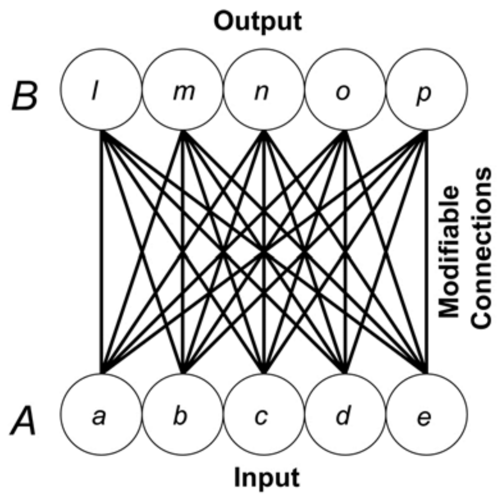
Quick Review PSYCO 452 Week 1 (January 2016)

Key Architecture: Distributed Associative Memory



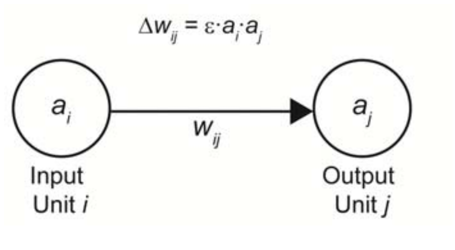
This network is a cued recall system. During learning, you present a pair of items to the network (one is a pattern of activity represented in the input units, the other is a pattern of activity represented in the output units). You then use a learning rule to store the association between the two patterns in the connection weights of the network. Later, you present a cue to the input units. This causes a signal to be sent through the weighted connections producing activity in the output units. Hopefully the activity produced there is the pattern associated with the cue that was learned earlier!

What I Forgot To Say In The Lecture:

The memory system above can use its one set of modifiable connections to store multiple stimulus-response pairs. This is another example of a distributed representation – different pattern pairs are distributed across the one set of weights.

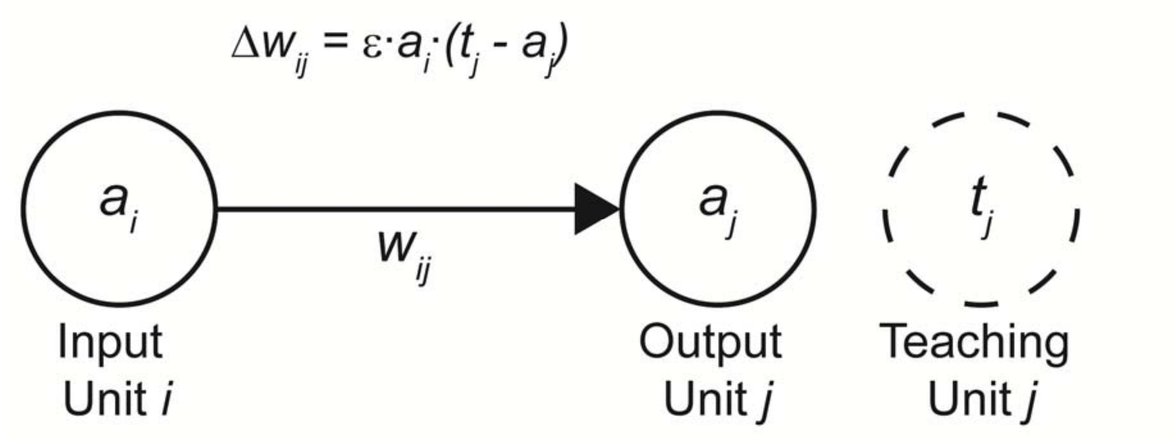
Hebb Learning

The key learning rule that we learned is the Hebb rule. It is used to store associations based on the law of contiguity. Basically, "processors that fire together wire together". This is implemented with a simple multiplication rule: a desired change in weight is a learning rate times the processor activity at the input end of the connection times the processor activity at the output end of the connection. The new weight equals the old weight plust the calculated weight change. This is illustrated in the figure below



The Delta Rule

You will discover in your homework assignments that the Hebb rule is very limited. One approach to fixing these problems involves measuring error in performance, and use this error to update the weights. So, you present the input pattern to the network. Then, you activate the output units. Next, you calculate output unit error. This is simply done by subtracting actual output unit activity (a_j) from the desired or target activity for that output unit (t_j). Then you use the Hebb rule, except that you multiply the learning rate times activity at the input end of the connection times the error at the output end of the connection in order to figure out the weight change. The delta rule is illustrated below.


$\Delta w_{ij} = \varepsilon \cdot a_i \cdot (t_j - a_j)$
a_i
w_{ij}
a_j
t_j
Input Unit *i*
Output Unit *j*
Teaching Unit *j*