Learning in spiking neural networks

CNS Lab

Computational **N**eural **S**imulations
Complex **N**etwork **S**tudies
Central **N**ervous **S**ystems
Cyber-**N**atural **S**ystems
Cryptography, **N**etworking, and **S**teganography
Cellular **N**euro**S**ciences
Cognitive **N**euro**S**ciences

Missouri S&T | Computer Science

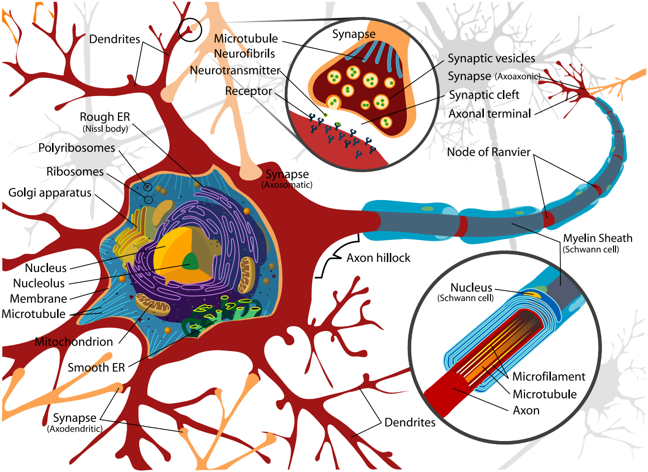
Counter to the current paradigm

“We shall envision the mind (or brain) as composed of many partially autonomous ”agents”—as a ”Society” of smaller minds. Each sub-society of mind must have its own internal epistemology and phenomenology, with most details private, not only from the central processes, but from one another.” (Minsky, K-Lines; 1980)

Outline

Neuron structure

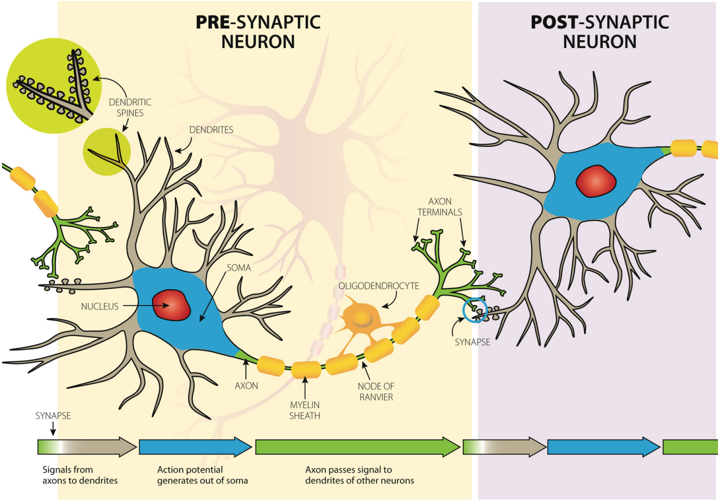
Synapses on dendrites (inputs) on soma (cell body) integrate and fire spikes down axons (output) toward synapses



Which of the full set of real biological features are enough for domain general learning? What can we eliminate?

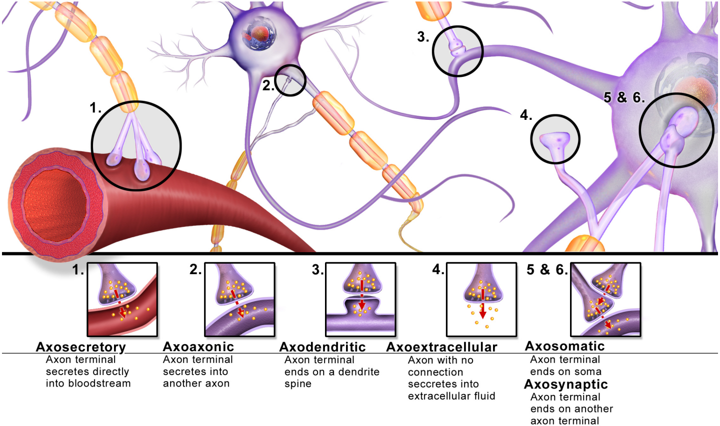
Outline

Synapse types



Remember for upcoming notation

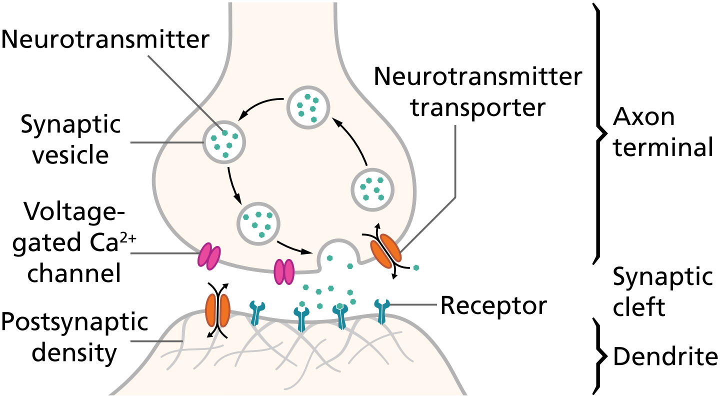
Connection types



Are these relevant computationally?

Outline

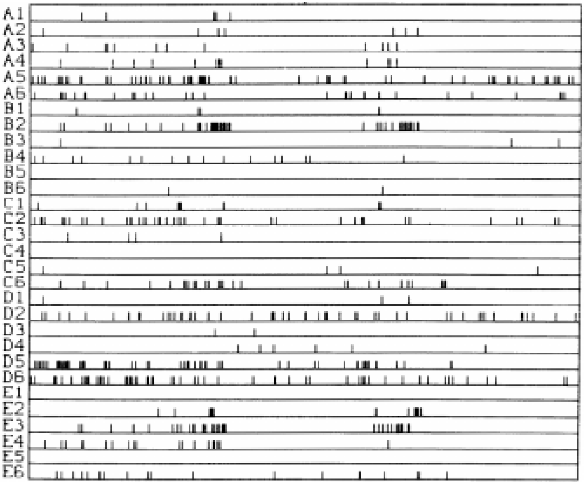
Synapses transmit chemically between neurons



This is where most of the learning appears to happen

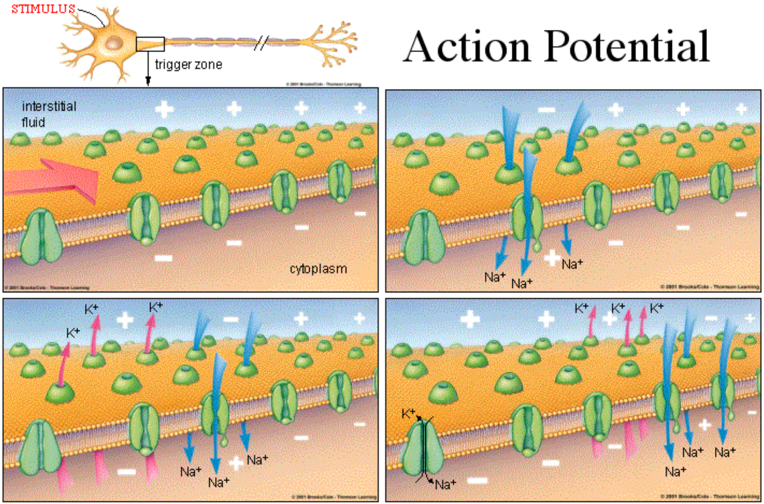
Outline

Action potentials

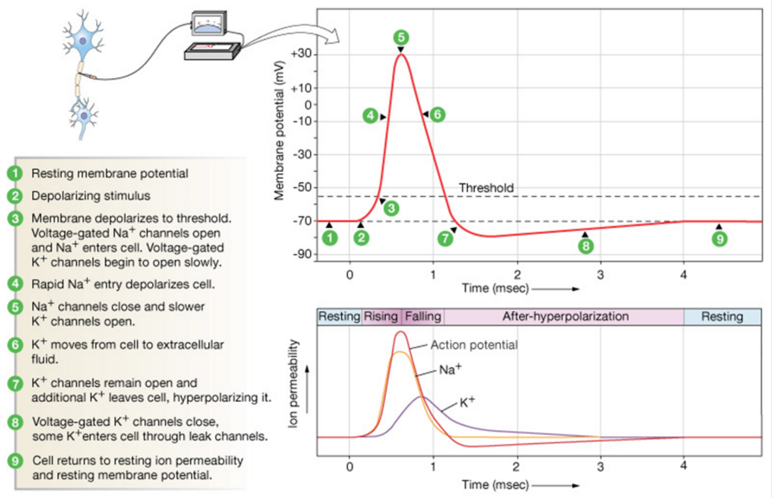


A 4 second recording of the neural activity recording from 30 neurons of the visual cortex of a monkey. Each vertical bar indicates a spike. The human brain can recognize a face within 150ms, which correlates to less than 3mm in this diagram; dramatic changes in firing frequency occur in this time span, neurons have to rely on information carried by solitary spikes.

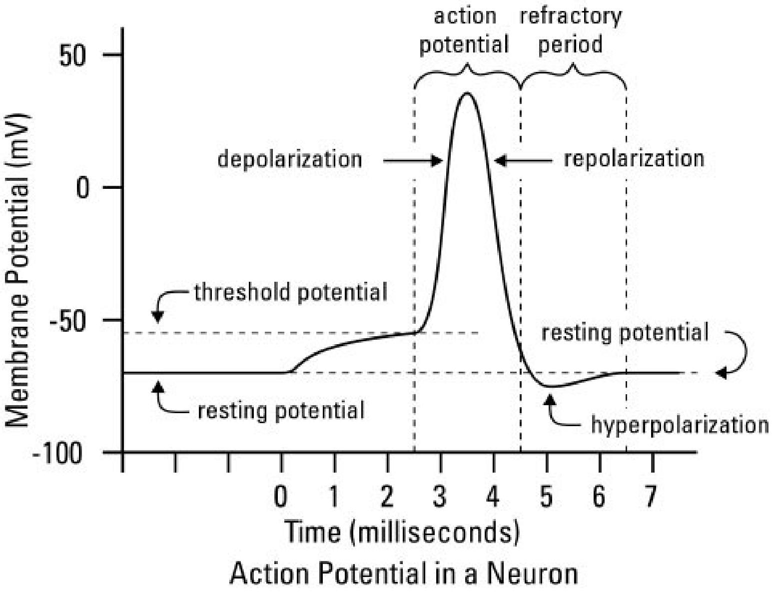
Action potentials



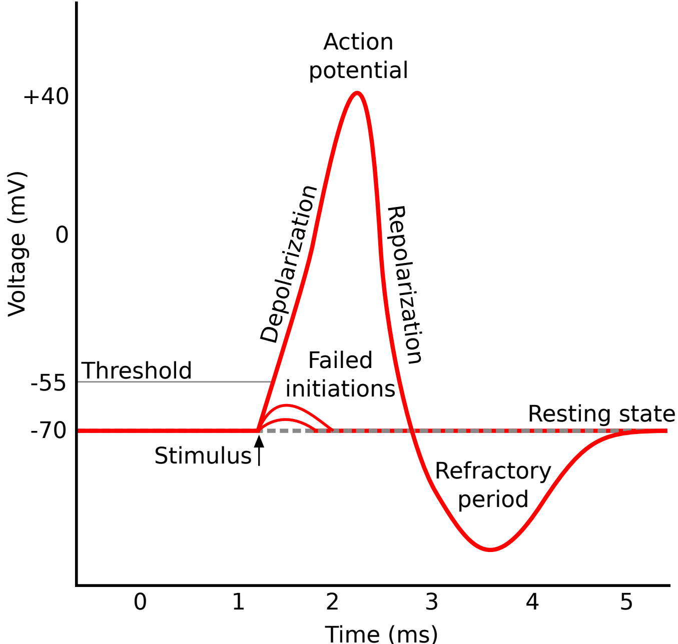
Neurons fire (spike) to transmit information (mostly)



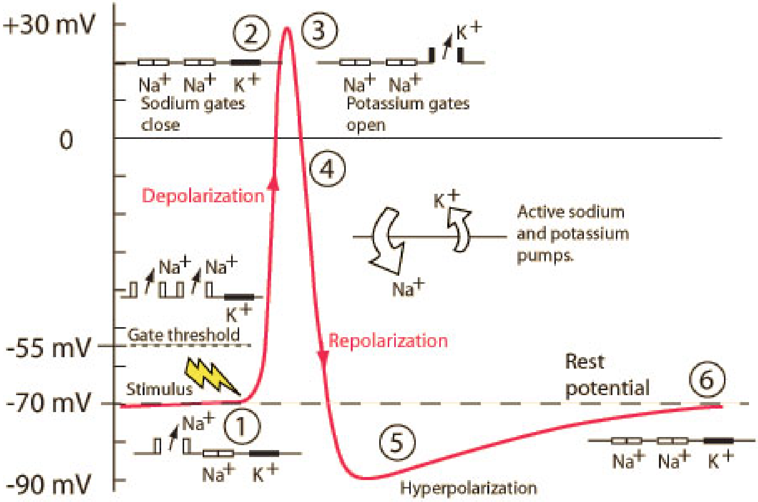
Action potentials



Action potentials



Action potentials



Action potentials

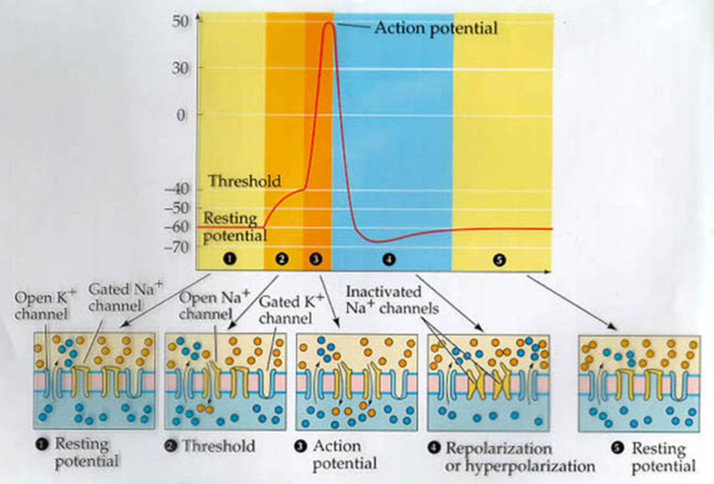
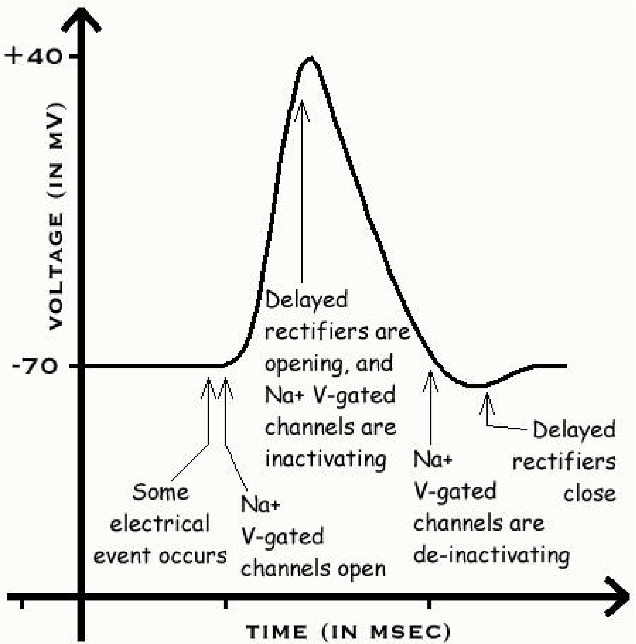
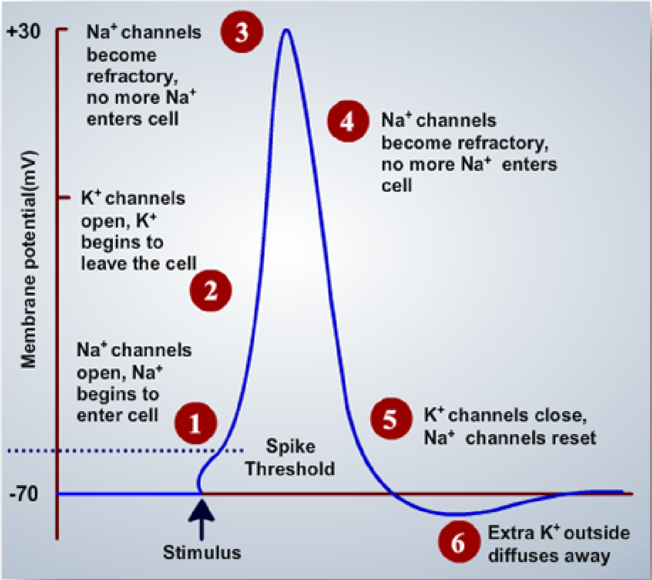


Figure 41.9 The Course of an Action Potential

Action potentials

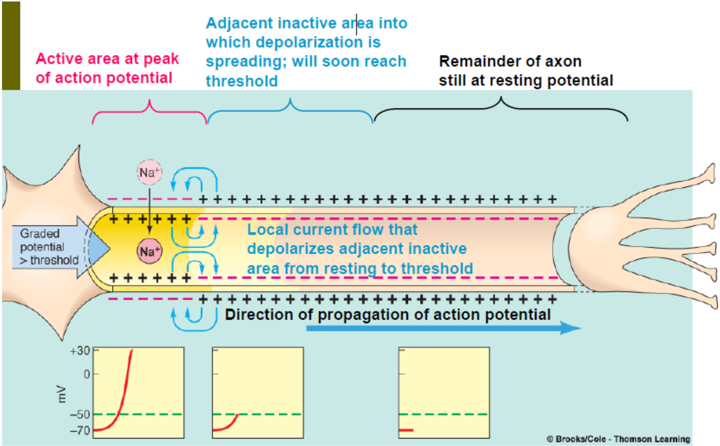


Action potentials

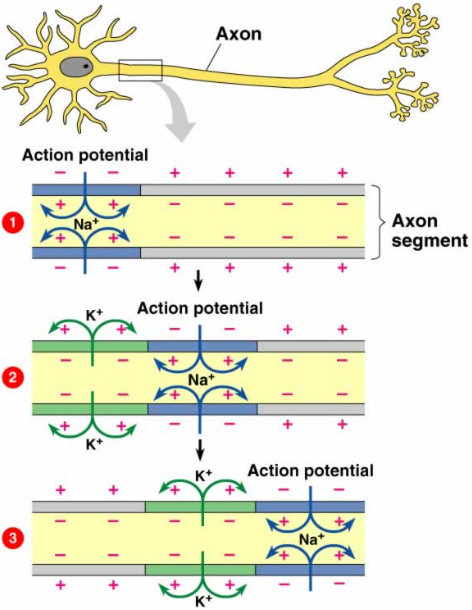


Outline

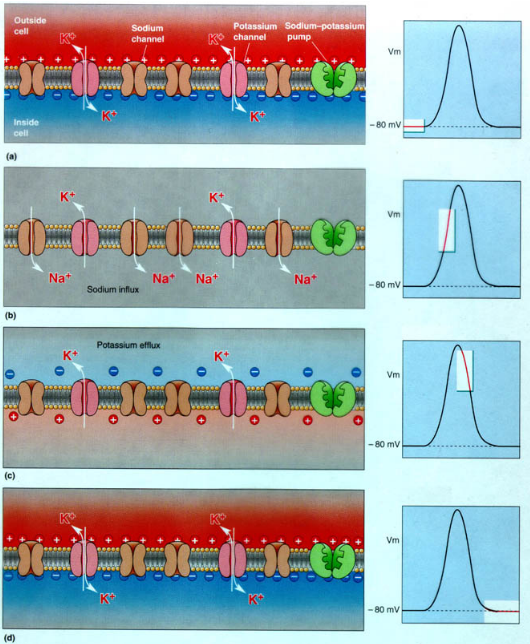
Transmission



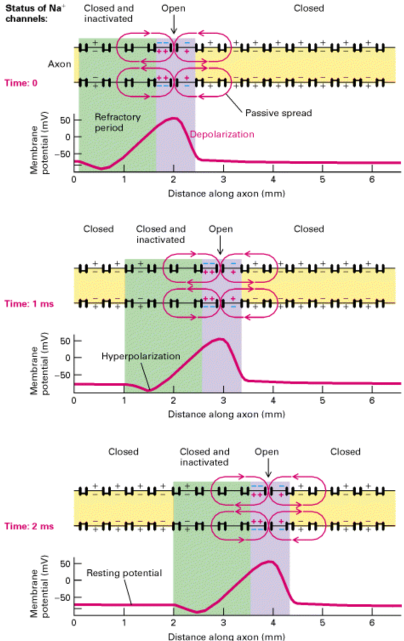
Transmission



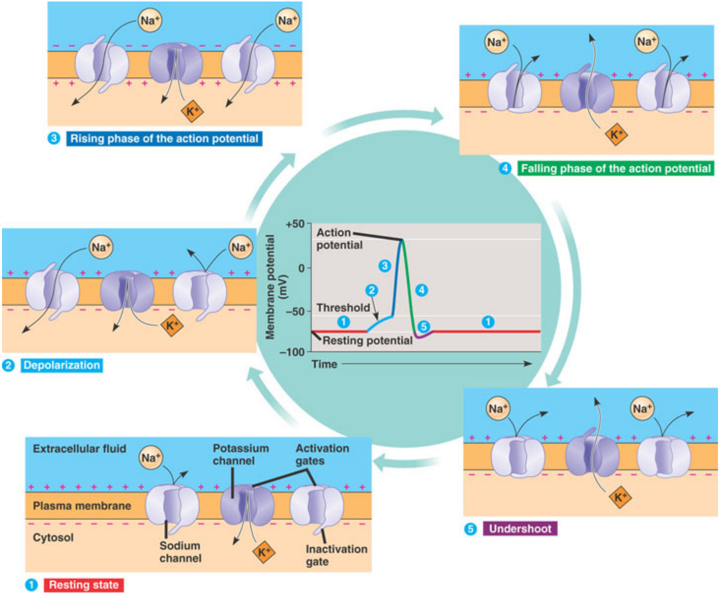
Transmission



Transmission

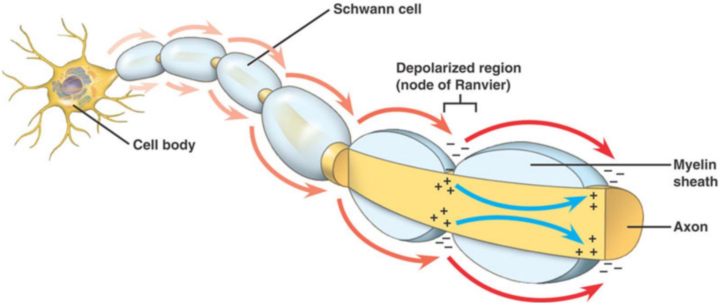


Channels

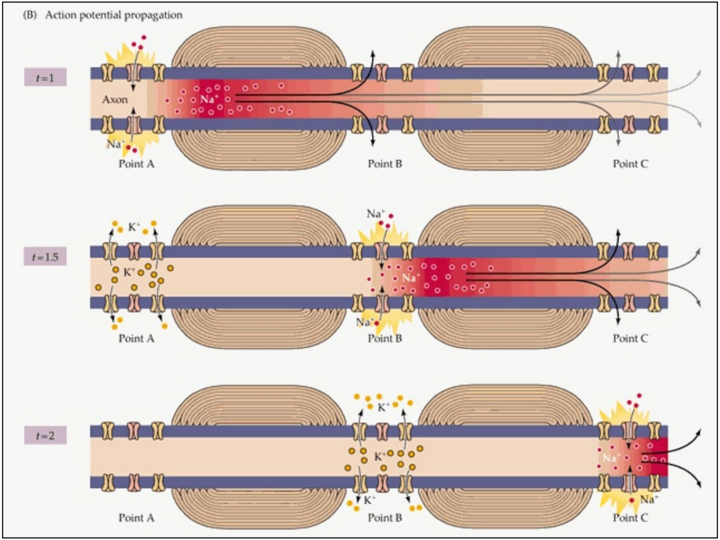


Na+
K+
3 Rising phase of the action potential
4 Falling phase of the action potential
2 Depolarization
5 Undershoot
1 Resting state
+50
0
−50
−100
Membrane potential (mV)
Action potential
Threshold
Resting potential
Time
Extracellular fluid
Potassium channel
Activation gates
Plasma membrane
Cytosol
Sodium channel
Inactivation gate

Saltatory conduction



Saltatory conduction



Outline

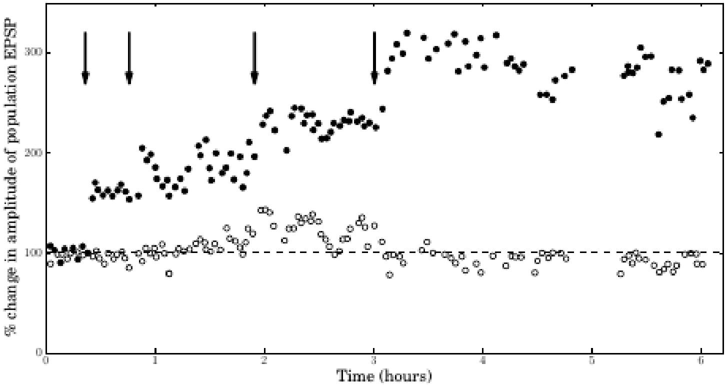
Neuron learning

- Neurons are the basis of learning, consciousness, etc.
- Neurons change their reactivity and "weights" to learn
  - Hebbian learning (ire together wire together)
- Long Term Potentiation / Depression (LTP/LTD)
- Short Term Potentiation / Depression (STP/STD)
- Glial learning ? (Human-mouse graft study!)
- Dopamine-induced reinforcement
- more????

Neuron learning

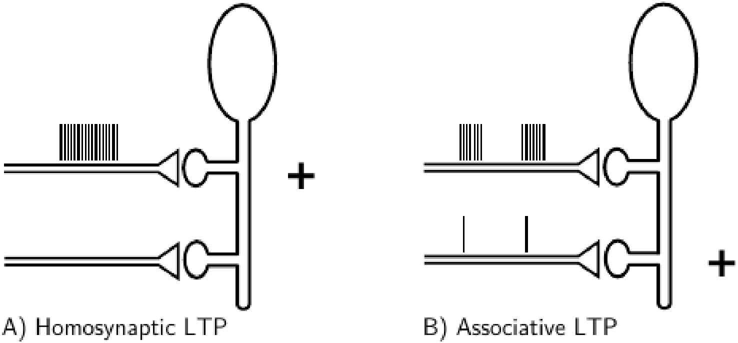
Phenomenon	Duration	Locus of induction
Short-Term Enhancement		
Paired-Pulse Facilitation (PPF)	100 ms	Pre
Augmentation	10 s	Pre
Post-Tetanic Potentiation (PTP)	1 min	Pre
Long-Term Enhancement		
Short-Term Potentiation (STP)	15 min	Post
Long-Term Potentiation (LTP)	>30 min	Pre and post
Depression		
Paired-Pulse Depression (PPD)	100 ms	Pre
Depletion	10 s	Pre
Long-Term Depression (LTD)	>30 min	Pre and post

LTP



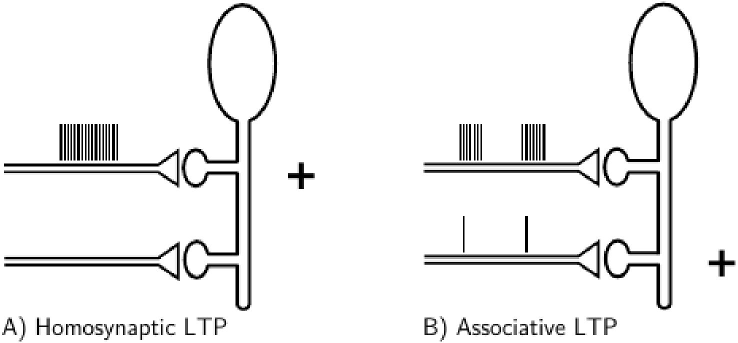
Black dots represent the EPSP of neurons in the stimulated pathway, white dots represent the EPSP in the unstimulated pathway. Tetanic stimulation was delivered at each arrow.

LTP



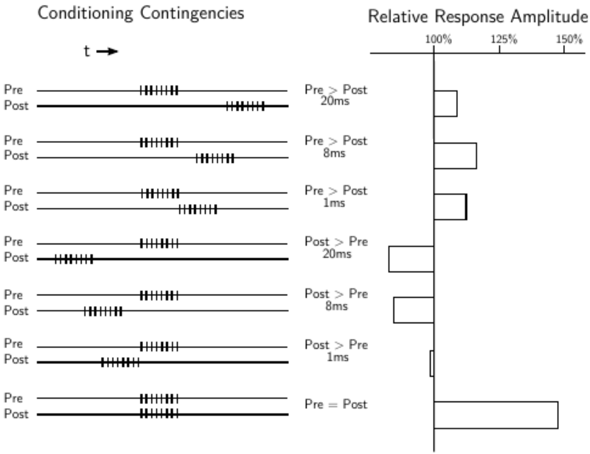
Cartoon depiction of the two classical methods of inducing LTP. Vertical lines do not depict spikes, but pulses applied to the presynaptic cell. + marks show which synapse is potentiated. (A) Homosynaptic (synapse-specific) LTP is induced by high-frequency tetanic stimulus (usually 100Hz for 1 second) of the presynaptic cell. (B) Associative LTP is induced by pairing a tetanic stimulus in one or more presynaptic cells with a low-frequency (usually 5Hz) stimulus in the presynaptic cell whose synapse is to be potentiated. Note that typically the synapses stimulated with the tetanic stimulation will also be potentiated.

LTD



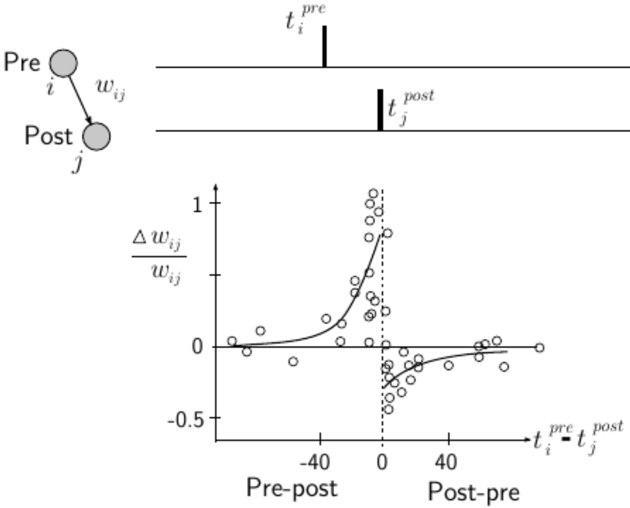
Cartoon depiction of the two classical methods of inducing LTD. Vertical lines do not depict spikes, but pulses applied to the presynaptic cell. - marks show which synapse is depressed. (A) Heterosynaptic LTD is induced with tetanic stimulation in some presynaptic cells; those that are not stimulated may become depressed. Note that the stimulated cells often have their synapses potentiated. (B) Homosynaptic LTD is induced with long period of low-frequency stimulation (typically 1 Hz for 10 minutes) of the presynaptic cell.

Order matters



Evidence that the temporal order of pre- and postsynaptic stimulation affects the induction of LTP/LTD. (Left) The stimulation protocol. Each vertical line represents a pulse of current. (Right) The ratio of the amplitude of the EPSP before the stimulation protocol and 20 minutes after the stimulation protocol. Note that depression happens when postsynaptic neurons are stimulated before presynaptic neurons, potentiation when presynaptic neurons are stimulated before postsynaptic neurons, and strong potentiation occurs when they are simultaneously stimulated.

Spike-timing dependent plasticity



The STDP curve. Each dot represents the relative change in synaptic strength after 60 pre-post or post-pre spike pairings.

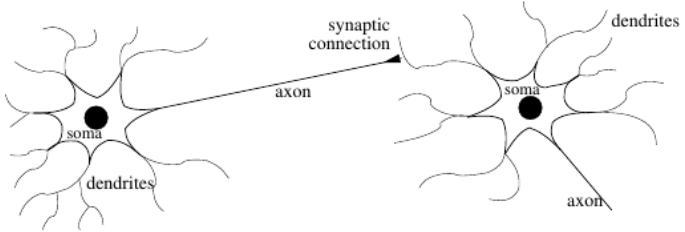
Outline

Neuron modeling

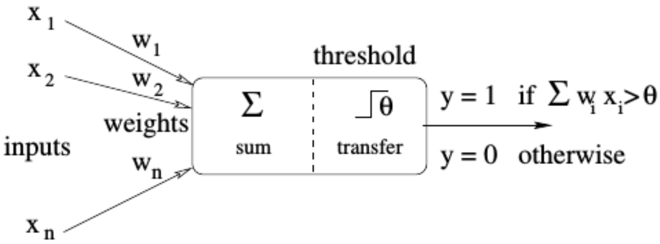
- Spike transmission
- Spike integration
- Thresholding
- etc.

Outline

Early artificial neuron models

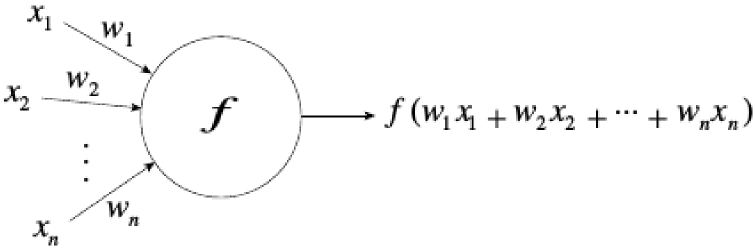


Elementary scheme of biological neurons



First mathematical model of artificial neuron

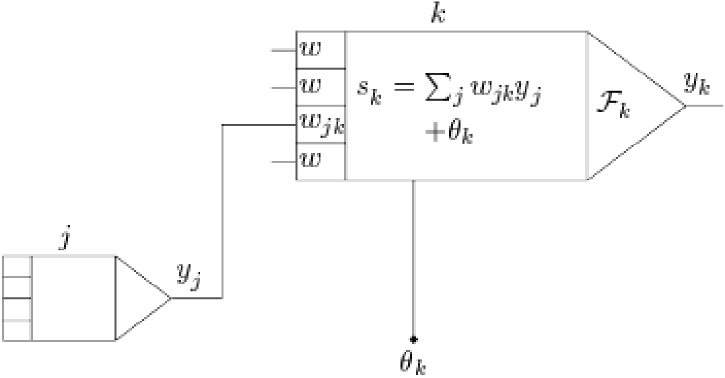
Basic neuron model



Neuron operations:

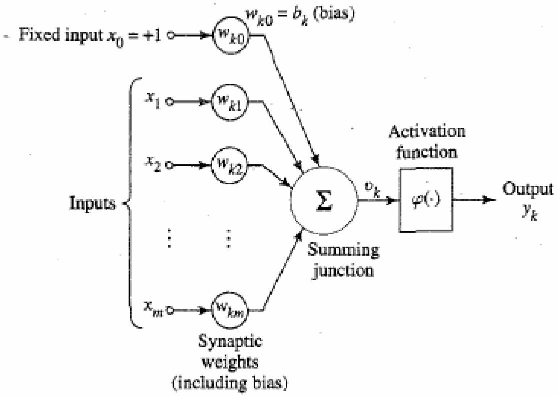
1. Sum (inputs x weights)
2. Apply activation function
3. Transmit signal

Basic neuron model



- Often a bias θ can be applied/learned

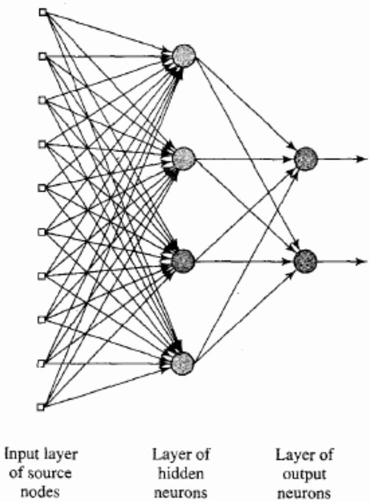
Basic neuron model



$$v_k = \sum_{j=0}^{m} w_{kj} x_j$$

$$y_k = \varphi(v_k)$$

Multi-layer feed forward network



Feed-forward networks with omniscient top-down knowledge are good at static feature extractions (e.g., AlphaGo)

From-scratch backprop beats you at digit class:

With no fancy libraries, from scratch, and this is the entire source (minus a 3-4 line main)

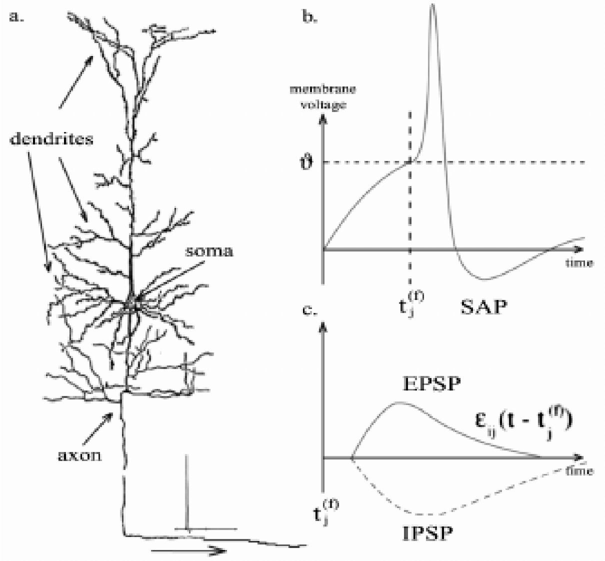
```
def sigmoid(x):
    return 1.0/(1.0 + np.exp(-x))

def sigmoid_prime(x):
    return sigmoid(x)*(1.0-sigmoid(x))

class NN:
    def __init__(self, nI=3, nH=4, nO=1):
        self.syn1 = 2 * np.random.random((nH, nO)) - 1

    def runNN(self, X):
        self.l0 = X
        self.l1 = sigmoid(np.dot(self.l0, self.syn0))
        self.l2 = sigmoid(np.dot(self.l1, self.syn1))
        return self.l2

    def backPropagate(self, y, N):
        l2_error = y - self.l2
        l2_delta = l2_error * sigmoid_prime(self.l2)
        l1_error = l2_delta.dot(self.syn1.T)
        l1_delta = l1_error * sigmoid_prime(self.l1)
        self.syn1 += (self.l1.T.dot(l2_delta)) * N
        self.syn0 += (self.l0.T.dot(l1_delta)) * N

    def train(self, X, y, max_iterations=10000, N=0.8):
        for roundNum in range(max_iterations):
            self.runNN(X)
            self.backPropagate(y, N)

    def test(self, X, y):
        final_prediction = self.runNN(X)
        return np.mean((final_prediction - y)**2)
```

What are brains for?

- "Deep" convolutional feed forward networks tend to be good for static feature extractions
- Recurrent networks tend to be better for control and time series

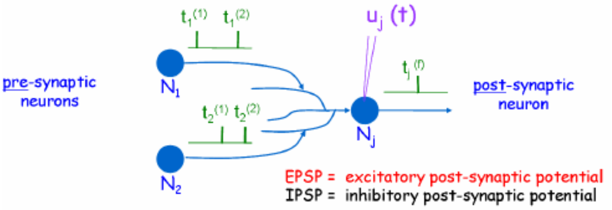
Outline

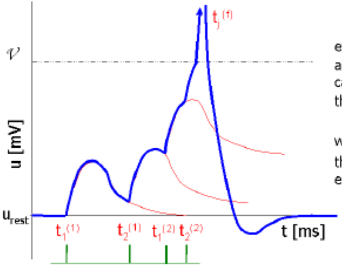
EPSP and IPSP



(a) Schematic drawing of a neuron. (b) Incoming postsynaptic potentials alter the membrane voltage so it crosses threshold value theta; the neuron spikes and goes into a refractory state. (c) Typical forms of excitatory and inhibitory postsynaptic potentials over time

Spike summation, spiking, hyper-polarization





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

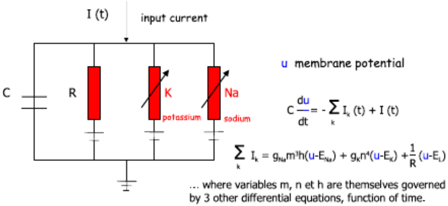
when u_j (t) reaches the threshold $\mathcal{V}$ the post-synaptic neuron N_j emits a spike, in its turn.

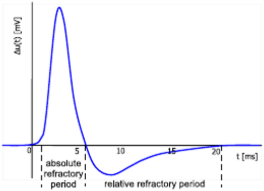
Outline

Hodgkin-Huxley

$$C\frac{du}{dt} = -g_{Na}m^3h(u-E_{Na}) - g_K n^4(u-E_K) - g_L(u-E_L) + I(t) \qquad (1)$$

$$\tau_n\frac{dn}{dt} = -[n-n_0(u)], \quad \tau_m\frac{dm}{dt} = -[m-m_0(u)], \quad \tau_h\frac{dh}{dt} = -[h-h_0(u)]$$



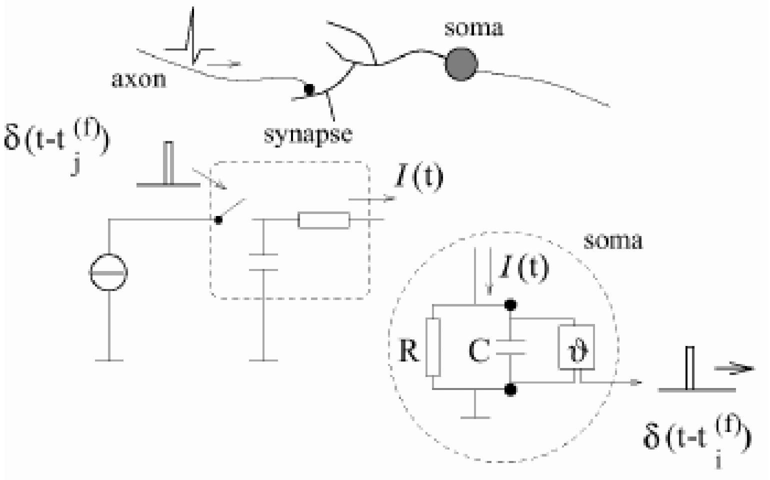


Dynamics of spike firing

Accurate in dynamics, but computationally inefficient, to the point of being useless for everything but validating faster models or bio-models...

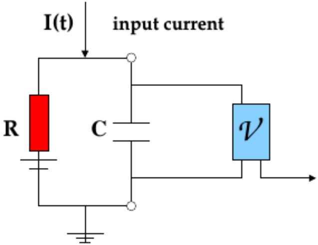
Outline

Integrate and fire



Schematic drawing of the integrate-and-fire neuron. On the left side, the low-pass filter that transforms a spike to a current pulse I(t) that charges the capacitor. On the right, the schematic version of the soma, which generates a spike when voltage u over the capacitor crosses threshold.

Integrate and fire



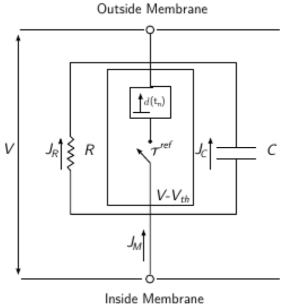
u being the membrane potential,

$$C\frac{du}{dt} = -\frac{1}{R}(u(t) - u_{rest}) + I(t)$$

spike firing time $t^{(f)}$ is defined by

$$u(t^{(f)}) = \vartheta \quad \text{with} \quad u'(t^{(f)}) > 0$$

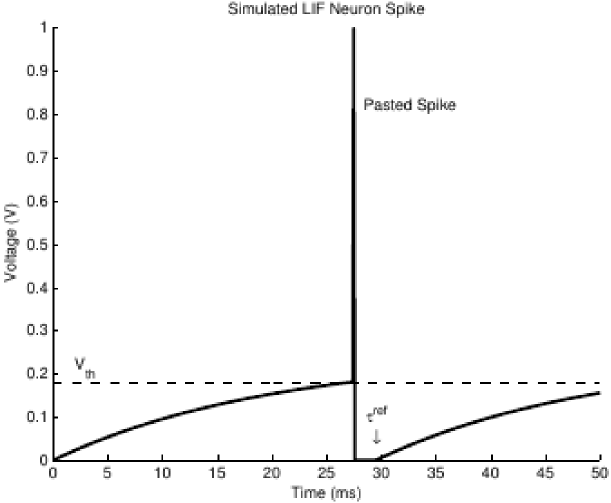
Leaky Integrate and fire (LIF)



Circuit diagram that corresponds to the leaky integrate-and-fire (LIF) neuron

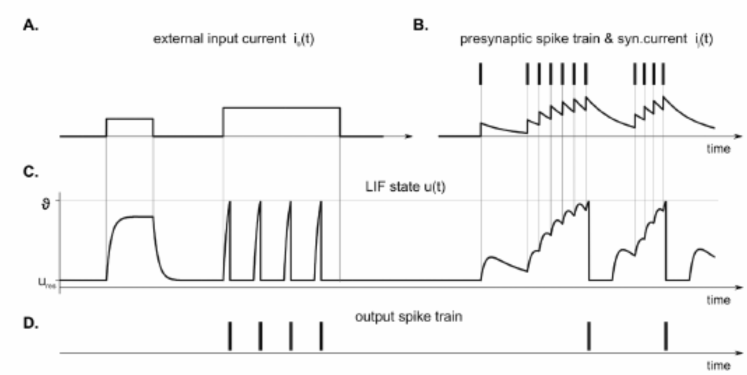
- $J(t)$ is weighted sum of inputs
- R is resistance
- C is membrane capacitance
- $V(t)$ is voltage at time t
- And the model is: $\frac{dV}{dt} = -\frac{1}{RC}(V(t) - J(t)R)$

Leaky Integrate and fire (LIF)



Membrane voltage of a LIF neuron with constant input J.

Leaky Integrate and fire (LIF)



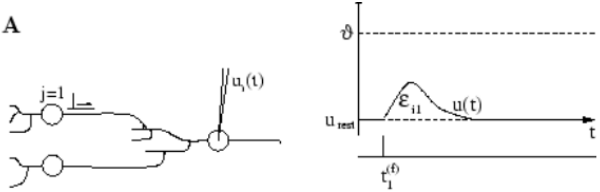
Time course of the membrane potential u(t) of a leaky-integrate-and-fire neuron LIF (panel C) driven by the external input current io(t) (shown in panel A) or by the synaptic current ij (t) evoked by the sample presynaptic spike train (panel B). Initially, the state u(t) of the LIF neuron is at the resting value ures. The currents io(t) and ij(t) increase the membrane potential towards the firing threshold theta. Whenever the threshold is crossed the neuron emits a spike and the membrane voltage u(t) is reset to a new value - here assumed ures. The firing times of the LIF neuron are shown as vertical bars in panel D.

Spike response model (SRM)

Potential: $$u_i(t) = \eta(t - \hat{t}_i) + \sum_j \sum_f \epsilon_{ij}(t - t_j^{(f)}) + u_{rest}$$

---

spike (stereotyped): $$\eta(t - t_i^{(f)})$$

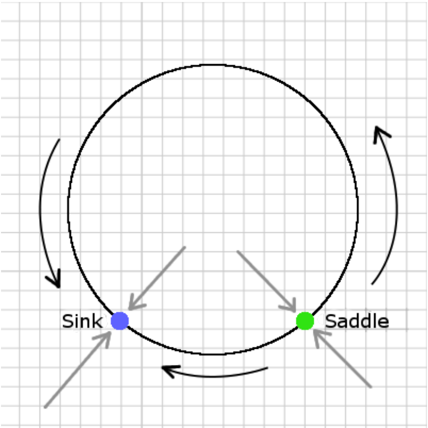


EPSP:

---

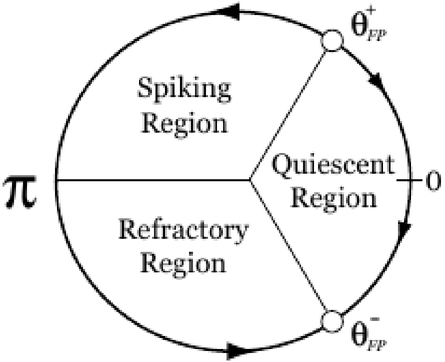
spike-condition: $$u_i(t) = \vartheta \text{ and } \frac{\mathrm{d}}{\mathrm{d}t} u_i(t) > 0 \implies t = t_i^{(f)}$$

Theta neuron



Dynamics of the theta model on the unit circle. Blue denotes a stable fixed point; Green denotes an unstable fixed point. By varying the input parameter, the two equilibria collide and form a stable limit cycle; Gray arrows indicate that the points are attracting in $\mathbb{R}^2$; Black arrows indicate the direction of movement along the unit circle

Theta neuron



The phase-trajectory in a Theta-neuron evolves according to:

$$\frac{d\theta}{dt} = (1 - \cos(\theta)) + \alpha I(t)(1 + \cos(\theta)),$$

where theta is the neuron phase, alpha is a scaling constant, and I(t) is the input current. The main advantage of the Theta-neuron model is that neuronal spiking is described in a continuous manner, allowing for more advanced gradient approaches

Ishikevich neuron

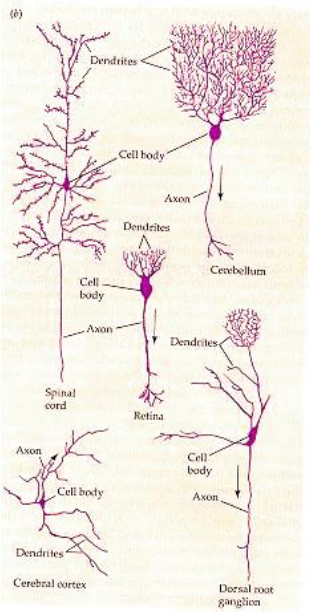
$$\frac{du}{dt} = 0.04u(t)^2 + 5u(t) + 140 - w(t) + I(t) \qquad \frac{dw}{dt} = a\left(bu(t) - w(t)\right) \tag{4}$$

with after-spike resetting: if $u \geq \vartheta$ then $u \leftarrow c$ and $w \leftarrow w + d$

Efficient enough, ran 100 million neuron simulation

Outline

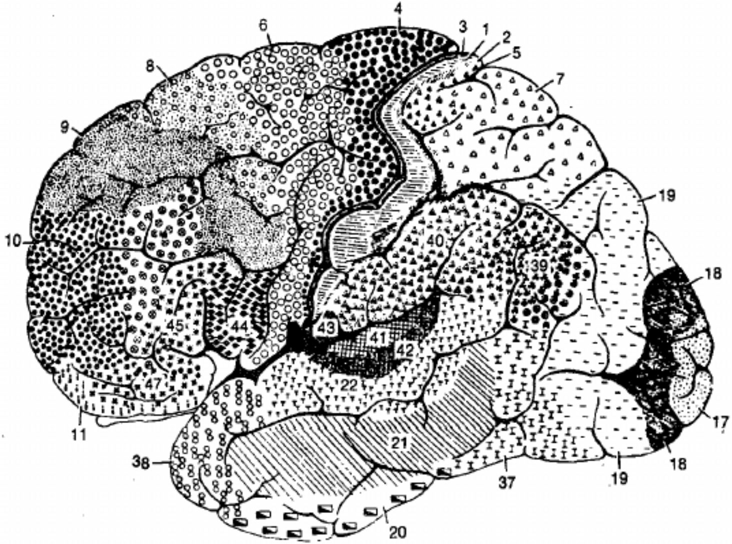
Diversity of neuron types



"What magical trick makes us intelligent? The trick is that there is no trick. The power of intelligence stems from our vast diversity (and size), not from any single, perfect principle." (Marvin Minsky, Society of Mind; 1987)

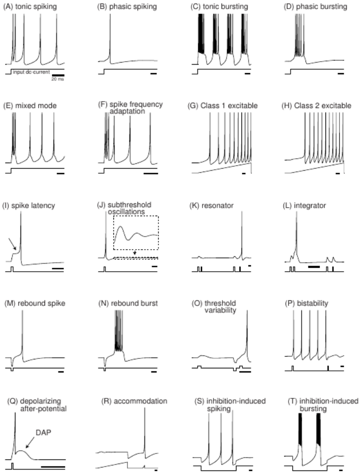
Diversity of neuron types cont...

Real neurons
Connections
Synapses
Action potentials
Transmission
Learning
Models
Original models
Modeling spikes
Detailed models
Efficient models
Comparisons
Coding
Plastic/Learn
Rate-based
Timing-based
Task learning
Rate vs. Timing
Supervised
Unsupervised
Reservoir
Reinforcement



Network structure varies on a macro scale.

Estimating variability in populations of neurons



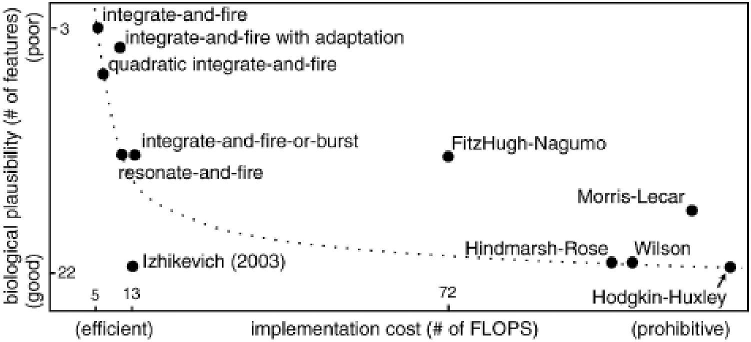
Not all neuron models can match this diversity (e.g., integrate and fire does not have refractory period)
Summary of the neuro-computational properties of biological spiking neurons. Shown are simulations of the same model (1) and (2), with different choices of parameters. Each horizontal bar denotes a 20-ms time interval.

Estimating variability in populations of neurons

Models	biophysically meaningful	tonic spiking	phasic spiking	tonic bursting	phasic bursting	mixed mode	spike frequency adaptation	class 1 excitable	class 2 excitable	spike latency	subthreshold oscillations	resonator	integrator	rebound spike	rebound burst	threshold variability	bistability	DAP	accommodation	inhibition-induced spiking	inhibition-induced bursting	chaos	# of FLOPS
integrate-and-fire	-	+	-	-	-	-	-	+	-	-	-	-	+	-	-	-	-	-	-	-	-	-	5
integrate-and-fire with adapt.	-	+	-	-	-	-	+	+	-	-	-	-	+	-	-	-	-	+	-	-	-	-	10
integrate-and-fire-or-burst	-	+	+		+	-	+	+	-	-	-	-	+	+	+	-	+	+	-	-	-		13
resonate-and-fire	-	+	+	-	-	-	-	+	+	-	+	+	+	+	-	-	+	+	+	-	-	+	10
quadratic integrate-and-fire	-	+	-	-	-	-	-	+	-	+	-	-	+	-	-	+	+	-	-	-	-	-	7
Izhikevich (2003)	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	13
FitzHugh-Nagumo	-	+	+	-		-	-	+	-	+	+	+	-	+	-	+	+	-	+	+	-	-	72
Hindmarsh-Rose	-	+	+	+			+	+	+	+	+	+	+	+	+	+	+	+	+	+		+	120
Morris-Lecar	+	+	+	-		-	-	+	+	+	+	+	+	+		+	+	-	+	+	-	-	600
Wilson	-	+	+	+			+	+	+	+	+	+	+	+	+	+		+	+				180
Hodgkin-Huxley	+	+	+	+			+	+	+	+	+	+	+	+	+	+	+	+	+	+		+	1200

Comparison of the neuro-computational properties of spiking and bursting models; "number of FLOPS" is an approximate number of floating point operations (addition, multiplication, etc.) needed to simulate the model during a 1 ms time span. Each empty square indicates the property that the model should exhibit in principle (in theory) if the parameters are chosen appropriately, but the author failed to find the parameters within a reasonable period of time.

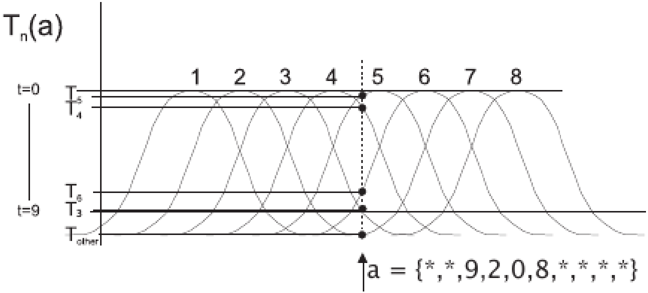
Estimating variability in populations of neurons

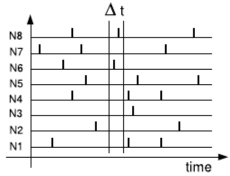


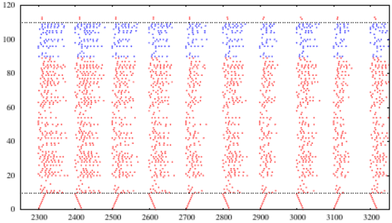
Comparison of the neuro-computational properties of spiking and bursting models; "num of FLOPS" is an approximate number of floating point operations (addition, multiplication, etc.) needed to simulate the model during a 1 ms time span. Each empty square indicates the property that the model should exhibit in principle (in theory) if the parameters are chosen appropriately, but the author failed to find the parameters within a reasonable period of time.

Outline

Input coding (sensory transducers)

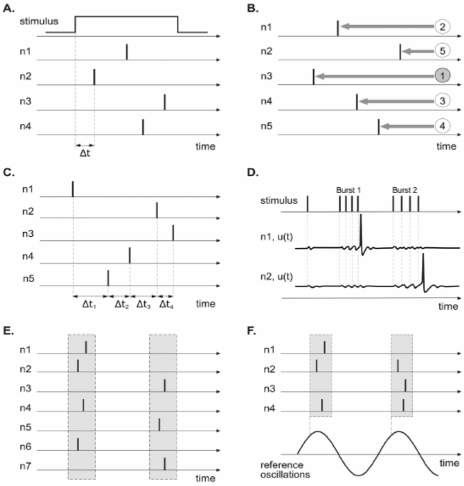






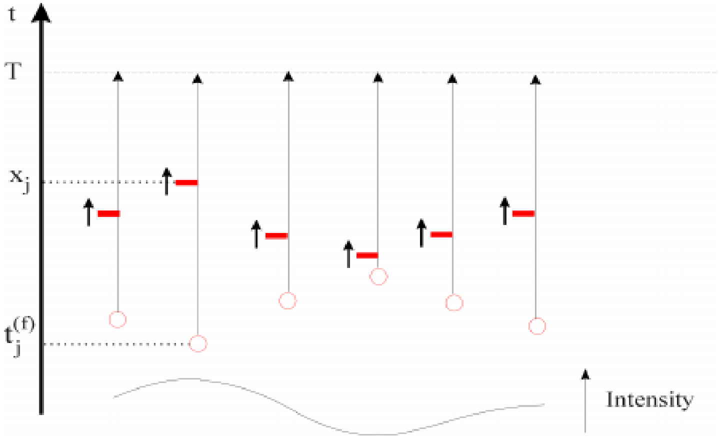
Distinction: inputs versus internals

Code types



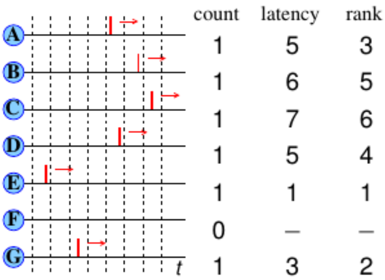
(A) time to first spike; (B) rank-coding or spike-order coding; (C) latency coding based on the exact timing of spikes; (D) resonant burst coding; (E) coding by synchrony; (F) phase coding. Legend: n1,...,n7 are the labels of neurons; the vertical bars in the particular plots represent the neural firing times; the numbers 1,...,5 in the circles indicate the order of spike arrival; Δt is the latency between the stimulus onset and the first spike; Δt1,...,Δt4 are the inter-spike latencies; u(t) is the neuron model state variable.

Code types



Coding by relative delay. The neurons in figure emit spikes at different moments (f) in time tj . The most strongly activated neuron fires first (i.e., second from left). Its spike travels a considerable distance along the axon, until last neuron fires (i.e., the fourth from left). The latencies xj are computed with respect to a reference time T

Rate coding, time coding, rank coding



Numeric examples:	count code	binary code	timing code	rank order
left (opposite) figure $n = 7, T = 7ms$	3	7	≈ 19	12.3
Thorpe et al. [164] $n = 10, T = 10ms$	3.6	10	≈ 33	21.8

Number of bits that can be transmitted

Outline

Weight updating

- The first was Hebbian: $\triangle\omega_{ij} = \kappa a_i a_j$ where:
  - ω_{ij} is the synaptic strenth of the synapse between the axon of the presynaptic neuron i and the dendrite of the poststynaptic neuron j.
  - $\triangle\omega_{ij}$ is change in synaptic strength
  - κ is the learning rate
  - a_i indexes the presynaptic neuron
  - a_j indexes the postsynapci neuron
  - "Activity" can be interpreted many different ways. The most common measure of activity is the firing rate of a neuron. However, membrane voltage, spike times, the amount of current flowing into the cell, filtered spike trains, and other measures can be interpreted as activity.

Outline

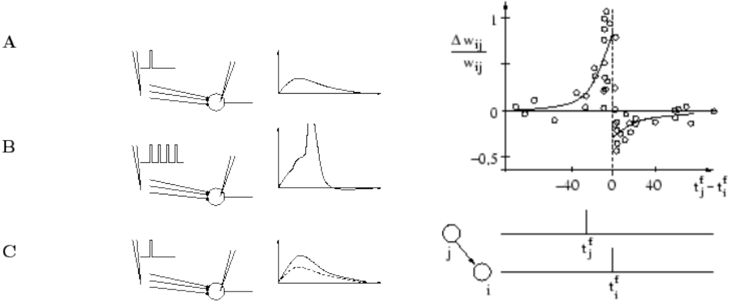
Various rate-schemes

$$\frac{\mathrm{d}}{\mathrm{d}t} w_{ij} = F(w_{ij}; \nu_i, \nu_j)$$

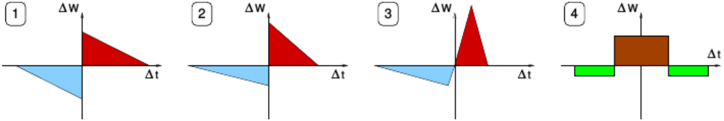
post ν_i	pre ν_j	$\mathrm{d}w_{ij}/\mathrm{d}t \propto \nu_i \nu_j$	$\mathrm{d}w_{ij}/\mathrm{d}t \propto \nu_i \nu_j - c_0$	$\mathrm{d}w_{ij}/\mathrm{d}t \propto (\nu_i - \nu_\theta) \nu_j$	$\mathrm{d}w_{ij}/\mathrm{d}t \propto \nu_i (\nu_j - \nu_\theta)$	$\mathrm{d}w_{ij}/\mathrm{d}t \propto (\nu_i - \langle \nu_i \rangle)(\nu_j - \langle \nu_j \rangle)$
ON	ON	+	+	+	+	+
ON	OFF	0	−	0	−	−
OFF	ON	0	−	−	0	−
OFF	OFF	0	−	0	0	+

Outline

Learning: Biology of STDP



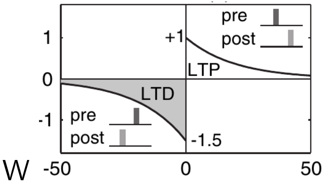
Model:



STDP (idealized, rather than online version)

$$\Delta w_j = \sum_{f=1}^{N} \sum_{n=1}^{N} W(t_i^n - t_j^f)$$

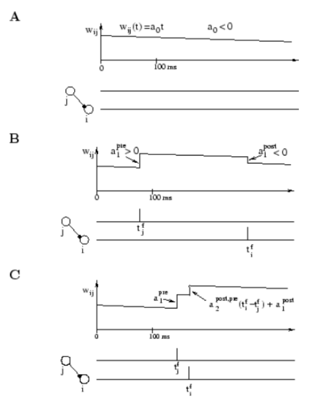
- Δw_j is synapse weight changefrom a presynaptic neuron j to i
- t_j^f with $f = 1, 2, 3, \ldots$ indexes presynaptic spikes
- t_j^n with $n = 1, 2, 3, \ldots$ indexes firing times of the postsynaptic neuron
- $W(x) = A_+ \exp(-x/\tau_+)$ *for* $x > 0$ (LTP curve below)
- $W(x) = -A_- \exp(x/\tau_-)$ *for* $x < 0$ (LTD curve below)
- A_+ and A_- constant changing amplitude
- $x = post - pre$
- $\tau_{+/-} = 10ms$ are a time constants



STDP Formulation

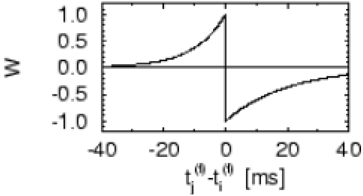
$$\frac{d}{dt} w_{ij}(t) = a_0 + S_j(t)\left[a_1^{pre} + \int_0^\infty a_2^{pre,\,post}(s)\, S_i(t-s)\, ds\right]$$
$$+\, S_i(t)\left[a_1^{post} + \int_0^\infty a_2^{post,\,pre}(s)\, S_j(t-s)\, ds\right],$$

$S_j(t) = \sum_f \delta(t - t_j^{(f)})$ and $S_i(t) = \sum_f \delta(t - t_i^{(f)})$ are pre- and postsynaptic spike trains



kernels $a_2^{post,\,pre}$ and $a_2^{pre,\,post}$ - inspired

$$W(s) = \begin{cases} A_+ \exp[s/\tau_1] & \text{for } s<0, \\ A_- \exp[-s/\tau_2] & \text{for } s>0, \end{cases}$$



Outline

Coincidence detectors

- Spiking networks are great at general coincidence detection, which results in many capabilities:

Outline

Analogue (**iff** hardware)>Spiking>Rate=artificial

- "Networks of noisy spiking neurons with temporal coding have a strictly larger computational power than sigmoidal neural nets with the same number of units."
- "In some cases, for example for stationary input, it will turn out that the spiking neuron models can be strictly reduced to rate models; in other cases such a reduction is not possible."
- "Spike-based and rate-based rules of plasticity are equivalent as long as temporal correlations are disregarded."
- "If rates vary rapidly, i.e. on the time scale of the learning window, then spike-time dependent plasticity is distinct from a rate-based formulation."

Outline

The supervised spike-time learning problem

Construct a network with arbitrary connection weights, ω. Given

- $S(t_j^d)$, the desired spike train of output neuron j, and
- $S(t_i)$, the spike train of an input neuron i,

modify ω such that

$$D\left(S(t_i), S(t_j)\right)$$

is minimized, where $D(S_1, S_2)$ is a measure of the dissimilarity between two spike trains [47, 214].

SpikeProp

- SpikeProp, operates much like traditional backpropagation in that it calculates the global error – the time difference between the spike train created by the network and the desired spike train – and assigns local error for each node, which is used to modify connection weights proportionally to the node's activity. Like backpropagation, however, the local error for each node depends on the connection weights of downstream neurons, making this algorithm biologically implausible. It also requires the network be feed-forward

SpikeProp, De-correlation backpropogation, FreqProp, ReSuMe, etc

Learning Method	Network Size	Epochs	Train	Test
Fisher Iris Dataset				
SpikeProp	50x10x3	1000	97.4%	96.1%
BP A	50x10x3	2.6e6	98.2%	95.5%
BP B	4x8x1	1e5	98.0%	90.0%
Theta Neuron BP	4x8x1	1080	100%	98.0%
Wisconsin Breast Cancer Dataset				
SpikeProp	64x15x2	1500	97.6%	97.0%
BP A	64x15x2	9.2e6	98.1%	96.3%
BP B	9x8x1	1e5	97.2%	99.0%
Theta Neuron BP	9x8x1	3130	98.3%	99.0%

SpikeProp does better than BackProp.

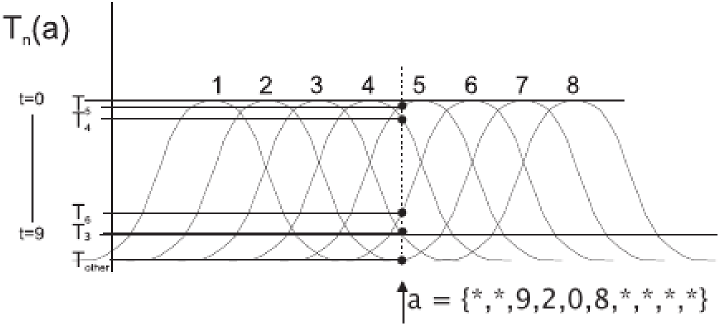
Outline

The unsupervised spike-time learning problem

Given input $\mathbf{X}$ and cost function $C(\mathbf{x}, \mathbf{y})$, generate output $\mathbf{Y}$ and minimize $C(\mathbf{x}, \mathbf{y})$.

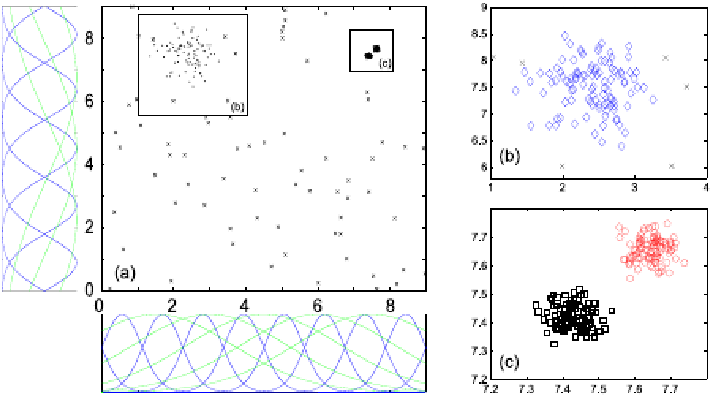
- Various rules:
  - Artola, Brocher, Singer (ABS) - rate based
  - Bienenstock, Cooper, Munro (BCM) - rate based
  - Spike-timing dependent plasticity rules - timing based
    - $\triangle\omega_{ij}(t^{pre}) = A^{-}exp(\frac{t^{post_l} - t^{pre}}{\tau^{-}})$
    - $\triangle\omega_{ij}(t^{post}) = A^{+}exp(\frac{t^{post_l} - t^{post}}{\tau^{+}})$
    - $t^{pre}$is time of a presynaptic spike
    - $t^{pre_1}$is time of the last presynaptic spike
    - t^{post} and t^{post_l} for postsynaptic
    - $A^{-}$is a negative constant representing the max amp post-pre depression
    - τ^{-} is time-constant controlling exponential decay
    - $A^{+}$and τ^{+}define positive pre-post part of the curve

Clustering



Encoding with overlapping Gaussian receptive fields. An input value a is translated into firing times for the input-neurons encoding this input-variable. The highest stimulated neuron (5), fires at a time close to 0, whereas less stimu- lated neurons, as for instance neuron 7, fire at increasingly later times.

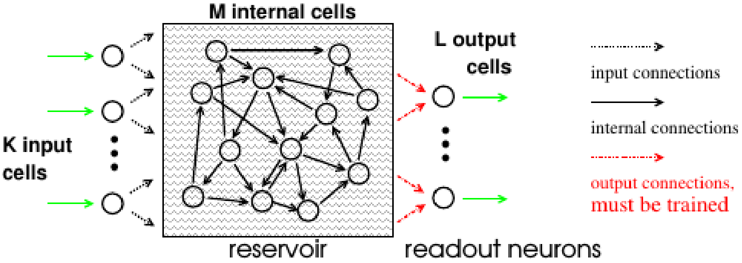
Clustering



Three clusters (upper left and upper right) of different scale with noise (crosses). (b,c) Insets: actual classification. Respective classes are marked with diamonds, squares, and circles. Noise outside the boxes or points marked by x's did not elicit a spike and were thus not attributed to a class. Side panels: graded receptive fields used.

Outline

Reservoir (supervised or reinforcement)



- Reservoir performance is improved with unsupervised STDP in reservoir.

Outline

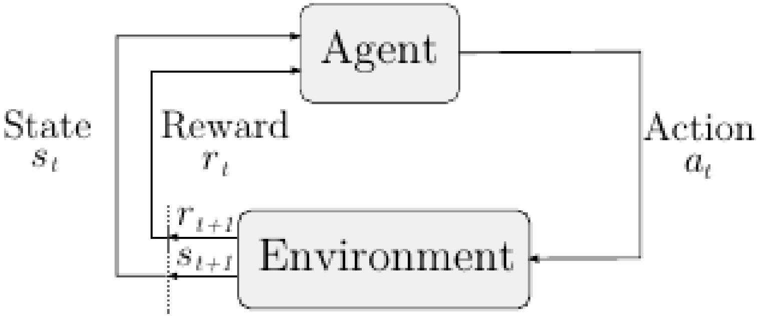
Reinforcement learning

Given some information about a Markov Decision Process,

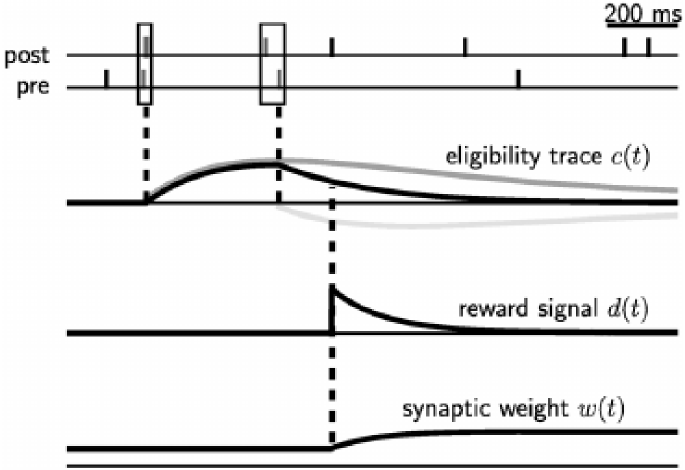
$$MDP = (\mathcal{S}, \mathcal{A}_s, \mathcal{P}_a(s, s'), \mathcal{R}_a(s, s')),$$

find a policy $\pi(s) = a$ such that following that policy maximizes

$$R = \sum_{t=0}^{\infty} \gamma^t r_{t+1}.$$

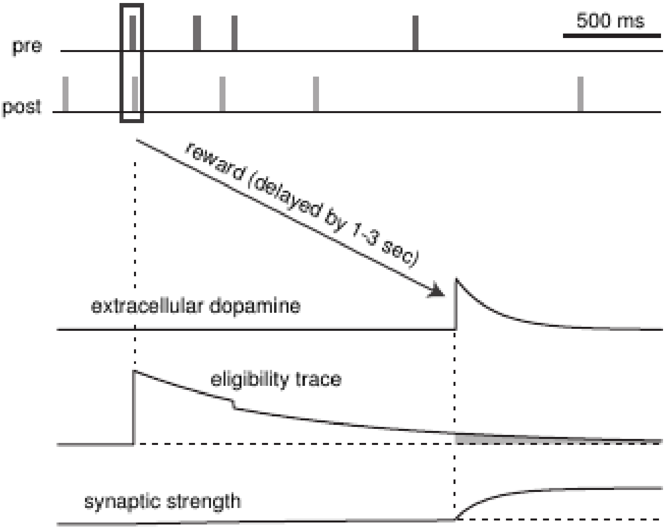


Reinforcement learning



Changes of the synaptic weight w(t) are proportional to the product of the STDP eligibility trace c(t) with the reward signal d(t). Contribution of a pre-before-post spike pair and a post-before-pre spike pair to the eligibility trace c(t) is illustrated at the top of the figure

Reinforcement learning



Eligibility trace modulates effects that would occur with normal STDP (see, for example, the dip after a post-pre pairing). The delivered reward allows the plasticity to occur, raising the synaptic strength

Implementation of R-STDP variants

Various ways to make a network do R-STDP:

- Neuron-level (each neuron integrates reinforcement, like dopamine) with various flavors within this set
- Actor-Critic: Population level (e.g., neuron compiler)

Variants on STDP:

- Triplets, etc