Modeling Networks of Neurons

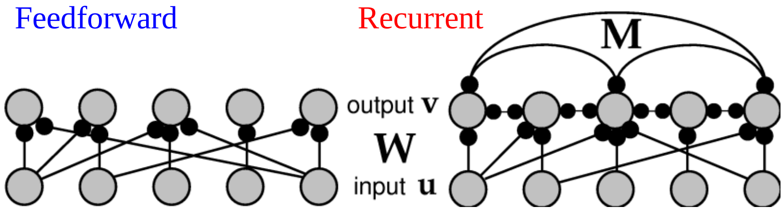
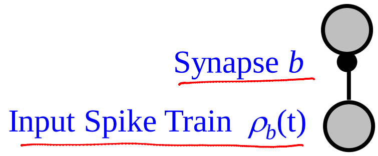


Image Source: Dayan & Abbott textbook

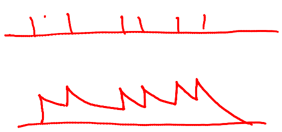
Modeling Networks: Spiking versus Firing Rate

- Option 1: Model networks using *Spiking* neurons
  - *Advantages*: Model computation and learning based on:
    - Spike Timing
    - Spike Correlations/Synchrony between neurons
  - *Disadvantages*: Computationally expensive
- Option 2: Use neurons with *firing-rate outputs (real valued outputs)*
  - *Advantages*: Greater efficiency, scales well to large networks
  - *Disadvantages*: Ignores spike timing issues
- Question: How are these two approaches related?

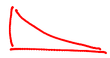
Recall: Linear Filter Model of a Synapse



$\rho_b(t) = \Sigma_i \, \delta(t-t_i)$ (t_i are the input spike times, δ = delta function)



Filter for synapse b = $K(t)$

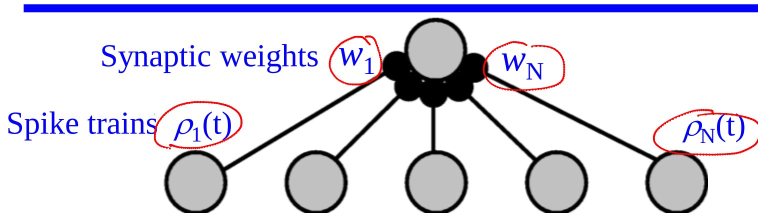


Synaptic conductance at b:

$$g_b(t) = g_{b,\max} \sum_{t_i < t} K(t - t_i)$$

$$= g_{b,\max} \int_{-\infty}^{t} K(t-\tau)\rho_b(\tau)\,d\tau$$

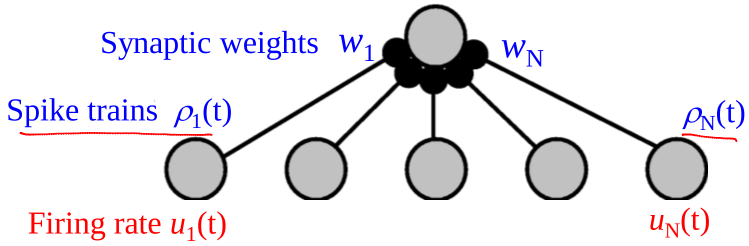
From a Single Synapse to Multiple Synapses



Total synaptic current $I_s(t) = \sum_{b=1}^{N} I_b(t)$

$$I_s(t) = \sum_{b=1}^{N} w_b \int_{-\infty}^{t} K(t-\tau)\rho_b(\tau)\,d\tau$$

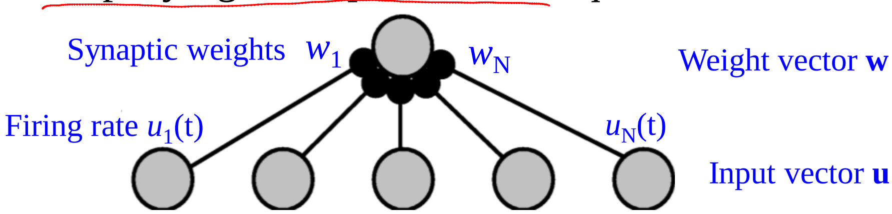
From Spiking to Firing Rate Model



Total synaptic current $I_s(t) = \sum_{b=1}^{N} w_b \int_{-\infty}^{t} K(t-\tau)\rho_b(\tau)d\tau$ **Spike train $\rho_b(t)$**

$$\approx \sum_{b=1}^{N} w_b \int_{-\infty}^{t} K(t-\tau)u_b(\tau)d\tau$$ **Firing rate $u_b(t)$**

Simplifying the Input Current Equation

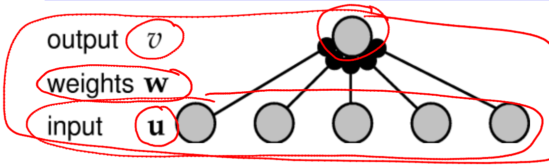


Suppose synaptic filter K is exponential: $K(t) = \frac{1}{\tau_s} e^{-\frac{t}{\tau_s}}$

Differentiating $\frac{d}{dt}$ $I_s(t) = \sum_b w_b \int_{-\infty}^{t} K(t-\tau)u_b(\tau)d\tau$ w.r.t. time t,

we get $$\tau_s \frac{dI_s}{dt} = -I_s + \sum_b w_b u_b$$
$$= -I_s + \mathbf{w} \cdot \mathbf{u}$$

Firing-Rate-Based Network Model



$\tau_s << \tau_r$

$I_s = w.u$

Output firing rate changes like this: $$\tau_r \frac{dv}{dt} = -v + F(I_s(t))$$

$\tau_r \frac{dv}{dt} = -v + F(w.u)$

Input current changes like this: $$\tau_s \frac{dI_s}{dt} = -I_s + \mathbf{w}\cdot\mathbf{u}$$

$\tau_r << \tau_s$

$v = F(I_s(t))$

STATIC INPUT

$\boxed{v_{ss} = F(w.u)}$

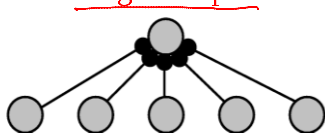
What if there are multiple output neurons?

Single Output

Scalar v

Vector $\mathbf{w}$

Vector $\mathbf{u}$



Input Vector

$$\boxed{\tau \frac{dv}{dt} = -v + F(\mathbf{w}\cdot\mathbf{u})}$$

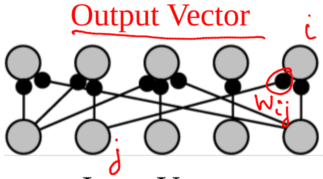
(Assuming relatively fast synapses, $I_s = \mathbf{w}\cdot\mathbf{u}$ at each t)

Output Vector

Vector $\mathbf{v}$

Matrix W

Vector $\mathbf{u}$



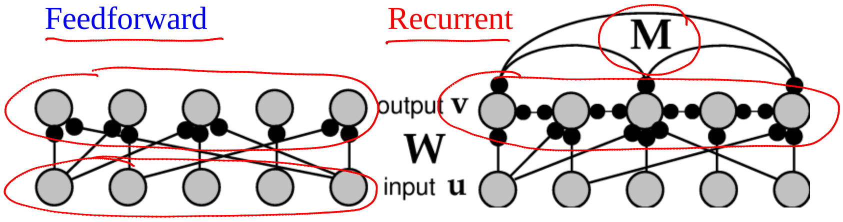
Input Vector

$$\boxed{\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + F(\mathrm{W}\mathbf{u})}$$

W_{ij}

Feedforward versus Recurrent Networks

Feedforward Recurrent



$$\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + F(\mathrm{W}\mathbf{u} + \mathrm{M}\mathbf{v})$$

Output Decay Input Feedback

For feedforward networks, M = matrix of zeros

Image Source: Dayan & Abbott textbook

Example: Linear Feedforward Network

Dynamics: $\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathrm{W}\mathbf{u}$

Steady State (set $d\mathbf{v}/dt$ to 0): $\mathbf{v}_{ss} = \mathrm{W}\mathbf{u}$

$$\mathrm{W} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

What is $\mathbf{v}_{ss}$?

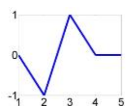
Linear Feedforward Network

$$\mathbf{v}_{ss} = \mathrm{W}\mathbf{u} = \overset{6\times5}{\begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix}} \underset{u}{\overset{5\times1}{\begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{bmatrix}}} = \underset{v_{ss}}{\overset{6\times1}{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}}}$$

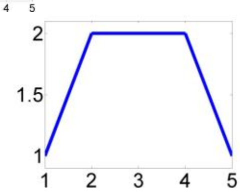
What is the network doing?

Network is performing Linear Filtering for Edge Detection

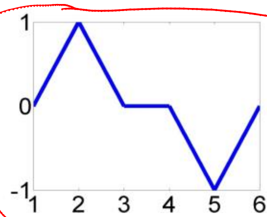
$\text{Filter} = [0 \quad -1 \quad 1 \quad 0 \quad 0]$
(and shifted versions in W)



$$\text{Input} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{bmatrix} \qquad \text{Output} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$



Input



Output

Example of Edge Detection in a 2D Image

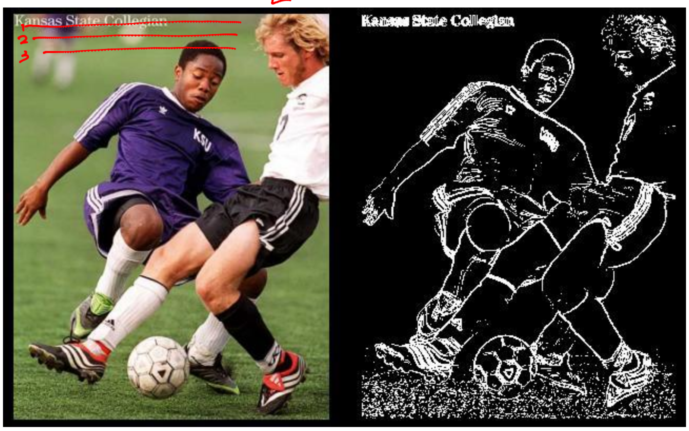
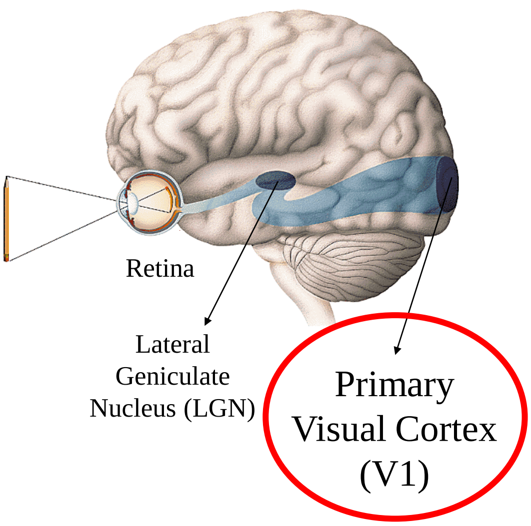
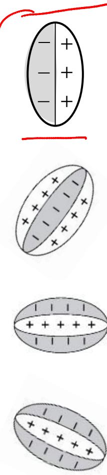


Image from http://www.alexandria.nu/ai/blog/entry.asp?E=51

Edge detectors in the brain

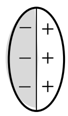




Examples of receptive fields in primary visual cortex (V1)

The Brain can do Calculus!

V1 neurons are basically computing derivatives!



W

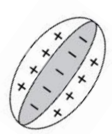
$$[0\quad -1\quad 1\quad 0\quad 0]$$

x x+1

$$\frac{df}{dx} = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$$

Wu

Discrete approximation $\approx f(x+1) - f(x)$



W

$$[0\quad 1\quad -2\quad 1\quad 0]$$

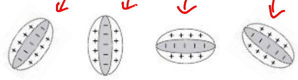
x-1 x x+1

$$\frac{d^2 f}{dx^2} = \lim_{h\to 0}\frac{f'(x+h)-f'(x)}{h}$$

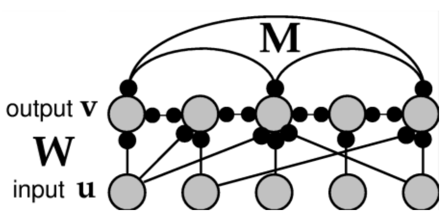
Wu

Disc. approx. $\approx \big(f(x+1) - f(x)\big) - \big(f(x) - f(x-1)\big)$

$$= f(x+1) - 2f(x) + f(x-1)$$



Next Lecture: Recurrent Networks



$$\tau\frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{W}\mathbf{u} + \mathbf{M}\mathbf{v}$$

Output Decay Input Feedback