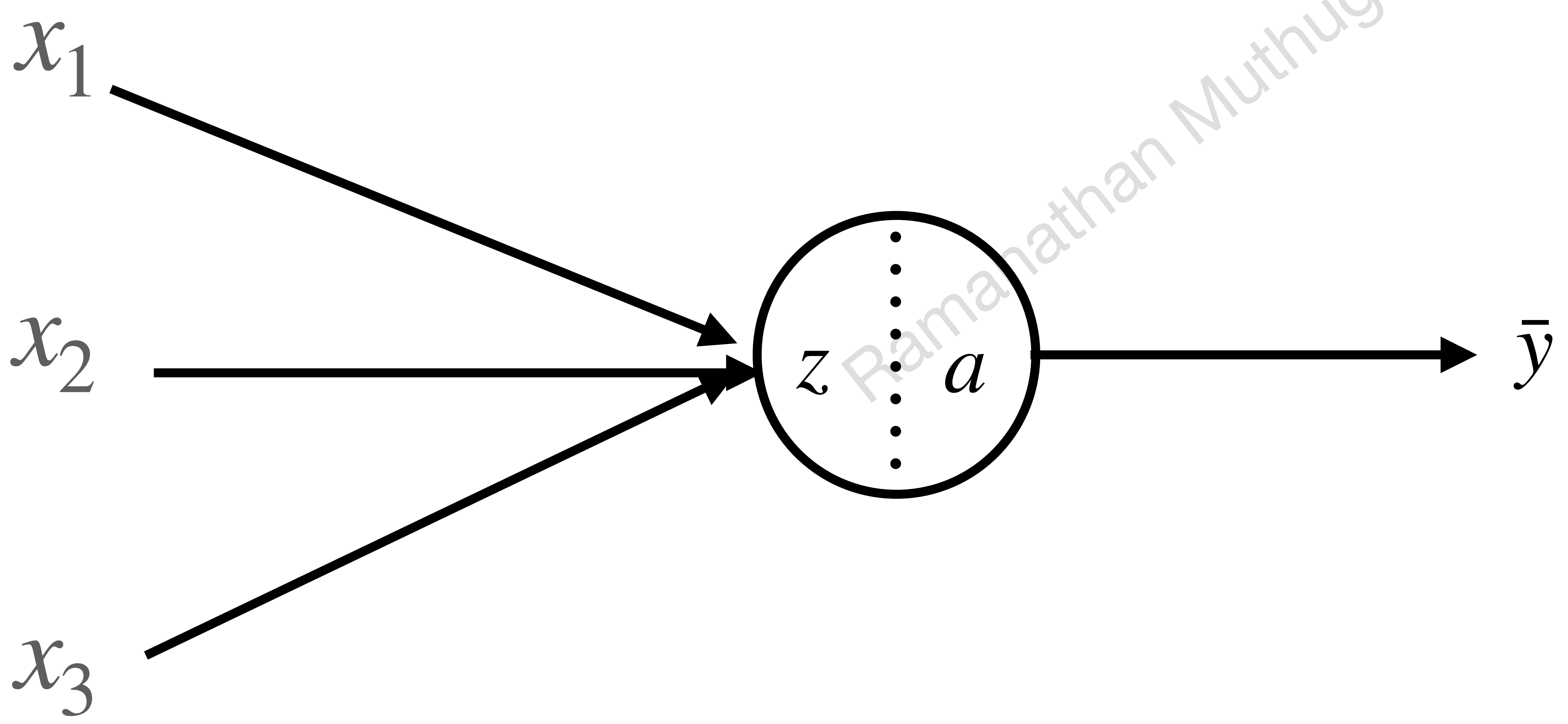
ED5340 - Data Science: Theory and Practise

L26 - Back-propagation in Neural Networks

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/

Logistic Regression - Pictorial



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

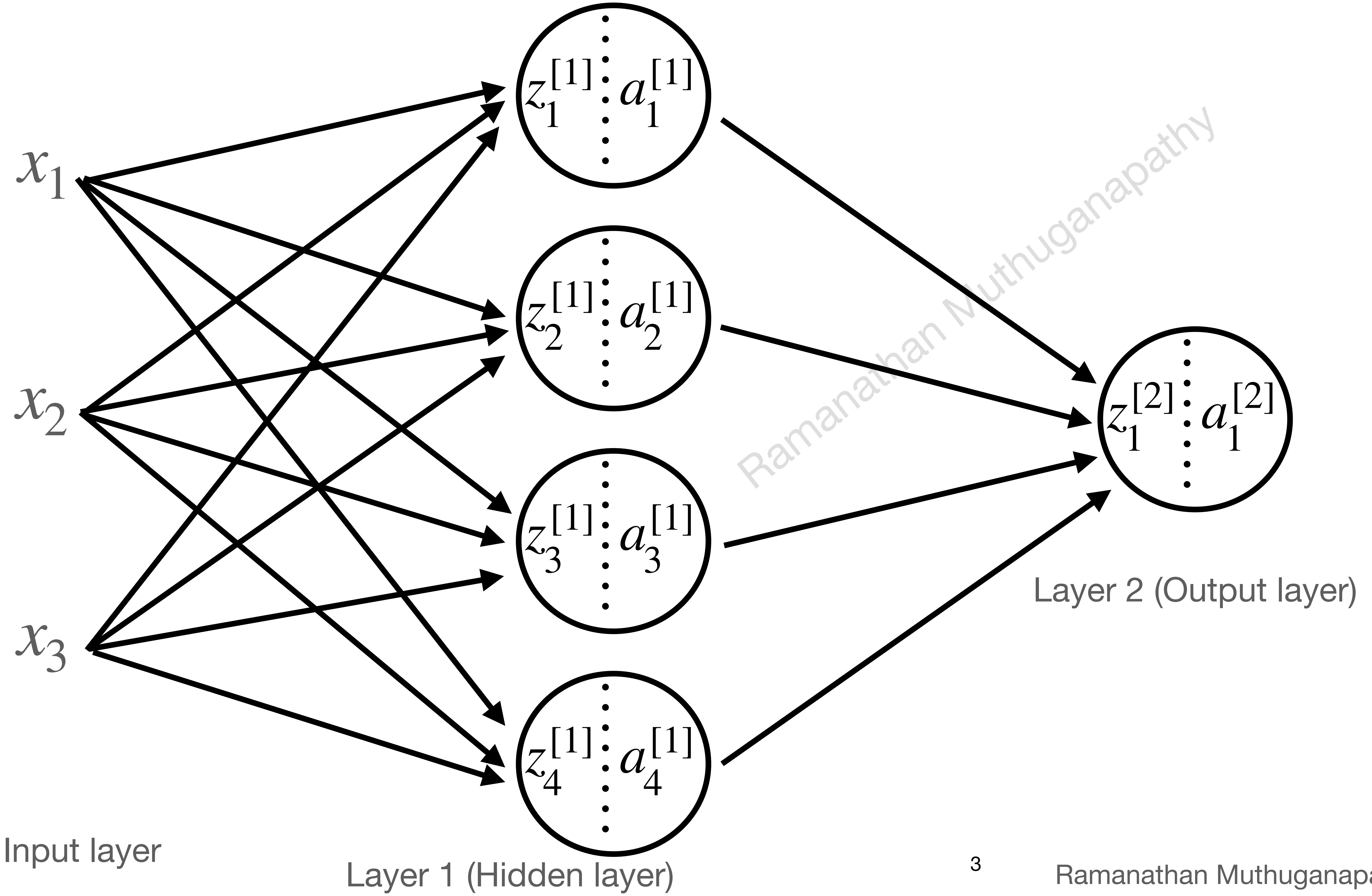
$$\bar{y} = a = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$J(a, y)$$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Neural Network - Pictorial

Two layer network! - Multi Layer Perceptron (MLP) - FFNN



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

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

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

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

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

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 $\mathbf{w}^{[1]^T}$

Vectorization (Matrix)

Layers 1 and 2 (for one sample) - dimensions

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

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

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

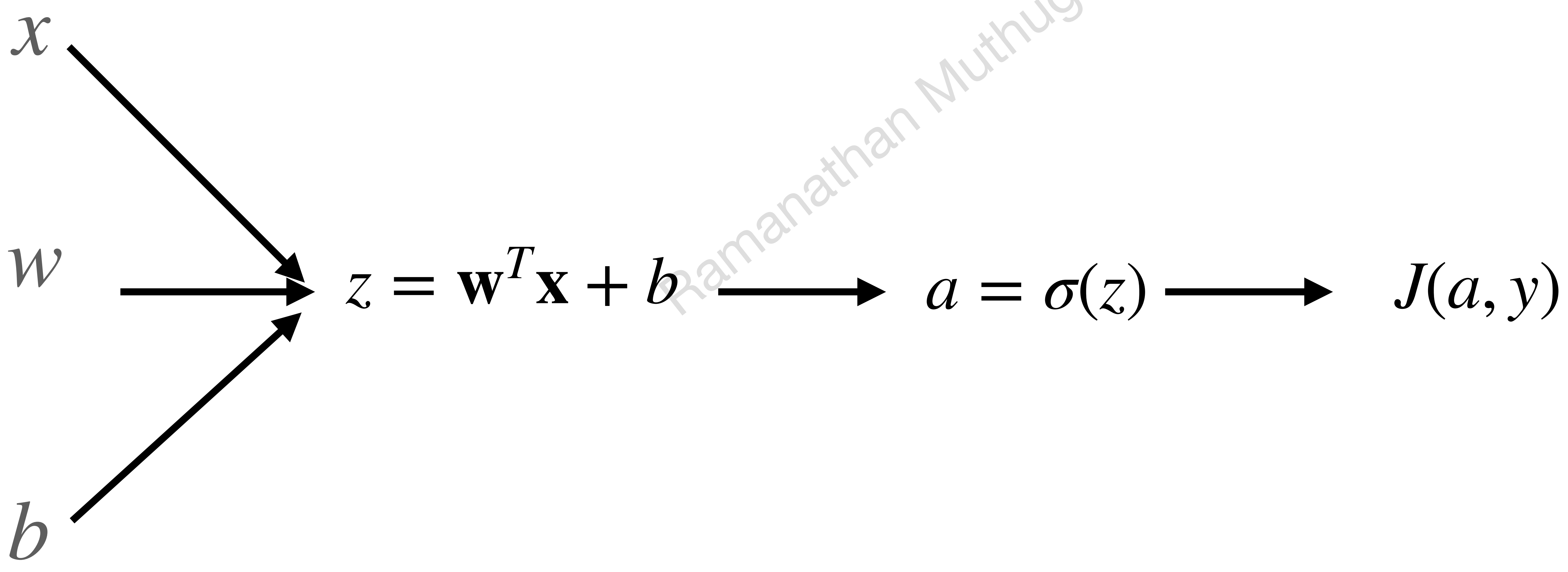
- $$\mathbf{X} = A^{[0]} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(m)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(m)} \\ x_3^{(1)} & x_3^{(2)} & \dots & x_3^{(m)} \end{bmatrix}$$

- $$Z^{[1]} = \begin{bmatrix} z_1^{[1](1)} & z_1^{[1](2)} & \dots & z_1^{[1](m)} \\ z_2^{[1](1)} & z_2^{[1](2)} & \dots & z_2^{[1](m)} \\ z_3^{[1](1)} & z_3^{[1](2)} & \dots & z_3^{[1](m)} \\ z_4^{[1](1)} & z_4^{[1](2)} & \dots & z_4^{[1](m)} \end{bmatrix}$$

Derivative of $\sigma(z)$

$$\sigma^1(z) = \sigma(1 - \sigma)$$

$$\frac{d\sigma}{dz} = a(1 - a)$$



Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Loss function and derivative

x

w

b

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

$$da = \frac{\partial J}{\partial a} \longleftarrow$$

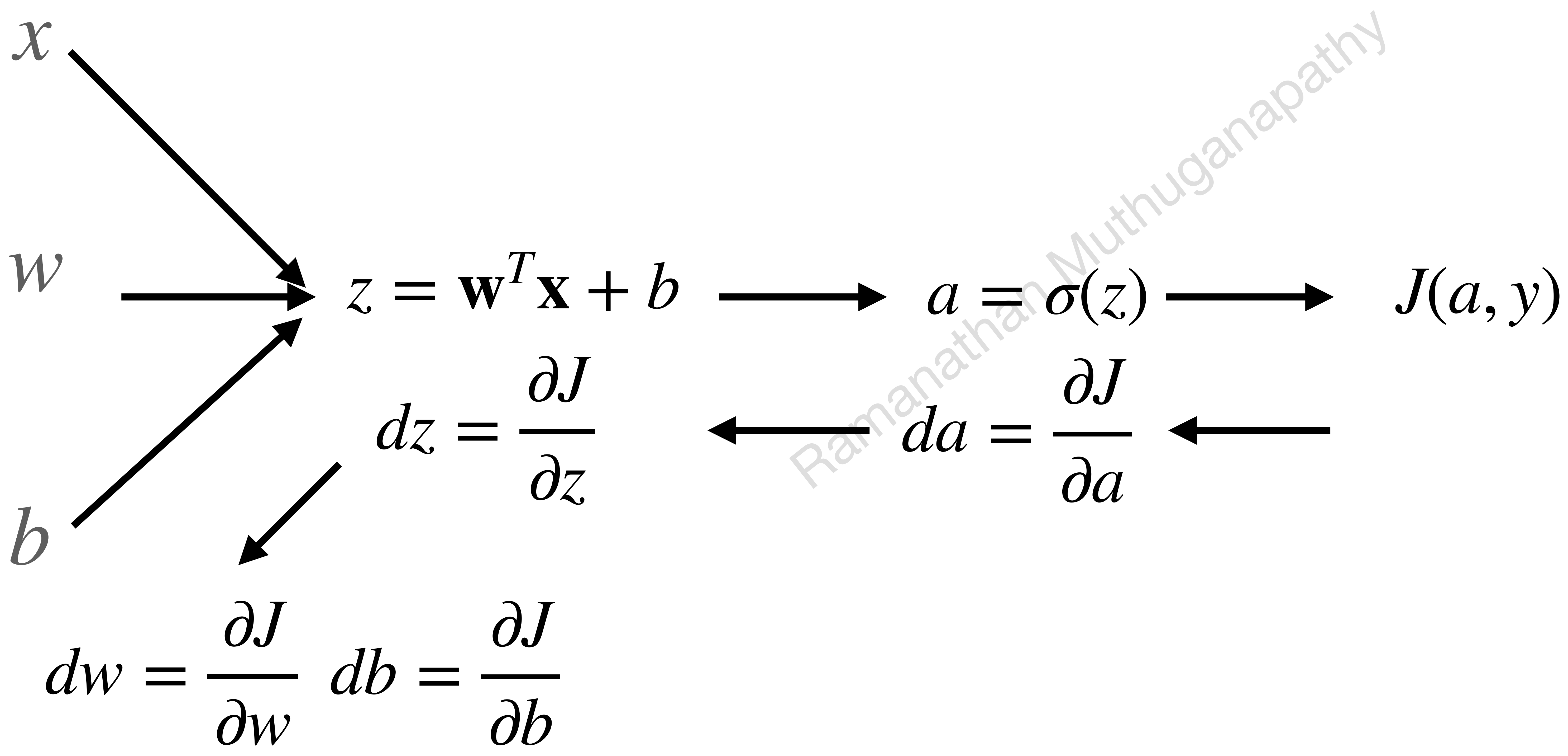
Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Loss function and derivative

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

$$dz = \frac{\partial J}{\partial z} \longleftarrow da = \frac{\partial J}{\partial a} \longleftarrow$$

Loss function and derivative



Loss function and derivative

$$J(a, y) = -y \log a - (1 - y)\log(1 - a)$$

$$\frac{\partial J}{\partial a} = -\frac{y}{a} + \frac{(1 - y)}{(1 - a)}$$

$$\begin{array}{l} x \searrow \\ w \rightarrow z = \mathbf{w}^T\mathbf{x} + b \longrightarrow a = \sigma(z) \longrightarrow J(a, y) \\ b \nearrow \end{array}$$

Loss function and derivative

$$J(a, y) = -y \log a - (1 - y)\log(1 - a)$$

$$\frac{\partial J}{\partial z} = \frac{\partial J}{\partial a}\frac{\partial a}{\partial z}$$

$$\frac{\partial J}{\partial z} = \frac{\partial J}{\partial a}\frac{\partial \sigma}{\partial z} \qquad \frac{\partial \sigma}{\partial z} = a(1 - a)$$

$$\frac{\partial J}{\partial z} = \left(-\frac{y}{a} + \frac{(y-1)}{(1-a)}\right)a(1 - a)$$

$$\frac{\partial J}{\partial z} = (a - y)$$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Loss function and derivative

$$\frac{\partial J}{\partial w} = \frac{\partial J}{\partial z}\frac{\partial z}{\partial w}$$

$$\frac{\partial J}{\partial w} = (a - y)x$$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Loss function and derivative

$$\frac{\partial J}{\partial b} = \frac{\partial J}{\partial z}\frac{\partial z}{\partial b}$$

$$\frac{\partial J}{\partial b} = (a - y)1$$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

For the NN

x

$$\mathbf{W}^{[1]} \longrightarrow z^{[1]} = \mathbf{W}^{[1]}x + b^{[1]} \longrightarrow a^{[1]} = \sigma(z^{[1]}) \xrightarrow{\mathbf{W}^{[2]},\ b^{[2]}} z^{[2]} = \mathbf{W}^{[2]}a^{[1]} + b^{[2]} \longrightarrow a^{[2]} = \sigma(z^{[2]}) \longrightarrow J(a^{[2]}, y)$$

$b^{[1]}$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

BP for the NN

x, $\mathbf{W}^{[1]}$, $b^{[1]}$ $\longrightarrow$

$$z^{[1]} = \mathbf{W}^{[1]}x + b^{[1]} \longrightarrow a^{[1]} = \sigma(z^{[1]}) \xrightarrow[b^{[2]}]{\mathbf{W}^{[2]}} z^{[2]} = \mathbf{W}^{[2]}a^{[1]} + b^{[2]} \longrightarrow a^{[2]} = \sigma(z^{[2]}) \longrightarrow J(a^{[2]}, y)$$

$$da^{[2]} \longleftarrow$$

BP for the NN

$$x,\ \mathbf{W}^{[1]},\ b^{[1]} \longrightarrow z^{[1]} = \mathbf{W}^{[1]}x + b^{[1]} \longrightarrow a^{[1]} = \sigma(z^{[1]}) \xrightarrow[b^{[2]}]{\mathbf{W}^{[2]}} z^{[2]} = \mathbf{W}^{[2]}a^{[1]} + b^{[2]} \longrightarrow a^{[2]} = \sigma(z^{[2]}) \longrightarrow J(a^{[2]}, y)$$

$$dz^{[2]} \longleftarrow da^{[2]} \longleftarrow$$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

BP for the NN

$$x,\ \mathbf{W}^{[1]},\ b^{[1]} \rightarrow z^{[1]} = \mathbf{W}^{[1]}x + b^{[1]} \rightarrow a^{[1]} = \sigma(z^{[1]}) \xrightarrow{\mathbf{W}^{[2]},\ b^{[2]}} z^{[2]} = \mathbf{W}^{[2]}a^{[1]} + b^{[2]} \rightarrow a^{[2]} = \sigma(z^{[2]}) \rightarrow J(a^{[2]}, y)$$

$$d\mathbf{W}^{[2]},\ db^{[2]} \leftarrow dz^{[2]} \leftarrow da^{[2]} \leftarrow$$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

BP for the NN

$$x,\ \mathbf{W}^{[1]},\ b^{[1]} \longrightarrow z^{[1]} = \mathbf{W}^{[1]}x + b^{[1]} \longrightarrow a^{[1]} = \sigma(z^{[1]}) \xrightarrow[b^{[2]}]{\mathbf{W}^{[2]}} z^{[2]} = \mathbf{W}^{[2]}a^{[1]} + b^{[2]} \longrightarrow a^{[2]} = \sigma(z^{[2]}) \longrightarrow J(a^{[2]}, y)$$

$$da^{[1]} \longleftarrow \begin{matrix} d\mathbf{W}^{[2]} \\ db^{[2]} \end{matrix} \longleftarrow dz^{[2]} \longleftarrow da^{[2]} \longleftarrow$$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

BP for the NN

$$x,\ \mathbf{W}^{[1]},\ b^{[1]} \longrightarrow z^{[1]} = \mathbf{W}^{[1]}x + b^{[1]} \longrightarrow a^{[1]} = \sigma(z^{[1]}) \xrightarrow[b^{[2]}]{\mathbf{W}^{[2]}} z^{[2]} = \mathbf{W}^{[2]}a^{[1]} + b^{[2]} \longrightarrow a^{[2]} = \sigma(z^{[2]}) \longrightarrow J(a^{[2]}, y)$$

$$dz^{[1]} \longleftarrow da^{[1]} \longleftarrow \begin{matrix} d\mathbf{W}^{[2]} \\ db^{[2]} \end{matrix} \longleftarrow dz^{[2]} \longleftarrow da^{[2]} \longleftarrow$$

BP for the NN

$$x,\ \mathbf{W}^{[1]},\ b^{[1]} \longrightarrow z^{[1]} = \mathbf{W}^{[1]}x + b^{[1]} \longrightarrow a^{[1]} = \sigma(z^{[1]}) \xrightarrow[b^{[2]}]{\mathbf{W}^{[2]}} z^{[2]} = \mathbf{W}^{[2]}a^{[1]} + b^{[2]} \longrightarrow a^{[2]} = \sigma(z^{[2]}) \longrightarrow J(a^{[2]}, y)$$

$$\begin{matrix} d\mathbf{W}^{[1]} \\ db^{[1]} \end{matrix} \longleftarrow dz^{[1]} \longleftarrow da^{[1]} \longleftarrow \begin{matrix} d\mathbf{W}^{[2]} \\ db^{[2]} \end{matrix} \longleftarrow dz^{[2]} \longleftarrow da^{[2]} \longleftarrow$$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Loss function and NN

$$J(a, y) = J(\mathbf{W}^{[1]}, b^{[1]}, \mathbf{W}^{[2]}, b^{[2]})$$

$$\frac{\partial J}{\partial \mathbf{W}^{[1]}}, \frac{\partial J}{\partial b^{[1]}}, \frac{\partial J}{\partial \mathbf{W}^{[2]}}, \frac{\partial J}{\partial b^{[2]}}$$

$$d\mathbf{W}^{[1]} = \frac{\partial J}{\partial \mathbf{W}^{[1]}}$$

$$db^{[1]} = \frac{\partial J}{\partial b^{[1]}}$$

$$d\mathbf{W}^{[2]} = \frac{\partial J}{\partial \mathbf{W}^{[2]}}$$

$$db^{[2]} = \frac{\partial J}{\partial b^{[2]}}$$

Gradient descent update

$$\mathbf{W}^{[1]} = \mathbf{W}^{[1]} - \alpha\, d\mathbf{W}^{[1]}$$

$$b^{[1]} = b^{[1]} - \alpha\, db^{[1]}$$

$$\mathbf{W}^{[2]} = \mathbf{W}^{[1]} - \alpha\, d\mathbf{W}^{[2]}$$

$$b^{[2]} = b^{[2]} - \alpha\, db^{[2]}$$

$$d\mathbf{W}^{[1]} = \frac{\partial J}{\partial \mathbf{W}^{[1]}}$$

$$db^{[1]} = \frac{\partial J}{\partial b^{[1]}}$$

$$d\mathbf{W}^{[2]} = \frac{\partial J}{\partial \mathbf{W}^{[2]}}$$

$$db^{[2]} = \frac{\partial J}{\partial b^{[2]}}$$

$$dZ^{[1]} = \frac{\partial J}{\partial Z^{[1]}}$$

$$dZ^{[2]} = \frac{\partial J}{\partial Z^{[2]}}$$

Back propagation

This set directly comes from logistic regression

$$\left.\begin{matrix} x \\ \mathbf{W}^{[1]} \\ b^{[1]} \end{matrix}\right\} \longrightarrow z^{[1]} = \mathbf{W}^{[1]}x + b^{[1]} \longrightarrow a^{[1]} = \sigma(z^{[1]}) \xrightarrow{\mathbf{W}^{[2]},\ b^{[2]}} z^{[2]} = \mathbf{W}^{[2]}a^{[1]} + b^{[2]} \longrightarrow a^{[2]} = \sigma(z^{[2]}) \longrightarrow J(a^{[2]}, y)$$

$$\longleftarrow \begin{matrix} d\mathbf{W}^{[2]} \\ db^{[2]} \end{matrix} \longleftarrow dz^{[2]} \longleftarrow da^{[2]} \longleftarrow$$

$$dz^{[2]} = a^{[2]} - Y \qquad \text{since} \quad \frac{\partial J}{\partial z} = (a - y)$$

$$dW^{[2]} = dz^{[2]}a^{[1]^T} \qquad \text{since} \quad \frac{\partial J}{\partial w} = (a - y)x$$

$$db^{[2]} = dz^{[2]}$$

Back propagation

$$dz^{[1]} = \frac{\partial J}{\partial z^{[1]}} = \frac{\partial J}{\partial z^{[2]}} \frac{\partial z^{[2]}}{\partial a^{[1]}} \frac{\partial a^{[1]}}{\partial z^{[1]}}$$

$$dz^{[1]} = \frac{\partial J}{\partial z^{[1]}} = W^{[2]^T} dz^{[2]} * a^{[1]^1}(z^{[1]})$$

(* denotes element-wise operation)

$$dW^{[1]} = dz^{[1]} x^T \qquad x = a^{[0]}$$

$$db^{[1]} = dz^{[1]}$$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Back propagation

Vectorized implementation

$$dZ^{[2]} = A^{[2]} - Y$$

$$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]^T}$$

$$db^{[2]} = \frac{1}{m} np.sum(dZ^{[2]}, axis = 1, keepdims = True)$$

$$dZ^{[1]} = A^{[2]} - Y$$

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$$

$$db^{[1]} = \frac{1}{m} np.sum(dZ^{[1]}, axis = 1, keepdims = True)$$

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Forward propagation

$$Z^{[1]} = \mathbf{W}^{[1]}\mathbf{X} + b^{[1]}$$

$$A^{[1]} = \sigma(Z^{[1]})$$

$$Z^{[2]} = \mathbf{W}^{[2]}A^{[1]} + b^{[2]}$$

$$A^{[2]} = \sigma(Z^{[2]})$$

Full procedure

Randomly initialise the weights

Repeat {

Comput FP

Compute BP (gradients)

Update the weights

} until convergence

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Deep neural networks

Architectures

- More number of hidden layers
- More units / hidden layer
- Advanced optimisation - stochastic gradient descent

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Deep neural networks

Examples (Images / Text / Speech)

- CNN (Images), Recurrent NN (text / speech)
- ResNet, VGGNet, GoogleNet (InceptionNet) etc.
- NeuralContours, Pix2Pix, RSCNN, etc.
- BRATS (medical imaging)

Deep neural networks

Point cloud data

- PointNet, PointNet++, PointCompletionNet (PCN)
- Point Transformer
- Attention-based

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Deep neural networks

Dataset - Images

- MNIST (handwritten digits)
- AlexNet
- ImageNet

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Deep neural networks

Dataset - 3D Models / Point Cloud / Sketches

- ShapeNet
- ModelNet
- CADNet
- CADSketchNet
- MCB (Mechanical Component Benchmark)
- ABC (A big CAD model dataset)

Deep neural networks

Dataset - 3D Models / Point Cloud / Sketches

- ShapeNet
- ModelNet
- CADNet
- CADSketchNet
- MCB (Mechanical Component Benchmark)
- ABC (A big CAD model dataset)

Deep neural networks

Typical problems that are done

- Classification
- Search and retrieval
- Shape completion
- Denoising
- Reconstruction
- Segmentation
- Sketch clean up

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Deep neural networks utilities
open sources

- TensorFlow
- Keras
- PyTorch

Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras