ECDBS 2006 – Neural network modelling

- Aims of the course:
  - To discuss the modelling enterprise: Why do it? Why use neural networks? What are the alternatives?
  - To discuss the basic elements of network models, and their relationship to brain processes.
  - To show how “real” models are put together, and discuss their relationship to experimental data.
  - To provide some hands on experience of constructing simple networks.

Structure of the course

- Lectures (Monday, Tuesday, Thursday).
- Demonstrations/tutorials (Monday, Tuesday). SNNS, TLearn, Matlab.
- Practical (group) work (Wednesday)
- Student presentations (Thursday/Friday):
  - Discussion of a network project carried out here.
  - Discussion of a proposed modelling investigation: empirical background, current theories, aim of modelling, proposed model architecture.

What are neural network models good for?

- Computational/mathematical specificity : explicit "effective" procedures for performing a particular task.
- Testing ideas 1: do your ideas really work, have you thought the problem through in sufficient detail?
- Testing ideas 2: do your ideas really produce the empirical effects you think they do? Can you generate new predictions?

- OK, but why neural networks, instead of, say, production systems, symbolic AI?

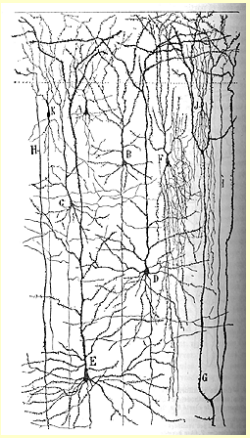
Why neural networks?

- Offers a constrained, uniform framework: The basic processes of most models are the same, and are fairly simple.
- Common language for cognition/brain sciences: models may be more neurally or cognitively oriented (data constraints), but they are easily relatable.
- Networks make errors, can be impaired: computational neuropsychology
- Previous successes! New work should relate to the best previous work, and many leading models are networks.

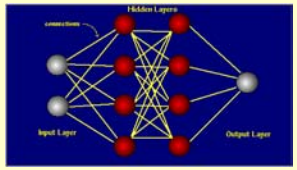
Basic elements of neural networks

- Revision: basic structures and processes in connectionist models
- Forms of representation using units and weights
- Learning: supervised and unsupervised
- Lesioning models: neural networks and cognitive neuropsychology

- The main features we will look at are:

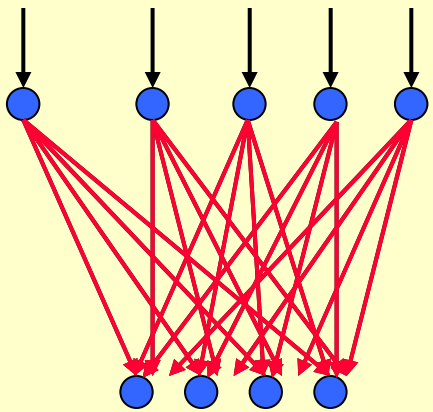


Brains		Neural Nets
Neurons	→	Nodes or Units
Firing Rate	→	Activation Level
Axons, Dendrites, Synapses	→	Connections
Excitatory v. Inhibitory Inputs	→	Excitatory v. Inhibitory Inputs



Nodes and Neurons

- The basic constituents of a neural network are Nodes or Units.



- The nodes are linked by excitatory and inhibitory connections

- When nodes are activated (e.g., by a stimulus), the activation spreads along the connections.

Nodes and Neurons

- Nodes in cognitive models are not equivalent to single neurons in the brain.

- However, they may act as an approximation to a group of neurons which tend to become active under the same circumstances (e.g., whenever you hear the voice of your best friend).

Firing Rate and Node Activation Level

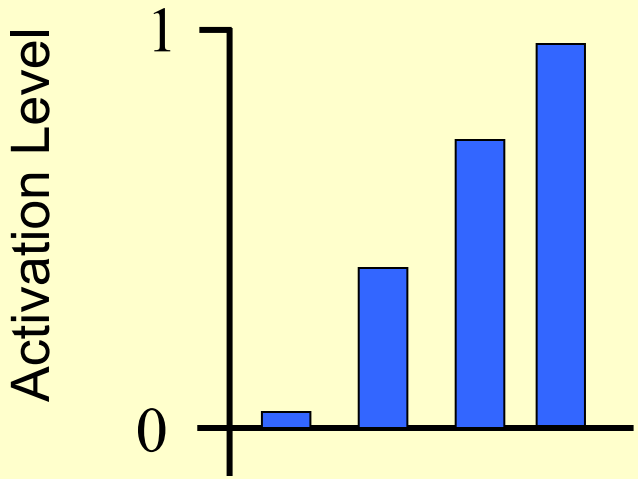
- An important aspect of representation in the brain is:
  - the firing rate of individual neurons
  - or the number of active neurons in a cooperative group of neurons.
- In both cases, we can talk about the degree of activation of the neural representation.

Firing Rate and Node Activation

- This “degree of activation” can represent quantitative variables, e.g.
  - Stimulus strength - e.g., intensity of a light source.
  - Confidence - how certain it is that the thing the neurons represent is perceived.
  - Response strength - e.g., grip force with the hand.
  - Response tendency - e.g., the strength of the urge to act in a certain way.

Firing Rate and Node Activation

- In artificial networks, this feature is captured by the <u>Activation Level</u> of Nodes, represented by a number generally between 0 and 1.



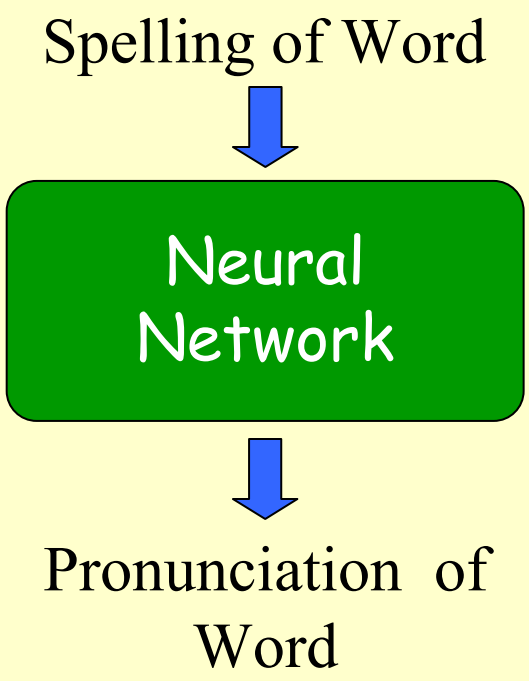
- 0 means a node is OFF – i.e., there is no activity.
- 1 means it is maximally active.
- Values between 0-1 represent the degree of activation.

Doubt and Ambiguity in Networks

- It is possible in network models for nodes representing different, incompatible, things to be active at the same time. This represents a state of uncertainty.
- For instance, how would you pronounce the nonword *grook?* As “gruck”, or “grewk”?
- Subjects name such “inconsistent” nonwords more slowly than stimuli such as *trank* (only one possible pronunciation, Glushko, 1970, J. Exp. Psych: Hum.Perc & Perf, 674-691).

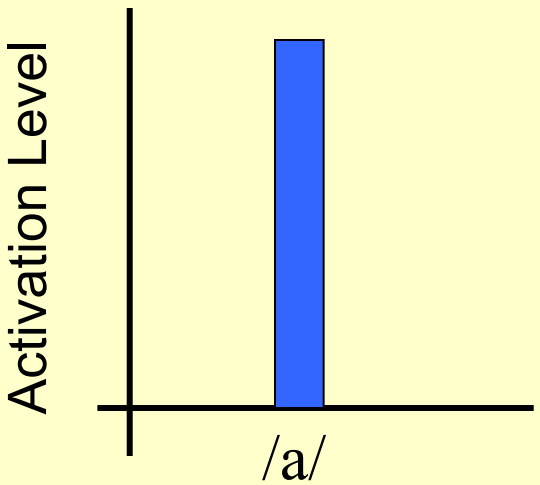
Doubt and Ambiguity in Networks

•Zorzi, Houghton & Butterworth* explain this effect with a neural network model of reading which can pronounce written nonwords.



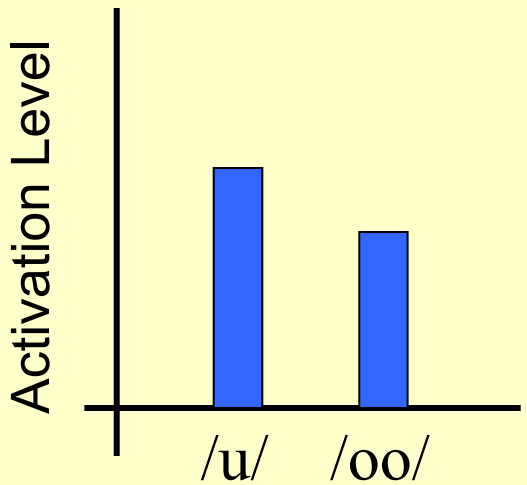
*1998. J. Exp. Psych: Hum. Perc. & Perf., 1131-1161

1. Reading a "regular" nonword - *trank*



Unambiguous vowel – fast RT.

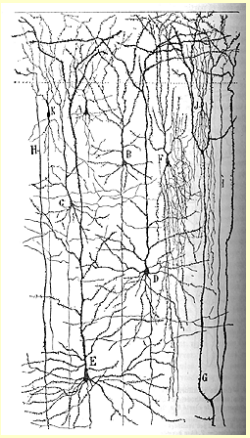
2. Reading an "inconsistent" nonword" - *grook*



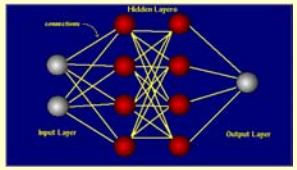
"In two minds" – RT slowed due to competition between vowels.

Graphs show activation of nodes representing vowel sounds

- The main features we will look at are:



Brains		Neural Nets
Neurons	→	Nodes or Units ✓
Firing Rate	→	Activation Level ✓
Axons, Dendrites, Synapses	→	Connections
Excitatory v. Inhibitory Inputs	→	Excitatory v. Inhibitory Inputs

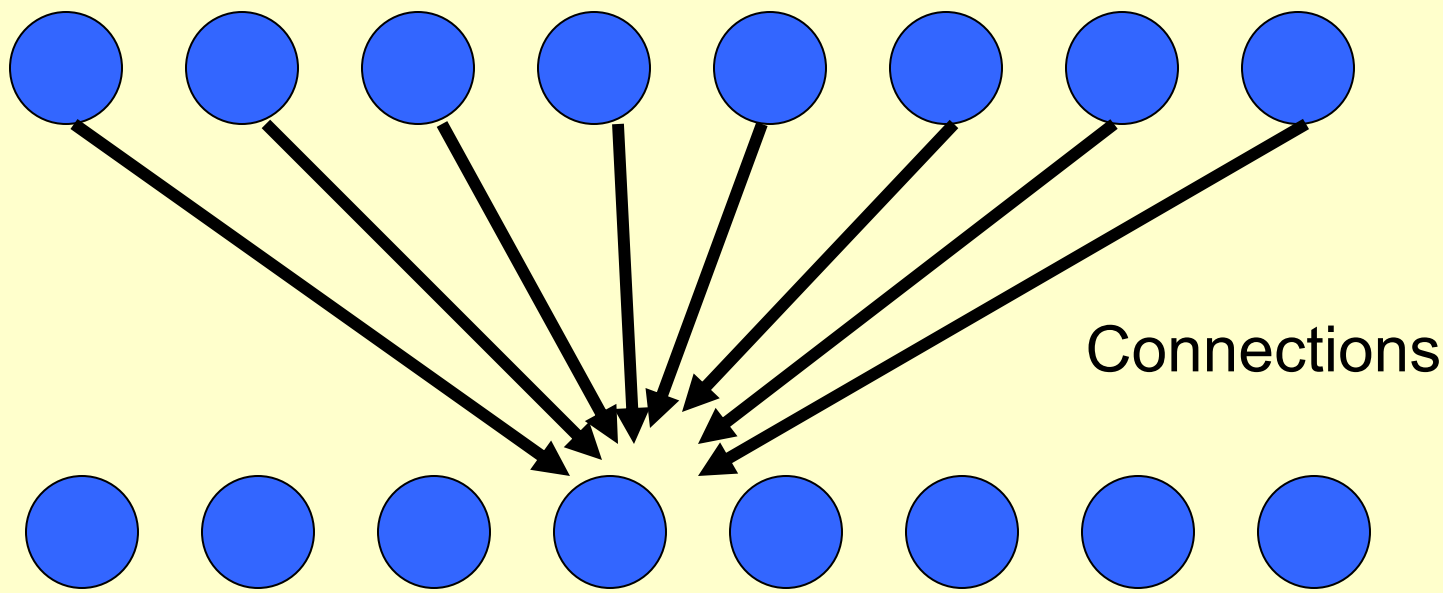


Connections in Neural Networks

- Nodes are connected, and the connections have “weights”, numbers representing the type and strength of the connection.
- Neural networks typically show both Convergent and Divergent Connectivity patterns...

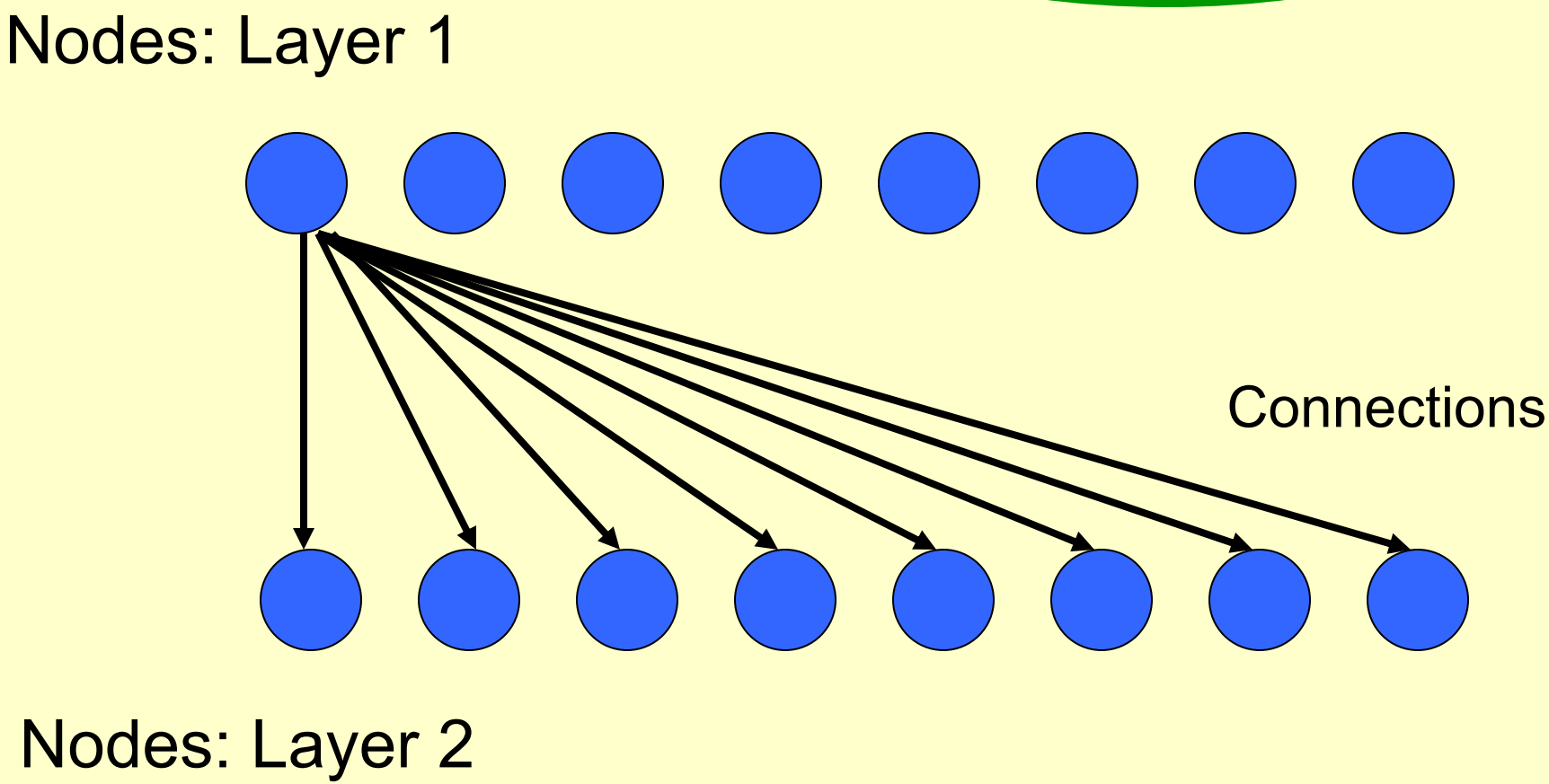
Connections between Nodes: Convergence and Divergence

Nodes: Layer 1

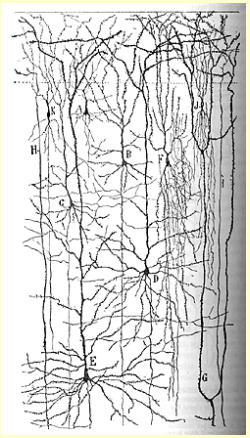


Nodes: Layer 2

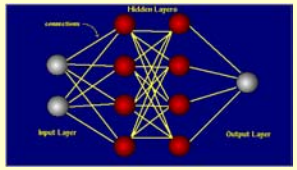
Connections between Nodes: Convergence and Divergence



- The main features we will look at are:



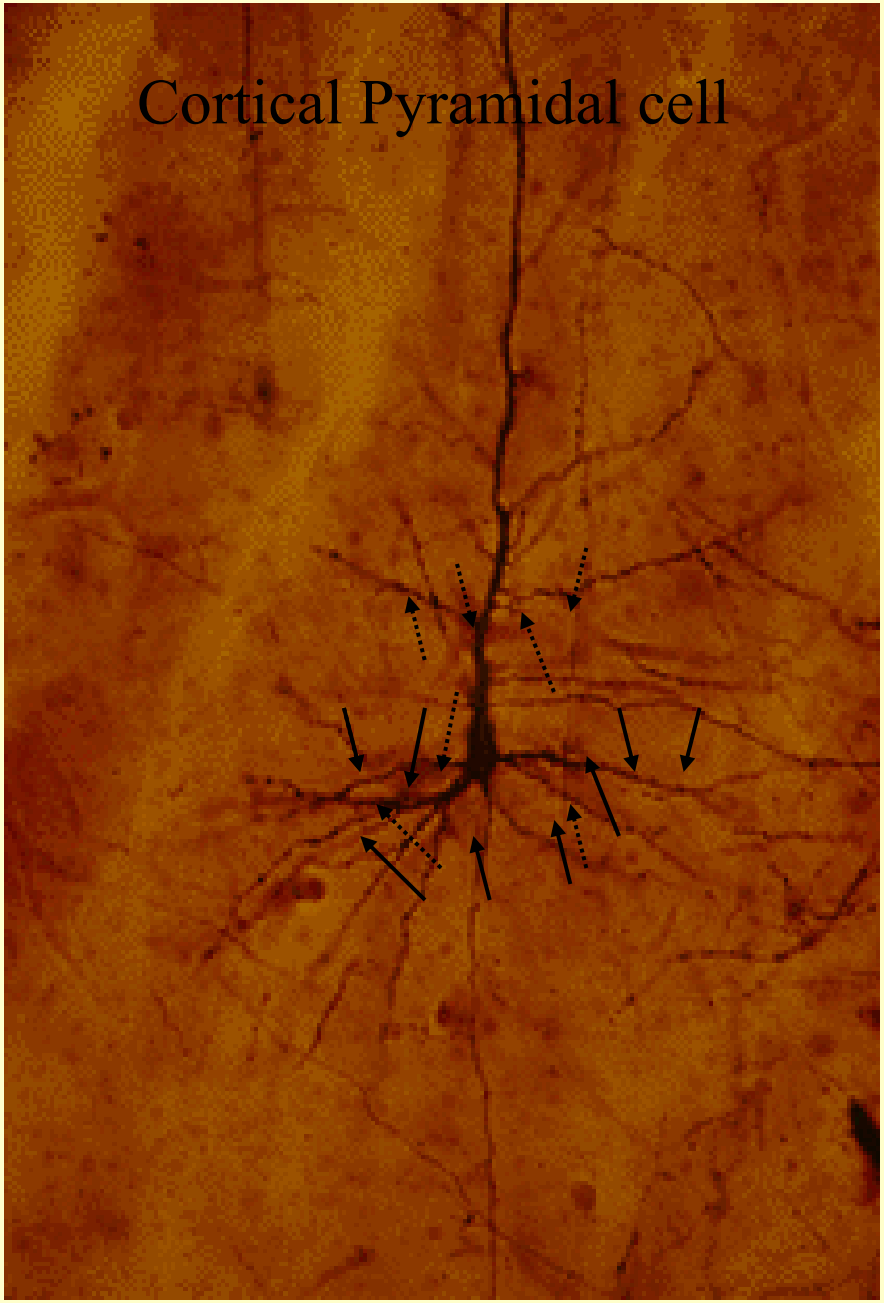
Brains		Neural Nets
Neurons	→	Nodes or Units ✓
Firing Rate	→	Activation Level ✓
Axons, Dendrites, Synapses	→	Connections ✓
Excitatory v. Inhibitory Inputs	→	Excitatory v. Inhibitory Inputs



Excitation and Inhibition

- In the brain, neurons interact via synapses, which may Excitatory or Inhibitory.
  - Action potentials arriving at excitatory synapses activate the post-synaptic neuron - increasing its firing rate.
  - Action potentials arriving at inhibitory synapses de-activate the post-synaptic neuron - decreasing its firing rate.

- The two types of synapse can occur together.
- For instance, they are found intermingled on the dendrites of pyramidal cells in the cortex.



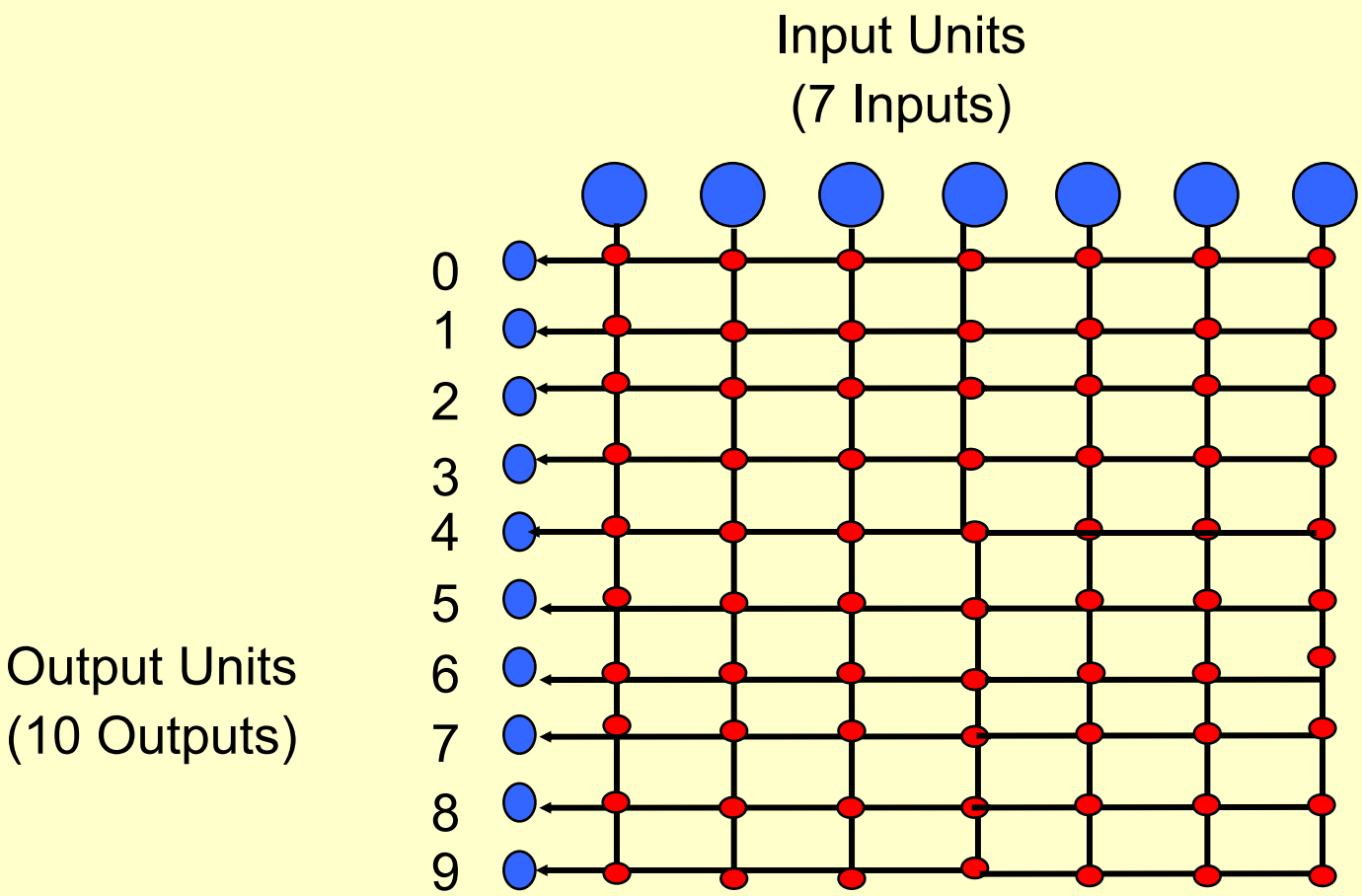
Excitatory Inputs ⟶

Inhibitory Inputs ⇢

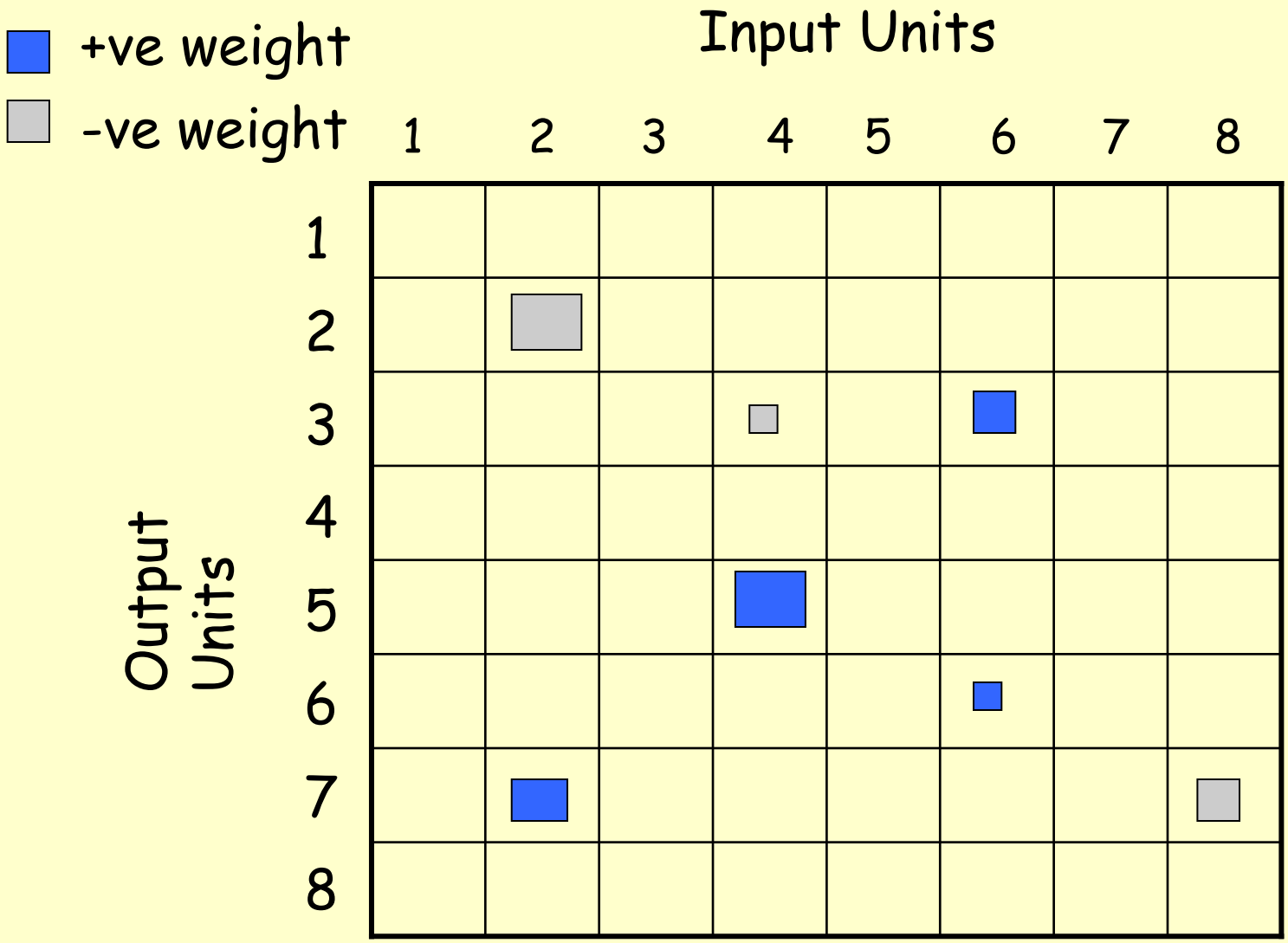
Excitation and Inhibition

- In neural networks, the equivalent are positive (excitatory) and negative (inhibitory) weights on the connections between nodes.
- The activation of units is multiplied by the weights on the connections to other units, producing either excitatory or inhibitory signals, depending on the weight.
- All the connections between 2 groups of units can be shown as a table, or 2-dimensional array (matrix).

Connection Weights in an Array (Matrix)



Connection Weights in an Array

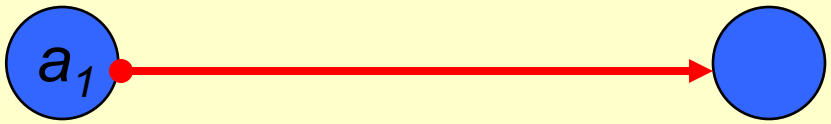


The "strength" of a connection weight is shown by the size of the rectangle

Neural Networks - spread of activation

Sending Unit - u_1

Receiving Unit - u_2



Connection Weight - $w_{1,2}$

- The activation of unit 1, a_1, flows along the weighted connection
- The input received by unit 2 is the activation times the weight:

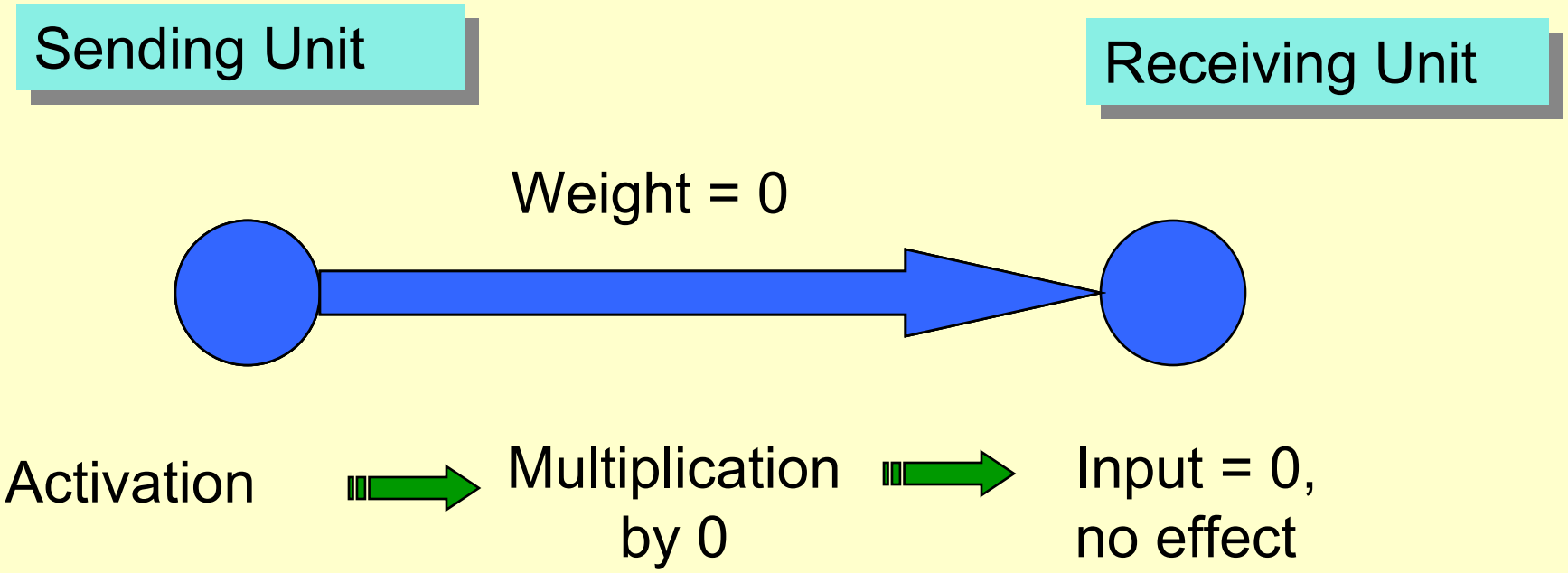
$$\text{input to } u_2 = a_1 \text{ X } w_{1,2}$$

Spread of Activation: The effects of weights

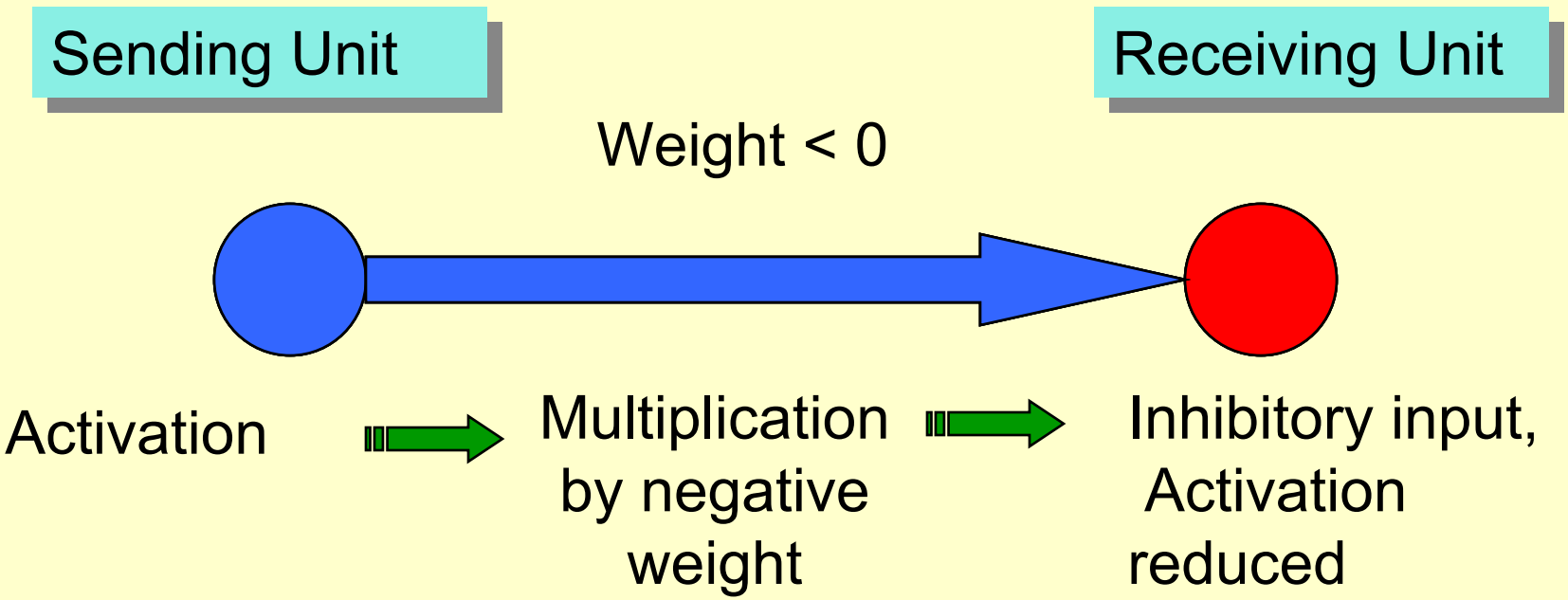
The way in which activation of one unit affects the activation of another to which it is connected depends on the weight on the connection.

1. Zero Weights - No spread of activation
2. Negative Weights - Inhibition, reduction in activation
3. Positive Weights - Excitation, increase in activation

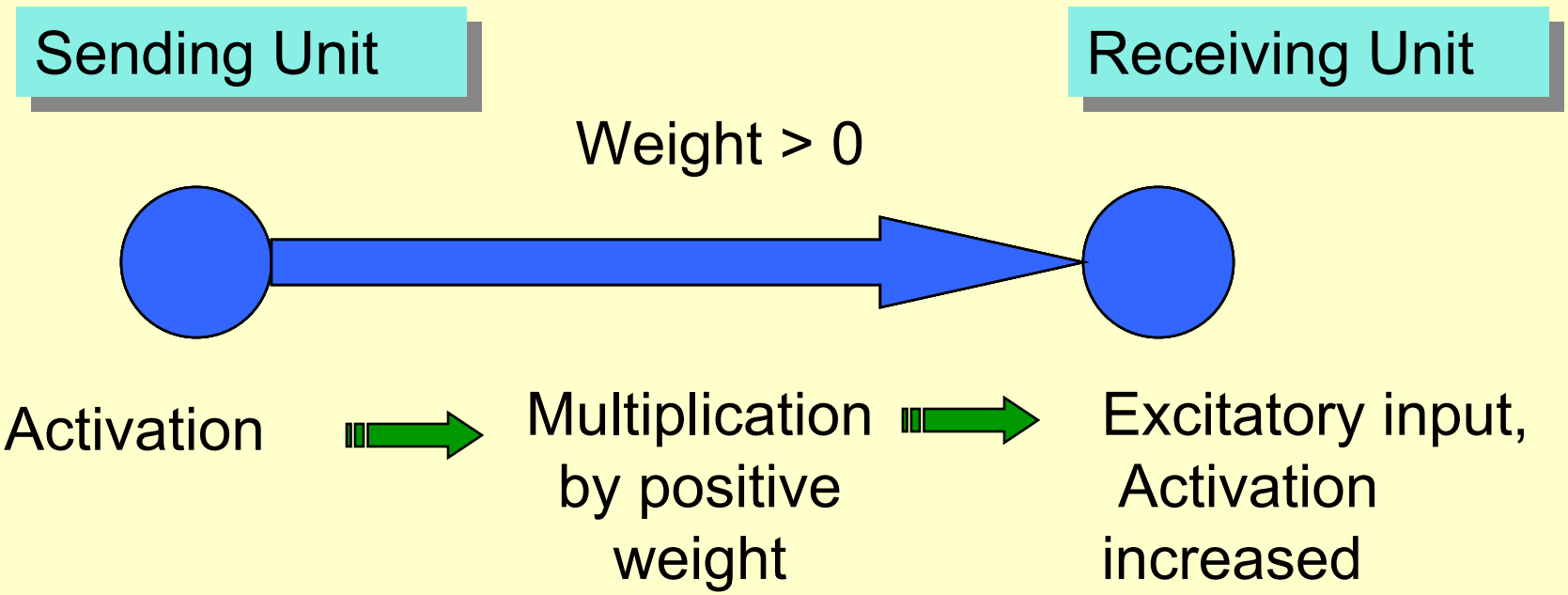
Spread of Activation 1: Zero Weights



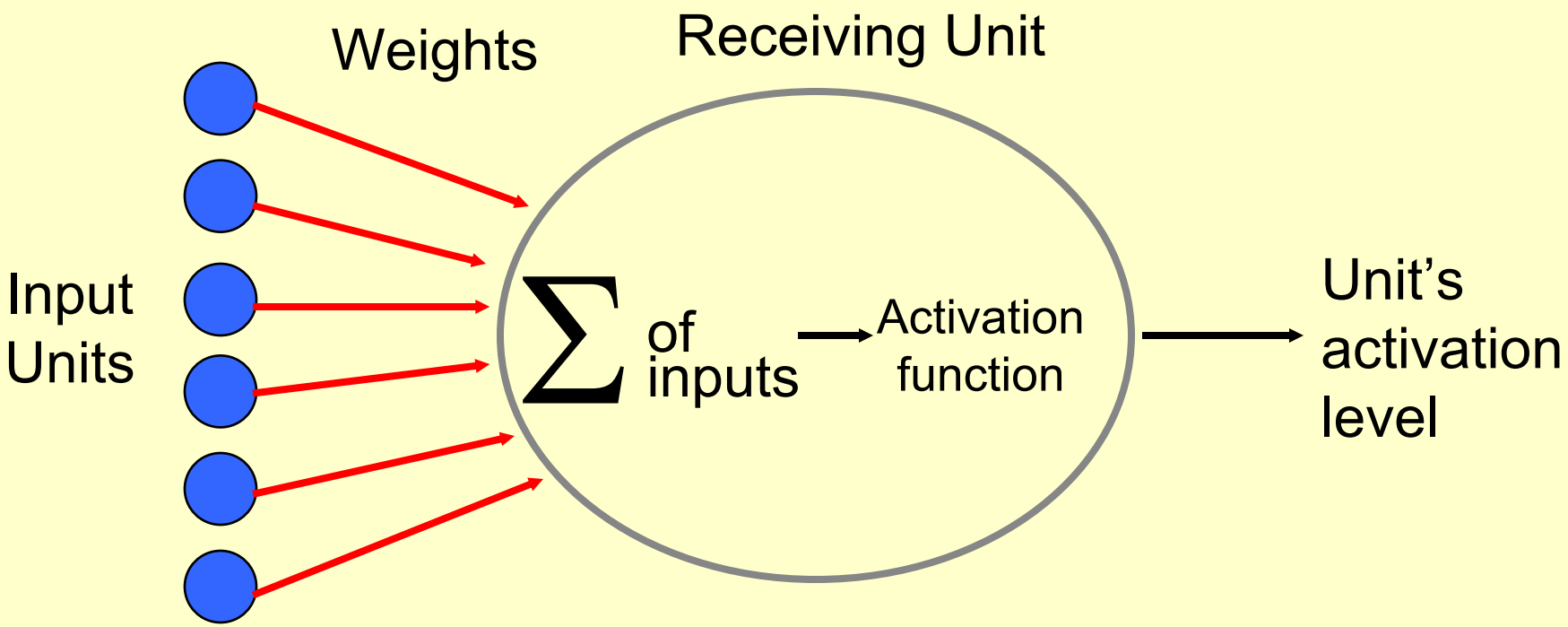
Spread of Activation 2: Negative Weights



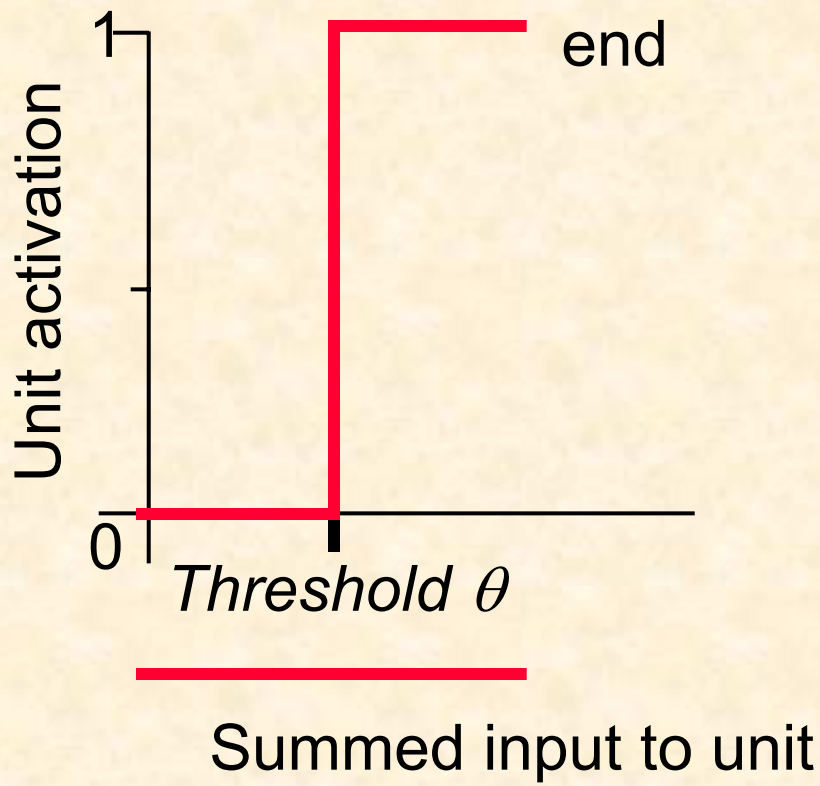
Spread of activation 3: Positive Weights



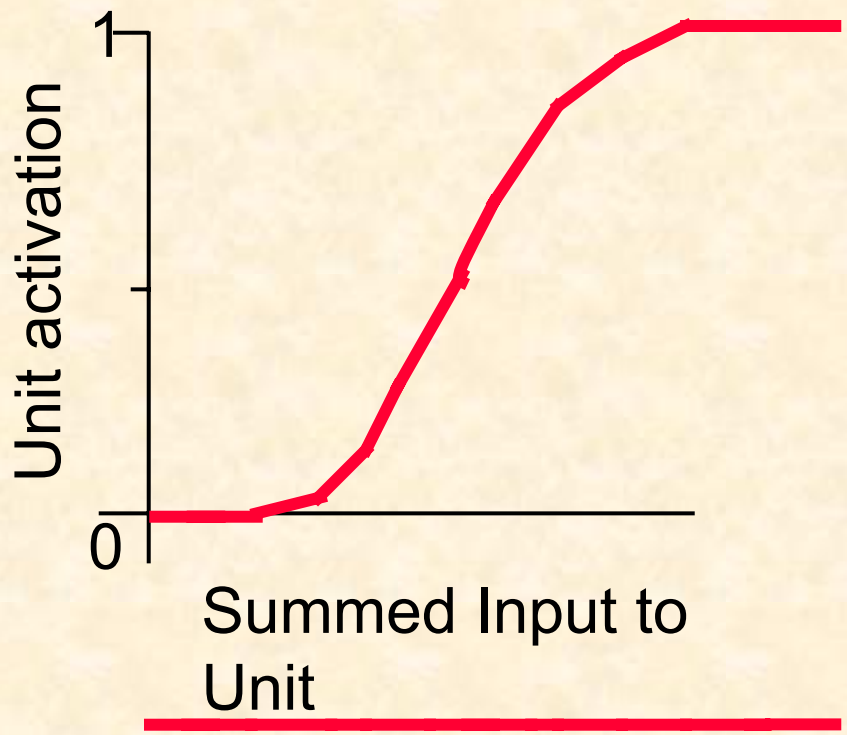
Units sum their inputs



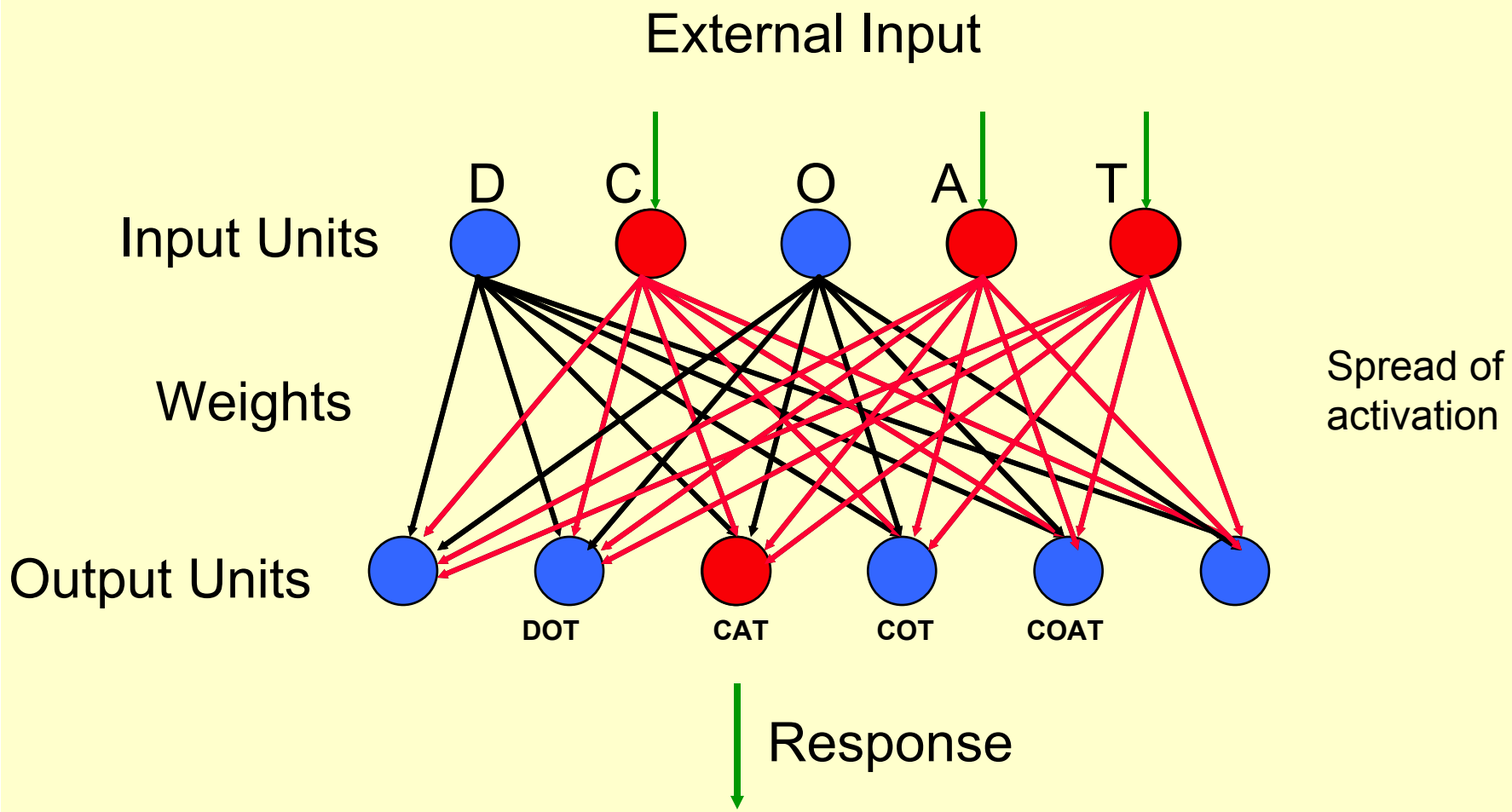
Activation Functions 1: Binary threshold



Activation Functions 2: Sigmoid (s-shaped)



The behaviour of networks depends on the weights between units



Changing weights changes the network's behaviour

- Spread of activation via weights transform patterns of activation in one "module" (set of nodes) into patterns in others (e.g., from input units representing letters to output units representing words).
- Getting the right output (response) for a given set of inputs depends on having the "right" weights.
- Rules for generating weights to get networks to perform well are called LEARNING ALGORITHMS.

Learning Algorithms

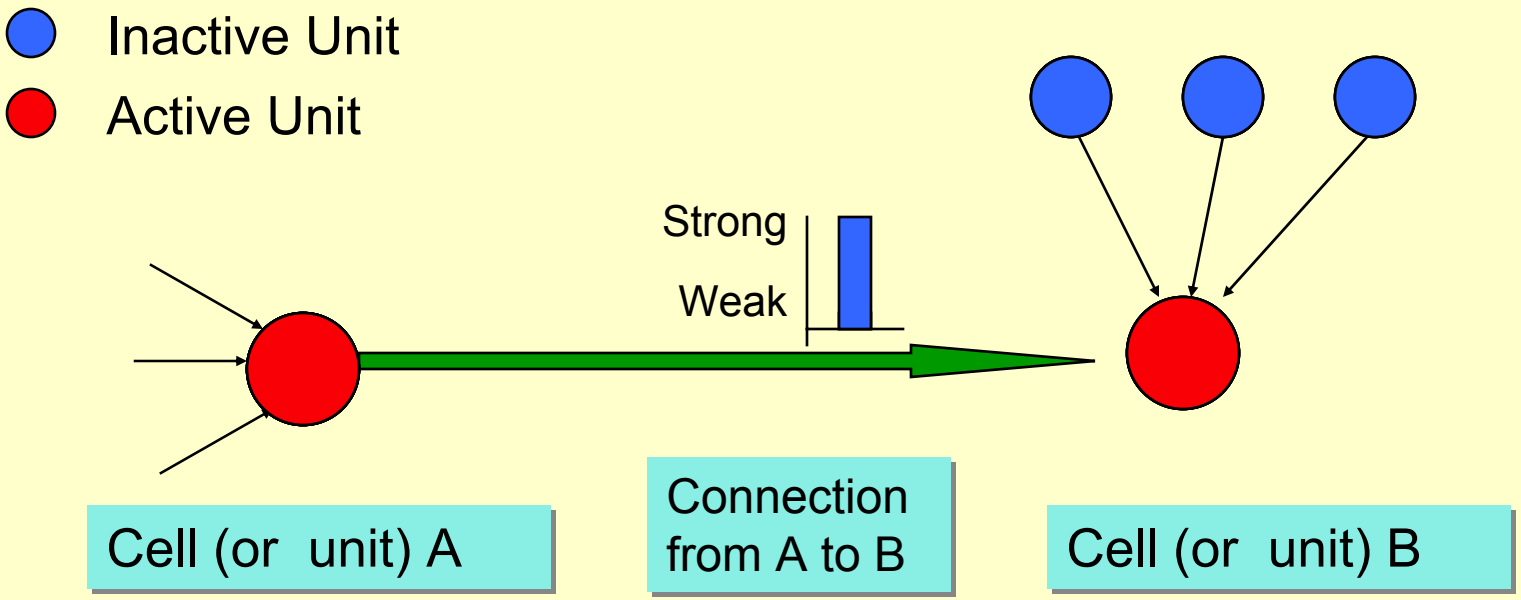
- Neural network learning algorithms change weights between units.
- **Unsupervised Learning:** weights are changed on "internal" criteria, without feedback on the network's performance.
- **Supervised Learning:** the network's responses to inputs are compared to feedback from an omniscient teacher. Learning reduces error.

The Hebb Rule

- From D.O. Hebb, (1949), The Organization of Behavior.

  "When an axon of cell A is near enough to excite a cell B and 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 efficiency, as one of the cells firing B, is increased."

The Hebb Rule



Stage 1: Weak connection: Cell A alone cannot activate cell B

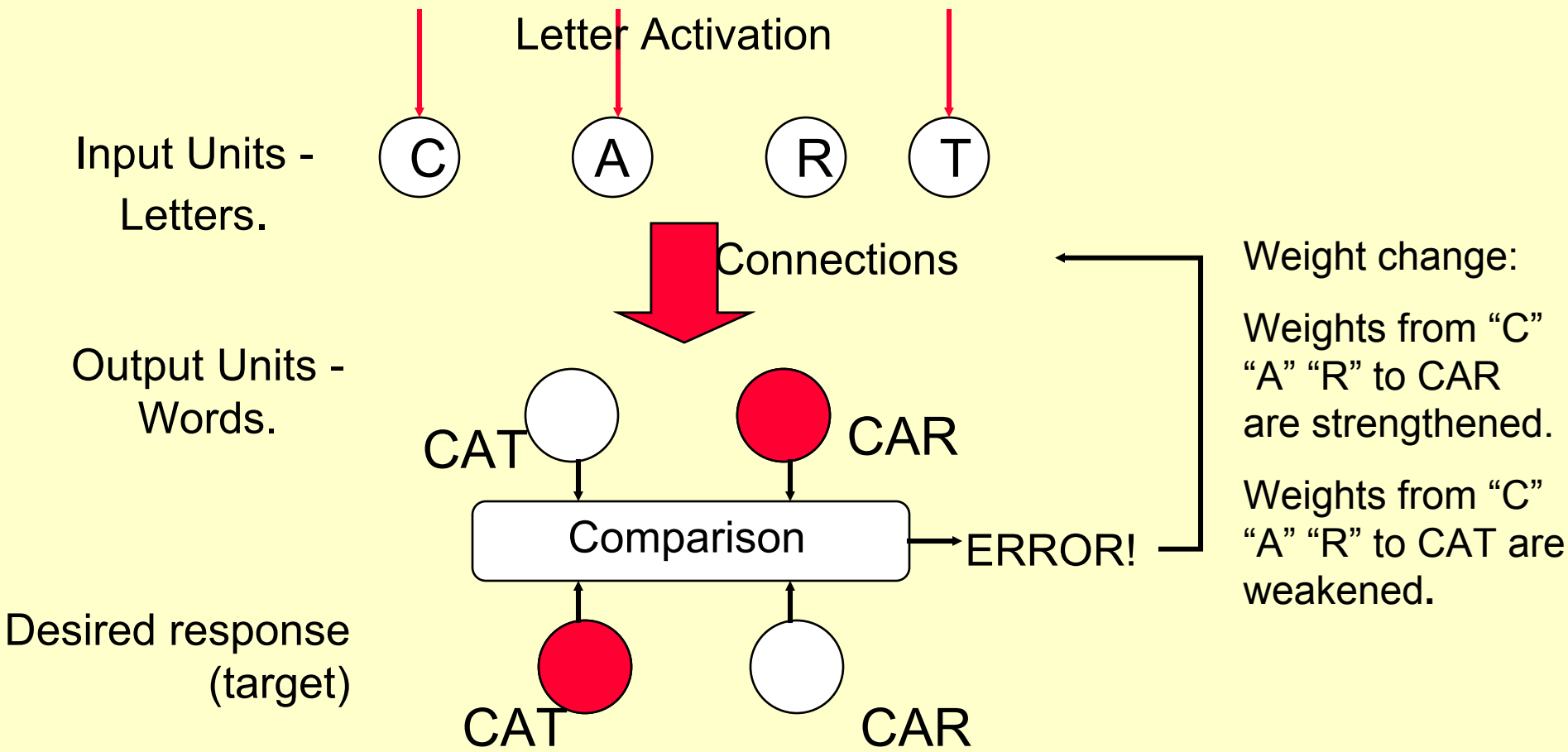
Stage 2: Cells A and B are simultaneously active: Connection strengthens

Stage 3: Strong connection: Cell A alone can activate cell B

Supervised Learning: The Delta Rule

- In supervised learning, the response (output) of a network to an input is compared to a TARGET (correct) response, and the ERROR made by any output unit is measured.
- The error is then used to change the weights in the network so that when the input is presented again the network's response will have improved (the error will be less).

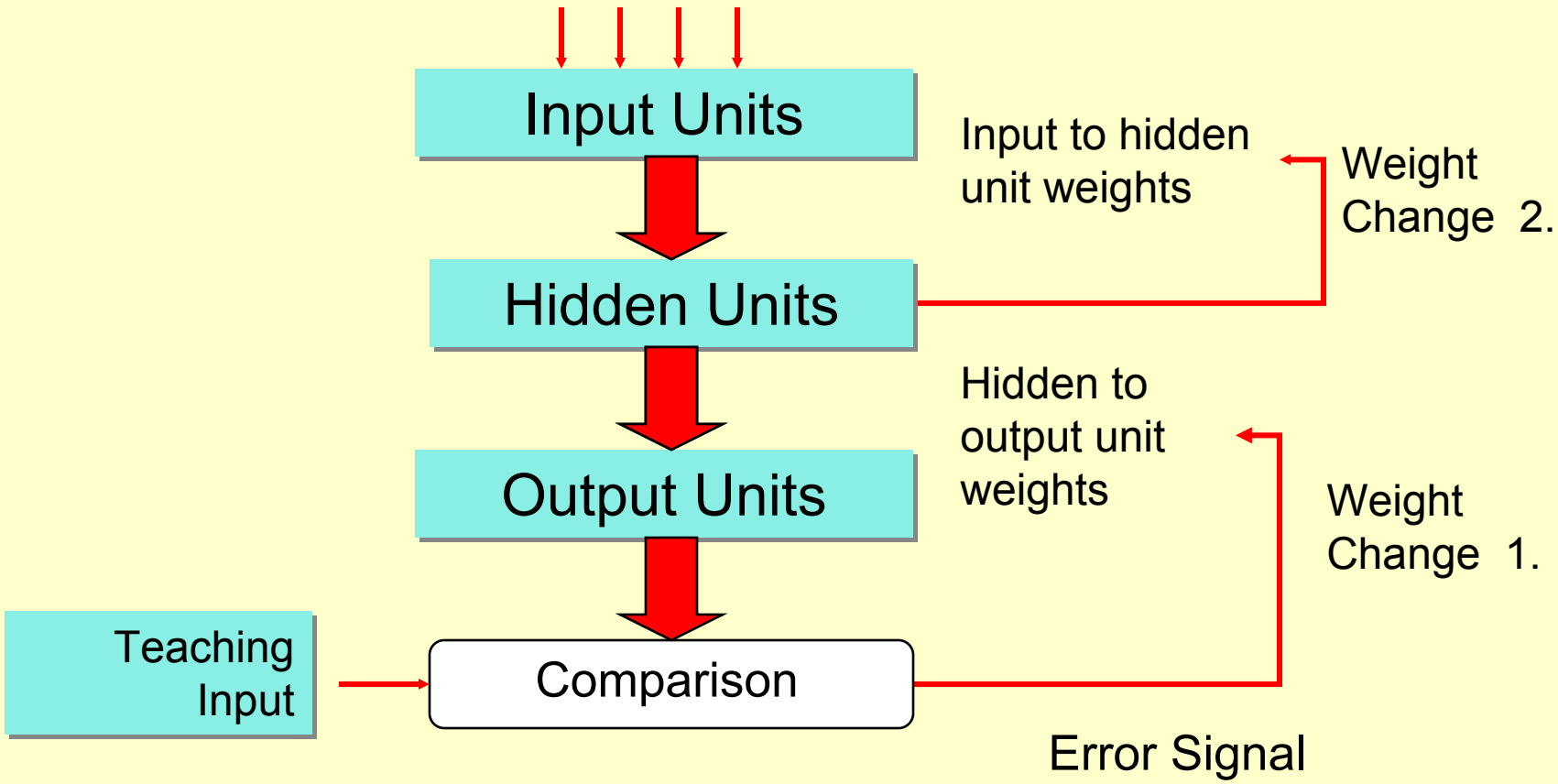
The delta rule - learning by error correction



Supervised learning for multi-layer nets: backpropagation

- The delta rule works for networks with 2 layers of units.
- Such networks have limitations on what they can in principle do - though anything they can do, they can learn to do using the delta rule
- More powerful networks can be produced using 3 layers of units - the units between input and output units are called “hidden units”
- These networks can be trained with a “generalization” of the delta rule, known as BACK-PROPAGATION.

Multi-layer networks and backpropagation



See e.g., Plaut et al., 1996, *Psychological Review, 103, 56-115*

Forms of representation in neural networks

1. Local Representations

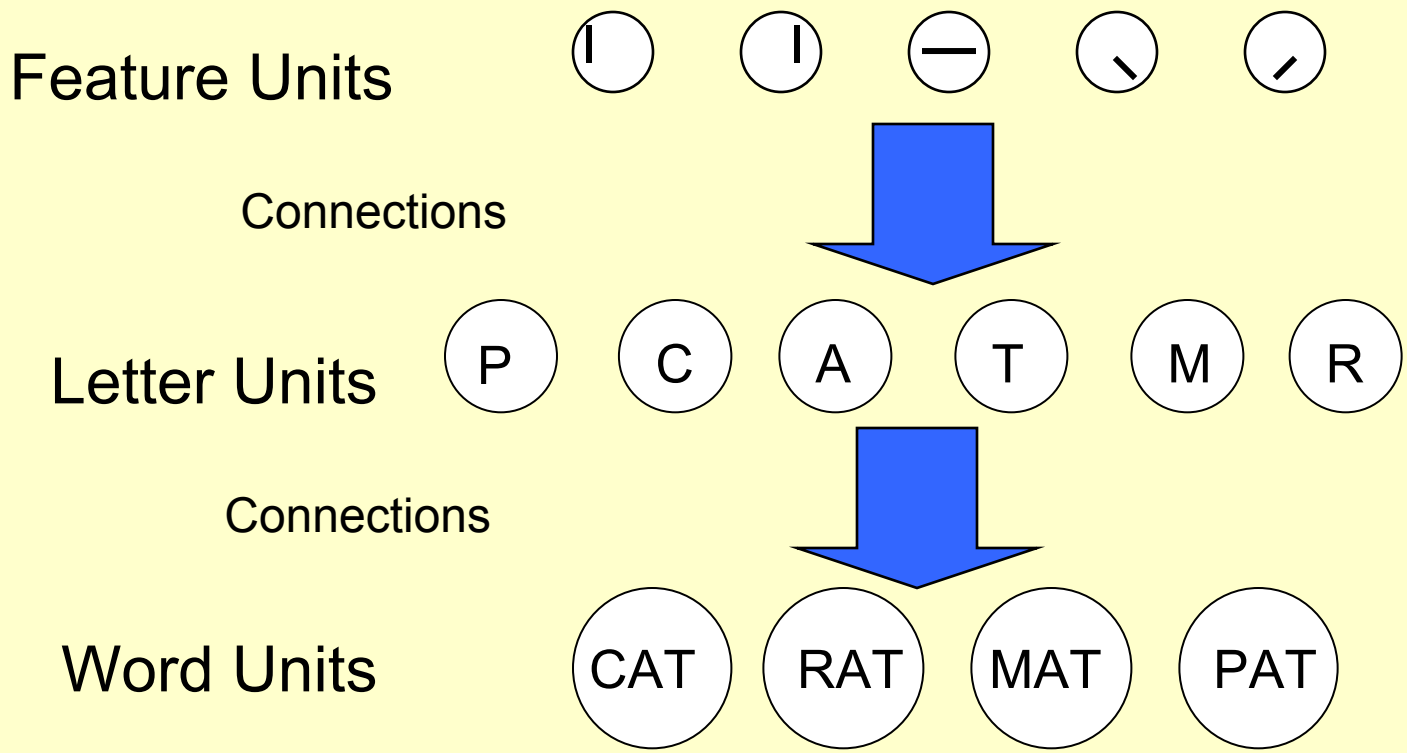
2. Distributed Representations

3. Population Coding

Local Representations

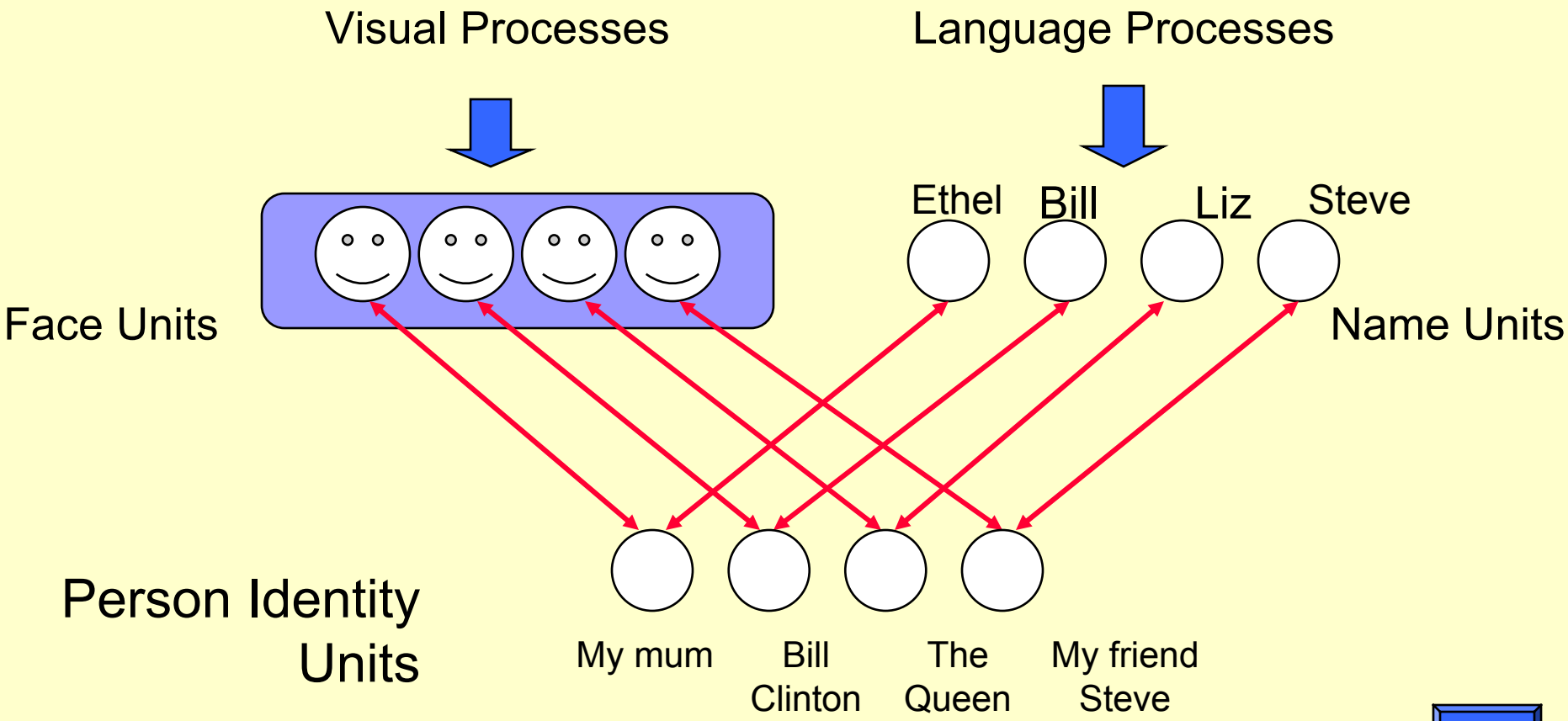
- Particular units stand for distinct represented items,
  - e.g., in a model of the mental lexicon with “lexical” units, each unit stands for a different word.
  - in a model of face recognition each unit stands for the identity of a distinct individual

Local Representations 1: Reading



McClelland & Rumelhart, 1981. *Psych. Review, 88, 375-407.*

Local Representations 2: Face Recognition

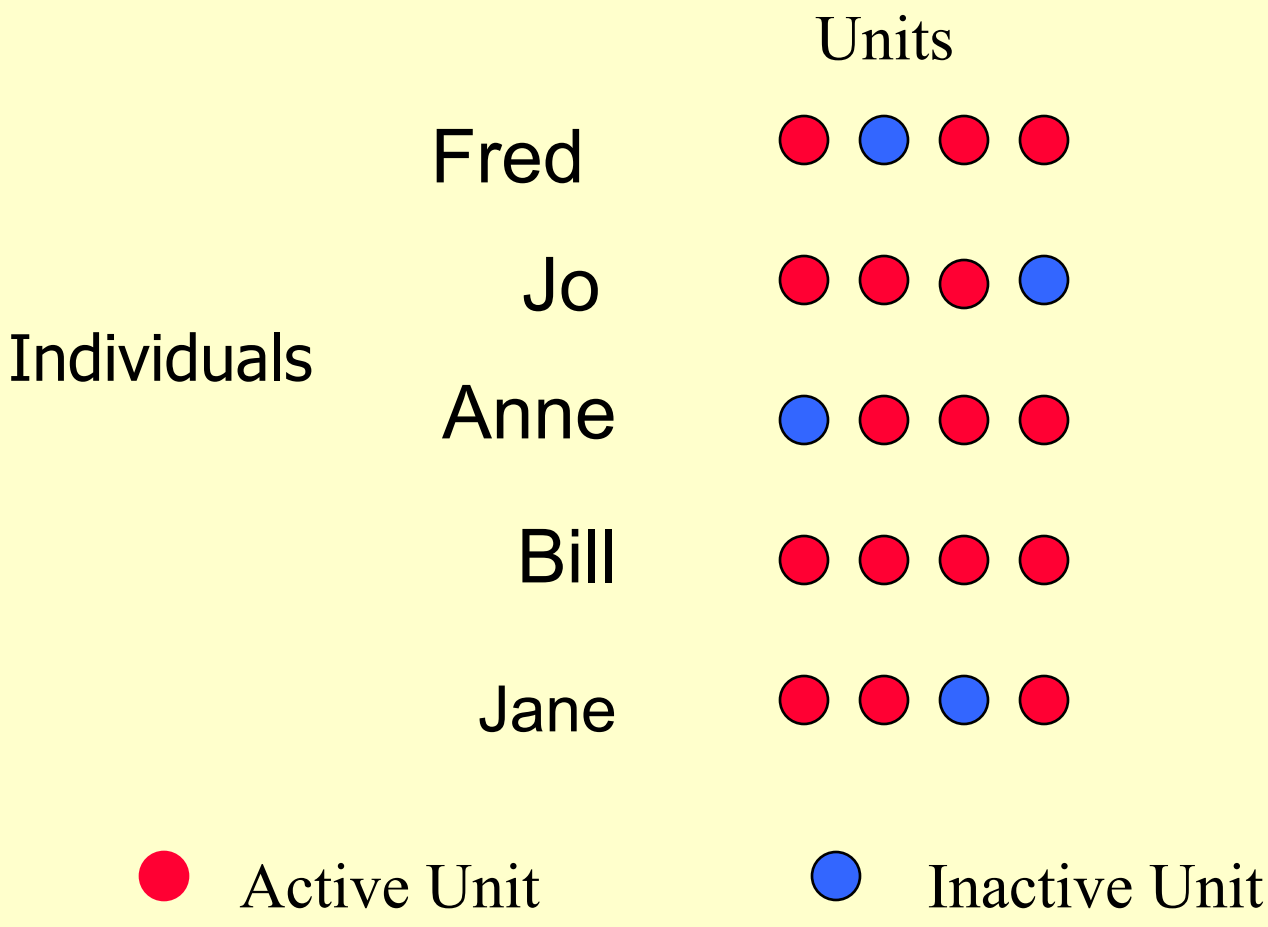


Burton et al., 1991. *Cognition, 39, 129-166.*

Distributed Representations

- Items are represented by *patterns of activation* over sets of units. Each item has its own distinct pattern, but patterns may overlap.
- Each unit may play in a part in the representation of many different items.

Distributed Representations



Advantages/Disadvantages

- Local representations: FOR:
  - Orthogonality (independence) good – individuation.
  - Multiple activations/competition (without need for attractors).
  - Simple learning rules (Hebb rule, Self-organising maps etc).
  - Transparency of model.

Advantages/Disadvantages

- Local representations: AGAINST
  - Orthogonality (independence) bad: similarity not encoded.
  - Wasteful – most units (neurons) not active most of the time.
  - Not enough neurons (?)
  - Learning new things: where do the units come from? What where they doing prior to learning?

Advantages/Disadvantages

- Distributed representations: FOR:
  - Efficient use of resources (up to 2^n patterns over n units)
  - Similar things can have similar activation patterns: generalisation of learning.
  - Learning new things: formation of new patterns over units being already used.

Advantages/Disadvantages

- Distributed representations: Against:
  - Lack of individuation: How do we distinguish multiple inputs?
  - Similar things can have similar patterns: *spurious* generalisation of learning.
  - Learning algorithms are typically more complex and biologically implausible.

Population coding

- Each unit (or neuron) represents a particular value of some variable, e.g., position, direction, but is broadly “tuned” to that value.
- Hence each unit will be active to some extent for values close to its preferred value.
- The actual represented value is computed from the distribution of activation in an “ensemble” (population) of units.

Population coding: Representation of direction

Unit activation:
Length = activation level
Angle = preferred direction

Resultant of population