Generalized Backpropagation

(Beta version 1.0)

Daniel Crespin
Facultad de Ciencias
Universidad Central de Venezuela

A Roberto Pérez Cabrera
por su admirable constructividad

Abstract

Global backpropagation formulas for differentiable neural networks are considered from the viewpoint of minimization of the quadratic error using the gradient method. The gradient of (the quadratic error function of) a processing unit is expressed in terms of the output error and the transposed derivative of the unit with respect to the weight. The gradient of the layer is the product of the gradients of the processing units. The gradient of the network equals the product of the gradients of the layers. Backpropagation provides the desired outputs or targets for the layers. Standard formulas for semilinear networks are deduced as a special case.

1.- **Introduction.** Let W be an open set in $\mathbf{R}^M$ and $Q : W \to \mathbf{R}$ a differentiable map. To obtain sequences $w^{(0)}, w^{(1)}, \ldots, w^{(k)}, \ldots$ with $Q(w^{(0)}) > Q(w^{(1)}) > \cdots > Q(w^{(k)}) \cdots$, the following two algorithms are often used.

Gradient algorithm: $w^{(k+1)} = w^{(k)} - \epsilon \nabla Q(w^{(k)})$.

Steepest descent algorithm: $w^{(k+1)}$ =point on the line $w^{(k)} + t\nabla Q(w^{(k)})$, $t \in \mathbf{R}$, where Q attains its minimum.

Numerous technical details are involved in the correct and effective application of these iterations but the calculation of the gradient is a basic step.

Problems solved with neural networks usually involve a set of inputs, a corresponding set of desired outputs or targets and a real valued error function

$Q(w)$ that measures, for a parameter $w \in \mathbf{R}^M$, how much the network misses the targets. The above algorithms are first choice procedures to obtain, starting with certain $w^{(0)}$, better parameter values $w^{(k)}$. See [10] for a treatment of the gradient method that leads to backpropagation in semilinear networks. On the other hand the steepest descent algorithm does not seem to have been much used for neural networks but could be useful.

In the present paper the calculation of ∇Q is carried out for general differentiable neural networks within the framework given in [1]. Quadratic error functions are used but the same procedure may be easily adapted to non-quadratic error functions, the ones associated with Minkowski metric $(\sum x_i^p)^{\frac{1}{p}}$ say, and the results could be of interest in non-linear regression.

The formulas obtained are global: Inputs and processing units of each layer are subject not to individual treatment but to typical explicit vector expressions that include all simultaneous components. This gives results in principle suited for vectorial processing. Gradients of (the quadratic error function of) processing units are expressed in terms of the output error and the transposed derivative of the unit with respect to w; gradients of layers are products of gradients of processing units; and the gradient of the network equals the product of the gradients of the layers. Backpropagation provides the desired outputs or targets for the layers preceding the last one. In particular, the explicit global formulas for semilinear networks are obtained. According to [5] perceptron neural networks that perform a given pattern recognition task can be constructed directly from the data provided; training is not needed. But in practical applications training, and backpropagation in particular, could still be useful to fine tune the weights. Hence, despite the direct methods, this paper may also be of practical interest.

2.- **Terminology and notation.** As a reference for elementary calculus consult [7] or [8]. Properties of transposes of linear maps can be found in [9]. For the neural network formalism adopted here see [1]. The following notation will be used. If $x = (x_1, \ldots, x_m)$, $x' = (x'_1, \ldots, x'_m)$ are in $\mathbf{R}^m$ then $\langle x, x' \rangle = \sum_{i=1}^m x_i x'_i$. Any linear map $L : \mathbf{R}^m \to \mathbf{R}$ has the form $L(x) = \langle a, x \rangle$ for a uniquely determined $a \in \mathbf{R}^m$. The transpose of the linear map $L : \mathbf{R}^m \to \mathbf{R}^n$ is $L^* : \mathbf{R}^n \to \mathbf{R}^m$ defined by the condition $\langle L(x), y \rangle = \langle x, L^*(y) \rangle$.

In particular if $L : \mathbf{R}^m \to \mathbf{R}$ is given by $L(x) = \langle a, x\rangle$ then $L^* : \mathbf{R} \to \mathbf{R}^m$ is given by $L^*(y) = ya$; also, $L^*(1) = a$. Matrices of linear maps will always be taken with respect to the canonical bases of euclidean spaces. If L has a $n \times m$ matrix (m_{ji}) with n rows and m columns then the matrix of L^* is the transposed $m \times n$ matrix $(m_{ji})^* = (m^*_{ij})$ with $m^*_{ij} = m_{ji}$.

Let $U \subseteq \mathbf{R}^m$ be open and let $f : U \to \mathbf{R}^n$ be differentiable at $\bar{x} \in U$; the derivative of f at $\bar{x}$ is a linear map $Df(\bar{x}) : \mathbf{R}^m \to \mathbf{R}^n$ whose value at $dx \in \mathbf{R}^m$ will be denoted $Df(\bar{x}) \cdot dx$. Let $f = (f_1, \ldots, f_n)$ with $f_j : U \to \mathbf{R}$. The matrix of $Df(\bar{x})$ is the jacobian of f at $\bar{x}$, $Jf(\bar{x}) = (\frac{\partial f_j}{\partial x_i}(\bar{x}))$, and the transposed derivative $D^*f(\bar{x}) : \mathbf{R}^n \to \mathbf{R}^m$ has matrix the transposed jacobian $J^*f(\bar{x}) = (\frac{\partial f_j}{\partial x_i}(\bar{x}))^*$. The gradient of f at $\bar{x}$ is by definition the vector $\nabla f(\bar{x}) = (\frac{\partial f}{\partial x_1}(\bar{x}), \ldots, \frac{\partial f}{\partial x_m}(\bar{x}))$; if the derivative of f at $\bar{x}$ can be expressed as $Df(\bar{x}) \cdot dx = \langle a, dx\rangle$ for some $a \in \mathbf{R}^M$ then necessarily $a = \nabla f(\bar{x})$. Except if otherwise indicated all maps will be assumed differentiable.

Let V be open in $\mathbf{R}^M \times \mathbf{R}^m$, $(dw, dx) \in \mathbf{R}^M \times \mathbf{R}^m = \mathbf{R}^{M+m}$ and $f : V \to \mathbf{R}^n$ differentiable. The partial derivatives of f at $(\bar{w}, \bar{x}) \in V$ with respect to $w \in \mathbf{R}^M$ and $x \in \mathbf{R}^m$ are the linear maps $D_w f(\bar{w}, \bar{x}) : \mathbf{R}^M \to \mathbf{R}^n$ and $D_x f(\bar{w}, \bar{x}) : \mathbf{R}^m \to \mathbf{R}^n$ defined by the expression $Df(\bar{w}, \bar{x}) \cdot (dw, dx) = (D_w f(\bar{w}, \bar{x}) \cdot dw, D_x f(\bar{w}, \bar{x}) \cdot dx)$. Their matrices are the $n \times M$ matrix $J_w f(\bar{w}, \bar{x}) = (\frac{\partial f_j}{\partial w_i}(\bar{w}, \bar{x}))$ and the $n \times m$ matrix $J_x f(\bar{w}, \bar{x}) = (\frac{\partial f_j}{\partial x_i}(\bar{w}, \bar{x}))$; these are composed from the first M and last m columns of $Jf(\bar{w}, \bar{x})$.

3.- **Errors.** Assume W is open in $\mathbf{R}^M$ and that $f : W \times \mathbf{R}^m \to \mathbf{R}^n$ is a differentiable parametric map; see [1]. Recall that $f_w(x) = f(w, x)$. Figure 1 illustrates f and its partials.

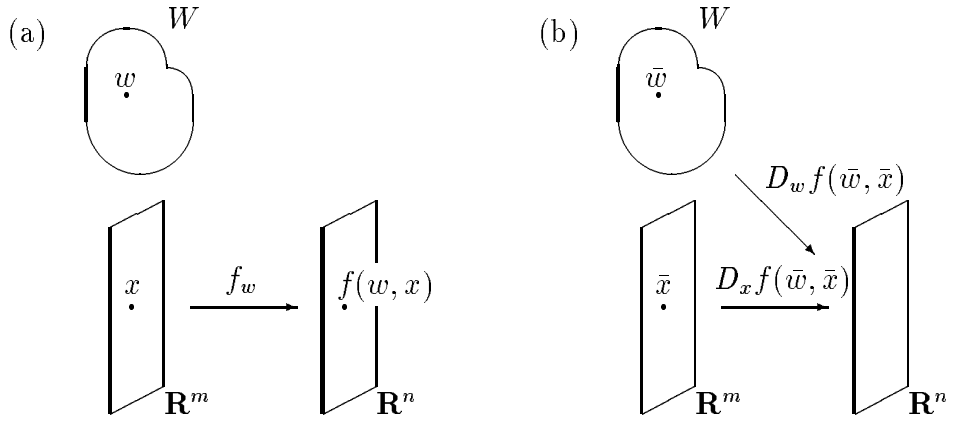

Figure 1: (a) Parametric map $f(w, x)$; (b) partial derivatives.

For fixed $\bar{x} \in \mathbf{R}^m$ as *input* and $\bar{y} \in \mathbf{R}^n$ as *desired output* or *target*, the *output error* is $\delta : W \to \mathbf{R}^n$ given by $\delta(w) = f(w, \bar{x}) - \bar{y}$, and the *quadratic error* is the function $Q : W \to \mathbf{R}$ defined by $Q(w) = \langle f(w, \bar{x}) - \bar{y}, f(w, \bar{x}) - \bar{y} \rangle = \langle \delta(w), \delta(w) \rangle$. The derivative of Q at $\bar{w}$ is the linear form in $\mathbf{R}^m$ given by

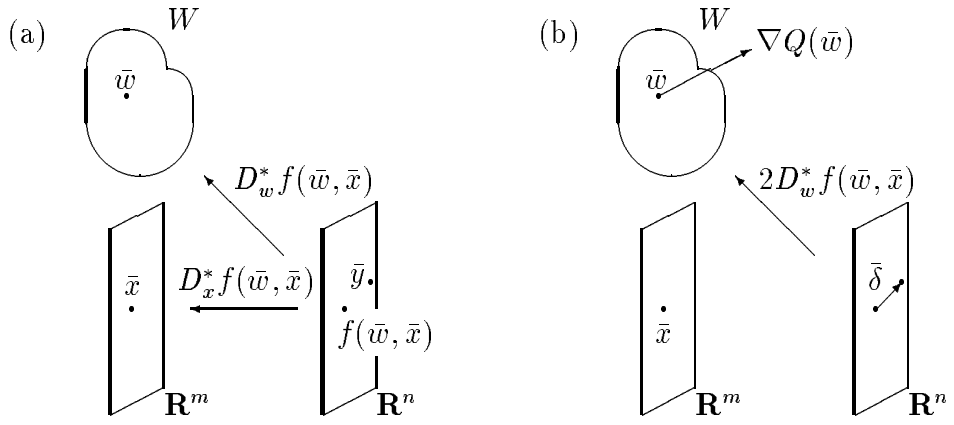

Figure 2: (a) Transposed partial derivatives of f; (b) gradient of Q

$DQ(\bar{w}) \cdot dw = 2\langle D_w f(\bar{w}, \bar{x}) \cdot dw, f(\bar{w}, \bar{x}) - \bar{y} \rangle$. Taking the transpose of the partial derivative of f this gives $DQ(\bar{w}) \cdot dw = \langle 2D_w^* f(\bar{w}, \bar{x}) \cdot [f(\bar{w}, \bar{x}) - \bar{y}], dw \rangle$, hence

$$\nabla Q(\bar{w}) = 2D_w^* f(\bar{w}, \bar{x}) \cdot \bar{\delta} \tag{1}$$

where $\bar{\delta} = \delta(\bar{w})$. This formula is basic and the rest of the paper concerns its application to calculate the gradients of quadratic errors.

4.- **Real valued parametric maps.** Assume $n = 1$, that is, let W be open in $\mathbf{R}^M$ and $f : W \times \mathbf{R}^m \to \mathbf{R}$ a differentiable map. The error $\bar{\delta}$ is now a real number and according to equation 1 above the gradient can be written as

$$\nabla Q(\bar{w}) = 2\bar{\delta} D_w^* f(\bar{w}, \bar{x}) \cdot 1 \tag{2}$$

5.- **Real valued paired maps.** In case $W = \mathbf{R}^m$ with $f : \mathbf{R}^m \times \mathbf{R}^m \to \mathbf{R}$ a paired map (see [1]) one has by definition $f(w_1, \ldots, w_m, x_1, \ldots, x_m) =$

$\phi(\xi_1(w_1,x_1),\ldots,\xi_m(w_m,x_m))$ so that $\frac{\partial \xi_{i'}}{\partial w_i}=0$ except for $i'=i$. Will calculate $\nabla Q(\bar{w})$ in terms of partial derivatives. Let $\bar{\xi}_i=\xi_i(\bar{w}_i,\bar{x}_i)$ and $\bar{\xi}=(\bar{\xi}_1,\ldots,\bar{\xi}_m)$ then

$$\frac{\partial f}{\partial w_i}(\bar{w},\bar{x})=\sum_{i'=1}^{m}\frac{\partial \phi}{\partial \xi_{i'}}(\bar{\xi}_1,\ldots,\bar{\xi}_m)\frac{\partial \xi_{i'}}{\partial w_i}(\bar{w}_i,\bar{x}_i)=\frac{\partial \phi}{\partial \xi_i}(\bar{\xi}_1,\ldots,\bar{\xi}_m)\frac{\partial \xi_i}{\partial w_i}(\bar{w}_i,\bar{x}_i)$$

and therefore for $dw=(dw_1,\cdots,dw_m)$

$$D_w f(\bar{w},\bar{x})\cdot dw=\sum_{i=1}^{m}\frac{\partial \phi}{\partial \xi_i}(\bar{\xi}_1,\ldots,\bar{\xi}_m)\frac{\partial \xi_i}{\partial w_i}(\bar{w}_i,\bar{x}_i)dw_i$$

thus, with $\bar{\delta}=f(\bar{w},\bar{x})-\bar{y}$, equation 1 implies

$$\nabla Q(\bar{w})=2\bar{\delta}\,(\frac{\partial \phi}{\partial \xi_1}(\bar{\xi}_1,\ldots,\bar{\xi}_m)\frac{\partial \xi_1}{\partial w_1}(\bar{w}_1,\bar{x}_1),\ldots,$$

$$\frac{\partial \phi}{\partial \xi_m}(\bar{\xi}_1,\ldots,\bar{\xi}_m)\frac{\partial \xi_m}{\partial w_m}(\bar{w}_m,\bar{x}_m))\qquad(3)$$

Note also the formulas

$$\frac{\partial f}{\partial x_i}(\bar{w},\bar{x})=\frac{\partial \phi}{\partial \xi_i}(\bar{\xi}_1,\ldots,\bar{\xi}_m)\frac{\partial \xi_i}{\partial x_i}(\bar{w}_i,\bar{x}_i)$$

$$D_x f(\bar{w},\bar{x})\cdot dw=\sum_{i=1}^{m}\frac{\partial \phi}{\partial \xi_i}(\bar{\xi}_1,\ldots,\bar{\xi}_m)\frac{\partial \xi_i}{\partial x_i}(\bar{w}_i,\bar{x}_i)dx_i$$

$$D_x^* f(\bar{w},\bar{x})\cdot\bar{\delta}=\bar{\delta}\,(\frac{\partial \phi}{\partial \xi_1}(\bar{\xi}_1,\ldots,\bar{\xi}_m)\frac{\partial \xi_1}{\partial x_1}(\bar{w}_1,\bar{x}_1),\ldots,$$

$$\frac{\partial \phi}{\partial \xi_m}(\bar{\xi}_1,\ldots,\bar{\xi}_m)\frac{\partial \xi_m}{\partial x_m}(\bar{w}_1,\bar{x}_m))\qquad(4)$$

6.- **Semilinear maps.** Assume now that $f:\mathbf{R}^{m+1}\times\mathbf{R}^m\to\mathbf{R}$ is semilinear, that is, $f(w_0,w_1,\ldots,w_m,x_1,\ldots,x_m)=\psi(w_0+w_1x_1+\cdots+w_mx_m)$. For $x=(x_1,\ldots,x_m)$ let $(\alpha,x)=(\alpha,x_1,\ldots,x_m)$ and let $w=(w_0,w_1,\ldots,w_m)$ so that, for example, $\langle w,(\alpha,x)\rangle=w_0\alpha+w_1x_1+\cdots+w_mx_m=\langle(\alpha,x),w\rangle$. Then the semilinear map can be written as $f(w,x)=\psi(\langle w,(1,x)\rangle)$. Either by direct application of the chain rule or from the formulas of the previous section

(with minor modifications due to the bias w_0), for $dw = (dw_0, dw_1, \ldots, dw_m)$ the following results

$$D_w f(\bar{w}, \bar{x}) \cdot dw = \psi'(\langle \bar{w}, (1, \bar{x}) \rangle) \, (dw_0 + \bar{x}_1 dw_1 + \cdots + \bar{x}_m dw_m)$$

$$D_w^* f(\bar{w}, \bar{x}) \cdot \bar{\delta} = \psi'(\langle \bar{w}, (1, \bar{x}) \rangle) \, \bar{\delta} \, (1, \bar{x}_1, \ldots, \bar{x}_m)$$

implying

$$\nabla Q(f)(\bar{w}) = 2\psi'(\langle \bar{w}, (1, \bar{x}) \rangle) \, \bar{\delta} \, (1, \bar{x}_1, \ldots, \bar{x}_m) \tag{5}$$

where ψ' denotes the derivative of ψ. Note also that

$$D_x f(\bar{w}, \bar{x}) \cdot dx = \psi'(\langle \bar{w}, (1, \bar{x}) \rangle) \, (\bar{w}_1 dx_1 + \cdots + \bar{w}_m dx_m)$$

$$D_x^* f(\bar{w}, \bar{x}) \cdot \bar{\delta} = \psi'(\langle \bar{w}, (1, \bar{x}) \rangle) \, \bar{\delta} \, (\bar{w}_1, \ldots, \bar{w}_m) \tag{6}$$

7.- **Vector valued parametric maps.** Consider now $f : W \times \mathbf{R}^m \to \mathbf{R}^n$ so that $f = (f_1, \ldots, f_n)$ with $f_j : W \times \mathbf{R}^m \to \mathbf{R}$, and let $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_n)$, $\bar{\delta} = (\bar{\delta}_1, \ldots, \bar{\delta}_n) = f(\bar{w}, \bar{x}) - \bar{y} = (f_1(\bar{w}, \bar{x}) - \bar{y}_1, \ldots, f_n(\bar{w}, \bar{x}) - \bar{y}_n)$. In this case $D_w f(\bar{w}, \bar{x}) \cdot dw = (D_w f_1(\bar{w}, \bar{x}) \cdot dw, \ldots, D_w f_n(\bar{w}, \bar{x}) \cdot dw)$ and therefore $D_w^* f(\bar{w}, \bar{x}) \cdot \bar{\delta} = D_w^* f_1(\bar{w}, \bar{x}) \cdot \bar{\delta}_1 + \cdots + D_w^* f_n(\bar{w}, \bar{x}) \cdot \bar{\delta}_n$. On the other hand the quadratic errors of the component functions f_j with targets $\bar{y}_j$ are $Q(f_j)(\bar{w}) = 2D_w^* f_j(\bar{w}, \bar{x}) \cdot \bar{\delta}_j$. Formula 1 applies and gives

$$\nabla Q(\bar{w}) = \nabla Q(f_1)(\bar{w}) + \cdots + \nabla Q(f_n)(\bar{w}) \tag{7}$$

In words, the gradient of the quadratic error of a vector valued parametric map equals the sum of the gradients of corresponding quadratic errors of the components. For the partial with respect to x and its transpose the formulas are

$$D_x f(\bar{w}, \bar{x}) \cdot dx = (D_x f_1(\bar{w}, \bar{x}) \cdot dx, \ldots, D_x f_n(\bar{w}, \bar{x}) \cdot dx)$$

$$D_x^* f(\bar{w}, \bar{x}) \cdot \bar{\delta} = D_x^* f_1(\bar{w}, \bar{x}) \cdot \bar{\delta}_1 + \cdots + D_x^* f_q(\bar{w}, \bar{x}) \cdot \bar{\delta}_q \tag{8}$$

8.- **Products.** The integers $m, M_1, \ldots, M_q, n_1, \ldots, n_q$ are positive, $M = M_1 + \cdots + M_q$, $n = n_1 + \cdots + n_q$. Let W_j be open in $\mathbf{R}^{M_j}$, $W = W_1 \times \cdots \times W_q \subseteq \mathbf{R}^{M_1} \times \cdots \times \mathbf{R}^{M_q} = \mathbf{R}^M$. Consider differentiable maps $f_j : W_j \times \mathbf{R}^m \to \mathbf{R}^{n_j}$ and their parametric product $f = f_1 \mathbin{\widehat{\times}} \cdots \mathbin{\widehat{\times}} f_q : W \times \mathbf{R}^m \to \mathbf{R}^n$ defined by the formula $f(w_1, \ldots, w_q, x) = (f_1(w_1, x), \ldots, f_q(w_q, x))$. For given input

$\bar{x} \in \mathbf{R}^m$, and target $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_q) \in \mathbf{R}^n$ let $\delta(w) = (\delta_1(w), \ldots, \delta_q(w)) = f(w, \bar{x}) - \bar{y} = (f_1(w_1, \bar{x}) - \bar{y}_1, \ldots, f_q(w_q, \bar{x}) - \bar{y}_q)$. The quadratic error of f satisfies $Q(f)(w) = \sum_{j=1}^{q} \langle f_j(w_j, \bar{x}) - \bar{y}_j, f_j(w_j, \bar{x}) - \bar{y}_j \rangle = \sum_{j=1}^{q} Q(f_j)(w_j)$. Taking a fixed $\bar{w} = (\bar{w}_1, \ldots, \bar{w}_q) \in W$ and letting $\bar{\delta} = (\bar{\delta}_1, \ldots, \bar{\delta}_q) = \delta(\bar{w}) = (f_1(\bar{w}_1, \bar{x}) - \bar{y}_1, \ldots, f_q(\bar{w}_q, \bar{x}) - \bar{y}_q)$ derivatives can be taken componentwise and it is obvious that

$$D_w f(\bar{w}, \bar{x}) \cdot dw = (D_{w_1} f_1(\bar{w}_1, \bar{x}) \cdot dw_1, \ldots, D_{w_q} f_q(\bar{w}_q, \bar{x}) \cdot dw_q)$$

Therefore $D_w^* f(\bar{w}, \bar{x}) \cdot \bar{\delta} = (D_{w_1}^* f_1(\bar{w}_1, \bar{x}) \cdot \bar{\delta}_1, \ldots, D_{w_q}^* f_q(\bar{w}_q, \bar{x}) \cdot \bar{\delta}_q)$, so $\nabla Q(\bar{w}) = (2D_{w_1}^* f_1(\bar{w}_1, \bar{x}) \cdot \bar{\delta}_1, \ldots, 2D_{w_q}^* f_q(\bar{w}_q, \bar{x}) \cdot \bar{\delta}_q)$ and this is the same as

$$\nabla Q(f)(\bar{w}) = (\nabla Q(f_1)(\bar{w}_1), \ldots, \nabla Q(f_q)(\bar{w}_q)) \tag{9}$$

Thus, for a parametric product the quadratic error has gradient equal to the product of the gradients of the quadratic errors of the factors. Similarly, $D_x f(\bar{w}, \bar{x}) \cdot dx = (D_x f_1(\bar{w}_1, \bar{x}) \cdot dx, \ldots, D_x f_q(\bar{w}_q, \bar{x}) \cdot dx)$ from where

$$D_x^* f(\bar{w}, \bar{x}) \cdot \bar{\delta} = D_x^* f_1(\bar{w}_1, \bar{x}) \cdot \bar{\delta}_1 + \cdots + D_x^* f_q(\bar{w}_q, \bar{x}) \cdot \bar{\delta}_q \tag{10}$$

9.- **Semilinear products.** An additional subindex has to be added to the notation of section 6. In the case of products of semilinear real valued parametric factors $f_j(w_{j0}, w_{j1}, \ldots, w_{jm}, x_1, \ldots, x_m) = \psi_j(w_{j0} + w_{j1}x_1 + \cdots + w_{jm}x_m)$ assume for simplicity that a unique threshold function $\psi = \psi_j$, $j = 1, \ldots, q$, is involved. Let $w_j = (w_{j0}, w_{j1}, \ldots, w_{jm})$, $w = (w_0, w_1, \ldots, w_m)$, $dw_j = (dw_{j0}, dw_{j1}, \ldots, dw_{jm})$ and $dw = (dw_1, \ldots, dw_q)$. The parametric product of semilinear functions is then $f = f_1 \widehat{\times} \cdots \widehat{\times} f_q : \mathbf{R}^{m+1} \times \overbrace{\cdots}^{q} \times \mathbf{R}^{m+1} \times \mathbf{R}^m = \mathbf{R}^{q(m+1)} \times \mathbf{R}^m \to \mathbf{R}^q$

$$\begin{aligned} f(w, x) &= f(w_1, \ldots, w_q, x) \\ &= (f_1(w_1, x), \ldots, f_q(w_q, x)) \\ &= (\psi(\langle w_1, (1, x) \rangle), \ldots, \psi(\langle w_q, (1, x) \rangle)) \end{aligned}$$

For notational convenience let $\bar{s}_j = \langle \bar{w}_j, (1, \bar{x}) \rangle = \bar{w}_{j0} + \bar{w}_{j1}\bar{x}_1 + \cdots + \bar{w}_{jm}\bar{x}_m$. From formulas 2 and 5 the quadratic error $Q(f)$ has gradient

$$\nabla Q(f)(\bar{w}) = 2(\psi'(\bar{s}_1)\,\bar{\delta}_1\,(1, \bar{x}_1, \ldots, \bar{x}_m), \cdots, \psi'(\bar{s}_q)\,\bar{\delta}_q\,(1, \bar{x}_1, \ldots, \bar{x}_m)) \tag{11}$$

If matrix notation is preferred define the *term by term* product of $n \times m$ matrices (a_{ji}) and (b_{ji}) as the $n \times m$ matrix $(a_{ji})\#(b_{ji}) = (a_{ji}b_{ji})$. It is then possible to write

$$\nabla Q(f)(\bar{w}) = 2 \begin{pmatrix} \psi'(\bar{s}_1)\,\bar{\delta}_1 & \psi'(\bar{s}_1)\,\bar{x}_1\,\bar{\delta}_1 & \cdots & \psi'(\bar{s}_1)\,\bar{x}_m\,\bar{\delta}_1 \\ \vdots & \vdots & & \vdots \\ \psi'(\bar{s}_q)\,\bar{\delta}_q & \psi'(\bar{s}_q)\,\bar{x}_1\,\bar{\delta}_q & \cdots & \psi'(\bar{s}_q)\,\bar{x}_m\,\bar{\delta}_q \end{pmatrix} =$$

$$2 \begin{pmatrix} \psi'(\bar{s}_1) & \cdots & \psi'(\bar{s}_1) \\ \vdots & & \vdots \\ \psi'(\bar{s}_q) & \cdots & \psi'(\bar{s}_q) \end{pmatrix} \# \begin{pmatrix} 1 & \bar{x}_1 & \cdots & \bar{x}_m \\ \vdots & \vdots & & \vdots \\ 1 & \bar{x}_1 & \cdots & \bar{x}_m \end{pmatrix} \# \begin{pmatrix} \bar{\delta}_1 & \cdots & \bar{\delta}_1 \\ \vdots & & \vdots \\ \bar{\delta}_q & \cdots & \bar{\delta}_q \end{pmatrix} \tag{12}$$

All matrices in this term by term product are $q \times (m+1)$ matrices. For the partial with respect to x equations 6 and 10 give

$$D_x f(\bar{w}, \bar{x}) \cdot dx = (\psi'(\bar{s}_1)\,\langle \bar{w}_1, (0, dx)\rangle, \ldots, \psi'(\bar{s}_q)\,\langle \bar{w}_q, (0, dx)\rangle)$$

and for the transpose

$$D_x^* f(\bar{w}, \bar{x}) \cdot \bar{\delta} = (\langle(\psi'(\bar{s}_1)\bar{w}_{11}, \ldots, \psi'(\bar{s}_1)\bar{w}_{q1}), \bar{\delta}\rangle, \ldots, \langle(\psi'(\bar{s}_q)\bar{w}_{1m}, \ldots, \psi'(\bar{s}_q)\bar{w}_{qm}), \bar{\delta}\rangle) \tag{13}$$

An equivalent matricial expression for 13 is

$$D_x^* f(\bar{w}, \bar{x}) \cdot \bar{\delta} = \begin{pmatrix} \bar{w}_{11} & \cdots & \bar{w}_{q1} \\ \vdots & & \vdots \\ \bar{w}_{1m} & \cdots & \bar{w}_{qm} \end{pmatrix} \begin{pmatrix} \psi'(\bar{s}_1) & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \psi'(\bar{s}_q) \end{pmatrix} \begin{pmatrix} \bar{\delta}_1 \\ \vdots \\ \bar{\delta}_q \end{pmatrix} \tag{14}$$

here the second matrix is $q \times q$ diagonal.

10.- **Compositions.** Consider open sets $W^k \subseteq \mathbf{R}^{M_k}$, $k = 1, \ldots, p$, $W = W^1 \times \cdots \times W^p$, and differentiable parametric maps $f^k : W^k \times \mathbf{R}^{n_k} \to \mathbf{R}^{n_{k+1}}$ with parametric composition $f = f^p \,\hat{\circ}\, \ldots \,\hat{\circ}\, f^1 : W \times \mathbf{R}^{n_1} \to \mathbf{R}^{n_{p+1}}$. Recall that for a given *first input* $x = x^1 = (x_1^1, \ldots, x_{n_1}^1) \in \mathbf{R}^{n_1}$ this is recursively defined

by the expression $f(w^1, \dots w^p, x) = f^p(w^p, f^{p-1}\hat{\circ}\cdots\hat{\circ}f^1(w^1, \dots, w^{p-1}, x))$; see [1] and Figure 3 below.

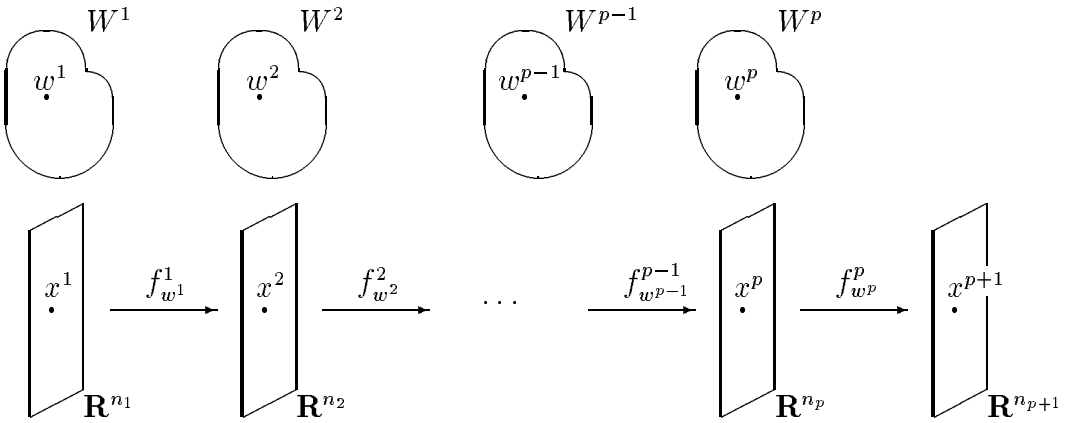

Figure 3: Neural network $f^p\hat{\circ}\cdots\hat{\circ}f^1$

Let $M = M_1 + \cdots + M_p$, $\bar{w} = (\bar{w}^1, \dots, \bar{w}^p) \in W$, $dw = (dw^1, \dots, dw^p) \in \mathbf{R}^M$ take $\bar{x} = \bar{x}^1 = (\bar{x}^1_1, \dots, \bar{x}^1_{n_1}) \in \mathbf{R}^{n_1}$ and define, for $k = 1, \cdots, p$, $\bar{x}^{k+1} = f^k(\bar{w}^k, \bar{x}^k)$. The point $\bar{x}^k$ is the k-th *input* in $\mathbf{R}^{n_k}$; see Figure 4. The chain rule implies the following formulas for the derivative of the parametric composition with respect to the parameter.

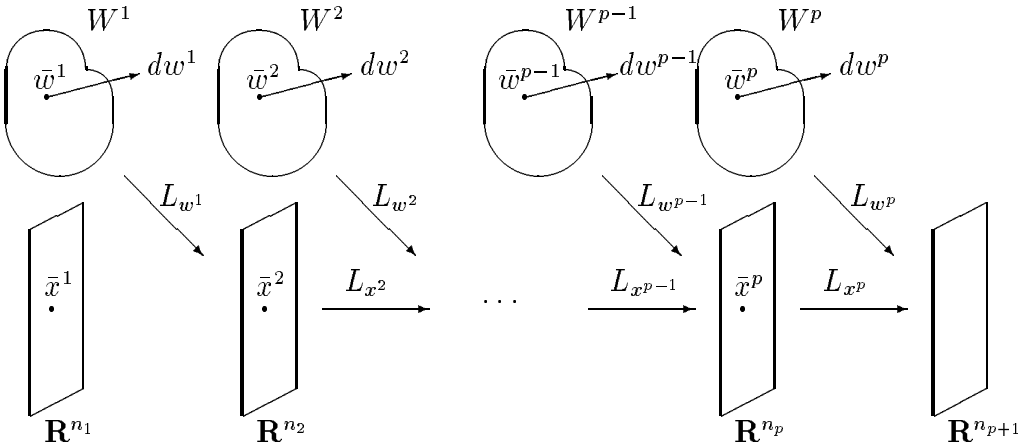

Figure 4: Partial derivative $D_w(f^p\hat{\circ}\cdots\hat{\circ}f^1)(\bar{w}, \bar{x}^1)\cdot dw$

$$
\begin{aligned}
&D_w f(\bar{w},\bar{x}^1)\cdot dw = \\
&\quad D_{x^p} f^p(\bar{w}^p,\bar{x}^p)\circ D_{x^{p-1}} f^{p-1}(\bar{w}^{p-1},\bar{x}^{p-1})\circ\cdots\circ D_{x^2} f^2(\bar{w}^2,\bar{x}^2)\circ \\
&\qquad\qquad D_{w^1} f^1(\bar{w}^1,\bar{x}^1)\cdot dw^1 \\
&\quad + D_{x^p} f^p(\bar{w}^p,\bar{x}^p)\circ D_{x^{p-1}} f^{p-1}(\bar{w}^{p-1},\bar{x}^{p-1})\circ\cdots\circ D_{x^3} f^3(\bar{w}^3,\bar{x}^3)\circ \\
&\qquad\qquad D_{w^2} f^2(\bar{w}^2,\bar{x}^2)\cdot dw^2 \\
&\quad +\cdots \\
&\quad + D_{x^p} f^p(\bar{w}^p,\bar{x}^p)\circ D_{x^{p-1}} f^{p-1}(\bar{w}^{p-1},\bar{x}^{p-1})\circ D_{w^{p-2}} f^{p-2}(\bar{w}^{p-2},\bar{x}^{p-2})\cdot dw^{p-2} \\
&\quad + D_{x^p} f^p(\bar{w}^p,\bar{x}^p)\circ D_{w^{p-1}} f^{p-1}(\bar{w}^{p-1},\bar{x}^{p-1})\cdot dw^{p-1} \\
&\quad + D_{w^p} f^p(\bar{w}^p,\bar{x}^p)\cdot dw^p
\end{aligned}
$$

In more compact notation let $L_{w^j} = D_{w^j} f^j(\bar{w}^j,\bar{x}^j)$ and $L_{x^j} = D_{x^j} f^j(\bar{w}^j,\bar{x}^j)$; the transposes $L^*_{w^j} = D^*_{w^j} f^j(\bar{w}^j,\bar{x}^j)$ and $L^*_{x^j} = D^*_{x^j} f^j(\bar{w}^j,\bar{x}^j)$ will be used later. Then

$$
\begin{aligned}
D_w f(\bar{w},\bar{x}^1)\cdot dw &= L_{x^p}\circ L_{x^{p-1}}\circ\cdots\circ L_{x^2}\circ L_{w^1}\cdot dw^1 \\
&+ L_{x^p}\circ L_{x^{p-1}}\circ\cdots\circ L_{x^3}\circ L_{w^2}\cdot dw^2 \\
&+ \cdots \\
&+ L_{x^p}\circ L_{x^{p-1}}\circ L_{w^{p-2}}\cdot dw^{p-2} \\
&+ L_{x^p}\circ L_{w^{p-1}}\cdot dw^{p-1} \\
&+ L_{w^p}\cdot dw^p
\end{aligned}
$$

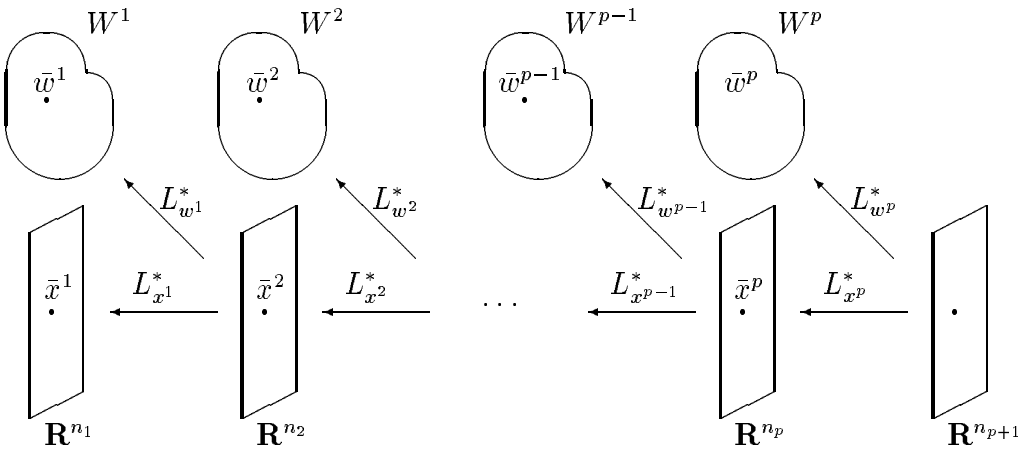

Figure 5: Transposes of partial derivatives

The gradient of the quadratic error $Q = Q(f)$ has p components: $\nabla Q(w) = (\nabla^{(1)}Q(w), \ldots, \nabla^{(p)}Q(w)) \in \mathbf{R}^M$; these can be calculated taking transposes in the previous formulas. For this, let $\bar{y} = \bar{y}^{p+1} \in \mathbf{R}^{n_{p+1}}$ be a *desired final output* or *final target* and define the *final error* as $\delta(w) = f(w, \bar{x}) - \bar{y} = \bar{x}^{p+1} - \bar{y}^{p+1}$. The quadratic error $Q = Q(f)(w) = \langle \delta(w), \delta(w) \rangle$ is then a function of w and if $\bar{\delta} = \bar{\delta}^{p+1} = \delta(\bar{w}) = f(\bar{w}, \bar{x}) - \bar{y}$,

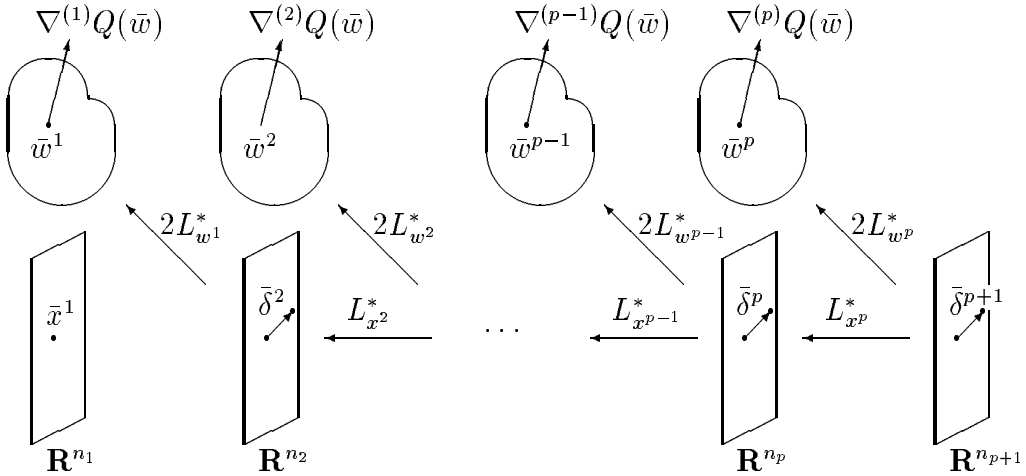

Figure 6: Gradient of quadratic error: $\nabla Q(f^p \hat{\circ} \cdots \hat{\circ} f^1)(\bar{w})$.

$$\nabla^{(p)}Q(\bar{w}) = 2D^*_{w^p} f^p\,(\bar{w}^p, \bar{x}^p) \cdot \bar{\delta}^{p+1}$$

$$\nabla^{(p-1)}Q(\bar{w}) = 2D^*_{w^{p-1}} f^{p-1}\,(\bar{w}^{p-1}, \bar{x}^{p-1}) \circ D^*_{x^p} f^p\,(\bar{w}^p, \bar{x}^p) \cdot \bar{\delta}^{p+1}$$

$$\nabla^{(p-2)}Q(\bar{w}) = 2D^*_{w^{p-2}} f^{p-2}\,(\bar{w}^{p-2}, \bar{x}^{p-2}) \circ D^*_{x^{p-1}} f^{p-1}\,(\bar{w}^{p-1}, \bar{x}^{p-1}) \circ D^*_{x^p} f^p\,(\bar{w}^p, \bar{x}^p) \cdot \bar{\delta}^{p+1}$$

$$\vdots$$

$$\nabla^{(2)}Q(\bar{w}) = 2D^*_{w^2} f^2(\bar{w}^2, \bar{x}^2) \circ D^*_{x^3} f^3\,(\bar{w}^3, \bar{x}^3) \circ \cdots \circ D^*_{x^{p-1}} f^{p-1}\,(\bar{w}^{p-1}, \bar{x}^{p-1}) \circ D^*_{x^p} f^p\,(\bar{w}^p, \bar{x}^p) \cdot \bar{\delta}^{p+1}$$

$$\nabla^{(1)}Q(\bar{w}) = 2D^*_{w^1} f^1(\bar{w}^1, \bar{x}^2) \circ D^*_{x^2} f^2\,(\bar{w}^2, \bar{x}^2) \circ \cdots \circ D^*_{x^{p-1}} f^{p-1}\,(\bar{w}^{p-1}, \bar{x}^{p-1}) \circ D^*_{x^p} f^p\,(\bar{w}^p, \bar{x}^p) \cdot \bar{\delta}^{p+1}$$

Or, with compact notation,

$$\begin{aligned}
\nabla^{(p)}Q(\bar{w}) &= 2L^*_{w^p}\cdot\bar{\delta}^{p+1}\\
\nabla^{(p-1)}Q(\bar{w}) &= 2L^*_{w^{p-1}}\circ L^*_{x^p}\cdot\bar{\delta}^{p+1}\\
&\vdots\\
\nabla^{(2)}Q(\bar{w}) &= 2L^*_{w^2}\circ L^*_{x^3}\circ\cdots\circ L^*_{x^p}\cdot\bar{\delta}^{p+1}\\
\nabla^{(1)}Q(\bar{w}) &= 2L^*_{w^1}L^*_{x^2}\circ\cdots L^*_{x^{p-1}}\circ L^*_{x^p}\cdot\bar{\delta}^{p+1}
\end{aligned}$$

Call *(p+1)*-th *error* to the already defined final error: $\bar{\delta}^{p+1}=\bar{\delta}=f(\bar{w},\bar{x})-\bar{y}\in\mathbf{R}^{n_{p+1}}$, and define the k-th *error* as $\bar{\delta}^k=D^*_{x^k}f^k(\bar{w}^k,\bar{x}^k)\cdot\bar{\delta}^{k+1}=L^*_{x^k}\cdot\bar{\delta}^{k+1}\in\mathbf{R}^{n_k}$, $k=p,p-1,\ldots,2,1$. One says that $\bar{\delta}^k$ is obtained *backpropagating* the error $\bar{\delta}^{k+1}$ by means of $D^*_{x^k}f^k(\bar{w}^k,\bar{x}^k)$. Define also the *desired k-th output* $\bar{y}^{k+1}\in\mathbf{R}^{n_{k+1}}$ as $\bar{y}^{k+1}=\bar{x}^{k+1}+\bar{\delta}^{k+1}$. The previous expressions for the components of the gradient can now be reformulated saying that the components of $\nabla Q(\bar{w})$ are the *liftings* to $\mathbf{R}^{M_k}$ via $D^*_{w^k}f^k(\bar{w}^k,\bar{x}^k)=L^*_{w^k}$ of the backpropagated errors

$$\begin{aligned}
\nabla^{(p)}Q(\bar{w}) &= 2\,L^*_{w^p}\cdot\bar{\delta}^{p+1}\\
\nabla^{(p-1)}Q(\bar{w}) &= 2\,L^*_{w^{p-1}}\cdot\bar{\delta}^{p}\\
&\vdots\\
\nabla^{(2)}Q(\bar{w}) &= 2\,L^*_{w^2}\cdot\bar{\delta}^{3}\\
\nabla^{(1)}Q(\bar{w}) &= 2\,L^*_{w^1}\cdot\bar{\delta}^{2}
\end{aligned}\tag{15}$$

For the maps $f^k=f^k(w^k,x^k):W^k\times\mathbf{R}^{n_k}\to\mathbf{R}^{n_{k+1}}$ take $x^k=\bar{x}^k$=k-th input$\in\mathbf{R}^{n_k}$, $y^{k+1}=\bar{y}^{k+1}$=k+1-th desired output$\in\mathbf{R}^{n_{k+1}}$ and consider the quadratic error $Q(f^k)(w^k)=\langle f^k(w^k,\bar{x}^k)-\bar{y}^{k+1},f^k(w^k,\bar{x}^k)-\bar{y}^{k+1}\rangle$. This has gradient $\nabla Q(f^k)(\bar{w}^k)=2D^*_{w^k}f^k(\bar{w}^k,\bar{x}^k)\cdot[f^k(\bar{w}^k,\bar{x}^k)-\bar{y}^{k+1}]=2\,L^*_{w^k}\cdot\bar{\delta}^{k+1}=\nabla^{(k)}Q(\bar{w})$, thus

$$\nabla(Q)(\bar{w})=(\nabla(Q(f^1))(\bar{w}^1),\ldots,\nabla(Q(f^p))(\bar{w}^p))\tag{16}$$

This says that for a neural network the quadratic error has gradient whose components are equal to the gradients of the quadratic error of the layers, calculated for the appropiated inputs and targets.

11.- **Backpropagation in neural networks.** Let $f^k:W^k\times\mathbf{R}^{n_k}\to\mathbf{R}^{n_{k+1}}$, $k=1,\ldots,p$, be differentiable, $W=W^1\times\cdots\times W^p$, and consider the neural network $f=f^p\,\hat{\circ}\cdots\hat{\circ}\,f^1:W\times\mathbf{R}^{n_1}\to\mathbf{R}^{n_{p+1}}$. Let the point $\bar{x}^1\in\mathbf{R}^{n_1}$ be an input, $\bar{y}^{p+1}\in\mathbf{R}^{n_{p+1}}$ a desired output and let $Q(w)$ be the quadratic

error. To minimize Q the gradient algorithm can be used; see section 1. The components $\nabla^k(f)(\bar{w})$ of $\nabla(f)(\bar{w})$ can be calculated in terms of the gradients $\nabla(f^k)(\bar{w}^k)$ of the quadratic error functions $Q(f^k)(w^k)$ backpropagating errors; see previous section. If the layers f^k are parametric products then the gradients $\nabla(f^k)(\bar{w}^k)$ can be expressed in terms of the gradients of the quadratic errors of the factors (processing units); see section 7.

In general the network has to be trained for several data: a finite set $\bar{X} \subseteq \mathbf{R}^{n_1}$ of inputs and targets $\bar{Y} \subseteq \mathbf{R}^{n_{p+1}}$. For each input $\bar{x}$ let its corresponding target be $\bar{y} = g(\bar{x})$, and define $Q^{\bar{x}}(f)(w) = \langle f(w,\bar{x}) - g(\bar{x}), f(w,\bar{x}) - g(\bar{x})\rangle$. The task is now to minimize the *total quadratic error* $Q^{\bar{X}}(f)(w) = \sum_{\bar{x}\in\bar{X}} Q^{\bar{x}}(f)(w)$. But since $\nabla Q^{\bar{X}}(f)(w) = \sum_{\bar{x}\in\bar{X}} \nabla Q^{\bar{x}}(f)(w)$ the calculation of $\nabla Q^{\bar{X}}(f)(w)$ reduces to the single input case.

If the network $f^p \hat{\circ} \cdots \hat{\circ} f^1$ is semilinear then $w = (w^1, \ldots, w^p)$, $w^k = (w^k_{j_{k+1}j_k})$ $\in \mathbf{R}^{n_{k+1}\times n_k}$ and the layer $f^k : \mathbf{R}^{n_{k+1}\times n_k} \times \mathbf{R}^{n_k+1} \to \mathbf{R}^{n_{k+1}}$ is a parametric product of n_{k+1} semilinear real valued units, that is, $f^k = f^k_1 \hat{\times} \cdots \hat{\times} f^k_{n_{k+1}}$ with $f^k_{j_{k+1}}(w^k_{j_{k+1}0}, w^k_{j_{k+1}1}, \ldots, w^k_{j_{k+1}n_k}, x^k_1, \ldots, x^k_{n_k}) = \psi(w^k_{j_{k+1}0} + w^k_{j_{k+1}1}x^k_1 + \cdots + w^k_{j_{k+1}n_k}x^k_{n_k})$. For a given single input $\bar{x}^1 = (\bar{x}^1_1, \ldots, \bar{x}^1_{n_k}) \in \mathbf{R}^{n_1}$, target $\bar{y}^{p+1} \in \mathbf{R}^{n_{p+1}}$ and parameter $\bar{w} = (\bar{w}^1, \ldots, \bar{w}^p)$ let $\bar{\delta}^{p+1} = (\bar{\delta}^{p+1}_1, \ldots, \bar{\delta}^{p+1}_{n_{p+1}})$ be the final error. Formula 14 gives the partial with respect to x of a semilinear product and it follows that the backpropagated errors are

$$\bar{\delta}^k = \begin{pmatrix} \bar{\delta}^k_1 \\ \vdots \\ \bar{\delta}^k_{n_k} \end{pmatrix} = \begin{pmatrix} \bar{w}^k_{11} & \cdots & \bar{w}^k_{n_{k+1}1} \\ \vdots & & \vdots \\ \bar{w}^k_{1n_k} & \cdots & \bar{w}^k_{n_{k+1}n_k} \end{pmatrix} \begin{pmatrix} \psi'(\bar{s}^k_1) & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \psi'(\bar{s}^k_{n_{k+1}}) \end{pmatrix} \begin{pmatrix} \bar{\delta}^{k+1}_1 \\ \vdots \\ \bar{\delta}^{k+1}_{n_{k+1}} \end{pmatrix}$$

where $k = p, p-1, \ldots 2$, $j_{k+1} = 1, \ldots, n_{k+1}$ and $\bar{s}^k_{j_{k+1}} = \langle \bar{w}^k_{j_{k+1}}, (1, \bar{x}^k)\rangle = \bar{w}^k_{j_{k+1}0} + \bar{w}^k_{j_{k+1}1}x^k_1 + \cdots + \bar{w}^k_{j_{k+1}n_k}x^k_{n_k}$. Formula 15 implies that the gradient has components $\nabla Q(f^k)(\bar{w}^k)$ and Formula 12 gives the partial with respect to w of a semilinear product implying that for semilinear networks the components of the gradient of the quadratic error are

$$2 \begin{pmatrix} \psi'(\bar{s}^k_1) & \cdots & \psi'(\bar{s}^k_1) \\ \vdots & & \vdots \\ \psi'(\bar{s}^k_{n_{k+1}}) & \cdots & \psi'(\bar{s}^k_{n_{k+1}}) \end{pmatrix} \# \begin{pmatrix} 1 & \bar{x}^k_1 & \cdots & \bar{x}^k_{n_k} \\ \vdots & \vdots & & \vdots \\ 1 & \bar{x}^k_1 & \cdots & \bar{x}^k_{n_k} \end{pmatrix} \# \begin{pmatrix} \bar{\delta}^{k+1}_1 & \cdots & \bar{\delta}^{k+1}_1 \\ \vdots & & \vdots \\ \bar{\delta}^{k+1}_{n_{k+1}} & \cdots & \bar{\delta}^{k+1}_{n_{k+1}} \end{pmatrix}$$

where each of the three matrices in the term by term products has n_{k+1} rows and $n_k + 1$ columns. In the case of several input-output pairs the gradients are added up as previously explained.

In practical applications the function ψ most often used is the sigmoid $\psi(t) = (1 + e^{-t})^{-1}$ which has derivative $\psi'(t) = e^{-t}(1 + e^{-t})^{-2}$.

REFERENCES

[1] Crespin, D. Neural Network Formalism.
[2] Crespin, D. Generalized Backpropagation (this present paper).
[3] Crespin, D. Geometry of Perceptrons.
[4] Crespin, D. Neural Polyhedra.
[5] Crespin, D. Pattern recognition with untrained perceptrons.
[6] Crespin, D. Feature Extraction. To appear.
[7] Abraham, R; Marsden, J.E.; Ratiu, T. Manifolds, Tensor Analysis and Applications.
[8] Lang, S. Analysis I.
[9] Lang, S. Linear Algebra.
[10] Hecht-Nielsen, R. Neurocomputing.

N.B.: Preprints [1]-[5] are available through World Wide Web at
" http:// euler.ciens.ucv.ve/Professors/dcrespin/Pub/ "

Daniel Crespin
dcrespin@euler.ciens.ucv.ve
Galipán, 24 de Diciembre de 1995.