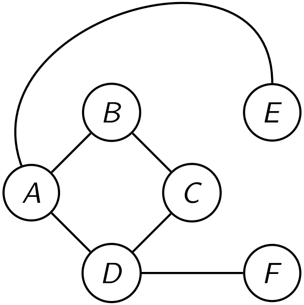
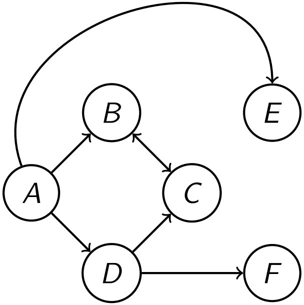
Lecture 1. Introduction to Graph Theory

What is a graph?

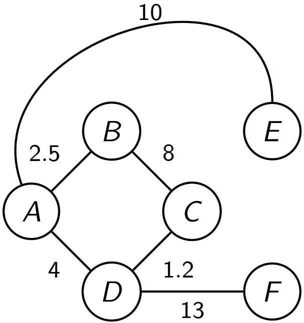
A graph is a structure consisting of vertices (or nodes: points in the graph) and edges (connections between pairs of vertices.)



(Undirected) Graph



Directed Graph

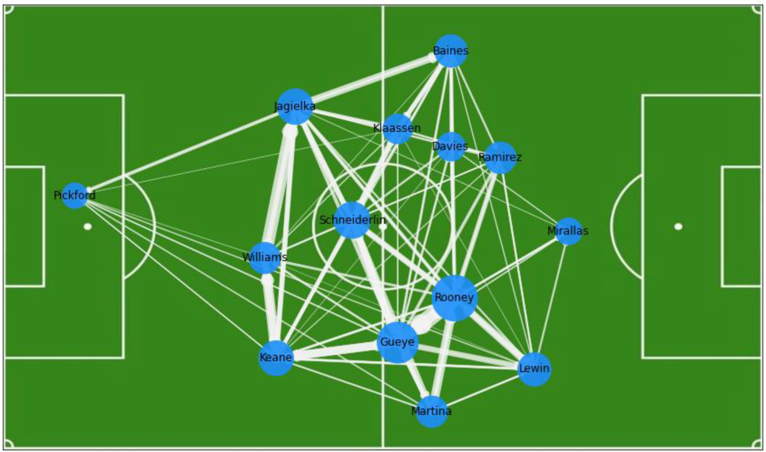


Weighted Graph

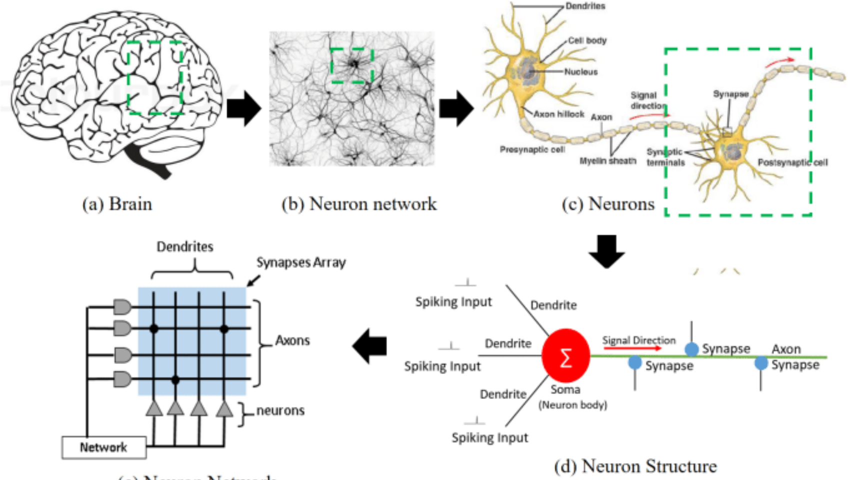
The vertices and edges of a graph might represent any number of different things, depending on the application

Applications of Graph Models

Example 1.1. Football Passing Networks



Example 1.2. A neural network is a computational model inspired by the structure and functioning of the human brain. It is widely used in machine learning tasks, such as image recognition, language processing, and data classification.



(a) Brain (b) Neuron network (c) Neurons

(d) Neuron Structure

(e) Neuron Network

A neural network can be modeled as a directed graph, where:

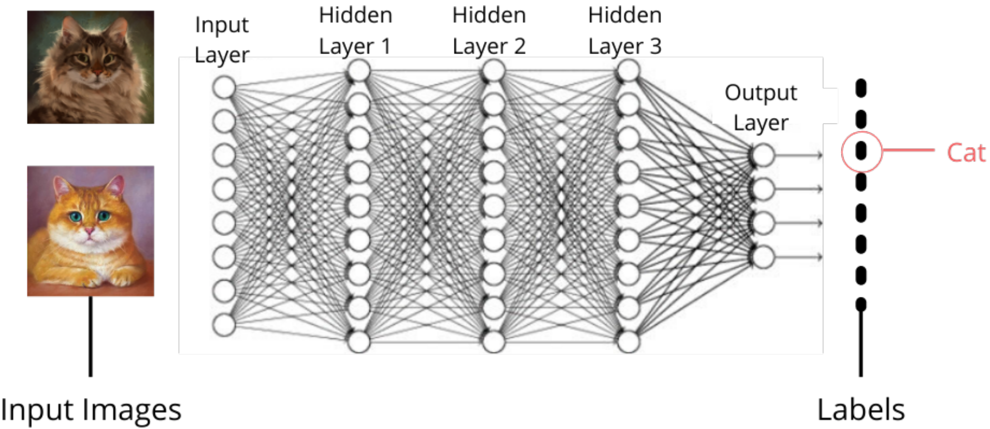
- **Vertices** represent neurons.
- **Edges** represent connections (synapses) between neurons, carrying weights to define the strength of the connections.

Structure of Neural Networks:

1. Input Layer: The first layer, where data enters the network.
2. Hidden Layers: Intermediate layers that process the data by performing weighted calculations and applying activation functions.
3. Output Layer: The final layer that produces predictions or classifications.

Information Flow in Neural Networks: Data flows from the input layer through the hidden layers and finally to the output layer. Each connection has a weight and can be adjusted during training to optimize performance.

Fully connected feed-forward neural network (perceptron):



Problem 1.3. Model a feed-forward neural network as a directed weighted graph with the following details:

- Input Layer: 2 neurons labeled I_1 and I_2.
- Hidden Layer: 3 neurons labeled H_1, H_2, H_3.
- Output Layer: 1 neuron labeled O_1.

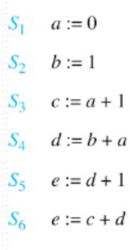
Graph Construction:

1. Draw vertices for each neuron.
2. Add directed edges to represent connections:
   - Each input neuron connects to every hidden neuron.
   - Each hidden neuron connects to the output neuron.
3. Assign weights to the edges (use arbitrary values).

Exercise 1.4. A precedence graph is a directed graph used in programming to represent the execution order of statements based on dependencies.

- Vertices: Represent program statements.
- Edges: A directed edge from vertex u to vertex v means that statement v cannot be executed until statement u has been executed.

Given the following statements and dependencies, draw a precedence graph. (Here, $S_1, S_2, \ldots$, are the order of computation process.)



S_1 $a := 0$

S_2 $b := 1$

S_3 $c := a + 1$

S_4 $d := b + a$

S_5 $e := d + 1$

S_6 $e := c + d$