Neural Network: Perceptron Example

CPSC 433: Artificial Intelligence
Fall 2024

Jonathan Hudson, Ph.D.
Assistant Professor (Teaching)
Department of Computer Science
University of Calgary

August 8, 2024

UNIVERSITY OF CALGARY

Outlines

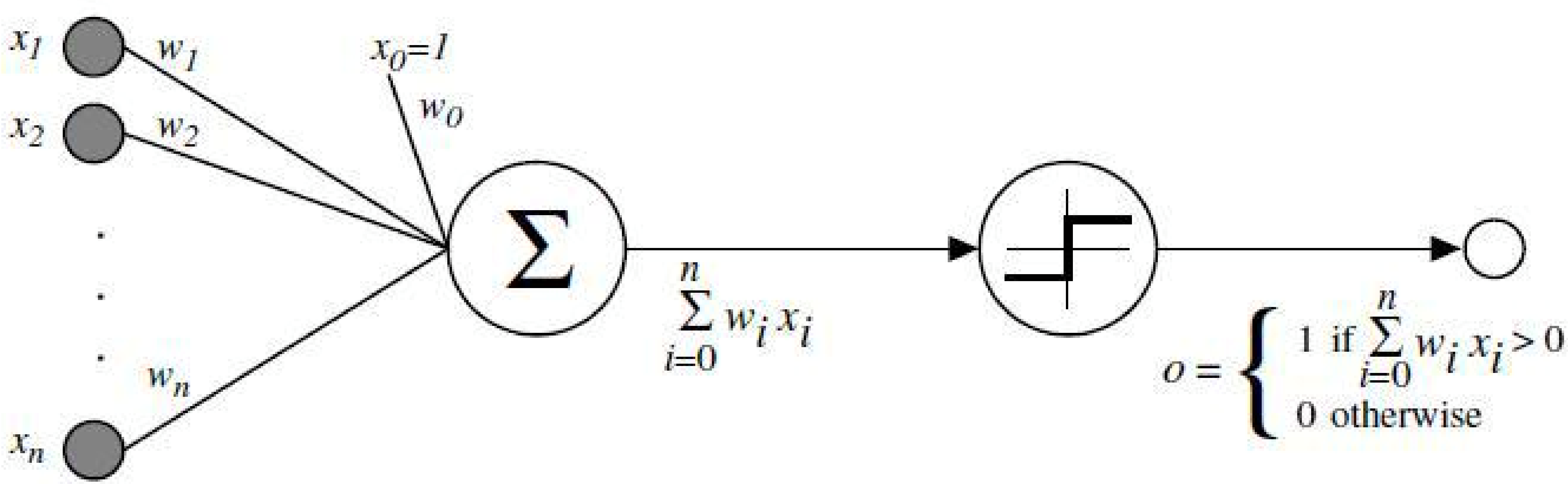
- Perceptron

Perceptron

Not a Dr. Who Robot (or maybe it is?)

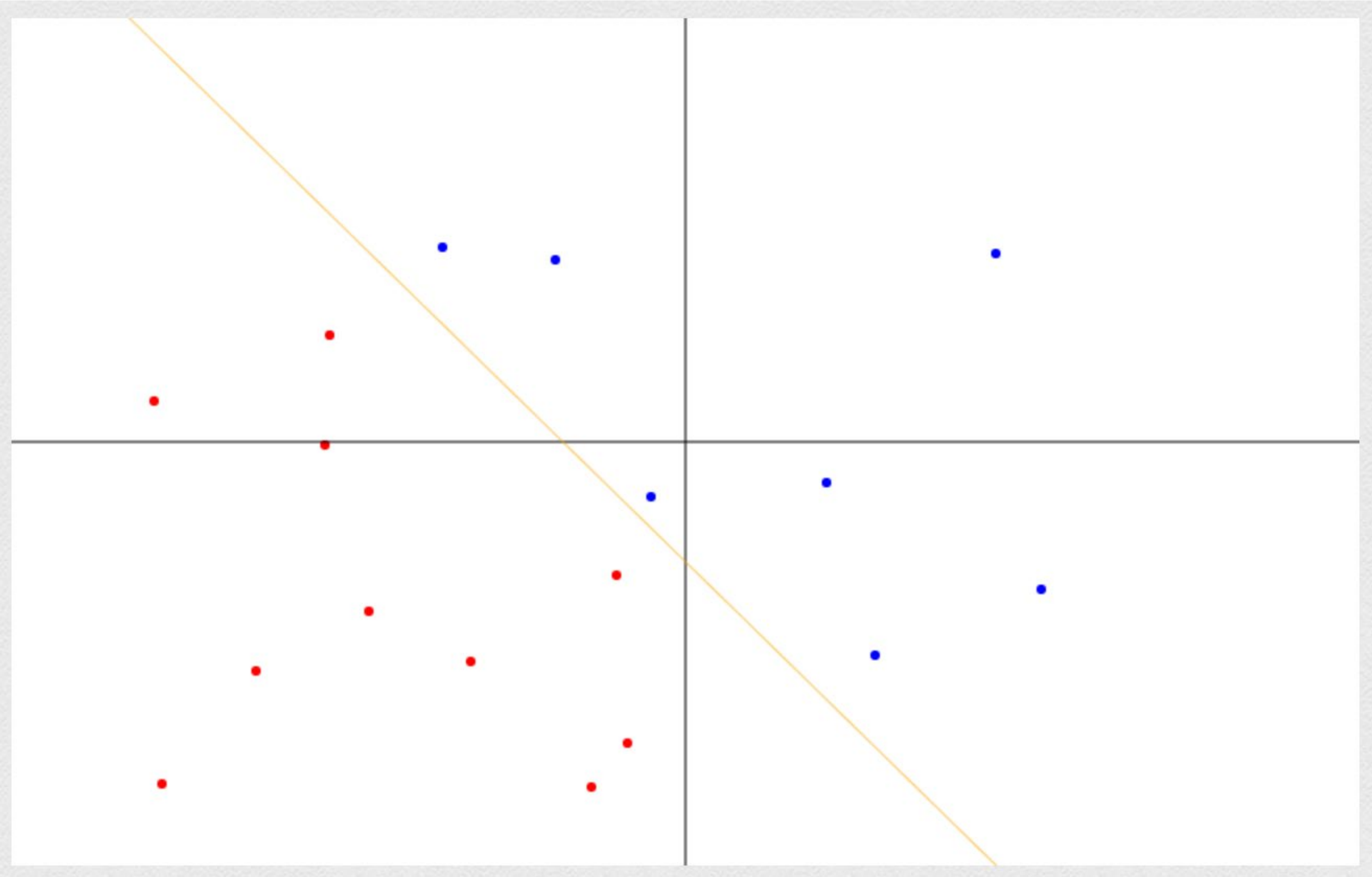
Perceptron

- Invented by Frank Rosenblatt (1957): simplified mathematical model of how the neurons in our brains operate


x_1
w_1
x_2
w_2
$x_0=1$
w_0
Σ
x_n
w_n
$\sum_{i=0}^{n} w_i x_i$
$o = \begin{cases} 1 \text{ if } \sum_{i=0}^{n} w_i x_i > 0 \\ 0 \text{ otherwise} \end{cases}$

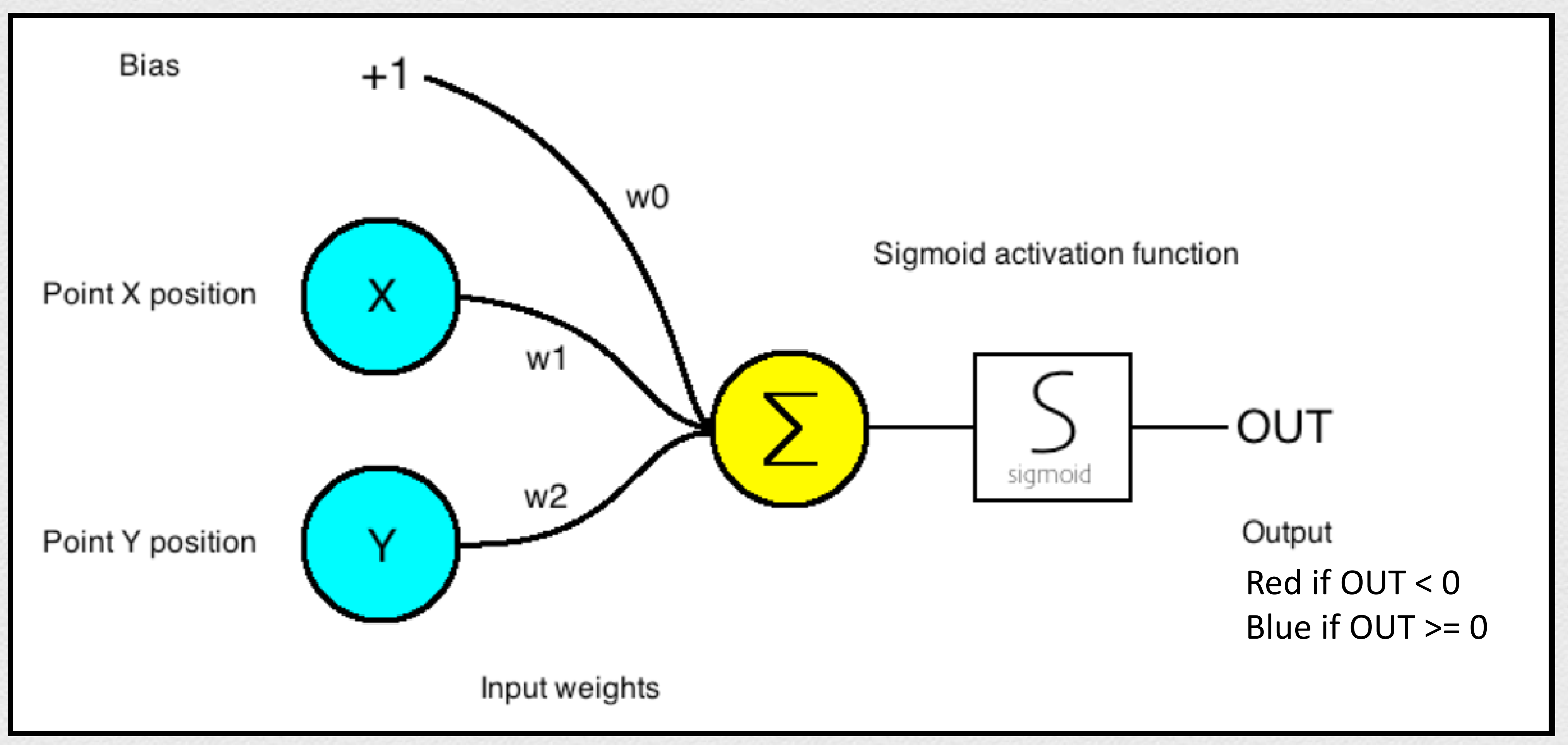
From: http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning/

Perceptron



https://vgarciasc.github.io/perceptron-viz/

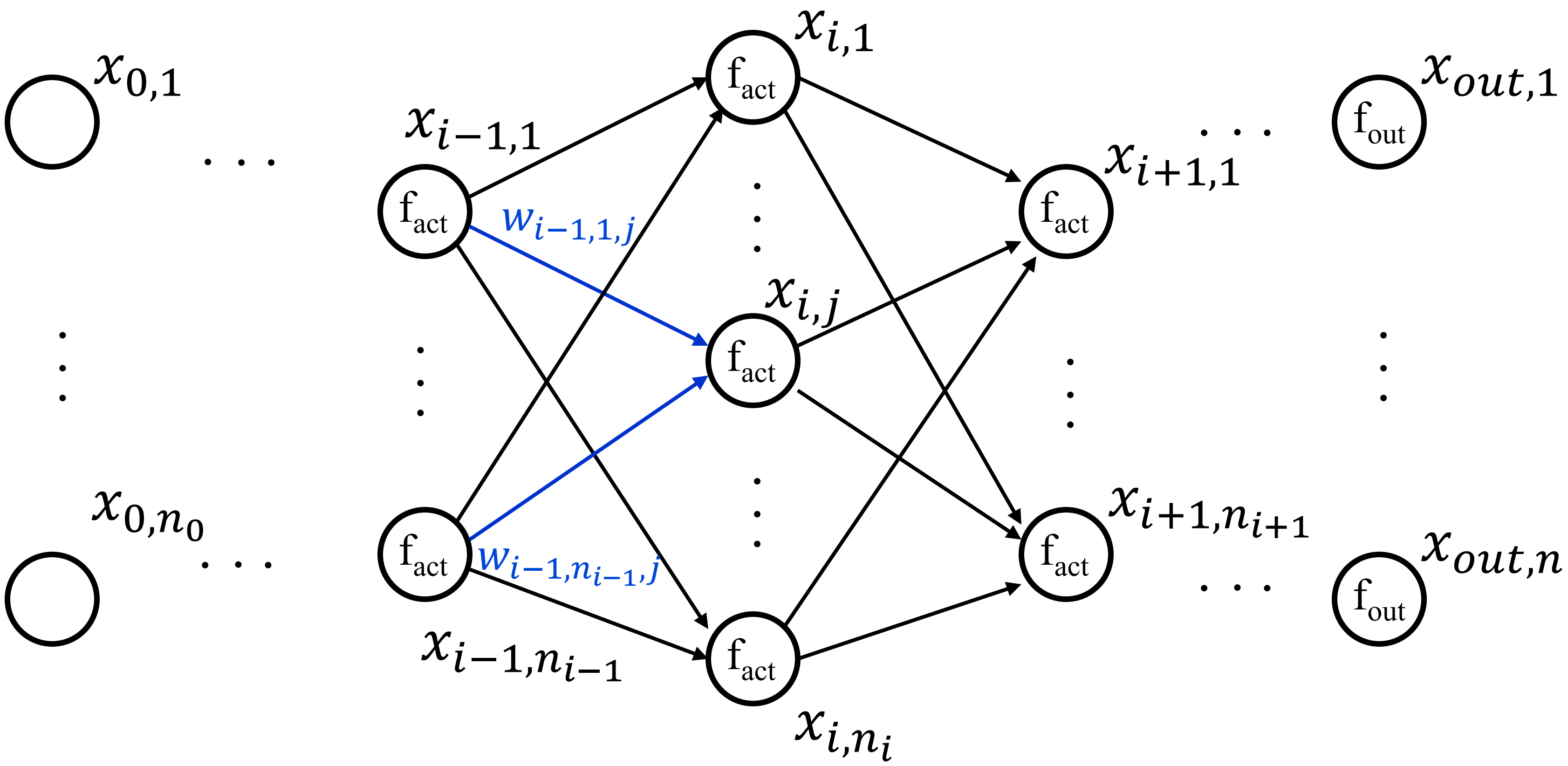
Perceptron



Red if OUT < 0

Blue if OUT >= 0

Basic data structures (reminder)



$$x_{i,j} = f_{act}\left(x_{i-1,1} * w_{i-1,1,j}, \ldots, x_{i-1,n_{i-1}} * w_{i-1,n_{i-1},j}\right)$$

Example: Perceptron (I)

- No hidden layers

- f_{act} $(a_1,\ldots,a_m) = sig(a_1+\ldots+a_m)$

- $f_{learn}(w_{0j1,old},d,x_{01},\ldots,x_{0k},c) = \ w_{0j1,old} + c*(d - sig(\sum x_{0i}*w_{0i1}))*x_{0j}$
-

  where d is the expected output of the web

- Usually there are more inputs to a perceptron than what is suggested by the function to learn: bias-nodes allow for more learning accuracy

Example: Perceptron (II)

Perform the learning for a perceptron with two input nodes and one bias node with constant value of 1 and a learning rate c = 0.3

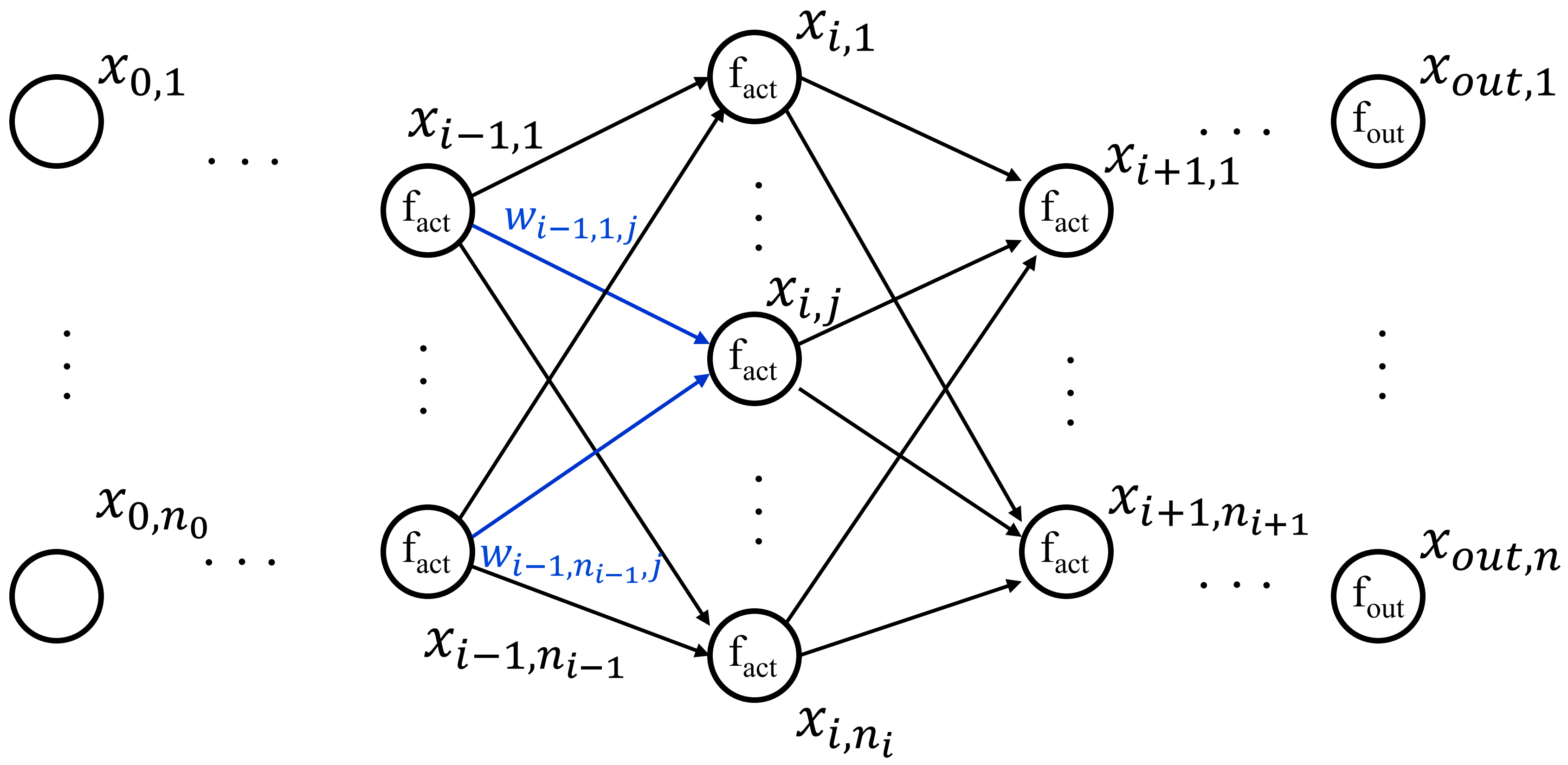
Let the initial weights be

$w_{011} = 0.3$; $w_{021} = 0.4$; $w_{031} = -0.2$

Training data:

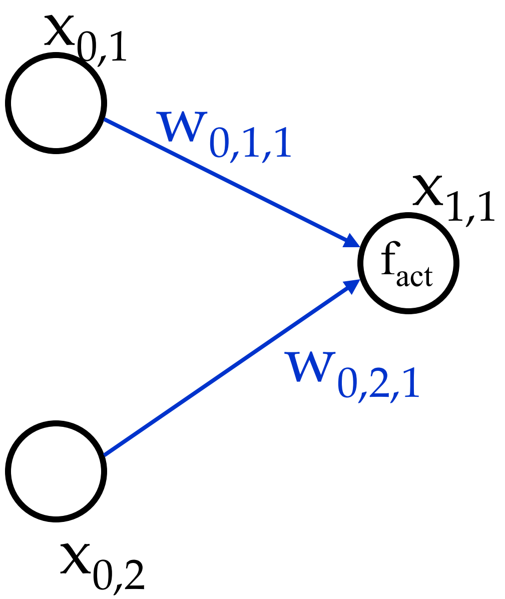
(1,1) -> 1; (9.4,6.4) -> -1; (8,7.7) -> -1; (0.5,2.2) -> 1

Basic data structures (start)



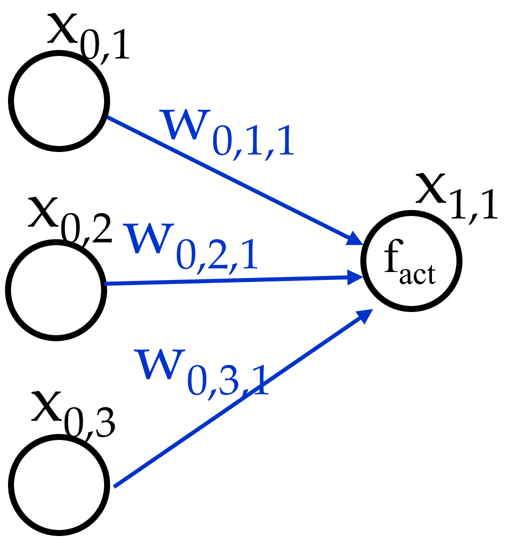
$$x_{i,j} = f_{act}\left(x_{i-1,1} * w_{i-1,1,j}, \dots, x_{i-1,n_{i-1}} * w_{i-1,n_{i-1},j}\right)$$

Basic data structures (two incoming)



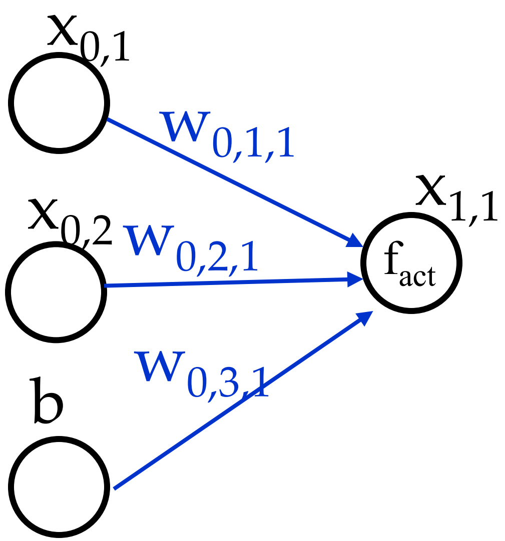
$$x_{11} = f_{act}(x_{0,1}*w_{0,1,1},x_{0,2}*w_{0,2,1})$$

Basic data structures (three incoming)



$x_{11} = f_{act}(x_{0,1}*w_{0,1,1},x_{0,2}*w_{0,2,1} ,x_{0,3}*w_{0,3,1})$

Basic data structures (like exmaple)



$$x_{11} = f_{act}(x_{0,1}*w_{0,1,1}, x_{0,2}*w_{0,2,1}, b*w_{0,3,1})$$

Example

Example: Perceptron (II)

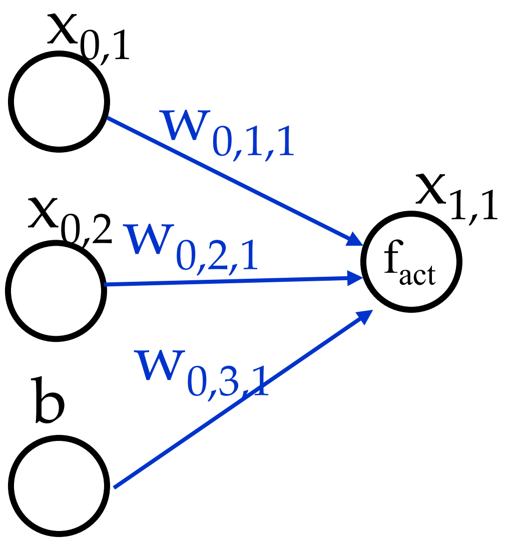
Perform the learning for a perceptron with two input nodes and one bias node with constant value of 1 and a learning rate c = 0.3

Let the initial weights be

w_{011} = 0.3 ; w_{021} = 0.4 ; w_{031} = -0.2

Training data:

(1,1) -> 1; (9.4,6.4) -> -1; (8,7.7) -> -1; (0.5,2.2) -> 1



Example: Perceptron (II)

Perform the learning for a perceptron with two input nodes and one bias node with constant value of 1 and a learning rate c = 0.3

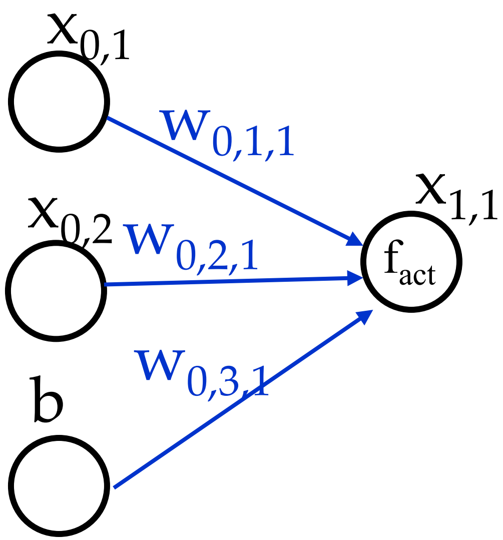
Let the initial weights be

$w_{011} = 0.3$; $w_{021} = 0.4$; $w_{031} = -0.2$

Training data:

(1,1) -> 1; (9.4,6.4) -> -1; (8,7.7) -> -1; (0.5,2.2) -> 1

- $f_{learn}(w_{0j1,old},d,x_{01},\ldots,x_{0k},c) = w_{0j1,old} + c*(d - sig(\sum x_{0i}*w_{0i1}))*x_{0j}$



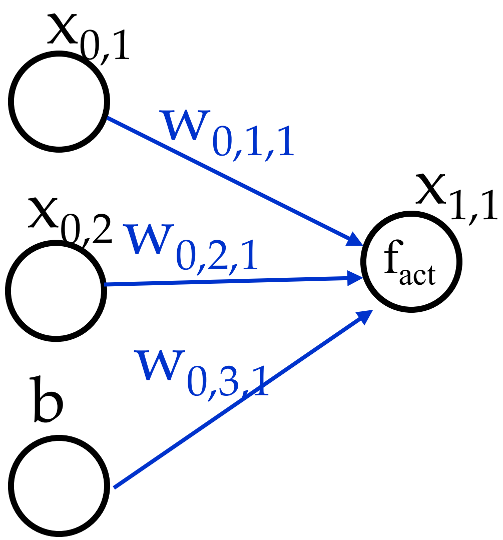
Example: Perceptron (II)

Perform the learning for a perceptron with two input nodes and one bias node with constant value of 1 and a learning rate c = 0.3

Let the initial weights be
$w_{011} = 0.3$; $w_{021} = 0.4$; $w_{031} = -0.2$

Training data:
(1,1) -> 1; (9.4,6.4) -> -1; (8,7.7) -> -1; (0.5,2.2) -> 1

- $w_{011,new} = w_{011,old} + c*(d - sig(\sum x_{0i}*w_{0i1}))*x_{01}$
- $w_{021,new} = w_{021,old} + c*(d - sig(\sum x_{0i}*w_{0i1}))*x_{02}$
- $w_{031,new} = w_{031,old} + c*(d - sig(\sum x_{0i}*w_{0i1}))*b$



Example: Perceptron (II)

Perform the learning for a perceptron with two input nodes and one bias node with constant value of 1 and a learning rate c = 0.3

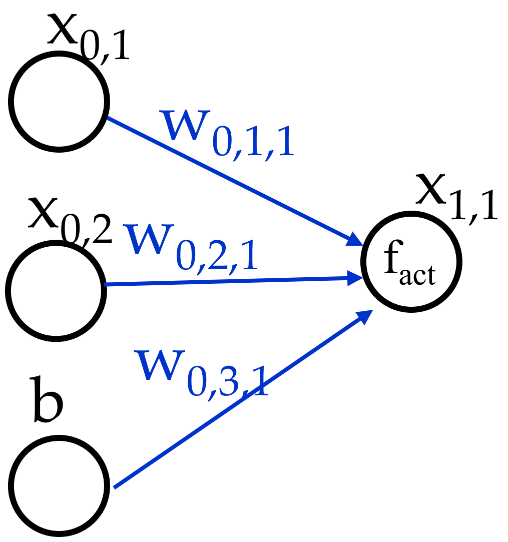
Let the initial weights be

$w_{011} = 0.3$; $w_{021} = 0.4$; $w_{031} = -0.2$

Training data:

(1,1) -> 1; (9.4,6.4) -> -1; (8,7.7) -> -1; (0.5,2.2) -> 1

- $w_{011,new} = 0.3 + 0.3 * (1 - \text{sig}(\sum x_{0i} * w_{0i1})) * 1$
- $w_{021,new} = 0.4 + 0.3 * (1 - \text{sig}(\sum x_{0i} * w_{0i1})) * 1$
- $w_{031,new} = -0.2 + 0.3 * (1 - \text{sig}(\sum x_{0i} * w_{0i1})) * 1$



Example: Perceptron (II)

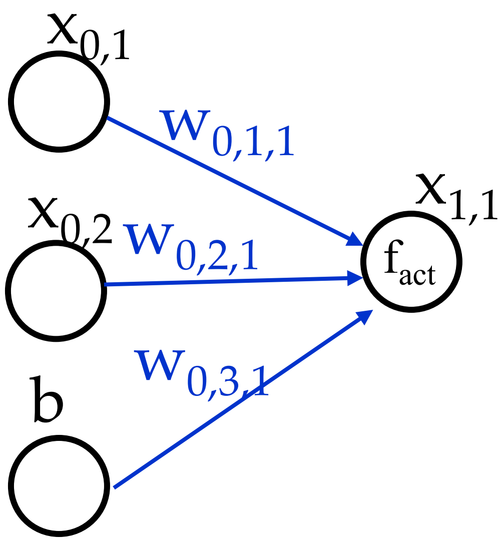
Perform the learning for a perceptron with two input nodes and one bias node with constant value of 1 and a learning rate c = 0.3

Let the initial weights be
w_{011} = 0.3 ; w_{021} = 0.4 ; w_{031} = -0.2

Training data:

(1,1) -> 1; (9.4,6.4) -> -1; (8,7.7) -> -1; (0.5,2.2) -> 1

- $w_{011,new} = 0.3 + 0.3 *(1 - \mathrm{sig}(\sum x_{0i} * w_{0i1}))*1$
- $w_{021,new} = 0.4 + 0.3 *(1 - \mathrm{sig}(\sum x_{0i} * w_{0i1}))*1$
- $w_{031,new} = -0.2 + 0.3 *(1 - \mathrm{sig}(\sum x_{0i} * w_{0i1}))*1$
- $\mathrm{sig}(\sum x_{0i} * w_{0i1}) = \mathrm{sig}(0.3*1 + 0.4*1 + -0.2*1) = \mathrm{sig}(0.5) = 1$



Heaviside step function

Example: Perceptron (II)

Perform the learning for a perceptron with two input nodes and one bias node with constant value of 1 and a learning rate c = 0.3

Let the initial weights be

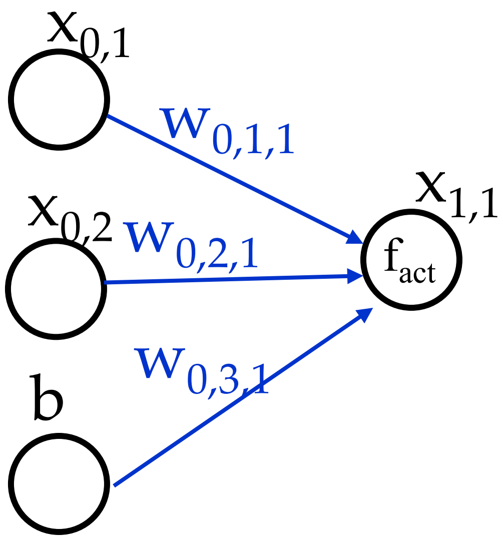
w_{011} = 0.3 ; w_{021} = 0.4 ; w_{031} = -0.2

Training data:

(1,1) -> 1; (9.4,6.4) -> -1; (8,7.7) -> -1; (0.5,2.2) -> 1

- $w_{011,new} = 0.3 + 0.3 *(1 - 1)*1 = 0.3$
- $w_{021,new} = 0.4 + 0.3 *(1 - 1)*1 = 0.4$
- $w_{031,new} = -0.2 + 0.3 *(1 - 1)*1 = -0.2$
- $sig(\sum x_{0i} * w_{0i1}) = sig(0.3*1 + 0.4*1 + -0.2*1) = sig(0.5) = 1$

Heaviside step function



Example: Perceptron (II)

Perform the learning for a perceptron with two input nodes and one bias node with constant value of 1 and a learning rate c = 0.3

Let the initial weights be

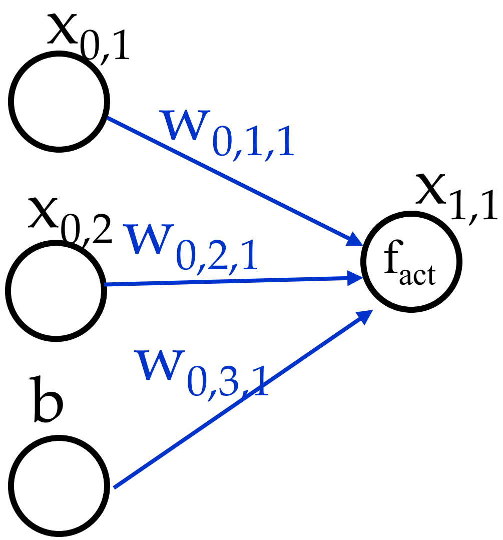
$w_{011} = 0.3$; $w_{021} = 0.4$; $w_{031} = -0.2$

Training data:

(1,1) -> 1; (9.4,6.4) -> -1; (8,7.7) -> -1; (0.5,2.2) -> 1

- $w_{011,new} = 0.3 + 0.3 *(-1 - 1)*9.4 = -5.34$
- $w_{021,new} = 0.4 + 0.3 *(-1 - 1)*6.4 = -3.44$
- $w_{031,new} = -0.2 + 0.3 *(-1 - 1)*1 = -0.8$
- $sig(\sum x_{0i}*w_{0i1}) = sig(0.3*9.4 + 0.4*6.4 + -0.2*1) = sig(5.18) = 1$ Restart!

Heaviside step function



Example: Perceptron (II)

Perform the learning for a perceptron with two input nodes and one bias node with constant value of 1 and a learning rate c = 0.3

Let the initial weights be

w_{011} = -5.34 ; w_{021} = -3.44 ; w_{031} = -0.8

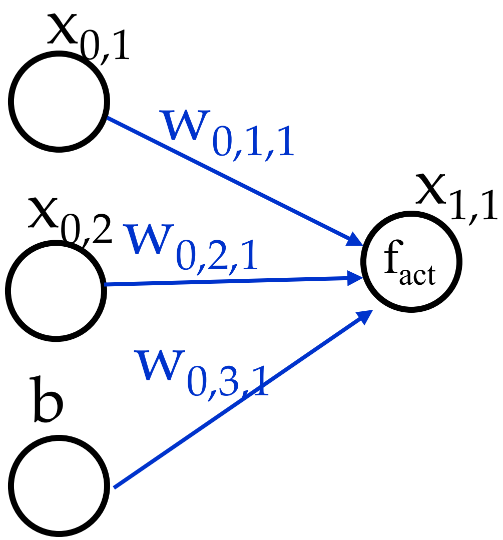
Training data:

(1,1) -> 1; (9.4,6.4) -> -1; (8,7.7) -> -1; (0.5,2.2) -> 1

- $w_{011,new}$ = -5.34 + 0.3 ∗(1 - -1)∗1 = -4.74
- $w_{021,new}$ = -3.44 + 0.3 ∗(1 - -1)∗1 = -2.84
- $w_{031,new}$ = -0.8 + 0.3 ∗(1 - -1)∗1 = -0.2
- sig(∑x_{0i}*w_{0i1}) = sig(-5.34*1 + -3.44*1 + -0.8*1) = sig(-9.58) = -1

Restart!

Heaviside step function



Example: Perceptron (Finally) (8th Try Through)

Perform the learning for a perceptron with two input nodes and one bias node with constant value of 1 and a learning rate c = 0.3

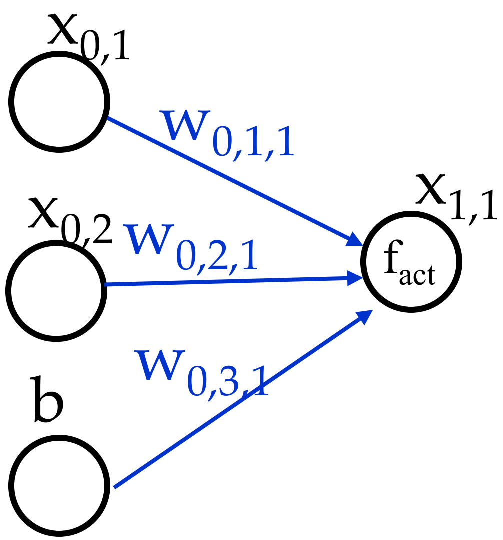
Let the initial weights be

w_{011} = -1.74 ; w_{021} = 0.16 ; w_{031} = 2.8

Training data:

(1,1) -> 1; (9.4,6.4) -> -1; (8,7.7) -> -1; (0.5,2.2) -> 1

- $w_{011,new} = -1.74 + 0.3 * (1 - 1) * 0.5 = 1.74$
- $w_{021,new} = 0.16 + 0.3 * (1 - 1) * 2.2 = 0.16$
- $w_{031,new} = 2.8 + 0.3 * (1 - 1) * 1 = 2.8$
- $sig(\sum x_{0i} * w_{0i1}) = sig(-1.74*0.5 + 0.16*2.2 + 2.8*1) = sig(2.282) = 1$



Heaviside step function

Example: Perceptron (Finally) (8th Try Through)

- Series of w011, w021, w031 values

0.30	0.40	-0.2
-5.34	-3.44	-0.8
-4.74	-2.84	-0.2
-4.14	-2.24	0.4
-3.54	-1.64	1.0
-2.94	-1.04	1.6
-2.34	-0.44	2.2
-1.74	0.16	2.8

Onward to … convolutional neural networks

Jonathan Hudson, Ph.D.
jwhudson@ucalgary.ca
https://cspages.ucalgary.ca/~jwhudson/

UNIVERSITY OF CALGARY