Associate Memory

- One of the major classes of neural networks
- Is a Store house of associated patterns encoded in some form.
- When store house is triggered or incited with a pattern, the associate pattern is recalled or output.
- Input could be an exact replica of the stored pattern or distorted or partial representation of stored pattern

Introduction

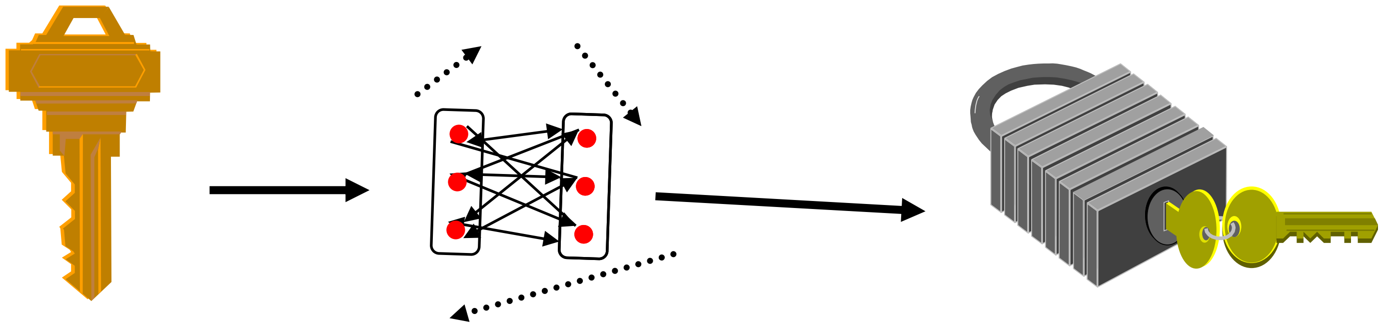
- Types of associative memory:
  - Heteroassociative memory
  - Autoassociative memory

Hetero Associative memory

✓If the associated pattern pairs (x,y) are different and the model recalls Y if given an X then this is called hetero association.

✓Useful for association of patterns

1. Hetero-associative X ------> Y

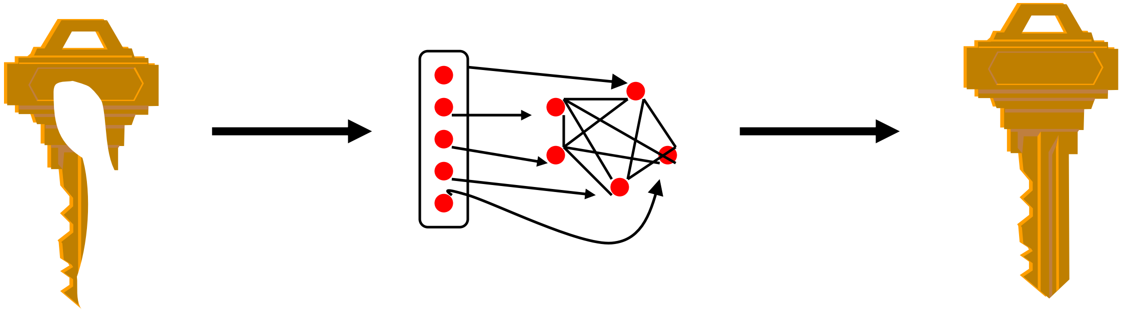


Auto associative memory

•If the associated pattern pairs (x,y) are same i.e X =Y then it is called auto association

•Useful for image refinement that is a given distorted or partial pattern, the whole pattern stored in its perfect form is recalled.

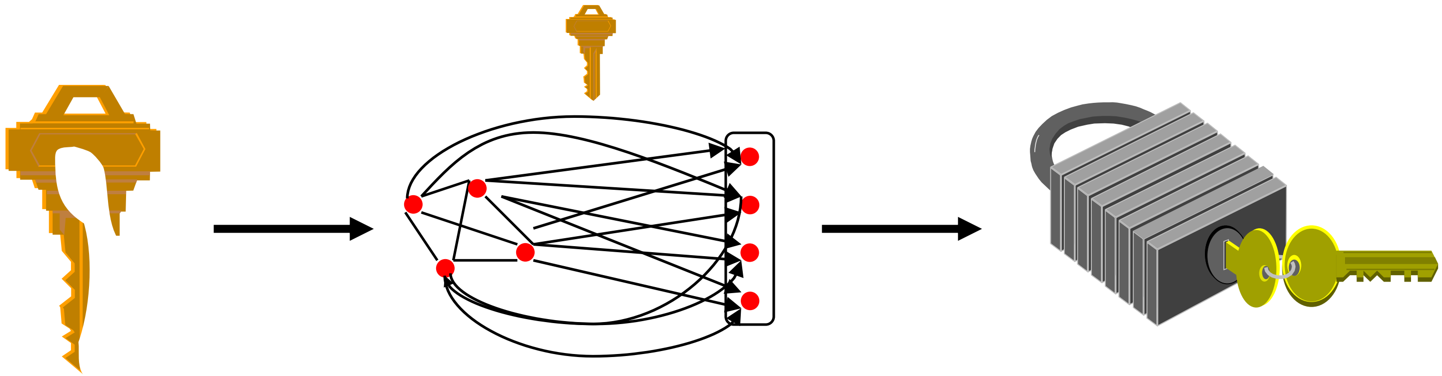
2. Auto-associative   X = Y



*Recognize noisy versions of a pattern

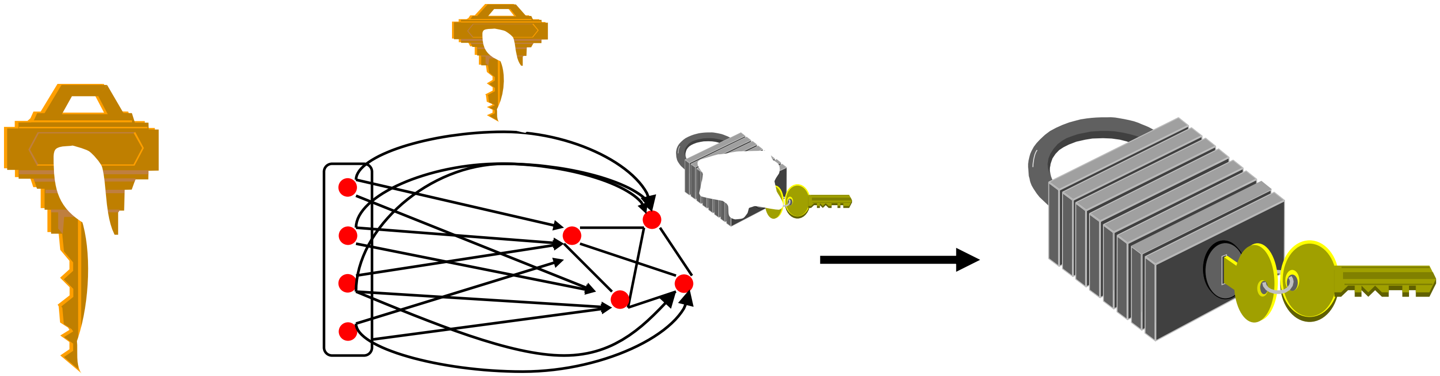
Hetero associative Network Types (2)

1. 1 Hetero-associative Input Correcting: X <> Y



*Input clique is auto-associative => repairs input patterns

1.2 Hetero-associative Output Correcting: X <> Y



*Output clique is auto-associative => repairs output patterns

Hebb’s Rule

Connection Weights ~ Correlations

``When one cell repeatedly assists in firing another, the axon of the first cell develops synaptic knobs (or enlarges them if they already exist) in contact with the soma of the second cell.” (Hebb, 1949)

↓

In an associative neural net, if we compare two pattern components (e.g. pixels) within many patterns and find that they are frequently in:

a) the same state, then the arc weight between their NN nodes should be positive

b) different states, then the arc weight between their NN nodes should be negative

<u>Matrix Memory:</u>

The weights must store the average correlations between all pattern components across all patterns. A net presented with a partial pattern can then use the correlations to recreate the entire pattern.

Quantifying Hebb's Rule

Compare two nodes to calc a weight change that reflects the state correlation:

Auto-Association: $$\Delta w_{jk} \propto i_k i_j$$

* When the two components are the same (different), increase (decrease) the weight

Hetero-Association: $$\Delta w_{jk} \propto i_k o_j$$

i = input component
o = output component

Ideally, the weights will record the average correlations across all patterns:

Auto: $$w_{jk} \propto \sum_{p=1}^{P} i_{pk} i_{pj}$$

Hetero: $$w_{jk} \propto \sum_{p=1}^{P} i_{pk} o_{pj}$$

Hebbian Principle: If all the input patterns are known prior to retrieval time, then init weights as:

Auto: $$w_{jk} \equiv \frac{1}{P} \sum_{p=1}^{P} i_{pk} i_{pj}$$

Hetero: $$w_{jk} \equiv \frac{1}{P} \sum_{p=1}^{P} i_{pk} o_{pj}$$

Weights = Average Correlations