Neural Networks
Lecture 5: Associative Memory

H.A. Talebi
Farzaneh Abdollahi

Department of Electrical Engineering

Amirkabir University of Technology

Winter 2011

Introduction

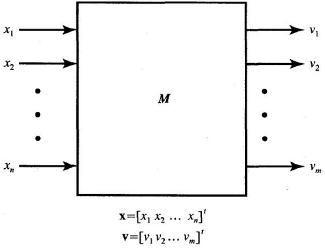
Static Memories

Introduction

- Learning can be considered as a process of forming associations between related patterns.
- For example visual image may be associated with another visual image, or the fragrance of fresh-mown grass may be associated with a visual image of feeling
- Memorization of a pattern could be associating the pattern with itself
- Therefore, in such networks the input pattern cause an output pattern which may be similar to the input pattern or related to that.
- An important characteristic of the association is that an input stimulus which is similar to the stimulus for the association will invoke the associated response pattern.

- For example, if we learn to read music, so that we associate with fingering on a stringed instrument, we do not need to see the same form of musical note we originally learned
  - If the note is larger, or handwritten , we still can recognize and play.
  - So after learning it is expected to make a good guess and provide appropriate response
- Another example, ability to recognize a person either in person or from a photo even his/her appearance has been changed
- This is relatively difficult to program by a traditional computer algs.
- Associative memories belong to class of NN that learn according to a certain recording algs.
- They require information a priori and their connectivity matrices (weights) most often need to be formed in advance
- Writing into memory produces changes in the neural interconnections
- Reading of the stored info from memory named recall, is a transformation of input signals by the network

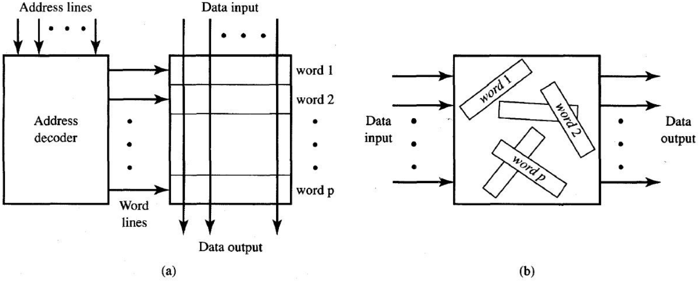
- Not usable addressing schemes exits in an associative memory
- All memory info is spatially distributed throughout the network
- The biological memory operates the same
- Associative memory enables a parallel search within a stored data
  - The purpose of search is to output one or all stored items that matches the search argument and retrieve it entirely or partially
- The fig. depicts a block diagram of an associative memory.



- The transformation is $v = M[x]$, M: a nonlinear matrix operator which has different meaning for each of memory models.

- For dynamic memories, M is time variable.
  - v is available at output at a later time than the input has been applied
- For a given memory model, M is usually expressed in terms of given prototype vectors that should be stored
- The algs of finding M are called recording or storage algs.
- The mapping in $v = M[x]$ preformed on x is called a retrieval.
- Retrieval may provide a desired an undesired solution prototype
- To have efficient retrieval some mechanisms should be developed
- Assume there are p stored pairs: $x^{(i)} \rightarrow v^{(i)}$ for $i = 1, ..., p$
- If $x^{(i)} \neq v^{(i)}$ for $i = 1, ..., p$ it is called heteroassociative memory
- If $x^{(i)} = v^{(i)}$ for $i = 1, ..., p$ it is called autoassociative memory
- Obviously the mapping of a vector $x^{(i)}$ into itself cannot be of any significance
- A more realistic application of autoassociative memory is recovery of undistorted prototype in response to a distorted prototype vector.

- Associative memory which uses NN concepts may resemble digital computer memory
- Let us compare their difference:
  - Digital memory is address-addressable memory:
    - data have input and output lines
    - a word line access the entire row of binary cells containing word data bits.
    - activation takes place when the binary address is decoded by an address decoder.



Addressing modes for memories: (a) address-addressable memory and (b) content-addressable memory.

- Associative memory is content addressable memory
  - The words are accessed based on the content of the key vector
  - When the network is excited by a portion of the stored date, the efficient response of autoassociative memory is the completed $x^{(i)}$ vector
  - In hetroassociative memory the content of $x^{(i)}$ provides the stored vector $v^{(i)}$
  - There is no storage for prototype $x^{(i)}$ or $v^{(i)}$ at any location of network
  - The entire mapping is distributed in the network.
  - The mapping is implemented through dense connections, feedback or/and a nonlinear thresholding operation
- Associative network memory can be
  - Static: networks recall an output response after an input has been applied in one feedforward pass, and, theoretically, without delay. They were termed instantaneous
  - Dynamic: memory networks produce recall as a result of output/input feedback interaction, which requires time.

- **Static memory**
  - Implement a feedforward operation of mapping without a feedback, or recursive update, operation.
  - They are sometimes called non-recurrent
  - The mapping can be expressed as

$$v^k = M_1[x^k]$$

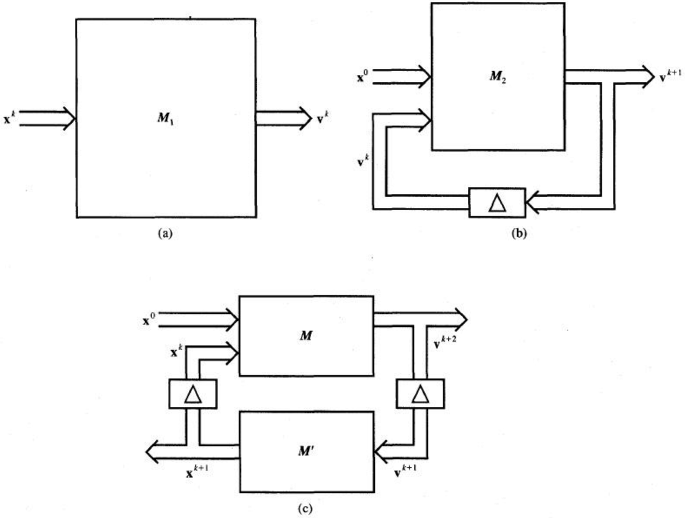
where k: index of recursion, M_1 operator symbol

- **Dynamic memory**
  - exhibit dynamic evolution in the sense that they converge to an equilibrium state according to the recursive formula

$$v^{k+1} = M_2[x^k, v^k]$$

  - This is a nonlinear difference equation.
  - Hopfield model is an example of a recurrent network for which the input x^0 is used to initialize v^0, i.e., $x^0 = v^0$, and the input is then removed.
  - So the formula will be simplified to

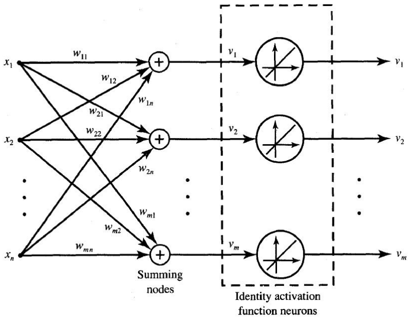
$$v^{k+1} = M_2[v^k]$$



Block diagram representation of associative memories: (a) feedforward network, (b) recurrent autoassociative network, and (c) recurrent heteroassociative network.

Linear Associator

- Traditional associative memories are of the ff and instantaneous.
- Their task: to learn the association within p vector pairs $\{x^{(i)}, v^{(i)}\}$, for $i = 1, 2, ..., p$.
- For the linear associative memory, an input pattern x is mapped to the output v by simply performing the matrix multiplication operation $v = Wx, \ x \in R^n, \ v \in R^m, \ W \in R^{m \times n}$
- $\therefore$ The general nonlinear mapping relationship $v = M_1[Wx]$ has been simplified to the linear form



- Objective: finding the matrix W to allow efficient storage of data within the memory
  - Given the pairs of vectors $\{s^{(i)}, f^{(i)}\}$, for $i = 1, 2, ..., p$.
  - We are looking for the mapping $f^{(i)} + \eta^i = Ws^{(i)}$ s.t. the length of the noise term vector, η^i is minimized.
  - Let us consider Hebbian learning rule

$$w'_{ij} = w_{ij} + f_i s_j$$

where f_i, s_j are ith and jth elements of vector f and s respectively and w_{ij} is the weight between them before update.

  - To generalize the formula: $W' = W + \mathbf{fs^T}$, where W : the weight matrix before update.
  - Initializing the weights in their unbiased position $W_0 = 0 \rightsquigarrow W' = \mathbf{fs^T}$
  - Since there are p pairs of patterns, the superposition of weights: $W' = \sum_{i=1}^{p} \mathbf{f}^{(i)}\mathbf{s^{(i)T}}$
  - The memory weight matrix W' above has the form of a cross-correlation matrix.

- Let us check whether or not the weight matrix W provides a noise-free mapping?
- Consider one of the stored vectors, $s^{(j)}$ as key vector at the input:

$$\begin{aligned} v &= \left(\sum_{i=1}^{p} \mathbf{f}^{(i)}\mathbf{s}^{(\mathbf{i})\mathbf{T}}\right)\mathbf{s}^{(\mathbf{j})} \\ &= \mathbf{f}^{(1)}\mathbf{s}^{(\mathbf{1})\mathbf{T}}\mathbf{s}^{(\mathbf{j})} + \ldots + \mathbf{f}^{(j)}\mathbf{s}^{(\mathbf{j})\mathbf{T}}\mathbf{s}^{(\mathbf{j})} + \ldots + \mathbf{f}^{(p)}\mathbf{s}^{(\mathbf{p})\mathbf{T}}\mathbf{s}^{(\mathbf{j})} \end{aligned}$$

- For ideal mapping, $\mathbf{s}^{(\mathbf{j})} \rightarrow \mathbf{f}^{(j)}$, we should have

$$v = \mathbf{f}^{(j)}$$

- $\therefore$ The orthogonal set of p input vectors $\{\mathbf{s}^{(i)}, \mathbf{s}^{(2)}, \ldots, \mathbf{s}^{(p)}\}$ ensures perfect mapping

$$\begin{cases} \mathbf{s}^{(\mathbf{i})\mathbf{T}}\mathbf{s}^{(\mathbf{j})} = 0 & i \neq j \\ \mathbf{s}^{(\mathbf{i})\mathbf{T}}\mathbf{s}^{(\mathbf{j})} = 1 & i = j \end{cases}$$

- This is a strict condition and may not always hold for all sets of vectors.