Information Processing and the Brain: Coursework
Sparse Coding

Samuel Sutherland-Dee, ss15060

1 Algorithm

The sparse coding algorithm proposed by Olshausen & Field [1] relies on the assumption that an image $I(x, y)$ can be represented as a linear superposition of over-complete basis functions $\phi_i(x, y)$ that are not necessarily orthogonal.

$$I(x, y) = \sum_i a_i \phi_i(x, y) \tag{1}$$

where a_i is a coefficient (or weight) assigned to each basis function. Each image will have a different set of coefficients in its reconstruction. The goal of the algorithm is to train a set of basis functions such that only very few are needed to reconstruct any image. In other words, for each image, only a few coefficients should be non-zero. The search for such basis functions can be formulated as an optimisation problem with the cost function:

$$E = -\left[\text{preserve information}\right] - \lambda\left[\text{sparseness of } a_i\right] \tag{2}$$

where λ is a positive constant that defines the importance of sparseness in the coefficients relative to the accuracy of the reconstruction. The first term is how well the reconstruction represents the image and is defined by the mean squared error between the actual image and it's reconstruction:

$$\left[\text{preserve information}\right] = -\sum_{x,y}\left[I(x, y) - \sum_i a_i \phi_i(x, y)\right]^2 \tag{3}$$

The second term should punish reconstructions where the activity is spread over many coefficients and reward sparse activity. The solution is to sum the coefficients activity passed through a non-linear function S:

$$\left[\text{sparseness of } a_i\right] = -\sum_i S\left(\frac{a_i}{\sigma}\right) \tag{4}$$

where sigma is a scaling constant. Some examples of S that the authors propose are displayed in Figure 1 and all of these examples prefer activity states with the fewest number of non-zero coefficients.

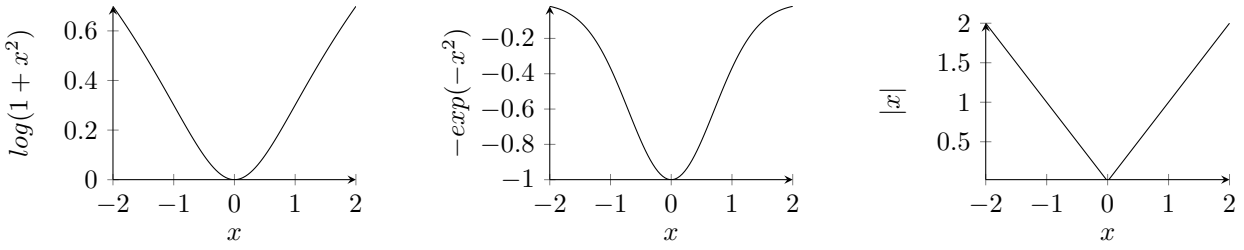


Figure 1: Example sparsity functions

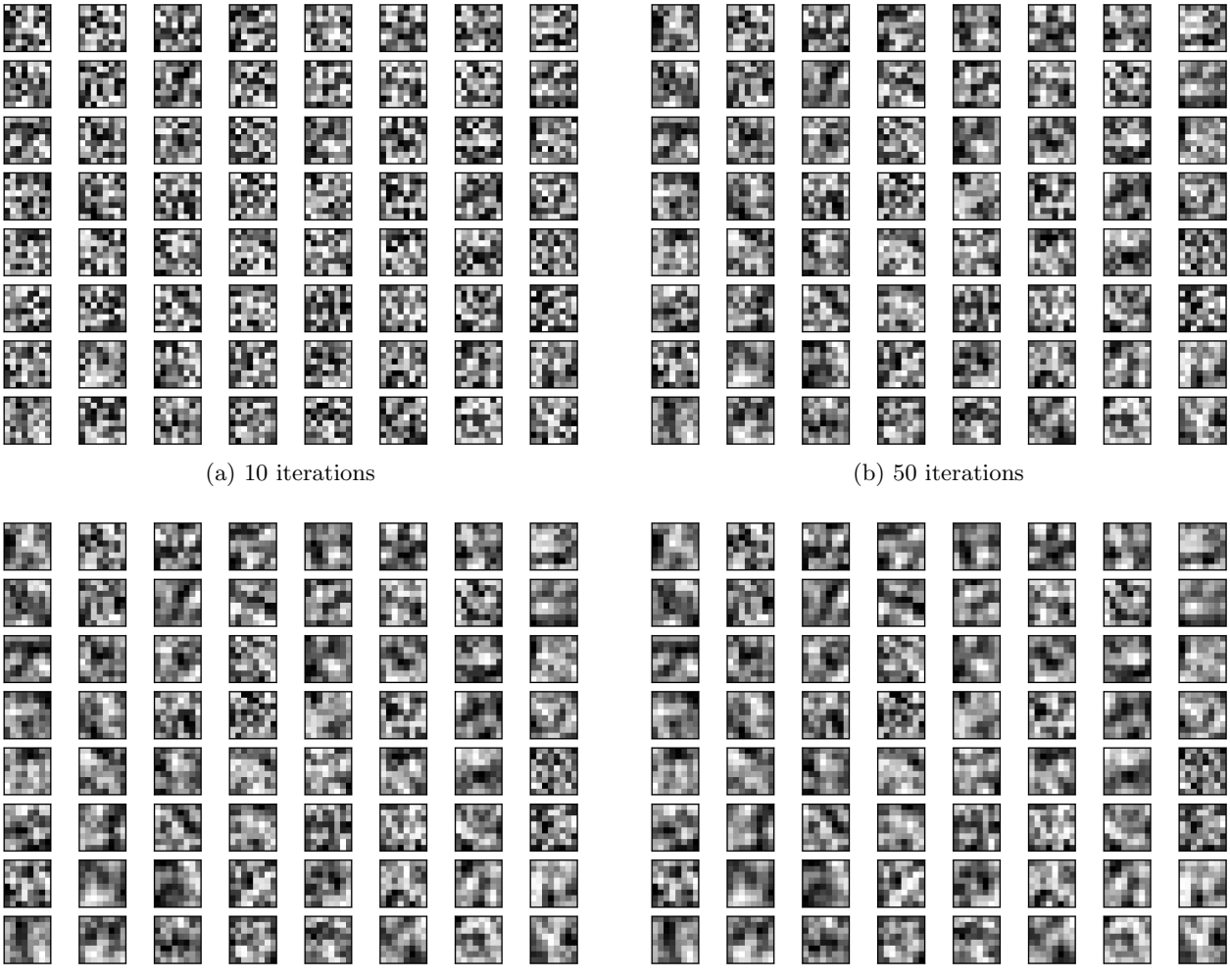
Learning is thus accomplished by minimising the cost function E by utilising gradient descent. For each iteration, a random batch of image patches are drawn and gradient descent is first performed on the a_i for each image in the batch. The basis images ϕ_i then evolve averaged over the images in the batch. For each image, the a_i are determined by the equilibrium to the equation:

$$\dot{a_i} = b_i + \sum_j C_{ij} a_j - \frac{\lambda}{\sigma} S'\left(\frac{a_i}{\sigma}\right) \tag{5}$$

where $b_i = \sum_{x,y} \phi_i(x,y) I(x,y)$ and $C_{ij} = \sum_{x,y} \phi_i(x,y)\phi_j(x,y)$. Once this gradient descent has converged then the coefficients a_i are the best sparse representation that can be achieved with the current basis images. The basis images can then be updated by gradient descent, calculating the gradient at each step by evaluating:

$$\Delta\phi_i(x_m, y_n) = \eta \left\langle a_i \left[I(x_m, y_n) - \hat{I}(x_m, y_n) \right] \right\rangle \tag{6}$$

where $\hat{I}(x_m, y_n) = \sum_i a_i \phi_i(x_m, y_n)$ is the reconstructed image and η is the learning rate. The algorithm allows for over-complete and non-orthogonal basis functions, with the coefficients determining the best basis functions to combine into the image reconstruction.



(a) 10 iterations

(b) 50 iterations

(c) 100 iterations

(d) 200 iterations

Figure 2: Basis images at various stages of training

The basis images at various stages of training can be seen in Figure 2. After 200 iterations the basis images

are fairly sparse and can be used to reconstruct images well. Due to the large computational time and cost of training, it was halted when the basis images were reasonable. To further improve runtimes, the Iterative Thresholding and Shrinkage Algorithm (ISTA) [2] was implemented instead. The algorithm works in a similar way, however instead of explicitly evaluating the sparseness of the coefficients, they are passed through a shrinkage function as seen in Figure 3. This has the effect of shrinking small coefficients to zero.

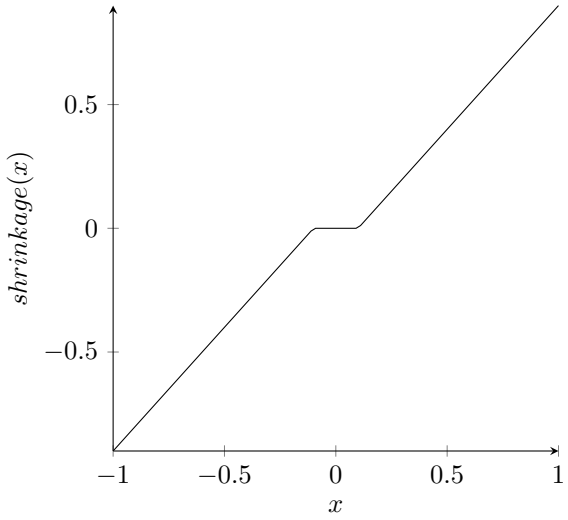
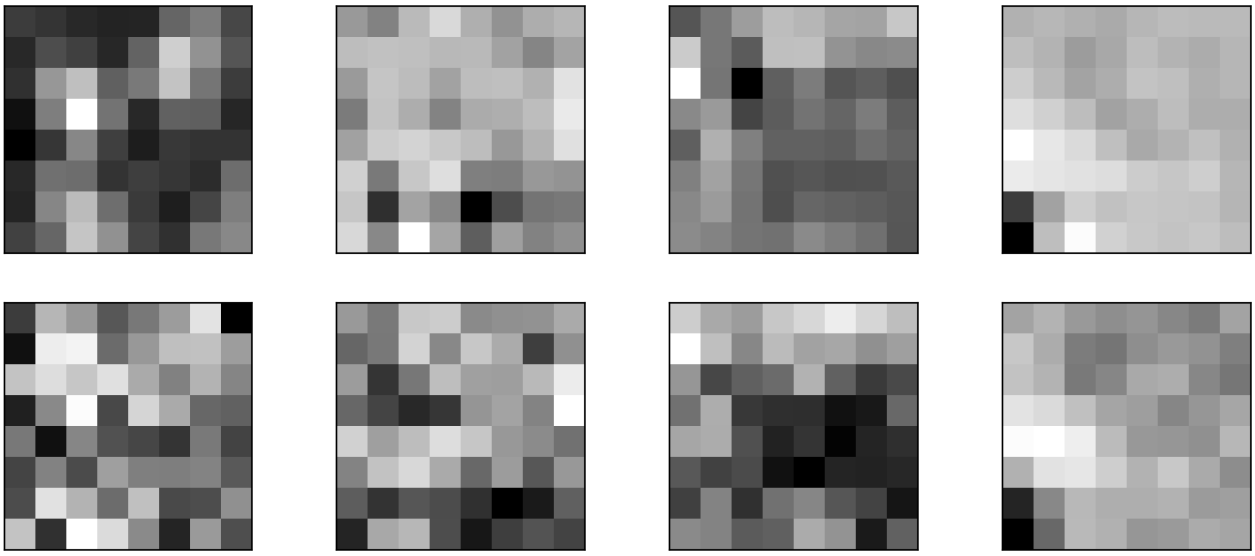


Figure 3: Shrinkage function used in ISTA

Figure 4 shows the actual image patches on the top row and their corresponding reconstructions on the bottom row, at various stages of training. The accuracy of the reconstruction can be seen to improve dramatically, from almost random pixel values at 10 iterations to a fairly accurate representation at 200 iterations.



(a) 10 iterations (b) 50 iterations (c) 100 iterations (d) 200 iterations

Figure 4: The top row contains random image patches extracted from the training set and the bottom row contains the reconstruction of these image patches at various stages of training.

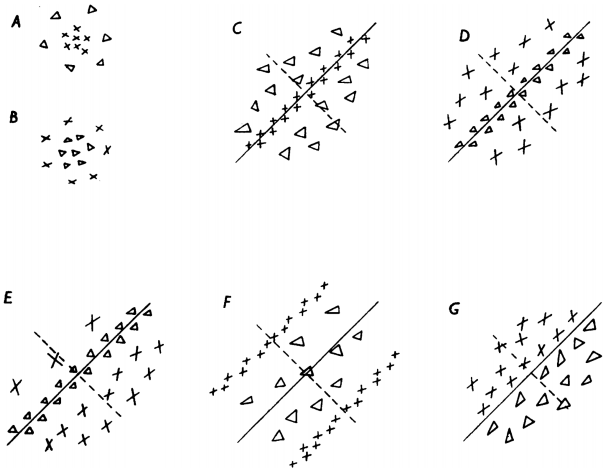
Included is the code that fits the basis images to the given image patches using the ISTA algorithm:

```
# basis_functions (num_basis, basis_width * basis_height) array
# image (batch_size, basis_width * basis_height) array
def ista(images, basis_functions, E):
   weights = np.random.uniform(0, 1, (batch_size, num_basis))

   for e in range(E):
      # calculate gradient
      reconstructed_images = weights.dot(basis_functions)
      gradient = -2.0 * ((images - reconstructed_images).dot(basis_functions.transpose()))
      # adjust weights
      weights = weights - ista_learning_rate * gradient
      # ensure sparsity
      weights = shrinkage(weights)

   return weight
```

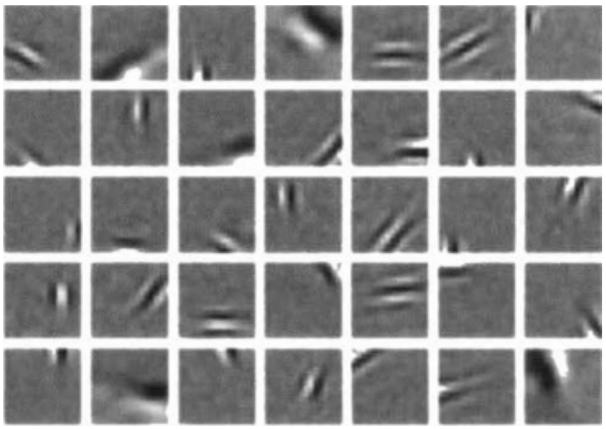
2 Relation to the Brain

Sparse coding is the idea that activity in the brain and therefore associated pieces of information is encoded in a small number of neurons. At one extreme, one piece of information could be encoded in the activation of one neuron, a so called local code. This suffers from poor representational capacity since N neurons can only encode N pieces of information. At the other extreme, the activation's of all N neurons could be used for every piece of information, a dense code. This in theory results in a higher representational capacity however suffers from the fact that all neurons can only represent one piece of information at a time, among other drawbacks. Sparse coding is seen as the compromise between dense and local codes, in which only a subset of the neuron population is required to represent a piece of information [3]. Sparse coding is more resilient to perturbation and more energy efficient than the other codes.

The algorithm implemented in Section 1 relates specifically to sparse coding in the visual system, specifically the primary visual cortex or V1. In an experiment, Vinje and Gallant [4] displayed natural images to macaques and measured the responses in V1. They found that during natural vision, the classical and non-classical receptive fields function together to form a sparse representation of the visual world. They also assert that this sparse encoding may be computationally efficient for both early vision and higher visual processing.



(a) Cat receptive fields discovered by Hubel and Wiesel [5]



(b) Sparse codes generated by the algorithm described in Section 1

Figure 5: Comparison of cat receptive fields and sparse coding generated basis images

In Figure 5 the similarity between receptive fields in a cat's visual cortex and the sparse basis functions generated by the algorithm in Section 1 is displayed. The cat receptive fields are the result of the famous Hubel and Wiesel experiment [5]. This is further evidence for the sparse coding model in the visual cortex and in addition, Olshausen and Field discuss how an over-complete basis set, such as that provided by the algorithm in Section 1 is a possible explanation for the weak forms of non-linearity observed in the response properties of cortical simple cells.

3 Advantages

One advantage of the sparse coding algorithm proposed in Section 1 is that the result of training is very biologically plausible, as discussed in Section 2. Another advantage is that unsupervised learning does not require labelling of the data before training unlike supervised learning and it also requires less data than supervised learning. The algorithm only uses natural images to train the sparse codes. Unsupervised learning in general is undertaken in the cerebral cortex and is the main driver of learning [6], however it is not clear how supervised learning can be directly translated into the brain, specifically the two phase nature in training-inference. It has been suggested that one network of neurons could provide some sort of training signal to another network to influence training [7], specifically the cerebellum generating these training signals [6].

4 Disadvantages

The major disadvantage of this algorithm is that it is extremely slow to train. Each iteration of the algorithm requires many computationally expensive gradient descent steps to fit the basis images to the image representation. The authors that proposed the algorithm trained their examples with 4000 iterations. Reinforcement learning requires no data since the subject trains by exploring the environment and discovering reward. In contrast to supervised and reinforcement learning, there is no explicit target output associated with each input; rather the result of unsupervised learning is which part of the input should be captured in the output of the network. This could be seen as a disadvantage in the sense that it is not necessarily adding explicit control or direction to computation in the brain [8].

References

[1] Bruno A Olshausen and David J Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". In: *Nature* 381.6583 (1996), p. 607.

[2] Ingrid Daubechies, Michel Defrise, and Christine De Mol. "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 57.11 (2004), pp. 1413–1457.

[3] Michael Beyeler et al. "Neural correlates of sparse coding and dimensionality reduction". In: *PLOS Computational Biology* 15.6 (June 2019), pp. 1–33. DOI: 10.1371/journal.pcbi.1006908. URL: https://doi.org/10.1371/journal.pcbi.1006908.

[4] William E Vinje and Jack L Gallant. "Sparse coding and decorrelation in primary visual cortex during natural vision". In: *Science* 287.5456 (2000), pp. 1273–1276.

[5] David H Hubel and Torsten N Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of physiology* 160.1 (1962), pp. 106–154.

[6] Kenji Doya. "Complementary roles of basal ganglia and cerebellum in learning and motor control". In: *Current opinion in neurobiology* 10.6 (2000), pp. 732–739.

[7] Eric I Knudsen. "Supervised learning in the brain". In: *Journal of Neuroscience* 14.7 (1994), pp. 3985–3997.

[8] Peter Dayan, Maneesh Sahani, and Grégoire Deback. "Unsupervised learning". In: *The MIT encyclopedia of the cognitive sciences* (1999).