linear Separability & Perceptrons

- Inputs x_1, x_2 …. values we use or control
- Activity $A = w_1x_1 + w_2x_2 + \theta$, a weighted function of the inputs
- A Monotonically increasing Activation Function, possibly coupled to some 'threshold logic' function.

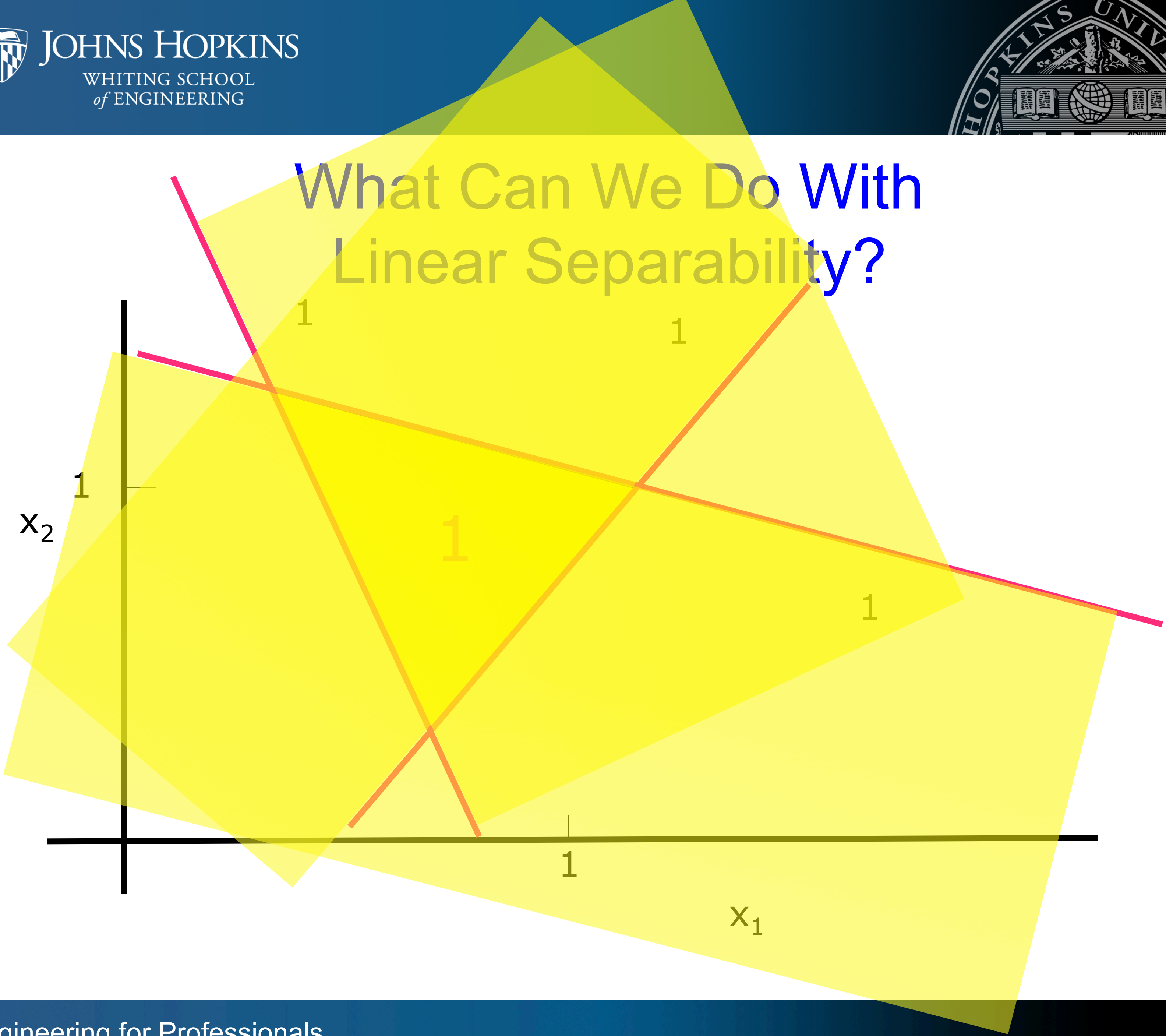
Linear Separability Bisects The Input Space



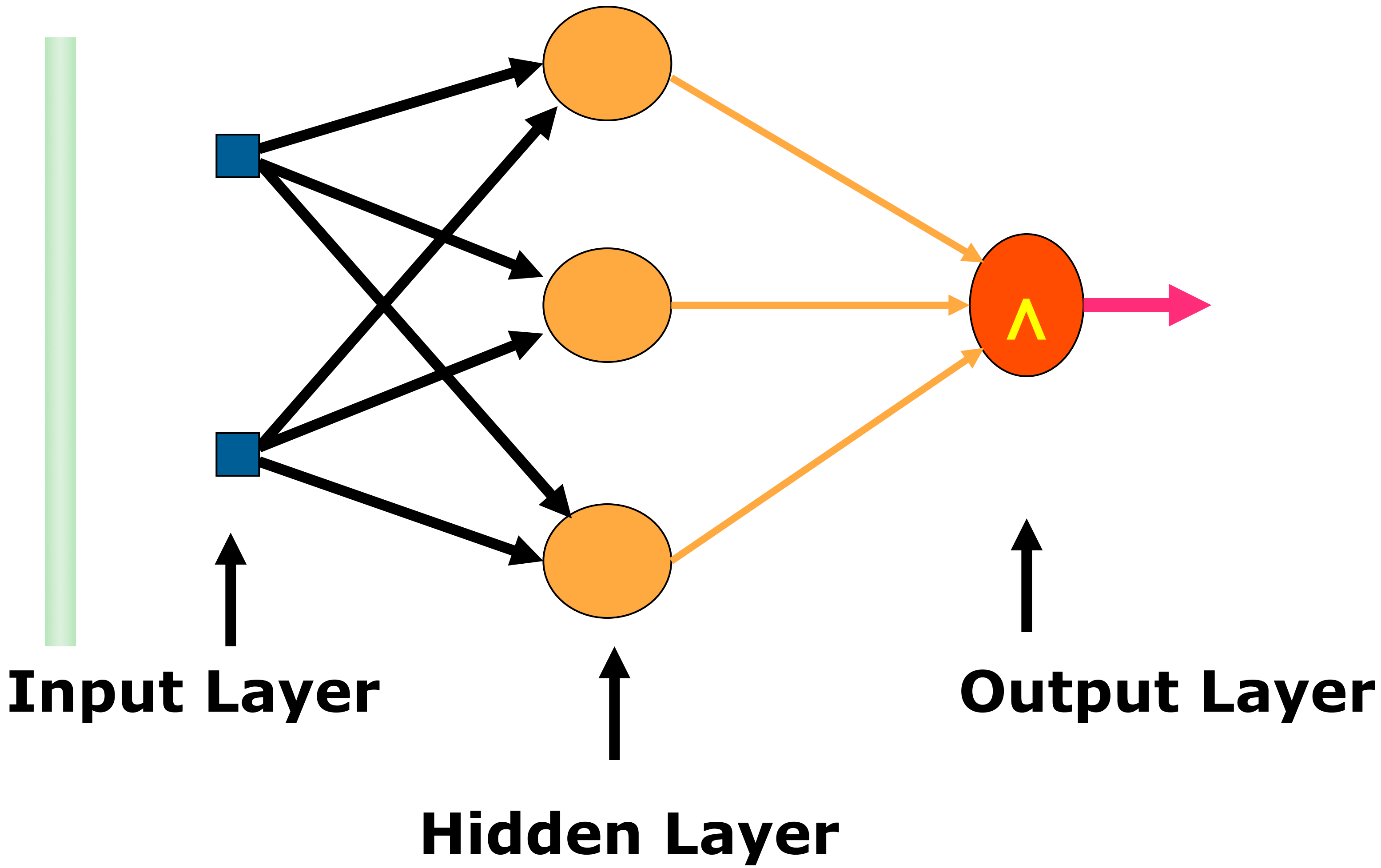
What Can We Do With Linear Separability?



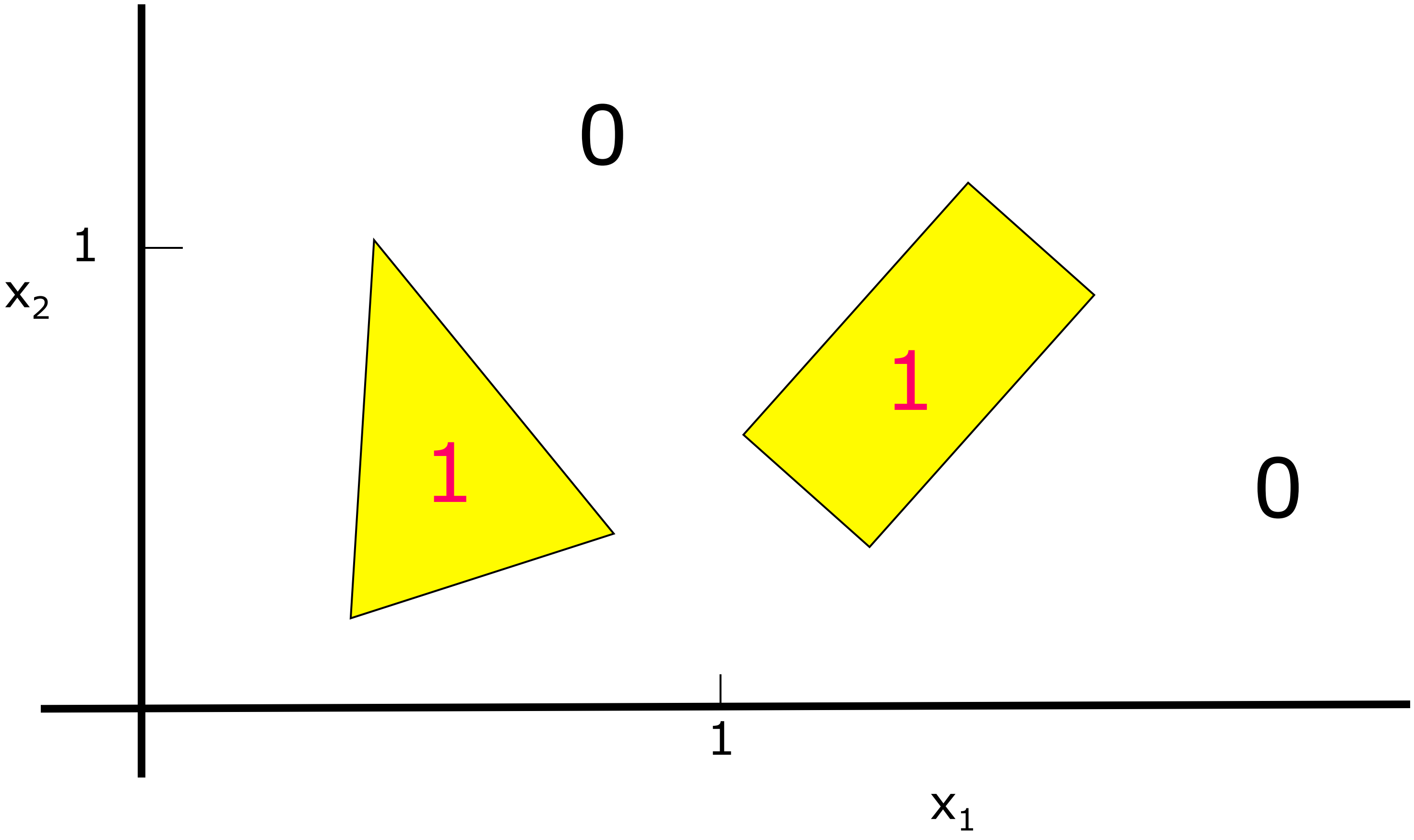
What Can We Do With Linear Separability?

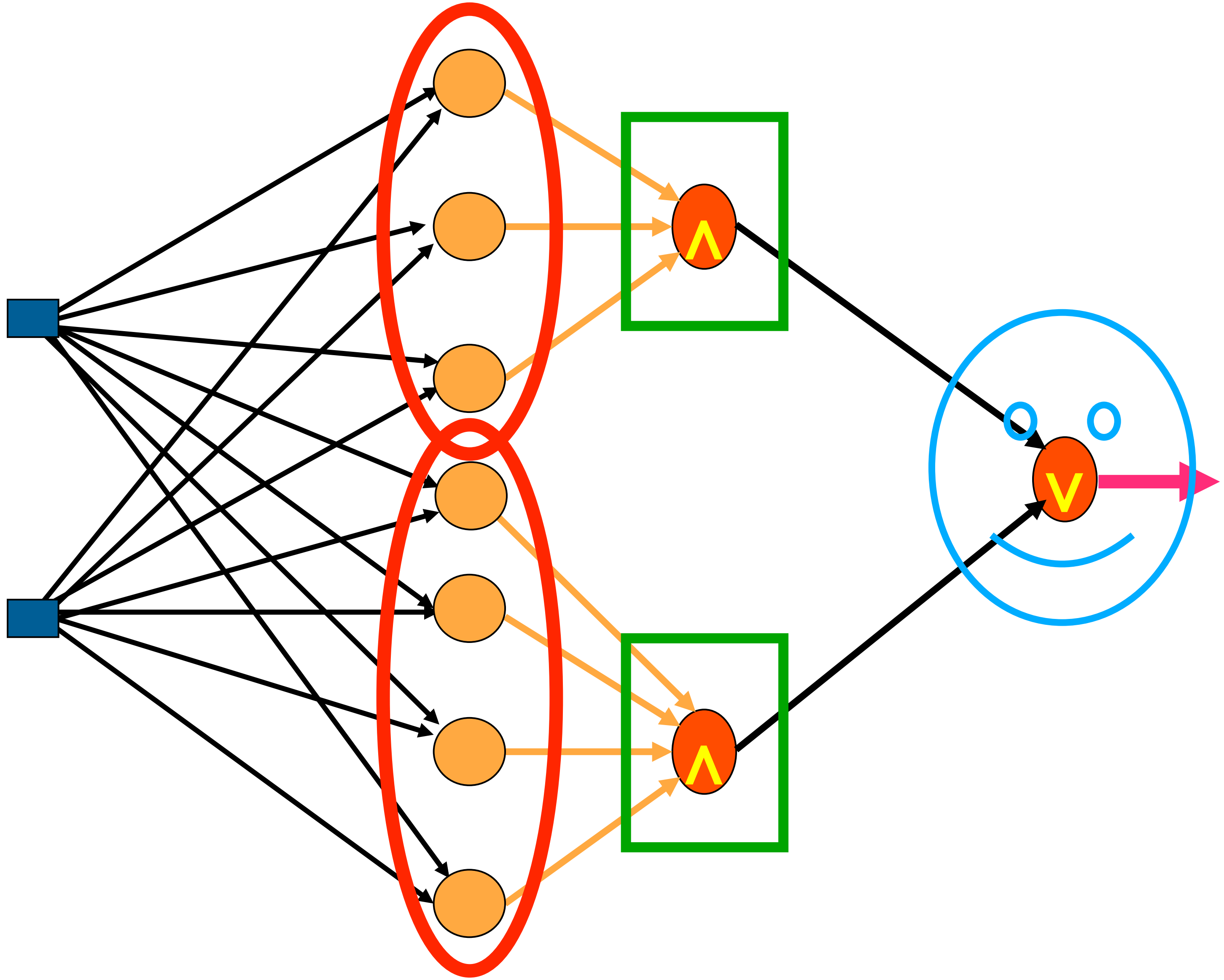


A Multi-layered Network



What Can We Do With Linear Separability?

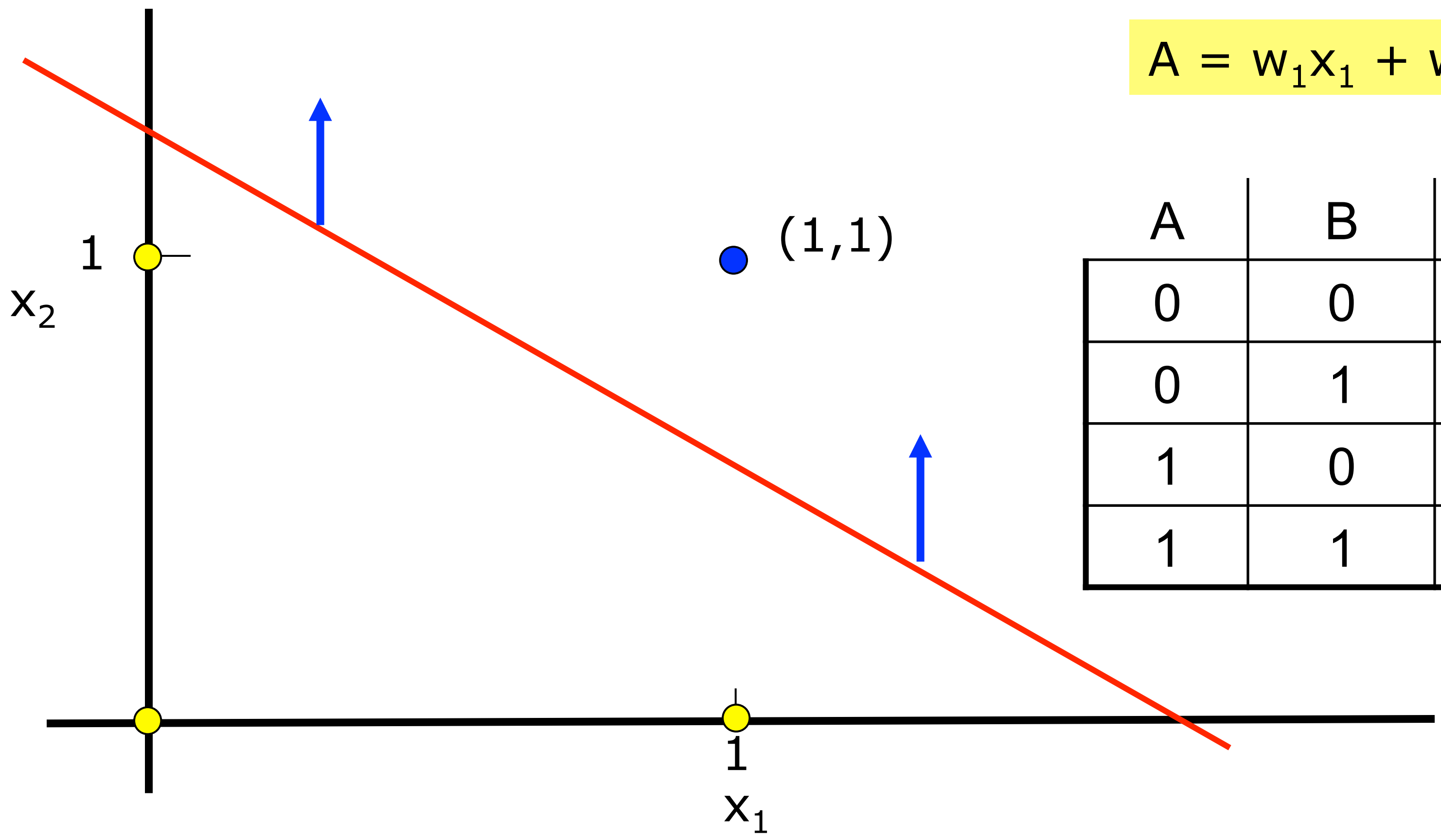




What Can We Do With Linear Separability?

- Segregate regions of the input-space
- Classification, categorization, labeling, etc.
- What do we need to do to enable this?
- Determine the weights!
- Is that all we need to do?

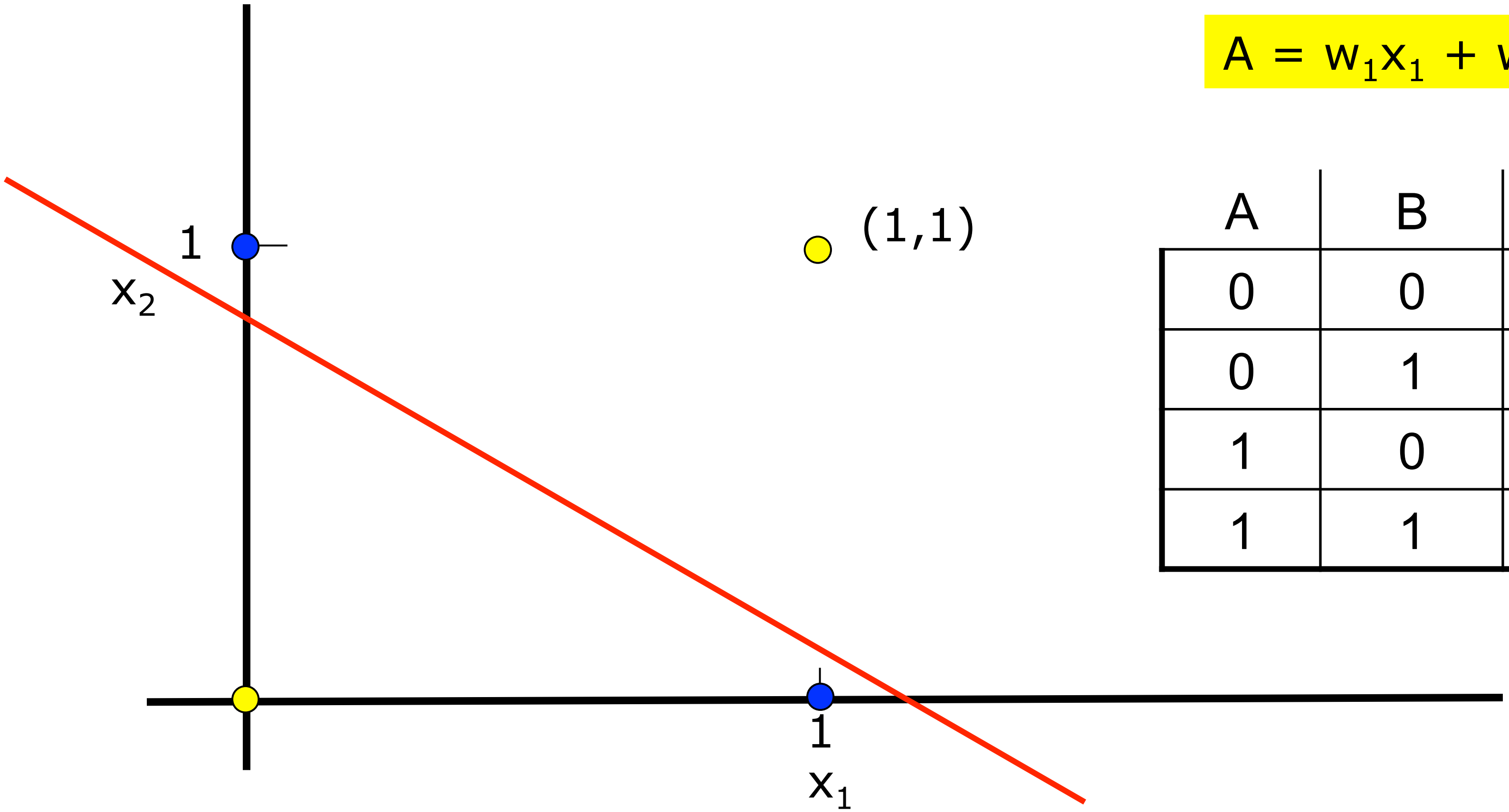
Can Do the AND and NAND



$A = w_1x_1 + w_2x_2$

A	B	A∧B
0	0	0
0	1	0
1	0	0
1	1	1

Can We Do XOR?



$A = w_1x_1 + w_2x_2$

A	B	A⊗B
0	0	0
0	1	1
1	0	1
1	1	0

Still, We Can't Solve XOR

With a Single Perceptron

$w_1x_1 + w_2x_2 + B = A$

x_1	x_2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

$0 + 0 + B < 0$

$0 + w_2x_2 + B >= 0$

$w_1x_1 + 0 + B >= 0$

$w_1x_1 + w_2x_2 + B < 0$

Adding together the two middle rows we get:

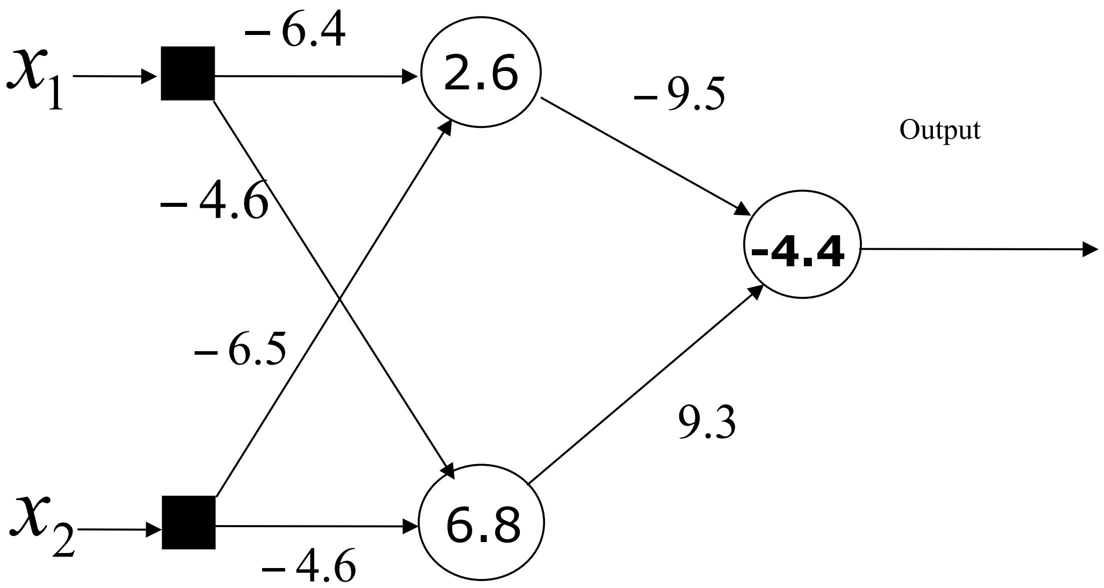
$w_1x_1 + w_2x_2 + 2B >= 0$

Adding together the first and last rows we get:

$w_1x_1 + w_2x_2 + 2B < 0$

Does there exist values of w_1 and w_2 that can yield this?
Is there any combination of values for w_1 and w_2?

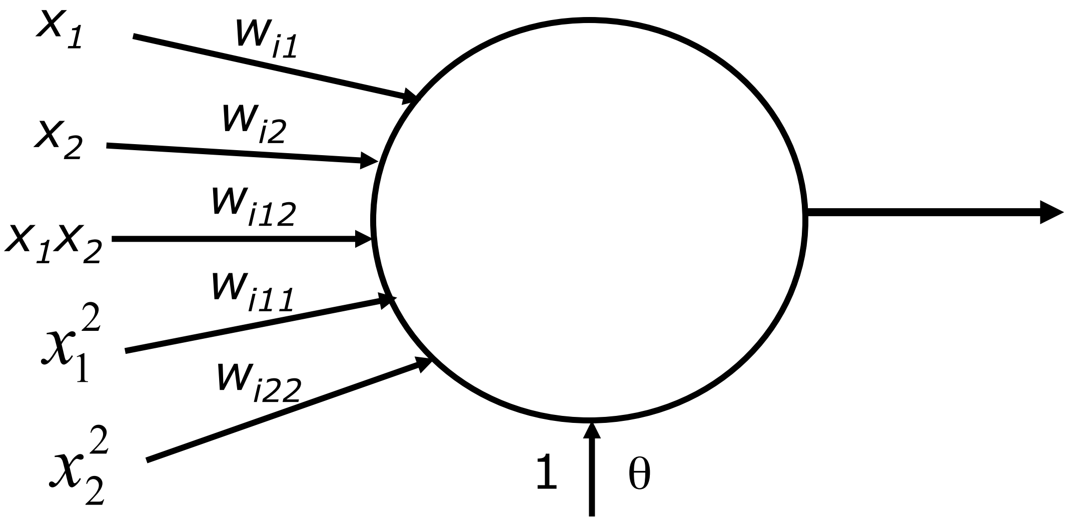
An Example of the XOR Problem



An Example of the XOR Problem

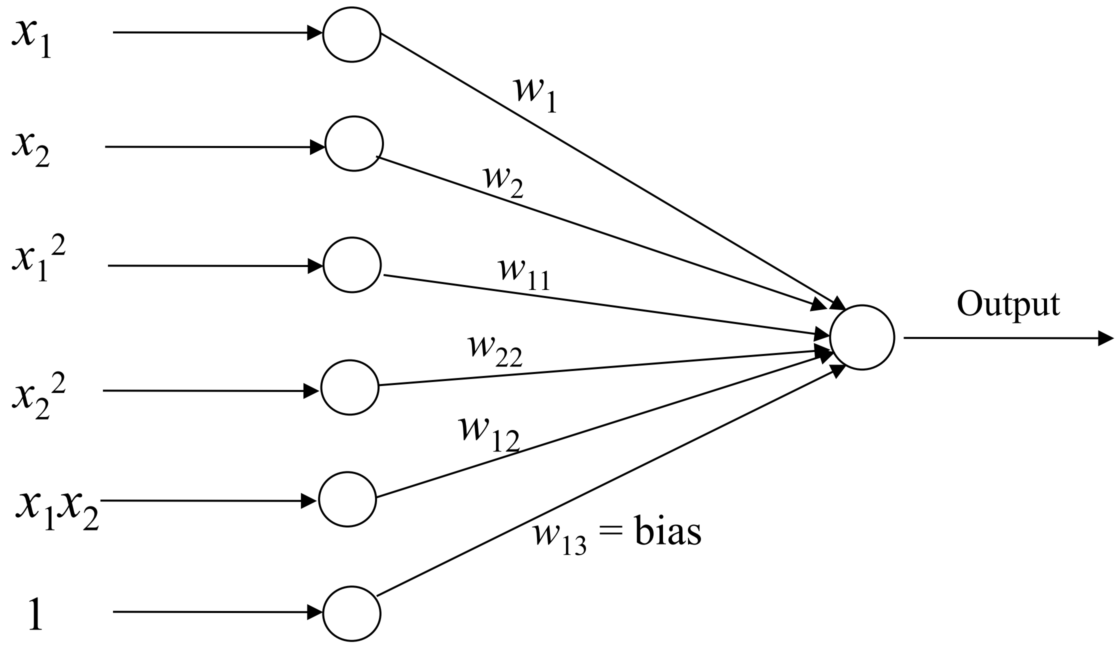
x_1	x_2	**I to HN**	**I + b**	**HN out**	**Input to Output Node**	**I + b**	**Output**
0	0	(0,0)	(2.6,6.8)	(0.93, 1.0)	0.465	-4	0.2->0
0	1	(-6.4,-4.6)	(-3.8,2.2)	(0.02,0.9)	8.37	3.96	0.98->1
1	0	(-4.6,-6.4)	<->	<->	<->	<->	<->
1	1	(-12.9,-9.2)	(-10.3,-2.4)	(0.0,0.08)	0.77	.77-4.4 =-3.6	0.03->0

A Second-Order Perceptron



$$y_i = \sum_{j=1} w_{ij} x_j + \sum_{\substack{j=1 \\ k=1}} w_{ikj} x_k x_j + \theta$$

XOR and 2nd Order Perceptron

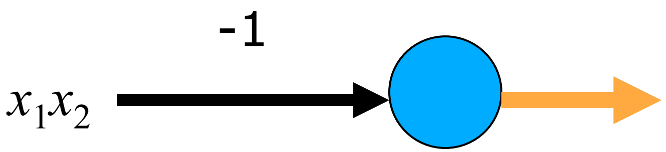


$$A = \sum_{j=1} w_{ij} x_j + \sum_{\substack{j=1 \\ k=1}} w_{ikj} x_k x_j + \theta$$

If we change the alphabet to 'bipolar' values of -1 and 1 AND set w_{12} = -1, then this can solve XOR.

XOR

Input		Output
x_1	x_2	
-1	-1	-1
-1	1	1
1	-1	1
-1	-1	-1



How do we set the weights in a complicated network like this?