Introduction to Neural Networks

Johns Hopkins University
Engineering for Professionals Program
605-447/625-438
Dr. Mark Fleischer

Module 8.1: Hebbian Learning

This Sub-Module Covers …

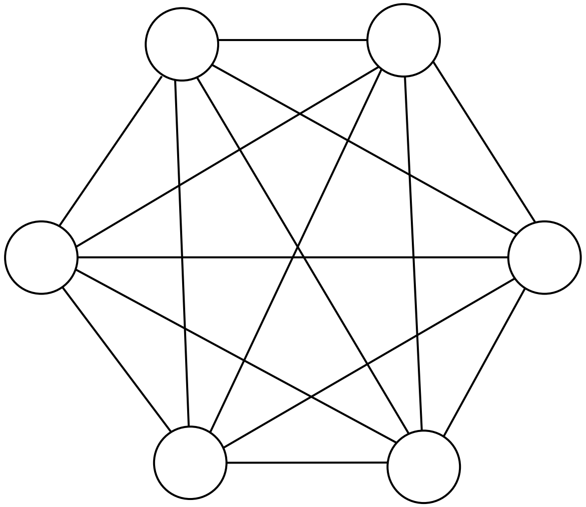
- The concept of Hebbian Learning.
- How to implement Hebbian learning paradigms with recurrent neural networks.
- How to model recurrent networks using matrix methods.

Dynamical Systems

- Neural Networks can be fashioned into dynamical systems.
- Feeding outputs back as inputs and cycling them ala fixed point exercise.
- How does such a system behave?
- Let's examine the most basic formulation.

Associative Neural Networks

- Examine the Hebbian Learning paradigm.
- Examine how and why the Hopfield Network works.



For n nodes, there are $\dfrac{n(n-1)}{2}$ connections which can be associated as the weights between each node.

With the addition of edges from each node to itself, and the symmetry of the nodes, we get n^2 weights.

Hebbian Learning

- In this type of net, and the FFBP algorithm, once the weights are established they are treated as fixed for the given input/output pairs that are used to train the net.
- Experiments with physiological neurons however reveal that the post-synaptic potential varies with time (as I mentioned during our first class).
- Moreover, the "weights" are also affected temporally by other efficacies for the same node. Some excite, others inhibit and if inhibitory potentials (negative weights) have "fired" in close temporal proximity to another excitatory potential, both may then have modified efficacies closer to their average (zero for example).
- Long-term potentiation of synapses, long-term depression of synapses occur if the potentials remain increased or decreased for extended periods of time.
- Thus, repeated temporal firings over time can lead to a more permanent modification of weights while short-term modifications do not last as long.

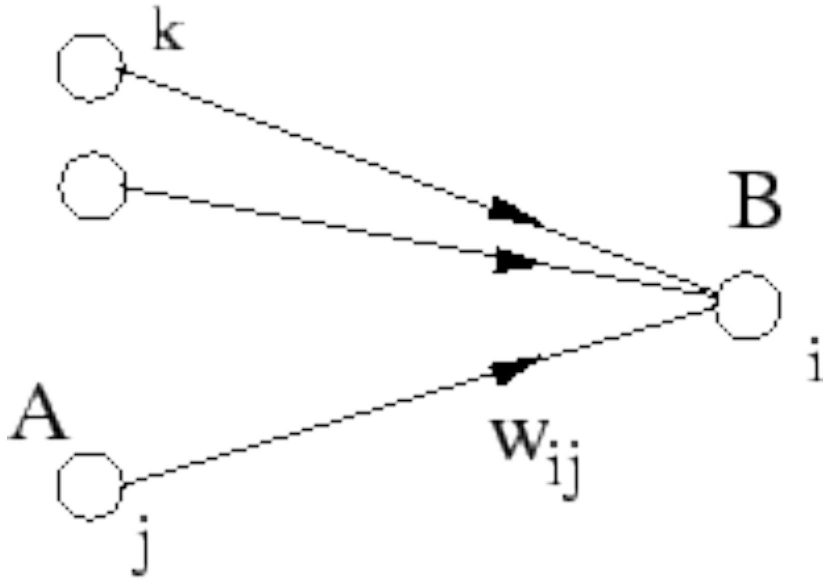
Hebbian Learning

- "Synaptic changes that are driven by correlated activity of pre- and post-synaptic neurons. This class of learning rule can be motivated by Hebb's principle and is therefore often called `Hebbian learning'."
- This is often described in terms of *synaptic plasticity*.

Hebbian Learning

- When an axon of cell *A* is near enough to excite cell *B* or repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that *A*'s synaptic efficacy, as one of the cells firing *B*, is increased.

Hebbian Learning



The change at synapse w_{ij} depends on the state of the presynaptic neuron *j* and the postsynaptic neuron *i* and the present efficacy w_{ij}.

Hebbian Learning

- Correlation-based learning is now generally called *Hebbian learning*. Correlations with firings on either side of the synapse are the basis of this type of learning which helps to stabilize the behavior of the neuron. Leads to the “cascade of associations” I mentioned in the first class.
- Long-lasting changes of synaptic efficacies were found experimentally by many researchers.
- How can we model this type of behavior?
- There are two aspects in Hebb's postulate that are particularly important, viz. ***locality*** and ***cooperativity***.
    - Locality means that the change of the synaptic efficacy can only depend on local variables, i.e., on information that is available at the site of the synapse, such as pre- and postsynaptic firing rate, and the actual value of the synaptic efficacy, but not on the activity of other neurons. Based on the locality of Hebbian plasticity we can make a rather general ansatz for the change of the synaptic efficacy,

Hebbian Idea

$$\frac{dw_{ij}}{dt} = f(w_{ij}, v_i, v_j)$$

The main idea of Hebbian Learning is that the behavior of other nodes can affect the synaptic weights of a given node in a manner that is either **reinforcing** or **inhibiting**.

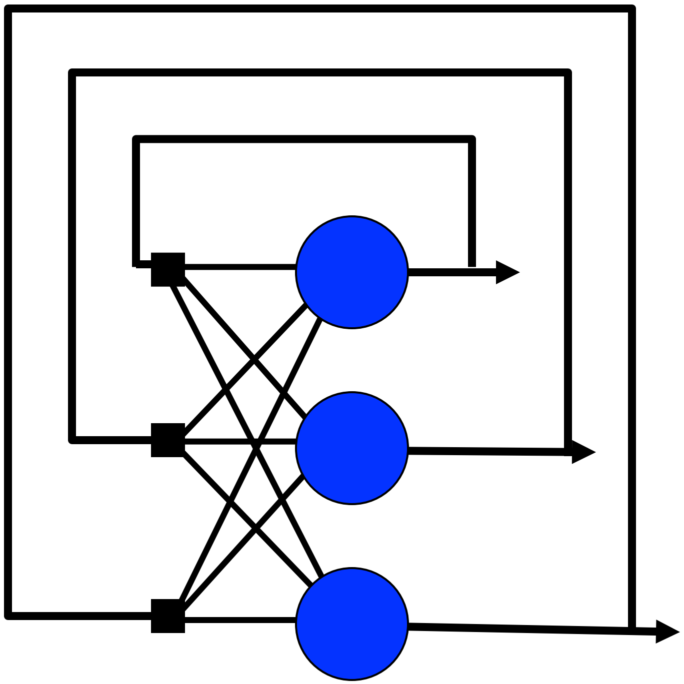
Network Structure

Let's start with simple ideas:

- One node for every input element of an exemplar.
- Two arcs between any two nodes (symmetry).
- No arc from a node to itself.
- Should map an exemplar to itself.

The Recurrent Network Topology

Let's view the network a bit differently.



Exemplar:

(1, -1, -1)

What is the weight from Node 1 to Node 2?

Network to Matrix

Each output x_i for a node *i* in a network of *N* perceptrons can be written as:

$$x_i = \sum_{j=1}^{N} w_{ij} x_j$$

But this is just the definition of the *i* th element of a vector after a matrix/vector multiplication!

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k$$

where index *k* corresponds to the iterate number.

Network to Matrix

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k$$

But how should we define **W**?

Sometimes in order to answer a question relating to a problem, we must change the problem!

$$\mathbf{p} = \mathbf{W}\mathbf{p}$$

Instead of deciphering a general dynamical system, let's examine a special case --- a fixed point!