recap

$n >> p$

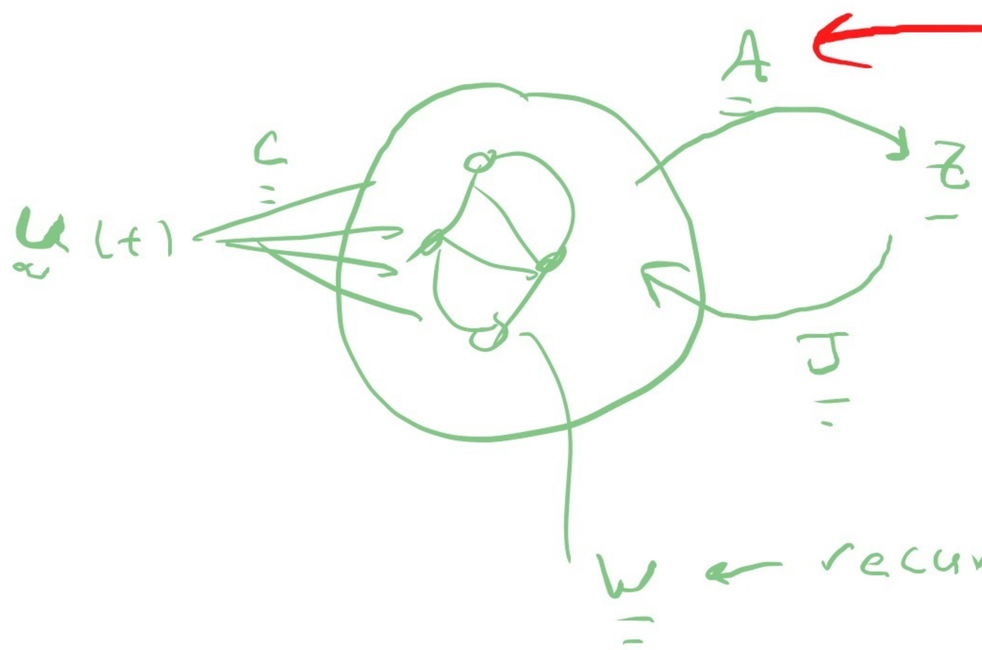
$$\tau \frac{dx_i}{dt} = \sum_{j=1}^{n} W_{ij} \phi(x_j) + \sum_{\mu} J_{i\mu} z_\mu + \sum_{\mu} C_{i\mu} u_\mu$$

feedback

control

$$z_\mu = \sum_{i=1}^{p} A_{\mu i} \phi(x_i)$$

↕ want to achieve a target $z_\mu^*(t)$



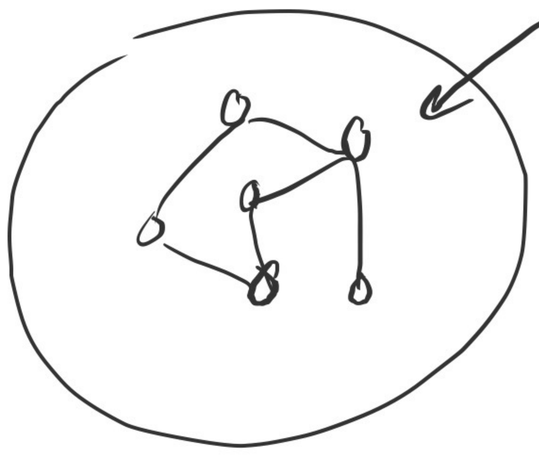
wide nonlinear network

Train only A

$W = 0$ (no recurrent connectivity)

$$\tau \frac{d\tilde{z}_\mu}{dt} = \sum_{i} A_{\mu i} \phi\left(\sum_{\nu} J_{i\nu} \tilde{z}_\nu + \sum_{\nu} C_{i\nu} u_\nu\right) - \tilde{z}_\mu$$

$$\tau \frac{d\tilde{z}_\mu}{dt} + \tilde{z}_\mu = z_\mu$$



recurrent network
- recurrent weights are plastic
- how do we train recurrent weights?

$$\tau \dot{x}_i = \sum_j W_{ij} \phi(x_j) + h_i(t) - x_i$$

control signal + feedback from z

$$z = \sum_i A_i \phi(x_i)$$

$$\mathcal{L} = \frac{1}{2} \int dt \, (z^* - z)^2 = \frac{1}{2} \int dt \, (z^* - \sum_i A_i x_i)^2$$

Minimize loss with respect to W_{ij} and A_i

W_{ij}: hard

A_i: easy

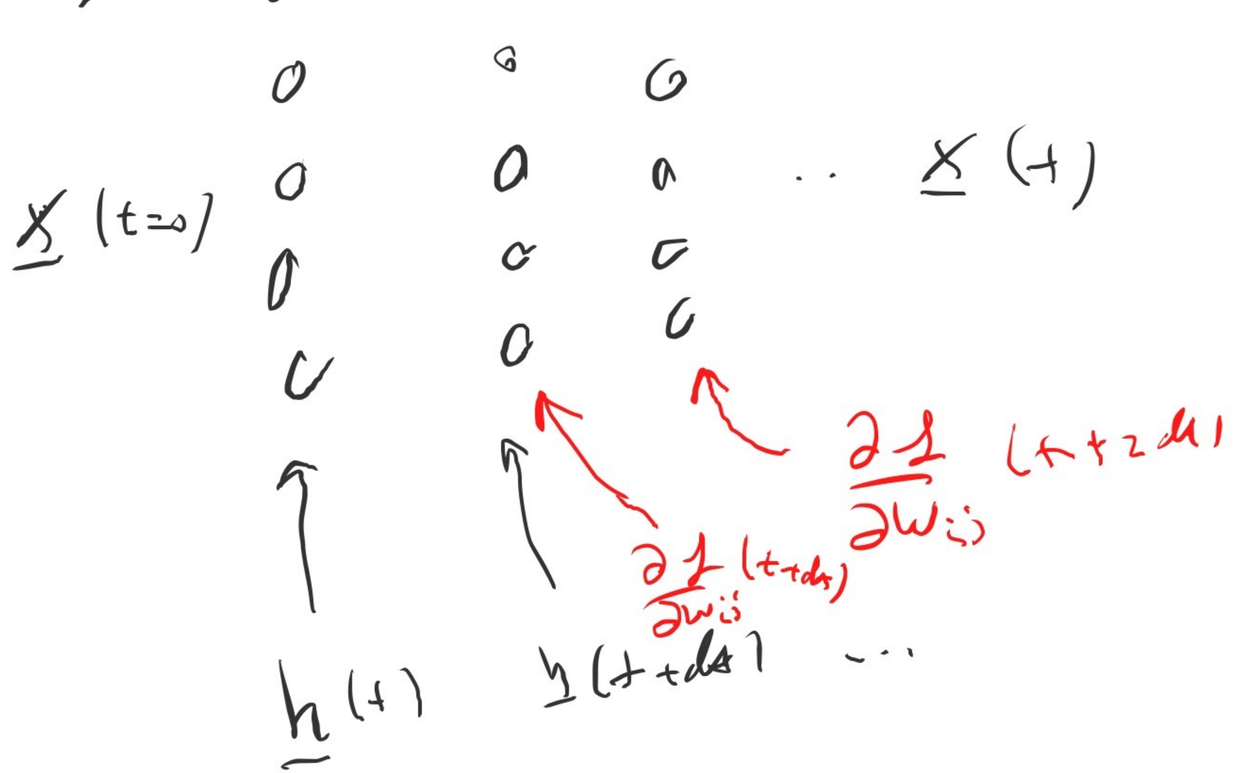
- discretize time

$$X_i(t+dt) = X_i(t)\left(1 - \frac{dt}{\tau}\right) + \frac{dt}{\tau}\left[\sum_j W_{ij}\,\phi(x_j) + h_i\right]$$

feedforward network with tied weights

Compute

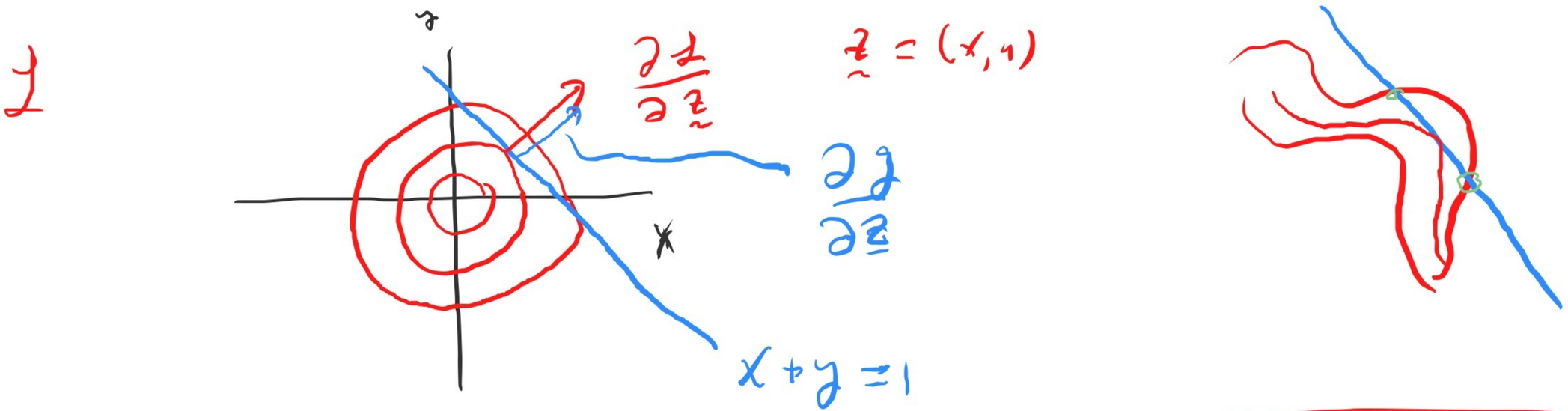
$\frac{\partial L}{\partial W_{ij}}$: compute via backprop



Lagrange multipliers

minimize $\frac{x^2+y^2}{2} = f(x,y)$

subject to $x+y=1$ $\quad g(x,y) = x+y-1$



at the minimum, gradient of loss and gradient of constraint are parallel

↑ extremum

$$\frac{\partial f}{\partial \underline{z}} + \lambda \frac{\partial g}{\partial \underline{z}} = 0$$

λ ← scalar variable to be determine

$$\frac{\partial}{\partial x} \quad x + \lambda = 0$$

$$\frac{\partial}{\partial y} \quad y + \lambda = 0$$

$$g(x,y) = ax + by - c$$

$$x + \lambda a = 0$$

$$y + \lambda b = 0$$

$$ax + by = c$$

$$x = y = -\lambda$$

$$g(x,y) = x + y - 1 = 0 \quad = -2\lambda - 1 = 0 \quad \lambda = -\frac{1}{2}$$

$$\Rightarrow \quad x = y = \frac{1}{2}$$

$$f(\underset{\sim}{x}) \quad \text{s.t.} \quad g_1(\underset{\sim}{x}) = 0 \,, \quad g_2(\underset{\sim}{x}) = 0$$

$$\frac{\partial f}{\partial \underset{\sim}{x}} + \lambda_1 \frac{\partial g_1}{\partial \underset{\sim}{x}} + \lambda_2 \frac{\partial g_2}{\partial \underset{\sim}{x}} = 0$$

$\underset{\sim}{x} \sim \mathbb{R}^n$

$$\hookrightarrow \begin{cases} \frac{\partial}{\partial \lambda_1} \left[f + \lambda_1 g_1 + \lambda_2 g_2 \right] = 0 \\ \frac{\partial}{\partial \lambda_2} \end{cases} \Big\} = 0$$

n eqns. , $n+2$ unknowns

$g_1 = 0$
$g_2 = 0$
} 2 more equations

$$\tau \frac{d\underline{x}}{dt} = \underline{\underline{w}} \cdot \phi(\underline{x}) + \underline{u} - \underline{x}$$

$$z = \underline{\underline{A}} \cdot \underline{x}$$

$$\mathcal{L} = \frac{1}{2} \int dt \left(z^{*}_{(t)} - \underline{\underline{A}} \cdot \underline{x}(t) \right)^2$$

$$+ \int dt \; \underset{\sim}{\lambda}(t) \cdot \left[\tau \frac{d\underline{x}}{dt} + \underline{x} - \underline{\underline{w}}\, \phi(\underline{x}) - \underline{u} \right]$$

↖ Lagrange multipliers

Minimize $\mathcal{L}$ w.r.t. $\underline{\underline{A}}$, $\underline{\underline{w}}$

Set $\dfrac{\partial \mathcal{L}}{\partial \underline{x}} = 0$ S.T. $[\;\;] = 0$

$$\frac{\partial \mathcal{L}}{\partial \underline{\lambda}} = 0$$

$$\dot{x} = -\omega x \qquad x(t=0) = 1$$

minimize $\frac{1}{2}\int_0^T dt\ (f(t) - A\,x(t))^2$

Lagrange multipliers

$$\mathcal{L} = \frac{1}{2}\int_0^T dt\ (f(t) - A\,x(t))^2 + \int_0^T dt\ \lambda(t)(\dot{x} + \omega x)$$

$$\dot{w} = -\gamma \frac{\partial \mathcal{L}}{\partial w}$$

$$\dot{A} = -\gamma \frac{\partial \mathcal{L}}{\partial A}$$

slow update rules (easy)

enforce:

$$\frac{\partial \mathcal{L}}{\partial \lambda(t)} = 0$$

$$\frac{\partial \mathcal{L}}{\partial x(t)} = 0$$

enforces constraints

gradients of loss parallel to gradients of constraints

$\mathcal{L} = \mathcal{L}(A, \omega, x(0), x(dt), x(2dt) \ldots \lambda(t), \lambda(2t) \ldots)$

$\frac{\partial \mathcal{L}}{\partial (\,)} = 0$ each param

$$\frac{\partial}{\partial \lambda(t)} \int_0^T dt' \; \lambda(t')\left(\dot{x}(t') + \omega x(t')\right)$$

functional derivative

$$\lambda \frac{d}{dt} x = \frac{d}{dt}(\lambda x) - \frac{d\lambda}{dt} x$$

$$\frac{\partial}{\partial \lambda(t)} \left[\int_0^T dt' \; -\dot{\lambda}(t') x(t') + \omega \lambda(t') x(t') \right] + \int_0^T dt' \; \lambda(t') \dot{x}(t')$$

$$+ \lambda(T) x(T) - \lambda(0) x(0) = \int_0^T dt' \; \frac{d}{dt'}\left(\lambda(t') x(t')\right) = \lambda(T) x(T) - \lambda(0) x(0)$$

$$= \dot{x}(t) + \omega x(t) = 0$$

$$t \neq T \qquad -\dot{\lambda} + \omega \lambda = 0$$

$$t = T \qquad \lambda(T) = 0$$

integrate backwards in time

$$\frac{\partial}{\partial x(t)} \sum_{t'=1, dt, 2dt, \ldots} \left[-\dot{\lambda}(t')\, x(t') + \omega\, \lambda(t')\, x(t') + \frac{1}{\tau} \left(f(t') - A x(t') \right)^2 \right] dt$$

$$+ \; x(T)\,\lambda(T) - x(0)\,\lambda(0)$$

$$\frac{\partial}{\partial x(T)} = \frac{\lambda(T)}{dt} + \left[-\dot{\lambda} + \omega\,\lambda - \left(f - A x \right) A \right] \frac{dt}{dt}$$

$$\simeq \lambda(T)$$

$$\frac{d\lambda}{dt} = \omega\lambda - A\underbrace{\left(f(t) - A x(t) \right)}_{\text{linearized dynamics}}$$

$$\lambda(T = \infty) = 0$$

$$\tau \dot{\underset{\sim}{x}} = \underline{\underline{w}} \cdot \phi(\underline{x}) + \underline{h}(t) - \underline{x} \qquad \underline{x}(t=0) = \underline{x}_0$$

$$\mathcal{L} = \frac{1}{2}\int_0^T dt' \left(z^*(t') - \underline{A} \cdot \underline{x}(t') \right)^2 + \int_0^T dt' \; \underline{\lambda}(t') \cdot \left[\tau \dot{\underline{x}} + \underline{x} - \underline{\underline{w}} \cdot \phi - \underline{h} \right]$$

$$0 = \frac{\partial \mathcal{L}}{\partial \underline{\lambda}(t)} = \tau \dot{\underline{x}} + \underline{x} - \underline{\underline{w}} \cdot \phi(\underline{x}) - \underline{h}(t) = 0$$

$$\int dt' \; \underset{\sim}{\lambda} \cdot \frac{d\underline{x}}{dt} = \int dt' \left[\frac{d}{dt'}(\underline{\lambda} \cdot \underline{x}) - \frac{d\underline{\lambda}}{dt'} \cdot \underline{x} \right]$$

$$= \underline{\lambda}(T) \cdot \underline{x}(T) - \underline{\lambda}(0) \cdot \underline{x}(0)$$

$$\mathcal{L} = \frac{1}{2}\int dt' \left(z^*(t') - \underline{\underline{A}}\cdot\underline{x}(t')\right)^2 + \left[\underline{\lambda}(T)\cdot\underline{\dot{x}}(T) - \underline{\lambda}(0)\cdot\underline{\dot{x}}(0)\right]T$$

$$+ \int_0^T dt' \left[\underline{\dot{\lambda}}\cdot\underline{x}(t') + \underline{\lambda}(t')\underline{\dot{x}} - \underline{\underline{w}}\cdot\underline{\phi}(\underline{x}) - \underline{h}(t')\right]$$

$$0 = \frac{\partial \mathcal{L}}{\partial \underline{\dot{x}}} = -\underline{\dot{\lambda}} - \underline{\underline{A}}\left(z^*(t') - \underline{\underline{A}}\cdot\underline{x}(t')\right)$$

$$+ \left[-\underline{\dot{\lambda}}T + \underline{\lambda}(t') - \underline{\underline{w}}\cdot\underline{\phi}'(\underline{x})\right] = 0$$

$$T\dot{\underline{\lambda}} = \underline{\lambda}\cdot\left[\underline{\underline{I}} - \underline{\underline{w}}\cdot\underline{\phi}'(\underline{x})\right] - \underline{\underline{A}}\left(\underline{z}^*(t') - \underline{\underline{A}}\cdot\underline{x}(t')\right)$$

linearized dynamics (integrating backwards in time)

$$\underline{\lambda}(t=T) = 0$$

$$\dot{\underset{\sim}{A}} = -\eta \frac{\partial \bar{E}}{\partial \underset{\sim}{A}} = \eta \int dt' \left(\bar{z}^*(t') - \underset{\sim}{A} \bar{x}(t') \right) \bar{x}(t')$$

$$\dot{\underset{\sim}{m}} = -\eta \frac{\partial \bar{E}}{\partial \underset{\sim}{m}} = \eta \int dt' \; \bar{\lambda}(t') \; \phi'(\bar{x}(t'))$$

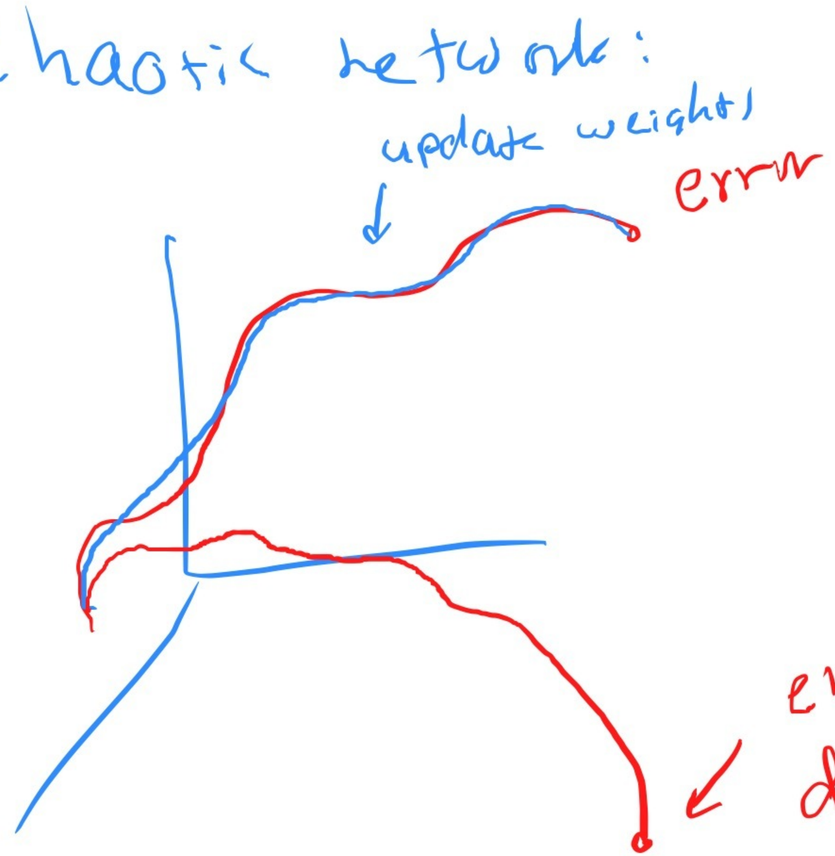
$$\frac{d\bar{\lambda}}{dt} = \left(\quad \right)$$

entirely non-trivial

Summarize

- problem faced by the brain (we believe) is to update recurrent weights.
- that's hard: change weights, and that affects the future.
- BPTT: run into the future, backprop the error

Chaotic network:



- BPTT can't deal with long time
  "long" depends on how chaotic
- clearly not biologically plausible

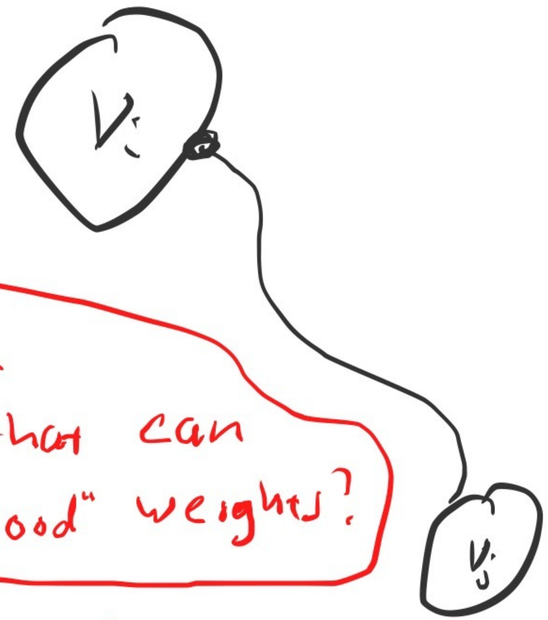
- Slight change of gears

- local learning rules

$$\tau \dot{V}_i = \phi\left(\sum_j W_{ij} V_j\right) - V_i$$

local learning rule:

$$\Delta W_{ij} = f(V_i, V_j)$$



is there a function that can produce "good" weights?

$$\Delta W_{ij} = f\left(f^{-1}\phi\left(\sum_k W_{ik} V_k(t)\right), V_j(t)\right)$$

$$\Delta W_{ij} = \tilde{f}(\text{all the weights})$$

↳ function of weights

$$\tau \dot{V}_i = \phi\left(\sum_j W_{ij} V_j\right) \Rightarrow V_i = f^{-1}\phi\left(\sum_k W_{ik} V_k(t)\right)$$

$\tau \to 0 \quad f^{-1}(\phi) = \phi$ ← Several time constants into the pot

$$V_i^{l+1} = \sum_j W_{ij}^{l} V_j^{l}$$

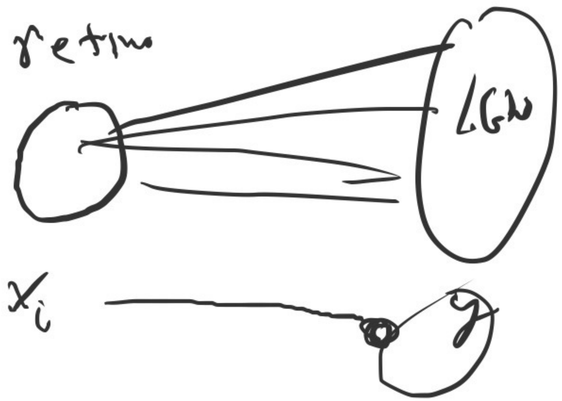
discrete sampled: V_j in

$$y = \sum_j W_j x_j - V_j^{l}$$

(V_0^{l+1}, W_{0j}^{l})

learning rule: $\dot{w}_i = y\, x_i$ (Hebb rule)

↖ averaged over lots of x_i's



$$\dot{w}_i = \sum_j w_j x_j x_i$$

average $\quad \rightarrow = C_{ji}$

$$\dot{w}_i = \sum_j w_j \langle x_j x_i \rangle$$

$$\dot{\underline{w}} = \underline{\underline{C}} \cdot \underline{w}$$

$$\underline{w} = \sum_k a_k \underline{v}_k \qquad \underline{\underline{C}} \cdot \underline{v}_k = \lambda_k \underline{v}_k$$

$$\underline{v}_k \cdot \underline{v}_\ell = \underline{\delta_{k\ell}}$$

$$\dot{\underline{w}} = \dot{a}_k \underline{v}_k = \underline{\underline{C}} \cdot \sum_k a_k \underline{v}_k = \sum_k a_k \underline{\underline{C}} \cdot \underline{v}_k = \sum_k a_k \lambda_k \underline{v}_k$$

$\dot{\underline{v}}_\ell$ $\qquad \boxed{\dot{a}_\ell = \lambda_\ell a_\ell}$ $\qquad a_\ell(t) = a_\ell(0)\, e^{\lambda_\ell t}$

- at long times, dominated by largest eigenvalue of covariance matrix
- picks out structure in the input

$w \to \infty$

fixed by Oja's rule

---

next time:

- Oja's rule
- feed forward networks
- attempts to make them biologically plausible