Neural networks: Associative memory

Prof. Sven Lončarić

sven.loncaric@fer.hr

http://www.fer.hr/ipg

Overview of topics

- Introduction
- Associative memories
- Correlation matrix as an associative memory
- Error correction learning
- Pseudoinverse matrix as an associative memory
- Discussion
- Problems

Introduction

- In neurobiological context, a memory represents relatively permanent neural changed caused by an interaction of an organism and environment
- For memory to be useful, it must be accessible by the neural system
- A memory is “filled” through a learning process
- Memories can be divided into:
  - Short term memory (contains the current state of the environment)
  - Long term memory (contains permanently stored knowledge)

Introduction

- This section deals with a distributed memory similar to the brain that uses associations
- Associative memory is an important part of human memory
- The main property of an associative memory is mapping of input patterns into output patterns of neural activity

Introduction

- During learning an input pattern called a key is presented to the memory that transforms it into a memorized pattern
- During recall a noised or incomplete version of the original key is presented to the memory
- Regardless of the imperfect input key, the associative memory outputs the corresponding memorized output pattern

Properties of associative memories

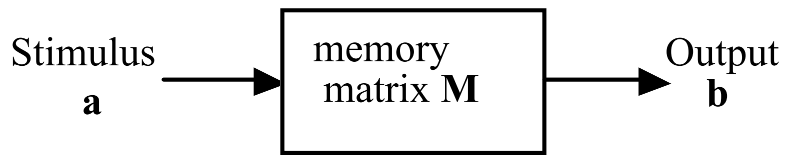
- An associative memory is distributed
- Input pattern (key) and output (memorized pattern) are vectors
- Information is stored in memory using a large number of neurons
- Information contained in the key determines the “address” of the pattern in the memory
- Memory is noise tolerant
- Possible interactions of memorized patterns – posibility of errors

Types of associative memories

- Autoassociative memory:
  - Input vector (key) is associated to itsself in the memory
  - Dimension of input and output vectors is the same
- Heteroassociative memory:
  - Arbitrary input vectors (keys) are associated with arbitrary memorized vectors
  - Dimension of input and output vectors can be different
- In both cases a memorized pattern can be recalled using an input vector that is incomplete or noised version of the original key

Types of associative memories

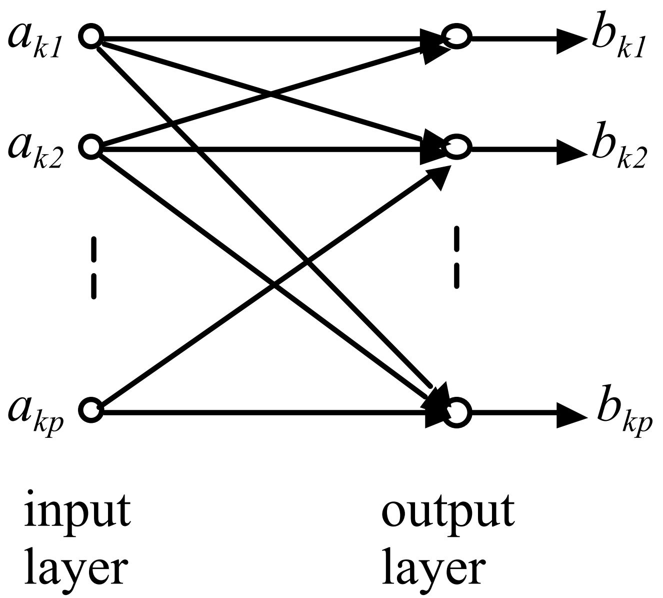
- Linear associative memory:
  - Neurons work in linear mode (linear combination)
  - Let **a** and **b** be input and output from associative memory – then the input-output mapping is represented by **b=Ma**, where **M** is the memory matrix



- Nonlinear associative memory
  - Input-output relation is described by: **b**=f(**M**,**a**)**a**, where f(.,.) is a nonlinear function

A model of associative memory

- A model of linear associative memory with artificial neurons is shown below:



Memory mapping

- Let us assume that we have a memory with one input layer and one layer with p linear neurons
- Let us assume that for input $\mathbf{a}_k$ we obtain output $\mathbf{b}_k$
- Let us assume that the memory stores q input-output pairs
- We could express the input-output relations as:

$$\mathbf{b}_k = \mathbf{W}(\mathrm{k})\mathbf{a}_k, \quad k = 1, \ldots, q$$

where $\mathbf{W}(\mathrm{k})$ is a matrix of dimensions $p \times p$ that depends only on $\mathbf{a}_k$ and $\mathbf{b}_k$

Memory mapping

- For q input-output pairs we obtain matrices $\mathbf{W}(1)$, …, $\mathbf{W}(q)$
- Furthermore, we could form a matrix of dimensions $p \times p$ that represents the sum of matrices $\mathbf{W}(k)$:

$$\mathbf{M} = \sum_{k=1}^{q} \mathbf{W}(k)$$

- Matrix **M** defines a relation between the input and the output patterns

Memory mapping

- A memory matrix M can also be represented using a recursive expression:

  $$\mathbf{M}_k = \mathbf{M}_{k-1} + \mathbf{W}(k), \quad k = 1, 2, \ldots, q$$

  where $\mathbf{M}_0 = \mathbf{0}$
- $\mathbf{M}_{k-1}$ is an old matrix value for the first k-1 associations
- $\mathbf{M}_k$ is an updated matrix that also takes into account the k-th association
- As the number of stored associations q increases the influence of individual new pairs to the memory decreases

Correlation matrix

- Let us assume that a memory has matrix **M** that represents the stored associations $\mathbf{a}_k$, $\mathbf{b}_k$, where $k = 1, 2, \ldots, q$
- An estimate of the matrix **M** (called correlation matrix) can be calculated using:

$$\hat{\mathbf{M}} = \sum_{k=1}^{q} \mathbf{b}_k \mathbf{a}_k^T$$

- The term $\mathbf{b}_k\mathbf{a}_k^T$ is the outer product of key $\mathbf{a}_k$ and stored pattern $\mathbf{b}_k$ and is a matrix of dimension $p \times p$

Correlation matrix

- This estimate can be written as:

$$\hat{\mathbf{M}} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_q \end{bmatrix} \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_q^T \end{bmatrix} = \mathbf{B}\mathbf{A}^T$$

where $\mathbf{A}=[\mathbf{a}_1\ \mathbf{a}_2\ \dots\ \mathbf{a}_q]$ and $\mathbf{B}=[\mathbf{b}_1\ \mathbf{b}_2\ \dots\ \mathbf{b}_q]$

- **A** is the matrix of input patterns (keys) of dimension $p \times q$
- **B** is the matrix of stored patterns of dimension $p \times q$

Memory recall

- Let us assume that a pattern $\mathbf{a}_j$ is input into associative memory that has stored q patterns
- The output of the memory will be:

$$\mathbf{b} = \hat{\mathbf{M}}\mathbf{a}_j = \sum_{k=1}^{q} \mathbf{b}_k \mathbf{a}_k^T \mathbf{a}_j = \sum_{k=1}^{q} \left(\mathbf{a}_k^T \mathbf{a}_j\right)\mathbf{b}_k$$

- Furthermore, we can rewrite this as:

$$\mathbf{b} = \left(\mathbf{a}_j^T \mathbf{a}_j\right)\mathbf{b}_j + \sum_{\substack{k=1 \\ k \neq j}}^{q} \left(\mathbf{a}_k^T \mathbf{a}_j\right)\mathbf{b}_k$$

Memory recall

- Let us assume that the input vectors (keys) are normalized: $\mathbf{a}_k^T\mathbf{a}_k=1$
- Then it holds that:

$$\mathbf{b} = \mathbf{b}_j + \mathbf{v}_j$$

where:

$$\mathbf{v}_j = \sum_{\substack{k=1 \\ k \neq j}}^{q} \left(\mathbf{a}_k^T \mathbf{a}_j\right) \mathbf{b}_k$$

Interpretation of memory recall

$$\mathbf{b} = \mathbf{b}_j + \mathbf{v}_j$$

- The first term in the above expression represents the desired memory recall for key $\mathbf{a}_j$
- The second term represents a "crosstalk" between key $\mathbf{a}_j$ and other stored keys (i.e. noise)
- If input patterns are statistically independent than the second term represents Gaussian noise
- This noise limits the number of reliably stored patterns

Interpretation of memory recall

$$\mathbf{v}_j = \sum_{\substack{k=1 \\ k \neq j}}^{q} \left(\mathbf{a}_k^T \mathbf{a}_j\right)\mathbf{b}_k$$

- Let us assume that the input vectors $\mathbf{a}_j$ (keys) compose an orthonormal set of vectors, i.e.

$$\mathbf{a}_k^T \mathbf{a}_j = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases}$$

- In that case the noise $\mathbf{v}_j$ jednak nuli
- The number of linearly independent vectors of dimension p is equal to p
- This means that the memory capacity is equal to the vector dimension (in this case p)

Discussion

- If a set of input patterns is not orthogonal it is possible to use Gram-Schmidt procedure to orthonormalize the ste of linearly independent vectors
- A disadvantage of this simple approach is that the memory has no way of error correction
- To overcome this drawback it is possible to use an error correction algorithm

Error correction learning

- Let $\mathbf{M}(n)$ be the matrix learned in step n
- Input vector $\mathbf{a}_k$ is presented to the memory and gives output vector $\mathbf{M}(n)\ \mathbf{a}_k$
- An error vector can be defined as:

$$\mathbf{e}_k(n) = \mathbf{b}_k - \mathbf{M}(n)\ \mathbf{a}_k$$

  where $\mathbf{b}_k$ is the output associated with input $\mathbf{a}_k$
- The error vector can be used for learning as:

(correction) = (learning-rate) × (error) × (input)

Error correction learning

- In this case we can write:

$$\Delta \mathbf{M}(n) = \eta \, \mathbf{e}_k(n) \, \mathbf{a}_k^T = \eta \, [\mathbf{b}_k - \mathbf{M}(n) \, \mathbf{a}_k] \, \mathbf{a}_k^T$$

- Correction $\Delta \mathbf{M}(n)$ is used to update matrix $\mathbf{M}$:

$$\mathbf{M}(n+1) = \mathbf{M}(n) + \Delta \mathbf{M}(n) \, , \quad \mathbf{M}(0)=\mathbf{0}$$

- A constant positive parameter η generates a short term memory because recent patterns will be better memorized compared to older patterns
- For this reason the parameter η is sometimes decreased with time n to approach zero when memory is full

Pseudoinverse memory

- Another type of linear associative memory is a memory that minimizes the error of associative memory:

$$e = \left\| \mathbf{B} - \hat{\mathbf{M}}\mathbf{A} \right\|$$

  where **A** is a key matrix of dimension *p×q,* and **B** is a matrix of desired outputs of dimension *p×q*
- Euclidean norm gives the error of the associative memory

Pseudoinverse memory

- Linear algebra tells us that the error *e* is minimal for:

$$\hat{\mathbf{M}} = \mathbf{B}\mathbf{A}^{+}$$

where $\mathbf{A}^{+}$ is a pseudoinverse matrix of matrix **A**

- The above equation is called the pseudoinverse learning rule and the memory is called pseudoinverse memory
- The sufficient condition for the perfect association is:

$$\mathbf{A}^{+}\mathbf{A} = \mathbf{I}$$

- Then it holds that:

$$\hat{\mathbf{M}}\mathbf{A} = \mathbf{B}\mathbf{A}^{+}\mathbf{A} = \mathbf{B}\mathbf{I} = \mathbf{B}$$

Discussion

- In some applications better resistance to noise is shown by correlation memory and in some other cases pseudoinverse memory shows better results

Problems

- Problem 3.1.
  - Modify expressions for correlation memory assuming different dimensions of input and output vector.
- Problem 3.2.
  - Let input vectors be defined as:

    $\mathbf{a}_1=[1\ 0\ 0\ 0]^T$, $\mathbf{a}_2=[0\ 1\ 0\ 0]^T$, $\mathbf{a}_3=[0\ 0\ 1\ 0]^T$

    and output vector as:

    $\mathbf{b}_1=[5\ 1\ 0]^T$, $\mathbf{b}_2=[-2\ 1\ 6]^T$, $\mathbf{b}_3=[-2\ 4\ 3]^T$

    Determine a memory matrix **M** and show that the memory recall is correct.