# **Rectified Gaussian Distributions and the Formation of Local Filters From Video Data**

#### Emilio Corchado, Darryl Charles and Colin Fyfe

Applied Computational Intelligence Research Unit Paisley University Scotland fyfe-ci0,char-ci0,corc-ci0@paisley.ac.uk

#### Abstract.

We investigate the use of an unsupervised artificial neural network to form a sparse representation of the underlying causes in a data set. By using fixed lateral connections that are derived from the Rectified Generalised Gaussian distribution, we form a network that is capable of identifying multiple cause structure in visual data and grouping similar causes together on the output response of the network. We show that the network may be used to form local spatiotemporal filters in response to real images contained in video data.

#### 1. Introduction.

The primary visual cortex is the part of the neocortex that receives visual input from the retina. It has been shown [1] that the primary visual cortex consists of cells that have different receptive field characteristics. Some simple cells in the primary visual cortex are called edge detectors because they respond most vigorously to a luminance edge in the proper orientation and of the proper polarity. Others are called line detectors or bar detectors because these respond maximally to bright or dark lines. It has been argued [2] that the visual cortex uses a factorial code to represent the visual environment. Based on this hypothesis, a number of researchers have designed artificial unsupervised neural networks to extract factorial codes that seem to capture part of the function of the primary visual cortex [3; 4]. Recently, [5] have found spatiotemporal filters working with natural scenes whose properties are similar to those of simple cells in the primary visual cortex of mammals.

In this paper we have taken another approach to the extraction of spatiotemporal filters from video data. We apply an unsupervised network architecture based on the rectified Gaussian distribution so that we can find the underlying structural causes in the data and through the use of lateral connections the network output response is ordered to reflect the nature of the causes in the data.

#### 2. The Rectified Gaussian Distribution.

The standard Gaussian distribution [7] is defined by:

$$P(y) = Z^{-1} e^{-\beta E(y)}$$
, with energy:  $E(y) = \frac{1}{2} y^{T} A y - b^{T} y$ .

The quadratic energy function E(y) is defined by the bias parameter b and the symmetric matrix A. The parameter  $\beta = 1/T$  is an inverse temperature and the factor Z normalizes the integral of P(y) to unity.

The Rectified Gaussian distribution (RGD) is a modification of the standard Gaussian distribution in which the variables are constrained to be non-negative, enabling the use of non-convex energy function. The modes of a Rectified Gaussian are the minima of the energy function, subject to non-negativity constraints. The modes of the distribution characterize much of its behaviour at low temperatures [6]. Lowering the temperature concentrates the distribution at the minimum of the energy function. The behaviour of the distribution is characterised by its mode when  $\beta$  is large, so investigation of the distribution of y concentrates on the values of y at its mode. Different regimes may be defined by determining different values of A and b. It is possible to identify values of A which give increasingly sparse firings and in the extreme a single neuron will respond to the whole data set. The sorts of energy function that can be used are only those where the matrix. A has the property that  $y^T Ay > 0$  for all y > 0. This condition is called copositivity. This property blocks the directions in which the energy diverges to negative infinity. Finding the modes of a rectified Gaussian is a problem in quadratic programming.

This paper will focus on the use of the cooperative distribution which is defined by:

$$A_{ij} = \delta_{ij} + \frac{1}{N} - \frac{4}{N} \cos\left(\frac{2\pi}{N}(i-j)\right) \text{ and } b_i = 1.$$

To speed learning, the A matrix may be simplified to:

$$A_{ij} = \left(\delta_{ij} - \cos(2\pi(i-j)/N)\right)$$

The matrix  $\mathbf{A}$  is used to form a relationship between the output neurons of the network in response to the data set, based on the distances between the outputs. The modes of this distribution can be thought of as existing on a ring in high dimensional space. Sampling from this distribution gives a bubble of activity which may be anywhere on this ring. The mode of the distribution can be approached by gradient descent on the derivative of the energy function with respect to y. This is:

$$\Delta y \propto -\frac{\partial E}{\partial y} = -(Ay - b) = b - Ay;$$

So the lateral connection (which acts after the feed forward but before the feedback) takes the form:

$$y_i(t+1) = [y_i(t) + \tau(b-Ay)]^+$$

where the rectification is necessary to ensure the y-values remain non-negative and the parameter  $\tau$  represents the strength of the lateral connections.

If the step size  $\tau$  is chosen correctly, this algorithm can provably be shown to converge to a stationary point of the energy function [7]. In practice, this stationary point is generally a local minimum.

# 3. The Algorithm.

We adapt a network architecture that has previously been used to find multiple cause structure in data[8], so that it now includes lateral connections on the outputs as derived from the RGD. The network operation is as follows:

Feedforward:  $y_{i} = \sum_{j=1}^{N} \omega_{ij} x_{j}, \quad \forall_{i}$ {Either Lateral Inhibition:  $y_{i}(t+1) = [y_{i}(t) - \tau(b - Ay)]^{+}$ {Or Lateral Excitation:  $y_{i}(t+1) = [y_{i}(t) + \tau(b - Ay)]^{+}$ Feedback:  $x_{j}(t+1) = x_{j}(t) - \sum_{i=1}^{M} \omega_{ij} y_{i}$ Weight Update:  $\Delta \omega_{ij} = \eta x_{j}(t+1) y_{i}$ With - N-dimensional input vector at time t, x(t). - M-dimensional output vector y. and where -  $\omega_{ij}$  are the weight vectors linking each input j to each output i.

-  $\eta$  is the learning rate coefficient.

Feedback is said to exist in a system whenever the output of an element in the system influences in part the input applied to that particular element, thereby giving rise to one or more closed paths for the transmission of signals around the system. It is used to maintain the equilibrium on the weight vectors. The lateral excitation is equivalent to gradient descent on the energy function and will be called "To-mode" in the following. The lateral inhibition is equivalent to gradient ascent on the energy function and will be called "From-mode".

## 4. Experiments.

We explore the performance of the network first in response to a well known artificial image data type, then to a real data set composed of video images. We are using Tomode and From-mode learning. The to-mode method encourages correlations between nearby outputs; from-mode method discourages correlations between nearby outputs.

## 4.1 Effect of changing some parameters with artificial input data.

In this section we investigate the effect of changing the parameter  $\tau$ , which represents the strength of the lateral connections, in the case of using To-mode and From-mode learning. We use the network as described in Section 3 and additive noise

has been added in a graduated manner across the outputs, with the first output having the least noise added, the last having the most. The addition of noise in this way forces the coding of the features to shift across the output space and improves the coding performance of the network [9]. To show the effect of this parameter for both sorts of learning all the parameters have been kept steady except  $\tau$ . We apply an artificial data set that is commonly used to test networks of this type. The inputs comprise of an 8×8 square grid which contains a mixture of horizontal and vertical bars [10] – each bar may appear at the input of the network with a probability of 1/8 (a bar pixel is black and has a value 1, background pixels are white with a value 0).

To-the-Mode learning: By altering the strength of the lateral connection parameter we affect the ability of the network to "gather" features together on the outputs. With a low valued  $\tau$ , we achieve a coding of both horizontal and vertical bars around a mode as predicted (Figure 1d). As we start to increase the value of  $\tau$ , the weak correlations between horizontal and vertical bars begin to have an impact on the learning. As the strength of the lateral connections becomes stronger the bars are still learned around a mode but now orientations start to separate (Figure 1c). Increasing the value of  $\tau$  further forces the network to learn only one orientation of bar (Figure 1b), however if the lateral connections are too strong then the coding of the bars may be squashed into an area of the output space that is too small for all of the bars to be coded individually (Figure 1a). The reason why one orientation of bar is suppressed (Figure 1b) is due to the pixel overlap between different orientations of bars, if the lateral excitation is strong enough between output neurons then a single output neuron may be able to switch its preference for a horizontal bar to a vertical one. For example, if at an early point in the learning process all of the bars have been learned and one single vertical bar is shown to the network then we will have two outputs (at least) responding to this pattern. The first one strongly because it is associated with that particular bar, and the second because the bar that it usually responds to has a pixel overlap with the first. Due to the strong positive lateral connections this second outputs response is increased significantly due to the correlation with the first, and so the Hebbian-style update of the weights causes this output to develop a preference for coding the same bar as the other output. As learning continues there are now 2 outputs learning the same bar but because these outputs are very strongly correlated then eventually only one output will take responsibility for this bar. In this way one orientation of bar may be eliminated from the coding-this may be horizontal or vertical.





Figure 1. Weights vectors obtained for four different values of the lateral connections parameter  $\tau$  using To-mode learning.

From-mode learning: This form of learning has been reported before[9]. This method can also be used to separate orientations of bar but this time using inhibitory connections on the lateral weights. Again for a low value of  $\tau$  ( $\tau$  =0.1) all of the features are gathered around the mode, however this time as we increase the value of  $\tau$  ( $\tau$  =0.3), bars of different orientation are pushed apart on the output space. This again is due to the pixel overlap in different orientations of bars. It is interesting to note that we can achieve separation of orientation of bars with both To-Mode and From-Mode learning, although the To-Mode method is less sensitive to the actual parameter values i.e. it is more robust.

# 4.2 Experiments using real input data.

To test the network we use real data comprising different movies of real life scenes. [5] used video clips from a great number of different sources. Thus movement in these images could be due to camera panning, movement of the targets, zooming of the camera etc. Our movies have been produced in such a way that single-cause movements have been captured. The movies are converted into a set of equally sized sequential still images and each image transformed to a storage matrix with each pixel having a value in the range [-127,128]. We sample from these images to produce the inputs to the network. The network used is the same as that described above, with the To-Mode operation. A random starting point is selected in the sequence of still images. We select as an input a square sample of size 12x12 pixels and then corresponding square samples in exactly the same positions in each of the next 11 still images in the sequence. This gives us a 12x12x12 sample patch and all of pixel values of this sample are presented to the network simultaneously as an input vector of length 1728. This process it repeated for every cycle of the learning algorithm.





Figure 2. The weight vectors connected to six different outputs of the network using To-Mode learning. Each rectangular box shows the weight vectors connected to a different output neuron (6 outputs in total). The representation is in the form of Hinton map (except black is positive in our case). Each of the 6 rectangles in Figure 2 comprises one vector of weights into outputs 15 to 20 inclusively. Within each rectangle we have split the weight vector into 12 squares each of which corresponds to 1 time slice and within each square we have arranged the weight vectors into a 12\*12 grid so that the physical representation of the magnitude of weights (the black circles) is easily understood in terms of where the greatest physical response to impulse will be.

Results achieved using To-Mode learning with this data set are shown Figure 2. This shows the weight vectors learned for the outputs numbered 15 to 20. (The number of output neurons of the network used for this experiment is 30). It can be seen that the response of the output is very local in space and time. Note that generally there is a gradual "drift" in the values of the weights in each weight vector over time, a phenomenon that may be related to the perception across the cortex of gradual movements of objects.

In the case of From-Mode learning there is not such continuity in the response of the output as there is for To-Mode learning. The response of the outputs is still very local in space and time as in the previous case.

## 5. Conclusion.

We have shown that an unsupervised ANN derived from the RGD is capable of identifying local features in artificial and real data. Using the well-known bars data we have further shown that we can separate orientation of bar type using To-Mode and From-Mode learning.

#### **References.**

 Hubel D. H. & Wiesel T. N., Receptive Fields, Binocular Interaction and Functional Architecture in the Cats Visual Cortex, Journal of Physiology (London) 160:106-154. 1962.
 Barlow H. B. Unsupervised learning. Neur. Comput. 1, 295-31. 1989.

[3] Field D. J. What is the goal of sensory coding? Neur. Comput. 6, 559-601. 1994.

[4] Olshausen B. A. & Field D. J. Sparse coding with an overcomplete basis set: a strategy employed by V1? Vision Res. 37, 3311-3325. 1997.

[5] van Hateren, J. H. & van der Schaaf, A. Independent component filters of natural images compared with simple cells in primary visual cortex. Proc. R. Soc. Lond. B 265, 359-366. 1998.
[6] Seung H. S., Socci N. D. and Lee D. D. The Rectified Gaussian Distribution, Advances in Neural Information Processing Systems, 10. 1998.

[7] Bertsekas D.P. Nonlinear programming. Athena Scientific, Belmont, MA, 1995.

[8] Charles D. and Fyfe C. Modelling Multiple Cause Structure Using Rectification

Constraints. Network: Computation in Neural Systems, 9:167-182, May 1998.

[9] Charles D. and Fyfe C. Rectified Gaussian Distribution and the Identification of Multiple Causes Structure in Data. ICANN 99 .1999.

[10] Földiák P., Models of sensory Coding PhD thesis, University of Cambridge. 1992.