Mackey Glass Time Series Prediction using LMS Algorithm

Jayasurya Sevalur Mahendran[1], Huseyin Seckin Demir[2], Kamalesh Kalirathinam[3]

Arizona State University
Tempe, AZ, 85281, USA
[1]jsevalur@asu.edu, [2]hdemir@asu.edu, [3]kamales1@asu.edu

***Abstract*— This paper describes the use of Least Mean Square Algorithm to predict Mackey Glass Time Series. A data set of 5000 samples are generated using Mackey-Glass equation among which 4000 samples are used for training and 1000 for testing. The algorithm is tested around different learning rates and tap values to gain behavioural insights of the algorithm by generating plots.**

***Keywords*— Mackey Glass, LMS algorithm, Runge-Kutta method**

I. PROBLEM FORMULATION

The **Mackey-Glass equation** is a nonlinear time delay differential equation; whose general form can be given as [1]:

$$\frac{dx(t)}{dt} = \beta \frac{x(t-\tau)}{1+x(t-\tau)^n} - \gamma x(t) \tag{1}$$

where β, γ, τ, n are real numbers, and x(t-τ) represents the value of the variable x at time (t−τ). Depending upon the values of these parameters, the equation displays a range of periodic and chaotic dynamics.

We are going to generate Mackey-Glass time series data by using The Runge-Kutta Method.

Any first order differential equation can be solved using the Runge-Kutta Method of order 4 using successive approximation technique [1].

$$k_1 = f(x_t, y_t) \tag{2}$$
$$k_2 = f\left(x_t + \frac{1}{2}h, y_t + \frac{1}{2}k_1 h\right) \tag{3}$$
$$k_3 = f\left(x_t + \frac{1}{2}h, y_t + \frac{1}{2}k_2 h\right) \tag{4}$$
$$k_4 = f(x_t + h, y_t + k_3 h) \tag{5}$$
$$y_{t+1} = y_t + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h \tag{6}$$

Where h is the step height.

A. Parameters and Initial Conditions

We have generated the Mackey-Glass time series data using the following parameters;

β = 0.2; γ = 0.1; τ = 17; x0 = 1.2; (initial value of x at t=0)
Δt= 0.1; Total number of samples = 5000

We have used [2] for creating the MATLAB code to generate our time series data.

II. METHODS

We have used Least Mean Square(LMS) algorithm to predict our Mackey Glass Time series data. The LMS algorithm was introduced in the year 1959 by Widrow and Hoff. The Least Mean Square Algorithm is an adaptive algorithm, that uses a gradient-based method of steepest decent [3]. LMS algorithm uses the estimates of the gradient vector from the available data. LMS utilizes an iterative procedure that makes successive corrections to the weight vector in the direction of the negative of the gradient vector which eventually leads to the minimum mean square error. It tries to reduce the Euclidean distances between the data points and the straight line fitted in case of linear regression.

Since the statistics is estimated continuously, the LMS algorithm can adapt to changes in the signal statistics; The LMS algorithm is thus an adaptive filter. Because of estimated statistics the gradient becomes noisy.

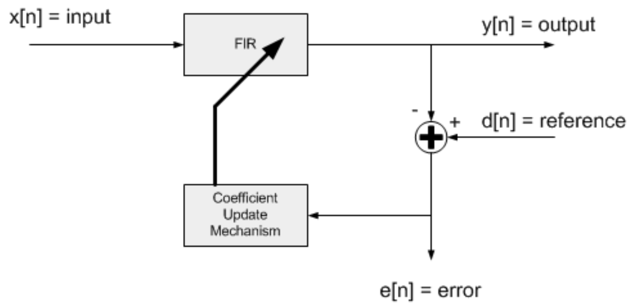

Figure 1. Least Mean Square Filter

An LMS filter consists of two components namely the FIR filter and the coefficient update mechanism. The LMS filter has two input signals namely the input signal and the reference signal. The "input" feeds the FIR filter while the "reference input" corresponds to the desired output of the FIR filter. The FIR filter coefficients are updated such that the reference input (desired output) matches the output of the FIR filter. The filter coefficients are updated based on the difference between the FIR filter output and the reference input. This "error signal" tends towards zero as the filter adapts. The LMS processing

functions accept the input and reference input signals and generate the filter output and error signal.
The output signal y[n] is computed by a standard FIR filter

$$y[n] = w[0] * x[n] + w[1] * x[n-1] + w[2] * x[n-2] + \cdots + w[numTaps-1] * x[n-numTaps+1] \qquad (7)$$

The error signal equals the difference between the reference signal d[n] and the filter output:

$$e[n] = d[n] - y[n] \qquad (8)$$

After each sample of the error signal is computed, the filter coefficients w[k] are updated on a sample-by-sample basis:

$$w[k] = w[k] + e[n] * \eta * x[n-k] \qquad (9)$$

For k=0, 1, ..., numTaps-1

where η is the step size and controls the rate of coefficient convergence.

We have used and modified [4] for implementing LMS algorithm for Mackey-Glass time series prediction.

In our implementation, we have randomly initialized the weight vector w, where the values are distributed uniformly in the range [0,1]. We have used 4 different values of η as 0.0005, 0.001, 0.005, 0.01. We also used 3 different values for numTaps which are 3, 5 and 7. Our results are given in the next section.

III. RESULTS

The mean squared errors (MSE) calculated on the test data using different learning rates and tap numbers are shown in Table 1. It is calculated as the mean of the squared error between the predicted value and actual value of the test data at each point.

TABLE I
TEST MSE FOR DIFFERENT LEARNING RATES AND FILTER ORDERS

Learning Rate	Number of Taps for LMS		
	3	5	7
0.0005	0.000186	0.001297	0.000034
0.001	0.000181	0.001325	0.000038
0.005	0.000221	0.001918	0.000068
0.01	0.000217	0.001471	0.000058

Even if there does not seem a sharp tendency in the table possibly because of random initialization of the weights, we can say that the minimum mean squared error is achieved by using the learning rate value as 0.0005 and tap value of 7.

Figure 2 demonstrates the convergence behaviour of the algorithm for three different cases where the number of taps are 3, 5 and 7. In each graph, squared error over the iterations are shown for different learning rates. As we expect, the algorithm converges faster when the learning rate is increased.

We can also see that the convergence speed is higher when the number of taps is increased. This result is also expected, since we utilize and update more number of weights in each iteration.

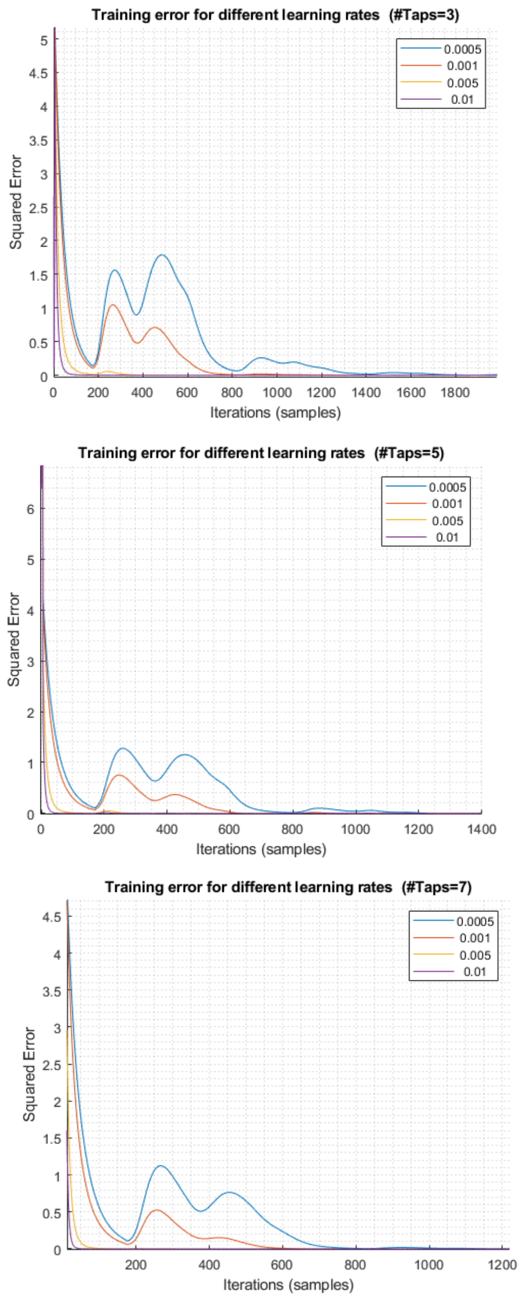

Fig. 2 Squared error during the training phase

In order to show the convergence behaviour of the LMS algorithm, we also included the actual values of training set and the predicted values of the training set for different tap numbers and learning rates in Figure 3. As the figure shows, predicted values approach faster to the actual values when the learning rate is higher.

Figure 4 shows the prediction results of the test data for different tap numbers and learning rates. We observe that the model successfully learns and predicts the unseen data from the previous samples.

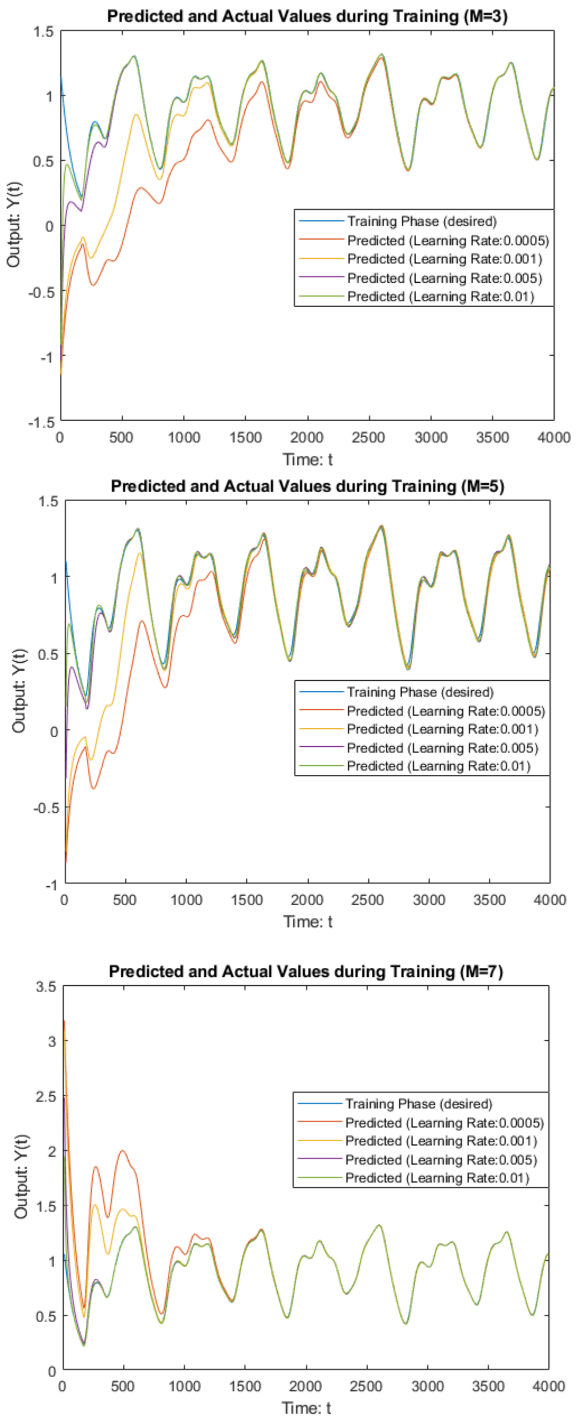

Fig. 3 Actual and predicted values of the training data for different tap numbers and learning rates

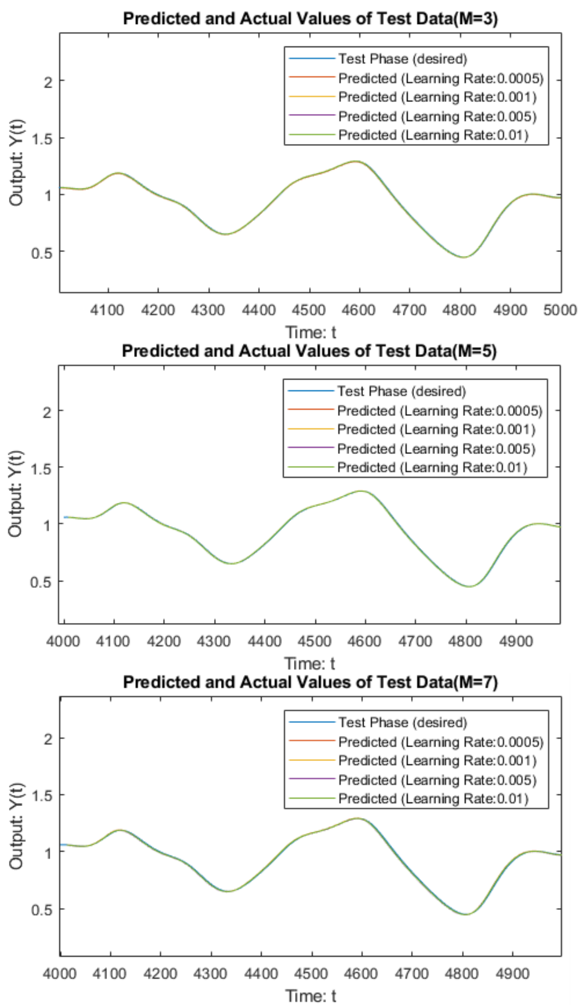

Fig. 4 Actual and predicted values of the test data for different tap numbers and learning rates

IV. Conclusions

Irrespective of the tap values and given the same randomized weights for training, a higher value of learning rate results in faster convergence and a lower learning rate results in a slower convergence. The converges is very much dependent on the correlation matrix and the sufficient condition for convergence is $0<\eta<2/\lambda_{max}$ where λ_{max} is the highest eigen value of the correlation matrix and η is the learning rate.

Some of the questions for future exploration can involve finding the optimal η for a given data set and a comparison between LMS, NLMS (Normalized Least Mean Square), RLS (Recursive least square) and FTRLS (Fast Traversal least square) algorithms.

References

[1] Leon Glass and Michael Mackey, *Mackey-Glass equation. Scholarpedia, 5(3):6908.*

[2] Marco Cococcioni, *Mackey-Glass time series generator*, https://www.mathworks.com/matlabcentral/fileexchange/24390-mackey-glass-time-series-generator, *online*

[3] *Kale, Karbhari Viswanath. Advances in computer vision and information technology. IK International Pvt Ltd, 2013.*

[4] Shujaat Khan, *Mackey Glass Time Series Prediction Using Least Mean Square (LMS), https://www.mathworks.com/matlabcentral/fileexchange /61017-mackey-glass-time-series-prediction-using-least-mean-square-lms?s_cid=ME_prod_FX*