Lecture 2: Feedforward networks

Jochen Braun

Otto-von-Guericke-Universität Magdeburg,
Cognitive Biology Group

Theoretical Neuroscience II, SS 2020

Credits:

Dayan & Abbot (2001), "Theoretical Neuroscience", MIT Press, Chapter 7.

Lecture 2: Feedforward networks

Classification: *Feedforward networks map many input patterns onto few output patterns. Thus, they perform classifications.* **Perceptron:** *A many-to-one network with a 'hard' activation function. It assigns input patterns to just two classes (binary output). This classifier is linear and a hyperplane in vector space of input patterns.*
Eye-hand coordination: *Projections between visual and motor cortex necessitate a change of coordinates (eye-coordinates to head-coordinates). Neurobiological FFN perform such coordinate transforms with* **gain modulation.** *We illustrate this by combining angular position s (in eye-coordinates) with gaze angle g to obtain angular position $s + g$ (in head-coordinates).*

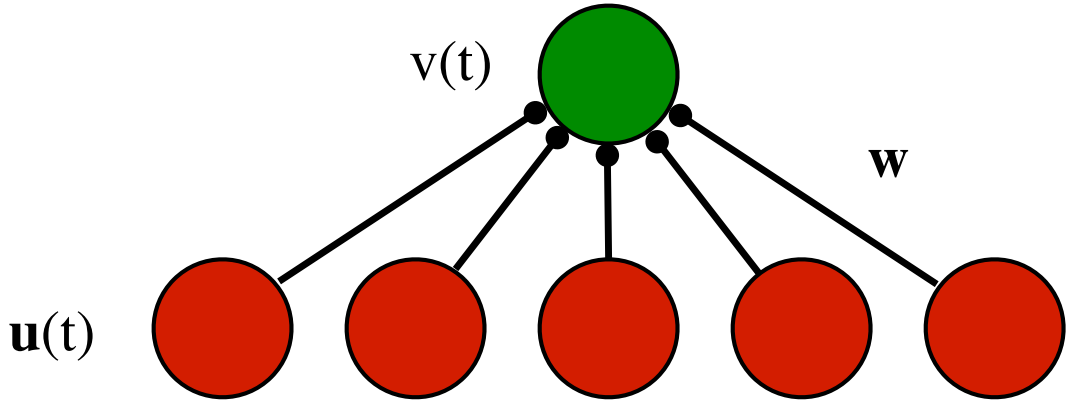
Organisation of lecture

- 1. Feedforward networks for classification
- 2. Perceptrons
- 3. Eye-hand coordination
- 4. Gain modulation
- 5. Generalization

1. Feedforward networks for classification

- *Feedforward networks* map activity patterns (inputs) onto other activity patterns (output).
- A *classification* maps many activity patterns (inputs) onto a few *classes* (categories).
- What *classifications* can be performed by *feedforward networks*?

Feedforward many-to-one



In a feedforward network, N_u input units with rates u_i converge via synapses with weights w_i on a single output units with rate v. For convenience, we collect input units and synaptic weight into vectors $\boldsymbol{u}$ and $\boldsymbol{w}$. The output rate is then approximated by

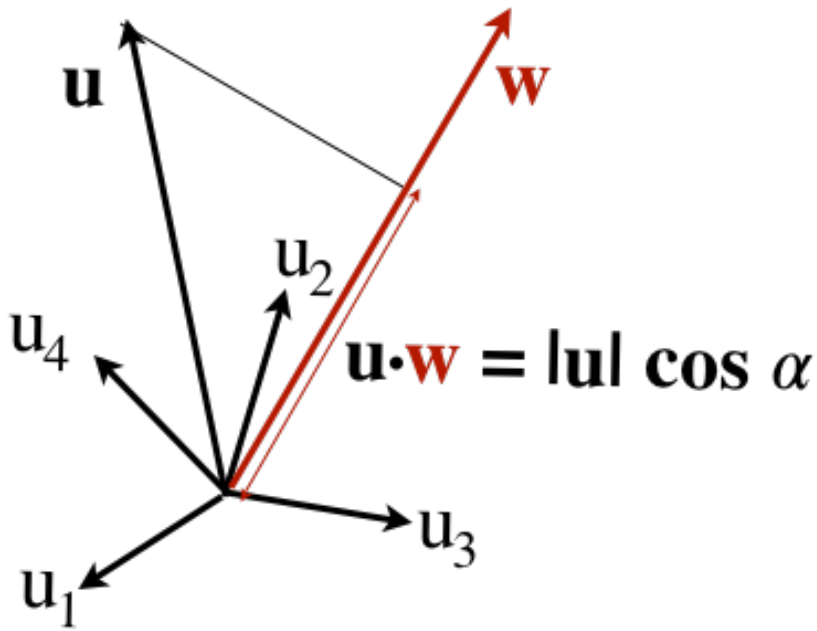
$$\tau \frac{dv}{dt} = -v + F\left(\boldsymbol{w} \cdot \boldsymbol{u}\right) \qquad \text{or} \qquad \tau_r \frac{dv}{dt} = -v + F\left(\sum_{i=1}^{N_u} w_i\, u_i\right)$$

The steady-state firing rate, v_∞, is

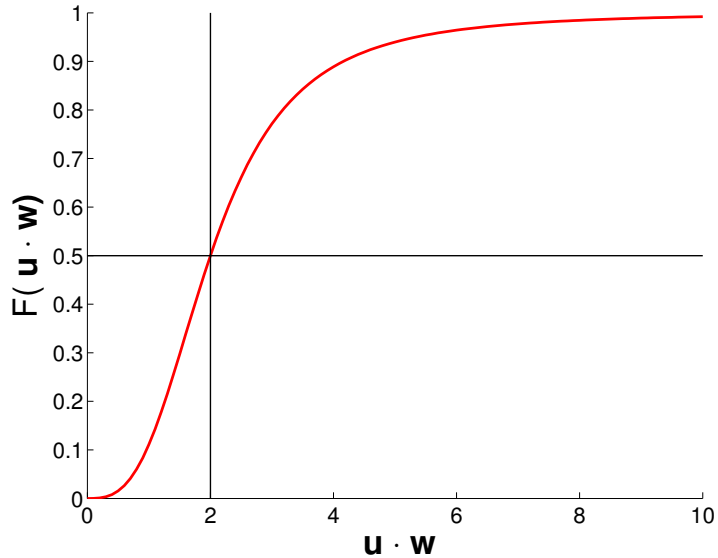
$$v_{ss} = F\left(\boldsymbol{w} \cdot \boldsymbol{u}\right)$$

Vector space interpretation

Consider an N_i-dimensional vector space, spanned by input activities $\boldsymbol{u} = \{u_i\}$. The weights $\boldsymbol{w} = \{w_i\}$ define a vector of unit length in this space ($|w| = 1$).
Dot product $\boldsymbol{u} \cdot \boldsymbol{w} = |u| \cos\alpha$ is the (positive or negative) length of the *projection* of the input vector $\boldsymbol{u}$ onto the weight vector $\boldsymbol{w}$.



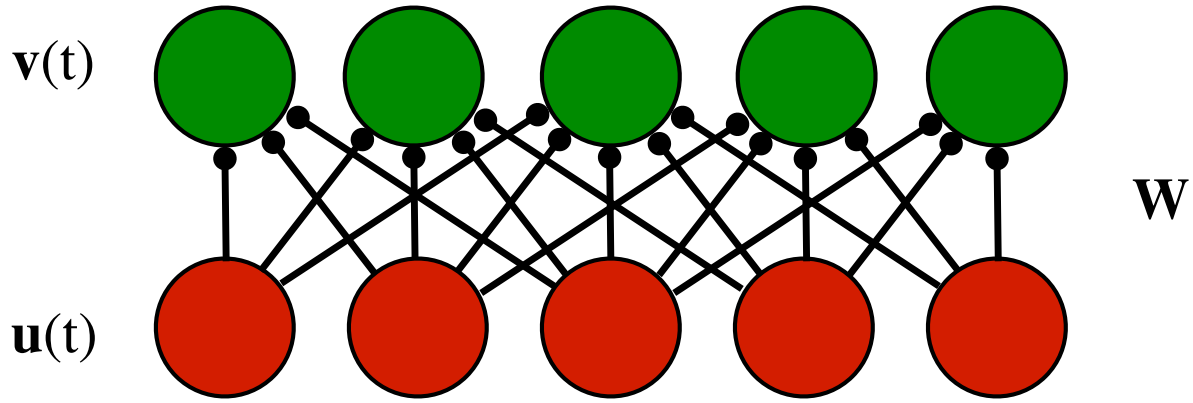
ctd



The steady-state response $v_{ss} = F(\boldsymbol{w} \cdot \boldsymbol{u})$ represents how well input vector $\boldsymbol{u}$ aligns with weight vector $\boldsymbol{w}$. If the alignment is sufficiently good, the steady-state response approaches unity. If it is not, the steady-state response remains near zero.

Thus, a feedforward projection computes how well the input vector matches the weight vector!

Feedforward many-to-many



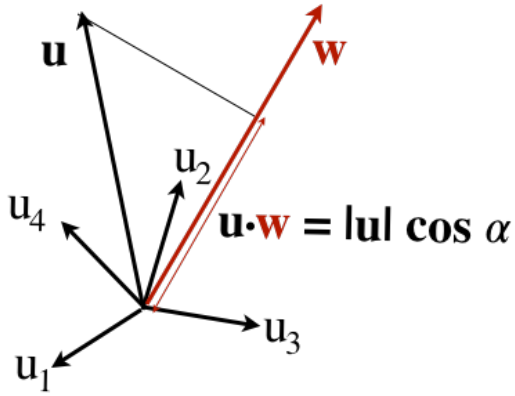
In a feedforward network, N_u input units with rates u_i converge via synapses with weights w_{io} onto N_v output units with rates v_o. For convenience, input and output units are collected in vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, and the synaptic weights into a matrix $\boldsymbol{W}$. The output rate is then approximated by

$$\tau \frac{d\boldsymbol{v}}{dt} = -\boldsymbol{v} + F\left(\boldsymbol{W} \cdot \boldsymbol{u}\right) \qquad \text{or} \qquad \tau_r \frac{dv_o}{dt} = -v_o + F\left(\sum_{i=1}^{N_u} w_{io}\, u_i\right)$$

The steady-state firing rate, $\boldsymbol{v}_{ss}$, is

$$\boldsymbol{v}_{ss} = \boldsymbol{F}\left(\boldsymbol{W} \cdot \boldsymbol{u}\right)$$

Vector space interpretation



For each output unit o, the steady-state response is determined by a vector dot product (independently of other units)

$$v_o^{ss} = F\left(\boldsymbol{w}_o \cdot \boldsymbol{u}\right) = \left[w_{1o} w_{2o} \ldots w_{N_i o}\right] \cdot \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N_i} \end{bmatrix}$$

The steady-state response depends on the *projection* of input vector $\boldsymbol{u}$ onto the weight vector $\boldsymbol{w}_o$ (*i.e.*, the o^{th} row of the weight matrix $\boldsymbol{W}$).

Summary

- A many-to-one *feedforward network* maps input activity vector $\boldsymbol{u}$ onto scalar output activity v (steady-state).
- A many-to-many *feedforward network* maps input activity vector $\boldsymbol{u}$ onto output activity vector $\boldsymbol{v}$ (steady-state).
- The mapping is determined by a synaptic vector $\boldsymbol{w}$ or a synaptic matrix $\boldsymbol{W}$:

$$v_{ss} = F\left(\boldsymbol{w} \cdot \boldsymbol{u}\right) \qquad \boldsymbol{v}_{ss} = F\left(\boldsymbol{W} \cdot \boldsymbol{u}\right)$$

- Thus, feedforward networks perform *vector dot product* operations.
- This insight is crucial for understanding the *classifications* that can be performed by feedforward networks.

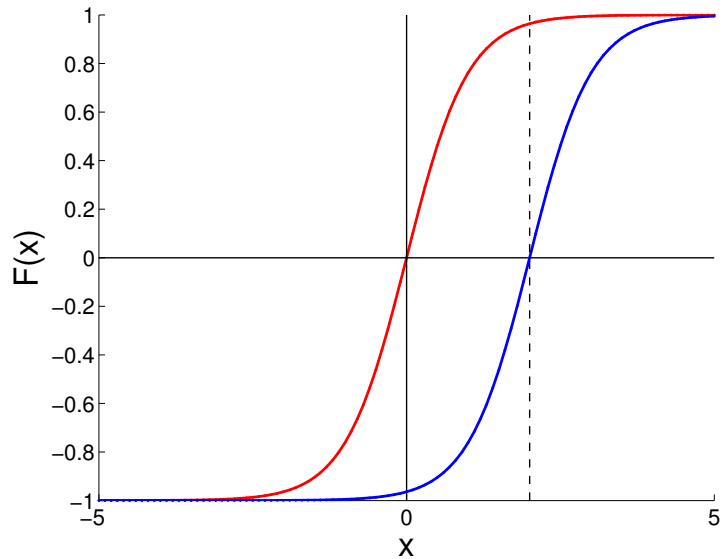
2. Perceptrons

- Idealized feedforward networks can function as *binary classifiers*, mapping input vectors onto binary values of ± 1 (neglecting 0) .
- Binary classifiers were called *perceptrons* by their inventor, Frank Rosenblatt (1958).
- Minsky and Papert showed that single-layer perceptron can distinguish only linearly separable patterns (1969, "Perceptrons: an introduction to computational geometry," MIT Press.)
- Multi-layer perceptrons are more capable.
- Perceptrons continue to be useful, but are no longer considered a valid model for cognitive function.

tanh activation function

What can and cannot perceptrons do? To understand, we simplify in several steps. As a first simplification, we replace the activation with the symmetric sigmoidal function tanh

$$F(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



The threshold and the limits of the tanh are

$$\lim_{x \to 0} \tanh(x) = \frac{1-1}{1+1} = 0 \qquad \lim_{x \to +\infty} \tanh(x) = \frac{e^x - 0}{e^x + 0} = 1 \qquad \lim_{x \to -\infty} \tanh(x) = \frac{0 - e^x}{0 + e^x} = -1$$

Hyperbolic tangent

With a tanh function, the steady-state mapping of a many-to-one feedforward network becomes

$$v = \tanh\left(\boldsymbol{w} \cdot \boldsymbol{u}\right)$$

Positive (negative) dot products $\boldsymbol{w} \cdot \boldsymbol{u}$ map onto positive (negative) outputs v.

With a threshold $\boldsymbol{b}$, additional mappings become possible

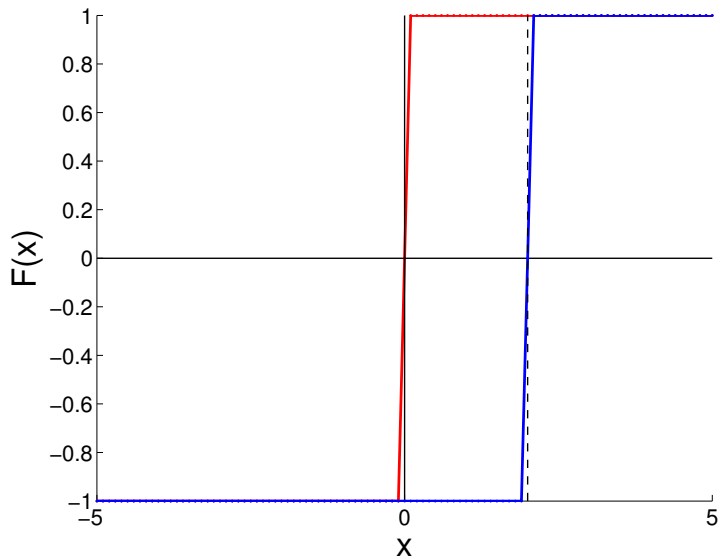
$$v = \tanh\left[\boldsymbol{w} \cdot \boldsymbol{u} + \boldsymbol{b}\right]$$

In both cases, all possible inputs u are mapped onto two output *classes* (positive and negative).

Sign function

Even easier to analyze are activation functions that approach the "hard delimiter" $sign(x)$

$$sign(x) = \begin{cases} +1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \qquad \lim_{\gamma \to \infty} \tanh(\gamma x) = sign(x)$$



The mapping equation becomes

$$v = \tanh\left[\boldsymbol{w} \cdot \boldsymbol{u}\right] = \tanh\left[\gamma\ \left(\gamma^{-1}\boldsymbol{w} \cdot \boldsymbol{u}\right)\right] \approx sign\left(\gamma^{-1}\boldsymbol{w} \cdot \boldsymbol{u}\right)$$

or

$$v = \tanh\left[\boldsymbol{w} \cdot \boldsymbol{u} - \boldsymbol{b}\right] \approx sign\left(\gamma^{-1}\boldsymbol{w} \cdot \boldsymbol{u} - \gamma^{-1}\boldsymbol{b}\right)$$

Hard feedforward networks

The steady-state mappings of many-to-one and many-to-many feedforward networks with a "hard" activation function are:

Many-to-one: $v^\mu = sign\left(\boldsymbol{m} \cdot \boldsymbol{u}^\mu\right)$ $\boldsymbol{m} = \gamma^{-1}\boldsymbol{w}$

Many-to-many: $\boldsymbol{v}^\mu = \mathbf{sign}\left(\boldsymbol{M} \cdot \boldsymbol{u}^\mu\right)$ $\boldsymbol{M} = \gamma^{-1}\boldsymbol{W}$

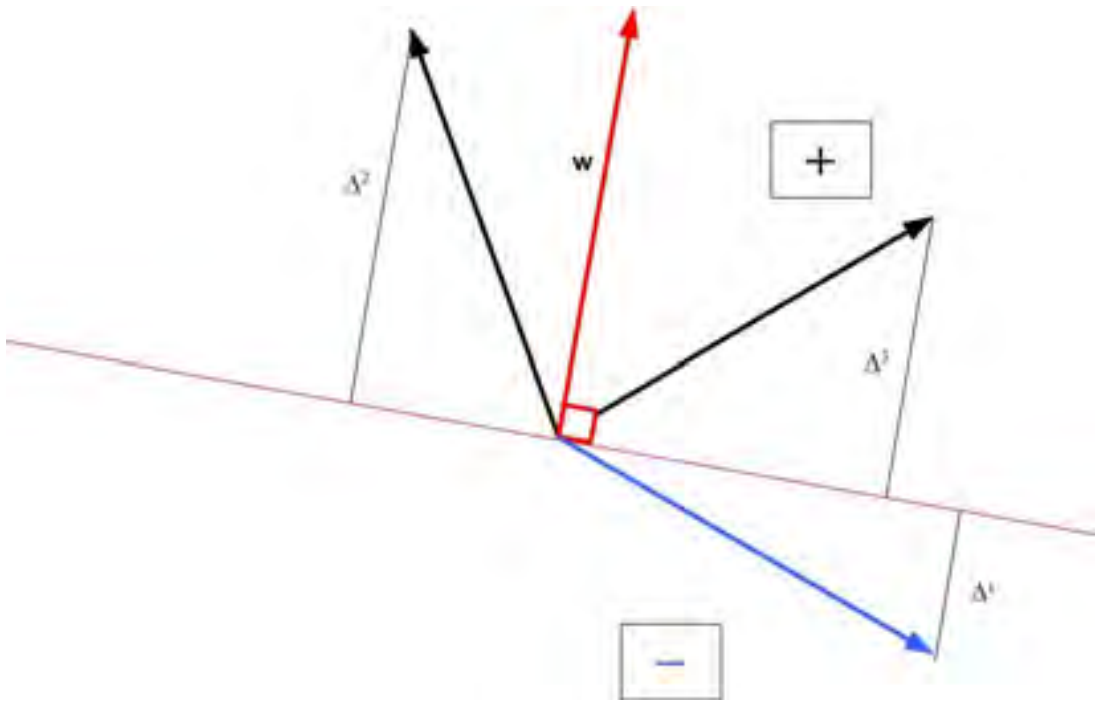
where μ indexes different input patterns, and γ is the amplification factor that turns *tanh* into *sign* (see previous slide).

Output vector $\boldsymbol{v}$ assume only values $\in \{-1, 1\}$. The value 0 occurs too rarely to matter.

Vector space interpretation

The weights $\boldsymbol{w}$ define a normal plane that classifies (almost) any input vector $\boldsymbol{u}^\mu$ into one of two "point classes":

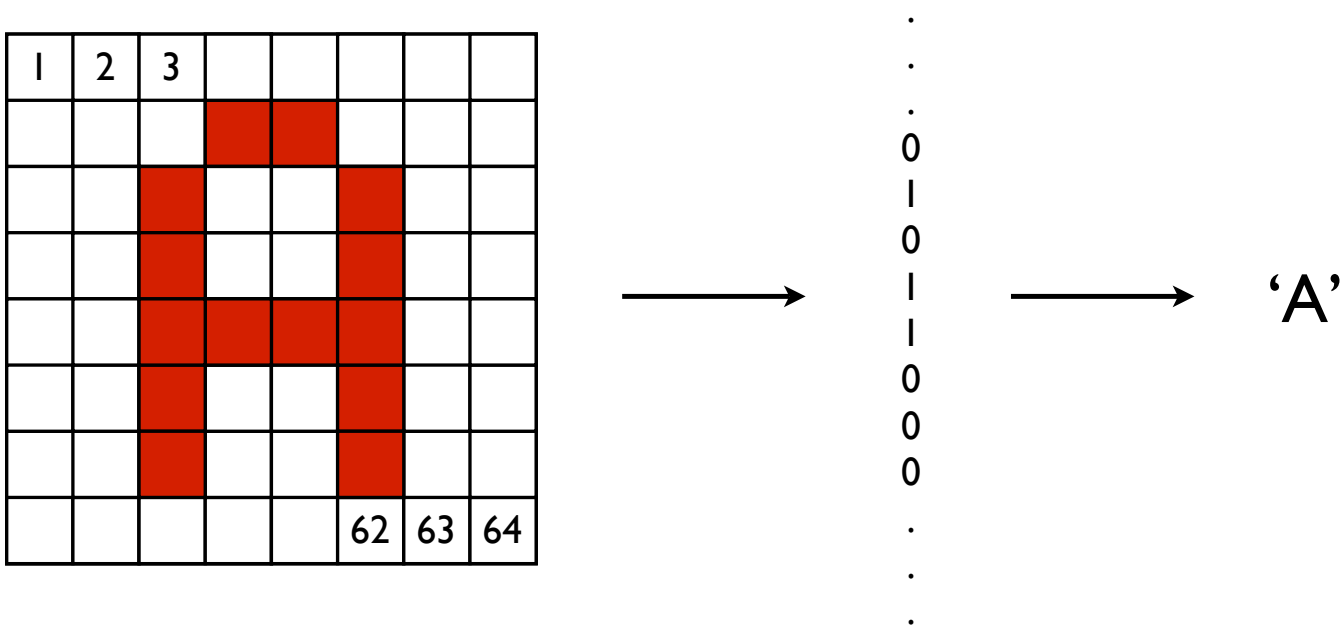
$$\boldsymbol{u}^\mu \to v^\mu \in \pm 1$$



A threshold shifts the plane away from the origin (see below).

What kinds of mappings/classifications are possible?

Consider a vector of sensory features as an input for object recognition:



Can a feedforward network perform such a classification?

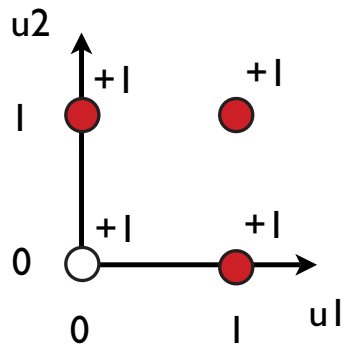
Binary mappings ('dichotomies')

Consider four possible activity patterns $\boldsymbol{u}$, with $u_{1,2} \in \{0, 1\}$. Define desired binary mapping ('dichotomy') $\boldsymbol{u} \to \{-1, +1\}$, either as lookup table or in vector space:

lookup table

u1	u2	v
0	0	-1
1	0	+1
0	1	+1
1	1	+1

vector space



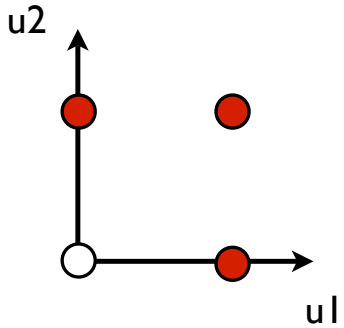
Seek to realize with binary perceptron

$$v = sign\left(\boldsymbol{w} \cdot \boldsymbol{u}^{\mu} + \boldsymbol{b}\right)$$

Perceptron works ...

Define desired mapping (left), seek to realize with binary perceptron (right):

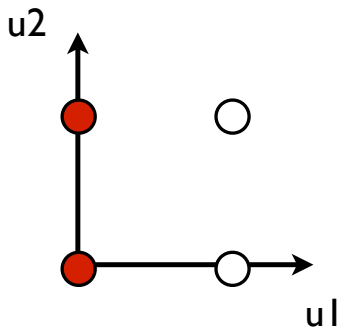
u1	u2	v



$$v = \text{sign}(\mathbf{w}\,\mathbf{u} + b)$$

$$w = (1,1) \qquad b = -0.5$$

u1	u2	v

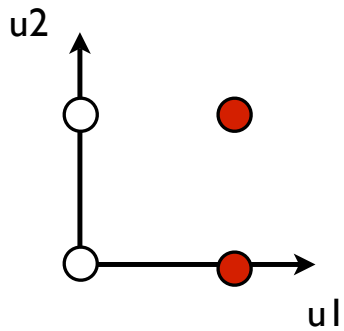


$$w = (-1,0) \qquad b = 0.5$$

... but not always ...

Define desired mapping (left), seek to realize with binary perceptron (right):

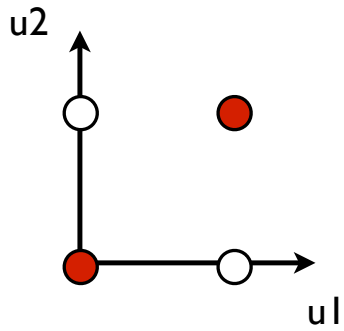
u1	u2	v
■		■
	■	
■	■	■



$$v = \text{sign}(\mathbf{w}\,\mathbf{u} + b)$$

$$w = (1,0) \qquad b = -0.5$$

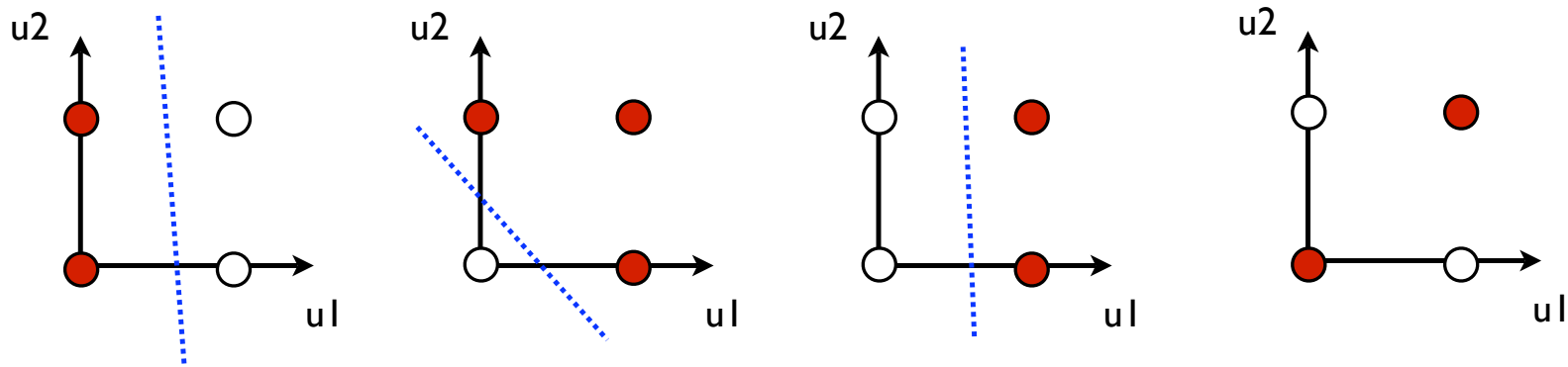
u1	u2	v
		■
■		
	■	
■	■	■



$$w = (?,?) \qquad b = ?$$

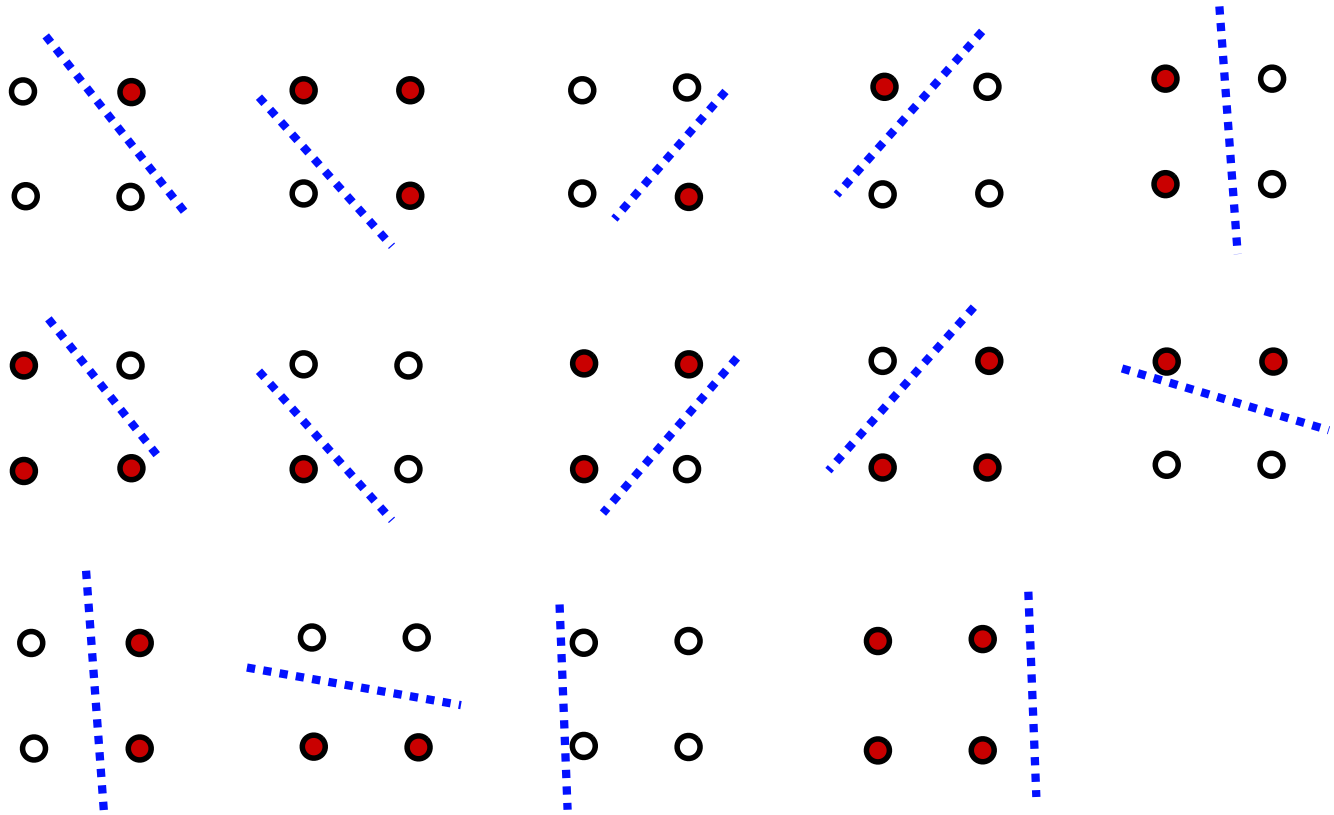
Linear separability

We find that perceptrons can realize only linearly separable mapping functions.
For two plus one input dimensions (see below), perceptrons can realize 14 out of 16 possible mappings.



(2 input dimensions, 4 input vectors, $2^4 = 16$ dichotomies).

Two binary inputs, with 16 possible mappings onto binary outputs. 14 mappings are linearly separable and can be realized with a ‘hard feedforward’ network.



Points to note

- In their most abstract form, feedforward networks become binary classifiers.

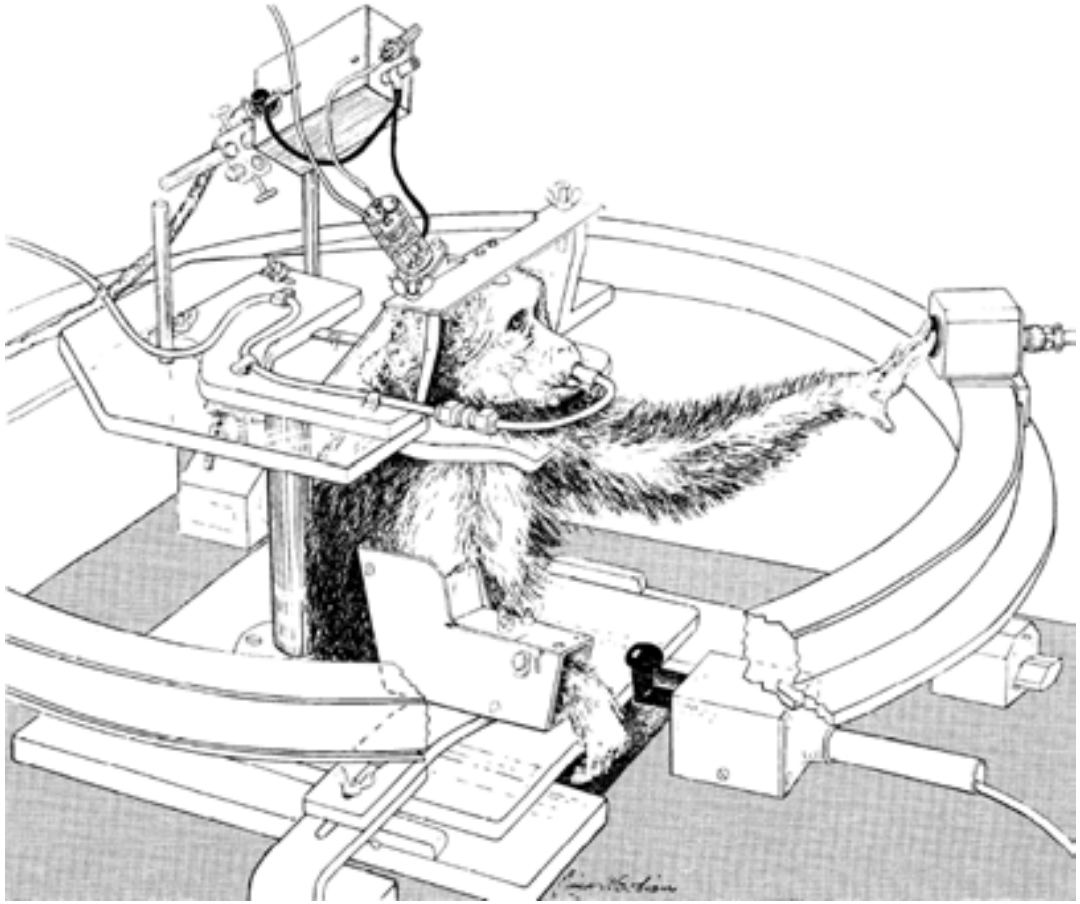
- They map any input vector $\boldsymbol{u}^{\mu}$ onto one of two output classes: $\{+1, -1\}$.
- The mapping has a straightforward vector interpretation: weights $\boldsymbol{w}$ define a normal plane (hyperplane), bisecting the input space.
- Inputs are mapped to either one or the other side of this plane.

- Feedforward networks can perform only *linearly separable* classifications, severely restricting their usefulness.

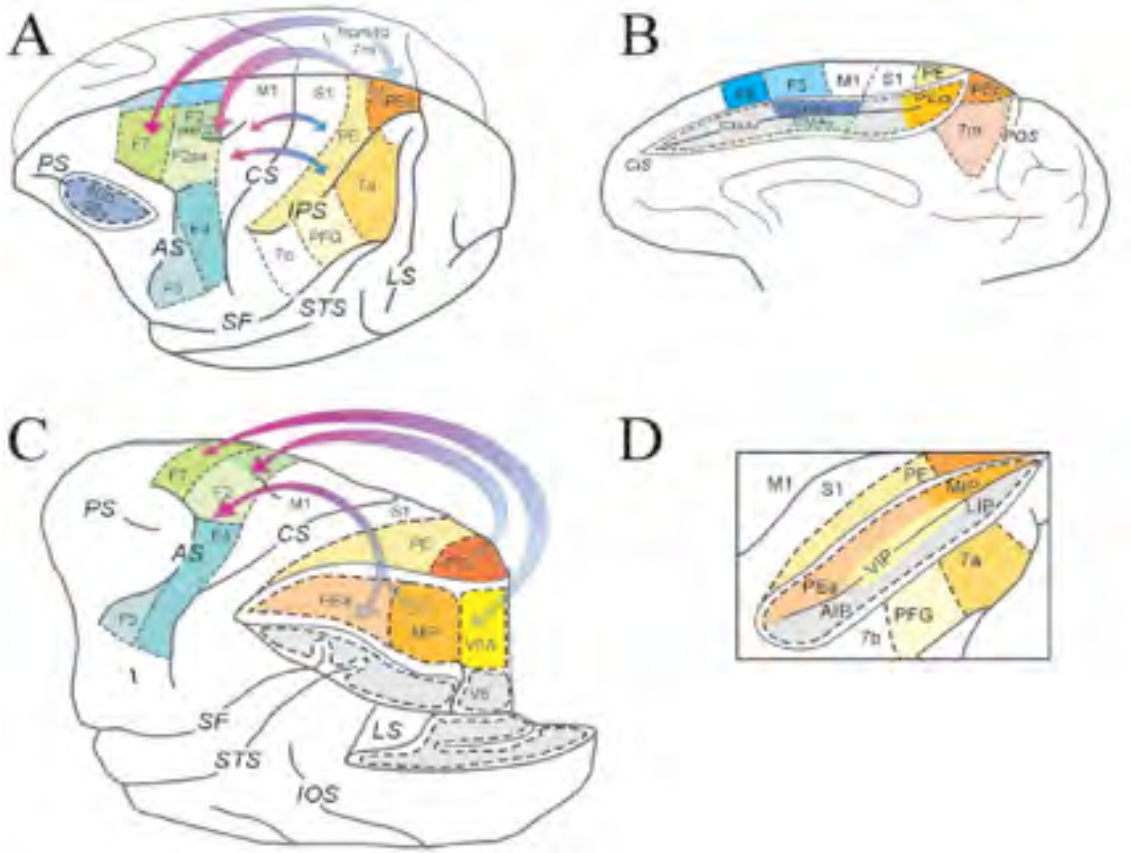
3. Eye-hand coordination

Reaching movements are visually guided. Visual information is acquired in retinal coordinates. Limb movements are controlled in body coordinates. Thus, eye-hand coordination involves coordinate transforms.

Visual guidance of reaching movements has been studied since ca. 1972. Vernon Mountcastle pioneered techniques for studying cortical activity in 'awake behaving' primates (macaque monkeys).

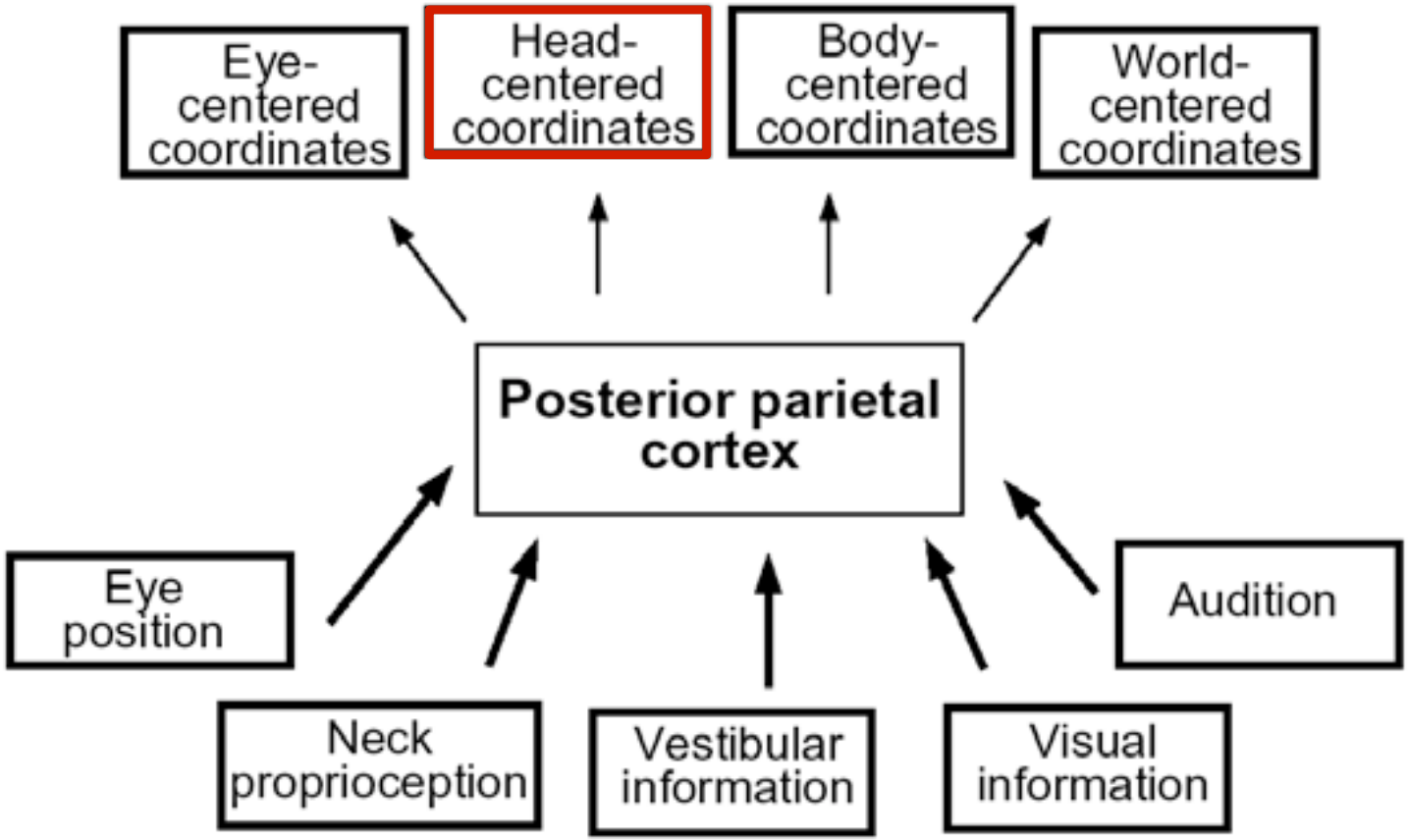


Visual guidance of reaching movements involves (i) visual areas in posterior parietal cortex and (ii) frontal areas in premotor cortex. The receptive fields of the former are in EYE coordinates, those of the latter in BODY coordinates.



The reference frame problem

- Translating from perception to action encounters a 'reference frame problem'.
- Visual information originates in EYE-centered coordinates. It is *ambiguous* and does not determine relative motion of body and environment.
- To determine relative motion, additional information is needed: gaze direction and vestibular inputs.
- Visual information is transformed into HEAD-centered coordinates by taking gaze direction into account.
- This format is more suitable for guiding reaching movements. It is also suitable for integrating vestibular, auditory, and tactile information.


Eye-centered coordinates
Head-centered coordinates
Body-centered coordinates
World-centered coordinates
Posterior parietal cortex
Eye position
Neck proprioception
Vestibular information
Visual information
Audition

4. Gain modulation

How can visual neurons with receptive fields in EYE coordinates drive pre-motor neurons with receptive fields in HEAD coordinates?

The answer was discovered by Richard Andersen and colleagues, who recorded in parietal cortex and found that visual receptive fields are often modulated by gaze direction (Zipser, Andersen, 1988).

Apparently, 'gain modulation' of visual responses in EYE coordinates facilitates the subsequent read-out in HEAD-coordinates.

Subsequently, the principle of 'gain modulation' was encountered elsewhere, too. For example, auditory responses in HEAD coordinates are gain modulated by neck direction, facilitating subsequent read-out in BODY-coordinates.

EYE coordinates s and HEAD coordinates $s + g$

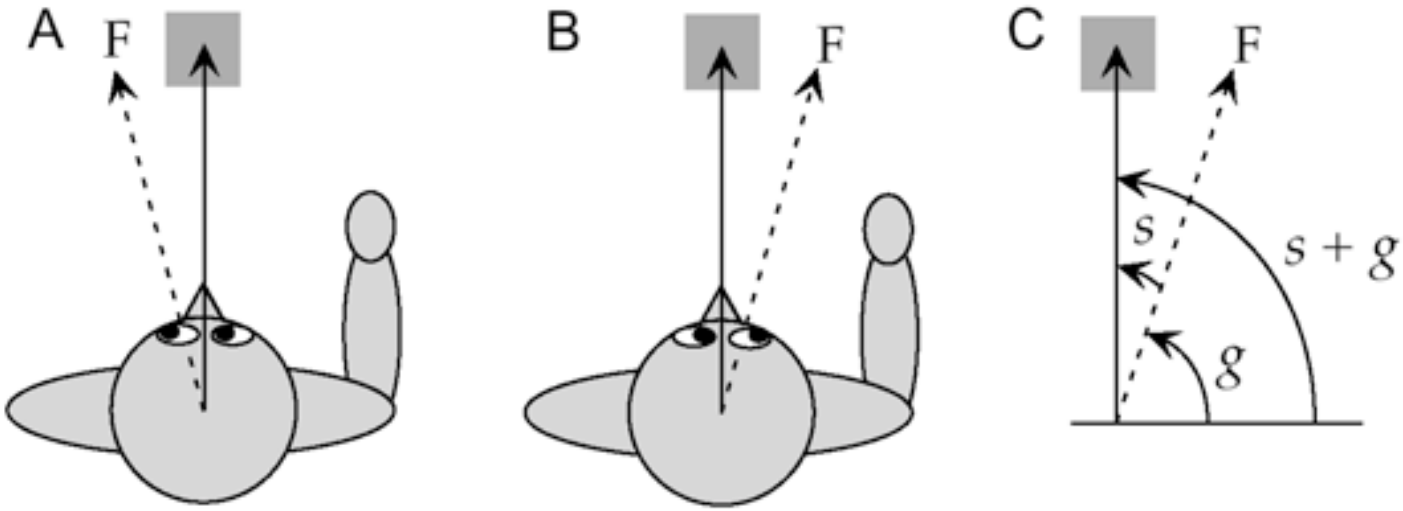


Figure 7.4: Coordinate transformations during a reaching task. A, B) The location of the target (the grey square) relative to the body is the same in A and B, and thus the movements required to reach toward it are identical. However, the image of the object falls on different parts of the retina in A and B due to a shift in the gaze direction produced by an eye rotation that shifts the fixation point F. C) The angles used in the analysis: s is the angle describing the location of the stimulus (the target) in retinal coordinates; g is the gaze direction angle, indicating the orientation of the eyes relative to the body. The direction of the target relative to the body is $s + g$.

Premotor neurons in HEAD coordinates $s + g$

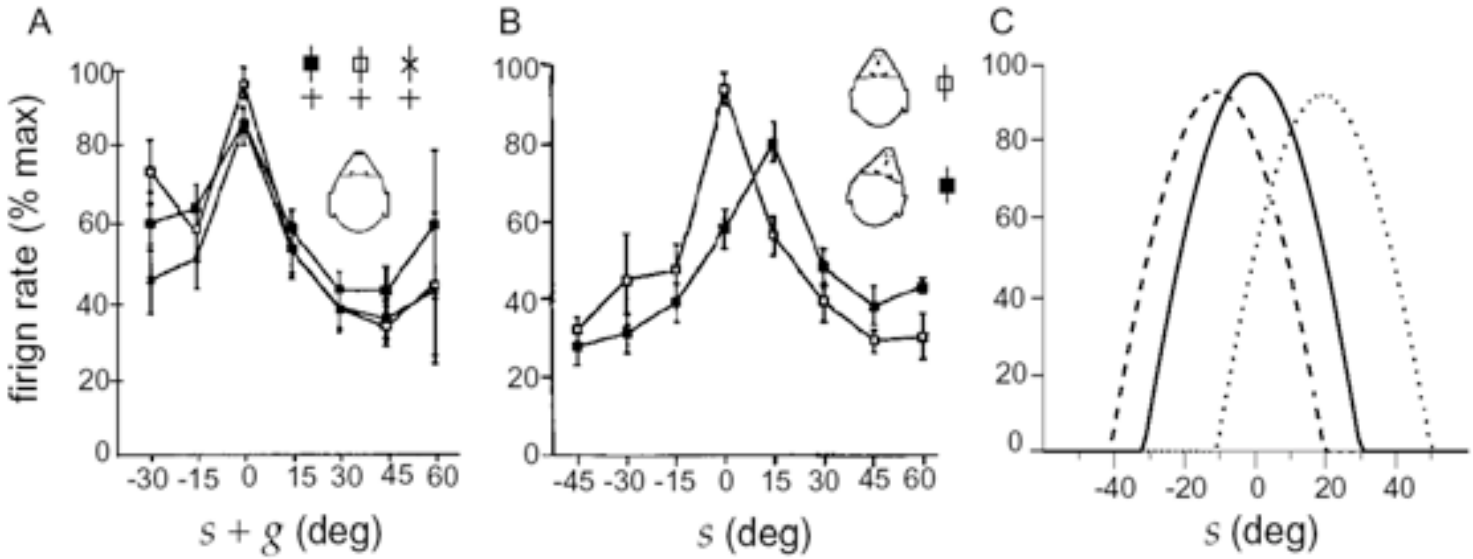
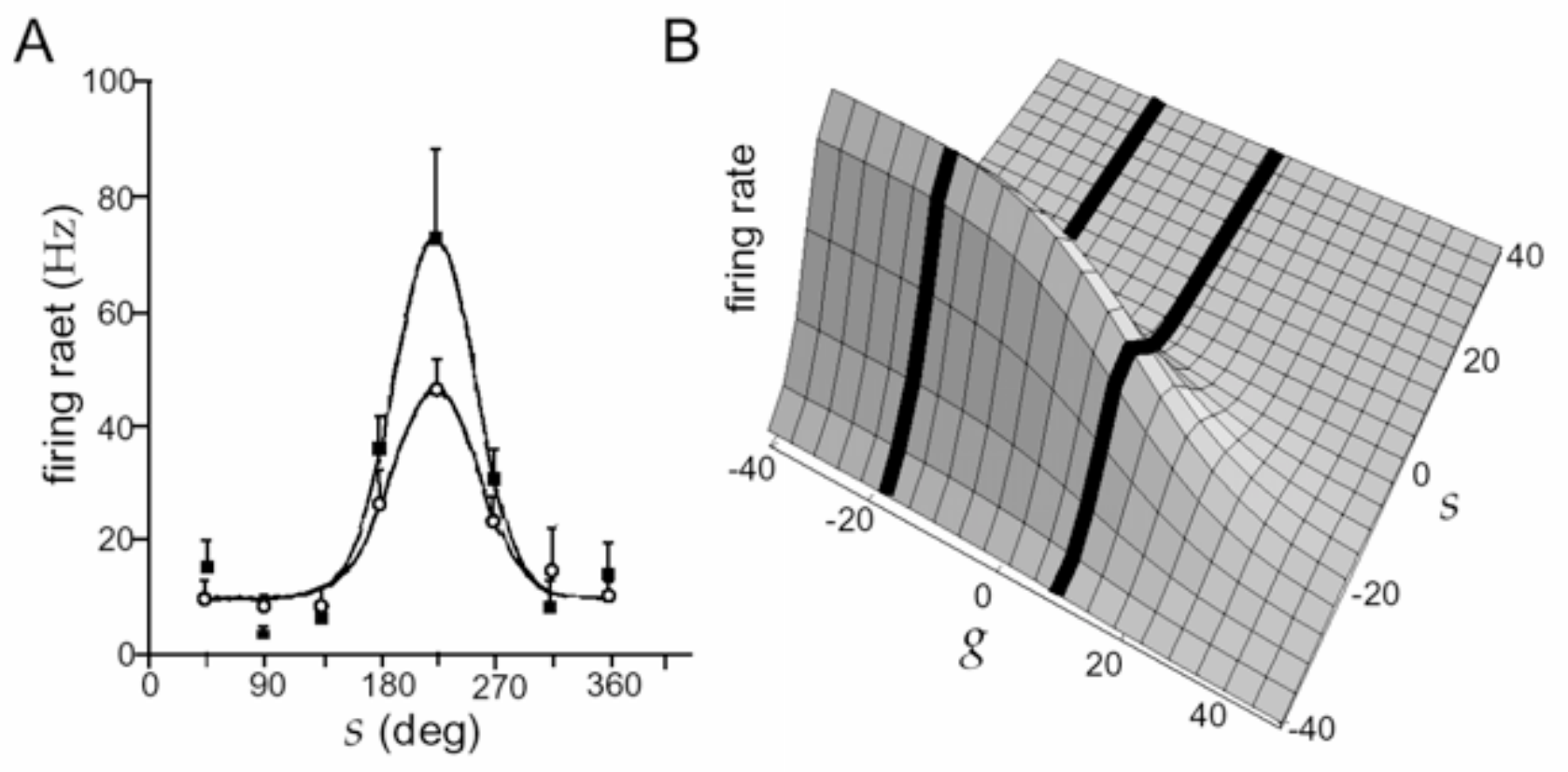


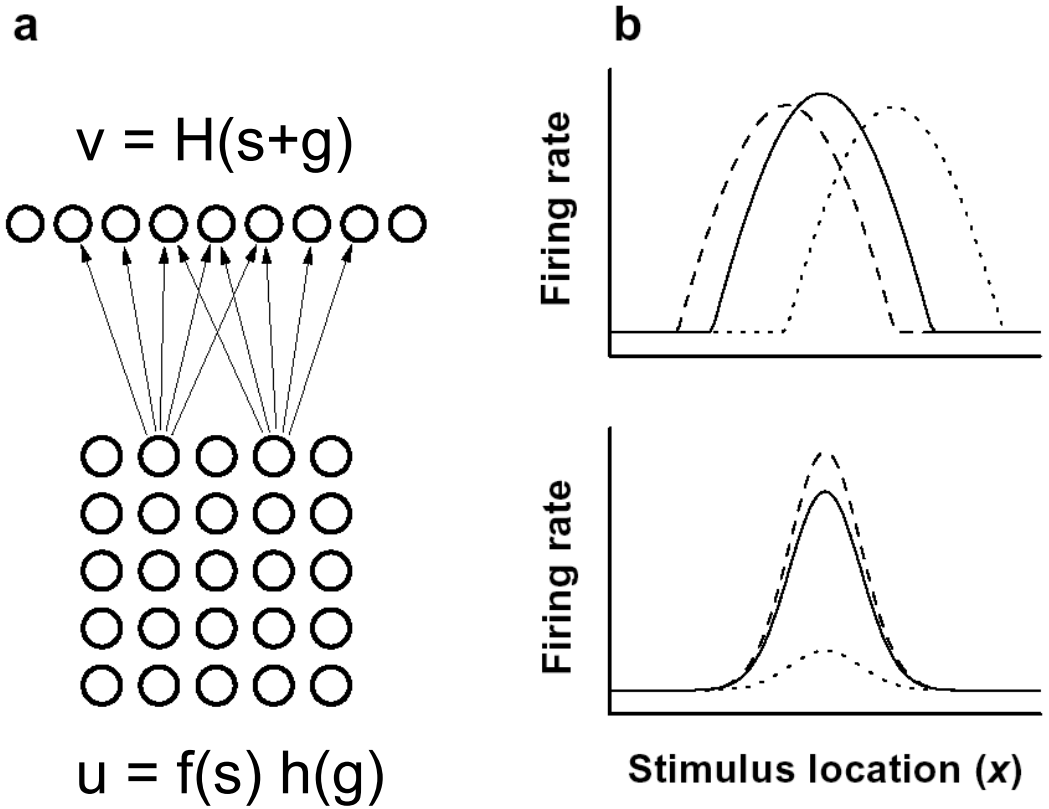
Figure 7.5: Tuning curves of a visually responsive neuron in the premotor cortex of a monkey. Incoming objects approaching at various angles provided the visual stimulation. A) When the monkey fixated on the three points denoted by the cross symbols, the response tuning curve did not shift with the eyes. In this panel, unlike B and C, the horizontal axis refers to the stimulus location in head-based, not retinal, coordinates ($s + g$, not s). B) Turning the monkey's head by 15° produced a 15° shift in the response tuning curve as a function of retinal location, indicating that this neuron encoded the stimulus direction in head-based coordinates. C) Model

EYE-coordinates s, gain-modulation by GAZE direction g



Input activity tuned in eye-coordinates, but gain-modulated by gaze-direction (from Dayan&Abbott).

Desire output in HEAD-coordinates $s + g$



The goal is output activity tuned in HEAD-coordinates (i.e., so that tuning in EYE-coordinates changes with gaze direction).

Visual neurons in parietal cortex

Following Pouget & Sejnowski (1995), we construct an input layer of parietal neurons i, with average response r depending *multiplicatively* on stimulus position s (in retinal coordinates) and on gaze position g (in head coordinates):

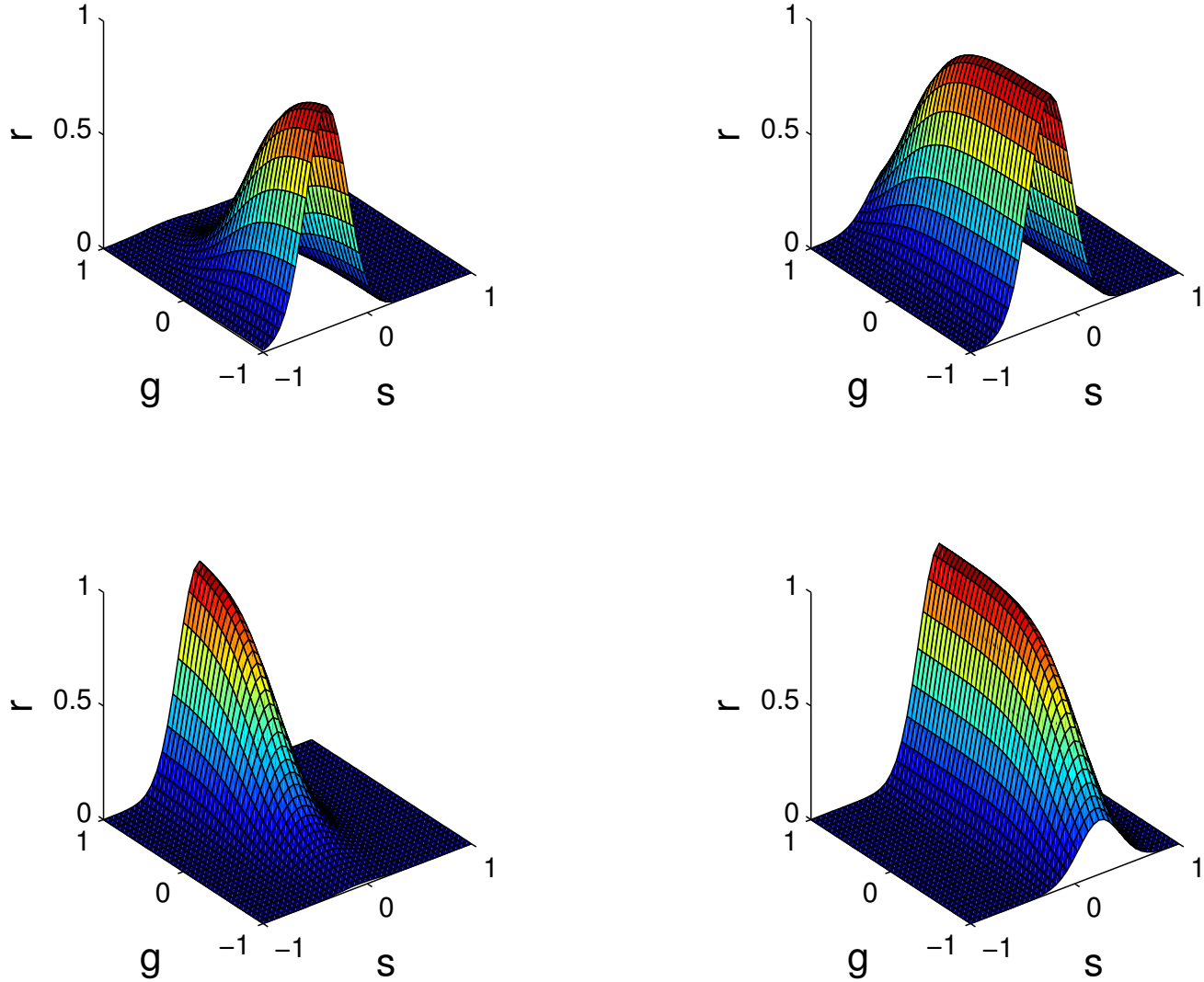
$$r = F(s, g) = f(s - \bar{s}_i)\, h(g - \bar{g}_i)$$

where $\bar{s}_i$ and $\bar{g}_i$ are *preferred stimulus* and *critical gaze* position of neuron i, respectively. Gaussian dependence (with width σ) on stimulus position $(s - \bar{s}_i)$

$$f(s - \bar{s}_i) = r_{max}\ \exp\left[-\frac{(s - \bar{s}_i)^2}{2\sigma^2}\right]$$

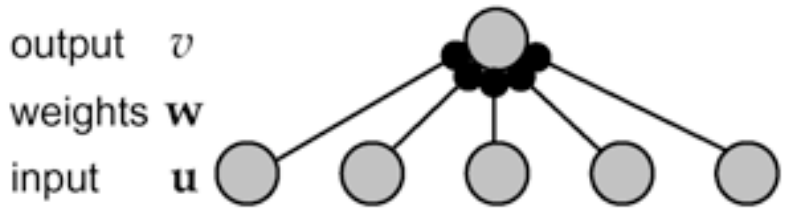
is multiplied with sigmoidal dependence (of sharpness κ) on gaze direction $(g - \bar{g}_i)$

$$h(g - \bar{g}_i) = \frac{\exp[\kappa(g - \bar{g}_i)]}{1 + \exp[\kappa(g - \bar{g}_i)]}$$



Input responses tuned in eye-coordinates, gain-modulated (positively or negatively) by gaze-direction.

Readout neurons in premotor cortex



Activity of readout neuron v in premotor cortex combines input from sensory neurons with a connectivity vector $\mathbf{w} = \{\ldots, w_i, \ldots\}$

$$\tau \frac{dv}{dt} = -v + F\left[\mathbf{w} \cdot \mathbf{u}\right] = -v + F\left[\sum_i w_i\, u_i\right]$$

$$v_{ss} = F\left[\sum_i w_i\, u_i\right] = F\left[\sum_i w_i(\bar{s}_i, \bar{g}_i)\, f(s - \bar{s}_i)\, h(g - \bar{g}_i)\right]$$

Feedforward connectivity (optional)

Can we find a connectivity $w_i(\bar{s}_i, \bar{g}_i)$ such that output v_{ss} becomes a function of $s+g$, $v_{ss} = H(s+g)$?

Assuming full coverage, for any neuron $(\bar{s}_i, \bar{g}_i)$ and any value of g, there is another neuron $(\bar{s}_j = \bar{s}_i - g, \bar{g}_j = \bar{g}_i + g)$ with identical response:

$$f(s-\bar{s}_i)\, h(g-\bar{g}_i) = f(s-\bar{s}_j)\, h(g-\bar{g}_j) = f(s+g-\bar{s}_i)\, h(-\bar{g}_i)$$

If feedforward weights satisfy

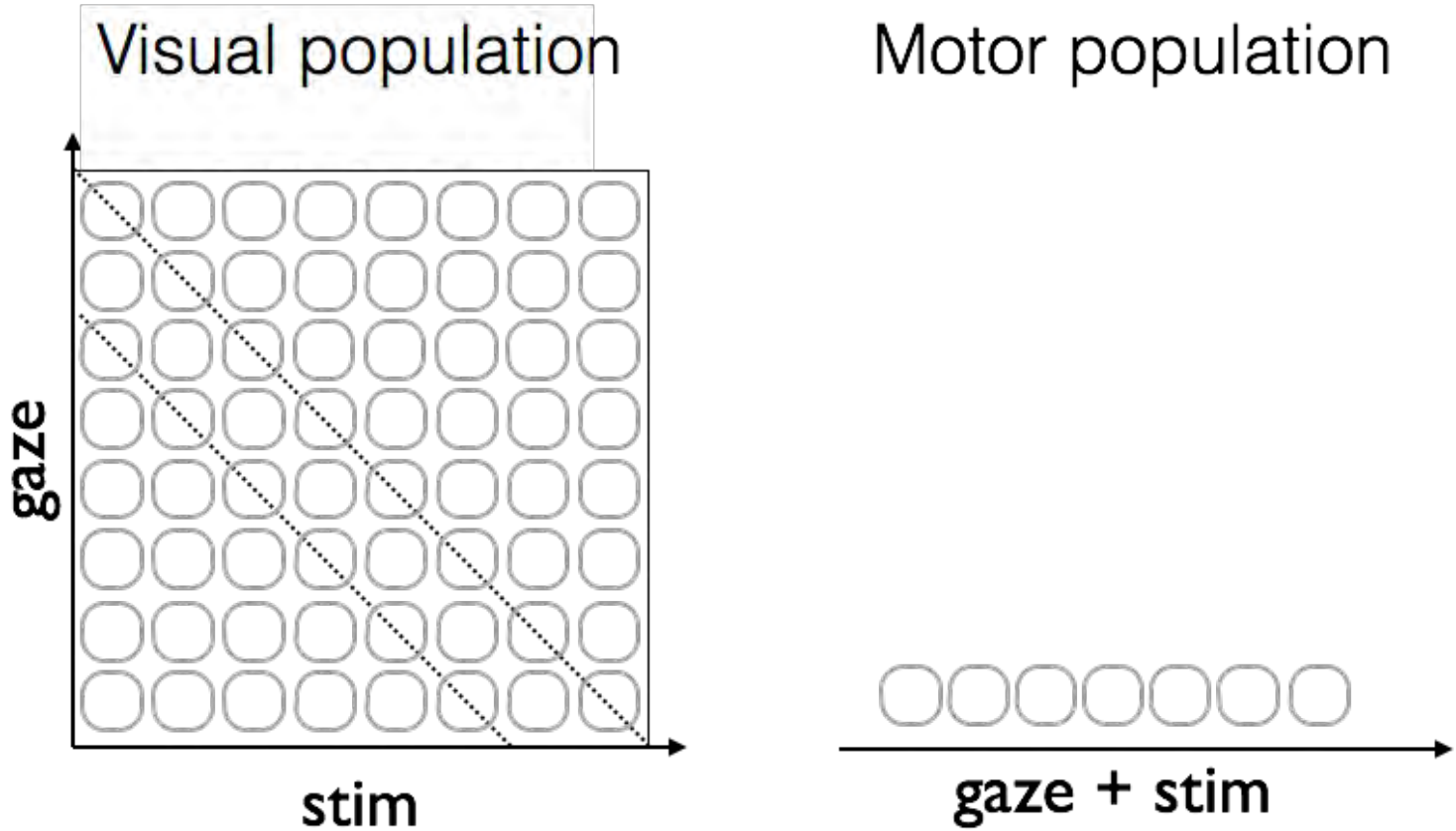
$$w_i(\bar{s}_i, \bar{g}_i) \stackrel{!}{=} w_i(\bar{s}_i + \bar{g}_i)$$

the output response can be written as

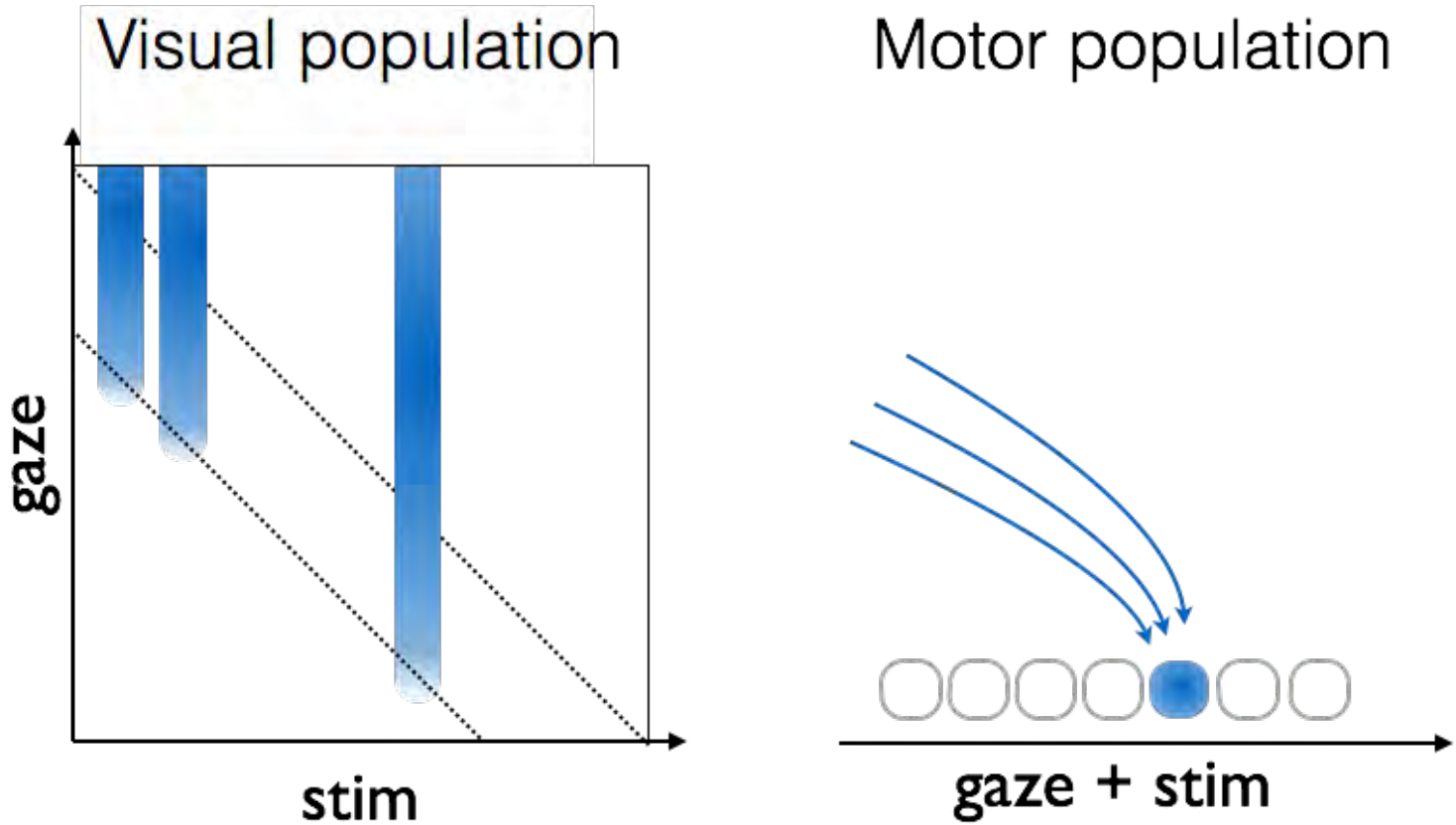
$$v_{ss} \;=\; F\left[\sum_j w_j(\bar{s}_i - g + \bar{g}_i + g)\, f(s+g-\bar{s}_i)\, h(-\bar{g}_i)\right] = H(s+g)$$

and is therefore a function of $s+g$.

Feedforward connectivity (schematic)

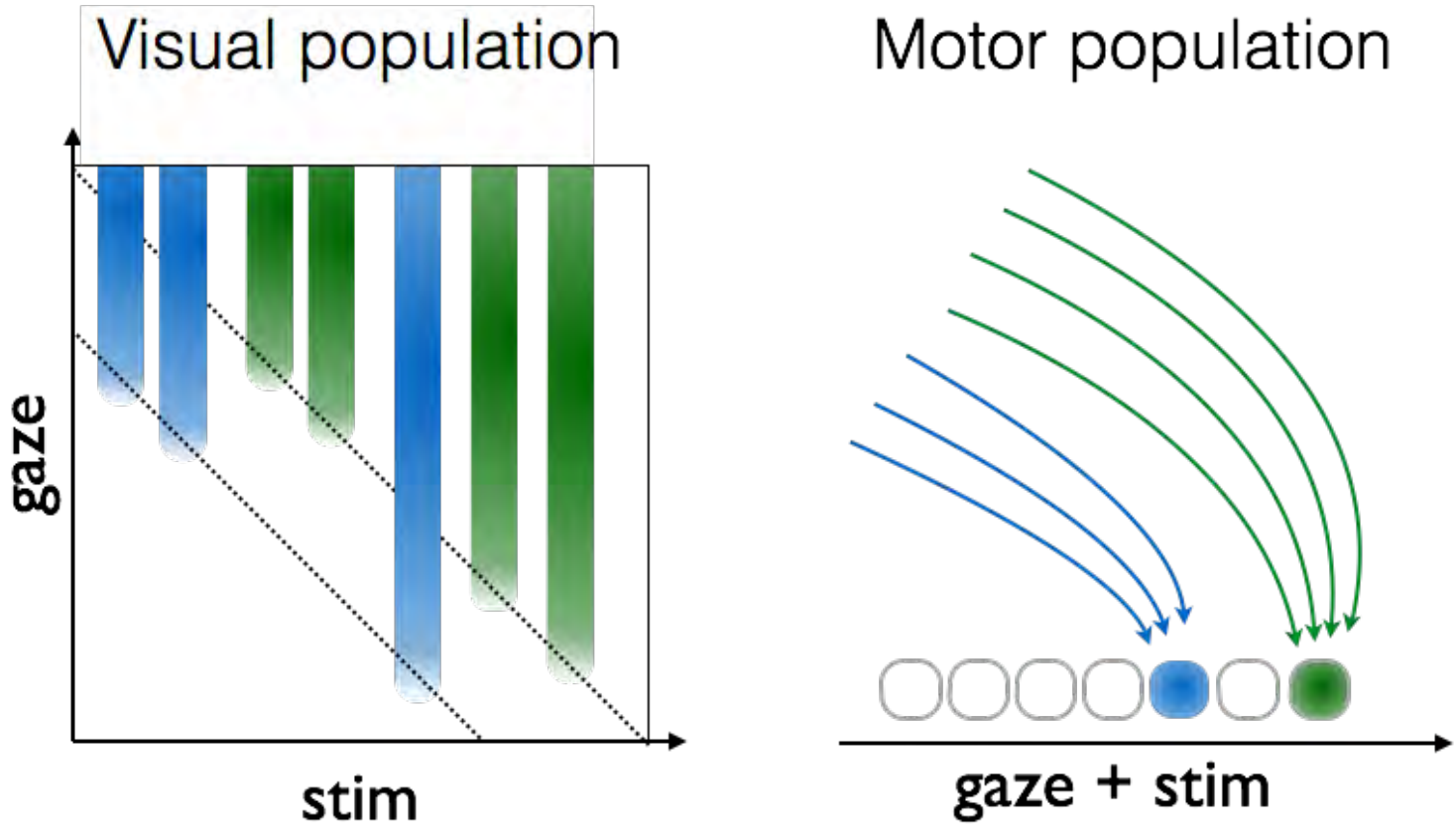


Feedforward connectivity (schematic)



Selected visual units converge to particular motor unit.

Feedforward connectivity (schematic)



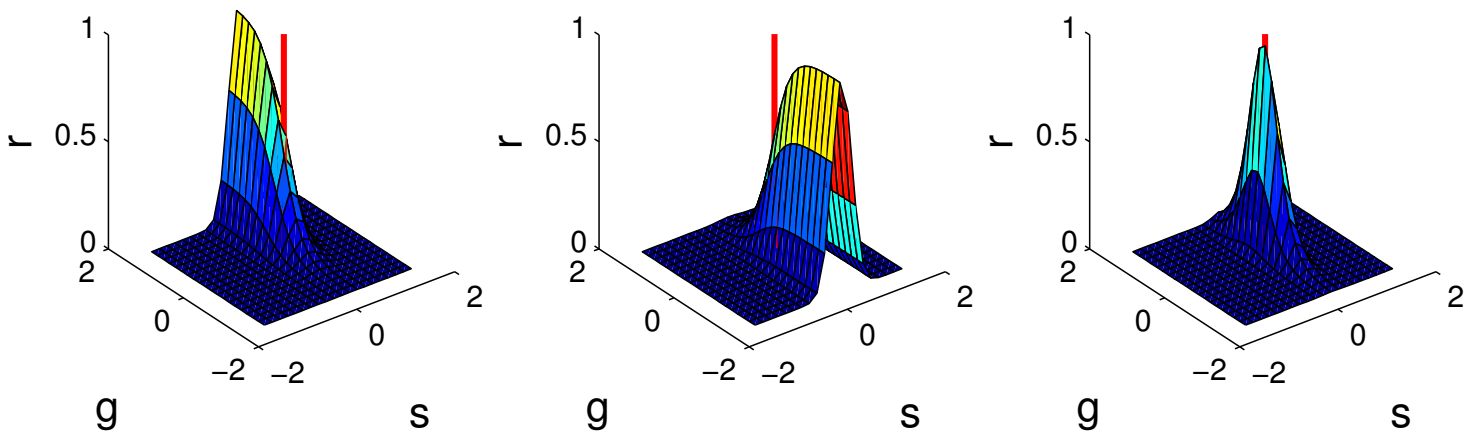
Selected other visual units converge to particular other motor unit.

Example: non-linear combination of visual inputs

We illustrate with an artificial example, combining complementary sensory outputs (with positive and negative sigmoid slopes) in a nonlinear manner:

$$v_{ss} \quad = \quad F\left[\sum_i w_i(\bar{s}_i + \bar{g}_i)\, r(\bar{s}_i, \bar{g}_i)\right]$$

$$r(\bar{s}_i, \bar{g}_i) = f^2(s - \bar{s}_i)\, h(g - \bar{g}_i)\, h(\bar{g}_i - g)$$

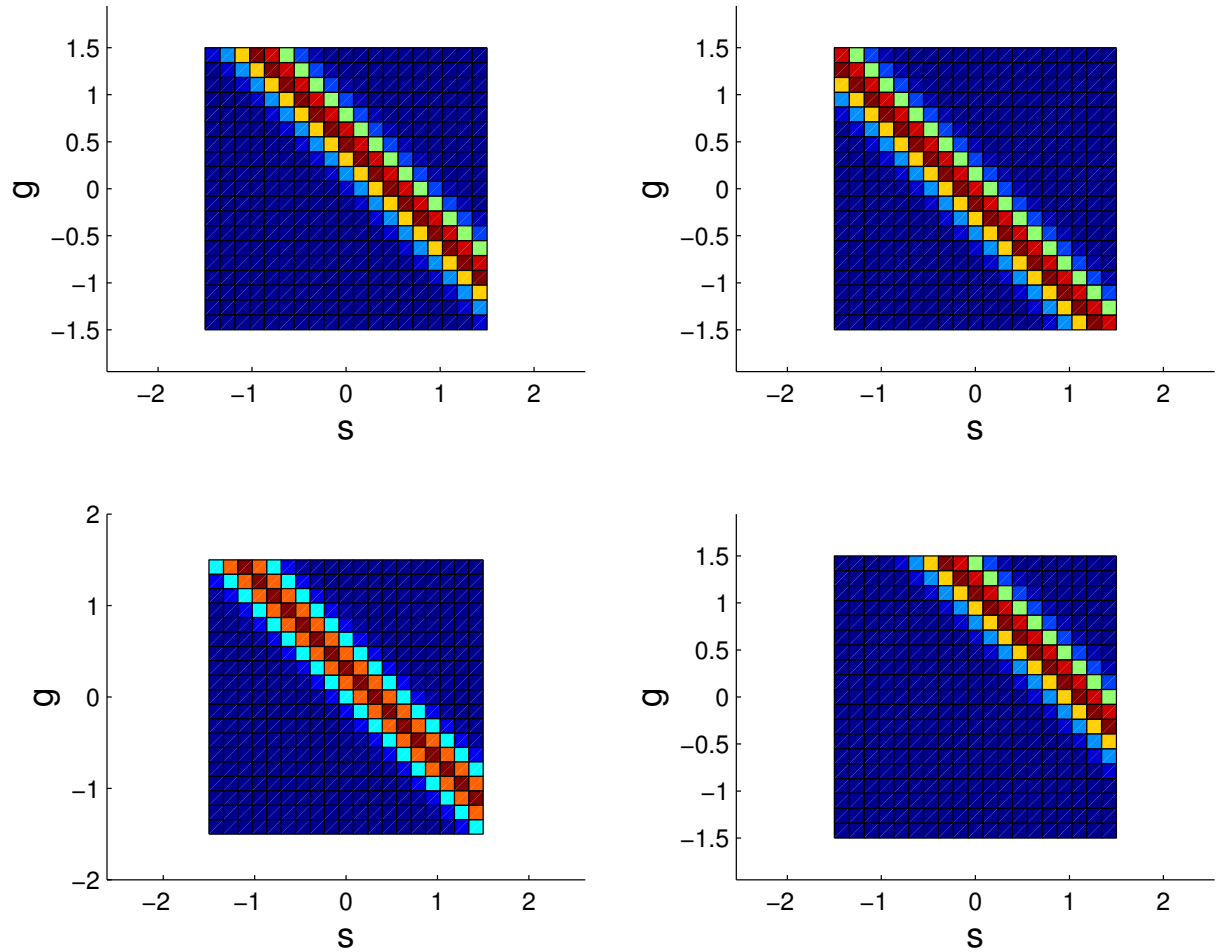


$f(s - \bar{s}_i)\, h(g - \bar{g}_i)$ $f(s - \bar{s}_i)\, h(\bar{g}_i - g)$ $r(\bar{s}_i, \bar{g}_i)$

Feedforward weights (to four example motor units)

Choosing feedforward weights to satisfy

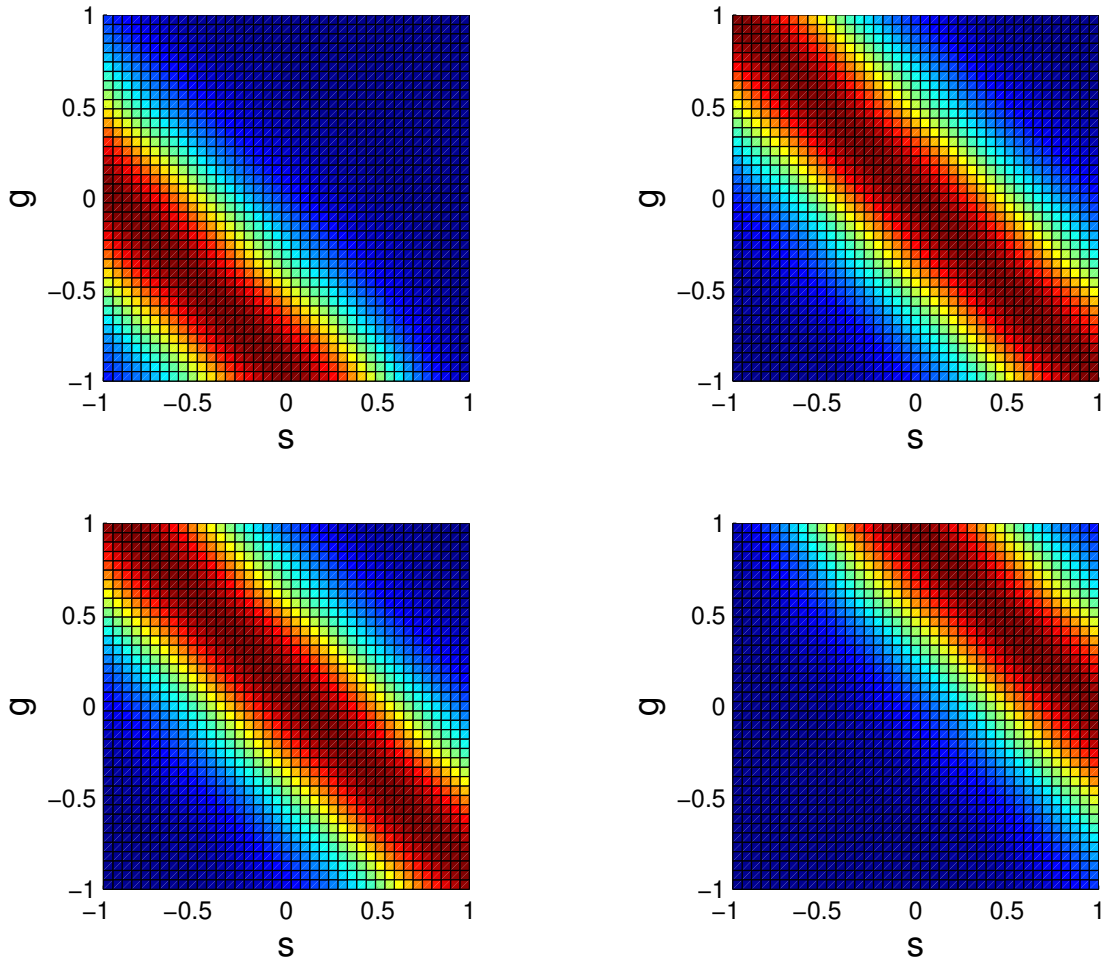
$$w_i(\bar{s}_i, \bar{g}_i) \stackrel{!}{=} w_i(\bar{s}_i + \bar{g}_i)$$



'Heat map' representation of weights $w_i(\bar{s}_i, \bar{g}_i)$ of four motor units.

Resulting selectivity (of four example motor units)

The resulting output neurons are selective for stimulus position in HEAD coordinates, $s + g$:



'Heat map' representations of s–g–tuning of four motor units.

Points to note

- **Visual neurons (in parietal cortex) are tuned in EYE coordinates, plus gain-modulated by GAZE direction.**
- **Motor neurons (in premotor cortex) are tuned in HEAD coordinates.**
- **Feedforward weights (from visual neuron i to motor neuron j) satisfying**

$$w_{ij} = w(\bar{s}_i, \bar{g}_i) \stackrel{!}{=} w(\bar{s}_i + \bar{g}_i)$$

  ensure that motor responses depend on $s + g$ (HEAD coordinates).
- **We have illustrated this by artificially constructing visual neurons selective for particular combinations $(\bar{s}_i, \bar{g}_i)$.**
- **In Exercise 3, you will use supervised learning to build such a feedforward projection.**

5. Generalization

There have been several reports of 'shifting' fields in sensory modalities. For example, visual and auditory receptive fields in parietal cortex may shift with eye position, planned eye position, arm or head position.

In the context of gain modulation, 'shifting' receptive fields present no mystery:

When a target population is driven by a gain-modulated source population, we expect to find neurons reflecting this gain modulation and other neurons (possibly nearby) with shifting receptive fields.

Final reflections

Many population coding schemes are conceivable. Which are chosen by the brain? For what reasons? The choice of a particular scheme may reflect considerations of ...

- ... accuracy: tuning functions and sampling match statistics of external events
- ... connectivity: functional organization to minimize wiring
- ... development: functional organization can be constructed with simple developmental rules
- ... further usefulness: optimize ease and efficiency of "read-out" by subsequent neurons.

Multiplicative gain modulation appears related to the latter issue.

Next:
3. Recurrent networks