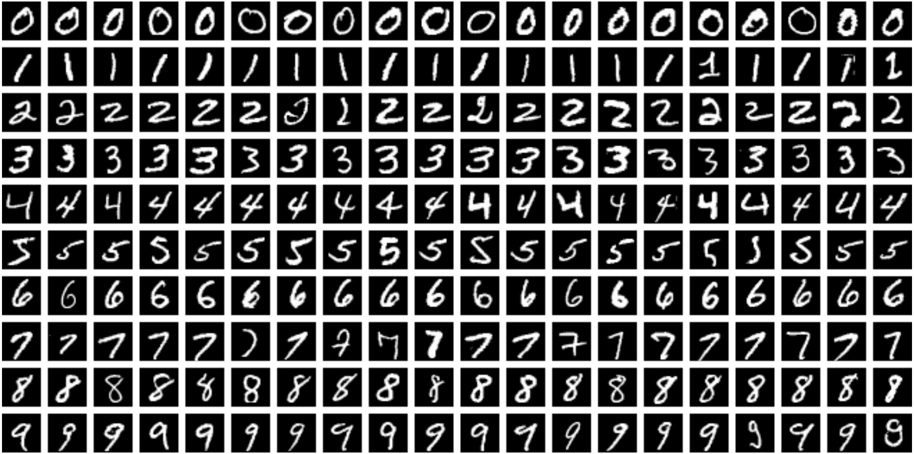
Introduction To Deep Learning

Bright Wiredu

November 7, 2025

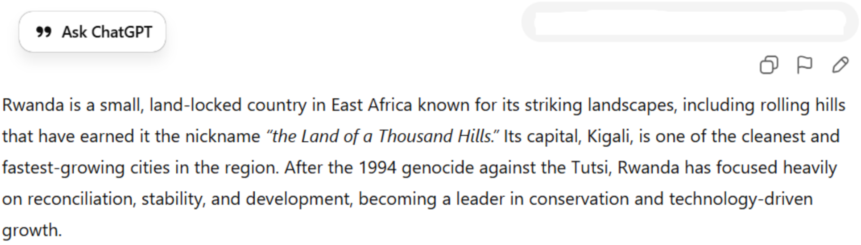
Outline

MNIST Handwriting Classification



Source : Wikipedia

Text Generation



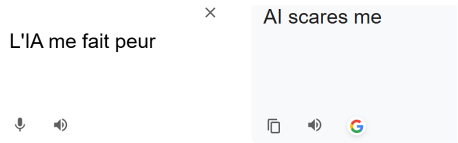


Image Generation

© 2025 Image credits: Hend. All rights reserved.

Self Driving Cars

History of Deep Learning

- **1952 – Stochastic Gradient Descent (SGD)**
- **1958 – Perceptron (Learnable Weights)** Introduced by Frank Rosenblatt.[1]
- **1986 – Backpropagation (Multilayer Perceptron)**
- **1995 – Deep Convolutional Neural Networks** Early deep CNNs (e.g., LeNet-5 by Yann LeCun)

(*Introduction to deep learning*. https://www.youtube.com/watch?v=ErnWZxJovaM&list=PLtBw6njQRU-rwp5__7C0oIVt26ZgjG9NI&index=12. Accessed: 2025-11-06. 2025)

[1] https://en.wikipedia.org/wiki/Frank_Rosenblatt

Objectives

- Understand the fundamentals of Deep Learning (DL)
- Explore key architectures such as Convolutional Neural Networks (CNNs)
- Understand common DL jargon and terminology
- Implement DL models using frameworks such as PyTorch and TensorFlow
- Know how to evaluate and optimize model performance through experimentation and tuning

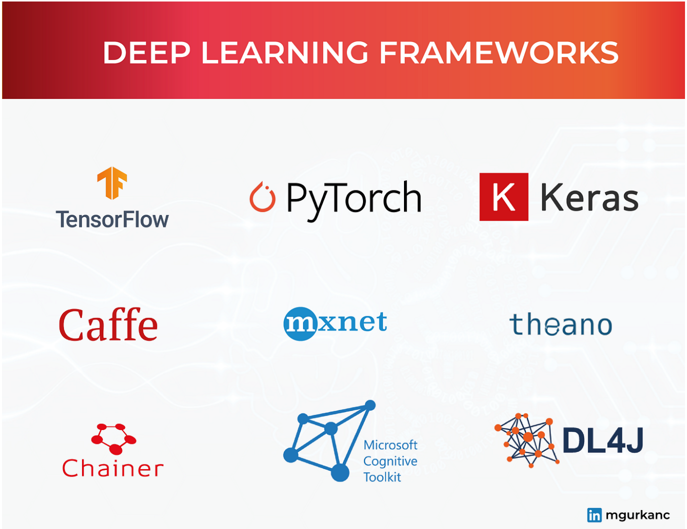
What is Deep Learning?

Deep learning is a subset of machine learning (ML) and artificial intelligence (AI) techniques that mimic human thinking processes (brain) by training machines with sufficiently large amounts of data. It is inspired by the human brain and is built upon artificial neural networks (ANNs) (deep neural networks). Compared to traditional ML, deep learning usually requires much larger amounts of data to learn effectively.

Machine Learning vs Deep Learning

- In traditional ML, you explicitly define the features for the model to learn from your dataset.
- In deep learning, the model automatically learns features from the data, eg. lines, curves, edges, and mid and higher-level latent representations.
- Deep learning requires much larger datasets for effective training compared to traditional ML.
- ML models may perform well on smaller datasets, while DL typically benefits from large-scale data and GPU acceleration.
- DL excels at complex tasks such as image recognition, speech processing, and natural language understanding.

Deep Learning Frameworks



Source: (Gürkan Çanakçı. *Choose the Deep Learning Frameworks: A Comprehensive Guide.* Accessed: 2025-11-07. 2023. URL:

ANN Architecture: Neuron Definition

- **Definition:** Let $w \in \mathbb{R}^d$, $W_0 \in \mathbb{R}$, $\sigma : \mathbb{R} \to \mathbb{R}$. A neuron is a function:

$$\Phi_{w,w_0,\sigma}(x) = \sigma\left(w_0 + \sum_{i=1}^{d} w_i x_i\right)$$

  When is Neuron called perceptron?

- **Vector Notation:**

$$\Phi_{w,w_0,\sigma}(x) = \sigma(w^\top x + w_0)$$

  - Convention: w and x are column vectors.
  - w is called the weight vector, b is the bias.

- **Activation Function:** Introduces non-linearity. Examples:

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \quad \mathrm{ReLU}(z) = \max(0, z), \quad \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Neuron Structure

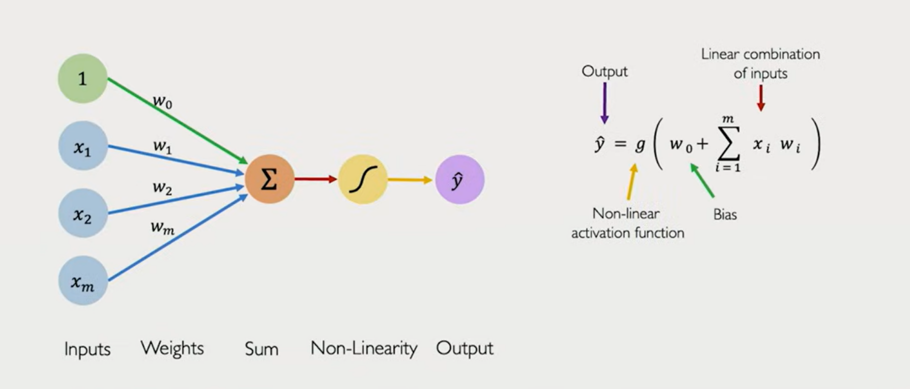


Figure: Single neuron[2]

[2] https://www.youtube.com/watch?v=ErnWZxJovaM&list=PLtBw6njQRU-rwp5__7C0oIVt26ZgjG9NI&index=12

Example: Implement a Neuron in PyTorch

Given:

$$W = \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \quad W_0 = 1.5, \quad \sigma(z) = \frac{1}{1 + e^{-z}}$$

Task: Compute the output of the neuron for an input $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ using PyTorch.

Try different values of x and observe the output.

ANN Architecture

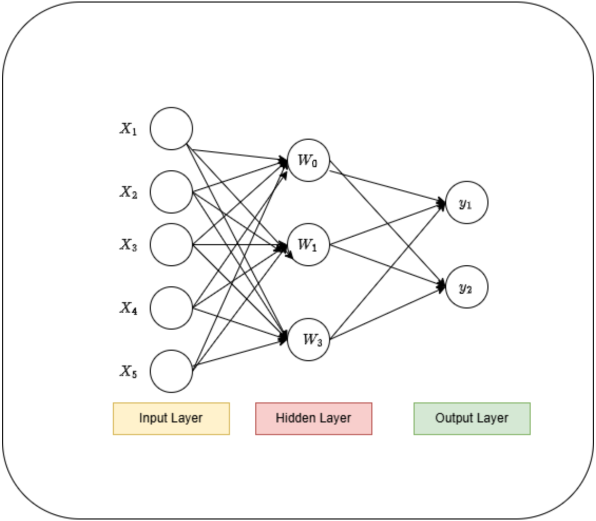
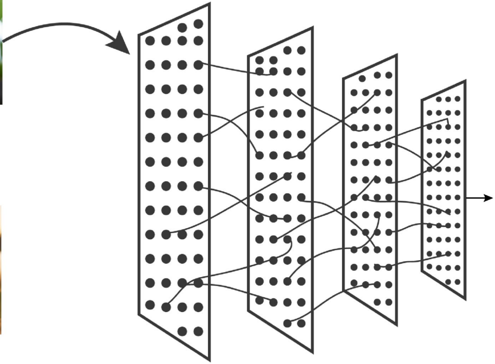


Figure: ANN Architecture

Real World Input

Cat

Dog



Output
Cat

Training and Learning

- **Forward Propagation:** Compute output:

$$a^{(l)} = f(W^{(l)} a^{(l-1)} + W_0^{(l)})$$

- **Loss Function:** Measures error, e.g., MSE:

$$L = \frac{1}{n} \sum_i (y_i - \hat{y}_i)^2$$

- **Backpropagation:** Compute gradients:

$$\frac{\partial L}{\partial W} \quad \text{and} \quad \frac{\partial L}{\partial W_0}$$

- **Optimization:** Update weights via GD:

$$W \leftarrow W - \eta \frac{\partial L}{\partial W}, \quad b \leftarrow b - \eta \frac{\partial L}{\partial W_0}$$

Gradient Descent

Algorithm

1. Initialize weights randomly $\sim\mathcal{N}(0, \sigma^2)$
2. Loop until convergence:
3. Compute gradient, $\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$
4. Update weights, $\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$
5. Return weights

Optimization Algorithms

- **Stochastic Gradient Descent (SGD)** – Basic gradient-based update:
$$\theta \leftarrow \theta - \eta \nabla_\theta L$$
- **Momentum** – Accelerates SGD by adding a velocity term.
- **AdaGrad** – Adaptive learning rate for each parameter based on past gradients.
- **RMSProp** – Uses exponentially decaying average of squared gradients to normalize updates.
- **Adam (Adaptive Moment Estimation)** – Combines Momentum + RMSProp; widely used.
- **Nadam** – Adam with Nesterov momentum.
- **Adadelta** – Variant of AdaGrad addressing rapid decay of learning rate.

Regularization Techniques

- **Dropout:** With certain probability, deactivate some neurons in a particular layer during training to prevent overfitting.
- **L1 / Lasso:** Adds $\lambda \sum |w_i|$ to loss $\rightarrow$ sparsity.
- **L2 / Ridge:** Adds $\lambda \sum w_i^2$ to loss $\rightarrow$ weight shrinkage.
- **Early Stopping:** Stop training when validation loss stops improving.

Limitations of Deep Learning

- **Large Data Requirements:** Deep learning models need huge datasets to learn effectively.
- **High Computational Power:** Training deep networks often requires GPUs.
- **Long Training Time:** Complex models can take hours or days to train.
- **Interpretability:** Deep models are often considered "black boxes".

Bibliography I

Çanakçı, Gürkan. *Choose the Deep Learning Frameworks: A Comprehensive Guide*. Accessed: 2025-11-07. 2023. URL: https://medium.com/@gurkanc/choose-the-deep-learning-frameworks-a-comprehensive-guide-66553788730d.

Introduction to deep learning. https://www.youtube.com/watch?v=ErnWZxJovaM&list=PLtBw6njQRU-rwp5__7C0oIVt26ZgjG9NI&index=12. Accessed: 2025-11-06. 2025.