Learning

Lecture 8

Examples of learning

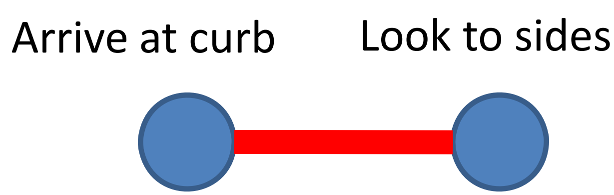
- Learning to play the piano
- Learning to win in chess
- Learning to cross the street safely
- Learning to catch prey
- Learning theoretical neuroscience
- The brain developing its representation of objects

Three types of learning

- Supervised learning: using feedback, learn to produce correct output given new input
- Reinforcement learning: learn to act in a way that maximizes future rewards
- Unsupervised learning: finding structure in data

Hebbian learning

- How neural networks change in response to input: activity-dependent synaptic plasticity
- Neural basis of learning and memory
- “Neurons that fire together, wire together”



- Can be supervised or unsupervised

Presynaptic activity: **u**

Postsynaptic activity: *v*

Weights: **w**

Simple Hebb rule: $$\tau \frac{d\mathbf{w}}{dt} = v\mathbf{u}$$

Averaged Hebb rule: $$\tau \frac{d\mathbf{w}}{dt} = \langle v\mathbf{u} \rangle$$

average over ensemble of inputs

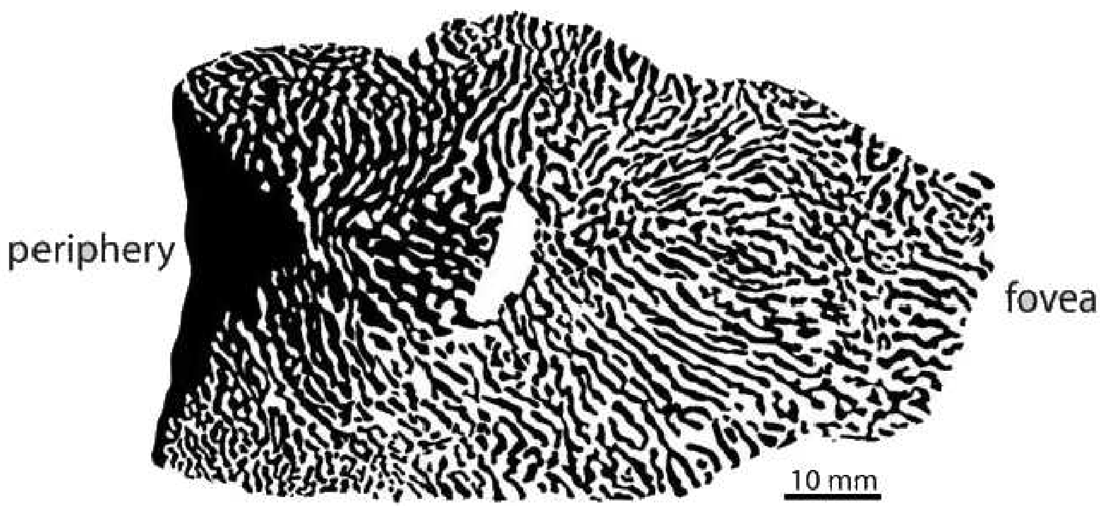
Unstable! Constraint needed to prevent weights from growing indefinitely.

Simple case: $v = \mathbf{w} \cdot \mathbf{u}$

Averaged Hebb rule: $\tau \frac{d\mathbf{w}}{dt} = \langle \mathbf{u}\mathbf{u} \rangle \mathbf{w} = \mathbf{Q} \cdot \mathbf{w}$

Correlation-based rule

With subtractive normalization and constraint
→ can predict ocular dominance columns



Adams et al., 2007

Supervised Hebbian learning

$$\tau \frac{d\mathbf{w}}{dt} = \langle v_s \mathbf{u}_s \rangle$$

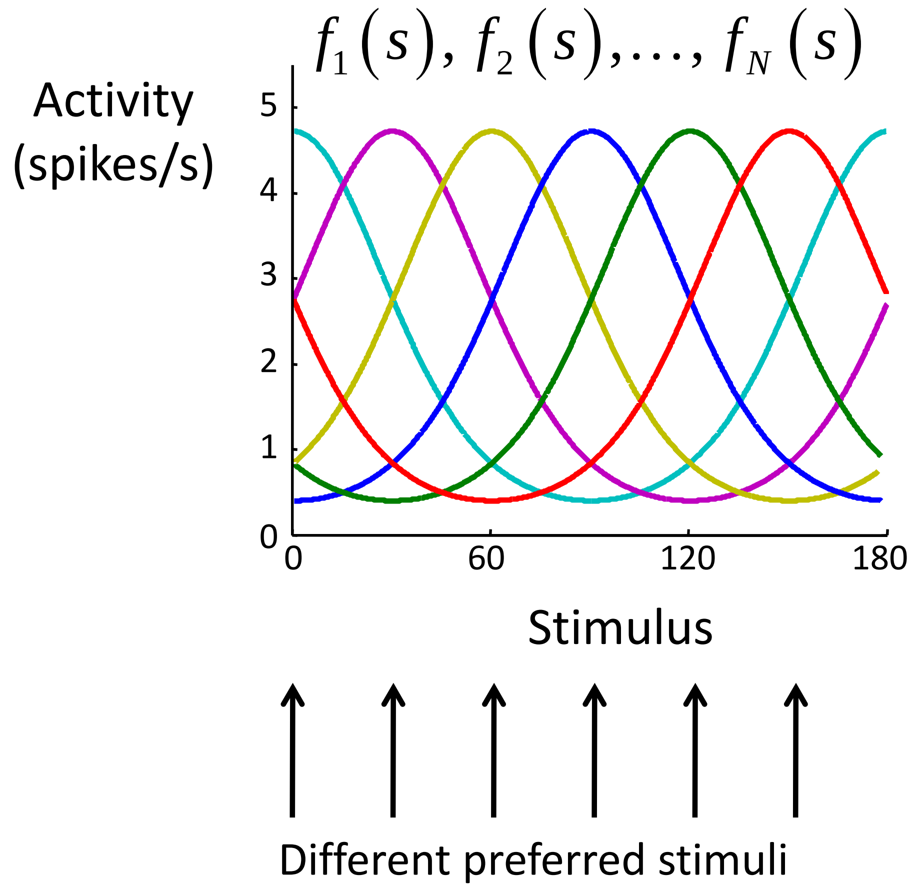
Paired samples

With decay: $\tau \frac{d\mathbf{w}}{dt} = \langle v_s \mathbf{u}_s \rangle - \alpha \mathbf{w}$

Steady state: $\mathbf{w} = \frac{1}{\alpha} \langle v_s \mathbf{u}_s \rangle$

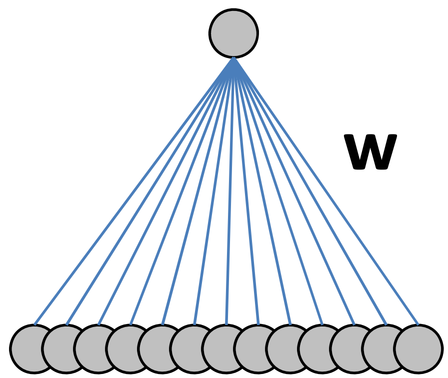
Weights proportional to input-output cross-correlation.

Neural networks for function approximation



Computing a new function

Output neuron *v*, should have tuning *h*(*s*)



Tuning curves **u**=**f**(*s*)

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

Learning rule

$$E = \left\langle \left(h\left(s_s\right) - \mathbf{w} \cdot \mathbf{f}\left(s_s\right) \right)^2 \right\rangle_{\text{training data}}$$

Gradient descent: $\mathbf{w} \rightarrow \mathbf{w} - \varepsilon \nabla_{\mathbf{w}} E$

$$\mathbf{w} \rightarrow \mathbf{w} + \varepsilon \left\langle \left(h\left(s_s\right) - v\left(s_s\right) \right) \mathbf{f}\left(s\right) \right\rangle_{\text{training data}}$$

Stochastic gradient descent (delta rule):

$$\mathbf{w} \rightarrow \mathbf{w} + \varepsilon \left(h\left(s_s\right) - v\left(s_s\right) \right) \mathbf{f}\left(s\right)$$

Representational learning

- How do neurons acquire their response selectivities?
- Natural images are richly structured and highly constrained.
- System learns statistical structure of visual images and builds a model to reproduce structure → *generative model*
- Use this to identify objects in particular images → *recognition model*

Types of representational learning

- Mixture of Gaussians
- Factor analysis
- Principal component analysis
- Independent component analysis
- Sparse coding
- Helmholtz machine

Sparse coding

- Inference on retinal images
- Model images as linear superposition of basis functions
- Sparseness: simple representation, minimize interference between different patterns of input, save energy
- Statistically independent: reduces redundancy

Image model

$$I(x,y) = \sum_i a_i \varphi_i(x,y) + \eta(x,y)$$

image

coefficients
(output activities)

Basis functions

Noise with variance σ^2

Infinite number of solutions for **a**, $\mathbf{\Phi}$, when basis set is *overcomplete* (more output units than input units)

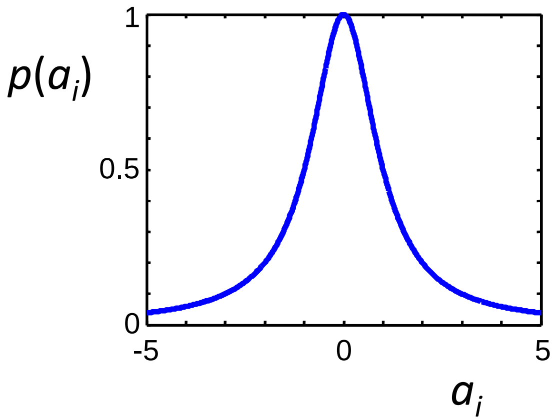
Questions

- When a given image **I** is presented, what should output activity, **a**, be? (recognition model)
- Across all images, what is the best choice of basis functions?

Prior over output activities

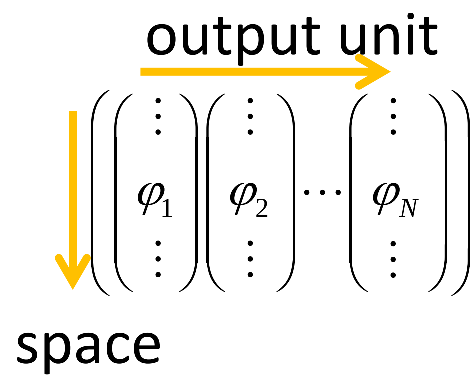
$$p(\mathbf{a}) = \prod_i p(a_i)$$

$$p(a_i) \propto e^{-S(a_i)}$$



Small coefficients are favored: sparseness

$\mathbf{\Phi}$: Matrix of basis functions



Assume $\mathbf{\Phi}$ fixed and known

Posterior over coefficients based on a given image **I**:

$$p(\mathbf{a} \mid \mathbf{I}, \mathbf{\Phi}) \propto p(\mathbf{I} \mid \mathbf{a}, \mathbf{\Phi})\, p(\mathbf{a} \mid \mathbf{\Phi})$$

$$p(\mathbf{I} \mid \mathbf{a}, \mathbf{\Phi}) \propto e^{-\frac{\|\mathbf{I}\text{-}\mathbf{\Phi}\mathbf{a}\|^2}{2\sigma^2}}$$

$$p(\mathbf{a} \mid \mathbf{\Phi}) \propto \prod_i e^{-S(a_i)}$$

Best possible coefficients:

$$\begin{aligned}\hat{\mathbf{a}} &= \underset{\mathbf{a}}{\operatorname{argmax}}\ p(\mathbf{a} \mid \mathbf{I}, \mathbf{\Phi}) \\ &= \underset{\mathbf{a}}{\operatorname{argmax}} \log p(\mathbf{a} \mid \mathbf{I}, \mathbf{\Phi}) \\ &= \underset{\mathbf{a}}{\operatorname{argmin}} \left(\frac{\|\mathbf{I} - \mathbf{\Phi}\mathbf{a}\|^2}{2\sigma^2} + \sum_i S(a_i) \right)\end{aligned}$$

Learning the coefficients through gradient descent:

$$\begin{aligned}\Delta\mathbf{a} &\propto -\nabla_{\mathbf{a}} \left(\frac{\|\mathbf{I} - \mathbf{\Phi}\mathbf{a}\|^2}{2\sigma^2} + \sum_i S(a_i) \right) \\ &= \frac{1}{\sigma^2} \mathbf{\Phi}^T \underbrace{(\mathbf{I} - \mathbf{\Phi}\mathbf{a})}_{\text{residual image}} - S'(\mathbf{a})\end{aligned}$$

Can be implemented in recurrent neural network

Learning the basis functions

- So far: fixed $\mathbf{\Phi}$, learned coefficients **a**
- What about different $\mathbf{\Phi}$?
- Maximize average log likelihood of parameters (minimize KL distance)

Objective function for learning: log likelihood of model $\mathbf{\Phi}$:

$$L = \left\langle \log p\left(\mathbf{I} \mid \mathbf{\Phi}\right) \right\rangle$$

Average over input images

$$p\left(\mathbf{I} \mid \mathbf{\Phi}\right) = \int p\left(\mathbf{I} \mid \mathbf{a}, \mathbf{\Phi}\right) p\left(\mathbf{a} \mid \mathbf{\Phi}\right) d\mathbf{a}$$

Gradient descent on $\mathbf{\Phi}$:

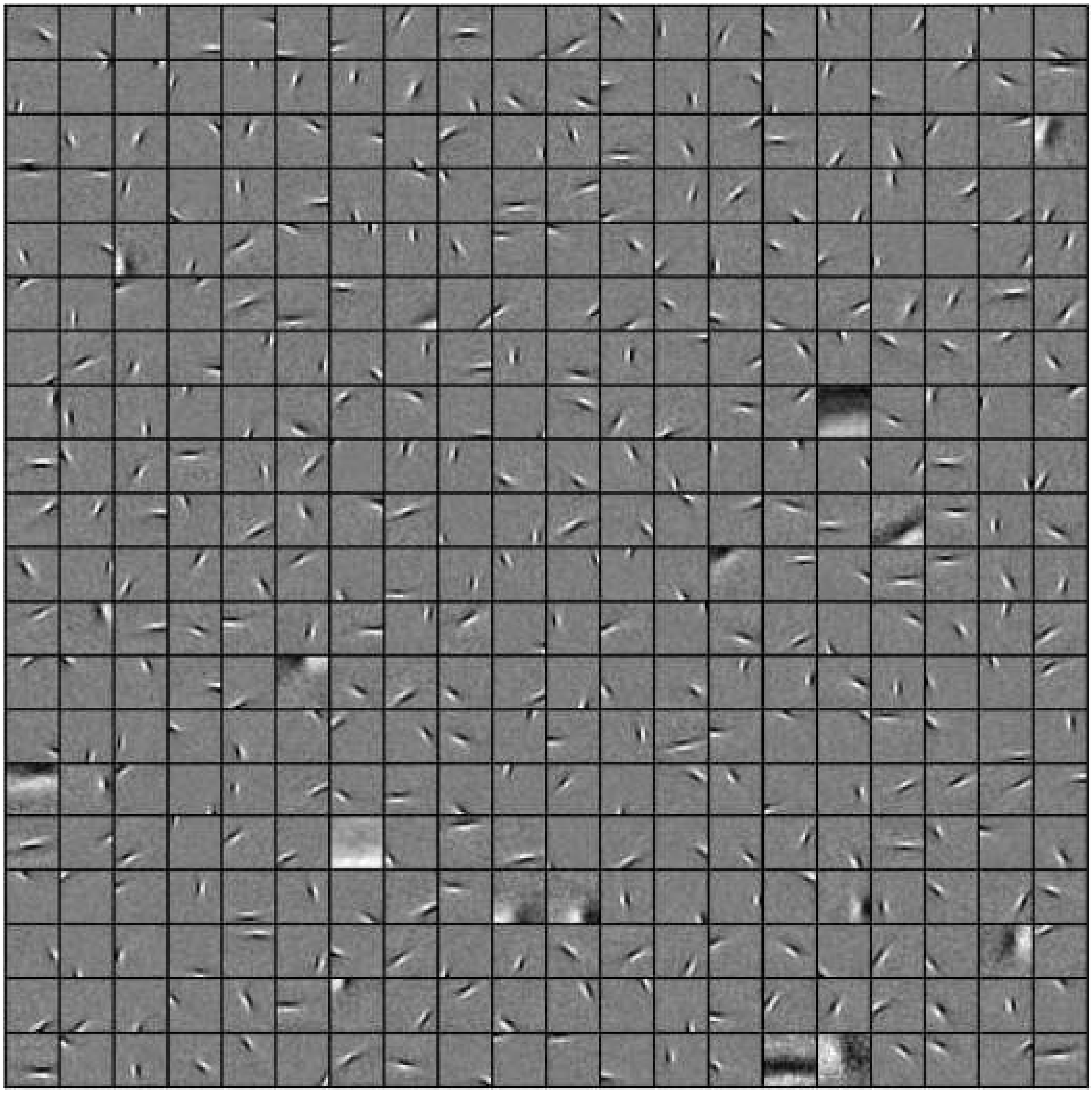
$$\Delta\mathbf{\Phi} \propto \frac{\partial L}{\partial \mathbf{\Phi}} = \frac{1}{\sigma^2} \left\langle \left\langle \left(\mathbf{I} - \mathbf{\Phi}\mathbf{a}\right)\mathbf{a}^{\mathrm{T}} \right\rangle_{p(\mathbf{a}|\mathbf{I},\mathbf{\Phi})} \right\rangle$$

Hebbian learning

Approximate by posterior maximum:

$$\Delta\mathbf{\Phi} \propto \left\langle \left(\mathbf{I} - \mathbf{\Phi}\hat{\mathbf{a}}\right)\hat{\mathbf{a}}^{\mathrm{T}} \right\rangle$$

Constraint needed to prevent growth without bound



Exercise

Reproduce this. Using sparse coding, learn Gabor-like basis functions from any set of photos. Make assumptions where necessary.

Due April 26 by email

Expectation maximization

- Objective function with two parameter sets:

$$F = \left\langle \log p\left(\mathbf{a}, \mathbf{\Phi}; \mathbf{I}\right)\right\rangle$$

(free energy)

- Step 1: fix $\mathbf{\Phi}$, find **a**(**I**) (expectation)
- Step 2: fix **a**, optimize $\mathbf{\Phi}$ (maximization)
- Repeat.
- Converges to local maximum

References

- Olshausen and Field, *Emergence of simple-cell receptive field properties by learning a sparse code for natural images*. Nature 381, 607-9
- Olshausen, *Sparse codes and spikes*. In *Probabilistic models of the brain* (MIT Press, 2002)
- Chapter 10 in Dayan and Abbott, *Theoretical Neuroscience* (MIT Press, 2001)

Lectures so far

- Neural population coding; how to decode
- Role of correlations in information processing
- Perception as Bayesian inference
- Cue combination
- Bayesian models of behavioral data
- Bayesian model comparison
- Neural implementation of Bayesian inference
- Models of perceptual decision-making
- Representational learning; sparse coding
- Thursday: optimal inference in higher-level cognition