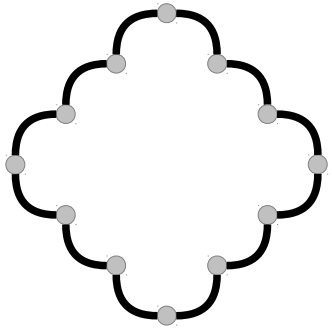
From structures to functions

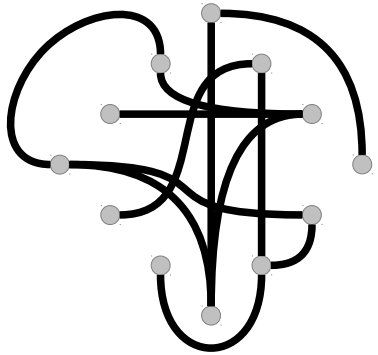
1 – Static Networks

2 – Dynamic Networks

Elements of graph theory

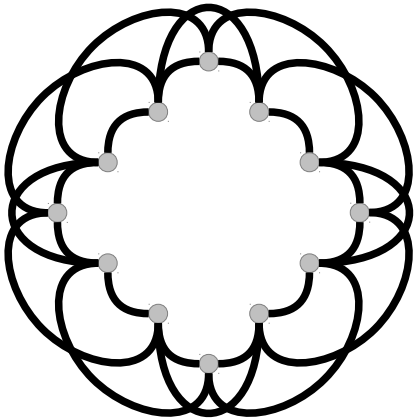


Ordered network
- 12 nodes
- 24 connections
- average connectivity: 2
- longest path: 6

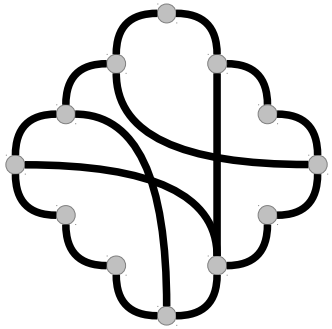


Random network
- 12 nodes
- 24 connections
- average connectivity: 2
- longest path: 7

Elements of graph theory

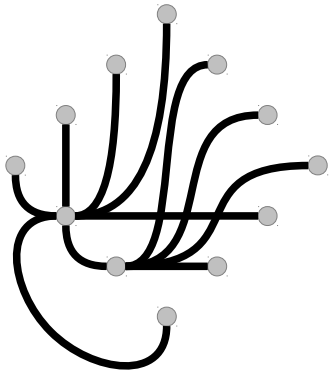


Ordered network
- 12 nodes
- 48 connections
- average connectivity: 4
- longest path: 3

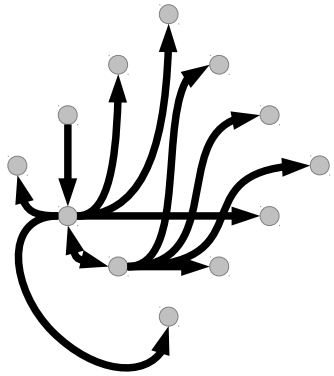


Small world network
- 12 nodes
- 32 connections
- average connectivity: 2.6
- longest path: 3

Elements of graph theory

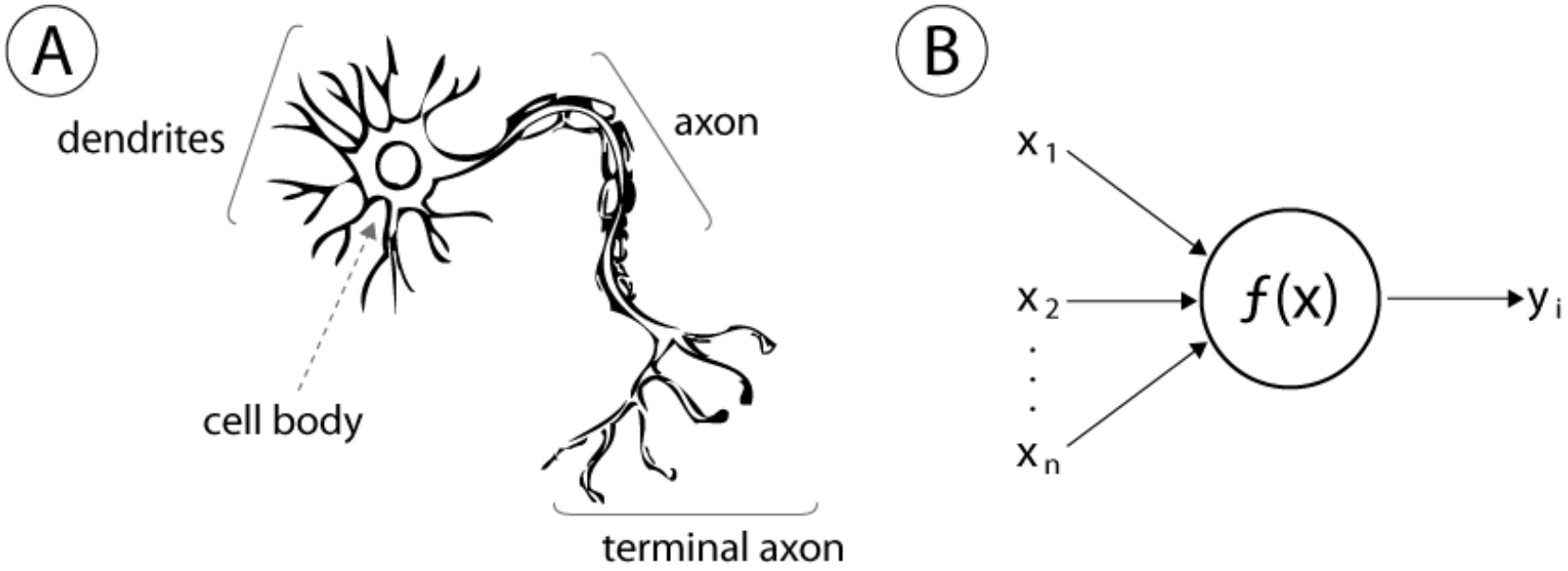


Scale-free network
- 12 nodes (2 hubs)
- 22 connections
- average connectivity: 1.8
- longest path: 3



Oriented graph
- 12 nodes
- 12 connections
- average connectivity: 1
- longest path: --

PDP, step 1: perceptron

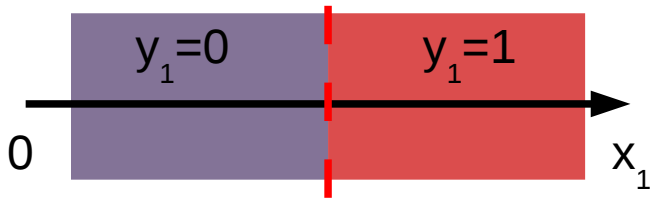


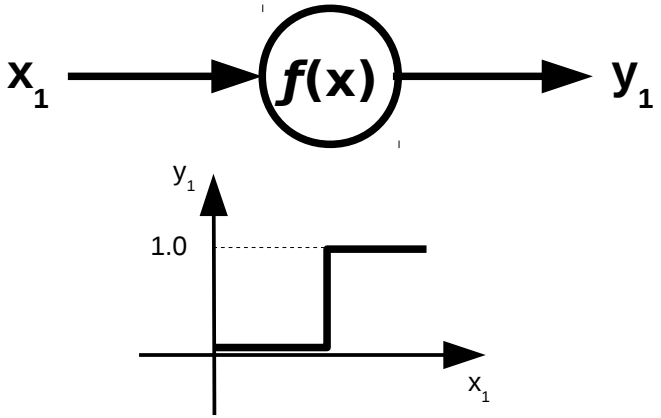
Parallel Distributed Processing (PDP), elements:

– Environment encoded into numerical input.
– Pattern of afferent connectivity, propagation rules.
– Internal computation (transfer function) resulting in activation states.
– Decoded output.

Image from: David Poole and Alan Mackworth, 2010

PDP, step 1: perceptron





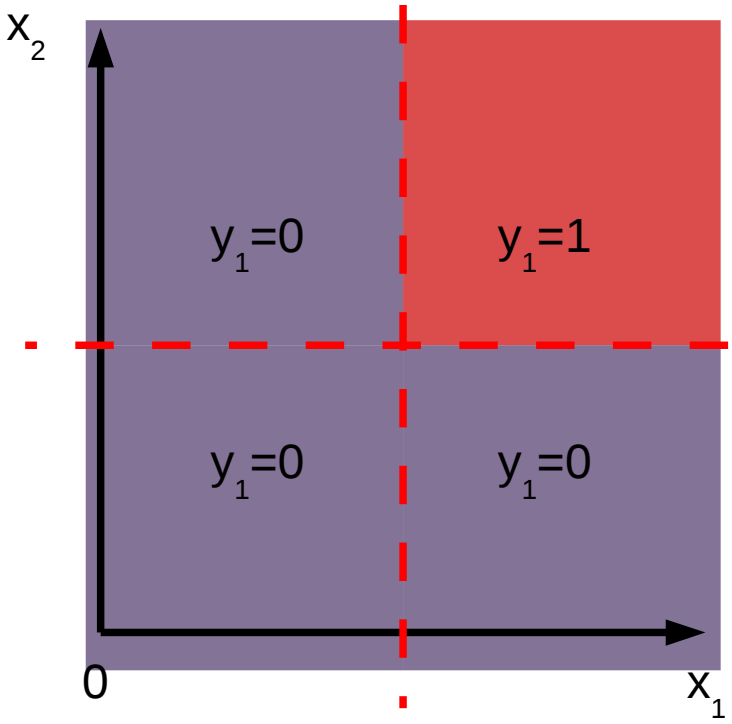
Output: 2 categories

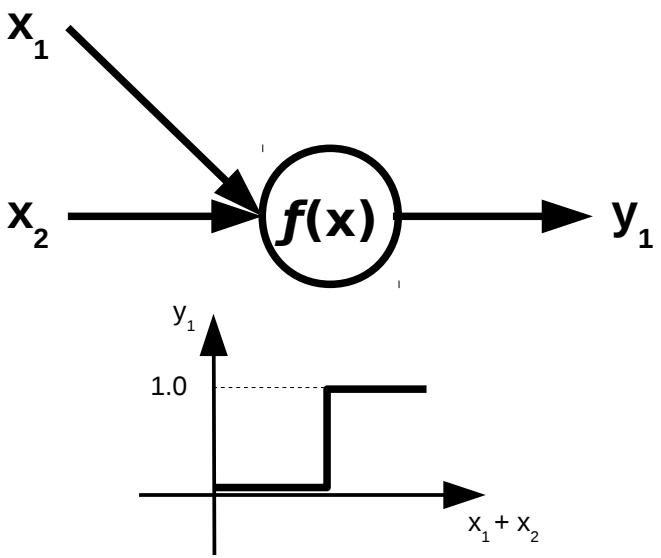
- $y_1=0$
- $y_1=1$

Perceptron

- Input: one dimension environment.
- Processing units (neurons): one
- Connectivity: one afferent, one efferent.
- Activation rule: threshold.

PDP, step 1: perceptron





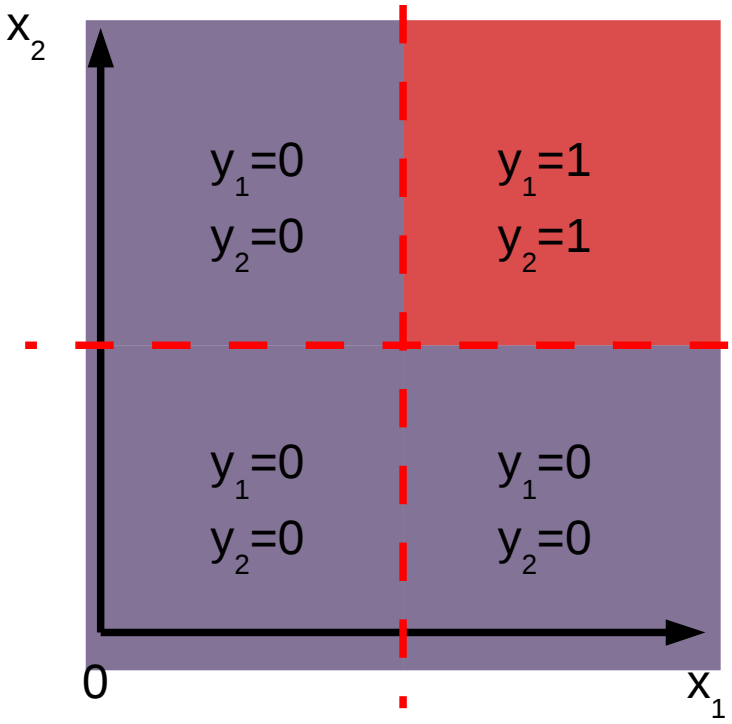
Output: 2 categories

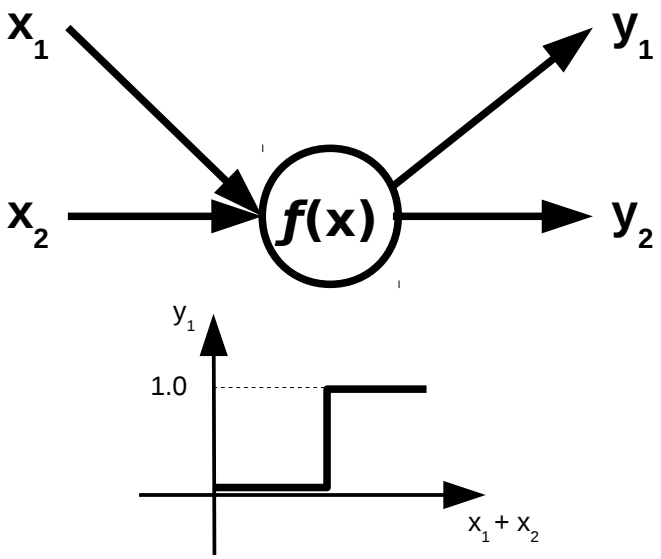
- $y_1=0$
- $y_1=1$

Perceptron

- Input: two dimension environment.
- Processing units (neurons): one
- Connectivity: two afferent, one efferent.
- Activation rule: threshold.

PDP, step 1: perceptron





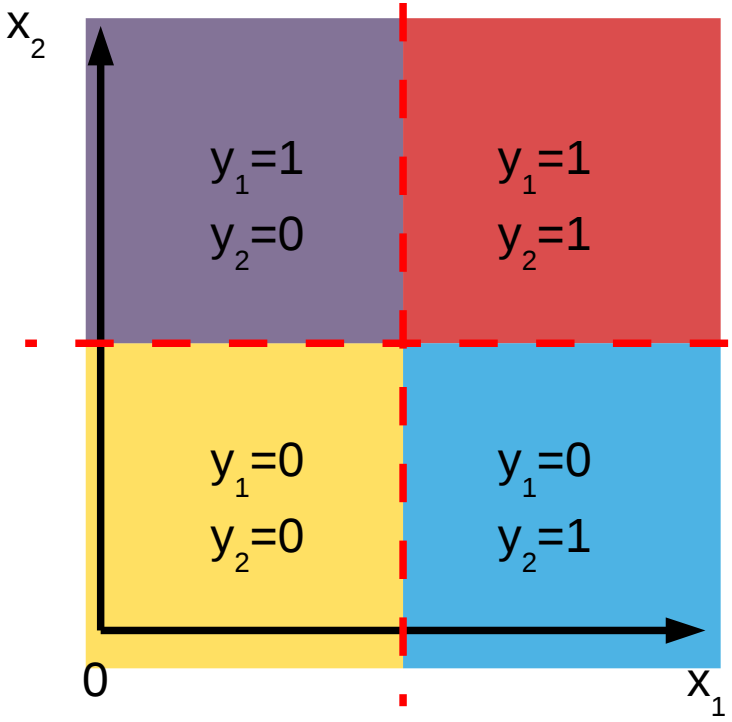
Output: 2 categories

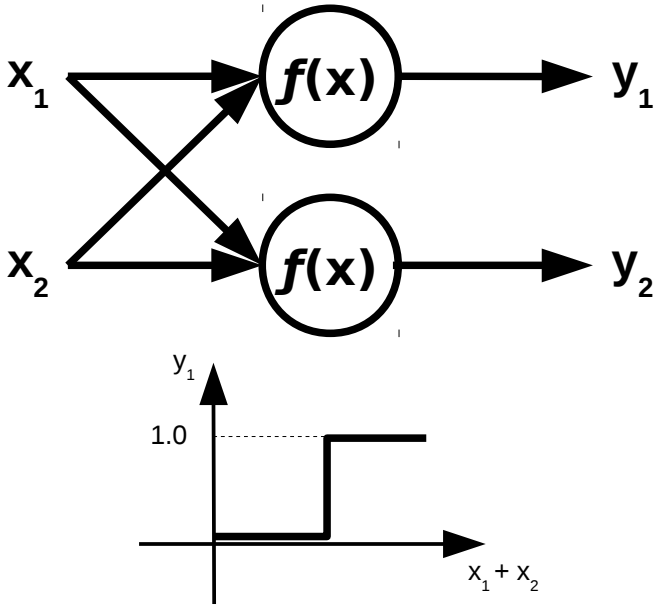
- $y_1=0$
- $y_1=1$
- $y_2= y_1$

Perceptron

- Input: two dimension environment.
- Processing units (neurons): one
- Connectivity: two afferent, two efferent.
- Activation rule: threshold.

PDP, step 1: perceptron





Output: 4 categories

- $y_1=0, y_2=0,$
- $y_1=1, y_2=0,$
- $y_1=0, y_2=1,$
- $y_1=1, y_2=1,$

Perceptron

- Input: two dimension environment.
- Processing units (neurons): two
- Connectivity: four afferent, two efferent.
- Activation rule: threshold.

PDP, step 2: multilayer

PDP, elements:

– Environment encoded into numerical input.

– Pattern of afferent connectivity, propagation rules.

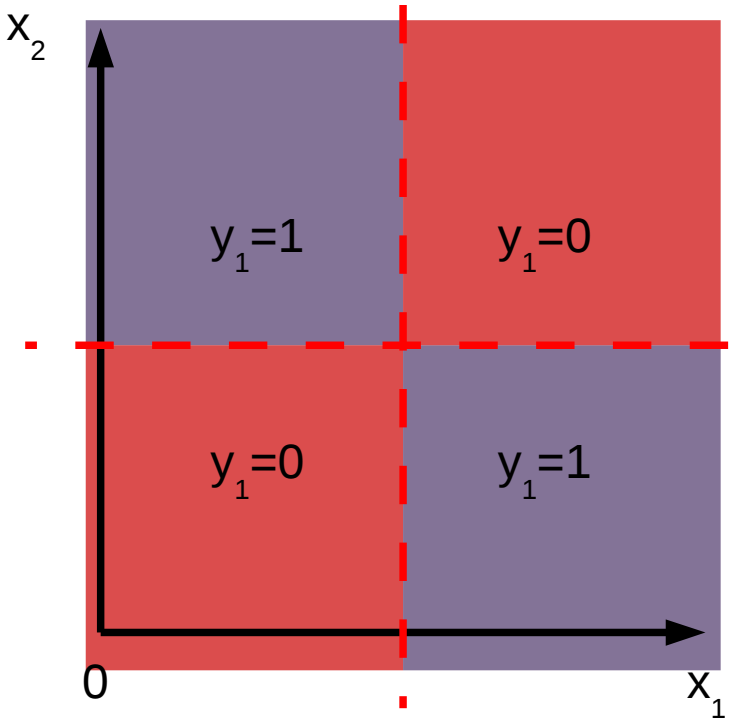
– Internal computation (transfer function) resulting in activation states.

– Sequential processing: the output of one layer becomes the input for another downstream layer.

– Decoded output.

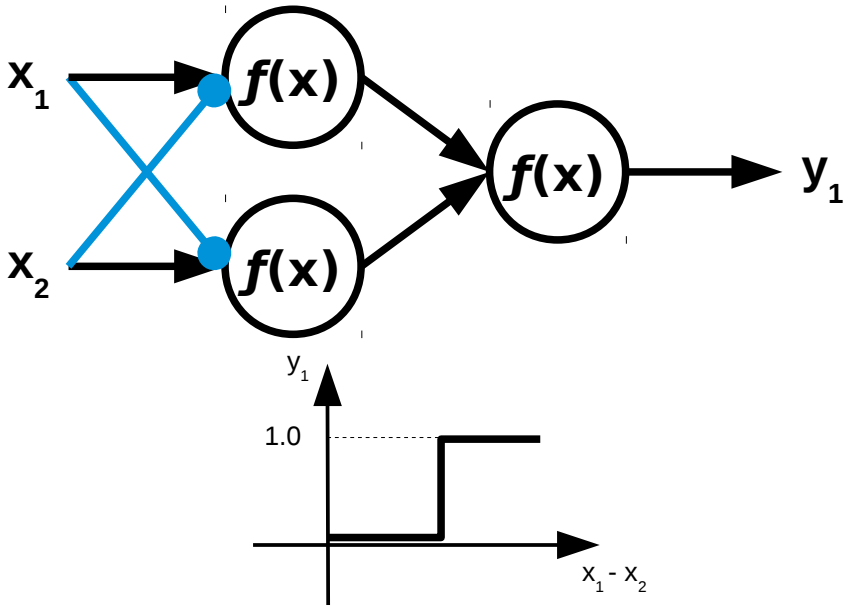
Image from: David Poole and Alan Mackworth, 2010

PDP, step 2: multilayer



Output: 2 categories

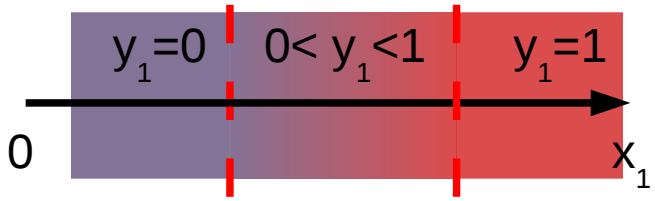
- $y_1=0$
- $y_1=1$

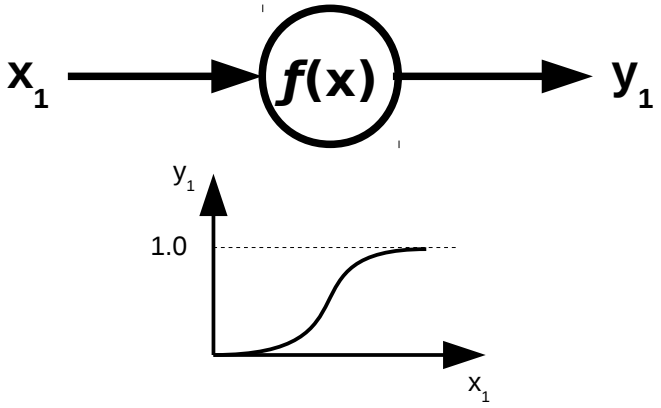


Feed-forward multilayer network

- Input: two dimension environment.
- Processing units (neurons): three.
- Connectivity: four afferent, two efferent (Layer 1), two afferent, one efferent (Layer 2).
- Activation rule: threshold.

PDP, step 3: time component





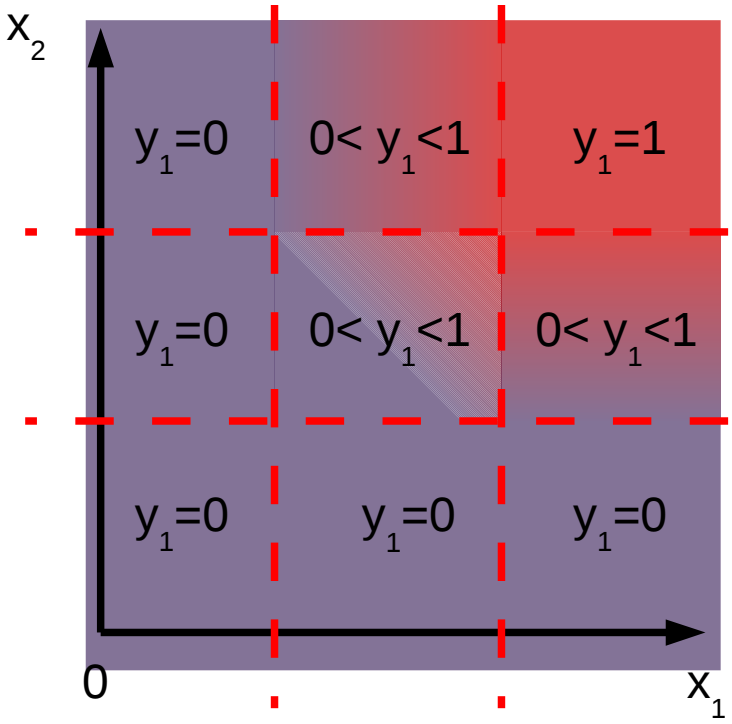
Output: encodes strength and duration of signal

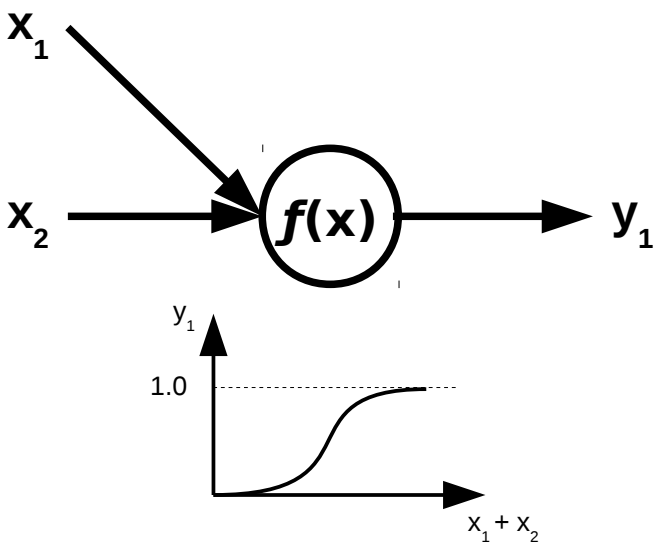
- $y_1=0$
- $0< y_1<1$
- $y_1=1$

Perceptron

- Input: one dimension environment.
- Processing units (neurons): one
- Connectivity: one afferent, one efferent.
- Activation rule: **leaky integrator**.

PDP, step 3: time component





Output: encodes strength and duration of signal

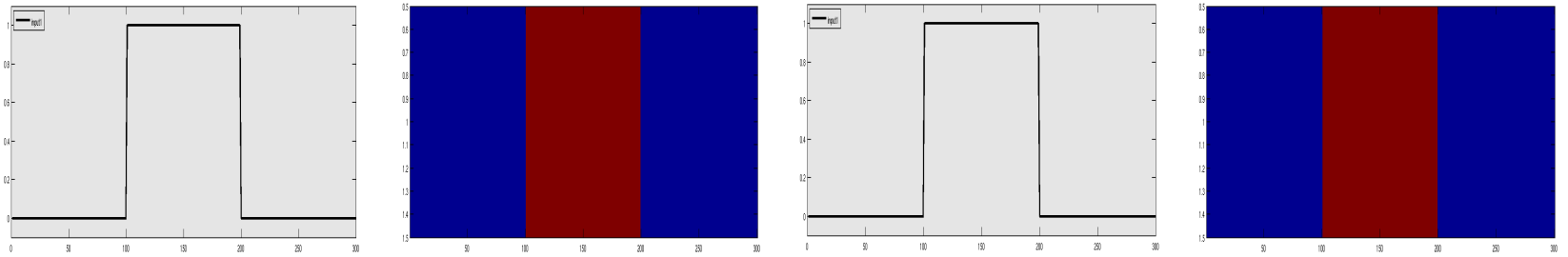
- $y_1=0$
- $0< y_1<1$
- $y_1=1$

Perceptron

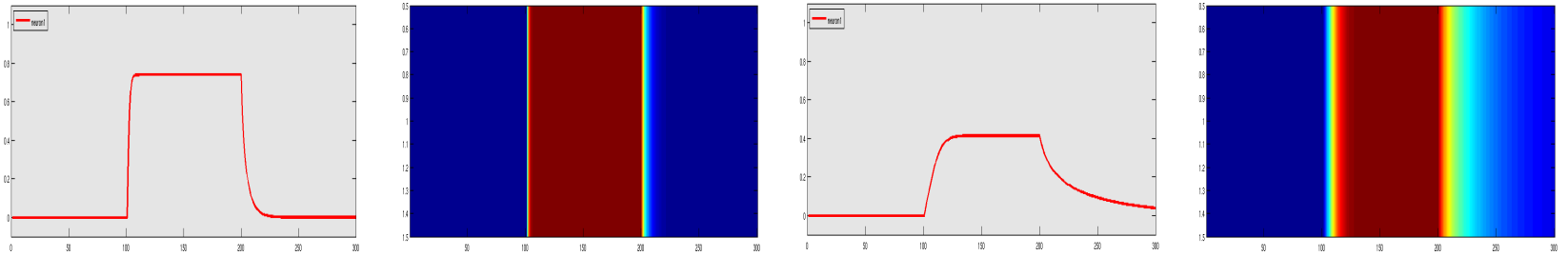
- Input: two dimension environment.
- Processing units (neurons): one
- Connectivity: two afferent, one efferent.
- Activation rule: **leaky integrator**.

PDP, step 3: time component

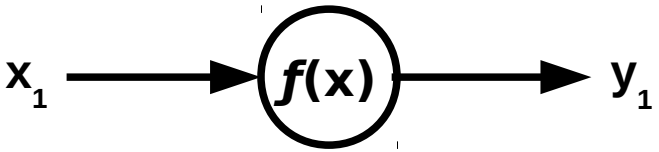
Input: x_1



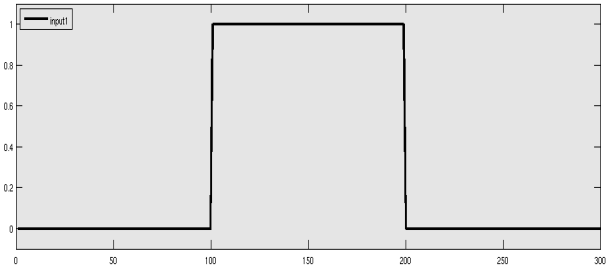
Output: y_1

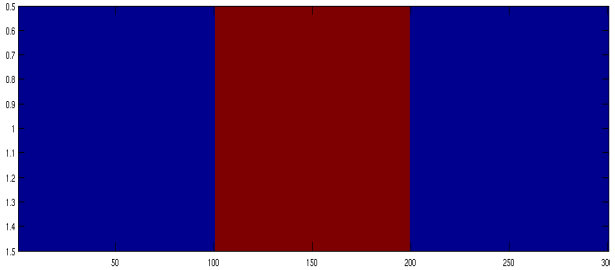


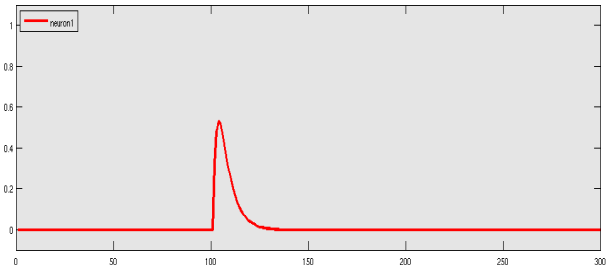
Quick leaky integrator **Slow leaky integrator**

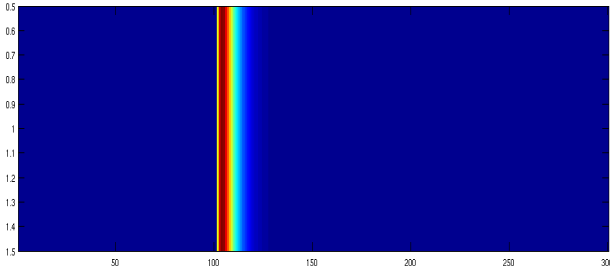


PDP, step 4: lateral and feedback connectivity

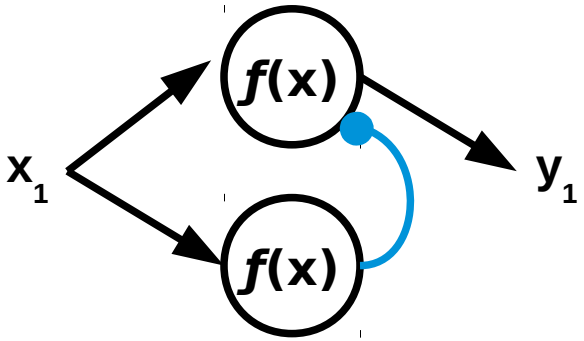




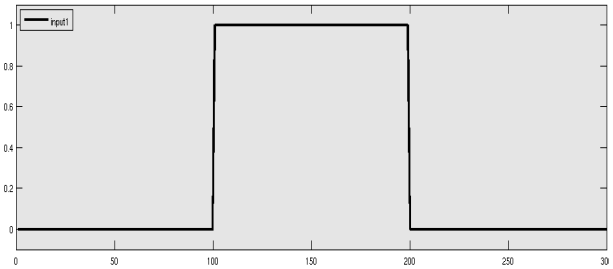


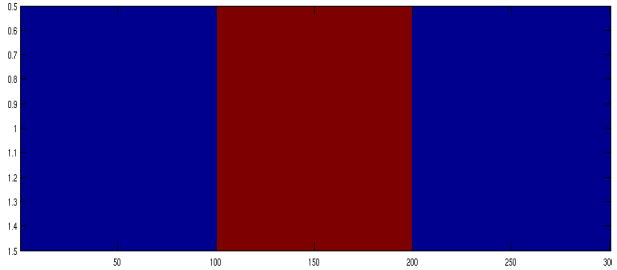


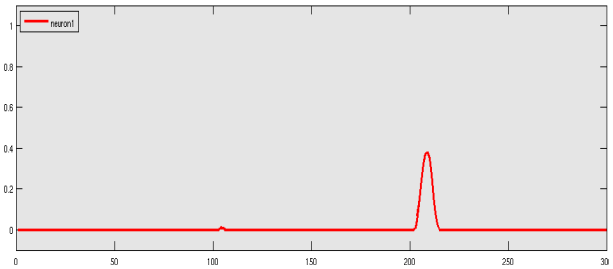
Function realised:
Onset detector

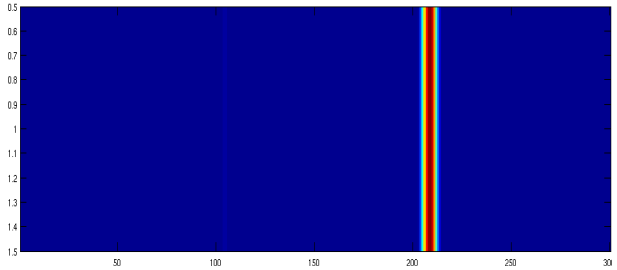


PDP, step 4: lateral and feedback connectivity

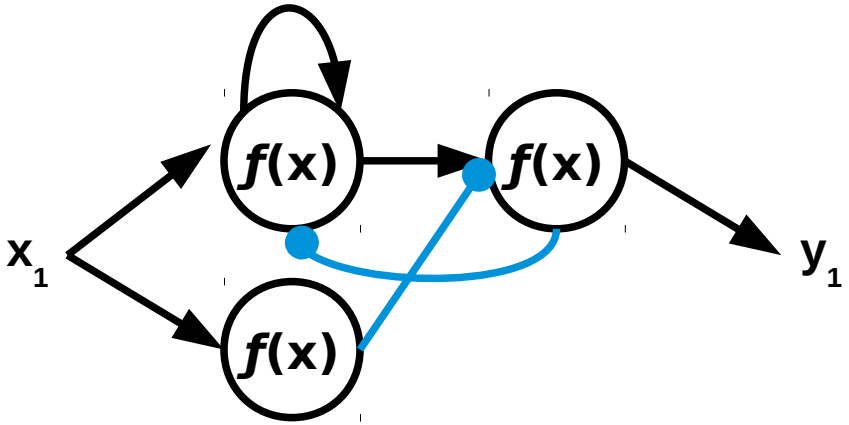








Function realised:
Offset detector



PDP, step 5: stability, attractors and memory

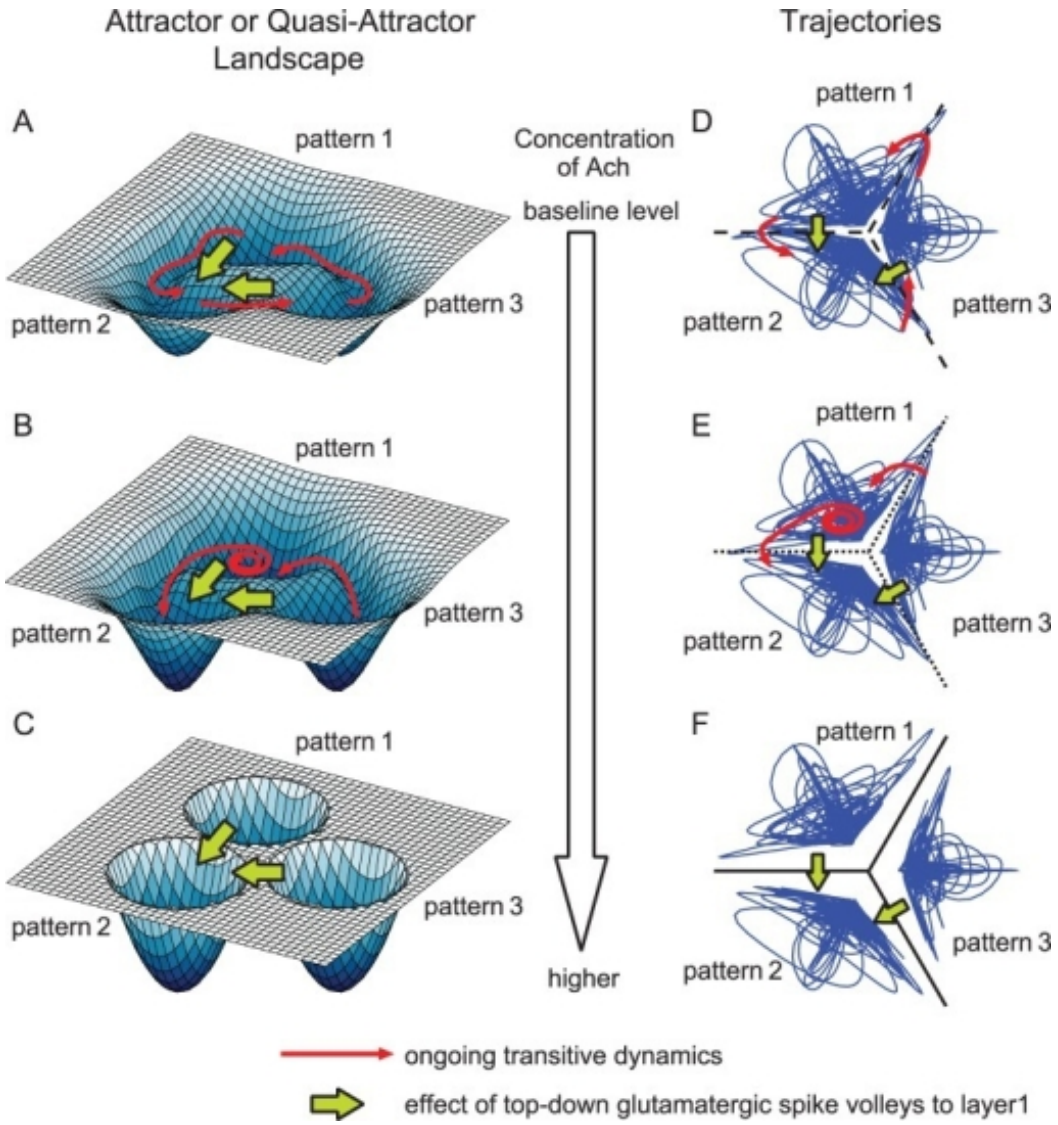
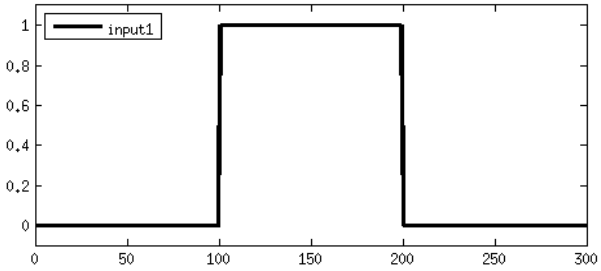
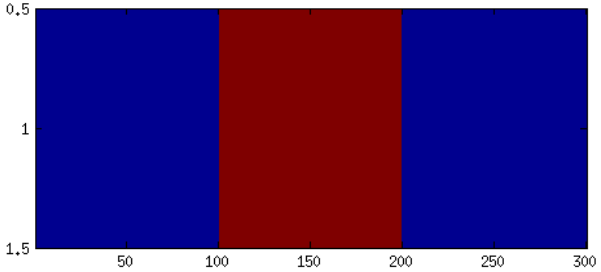
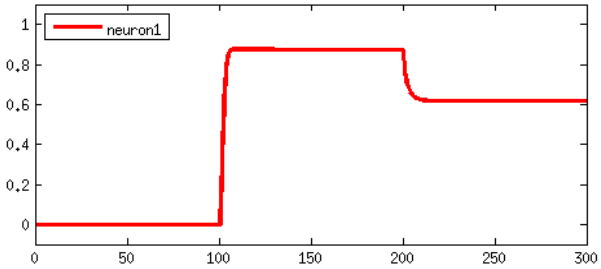


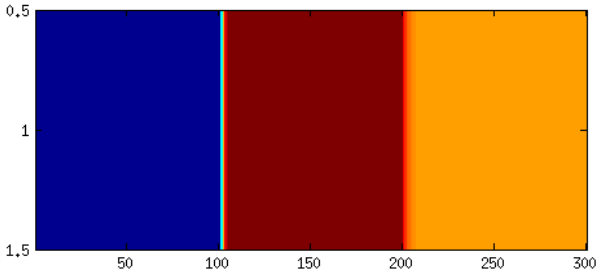
Image from: Kanamaru T, Fujii H, Aihara K - PLoS ONE (2013)

PDP, step 5: stability, attractors and memory

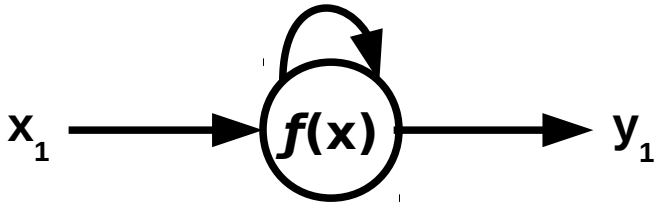




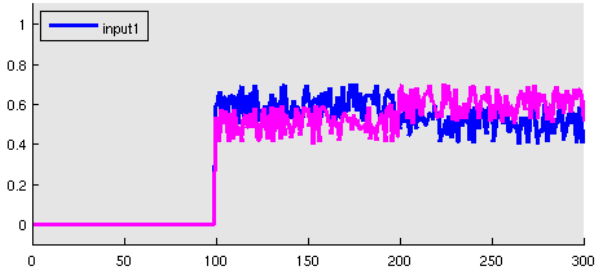


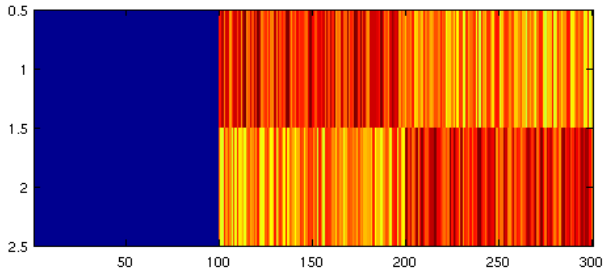


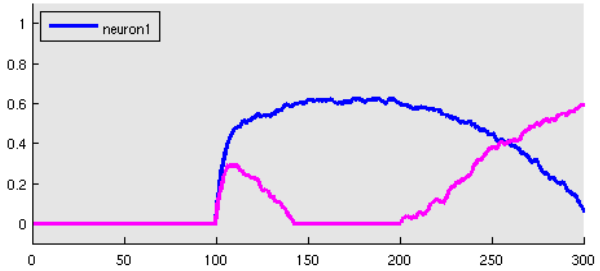
Function realised:
Maintenance

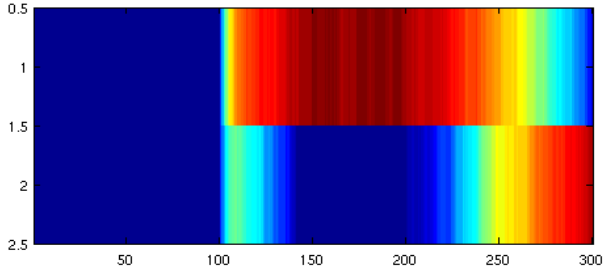


PDP, step 5: stability, attractors and memory

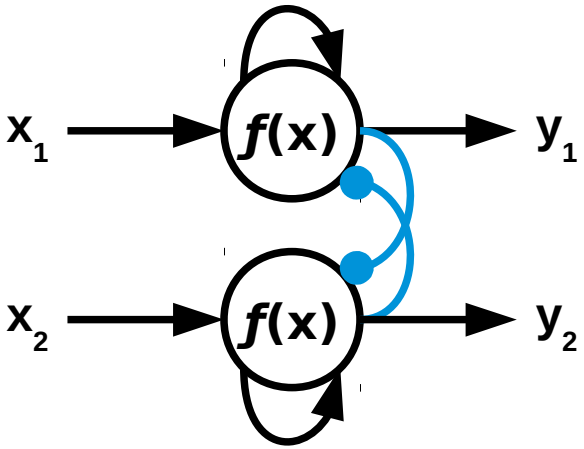




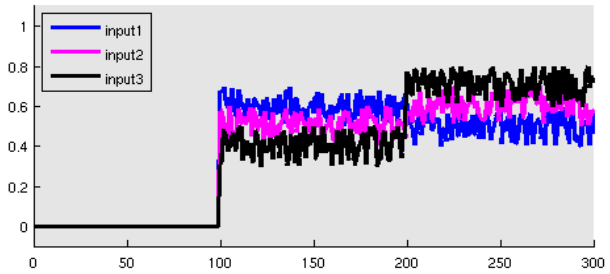


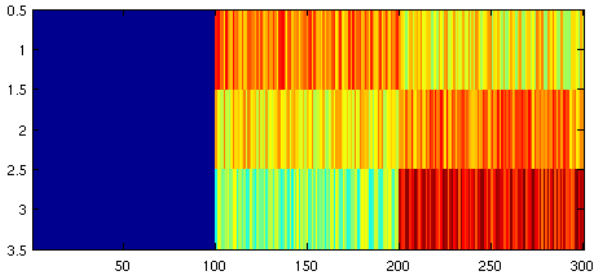


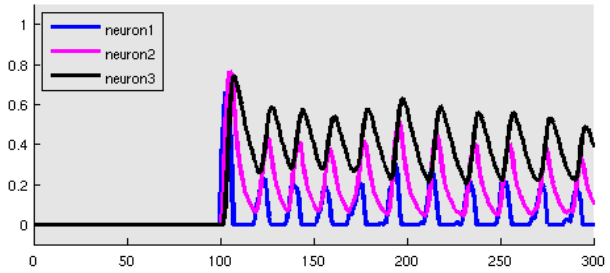
Function realised:
Attractor state network

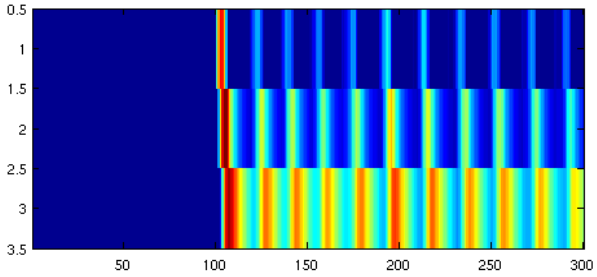


PDP, step 6: pattern generators

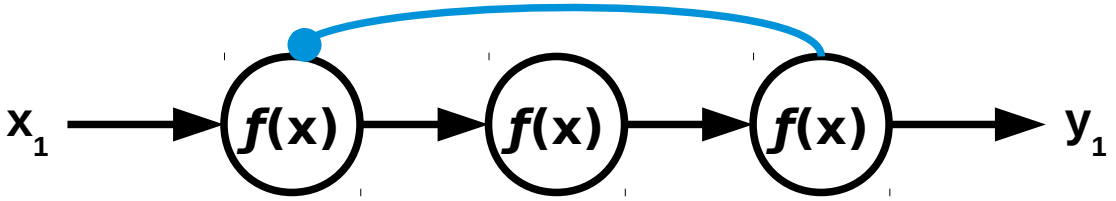


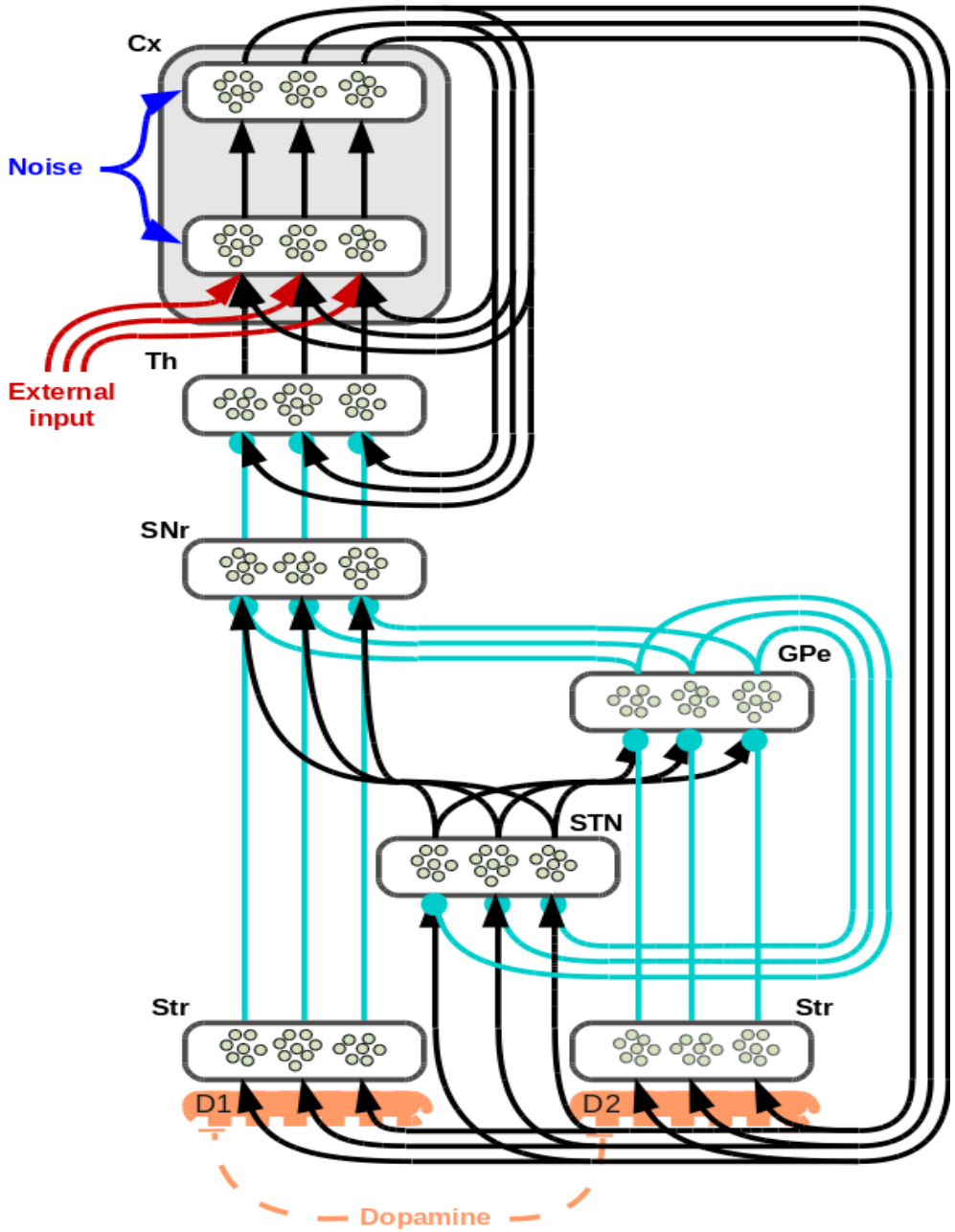






Function realised:
Pattern generator





What kind of dynamics can we expect from this?

From structures to functions

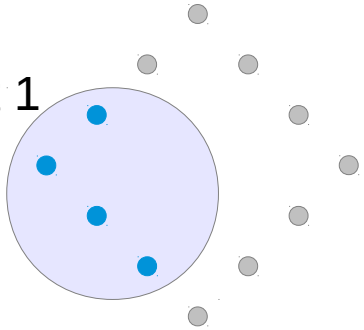
1 – Static Networks

2 – Dynamic Networks

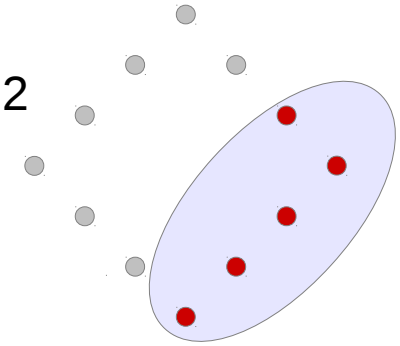
PDP, step 7: learning

PDP, step 7: (unsupervised) learning

Sensory input 1
(e.g. orange)



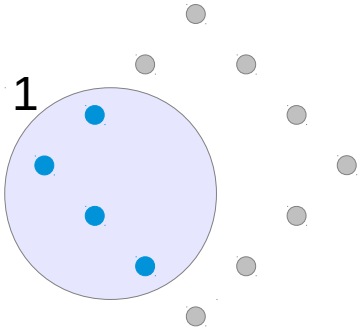
Sensory input 2
(e.g. kiwi)



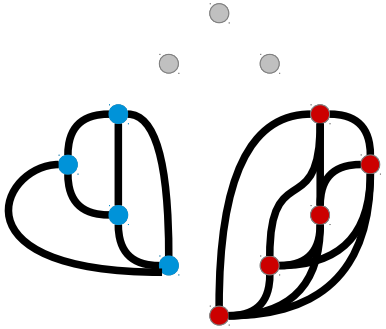
Different sensory stimuli
activate different units

PDP, step 7: (unsupervised) learning

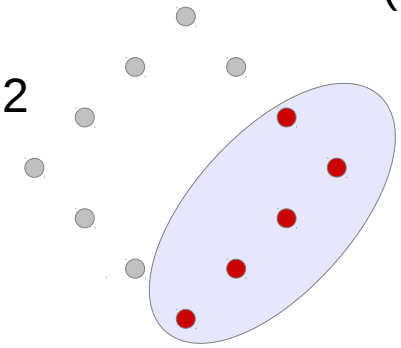
Sensory input 1
(e.g. orange)



TIME
(several exposures)



Sensory input 2
(e.g. kiwi)



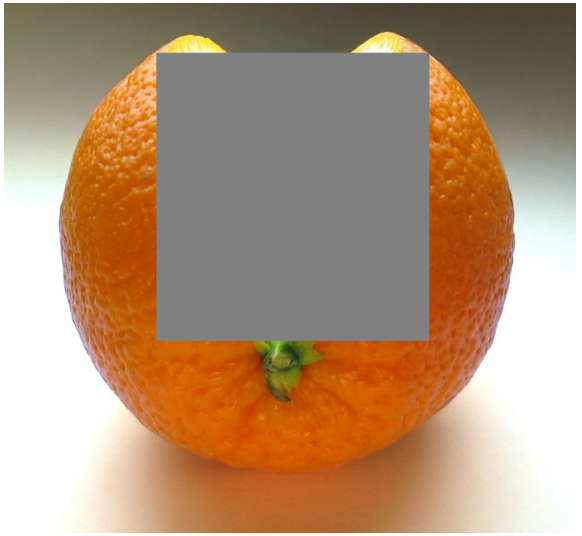
Different sensory stimuli
activate different units

Neurons firing together,
wire together, creating
clusters.
First Hebbian rule:

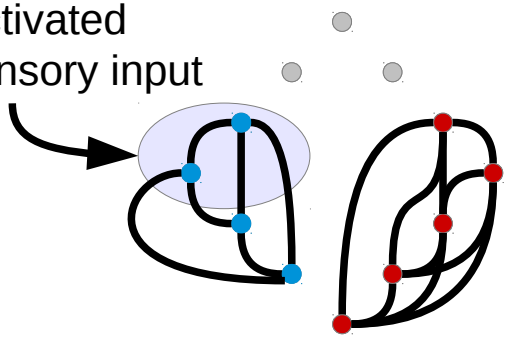
$$\Delta w_{ij} = (y_i)(y_j)$$

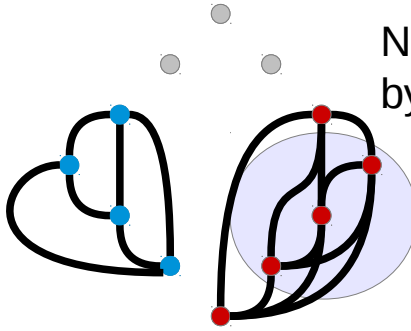
PDP, step 7: (unsupervised) learning

Completion: after learning, the system recognises incomplete features to associate the sensory stimulus to a known category.



Nodes activated
by the sensory input

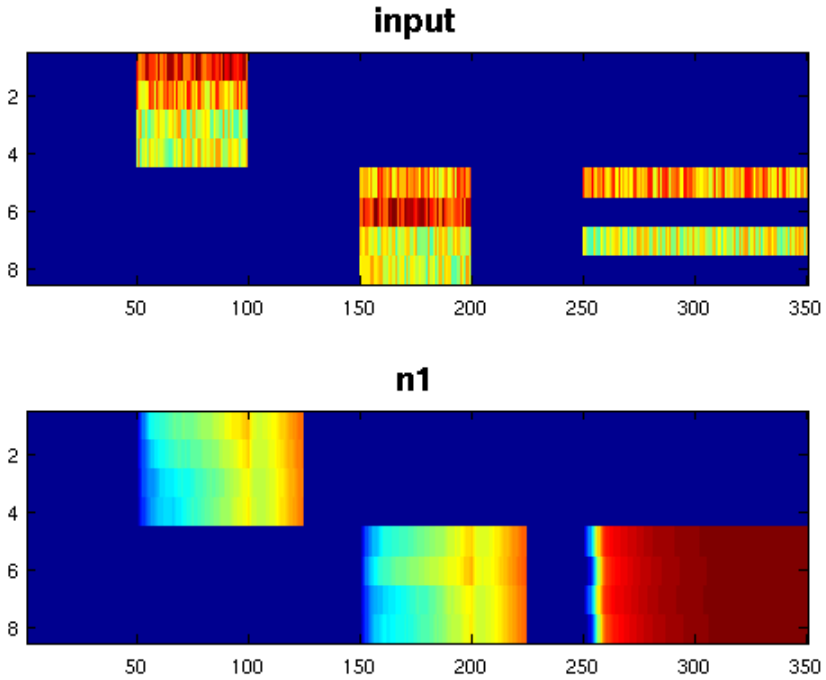


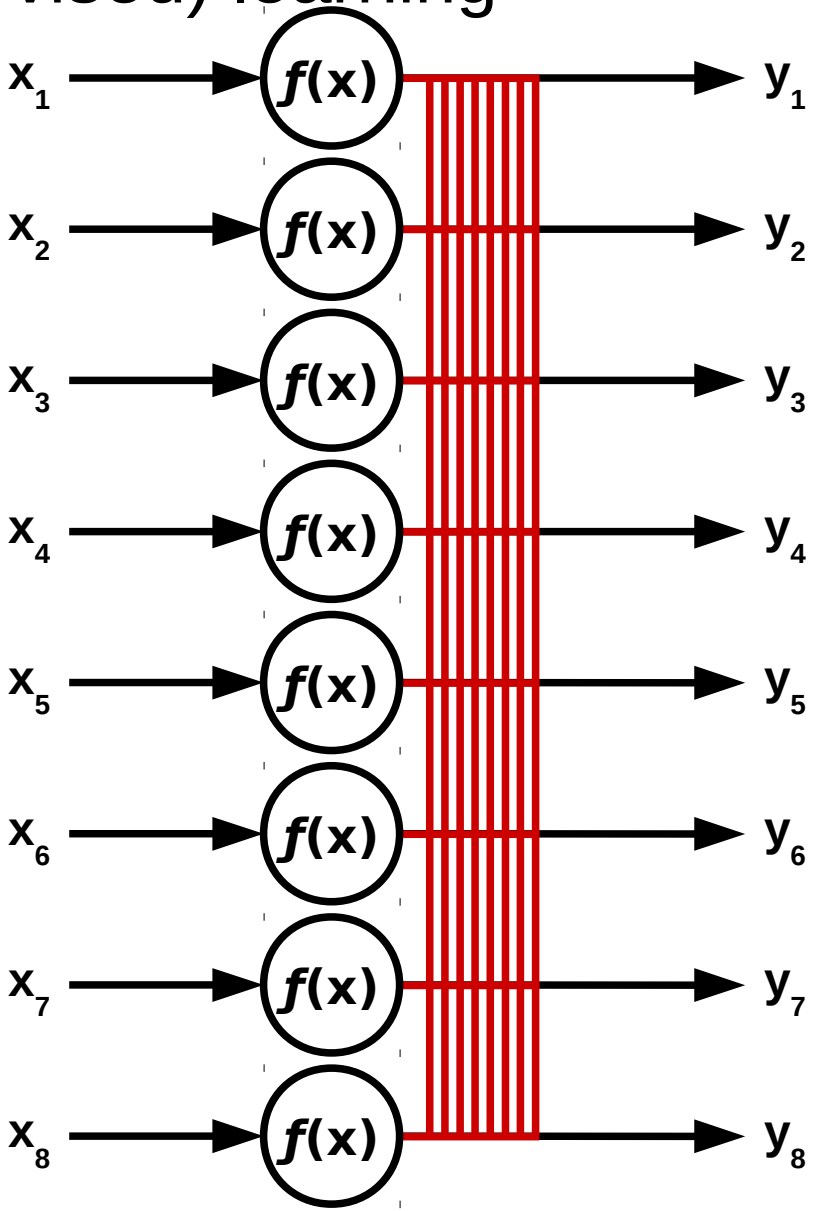


Nodes activated
by the sensory input

PDP, step 7: (unsupervised) learning

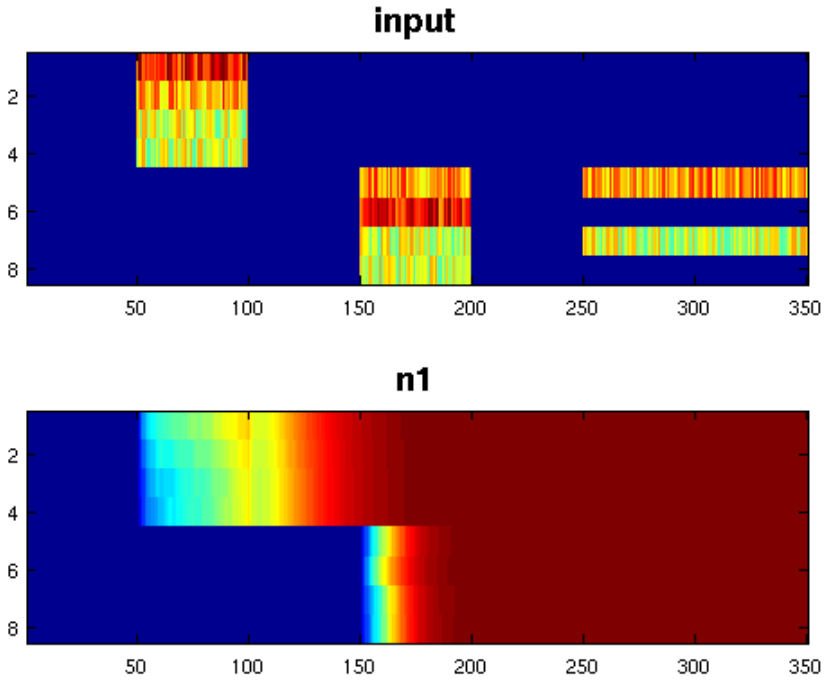
Target neural activity:



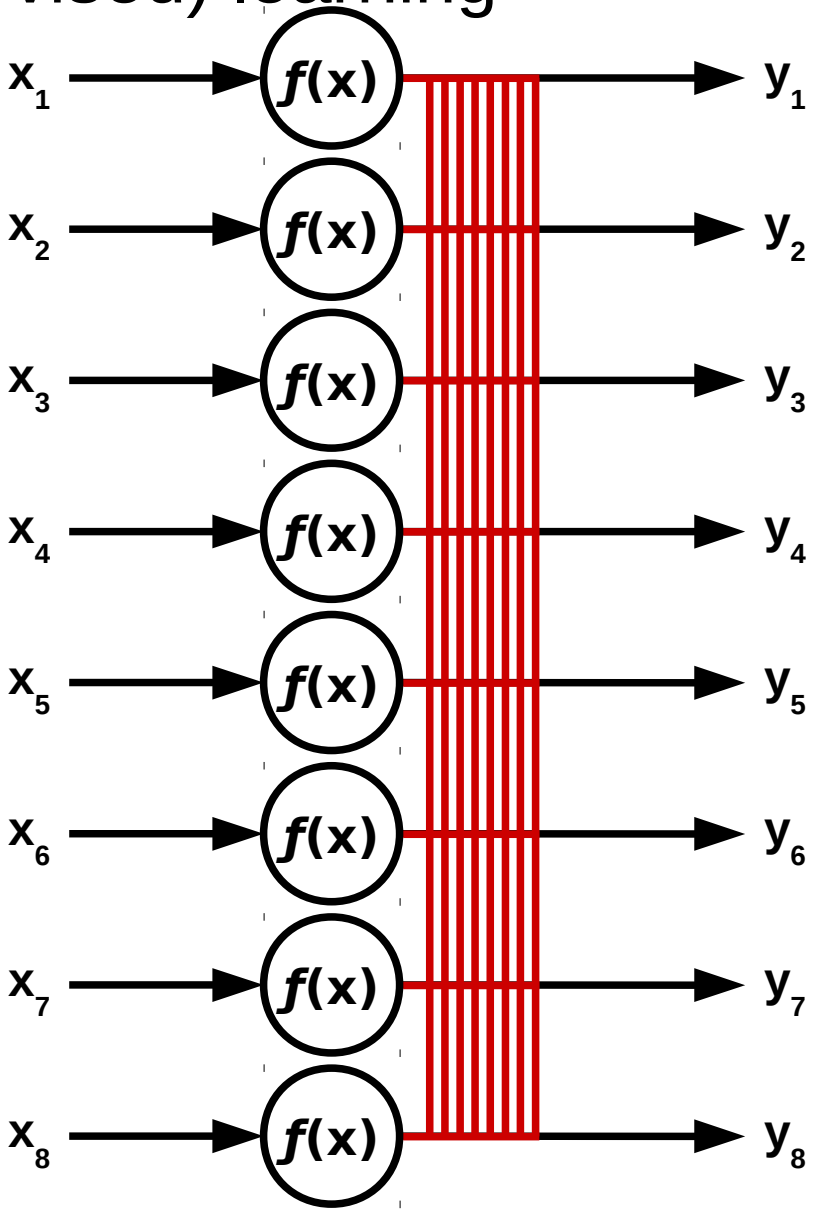


PDP, step 7: (unsupervised) learning

Actual neural activity:

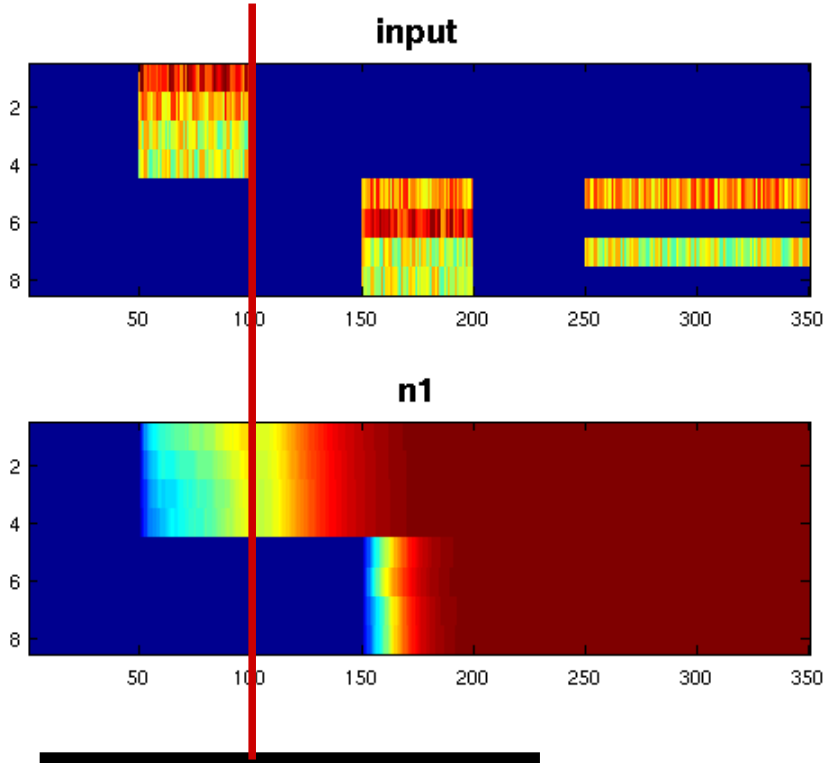


Completely interconnected neural network: why?

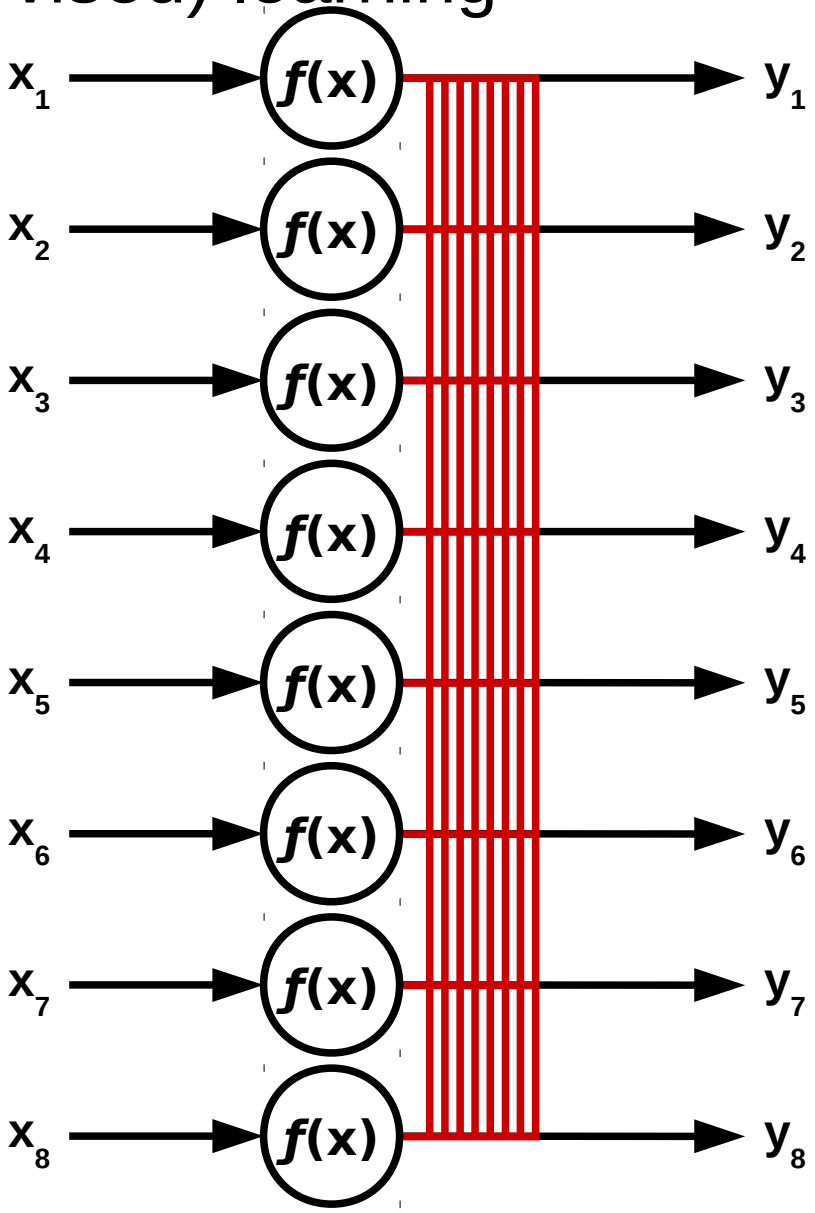


PDP, step 7: (unsupervised) learning

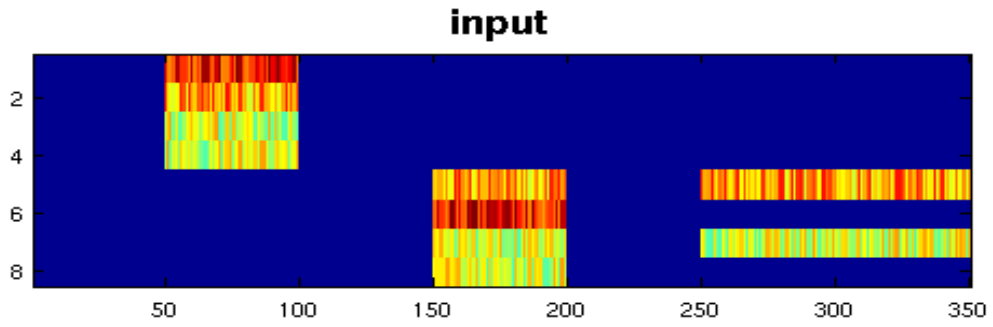
Actual neural activity:



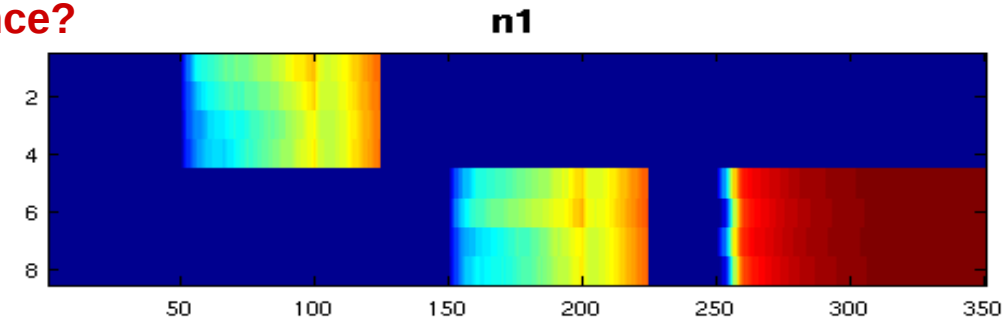
Undesired maintenance and memory effect!



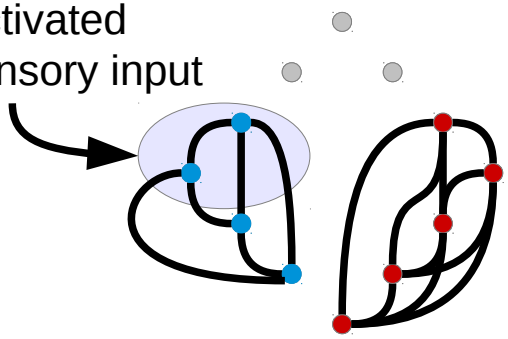
PDP, step 7: (unsupervised) learning

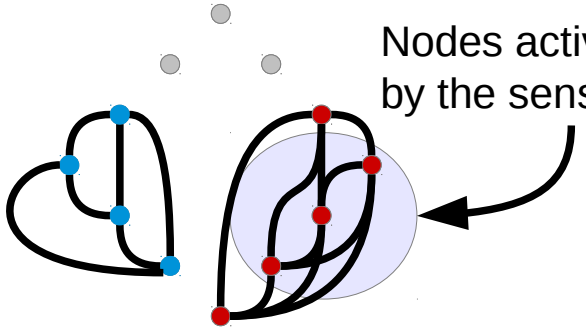


How do we suppress undesired maintenance?



Nodes activated by the sensory input

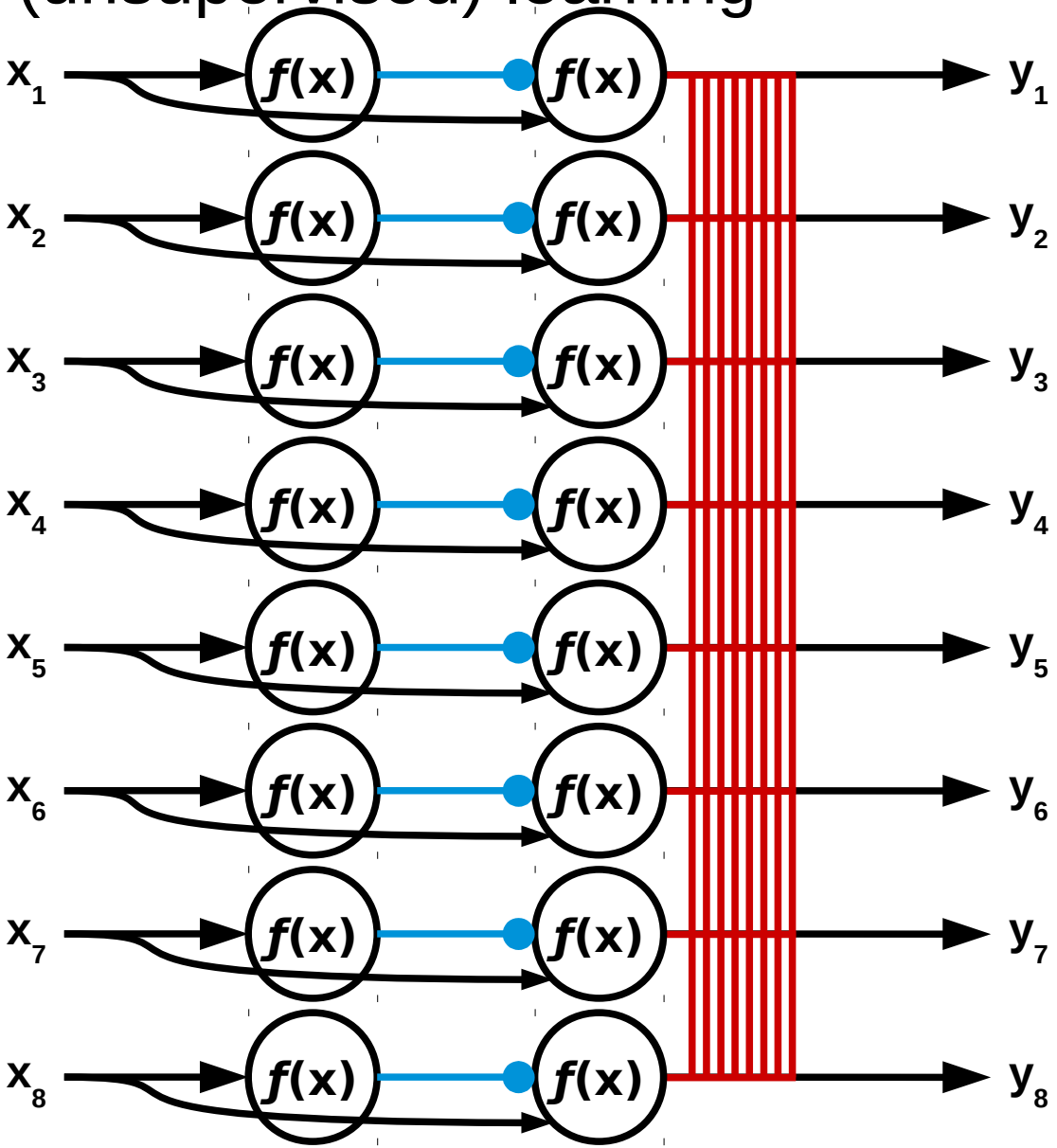




Nodes activated by the sensory input

PDP, step 7: (unsupervised) learning

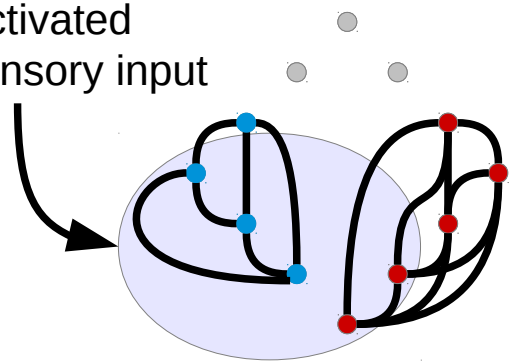
How do we suppress undesired maintenance?
A possible solution grounded on inhibitions.



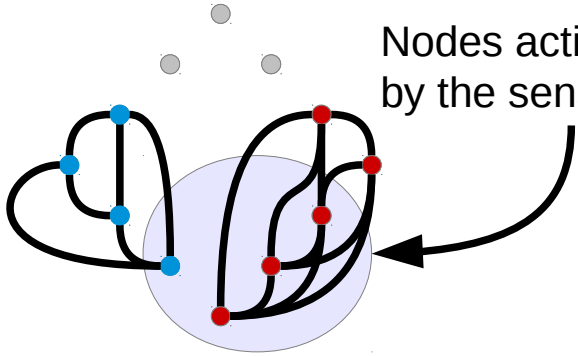
PDP, step 7: (unsupervised) learning

Interference: after learning, the system must suppress conflicting stimuli to associate the sensory stimulus to a known category.

Nodes activated
by the sensory input

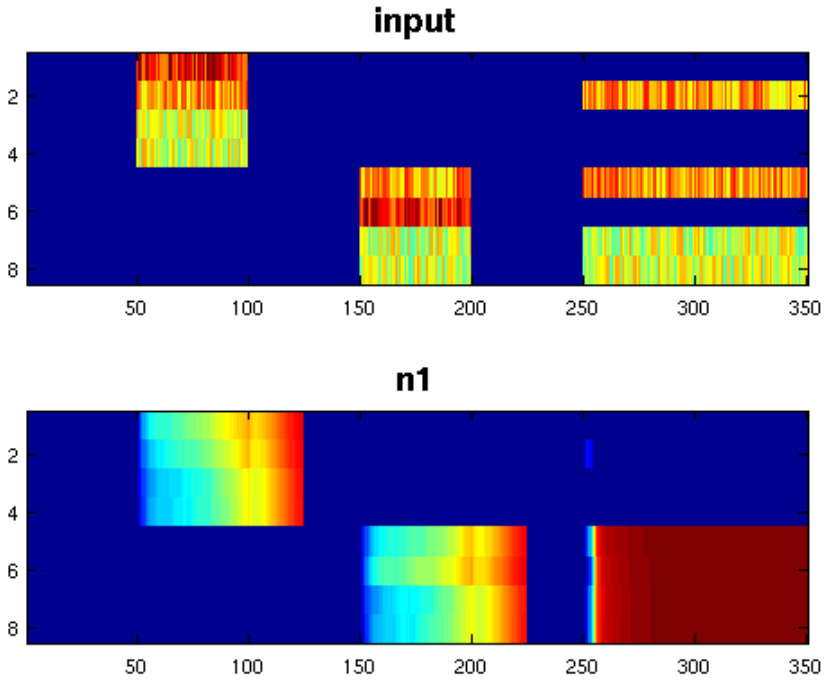


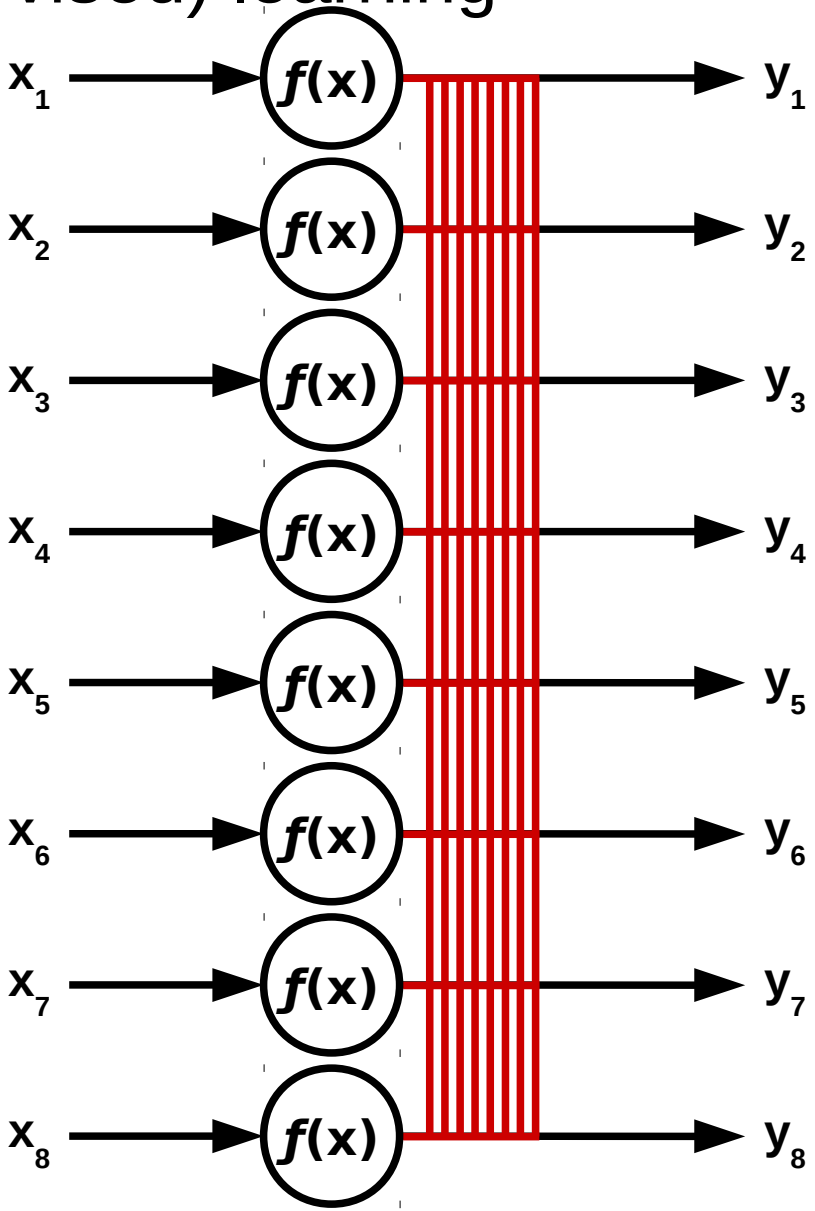
Nodes activated
by the sensory input



PDP, step 7: (unsupervised) learning

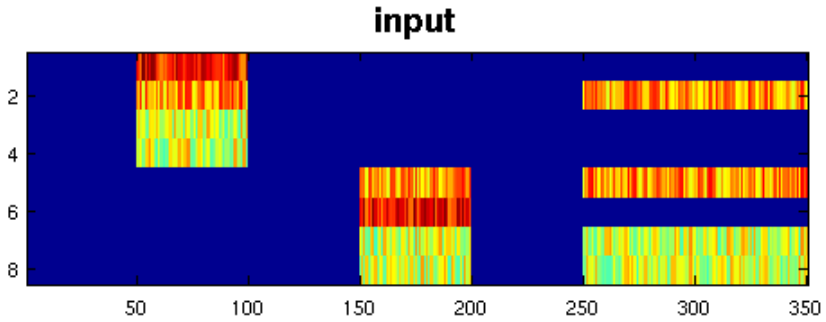
Target neural activity:

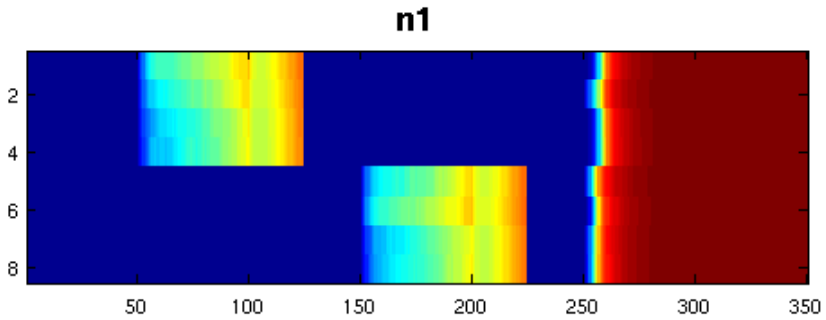




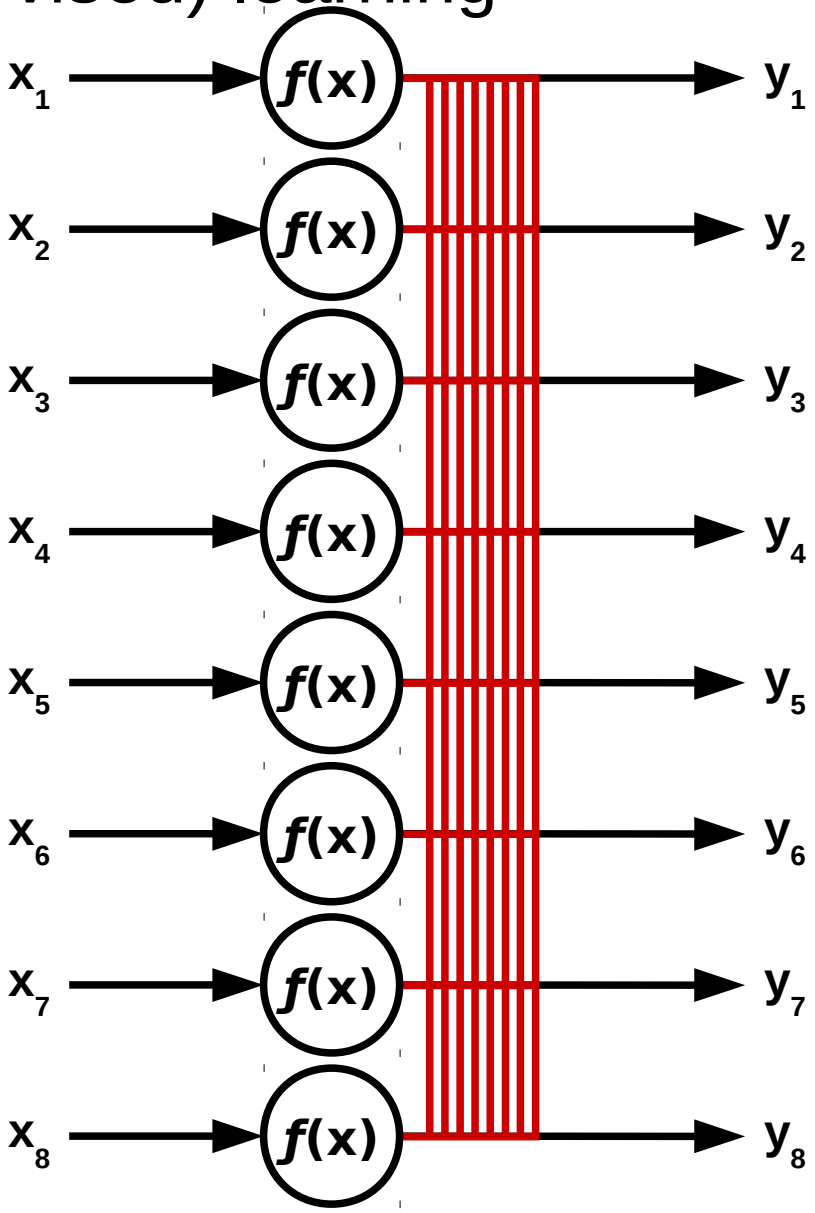
PDP, step 7: (unsupervised) learning

Actual neural activity:



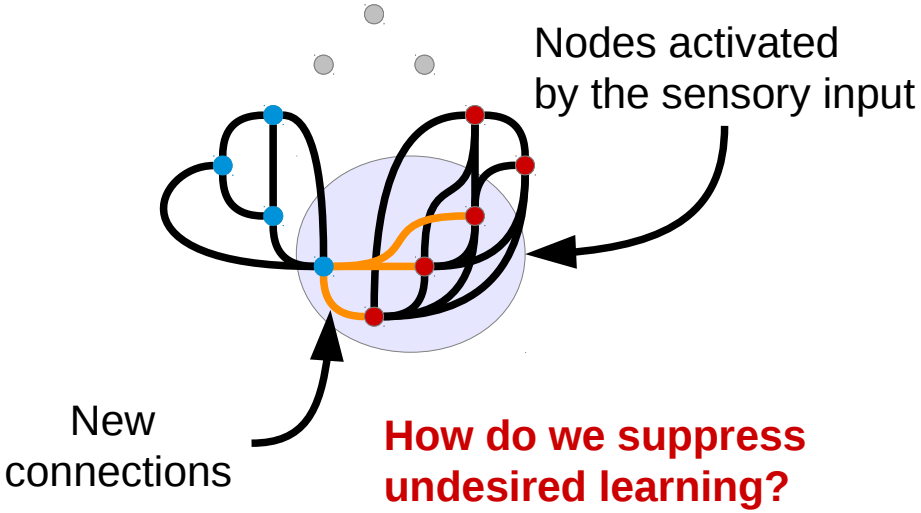


Undesired learning: Oranges and kiwis now belong (again!) to the same category



PDP, step 7: (unsupervised) learning

Interference: after learning, the system must suppress conflicting stimuli to associate the sensory stimulus to a known category.



How do we suppress undesired learning?

PDP, step 7: (unsupervised) learning

The three Hebbian learning rules in auto associative networks:

1 – Hebb: $\Delta w_{ij} = (y_i)(y_j)$

2 – Post-synaptic $\Delta w_{ij} = (y_i)(2y_j - 1)$

3 – Pre – synaptic $\Delta w_{ij} = (2y_i - 1)(y_j)$

Generally considered valid for the paleocortex (e.g. hippocampus)

PDP, step 7: (unsupervised) learning

The four Hebbian learning rules in auto associative networks:

1 – Hebb: $\Delta w_{ij} = (y_i)(y_j)$

2 – Post-synaptic $\Delta w_{ij} = (y_i)(2y_j - 1)$

3 – Pre – synaptic $\Delta w_{ij} = (2y_i - 1)(y_j)$

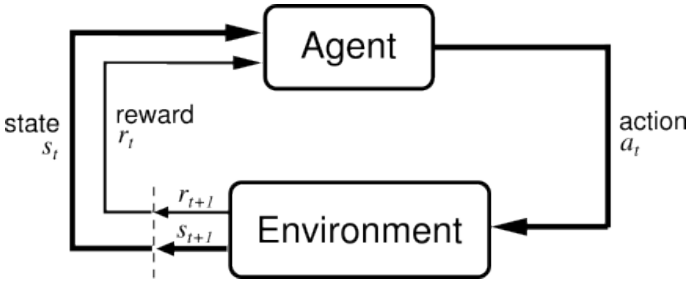
4 – Hopfield $\Delta w_{ij} = (2y_i - 1)(2y_j - 1)$

Hopfield rule increases synaptic strength if pre- and post- synaptic units are inactive together.

PDP, step 7: (supervised) learning

The role of dopamine:

- reward signal (shift is caused by CS-R association).
- prediction error (shift is caused by prediction).
- novelty signal (shift is caused R becoming expected and CS unexpected).
- signal of intrinsic/extrinsic motivation (shift is caused by CS-R association).



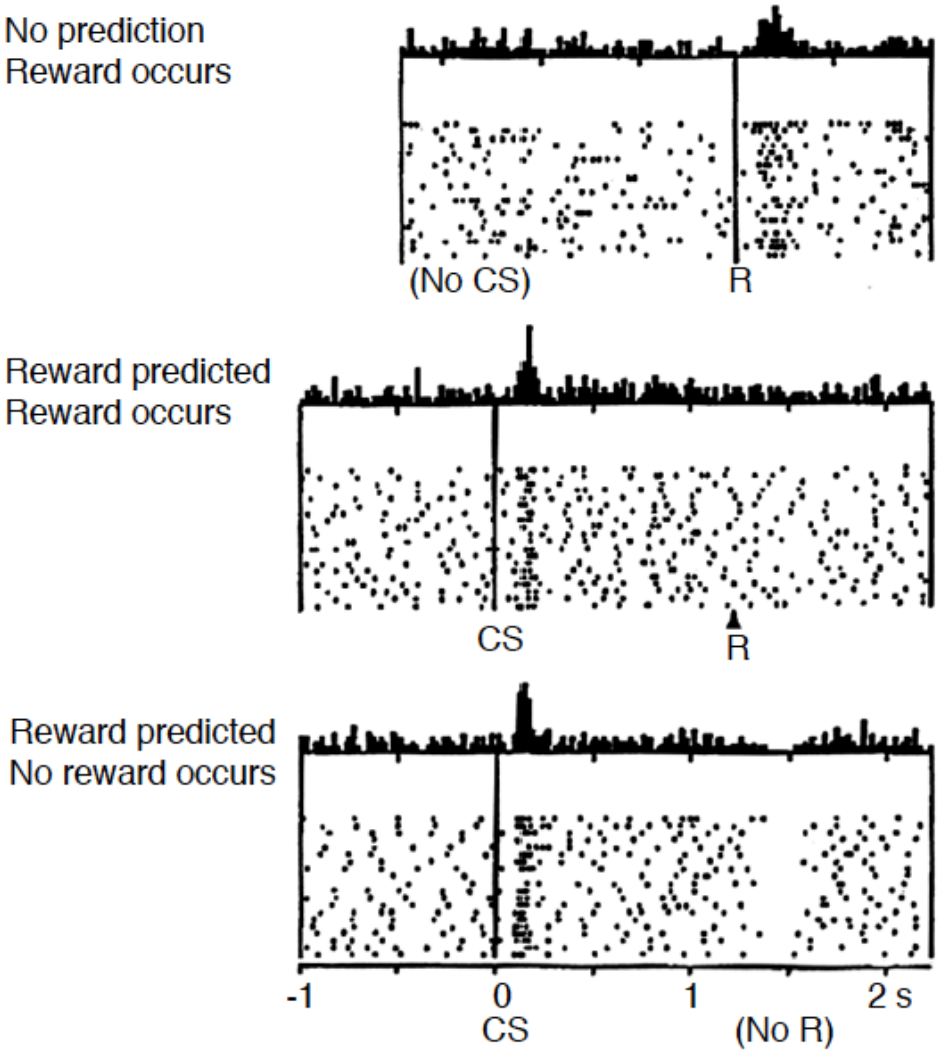
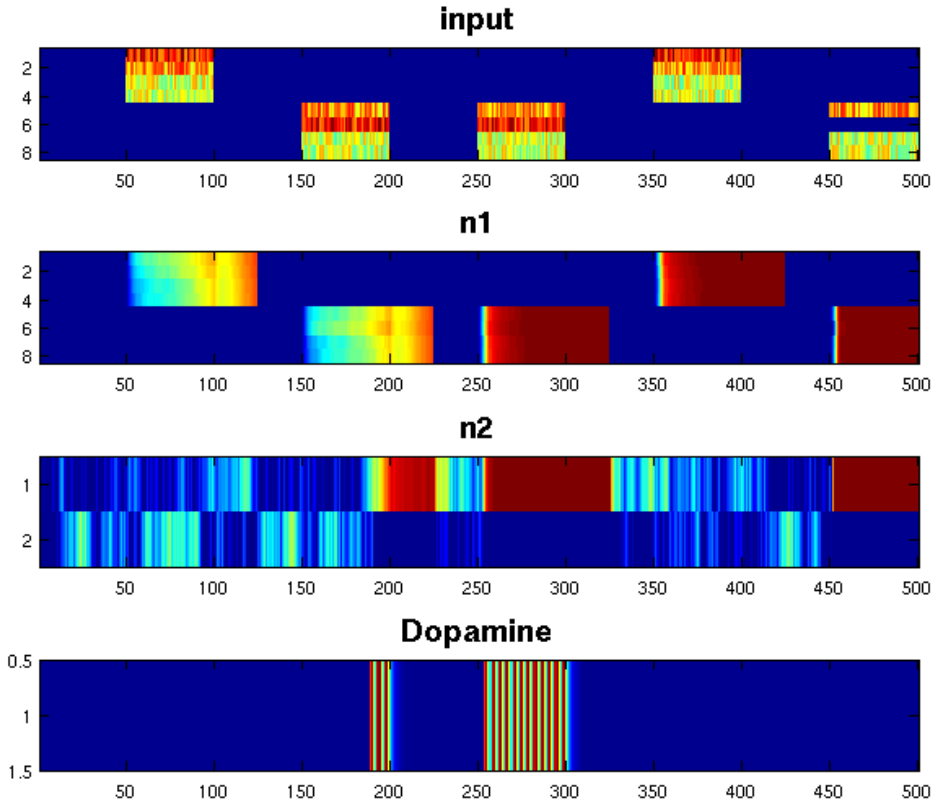
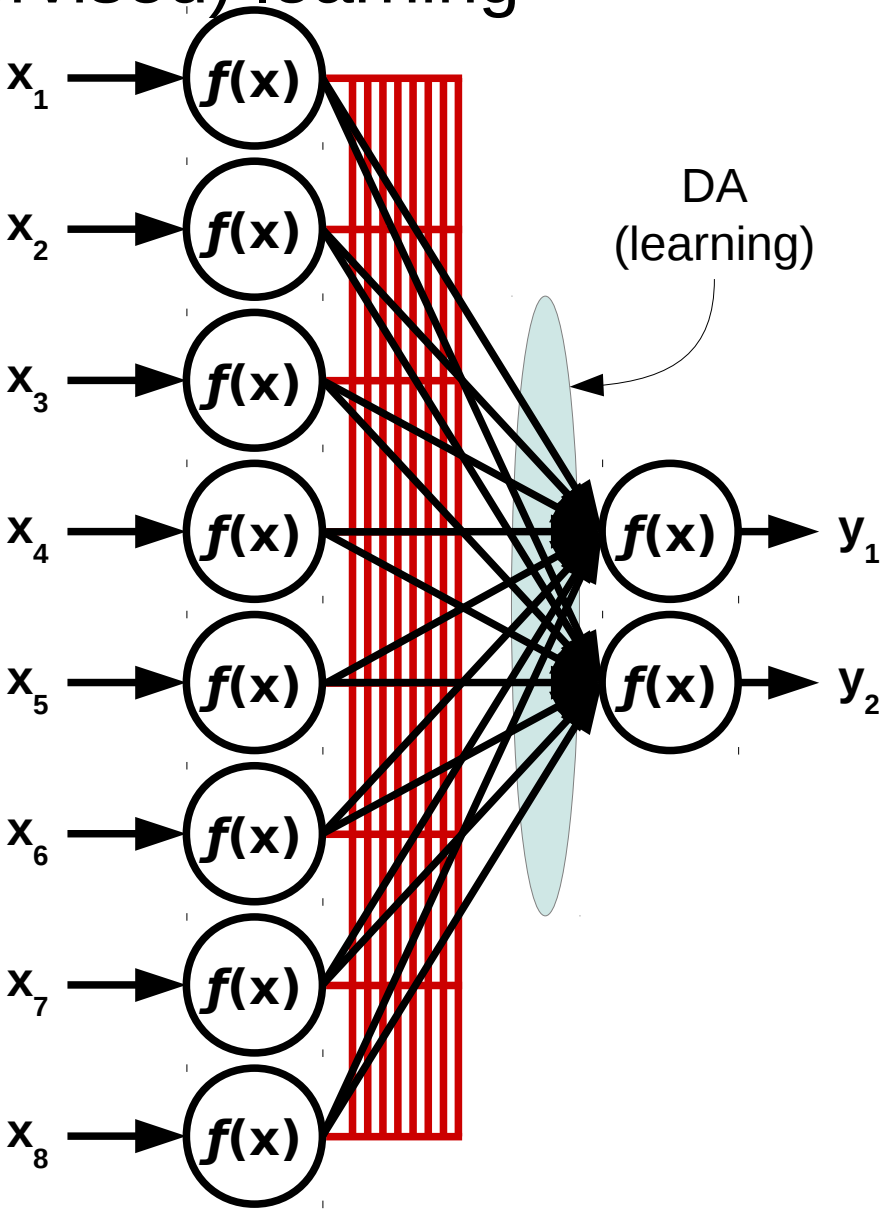


Image from: Schultz W., Dayan P., Montague R. R. A (1997) Science.

PDP, step 7: (supervised) learning





PDP, step 7: (supervised) learning

Dopamine dependent Hebbian learning:

1 – Hebb: $\Delta w_{ij} = (y_i)(y_j)(da - th)$

2 – Post-synaptic $\Delta w_{ij} = (y_i)(2y_j - 1)(da - th)$

3 – Pre – synaptic $\Delta w_{ij} = (2y_i - 1)(y_j)(da - th)$

PDP, step 8: neuromodulation

Standard leaky integrator

$$\tau_g \dot{u}_j = -u_j + b_j + \sum w_{ji}\, y_i$$

$$y_j = [\tanh(u_j - \theta)]^+$$

Modified leaky integrator
(d=dopamine release)

$$\tau_g \dot{u}_j = -u_j + b_j + (\varepsilon + \lambda d) \sum w_{ij}\, y_i$$

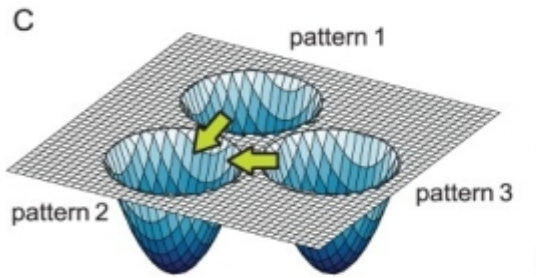
$$y_j = [\tanh(u_j - \theta)]^+$$

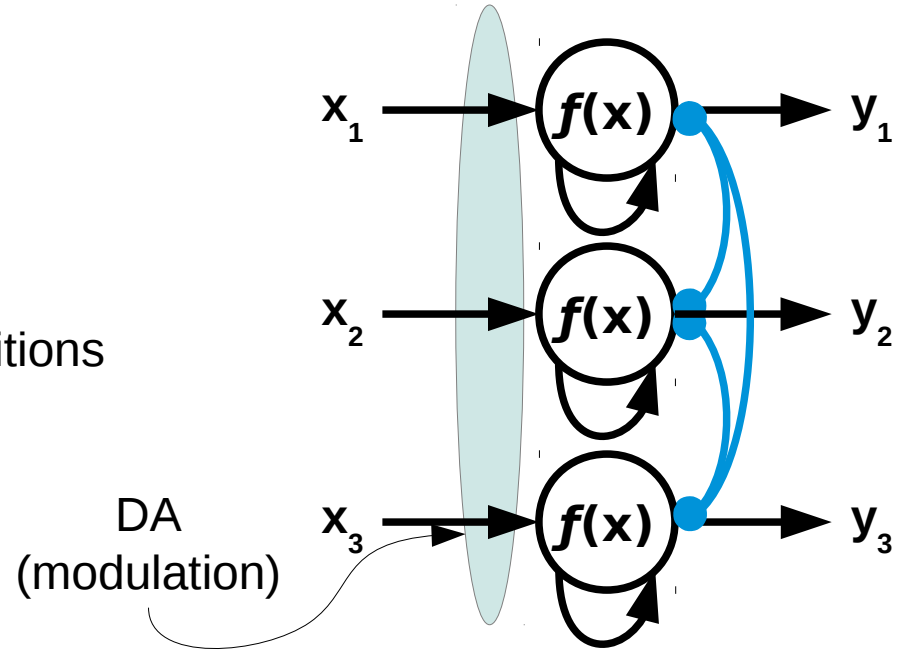
Case study: selection in the basal ganglia

Functions implemented:
- sustained selection (attractors)
- noise suppression
- modulated by dopamine
(e.g. Parkinson)

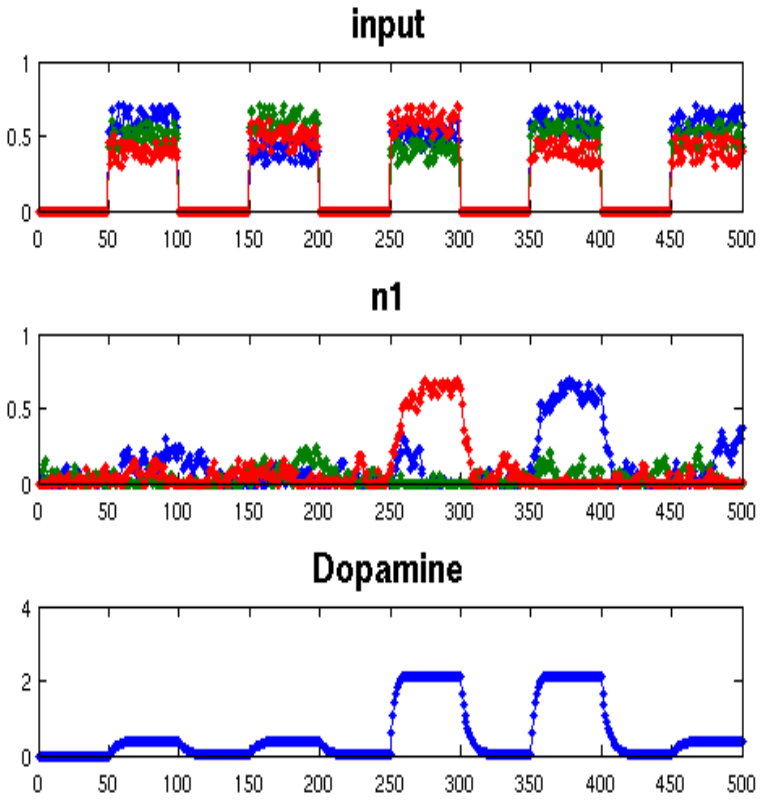
Structural features:
- dominance of GABA

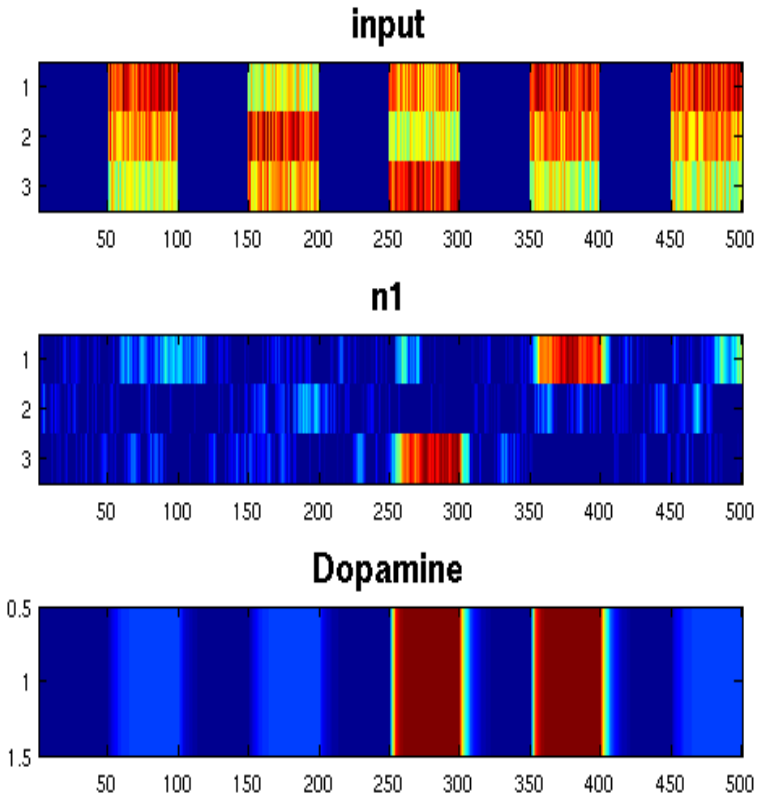
Solution:
competition via lateral inhibitions





Case study: selection in the basal ganglia





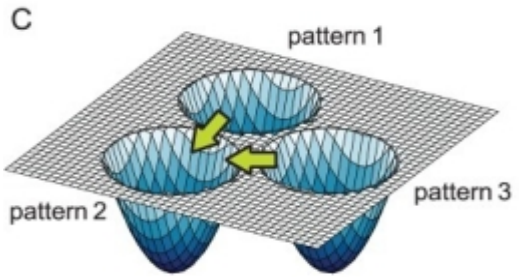
Case study: selection in the basal ganglia

Functions implemented:
- sustained selection (attractors)
- noise suppression
- modulated by dopamine
(e.g. Parkinson)

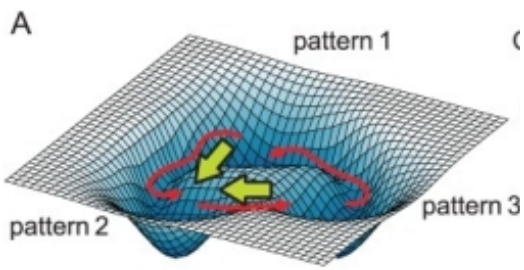
Structural features:
- dominance of GABA

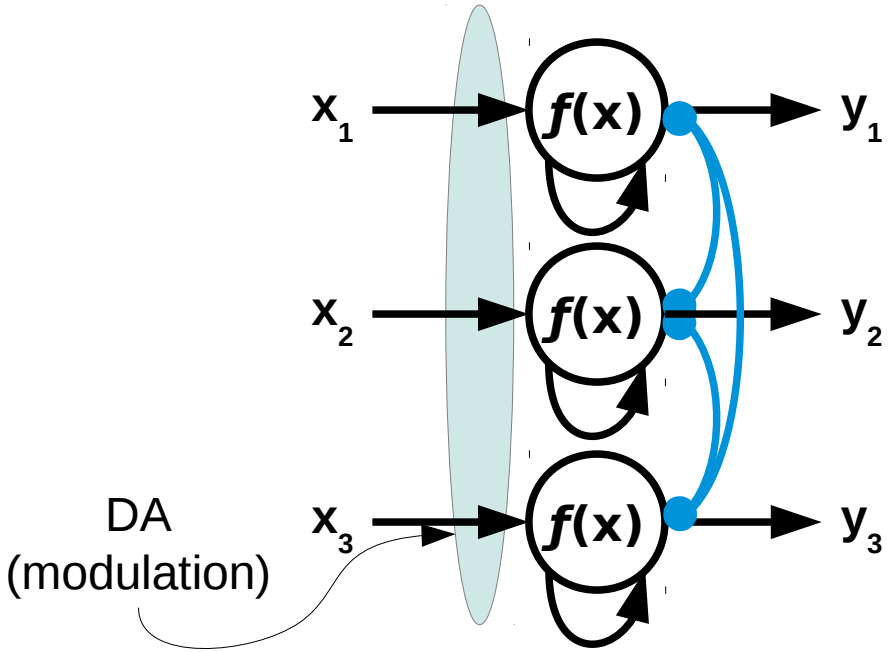
Solution:
competition via lateral inhibitions

High DA release:
strong attractors



Low DA release:
weak attractors



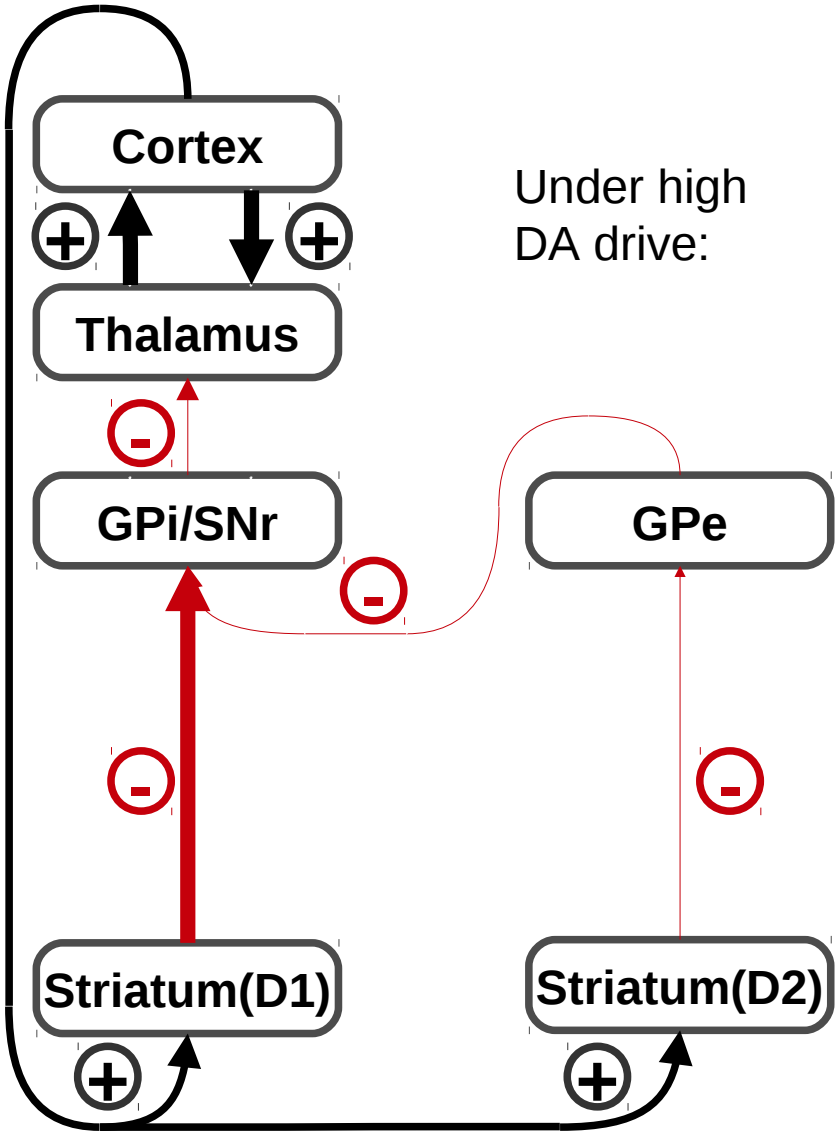


Case study: selection in the basal ganglia

Functions implemented:
- sustained selection (attractors)
- noise suppression
- modulated by dopamine
(e.g. Parkinson)

Structural features:
- dominance of GABA (efferent)

Solution:
Competing pathways.

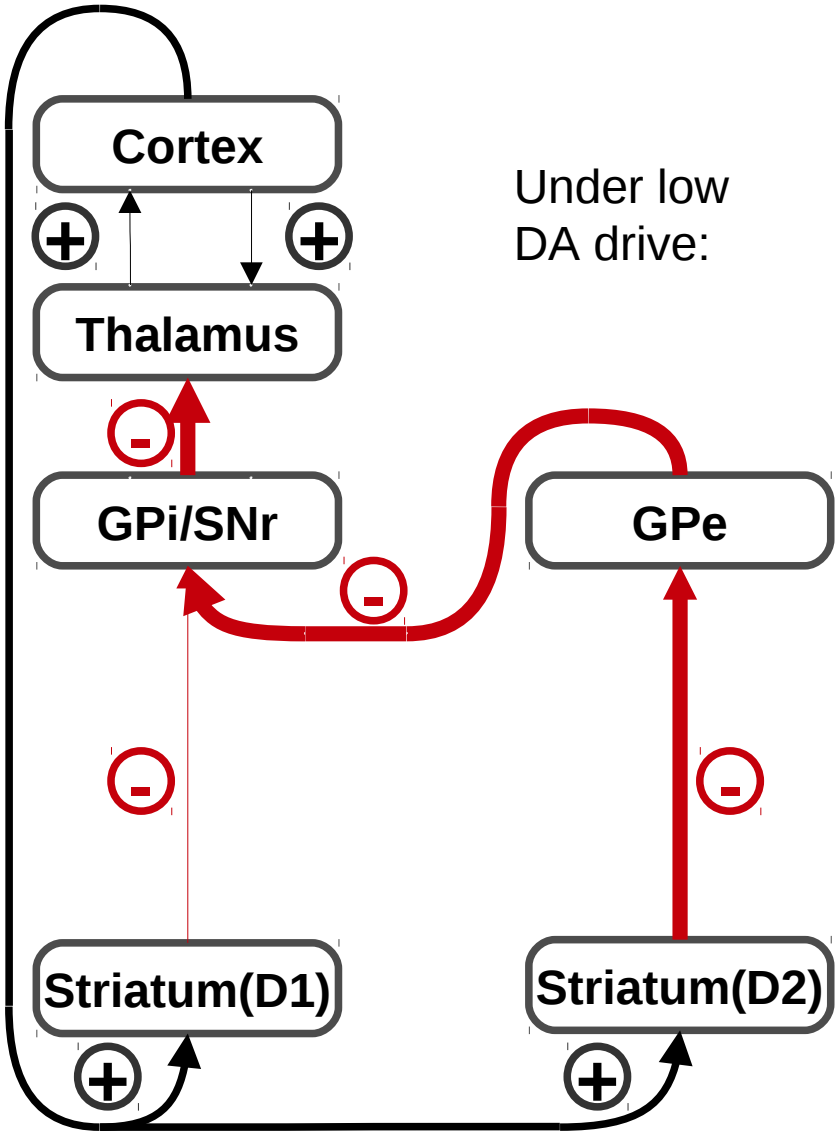


Case study: selection in the basal ganglia

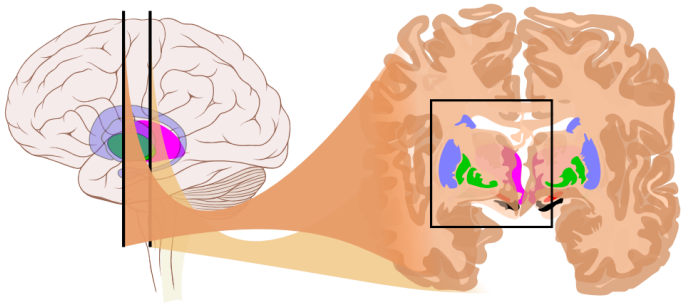
Functions implemented:
- sustained selection (attractors)
- noise suppression
- modulated by dopamine
(e.g. Parkinson)

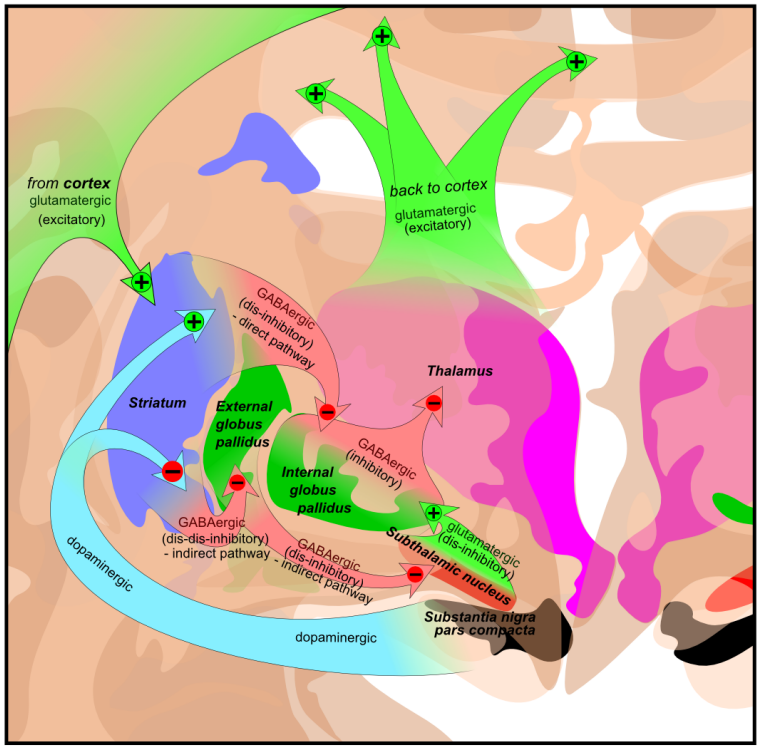
Structural features:
- dominance of GABA (efferent)

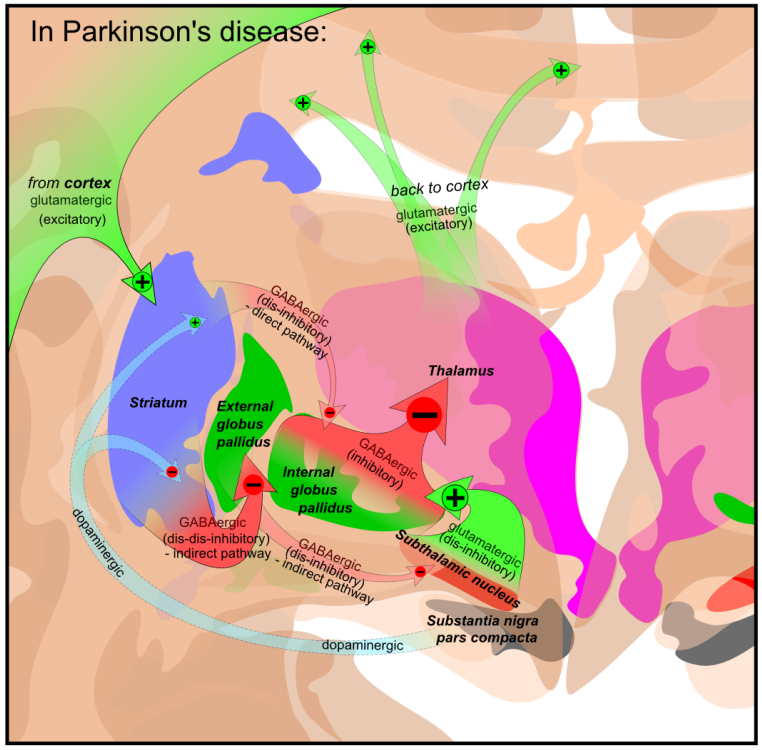
Solution:
Competing pathways.



Case study: selection in the basal ganglia







Images from: wikimedia commons, Basal Ganglia

Case study: selection in the basal ganglia

Functions implemented:
- sustained selection (attractors)
- noise suppression
- modulated by dopamine
(e.g. Parkinson)

Structural features:
- dominance of GABA (efferent)

Solution:
...slighlty more complex.