ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 217-222

Weight Perturbation Learning Algorithm with Local Learning Rate Adaptation for the Classification of Remote-Sensing Images

F. Diotalevi and M. Valle

Department of Biophysical and Electronic Engineering
University of Genoa, Via Opera Pia 11/a, I-16145
Genoa - Italy
{diotalevi, valle}@dibe.unige.it

Abstract. The weight perturbation learning algorithm was formerly developed by hardware designers for its friendly features in the perspective of the analog on-chip implementation. Therefore it has not been used for real-world applications but it has been verified only on test problems. To significantly increase its attitude for the on-chip implementation, we proposed a local learning rate adaptation technique, which anyway, increases also the performance. At the same time to demonstrate the efficiency of the weight perturbation algorithm, in this paper we report the results of the application of the proposed algorithm to the classification of remote-sensing images. Our results compare favorably with those reported in the literature and demonstrate the soundness of the proposed approach.

1. Introduction

Artificial Neural Networks (NNs) are an efficient solution for solving many real world problems. At present, there is a growing interest in applications like Optical Characters Recognition (OCR), industrial quality control analysis and many others in which Neural Networks (NNs) can be effectively employed [1]. Many researchers have recently proposed circuit architectures for the analog VLSI implementation of Multi Layer Perceptron (MLP) based networks to achieve low power consumption, small size and high speed (i.e. portable equipment) [2]. The Weight Perturbation (WP) algorithm, was formerly developed to simplify the circuit implementation [3], [4] and although it looks more attractive then BP for the analog VLSI implementation, its efficiency in solving real world problems has not yet been heavily investigated.

In this paper, we want to evaluate and validate the WP learning algorithm in a real world application like classification of remote-sensing images. We compare our results with those obtained in [5]; this paper is a remarkable example of the application on NNs for the classification of remote-sensing images. We adopt a local learning rate architecture with an adaptive management strategy. In Section 2 we analyze the WP learning algorithm and focus our attention on the adaptive and local management of the learning rate. In Section 3 the case study is investigated while

results are reported in Section 4 and discussed in Section 5 where conclusions are also drawn.

2. The weight perturbation learning algorithm with local and adaptive learning rate

Using gradient descent optimization techniques, the learning task is accomplished by minimizing, with respect to the synaptic weights values, the output error function ε [6]. The weight update learning rule is:

$$\Delta w_{ij} = -\eta \frac{\partial \varepsilon}{\partial w_{ij}} \tag{1}$$

where η is the learning rate, ε represents the output error function to be minimized and w_{ij} is the synaptic weight that connects the i^{th} neuron to the j^{th} neuron. The main computational issue (from the circuit point of view) is the computation of $\partial \varepsilon / \partial w_{ij}$.

The WP algorithm estimates rather than calculates the gradient's value of the output error function. This method estimates the gradient simply through its incremental ratio. If the weight perturbation $p_{ij}^{(n)}$ is small enough, we can neglect the higher order terms and write:

$$\Delta w_{ij} \cong -\eta \frac{\varepsilon(w_{ij} + p_{ij}^{(n)}) - \varepsilon(w_{ij})}{p_{ij}^{(n)}} \tag{2}$$

where $p_{ij}^{(n)}$ is the perturbation injected in the w_{ij} synaptic weight at the n^{th} iteration and Δw_{ij} is the value used to update the weight w_{ij}. The difference between the output error function before and after the perturbation of the generic weight w_{ij} (see Eq. 2) is used to estimate the gradient's value with respect to w_{ij} [3].

WP is a gradient descent method and then it's possible to being trapped in local minima [3]. The shown algorithm is the simplest form of the WP algorithm: only one synapse's weight is perturbed at a time; we call this technique as sequential [3]. The sequential process, however, can be slow for large size NNs. To solve this problem, some different approaches have been proposed [7].

For the case study presented in this paper we chose the fully parallel perturbation strategy [7], [9], i.e. a synchronous parallel perturbation of the weights. Moreover it gives an efficient and modular learning architecture that can be easily mapped into analog VLSI hardware [8].

For circuit implementation issues, we consider every weight perturbation $p_{ij}^{(n)}$ as equal in value but random in sign [7]:

$$p_{ij}^{(n)} = pert_{ij}^{(n)} step \tag{3}$$

where *step* is the value of every perturbation of every synaptic weight w_{ij}, while $pert_{ij}^{(n)}$ can assume *+1* or *–1* with equal probability.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 217-222

We can rewrite Eq. 2 as follows:

$$\Delta w_{ij} \cong -\eta \frac{\varepsilon(w + p_{ij}^{(n)}) - \varepsilon(w_{ij})}{p_{ij}^{(n)}} = -\eta \frac{\Delta\varepsilon}{step} pert_{ij}^{(n)} = -\eta/_{step} \Delta\varepsilon \cdot pert_{ij}^{(n)} \qquad (4)$$

We can combine the information of the term *step* in the η value, i.e.:

$$\Delta w_{ij} = -\eta' \cdot \Delta\varepsilon \cdot pert_{ij}^{(n)} \qquad \eta' = \eta/_{step} \qquad (5)$$

$$pert_{ij}^{(n)} = \begin{cases} +1 \\ -1 \end{cases} \quad with\ equal\ probability \qquad (6)$$

To compute the synapse's weight w_{ij}, we only need to compute $\Delta\varepsilon$ and to known $pert_{ij}^{(n)}$.

2.1 Improvements in the learning convergence speed

To accelerate the learning process, we adopted an adaptive and local learning rate management strategy [10]: each synapse has its own local learning rate η_{ij} and the value of each learning rate is changed adaptively following the behaviour of the local gradient error function ($\partial\varepsilon/\partial w_{ij}$). More precisely, η_{ij} is increased when during at least two successive iterations, the signs of the term $\partial\varepsilon/\partial w_{ij}$ are equal, and it is decreased when the signs of the term $\partial\varepsilon/\partial w_{ij}$, during two consecutive iterations, are opposite.

So the local learning rate update rule can be formulated as follows:

$$\eta_{i,j}^{\ l}(t+1) = \begin{cases} \eta_{i,j}^{\ l}(t)\left[\frac{\eta^{\max}}{\eta_{i,j}^{\ l}(t)}\right]^{\gamma} & when\ S^{l}_{ij}(t) = S^{l}_{ij}(t-1) \\ \eta_{i,j}^{\ l}(t)\left[\frac{\eta^{\min}}{\eta_{i,j}^{\ l}(t)}\right]^{\gamma} & when\ S^{l}_{ij}(t) \neq S^{l}_{ij}(t-1) \end{cases} \quad where\ S^{l}_{ij}(t) = sign(\partial\varepsilon / \partial w_{ij}^{\ l}) = \begin{cases} +1\ if\ \partial\varepsilon / \partial w_{ij}^{\ l} > 0 \\ -1\ if\ \partial\varepsilon / \partial w_{ij}^{\ l} < 0 \end{cases}$$

where, taking in account the generic synapse connecting the ith neuron of the lth layer and the jth neuron of the (l-1)th layer, $S^{l}_{ij}(t)$ is the sign of the gradient component $\partial\varepsilon(t)/\partial w_{ij}^{\ l}$, and $\eta^{l}_{ij}(t)$ is the learning rate value at the tth iteration; η^{max} and η^{min} are respectively the maximum and minimum values of the learning rate and γ is the learning rate adaptation coefficient [$\gamma \in (0 \div 1)$].

2.2 The learning algorithm

In Fig. 1 the WP algorithm used in our experiment is summarized. Please, note that: η_{ij} is the local learning rate, ε represents the output error function that must be minimized and w_{ij} is the synaptic weight that connects the ith neuron to the jth neuron. The *step* parameter is the value of the perturbation of each synaptic weight w_{ij}, while $pert_{ij}$ can assume *+1* or *–1* with equal probability.

The learning strategy adopted is the by-pattern approach [1].

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 217-222

The *by-pattern* examples presentation procedure introduces some randomness in the learning process that often may help in escaping from the local minima of the output error function $\mathcal{E}$; moreover, this technique is usually faster and more effective when the training set is composed of thousands of pattern examples.

```
for(each epoch){
   set each η_ij to η^min;
   set each S^l_ij(t) at 1 (for example);
   set each pert_ij^(0) at a random value;
   For(each pattern of the training set)
     {Choose a pattern in random way and put it in
          input to the network;
      Feed-Forward phase;
      Compute ε(w_ij);
      Weight Perturbation;
      Feed-Forward phase;
      Compute ε(w_ij+step·pert_ij);
      Compute Δw_ij=-η_ij·[ε(w_ij+step·pert_ij)- ε(w_ij)]·pert_ij;
      Each η_ij is adaptively updated;
      WeightUpdate;}}
```

Fig. 1 The proposed WP algorithm.

3. The case study: classification of remote-sensing images

The case study and the data sets are based on those of [5]. The data base is a multisource data set composed of images of the same geographic area acquired by two different types of airborne sensors: a Daedalus 1268 airborne thematic mapper (ATM) scanner and, a PLC-band, fully polarimetric, NASA/JPL SAR sensor. The selected data set refers to a section of 250×350 pixels of a scene acquired in a agricultural area near Feltwell, U.K.. The available ground truth was used to prepare a reference map to assess the classification error. Five land cover classes corresponding to five types of crops (sugar beets, stubble, bare soil, potatoes and carrots) are considered. Applying a simple running mean filtering to both the ATM (5×5 window) and the SAR (9×9 window) images, the noise affecting the intensity values of the images is reduced [5]. In [4] the agricultural fields images were randomly subdivided into two disjoint sets: 5124 pixels were selected from the fields of one set and used as training set, and 5820 pixels were selected from the fields of the other set and used as test set.

To evaluate more correctly the learning performance, in the experiments here presented, the pixels used in [5] as test set have been furthermore partitioned in two data set (according to [1]): 2909 pixels are used for the validation set and 2911 pixels are used for the test set. The resulting data set, is shown in Tab. 1.

Fifteen channels were selected to form a feature vector for each pixel: they selected the six ATM channels corresponding to TM channels in the visible and infrared

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 217-222

spectrum (except the thermal band) and the nine SAR channels in the PLC-band and HH-, HV-, VV-polarizations. Each feature has been normalized in the range [-1,+1].

Tab. 1 The data base used in the experiments.

		Original DB [5]			
		Test set	**Training set**	**Validation set**	**Test set**
Number of pixels (i.e. patterns) for each class	Sugar Beets	2043	1488	1021	1022
	Stubble	1371	1070	686	685
	Bare Soil	555	341	277	278
	Potatoes	884	1411	442	442
	Carrots	967	814	483	484
	Overall	5820	5124	2909	2911
			Our DB		

4. Experimental results

The size of the Multi Layer Perceptron used in the simulations was: 15×15×5 neurons. The *step* parameter (see Eq. 3) has been kept fixed to 10^{-3}. We save the weights configuration when the output error function ε, computed on the validation set, reaches the minimum [1] (i.e. at the overfitting point). The learning rate was:

- $\eta^{max}=10^{-3}$ and $\eta^{min}=10^{-5}$ for the WP with adaptive and local learning rate,
- $\eta=10^{-4}$ for the WP with fixed learning rate.

Tab. 2 Classification error (mean and standard deviation) computed over 15 trials.

Land-cover classes	**Output error value**							
	Results reported in [5]				**Our results**			
	Classical technique		**Technique proposed in [5]**		**Adaptive local learning rate**		**Fixed learning rate**	
	Mean	**St. deviation**	**Mean**	**St. deviation**	**Mean**	**St. deviation**	**Mean**	**St. deviation**
Sugar beets	1.8%	1.97	0.7%	1.43	1.6%	0.47	1.3%	0.39
Stubble	19.6%	3.78	13.0%	4.29	13.9%	3.19	14.1%	0.41
Bare soil	42.5%	16.91	17.1%	1.01	41%	5.77	39.1%	2.16
Potatoes	26.0%	6.53	23.7%	0.84	3.8%	1.60	6.4%	0.75
Carrots	20.6%	6.19	13.6%	0.25	12.6%	1.60	13.5%	0.62
Overall	16.4%	2.19	10.8%	0.81	10.4%	1.06	10.7%	0.10

From table Tab. 2 it is easy to deduce that the WP algorithm reduces the overall classification error and shows a more stable behavior versus the initial weight sets. In particular, the overall classification error made by the WP with adaptive local learning rate was found in the range 8.9% to 11.5%, the mean and the standard deviation being equal to 10,4% and 1.06 respectively. In the case of the WP with fixed learning rate the overall classification error was in the range from 10.6% to 10.8%, the mean and the standard deviation being equal to 10,7% and 0.1 respectively.

ESANN'2001 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 25-27 April 2001, D-Facto public., ISBN 2-930307-01-3, pp. 217-222

5. Conclusions

The aim of our work is to demonstrate the capabilities of the WP learning algorithm for the classification of remote sensing images: we adopted a local learning rate with adaptive management: our results are fully comparable with those obtained in [5]. This implies that WP can be efficiently used to solve classification problems in the remote-sensing images field. In particular, the results for the WP with fixed learning rate and with fully parallel perturbation strategy, proves the soundness of the VLSI architecture introduced in [8].

Acknowledgments

The authors wish to thank Prof. B. Serpico of DIBE and Hunting Technical Services Ltd for providing the remote sensing images used in the experiments.

References

1. L. Tarassenko, "A Guide to Neural Computing Applications", J. Wiley & Sons Inc., New York–Toronto, 1998.
2. D.Baratta, et al., "Microelectronic Implementation of Artificial Neural Networks," *In Proc. of the 5th Electronic Devices and Systems Conference*, EDS'98, p.p. VI-IX, Brno, Czech Republic, 1998.
3. M. A. Jabri, et al., *Adaptive Analog VLSI Neural Systems*, Chapman & Hall, 1996.
4. M. Jabri et al., "Weight Perturbation: An Optimal Architecture and learning Technique for Analog VLSI Feedforward and Recurrent Multilayer Networks," *IEEE trans. Neural Networks*, vol. 3 (1), pp. 154-157, 1992
5. L. Bruzzone et al., "A Technique for the Selection of Kernel-Function Parameters in RBF Neural Networks for Classification of Remote-Sensing Images", *IEEE Tran. On Geoscience and Remote Sensing*, vol. 37, No. 2, pp. 1179-1184, 1999.
6. J. Hertz, A. Krogh, and R.G. Palmer, "Introduction to the theory of the Neural Computation," *Addison-Wesley Publishing Company*, 1981.
7. F. Diotalevi, et al., "Evaluation and Validation of Local and Adaptive Weight Perturbation Learning Algorithms for Optical Characters Recognition Applications," *In Proceeding of Soft Computing*, Genova (Italy), June 1 - 4 1999, pp. 508-512.
8. F. Diotalevi, et al., "An Analog on-chip Learning Circuit Architecture ot the Weight Perturbation Algorithm", *in Proc. ISCAS2000*, Geneva (Switzerland), 28-31 May, 2000, pp. 419 - 422.
9. J. Alspector and D. Lippe, "A Study of Parallel Perturbative Gradient Descent," *in Advances in Neural Information Processing Systems* (NIPS96), pp. 803-810, 1996
10. G. M. Bo et al., "Analog VLSI on-chip Learning Neural Network with Learning Rate Adaptation," in chapter 14 of G. Cauwenberghs and M. A. Bayoumi eds., "Learning on silicon, adaptive VLSI neural systems," Kluwer Academic Pub. 1999.