ARTIFICIAL NEURAL NETWORK

Swetha R.

Department of Electronics and Communication Engineering

ARTIFICIAL NEURAL NETWORK

Class 8

LEAST-MEANS-SQUARE(LMS) ALGORITHM

Swetha R.
Department of Electronics and Communication Engineering

LEAST-MEANS-SQUARE(LMS)

- In Least-Mean Square (LMS) , developed by Widrow and Hoff (1960), was the first linear adaptive filtering algorithm (inspired by the perceptron) for solving problems such as prediction
- Some features of the LMS algorithm:
  - Linear computational complexity with respect to adjustable parameters.
  - Robust with respect to external disturbance

LEAST-MEANS-SQUARE(LMS)

- The aim of the LMS algorithm is to minimize the instantaneous value of the cost function E (w) :

$$E(w) = \frac{1}{2} e^2(n)$$

- where e(n) is the error signal measured at time n.
- Differentiation of E (w), with respect to w, yields :

$$\frac{\partial E(w)}{\partial w} = e(n) \frac{\partial e(n)}{\partial w}$$

LEAST-MEANS-SQUARE(LMS)

As with the least-square filters, the LMS operates on linear neuron, we can write:

$$e(n) = d(n) - X^T(n)w(n)$$

$$\frac{\partial e(n)}{\partial w} = - X(n)$$

$$g(n) = \frac{\partial E(w)}{\partial w} = - e(n)X(n)$$

LEAST-MEANS-SQUARE(LMS)

Weight update equation in steepest descent algorithm

$$w(n+1) = w(n) - \eta g(n)$$

Therefore,

$$\boxed{w(n+1) = w(n) + \eta e(n)X(n)}$$

Where η is the learning rate parameter

LEAST-MEANS-SQUARE(LMS)

- The **inverse** of the **learning-rate** acts as a **memory** of the LMS algorithm. The smaller the learning-rate η , the longer the memory span over the past data,
  - which leads to more accurate results but with slow convergence rate.
- In the **steepest-descent** algorithm the weight vector w(n) follows a **well-defined** trajectory in the weight space for a prescribed η.

LEAST-MEANS-SQUARE(LMS)

- In contrast, in the **LMS** algorithm, the weight vector w(n) traces a **random** trajectory. For this reason, the LMS algorithm is sometimes referred to as "**stochastic gradient algorithm**."
- Unlike the steepest-descent, the LMS algorithm **does not** require knowledge of the **statistics** of the **environment**. It produces an **instantaneous estimate** of the weight vector.

LEAST-MEANS-SQUARE(LMS)

TABLE 3.1 Summary of the LMS Algorithm

Training Sample: Input signal vector $= \mathbf{x}(n)$

Desired response $= d(n)$

User-selected parameter: η

Initialization. Set $\hat{\mathbf{w}}(0) = \mathbf{0}$.

Computation. For $n = 1, 2, \ldots,$ compute

$$e(n) = d(n) - \hat{\mathbf{w}}^T(n)\mathbf{x}(n)$$

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \eta\mathbf{x}(n)e(n)$$

Virtues and Limitation of the LMS Algorithm:

Computational Simplicity and Efficiency:

- The Algorithm is very simple to code, only two or three line of code.
- The computational complexity of the algorithm is linear in the adjustable parameters.

Virtues and Limitation of the LMS Algorithm:

Robustness

Since the LMS is model independent, therefore it is robust with respect to disturbance,(small model uncertainty and small disturbances (i.e., disturbances with small energy) can only result in small estimation errors (error signals)).

LEAST-MEANS-SQUARE(LMS)

Factors Limiting the LMS Performance:

The primary limitations of the LMS algorithm are:

- Its **slow rate** of convergence (which become serious when the dimensionality of the input space becomes **high**)
- Its **sensitivity** to variation in the eigen structure of the input. (it typically requires a number of iterations equal to about 10 times the dimensionality of the input data space for it to converge to a stable solution)

LEAST-MEANS-SQUARE(LMS)

Factors Limiting the LMS Performance:

- The sensitivity to changes in environment become particularly acute when the condition number of the LMS algorithm is high.
- The condition number,

$$\chi(R) = {}^{\lambda_{\max}}/_{\lambda_{\min}}$$

- Where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalues of the correlation matrix, R_x.

LEAST-MEANS-SQUARE(LMS)

Convergence of the LMS Algorithm:

- Convergence is influenced by the statistical characteristics of the input vector x(n) and the value assigned to the learning-rate parameter η
- By invoking the elements of independence theory and assuming the learning- rate parameter η is sufficiently small, it is shown in Haykin (1996) that the LMS is convergent in the mean square provided that η satisfies the condition

$$0 < \eta < \frac{2}{\lambda_{max}}$$

where, λ_{max} is the largest eigenvalue of the correlation matrix R_x,

LEAST-MEANS-SQUARE(LMS)

Convergence of the LMS Algorithm:

In typical applications of the LMS algorithm, knowledge of λ_{max} is not available. To overcome this difficulty, the trace of R_x, may be taken as a conservative estimate for λ_{max}, the condition is reformulated as

$$0 < \eta < \frac{2}{tr[R_x]}$$

Where, λ_{max} is the largest eigenvalue of the correlation matrix R_x,

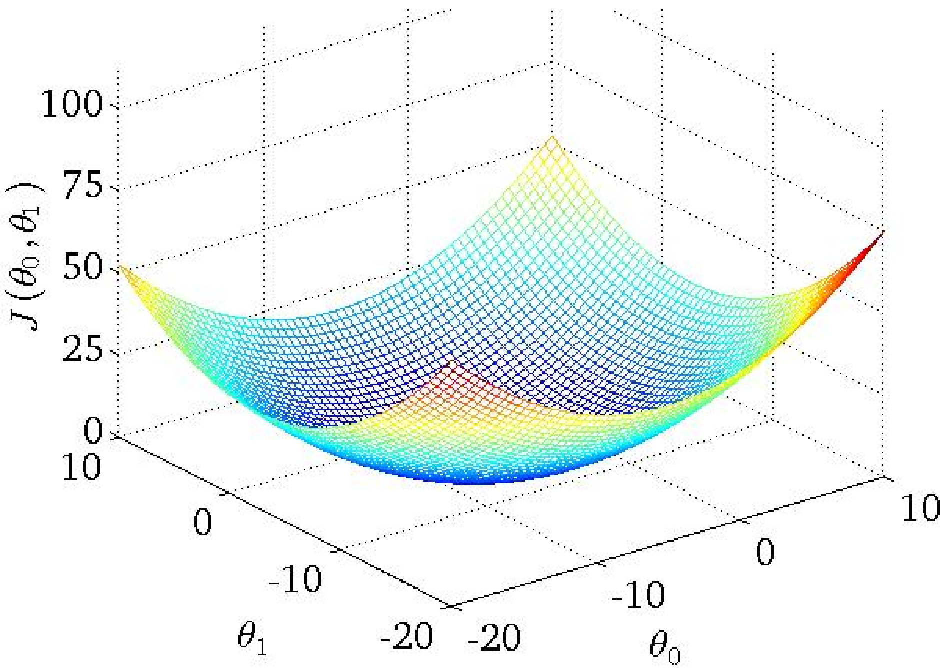
LEASE-MEANS-SQUARE(LMS)

Convergence of the LMS Algorithm:

By definition, the trace of a square matrix is equal to the sum of its diagonal elements. Since each diagonal element of the correlation matrix R_x equals the mean-square value of the corresponding sensor input

$$0 < \eta < \frac{2}{\text{sum of mean}-\text{square values of the sensor inputs}}$$

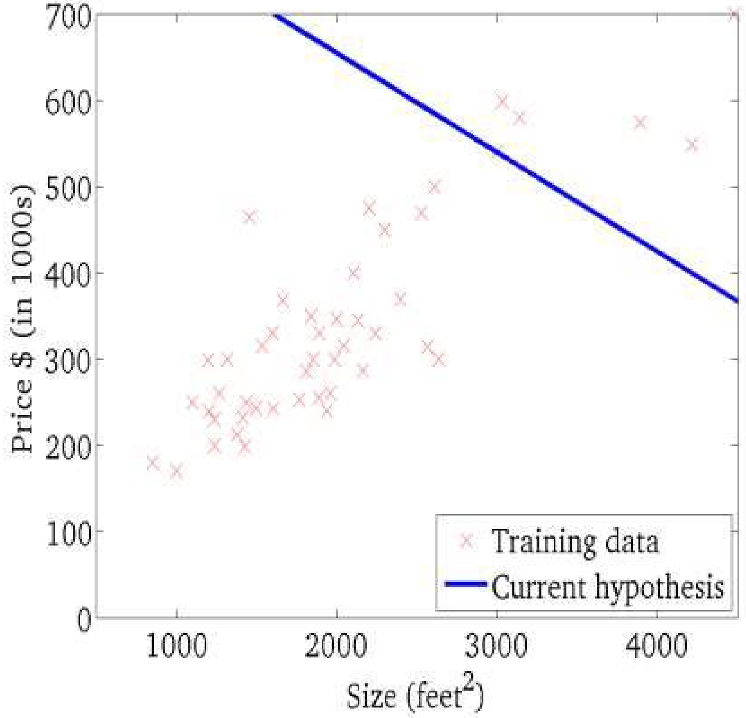
LEAST-MEANS-SQUARE(LMS)
Computational Example:

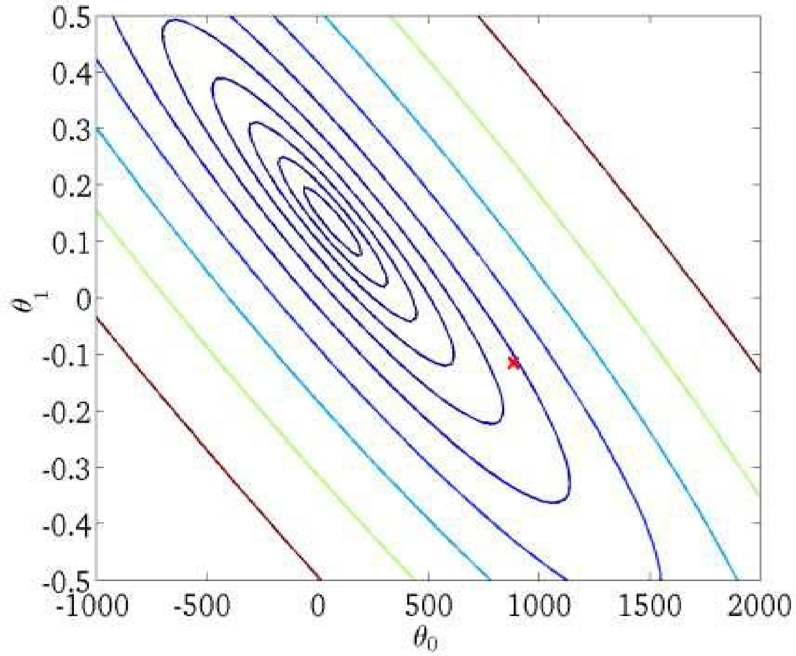


Usually *linear models* produce concave cost functions

LEAST-MEANS-SQUARE(LMS)

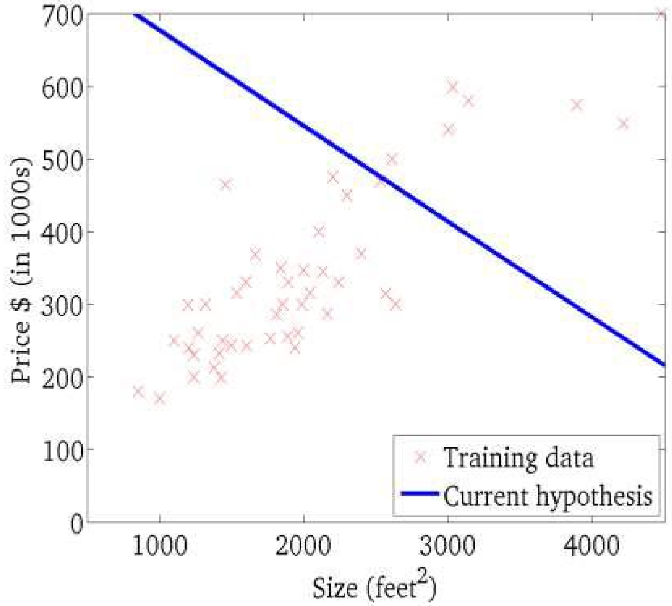
Computational Example:

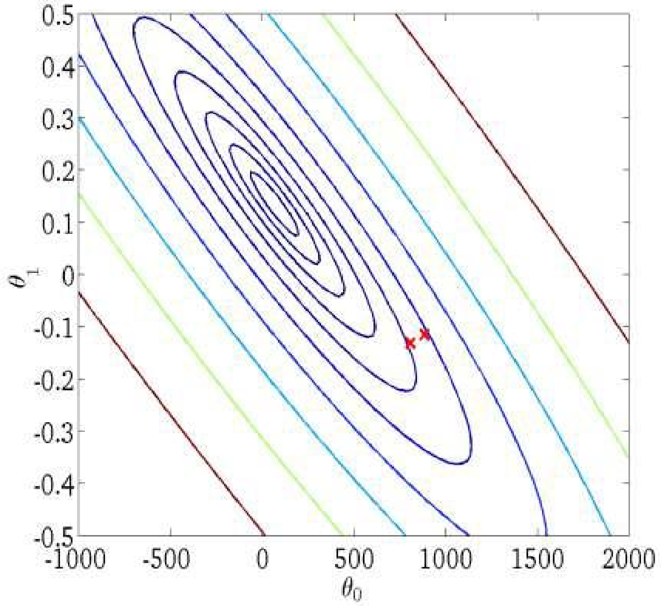




LEAST-MEANS-SQUARE(LMS)

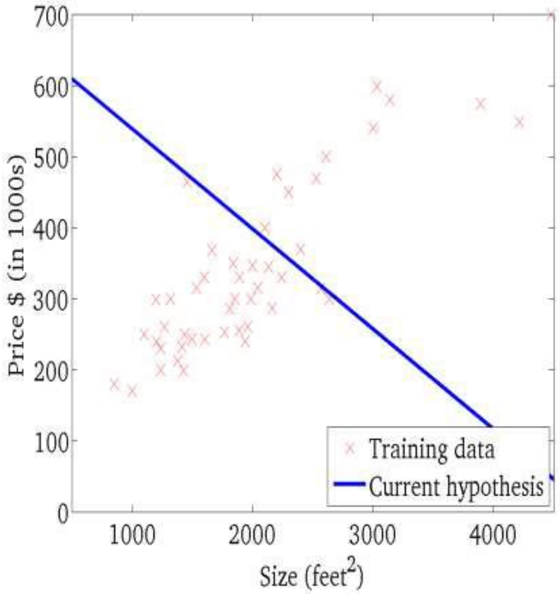
Computational Example:

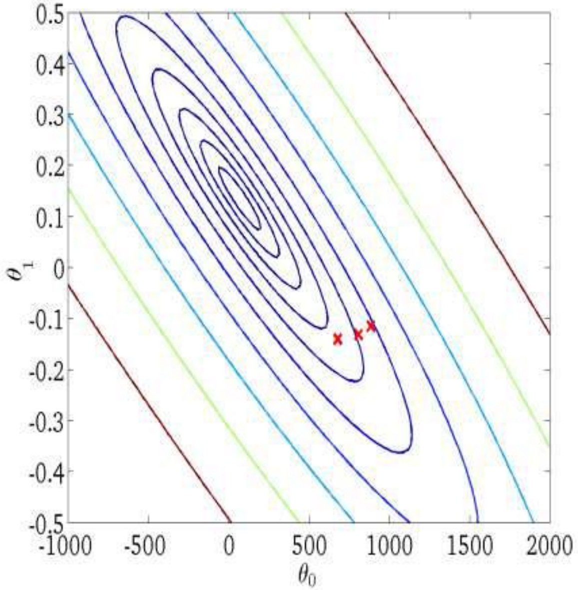




LEAST-MEANS-SQUARE(LMS)

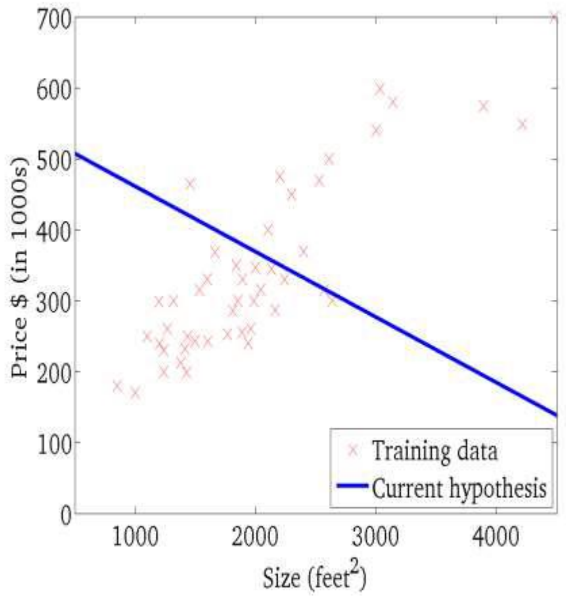
Computational Example:

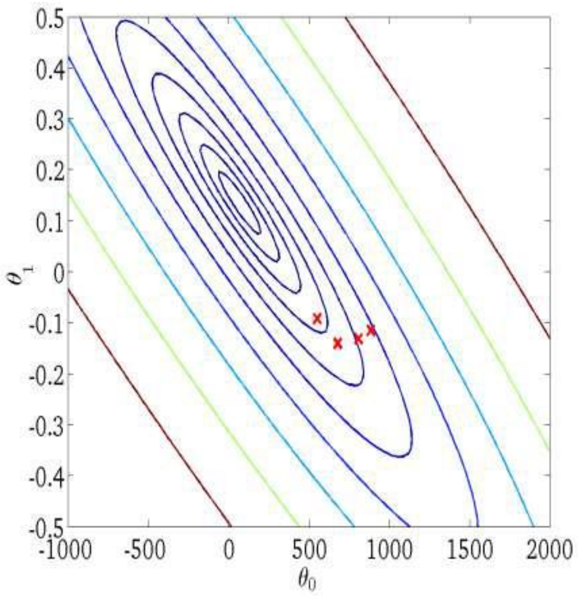




LEAST-MEANS-SQUARE(LMS)

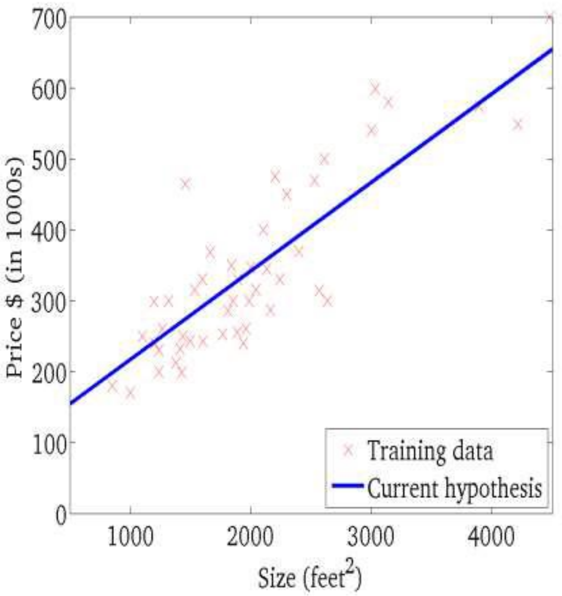
Computational Example:

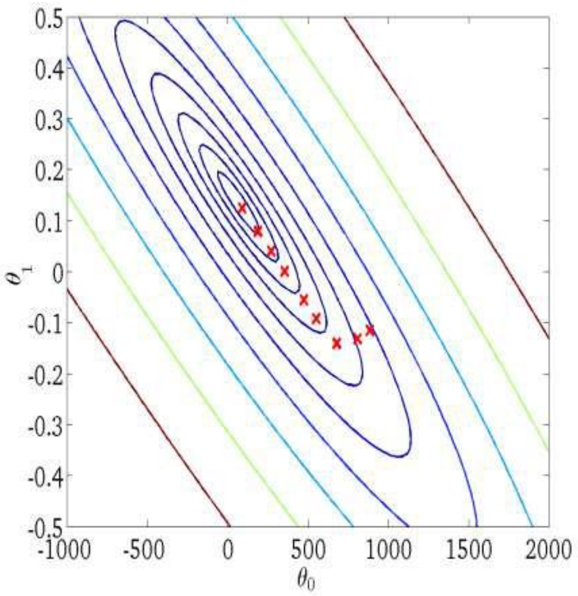




LEAST-MEANS-SQUARE(LMS)

Computational Example:





THANK YOU

Swetha R.
Department of Electronics and Communication Engineering

swethar@pes.edu

+91 80 2672 1983 Extn 753