Spiking Neuron Networks A Survey

Article by Helene Paugam-Moisy
Presentation by Jeremy Wurbs
May 3, 2010

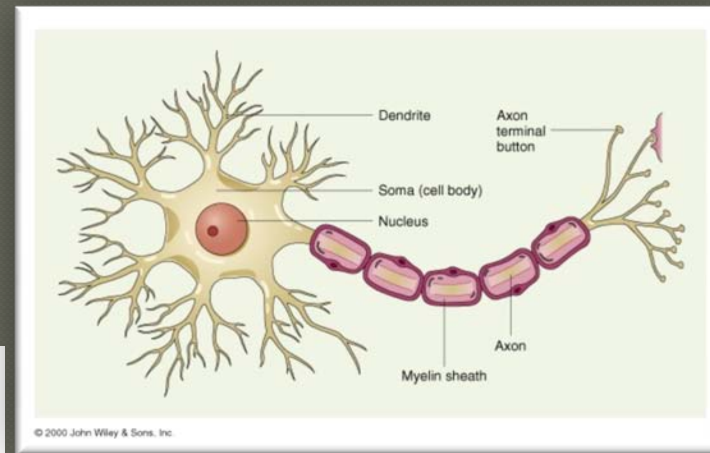
Overview

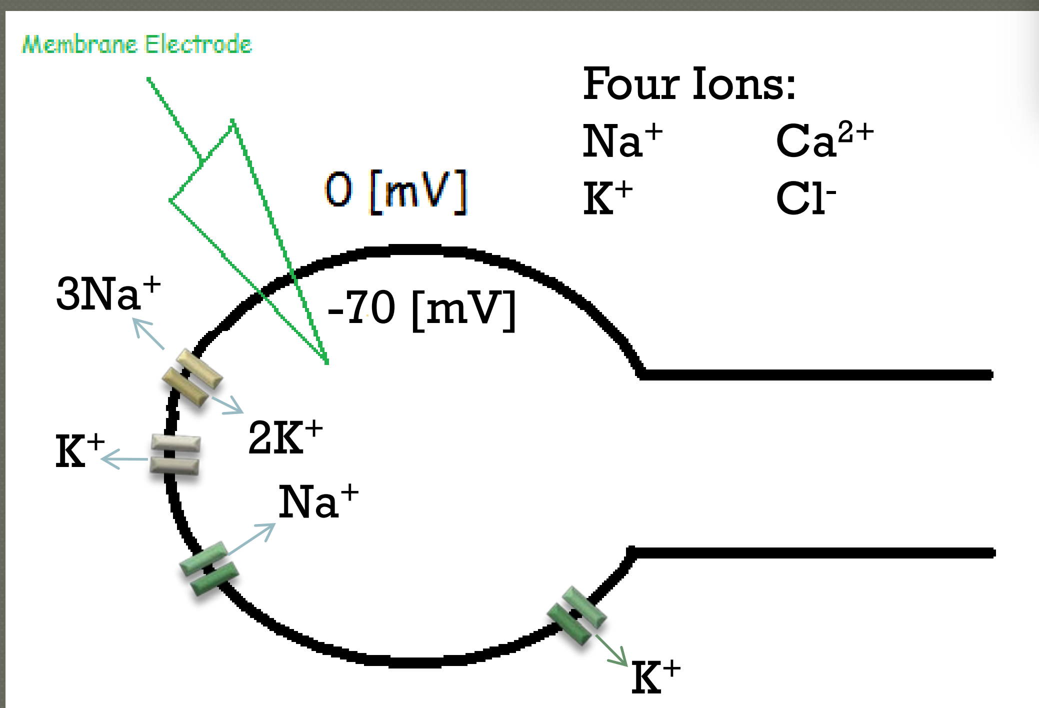
- Motivation
- Biology
- SNN Models
- Temporal Coding
- ESN's and LSM's
- Computational Power of SNNs
- Training/Learning with SNNs
- Software/Hardware Implementation
- Applications
- Discussion

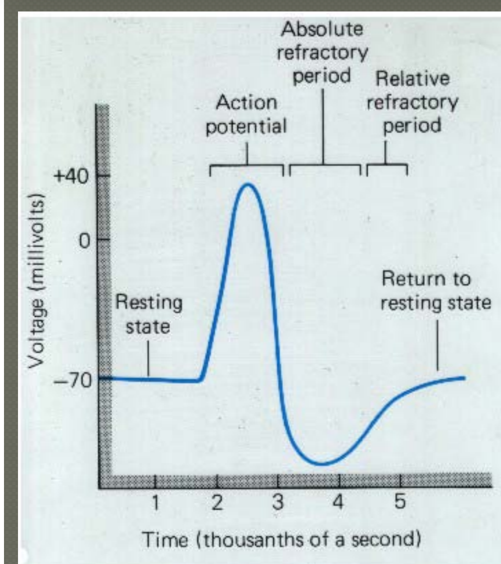
Neural Network Classification

- 1st Generation:
  - Perceptrons, Hopfield Networks, MLP with threshold units
- 2nd Generation:
  - Networks with non-linear activation units and real-valued, continuous set of output units
- 3rd Generation:
  - Spiking neuron networks, using firing times of neurons for information encoding

The Biological Neuron

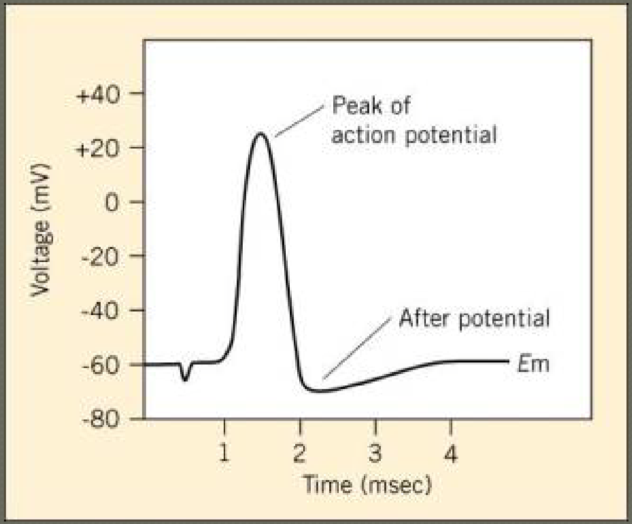


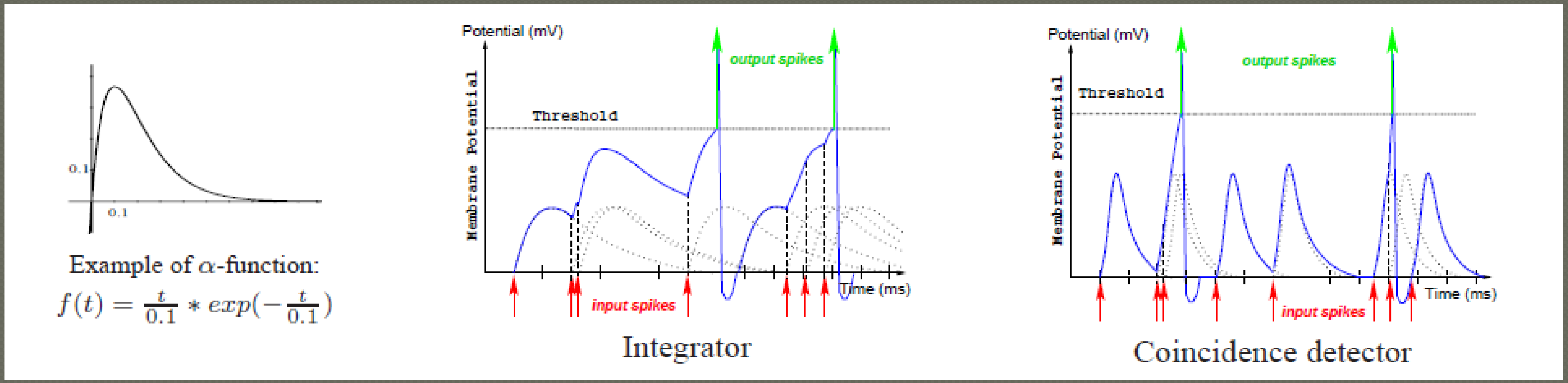




Traditional Spike Representation

- Alpha Function
- Integrator
- Coincidence Detector

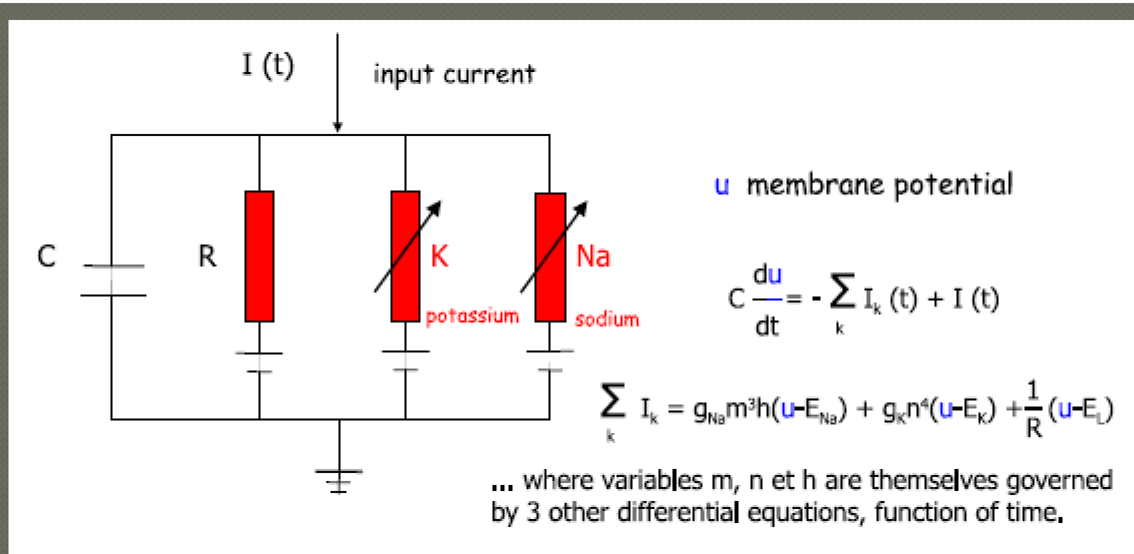




Hodgkin-Huxley Model

- Models membrane potential
  - Conductance-based
  - Defined in 1952 (Note: Na-K Pump disc. in 1957)

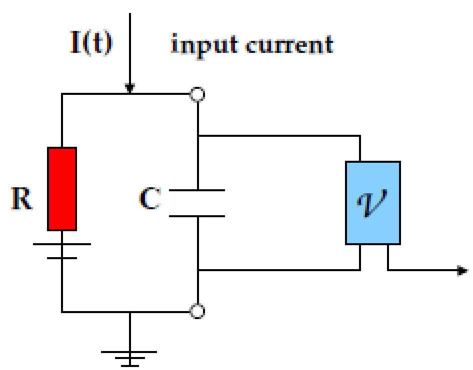
$$C\frac{du}{dt} = -g_{Na}m^3h(u-E_{Na}) - g_K n^4(u-E_K) - g_L(u-E_L) + I(t)$$



Leaky Integrate & Fire Model

- Considers spike as event
- Ions leak out, requiring time constant, τ

$$\tau_m \frac{du}{dt} = u_{rest} - u(t) + RI(t)$$



$\mathbf{u}$ membrane potential

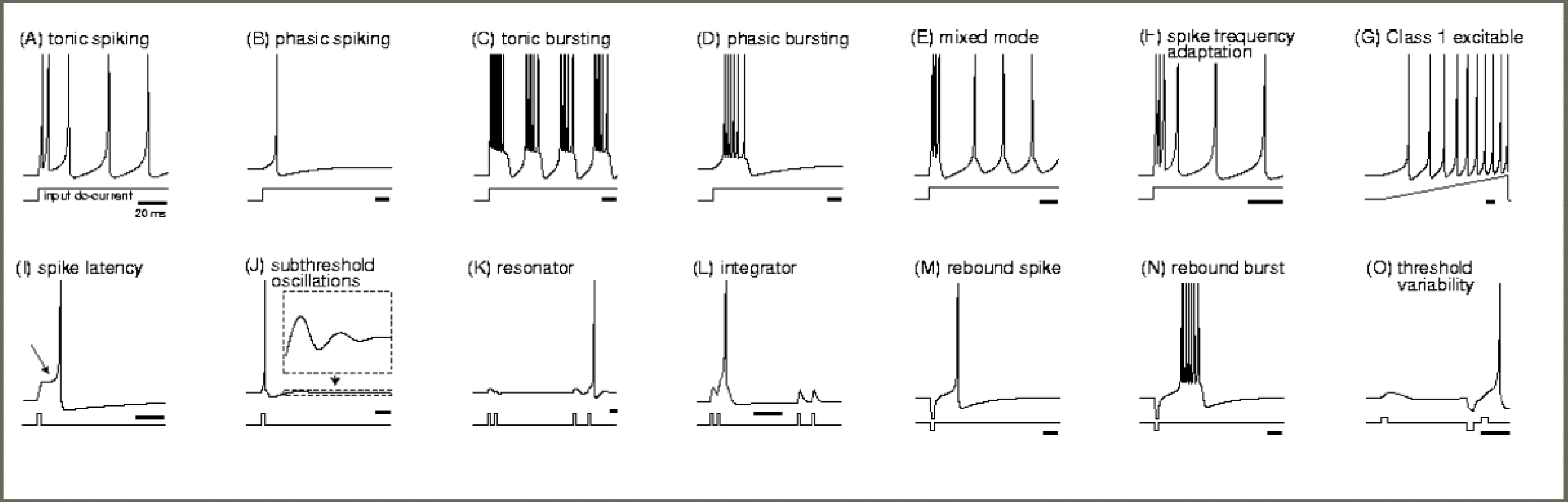
$$C\frac{d\mathbf{u}}{dt} = -\frac{1}{R}\mathbf{u}(t) + I(t)$$

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

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

Izhikevhich's Firing Behaviors

- 20 Possible Neuron Firing Behaviors
- LIF can only accommodate 3 (A,G, & L)



Izhikevich Neuron Model

- Two variables
  - Voltage Potential (*v*)
  - Membrane Recovery (activation of K currents and inactivation of Na currents) (*u*)
  - W is the weighted input(s), *a, b, c* & *d* are abstract parameters of the model

$$\frac{du}{dt} = a(bv - u)$$

$$\frac{dv}{dt} = .04v^2 + 5v + 140 - u + W$$

- When (*v* > threshold), *v* and *u* are reset:

$$v \rightarrow c$$
$$u \rightarrow u + d$$

Spike Response Model

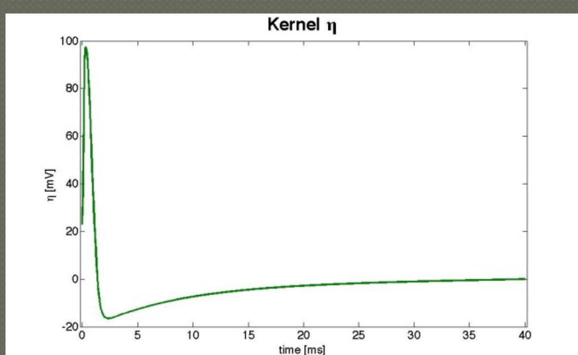
- Adds a refractory period

$$u_j(t) = \sum_{t_j^{(f)} \in \mathcal{F}_j} \eta_j \left(\underbrace{t - t_j^{(f)}}_{s}\right) + \sum_{i \in \Gamma_j} \sum_{t_i^{(f)} \in \mathcal{F}_i} w_{ij} \epsilon_{ij} \left(\underbrace{t - t_i^{(f)}}_{s}\right) + \underbrace{\int_0^{\infty} \kappa_j(r) I(t-r) dr}_{\text{if external input current}} \tag{4}$$

Spike & Spike Reset

Weighted Sum of Inputs

External Current



$$\eta_j(s) = -\eta_0 \, exp\left(-\frac{s - \delta^{abs}}{\tau}\right) \mathcal{H}(s - \delta^{abs}) - K \mathcal{H}(s) \mathcal{H}(\delta^{abs} - s)$$

$$\eta_j(s) = -\vartheta \, exp\left(-\frac{s}{\tau}\right) \mathcal{H}(s)$$

$$\epsilon_{ij}(s) = \frac{s - d_{ij}^{ax}}{\tau_s} \, exp\left(-\frac{s - d_{ij}^{ax}}{\tau_s}\right)$$

$$\epsilon_{ij}(s) = \left[exp\left(-\frac{s - d_{ij}^{ax}}{\tau_m}\right) - exp\left(-\frac{s - d_{ij}^{ax}}{\tau_s}\right)\right] \mathcal{H}(s - d_{ij}^{ax})$$

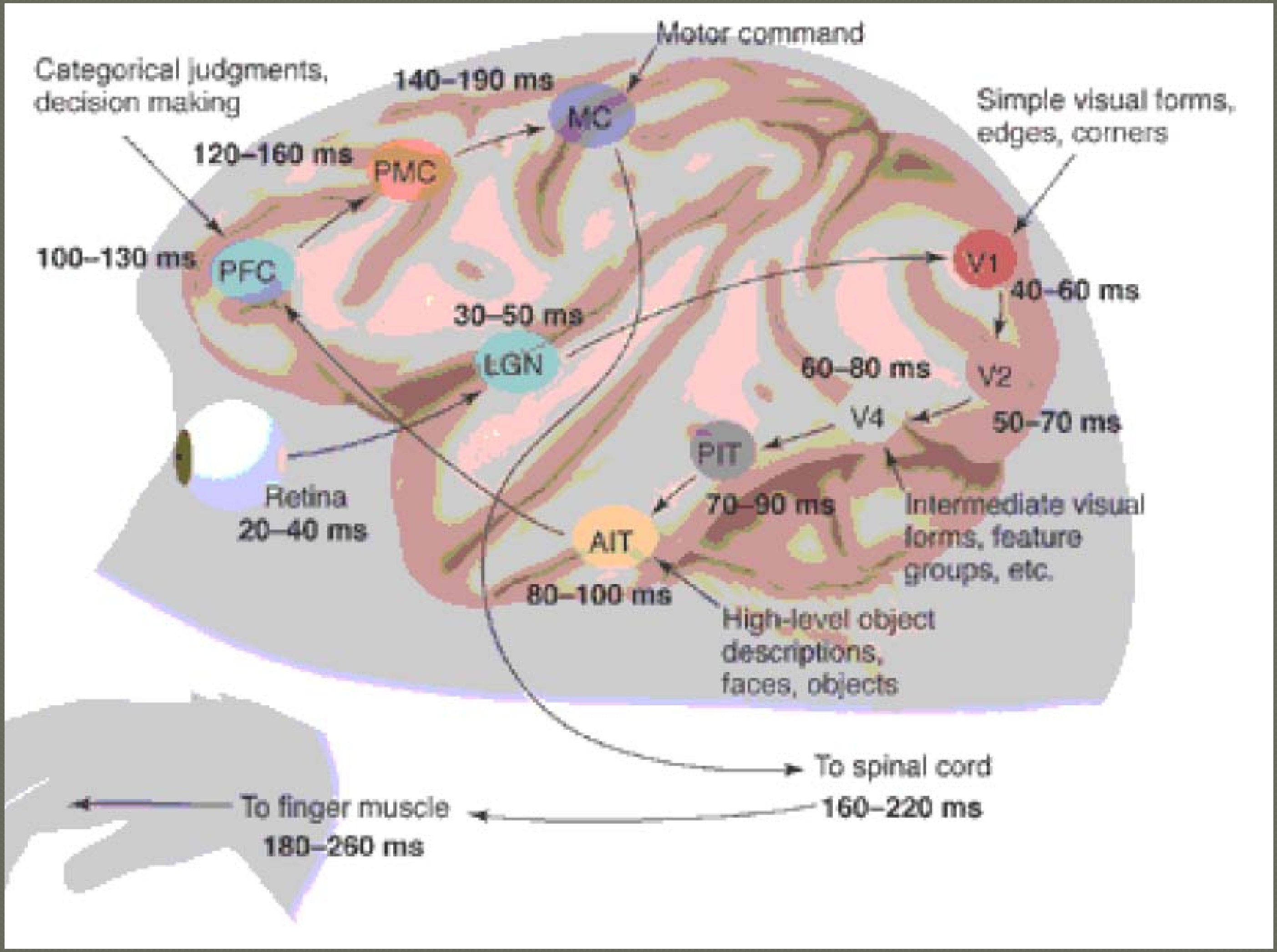
Model Advantages

- Hodgkin-Huxley
  - Accurate Modeling
  - Predicts membrane potentials due to pharmacological blocking of ion channels
- Integrate & Fire
  - Easy implementation
  - Computation-light
- Spike Response Model
  - Includes refractory phase

Rate Coding v. Temporal Coding

- Rate Coding
  - Information transmitted by rates
  - I.E. number of spikes per unit time
- Temporal Coding
  - The exact timing of spikes matter

Rate Coding v. Temporal Coding



(Trad.) Temporal Computing

$$u_j(t) = \sum_{t_j^{(f)} \in \mathcal{F}_j} \eta_j\left(t - t_j^{(f)}\right) + \sum_{i \in \Gamma_j} \sum_{t_i^{(f)} \in \mathcal{F}_i} w_{ij}\epsilon_{ij}\left(t - t_i^{(f)}\right)$$

$$\sum_{i \in \Gamma_j} w_{ij}\epsilon_{ij}\left(t_j - t_i\right) = \sum_{i \in \Gamma_j} w_{ij}\lambda_{ij}\left(t_j - t_i - \Delta_{ij}\right) = \vartheta$$

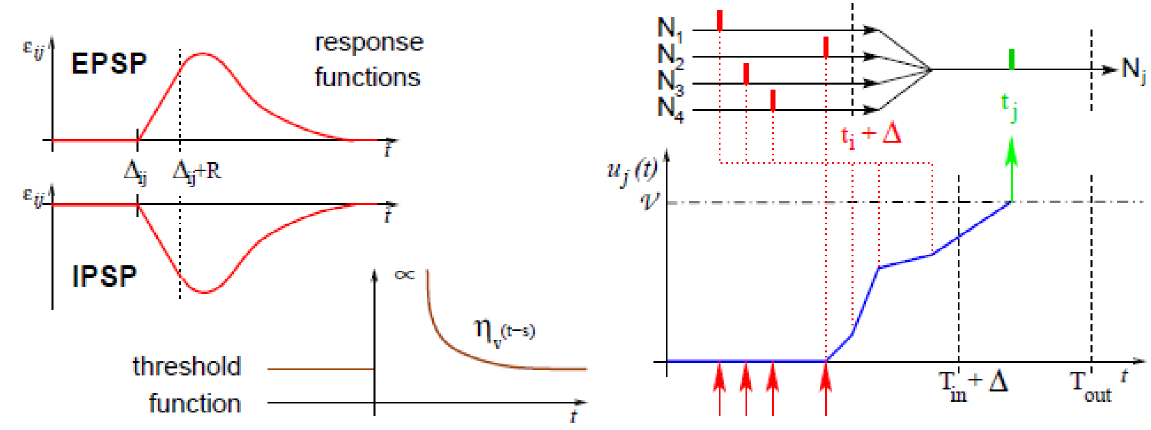


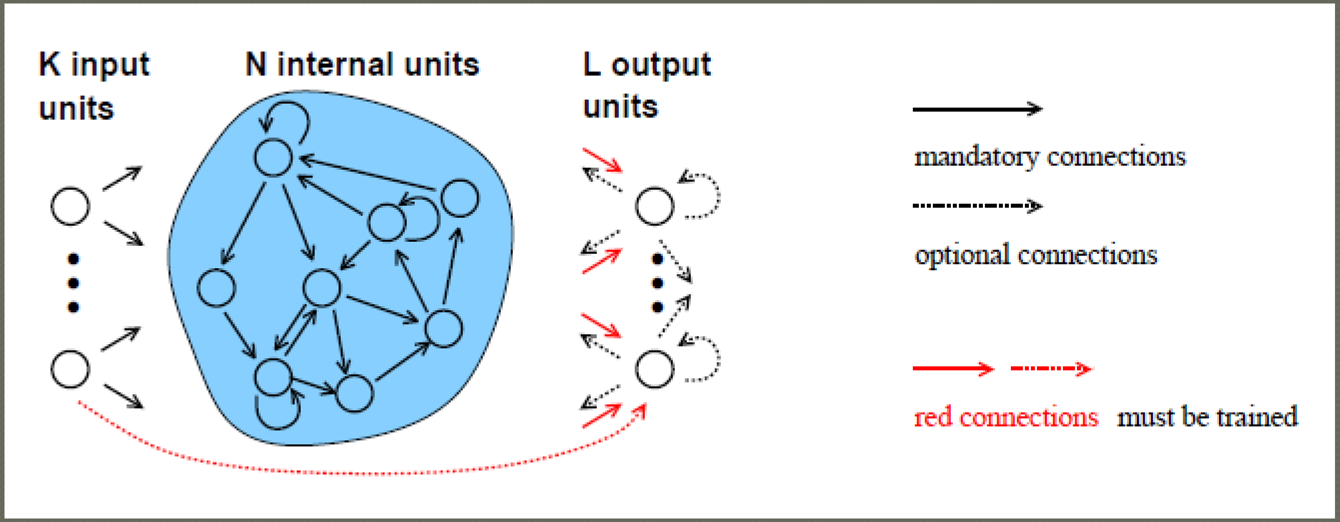
Figure 9: Shapes of postsynaptic potential (EPSP or IPSP) for computing a weighted sum in temporal coding. Right: Example variation of neuron N_j membrane potential for computing $\sum_{i \in \Gamma_j} \alpha_{ij} x_i$ and resulting firing time t_j. All the delays Δ_{ij} have been set equal to Δ. Neuron N_4 (third firing) is inhibitory whereas the other three are excitatory. The slopes of the PSPs are modulated by the synaptic efficacies w_{ij}.

Network Topology and Dynamics

- Reviewed models describe single neurons, still need to create networks
- Traditional Architectures
  - Use temporal coding to reduce SNN to NN
  - Refer to previous slide
- Echo State Networks & Liquid State Machines

Echo State Networks

- Produce an echo state network
- Sample network training dynamics
- Compute output weights, use any linear regression algorithm
- SNs implemented in ESN outperform traditional ESNs



Liquid State Machines

- Turns time varying input into a spatiotemporal pattern of activation
- Large number of non-linear activation states
- Activations go into readout neuron(s) (linear discriminate units)

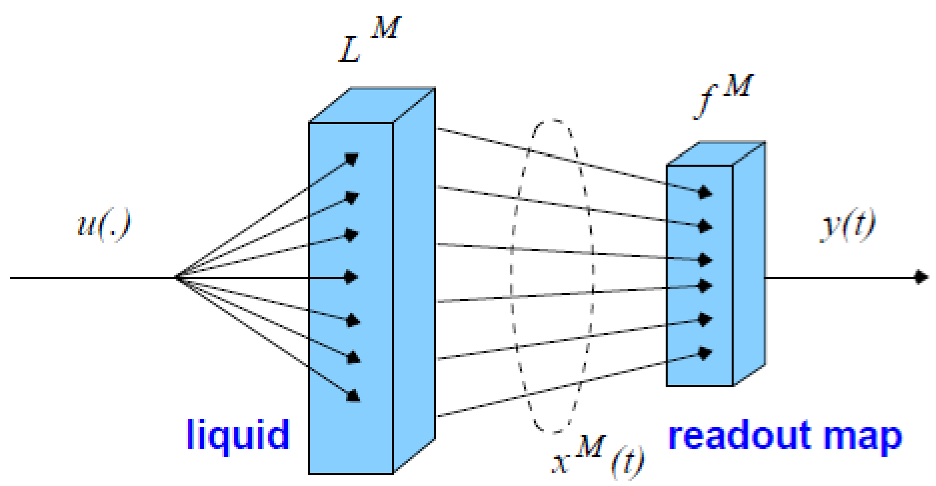


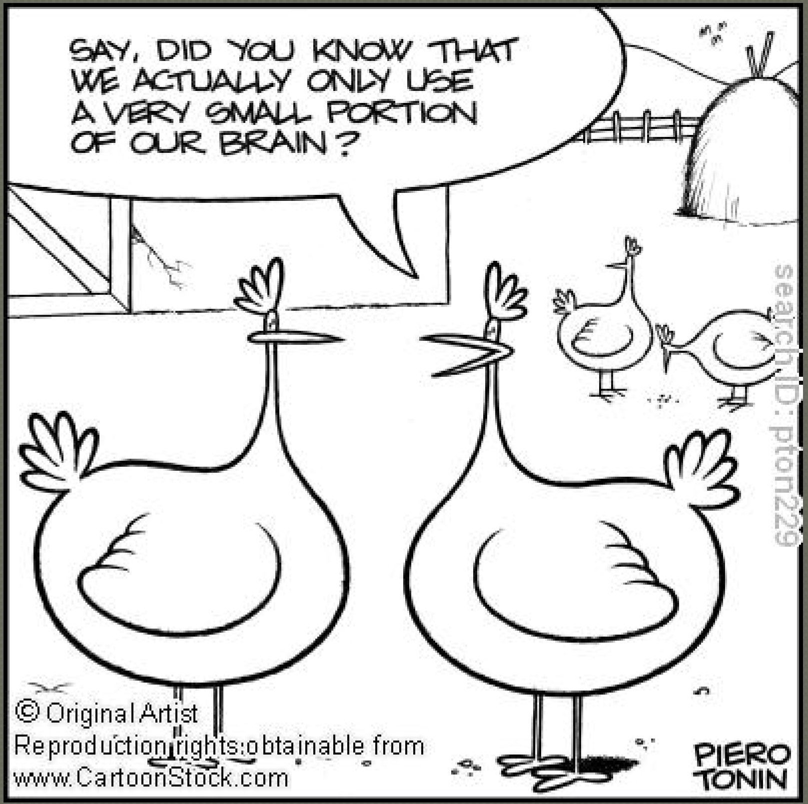
Figure 11: Architecture of a "Liquid State Machine". A continuous stream of values $u(.)$ is injected as input into the liquid filter L^M. A sufficiently complex excitable "liquid medium" creates, at time t, the *liquid state* $x^M(t)$, which is transformed by a memoryless *readout map* f^M to generate output $y(t)$.

Cell Assemblies

- “A group of neurons with strong mutual excitatory connections.”
- Excite one, excite all (many)
- “Grandmother Neural Groups”
- Synfire chain: pool of in-sync neurons
- Transient synchrony
  - Leads to collective sync. event; computational building block, “many variables are cur. ~equal”
- Polychronization
  - "reproducible time-locked but not synchronous firing patterns"

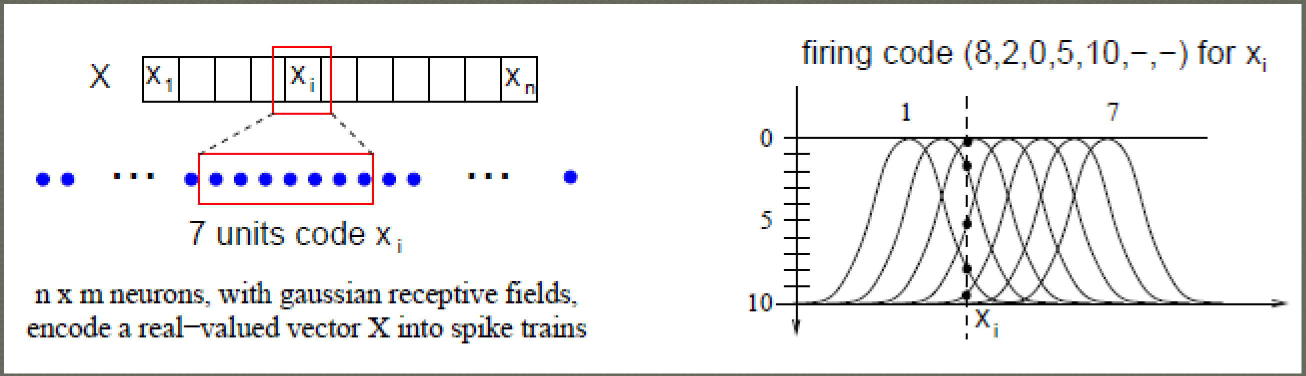
Learning Rules

- Traditional Methods
- New SNN Methods



Applying Traditional Learning Rules to SNN

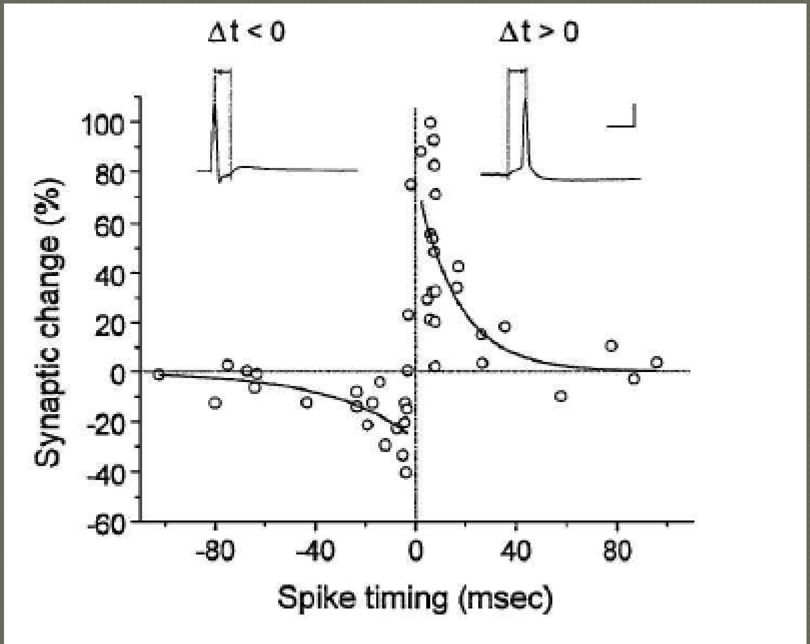
- Hopfield Networks (Maass & Natschlager)
- Kohonen SOMs (Ruf & Schmitt)
- RBF Networks (Natschlager & Rug)
- ML RBF Networks (Bohte, La Poutre & Kok)
- SNN shown to be universal function approximaters

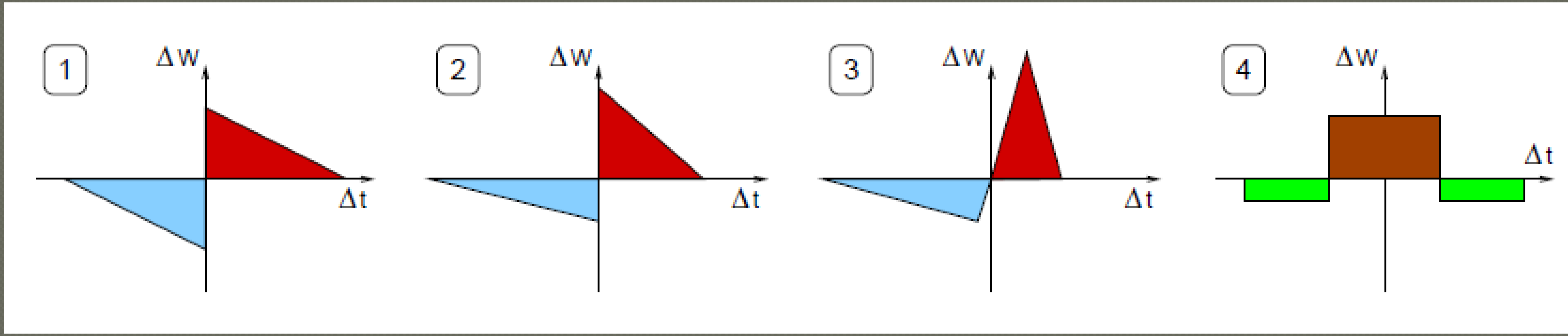


Hebbian-based Learning

- "When a pre-synaptic neuron repeatedly fires right before a post-synaptic neuron fires, the weight between the two neurons increases."
- Hebbian Properties
  - Synaptic Scaling
  - Synaptic Redistribution
  - Spike-timing dependent synaptic plasticity

Spike-timing Dependent Synaptic Plasticity (STDP)





SNN Learning Theory Models

- Maximization of mutual information
- BCM model
- Minimization of entropy
  - Minimize the response variability in the post-synaptic neuron given a particular input pattern

Software & Hardware Implementation

- Event-driven Simulation
  - Vs. time-driven simulation
  - Most of the time neurons aren't firing, so
  - Calculate when firing events occur, not what every neuron is doing at every time step
  - Delayed firing problem
- Parallel
  - SpikeNET
  - DAMNED simulator

Applications

- Hopfield and Brody, Digit Recognition
  - Generalize from small number of examples
  - Robust to noise
  - Uses temporal integration of transient synchrony
  - Time warp invariant
  - A set of neurons fire synchronously to a particular input (transient synchrony)
- Many examples in
  - Speech processing
  - Computer Vision

Discussion

- Spiking Neuron Networks
  - Biologically motivated
  - Computationally difficult without simplification
  - Traditional learning rules don’t take advantage of timing sequencing
  - New learning rules will have to be forthcoming before SNN show their potential