Data Mining and Machine Learning

Introducti Neural

Networks

Peter Jančovič

Objectives

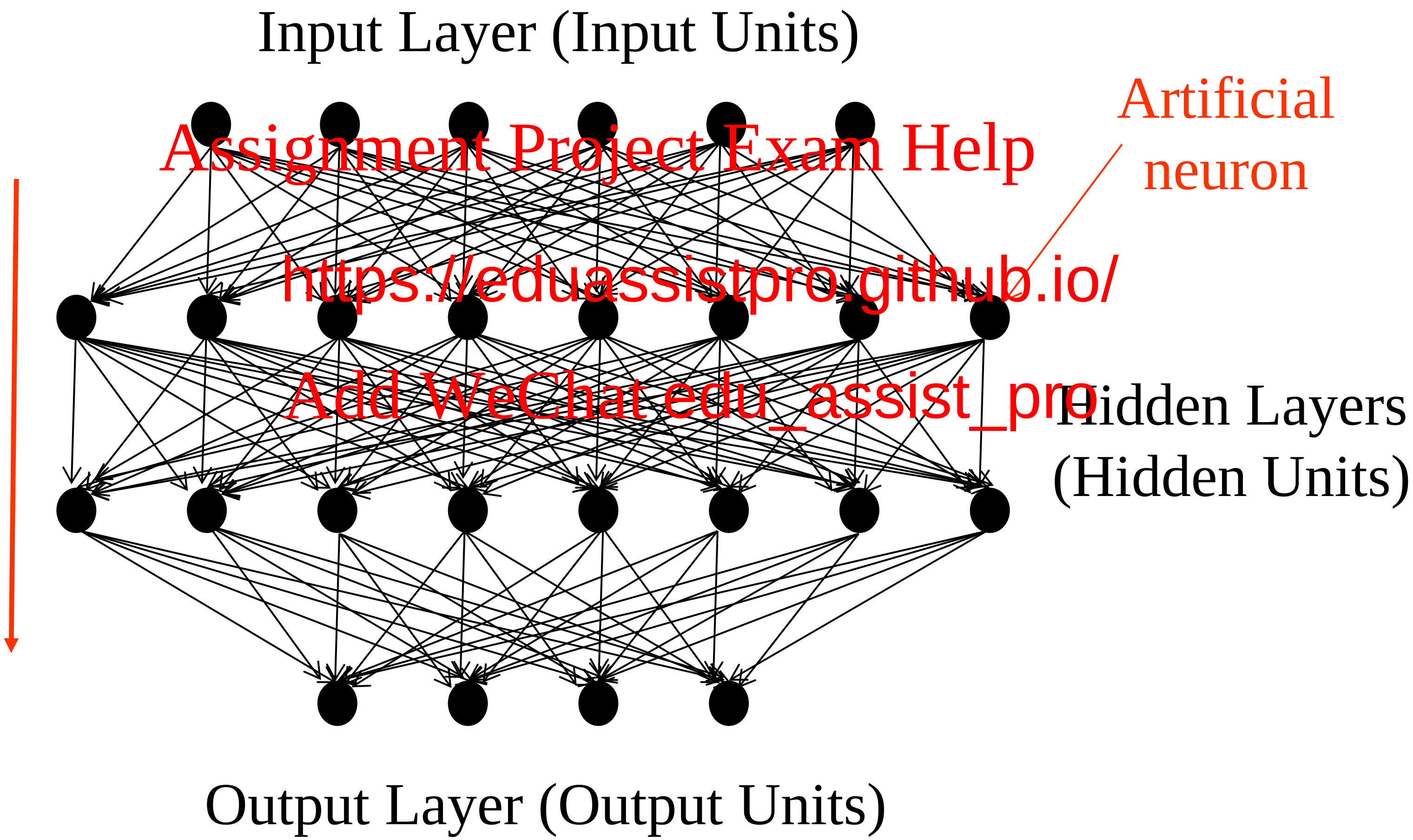
- Introduce Artificial Neural Networks (ANNs)
- Feed-forward ANNs – <u>**Multi-Layer Perceptrons**</u> (MLPs)
- Basic MLP c
- Geometric interpretation o

Artificial Neural Networks

- (Artificial) Neural Networks (NNs) offer another approach to data analysis
- Popularised in 1980s, resurgence in 2000s
- “Machine le ty “AI”) often synonymous with the use
- Inspiration for the basic el a NN (artificial neuron) comes from biology, but analogy stops there
- ANNs are just a computational device for processing patterns – not “artificial brains”

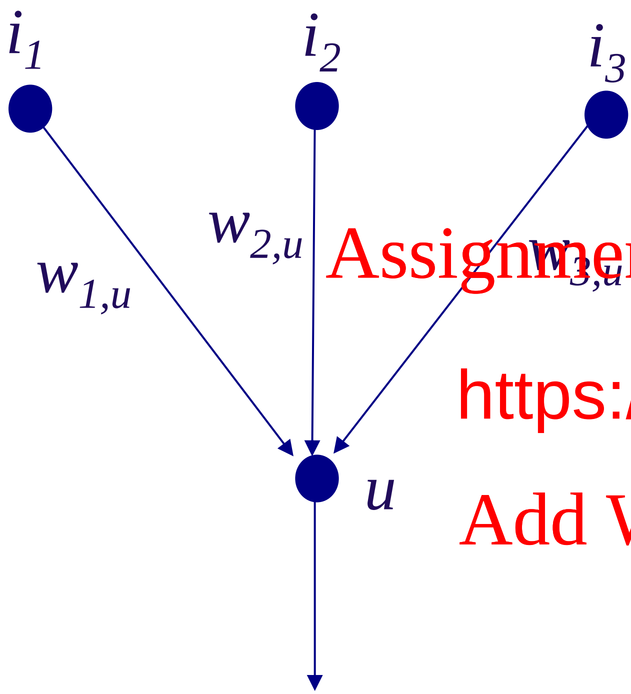
Feed-forward Neural Networks

<u>Multi-Layer Perceptron</u> - Feed-Forward Neural Network



A simple model of a neuron

A Simple Artificial Neuron

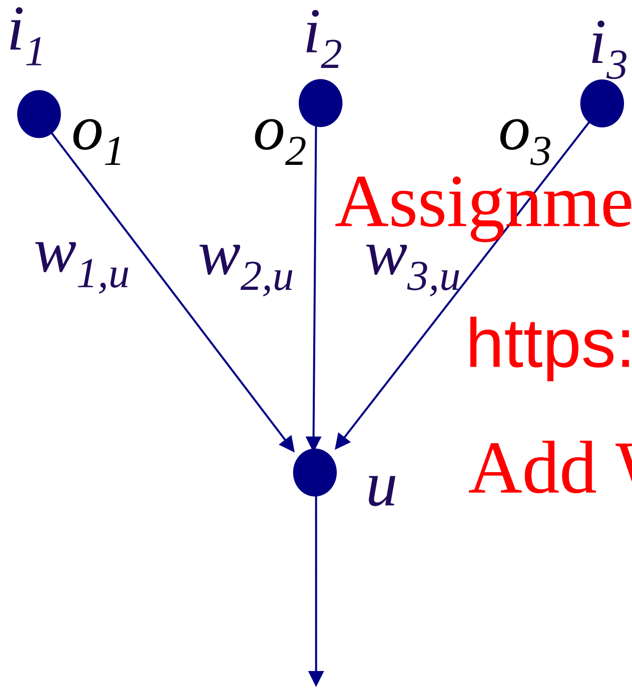


- Basic idea –
  - if the input i_u to unit u is big enough, the neurone

w

- How do we calculate the input to u?

Artificial Neurone (2)



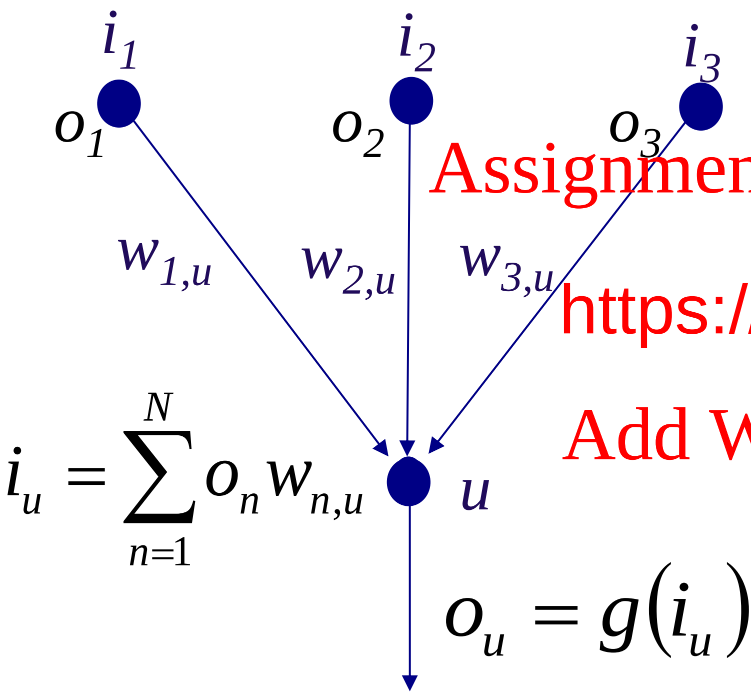
- Suppose the inputs to units 1, 2 and 3 are i_1, i_2 and i_3 and these are also the outputs o_1, o_2 and o_3
- Then the input to u is:

$$i_u = o_1 w_{1,u} + o_2 w_{2,u} + o_3 w_{3,u}$$

l, for an artificial n t receives input from N units, the input to unit u is:

$$i_u = \sum_{n=1}^{N} o_n w_{n,u}$$

The sigmoid activation function



- The activation function defines the output of a neuron - whether the neuron should "fire"
- A typical activation function is the [illegible] tion g:

$$g(x) = \frac{1}{1+e^{-kx}}$$

- The output of u is then:

$$o_u = g(i_u)$$

Activation functions

- Linear activation function (output equals input):

$$g(x) = x$$

- Sigmoid activation function:

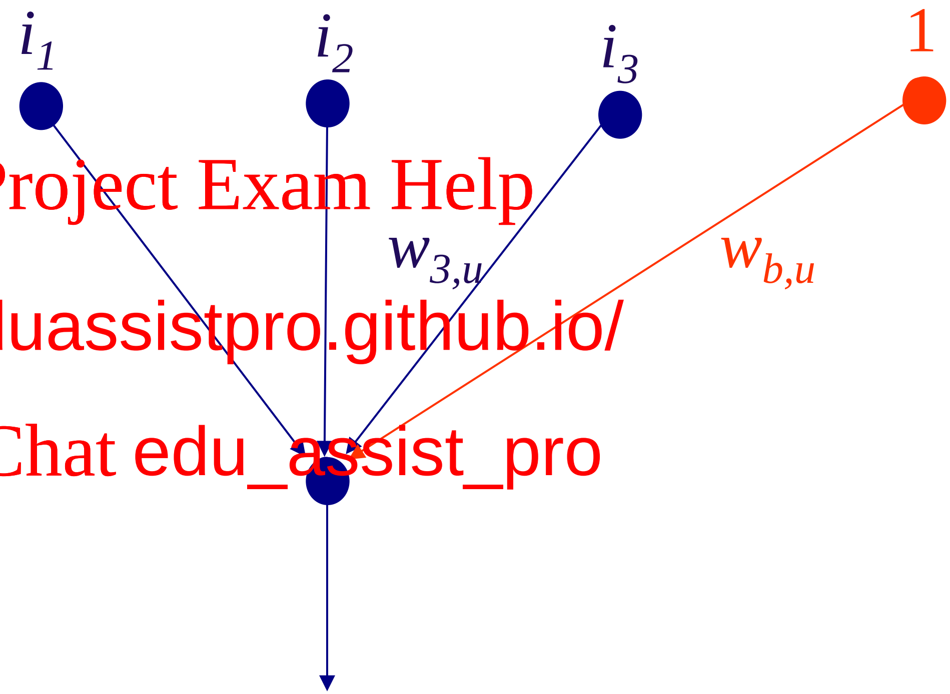
$$g(x) = \frac{1}{1 + e^{-kx}}$$

- The sigmoid is a ‘soft’ threshold function

Sigmoid activation function

The ‘bias’

- As described, the neuron will ‘fire’ only if its input is greater than 0
- We can chang value of the po firing by introducing a bias
- This is an additional input unit whose input is fixed at 1



How the bias works…

- According to the sigmoid activation function, the artificial neuron *u* ‘fires’ if the input to *u* is greater than or equal to 0

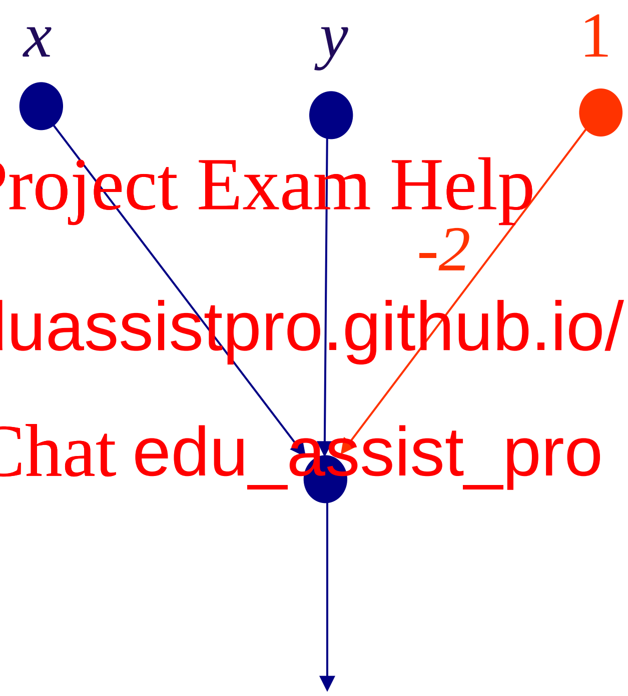
- i.e: $i_u = o_1 w_{1,u} + o_2 w_{2,u} + o_3 w_{3,u} + w_{b,u} \geq 0$

- But this happens only if

$$i_1 w_{1,u} + i_2 w_{2,u} + i_3 w_{3,u} \geq -w_{b,u}$$

Example (2D)

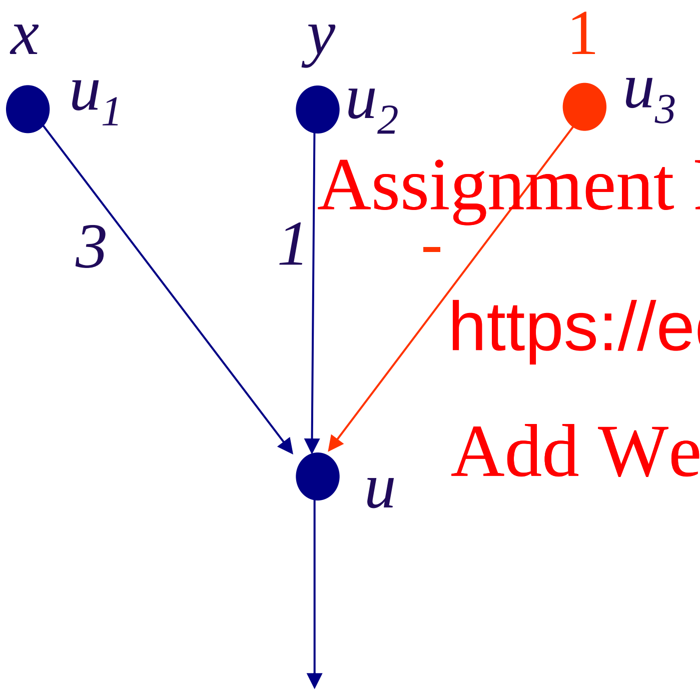
- Suppose u has a sigmoid activation function. Then, for these values of weight, u will

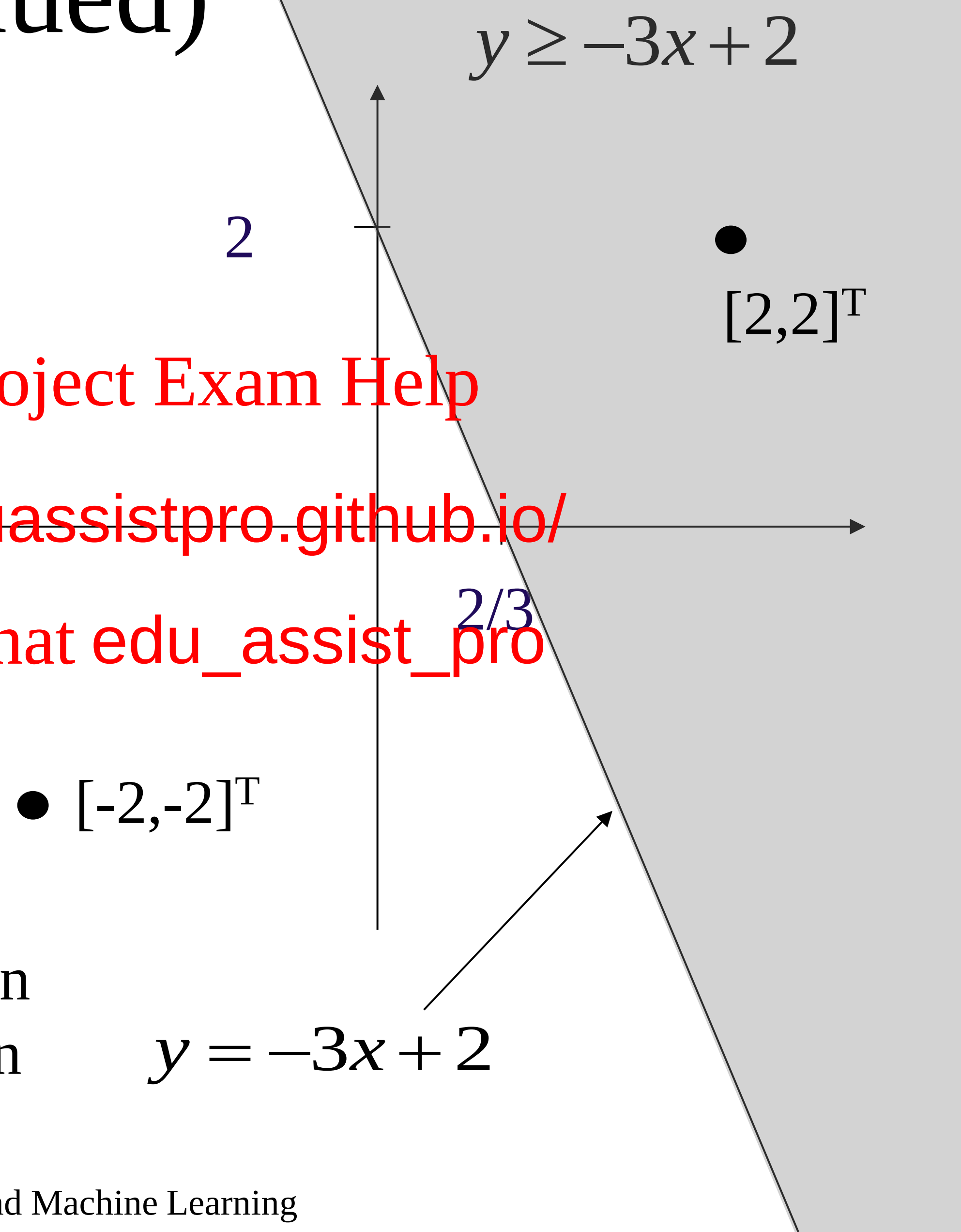


$$i_u = 3x + y - 2 \geq 0$$

$$\text{i.e. } y \geq -3x + 2$$

Example (continued)



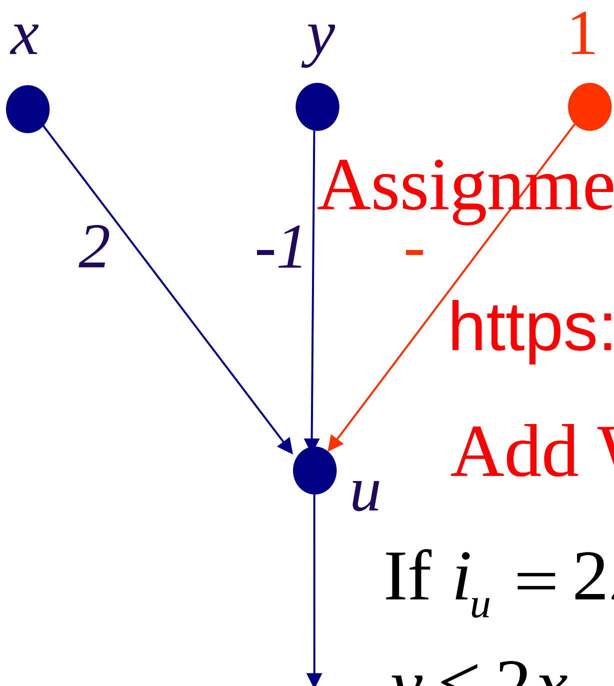


A single artificial neuron defines a linear decision boundary

Example (continued)

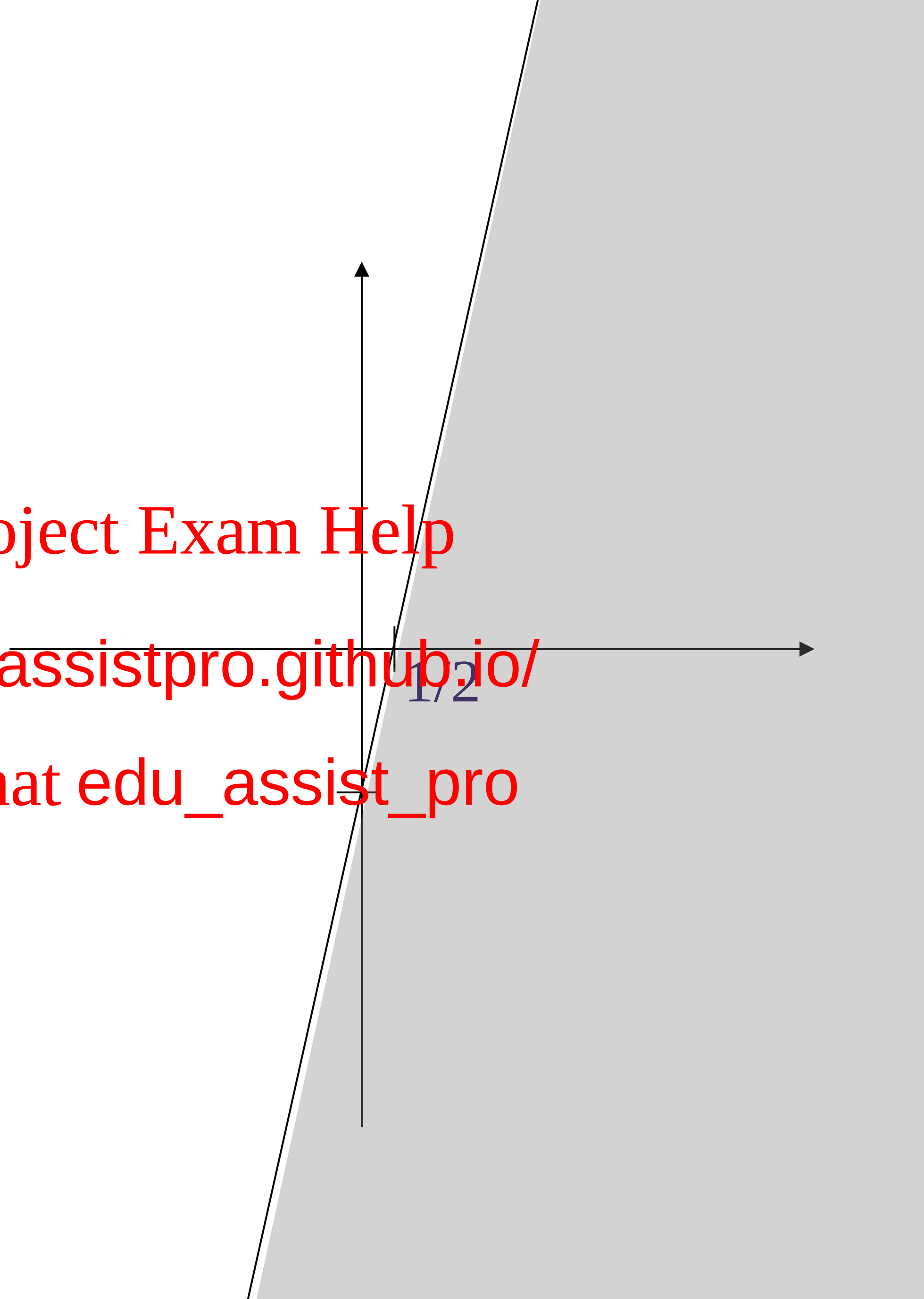
- Assume
  - Linear activation functions for units u_1, u_2 and u_3
  - Sigmoid activation function for u
- <u>Case1</u>: inpu u_2 is 2, then:
  - Input i_u to u
  - Hence output o_u from u is g
- <u>Case</u> 2: input to u_1 is -2 and input to u_2 is -2, then:
  - Input i_u to u is $-2 \times 3 + -2 \times 1 + 1 \times (-2) = -10$
  - Hence output o_u from u is $g(-10) = 4.54 \times 10^{-5} \approx 0$

Example 2

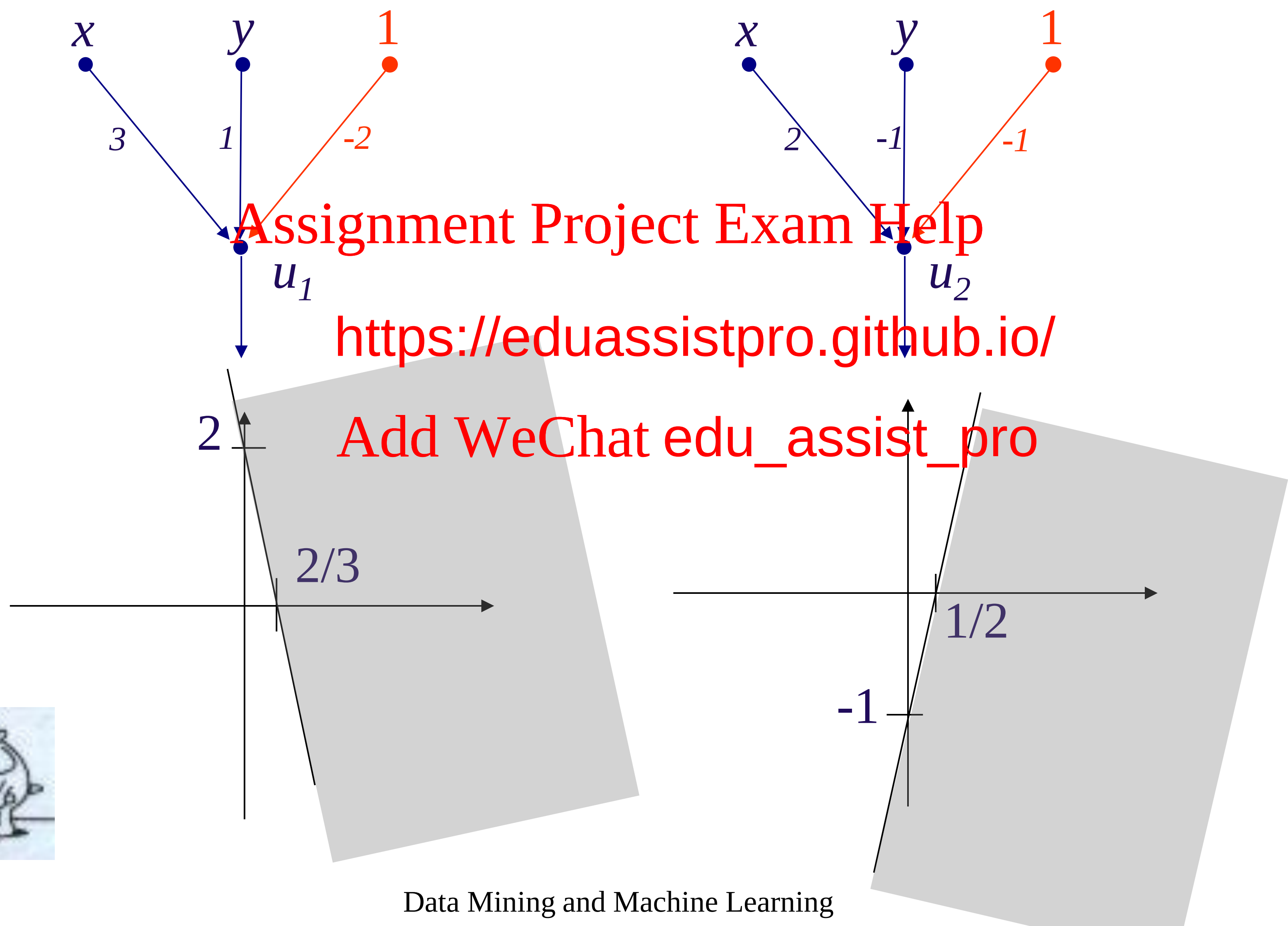


If $i_u = 2x$

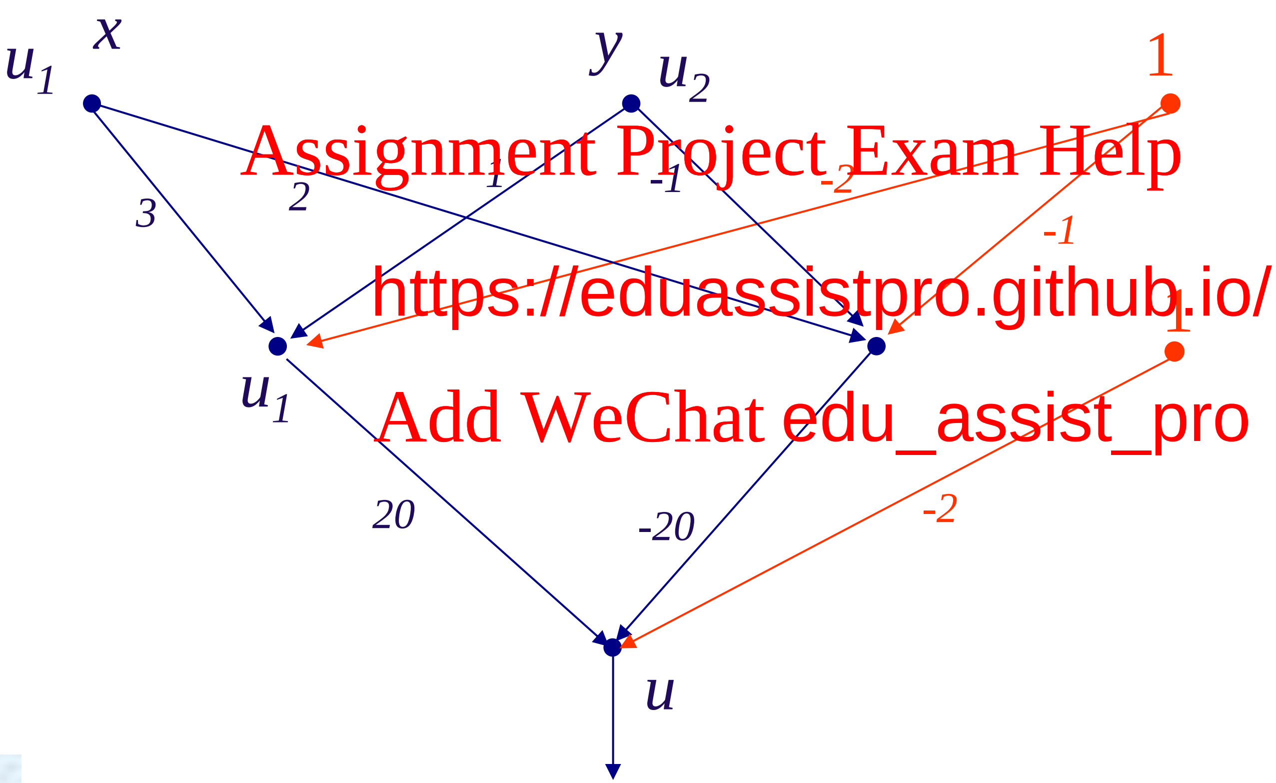
$y \leq 2x - 1$



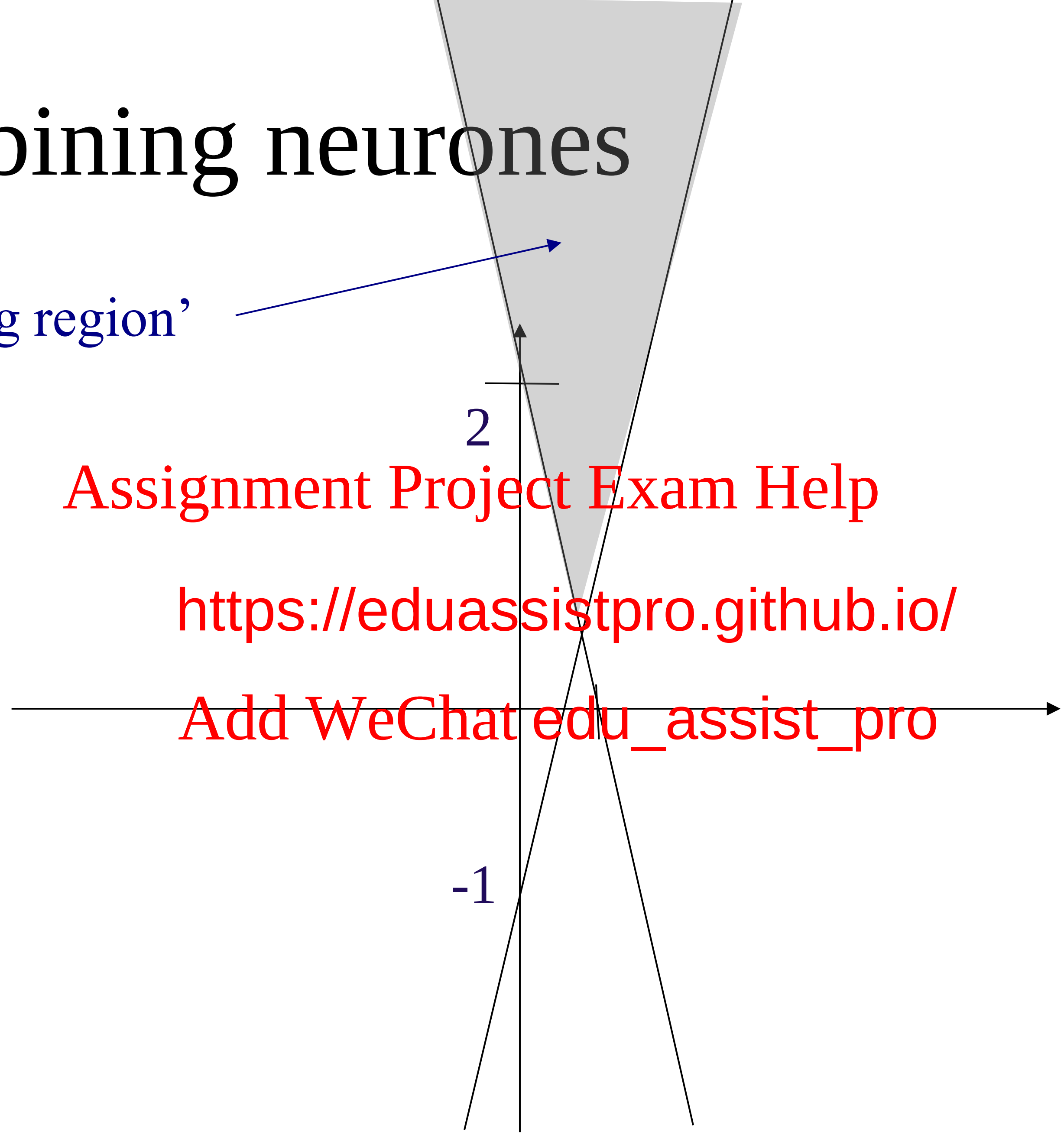
Combining 2 Artificial Neurons



Combining neurons – artificial neural networks



Combining neurones



Combining neurons

- Input to u_1 is $3x + y - 2$
- Input to u_2 is $2x - y - 1$

- When $x = 3$,
  - Input i_{u1} to

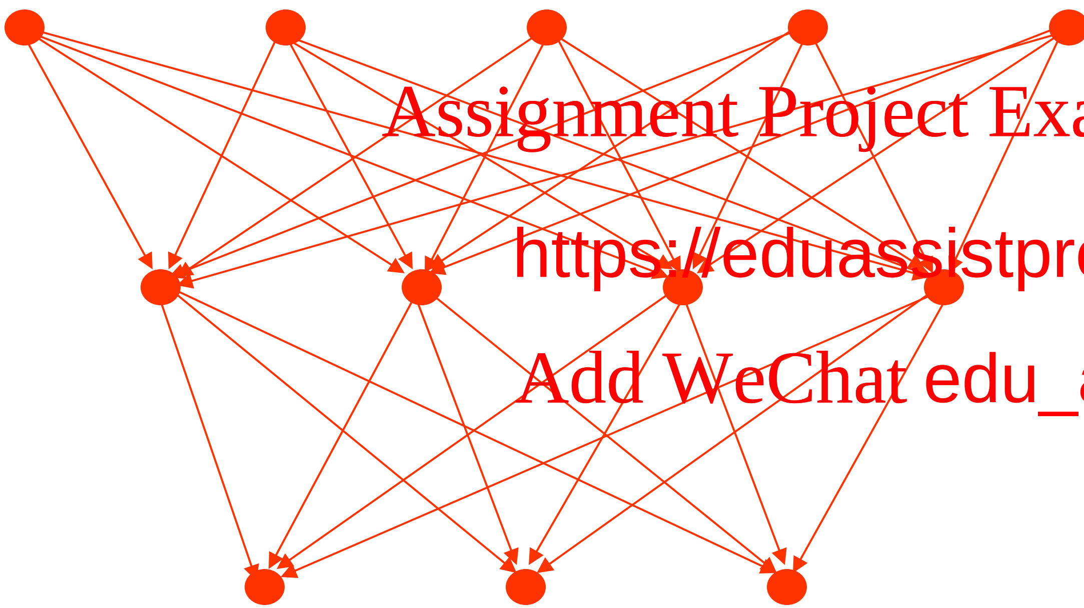
  - Output o_{u1} from u_1 is 1, out u_2 is 0.993

  - Input i_u to u is $1 \times 20 + 0.9$ 2 = -1.88
  - Output o_u from u *is* $g(-1.88) = 0.13$

Outputs

i_1	i_2	o_u
3	0	0.13
0.5		1.00
0.5		0.00
-1	0	0.06

Single hidden layer Multi-Layer Perceptron (MLP)



I units in Input layer

$H \times I$ weight matrix $\boldsymbol{W^1}$

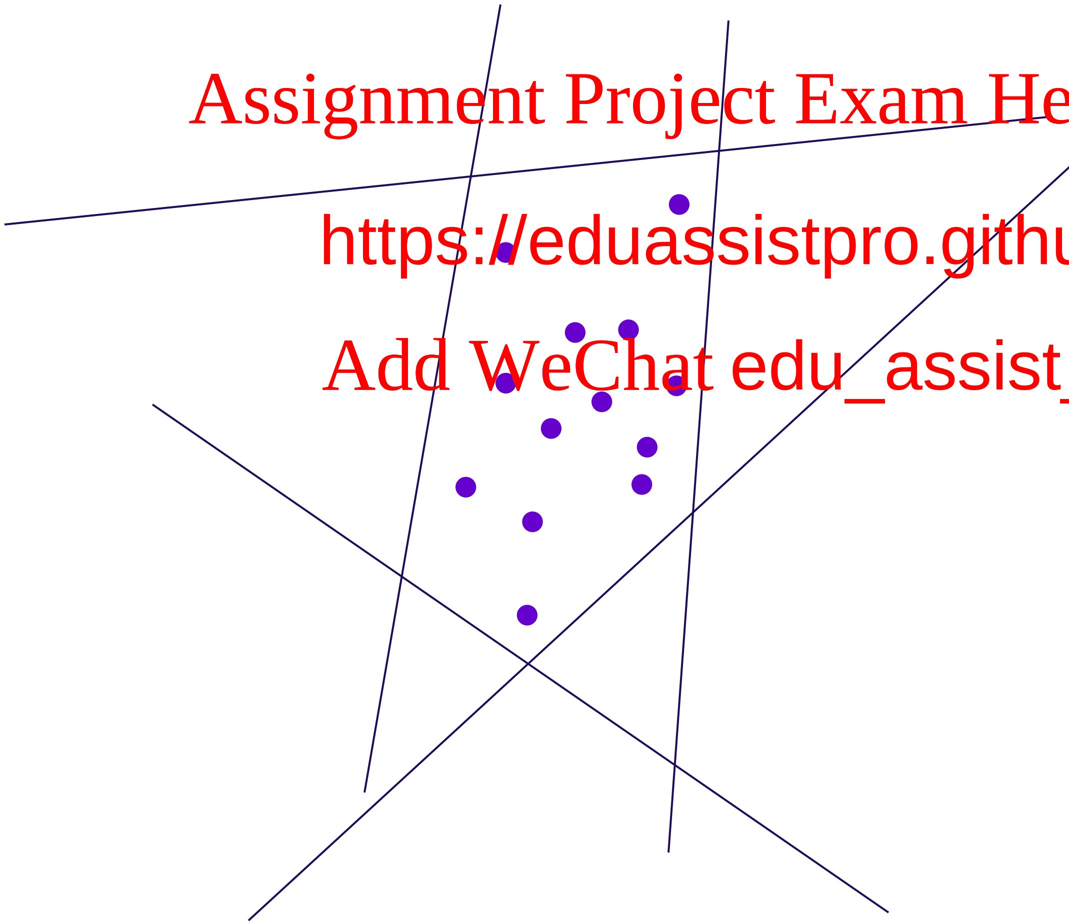
nits in Hidden layer

weight matrix $\boldsymbol{W^2}$

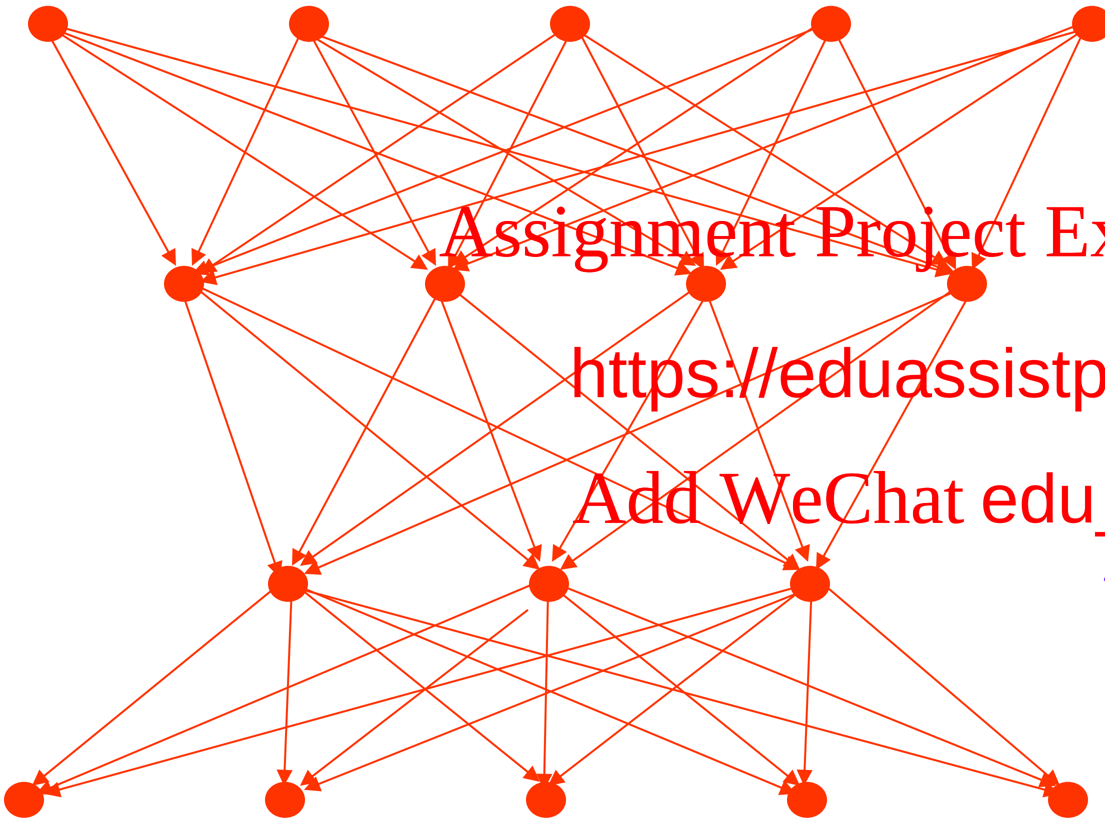
O units in Output layer

Single hidden layer MLP

- Can characterize arbitrary convex regions
- Defines the region using linear decision boundaries



Two hidden layer MLP



I units in Input layer

$H_1 \times I$ weight matrix $\boldsymbol{W^1}$

nits in first hidden layer

$\times H_1$ weight matrix $\boldsymbol{W^2}$

$_2$ in second hidden layer

$O \times H_2$ weight matrix $\boldsymbol{W^3}$

O units in Output layer

Two hidden layer MLP

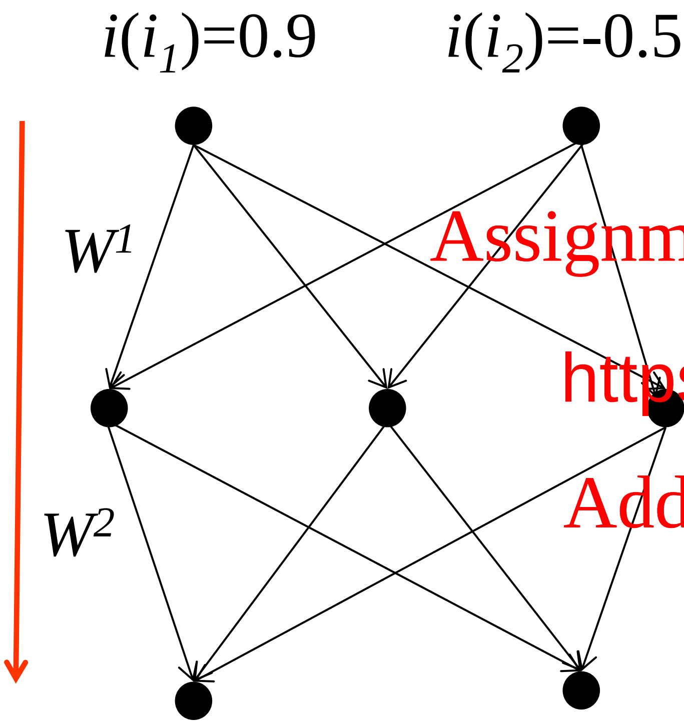
- An MLP with two hidden layers can characterize arbitrary shapes

- First hidden layer characterises convex regions
- Second hidd e convex regions
- In theory, there is no adva ving more than two hidden layers
- In practice multiple hidden layer “deep” neural networks give best performance (e.g. Speech recognition)

Formal definition: MLP with a single hidden layer

- A single hidden layer MLP consists of:
  1. A set of I input units, and for each input unit i an activation function g_i (typically linear)
  2. A set of H hidden unit h an activation oid)
  3. A set of O output units, an utput unit o an activation function g_o
  4. An H x I weight matrix W^1, which maps the outputs of the input units to the inputs of the hidden units
  5. An O x H weight matrix W^2, which maps the outputs of the hidden units to the inputs of the output units

Example



- 2 unit input layer, linear activation (I = 2)
- Single 3 unit hidden layer, sigmoid activation (H = 3)
- er, linear activation
- A 3 x trix W^1 between input ayer
- A 2 x 3 weight matrix W^2 between hidden and output layer

Example continued

$$W^1 = \begin{bmatrix} 2.6 & -1.7 \\ 0.2 & 1.0 \\ -4.0 & 2.5 \end{bmatrix}, \quad {}^{2} \begin{bmatrix} 1.0 & -0.5 & 1.0 \\ & & 0 \end{bmatrix}$$

$$\text{Input} = \begin{bmatrix} 0.9 \\ -0.5 \end{bmatrix}$$

$$\text{Output from first layer} = \begin{bmatrix} 0.9 \\ -0.5 \end{bmatrix} \text{(linear activation)}$$

Example (continued)

Inputs to hidden layer:

$$i(h_1) = w^1_{11}o_1 + w^1_{12}o_2 = 2.6 \times 0.9 + (-1.7) \times \left(-0.5\right) = 2.34 + 0.85 = 3.19,$$

$$i\left(h_2\right) = w^1_{21}o_1 + w^1_{22}o_2 = 0.2 \times 0.9 + 1.0 \times (-0.5) = 0.18 - 0.5 = -0.32,$$

$$i\left(h_3\right) = w^1_{31}o_1 + w^1_{32}o_2 = (-4.0) \times 0.9 + 2.5 \times (-0.5) = -3.6 - 1.25 = -4.85$$

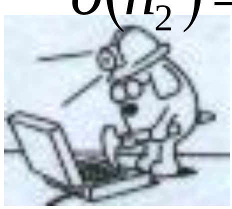
In matrix notation:

$$i\left(h\right) = W^1 o$$

Outputs from hidden layer:

$$o(h_1) = \frac{1}{1+e^{-3.19}} = 0.96,$$

$$o\left(h_2\right) = \frac{1}{1+e^{0.32}} = 0.42,\ o\left(h_3\right) = \frac{1}{1+e^{4.85}} = 0.008.$$



Example (continued)

Inputs to the output layer:

$$i(o_1) = w_{11}^2 \times o(h_1) + w_{12}^2 \times o(h\ ) + w^2\ \times o(h\ )$$

$$i(o_1) = 1 \times 0.96 + (-0.5) \qquad 1 + 0.008 = 0.758.$$

$$i(o_2) = w_{21}^2 \times o(h_1) + w_{22}^2 \times o(h_2) + w_{23}^2 \times o(h$$

$$i(o_2) = 0.5 \times 0.96 + 0.6 \times 0.42 - 1 \times 0.008 = \quad -0.008 = 0.742.$$

In matrix notation:

$$i(o) = W^2 o(h)$$

Linear output unit activation:

$$o(o_1) = 0.758, o(o_2) = 0.742.$$

Summary

- Introduction to neural networks
- Definition of an ‘artificial neurone’
- Activation functions – linear and sigmoid
- Linear boun e neurone
- Convex regi LP
- Two-level MLPs
- Forward propagation in an MLP (calculation)