בס"ד

NN&DL

DL Specialization

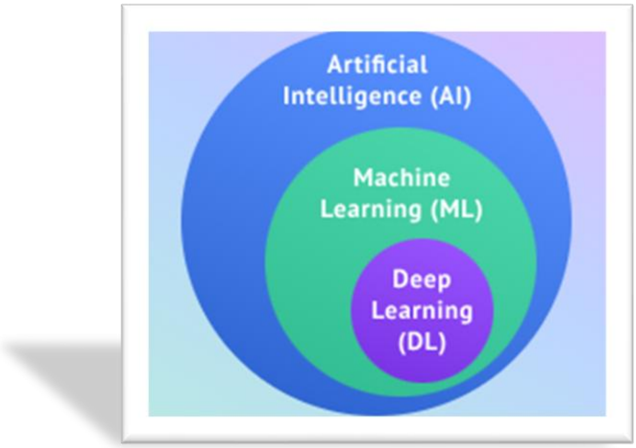
Table of Contents

[DATE]

ANDREW NG

Introduction to ML & DL

So what's the connection between AI, ML and DL ?



Connection between AI, ML, and DL:

- **AI (Artificial Intelligence)** is the broadest concept — making machines "intelligent," i.e., capable of performing tasks that normally require human intelligence.
- **ML (Machine Learning)** is a subset of AI — instead of explicitly programming rules, the system *learns patterns from data*.
- **DL (Deep Learning)** is a subset of ML — it uses **neural networks with many layers** to learn complex patterns, especially in unstructured data.

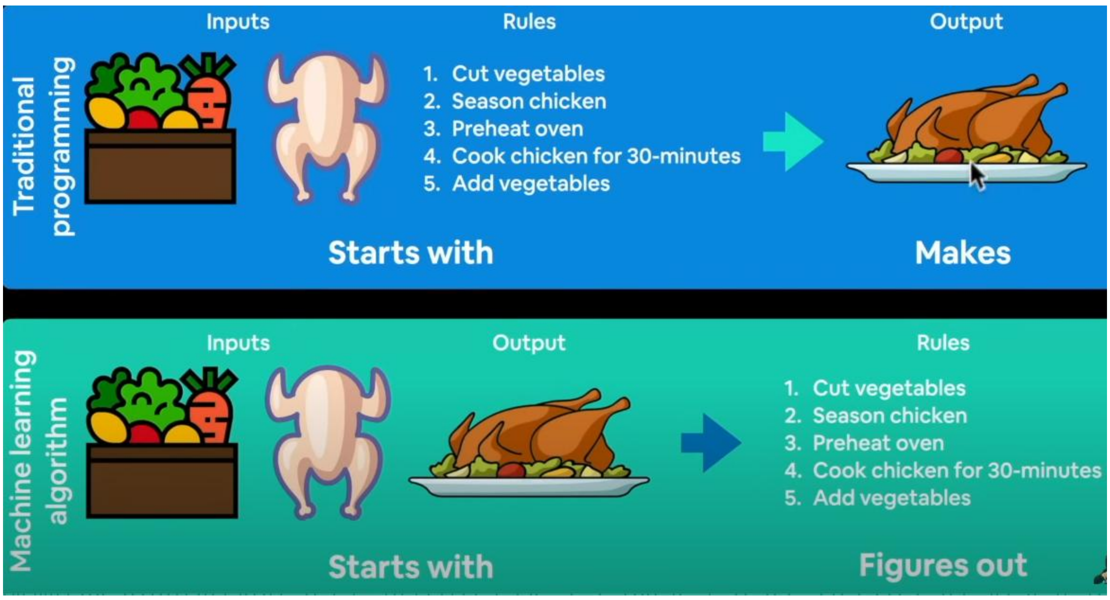
<u>And the definitions (informally):</u>

ML (Machine Learning) – is turning things (data) into numbers and finding patterns (code & math) in those numbers.
Works well with **structured data** (tables, spreadsheets). Can also work with unstructured data.

DL (Deep Learning) – A type of ML using neural networks to find patterns in **unstructured data** (images, audio, text, video).

Traditional programing vs ML programing

The difference between traditional programming and ML programming is presented very good at the next diagram:



Yarin Gaida

Traditional programming has	ML programming has
Inputs	Input
Rules	Output

Why should one use ML (or DL)?

ML/DL is useful for complex problems where there are too many rules to explicitly program. The key question is: Can you think of all the rules?

- For simple problems, like a chicken meal recommendation based on ingredients, it's feasible to define rules manually.
- For highly complex tasks, like self-driving cars, it's nearly impossible to list every scenario and corresponding action:
    - How to back off the street
    - How to park
    - How to turn left
    - How to react to unexpected obstacles
    - And countless other edge cases

In such cases, ML/DL learns patterns from data and generalizes to situations that weren't explicitly coded, making it far more scalable and practical than traditional programming.

"I think you can use ML for literally anything as long as you can convert in unto numbers and program it to find patterns. Literally it could be anything, any input or output from the universe"

Types of learning:

Supervised Learning

- have both data and labels.
- Example: 1,000 cat (label) photos (data) and 1,000 dog photos, and you know which photo is which.
- The algorithm learns to map input → output (photo → label).

Unsupervised Learning

- have data but no labels.
- Example: Only cat and dog photos, but you don't know which is which.
- The algorithm finds patterns or structure in the data (e.g., cluster - group of things that are similar to each other), but it doesn't know what each cluster "means" by default.

Transfer Learning

- Take a model that has already learned patterns from one dataset and reuse it for a new, related task.
- Example: A model trained to recognize cats and dogs can be adapted to recognize other animals with fewer new labeled examples.

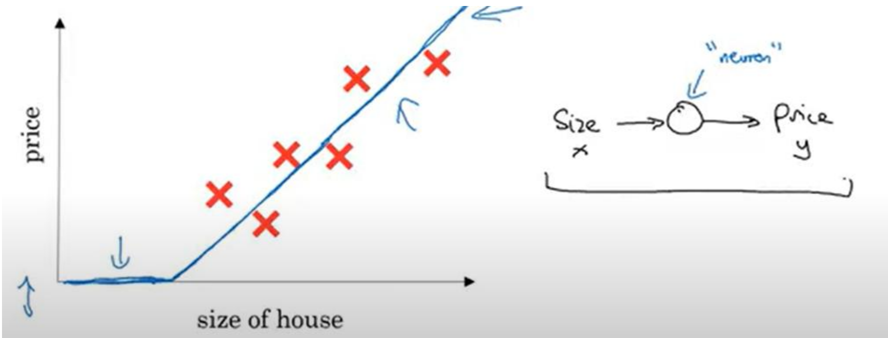
Reinforcement Learning

- Learn by trial and error through interaction with an environment.
- The agent takes actions and receives feedback in the form of rewards (good) or penalties (bad).
- Over time, the agent learns a strategy (policy) that maximizes long-term rewards.
- Example: A self-driving car gets a reward for staying in its lane and a penalty for crashing. Over many trials, it learns safe driving behavior.

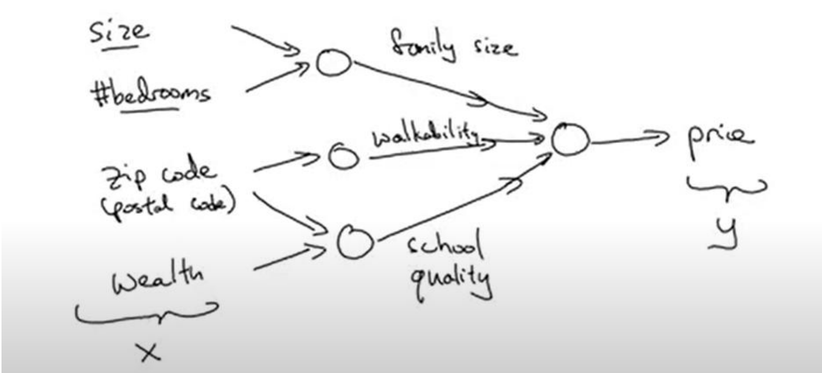
Neural Networks and DL

Definition of NN

A neural network is made of **neurons (nodes)** connected in layers.
Each neuron is basically a **function**: it takes inputs, applies some weights + an activation function, and produces an output.
By stacking many neurons together, the network can learn complex patterns from data.
(Example: In image recognition, some neurons learn to detect edges, others shapes, and deeper layers recognize objects)
An example from Andrew:



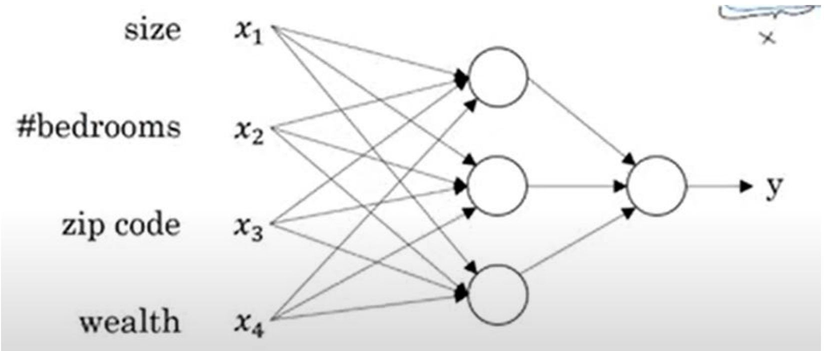
Now besides calculating the price of a house just by its size we consider other influencers:



Yarin Gaida

"Kind of magic of a neural network is that when you implement it you need to give it just the input x and output y – for a number of examples in the training set – and all this things in the middle - it will figure out by itself."

Now in more general way:



So that each node takes all 4 inputs – rather that saying what each node represents, we let the neural network decide whatever you want each node to be.
In this case we say that the unput layer and the layer in the middle of the neural network are **densely connected.**

Supervised Learning

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

Why does audio to text transcript supervise learning?

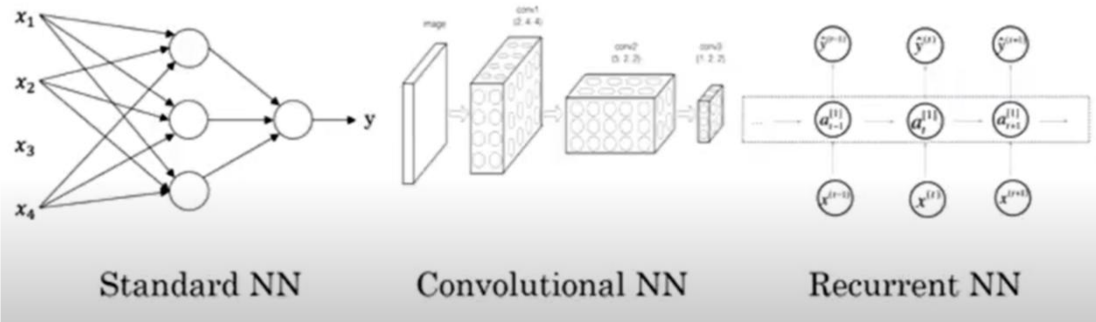
- Supervised learning because input = audio, label = text transcript
- Neural network learns mapping from sound patterns to words

Why does image to Position of other cars supervise learning?

- Supervised learning because input = camera/radar data, label = positions of cars
- Neural network learns to detect and locate cars based on annotated data
- Harder than audio-to-text because the input is high-dimensional (images + radar signals) and the output is continuous coordinates, not discrete labels

(**Unsupervised learning** is when we **do not have labels / outputs** for our data. We only have inputs, and the algorithm tries to find patterns, structure, or groupings by itself)

NN examples



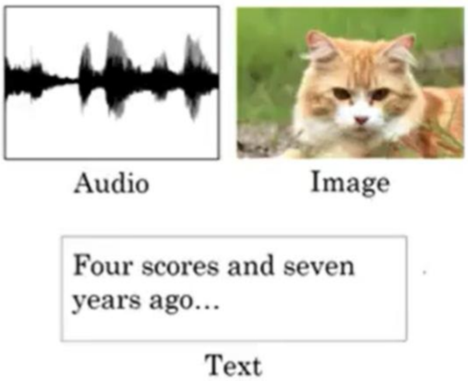
Structed data VS. Unattracted data

Structured Data

Size	#bedrooms	...	Price (1000$s)
2104	3		400
1600	3		330
2400	3		369
⋮	⋮		⋮
3000	4		540

User Age	Ad Id	...	Click
41	93242		1
80	93287		0
18	87312		1
⋮	⋮		⋮
27	71244		1

Unstructured Data



Neural Networks Basics

Binary classification