Parallel tracing of multiple trajectories in gradient descent algorithm with Cell Broadband Engine

Yuri Boiko and Gabriel Wainer
Carleton University
1125 Colonel By Drive,
Ottawa, ON, K1S 5B6 CANADA
yuri.boiko@rocketmail.com, gwainer@sce.carleton.ca

Keywords: gradient descent algorithm; Cell Broadband Engine; multiple trajectories.

Abstract

Explored here is the ability of Cell B.E. to efficiently reveal viable solutions of nonlinear function approximation with multilayer perceptron (MLP) employing gradient descent algorithm. The capacity of Cell BE to asynchronously trace several trajectories of implemented gradient descent algorithm from random set of starting points offers advantage of revealing statistical trends and classifying viable optimal approximations delivered by simulated function generator. Approximation conditions of surfaces of 2nd and 3rd order with saddle points, such as hyperbolic paraboloid $z=x^2-y^2$, and Monkey saddle $z=x^3-3xy^2$, are determined via implementation of gradient descent algorithm (its back propagation version) for 3 layers MPL. Demonstrated are conditions of generating function approximations with (1)highly irregular error distribution, (2)close to uniform error distribution as well as (3)enhanced approximation. In the last case the overall error is significantly smaller than that programmed in the algorithm to be attained via training patterns. Such enhanced solutions offer advantage of attaining highly accurate function representation within minimized resources of MLP (i.e. with minimized number of hidden neurons in the MLP).

1. INTRODUCTION

Efficient function generators are needed in the areas of signal processing systems, such as those speech recognition [1], text-to-speech synthesizers [2], Optical Character Recognition (OCR), data mining, image compression, medical diagnosis, Automatic Speech Recognition (ASR). Artificial Neural Networks are recognized to be able of delivering efficient technical solutions, which gain recent advancements with ability to adapt neuron's activation function [3-5]. Implementing algorithms for generating specific nonlinear functions to optimize the respective MLP design has importance in view of further implementation of it in the hardware. Recent advancement in Cell B.E. development offers efficient tool for algorithms implementation in parallel architecture, which significantly accelerates the simulation process, as well as allows for simultaneous tracing of multiple trajectories of such algorithms from randomly selected starting points, thus efficiently observing the diversity of possible solutions and allowing for efficient selection of the better ones.

2. PROBLEM FORMULATION AND IMPLEMENTATION

2.1 Nonlinear functions selection

For this study nonlinear functions were selected of 2nd and 3rd order with saddle points, which offers sufficient challenge for achieving approximation via 3-in-1-out MLP geometry. Specifically, the first function was the standard saddle surface or hyperbolic paraboloid $z=x^2-y^2$, shown in Fig. 1 in it's analytical representation. Second selected function was "monkey saddle" defined by the equation $z = x^3 - 3*x*y^2$ as shown in Fig.2. MLP consisted of three layers with 3 inputs (x, y, 1), one hidden layer with tanh(.) activation function for the neurons and linear output neuron, thus allowing 3-D function generation.

2.2 Simulation environment

As a simulation environment a multithreaded sinergistic mode on Cell B.E. was implemented by allowing PPU to initiate asynchronous threads of

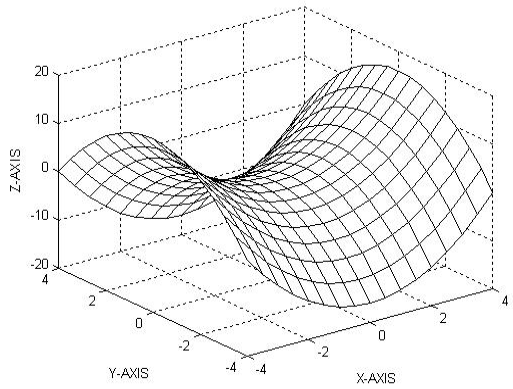

Figure 1. Analytical representation of hyperbolic paraboloid $z=x^2-y^2$ for the area under simulation.

.

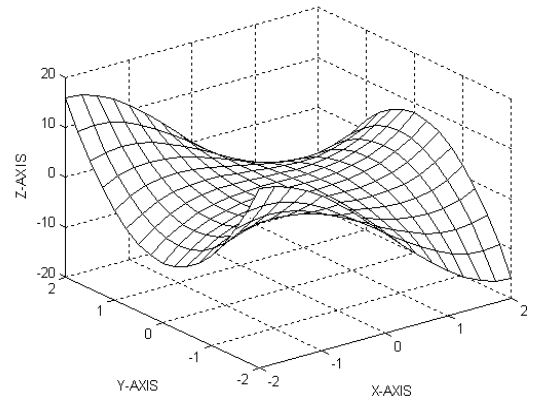

Figure 2. Analytical representation of "monkey saddle" $z = x^3 - 3*x*y^2$ for the simulated area.

gradient descent algorithm on available SPU's concurrently. Solutions were attained via subjecting MLP to training for selected training points until the convergence lead to attaining the desired level of mean squared error ε. The output parameters of each SPU run are weights of the MLP's neurons, which compose the neural function generator/ approximator. The algorithm solutions, which are above mentioned sets of the input weights for MLP's neurons, are classified according to the statistical characteristics of the function approximation they provide, such as under-fitting, over-fitting, minimal mean squared error fitting. The starting points (i.e. weights of the neurons) for the gradient descent algorithm deterministically define the convergence route to the solution, which satisfies the conditions of optimality in that the average mean square error for a predetermined set of training points falls below chosen limit ε..Evaluation of the quality of the achieved solution was conducted by employing two more error parameters: (i) mean squared error for the most remote from training coordinate points E_t, and (ii) overall mean squared error E_d for combined coordinates including joint set of X and Y, both including all of training and testing (used in (i)) coordinates.

2.2 Solutions attained

2.2.1 Hyperbolic paraboloid $z=x^2-y^2$ case

For hyperbolic paraboloid, the algorithm convergence is achieved at minimum 7 hidden neurons and the training points sets of X=Y=[3.5 2.5 1.5 0.5 -0.5 -1.5 -2.5 -3.5]. The solution attained for this set of training points and involving 10 hidden neurons is shown in Fig.3.

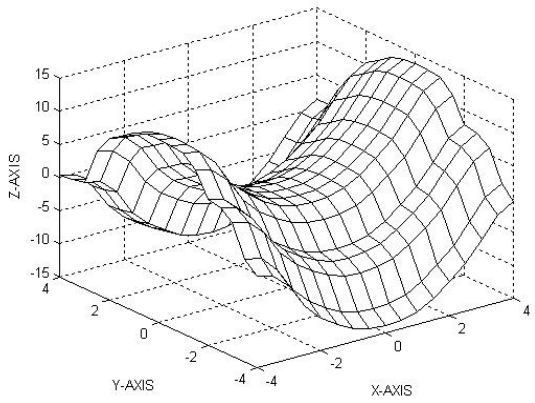

Figure 3. The function generated to approximate hyperbolic paraboloid with 10 hidden neurons and training points sets of X=Y=[3.5 2.5 1.5 0.5 -0.5 -1.5 -2.5 -3.5]. ε.= 0.000998, E_t=0.352983 and E_d=0.681416.

Error parameters for the function in Fig.3 are ε.= 0.000998, E_t=0.352983 and E_d=0.681416, while sets of testing points were $X_t=Y_t$=[4.0 3.0 2.0 1.0 0.0 -1.0 -2.0 -3.0 -4.0] and sets of overall evaluation points were $X_d=Y_d$=[4.0 3.5 3.0 2.5 2.0 1.5 1.0 0.5 0.0 -0.5 -1.0 -1.5 -2.0 -2.5 -3.0 -3.5 -4.0]. It is seen that because $\varepsilon. << E_t < E_d$, the errors are minimal near

the training points while grow significantly for any other points, to which the MLP was subjected to. The tests reveal that significant enhancement of the quality of the approximation can be achieved by scaling down by factor of 2 the distance between training points, i.e. for X=Y=[1.75 1.25 0.75 0.25 -0.25 -0.75 -1.25 -1.75]. Under these circumstances the instances took place when the overall attained error was much smaller than the preprogrammed error for the training points, i.e.. $E_d << \varepsilon$. This case is illustrated in Fig.4.

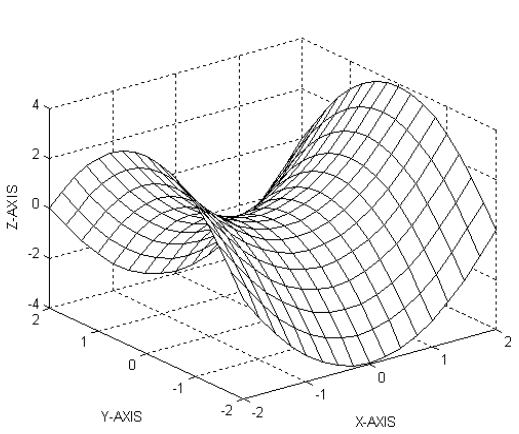

Figure 4. Enhanced approximation of the hyperbolic paraboloid with 10 hidden neurons and training points sets of X=Y=[1.75 1.25 0.75 0.25 -0.25 -0.75 -1.25 -1.75]; ε.= 0.000999, E_t=0.007481 and E_d=0.000007.

For the case of Fig.4 the E_t value was measured for sets $X_t=Y_t$=[2.0 1.5 1.0 0.5 0.0 -0.5 -1.0 -1.5 -2.0] , while E_d was attributed to the sets of points $X_d = Y_d$ = [2.0 1.75 1.5 1.25 1.0 0.75 0.5 0.25 0.0 -0.25 -0.5 -0.75 -1.0 -1.25 -1.5 -1.75 -2.0]. It is seen that distribution of errors for various sets of points changes drastically, overall being in favor of points to which MLP had no exposure (because $E_d << \varepsilon$). It is worth to note that ratio ε / E_d = 0.000999 /0.000007 = 142.7, i.e. overall enhancement attained in mean squared error is more than 2 orders of magnitude, which suggests that high quality of approximation is achievable at moderate expense (same 10 hidden nodes as in Fig.3) when enhanced mode is attainable for the gradient descent.

2.2.2 "Monkey saddle" $z = x^3 - 3*x*y^2$ case

For the case of "monkey saddle", the convergence of the gradient descent algorithm become attainable with the least number of hidden nodes being 10 and under training sets of points X=Y= [1.75 1.25 0.75 0.25 -0.25 -0.75 -1.25 -1.75]. Worth while range of errors is attainable is number of hidden nodes is increased to 12, th result of which is shown in Fig.5.

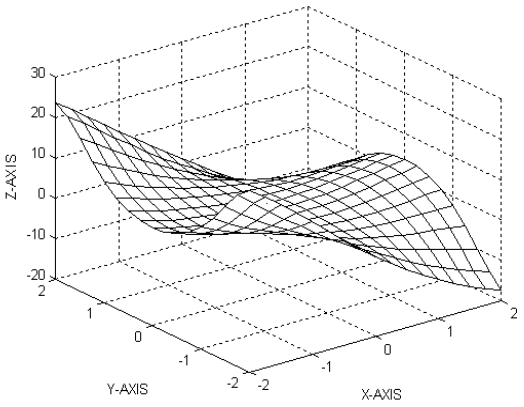

Figure 5. The function generated to approximate "monkey saddle" with 12 hidden neurons and training points sets of X=Y=[1.75 1.25 0.75 0.25 -0.25 -0.75 -1.25 -1.75]. ε.= 0.050009, E_t=0.136521 and E_d=0.181012.

The quality of the approximation is still low, particularly because of the relation $\varepsilon < E_t < E_d$, which means that best approximation is attained at training points only and errors increase for the rest of the points. Coordinates were $X_t=Y_t$=[2.0 1.5 1.0 0.5 0.0 -0.5 -1.0 -1.5 -2.0] and $X_d = Y_d$ = [2.0 1.75 1.5 1.25 1.0 0.75 0.5 0.25 0.0 -0.25 -0.5 -0.75 -1.0 -1.25 -1.5 -1.75 -2.0].

Scaling down the distance between training points by factor of 2, which was very much successful for the hyperbolic paraboloid, brings a success, which is worth mentioning by illustrating it in Fig.6, even though enhancement attained is not as dramatic as before. Coordinates in question become $X_t = Y_t$ = [1.0 0.75 0.5 0.25 0.0 -0.25 -0.5 -0.75 -1.0] and $X_d = Y_d$ = [1.0 0.875 0.75 0.625 0.5 0.375 0.25 0.125 0.0 -0.125 -0.25 -0.375 -0.5 -0.625 -0.75 -0.875 -1.0].

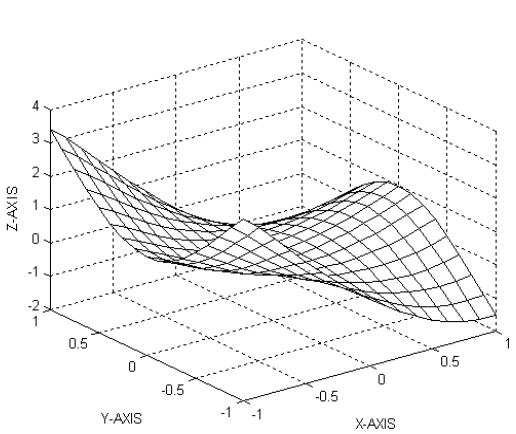

Figure 6. The function generated to approximate "monkey saddle" with 12 hidden neurons and training points sets of X = Y = [0.875 0.625 0.375 0.125 -0.125 -0.375 -0.625 -0.875]; ε.= 0.050001, E_t=0.023518 and E_d=0.030031.

The attained error parameters become ε.= 0.050001, E_t=0.023518 and E_d=0.030031, i.e. enhanced mode is attained for points most remotely located from the training points (which was not the case in hyperbolic paraboloid), and this brings overall enhancement by the factor of ε /E_d = 0.050001 /0.030031 = 1.66 with the best enhancement for points most remote from training ones: ε./E_t= 0.050001/ 0.023518 = 2.1. Distribution of errors for the "monkey saddle" case is much more uniform than that for the hyperbolic paraboloid for either mode of approximation – enhanced or not.

3. CONCLUSIONS

The following conclusions can be drawn with regard to suitability of Cell B.E. for gradient descent algorithm implementation for nonlinear function approximation of second and third order with saddle points.

1. Cell B.E. is allowing for efficient tracing of multiple trajectories of the gradient descent algorithm, revealing best approximation conditions.
2. For the hyperbolic paraboloid $z=x^2-y^2$ the enhanced mode of function approximation is attained gaining overall mean squared error reduction of more than two orders of magnitude (ε./E_d= 142).
3. For "monkey saddle" $z = x^3 - 3*x*y^2$ the enhanced mode of function approximation is also attainable and strongest demonstrated enhancement is attained for points most remote from the training ones (ε. / E_t = 2.1), gaining overall mean squared error reduction by factor of ε./E_d= 1.66.

References:

[1] Computational Auditory Scene Analysis: Principles, Algorithms, and Applications, Wang, D. and Brown, G.J., Eds. (2006).

[2] Shaw-Hwa Hwang Sin-Horng Chen, "Neural network synthesiser of pause duration for Mandarin text-to-speech,"- Electronics Letters, Vol. 28 (8), pp.720-721 (1992).

[3] D. Larkin, A. Kinane, V. Muresan and N. O'Connor, "An Efficient Hardware Architecture for a Neural Network Activation Function Generator,"- in: Advances in Neural Networks - ISNN 2006, Ed.Springer Berlin / Heidelberg, Vol.3973, pp.1319-1327 (2006). http://www.springerlink.com/content/h27j70h25680j765/fulltext.pdf

[4] J.J. Soraghan, A. Hussain, "Neural network for predicting values in non-linear functional mappings,"- US Patent 6,453,206 (2002). http://www.freepatentsonline.com/6453206.html

[5]. H.K. Rising, "Method and apparatus of compressing images using localized radon transforms,"- US Patent 6,424,737, (2002) - http://www.freepatentsonline.com/6424737.html