ED5340 - Data Science: Theory and Practise

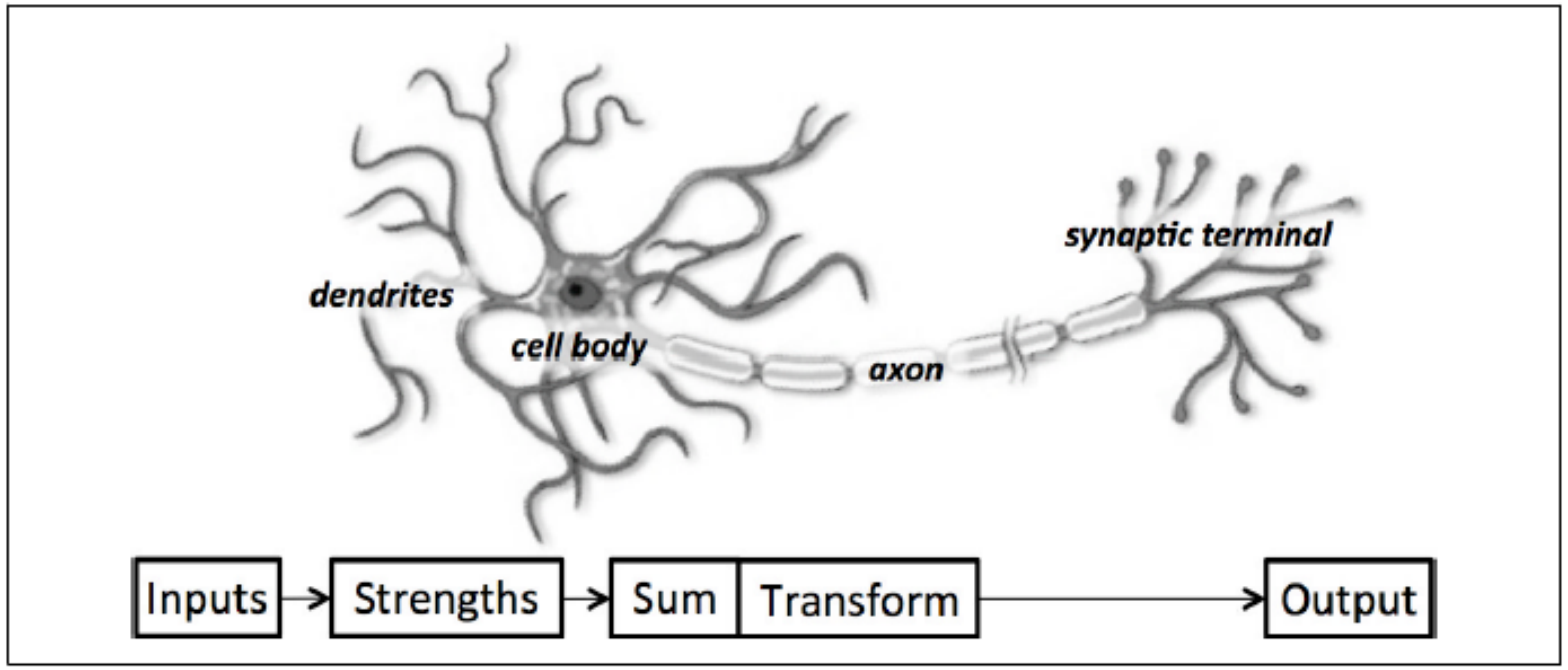
L25 - Neural Networks

Ramanathan Muthuganapathy

Ramanathan Muthuganapathy (https://ed.iitm.ac.in/~raman)
Course web page: https://ed.iitm.ac.in/~raman/datascience.html
Moodle page: Available at https://courses.iitm.ac.in/

Biological Neuron

Fundamentals of deep learning - Nikhil Buduma

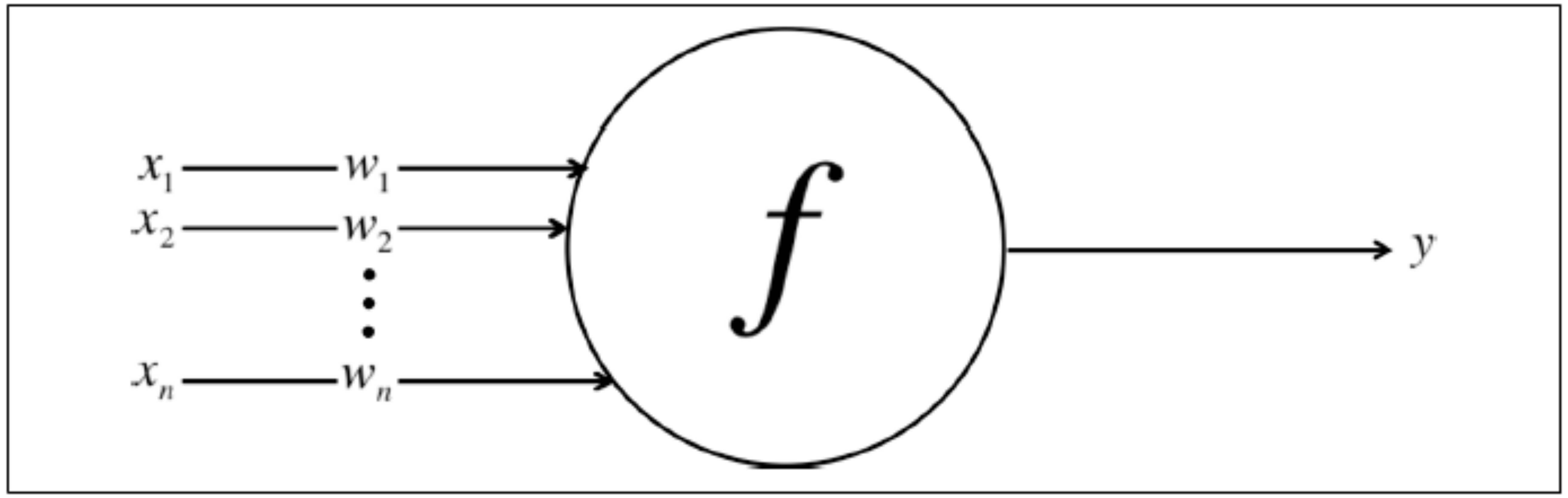


- Fundamental unit of human brain - neuron
- Small region - 10000 neurons -
    - each neutron having 6000 connections with others
- Input from Antinna-like structures called dendrites ——> neurons ——> connections (wighted) ——> other neurons ——> output

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

ANN

Fundamentals of deep learning - Nikhil Buduma



- Replicate the biological neuron
- Inputs are features
    - weighted and send to neuron via connections through functions
- Output

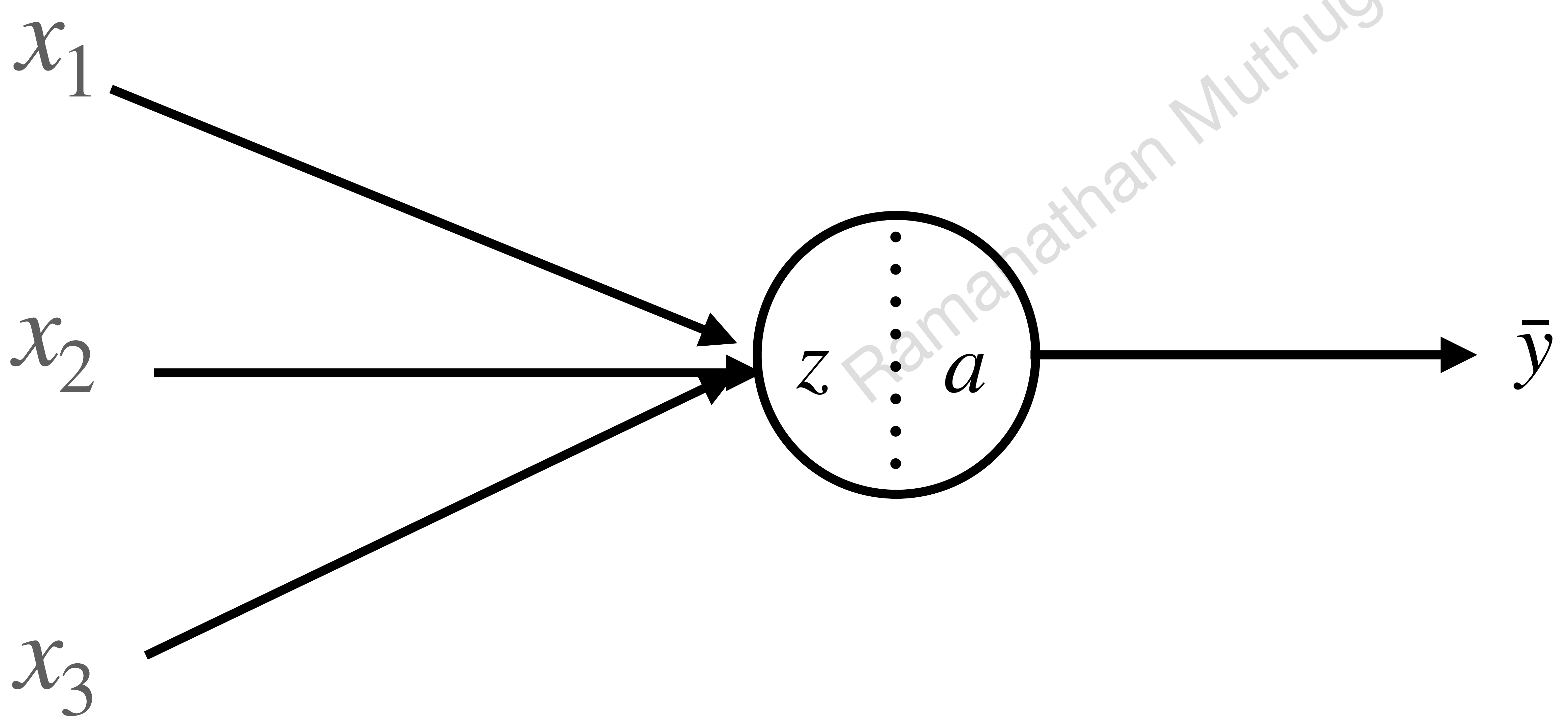
Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

You already know NN

- Logistic regression
  - Computing z and σ
  - Gradient descent
- Stacking of several LogReg

Ramanathan Muthuganapathy

Logistic Regression - Pictorial



$$z = \mathbf{w}^T\mathbf{x} + b$$

$$\bar{y} = a = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$J(a, y)$$

Ramanathan Muthuganapathy

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Logistic Regression - Pictorial

- X - input
- W - weights
- b = bias
- a = activation function (sigmoid / logistic)
- J is the loss function

$$x,\ w,\ b \longrightarrow z = \mathbf{w}^T\mathbf{x} + b \longrightarrow a = \sigma(z) \longrightarrow J(a, y)$$

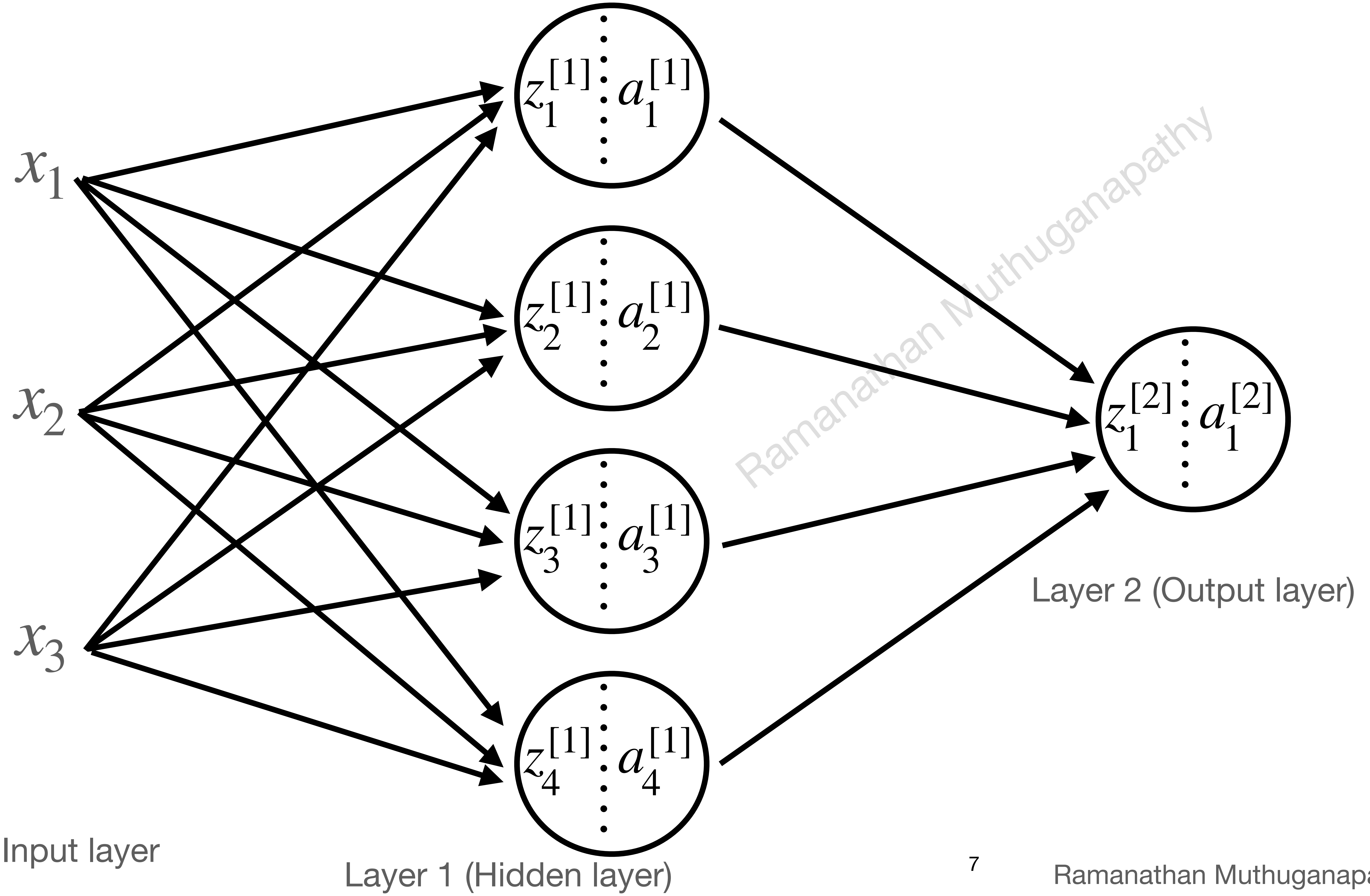
- $\mathbf{w}^T = [w_1 \quad w_2 \quad w_3]$
- $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

Ramanathan Muthuganapathy

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Neural Network - Pictorial

Two layer network!



Ramanathan Muthuganapathy

$$z = \mathbf{w}^T\mathbf{x} + b$$

$$a = \sigma(z) = \frac{1}{1 + e^{-z}}$$

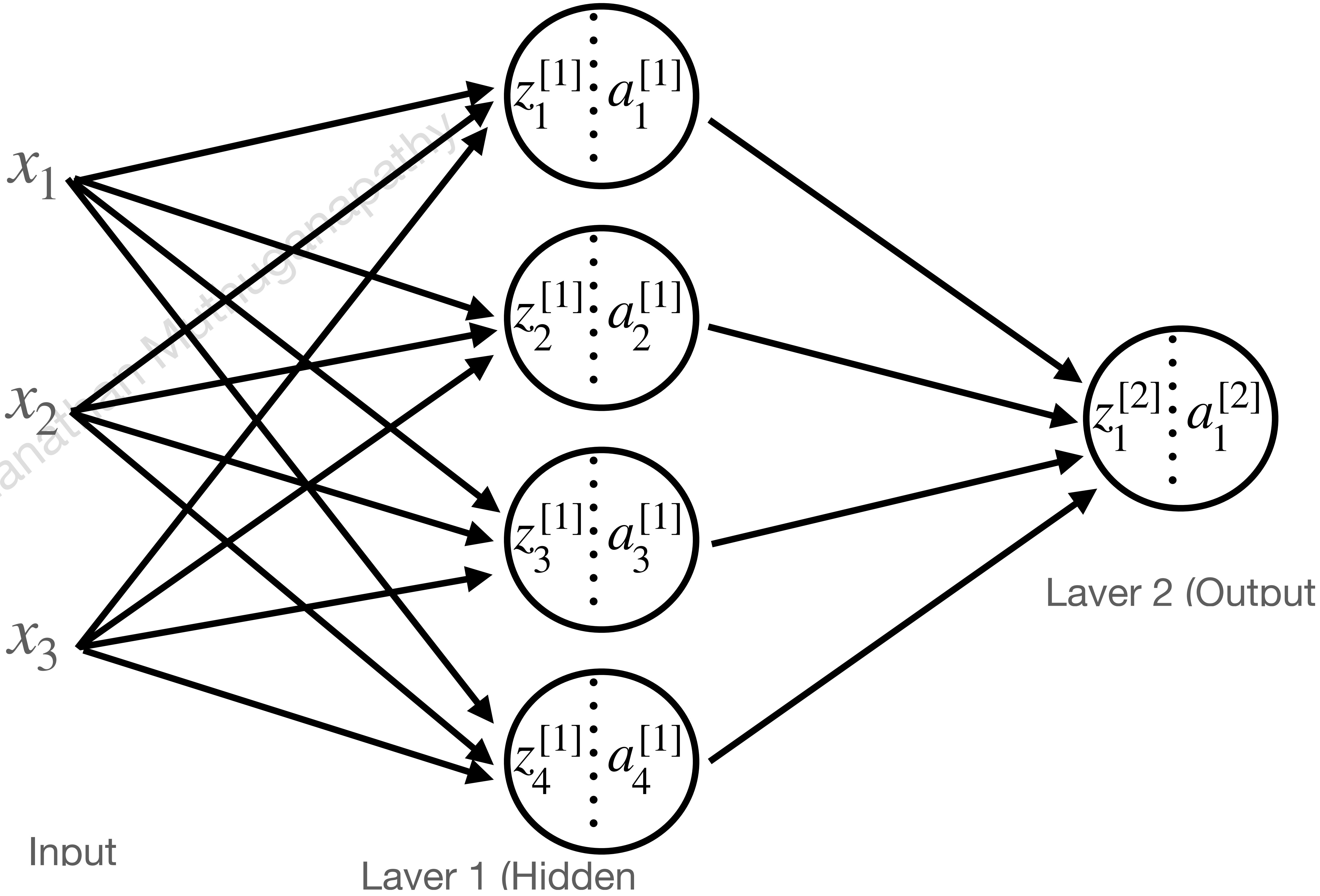
$$\bar{y} = a_1^{[2]}$$

$$J(y, \bar{y}) = J(a, y)$$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Terminologies in NN

- Unit / Neuron / Node
- Connection
- Layer
    - Input (not counted)
    - Hidden
    - Output
- Activation Function (here, sigmoid)

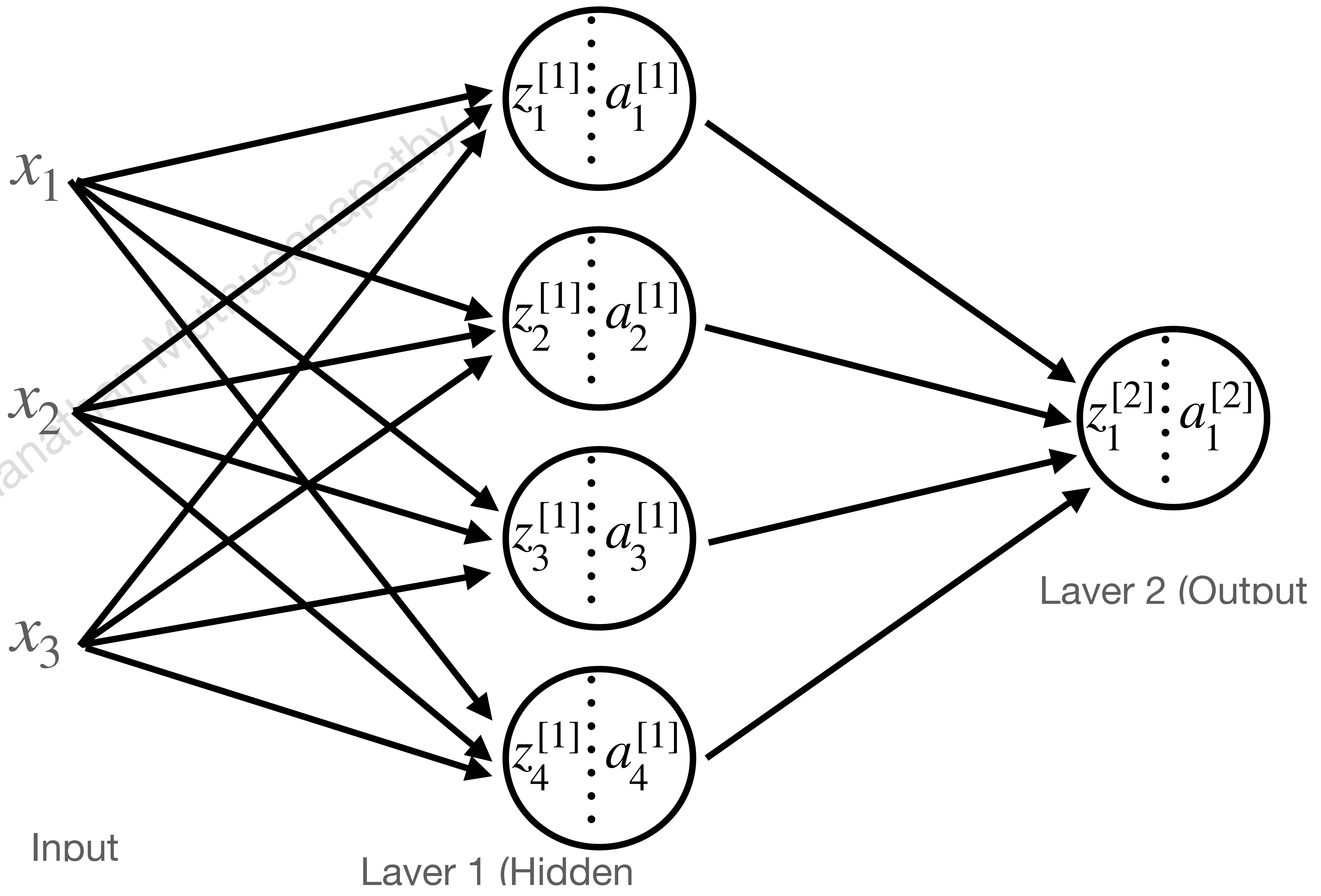


Ramanathan Muthuganapathy

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Terminologies in NN

- Forward-propagation
    - Compute Activation Function for each layer (unit in the layer)
- Back-propagation
    - Computing Gradients
- Fully-connected layer

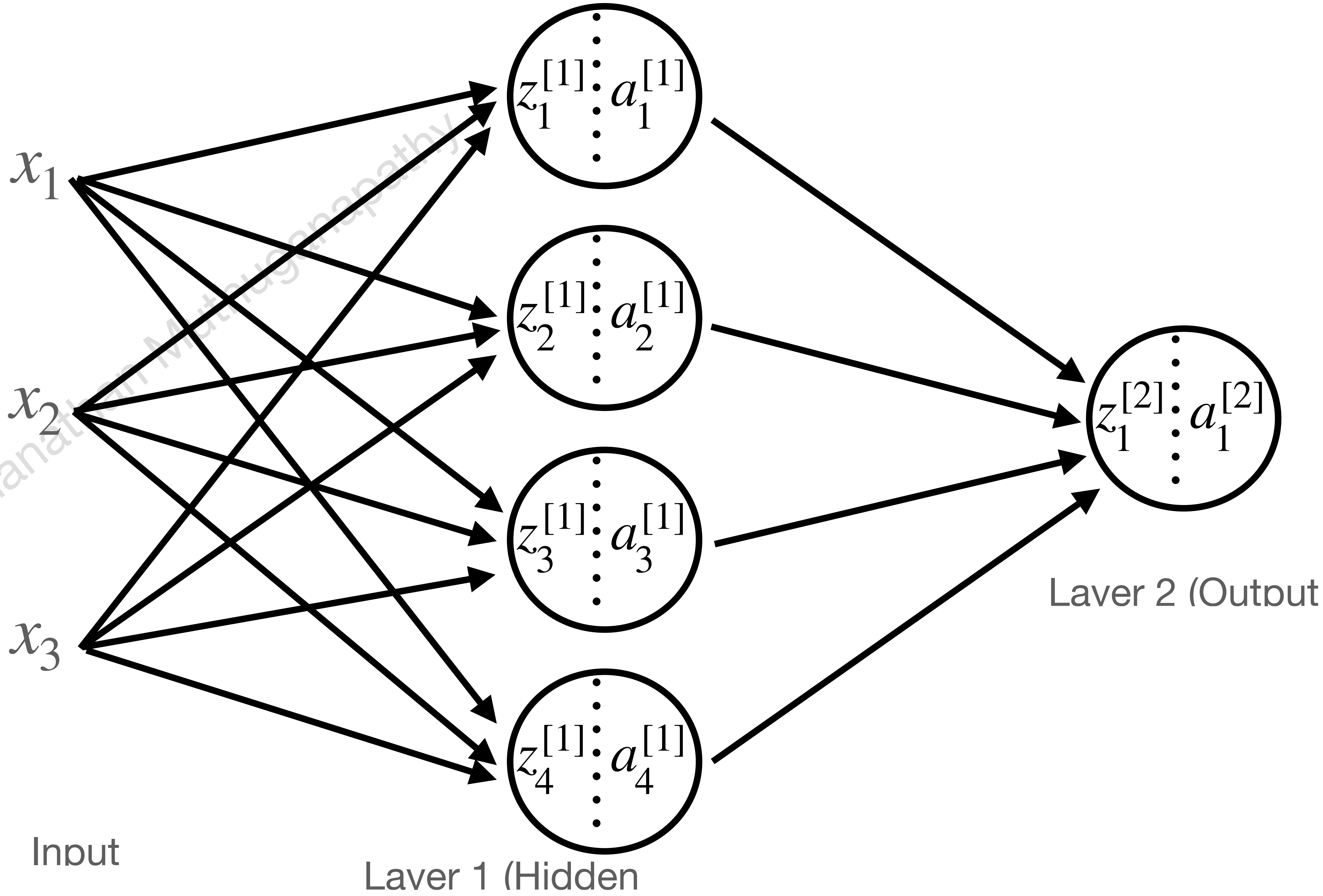


Ramanathan Muthuganapathy

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Notations in NN (Andrew Ng)
MLR follows the reverse!

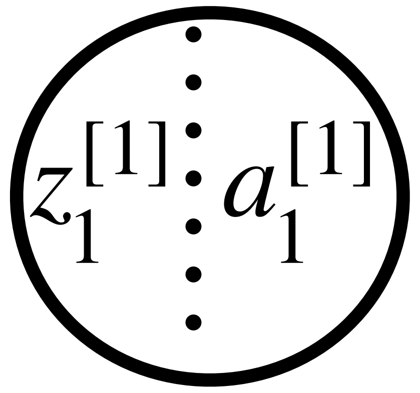
- Superscript in square bracket - Layer number
- Superscript in C - bracket - Sample number
- Subscript - Unit number in the layer

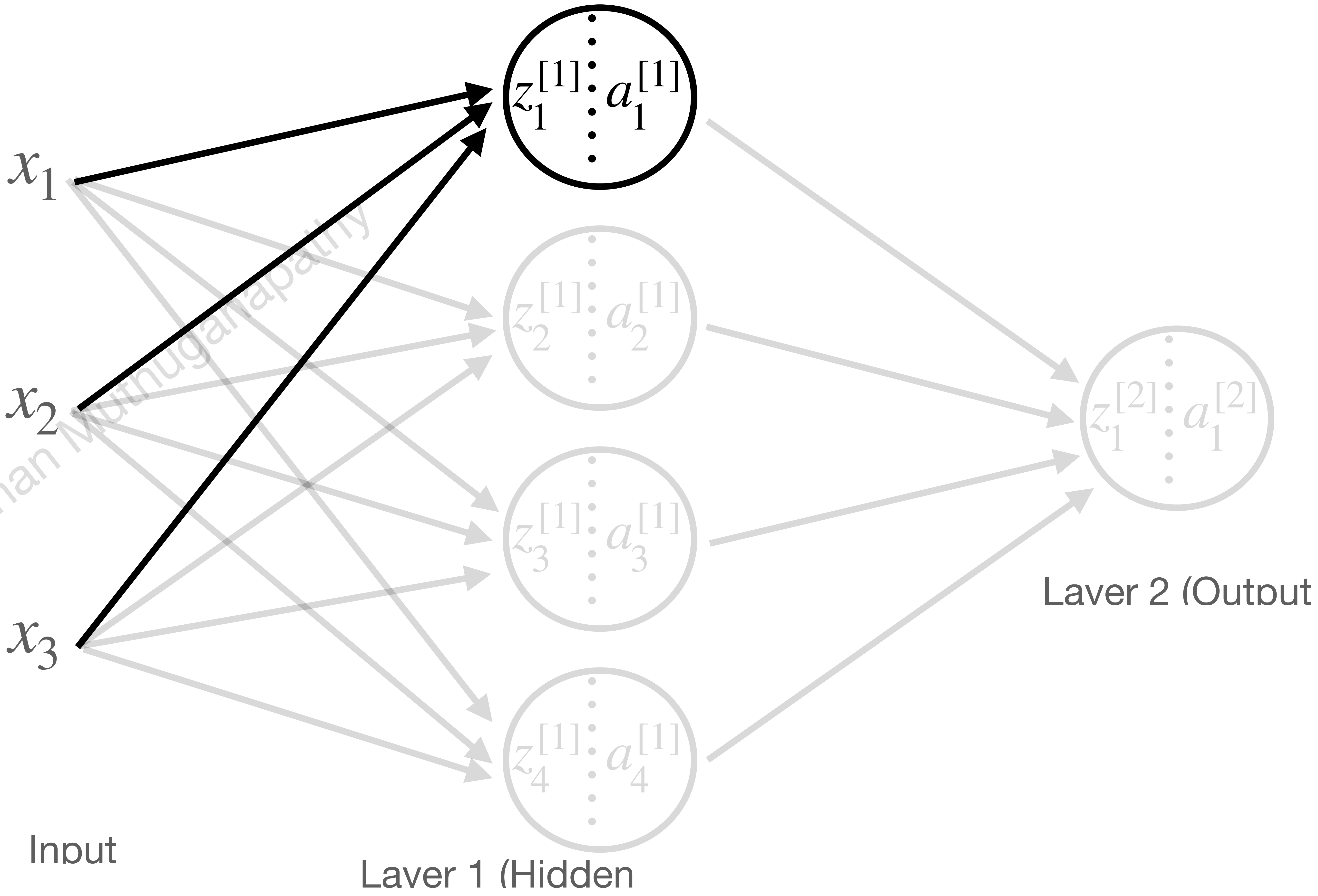


Ramanathan Muthuganapathy

Forward propagation

In unit 1 of layer 1





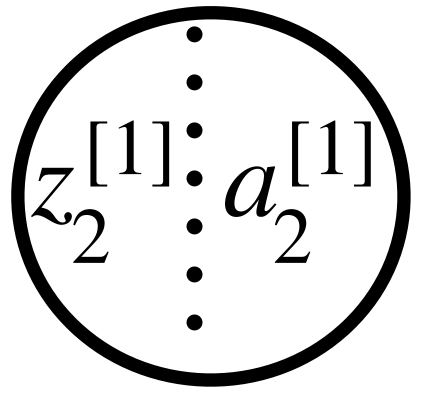
$$z_1^{[1]} = \mathbf{w}_1^{[1]^T}\mathbf{x} + b_1^{[1]} \longrightarrow a_1^{[1]} = \sigma(z_1^{[1]})$$

- $\mathbf{w}_1^{[1]^T} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} & w_{1,3}^{[1]} \end{bmatrix}$

Ramanathan Muthuganapathy

Forward propagation

In unit 2 of layer 1



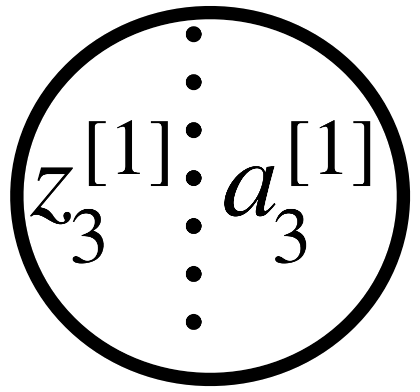
$$z_2^{[1]} = \mathbf{w}_2^{[1]^T}\mathbf{x} + b_2^{[1]} \longrightarrow a_2^{[1]} = \sigma(z_2^{[1]})$$

- $\mathbf{w}_2^{[1]^T} = \begin{bmatrix} w_{2,1}^{[1]} & w_{2,2}^{[1]} & w_{2,3}^{[1]} \end{bmatrix}$

Ramanathan Muthuganapathy

Forward propagation

In unit 3 of layer 1



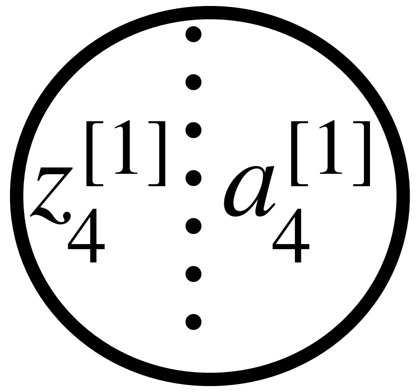
$$z_3^{[1]} = \mathbf{w}_3^{[1]^T}\mathbf{x} + b_3^{[1]} \longrightarrow a_3^{[1]} = \sigma(z_3^{[1]})$$

- $\mathbf{w}_3^{[1]^T} = \begin{bmatrix} w_{3,1}^{[1]} & w_{3,2}^{[1]} & w_{3,3}^{[1]} \end{bmatrix}$

Ramanathan Muthuganapathy

Forward propagation

In unit 4 of layer 1



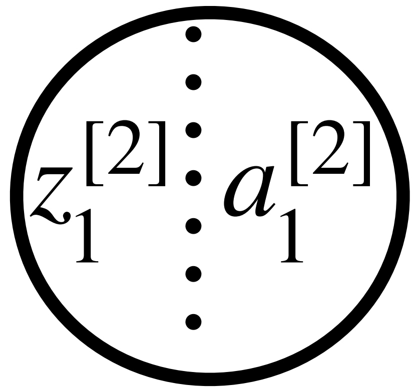
$$z_4^{[1]} = \mathbf{w}_4^{[1]^T}\mathbf{x} + b_4^{[1]} \longrightarrow a_4^{[1]} = \sigma(z_4^{[1]})$$

- $\mathbf{w}_4^{[1]^T} = \begin{bmatrix} w_{4,1}^{[1]} & w_{4,2}^{[1]} & w_{4,3}^{[1]} \end{bmatrix}$

Ramanathan Muthuganapathy

Forward propagation

In unit 1 of layer 2



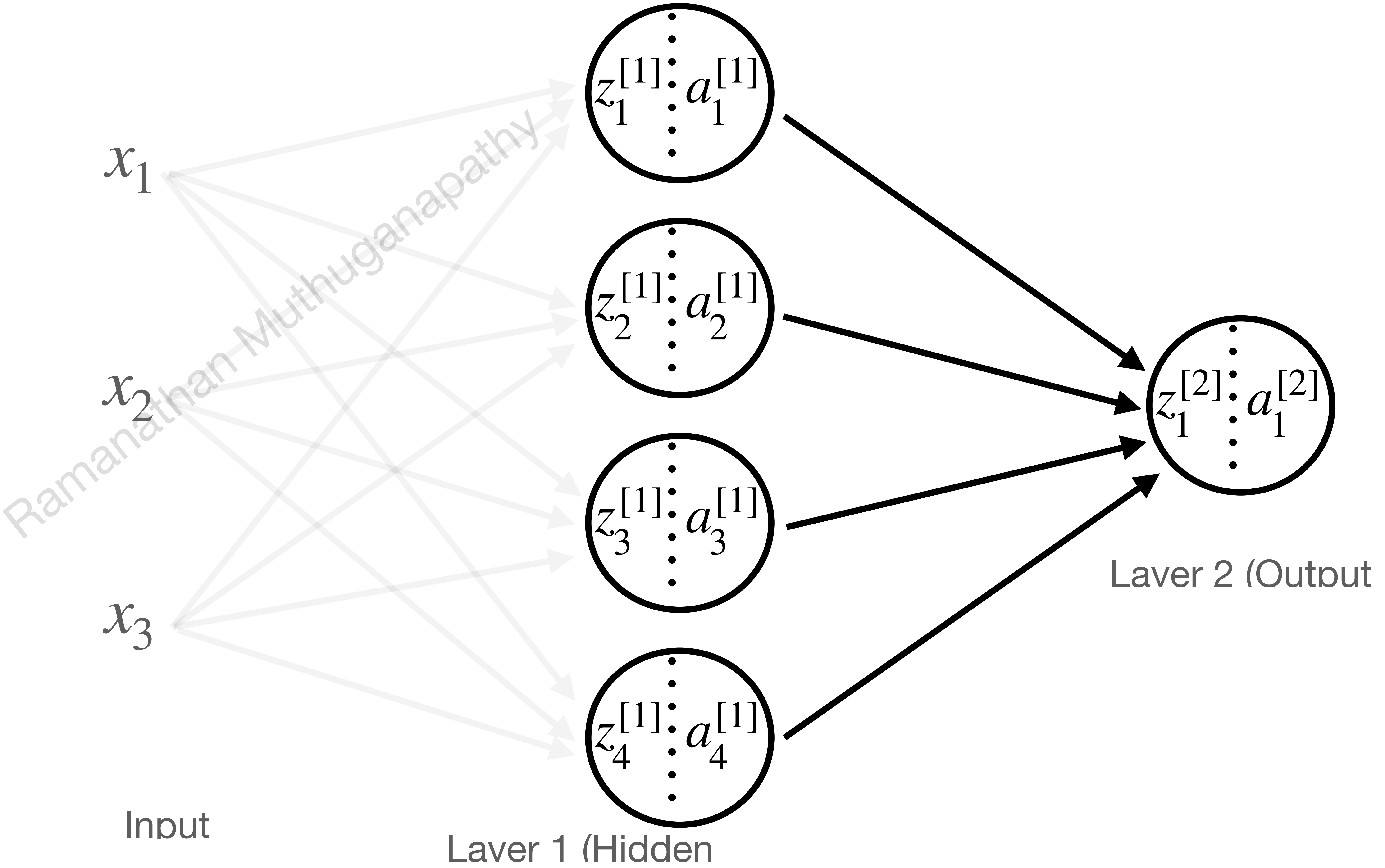
$$z_1^{[2]} = \mathbf{w}_1^{[2]^T} a^{[1]} + b_1^{[2]} \longrightarrow a_1^{[2]} = \sigma(z_1^{[2]})$$

- $\mathbf{w}_1^{[2]^T} = \begin{bmatrix} w_{1,1}^{[2]} & w_{1,2}^{[2]} & w_{1,3}^{[2]} & w_{1,4}^{[2]} \end{bmatrix}$
- $a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix}$

Ramanathan Muthuganapathy

Input
Layer 2

- Input for layer 2 is the output from layer 1



Ramanathan Muthuganapathy

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Vectorization (Matrix)

Layer 1

- $\mathbf{w}_1^{[1]^T} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} & w_{1,3}^{[1]} \end{bmatrix}$
- $\mathbf{w}_2^{[1]^T} = \begin{bmatrix} w_{2,1}^{[1]} & w_{2,2}^{[1]} & w_{2,3}^{[1]} \end{bmatrix}$
- $\mathbf{w}_3^{[1]^T} = \begin{bmatrix} w_{3,1}^{[1]} & w_{3,2}^{[1]} & w_{3,3}^{[1]} \end{bmatrix}$
- $\mathbf{w}_4^{[1]^T} = \begin{bmatrix} w_{4,1}^{[1]} & w_{4,2}^{[1]} & w_{4,3}^{[1]} \end{bmatrix}$

- $\mathbf{W}^{[1]} = \begin{bmatrix} \mathbf{w}_1^{[1]^T} \\ \mathbf{w}_2^{[1]^T} \\ \mathbf{w}_3^{[1]^T} \\ \mathbf{w}_4^{[1]^T} \end{bmatrix}$
- Dropping the transpose
- using W (Capital)
- $\mathbf{W}^{[1]}$ instead of $\mathbf{w}^{[1]^T}$

Ramanathan Muthuganapathy

Vectorization (Matrix)

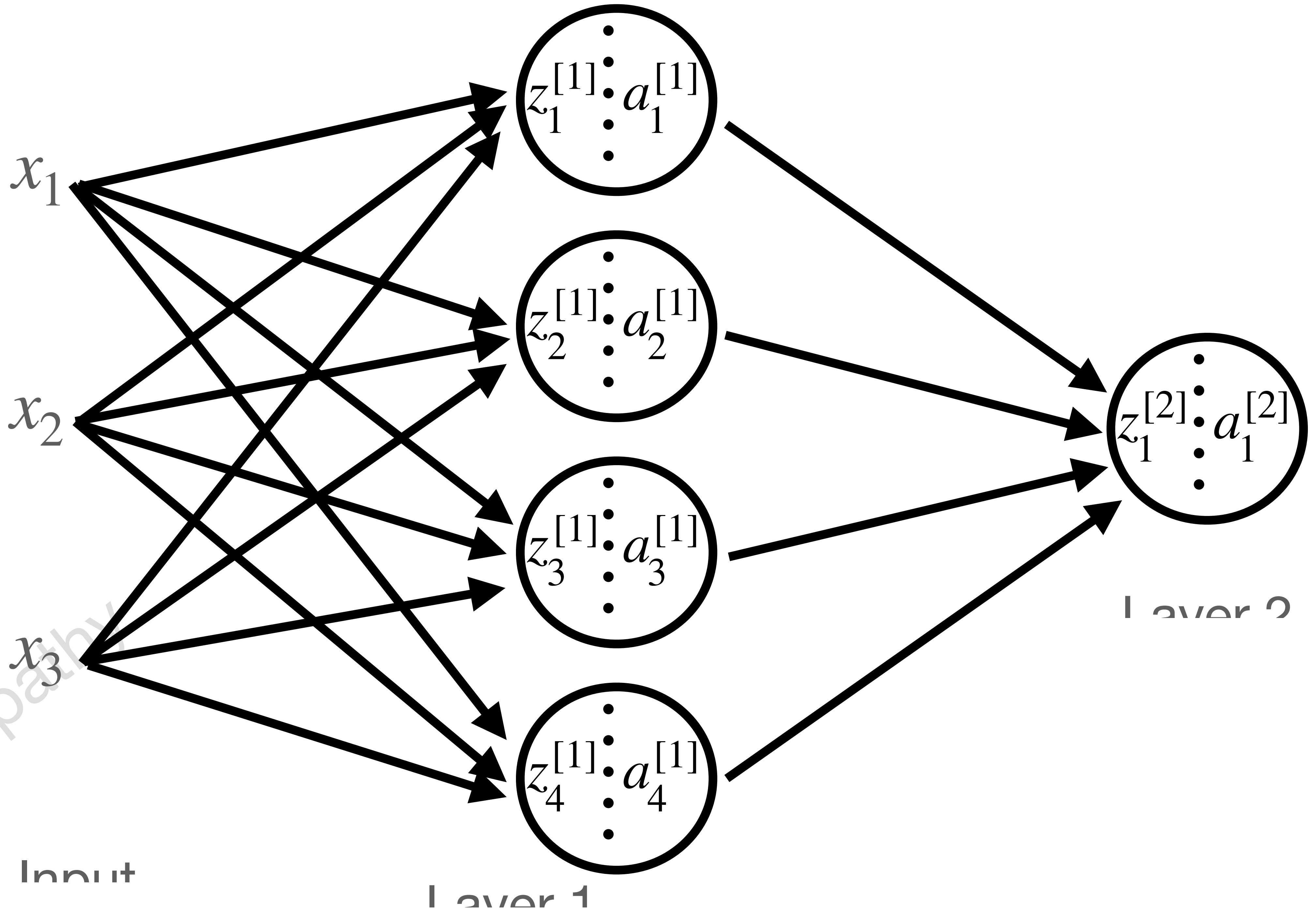
Layer 1

- $$\mathbf{b}^{[1]} = \begin{bmatrix} \mathbf{b}_1^{[1]} \\ \mathbf{b}_2^{[1]} \\ \mathbf{b}_3^{[1]} \\ \mathbf{b}_4^{[1]} \end{bmatrix}$$

Ramanathan Muthuganapathy

Vectorization (Matrix)

Layer 1

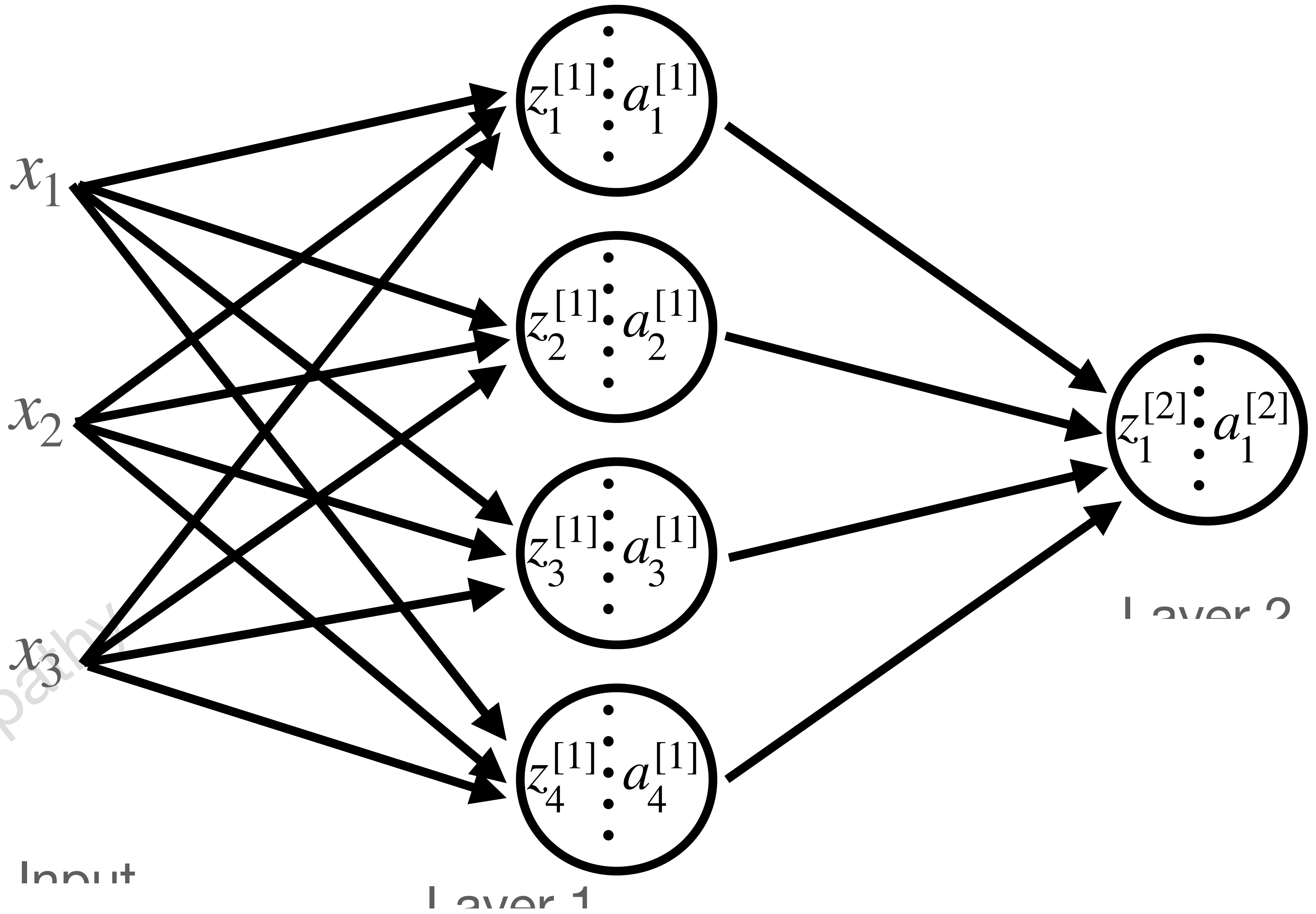


- $$z^{[1]} = \begin{bmatrix} \mathbf{w}_1^{[1]^T}\mathbf{x} + b_1^{[1]} \\ \mathbf{w}_2^{[1]^T}\mathbf{x} + b_2^{[1]} \\ \mathbf{w}_3^{[1]^T}\mathbf{x} + b_3^{[1]} \\ \mathbf{w}_4^{[1]^T}\mathbf{x} + b_4^{[1]} \end{bmatrix} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix} = \mathbf{W}^{[1]}\mathbf{x} + b^{[1]}$$

Ramanathan Muthuganapathy

Forward propagation

Output from layer 1 (i/p to layer 2)

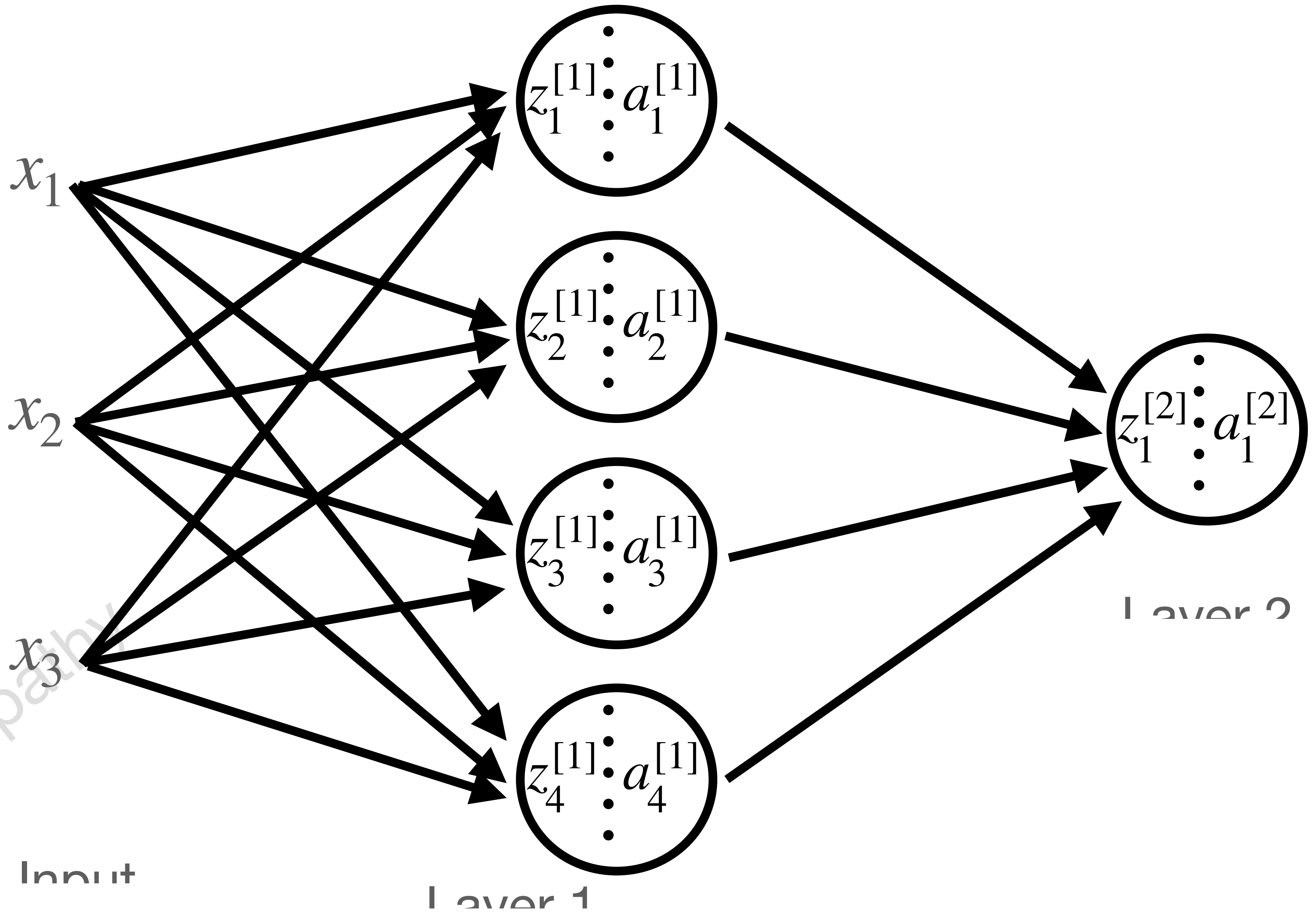


- $$a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix} = \begin{bmatrix} \sigma(z_1^{[1]}) \\ \sigma(z_2^{[1]}) \\ \sigma(z_3^{[1]}) \\ \sigma(z_4^{[1]}) \end{bmatrix} = \sigma(z^{[1]})$$

Ramanathan Muthuganapathy

Forward propagation

Unit 1 of layer 2



- $z^{[2]} = \mathbf{W}^{[2]}a^{[1]} + b^{[2]}$
- $a^{[2]} = \sigma(z^{[2]})$

Ramanathan Muthuganapathy

Vectorization (Matrix)

Layers 1 and 2 (for one sample)

- $z^{[1]} = \mathbf{W}^{[1]}\mathbf{x} + b^{[1]}$
- $a^{[1]} = \sigma(z^{[1]})$
- $z^{[2]} = \mathbf{W}^{[2]}a^{[1]} + b^{[2]}$
- $a^{[2]} = \sigma(z^{[2]})$

Ramanathan Muthuganapathy

Vectorization (Matrix)

Layers 1 and 2 (for one sample) - dimensions

Ramanathan Muthuganapathy

- $z^{[1]} = \mathbf{W}^{[1]}\mathbf{x} + b^{[1]}$ = (4X3) (3X1) + (4X1) = (4X1)
- $a^{[1]} = \sigma(z^{[1]})$ = (4X1)
- $z^{[2]} = \mathbf{W}^{[2]}a^{[1]} + b^{[2]}$ = (1X4) (4X1) + (1X1) = 1X1
- $a^{[2]} = \sigma(z^{[2]})$ = (1X1) = $\bar{y}$

Vectorization (Matrix)

Layers 1 and 2 (for m samples)

Ramanathan Muthuganapathy

- $z^{[1](i)} = \mathbf{w}^{[1]^T}\mathbf{x}^{(i)} + b^{[1]} = \mathbf{W}^{[1]}\mathbf{x}^{(i)} + b^{[1]}$
- $a^{[1](i)} = \sigma(z^{[1](i)})$
- $z^{[2](i)} = \mathbf{W}^{[2]}a^{[1](i)} + b^{[2]}$
- $a^{[2](i)} = \sigma(z^{[2](i)})$

Vectorization (Matrix)

Layers 1 and 2 (for m samples)

Ramanathan Muthuganapathy

- $z^{[1](i)} = \mathbf{W}^{[1]}\mathbf{x}^{(i)} + b^{[1]}$ = (4X3) (3Xm) + (4X1) = (4Xm)
- $a^{[1](i)} = \sigma(z^{[1](i)})$ = (4Xm)
- $z^{[2](i)} = \mathbf{W}^{[2]}a^{[1](i)} + b^{[2]}$ = (1X4) (4Xm) + (1X1) = 1Xm
- $a^{[2](i)} = \sigma(z^{[2](i)})$ = (1X m) = $(\bar{y}^{(1)}\,\bar{y}^{(2)}\ldots\ldots\bar{y}^{(m)})$

Vectorization (Matrix)

Layers 1 and 2 (for m samples)

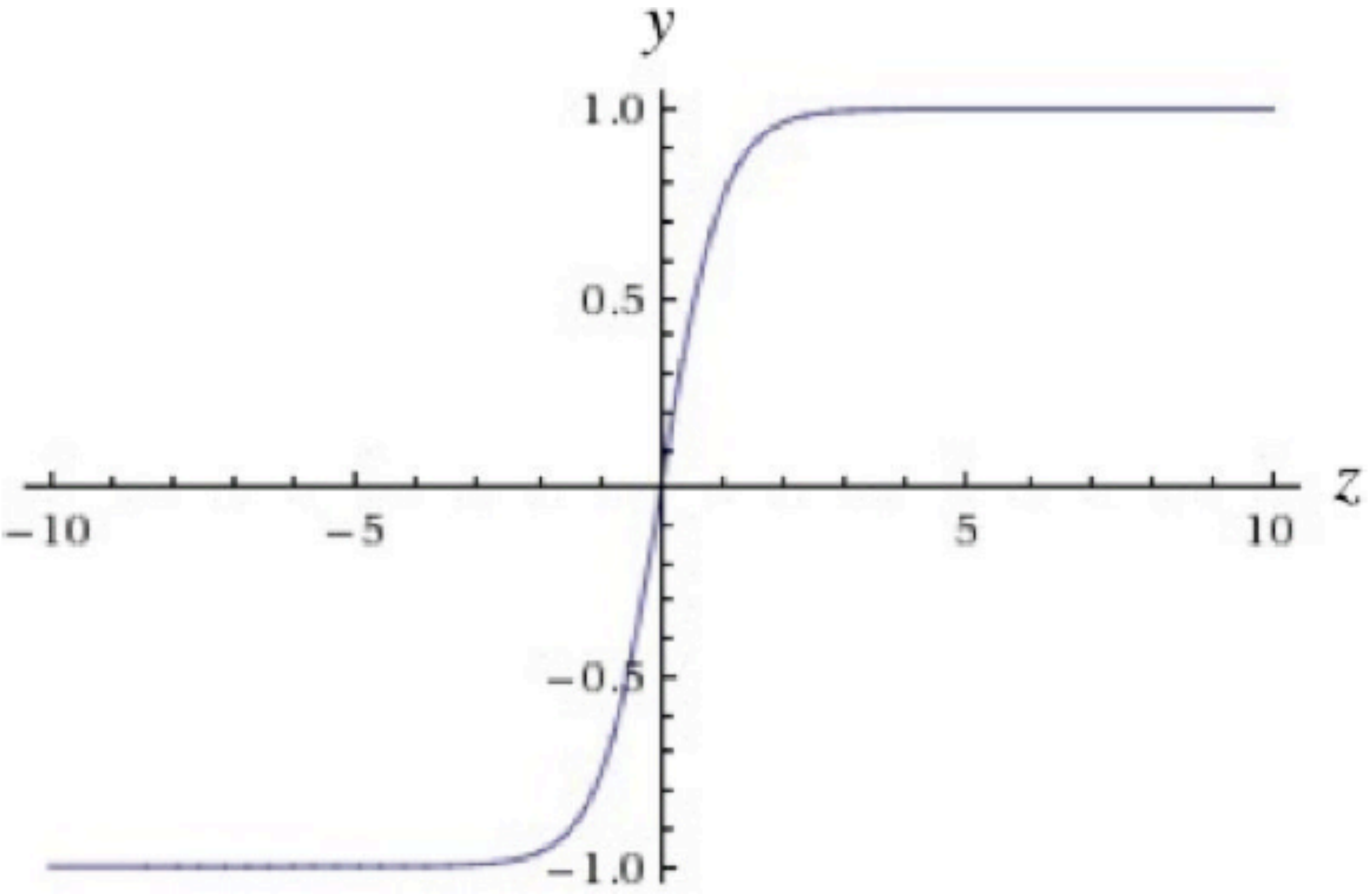
- $Z^{[1]} = \mathbf{W}^{[1]}\mathbf{X} + b^{[1]}$
- $A^{[1]} = \sigma(Z^{[1]})$
- $Z^{[2]} = \mathbf{W}^{[2]}A^{[1]} + b^{[2]}$
- $A^{[2]} = \sigma(Z^{[2]})$

- $$\mathbf{X} = A^{[0]} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(m)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(m)} \\ x_3^{(1)} & x_3^{(2)} & \dots & x_3^{(m)} \end{bmatrix}$$
- $$Z^{[1]} = \begin{bmatrix} z_1^{[1](1)} & z_1^{[1](2)} & \dots & z_1^{[1](m)} \\ z_2^{[1](1)} & z_2^{[1](2)} & \dots & z_2^{[1](m)} \\ z_3^{[1](1)} & z_3^{[1](2)} & \dots & z_3^{[1](m)} \\ z_4^{[1](1)} & z_4^{[1](2)} & \dots & z_4^{[1](m)} \end{bmatrix}$$

Ramanathan Muthuganapathy

Other activation functions

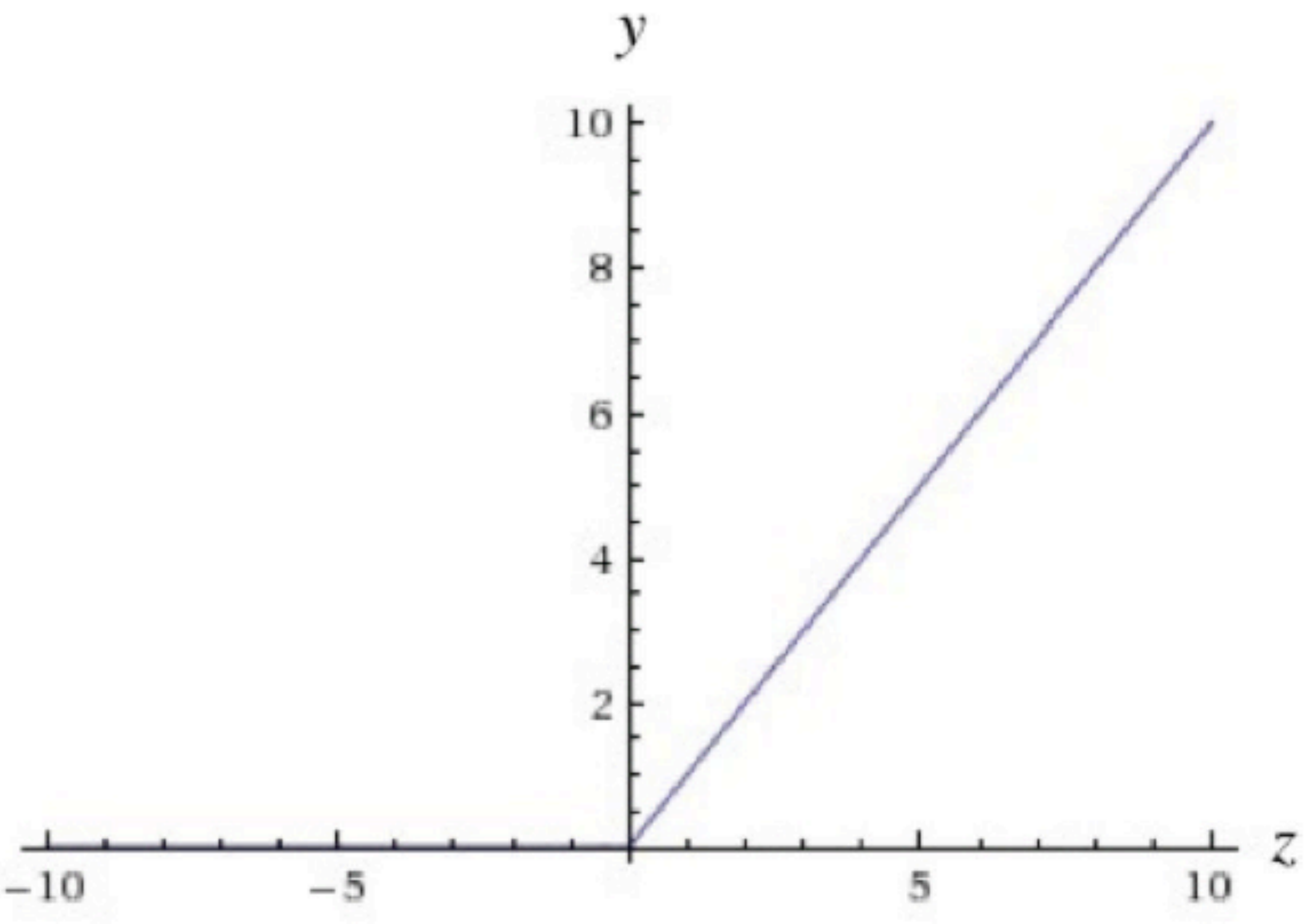
- $\tanh(z) = \dfrac{e^z - e^{-z}}{e^z + e^{-z}}$



Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Other activation functions

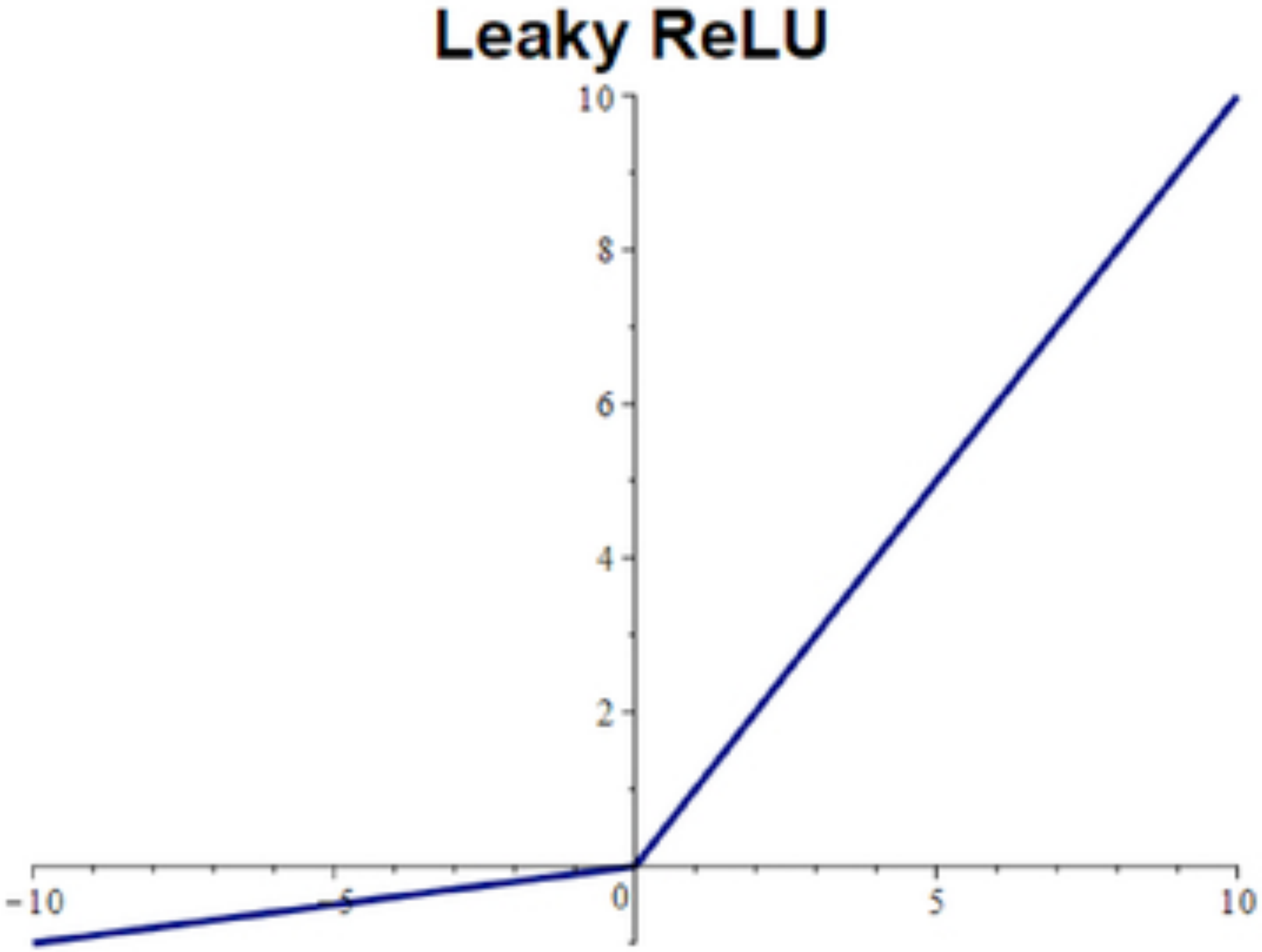
- ReLU(z) = max(0, z)



Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Other activation functions

- Leaky ReLU = max(0.01z, z)



Ra

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras