AFIT/GE/ENG/16-..

SIMULATED QUANTUM ANISOTROPIC ANNEALING APPLIED TO ARTIFICIAL NEURAL NETWORK WEIGHT SELECTION

THESIS

Justin Fletcher
First Lieutenant, USAF

AFIT/GE/ENG/16-..

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

AFIT/GE/ENG/16-..

SIMULATED QUANTUM ANISOTROPIC ANNEALING APPLIED TO ARTIFICIAL NEURAL NETWORK WEIGHT SELECTION

THESIS

Presented to the Faculty of the Electrical and Computer Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Computer Science

Justin Fletcher, B.S. CEC

First Lieutenant, USAF

June, 2016

AFIT/GE/ENG/16-..

SIMULATED QUANTUM ANISOTROPIC ANNEALING APPLIED TO ARTIFICIAL NEURAL NETWORK WEIGHT SELECTION

Justin Fletcher, B.S. CEC

First Lieutenant, USAF

Approved:

Dr. Michael J. Mendenhall Thesis Advisor	Date
Dr. Gilbert L. Peterson Committee Member	Date
Capt. Charlton D. Lewis Committee Member	Date

Preface

Justin Fletcher

Table of Contents

Page

List of Figures

Figure Page

List of Tables

Table Page

List of Symbols

Symbol Page

List of Abbreviations

Abbreviation Page

ANN Artificial Neural Network ix

ANN

AFIT/GE/ENG/16-..

Abstract

SIMULATED QUANTUM ANISOTROPIC ANNEALING APPLIED TO ARTIFICIAL NEURAL NETWORK WEIGHT SELECTION

I. Introduction

In chapter three the traversal a error manifold in the problem configuration space is discussed at length. Several traversal methodologies are proposed and evaluated.

II. Background

This chapter serves as a comprehensive review of the physical and computational concepts material to the topic of this thesis. A broad overview of artificial neural networks and the application and history thereof is presented. Next, simulated annealing is described, along with a summary of some related works and a description of the physical inspiration for the algorithm. The chapter concludes with a very brief overview of the quantum mechanics, with emphasis placed on those concepts which will be employed throughout this thesis document. Finally, the notation and terminology conventions adopted in this document are established.

2.1 Artificial Neural Networks

It has long been recognized (source maybe?) that the capacity of biological information processing systems to flexibly and quickly process large quantities of data greatly exceeds that of sequential computing machinery. This information processing capability arises from the complex, nonlinear, parallel nature of biological information processors (Source? Hayken?). The family of models designed to replicate this powerful information processing architecture are collectively called artificial neural networks. In the most general sense, artificial neural networks are parallel distributed information processors (1) comprising many simple processing elements. Networks store information about experienced stimuli and can make that information available. In such a network, interneuron connection strengths are used to encode information, and are modified via a learning strategy. Artificial neural networks are characterized by three features: a network topology or architecture, an activation function, and a learning strategy. Each will be discussed in the following sections. I will also review the history of artificial neural networks, and the biological inspiration for the computational model.

2.1.1 Biological Inspiration.

Fig[an image of a neuron, mapped to a schematic of a neuron, mapped to a processing element]

Integration of magnitude-encoded, rather than frequency-encoded, signals. Threshold functions relationship to the biological shape, size... Papers needed.

Biological neural networks are many orders of magnitude slower than those based in ... It in not the size of the network or the number of interconnections alone which confer upon the human brain its remarkable efficiency (Faggin, 1991). Though size and connectivity are necessary, it is the structure, or topology of the network of interconnections that en

2.1.2 Historical Overview. The study of artificial neural networks began with a 1943 paper (4) by McCulloch and Pitts. In this paper, McCulloch and Pitts united, for the first time, neurophysiology and formal logic in a model of neural activity. This landmark paper marked the beginning of not only the computational theory of neural networks but also the computational theory of mind, and eventually led to the notion of finite atomata (6). In this paper McCulloch and Pitts introduced a very simple model of a neuron, which acted as a threshold-based propositional logic unit. Significantly, McCulloch and Pitts showed that a network of these neuron models, interconnected, could represent a proposition of arbitrarily-high complexity. Said differently, a network of these neurons can represent any logical proposition. These models, often called McCulloch-Pitts Neurons, allow only discrete input values which are summed and compared to a threshold value during a fixed time quantum, and do not posses any learning mechanism. Theses neurons are able to incorporate inhibitory action, but the action is absolute and inhibits the activation of the neuron without regard to any other considerations. These models are of theoretical significance, but cannot be applied to practical problems.

Though McCulloch and Pitts made mention of learning in their 1943 paper, thirteen years would pass before this concept was formalized into a mathematical and computational model. In 1956 Rochester, Holland, Haibt, and Duda (7) presented the first attempt at using a physiologically-inspired learning rule to update the synaptic weights of a neural network. This model was based on the correlation learning rule postulated in 1949 by Hebb[1]. In his book *The Organization of Behavior*, Hebb suggested that synaptic plasticity, that is the capacity of synaptic strengths to change, is driven by metabolic and structural

[1]It should be mentioned that, while Hebb was the first to postulate the correlation learning rule as it relates to neurons and synaptic connection strength, the abstract rule was foreshadowed as early as 1890 by William James (**?**) in Chapter XVI of *Psychology (Briefer Course)*.

changes in the both neurons near the synaptic cleft (2) such that if two cells often fired simultaneously the efficiency with which they cause one another to fire will increase. This efficiency is now called a synaptic weight. Rochester et. al. showed that the addition of variable synaptic weights alone was not sufficient to produce a network capable of learning. It was also necessary that these weights be capable of assuming inhibitory values.

The next major contribution to the field would come in 1958 with Rosenblatt's introduction of the perceptron (**?**). The perceptron was the first (**?**) well formed, computationally oriented neural network. Crucially, and unlike most preceding neural models, the model Rosenblatt presented in his 1958 paper was associative. That is, the model learned to associate stimuli with a response. This learning is accomplished by modifying the synaptic weights of the model such that the difference between an input pattern and the desired output pattern is minimized. The responsibility for the error, or difference between the correct and computed output patterns, is divided among the weights in proportion to their magnitude. Thus, large synaptic weights will be reduced more than small synaptic weights for a large, positive error. This weight update strategy is represented mathematically as:

$$w_i(t+1) = w_i(t) + \alpha(d_j - y_j)x_{j,i} \tag{2.1}$$

where $w_i(t)$ the synaptic weight for feature i at discrete time t, α is the tunable learning rate parameter, d_j is the desired output, y_j is the computed output, and x_j, i is the input pattern. This method constitutes a from of reinforcement learning.

Rosenblatt's perceptron was found to be successful at predicting the correct response class for stimuli only if the responses were correlated. It was not until Block's 1962 publication that the reason for this observed performance was elucidated. In this paper, Block presented two key findings: first, that simple perceptrons require linearly separable classes to achieve perfect classification and second, the perceptron convergence theorem (**?**). Linear separability is the ability of the response classifications to be separated by a hyperplane in the n-dimensional space of the input stimuli to which they correspond. The requirement of linear separability arises directly from the way in which the output of a perceptron

response unit is calculated. The output of a perceptron response unit is given by:

$$y_j = \begin{cases} -1 & \sum_{i=1}^{n} w_{i,j} x_i \leq \Theta \\ +1 & \sum_{i=1}^{n} w_{i,j} x_i > \Theta \end{cases} \tag{2.2}$$

where y_j is the response value of response unit j, $w_{i,j}$ is the synaptic weight of the connection between activation unit i and response unit j, x_i is the activation value of activation unit i, and Θ is the threshold value of the perceptron. Block's crucial observation was that the form of the summation in the response determining equation is isomorphic to a hyperplane in an n-dimensional space. Thus, in order for the perceptron to achieve perfect classification a hyperplane must be able to separate them in the n-dimensional input space. The corollary of this observation it the perceptron convergence theorem. The theorem proves that for some learning rules, if a perfect classification is possible it will be found by the perceptron. Specifically, the class of learning rules which were found effective were those which did not change synaptic weights when a correct classification occurs. While the condition does ensure convergence, it can often result in very slow convergence. Considerably faster guaranteed convergence can be achieved using a error gradient descent learning rule (**?**), as described by Widrow and Hoff.

Minsky breaks everything

then multilayer stuff This, in turn, implies that neural networks can serve as universal function approximaters. (Cybenko) (Kurt Hornik (1991) "Approximation Capabilities of Multilayer Feedforward Networks", Neural Networks, 4(2), 251257)

then theres backprop

2.1.3 Network Topology. Feed forward recursive

2.1.4 Activation Functions.

2.1.5 Learning Strategies. General discussion.

Reference (Mendel and McClaren, 1970) and (Haykin, pg 50)

2.1.5.1 Back Propagation Training. Consider pandimonium by slefedger

Discuss (Rumelhart, Hinton, Williams, 1986)

Derive back prop. (Haykin, pg 161)

Discuss the implication of local minima.

Fig[Error surface with backprop]

2.1.5.2 Simulated Annealing. (Ackley, Hinton, and Sejnowski 1985)

This Boltzmann machine is also of historical importance, as it was the first successfully-implemented multilayered neural network (1).

2.2 Related Works in Simulated Annealing

Read (Haykin pg 556)

Read (Haykin pg 560)

Simulated annealing (SA) is a stochastic optimization algorithm which can be used to find the global minimum of a cost function mapped from the configurations of a combinatorial optimization problem. The concept of simulated annealing was introduced in by Kirkpatrick et al. in (3) as an application of the methods of statistical mechanics to the problem of discrete combinatorial optimization. Specifically, simulated annealing is an extension of the Metropolis-Hastings (5) algorithm which can be used to estimate the ground energy state of a many-body systems at thermal equilibrium. Kirkpatrick et al. applied the Metropolis-Hastings algorithm sequentially, with decreasing temperature values in order to approximate a solid slowly cooling to low temperatures. Later work by Geoff (Geoff) and Cortana et al. (Cortana) extended simulated annealing to the continuous domain. The basic simulated annealing algorithm is presented in [algo1].

[algo1]

The physical inspiration for simulated annealing. See (Haykin pg 546)

In the parlance of simulated annealing (3) a system at its maximum temperature is said to be *melted*. In the melted state, most perturbations of the system configuration are

accepted by the algorithm. Analogously, a system that has a temperature of zero, which indicates that the algorithm cannot move to any higher-error state, is said to be *frozen*. Note that a frozen system may still be perturbed into a lower-energy state.

When considering only the influence of classical thermal fluctuations in particle energy levels, the probability of a particle traversing a barrier of height ΔV at a temperature T is on the order of:

$$\mathcal{P}_t = e^{-\frac{\Delta V}{T}} \tag{2.3}$$

2.2.1 Reheating.

2.2.2 Application of Simulated Annealing to ANN Synaptic Weight Selection.

2.3 Related Works in Quantum Mechanics

Quantum mechanics is the branch of physics concerned with the physical laws of nature at very small scales. Many aspects of physical reality are observable only at these scales. Several techniques described in this document are either inspired by, or are simple models of quantum mechanical processes. These concepts are very briefly reviewed in this section.

2.3.1 Quantum Tunneling. One of the quantum phenomena for which there is no classical analog is potential barrier penetration, also known as quantum tunneling. This phenomenon arises from the probabilistic and wavelike behavior of particles in quantum physics. Tunneling plays a significant role in the behavior of bound and scattering quantum mechanical systems.

A particle with energy E incident upon a potential energy barrier of height $\Delta V > E$ has a non-zero probability of being found in, or past, the barrier. Classically, this behavior is forbidden. The probability of tunneling, $\mathcal{P}_t$, through a step barrier of height ΔV is described by:

$$\mathcal{P}_t = e^{-\frac{w\sqrt{\Delta V}}{\Gamma}} \tag{2.4}$$

where Γ is the tunneling field strength [Ref: Multivariable Opt: QAC - Mukherjee]. Figure [1] depicts a one-dimensional example of quantum tunneling.

[Figure 1]

2.3.2 Quantum Annealing. Quantum annealing is the use of quantum, rather than thermal fluctuations to traverse the free energy landscape of a system. This is accomplished by introducing an additional Hamiltonian term that does not commute with the classical Hamiltonian. This non-commutation implies that [What does it mean?... The non-commutative causes the quantum effects...but how?]. The term is introduced to account for the presence of a tunneling field which controls the frequency with which quantum fluctuations occur in the system. In effect, this term controls the relative importance of quantum effects on the behavior of the modeled system. This term, much like thermal energy in simulated annealing, is gradually reduced over the course of the simulation. [Par Source: Quantum annealing in a kinetically constrained system] The time dependent Schrdinger equation [2] for such a system has the form:

$$[\lambda(t)H' + H_0]\psi = i\hbar\frac{\partial\psi}{\partial t} \tag{2.5}$$

[Eq from Mult Opt... Murherjee] where $\lambda(t)$ is the time-variance function of the tunneling field, H' is the Hamiltonian term describing the tunneling field, and H_0 is the classical Hamiltonian.

The fluctuations induced by the tunneling field are tunneling events, which transition the system from one configuration to a different, lower-energy configuration directly, without assuming any of the higher energy configurations between the two. Said differently, the quantum tunneling field enables the penetration of energy barriers. The addition of these quantum fluctuations also ensures that each possible state of the system can be reached (KCS Paper).

[2]Note that the presence of the Schrdinger equation in section does not imply that quantum annealing requires the annealed system must be an approximation to a wavefunction. It merely serves as an exposition of the properties of physical system which is modeled.

2.4 Notation and Terminology Conventions

There is a great deal of academic writing describing quantum annealing in the language of physics, but very little writing describing the concept from an algorithmic perspective. For this reason a new, more specific term is introduced in this document. Simulated quantum annealing (SQA) is the quantum mechanical counterpart of simulated thermal annealing.

The term neuron will be used in this document to describe the information processing elements of a neural network. This convention is selected both for conciseness and for the useful adjectival form, neural, which will be of great explanatory utility in the coming chapters.

Read (Haykin pg 561) Table 11.1

III. Methodology

3.1 Simulated Quantum Annealing

It is instructive to contrast equations 2.1 and 2.2. Both describe the same value, but the importance of the width and height of the traversed barrier in the two equations is considerably different. For systems in which quantum tunneling is possible, the probability of penetrating a barrier of height ΔV is increased by a factor of approximately $e^{\Delta V}$, for large values of ΔV. This relationship is depicted graphically in Fig. 2 which shows the probability of barrier traversal for a system which allows quantum fluctuations, divided by the same probability for a system which only considers thermal fluctuations. Therefore, physical models which considers quantum effects are much more likely predict penetration of tall, thin energy barriers than those which only include classical thermal effects.

[Figure 2]

3.2 Traversing the Error Manifold

3.2.1 Unidimensional Weight Perturbation.

3.2.2 Omni-dimensional Weight Perturbation.

3.2.3 Constrained Step-Size Omni-dimensional Weight Perturbation.

3.2.4 Quantum Weight Perturbation. It is shown in Proof [n] that this algorithm is certain to eventually find the minimum possible

3.2.5 Stochastic-Anisotropic Quantum Weight Perturbation.

3.2.6 Quantum Annealing. [After discussing the way in which the algorithm is implemented] ...The net effect of this design is to allow the algorithm to move from a local minima configuration, to a different, lower-error configuration, without requiring the evaluation of intervening, higher-error configurations. This means that the probability of "tunneling" to a state

IV. Results

Is my algorithm computationally efficient as in Haykin, pg 229?

V. Conclusion

Appendix A. First appendix title

A.1 In an appendix

This is appendix section A.1.

Note: I highly recommend you create each chapter in a separate file including the `\chapter` command and `\include` the file. Then you can use `\includeonly` to process selected chapters and you avoid having to latex/preview/print your entire document every time.

Bibliography

1. Haykin, Simon. *Neural networks: a comprehensive foundation*. Upper Saddle River, N.J: Prentice Hall, 1999.

2. Hebb, D. O. *The organization of behavior; a neuropsychological theory, (by) D.O. Hebb. Science Editions*. New York: John Wiley and Sons, 1967.

3. Kirkpatrick, S., et al. "Optimization by simulated annealing," *SCIENCE*, *220*(4598):671–680 (1983).

4. McCulloch, Warren S. and Walter Pitts. "Neurocomputing: Foundations of Research." edited by James A. Anderson and Edward Rosenfeld, chapter A Logical Calculus of the Ideas Immanent in Nervous Activity, 15–27, Cambridge, MA, USA: MIT Press, 1988.

5. Metropolis, N., et al. "Equation of State Calculations by Fast Computing Machines," *21*:1087–1092 (June 1953).

6. Piccinini, Gualtiero. "Computational Explanation in Neuroscience," *Synthese*, *153*(3):343–353 (2006).

7. Rochester, N., et al. "Tests on a cell assembly theory of the action of the brain, using a large digital computer," *Information Theory, IRE Transactions on*, *2*(3):80–93 (September 1956).

Vita

Insert your brief biographical sketch here. Your permanent address is generated automatically.

Permanent address: 452 Orchard Drive
Oakwood, Ohio 45419