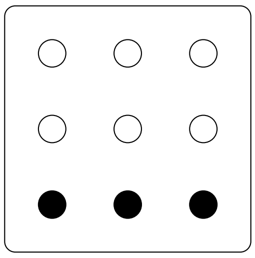
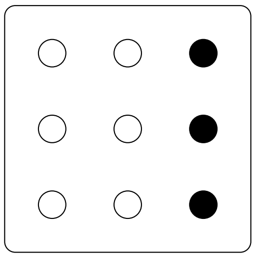
Features

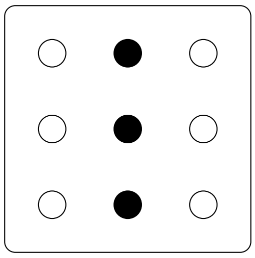
Knowing why V1 receptive fields have the particular stucture they do is likely to tell us something about what it is that the brain does to information in the sensory pathways. One idea is that it is related to feature extraction. To motivate this we will consider a ficticious world of simplified creatures; we are one of these creates and wish to decide how to react to other creatures we encounter. As in the real world, when we encounter a creature we need to decide between what are sometimes called the three Fs: fighting, fleeing and mating. Now imagine that the creatures all have a three by three pattern on their stomachs:



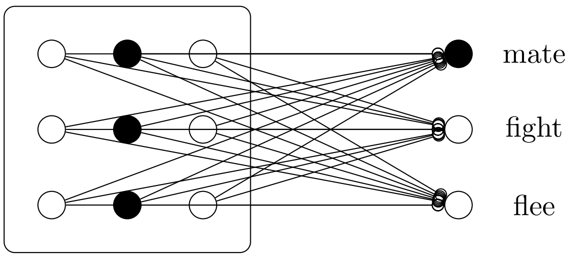
and that creatures to fight have a horizontal strip to the top or the bottom, as above, creatures to flee from, a vertical strip on the left or right, for example



and creatures to mate with, a central line, either horizontal or vertical like:

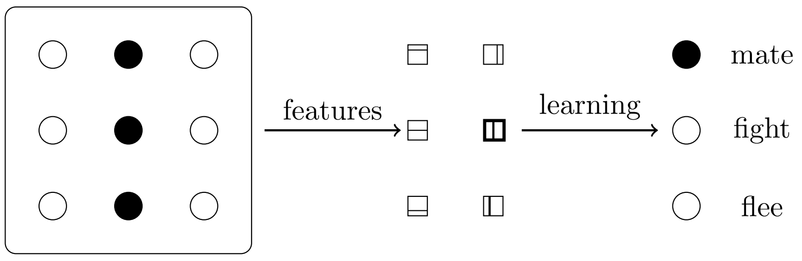


Now, imagine processing this information so as to rapidly decide what to do, the simplest neural network to process the patterns would look like



Clearly this would be very hard to learn; the connection from, say the bottom middle node is on in two of the patterns above despite these patterns corresponding to different types of creature.

A far better strategy would be to first learn features and then learn the association between these features and the creature type. Here the features are clearly the horizontal and vertical bars



In this case the problem has been split in two; first the connections summarized as 'features' above are learned from the data, possibly using the sort of correllation structure learning provided for by STDP, the interpretation of these feature is then learned, this is clearly easier, the connections summarized as 'learning' have a far simpler task, which is good, since it is crucial to learn this sort of salient ecological information quickly.

Feature selection

Here we will consider what properties we would expect features to have. To do this lets imagine that there are neurons that correspond to the features and their activity represents the image. Say the feature neurons each has a receptive field and responds linearly and for simplicity we will leave out the background firing rate: for the sth feature neuron

$$a_s = \sum_{ij} w_{ij}^s I_{ij} \tag{1}$$

and conversely, the output can be represented by

$$I_{ij} = \sum_s a^s W_{ij}^s \tag{2}$$

The slightly confusing thing here is that we are moving between the linear model and the reconstruction. We do this all the time with vectors:

$$\mathbf{v} = v_1\mathbf{i} + v_2\mathbf{j} + v_3\mathbf{k} \tag{3}$$

is the reconstruction where the corresponding project, for example

$$v_1 - \mathbf{v} \cdot \mathbf{i} \tag{4}$$

is like the linear model. However, the situation in this case is more striaght-forward, because the basis vectors are orthonormal

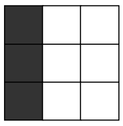
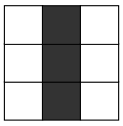
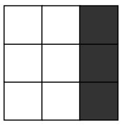
$$\mathbf{i} \cdot \mathbf{j} = \mathbf{j} \cdot \mathbf{k} = \mathbf{k} \cdot \mathbf{i} = 0 \tag{5}$$

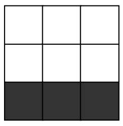
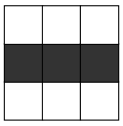
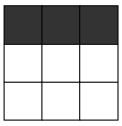
and

$$\mathbf{i} \cdot \mathbf{i} = \mathbf{j} \cdot \mathbf{j} = \mathbf{k} \cdot \mathbf{k} = 1 \tag{6}$$

the same basis vector appears in the reconstruction and the projection: the coeffient v_1 of $\mathbf{i}$ in the reconstruction is the projection of $\mathbf{v}$ onto $\mathbf{i}$. However, in the case of vision the basis elements, the W^s_{ij} and w^s_{ij} are not orthonormal and therefore are not the same, working out the relationship between involves vectorizing the matrix indices i and j, so we won't go into it here, morally one is the inverse transpose of the other. In fact, here we will consider an example where the dimensions are different, where the image patches are 3×3 but there are only six features, so $s = 1 \dots 6$. This means that the reconstructed image may not be equal the original image and will just be an approximation to it.

As an example, lets have

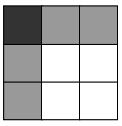
$W^1 =$  $W^2 =$  $W^3 =$ 

$W^4 =$  $W^5 =$  $W^6 =$ 

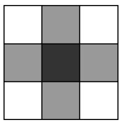
where the almost-black corresponds to one and white to zero, so put another way

$$[W^1_{ij}] = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{7}$$

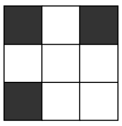
Now consider the example visual input

$I =$ 

This would correspond to $a = (0.5, 0, 0, 0, 0, 0.5)$ or

$I =$ 

corresponds to $a = (0, 0.5, 0, 0, 0.5, 0)$ whereas

$I =$ 

lies outside the six-dimensional subspace spanned by the features.

The question now is what principle to use to select the features. One idea, due to [], is to use sparseness. Going back to the example with creatures with patterned bellies, looking at flee-from animal, for example, three dots are black so among neurons that code for individual dots three would be active, but among neurons coding for vertical or horizontal lines, only one would be active.

Of course this is a very artificial made-up example, the it thought that 'sparseness' is a good way to define features [1, 2]. Very roughly, the fewer neurons needed to reconstruct an image, the more of the image each neuron is coding for; for this to work without having a vast number of neurons covering every possible combination of pixels, the neurons must code for features, piece of image that occur regularly. The assumption, in short, is that all the images are mostly made of the same few building blocks.

The idea is as follows, let I be a image and $\tilde{I}$ an approximation to that image formed using features

$$\tilde{I}_{ij} = \sum_s a_s W^s_{ij} \tag{8}$$

Now W^s could be under-complete, like above, or over-complete, as it may be in the visual system, or the dimensions could be chosen to match. Either way, even if the basis is not under-complete, $\tilde{I}$ is an approximation because the a_s are not just chosen to give an accurate reconstruction, but to do so in a sparse way; the are chosen to minimize

$$E = \sum_{ij} (I_{ij} - \tilde{I}_{ij})^2 + \beta \sum_s f(a_s) \tag{9}$$

This has two terms, the first measures the square error between I and $\tilde{I}$, the second is intended as a measure of sparseness, there are different choices possible, one example would be

$$f(x) = \log_2(1 + x^2) \tag{10}$$

Now, just looking at two dimensions $(1, 0)$ gives

$$\sum_s f(a_s) = \log_2(2) + \log_2(1) = 1 \tag{11}$$

whereas the less sparse $(1/\sqrt{2}, 1/\sqrt{2})$, which as a vector is the same length, gives

$$\sum_s f(a_s) = 2 \log_2(3/2) = 1.17 \tag{12}$$

which is larger. The β here determines the trade-off between accuracy and sparseness, if β is small the square error is more important, if β is big the sparseness is.

The idea now is to take a corpus of image patches and find the best features, the ones that on average give the lowest values of E. The algorithm proceeds in two stages, for each image patch the a_s are chosen to minimise E basically by numerically solving the system of differential equations

$$\frac{\partial E}{\partial a_s} = 0 \tag{13}$$

Next, using the results of this calculation for all the images in the corpus, the features W^s_{ij} are adjusted

$$W^s_{ij} \rightarrow W^s_{ij} - \eta \frac{\partial \langle E \rangle}{\partial W^s_{ij}} \tag{14}$$

where η is a learning rate and $\langle E \rangle$ this average error. This should reduce the average error in the next run through the corpus. This is repeated until the best features are found. The details of how E is minimized over possible choices of the a_s and how to adjust the W^s_{ij} can be found in [1, 2]. The results are shown in Fig. **??** and, ignoring the complication that these are

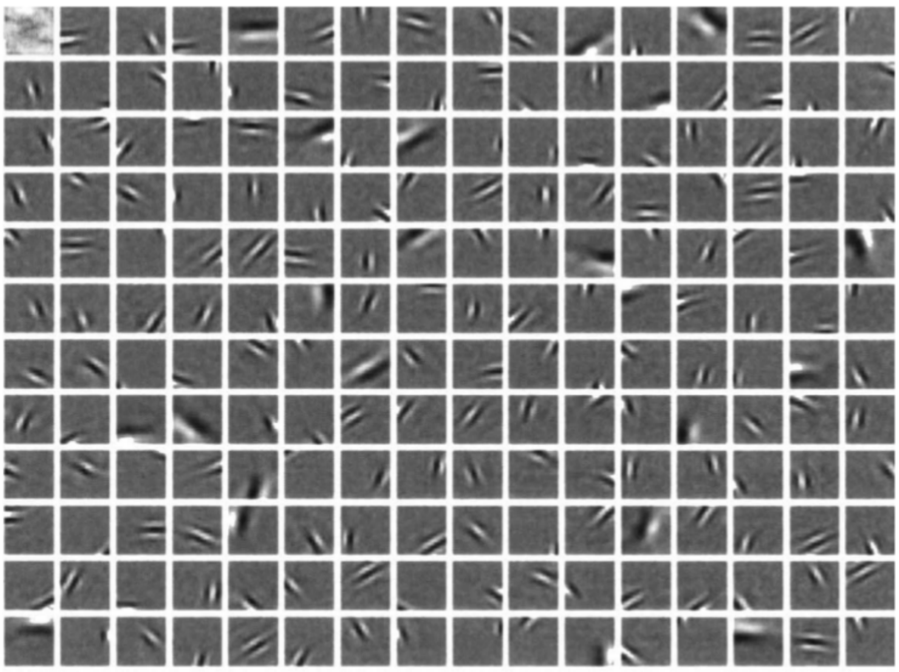


Figure 1: Sparse filters; the optimal features W^s_{ij} using 16×16 image patches cut from a corpus of pictures of the American northwest. [From [1]].

not actually the receptive fields, they do clearly resemble the receptive fields measured from V1.

As described, none of this seems very biological, the sparse filters were discovered using numerical optimization routines. However, there are biologically plausible implementations using Hebbian learning, see for example [3]. There are other approach which parallel sparseness as a way of distinguishing features, for example, Infomax, which examines the informativeness of putative features [].

References

[1] Olshausen BA, Field DJ (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381: 607–609.

[2] Olshausen BA, Field DJ (1997) Sparse coding with an overcomplete basis set: A strategy employed by V1?. Vision Research 37: 3311–3325.

[3] O'Reilly RC, Munakata Y (2000) Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain. MIT Press.

[4] Bell AJ, Sejnowski TJ (1995) An information-maximization approach to blind separation and blind deconvolution. Neural Computation 7: 1129–1159.