Artificial Neural Networks

Lecture 13 – HCCDA-AI

Imran Nawar

19 July 2025

Course Progress: Where We Are and What's Ahead

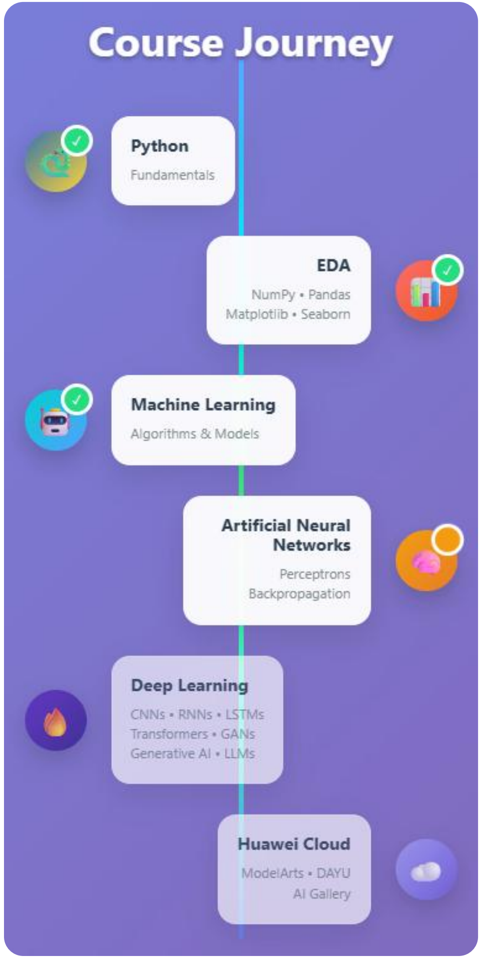
Python

- Python Fundamentals

Exploratory Data Analysis

- NumPy
- Pandas
- Data Visualization
  - Matplotlib
  - Seaborn

- **Machine Learning**
- **Artificial Neural Networks**
- **Deep Learning**
  - Convolutional Neural Networks (Deep Computer Vision)
  - Sequence Learning
  - Deep Generative AI
  - Large Language Models
- **Huawei Cloud AI Services**
  - ModelArts (AI Development Platform)
  - DAYU (Data Processing)
  - AI Gallery & Pre-trained Models
  - ...



Course Progress Overview

- **Lecture 1**:
  - Introduction to Programming, Installation and Setup
  - Variables, Data Types → *int, float, str, bool, list, tuple, dict*
  - Conditional Statements → *if, if-else, if-elif-else*
  - Loops → *while, for (range(), zip(), enumerate(), break, continue, pass)*

- **Lecture 2**:
  - Functions, Types of arguments → *positional , keyword, *args*
  - Programming Paradigms
  - Object Oriented Programming: Classes, Objects, Attributes, Methods
  - Constructor → __init__ Method

- **Lecture 3**:
  - Advanced OOP → (*Inheritance, Polymorphism, Abstraction, Encapsulation*)

- **Lecture 4**:
  - Exception Handling, File Handling
  - Exploratory Data Analysis, Python libraries overview for EDA
  - NumPy Library → *numpy array, slicing, indexing* etc.

Course Progress Overview

- **Lecture 5**:
  - Exploratory Data Analysis, Python libraries overview for EDA
  - Pandas Library → *Series, DataFrame, indexing and selection*
  - Handling missing data → *isnull(), fillna(), dropna()*
  - Data Visualization with pandas, Lab → *Titanic dataset analysis*

- **Lecture 6**:
  - Data Visualization → *Overview, Importance, what is data, sources of data*
  - Common Types of Plots → *Bar, line, scatter, histogram, pie chart*
  - Data Visualization Tools → *Matplotlib, Seaborn, Plotly*
  - Advanced Plotting Techniques → *Subplots, 3D plots, Network Graphs, Choropleth Maps, Contour plots*
  - Time Series Data, Interactive Visualization, Forecasting Stock Prices with LSTM

- **Lecture 7**:
  - Machine Learning → *Traditional Programming vs. ML, Mathematics for ML*
  - Types of ML → *Supervised, Unsupervised, Reinforcement Learning*
  - Key Concepts → *Data and Features, Algorithms and Models, Training and Testing*
  - Common Algorithms, Tools and Frameworks, Challenges and Limitations, Future Trends

Course Progress Overview

- **Lecture 8**:
  - Simple Linear Regression
  - Hypothesis → $h_\theta(x) = \theta_0 + \theta_1 x$
  - Cost Function, Gradient Descent algorithm, Gradient of cost function
  - Code → *Manual Implementation of Gradient Descent*

- **Lecture 9**:
  - Version Control → *Git, GitHub, What is Repository, Creating Repositories.*
  - Updating repositories, .gitignore file, requirments.txt, Basic Git Flow
  - Contributing to open source projects
  - Multiple Linear Regression → *Hypothesis, Cost Functions, Gradient Descent*

- **Lecture 10**:
  - Normal Equation, Polynomial Regression, Code for Multiple Linear Regression
  - Logistic Regression → *Hypothesis, Sigmoid Function, Cost Function, Gradient Descent*
  - Code for Logistic Regression

- **Lecture 11**:
  - Regularization → *The problem of overfitting*
  - Practical Machine Learning with Scikit Learn

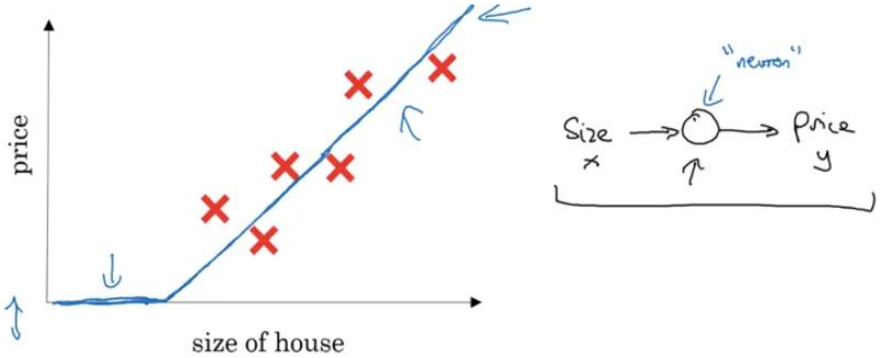
Course Progress Overview

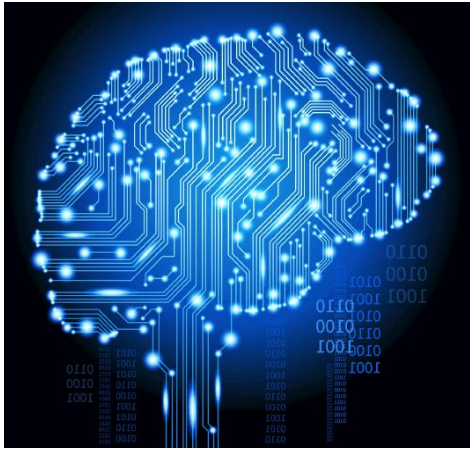
- **Lecture 12**:
  - Regression Project → House Price Prediction
  - Feature Descriptors and Face Detection → *ORB, SIFT, Hough Transform, Haar Cascade*
  - Classification Project → *Image Classification*

- **Coming Up Next**: **Artificial Neural Networks**

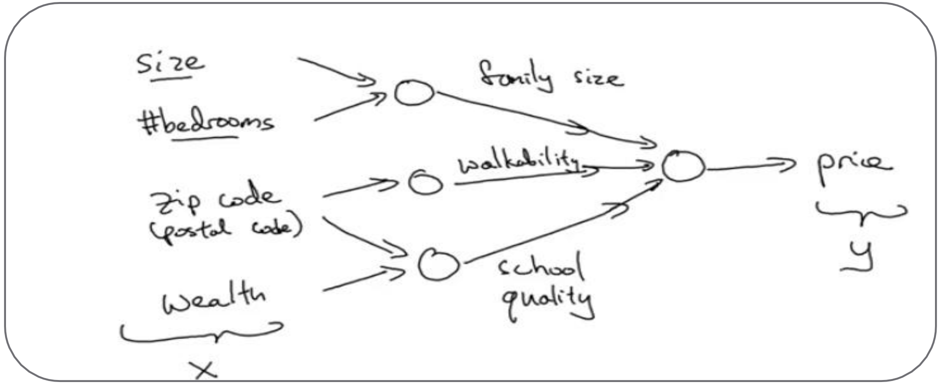
Artificial Neural Networks (ANNs)

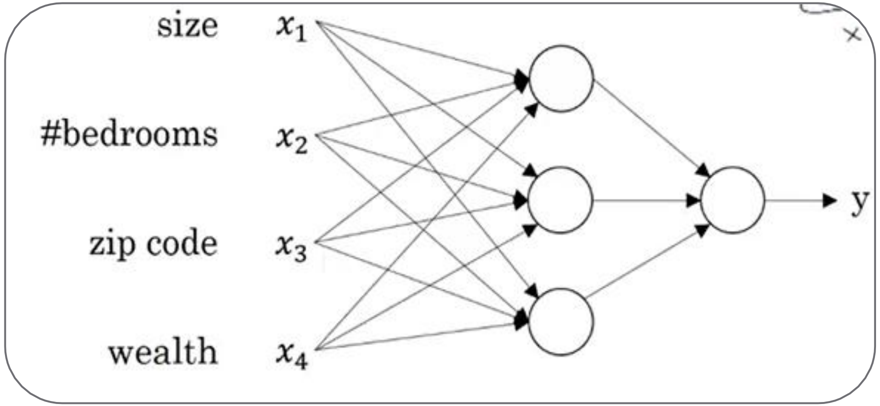
- Neural networks are a core technique in modern AI.
- Inspired by how the human brain processes information.
- It consists of interconnected nodes (artificial neurons) that transform input data through weighted connections and activation functions.
- ANNs are used for classification, prediction, and pattern recognition across various domains.





Housing Price Prediction





Given these input features, the job of the neural network will be to predict the price y.

Supervised Learning with Neural Networks

Input (x)	Output (y)	Application
Home features	Price	Real Estate
Image	Objects (1, …., 1000)	Photo tagging
Audio	Text Transcript	Speech recognition
English	Chinese	Machine translation
Image, Radar Info	Position of other cars	Autonomous driving

Standard NN

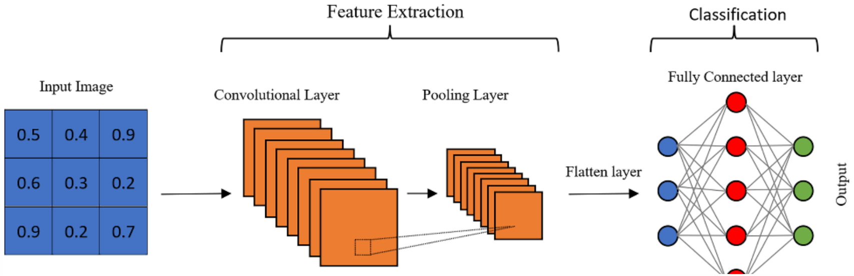
CNN

RNN

Custom

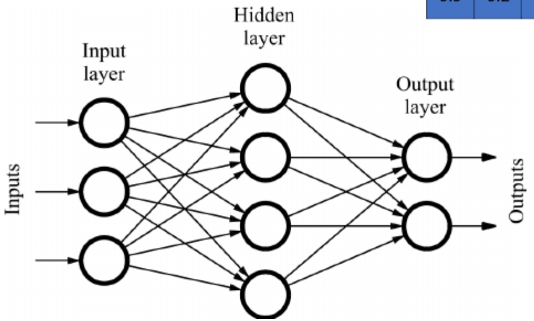
Neural Network Examples

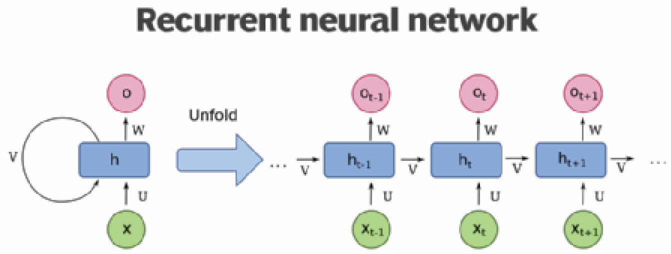
Type	Description	Application
Feedforward NN	Standard fully-connected ANN	Tabular data
CNN	Convolutional Neural Network	Image recognition
RNN	Recurrent Neural Network	Time series, text, speech



CNN

Feedforward NN

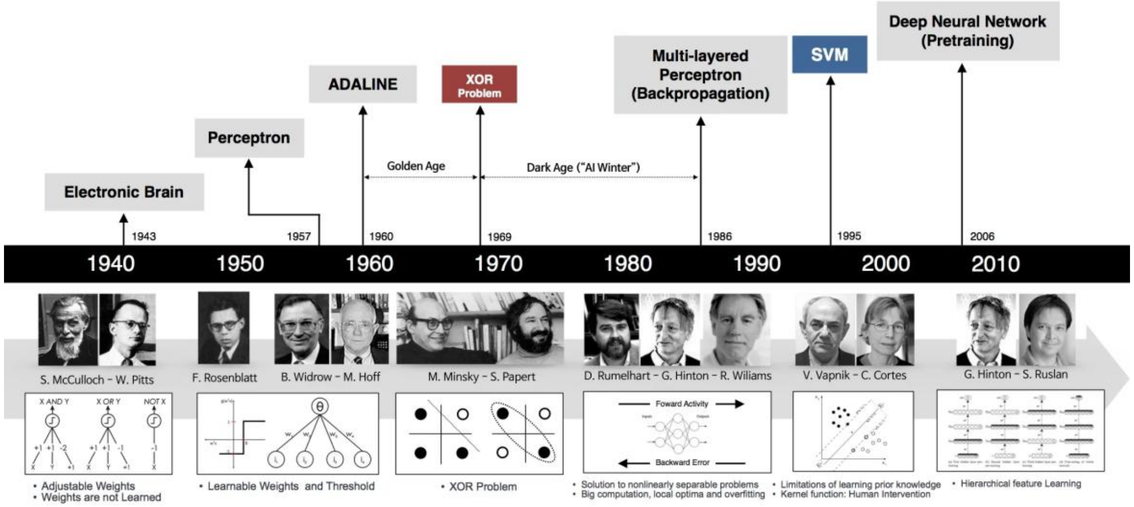




Evolution of Neural Networks

Origins:

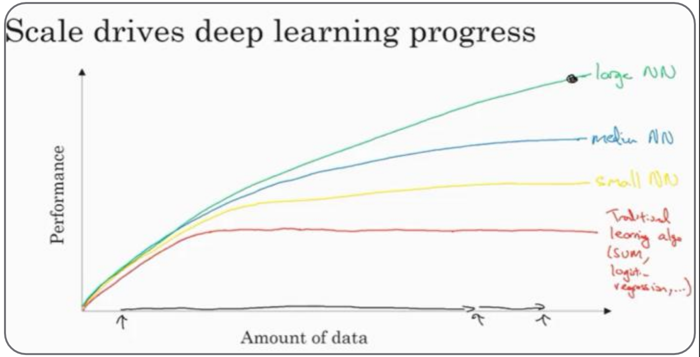
- Inspired by the structure of the human brain.
- Early research dates back to 1943 (McCulloch & Pitts models).
- Gained popularity in the 1980s and early 1990s.
- Interest declined in the late 1990s due to computational limitations and the rise of simpler model like Support Vector Machines (SVMs).



Recent Resurgence of Neural Networks

Why Now?

- Around **2005**, interest in neural networks surged again due to:
  - **Improved hardware** (GPUs, TPUs for parallel processing).
  - **Large datasets** (Big Data, Internet-scale data availability).
  - **Algorithmic advancements** (better activation functions, optimization techniques like Adam).

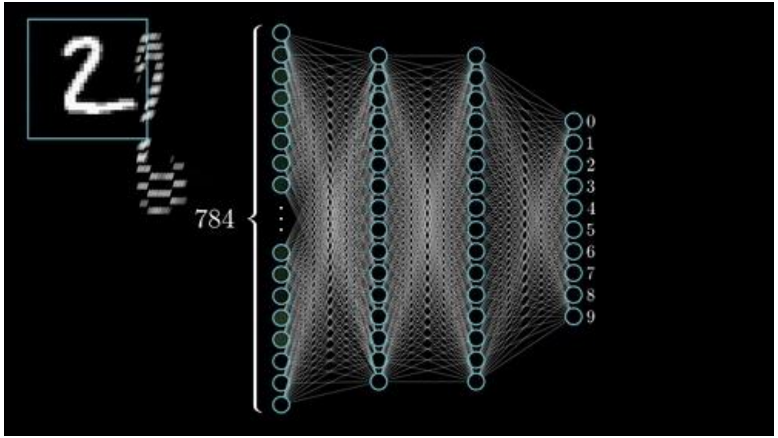


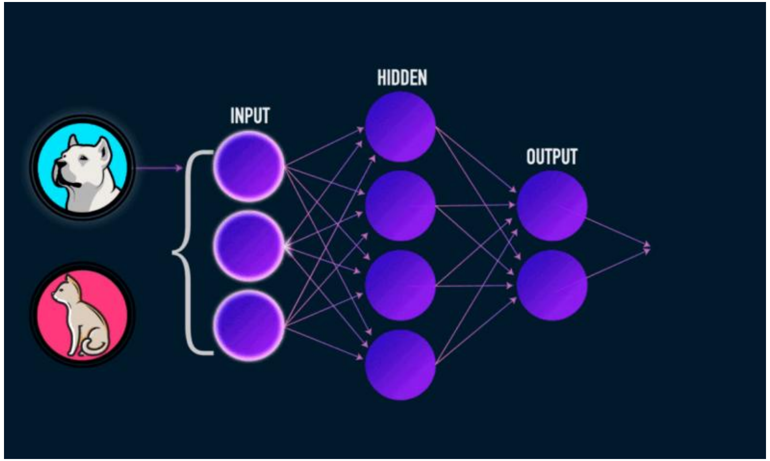
Why Neural Networks Matter?

- Neural networks power state-of-the-art applications in:
  - **Speech Recognition** (e.g., Siri, Google Assistant)
  - **Computer Vision** (e.g., Face Recognition, Object Detection)
  - **Natural Language Processing (NLP)** (e.g., ChatGPT, Google Translate)
- With enough data and computational power, they often outperform traditional machine learning.

Neural Networks – How They Work

- Neural networks learn patterns by adjusting weights of connections between neurons.
- Learning is done through backpropagation, which minimizes prediction errors using gradient descent.
- Deep networks, known as Deep Neural Networks (DNNs), consist of multiple hidden layers for hierarchical feature learning.





Neural Networks: Representation

Neurons and the brain

Biological Inspiration: The Human Brain

Biological Neurons:

- Humans have ~86 billion neurons.
- Neurons communicate via electrical and chemical signals.
- Each neuron may connect to thousands of others.
- Information processing occurs through complex networks.

From Biology to Computation:

- Artificial neurons are simplified mathematical models.
- They capture the essence of biological neural communication:
  - Receive inputs (dendrites)
  - Process signals (cell body)
  - Transmit output (axon)
- Neurons receive, process, and send signals
- Artificial neurons mimic this behavior using simple mathematical functions
- While vastly simplified, this model has proven remarkably effective

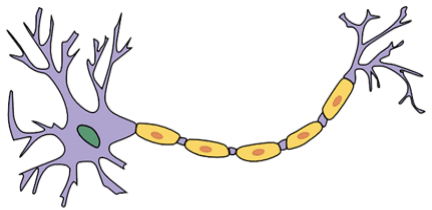


Fig: Biological Neuron

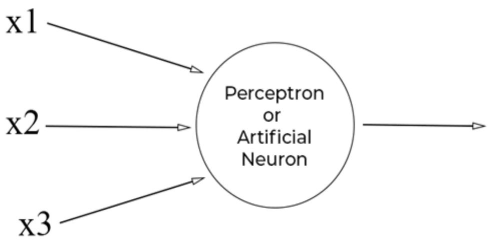
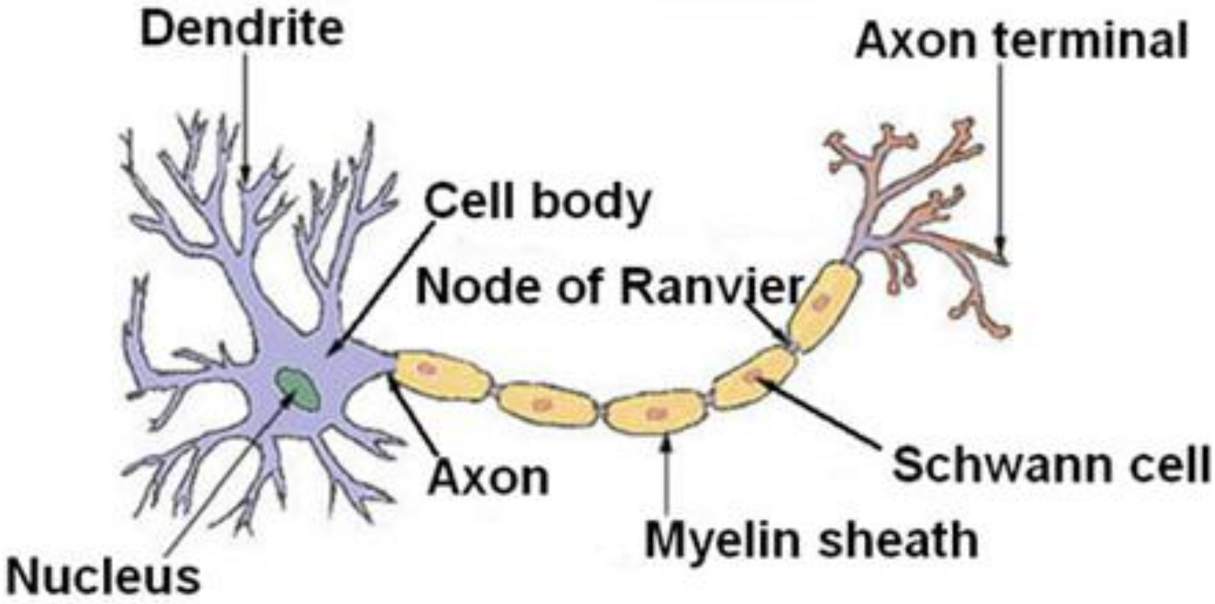
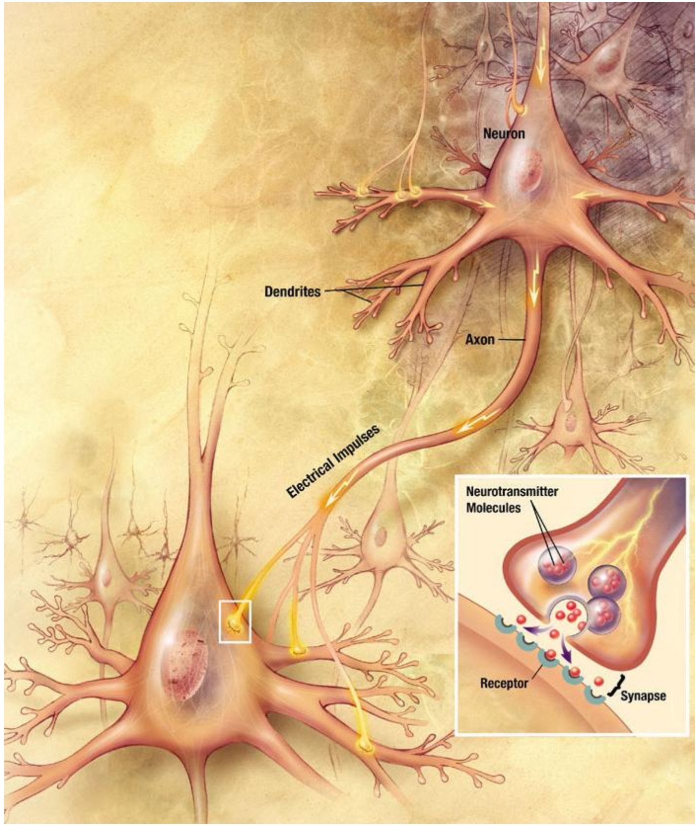


Fig: Artificial Neuron

Neuron in the brain



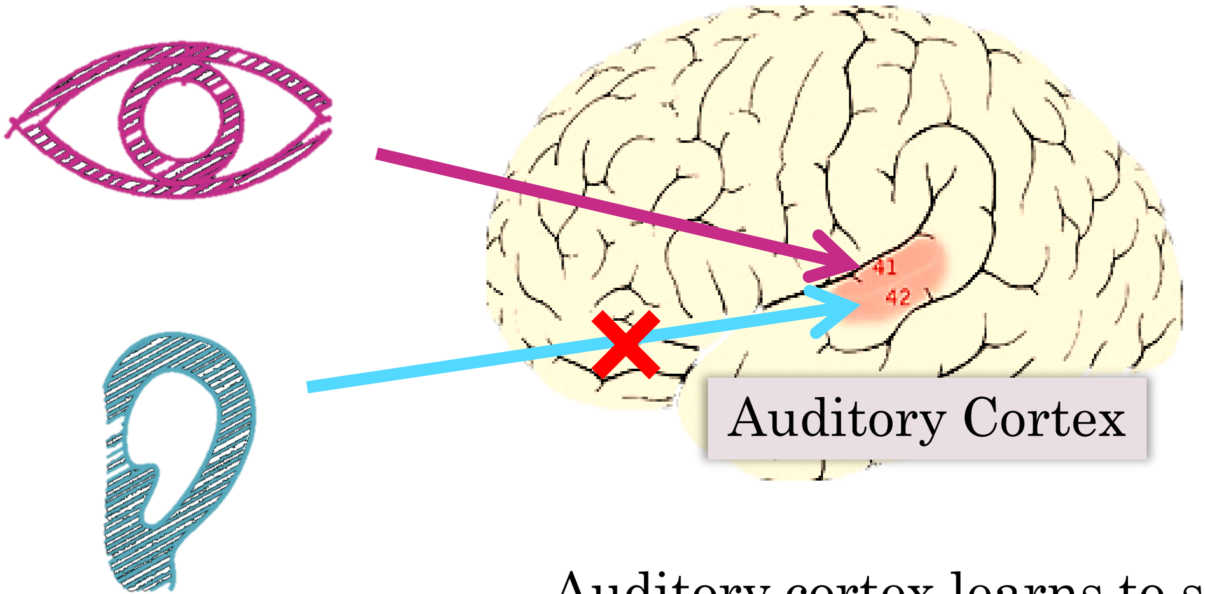
Neurons in the brain



[Credit: US National Institutes of Health, National Institute on Aging]

The “one learning algorithm” hypothesis

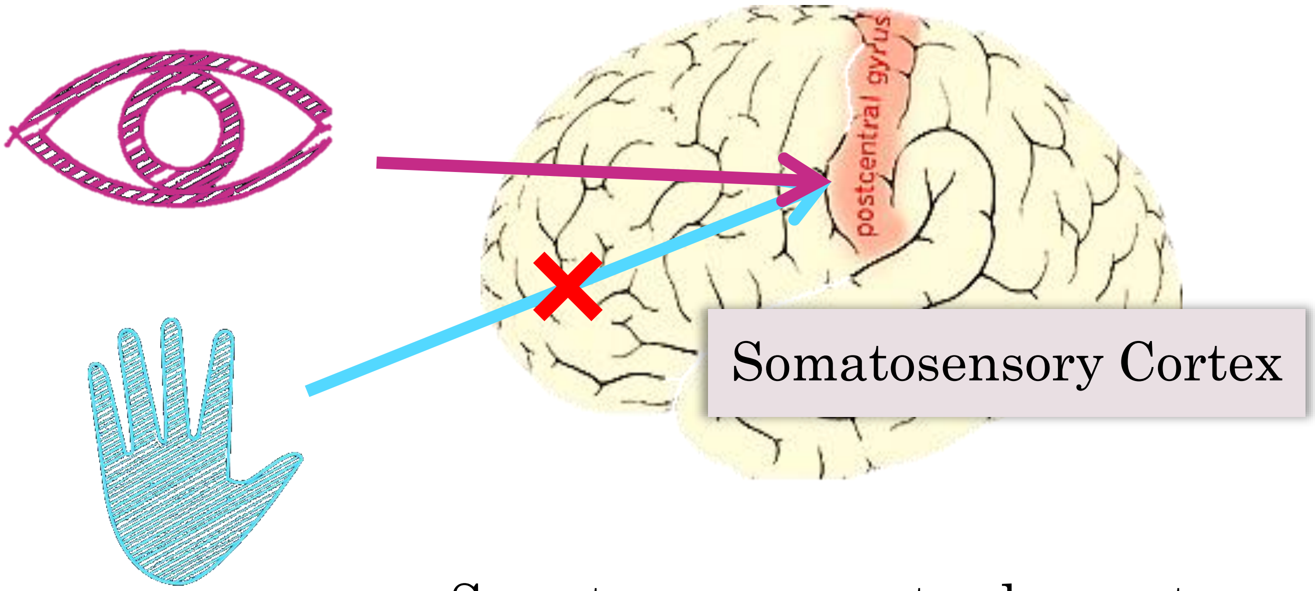
- This hypothesis suggests that the human brain does not operate with thousands of specialized algorithms.
- Instead, it relies on a single general learning mechanism to perform different tasks.
- Artificial neural networks aim to replicate this capability in a computational model.



Auditory cortex learns to see

[Roe et al., 1992]

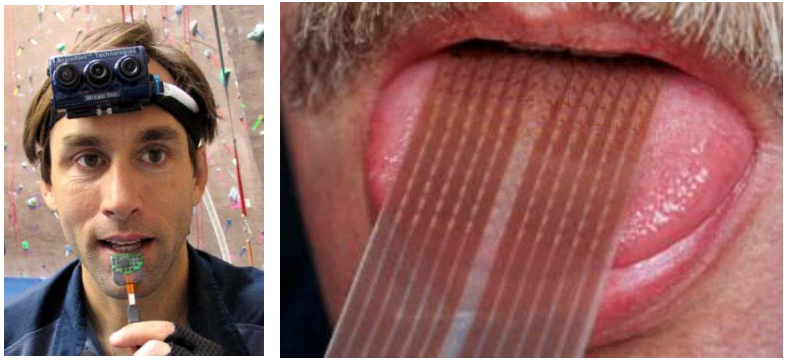
The "one learning algorithm" hypothesis



Somatosensory cortex learns to see

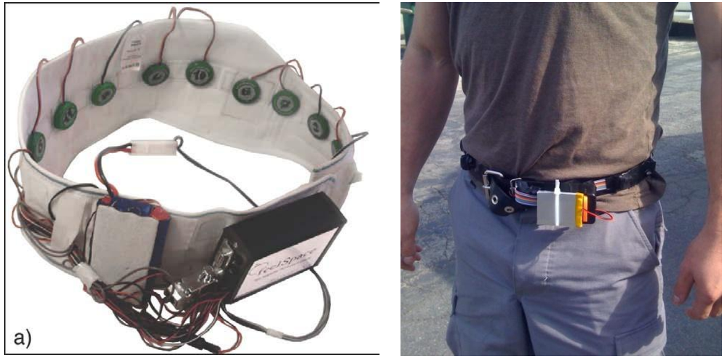
[Metin & Frost, 1989]

Sensor representations in the brain

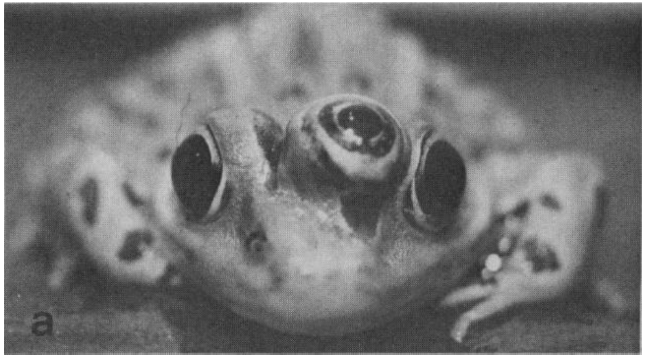


Seeing with your tongue

Human echolocation (sonar)



Haptic belt: Direction sense



Implanting a 3rd eye

[BrainPort; Welsh & Blasch, 1997; Nagel et al., 2005; Constantine-Paton & Law, 2009]

Neural Networks: Representation

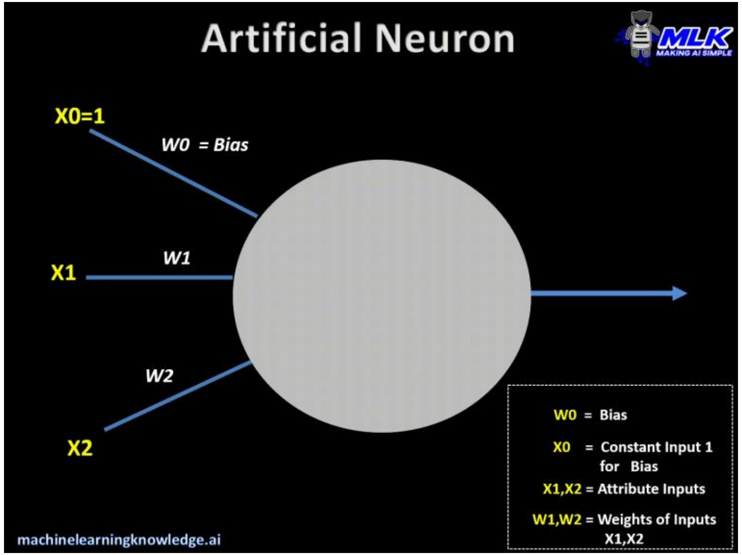
Model Representation I

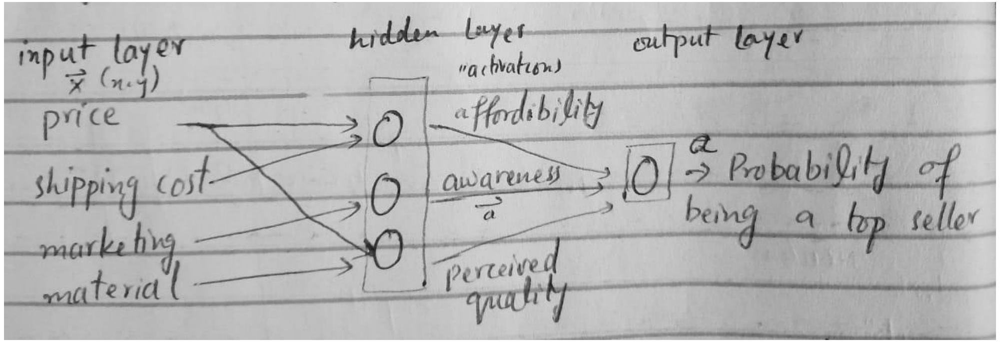
Neural Network Architecture: Layers and Connections

- A neuron in an artificial neural network is a simplified model compared to biological neurons.

- A neural network consists of:
  - **Input Layer**: Takes raw data (features) as input.
  - **Hidden Layers**: Apply transformations to learn representations.
  - **Output Layer**: Produces final predictions (classification, regression, etc.).

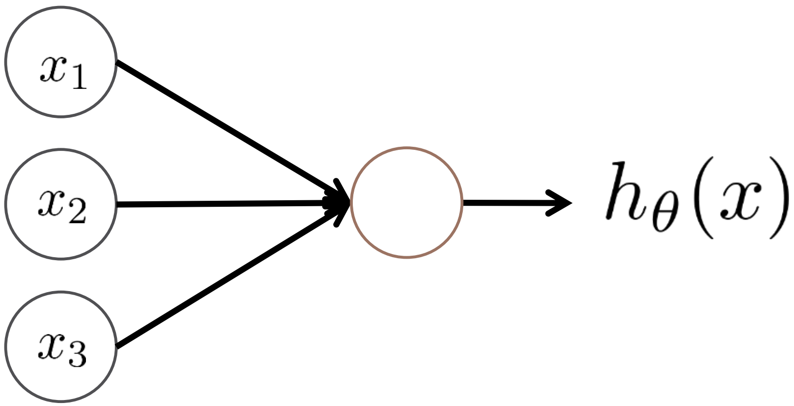
- **Fully connected networks (Dense Networks)**:
  - Each neuron in a layer connects to all neurons in the next layer.





Neuron model: Logistic unit

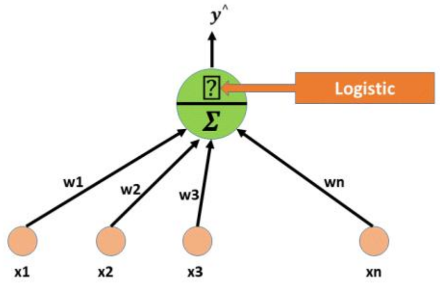
$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

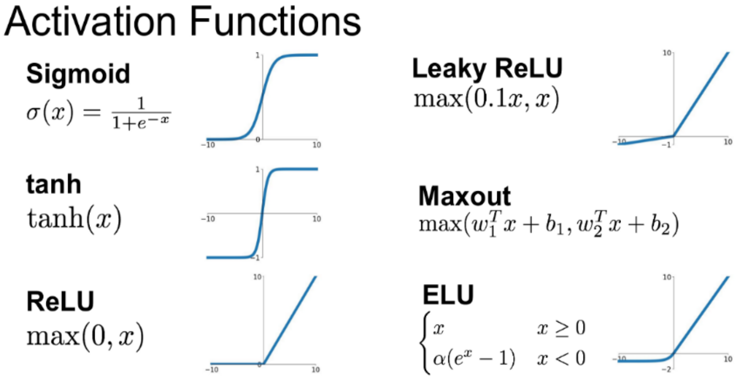


Sigmoid (logistic) activation function.

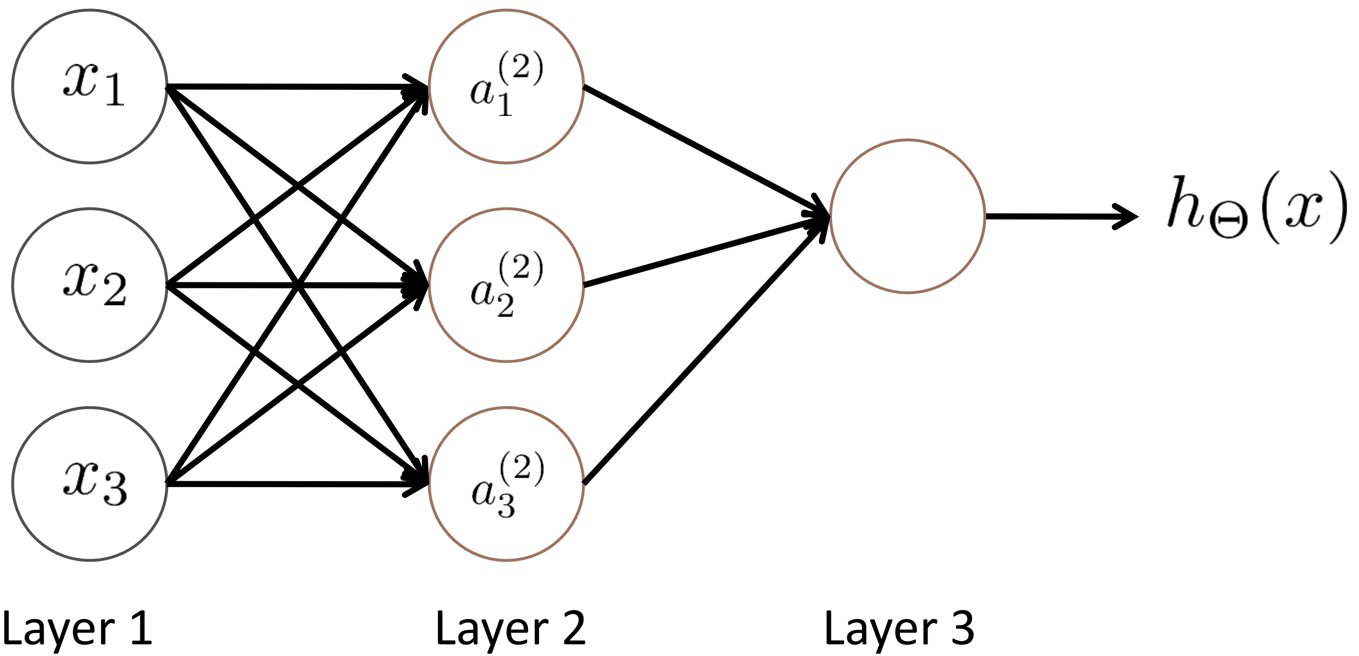
Activation Functions

- Activation functions introduce non-linearity into the neural network, enabling it to learn and model complex patterns beyond linear relationships.
- **Common activation functions:**
  - **Sigmoid:**
    - **Output range:** *(0, 1)*
    - Good for probability based outputs (e.g., binary classification)
    - Can suffer from vanishing gradients, especially in deep networks
  - **Tanh (Hyperbolic Tangent):**
    - **Output range:** *(-1, 1)*
    - Centered around zero → better for optimization than sigmoid
    - Still prone to vanishing gradient issues.
  - **ReLU: (Rectified Linear Unit):**
    - **Output:** *max(0, x)*
    - Most widely used due to computational efficiency
    - Helps reduce vanishing gradient problem
    - Can lead to “dead neurons” (zero gradients) if inputs are always negative.



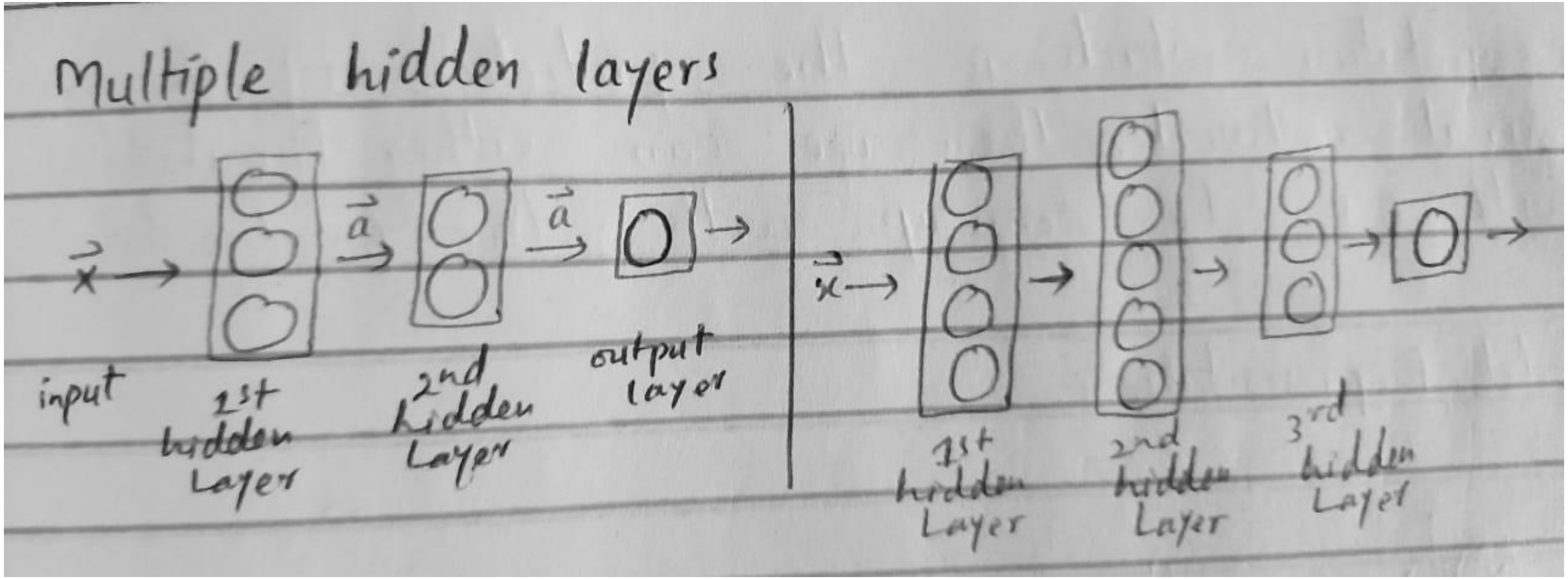


Neural Network

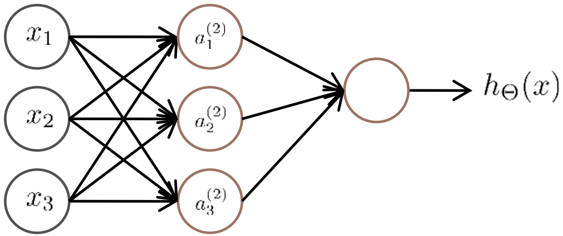


Multiple Hidden Layers

- Every layer in a neural network applies a transformation to its inputs.
- Deeper networks extract **hierarchical features**, enabling:
    - Early layers to learn **simple features** (edges, textures).
    - Deeper layers to capture **complex patterns** (objects, faces, words).
- The final output layer predicts the **desired outcome**.



Neural Network



$a_i^{(j)} =$ "activation" of unit i in layer j

$\Theta^{(j)} =$ matrix of weights controlling function mapping from layer j to layer $j+1$

Computational Steps:

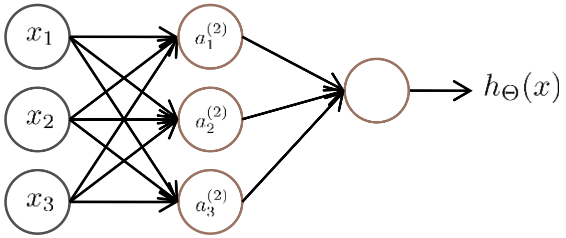
$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$
$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

If network has s_j units in layer j, s_{j+1} units in layer $j+1$, then $\Theta^{(j)}$ will be of dimension $s_{j+1} \times (s_j + 1)$.

Neural Networks: Representation

Model Representation II

Forward propagation: Vectorized implementation



$$a_1^{(2)} = g(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3)$$

$$h_\Theta(x) = g(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)} + \Theta_{13}^{(2)}a_3^{(2)})$$

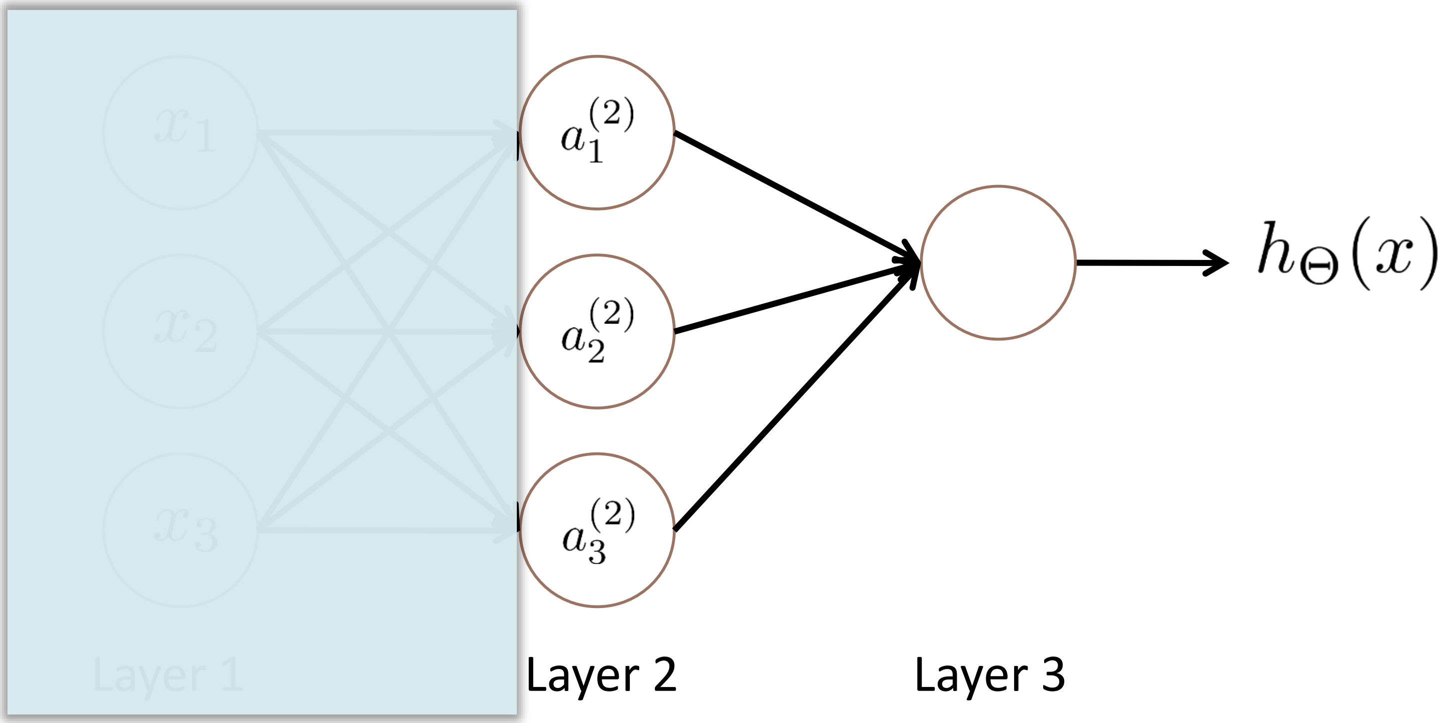
$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix}$$

$$z^{(2)} = \Theta^{(1)}x$$
$$a^{(2)} = g(z^{(2)})$$

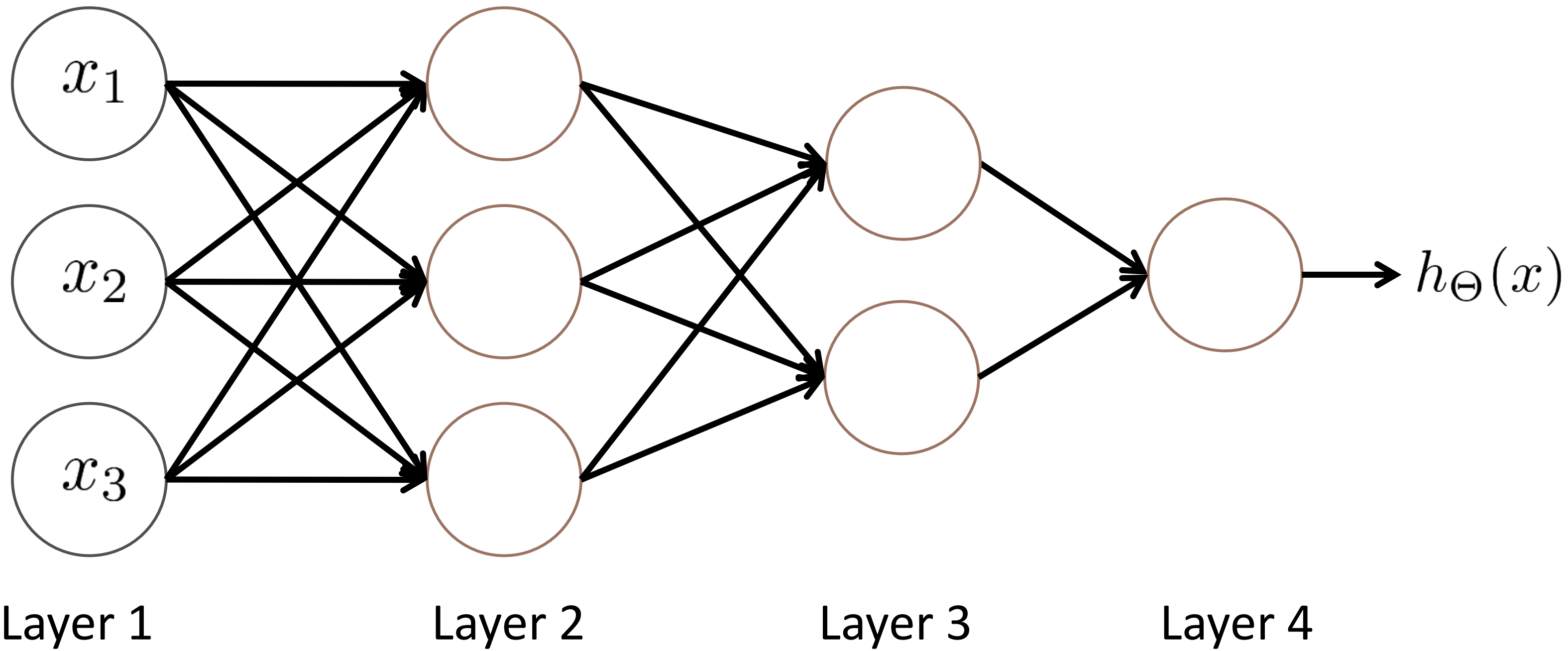
Add $a_0^{(2)} = 1$.

$$z^{(3)} = \Theta^{(2)}a^{(2)}$$
$$h_\Theta(x) = a^{(3)} = g(z^{(3)})$$

Neural Network learning its own features



Neural Network Architecture



How Neural Networks Learn

- Neural networks adjust weights and biases using labeled training data.

1. **Forward Propagation:** Compute predictions
2. **Loss Function:** Measure error
3. **Backpropagation:** Compute gradients
4. **Optimization:** Update weights using gradient descent

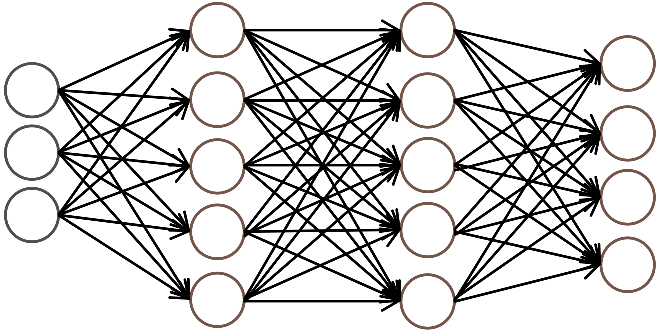
Multi-Class Classification with ANNs

Pedestrian

Car

Motorcycle

Truck



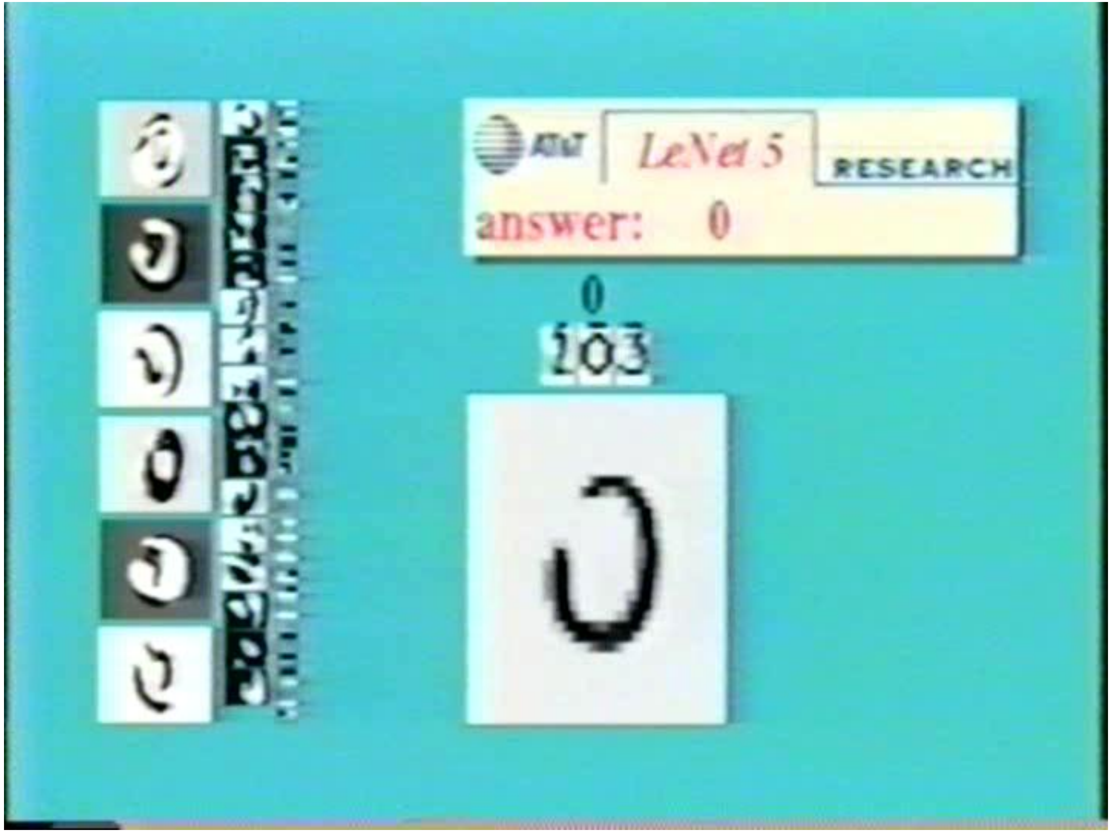
$$h_\Theta(x) \in \mathbb{R}^4$$

Our neural network output a vector of four numbers

Want $h_\Theta(x) \approx \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$, etc.

when pedestrian when car when motorcycle

Handwritten digit classification

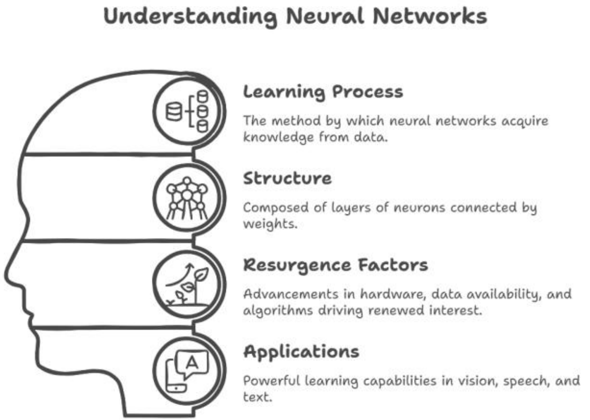


[Courtesy of Yann LeCun]

Summary

Neural Networks:

- Are inspired by the human brain
- Learn patterns from data through weights and updates.
- Work well on complex tasks like image and speech recognition
- Benefit from deeper architectures (DNNs)
- Require understanding of activation functions, layers, and the learning process.



Thank You