KAUST Academy & Tech Camp AI Week

Presented By: Ali Alqutayfi & Hassain Alsayhah

Linear Regression

Logistic Regression

Neural Networks

Deep Learning

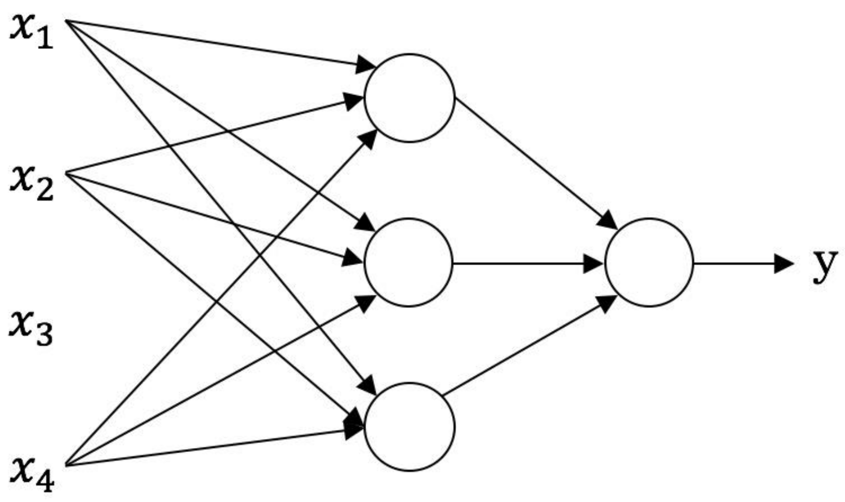
Artificial Intelligence and Machine Learning

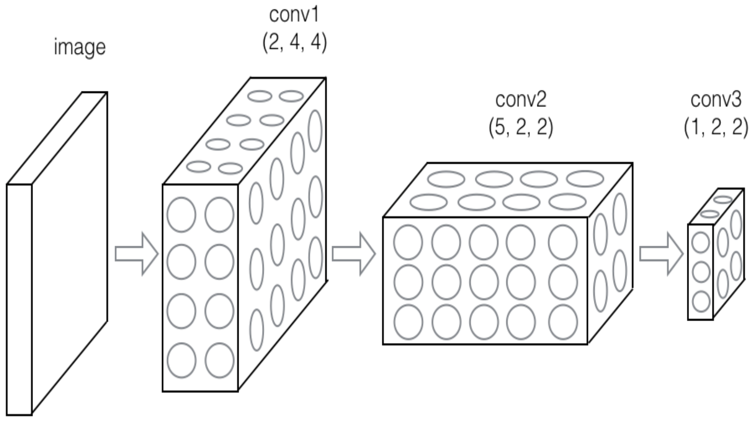
Neural Networks

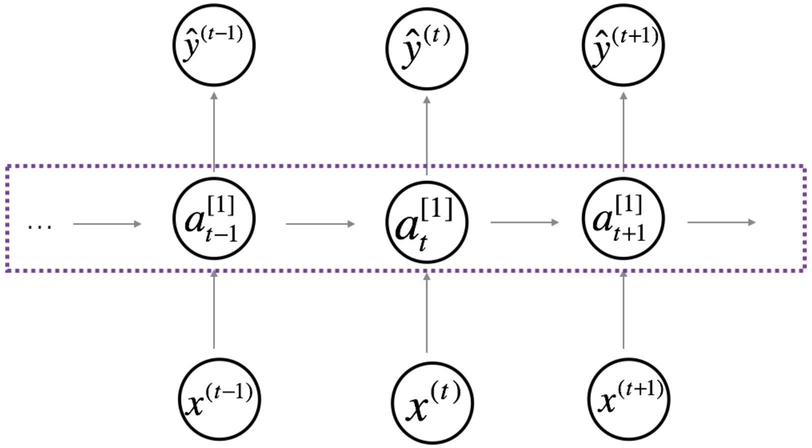
Lecture Outline

- Neural Networks
  - Forward pass
  - Backward pass
- Activation Functions

What is a Neural Network?





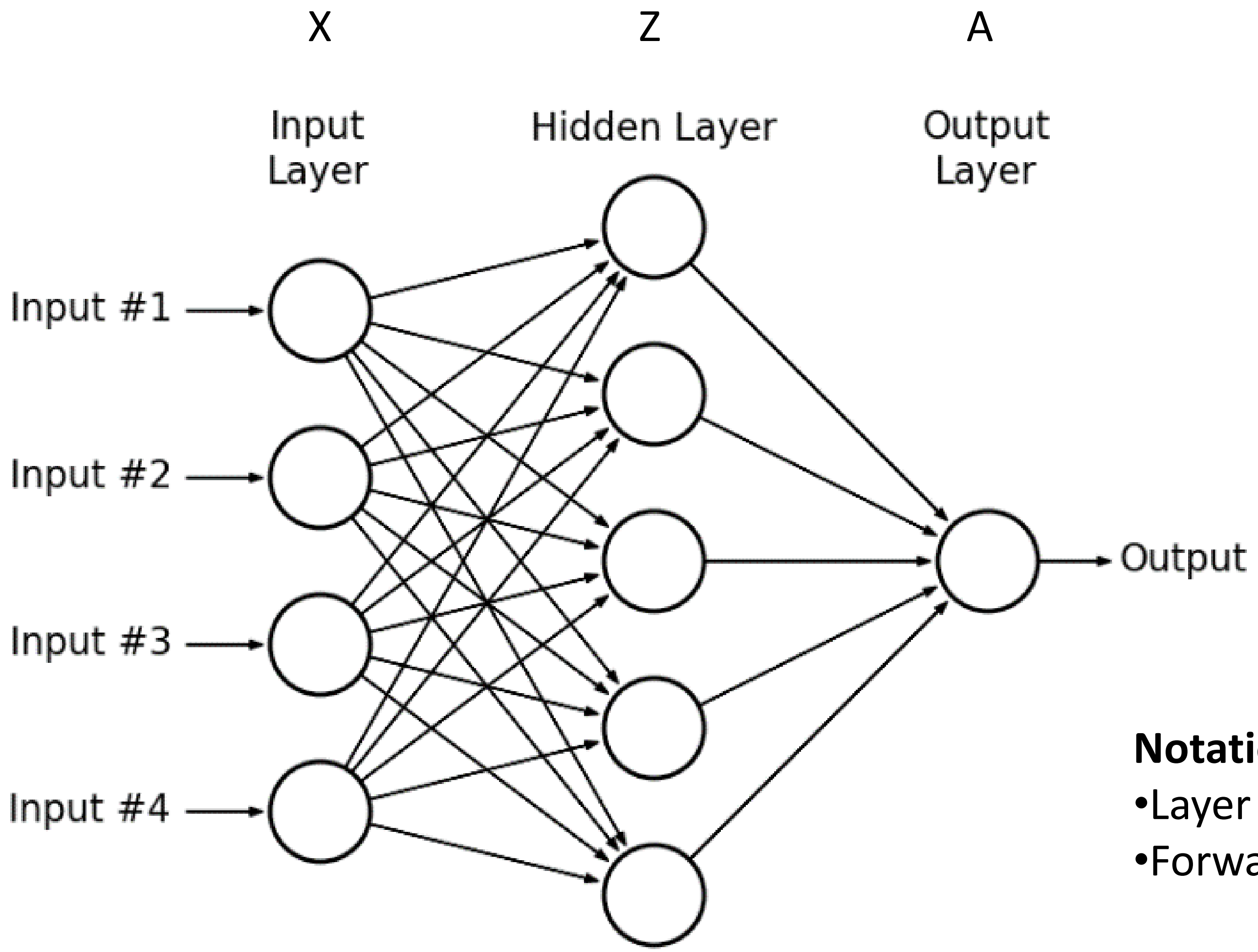


Standard NN

Convolutional NN

Recurrent NN

Neural Network Architecture

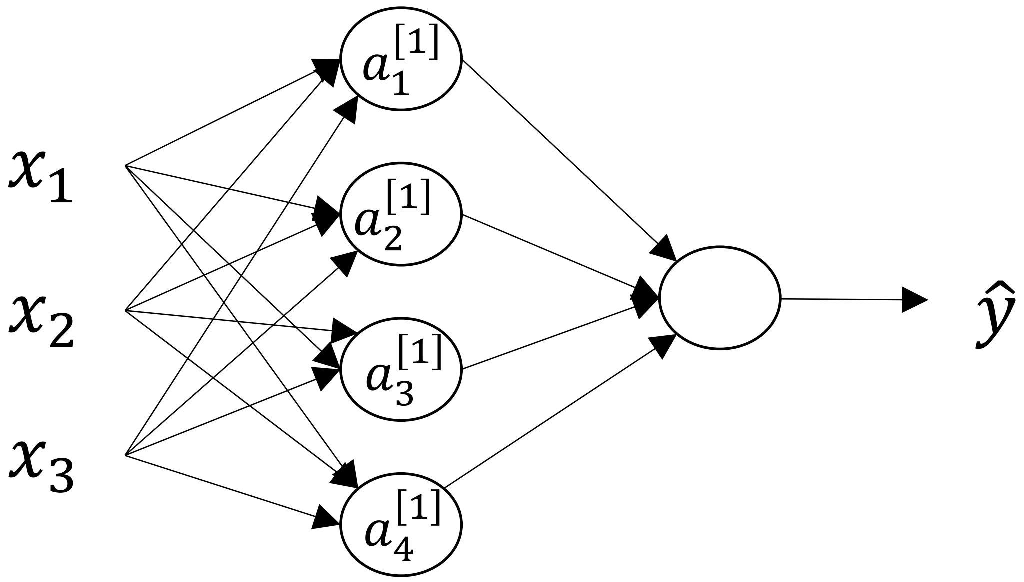


Notation:

•Layer l: W[l] (weights), b[l] (bias), a[l] (activations)

•Forward pass: z[l] = W[l]a[l-1] + b[l], a[l] = σ(z[l])

Forward Propagation



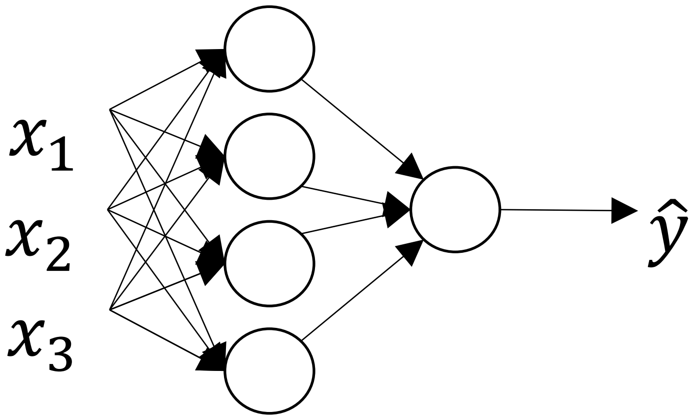
$$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]},\ a_1^{[1]} = \sigma(z_1^{[1]})$$

$$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]},\ a_2^{[1]} = \sigma(z_2^{[1]})$$

$$z_3^{[1]} = w_3^{[1]T} x + b_3^{[1]},\ a_3^{[1]} = \sigma(z_3^{[1]})$$

$$z_4^{[1]} = w_4^{[1]T} x + b_4^{[1]},\ a_4^{[1]} = \sigma(z_4^{[1]})$$

Vectorizin



$$X = \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ | & | & & | \end{bmatrix}$$

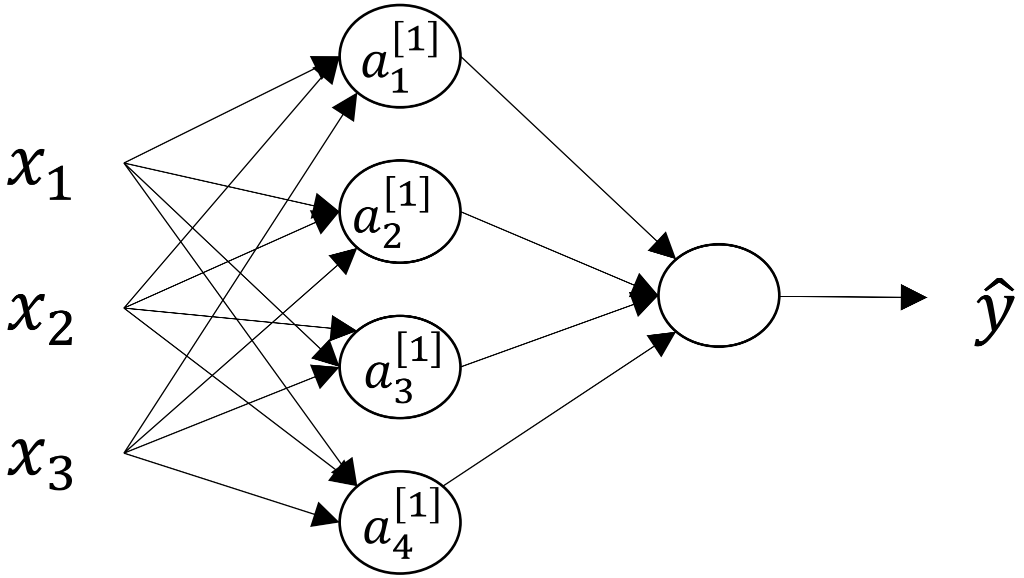
$$A^{[1]} = \begin{bmatrix} | & | & & | \\ a^{[1](1)} & a^{[1](2)} & \cdots & a^{[1](m)} \\ | & | & & | \end{bmatrix}$$

```
for i = 1 to m
```

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

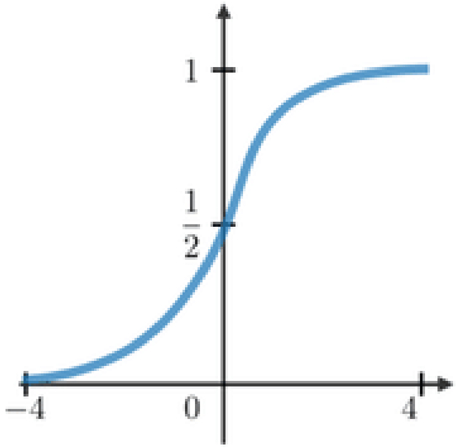
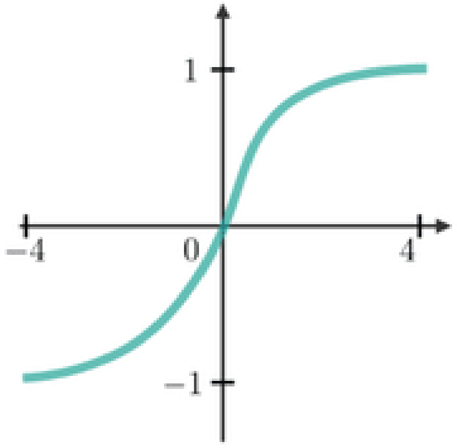
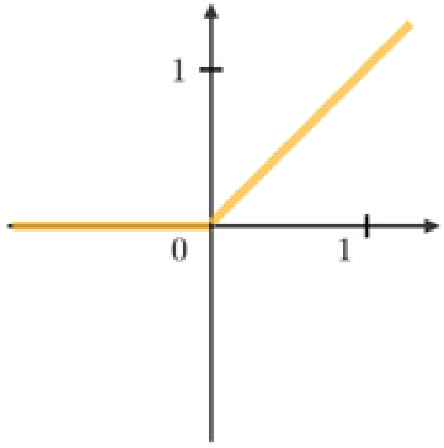
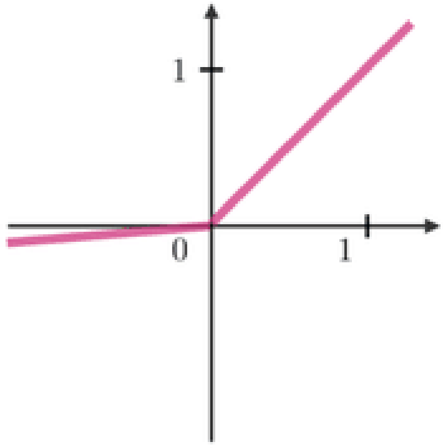
$$Z^{[1]} = W^{[1]}X + b^{[1]}$$
$$A^{[1]} = \sigma(Z^{[1]})$$
$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$
$$A^{[2]} = \sigma(Z^{[2]})$$

Forward Propagation



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

Activation Functions

Sigmoid	Tanh	ReLU	Leaky ReLU
$g(z) = \frac{1}{1+e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$	$g(z) = \max(\epsilon z, z)$ with $\epsilon \ll 1$
			

Why non-linear? Without activation functions, the network becomes a linear model regardless of depth.

Backpropagation & Gradient Descent

Core Algorithm:

- **Forward pass**: Compute predictions
- **Compute loss**: $L(\hat{y}, y)$
- **Backward pass**: Calculate gradients
- **Update parameters**: $W := W - \alpha \cdot dW$, $b := b - \alpha \cdot db$

Computing gradients

Logistic regression

$$\begin{tikzcd}
x \arrow[dr] & & & \\
w \arrow[r] & \boxed{z = w^T x + b} \arrow[r] & \boxed{a = \sigma(z)} \arrow[r] & \boxed{\mathcal{L}(a, y)} \\
b \arrow[ur] & & &
\end{tikzcd}$$

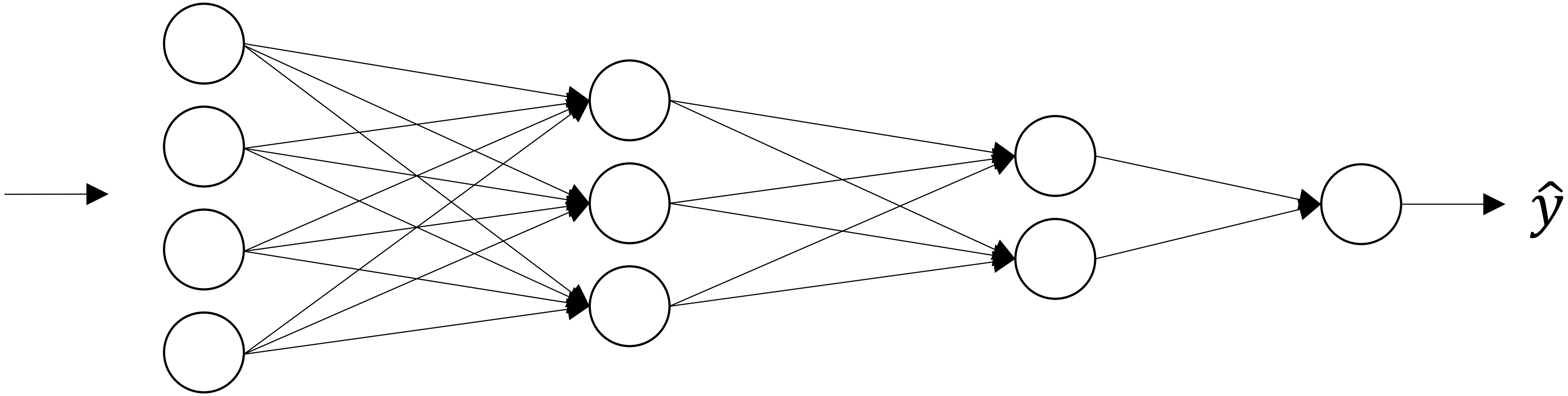
Andrew Ng

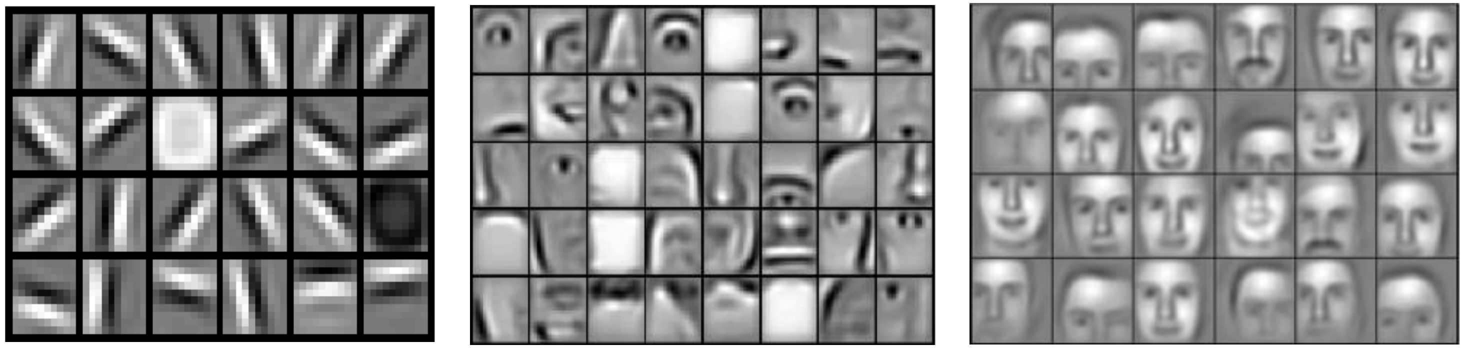
Computing gradients

Computing gradients

Matrix Dimensions

Intuition about deep representation





Why Deep Learning Works Now?

Three Key Factors:

- 🗃️ **Data**: Massive datasets available
- 💻 **Computation**: GPUs, specialized hardware
- 🔬 **Algorithms**: Better architectures, optimization

Supervised Learning

Input(x)	Output (y)	Application
Home features	Price	Real Estate
Ad, user info	Click on ad? (0/1)	Online Advertising
Image	Object (1,…,1000)	Photo tagging
Audio	Text transcript	Speech recognition
English	Chinese	Machine translation
Image, Radar info	Position of other cars	Autonomous driving

Let’s Code

We will implement Neural Networks using the following methods:

- Using NumPy
- Using Pytorch: A library specialized in building AI architectures

Exercise

Search about CIFAR10 dataset and build a classification model for it using the following ways:

- NumPy
- Pytorch