MACHINE LEARNING

WEEK 4: INFORMATION THEORY

LECTURE 1: INFORMATION, ENTROPY, KL DISTANCE

Information

The information content of a set of N_m messages is defined to be

$$I_m = \log N_m$$

Consider the case of a binary channel with only ones and zeros. Further constrain all messages to be of length m and to have exactly m_1 ones and m_2 zeros. Naturally $m_1 + m_2 = m$. The number of different possible messages of this distribution of ones and zeros is just

$$N_m = \left(\begin{array}{c} m \\ m_1 \end{array} \right) = \frac{m!}{m_1! m_2!}$$

Thus the information in the ensemble of these messages is just

$$I_m = \log N_m = \log m! - \log m_1! - \log m_2!$$

Entropy = Ave Information

If the $m_i, i = 1, 2$ are so large that $\log m_i \gg 1$, then you can approximate the preceding equation as

$$\log N_m = m \log m - m_1 \log m_1 - m_2 \log m_2$$

So the average information, or entropy, $H = I_m/m$, can be obtained as:

$$H = \log m - \frac{m_1}{m} \log m_1 - \frac{m_2}{m} \log m_2$$

This can be rearranged as

$$-\frac{m_1}{m} \log \frac{m_1}{m} - \frac{m_2}{m} \log \frac{m_2}{m}$$

Finally, interpreting $\frac{m_i}{m}$ as the probability p_i leads to

$$H = -\sum_{i=1}^{2} p_i \log p_i$$

Entropy

$$\mathrm{H}[x] = -\sum_{x} p(x) \log_2 p(x)$$

Important quantity in

- coding theory
- statistical physics
- machine learning

Entropy

Coding theory: x discrete with 8 possible states; how many bits to transmit the state of x?

All states equally likely

$$\mathrm{H}[x] = -8 \times \frac{1}{8} \log_2 \frac{1}{8} = 3 \text{ bits.}$$

Minimum code length

An information rate of $-\sum p_i \log p_i$ is the best we can do. Thus we expect that the average rate (or length) is going to be greater than this—that is, that

$$-\sum p_i \log p_i \leq \sum p_i l_i$$

where l_i is the length of the i^{th} code word. From this it is seen that equality occurs when

$$l_i = -\log p_i$$

and this is in fact the best strategy for picking the lengths of the code words.

Entropy

x	a	b	c	d	e	f	g	h
$p(x)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{1}{64}$
code	0	10	110	1110	111100	111101	111110	111111

$$\begin{aligned} \mathrm{H}[x] &= -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{16}\log_2\frac{1}{16} - \frac{4}{64}\log_2\frac{1}{64} \\ &= 2 \text{ bits} \end{aligned}$$

$$\begin{aligned} \text{average code length} &= \frac{1}{2}\times 1 + \frac{1}{4}\times 2 + \frac{1}{8}\times 3 + \frac{1}{16}\times 4 + 4\times\frac{1}{64}\times 6 \\ &= 2 \text{ bits} \end{aligned}$$

Entropy

In how many ways can N identical objects be allocated M bins?

$$W = \frac{N!}{\prod_i n_i!}$$

$$\mathrm{H} = \frac{1}{N} \ln W \simeq - \lim_{N \to \infty} \sum_i \left(\frac{n_i}{N} \right) \ln \left(\frac{n_i}{N} \right) = - \sum_i p_i \ln p_i$$

Entropy maximized when $\forall i : p_i = \frac{1}{M}$

Differential Entropy

Put bins of width ¢ along the real line

$$\lim_{\Delta \to 0} \left\{ -\sum_i p(x_i) \Delta \ln p(x_i) \right\} = -\int p(x) \ln p(x) \, \mathrm{d}x$$

Differential entropy maximized (for fixed σ^2) when

$$p(x) = \mathcal{N}(x|\mu, \sigma^2)$$

in which case

$$\mathrm{H}[x] = \frac{1}{2} \left\{ 1 + \ln(2\pi\sigma^2) \right\}.$$

Conditional Entropy

$$\mathrm{H}[\mathbf{y}|\mathbf{x}] = -\iint p(\mathbf{y},\mathbf{x}) \ln p(\mathbf{y}|\mathbf{x}) \, \mathrm{d}\mathbf{y} \, \mathrm{d}\mathbf{x}$$

$$\mathrm{H}[\mathbf{x},\mathbf{y}] = \mathrm{H}[\mathbf{y}|\mathbf{x}] + \mathrm{H}[\mathbf{x}]$$

The Kullback-Leibler Divergence

$$\begin{aligned}\mathrm{KL}(p\|q) &= -\int p(\mathbf{x}) \ln q(\mathbf{x}) \, \mathrm{d}\mathbf{x} - \left(-\int p(\mathbf{x}) \ln p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \right) \\ &= -\int p(\mathbf{x}) \ln \left\{ \frac{q(\mathbf{x})}{p(\mathbf{x})} \right\} \mathrm{d}\mathbf{x}\end{aligned}$$

$$\mathrm{KL}(p\|q) \simeq \frac{1}{N} \sum_{n=1}^{N} \{ -\ln q(\mathbf{x}_n|\boldsymbol{\theta}) + \ln p(\mathbf{x}_n) \}$$

$$\mathrm{KL}(p\|q) \geqslant 0 \qquad\qquad \mathrm{KL}(p\|q) \not\equiv \mathrm{KL}(q\|p)$$

Mutual Information

$$\begin{aligned} \mathrm{I}[\mathbf{x},\mathbf{y}] &\equiv \mathrm{KL}(p(\mathbf{x},\mathbf{y})\|p(\mathbf{x})p(\mathbf{y})) \\ &= -\iint p(\mathbf{x},\mathbf{y}) \ln\left(\frac{p(\mathbf{x})p(\mathbf{y})}{p(\mathbf{x},\mathbf{y})}\right) \mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{y} \end{aligned}$$

$$\mathrm{I}[\mathbf{x},\mathbf{y}] = \mathrm{H}[\mathbf{x}] - \mathrm{H}[\mathbf{x}|\mathbf{y}] = \mathrm{H}[\mathbf{y}] - \mathrm{H}[\mathbf{y}|\mathbf{x}]$$

Minimum Description Length

The idea is that of sending a message that describes a theory. One way to do so would be just to send all the data, as in a sense this is a literal description of the theory. But the intuition behind Occam's razor is to favor compact theories. MDL captures this by allowing the message to have the form of a description of the theory plus a description of the data when encoded by the theory. The assumption is that the sender and receiver agree on the semantics of a language for the message and the cost is then the length of the code for the message. Thus the combined length of the message, $L(M, D)$, is a sum of two parts,

$$|L(M, D)| = |L(M)| + |L(D \text{ encoded using } M)|$$

In terms of Bayes' rule, the probability of a model given data can be expressed as

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}$$

Picking the best model can be expressed as maximizing $P(M|D)$, or

$$\max_M P(D|M)P(M)$$

You do not have to consider $P(D)$ here because it is constant across all models. Now maximizing this expression is equivalent to maximizing its logarithm, as the logarithm is monotonic and will not affect the outcome. So

$$\max_M [P(D|M)P(M)] = \max_M [\log P(D|M) + \log P(M)]$$

and this is the same as minimizing its negative:

$$\min_M [-\log P(D|M) - \log P(M)] \tag{2}$$

But now remember the earlier result that for a minimal code that has probability of being sent P, the length of the code is

$$-\log P \tag{3}$$

MDL cost function

Assume the residuals are distributed in the form of a Gaussian with variance α. Then

$$p(D|M) = \left(\frac{1}{2\pi\alpha}\right)^{\frac{N}{2}} e^{-\frac{1}{2\alpha}\sum_{i=1}^{n}(x_i - m_i)^2}$$

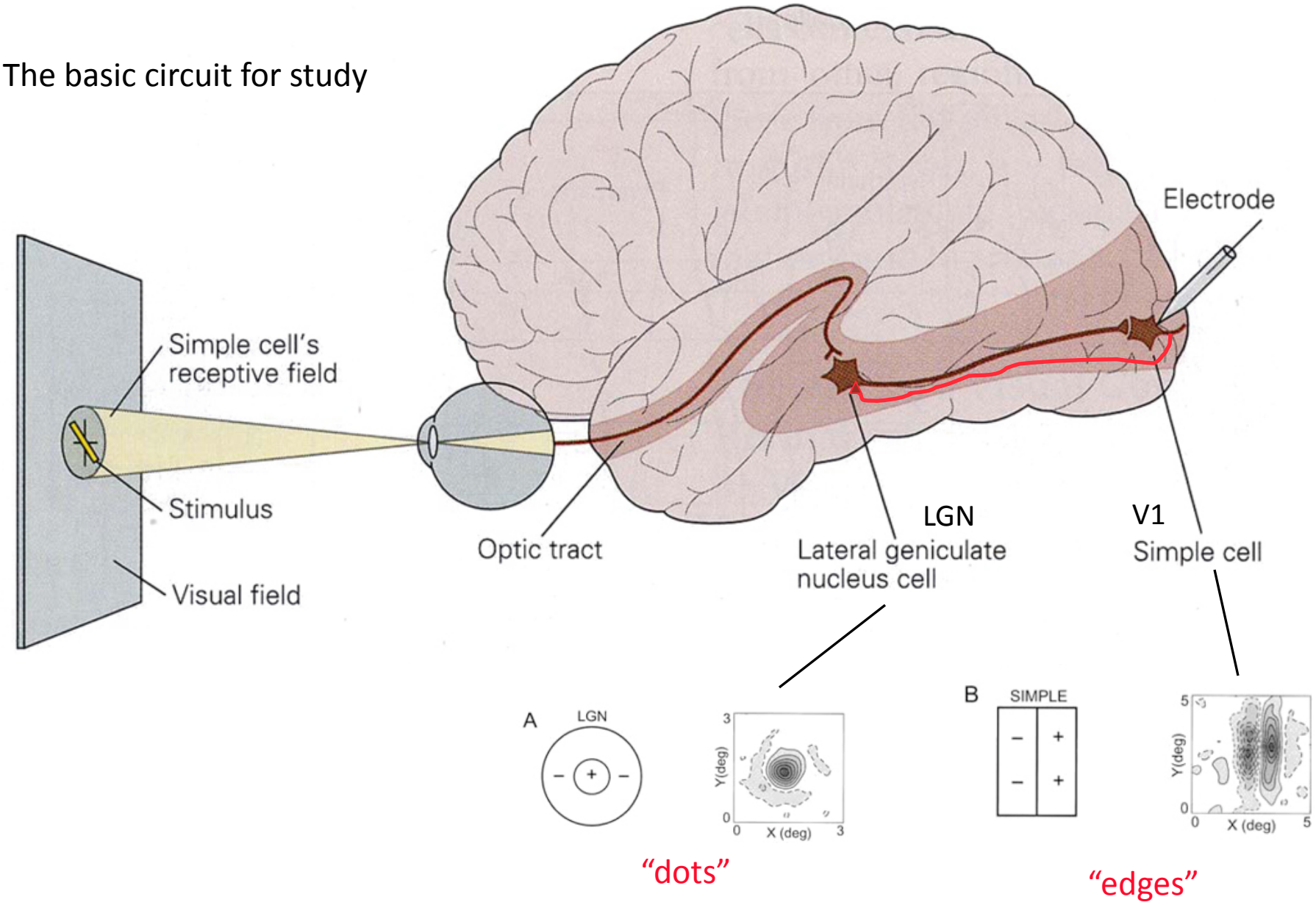
If in turn the model is a neural network with a set of parameters $w_i, i = 1, \ldots, W$, then we can assume that they also are distributed according to a Gaussian, with variance β. Therefore,

$$p(M) = \left(\frac{1}{2\pi\beta}\right)^{\frac{W}{2}} e^{-\frac{1}{2\beta}\sum_i w_i^2}$$

Substituting these two equations into Equation 2,

$$\min_M[-\log P(D|M) - \log P(M)] = \frac{1}{2\alpha}\sum_{i=1}^{n}(x_i - m_i)^2 + \frac{1}{2\beta}\sum_i w_i^2 + \text{ const.}$$

The basic circuit for study



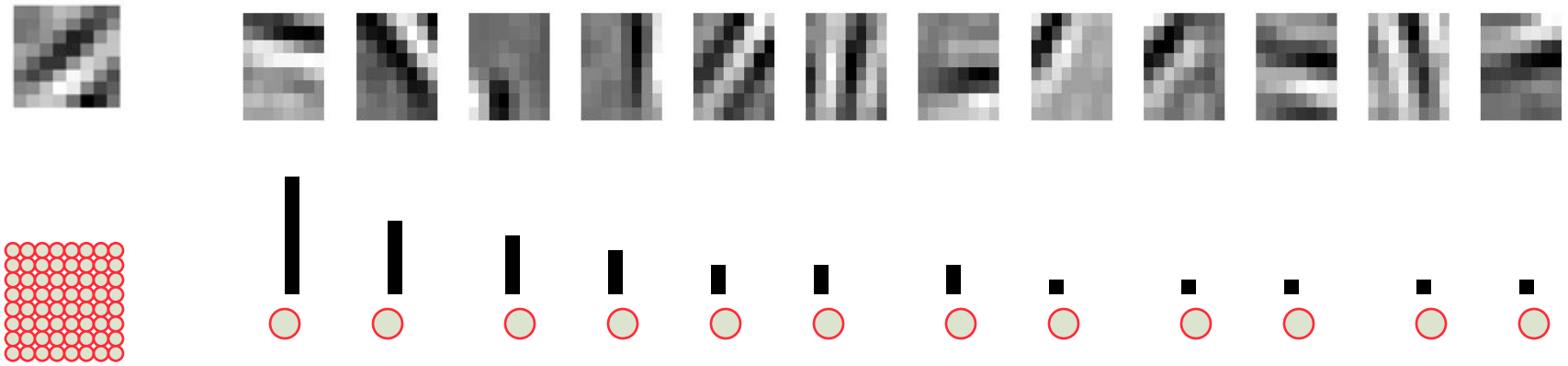
DeAngelis et al. (1995)

Approximating an image patch w basis functions

The outputs
of 64cells
in the LGN …

… can be coded with only twelve V1 cells …

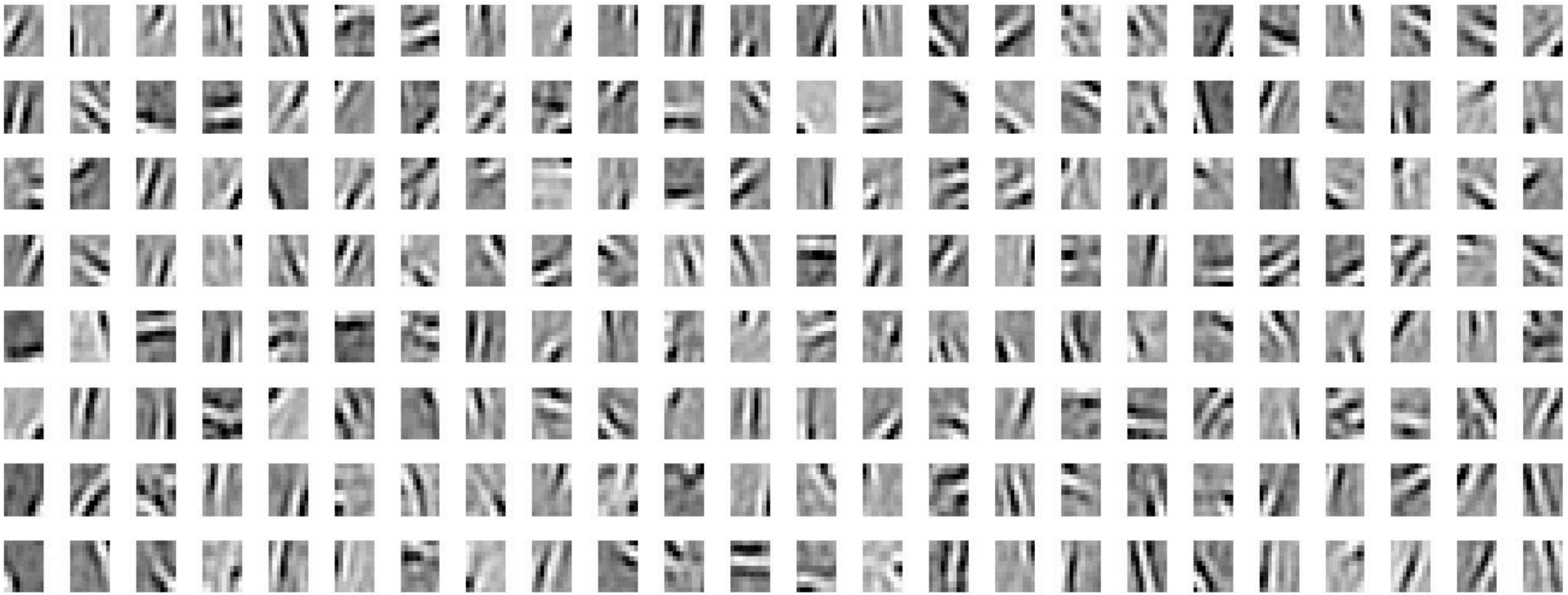
… where each cell has 64 synapses



LGN
Thalamic
nucleus

V1
striate cortex

The neural coding library of learned RFs

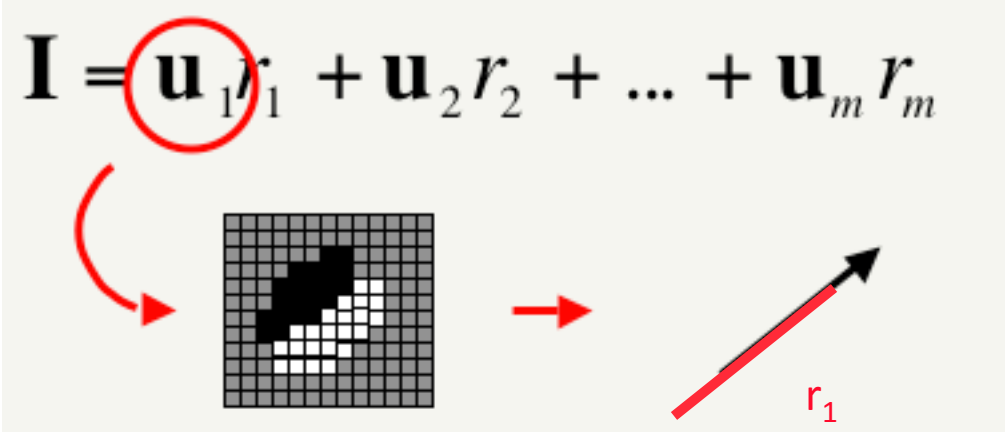


Because there are more than we need - *Overcomplete* (192 vs 64) - the number of cells that need to send spikes at any moment is *Sparse* (12 vs 64).

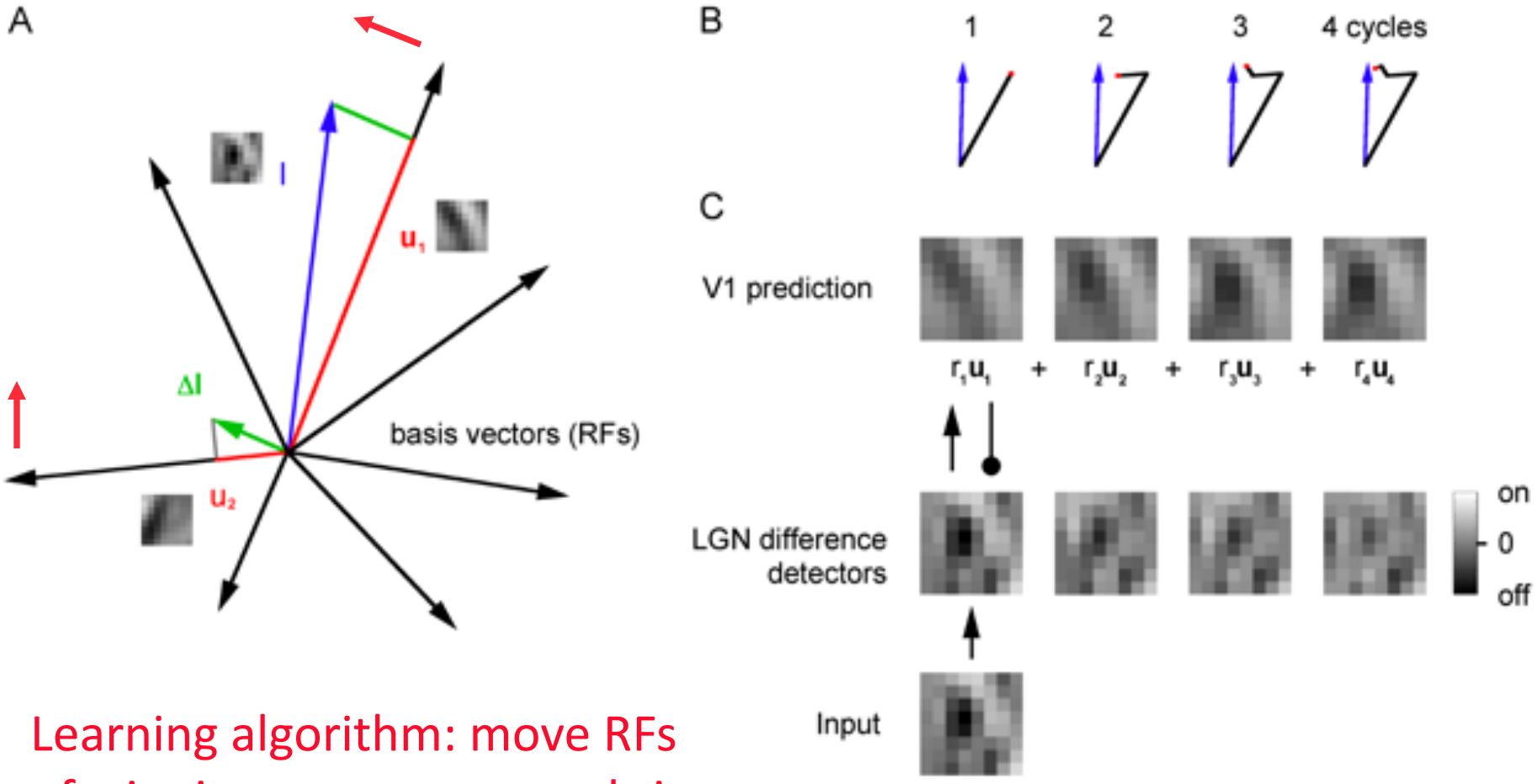
Approximating an image patch w basis functions

LGN V1

RF



The Current Best Algorithm: Matching Pursuit



Learning algorithm: move RFs of winning neurons towards inputs