Off-line Selection of Learning Rate for Back-Propagation Neural Ntwork using Evolutionary Adaptation

진화 적응성을 이용한 신경망의 학습율 선택

김 흥 범*, 정 성 훈**, 김 탁 곤*, 박 규 호*
Heung Bum Kim*, Sung Hoon Jung**, Tag Gon Kim*, Kyu Ho Park*

ABSTRACT

In training a back-propagation neural network, the learning speed of the network is greatly affected by its learning rate. Most of off-line fashioned learning-rate selection methods, however, are empirical except for some deterministic methods. It is very tedious and difficult to find a good learning rate using the empirical methods. The deterministic methods cannot guarantee the quality of the quality of the learning rate. This paper proposes a new learning-rate selection method. Our off-line fashioned method selects a good learning rate through stochastically searching process using evolutionary programming. The simulation results show that the learning speed achieved by our method is superior to that of deterministic and empirical methods.

요 약

신경망을 학습하는데 있어서, 망의 학습속도는 학습율에 의해 크게 좌우된다. 그러나, 대부분의 정적인 학습율 선택 방법들은 몇몇 결정적인 방법들을 제외하곤 경험적인 방식에 의존해 왔다. 경험적인 방식을 사용하여 좋은 학습율을 찾아내는 것은 매우 지루하고 어려운 일이다. 또한 결정적인 방법들은 학습율의 질을 보장하지는 못한다. 본 논문에서 우리는 새로운 학습율 선택 방법을 제안한다. 우리의 방법은 진화 프로그래밍기법을 사용하여 통계적인 방식으로 접근함으로써 좋은 학습율을 찾을 수 있다. 모의 실험을 통하여 우리의 방식이 경험적인 방식들이나 결정적인 방식보다 우수함을 보였다.

I. Introduction

The training speed of the back-propagation neural network greatly depends on the value of the learning rate, η [2]. Most of off-line fashioned learning-rate selection method have frequently been empirical [1, 6]. It is very tedious and difficult to find a good learning rate using the empirical method. Eaton has suggested a learning rate selection method [2] that deterministically computes the learning rate using the number of training samples. This method may select a bad learning rate because its selection equation depends only on the number of sample patterns. Thus, this method cannot guarantee the quality of the learning rate. In this paper, we propose an off-line selection method

*Department of Electrical Engineering, KAIST
한국과학기술원 전기 및 전자공학과
** School of Information and Computer Engineering, Hansung Univ.
한성대학교 정보전산학부

that can pick out a good learning rate using the evolutionary programming technique.

Our method mostly outperforms the Eaton method because our method searches a good learning rate through stochastically searching process. The learning rates in our method are evaluated and generated iteratively by the evolutionary programming technique. As the process goes on, the learning rates in evolutionary programming become better and better, and the best learning rate is selected finally. This method is not an application specific method because it selects the best learning rate within iteratively evaluated learning rates. Thus this method can be widely employed in any type of problems. It was shown through simulation that the learning speed with the learning rate selected by our method is superior to those by the empirical method and the deterministic method [2].

Ⅱ. Modification of Evolutionary Programming

Evolutionary programming [3] is an algorithm that simulates natural evolution. It is a type of the systematic multi-agent stochastic search methodology. This algorithm consists of four main processes: evaluating fitness, competing with members, selecting parents, and generating offspring. In order to find a good solution for a problem, these processes are repeated. To operate the evolutionary programming, three important parameters-population, objective function, and perturbation factor-must be determined [3, 6]. The population is a set of solutions. The objective function is to measure the fitness of each solution. The perturbation factor is used to create new generations from parents.

In our application, the learning rates for a neural network are mapped to the solutions within the population. We defined the objective function as the inverse value of the total sum squared error(TSSE) of the network. The perturbation factor is defined in terms of the objective function and other elements of the network. The objective function and the perturbation factor are defined as follows.

Definition 1: Objective Function

Lex X_i be a solution (learning rate) of population P. Then an *objective function* is defined as:

$$F(X_i) = \frac{1}{FSSE} = \frac{1}{\sum^{R_{p-1}} \sum^{K_{k-1}} (d_{pk} - o_{pk}^{X_i})^2}$$

where d_{pk} and $o_{pk}^{X_i}$ are the desired output and the actual output of a pattern p in an output neuron k on a vector X_i, respectively; R and K are the number of patterns and output neurons, respectively.

Definition 2: Perturbation Factor (Standard Deviation)

Let X_i be a solution of the population. Then a *perturbation factor* is defined as:

$$\sigma_{X_i} = \frac{1}{\sqrt{F(X_i)} * R * K}$$

where R and K are the same as in the above definition.

Using the definitions 1 and 2, we devised an algorithm for off-line selection of learning rate as follows.

Algorithm 1: Off-line Selection of Learning Rate( )

// $2m$: the number of elements in population, (m parents $+m$ offspring) //
// X_i: the ith solution of population P //
// $F(X_i)$: the objective function of X_i //
// J_i: the fitness score of X_i, $J_i \leftarrow F(X_i)$ //
// W_i: the winning number of X_i in competition //
// σ_i: the perturbation factor of X_i, $\sigma_i \propto \frac{1}{J_i}$ //
// g: the number of generation //
// S_k: the scores and solutions of parents in kth generation //
1. initialize population $P = [X_0, X_1, \ldots, X_{2m-1}]$
2. **for** $k = 0$ to $g-1$
3.   **for** $i = 0$ to $2m-1$
4.   assign a score J_i to each solution X_i

end for
for $i=0$ to $2m-1$
    compete X_i randomly with the others X_j ($\neq X_i$) and count winning number W_i
end for
for $i=0$ to $m-1$
    select m parents solutions in population according to the number order of W_i
    generate offspring by modifying X_i in parents with random peturbation factor σ_{X_i}
end for
store the scores and solutions of parents to S_k
end for
obtain the appropriate solution from S_0 to S_{g-1}

Ⅲ. Simulation

In order to verify our algorithm, we employ to two examples, Exclusive OR and 10-bit modulus problem. For the first experiments, the variables for the off-line selection algorithm are set up as follows: initial solutions in a population are randomly picked out within 0.04 and 0.4. The number of population is 100. The number of competition is 10. This algorithm spends approximately 2.8 sec of the CPU time of sparc-2 workstation on determining the most excellent learning rate.

We employed a benchmark problem[4], Exclusive OR 2-inputs pattern classification, and a three-layered (2-2-1) neural network structure[5]. Three methods (ours, Eaton, and the empirical) were examined 10 times respectively. Figure 1 shows the result of the simulation with the average value of 10 runs.

In the second experiment, we change the number of population and competition to 50 and 5, respectively. CPU time in ALPHA workstation needs 74.6 sec for selecting a learning rate with our algorithm. The 10-bit modulus problem, as a classification problem, is to categorize a 10-bit binary input to a decimal modulus output. Thus, this problem takes 10-bit binary inputs and generates 10 modulus outputs. For

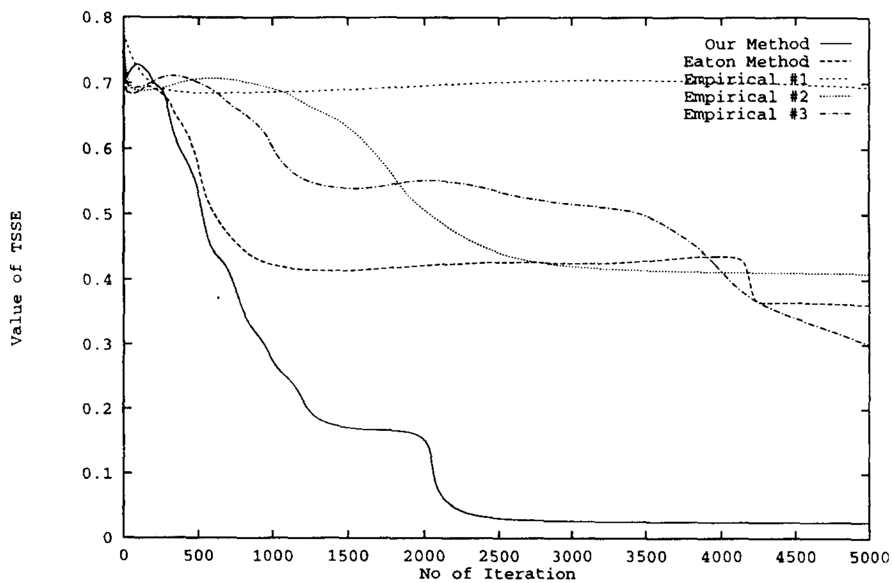

Fig. 1 TSSE *vs.* No of iteration in 2-2-1 network using Exclusive OR 2-inputs pattern

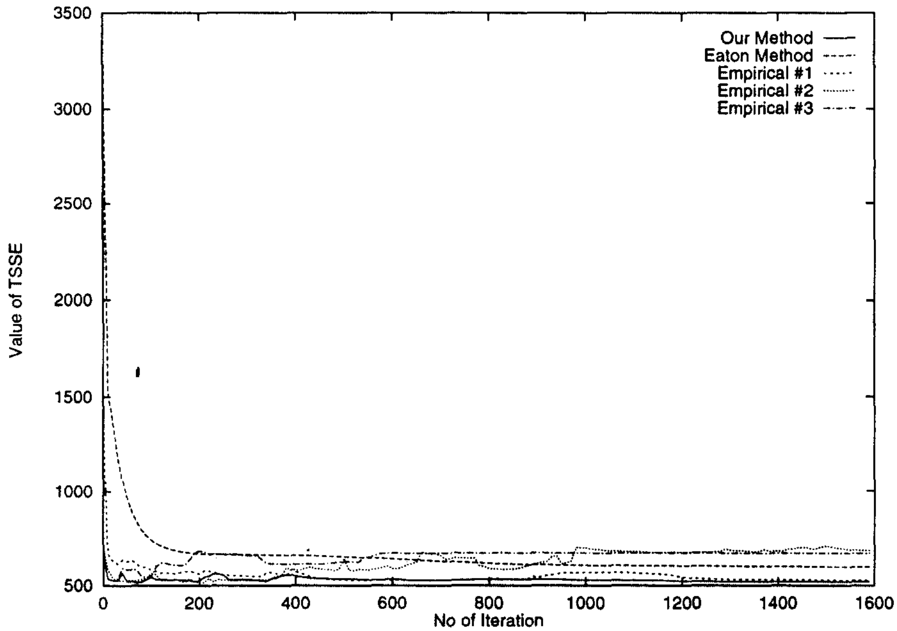

Fig. 2 TSSE *vs.* No of iteration in 10-10-10 network using 10-bit modulus problem

example, if a 10-bit binary 0000001011 is provided, then only the second output becomes one and all other output become zeros, 0100000000. This is because the binary number corresponds to modulus 1. The number of sample patterns is 1024. The patterns were provided into 10-10-10 network, and the result of 10 averaging runs exists in Figure 2.

In Figure 1 and 2, the learning rate of Eaton method [2] is given as $\frac{1.5}{\sqrt{N^{2_1}+N^{2_1}+\cdots+N^{2_m}}}$ (N=the number of patterns, m=the number of types). The result of Eaton method has the learning rate $m=1$ and the momentum coefficient $a=0.9$. For the empirical method, we selected three learning rates (Empirical #1, #2, and #3, i. e., $\eta=0.04$, 0.22, and 0.4 respectively). The TSSE of our method is mostly lower than that of Eaton method and empirical #1, #2, and #3 during the training process. Therefore, our selection method outperforms other selection methods in training speed stochastically.

Our method can select good learning rate than other methods because evolutionary programming generates better and better solutions from parents through evolutionary adaptation. In two experiments our method is superior to Eaton method. In the view point of training speed, the result of Exclusive OR 2bit problem shows stable property lower TSSE for all methods. In 10-bit modulus problem Standard Method #2 and #3 become unstable according to increasing iteration. But our method represents more stable feature.

It is hard problem to select learning rate for training neural network because the rate is changed to the structures of neural network, the kind of sample patterns, and application domains. Specially, the learning rate of empirical selection can make neural network unstable during training. By two tests, off-line algorithm can provide better learning rate for each applicable domain, and training fashion demonstrates more stable than other methods.

Ⅳ. Conclusion

A new method for off-line selection of learning rate is proposed. Our method can select a better learning rate than other methods because evolutionary programming generates better and better solutions from parents through evolutionary adaptation. Using Exclusive OR 2-inputs and 10-bit modulus problem, we simulated a neural network with 2-2-1 and 10-10-10 structure respectively. In terms of the training speed and stability, the simulation result shows that our selection method is superior to Eaton selection[2] and empirical selection methods.

References

1. Rumelhart, D. E., McClelland, J. L., and the PDP Group 1986:'Parallel Distributed Processing Vol. 1'. Cambridge, MA, MIT Press.
2. H. A. C.~Eaton and T. L. Olivier:'Learning Coefficient Dependence of Training Set Size'. Neural Networks, vol. 5, pp. 283-288, 1992.
3. D.B. Fogel:'An Introduction to Simulated Evolutionary Optimization'. IEEE Tr. on NN, vol. 5, no. 1, pp. 3-14, Jan., 1994.
4. P.R. Nachtsheim:'A First Order Adaptive Learning Rate Algorithm for Back Propagation Networks'. IEEE International Conference of Neural Networks, pp. 257-262, 1994.
5. M.K. Weir:'A Method for Self-Determination of Adaptive Learning Rates in Back Propagation'. Neural Networks, vol. 4, pp. 371-379, 1991.
6. Heung Bum Kim, Sung Hoon Jung, Tae Gon Kim, and Kyu Ho Park:'A Fast Learning Method for Back-Propagation Neural Network by Evolutionary Adaptation of Learning Rates'. Journal of Neurocomputing. "accepted for publication".

김 홍 범(Heung Bum Kim) 정회원

1979년 : 한국항공대학교 항공전자공학과 졸업(공학사)
1979년~현재 : 국방과학연구소 선임연구원
1989년 : 한국과학기술원 전기 및 전자공학과 졸업(공학석사)
1993년~현재 : 한국과학기술원 전기 및 전자공학과 박사과정
※주관심분야 : 뉴로컴퓨팅, 모델링/시뮬레이션, 분산 문제 해결

정 성 훈(Sung Hoon Jung) p.51 참조

김 탁 곤(Tag Gon Kim) 정회원

1975년 : 부산대학교 전자공학과 졸업(공학사)
1980년 : 경북대학교 전자공학과 졸업(공학석사)
1988년 : 아리조나대학교 전기 및 전산공학과 졸업(공학박사)
1980년~1983년 : 부산수산대학교 전자통신공학과 조교수
1989년~1991년 : 캔사스대학교 전기 및 컴퓨터공학과 조교수
1991년~1993년 : 한국과학기술원 전기 및 전자공학과 조교수
1993년~현재 : 한국과학기술원 전기 및 전자공학과 부교수
※주관심분야 : 모델링 이론, 병렬/지능형 시뮬레이션, 컴퓨터 시스템 해석

박 규 호(Kyu Ho Park) 정회원

1973년 : 서울대학교 전자공학과 졸업(공학사)
1975년 : 한국과학기술원 전기 및 전자공학과 졸업(공학석사)
1983년 : 프랑스 파리 11대학 졸업(공학박사)
1975년~1978년 : 동양정밀 개발과장
1983년~1988년 : 한국과학기술원 전기 및 전자공학과 조교수
1988년~1992년 : 한국과학기술원 전기 및 전자공학과 부교수
1992년~현재 : 한국과학기술원 전기 및 전자공학과 교수
※주관심분야 : 병렬처리, 컴퓨터구조, 컴퓨터 비젼