LMS Algorithm.

(1) The performance index for ADLINE algorithm is MSE.

$$F(x) = E[e^2]$$
$$= E[(t-a)^2]$$
$$= E[(t - x^T z)^2] \quad \text{--- (1)}$$

because, we know

$$a = W^T p + b$$

which can be written as:-

$$\underline{a = x^T z.}$$

eqn.(1) can be written as:-

$$F(x) = E\left[t^2 - 2t x^T z + (x^T z . z^T x)\right]$$
$$= E[t^2] - 2x^T E[tz] + x^T E[zz^T]x. \quad \text{--- (2)}$$

This can be further written as follows:

$$F(x) = c - 2x^T h + x^T R x \quad \text{--- (3)}$$

where,

$$c = E[t^2],$$
$$h = E[tz],$$
$$R = E[zz^T].$$

h gives cross-correlation between the i/p vector and associated target, R is the i/p correlation matrix.

~~Now~~ the general form of quadratic function is

$$F(x) = \frac{1}{2} x^T A x + d^T x + C \qquad \text{--- (4)}$$

Now, if we compare eqn.(3) with (4) then we have the following:—

$$F(x) = C + d^T x + \frac{1}{2} x^T A x \qquad \text{--- (5)}$$

So, $d = -2h$ and ~~B~~ $A = 2R$.

A is a hessian matrix and if the eigenvalues of the hessian matrix are positive then we ~~have~~ ~~our~~ surely get global minimum.

R (correlation) → +ve → unique global optimum/minimum value.

R (correlation) → −ve → no minimum/weak minimum.

Calculation of the ~~[illegible]~~ gradient of eqn.(5)

$$\nabla F(x) = \nabla\left(c + d^T x + \frac{1}{2} x^T A x\right)$$

$$= d + AX$$

$$= -2h + 2RX \qquad \text{--- (6)}$$

Now to find the stationary pt. $F(x) = 0$

hence

$$-2h + 2RX = 0$$

$$\Rightarrow \boxed{X = R^{-1} h.} \qquad \text{--- (7)}$$

→ The goal is to locate the minimum point and it is only possible if performance index is known.

→ We can find minimum point directly from eqn. 7 ~~too~~ if the h and R value is possible to calculate.

→ If it is not possible then gradient ~~of~~ calculation of eqn. 6 and may be steepest descent algorithm ~~may~~ ~~to~~ help us to find the minimum. In real life problem it is not desirable or convenient to calculate h and R. For this reason we will use an approximate steepest descent algorithm

→ MSE : $F(x) = (t(k) - a(k))^2$

$$= e^2(k)$$

$e(k)$ is the ~~k~~ error observed in the k^{th} ~~...~~ iteration.

Gradient in each iteration :

$$\nabla F(x) = \nabla e^2(k)$$

→ As the weight and bias are the two parameter which adjusted in each iteration The error calculation ~~is~~ depends on the weight and bias. So, the derivatives ~~adjust~~ weight and bias is calculated in terms of weight and bias to find the ~~...~~ unique weight and bias matrix which provides minimum error.

→ Hence,

$$\nabla e^2(K) = \frac{\partial e^2(K)}{\partial w_j}$$

$$= 2e(K)\frac{\partial e(K)}{\partial w_j}$$

and

$$\nabla e^2(K) = \frac{\partial e^2(K)}{\partial b}$$

$$= 2e(K)\frac{\partial e(K)}{\partial b}$$

Now,

$$\frac{\partial e(K)}{\partial w_j} = \frac{\partial}{\partial w_j}[(t(K)-a(K))]$$

$$= \frac{\partial}{\partial w_j}\left[t(K) - (w^T p(K) + b)\right]$$

$$= -p_j(K)$$

Similarly,

$$\frac{\partial e(K)}{\partial b} = \frac{\partial}{\partial b}[t(K) - a(K)]$$

$$= \frac{\partial}{\partial b}\left[t(K) - (w^T p(K) + b)\right]$$

$$= -1$$

So, $\nabla F(x) = \nabla e^2(k)$
$= -2e(k)z(k)$ — (13)

Now, according to ~~steepest~~ descent algorithm

$x_{k+1} = x_k - \alpha \nabla F(x)$ — (14)

if we substitute eqn. (13) into (14)
then
$x_{k+1} = x_k + 2\alpha e(k) z(k)$ — (15)

eqn. (15) can be written as :—
$w(k+1) = w_k + 2\alpha e(k) p(k)$
and $b(k+1) = b_k + 2\alpha e(k)$