IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 6, Ver. III (Nov.-Dec. 2016), PP 92-94
www.iosrjournals.org

A novel approach on memory less BFGS neural network training algorithm

S Shoba Rani[1], Dr.D.Nagendra Rao[2], Dr.S Vatsal[3], R.Chennakeshavulu[4]
[1]*Associate Professor, Computer Science and Engineering Vardhaman College of Engineering, Shamshabad, Telangana, India.*
[2]*Principal & Professor, Electronics and Communication Engineering,JBREC, Moinabad, Telangana, India*
[3] *Professor, ElectricalandElectronicsEngineering, Institute of Aeronatical Engineering , Telangana, India*
[4]*Associate Professor, Electronics and Communication Engineering, JBIET, , Moinabad Telangana, India*

Abstract: *A curvilinear algorithm [1] model is presented here for training neural networks which are based on modifications of the memory less BFGS [2] method. The method supports curvilinear [3] search. A memory quasi-Newton direction and a direction of negative curvature are described here. Additionally negative curvature [4] direction is allowed by avoiding any storage and matrix factorization. The efficiency of the training process is achieved by simulation results.*
Index Terms*: Neural networks, memory less BFGS, negative curvature direction, curvilinear search*.

I. Introduction

According to the trading problem of neural networks, reducing the set of weights w to minimize the error function E(W) . It is the sum of square of errors in outputs. Traditionally gradient-based algorithm formula is

$W_k+1=w_k + \eta_k p_k$

Where k is defined as current iteration called as epoch. W_o belongs to P_k is a starting point. η_k is a stepsize [5](or learning rate) with $\eta_k > 0$ and p_k is a descent search direction, i.e., $g_k^T p_k < 0$. The gradient are easily obtained by means of back propagating errors through the network layers. These suggestions were been proposed in literatures in order to define the search direction p_k. The most elaborated directions are limited memory quasi-Newton[6] direction which is defined by Hessian[7] approximation using curvature from most recent iterations. Recently a method is proposed which exploits the Eigen structure of the memory less[8] BFGS matrices without using matrix factorization and storage. Consequently, a direction of negative curvature can be computed analytically avoiding the storage of any matrix. So, the proposal for a curvilinear scheme of a memory less quasi-Newton method for training came into existence. The proposed algorithm utilizes a pair of directions; a memory less quasi-Newton direction and a direction of negative curvature, i.e., directions d such that $d_{k'}$ $2E(w)d < 0$, and it is based on the following iterative form $wk+1 = w_k + \eta_k p_k$, if B_k is positive definite; $w_k + \eta_k^2 p_k + \eta_k d_k$, otherwise where p_k is memory less quasi-Newton direction, dk is direction of negative curvature and B_k is memory less BFGS Hessian approximation. When B_k defines positive definite, proposed iterative scheme standard line search procedure (see [3], [19]). In different case, the iterative scheme searches along the curvilinear path

$w_k+1 = w_k + \eta_2 kp_k + \eta_k d_k$

which first proposed by Mor´e and Sorensen [15]. The proposed method preserves strong convergence[9] properties provided by quasi-Newton direction when d_k is positive definite. Additionally, it exploits the nonconvexity[10] of the error surface through the computation of the negative curvature direction without using any storage and matrix factorization. Euclidean norm and n the dimension of the error function.

II. Properties of the memory less BFGS matrices

The memory less BFGS algorithms are computed on L-BFGS philosophy using information from recent iteration. L-BFGS makes d_k share many features with other quasi-Newton algorithms, but it is very different in the matrix-vector multiplication for finding the search direction. , where is the current derivative and is the inverse of the Hessian matrix[12]. There is multiple published approaches using a history of updates to form the direction vector[13]. Here, we give a common approach, the so-called "two loop recursion.

We'll take as given, the position at each iteration, and the function being minimized, and all vectors are column vectors. We assume that the last stored updates of the form and we'll define, and that the 'initial' approximate of the inverse Hessian is according to our estimate when iteration begins with. Then we can compute the (uphill[14]) direction.

The formulation is validated and we are using minimizing or maximizing. Note that if we are minimizing, the search direction will be negative of k (since k is "uphill"), and if we are maximizing, it will be

negative rather than positive. This would typically do a backtracking line search in the search direction. Commonly, the inverse Hessian represented as a diagonal matrix, then it initially sets as required only an element-by-element[18] multiplication.

These two loop update works only for inverse Hessian. Approaches to implementing L-BFGS using the direct approximation. This would typically do a backtracking line search in the search direction. Commonly, the inverse Hessian represented as a diagonal matrix, then it initially sets as required only an element-by-element multiplication. These two loop update works only for inverse Hessian. Approaches to implementing L-BFGS Hessian have developed, as other means of approximating the inverse Hessian.

$W_{k+1} = W_k + \eta_k P_k$ if P_k is positive definite,

Otherwise $W_{k+1} = W_k + \eta_k^2 P_k + \eta_k d_{k..}$

These two methods uses both the first and second derivatives of the function. However, BFGS has proven that it has very good performance even for non-smooth optimizations. In quasi-Newton methods, the Hessian of second derivatives[15] are not evaluated directly. Instead, the Hessian matrix is approximation of using rank-one[16] updates specified by gradient evaluations (or approximate gradient evaluations). Quasi-Newton methods are generalizations of the known secant method[17] to find the root of the first derivative for multidimensional problems. In the multi-dimensional problems, new secant equation does not specify a unique solution, and quasi-Newton methods differ in how they constrain the solution. So, the BFGS method is known as the most popular members of the class which is known as common use is L-BFGS[20], is a limited-memory version of BFGS which is particularly suited for problems with very large numbers of variables (e.g., >1000). The BFGS variant handles simple box constraints.This search direction P_k at stage *k* is given by solution of the analogue of Newton equation which is an approximation of Hessian matrix and is updated iteratively at each stage, and is gradient of the function is evaluated at W_k. A line search in this direction P_k which is used to find the next point W_{k+1}. Instead of the required full Hessian matrix at the point w_{k+1} which is to be computed as W_{k+1}, the approximation Hessian stage *k* is updated by the addition of two matrices.

Both W_k and P_k are symmetric rank-one matrices but have different (matrix) bases. The symmetric rank made by one assumption. So these equivalents, W_k and P_k construct a new rank-two update matrix which robust against the scale problem. The quasi-Newton condition imposed on this updated. In the paper of conjugate we develop a limited memory conjugate gradient method which corrects the loss of orthogonality that will occur in ill-conditioned optimization problems. In our method, we check distance between current gradient and the space P_k spanned by the recent prior search directions. When distance becomes sufficiently small, then the orthogonality property has been lost, and here we optimize the objective function over P_k until achieving a gradient that is approximately orthogonal to P_k. This approximation orthogonality condition is eventually fulfilled by first-order[22] optimality conditions for local minimizer in the subspace. The algorithm continuously operate in this same way: We apply the conjugate gradient iteration until distance between current gradient and P_k becomes sufficiently small, and then we solve a subspace problem to obtain an iterate for which gradient is approximately orthogonal to P_k. Our limited memory algorithm has connections with both L-BFGS of Necedal [20] and Liu and Nocedal [16], and with reduced Hessian method of Gill and Leonard [10, 11]. Unlike either of these limited memory approaches, we do not always use memory to construct new search direction. This memory is used to monitor orthogonality of the search directions; and when orthogonality is lost, memory is used to generate a new orthogonal search direction. Our rational for not using the memory to generate the current search when orthogonality[21] holds is that conjugate.

III. Algorithm

From an initial guessing we approximate the Hessian matrix the following steps are repeated as stages to the solution.

1. Obtain a direction by the solving.
2. Perform line search to find an acceptable step size in the direction found in the first step, then update .
3. If stepsize found as negative then isolate
4. Otherwise, it is treated as positive.
5. stop

Which denotes the objective function to be minimized. This can be checked by observing under the norm of gradient. Practically, which can be initialized with, so the first step will be equivalent to a gradient descent, but the further steps are more and more refined by, the approximation to the Hessian. This first step of algorithm is carried out by using inverse of the matrix, which can also be obtained efficiently by applying the Sherman–Morrison formula to the algorithm, this will be computed efficiently without using temporary matrices, recognizing that it is symmetric, and is that which is scalar, using an expansion .

In the statistical estimation problems (such as maximum likelihood or the inference), credible intervals or confidence intervals for solution will be estimated from inverse of the final Hessian matrix. However, these quantities are technically defined by true Hessian matrix, and the BFGS approximation may not converge to the true Hessian matrix.

IV. Conclusions

The work, have proposed a new curvilinear method for training the neural networks which is based on the analysis of the eigen structure of the memory less BFGS matrices. This method preserves the strong convergence properties provided by the new quasi-Newton direction while at the same time it exploits the nonconvexity of the error surface through the negative curvature direction without using any storage and matrix factorization. Based on the fact that the algorithm uses only inner products and vector summations, this proposed method is suitable for training large scale neural networks. Our numerical experiments have shown that the method outperforms other popular training methods on famous benchmarks.

References

[1]. A.D. Anastasiadis, G.D. Magoulas, and M.N. Vrahatis. New globally convergent training scheme based on the resilient propagation algorithm. Neurocomputing, 64:253–270, 2005.
[2]. M.S. Apostolopoulou, D.G. Sotiropoulos, and P. Pintelas. Solving the quadratic trust-region subproblem in a low-memory BFGS framework. Optimization Methods and Software,23(5):651–674, 2008.
[3]. L. Armijo. Minimization of functions having Lipchitz continuous partial derivatives. Pacific Journal of Mathematics, 16:1–3, 1966.
[4]. J. Barzilai and J.M. Borwein. Two point step size gradient methods. IMA Journal of Numerical Analysis, 8:141–148, 1988.
[5]. R. Battiti. First and second order methods for learning: between steepest descent and Newton's method. Neural Computation,4:141–166, 1992.
[6]. E. Dolan and J.J. Mor´e. Benchmarking optimization software with performance profiles. Mathematical Programming, 91:201–213, 2002.
[7]. J.Ch. Gilbert and X. Jonsson. LIBOPT-An enviroment for testing solvers on heterogeneous collections of problems - version 1.CoRR, abs/cs/0703025, 2007.
[8]. D. Goldfarb. Curvilinear path steplength algorithms for minimization which use directions of negative curvature. Mathematical Programming, 18:31–40, 1980.
[9]. J. Hertz, A. Krogh, and R. Palmer. Introduction to the Theory of Neural Computation. Addison-Wesley, Reading, MA, 1991.
[10]. P. Horton and K. Nakai. A probabilistic classification system for predicting the cellular localization sites of proteins. In 4thInternational Conference on Intelligent Systems for Molecular Biology, pages 109–115, 1996.
[11]. P. Horton and K. Nakai. Better prediction of protein cellular localization sites with the k Nearest Neighbors classifier. In Intelligent Systems in Molecular Biology, pages 368–383, 1997.
[12]. I. Ipsen. Computing an eigenvector with inverse iteration. SIAM Review, 39:254–291, 1997.
[13]. R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In International JointConference on Artificial Intelligence, pages 223–228. AAAIPress and MIT Press, 1995.
[14]. D.C. Liu and J. Nocedal. On the limited memory BFGS methodfor large scale optimization methods. Mathematical Programming, 45:503–528, 1989.
[15]. J.J. Mor´e and D . Sorensen. On the use of directions of negative curvature in a modified Newton method. MathematicalProgramming, 16:1–20, 1979.
[16]. P.M. Murphy and D.W. Aha. UCI repository of machine learning databases. Irvine, CA: University of California, Departmentof Information and Computer Science, 1994.
[17]. D. Nguyen and B. Widrow. Improving the learning speed of 2-layer neural network by choosing initial values of adaptive weights. Biological Cybernetics, 59:71–113, 1990.
[18]. J. Nocedal. Updating quasi-Newton matrices with limited storage. Mathematical Computing, 35(151):773–782, 1980.
[19]. J. Nocedal and S. J. Wright. Numerical Optimization. SpringerVerlag, New York, 1999.
[20]. L. Prechelt. PROBEN1-A set of benchmarks and benchmarking rules for neural network training algorithms. Technical Report 21/94, Fakultt fr Informatik, University of Karlsruhe, 1994.
[21]. D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learninginternal representations by error propagation. In Parallel Distributed Processing: Explorations in the Microstructure of Cognition, pages 318–362, Cambridge, Massachusetts, 1986.
[22]. M. Zhang and P. Wong. Genetic programming for medical classification: a program simplification approach. Genetic Programming and Evolvable Machines, 9:229–255, 2008.2009 7th IEEE International Conference on Industrial Informatics (INDIN 2009).

Biography

S.Shoba Rani received B.Tech degree in Computer Science and Engineering from Swami Ramananda Tirtha Institute of Science & Technology, in 2003 and M.Tech degree in Software Engineering from JNTUH, Hyderabad, India in 2008. Pursuing Ph.d from JNTU hyederabad. At present working as Associate Professor in the department of Computer Science and Engineering at Vardhaman College of Engineering, Hyderabad, India. I have totally teaching experience 11 years.

R.Chennakeshavulu received B.E degree in *Electronics and Communication Engineering* from Chaitnya Bharati Institue of Technology(CBIT), in 2001 and M.Tech degree in DSCE from JNTUH, Hyderabad, India in 2009.Pursuing Ph.d from JNTU hyederabad. At present working as Associate Professor in the department of *Electronics and Communication Engineering, JBIET, Moinabad. He have total teaching experience 13 years.*