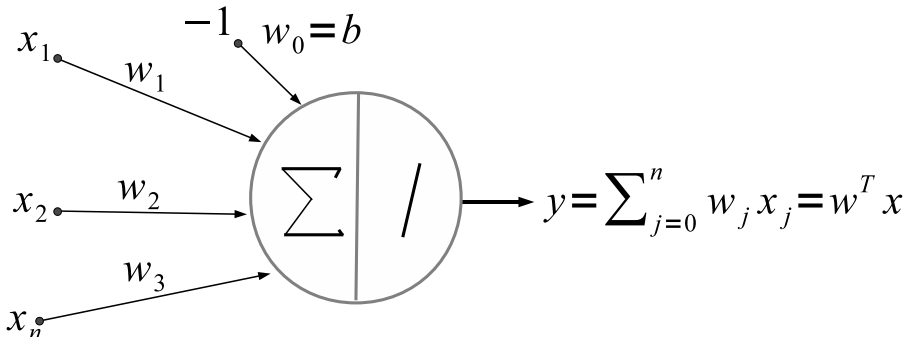
Machine Learning Group

Ovidiu Calin, September 29, 2017

Adaline neuron

The neuron has n inputs, which are random variables, $X_1, \cdots, X_n$. The weight for the input X_j is a number denoted by w_j. The bias is considered as weight $w_0 = b$. Consider $X_0 = -1$, constant. We shall adopt the vectorial notation

$$X = \begin{pmatrix} X_0 \\ X_1 \\ \cdots \\ X_n \end{pmatrix}, \qquad w = \begin{pmatrix} w_0 \\ w_1 \\ \cdots \\ w_n \end{pmatrix}$$


-1
$w_0 = b$
x_1
w_1
x_2
w_2
w_3
x_n
$y = \sum_{j=0}^{n} w_j x_j = w^T x$

Given that the activation function is the identity, the neuron output is given by the one-dimensional random variable

$$Y = \sum_{j=0}^{n} w_j X_j = w^T X = X^T w.$$

The desired output, is given by a one-dimensional random variable Z. The idea is to tune the parameter vector w such that Y learns Z; the optimal parameter is

$$w^* = \arg\min_w \mathbb{E}[(Z - Y)^2].$$

Exact solution The error function is quadratic in w

$$\begin{aligned}
\mathbb{E}[(Z-Y)^2] &= \mathbb{E}[Z^2 - 2ZY + Y^2] \\
&= \mathbb{E}[Z^2 - 2ZX^T w + (w^T X)(X^T w)] \\
&= \mathbb{E}[Z^2] - 2\mathbb{E}[ZX^T]w + w^T \mathbb{E}[XX^T]w \\
&= c - 2bw + w^T A w,
\end{aligned}$$

where $c = \mathbb{E}[Z^2]$ is the second centered moment of the target Z, $b = \mathbb{E}[ZX^T]$ is a vector that measures the cross correlation between the input and the output, and

$$A = \mathbb{E}[XX^T] = \begin{pmatrix} \mathbb{E}[X_0X_0] & \mathbb{E}[X_0X_1] & \cdots & \mathbb{E}[X_0X_n] \\ \mathbb{E}[X_1X_0] & \mathbb{E}[X_1X_1] & \cdots & \mathbb{E}[X_1X_n] \\ \cdots & \cdots & \cdots & \cdots \\ \mathbb{E}[X_nX_0] & \mathbb{E}[X_nX_1] & \cdots & \mathbb{E}[X_nX_n] \end{pmatrix}$$

is a matrix describing the autocorrelation of inputs. In the following analysis we shall assume that A is non-degenerate and positive definite (i.e. the inputs are coherent).

The quadratic error function

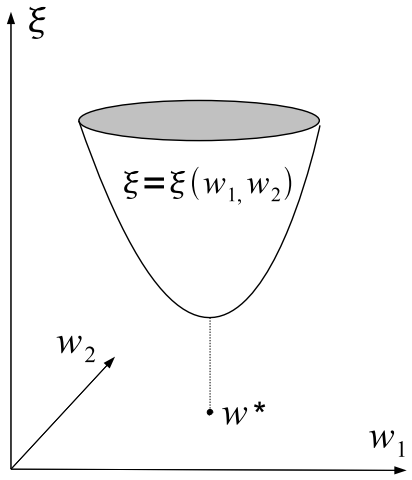
$$\xi(w) = c - 2bw + w^T Aw,$$

has the gradient $\nabla_w \xi(w) = 2Aw - 2b$. The optimal weight w^* is obtained as a solution of $\nabla_w \xi(w) = 0$, which becomes the linear system $Aw = b$. Since A is nondegenerate, the system has the unique solution $w^* = A^{-1}b$. This is a minimum point, since the Hessian of the error is given by $H_\xi(w) = 2A$, with A positive definite.

• In real life, for n is very large, it is computationally expensive to find the inverse A^{-1}. Hence, the need of a faster method to produce w^*, even if only as an approximation.

• This can be seen as a tradeoff between the solution accuracy and the computer time spent to find the optimum.

The gradient descent method is more practically. The error function $\xi(w)$ is convex and has a minimum:



We start from an arbitrary initial weight vector $w^{(0)} = (w_0^{(0)}, \cdots, w_n^{(0)})^T \in \mathbb{R}^n$.

Construct the approximation sequence $(w^{(j)})_j$

$$\begin{aligned} w^{(j+1)} &= w^{(j)} - \delta \nabla_w \xi(w^{(j)}) \\ &= w^{(j)} - 2\delta(Aw^{(j)} - b) \\ &= (\mathbb{I}_n - 2\delta A)w^{(j)} + 2\delta b \\ &= Mw^{(j)} + 2\delta b, \end{aligned}$$

where $M = \mathbb{I}_n - 2\delta A$.

Iterating, we obtain

$$\begin{aligned} w^{(j)} &= M^j w^{(0)} + (M^{j-1} + M^{j-2} + \cdots + M + \mathbb{I}_n)\, 2\delta b \\ &= M^j w^{(0)} + (\mathbb{I}_n - M^j)(\mathbb{I}_n - M)^{-1}\, 2\delta b \\ &= M^j w^{(0)} + (\mathbb{I}_n - M^j) A^{-1} b. \end{aligned} \tag{0.1}$$

Assume the learning rate $\delta > 0$ is chosen such that $\lim_{j\to\infty} M^j = \mathbb{O}_n$. Then taking the limit in the previous formula yields $w^* = \lim_{j\to\infty} w^{(j)} = A^{-1}b$, which recovers the previous result.

Return to the assumption $\lim_{j\to\infty} M^j = \mathbb{O}_n$. Since M is symmetric it has real eigenvalues. It suffices showing that the eigenvalues $\{\lambda_i\}$ of M are bounded, with $|\lambda_i| < 1$.

We do this in two steps:

Step 1. Show that $\lambda_i < 1$.

Step 2. Show that $\lambda_i > 0$ for δ small enough.

Step 1. Show that $\lambda_i < 1$.
If λ_i is an eigenvalue of M, $\det(M - \lambda_i \mathbb{I}_n) = 0$. Substituting for M, we get $\det\left(A - \frac{1-\lambda_i}{2\delta}\mathbb{I}_n\right) = 0$. This implies that $\alpha_i = \frac{1-\lambda_i}{2\delta}$ is an eigenvalue of A. Since A is positive definite and non-degenerate, it follows that $\alpha_i > 0$, which implies that $\lambda_i < 1$.

Step 2. Show that $\lambda_i > 0$ for δ small enough.
The condition $\lambda_i > 0$ is equivalent to $\frac{1-\lambda_i}{\alpha_i} < \frac{1}{\alpha_i}$, where we used that A has positive eigenvalues. This can be written in terms of δ as $2\delta < \frac{1}{\alpha_i}$. Hence, the learning rate has to be chosen such that

$$0 < \delta < \min_i \frac{1}{2\alpha_i}. \tag{0.2}$$

The closed-form expression (0.1) does not have much practical use, since it contains the inverse A^{-1}. In real life we use the iterative formula

$$w^{(j+1)} = Mw^{(j)} + 2\delta b,$$

where the learning rate δ satisfies the inequality (0.2).

We shall estimate next the error at the jth iteration, $\epsilon_j = |w^{(j)} - w^*|$:

$$\begin{aligned} w^{(j)} - w^* &= M^j w^{(0)} + (\mathbb{I}_n - M^j)w^* - w^* \\ &= M^j(w^{(0)} - w^*). \end{aligned}$$

Using that $|Mx| \leq \|M\||x|$ for all $x \in \mathbb{R}^n$, we have

$$\epsilon_j = |w^{(j)} - w^*| = |M^j(w^{(0)} - w^*)| \leq \|M\|^j |(w^{(0)} - w^*)| = \mu^j d,$$

where $\mu = \|M\|$ is the norm of M (considering M as a linear operator), and $d = |(w^{(0)} - w^*)|$ is the distance from the initial value to the limit. Since the learning rate δ is chosen such that $\mu \in (0, 1)$, then the sequence $\mu^j d \to 0$ as $j \to \infty$.

Gradient estimates The error function $\xi(w) = \mathbb{E}[(Z-Y)^2]$ has to be estimated from measurements, $(x^{(j)}, z^{(j)})$, where ${x^{(j)}}^T = (x_0^{(j)} \cdots x_n^{(j)})$. The empirical error is

$$\widehat{\xi}(w) = \frac{1}{m}\sum_{j=1}^{m}(z^{(j)} - w^T x^{(j)})^2 = \frac{1}{m}\sum_{j=1}^{m}\epsilon_j^2,$$

where $\epsilon_j = z^{(j)} - w^T x^{(j)}$ is the error between the desired value $z^{(j)}$ and the output value $w^T x^{(j)}$.

Crude estimation: uses a single sample error, i.e. the previous sum is replaced by only one term, $\widehat{\xi}(w) = \epsilon_j^2$. In this case the gradient is estimated as

$$\begin{aligned} \nabla_w \widehat{\xi}(w) &= \nabla_w \epsilon_j^2 = 2\epsilon_j \partial_w \epsilon_j = 2\epsilon_j \partial_w (z^{(j)} - w^T x^{(j)}) \\ &= -2\epsilon_j x^{(j)}. \end{aligned}$$

Applying now the gradient descent method, we obtain

$$\begin{aligned} w^{(j+1)} &= w^{(j)} - \delta \nabla_w \widehat{\xi} \\ &= w^{(j)} - \delta(-2\epsilon_j x^{(j)}) = w^{(j)} + 2\delta\epsilon_j x^{(j)}. \end{aligned}$$

Substituting for $\epsilon_j = z^{(j)} - w^T x^{(j)}$, yields

$$w^{(j+1)} = w^{(j)} + 2\delta(z^{(j)} - w^T x^{(j)})x^{(j)}, \qquad (0.3)$$

where $x^{(j)}$ is the jth measurement of the input X.