Artificial neural networks

The Delta rule

- We wish to fit $y(\boldsymbol{x}) = g\left(\boldsymbol{w}^\top \boldsymbol{x}\right)$ to a set of learning examples $\{(\boldsymbol{x}_1, t_1), \ldots, (\boldsymbol{x}_N, t_N)\}$, where $\boldsymbol{x}_n \in \mathbb{R}^d, t_n \in \mathbb{R}$

- Define the (empirical) **mean-square error** of the network as:

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}(t_n - y(\boldsymbol{x}_n))^2 = \frac{1}{2}\sum_{n=1}^{N}\left[t_n - g\left(\sum_{i=1}^{d} w_i x_{n,i} + w_0\right)\right]^2$$

Artificial neural networks

The Delta rule

Let $f : \mathbb{R}^r \to \mathbb{R}$ differentiable; we wish to minimize it by making changes in its variables. Then the increment in each variable is proportional to the corresponding derivative: $x_i(t+1) := x_i(t) + \Delta x_i(t)$, with

$$\Delta x_i(t) = -\alpha \left.\frac{\partial f}{\partial x_i}\right|_{\boldsymbol{x}=\boldsymbol{x}(t)}, \; \alpha > 0, i = 1, \ldots, r$$

Illustration ($r = 1$). Let $f(x) = 3x^2 + x - 4$ and take $\alpha = 0.05$. We have $f'(x) = 6x + 1$. Then $x(0) = 1, x(1) = x(0) - \alpha f'(1) = 1 - 0.05 \cdot 7 = 0.65, x(2) = 0.65 - 0.05 \cdot 4.9 = 0.405, \ldots$. We find $\lim_{i\to\infty} x(i) = -\frac{1}{6}$.

Artificial neural networks

The Delta rule

In our case, the function to be minimized is the empirical error and the variables are the weights $\boldsymbol{w}$ of the network:

$$\Delta w_j(t) = -\alpha \left.\frac{\partial E(\boldsymbol{w})}{\partial w_j}\right|_{\boldsymbol{w}=\boldsymbol{w}(t)}, \ \alpha > 0, \ j = 0, \ldots, d$$

$$\frac{\partial E(\boldsymbol{w})}{\partial w_j} = -\sum_{n=1}^{N} (t_n - y(\boldsymbol{x}_n)) g'(\boldsymbol{w}^\top \boldsymbol{x}_n) x_{n,j}$$

- $t_n - y(\boldsymbol{x}_n)$ is called the **delta**
- $\boldsymbol{w}^\top \boldsymbol{x}_n$ is called the **net input**

Artificial neural networks

The Delta rule

$$\longrightarrow \Delta w_j(t) = \alpha \sum_{n=1}^{N} (t_n - y(\boldsymbol{x}_n)) g'(\boldsymbol{w}(t)^\top \boldsymbol{x}_n) x_{n,j}$$

When g is the identity, we get the α-**LMS Learning Rule**:

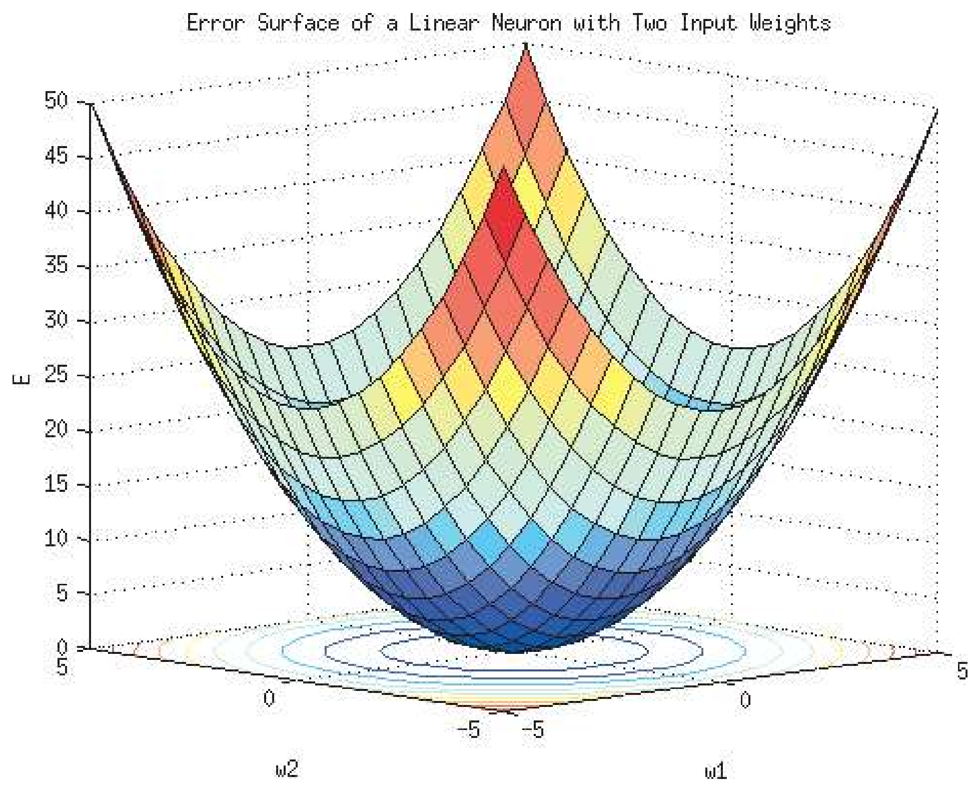
$$\Delta w_j(t) = \alpha \sum_{n=1}^{N} (t_n - y(\boldsymbol{x}_n)) x_{nj} = \alpha \sum_{n=1}^{N} (t_n - \boldsymbol{w}^\top \boldsymbol{x}_n) x_{n,j}$$

- This technique represents a **linear regressor** where the regression coefficients are estimated iteratively (probably the most analyzed and applied **learning rule**)
- This is a form of learning (because of the adaptation to the data) but it is **not incremental**: we need all the examples from the beginning (**"batch"** learning)

Artificial neural networks

The Delta rule

The function $E(\boldsymbol{w})$ is convex in $\boldsymbol{w}$: it defines a convex hyper-paraboloidal surface with a single **global minimum** $\boldsymbol{w}^*$



Artificial neural networks

The Delta rule

1. The constant α controls the stability and speed of convergence. If chosen sufficiently small, the gradient descent procedure asymptotically converges towards $\boldsymbol{w}^*$, regardless of the initial vector $\boldsymbol{w}(0)$

2. A sufficient condition is $0 < \alpha < \frac{2}{\lambda_{max}}$, where λ_{max} is the largest eigenvalue of the input auto-correlation matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] \approx XX^\top$

In practice, one may use $\alpha < \frac{2}{\sum_{n=1}^{N} \|\boldsymbol{x}_n\|^2}$, since $\lambda_{max} < \mathrm{Tr}(E[\boldsymbol{x}\boldsymbol{x}^\top]) \approx \mathrm{Tr}(XX^\top) = \sum_{n=1}^{N} \|\boldsymbol{x}_n\|^2$

Artificial neural networks

The Delta rule

In the **on-line** version, we also begin with $\boldsymbol{w}(0)$ arbitrary and apply:

$$\Delta w_j(t) = \alpha_t \Big(t_{n(t)} - y(\boldsymbol{x}_{n(t)}) \Big) x_{n(t),j}$$

- At each step t, the example $n(t)$ is drawn at random from $\{1, \ldots, N\}$
- It can be shown that if $\sum\limits_{t \geq 0} \alpha_t = \infty$ and $\sum\limits_{t \geq 0} \alpha_t^2 < \infty$, then $\boldsymbol{w}(t)$ converges to the **global minimum** $\boldsymbol{w}^*$ in the mean square sense:

$$\lim_{t \to \infty} \left\| \boldsymbol{w}(t) - \boldsymbol{w}^* \right\|^2 = 0$$

One such procedure is $\alpha_t = \frac{\alpha}{t+1}$, with $\alpha > 0$