UDC 519.7:004.8

K.A.KUZNETSOV, Ph.D., Dnepropetrovsk National University
O.I.PEREDERIEIEVA, National Mining University

A DETERMINISTIC ANNEALING ALGORITHM FOR NEURAL NET LEARNING

В статті проведений порівняльний аналіз роботи алгоритму зворотного розповсюдження помилки та алгоритмів імітації відпалу для задач навчання нейронних мереж. Запропоновані адаптивні схеми налаштування параметрів алгоритмів детермінованого відпалу та проведено експериментальне дослідження їх впливу на якість отримуваного розв‘язку.

This article compares backpropagation and simulated annealing algorithms of neural net learning. Adaptive schemes of the deterministic annealing parameters adjustment were proposed and experimental research of their influence on solution quality was conducted.

Introduction. Simulated neural networks are nowhere near the grand goal of providing a functional equivalent of human brain. But the simple structures already demonstrate very powerful capabilities of problem-solving through self-adaptive computer learning.

In this work we are focusing on one of the most widespread neural net classes – the feed-forward neural network (FNN). For any transfer functions and structure topologies FNN is defined as mapping of inputs X and synoptic weights W into outputs O: $O=\phi(X,W)$. Supervised learning consists of finding particular mapping ϕ that not only approximates input patterns correctly but has the property of generalization for test patterns as well. The learning process is performed with fixed topology and activation functions only through synoptic weights adjustment. In other words for a given set of input images $X=\{(I_1,D_1),\dots(I_m,D_m)\}$ learning process represents a problem of minimization of the net error function.

$$\min_{W} E(W)=\min_{W}\sum_{i=1}^{m}\varepsilon\left(W,I_i,D_i\right) \qquad (1)$$

One popular (but not unique) type of error function is the squared-error function: $\varepsilon(W,I_i,D_i)=(\phi(I_i,W)-D_i)^2$. The quality of a learned network is estimated by its error within a given set of training patterns and/or by the error for the test patterns.

It means that supervised neural network learning can be considered a non-linear optimization problem in the space of the synoptic weights. The problem (1) is NP-hard in general case and therefore chances for finding global optima are few especially for the large multi-layered networks. Thus the development of the new learning algorithms and the adaptation of the available meta-heuristics to the network learning are the key problems.

In our work we are trying to fill some noticeable gaps in the knowledge of deterministic annealing techniques application to the FNN learning problem.

1. The Drawbacks of the Gradient-Based Learning Algorithms. The most commonly used *FNN* learning algorithm is the method of backpropagation error (*BP*). BP is gradient descent method in the synoptic weight space that minimizes deviation of the network outputs from desired ones. The detailed description of the *BP* method is depicted in numerical sources ([1] for example) and omitted here. The *BP* method has many successful applications but in many cases there are serious problems with it.

The first problem lays in the necessity of the gradient computation resulting in a differentiability of the error function. A possible conclusion is that *BP* algorithm doesn’t work within networks with non-differential optimality criteria and with discontinuous transfer functions.

The second problem is in the significant difficulties that run into gradient-based search schemes on the extremely rigged or near-plateau landscapes of the error function. Many authors [2] pointed out the fact that surface of the error function had many extremes. Obviously any methods that don't provide ways of escaping local optima traps will have difficulties finding near-optimal solution.

We may highlight that the second problem is aggravated by the *reduction* of the network size. The use of the small (by size) networks has wide range of the apparent advantages. First, such networks are easily designed and work faster both in software and hardware realizations. Second, they have better generalization possibilities as they don’t try to adapt synoptic weights to the input patterns with the ultimate accuracy. Third, they have less local optima. But it returns in extremely rugged surface of the error function landscape and decrease in possibility of finding global optima by the *BP* algorithm from the random starting point. This phenomenon explains the fact that gradient-based schemes commonly find near optimal solutions for the large networks other than for the small ones.

The remedies for the error function local optima in the context of the gradient scheme usually lay in the design of the multi-start algorithms with adaptive choice of the starting point [3]. Random initialization of the synoptic weights or initialization by the algorithm of Nguen-Widrow [4] is not appropriate if the optimal weights are large. Many modifications of the *BP* algorithm are known – conjugate gradient, quasi-Newton, Levenberg-Marquardt etc. The above methods partially tackle the problems mentioned but still have many limitations.

Many distinctly different algorithms of neural network design and learning were created in attempt to overcome the BP method limitations. Much attention was paid to the application of the evolutionary algorithms to the network design and learning. A comprehensive overview of the research can be found in [5]. The approaches are very promising in terms of the quality of the obtained solution but computational time for the realization is often exceeds acceptable time limit.

We designed a new algorithm for neural network learning that allows obtaining near-optimal solutions fast, making it suitable for online applications.

2. The simulated annealing algorithms. We consider simulated annealing method (SA) as a basis for our version of network learning algorithm. SA was first presented as optimization technology in [6] for computer modeling equilibrium in the statistical physics (based on Monte Carlo techniques). Today, this algorithm is popular in practical applications because of its simplicity, flexibility and efficiency, as well as among theorists for the possibility to analytical examination of its properties and proof of asymptotic convergence.

Simulated annealing algorithm belongs to a class of threshold local search algorithms. Scheme of basic algorithm (also called Metropolis algorithm) can be represented as follows

Algorithm 1. Metropolis algorithm

```
1. select initial state (value of network weights and biases) S
2. select temperature value T>0
3. repeat
    (a) select new state $S'$ from the neighborhood N(S)
    (b) $\Delta E = E(S') - E(S)$, where $E(S)$ is the energy of state S
    (c) if $\Delta E < 0$ accept new state $S \leftarrow S'$
    (d) else if $e^{(-\Delta E/T)} < rand(0,1)$ (2) accept new state $S \leftarrow S'$
    (e)     else reject new state
4. until stop criteria
```

Each new system state is a stochastic perturbation of current one. We will define this perturbation for the space of the synoptic weights as a zero-mean standard deviation. The transition to a new state depends on the energy difference of the current state and the perturbed one. Algorithm 1 allows transitions in any state that reduces the system energy or satisfies stochastic condition (3d). Algorithm stops if it couldn't move to the new state during certain amount of attempts. It means the quasi-equilibrium is achieved. After reaching equilibrium the temperature value can be updated according to a cooling schedule.

Condition (3d) consists of random sampling and exponentiation and therefore takes a significant part of computational cost especially in the low energy states. One of the ways of Metropolis algorithm improvement is to replace this condition with simpler one without sacrifice of solution quality. The idea of such substitution belongs to Creutz [7], and the algorithm is known as microcanonical Monte Carlo simulation method or demon algorithm. In the original form demon algorithm was not aimed at obtaining low energy states, and wasn't directly used for optimization.

Algorithm 2. Creutz's demon algorithm

```
1. select initial state S
2. select demon energy D>0
3. repeat
    (a) select new state $S'$ from the neighborhood N(S)
    (b) $\Delta E = E(S') - E(S)$
    (c) if $\Delta E \le D$ accept new state and renew demon energy
        $S \leftarrow S'$, $D \leftarrow D - \Delta E$
        else reject new state
4. until stop criteria
```

Generating a new state (3a) is similar to the algorithm 1. The transition to a new state occurs if this state reduces system energy. This lost energy accumulates in artificial variable called demon. An increase of the system energy is permitted only if the demon can give the system necessary energy lost in this case. Obviously, the value $E(S) + D = C$ is a constant for any state of obtained Markov chain.

The acceptance function (3c) of the algorithm 2 is deterministic and simpler to calculate than the same one of the Metropolis algorithm. Exponentiation and generation of a random number are replaced with comparing and subtracting. The sequence of the demon algorithm states is stochastic, but all its randomness arises through the generating function (3a).

3. Demon algorithms and optimization. Any optimization problem can be interpreted as minimization of energy function (fitness function) in the acceptable states. The modification of demon algorithm for the system transformation from initial state into low-energy one, as it is required by optimization was proposed in [8]. These methods are based on different strategies of the demon energy reducing:

– "annealing" (reduction) of demon value, similar to a temperature decrease in the simulation annealing method [6];

– setting low enough upper threshold for demon value, which indirectly reduces system energy.

Here are the algorithms that implement the proposed schemes:

Algorithm 3. Bounded demon algorithm

```
1. select initial state S
2. select initial demon energy $D = D_0 > 0$
3. repeat
      select new state $S'$
      $\Delta E = E(S') - E(S)$
      if $\Delta E \le D$ accept new state and renew demon energy
      $S \leftarrow S'$, $D \leftarrow D - \Delta E$
      else reject new state
      if $D > D_0$,   $D \leftarrow D_0$ - truncation of upper value of demon
4. until stop criteria
```

Algorithm 4. Annealed demon algorithm

```
1. select initial state S
2. select initial demon energy $D = D_0 > 0$
3. repeat
      select new state $S'$
      $\Delta E = E(S') - E(S)$
      if $\Delta E \le D$ accept new state and renew demon energy
      $S \leftarrow S'$, $D \leftarrow D - \Delta E$
      else reject new state
      if quasi-equilibrium is achieved
      $D = \alpha * D$ - reduce demon value
4. until stop criteria
```

Each of these methods can be improved [8] by including random standard deviation to the value of demon energy. These algorithms will behave similarly to deterministic algorithms 3 and 4. However, we would like to avoid such varying

increases of the computational complexity of the methods. Therefore, in this work we confined ourselves to consideration of deterministic modifications of simulated annealing algorithms.

The method of threshold accepting (TA) also belongs to the class of deterministic simulated annealing algorithms. This method [9] can be stated as:

Algorithm 5. Threshold accepting

```
1. select initial state S
2. select initial threshold T
3. repeat
      select new state $S'$
      $\Delta E = E(S') - E(S)$
      if $\Delta E \le T$ accept new state $S \leftarrow S'$
      else reject new state
      if quasi-equilibrium is achieved, reduce T according to cooling
      schedule
4. until stop criteria
```

TA algorithm is similar to both the annealed demon algorithm and the bounded demon algorithm [8]. Significant differences are as follows:

- threshold does not absolve or absorb energy unlike demon energy
- increase of upper limit of energy on each step is fixed
- unlimited energy increase is possible and it allows to escape local minima of any depth
- original work [9] presented only linear scheme of energy reduce.

In algorithm 3, value of the upper boundary of demon can be set higher than the threshold value of TA, as well as average value of demon energy is usually significantly less than the initial values.

4. Adjustment parameters of the algorithms. Demon algorithms were usually applied with simple cooling schedules and empirically selected parameters. In this paper we attempt to adapt recent results, related to speed up convergence of simulated annealing algorithms to those computing schemes.

4.1. The choice of initial threshold. Certain amount of random neural networks is generated, errors of each network on learning data set are determined and then standard deviation σ of these errors is calculated. The initial threshold (initial value of demon energy), according to [10], is selected as follows: $t_0 \ge \sigma$

In our paper we set $t_0 = \sigma$.

4.2. Cooling schedule. There is rich variety of cooling schedules in the theory of annealing algorithms. In this work we consider the most used ones – fixed and adaptive. Fixed cooling schedule does not depend on the state of the Markov chain and usually looks like:

$$t_k = t_0\cdot\alpha^k,$$

where α is some constant ($0 < \alpha < 1$) that is usually selected within [0.90 , 0.99]. In our paper, we used this cooling schedule in the annealed demon algorithm and threshold accepting ($\alpha = 0.95$).

Adaptive cooling schedule proposed in [11] can be given as:

$$t_k = t_{k-1}\cdot\left(1+\frac{t_{k-1}\ln(1+\delta)}{3\sigma_{k-1}}\right)^{-1}$$

where δ (the distance parameter) is a small positive constant. In this work, adaptive cooling schedule is used in annealing demon algorithm with distance parameter $\delta = 0.085$.

4.3. The stop criterion. The stop criterion depends on the specific implementation of the algorithm and the current task. We use the stop criterion proposed in [12]:

$$\left(\frac{\sigma_f^2}{t_f\left|\mu_0-\mu_f\right|}\right)<\theta,$$

where θ (the stop parameter) is a small positive constant. We set $\theta = 0.00001$ according to [12]. Moreover, the algorithm stops after a certain amount of iterations (epochs). This additional stop criterion is implemented to limit the maximum working time of the algorithm.

5. Case Study – a Breast Cancer Data Set. For comparison purposes of the discussed methods we used data of the breast cancer diagnostics obtained in [13]. This data set is permanently stored into the repository of the *UCI Machine Learning Group* http://mlearn.ics.uci.edu/databases/ as "*Wisconsin breast cancer database*".

Every sample consists of 9 attributes and the class attribute (0 for benign, 1 for malignant). All attributes are in the domain [10]. Class distribution is benign: 458 (65.5%) and malignant: 241 (34.5%)

We use the network with 8 neurons on the hidden layer and the sole output. All transfer functions were hyperbolic tangents.

All algorithms were implemented in *MATLAB 7.0.* Data file was divided into two parts: 60% for learning purposes and 40% for tests. As the network error function we used squared-error (*Matlab* function *mse*). Then using the learned network we defined error rate for the test patterns. We measured the learning time as well. Our results are shown in Table 1.

Table 1

Test results

Algorithm		**Backpropagation (*traingd*)**	**Annealed Demon (AD)**	**Threshold Accepting (TA)**
Squared Error (for learning patterns)	Min	0.0067	0.0053	0.0044
	Max	0.0113	0.0242	0.0125
	Average	0.0091	0.0114	0.0076
Error, %(for test patterns)	Min	3.1128	3.1128	2.1713
	Max	4.2802	4.6693	3.5920
	Average	3.6316	3.8132	2.5853
Leaning Time, sec	Min	120.6090	103.9540	127.3280
	Max	140.1870	135.8290	136.0630
	Average	129.6770	128.2127	131.3604

The comparative performances of the algorithms are shown in Figure 1. The graph corresponds to the leaning process with best test performance. It is notable that the learning squared errors weren't minimal for those cases. We restricted the number of epochs in such a way that the learning periods were comparable. These numbers of epochs are 35000 for *BP* method, 4650 for TA and 5000 for AD.

The conducted empirical studies indicate that the *TA* algorithm exceeds competitors both in the quality and in the time for the solution obtaining. We observed this fact in the learning phase as well as in the test.

Conclusion. In our work we performed comparison of the backpropagation algorithm to some deterministic annealing techniques for the *FNN* learning. The empirical studies shown that deterministic annealing algorithms may successfully compete with *BP*. Threshold accepting with accurate parameter adjustment can escape from local optima of any depth and therefore outperforms *BP* under comparable working time. Furthermore, in the quasi-equilibrium state *TA* method is close to local search resulting in better performance in the optima neighborhood. However, these benefits essentially depend on parameter values. We introduced some techniques for adaptive parameter adjustment that improve the learning speed and quality.

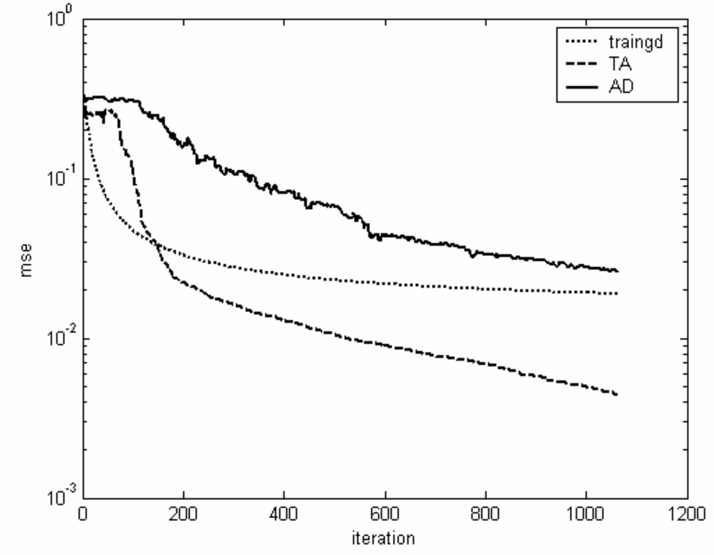

Fig. 1. Comparison of best result obtained by BP, AD and TA

References: 1. *Haykin S.* Neural networks: a comprehensive foundations.-McMillan.-1994. **2.** *Shang Yi, Wah Benjamin W.* Global optimization for neural network training. - Coordinated science laboratory, University of Illinois at Urbana-Champaign, June 24, 1996. **3.** *Duch W., Korczak J.*, Optimization and global minimization methods suitable for neural networks, department of computer methods, Nicholas Copernicus University, Poland, Laboratoire des sciences de l`Image, de l`Informatique et de la Télédétection, CNRS, Université Louis Pasteur, France, 1998. **4.** *Nguyen D., Widrow B.*, Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. Proceedings of the International Joint Conference on Neural Networks, 1990.-№3.-:21-26. **5.** *Whitley D., Kauth J.* GENITOR: A different genetic algorithm, Proc. Rocky Mtn. Conf. on AI, 118-130, 1988. **6.** *Kirkpatrick S., Gelatt C., Vecchi M.*, Optimization by simulated annealing, Science, vol. 220, pp. 671-680, 1983. **7.** *Creutz M.*, Microcanonical Monte Carlo simulation, Physical Review Letters, vol. 50, no 19, pp. 1411-1414, 1983. **8.** *Wood I A., Downs T.*, Demon algorithms and their application to optimization problems. In Proceedings International Joint Conference on Neural Networks 2, Anchorage, Alaska, USA, pp. 1661-1666, 1998. **9.** *Dueck G., Scheuer T.*, Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing, Journal of Computational Physics, vol.90, pp. 161-175, 1990. **10** *White S.R.*, Concepts of scale in simulated annealing, Proc. IEEE ICCD, Port Chester, NY, 646-651, 1984. **11.** *Aarts E. H. L., van Laarhoven P. J. M.*, A new polynomial-time cooling schedule, Proc IEEE ICCAD-85, Santa Clara, CA, 206-208, 1985. **12.** *Otten R.H.J.M., van Ginneken L.P.P.P.*, Annealing applied to floorplan design in a layout compiler, Proc Automation `86, Houston, TX, 185-228, 1986. **13.** *Mangasarian O.L., Setiono R., Wolberg W.H.* Pattern recognition via linear programming: Theory and application to medical diagnosis.- Large Scale Numerical Optimization.- SIAM Publications.- p. 22-30, 1990

Поступила в редколлегию 19.02.08

UDC 681.518

C. RUSS, Institute of Applied Informatics, Alpen-Adria University Klagenfurt, Austria, cruss@uni-klu.ac.at

SPONTANEOUS DIFFUSION OF INFORMATION IN ONLINE SOCIAL NETWORKS

Онлайнові соціальні мережі (ОСМ) є новими типами веб-сервісів, які пропонують онлайновим суспільствам середовище для гуртування та віртуального спілкування. Як наслідок, такі віртуальні мережі соціальних зв‘язків мають високий потенціал для впливового прийняття рішень та розповсюдження інформації «з вуст в уста», але, з іншого боку, вони також можуть розповсюджувати чутки, плітки та некоректну інформацію. Потенціал цих мереж також розпізнається сервіс-провайдерами, маркетологами та виробниками товарів. Вони усі бажають використовувати ці існуючі комунікаційні канали для розповсюдження реклами продуктів безпосередньо користувачам. Але не усі такі спроби є успішними. Ця робота робить спробу пояснити, чому ОСМ є добрим середовищем для спонтанного розповсюдження інформації та які етапи повинні бути виконані для досягнення оптимального рівня розповсюдження для одного елемента інформації. Ми починаємо з розгляду моделі гіперциклів Гартнера, яка пояснює надмірний ентузіазм при впровадженні нових технологій. Далі ми вводимо концепцію «соціального забруднення» та інфекційного розповсюдження інформації. Базова ідея нашого підходу полягає в тому, що онлайнові індивідуали прихильні до колективної поведінки, якщо вони віртуальну поведінку та дії інших. Цей принцип «спрямованості на інших» може генерувати ланцюгову реакцію інфекційних імітацій які інколи можуть розповсюджуватись неконтрольовано через соціальні мережі, подібно до епідемії.

Online Social Networks (OSN) are new types of web services which provide online communities an environment to gather and meet virtually. The online users are connected to each other via links of trust and utilize the features of the OSN to interact and communicate in an easy socio-technical way. Hence these virtual networks of social relationships have a high potential for influential decision-making and the word of mouth spread of information, but also for spreading fads, rumors, and erroneous information. The power of these new forms of social networks is also recognized by service providers, marketers and vendors of consumer goods. They would all like to (mis)use these existing communication channels to spread product placements, advertising and promotions directly to the connected users. However, just like the old economy businesses, not all attempted marketing initiatives are successful. Most of them fail or do not reach the desired audience. This paper tries to explain why OSN are a good environment for spontaneous diffusion of information and what phases of development need to be accomplished to reach the optimal spreading rate for one piece information. Therefore, we start with a look at the "Hype Cycle" model of Gartner to explain over-enthusiasm for new technology adoptions. Next we introduce the concept of "social contagion" and the infections spread of information. After a short introduction of OSN, we try to illustrate the phases of a social online contagion development process which can lead to spontaneous and uncontrolled diffusion of information, messages or ideas. The core statement of our approach is that online individuals tend to behave collectively if they observe the virtual behaviors and actions of others. This principle of "other-directedness" can generate a chain reaction of infectious imitation which can sometimes spread uncontrolled through the interconnected social network like an epidemic. This helps to explain why some online information waves can grow extraordinarily high and others fall.