Speeding up backpropagation with Multiplicative Batch Update Step

Pedro Cruz
Department of Mathematics, University of Aveiro, Portugal
E-mail: jpedro@mat.ua.pt

Abstract

Updating steps in a backpropagation neural network with multiplicative factors $u > 1$ and $d < 1$ has been presented by several authors. The istatistics field of Stochastic Approximation has a close relation with backpropagation algorithms. Recent theoretical results in this field show that for functions of one variable, different values of u and d can produce very different results: fast convergence at the cost of a poor solution, slow convergence with a better solution, or produce a fast move towards a solution but without converging. To speed up backpropagation in a simple manner we propose a batch step adaptation technique for the online backpropagation algorithm based on theoretical results on simple cases.

1 Introduction

Speeding up backpropagation has been a constant challenge and several techniques has been applied like using second order information [4].

Also several works on multiplicative step update where developed [5,6,8] and each of them uses the common update algorithm

$$\gamma_t = \begin{cases} \gamma_{t-1} \times u & \text{if condition C} \\ \gamma_{t-1} \times d & \text{otherwise} \end{cases} \tag{1}$$

where $u > 1$ and $0 < d < 1$ are real parameters. Possibly variable γ_t is guaranteed to be limited $0 < \Gamma_d < \gamma_t < \Gamma_u$.

Condition C could be a gradient condition [5] or a condition on the quadratic error [6,8]. In both cases the quadratic error

$$E_t = E(W_t, x, d) = (1/2)\sum_{i=1}^{n}(\mathrm{NET}_i(W_t, x) - d)^2, \tag{2}$$

is to be minimized, with (x, d) being the pattern presented at the iteration, W_t is the weight vector that describes net state, and NET_i is the ith output of the neural network.

Gradient condition uses change of signs: if

$$\frac{\delta E_t}{\delta w} \times \frac{\delta E_{t-1}}{\delta w} > 0 \tag{3}$$

then step must be incremented otherwise reduced. Different updates are going to occur for each weight w in the net parameters $W = (w_0, w_1, \ldots, w_n)$.

Another type of condition is the error condition which is based on increases or decreases of global error ΔE_t causing the step to change.

We propose an algorithm that uses a gradient condition for step update. Next we present theoretic results that are guides to the proposed algorithm.

2 Stochastic Approximation and New results

Many problems in Stochastic Approximations are described as the following. Consider the problem of searching for the zero point of a function, according to the stochastic approximation procedure

$$\begin{aligned} x_{t+1} &= x_t - \gamma_t y_t, \quad (4)\\ y_t &= \varphi(x_t) + \xi_t \quad (5)\end{aligned}$$

where φ is the function under consideration, x_t means the tth approximation of the zero x^* of φ, y_t is the tth disturbed observation of φ with random disturbance ξ_t.

If step size values γ_t of the procedure (4), (5) are deterministic and satisfy $\sum \gamma_t = \infty$, $\sum \gamma_t^2 < \infty$, the sequence of x_t is proved to converge to x^* [2]. There are various versions of (4) and (5) aiming at accelerating convergence of x_t. We are concentrating here on the methods using step size adaptation in the course of algorithm, decreasing γ_t every time that the two consecutive differences $\Delta x_{t-1} = \gamma_{t-1}y_{t-1}$ and $\Delta x_t = \gamma_t y_t$ have the same sign, and increasing γ_t, otherwise.

Consider the following update rule

$$\gamma_t = \begin{cases} \min\{u\gamma_{t-1}, \Gamma\} & \text{if} \quad y_{t-1}y_t > 0, \\ d\gamma_{t-1} & \text{if} \quad y_{t-1}y_t \leq 0, \end{cases} \tag{6}$$

$t = 2, 3, \ldots$ is used. Here $0 < d < 1 < u$, $0 < \gamma_0 \leq \Gamma_u$, $\gamma_1 \in \{\min\{u\gamma_0, \Gamma\}, d\gamma_0\}$, Γ_u is a positive constant. Let us point out the main differences with standard algorithm. Suppose that $\{\xi_t\}$ is a sequence of i.i.d.r.v. with zero mean, besides $\mathrm{P}(\xi_t > 0) = \mathrm{P}(\xi_t < 0)$. Under some additional assumptions on φ, ξ_t, and Γ_u, stated below, the process defined by (4), (5), (6) a.s. diverges if

$ud > 1$, and converges if $ud < 1$. In fact this algorithm can converge to a region near one of the zeros of φ. See [7] for details.

3 Batch Step Update and Backpropagation

Using the above algorithm with feed forward neural networks is not possible since partial derivatives changes sign very frequently indicating a non smooth surface and this cause a fast step decrease and a very poor solution is reached.

However, using constant step, one can observe two facts about a weight w. We use w_t to indicate the sequence of values of some predetermined weight of the vector of parameters W. First fact is that during pattern presentation w_t oscilates frequently, causing the training to freeze. The second fact is that observing w_t values after the full training set has been used one notice that w_t doesn't oscilate, doing some progression on the same direction each batch time. Change of weight direction occurs but not frequently as after each pattern presentation.

Consider the following 'time' definitions: time T indicate the batch number and is incremented after B pattern presentations; time t is incremented after each pattern presentation.

We propose the following batch update rule for step

$$\gamma_T = \begin{cases} \min\{\gamma_{T-1} \times u, \Gamma_u\} & \text{if } \Delta w_T \times \Delta w_{T-1} > 0 \\ \max\{\gamma_{T-1} \times d, \Gamma_d\} & \text{if } \Delta w_T \times \Delta w_{T-1} \leq 0 \end{cases} \tag{7}$$

for $T = 1, 2, \ldots$ with

1. $u > 1$ and $0 < d < 1$;
2. γ_T is bounded by Γ_d and Γ_u constants;
3. γ_0 some constant that could be much less than Γ_u since step can grow.

After the step update rule we define the weight update rule. It is known that online training has been shown to produce better solutions than weight batch update. So, each weight is going to be updated after each pattern presentation. We propose the following update algorithm for a single weight w

$$w_t = w_{t-1} - \gamma_{t/B} \frac{\delta E(W_{t-1}, x_t, d_t)}{\delta w} \tag{8}$$

with $t = 1, \ldots$,and where t/B uses integer division, and $E(W, x, d)$ is defined in (2).

4 A numerical study

We use the MNIST digit database [3] to study the new algorithm performance but using only digits $\{0, 1, 2, 3\}$.

Four identical shape neural networks where used, one for each digit. Each net has the following structure: 28×28 inputs, 10 hidden units and two outputs (first output is 1 when a good pattern is presented, otherwise is valued -1, and bad patterns are classifiyed 1 in the second output, otherwise -1).

Each digit has aproximatly 6000 exemplars (called *good patterns*). The remaning digits are 54000 *bad patterns*). So, the training set has 60000 patterns. The test set has 1000 patterns (approximately 1000 samples of each digit).

Each net was trained using the 6000 'good patterns' against other 6000 randomly choosen 'bad patterns'. So, total training set is 12000. We used these parameters: the number of t iterations was 5×12000, the step was updated every $B = 3000$ (1/4 of training set dimension) and $\Gamma_u = 0.001$ (that is $1/(\text{large fan in})$, and $\Gamma_d = 10^{-7}$. Initial step was $\gamma_0 = 0.0001$. This two values where randomly set.

Figure 1 describes the minimum wrong values on the test set in all the 5×12000 iterations. The step schedules where: **Bar 1** $u = 1.1$ and $d = 0.9/1.1$ ($ud < 1$), **Bar 2** $u = 1.1$ and $d = 1/1.1$ ($ud = 1$), **Bar 3** $u = 1.1$ and $d = 1.08/1.1$ ($ud > 1$) and last **Bar 4** with constant step $\gamma = \gamma_0 = 0.0001$ ($u = d = ud = 1$).

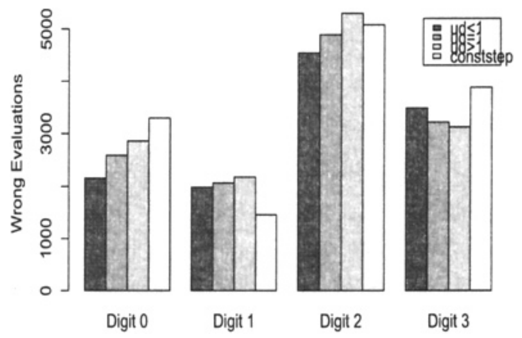

Fig. 1. Number of wrong classifications vs. step schedule in a short period of training steps. The dark bar is $ud < 1$ case, then $ud = 1$, $ud > 1$ and finally constant step.

The main observation is that the algorithm with step update rule $ud < 1$ behaves better in 3 cases: digits $\{0, 2, 3\}$.

The number of wrong classifications are very high for two reasons: small number of iterations and the described method of training makes harder for the net to classify with the simple structure.

As a conclusion of this work we can say that a very simple familly of algorithms for step update was presented and these simple results are encouraging. However, more work on these methods should be done.

5 Acknowledgements

The author wishes to thank the financial support by the Portuguese program PRODEP 'Medida 5 - Acção 5.3 - Formação Avançada de Docentes do Ensino Superior - Concurso nr. 2/5.3/PRODEP/2001'.

References

[1] Hertz, J., Krogh, A., Palmer, R. G., (1991) Introduction to the theory of neural computation, ISBN 0-201-50395-6, Addison-Wesley Longman Publishing Co., Inc.

[2] Kushner, Harold J., Yin, G.George (1997) Stochastic approximation algorithms and applications, Applications of Mathematics. 35. Berlin: Springer. xxi, 417 p.

[3] LeCun, Y., Bottou, L., Bengio Y., Haffner, P. (1998) Gradient-based learning applied to document recognition. Proc. of the IEEE, vol. 86, nr. 11, 1998.

[4] Spall, J. C., (2000) Adaptive stochastic approximation by the simultaneous perturbation method., IEEE Trans. Autom. Control, vol.45, nr.10: 1839–1853.

[5] Silva, F. M., Almeida, L. B. (1990),Speeding up backpropagation, Advanced Neural Computers, ed. R. Eckmiller, Elsevier Science Publishers, Amsterdam, pp151-158.

[6] Salomon, R., Hemmen, van J. L., (1996), Accelerating Backpropagation through Dynamic Self-Adaptation, Neural Networks, vol. 9, nr. 4: 589–601.

[7] Plakhov, A., Cruz, P. (to be published), A stochastic approximation algorithm with multiplicative step size adaptation.

[8] Battiti, R., (1992) First and second order methods for learning: between steepest descent and Newton's method, Neural Computation, vol. 4, nr. 2: 141–166, The MIT Press, Cambridge–Massachusetts.