2008 International Conference on Computer Science and Software Engineering

Improved Simultaneous Perturbation Stochastic Approximation and Its Application in Reinforcement Learning

Xiumei Yue
Department of Electrical and Electronic Engineering
Huangshi Institute of Technology, China
wangyue_120109@163.com

Abstract: **In the optimization problem which only measurements of the objective function are available, it is difficult or impossible to directly obtain the gradient of the objective function. Although the second order simultaneous perturbation stochastic approximation (2SPSA) algorithm solves this problem successfully by efficient gradient approximation that relies on measurements of the objective function, the accuracy of the algorithm depends on the matrix conditioning of the objective function Hessian. In order to eliminate the influence caused by the objective function Hessian, this paper uses nonlinear conjugate gradient method to decide the search direction of the objective function. By synthesizing different nonlinear conjugate gradient methods, it ensures each search direction to be descensive. Besides the search direction improvement, this paper also uses inexact line searches to decide the stepsize of movement. With the descensive search direction and appropriate stepsize, the improved SPSA converges faster than the 2SPSA. Through applying to reinforcement learning, the virtues of the improved SPSA are validated.**

Ⅰ INTRODUCTION

In engineering, physical and social science field, there are many optimization problems which only the measurements of the objective function are available. Because it is difficult or impossible to directly obtain the gradient of the objective function, stochastic approximation algorithms such as Kiefer-Wolfowitz finite difference gradient approximation [1] and simultaneous perturbation stochastic approximation [2-3] were proposed to solve these problems. Contrasted with the finite difference approaches which require a number of function measurements proportional to the dimension of the gradient vector, the SPSA algorithm significantly reduces the number of measurements required in many multivariate problems of practical interest. The latest improved SPSA estimates gradient relying on only one measurement of the objective function [4]. Based on the simultaneous perturbation theory, the SPSA algorithm estimates the gradient of objective function $f(\theta)$ as

$$\hat{g}_k(\hat{\theta}_k)=\frac{f(\hat{\theta}_k+c_k\Delta_k)-f(\hat{\theta}_k-c_k\Delta_k)}{2c_k}\begin{bmatrix}\Delta_{k1}^{-1}\\ \Delta_{k2}^{-1}\\ \vdots\\ \Delta_{kp}^{-1}\end{bmatrix} \quad (1)$$

Where p is dimension number, Δ_k is a p-dimensional perturbation vector, c_k is a gain sequence. Although the measurements are few, the estimation of gradient is efficient. With the step vector α_k, the first order SPSA estimates $\hat{\theta}_k$ of a solution θ^* according to (2) as

$$\hat{\theta}_{k+1}=\hat{\theta}_k-\alpha_k\hat{g}_k(\hat{\theta}_k) \quad (2)$$

In order to accelerate the convergence speed, the 2SPSA algorithm estimates the acceleration properties associated with deterministic algorithms of Newton–Raphson form as

$$\begin{cases}\hat{\theta}_{k+1}=\hat{\theta}_k-\alpha_k\bar{\bar{H}}_k^{-1}\hat{g}_k(\hat{\theta}_k)\\ \bar{\bar{H}}_k=f_k(\bar{H}_k)\\ \bar{H}_k=\dfrac{k}{k+1}\bar{H}_{k-1}+\dfrac{1}{k+1}\hat{H}_k\end{cases} \quad (3)$$

However, at finite iterations, the accuracy of the algorithm depends on the matrix conditioning of the loss function Hessian. The error of 2SPSA algorithm for a loss function with an ill-conditioned Hessian is greater than the one with a well-conditioned Hessian. Besides the accuracy problem, the Newton–Raphson algorithm itself can not ensure each search direction always to be a descensive direction [5]. Then the objective value will not minish even if the approximation of Hessian matrix is accurate. In addition, the step size of movement is also important for the 2SPSA which simulates the Newton–Raphson algorithm. Large or short step size can not ensure fast down in convergence. Regard to the diminishing stepsize of SPSA, it is not generated according to the Wolfe condition of linear search [6]. Then the association which consists of accurate Hessian matrix, descensive search direction and bad stepsize will also cause worse objective value. In order to accelerate convergence property, it is necessary to improve the gradient and step strategy of SPSA.

The remainder of this paper is structured as follows: section □introduces the nonlinear conjugate gradient method in the SPSA. In section □, we analyze the reinforcement learning model of AC drive system. In section □, a simulation is used to validate the feasibility of the proposed improvement in SPSA. At last, through analyzing of experiment results, the conclusion is presented in section □.

□.NONLINEAR CONJUGATE GRADIENT METHOD IN THE SPSA

In numerical analysis field, the conjugate gradient method is the best way to approximate the solution of a problem. The conjugate gradient method satisfies the recurrence

$$x_{k+1}=x_k+\alpha_k d_k \quad (4)$$

Where the stepsize α_k is positive and the directions d_k

978-0-7695-3336-0/08 $25.00 © 2008 IEEE
DOI 10.1109/CSSE.2008.1019

are generated by the rule:

$$\begin{cases} d_{k+1} = -g_{k+1} + \beta_k d_k \\ d_0 = -g_0 \end{cases} \tag{5}$$

Here $g_k = \nabla f(x_k)$ is gradient vector of objective function. There are many different versions of the conjugate gradient method corresponding to different choices of β_k. Well-known conjugate gradient methods include the Fletcher-Reeves method [7], Polak-Ribiere-Polyak method [8-9], Dai-Yuan method [10] and Hestenes-Stiefel method [11]. They are specified by formula (6) respectively.

$$\begin{cases} \beta_k^{FR} = \|g_k\|^2 \Big/ \|g_{k-1}\|^2 & \beta_k^{PRP} = g_k^T y_{k-1} \Big/ \|g_{k-1}\|^2 \\ \beta_k^{DY} = \|g_k\|^2 \Big/ d_{k-1}^T y_{k-1} & \beta_k^{HS} = g_k^T y_{k-1} \Big/ d_{k-1}^T y_{k-1} \end{cases} \tag{6}$$

Where $y_{k-1} = g_k - g_{k-1}$ and $\|\bullet\|$ stands for the Euclidean norm of vectors. When f is quadratic and α_k is chosen to minimize f in the search direction d_k, these choices are all equivalent, but for a general nonlinear function, different choices have quite different convergence properties. It is proved that the global convergence of the FR method for nonconvex functions with the strong Wolfe line search if the parameter $\sigma < 0.5$ [12]. The PRP method with exact line search may cycle without approaching any stationary point, see Powell's counter-example [13]. Although one would be satisfied with its global convergence properties, the FR method sometimes performs much worse than the PRP method in real computations. A similar case happens to the DY method and the HS method. To combine the good numerical performance of the PRP and HS methods and the nice global convergence properties of the FR and DY methods, Dai and Yuan proposed a hybrid conjugate gradient method as [14]

$$\beta_k = \max\{0, \min\{\beta_k^{DY}, \beta_k^{HS}\}\} \tag{7}$$

This hybrid conjugate gradient method shows better convergence property than others since of it ensures each d_k to be a descensive search direction. Then the recurrence of estimating $\hat{\theta}_k$ based on the nonlinear conjugate gradient method is

$$\begin{cases} \hat{\theta}_{k+1} = \hat{\theta}_k + \alpha_k d_k \\ d_{k+1} = -\hat{g}_k(\hat{\theta}_k) + \beta_k d_k \\ \beta_k = \max\{0, \min\{\beta_k^{DY}, \beta_k^{HS}\}\} \end{cases} \tag{8}$$

Through analyzing the above formula (8), we find that the convergence speed of SPSA is not relative to matrix conditioning of the objective function Hessian which exists in 2SPSA. So the convergence speed is decided by $\hat{g}_k$ and α_k. Because Spall has proved the efficiency of simultaneous perturbation in gradient approximation [2], then it necessary to find a effective stepsize strategy which can minish the objective value based on the exact $\hat{g}_k$ and β_k. As standard conjugate gradient method required, α_k should satisfy Wolfe conditions as follow

$$\begin{cases} f(\hat{\theta}_k + \alpha_k d_k) \le f(\hat{\theta}_k) + \delta \alpha_k d_k^T \hat{g}_k \\ \sigma_1 d_k^T \hat{g} \le d_k^T \hat{g}_k(\hat{\theta}_k + \alpha_k d_k) \le -\sigma_2 d_k^T \hat{g}_k \end{cases} \tag{9}$$

Where $0 < \delta < 1$, $\sigma_1 \in (\delta, 1)$ and $\sigma_2 \ge 0$. Because only measurements of the objective function are available, it is impossible to execute line search according to Wolfe conditions. In standard SPSA, α_k is calculated as

$$\alpha_k = a / (A + k + 1)^{\sigma} \tag{10}$$

Spall [2] chooses $\sigma = 0.602$, A as 10 %(or less) of the maximum number of expected/allowed iterations and a as $a/(A+1)^{\sigma}$ times the magnitude of elements in $\hat{g}_0(\hat{\theta}_0)$. With the increase of k, α_k is decreasing. If α_k causes a worse objective value, the optimal solution must stay at $\hat{\theta}_k$ and look for a new α_k according to (10) at the next iteration. Without an appropriate stepsize, the optimal solution will stay at $\hat{\theta}_k$ forever and slower the convergence speed greatly. For instance, as Fig. 1

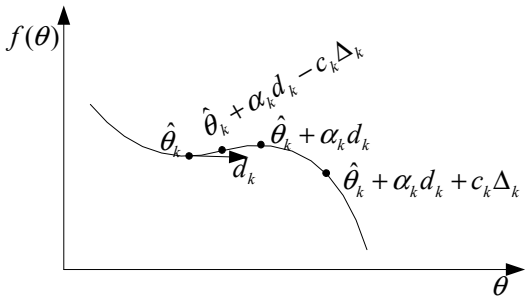

Fig. 1 The search process of SPSA

shows, $f(\hat{\theta}_k + \alpha_k d_k) > f(\hat{\theta}_k)$, then it is necessary to look for another α_k to minish objective function value as standard SPSA required. But we ignore the perturbations which minish the objective function value. In Fig.1, the objective values of four points are that $f(\hat{\theta}_k + \alpha_k d_k + c_k\Delta_k) < f(\hat{\theta}_k) < f(\hat{\theta}_k + \alpha_k d_k - c_k\Delta_k) < f(\hat{\theta}_k + \alpha_k d_k)$. Obviously, the optimal solution is $\hat{\theta}_k + \alpha_k d_k + c_k\Delta_k$. At this time, we can take $\hat{\theta}_{k+1} = \hat{\theta}_k + \alpha_k d_k + c_k\Delta_k$ as the result generated according to formula (8) with the appropriate stepsize $\hat{\alpha}_k = \alpha_k + c_k\Delta_k / d_k$. Then the $\hat{\theta}_k + \alpha_k d_k$ and $\hat{\theta}_k + \alpha_k d_k - c_k\Delta_k$ can be looked as perturbations. The new expressions of perturbations are

$$\begin{cases} \hat{\theta}_k + \alpha_k d_k = (\hat{\theta}_k + \alpha_k d_k + c_k\Delta_k) - c_k\Delta_k \\ \hat{\theta}_k + \alpha_k d_k - c_k\Delta_k = (\hat{\theta}_k + \alpha_k d_k + c_k\Delta_k) - 2c_k\Delta_k \end{cases} \tag{11}$$

With the perturbations, we can estimate the gradient at $\hat{\theta}_{k+1}$ as

$$\hat{g}_k(\hat{\theta}_{k+1}) = \frac{f(\hat{\theta}_k + \alpha_k d_k) - f(\hat{\theta}_k + \alpha_k d_k - c_k\Delta_k)}{c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \Delta_{k2}^{-1} \\ \vdots \\ \Delta_{kp}^{-1} \end{bmatrix} \quad (12)$$

Based on the above analysis, if $f(\hat{\theta}_k + \alpha_k d_k - c_k\Delta_k) = \min\{f(\hat{\theta}_k + \alpha_k d_k), f(\hat{\theta}_k), f(\hat{\theta}_k + \alpha_k d_k - c_k\Delta_k), f(\hat{\theta}_k + \alpha_k d_k + c_k\Delta_k)\}$, then $\hat{\theta}_{k+1} = \hat{\theta}_k + \alpha_k d_k - c_k\Delta_k$ and appropriate stepsize $\hat{\alpha}_k = \alpha_k - c_k\Delta_k / d_k$,the estimation of gradient at $\hat{\theta}_{k+1}$ is

$$\hat{g}_k(\hat{\theta}_{k+1}) = \frac{f(\hat{\theta}_k + \alpha_k d_k) - f(\hat{\theta}_k + \alpha_k d_k + c_k\Delta_k)}{c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \Delta_{k2}^{-1} \\ \vdots \\ \Delta_{kp}^{-1} \end{bmatrix} \quad (13)$$

If $f(\hat{\theta}_k) < \min\{f(\hat{\theta}_k + \alpha_k d_k), f(\hat{\theta}_k + \alpha_k d_k - c_k\Delta_k), f(\hat{\theta}_k + \alpha_k d_k + c_k\Delta_k)\}$, the only way is to keep optimal solution at $\hat{\theta}_k$ and look for a new α_k at next iteration. In fact, the above process of looking for the minimal objective value among $f(\hat{\theta}_k + \alpha_k d_k)$, $f(\hat{\theta}_k + \alpha_k d_k + c_k\Delta_k)$, $f(\hat{\theta}_k)$ and $f(\hat{\theta}_k + \alpha_k d_k - c_k\Delta_k)$ is a line search process in essential. Based on the exact estimation of gradient, it will accelerate the convergence speed of SPSA.

THE IMPROVED SPSA IN RL

In order to validate the feasibility of the proposed algorithm, this paper applies it to the reinforcement learning which designs a controller for asynchronous (AC) motor drive system. Fig. 2 illustrates the theory of reinforcement learning in AC motor drive system.

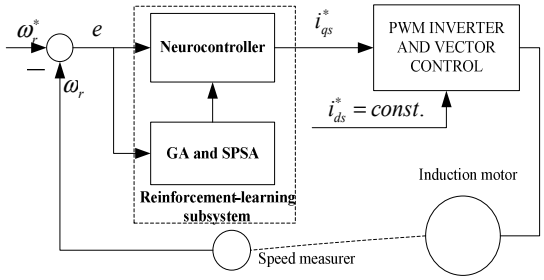

Fig. 2 Theory of neurocontroller designing in AC motor drive system

The dashed square is the reinforcement learning subsystem which consists of genetic algorithm (GA) and SPSA algorithm. Both GA and SPSA are stochastic approximation algorithms. Although GA has good global search ability than SPSA algorithm, it descends slowly at local area. So this paper applies SPSA algorithm to search optimal solution when GA is vibrating at local area. This hybrid algorithm can accelerate the learning speed of reinforcement learning. Obviously, the reinforcement learning is a good test which will validate the fast convergence speed of the improvement SPSA algorithm. For the drive system, the electromagnetic torque T_e and load state equation in the rotor flux oriented scheme at steady state are

$$\begin{cases} T_e = \frac{3}{2}(\frac{p}{2})\frac{L_m}{L_r}\hat{\psi}_r i_{qs} \\ T_e - T_L = (\frac{2}{p})J\frac{dw_r}{dt} \end{cases} \quad (14)$$

Where p is the number of magnetic pole, L_m is magnetizing inductance, L_r is rotor inductance, $\hat{\psi}_r$ is the rotor flux, T_L is the load torque, J is the inertia, w_r is the speed of rotor. The aim of reinforcement learning is to design a speed controller which supplies the reference torque current i_{qs}^*. In this AC drive system, the steady state error is the learning goal. It means that the controller should own the optimal mapping function of w_r and i_{qs} which ensure drive system have the minimal steady state error. Generally speaking, this reinforcement learning is a minimization problem which the objective function is

$$f = \sum_{t_1}^{t_2} |e(t)| \quad (15)$$

Where $e(t)$ is the steady state error, t_1, t_2 is is the starting and ending time of evaluation.

. SIMULATION RESULTS AND ANALYSIS

The simulation is established by simulink kit of MATLAB as Fig. 2. The model parameters are set according to the factual AC motor in the laboratory: L_m=0.1024H, L_r=0.1088H, L_s=0.1063H, R_r=0.531Ω, R_s=0.813Ω, p=2, J=0.02kgm^2, rated power P_n=5.5kw, rated voltage U_n=380V. Although the parameters are exact in simulation, the drive system is unknown to the controller designer. Because most control system is a dynamic system in real world, we choose a recurrent neural network as the neurocontroller. The topology of the initial network shows as Fig. 3.

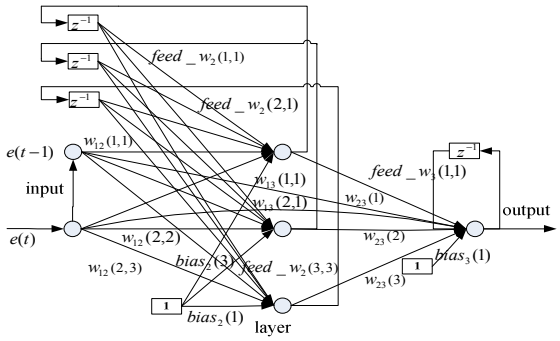

Fig. 3 The initial topology of neurocontroller

In Fig. 3, w_{ij} is the weight matrix which connects up layer i and layer j. $feed\_w_i$ is the weight matrix of the i^{th} layer which feeds back neuron output to other neurons of the same layer. $bias_i$ is bias matrix of the i^{th} layer. In order to take advantage of different activation function of neural network, we also use a new activation function as

$$trans(x) = trans\_w_i \times \begin{bmatrix} \tan sig(x) \\ hard\lim s(x) \\ purelin(x) \\ radbas(x) \end{bmatrix} \quad (16)$$

Where $trans\_w_i$ is weight matrix of activation function, *tansig*(x), *hardlims*(x), *purelin*(x), and *radbas*(x) are four ordinary activation functions.

The initial settings of GA like these: the population size is 40, Crossover probability P_c=0.8, mutation probability P_m=0.01, selection method is roulette wheel selection, max generation N=200. When GA vibrates at local area, the SPSA algorithm begins to search the optimal solution instead of it. The parameters like a , c , A and σ are initialized according to standard SPSA in [2]. The hybrid algorithm of GA and SPSA operates as fig.4.

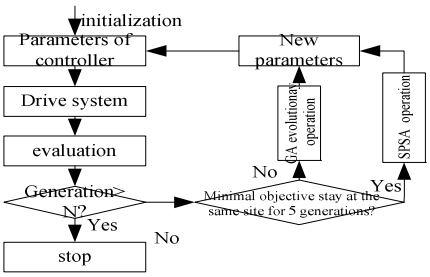

Fig. 4 the operation flow of the hybrid algorithm

In order to validate the improvement of the convergence speed, Fig.5 shows the minimal objective value during the reinforcement learning when learned based on 2SPSA algorithm and the improved SPSA algorithm respectively. As the Fig.5 shows, the improved SPSA algorithm accelerates convergence speed obviously. In Fig. 5, there are some horizontal lines which imply the search stay at the same point during the 2SPSA search process. The two reasons bringing on the above phenomenon are that the search direction is not descensive and the stepsize does not fit in with the search direction. These shortcomings slower the convergence speed of 2SPSA greatly. Compared with the 2SPSA algorithm, the improved SPSA algorithm eliminates the shortcomings and accelerates convergence speed. In Fig. 4, the minimal objective values keep descending during the improved SPSA algorithm search process. Although 2SPSA algorithm begins to search at the 45th generation and the improved SPSA algorithm begins to search at the 55th generation, the convergence speed of the latter is much faster.

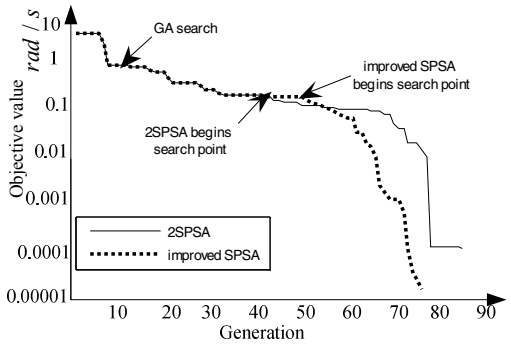

Fig. 5 the minimal objective value during the reinforcement learning

Ⅴ. CONCLUSION

This paper improves two facets of the SPSA algorithm. The first improvement which applies nonlinear conjugate gradient method to SPSA algorithm ensures the search direction to be descensive. The second improvement which executes inexact line search can find an appropriate stepsize corresponding to descensive search direction. Based on the exact gradient approximation, these improvements can accelerate the convergence speed of SPSA. Through applying the improved SPSA algorithm to reinforcement learning of AC drive system, its virtues are validated completely. On the other hand, the simulation results also show that the SPSA is a good method to solve reinforcement learning problem.

REFERENCE

[1]. Spall JC. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions on Automatic Control 1992; 37:332–341.

[2]. Spall JC. Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimization. IEEE Transactions on Aerospace and Electronic Systems, Vol. 34, NO. 3,1998.

[3]. Spall JC. Accelerated Second-Order Stochastic Optimization Using Only Function Measurements. Proceedings of the 36th Conference on Decision & Control San Diego, Califomia USA - December 1997.

[4]. Jamie R. Wieland, Bruce W. Schmeiser. Stochastic Gradient Estimation Using A Single Design Point. Proceedings of the 2006 Winter Simulation Conference. 2006.

[5]. Dai Y H, Yuan Y. Nonlinear Conjugate Gradient Methods. Shanghai, Shanghai Science and Technology Press, 2000.

[6]. WOLFE, P. Convergence Conditions for Ascent Methods II: Some Corrections. SIAM Rev.1971, 13,185–188.

[7]. Fletcher, R. and Reeves, C. Function Minimization by Conjugate Gradients. Comput. J. 1964, 7,149–154.

[8]. Polak, E. and Ribière, G.. Note sur la convergence de directions conjuguées. Rev. Francaise Informat Recherche Operationnelle, 3e Année, 16: 35–43. 1969

[9]. Polyak, B. T. The Conjugate Gradient Method in Extreme Problems. USSR Comp. Math. Math. Phys. 9, 94–112, 1969.

[10]. Hestenes, M. R. and Stiefel, E. L. Methods of Conjugate Gradients for Solving Linear Systems. J. Res. Nat. Bur. Standards 49, 409–436, 1952.

[11]. Dai, Y. H. and Yuan, Y. A Nonlinear Conjugate Gradient Method with A Strong Global Convergence Property. SIAM J. Optim. 10, 177–182, 2000.

[12]. M. Al-Baali, Descent property and global convergence of the Fletcher–Reeves method with inexact line search, IMA J. Numer. Anal. 5 (1985): 121–124.

[13]. M.J.D. Powell, Nonconvex minimization calculations and the conjugate gradient method, Lecture Notes in Math. 1066 (1984): 121–141.

[14]. Y.H. Dai, Y. Yuan, An efficient hybrid conjugate gradient method for unconstrained optimization, Ann. Oper. Res. 103 (2001): 33–47.