Learning and Meta-learning

- **computation**
  - making predictions
  - choosing actions
  - acquiring episodes
  - *statistics*
- **algorithm**
  - gradient ascent (*eg* of the likelihood)
  - correlation
  - Kalman filtering
- **implementation**
  - Hebbian synpatic plasticity
  - neuromodulation

Adaptation and Learning

Two strategies:

bottom-up

- rules of neural plasticity (Hebbian; sliding threshold)
- mathematical application (development)
- computational significance (PCA; projection pursuit; probabilistic model fitting)

top-down

- function(s) $E(\mathbf{w})$
- $\Delta\mathbf{w} \propto -\nabla_{\mathbf{w}} E(\mathbf{w})$
- delta rule; basis functions

Types of Learning

supervised	$\mathbf{v}\|\mathbf{u}$	inputs $\mathbf{u}$ and *desired* or *target* outputs $\mathbf{v}$ both provided
reinforce	$\max r\|\mathbf{u}$	input $\mathbf{u}$ and scalar *evaluation* r often with *temporal* credit assignment problem
unsupervised	$\mathbf{u}$	or *self-supervised* learn structure from statistics

These are closely related:

supervised learn $P[\mathbf{v}|\mathbf{u}]$

unsupervised learn $P[\mathbf{v}, \mathbf{u}]$

Hebb

Famously suggested:

> if cell A consistently contributes to the activity of cell B, then the synapse from A to B should be strengthened

- strong element of *causality*
- what about weakening (LTD)?
- multiple timescales – STP to protein synthesis
- multiple biochemical mechanisms
- systems:
  - hippocampus – multiple sub-areas
  - neocortex – layer and area differences
  - cerebellum – LTD is the norm

Neural Rules

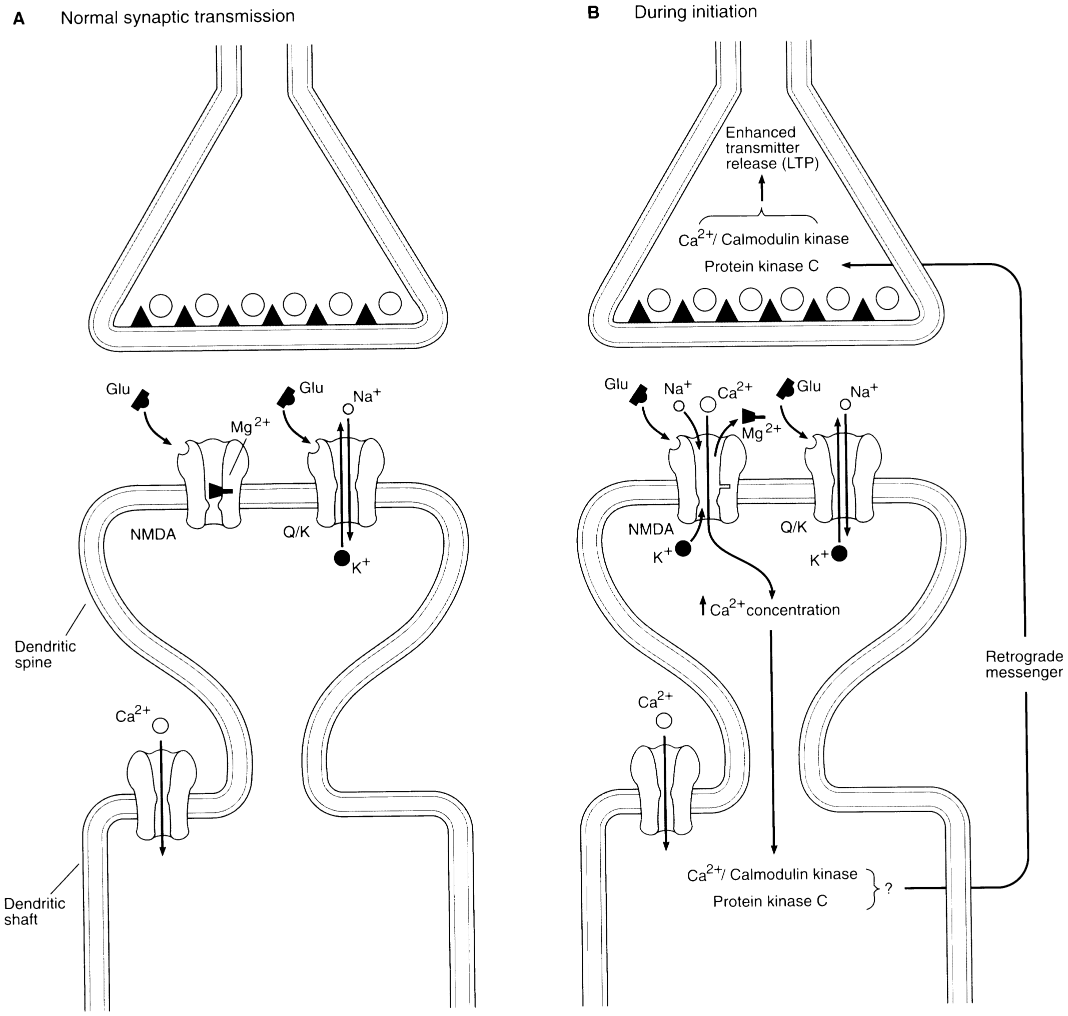
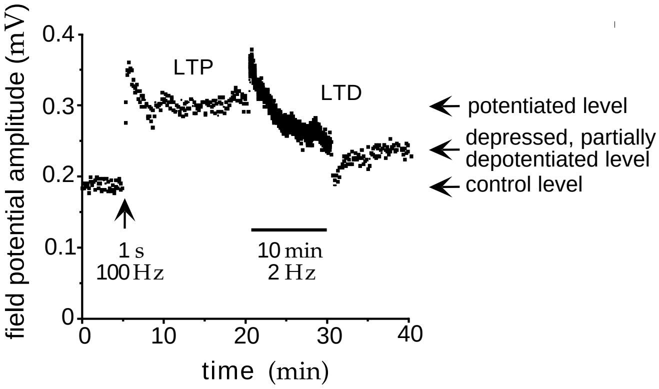


FIGURE 65–11



Stability and Competition

Hebbian learning involves *positive feedback.*

Control by:

LTD usually not enough – covariance *versus* correlation

saturation prevent synaptic weights from getting too big (or too small) – triviality beckons

competition spike-time dependent learning rules

normalization over pre-synaptic or post-synaptic arbors

- subtractive: decrease all synapses by the same amount whether large or small
- multiplicative: decrease large synapses by more than small synapses

Preamble

Linear firing rate model

$$\tau_r \frac{dv}{dt} = -v + \mathbf{w} \cdot \mathbf{u} = -v + \sum_{b=1}^{N_u} w_b u_b$$

assume that τ_r is small compared with the rate of change of the weights, then

$$v = \mathbf{w} \cdot \mathbf{u}$$

during plasticity

Then have

$$\tau_w \frac{d\mathbf{w}}{dt} = f(v, \mathbf{u}, \mathbf{w})$$

Supervised rules use targets to specify v – neural basis.

The Basic Hebb Rule

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{u}v$$

averaged $\langle\rangle$ over input statistics gives

$$\tau_w \frac{d\mathbf{w}}{dt} = \langle \mathbf{u}v \rangle = \langle \mathbf{u}\mathbf{u} \cdot \mathbf{w} \rangle = \mathbf{Q} \cdot \mathbf{w}$$

where $\mathbf{Q}$ is the input correlation matrix.

Positive feedback instability

$$\tau_w \frac{d}{dt}|\mathbf{w}|^2 = 2\tau_w \mathbf{w} \cdot \frac{d\mathbf{w}}{dt} = 2v^2$$

Also have discretised version

$$\mathbf{w} \rightarrow \mathbf{w} + \frac{T}{\tau_w}\mathbf{Q} \cdot \mathbf{w}\,.$$

integrating over time, presenting patterns for T seconds.

Covariance Rule

Since LTD really exists, contra Hebb:

$$\tau_w \frac{d\mathbf{w}}{dt} = \quad \mathbf{u} \quad (v - \theta_v)$$

or

$$\tau_w \frac{d\mathbf{w}}{dt} = (\mathbf{u} - \boldsymbol{\theta}_u)\ v$$

If $\theta_v = \langle v \rangle$ or $\boldsymbol{\theta}_u = \langle \mathbf{u} \rangle$ then

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{C} \cdot \mathbf{w}$$

where $\mathbf{C} = \langle (\mathbf{u} - \langle \mathbf{u} \rangle)(\mathbf{u} - \langle \mathbf{u} \rangle) \rangle$ is the input covariance matrix.

Still unstable

$$\tau_w \frac{d}{dt} |\mathbf{w}|^2 = 2v(v - \langle v \rangle)$$

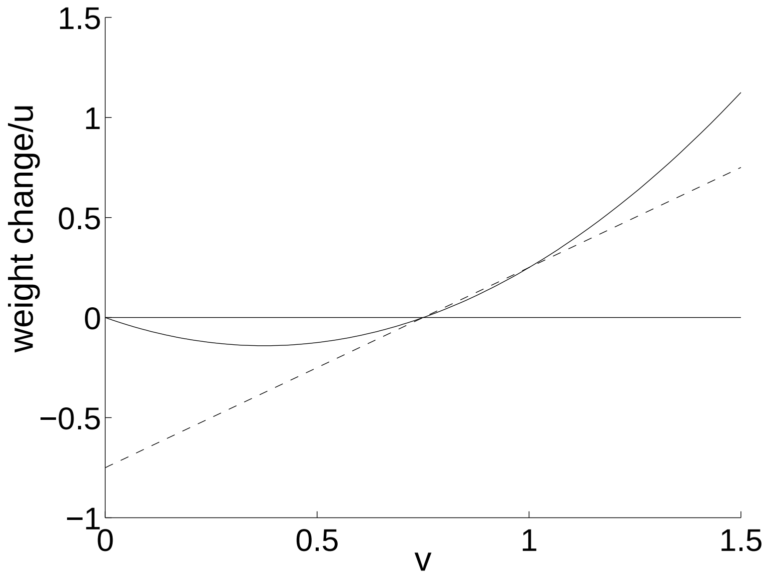
which averages to the (positive) covariance of v.

BCM Rule

Odd to have LTD with $v = 0$ or $\mathbf{u} = \mathbf{0}$.

Evidence for

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u}\left(v - \theta_v\right) .$$



If θ_v *slides* to match a high power of v

$$\tau_\theta \frac{d\theta_v}{dt} = v^2 - \theta_v$$

with a fast τ_θ, then get *competition* between synapses – intrinsic stabilization.

Subtractive Normalisation

Could normalise $|\mathbf{w}|^2$ or

$$\sum w_b = \mathbf{n} \cdot \mathbf{w} \quad \mathbf{n} = (1, 1 \ldots, 1)$$

For subtractive normalisation of $\mathbf{n} \cdot \mathbf{w}$:

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u} - \frac{v(\mathbf{n} \cdot \mathbf{u})}{N_u}\mathbf{n}$$

with dynamic subtraction, since

$$\tau_w \frac{d\mathbf{n} \cdot \mathbf{w}}{dt} = v\mathbf{n} \cdot \mathbf{u} \left(1 - \frac{\mathbf{n} \cdot \mathbf{n}}{N_u}\right) = 0\,.$$

as $\mathbf{n} \cdot \mathbf{n} = N_u$.

Strongly competitive – typically all the weights bar one go to 0. Therefore use upper saturating limit.

The Oja Rule

A multiplicative way to ensure $|\mathbf{w}|^2$ is constant

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u} - \alpha v^2 \mathbf{w}$$

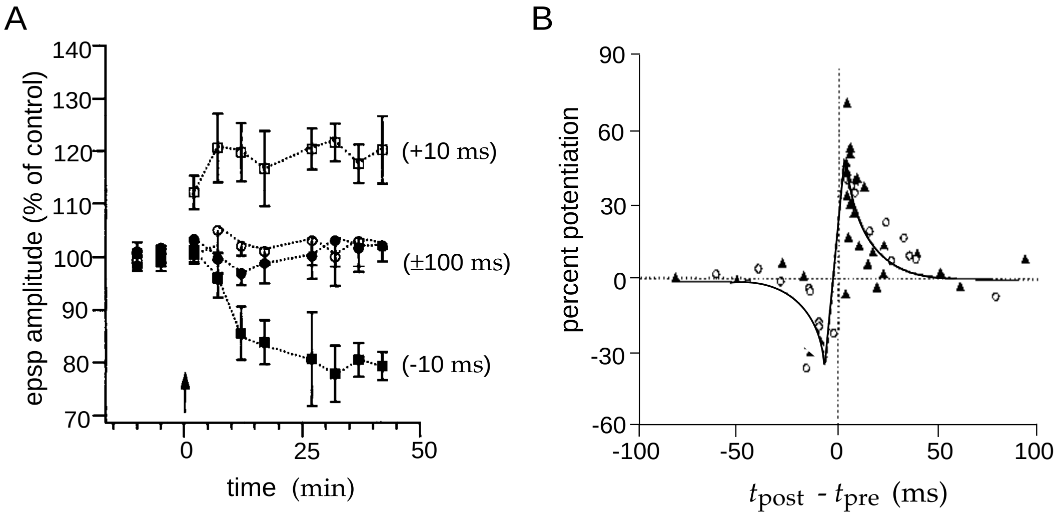
gives

$$\tau_w \frac{d|\mathbf{w}|^2}{dt} = 2v^2(1 - \alpha|\mathbf{w}|^2) \, .$$

so $|\mathbf{w}|^2 \to 1/\alpha$.

Dynamic normalisation – could also enforce normalisation all the time.

Timing-Based Rules



slice cortical pyramidal cells; Xenopus retinotectal system

- window of 50ms
- gets Hebbian causality right
- rate-description

$$\tau_w \frac{d\mathbf{w}}{dt} = \int_0^\infty d\tau \; (H(\tau)v(t)\mathbf{u}(t-\tau) + H(-\tau)v(t-\tau)\mathbf{u}(t)) \; .$$

- spike-based description necessary if an input spike can have a measurable impact on an output spike.
- critical factor is the overall integral – net LTD with 'local' LTP.
- partially self-stabilizing

Single Postsynaptic Neuron

Basic Hebb rule:

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{Q} \cdot \mathbf{w}$$

analyse using an eigendecomposition of $\mathbf{Q}$:

$$\mathbf{Q} \cdot \mathbf{e}_\mu = \lambda_\mu \mathbf{e}_\mu \qquad \lambda_1 \geq \lambda_2 \ldots$$

Since $\mathbf{Q}$ is symmetric and positive (semi-)definite

- complete set of real orthonormal evecs
- with non-negative eigenvalues
- whose growth is decoupled

Write

$$\mathbf{w}(t) = \sum_{\mu=1}^{N_u} c_\mu(t) \mathbf{e}_\mu$$

then

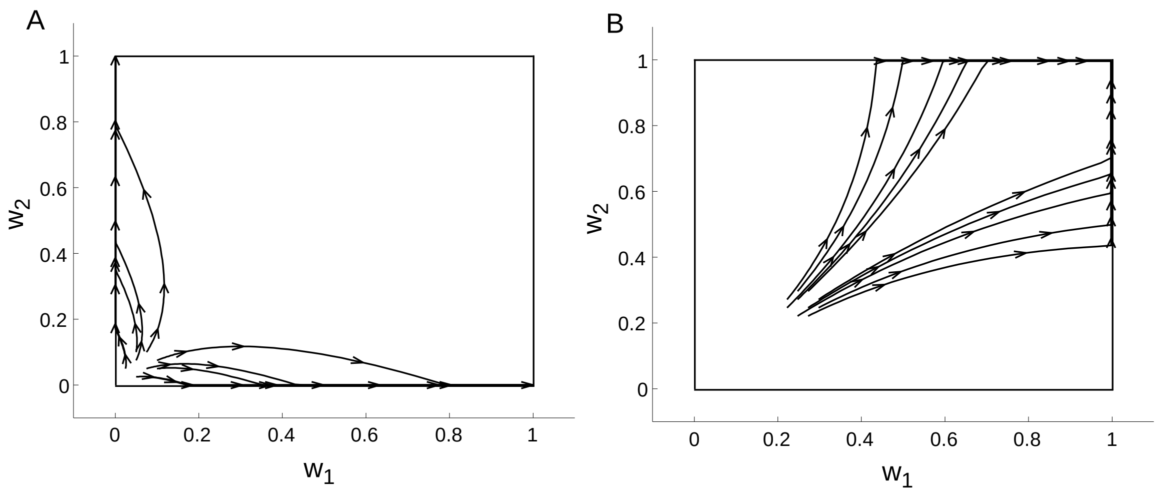
$$c_\mu(t) = c_\mu(0) \exp\left(\lambda_\mu \frac{t}{\tau_w}\right)$$

and $\mathbf{w}(t) \to \alpha(t)\mathbf{e}_1$ as $t \to \infty$

Constraints

$\alpha(t) = \exp(\lambda_\mu t/\tau_w) \to \infty$.

- Oja makes $\mathbf{w}(t) \to \mathbf{e}_1/\sqrt{\alpha}$
- saturation can disturb outcome



- subtractive constraint
  $\tau_w \dot{\mathbf{w}} = \mathbf{Q} \cdot \mathbf{w} - \frac{(\mathbf{w}\cdot\mathbf{Q}\cdot\mathbf{n})\mathbf{n}}{N_u}$.

  Sometimes $\mathbf{e}_1 \propto \mathbf{n}$ – so its growth is stunted; and $\mathbf{e}_\mu \cdot \mathbf{n} = 0$ for $\mu \neq 1$ so

$$\mathbf{w}(t) = (\mathbf{w}(0) \cdot \mathbf{e}_1)\,\mathbf{e}_1 + \sum_{\mu=2}^{N_u} \exp\left(\frac{\lambda_\mu t}{\tau_w}\right)(\mathbf{w}(0) \cdot \mathbf{e}_\mu)\,\mathbf{e}_\mu$$

Translation Invariance

Particularly important case for development has

$$\langle u_b \rangle = \langle u \rangle \qquad \mathbf{Q}_{bb'} = \mathcal{Q}(b - b')$$

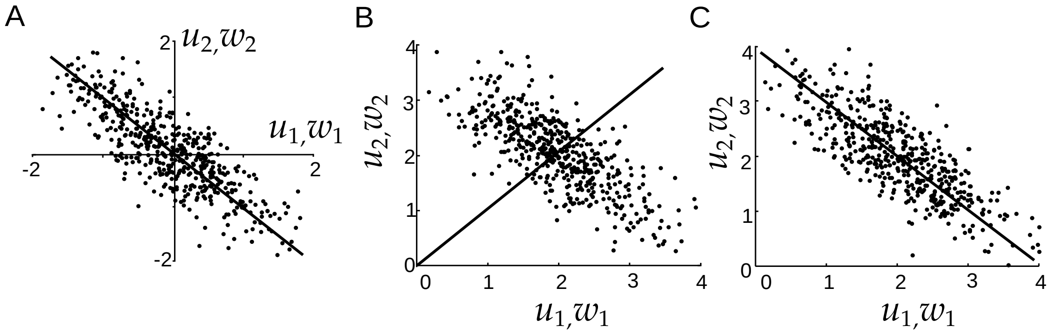
Write $\mathbf{n} = (1, \ldots, 1)$ and $\mathbf{J} = \mathbf{n}\mathbf{n}^T$, then

$$\mathbf{Q}' = \mathbf{Q} - N\langle u \rangle^2 \mathbf{J}$$

1. $\mathbf{e}_\mu \cdot \mathbf{n} = 0$, *AC modes* are unaffected
2. $\mathbf{e}_\mu \cdot \mathbf{n} \neq 0$, *DC modes* are affected
3. $\mathbf{Q}$ has discrete sines and cosines as eigenvectors
4. fourier spectrum of $\mathcal{Q}$ are the eigenvalues

PCA

What is the significance of $\mathbf{e}_1$?



- optimal linear reconstruction: minimise

$$E(\mathbf{w}, \mathbf{g}) = \left\langle |\mathbf{u} - \mathbf{g}v|^2 \right\rangle$$

- information maximisation:

$$\mathcal{I}[v, \mathbf{u}] = \mathcal{H}[v] - \mathcal{H}[v|\mathbf{x}]$$

  under a linear model

- assume $\langle \mathbf{u} \rangle = \mathbf{0}$ or use $\mathbf{C}$ instead of $\mathbf{Q}$.

Linear Reconstruction

$$\begin{aligned} E(\mathbf{w}, \mathbf{g}) &= \left\langle |\mathbf{u} - \mathbf{g}v|^2 \right\rangle \\ &= \mathcal{K} - 2\mathbf{w} \cdot \mathbf{Q} \cdot \mathbf{g} + \|\mathbf{g}\|^2 \mathbf{w} \cdot \mathbf{Q} \cdot \mathbf{w} \end{aligned}$$

quadratic in $\mathbf{w}$ with minimum at

$$\mathbf{w}^* = \frac{\mathbf{g}}{\|\mathbf{g}\|^2}$$

making

$$E(\mathbf{w}^*, \mathbf{g}) = \mathcal{K} - \frac{\mathbf{g} \cdot \mathbf{Q} \cdot \mathbf{g}}{\|\mathbf{g}\|^2}.$$

look for soln with $\mathbf{g} = \sum_k (\mathbf{e}_k \cdot \mathbf{g}) \mathbf{e}_k$ and $\|\mathbf{g}\|^2 = 1$:

$$E(\mathbf{w}^*, \mathbf{g}) = \mathcal{K} - \sum_{k=1}^{N} (\mathbf{e}_k \cdot \mathbf{g})^2 \lambda_k$$

clearly has $\mathbf{e}_1 \cdot \mathbf{g} = 1$ and $\mathbf{e}_2 \cdot \mathbf{g} = \mathbf{e}_3 \cdot \mathbf{g} = \ldots = \mathbf{0}$

Therefore $\mathbf{g}$ and $\mathbf{w}$ both point along principal component

Infomax (Linsker)

$$\text{argmax}_{\mathbf{w}} \mathcal{I}[v, \mathbf{u}] = \mathcal{H}[v] - \mathcal{H}[v|\mathbf{u}]$$

Very general unsupervised learning suggestion:

- $\mathcal{H}[v|\mathbf{u}]$ is not quite well defined unless $v = \mathbf{w} \cdot \mathbf{u} + \eta$ where η is arbitrarily deterministic
- $\mathcal{H}[v] = \frac{1}{2} \log 2\pi e \sigma^2$ for a Gaussian.

If $P[\mathbf{u}] \sim \mathcal{N}[\mathbf{0}, \mathbf{Q}]$ then

$$v \sim \mathcal{N}[0, \mathbf{w} \cdot \mathbf{Q} \cdot \mathbf{w} + v^2]$$

maximise $\mathbf{w}\mathbf{Q}\mathbf{w}^T$ subject to $\|\mathbf{w}\|^2 = 1$

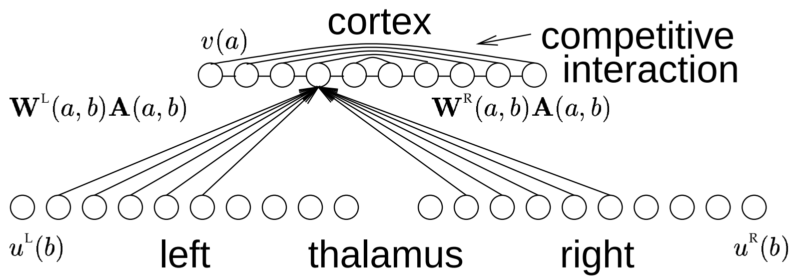
Same problem as above: implies that

$$\mathbf{w} \propto \mathbf{e}_1.$$

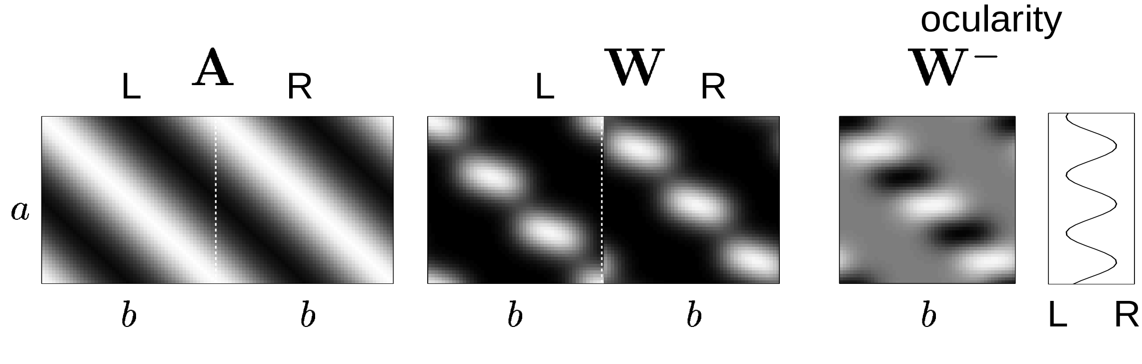
note the *normalisation*

If non-Gaussian, only maximising an *upper bound* on $\mathcal{I}[v, \mathbf{u}]$.

Ocular Dominance



- retina-thalamus-cortex
- OD develops around eye-opening
- interaction with refinement of topography
- interaction with orientation
- interaction with ipsi/contra-innervation
- effect of manipulations to input



Start Simple

Consider one input from each eye

$$v = w_{\mathsf{R}} u_{\mathsf{R}} + w_{\mathsf{L}} u_{\mathsf{L}} \,.$$

Then

$$\mathbf{Q} = \langle \mathbf{u}\mathbf{u} \rangle = \begin{pmatrix} q_{\mathsf{S}} & q_{\mathsf{D}} \\ q_{\mathsf{D}} & q_{\mathsf{S}} \end{pmatrix}$$

has

$$\mathbf{e}_1 = (1,1)/\sqrt{2} \qquad \lambda_1 = q_{\mathsf{S}} + q_{\mathsf{D}}$$
$$\mathbf{e}_2 = (1,-1)/\sqrt{2} \qquad \lambda_2 = q_{\mathsf{S}} - q_{\mathsf{D}}$$

so if $w_+ = w_{\mathsf{R}} + w_{\mathsf{L}}, w_- = w_{\mathsf{R}} - w_{\mathsf{L}}$ then

$$\tau_w \frac{dw_+}{dt} = (q_{\mathsf{S}} + q_{\mathsf{D}}) w_+ \qquad \tau_w \frac{dw_-}{dt} = (q_{\mathsf{S}} - q_{\mathsf{D}}) w_- \,.$$

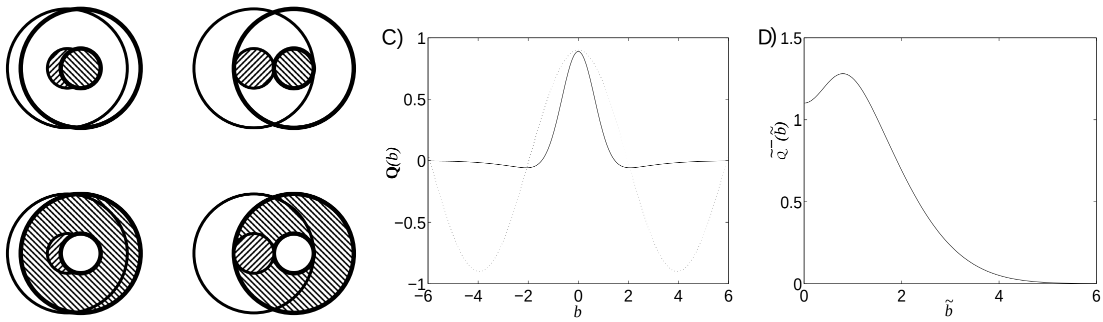
Since $q_{\mathsf{D}} \geq 0$, $w+$ dominates – so use subtractive normalisation

$$\tau_w \frac{dw_+}{dt} = 0 \qquad \tau_w \frac{dw_-}{dt} = (q_{\mathsf{S}} - q_{\mathsf{D}}) w_- \,.$$

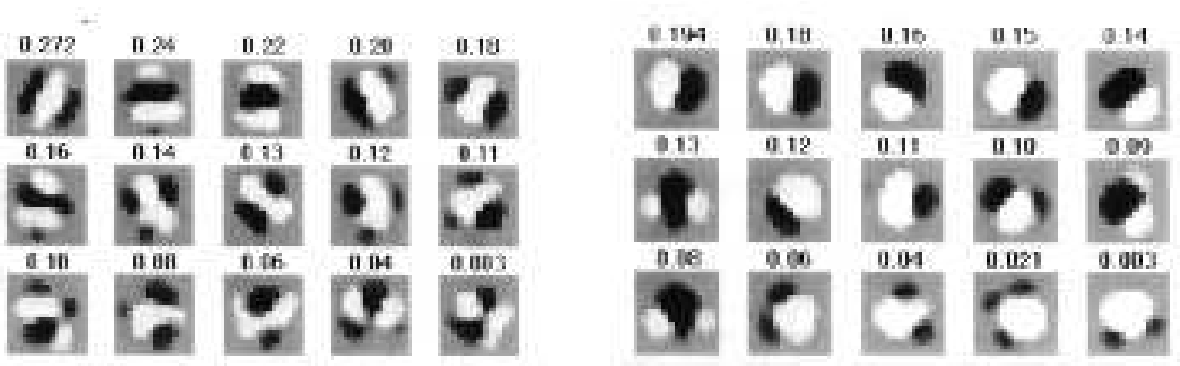
so $w_- \to \pm\omega$ and one eye dominates.

Orientation Selectivity

Model is exactly the same – input correlations come from ON/OFF cells:



Now dominant mode of $\mathbf{Q}^{-}$ has spatial structure:



centre-surround version also possible, but is usually dominated because of non-linear effects.

Temporal Hebbian Rules

Look at rate-based temporal model as

$$\mathbf{w} = \frac{1}{\tau_w} \int_0^T dt \, v(t) \int_{-\infty}^{\infty} d\tau \, H(\tau) \mathbf{u}(t - \tau)$$

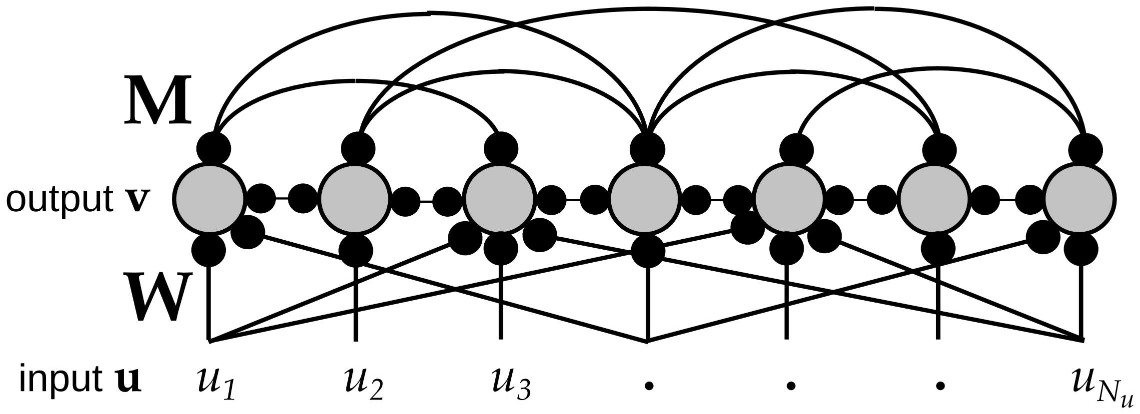
ignoring some edge effects.

Correlate

- output $v(t)$ with
- filtered version of the input $\int_{-\infty}^{\infty} d\tau \, H(\tau) \mathbf{u}(t - \tau)$

ie look for structure at the scale of the temporal filter

Multiple Output Neurons



Fixed recurrent connections

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

leads to

$$\begin{aligned} \mathbf{v} &= \mathbf{W} \cdot \mathbf{u} + \mathbf{M} \cdot \mathbf{v} \\ &= \mathbf{K} \cdot \mathbf{W} \cdot \mathbf{u} \end{aligned}$$

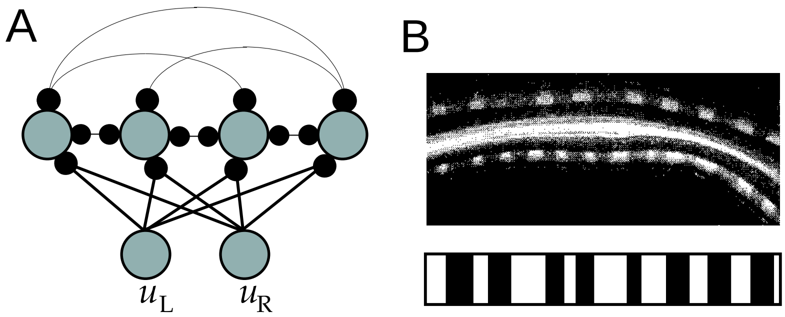
where $\mathbf{K} = (\mathbf{I} - \mathbf{M})^{-1}$.

Thus with Hebbian learning

$$\tau_w \frac{d\mathbf{W}}{dt} = \langle \mathbf{v}\mathbf{u} \rangle = \mathbf{K} \cdot \mathbf{W} \cdot \mathbf{Q}$$

and we can analyse the eigeneffect of $\mathbf{K}$.

Ocular Dominance Revisited

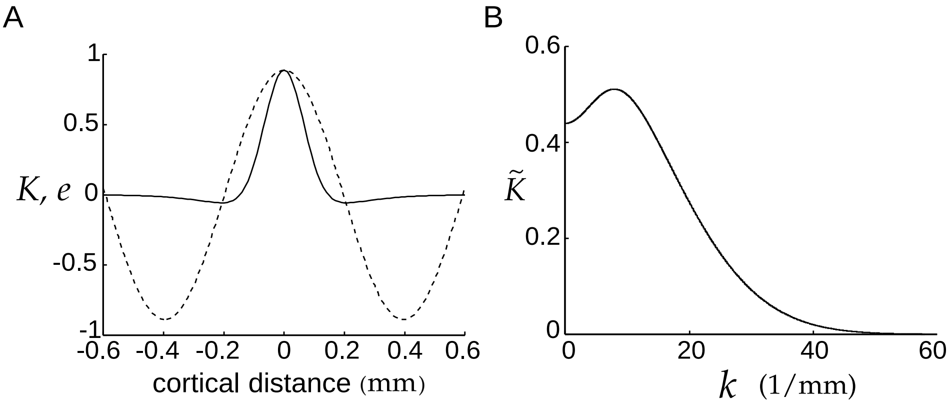


Write $\mathbf{w}_+ = \mathbf{w}_\mathsf{R} + \mathbf{w}_\mathsf{L}, \mathbf{w}_- = \mathbf{w}_\mathsf{R} - \mathbf{w}_\mathsf{L}$, for the *projective* weights, then

$$\tau_w \frac{d\mathbf{w}_+}{dt} = (q_\mathsf{S} + q_\mathsf{D})\mathbf{K} \cdot \mathbf{w}_+ \qquad \tau_w \frac{d\mathbf{w}_-}{dt} = (q_\mathsf{S} - q_\mathsf{D})\mathbf{K} \cdot \mathbf{w}_-$$

Since $\mathbf{w}_+$ is clamped by subtractive normalisation, just interested in the pattern of $\pm$ in $\mathbf{w}_-$.

Since $\mathbf{K}$ is Töplitz – eigenvectors are waves; eigenvalues come from the Fourier transform.



Comp Hebbian Learning

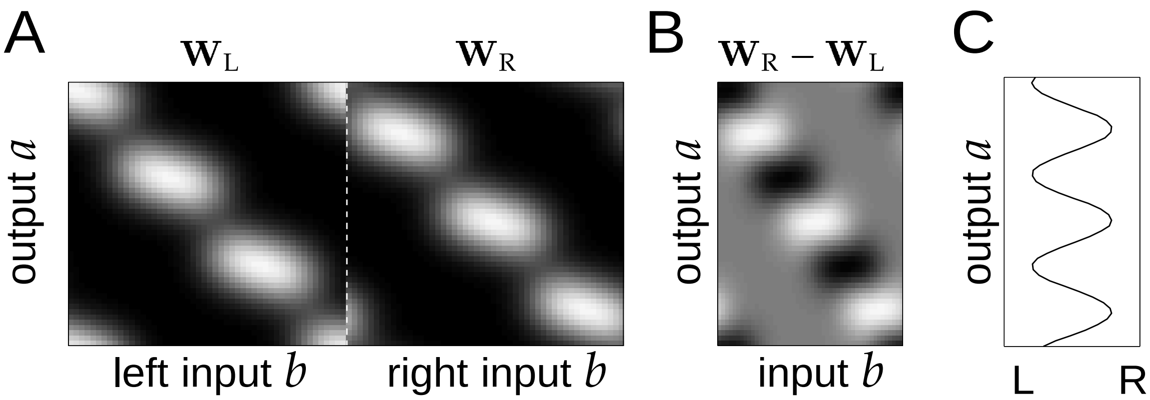
Use a competitive non-linearity

$$z_a = \frac{\left(\sum_b W_{ab} u_b\right)^\delta}{\sum_{a'} \left(\sum_b W_{a'b} u_b\right)^\delta}$$

in conjunction with a postive interaction term

$$v_a = \sum_{a'} M_{aa'} z_{a'} \,.$$

and standard Hebbian learning:



Features:

ocularity $\sum_b \mathbf{W}_-$

topography '$\sum_b \mathbf{W}_+ \vec{x}_b$'

Feature-Based Models

Reduced descriptions $(x, y, z, r\cos(\theta), r\sin(\theta))$

x, y topographic location

z ocularity ($\in [-1, 1]$)

r orientation *strength*

θ orientation

matching replace $[\mathbf{W} \cdot \mathbf{u}]_a$ by

$$\exp\left(-\sum_b (u_b - W_{ab})^2 / 2\sigma_b^2\right)$$

plus softmax competition and cortical interaction

learning *self organizing map*

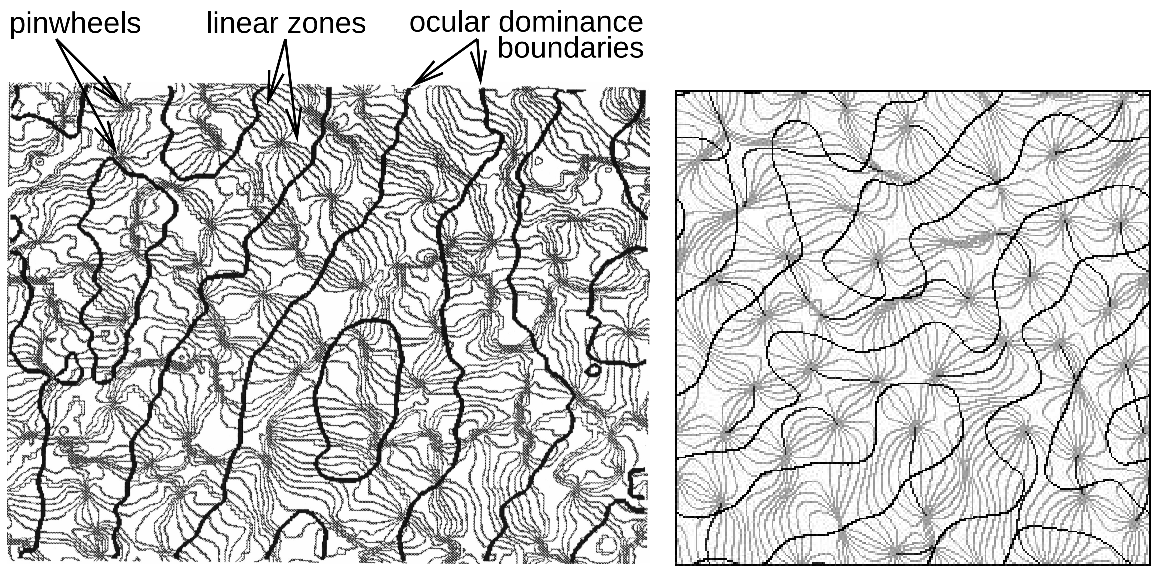
$$\tau_w \frac{dW_{ab}}{dt} = \langle v_a (u_b - W_{ab}) \rangle \, .$$

or *elastic net* – **only** competition and

$$\tau_w \frac{dW_{ab}}{dt} = \langle v_a (u_b - W_{ab}) \rangle + \beta \sum\nolimits_{a' \in \mathcal{N}(a)} (W_{a'b} - W_{ab})$$

Large-Scale Results

meshing of the patterns of OD and OR:



overall pattern of OD stripes *vs* elastic net simulation

Redundancy

Multiple units $\rightarrow$ redundancy:

- Hebbian learning – all units the same
- fixed output connections – inadequate

One possibility is decorrelation:

$$\langle \mathbf{v}\mathbf{v} \rangle = \mathbf{I}\,.$$

If Gaussian, then complete factorisation.

Three approaches:

Atick & Redlich force $n \rightarrow n$ mapping and decorrelate using anti-Hebbian learning.

Földiák use Hebbian and anti-Hebbian learning to learn feedforward and lateral weights.

Sanger explicitly subtract off first component from subsequent ones.

Williams subtract off predicted portion of $\mathbf{u}$

Goodall

$$\mathbf{v} = \mathbf{W} \cdot \mathbf{u} + \mathbf{M} \cdot \mathbf{v}$$

Anti-Hebbian learning is ideal for lateral weights:

- if v_a and v_b are correlated
- make $\mathbf{M}_{ab} = \mathbf{M}_{ba}$ negative
- which reduces the correlation

Goodall $n \to n$ with $\mathbf{W} = \mathbf{I}$ so:

$$\mathbf{v} = (\mathbf{I} - \mathbf{M})^{-1} \cdot \mathbf{x} = \mathbf{K} \cdot \mathbf{x}.$$

Then

$$\tau_M \dot{\mathbf{M}} = -\mathbf{uv} + \mathbf{I} - \mathbf{M}$$

At $\dot{\mathbf{M}} = \mathbf{0}$

$$\langle \mathbf{uu} \cdot \mathbf{K} \rangle = \mathbf{K}^{-1} \qquad \mathbf{K} \cdot \mathbf{Q} \cdot \mathbf{K} = \mathbf{I}.$$

So

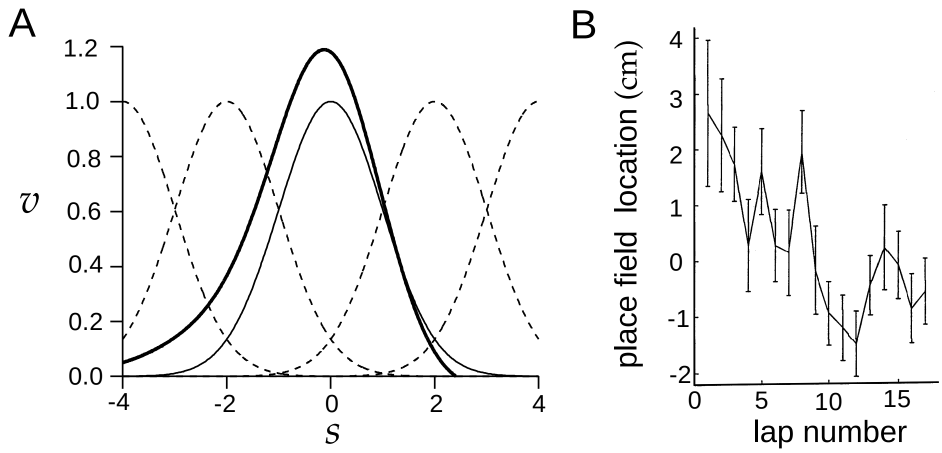
$$\langle \mathbf{uu} \rangle = \langle \mathbf{K} \cdot \mathbf{uu} \cdot \mathbf{K} \rangle = \mathbf{I}$$

as required.

Temporal Plasticity

Using the temporal rule:

$$\tau_w \frac{d\mathbf{w}}{dt} = \int_0^\infty d\tau \; (H(\tau)v(t)\mathbf{u}(t-\tau) + H(-\tau)v(t-\tau)\mathbf{u}(t))$$



- $s_a = -2$ is active before $s_a = 0$
- synapse $-2 \to 0$ gets strengthened
- $s_a = 0$ extends its firing field *backwards*

Supervised Learning

Consider case of learning pairs $\mathbf{u}^m, v^m$:

classification binary v^m to classify real-valued $\mathbf{u}^m$.

regression real-valued mapping from $\mathbf{u}^m$ to v^m.

storage learn the relationships in the data

generalisation infer a functional relationship from limited examples

error-correction mistakes drive adaptation

Hebbian plasticity:

$$\tau_w \frac{d\mathbf{w}}{dt} = \langle v\mathbf{u} \rangle = \frac{1}{N_\mathsf{S}} \sum_{m=1}^{N_\mathsf{S}} v^m \mathbf{u}^m \, .$$

and (multiplicative) weight decay

$$\tau_w \dot{\mathbf{w}} dt = \langle v\mathbf{u} \rangle - \alpha \mathbf{w} \, ,$$

makes $\mathbf{w} \to \langle v\mathbf{u} \rangle / \alpha$. No positive feedback.

Classification and the Perceptron

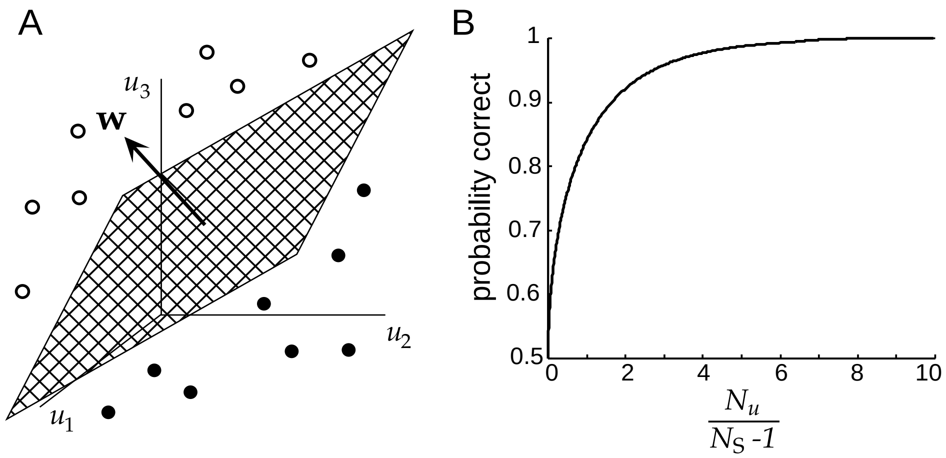
Classification rule

$$v = \begin{cases} 1 & \text{if} \quad \mathbf{w} \cdot \mathbf{u} - \gamma \geq 0 \\ 0 & \text{if} \quad \mathbf{w} \cdot \mathbf{u} - \gamma < 0 \end{cases}$$

Can use supervised Hebbian learning

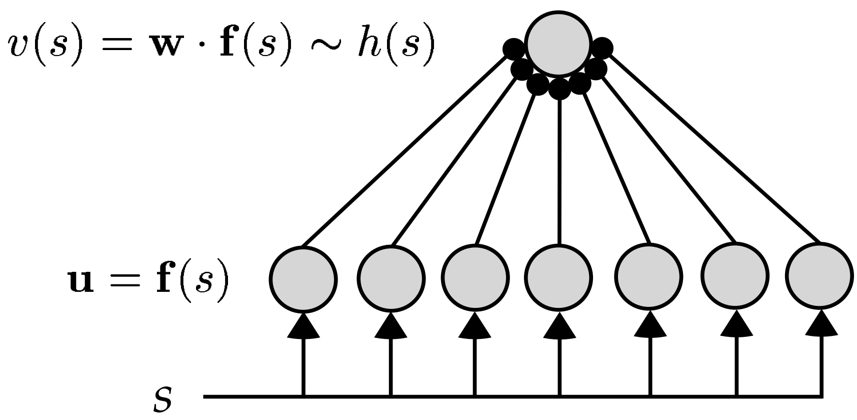
$$\mathbf{w} = \frac{2}{N_u} \sum_{m=1}^{N_S} v^m \mathbf{u}^m \, .$$

but works quite poorly for random patterns



Function Approximation

Basis function network



output $v(s) = \mathbf{w} \cdot \mathbf{u} = \mathbf{w} \cdot \mathbf{f}(s)$

error $E = \frac{1}{2} \left\langle (h(s) - \mathbf{w} \cdot \mathbf{f}(s))^2 \right\rangle$

reaches a minimum at (normal equations)

$$\langle \mathbf{f}(s)\mathbf{f}(s) \rangle \cdot \mathbf{w} = \langle \mathbf{f}(s) h(s) \rangle \ .$$

Hebbian Function Approximation

When does the Hebbian $\mathbf{w} = \langle \mathbf{f}(s)h(s)\rangle/\alpha$ satisfy the normal equations

$$\langle \mathbf{f}(s)\mathbf{f}(s)\rangle \cdot \mathbf{w} = \langle \mathbf{f}(s)h(s)\rangle \ ?$$

1. input patterns are orthongonal

$$\langle \mathbf{f}(s)\mathbf{f}(s)\rangle = \mathbf{I}$$

2. tight frame condition

$$\mathbf{f}(s^m) \cdot \mathbf{f}(s^{m'}) = c\delta_{mm'}$$

as then

$$\begin{aligned}\langle \mathbf{f}(s)\mathbf{f}(s)\rangle \cdot \mathbf{w} &= \frac{\langle \mathbf{f}(s)\mathbf{f}(s)\rangle \cdot \langle \mathbf{f}(s)h(s)\rangle}{\alpha} \\ &= \frac{1}{\alpha N_{\mathsf{S}}^2}\sum_{mm'} \mathbf{f}(s^m)\mathbf{f}(s^m) \cdot \mathbf{f}(s^{m'})h(s^{m'}) \\ &= \frac{c}{\alpha N_{\mathsf{S}}^2}\sum_{m} \mathbf{f}(s^m)h(s^m) \\ &= \frac{c}{\alpha N_{\mathsf{S}}}\langle \mathbf{f}(s)h(s)\rangle\end{aligned}$$

V1 forms an approximate tight frame

Error-Correcting Rules

Hebbian plasticity is independent of the performance of the network

Perceptron learning rule:

- if $v(\mathbf{u}^m) = 0$ when $v^m = 1$,
- modify $\mathbf{w}$ and γ to increase $\mathbf{w} \cdot \mathbf{u}^m - \gamma$

easiest rule:

$$\mathbf{w} \rightarrow \mathbf{w} + \epsilon_w \left(v^m - v(\mathbf{u}^m)\right) \mathbf{u}^m$$
$$\gamma \rightarrow \gamma - \epsilon_w (v^m - v(\mathbf{u}^m))$$

implies that

$$\Delta \left(\mathbf{w} \cdot \mathbf{u}^m - \gamma\right) = \epsilon_w (v^m - v(\mathbf{u}^m)) \left(|\mathbf{u}^m|^2 + 1\right)$$

which has just the right sign. In fact, guaranteed to converge.

note the discrete nature of the weight update

The Delta Rule

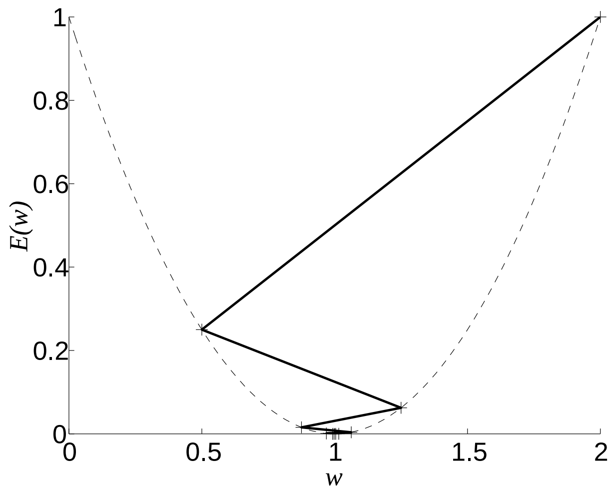
Defintion of the task in $E(\mathbf{w})$ – how well (poorly) do synaptic weights $\mathbf{w}$ perform?

Gradient descent:

$$\mathbf{w} \to \mathbf{w} - \epsilon_w \nabla_{\mathbf{w}} E(\mathbf{w})$$

since if $\mathbf{w}' = \mathbf{w} - \epsilon \nabla_{\mathbf{w}} E(\mathbf{w})$, then to first order in ϵ_w:

$$\begin{aligned} E(\mathbf{w} - \epsilon_w \nabla_{\mathbf{w}} E) &= E(\mathbf{w}) - \epsilon_w \left|\nabla_{\mathbf{w}} E\right|^2 \\ &\leq E(\mathbf{w}) \end{aligned}$$



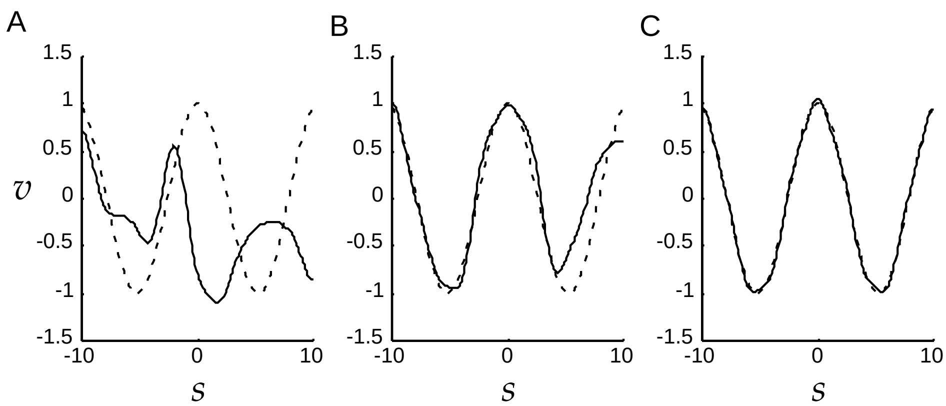
Stochastic Gradient Descent

$E(\mathbf{w}) = \frac{1}{2}\left\langle (h(s) - \mathbf{w}\cdot\mathbf{f}(s))^2 \right\rangle$ is an average over many examples.

Use random input-output paris $s^m, h(s^m)$ and change

$$\mathbf{w} \to \mathbf{w} - \epsilon_w \nabla_{\mathbf{w}} (h(s^m) - v(s^m))^2/2$$
$$= \mathbf{w} + \epsilon_w (h(s^m) - v(s^m))\mathbf{f}(s^m)$$

called stochastic gradient descent.



Contrastive Hebbian Learning

The delta rule

$$\mathbf{w} \to \mathbf{w} + \epsilon_w \left(v^m \mathbf{u}^m - v(\mathbf{u}^m)\mathbf{u}^m\right)$$

involves:

Hebbian learning $v^m \mathbf{u}^m$ based on *target*

anti-Hebbian learning $-v(\mathbf{u}^m)\mathbf{u}^m$ based on *outcome*

learning stops when outcome = target

Generalize to a *stochastic* network

$$P[\mathbf{v}|\mathbf{u};\mathbf{W}] = \frac{\exp(-E(\mathbf{u},\mathbf{v}))}{Z(\mathbf{u})}$$
$$Z(\mathbf{u}) = \sum_{\mathbf{v}} \exp(-E(\mathbf{u},\mathbf{v}))$$

weights $\mathbf{W}$ generate a *conditional* distribution
eg with quadratic form $E(\mathbf{u},\mathbf{v}) = \mathbf{u} \cdot \mathbf{W} \cdot \mathbf{v}$

Goal of Learning

Natural quality measure for $\mathbf{u}$:

$$D_{\mathsf{KL}}(P[\mathbf{v}|\mathbf{u}], P[\mathbf{v}|\mathbf{u};\mathbf{W}]) = \sum_{\mathbf{v}} P[\mathbf{v}|\mathbf{u}] \ln\left(\frac{P[\mathbf{v}|\mathbf{u}]}{P[\mathbf{v}|\mathbf{u};\mathbf{W}]}\right)$$

$$= -\sum_{\mathbf{v}} P[\mathbf{v}|\mathbf{u}] \ln\left(P[\mathbf{v}|\mathbf{u};\mathbf{W}]\right) + K\,,$$

average over $\mathbf{u}^m$; $\mathbf{v}^m$ is *sample* of $P[\mathbf{v}|\mathbf{u}^m]$

$$\left\langle D_{\mathsf{KL}}(P[\mathbf{v}|\mathbf{u}], P[\mathbf{v}|\mathbf{u};\mathbf{W}])\right\rangle \sim -\frac{1}{N_{\mathsf{S}}}\sum_{m=1}^{N_{\mathsf{S}}} \ln\left(P[\mathbf{v}^m|\mathbf{u}^m;\mathbf{W}]\right)$$

amounts to maximum likelihood learning.

$$\frac{\partial \ln P[\mathbf{v}^m|\mathbf{u}^m;\mathbf{W}]}{\partial W_{ab}} = \frac{\partial}{\partial W_{ab}}\left(-E(\mathbf{u}^m,\mathbf{v}^m) - \ln Z(\mathbf{u}^m)\right)$$

$$= v_a^m u_b^m - \sum_{\mathbf{v}} P[\mathbf{v}|\mathbf{u}^m;\mathbf{W}] v_a u_b^m\,.$$

is also Hebb – ⟨anti-Hebb⟩
positive – ⟨negative⟩

use Gibbs sampling for $\mathbf{v}^- \sim P[\mathbf{v}|\mathbf{u}^m;\mathbf{W}]$

unsupervised version is just the same

Function Approximation

Two ignored concerns:

1. examples may be corrupted by noise

$$\mathcal{D} \equiv \{(s^m, v^m)\} \quad v^m = h(s^m) + \eta^m$$

2. want $v(s) = h(s)$ even *away* from training set – in general impossible – need *prior* information *eg* smoothness of $h(\mathbf{s})$

Gaussian process says that given $s^1, s^2, \ldots, s^K$, $h(s^1), h(s^2), \ldots, h(s^K)$ are jointly Gaussian with

- **0** mean (short of expectations about bias
- covariance matrix determined by prior expectations *eg*

$$\Sigma_{\mu\nu} = \Sigma\left(|s^\mu - s^\nu|\right)$$

Then distribution of $h(s)$ given $\mathcal{D}$ is Gaussian with mean that depends linearly on $v^1, \ldots, v^{N_\mathrm{s}}$.

Functional Priors

Alternatively, specify:

$$P[h] \propto e^{-\frac{1}{2}\phi[h]},$$

where $\phi[h]$ is a *functional, eg:*

$$\phi[h] = \int_{\tilde{s}} d\tilde{s} \, \frac{|\tilde{h}(\tilde{s})|^2}{\tilde{\mathcal{G}}(\tilde{s})}$$

in the Fourier domain. $\tilde{\mathcal{G}}(\tilde{s})$ penalises high frequency wobbles in $h(s)$.

$$\tilde{\mathcal{G}}(\tilde{s}) = e^{-|\tilde{s}|^2/\delta}$$

If η^μ are iid Gaussians, then likelihood is:

$$P[\mathcal{D}|h] \propto \exp\left(-\frac{1}{2\sigma^2}\sum_{m=1}^{N_\mathrm{S}}(v^m - h(s^m))^2\right)$$

So posterior distributions over $h(s)$ is:

$$\begin{aligned}\log P[h|\mathcal{D}] &= \log P[\mathcal{D}|h]P[h] + \mathcal{K} \\ &= -\frac{1}{2\sigma^2}\sum_{m=1}^{N_\mathrm{S}}(v^m - h(s^m))^2 - \frac{1}{2}\phi[h] + \mathcal{K}'\end{aligned}$$

Regularization Theory

Find MAP solution by minimising:

$$E[h] = \frac{1}{\sigma^2} \sum_{m=1}^{N_S} (v^m - h(s^m))^2 + \phi[h] \,,$$

Turns out that

$$h(s) = \sum_{m=1}^{N_S} w_m \mathcal{G}(s - s^m) + k(s)$$

where $\mathcal{G}(s - s^\mu)$ is IFT of smoothness prior and $k(s)$ satisfies $\phi[k] = 0$. The weights are given by

$$\mathbf{w} = (\mathbf{G} + \sigma^2 \mathbf{I})^{-1} \mathbf{v}.$$

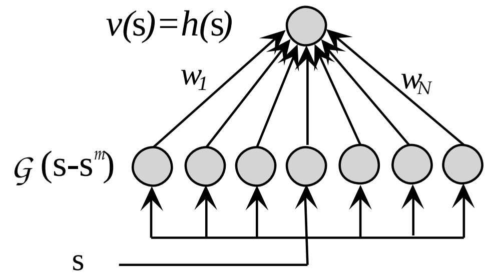
where $\mathbf{G}_{mn} = \mathcal{G}(s^m - s^n)$ recovering linear dependence on $\mathbf{v}$.

For Gaussian $\tilde{\mathcal{G}}(\tilde{s})$,

$$\mathcal{G}(s - s^m) \propto e^{-\delta |s - s^m|^2}$$

is a radial basis function:

Regularization Theory



- one basis function for each example
- basis functions determined by prior
- as $\sigma^2 \to 0$, MAP solution interpolates y^μ
- use Bayesian hyperparameters to manipulate the norm
- fixed basis functions $\Rightarrow$ inference over weights:

$$E(\mathbf{w}) = \frac{1}{\sigma^2} \sum_{m=1}^{N_S} (v^m - h(s^m))^2 + \phi(\mathbf{w}),$$

eg $\phi(\mathbf{w}) = \|\mathbf{w}\|$ is Gaussian

- or learn the basis functions themselves using backprop.