5+1 Pitch Talks – Thursday

- Orren Ravid, Samyak Gupta: SNN contour recognition and artifact generation
- Priyanka Dhulkhed, Ruky Rupasinghe, Abraham Arce: Emotion Recognition from speech
- Abdulaziz, Steven Banning: Unsupervised speech recognition
- Ryan D'souza, Michal Kobylarz, Akash Parikh, Rahul Pant: Gender Identification
- Meet Mukadam, Mandhara Jayaram: Edge Detection.
- Tina Janulis, Saraann Stanway, Shreya Sethi, Fan Liu: Emoticon recognition

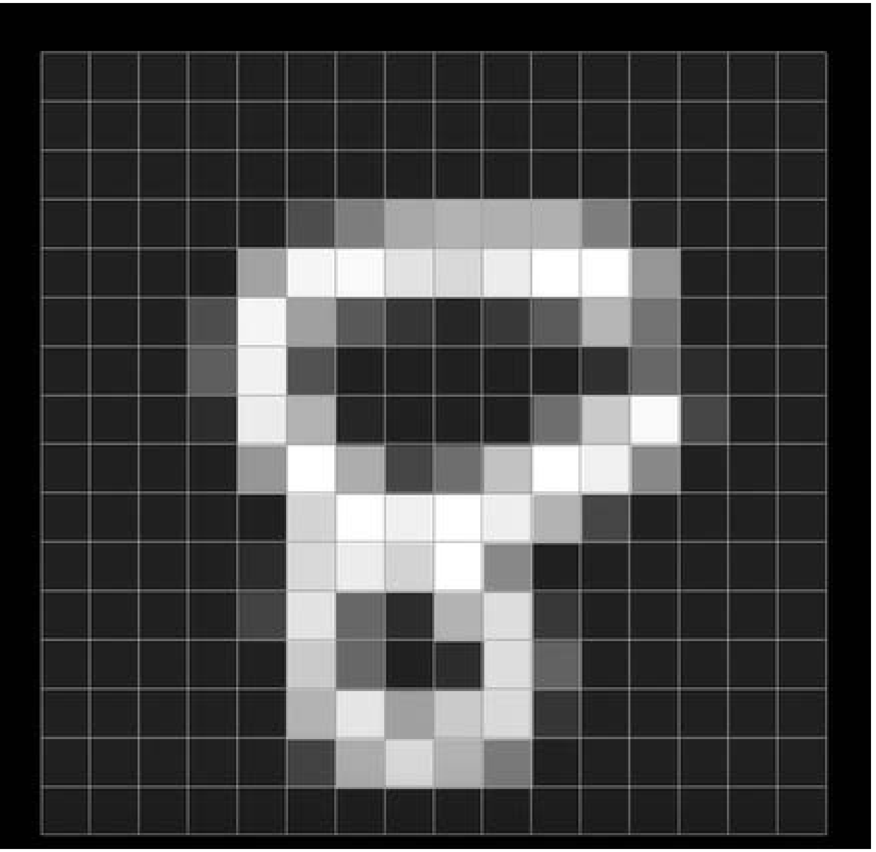
Paper presentations

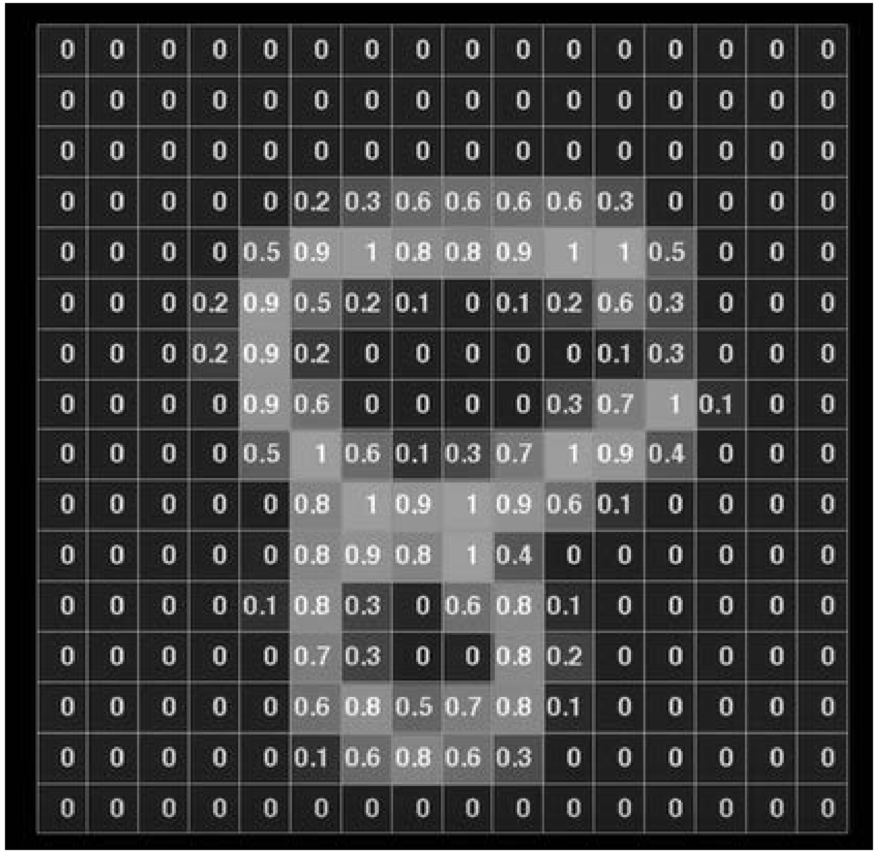
April 23rd		
Aditya S.	Hemanth C.	26_Tempotron_2006
Moulindra M.	Sandeep p.	5_SNN review 2009
Tina J.	Shreya S.	2_Eliasmith Science 2012
Anna F.	Jeremy K.	27_Abbott_2015
Shuaishuai S	Xu Y.	28_Sejnowski_Gradient Descent 2018
April 25th		
Andrea W.	Kyle C.	20_Donoghue 2010
Chris I.	Aviv K.	11_Wade 2010
Aravind	Kunal M.	12_Wang 2008
Blue A.	Abraham A.	18_Perez-Pena 2013
Rukmal R.	Priyanka D.	15_Baladron 2015
April 30th		
Mikolaj P.	Sophia D.	22_Maass_2019
Srihari S.	Anjali M.	21_Davies Loihi 2018
Ryan D.	Michal K.	23_Maass_2018
Mandhara J.	Meet M.	24_Sommer_2019
Bhargav D.	Harsh P.	25_Zhang_2019

Information encoding: Time

- Information is carried in **the individual action potentials**, rather than aggregate measures such as “firing rate”
- Rather than the form of the action potential, it is the *number* and the *timing* of spikes that matter
- In fact, it has been established that **the exact timing of spikes can be a means for coding information**, for instance in the electro-sensory system of electric fish, in the auditory system of echo-locating bats, and in the visual system of flies

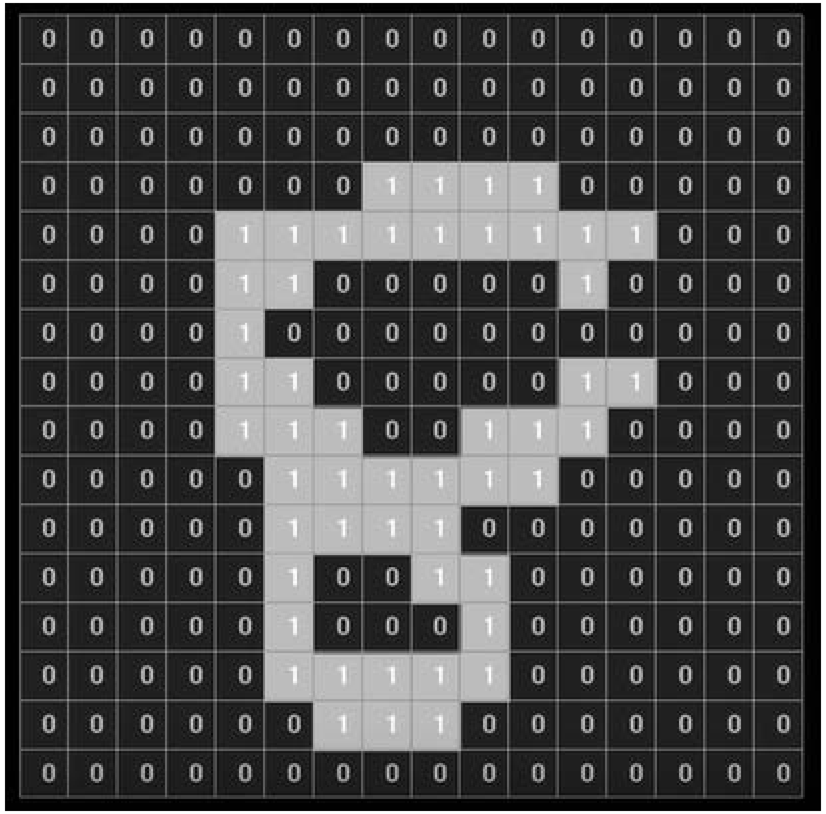
Image to spikes





0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0.2	0.3	0.6	0.6	0.6	0.6	0.3	0	0	0	0
0	0	0	0	0.5	0.9	1	0.8	0.8	0.9	1	1	0.5	0	0	0
0	0	0	0.2	0.9	0.5	0.2	0.1	0	0.1	0.2	0.6	0.3	0	0	0
0	0	0	0.2	0.9	0.2	0	0	0	0	0	0.1	0.3	0	0	0
0	0	0	0	0.9	0.6	0	0	0	0	0.3	0.7	1	0.1	0	0
0	0	0	0	0.5	1	0.6	0.1	0.3	0.7	1	0.9	0.4	0	0	0
0	0	0	0	0	0.8	1	0.9	1	0.9	0.6	0.1	0	0	0	0
0	0	0	0	0	0.8	0.9	0.8	1	0.4	0	0	0	0	0	0
0	0	0	0	0.1	0.8	0.3	0	0.6	0.8	0.1	0	0	0	0	0
0	0	0	0	0	0.7	0.3	0	0	0.8	0.2	0	0	0	0	0
0	0	0	0	0	0.6	0.8	0.5	0.7	0.8	0.1	0	0	0	0	0
0	0	0	0	0	0.1	0.6	0.8	0.6	0.3	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

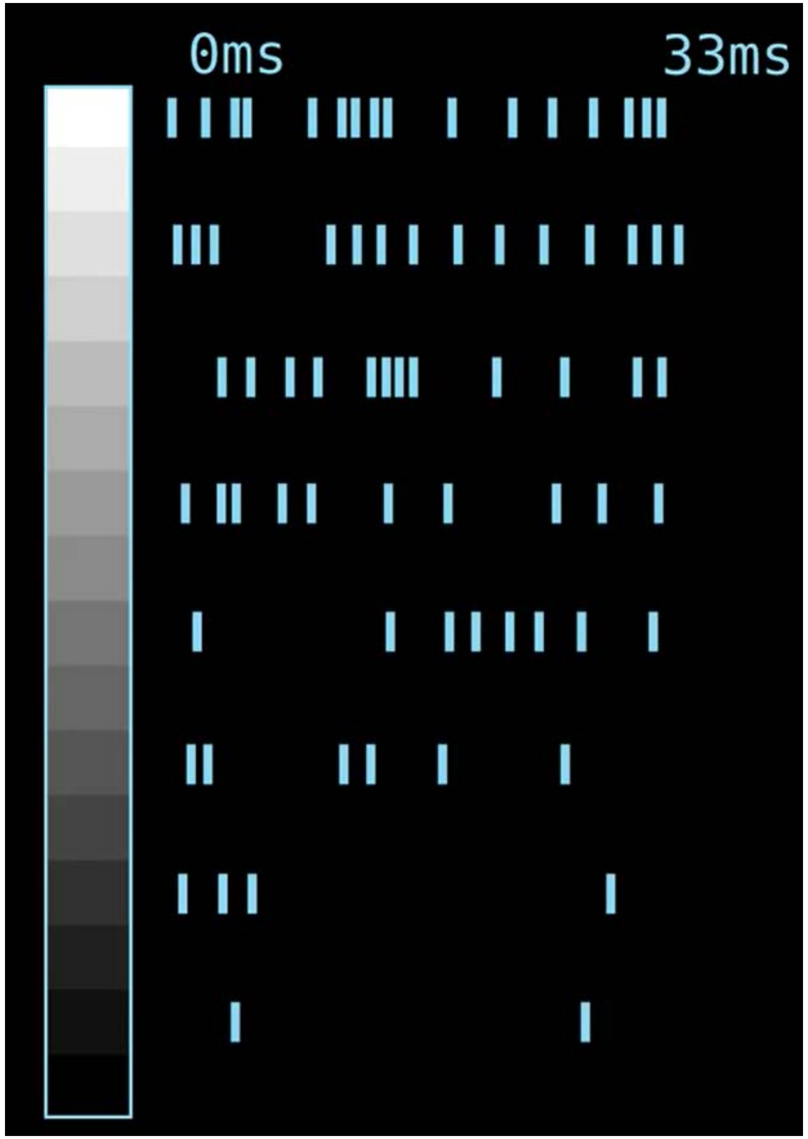
Encoding Images as Neural Spikes



Thresholding   1 spike for 1
0 spikes for 0

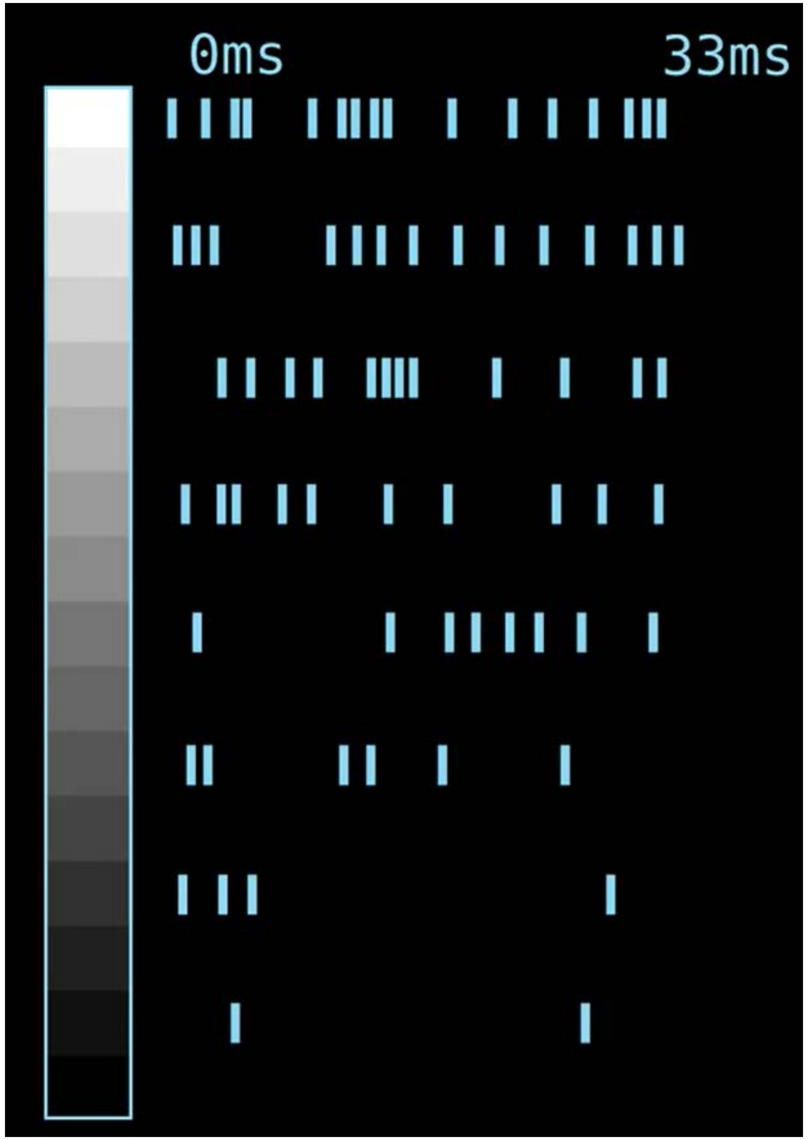
Encoding complicated images

Number of Spikes proportional to the intensity level of the edge

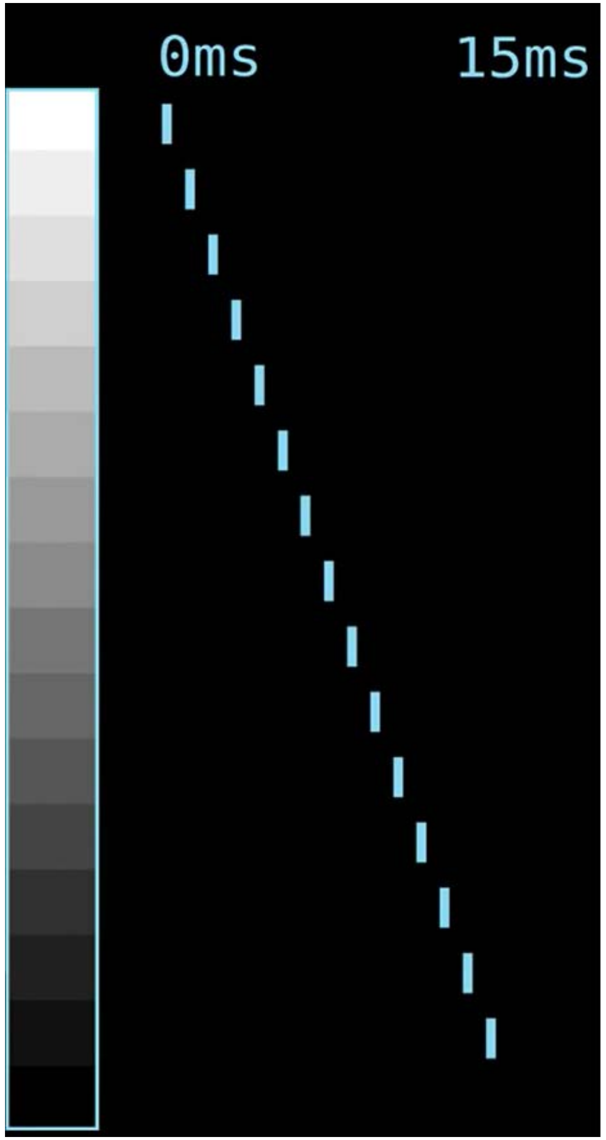


Timing of the spikes does not matter – we are just counting them

Rate-Encoding



Time-Encoding



This is how your retina works!

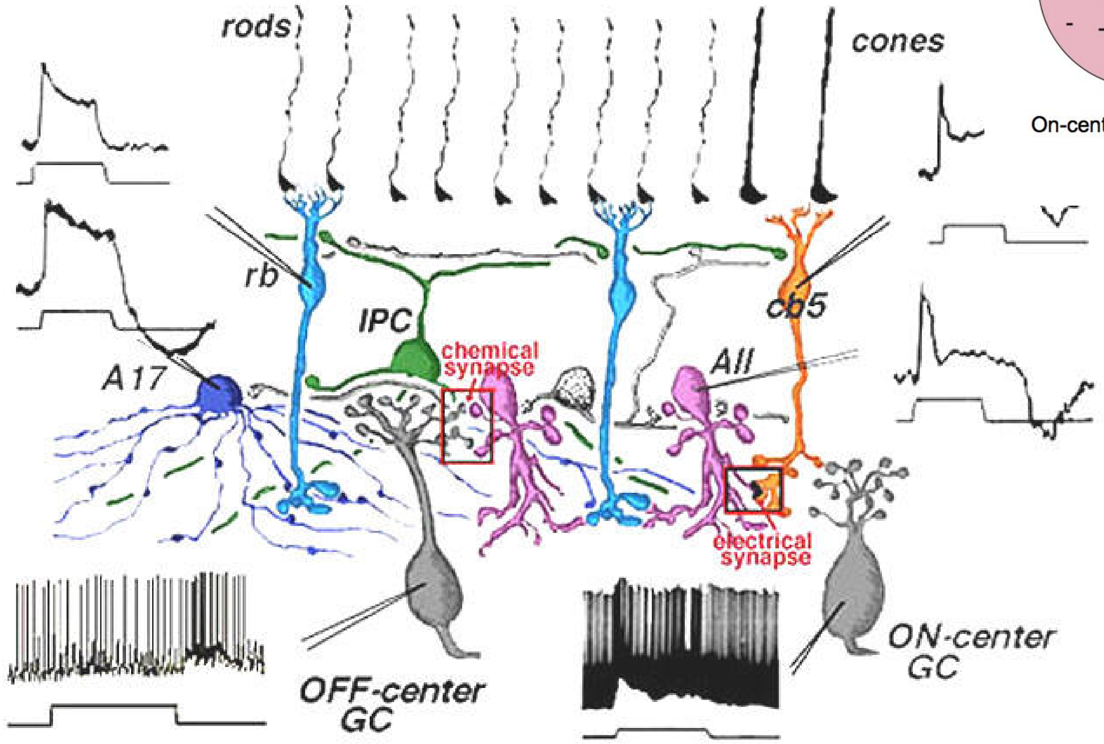
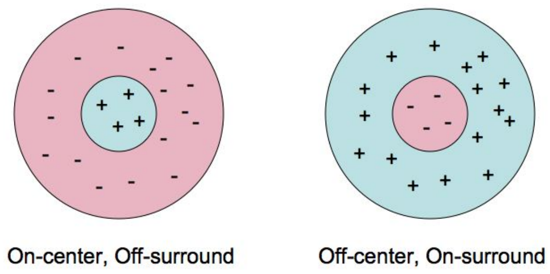
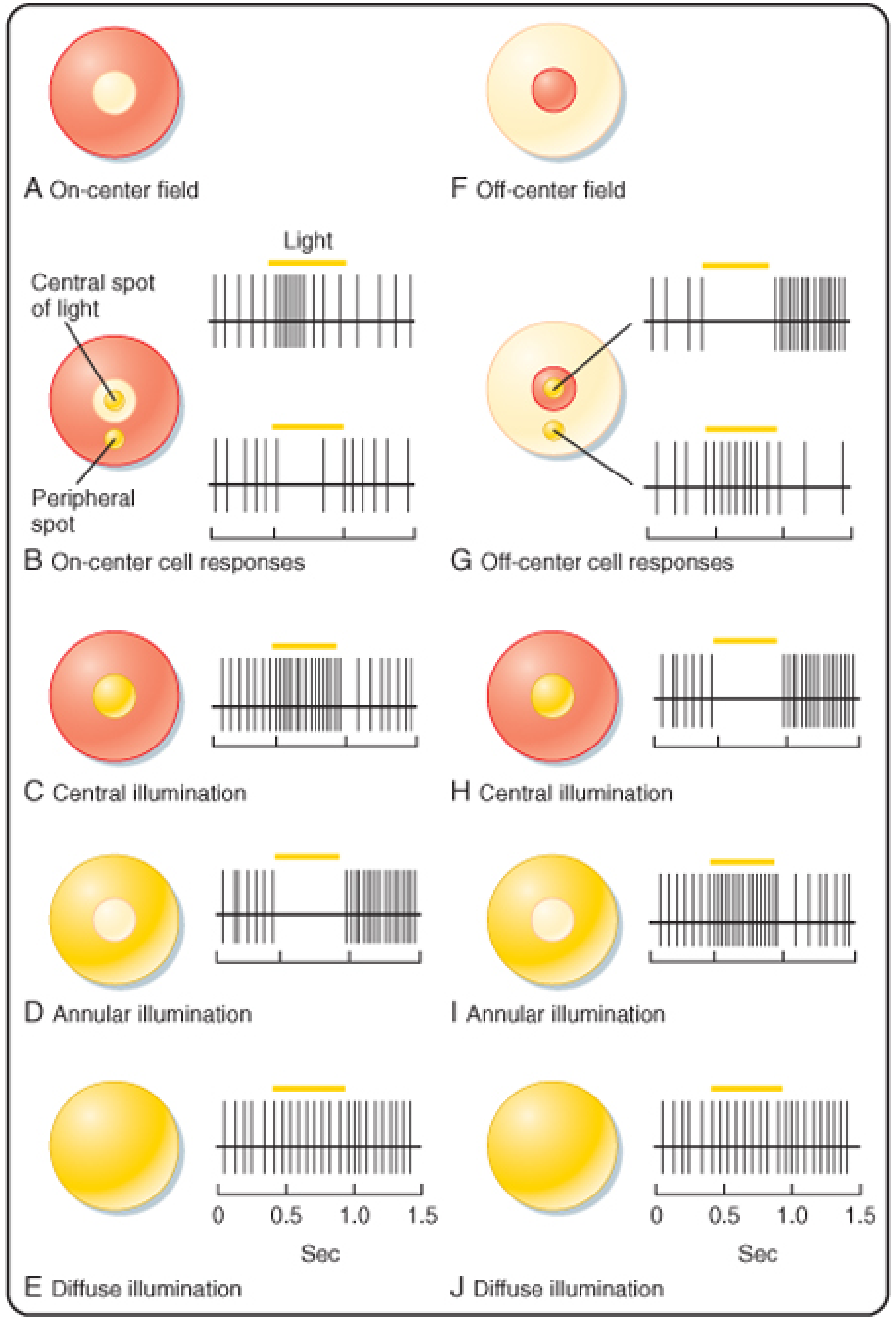


Fig. 20. Summary diagram of the rod pathway neurons and their responses. Amacrine cells intervene between rod bipolar and ON and OFF center ganglion cells.

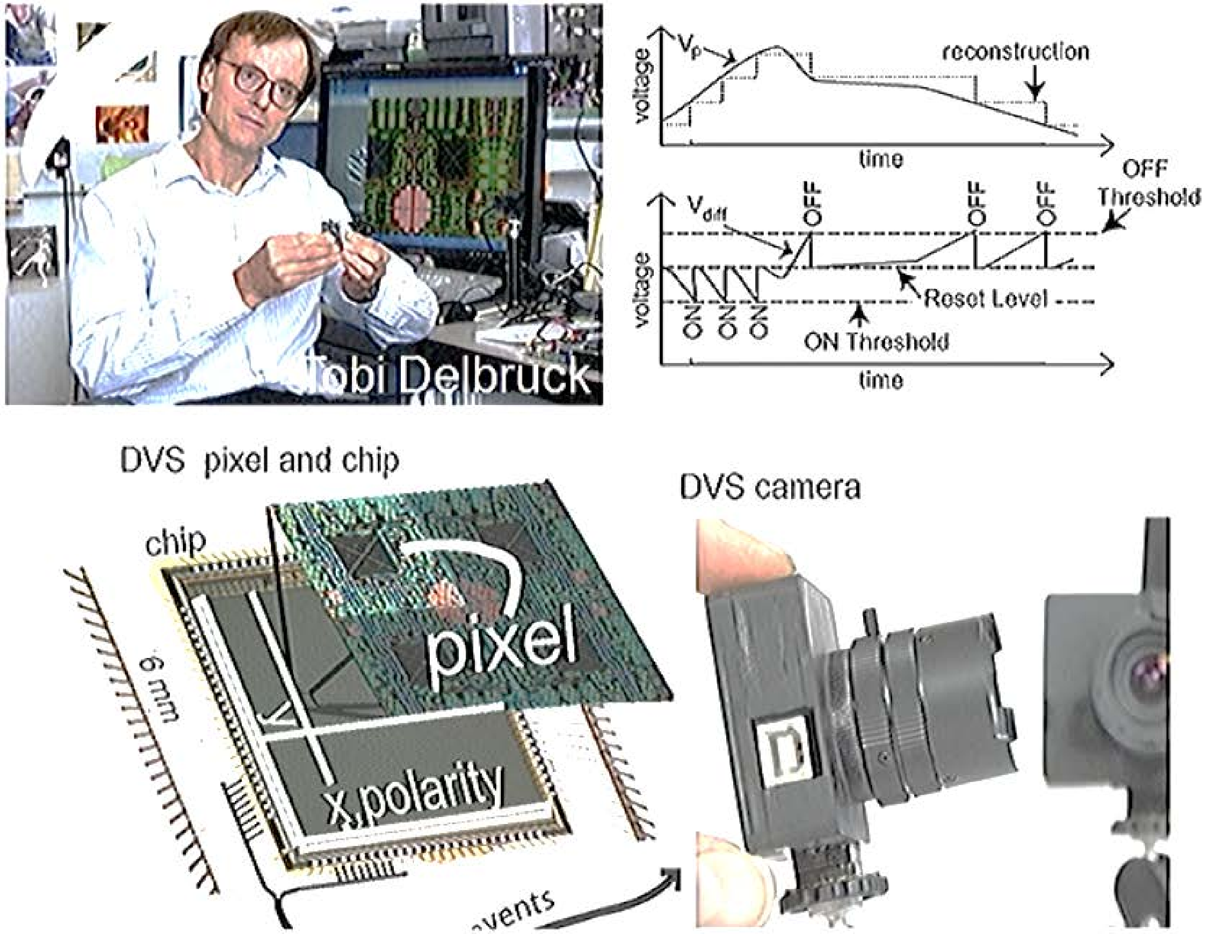
Receptive Fields





Koeppen & Stanton: Berne and Levy Physiology, 6th Edition.

Neuromorphic camera (University of Zurich)



Lichtsteiner, Posch and Delbruck, 2008

A simple camera that encodes change in intensity levels in the form of spikes

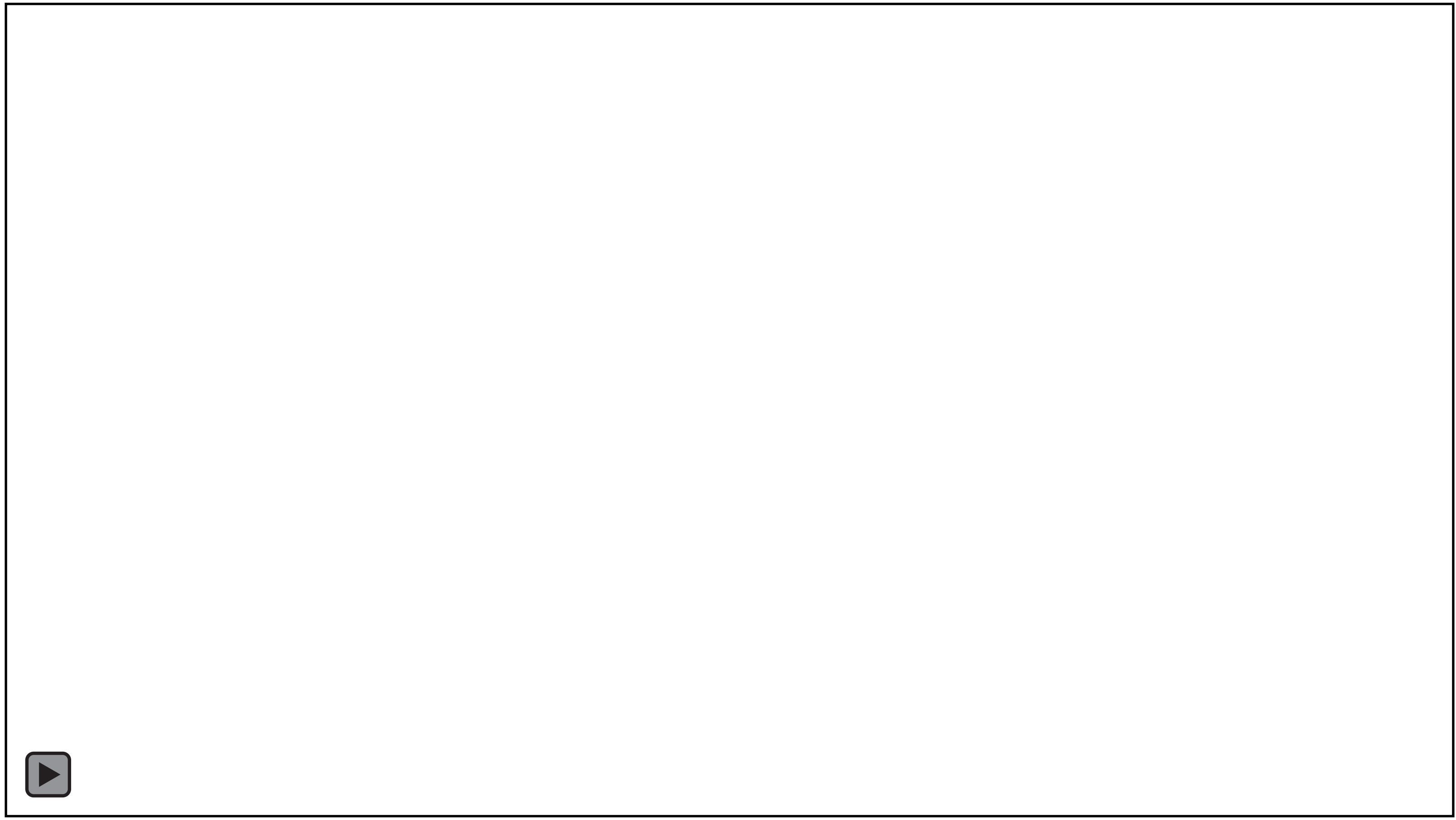
Each pixel has two different outputs

The first output gives a spike when the voltage goes up by an increment

The second output gives a spike when the voltage goes down by an increment

What are we gaining?

- No frame buffer
- No loss of information



Computing with SNNs

- Since the basic principle underlying SNNs is so radically different, it is not surprising that much of the work on traditional neural networks, such as **learning rules** and theoretical results, has to be adapted, or even has to be fundamentally rethought.
- The first difficult task is to define "the" model of neuron, as there exist numerous variants already
  - H-H model
  - Integrate and Fire model
  - Izhikevich model
  - Spike Response Model
- The second task is to model (and harness) synaptic plasticity
  - Different methods for learning in SNNs (e.g., Reservoir Computing, STDP, Tempotron)

Neural Computation Theories of Learning

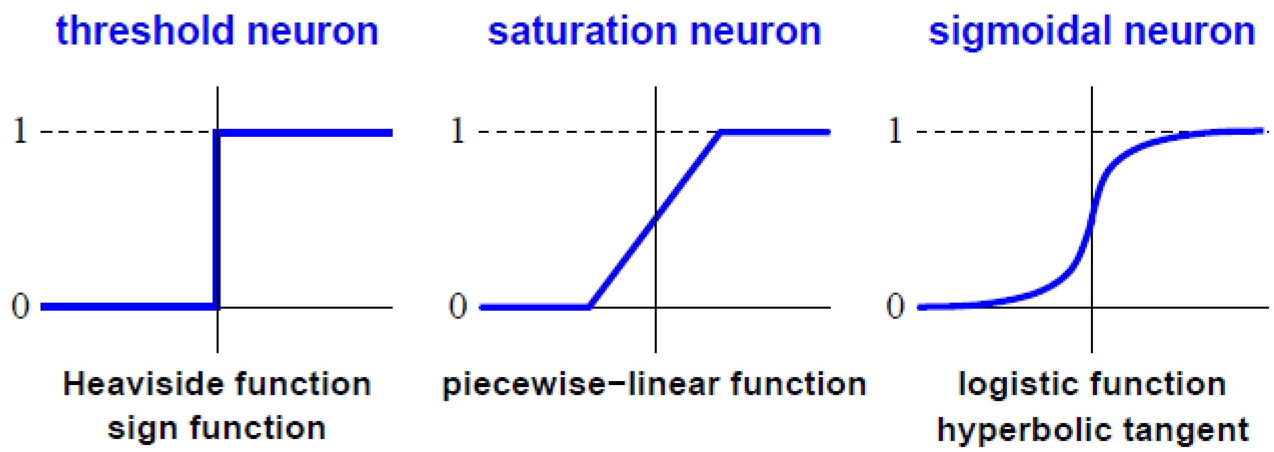
- Introduction
- Hebbian Learning
- Unsupervised Hebbian Learning
- Supervised Learning
- Reinforcement Learning
- Spike-Timing Dependent Plasticity
- Plasticity of Intrinsic Excitability
- Homeostatic Plasticity
- Complexity of Learning

Computing with SNNs

- Since the basic principle underlying SNNs is so radically different, it is not surprising that much of the work on traditional neural networks, such as **learning rules** and theoretical results, has to be adapted, or even has to be fundamentally rethought.
- The first difficult task is to define “the” model of neuron, as there exist numerous variants already
  - H-H model
  - Integrate and Fire model
  - Izhikevich model
  - Spike Response Model
- The second task is to model (and harness) synaptic plasticity
  - Different methods for learning in SNNs (e.g., Reservoir Computing, STDP, Tempotron)

From Biological Computing to Artificial Neural Networks

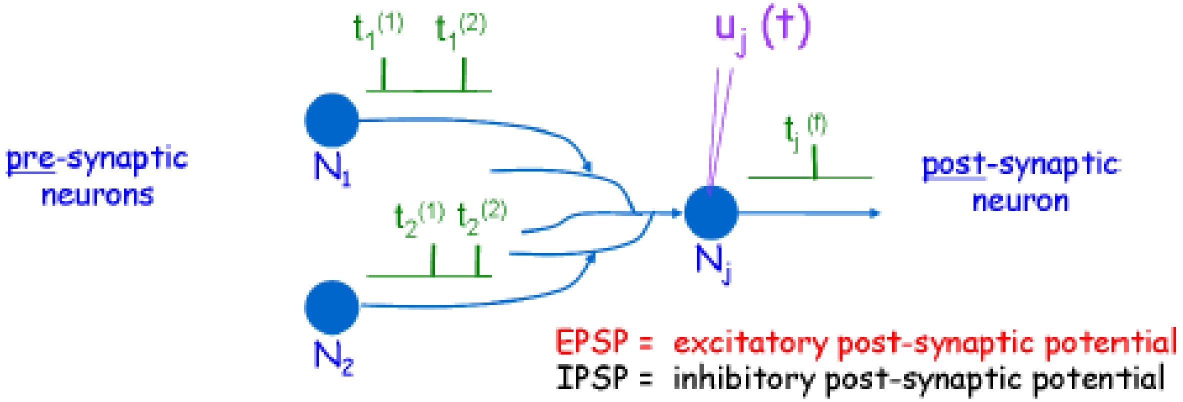
Functions connecting the input to the output

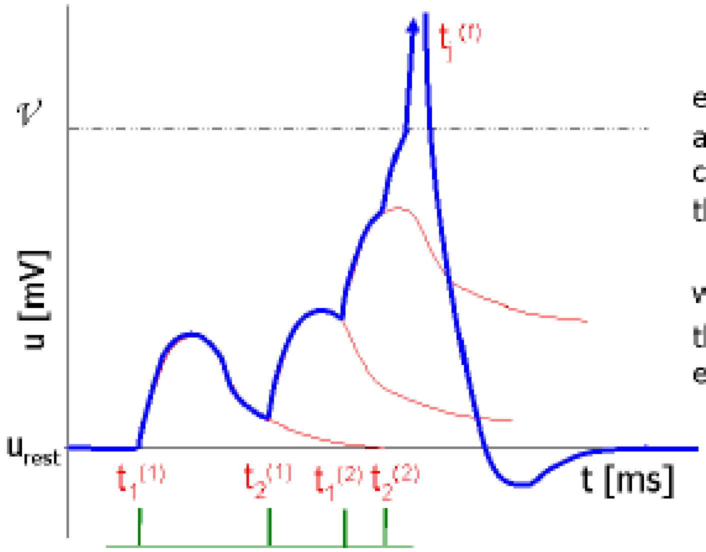


The output is calculated based on the dot product < X, W > computation

Spiking Neural Networks – the 3rd Gen ANNs

Now the Input – Output relationship is no longer some arbitrary function but resembles the voltage fluctuations in the simulated neuron's membrane





each pre-synaptic spike generates an EPSP (red curves) or IPSP in case of negative weight ; they are all added (blue line) ;

when u_j (t) reaches the threshold $\mathcal{V}$ the post-synaptic neuron N_j emits a spike, in its turn.

A model of spiking neuron: N_j fires a spike whenever the weighted sum of incoming EPSPs generated by its pre-synaptic neurons reaches a given threshold. The graphic (right) shows how the membrane potential of N_j varies through time, under the action of the four incoming spikes (left).

Neural Computation Theories of Learning

- Introduction
- **Hebbian Learning**
- Unsupervised Hebbian Learning
- Supervised Learning
- Reinforcement Learning
- Spike-Timing Dependent Plasticity
- Plasticity of Intrinsic Excitability
- Homeostatic Plasticity
- Complexity of Learning

Hebb, 1949

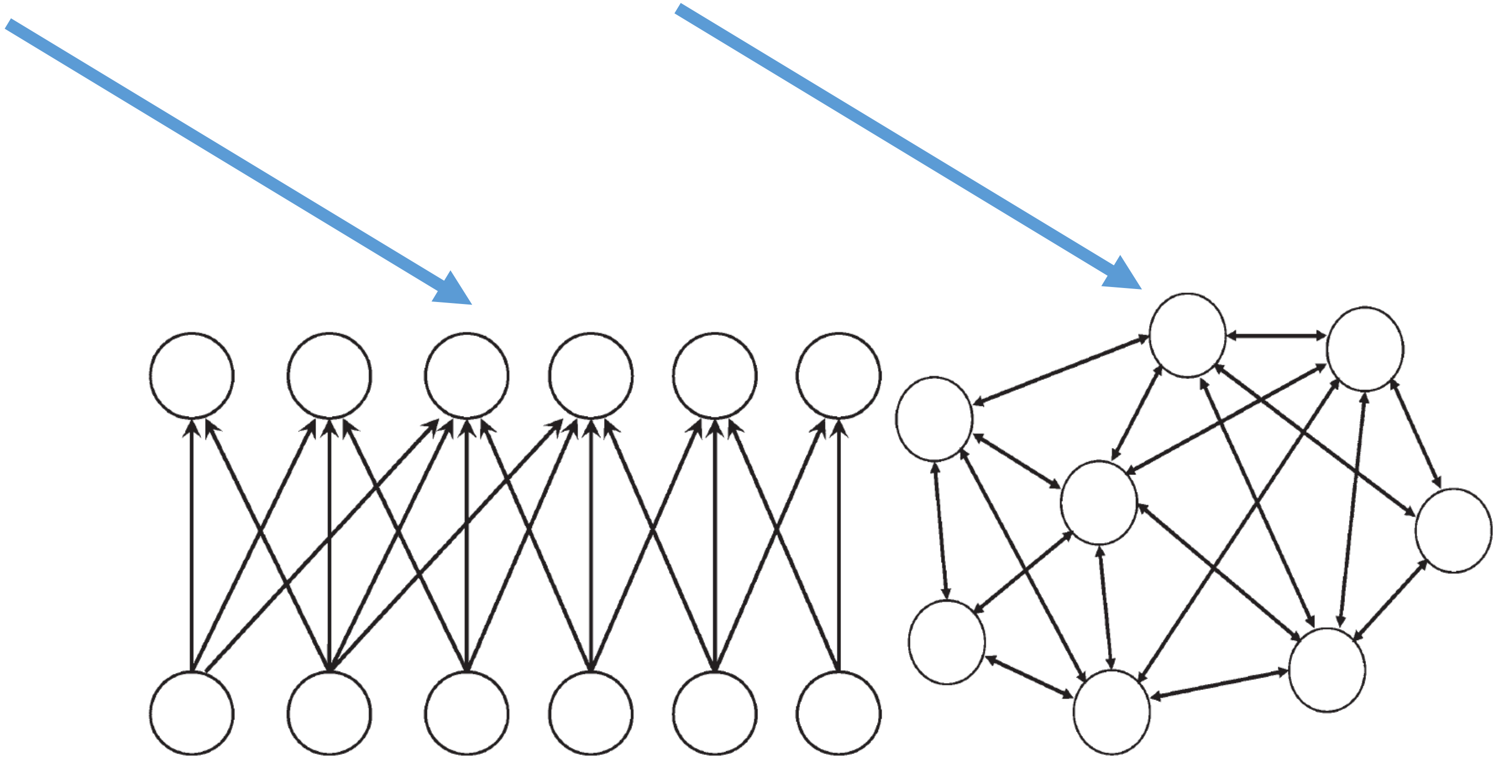
When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased

Neurons that fire together, wire together

Introduction

- Hebb’s postulate has served as the starting point for studying the learning capabilities of ANN and for the theoretical analysis and computational modeling of biological neural systems.
- The architecture of an ANN determines its behavior and learning capabilities. The architecture of a network is defined by the connections among the artificial neural units and the function that each unit performs on its inputs.
  - Two general classes are feedforward and recurrent architecture. The simplest feedforward network has one layer of input units and one layer of output units. All connections are unidirectional and project from the input units to the output units.
    - The **perceptron** is an example of a simple feedforward network (Rosenblatt, 1958). It can learn to classify patterns from examples. It turned out that the perceptron can only classify patterns that are linearly separable – that is, if the positive patterns can be separated from all negative patterns by a plane in the space of input patterns.
    - More powerful multilayer feedforward networks can discriminate patterns that are not linearly separable. In a multilayer feedforward network, the ‘hidden’ layers of units between the input and output layers allow more flexibility in learning features.

Feedforward vs. Recurrent Networks



A simple recurrent network can have a uniform architecture such as all-to-all connectivity combined with symmetric weights between units, as in a Hopfield network (Hopfield, 1982), or it can be a network with specific connections designed to model a particular biological system.

Learning Algorithm

A learning algorithm specifies how and under what conditions a learning rule or a combination of learning rules should be applied to adjust the network parameters.

For a simple task, it is possible to invent an algorithm that includes only one type of learning rule, but for more complex problems, an algorithm may involve a combination of several different learning rules.

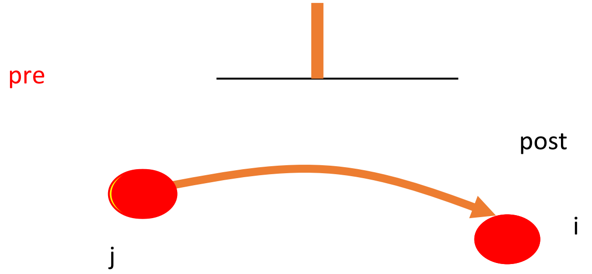
Learning in networks

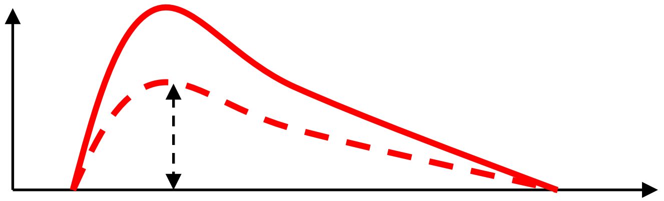
Modeling learning processes in networks implies that the strengths of connections and other parameters are adjusted according to a **learning rule.**

- Other parameters that may change include the threshold of the unit, time constants, and other dynamical variables.
- A learning rule is a dynamical equation that governs changes in the parameters of the network.

- There are **three main categories** of learning rules: unsupervised, supervised, and reinforcement.
  - Unsupervised learning rules are those that require no feedback from a teaching signal.
  - Supervised learning rules require a teacher, who provides detailed information on the desired values of the output units of the network, and connections are adjusted based on discrepancies between the actual output and the desired one.
  - Reinforcement learning is also error correcting but involves a **single scalar** signal about the overall performance of the network. Thus, reinforcement learning requires less-detailed information than supervised learning.

Learning through synaptic changes: Classification





Induction of changes

- fast (if stimulated appropriately)
- slow (homeostasis)

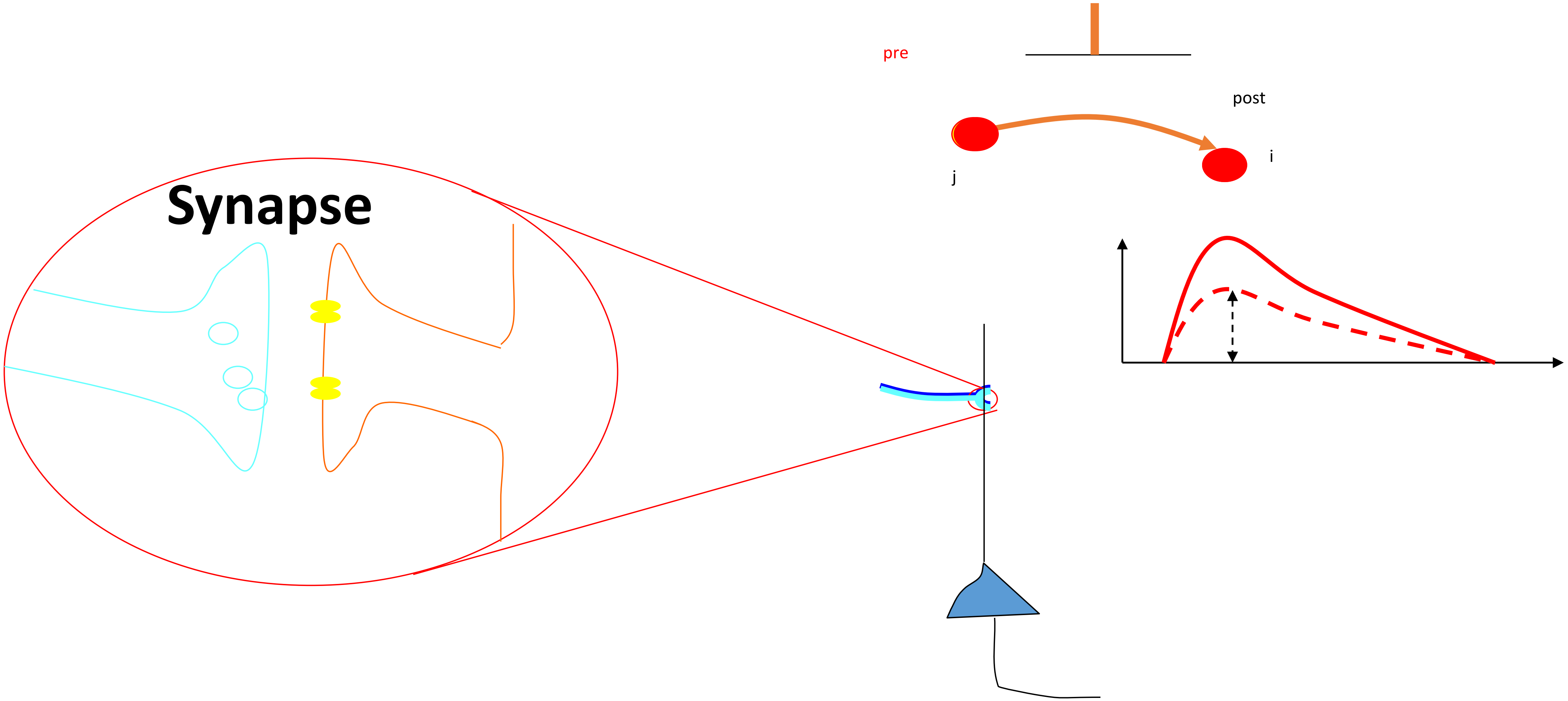
Persistence of changes

- long (LTP/LTD)
- short (short-term plasticity)

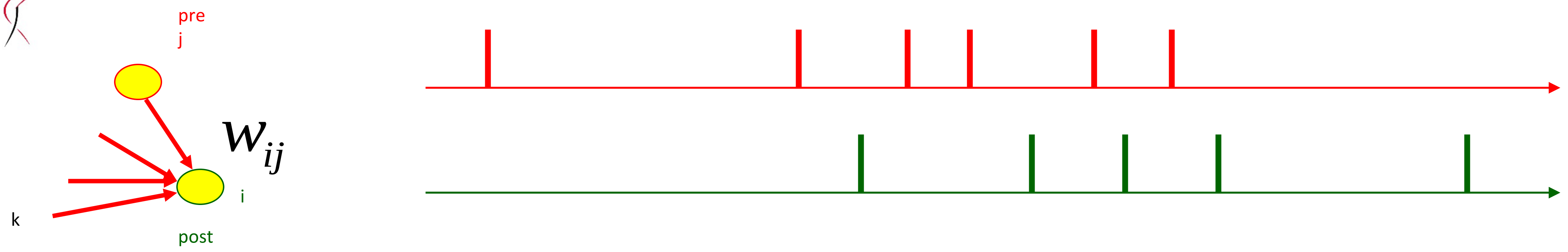
Functionality

- useful for learning a new behavior
- useful for development (wiring for receptive field development)
- useful for activity control in network
- useful for coding

Synaptic Plasticity



Synaptic Plasticity & Hebbian Learning



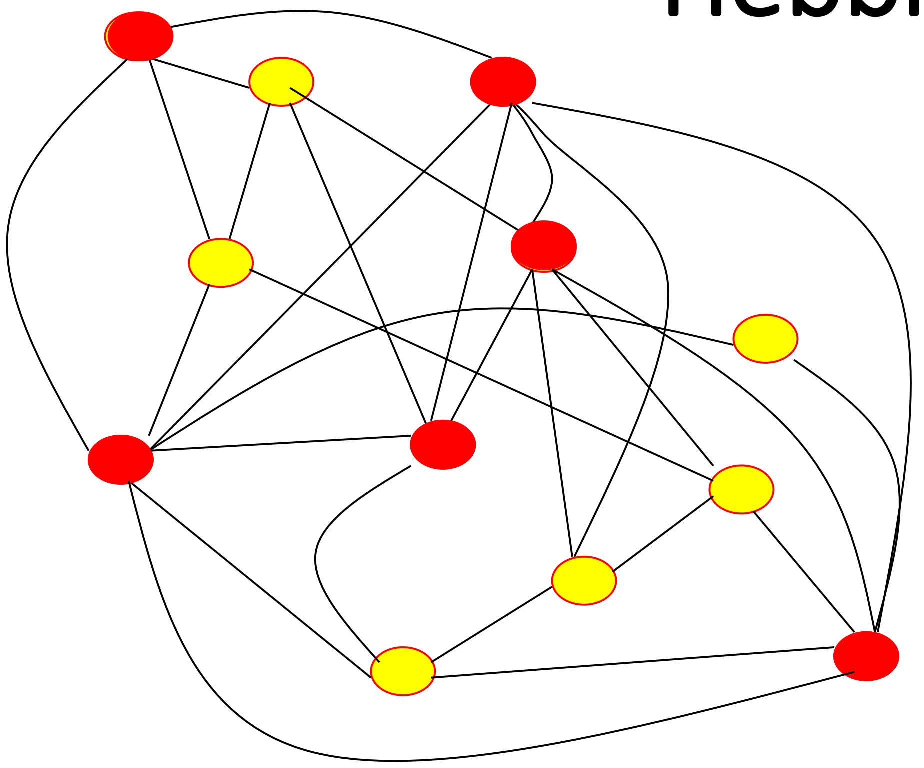
When an axon of cell j repeatedly or persistently takes part in firing cell i, then j's efficiency as one of the cells firing i is increased

Hebb, 1949

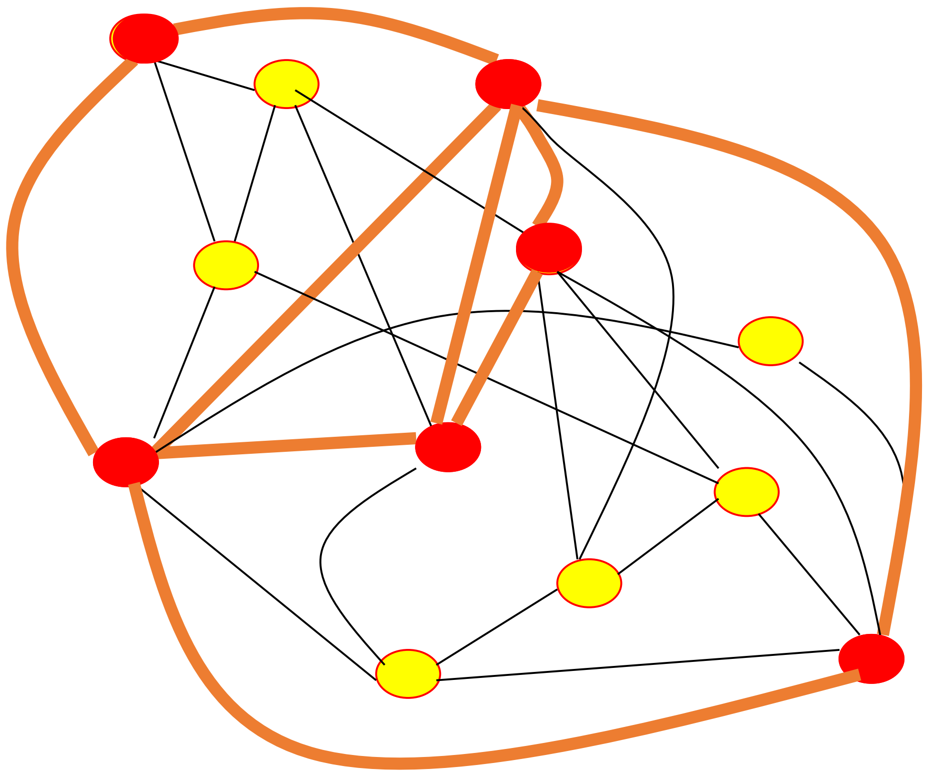
- local rule
- simultaneously active (correlations)

Correlation based Learning

Hebbian Learning



Correlation based Learning

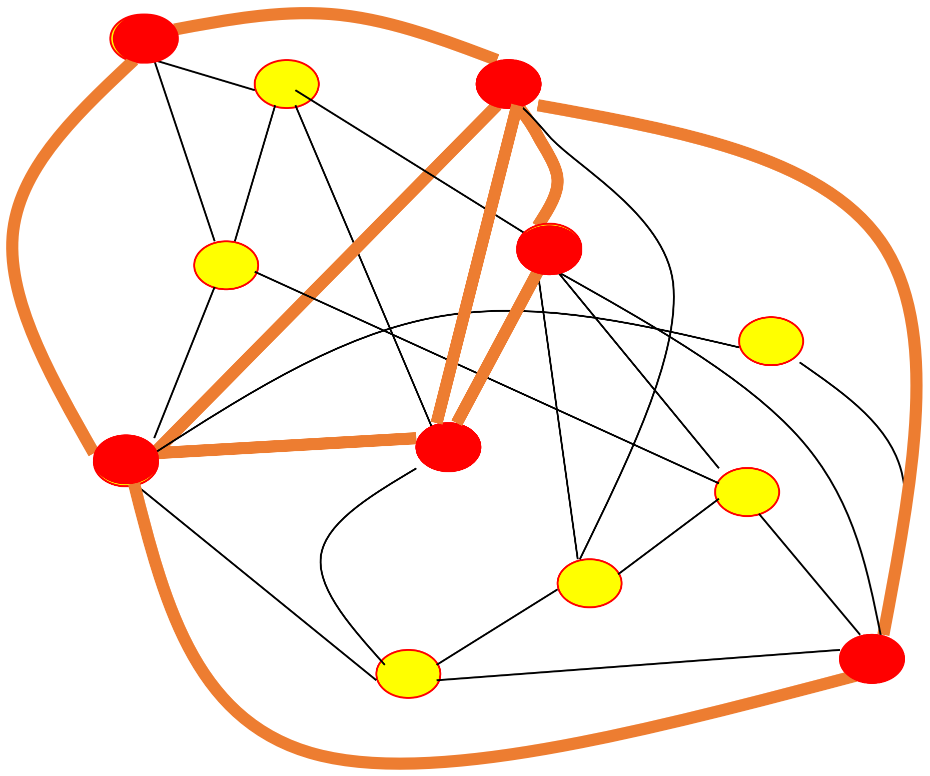


item memorized

Correlation based Learning

Recall:

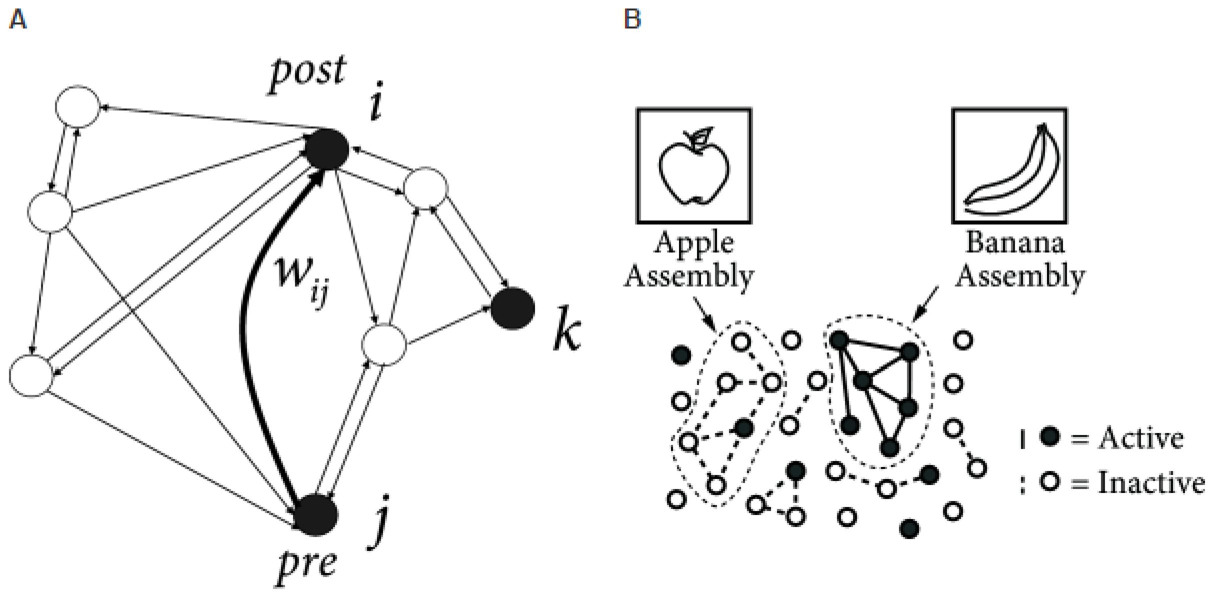
Partial info



item recalled

Correlation based Learning

- Hebbian learning is unsupervised
  - There is no notion of 'good' or 'bad' changes of a synapse
- Synaptic changes happen whenever there is joint activity of pre- and postsynaptic neurons
  - Firing patterns may reflect sensory stimulation or ongoing brain activity, but there is no feedback signal from a 'supervisor' or from the environment



Hebbian learning. A. The change of a synaptic weight w_{ij} depends on the state of the presynaptic neuron j and the postsynaptic neuron i and the present efficacy w_{ij}, but not on the state of other neurons k. B. Hebbian learning strengthens the connectivity within assemblies of neurons that fire together, e.g. during the perception of banana. Schematic figure.