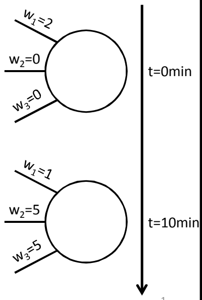
CISC 3250
Systems Neuroscience

Neuroplasticity: Learning in Neurons



Professor Daniel Leeds
dleeds@fordham.edu
JMH 328A

1

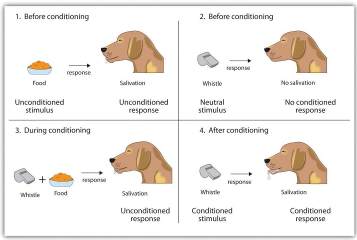
Two forms of plasticity

- **Structural plasticity**: generation of new connections between neurons
- **Functional plasticity**: changing strength of connections between neurons

Hebbian plasticity:
"cells that fire together, wire together"

2

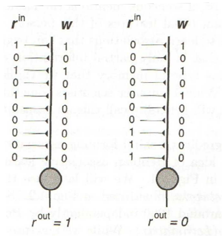
Cognitive level: Conditioning



Associating both smell and whistle with food
- **Unconditioned stimulus**: smell – already associated with food
- **Conditioned stimulus**: whistle – indicates food coming

Computing level: Associator network

Define input $h = \sum_i w_i r_i^{in}$
Neuron fires when $h > 1.5$ – step activation function



Smell input   Whistle input

At each learning step, add 0.1 to weights of pre-synaptic inputs co-occuring with post-synaptic firing

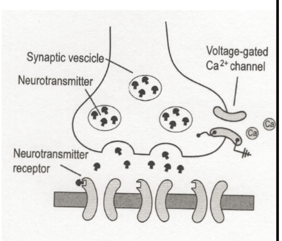
4

Chemical level: NT receptors

Increase weight by improving NT detection

Post-synaptic:
- Insert more receptors into dendrite membrane
- Improve performance of receptors

Pre-synaptic:
- Increase amount of NT released



5

Marr's levels of analysis

- **Computational theory:** Learn associations among sensations
- **Representation and algorithm:** Associate each sense with set of neural outputs, adjust weights on these outputs into another neuron
- **Hardware implementation:** Insert/remove NT receptors from dendrites

6

Features of associators

- Pattern completion/ generalization

- Fault tolerance
  - Selected dendrites miss input, post-synaptic neuron still fires
- Learning prototypes

  - Neuron firing for common combinations

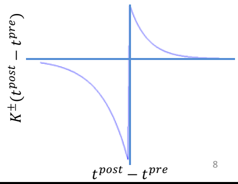
7

Math of Hebbian spike learning

- Pre-synaptic spike followed by post-synaptic spike -> increase weight
- Post-synaptic spike followed by pre-synaptic spike -> decrease weigh

$$\Delta w_{ij}^{\pm} = \epsilon^{\pm}(w) K^{\pm}(t^{post} - t^{pre})$$

Prevent weights from increasing to ∞



8

Math of Hebbian rate learning

"Cells that fire together, wire together"

$$\Delta w_{ij} = \epsilon r_i r_j$$

9

Weight decay

- Synaptic weights are finite
- Propose learning rules that keep weights bounded

$$\Delta w_{ij} = r_i r_j - c w_{ij}$$

$$\Delta w_{ij} = r_i (r_j - w_{ij})$$ – *Willshaw*

$$\Delta w_{ij} = r_i r_j - (r_i)^2 w_{ij}$$

- Or, preserve total synaptic weight across network:

$$w_{ij} \leftarrow \frac{w_{ij}}{\sum_j w_{ij}}$$

10