International Journal of Neural Systems, Vol. 7, No. 3 (July, 1996) 257–262

A REGULARIZATION TERM TO AVOID THE SATURATION OF THE SIGMOIDS IN MULTILAYER NEURAL NETWORKS*

LLUIS GARRIDO† and SERGIO GÓMEZ‡
Departament d'Estructura i Constituents de la Matèria,
Facultat de Física, Universitat de Barcelona,
Diagonal 647, E-08028 Barcelona, Spain

VICENS GAITÁN
†*Institut de Física d'Altes Energies, Universitat Autònoma de Barcelona,*
E-08193 Bellaterra (Barcelona), Spain

MIQUEL SERRA-RICART
Instituto de Astrofísica de Canarias, E-38200 La Laguna (Tenerife), Spain

Received 25 July 1995
Revised 25 February 1996
Accepted 15 May 1996

In this paper we propose a new method to prevent the saturation of any set of hidden units of a multilayer neural network. This method is implemented by adding a regularization term to the standard quadratic error function, which is based on a repulsive action between pairs of patterns.

1. Introduction

Consider a multilayer neural network consisting of L layers with $n_1, \ldots, n_L$ units respectively. The equations governing the state of the net are

$$\xi_i^{(\ell)} = g\left(\sum_{j=1}^{n_{\ell-1}} \omega_{ij}^{(\ell)} \xi_j^{(\ell-1)} - \theta_i^{(\ell)}\right), \quad i = 1, \ldots, n_\ell, \quad \ell = 2, \ldots, L, \tag{1}$$

where $\boldsymbol{\xi}^{(\ell)}$ represents the state of the neurons in the ℓ-th layer, $\{\omega_{ij}^{(\ell)}\}$ the weights between units in the $(\ell-1)$-th and the ℓ-th layers, $\theta_i^{(\ell)}$ the threshold of the i-th unit in the ℓ-th layer, and g is the activation function, usually taken to be the sigmoid

$$g(h) = \frac{1}{1+e^{-h}}. \tag{2}$$

This kind of network may be seen as a composition of $\ell - 1$ functions:

$$\mathbb{R}^{n_1} \xrightarrow{f^{(2)}} [0,1]^{n_2} \xrightarrow{f^{(3)}} [0,1]^{n_3} \longrightarrow \cdots \xrightarrow{f^{(L)}} [0,1]^{n_L}.$$

Therefore, if we have a certain set of input patterns $\{\mathbf{x}^\mu \in \mathbb{R}^{n_1}, \mu = 1, \ldots, p\}$, we can study how the shape of the distribution changes when it is introduced in the net.

The parameters pertaining to multilayer neural networks are usually set with the aid of a learning algorithm such as back-propagation[6] or any of its variants, which make use of the steepest descent

*This research is partly supported by the 'Comissionat per Universitats i Recerca de la Generalitat de Catalunya' and by EU under contract number CHRX-CT92-0004.
‡Present address: Dept. d'Enginyeria Informàtica (ETSE), Univ. Rovira i Virgili, Crt. de Salou s/n (Complex educatiu), E-43006 Tarragona (Spain).

method and of the chain rule so as to minimize the squared error function

$$E = \frac{1}{2}\sum_{\mu=1}^{p}\sum_{i=1}^{n_L}(\xi_i^{(L)}(\mathbf{x}^\mu) - z_i^\mu)^2\,, \qquad (3)$$

where $\mathbf{z}^\mu$ is the desired output for the input $\mathbf{x}^\mu$, and $\boldsymbol{\xi}^{(L)}(\mathbf{x}^\mu)$ is its corresponding output through the net.

A phenomenon which may arise due to the use of sigmoidal units is the *saturation* of the outputs of some of the hidden units. By saturation we mean that the output of a neuron is almost constant for most of the input patterns, since the activations they generate are far beyond the 'linear' regime of the sigmoid. This behaviour should not represent any problem in most of the cases. Nevertheless, if the net has to minimize the loss of information from layer to layer, saturation should be avoided. This is what happens, for instance, in self-supervised back-propagation and, in particular, in data compression.

Self-supervised back-propagation is the particular case of back-propagation performed with desired outputs equal to their inputs ($\mathbf{z}^\mu = \mathbf{x}^\mu$, $\mu = 1, \ldots, p$). Thus, the net carries out an *unsupervised learning* of the input distribution. From a theoretical point of view, it is most interesting to note that in the simplest case, i.e. with just one hidden layer, linear activation functions and $n_2 \leq n_1$, self-supervised back-propagation is equivalent to *principal component analysis.*[7] This means that the network projects the input distribution onto the span of its first n_2 principal components, thus capturing as much information as possible in a linear case. One step forward consists in using sigmoidal activation functions, which introduce non-linearities.[8] However, it can be shown that neither one nor two hidden layers suffice to allow the non-linearities to improve the information preservation.[4,9]

Self-supervised back-propagation applied to a multilayer network with a hidden layer carrying a minimum number of units can be used for *data compression.* For instance, this method was successfully employed by Cottrell, Munro and Zipser to the problem of image compression.[1] They trained a multilayer net, whose architecture was $L = 3$, $n_1 = n_3 = 64$ and $n_2 = 16$, with patterns consisting of cells 8×8 pixels chosen randomly on a certain image.

In general, supposing that the activations of the hidden units are discretized to a precision lower than or equal to the number of grey-levels of the inputs, the resulting net can be decomposed into

$$\text{Input} \in [0,1]^{n_1} \xrightarrow{f_c} \text{Compressed} \in [0,1]^{n_H} \xrightarrow{f_d} \text{Output} \in [0,1]^{n_L}\,,$$

where the 'bottle-neck' layer is the H-th one, and

$$f_c = f^{(H)} \circ \cdots \circ f^{(2)}\,,$$
$$f_d = f^{(L)} \circ \cdots \circ f^{(H+1)}\,,$$

are the compression and decompression functions respectively. It is clear that, since the bottle-neck layer should carry as much information of the input distribution as possible, saturation becomes a dangerous problem.

In the next section we will introduce a new regularization term to prevent the saturation of any set of hidden units. Our method could be applied whenever this situation is detected. To study the effects of this term we have applied it, in Sec. 3, to the example of image compression with neural nets,[1] to see if the quality of the decompressed images are improved.

2. Saturation and a Regularization Term

Suppose that we have detected that a subset of units in a certain hidden layer are saturated. To simplify the notation, we will consider that these are all the n_H units in the H-th hidden layer. As a consequence, we know that the distribution $\{\boldsymbol{\xi}^{(H)}(\mathbf{x}^\mu), \mu = 1, \ldots, p\}$ fails to fill all the available space $[0,1]^{n_H}$, and hence a lot of information is lost (the ideal solution would be a flat distribution).

The solution we propose is the addition of a *regularizarion term* to the quadratic error E of Eq. (3), in a similar fashion to the case of weight decay,[2] but with a completely different aim (weight decay is mainly used to control the size of the weights). More precisely, we introduce a *repulsive term* between pairs of $\boldsymbol{\xi}^{(H)}$ patterns, with *periodic boundary conditions* in the $[0,1]^{n_H}$ space. That is, our effective error function is

$$E + \lambda E^{(\text{repulsive})}\,, \qquad (4)$$

where λ is a constant and

$$E^{(\text{repulsive})} = -\frac{1}{2}\sum_{\mu\neq\nu}\sum_{k=1}^{n_H}\min\{|\xi_k^{(H)}(\mathbf{x}^\mu)-\xi_k^{(H)}(\mathbf{x}^\nu)|, (1-|\xi_k^{(H)}(\mathbf{x}^\mu)-\xi_k^{(H)}(\mathbf{x}^\nu)|)\}. \quad (5)$$

The reasons why we have chosen Eq. (5) are the following. We tried several repulsive potentials between pairs $(\boldsymbol{\xi}^{(H)}(\mathbf{x}^\mu), \boldsymbol{\xi}^{(H)}(\mathbf{x}^\nu))$ in order to separate them as much as possible. However, simulations showed that they tended to accumulate half of the patterns in one 'corner' of the $[0, 1]^{n_H}$ space, and the others in the opposite one. Thus, to avoid this possibility we introduced periodic boundary conditions, which is reflected in Eq. (5) through the presence of the $\min\{\cdot, \cdot\}$ function. Once this fact was realized, the absolute value repulsive term worked much better than the rest.

The new λ parameter takes into account the relative importance of the repulsive term with respect to the quadratic error. Its value can be adjusted by means of bayesian techniques,[3] but our computer simulations indicate that good results are obtained if λ is chosen such that both errors E and $E^{(\text{repulsive})}$ are of the same order of magnitude.

A straightforward generalization consists in adding one regularization term for each hidden layer exhibiting saturation problems, but usually it should only be applied to the layer containing the minimum number of units.

3. Application to Image Compression

Let us now see the effect of the regularization term in Eq. (5) when applied to image compression, using the following scheme based on self-supervised back-propagation (for simplicity we assume that the image is 1024×1024, with 256 grey levels):

1. A 16:25:12:2:12:25:16 multilayer neural network is initialized. In this case, the bottle-neck layer is the fourth one.
2. A cell of 4×4 pixels is chosen at random on the image, and is then introduced into the net as an input pattern.
3. Next, the error between the output and input patterns is back-propagated through the net.
4. The last two steps are repeated until enough iterations have been performed.
5. f_c is read as the half of the network from the input layer to the two neck units, and f_d as the other half from the bottle-neck layer to the output units. Thus, the compression transforms a 16-dimensional distribution into a simpler 2-dimensional one.
6. The original image is divided into its 65 536 cells, 4×4 pixels each, and f_c is applied to all these subarrays. The two outputs per cell (the neck states) are stored, as the compressed image. Thus, 16 pixels are replaced by two numbers which, if stored with a precision of 1 byte each, give a minimum compression rate of 8 (a further reversible compression method could be applied to this compressed image, increasing by some amount the final compression rate).
7. Once the image has been compressed, f_c is discarded or used to compress other similar images.
8. To improve the performance of f_d a new 10:12:25:16 network is trained to decompress the image, using the neck-states of a cell plus those of their four neighbours as inputs, and the pixel values of the central cell itself as outputs.

Figure 1 displays the 2-dimensional compressed distribution corresponding to a 16-dimensional data distribution (obtained from four typical radiographies) after a sufficiently large number of training steps. It is clear that the result achieved is not the best since most of the space available is free, and the distribution is practically 1-dimensional. The reason why this distribution takes on such a funny shape lies in the mentioned saturation of the output sigmoids of f_c. Consequently, the loss of information is greater than desirable.

Figure 2 shows the compressed distribution of the same data as in Fig. 1, but after the addition of our regularization term in Eq. (5). Although Fig. 2 is not a truly uniform distribution, at least it now looks really 2-dimensional, and the saturation phenomenon has disappeared. Moreover, the quadratic part E of the regularized error in Eq. (4) falls from $E = 115.0$ in Fig. 1, to $E = 56.9$

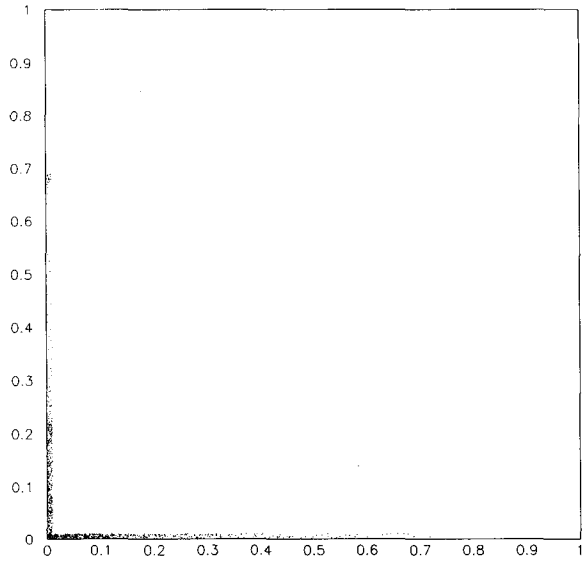

Fig. 1. Distribution of compressed images obtained using the self-supervised back-propagation.

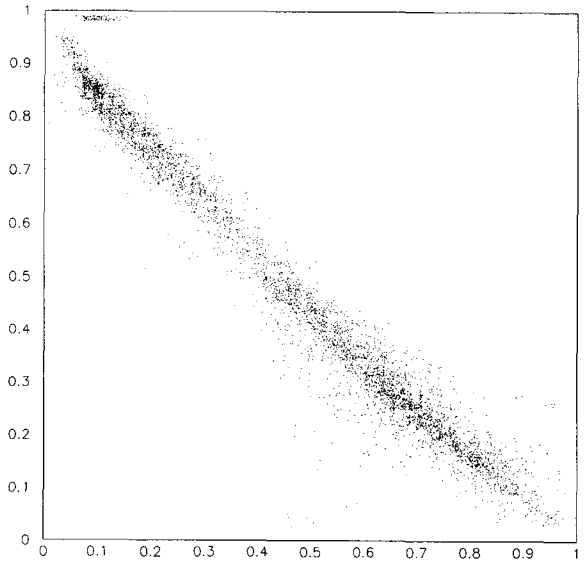

Fig. 2. Distribution of compressed images attained by means of self-supervised back-propagation with a repulsive term.

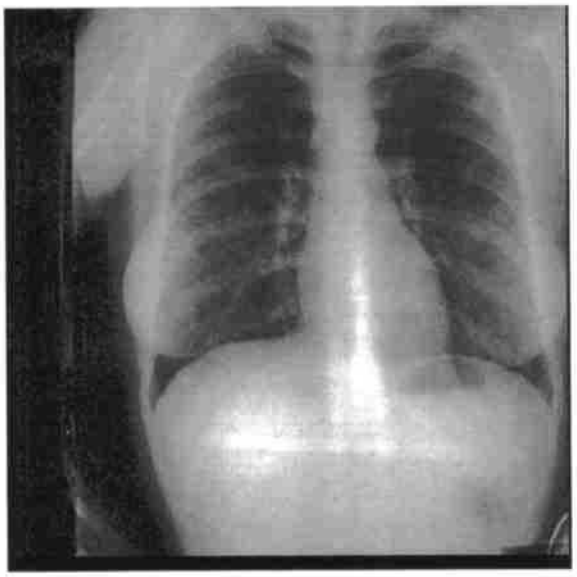

Fig. 3. Original thorax.

Fig. 4. Difference between the original thorax and the decompressed one.

in Fig. 2. This means that our method has moved the system away from a difficult local minimum.

To see if the elimination of the saturation has also served for the improvement of the quality of the decompressed images, we have applied it to the thorax in Fig. 3. Since the differences between the original image and the decompressed ones are hard to see, it is better to show directly the absolute value of the differences between both images. Thus, Fig. 4 corresponds to a standard self-supervised learning, and Fig. 5 to a learning using the repulsive term.

Fig. 5. Difference between the original of the thorax and the learnt using the repulsive term.

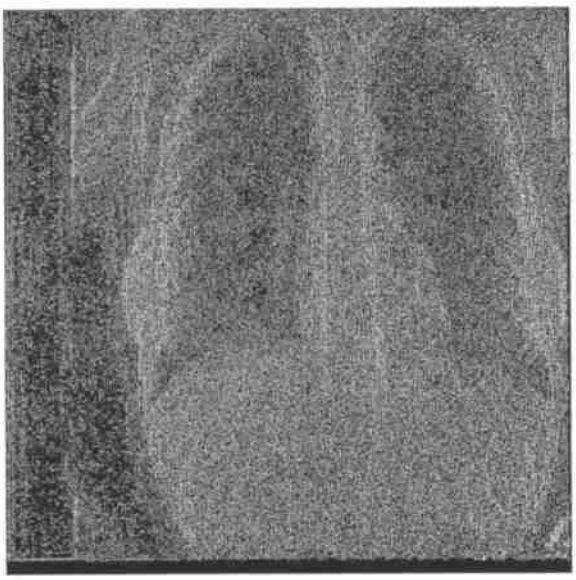

Fig. 6. Difference between the original of the thorax and the compressed and decompressed using the JPEG algorithm.

Table 1. Compression rates and quadratic errors for the three methods.

Method	Compression Rate	Error
Without repulsive term	16.85	21.52
With repulsive term	10.54	14.94
JPEG	10.81	13.95

Comparing them, it is clear that less structure is lost in the second case. Moreover, the compression rates and quadratic errors of these two images are given in Table 1. As we can see, the inclusion of the repulsive term has the effect of decreasing the error at expenses of decreasing the compression rate.

Finally, we have compared the goodness of our method using the repulsive term with the standard image compression method known as JPEG.[5] In Fig. 6 we show the result of JPEG for a compression rate nearly equal to that in Fig. 5. This compression rate and the corresponding quadratic error for this image are also shown in Table 1, finding a similar error than using the repulsive term. From these errors and the figures we see that the results of our method are competitive. However, from a practical point of view, JPEG is preferable since it needs much less CPU recurses.

4. Conclusions

We have proposed a regularization term to prevent the saturation of any set of hidden units of multilayer neural networks, which is based on a repulsive action between pairs of patterns. It has been tested in an example of image compression, showing an improvement of the image quality as demonstrated by the image difference and the quadratic error between the original image and the processed one.

References

1. G. W. Cottrell, P. Munro and D. Zipser 1987, "Learning internal representations from grey-scale images: An example of extensional programming," in *Proc. 9th Annual Conf. Cognitive Sci. Soc.* (Hillsdale, Erbaum), pp. 462–473.
2. J. A. Hertz, A. Krogh and R. G. Palmer 1991, *Introduction to the theory of neural computation* (Addison-Wesley, Redwood City, California).
3. D. J. C. MacKay 1992, "A practical Bayesian framework for backpropagation networks," *Neural Computation* **4**, 448–472.
4. E. Oja 1991, "Artificial neural networks," in *Proc. 1991 Int. Conf. Artificial Neural Networks*, eds. T. Kohonen, K. Mäkisara, O. Simula and J. Kangas (North-Holland, Amsterdam), p. 737.

5. W. Pennebaker 1990, "JPEG Technical Specification, Revision 8," Working Document No. JTC1/SC2/WG10/JPEG-8-R8.
6. D. E. Rumelhart, G. E. Hinton and R. J. Williams 1986, "Learning representations by back-propagating errors," *Nature* **323**, 533–536.
7. T. D. Sanger 1989, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks* **2**, 459–473.
8. E. Saund 1989, "Dimensionality-reduction using connectionist network," *IEEE Trans. Pattern Analysis and Machine Intelligence* **11**, 304.
9. M. Serra-Ricart, X. Calbet, Ll. Garrido, V. Gaitan 1993, "Multidimensional analysis using artificial neural networks: Astronomical applications," *Astronomical Journal* **106**, 1685.