3G3 Lab Report: Coding in Visual Cortex

Shanzi(Monica) Ran
sr2021@cam.ac.uk

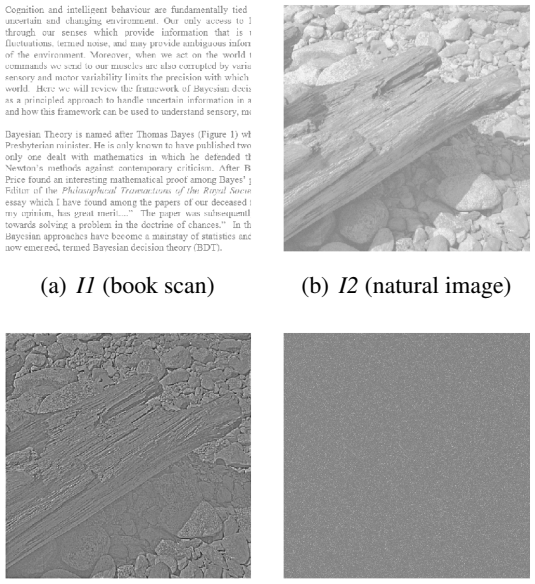
February 8, 2025

1 Introduction

Neural coding is a widely studied framework for sensory information processing, aiming to maximize the metabolic or informational efficiency of sensory representations [1]. This report compares two coding strategies at the primary visual cortex (V1):

- Compact coding: A small population of neurons represents stimuli with frequent activation, reducing redundancy in visual representation [2, 3].
- Sparse coding: Stimuli are represented by a small number of basis functions from a large set, producing a dispersed and sparse representation of sensory input [2, 3].

Through linear algebra simulations, this report explores these coding schemes using both natural and generated images, analyzing how image statistics influence the representation and performance of coding schemes.



(a) *I1* (book scan) (b) *I2* (natural image)

(c) *I2w* (whitened I2) (d) *I3* (random white noise)

Figure 1: *Examples of images from each of the categories*

2 Results and Discussion

2.1 Datasets

A1 Natural images, such as scenes from the natural world [4], exhibit high local pairwise correlation, resulting in a significantly smaller state space compared to randomly generated noise. The greyscale image sets used in this exercise (Fig 1) are categorized as follows: *I1* consists of natural scenes, *I2* contains book scans, and *I3* comprises synthetic white noise images. Additionally, *I2w* represents a whitened version of *I2*, where whitening reduces redundancy by decorrelating adjacent pixels and normalizing variance. **EOA**

2.2 Compact Coding

2.2.1 Dimension Reduction

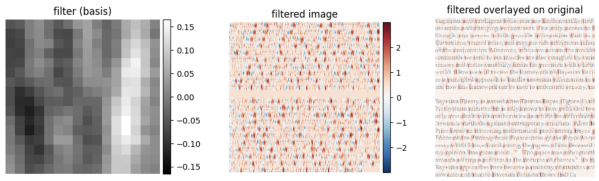
The dimension reducing compact coding algorithm employed in this exercise is Principal Components Analysis (PCA), which produces a hierarchy of eigenvectors of the covariance matrix according to the eigenvalues (variance). **A2** Using this hierarchy, it is observed in the exercise that 102 principal components (PC) capture 90% of the variance in the semi-random book scan *I1* that contains man-made features [3] **EOA**, **A4** 191 components are needed for *I3*, reflecting its random nature and full state space **EOA**, and **A5** 16 components suffice for *I2*, consistent with the high redundancy of natural images **EOA**. Nevertheless, these values all show a reduction from the original 256-dimensional space.

2.2.2 Interpretation of Bases by PCA

Due to statistical differences across image categories, the PCA basis functions for datasets *I1*, *I2*, and *I3* will be analyzed separately in the following section.

PCA for *I1* **A3** For book scan *I1*, each PC captures distinct printing features and setting. For example, the basis function in Fig 2 captures the spacing between words on the page. The density distribution of the

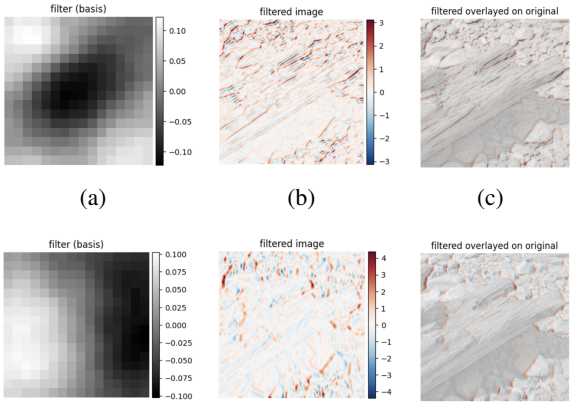
spacing is also depicted in the filter shape. **EOA**



(a) (b) (c)

Figure 2: *An example of basis function for book scan I1: (a)basis filter, (b)filtered image, (c)filtered image overlayed on original image*

PCA for *I2* **A6** For natural image in *I2*, PCA performs a feature extraction transformation similar to principal edge detection. It is also observed that the bases are ordered by increasing complexity, and the whole set of PCs is constructed by rotations of basis coordinates.**EOA**



(a) (b) (c)

(d) (e) (f)

Figure 3: *Two examples of basis functions for natural image I2, where (a)(d) are basis filters, (b)(e) are filtered images, and (c)(f) are filtered image overlayed on original image. It can be observed that (a)(b)(c) depicts a filter detecting edge in diagonal direction, while (d)(e)(f) depicts another filter detecing edge concentrated in the corner orthogonal to the first filter.*

A5 For example, two of the PC bases shown in Fig 3 demonstrate different edge detecting directions, which highlights edges aligned to the filter negative weight areas accordingly. This result aligns with expectation as edges representing first-order derivative change in the observed direction are indeed fundamental components constructing an image. It may also be noted that the filter eliminates redundancy by reducing contrast in areas not aligned to the filer. **EOA**

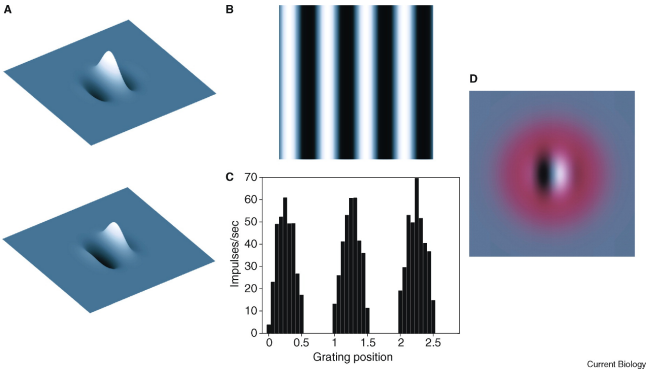
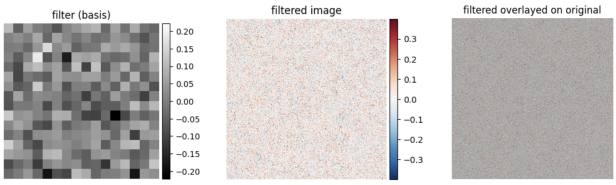


Figure 4: *The simple cell RF can be constructed by weighing a sinusoid using two Gaussian filters. [5]*

A7 Despite rough edges caused by limited subimage resolution, these basis vectors obtained from PCA demonstrate many similarities with receptive fields (RF) of simple cells (Fig 4) as they all demonstrate linear filtering properties with clear distinction between inhibitory regions (with negative filter weight) and excitatory regions (with positive filter weight). However, the inherent linearity of PCA makes accurate representation of non-linear filters of higher-order difficult even with higher complexity. In addition, because the PCs are independent of the phase spectrum of input stimuli [3], the PCA method is also unable to capture local characteristics precisely as a neuron would in V1. **EOA**

PCA for *I3* **A4** Unlike *I1* and *I2*, PCA on the *I3* random white noise dataset shows no clear trend due to the low correlation between local pixels. However, as seen in Fig 5, the filter effectively captures features like immediate intensity contrast. **EOA**



(a) (b) (c)

Figure 5: *An example of basis function for random noise I3: (a)basis filter, (b)filtered image, (c)filtered image overlayed on original image*

Discussion Comparing PCA performance across the three datasets, it is evident that compact coding is most effective for images with high local covariance and

smaller state spaces, such as natural images, and struggles with randomness. Even for natural images like *I2*, PCA fails to capture certain boundary details processed by V1 neurons, highlighting the need to explore alternative sensory coding strategies, such as sparse coding.

2.3 Sparse Coding

2.3.1 Iterative optimisation

In the sparse coding framework, the performance of a code is assessed by the cost function

$$\mathscr{C}(\mathbf{B},\mathbf{A}) = \frac{1}{K}\sum_{k=1}^{K}\left(\sum_{j}\left[S_{kj} - \sum_{i} A_{ki}B_{ij}\right]^2 + \lambda\sum_{i}\log\left(1+\frac{A_{ki}^2}{\sigma^2}\right)\right) \tag{1}$$

which is then minimised with respect to both the basis functions $\mathbf{B} = \{B_{ij}\}$ and the activations $\mathbf{A} = \{A_{ki}\}$. This optimisation is conducted iteratively by first minimising $\frac{\partial\mathscr{C}(\mathbf{A})}{\partial\mathbf{A}}$ for a fixed $\mathbf{B}$, then minimising for $\frac{\partial\mathscr{C}(\mathbf{B})}{\partial\mathbf{B}}$ for the optimum $\mathbf{A}$.

Conveniently writing the expression using the error matrix $\mathbf{E} = \mathbf{S} - \mathbf{AB}$ and trace function, the partial derivatives are found as follows:

$$\frac{\partial\mathscr{C}(\mathbf{A})}{\partial\mathbf{A}} = \frac{2}{K}\left(\mathbf{ABB}^\top - \mathbf{SB}^\top\right) + \frac{2\lambda}{K}\left(\frac{\mathbf{A}}{\sigma^2+\mathbf{A}^2}\right) \tag{2}$$

and **A8**

$$\frac{\partial\mathscr{C}(\mathbf{B})}{\partial\mathbf{B}} = \frac{2}{K}\left(\mathbf{A}^\top\mathbf{AB} - \mathbf{A}^\top\mathbf{S}\right) \tag{3}$$

EOA

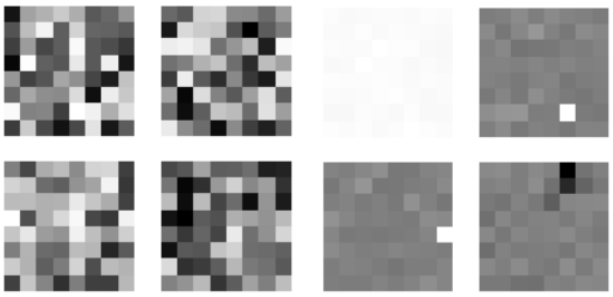
A9 During the implementation of this optimization algorithm (details in Appendix), $\mathbf{B}$ is normalized at each iteration to improve numerical stability, as it undergoes large updates with a learning rate $\eta = 0.2$. Without normalization, the basis function values in $\mathbf{B}$ could easily explode or vanish due to the unpredictable and potentially large gradient $\frac{\partial\mathscr{C}(\mathbf{B})}{\partial\mathbf{B}}$. **EOA**

2.3.2 Interpretation of Bases by Sparse Coding

Since sparse coding is an infinite iterative process, both the basis functions and their evolution are important. This section compares the sparse coding results for the standard datasets *I2(I2w)* and *I3*, representing extreme cases of natural and artificial images respectively.

Sparse coding basis for *I3* **A10** The basis functions obtained from sparse coding differ significantly from those produced by PCA, as shown in Fig 6. Early iterations yield ambiguous receptive fields similar to PCA (Fig 6(a)), but this resolves as the optimization progresses. By iteration 1000 (Fig 6(b)), the receptive fields become sparser and more dispersed, reflecting the goal of sparse coding of promoting overcomplete representations to increase population sparseness and simulate lower action potential requirements for the same neuron activities [2].

Observing the trend at different iteration counts, it can be predicted that as number of iteration reaches infinity, each basis function of *I3* will approach a highly simplified filter with only one feature pixel in a background of uniform intensity, which corresponds to zero overall mean (normalised to fill the greyscale). Therefore, the dataset *I3* could also be reconstructed perfectly by linear superposition and combination of these random filters, given a fully trained set of basis functions. **EOA**



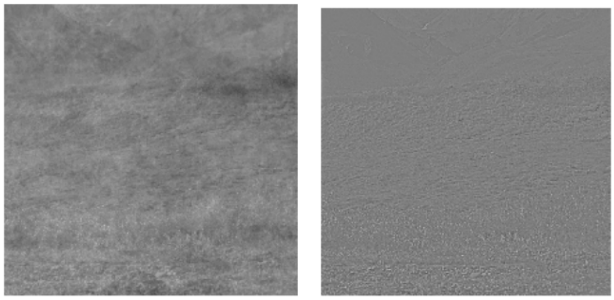
(a) 50 iterations (b) 1000 iterations

Figure 6: *An example of a group of four basis functions of I3 generated by sparse coding, (a)at 50 iterations and (b) at 1000 iterations.*

Sparse coding basis for *I2* Since *I2* dataset consists of natural images which very different variances along different directions in the n_{pix}-dimensional image space, it is pre-processed by whitening to *I2w* as adjacent correlation is removed and variance unified. This whitening process also captures how the retinal ganglion cells decorrelate natural visual stimuli to reduce redundancy and suppress noise for each individual neuron [6].

BA1 In order to implement this batch normalisation method, the PCA-whitening technique is experimented (see details in Appendix). PCA-whitening is a process where the original dataset is multiplied by the matrix of eigenvectors of the covariance matrix, which is then normalised using the square root of its singular values [6]. The covariance matrix is analysed using Singular Value Decomposition (SVD) for the PCA process. Because the products of this pre-processing step is to be used in training, the full 256×256 pixels matrix is used for SVD, and a randomly sampled set of subimages is used then to calculate the amount of required scaling for efficiency considerations.

However, as shown in Fig 7, the pre-whitening result from PCA-whitening is not optimum and appears to be not sphericalised enough when comparing to its contemporary in *I2w*. This is conjured to be an effect of the PCA-whitening strategy, which can be modified and likely improved by more complicated algorithms like ZCA. **EOA**



(a) (b)

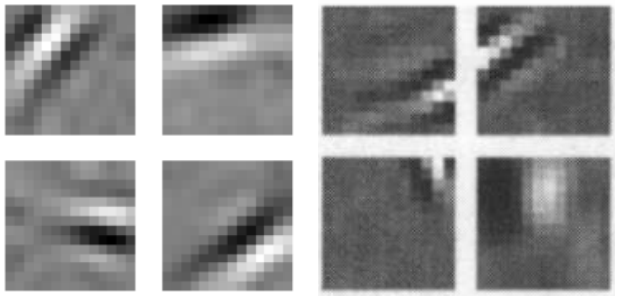
Figure 7: *An example of pre-whitened image from I2: (a)experimental results from this lab exercise and (b)the corresponding image in dataset I2w*

A11 Although both demonstrate directional discrimination to some extent, it is obvious that the basis functions learned by sparse coding in Fig 8(a) demonstrates better localisation and clear boundaries for inhibitory and excitatory regions. This also aligns well with the localised and oriented structure of V1 simple cell RF linear filters. **EOA**

2.3.3 Measure of Sparseness

BA2 When describing a variable with the term "sparse", two scenarios might be indicated:

- Population sparseness: The average number of neurons active at any time,
- Lifetime sparseness (kurtosis): Each neuron has a high kurtosis in its lifetime activation responses. [1]



(a) (b)

Figure 8: *An example of a group of four basis functions of I2w generated by sparse coding. Notice the similarity between (a)experimental results from this lab exercise (3500 iterations) and (b) basis functions learned by sparse coding algorithm in the original paper by Olshausen & Field [7].*

In this lab exercise, since temporal data over a certain time range is unattainable, it is assumed that the sparseness to be measured describes population sparseness. One possible way of measuring this quantity is the "effective PCA" count as discussed in Sec 2.2.1. The kurtosis of response of the entire population to a single visual stimulus could may also be an important indicator, which can be calculated as:

$$K_P = \left\{ \frac{1}{N} \sum_{j=1}^{N} \left(\frac{r_j - \bar{r}}{\sigma_r} \right)^4 \right\} - 3[1] \qquad (4)$$

EOA

3 Conclusion

In this lab exercise, the performance and characteristics of compact coding and sparse coding are examined as two common strategies for sensory coding. It is observed that compact coding through PCA generates fewer basis functions compared to the overcomplete basis set obtained from sparse coding. However, PCA also falls short in accurately representing the complexity of neuron activity in V1. Sparse coding, on the other hand, provides a more biologically realistic and accurate model of V1 neuron responses by producing sparse, distributed representations, and is hence more

suitable for noisy or random stimuli.

These results align with the widely accepted view that sparse coding is a more powerful and efficient strategy for sensory coding, particularly for representing natural images. Nonetheless, PCA remains useful for identifying key features in data through principal components and offers an effective preprocessing method, such as whitening, which reduces redundancy and simplifies subsequent analyses. Each with their own strengths and weaknesses, compact and sparse coding are important complimentary tools fostering better understanding of the myths of sensory coding.

References

[1] B. Willmore and D. Tolhurst, "Characterizing the sparseness of neural codes," *Network: Computation in Neural Systems*, vol. 12, pp. 255–270, 01 2001.

[2] B. Olshuasen and D. Field, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, pp. 481–487, 08 2004.

[3] D. J. Field, "What is the goal of sensory coding?," *Neural Computation*, vol. 6, pp. 559–601, 07 1994.

[4] S.-C. Zhu and Y. N. Wu, "Statistics of natural images," *Springer eBooks*, pp. 19–35, 01 2023.

[5] P. Lennie, "Receptive fields," *Current Biology*, vol. 13, pp. R216–R219, 2003.

[6] E. M. Dodds, J. A. Livezey, and M. R. DeWeese, "Spatial whitening in the retina may be necessary for v1 to learn a sparse representation of natural scenes," *bioRxiv (Cold Spring Harbor Laboratory)*, 09 2019.

[7] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision Research*, vol. 37, pp. 3311–3325, 12 1997.

Appendix: Code implementations

```
def cost(A, B, S, lambd, sigma):
"""
cost␣function

Parameters
----------
A:␣activations␣with␣shape␣(n_sub,␣n_bas)
B:␣bases␣functions␣(n_bas,␣n_pix)
S:␣matrix␣of␣image␣patches␣(n_sub,␣n_pix)
lambd:␣weighs␣the␣importance␣of␣
    reconstruction␣error␣and␣sparsity
sigma:␣activation␣scale␣hyper-parameter␣
    in␣the␣cost␣function

Returns
-------
c:␣average␣cost␣for␣the␣image␣batch␣(thus
    ␣c␣=␣err␣+␣sparsity;␣see␣next␣two␣
    lines)
err:␣reconstruction␣error␣per␣batch
sparsity:␣sparsity␣penalty␣per␣batch,␣
    including␣the␣lambd␣factor
"""
n_sub = np.shape(A)[0]

err = (np.trace(S @ S.T) - 2 * np.trace(S
    @ B.T @ A.T) + np.trace(A @ B @ B.T @
    A.T)) / n_sub

sparsity = lambd * np.sum(np.log(1 + (A
    ** 2) / (sigma ** 2))) / n_sub

c = err + sparsity

return c, err, sparsity
```

```
def dcost_A(A, B, S, lambd, sigma):
"""
gradient␣of␣the␣cost␣function␣with␣
    respect␣to␣A␣(i.e.,␣dC/dA)

Parameters
----------
A:␣activations␣with␣shape␣(n_sub,␣n_bas)
B:␣bases␣functions␣(n_bas,␣n_pix)
S:␣image␣patches␣(n_sub,␣n_pix)
lambd:␣weighs␣the␣importance␣of␣
    reconstruction␣error␣and␣sparsity
sigma:␣activation␣scale␣hyper-parameter␣
    in␣the␣cost␣function

Returns
-------
dc:␣gradient␣of␣the␣cost␣function␣dC/dA

"""
n_sub = np.shape(A)[0]

dc = (2 / n_sub) * ((A @ B @ B.T) - (S @
    B.T)) + (2 * lambd / n_sub) * (A / (
    sigma**2 + A**2))

return dc
```

```
def dcost_B(A, B, S, lambd, sigma):
"""
gradient␣of␣the␣cost␣function␣with␣
    respect␣to␣B␣(i.e.,␣dL/dB)

Parameters
----------
A:␣activations␣with␣shape␣(n_sub,␣n_bas)
B:␣bases␣functions␣(n_bas,␣n_pix)
S:␣image␣patches␣(n_sub,␣n_pix)
lambd:␣weighs␣the␣importance␣of␣
    reconstruction␣error␣and␣sparsity
```

```
sigma: activation scale hyper-parameter
    in the cost function

Returns
-------
g: gradient of the cost function dL/dB
"""
n_sub = np.shape(A)[0]

g = 2 * (A.T @ A @ B - A.T @ S)/ n_sub

return g
```

```
    def pca_whiten_image_set(
        images,
        patch_size=8,
        n_patches=1000):
"""
PCA-whiten a set of images using
    statistics from random patches

Parameters
----------
images: numpy.ndarray
Set of images with shape (n_images,
    n_pixels) e.g. (9, 512*512)
patch_size: int
Size of square patches to extract (
    default 8)
n_patches: int
Number of random patches to sample (
    default 1000)

Returns
-------
whitened_images: numpy.ndarray
PCA-whitened images with same shape as
    input
"""
# Full image SVD to get PCA
_, s, vt = np.linalg.svd(images - np.mean
    (images, 0), full_matrices=False)

# Sample patches to compute whitening
    scaling needed while maintaining
    efficiency
patches = sample_patches(images,
    patch_size, n_patches)
_, patch_s, _ = np.linalg.svd(patches -
    np.mean(patches, 0), full_matrices=
    False)

scaling = np.std(patch_s)

whitened_images = np.zeros_like(images)
for i, img in enumerate(images):
img_centered = img - np.mean(img)
img_proj = img_centered @ vt.T

# Scale the projections using the
    singular values from patches
whitened_proj = img_proj / (scaling + 1e
    -5) # Regularisation for numerical
    safety

whitened_images[i] = whitened_proj @ vt

return whitened_images
```