# Backpropagation

## I. Differential Calculus Framework

### 1.1 Differential

Let $E, F$ be finite-dimensional real normed vector spaces, $U \subset E$ open, and $f : U \to F$.

**Definition 1.1.** The function $f$ is *differentiable at* $a \in U$ if there exists a linear map $\varphi : E \to F$ such that
$f(a+h) = f(a) + \varphi(h) + o(\|h\|)$
as $h \to 0$. The linear map $\varphi$ is unique and called the *differential of* $f$ *at* $a$, denoted $Df(a) : E \to F$.

### 1.2 Jacobian Matrix

Fix bases for $E \cong \mathbb{R}^n$ and $F \cong \mathbb{R}^m$. The differential $Df(a)$ is represented by the *Jacobian matrix*

$$J_f(a) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(a) & \cdots & \frac{\partial f_1}{\partial x_n}(a) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(a) & \cdots & \frac{\partial f_m}{\partial x_n}(a) \end{pmatrix} \in \mathbb{R}^{m \times n}$$

where $[Df(a)(h)]_i = \sum_{j=1}^{n} J_f(a)_{ij} h_j$.

**Special case.** When $F = \mathbb{R}$ (scalar-valued), $J_f(a) \in \mathbb{R}^{1 \times n}$ is identified with the *gradient*

$$\nabla f(a) = \begin{pmatrix} \frac{\partial f}{\partial x_1}(a) \\ \vdots \\ \frac{\partial f}{\partial x_n}(a) \end{pmatrix} \in \mathbb{R}^n$$

## 1.3 Chain Rule

**Theorem 1.2 (Chain Rule for Differentials).** Let $f : U \to V$ be differentiable at $a \in U \subset E$ and $g : V \to W$ be differentiable at $f(a) \in V \subset F$. Then $g \circ f$ is differentiable at $a$ with
$D(g \circ f)(a) = Dg(f(a)) \circ Df(a)$

**Corollary 1.3 (Chain Rule for Jacobians).** With notation as above,
$J_{g \circ f}(a) = J_g(f(a))J_f(a)$

**Corollary 1.4 (Scalar Chain Rule).** If $h = g \circ f$ where $g : \mathbb{R}^m \to \mathbb{R}$ and $f : \mathbb{R}^n \to \mathbb{R}^m$, then
$\nabla h(a) = J_f(a)^T \nabla g(f(a))$
or componentwise,
$\frac{\partial h}{\partial x_j}(a) = \sum_{i=1}^m \frac{\partial g}{\partial y_i}(f(a)) \frac{\partial f_i}{\partial x_j}(a)$

## 1.4 Diagonal Matrices for Component-wise Operations

For a differentiable function $\sigma : \mathbb{R} \to \mathbb{R}$ applied component-wise to $z = (z_1, \ldots, z_n) \in \mathbb{R}^n$, define $\sigma(z) = (\sigma(z_1), \ldots, \sigma(z_n))$.

**Lemma 1.5.** The Jacobian of $z \mapsto \sigma(z)$ at $z$ is the diagonal matrix
$J_\sigma(z) = D(z) := \operatorname{diag}(\sigma'(z_1), \ldots, \sigma'(z_n)) \in \mathbb{R}^{n \times n}$

---

# II. Neural Network as Function Composition

## 2.1 Layer Structure

From the article (Section 3), a neural network with $L$ layers computes:

$$a^{[1]} = x \in \mathbb{R}^{n_1} \tag{6}$$
$$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]} \in \mathbb{R}^{n_l} \quad (l = 2, \ldots, L) \tag{7}$$
$$a^{[l]} = \sigma(z^{[l]}) \in \mathbb{R}^{n_l} \tag{8}$$

## 2.2 Cost Function

For a single training point $(x, y)$, the cost is (equation 5.1):
$C = C(W^{[2]}, \dots, W^{[L]}, b^{[2]}, \dots, b^{[L]}) = \frac{1}{2}\|y - a^{[L]}\|_2^2$

## 2.3 Compositional notation

Define for each layer $l \geq 2$ the functions:

- **Affine map**: $\phi^{[l]} : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}, \phi^{[l]}(a) = W^{[l]}a + b^{[l]}$
- **Activation**: $\sigma : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}$, component-wise

Then $a^{[l]} = \sigma \circ \phi^{[l]}(a^{[l-1]})$ and the network output is
$a^{[L]} = F(x) = (\sigma \circ \phi^{[L]}) \circ \cdots \circ (\sigma \circ \phi^{[2]})(x)$

The cost becomes $C = \frac{1}{2}\|y - F(x)\|_2^2$.

## 2.4 The Backpropagation Problem

**Goal:** Compute $\frac{\partial C}{\partial W_{jk}^{[l]}}$ and $\frac{\partial C}{\partial b_j^{[l]}}$ for all layers $l = 2, \dots, L$ and all indices.

**Key observation:** By the chain rule, these derivatives can be expressed in terms of $\frac{\partial C}{\partial z_j^{[l]}}$.

**Definition 2.1.** Define the *error vector* at layer $l$:

$$\delta^{[l]} = \begin{pmatrix} \frac{\partial C}{\partial z_1^{[l]}} \\ \vdots \\ \frac{\partial C}{\partial z_{n_l}^{[l]}} \end{pmatrix} \in \mathbb{R}^{n_l}$$

---

# III. Backpropagation Equations

## 3.1 Output Layer Error

**Proposition 3.1.**
$\delta^{[L]} = D^{[L]}(a^{[L]} - y)$
where $D^{[L]} = \text{diag}(\sigma'(z_1^{[L]}), \ldots, \sigma'(z_{n_L}^{[L]}))$.

**Proof.** Since $C = \frac{1}{2}\sum_{i=1}^{n_L}(y_i - a_i^{[L]})^2$ and $a_j^{[L]} = \sigma(z_j^{[L]})$:
$\frac{\partial C}{\partial z_j^{[L]}} = \frac{\partial C}{\partial a_j^{[L]}} \frac{\partial a_j^{[L]}}{\partial z_j^{[L]}} = -(y_j - a_j^{[L]})\sigma'(z_j^{[L]}) = (a_j^{[L]} - y_j)\sigma'(z_j^{[L]})$
Thus $\delta_j^{[L]} = \sigma'(z_j^{[L]})(a_j^{[L]} - y_j) = [D^{[L]}(a^{[L]} - y)]_j$. $\square$

## 3.2 Backward Recursion

**Proposition 3.2.** For $l = L-1, L-2, \ldots, 2$:
$\delta^{[l]} = D^{[l]}(W^{[l+1]})^T \delta^{[l+1]}$
where $D^{[l]} = \text{diag}(\sigma'(z_1^{[l]}), \ldots, \sigma'(z_{n_l}^{[l]}))$.

**Proof.** By the chain rule (Corollary 1.4):
$\frac{\partial C}{\partial z_j^{[l]}} = \sum_{k=1}^{n_{l+1}} \frac{\partial C}{\partial z_k^{[l+1]}} \frac{\partial z_k^{[l+1]}}{\partial z_j^{[l]}}$

Since $z_k^{[l+1]} = \sum_{s=1}^{n_l} W_{ks}^{[l+1]} a_s^{[l]} + b_k^{[l+1]}$ and $a_j^{[l]} = \sigma(z_j^{[l]})$:
$\frac{\partial z_k^{[l+1]}}{\partial z_j^{[l]}} = W_{kj}^{[l+1]}\sigma'(z_j^{[l]})$

Therefore:
$\delta_j^{[l]} = \sigma'(z_j^{[l]}) \sum_{k=1}^{n_{l+1}} W_{kj}^{[l+1]} \delta_k^{[l+1]} = \sigma'(z_j^{[l]})[(W^{[l+1]})^T \delta^{[l+1]}]_j$
which gives the matrix form. $\square$

## 3.3 Parameter Gradients

**Proposition 3.3.** For all layers $l = 2, \ldots, L$:
$\frac{\partial C}{\partial b_j^{[l]}} = \delta_j^{[l]}$
$\frac{\partial C}{\partial W_{jk}^{[l]}} = \delta_j^{[l]} a_k^{[l-1]}$

**Proof.** (a) Since $z_j^{[l]} = \sum_s W_{js}^{[l]} a_s^{[l-1]} + b_j^{[l]}$, we have $\frac{\partial z_j^{[l]}}{\partial b_j^{[l]}} = 1$ and $\frac{\partial z_s^{[l]}}{\partial b_j^{[l]}} = 0$ for $s \neq j$.

Thus:
$\frac{\partial C}{\partial b_j^{[l]}} = \sum_{s=1}^{n_l} \frac{\partial C}{\partial z_s^{[l]}} \frac{\partial z_s^{[l]}}{\partial b_j^{[l]}} = \delta_j^{[l]}$

(b) Similarly, $\frac{\partial z_j^{[l]}}{\partial W_{jk}^{[l]}} = a_k^{[l-1]}$ and $\frac{\partial z_s^{[l]}}{\partial W_{jk}^{[l]}} = 0$ for $s \neq j$. Thus:
$\frac{\partial C}{\partial W_{jk}^{[l]}} = \delta_j^{[l]} a_k^{[l-1]}$ $\square$

**Matrix form.** These relations can be written as:
$\frac{\partial C}{\partial b^{[l]}} = \delta^{[l]}, \quad \frac{\partial C}{\partial W^{[l]}} = \delta^{[l]} (a^{[l-1]})^T$

---

## IV. Backpropagation Algorithm

**Input:** Training point $(x, y)$, current parameters $(W^{[l]}, b^{[l]})_{l=2}^{L}$

**Forward pass:** Compute $a^{[1]} = x$, then for $l = 2, \ldots, L$:
$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}, \quad a^{[l]} = \sigma(z^{[l]})$

**Backward pass:**

1. Compute $\delta^{[L]} = D^{[L]}(a^{[L]} - y)$
2. For $l = L-1, \ldots, 2$: $\delta^{[l]} = D^{[l]}(W^{[l+1]})^T \delta^{[l+1]}$

**Gradients:** For $l = 2, \ldots, L$:
$\frac{\partial C}{\partial W^{[l]}} = \delta^{[l]} (a^{[l-1]})^T, \quad \frac{\partial C}{\partial b^{[l]}} = \delta^{[l]}$

**Computational complexity:** $O(\text{network size})$ — each connection is visited once in each pass.

---

## V. Connection to Article Notation

The article uses the Hadamard (component-wise) product $\circ$ where we use diagonal matrices:
$\sigma'(z^{[l]}) \circ v \equiv D^{[l]} v$

This equivalence allows:

- Article equation (5.5): $\delta^{[L]} = \sigma'(z^{[L]}) \circ (a^{[L]} - y) \equiv D^{[L]}(a^{[L]} - y)$
- Article equation (5.6): $\delta^{[l]} = \sigma'(z^{[l]}) \circ (W^{[l+1]})^T \delta^{[l+1]} \equiv D^{[l]}(W^{[l+1]})^T \delta^{[l+1]}$

The diagonal matrix viewpoint makes explicit the linear algebraic structure of backpropagation as a sequence of linear transformations dictated by the chain rule.

---

## VI. Worked Example: (2,3,2) Network with Linear Output

### 6.1 Network Architecture

Consider a network with:

- **Input layer**: $n_1 = 2$
- **Hidden layer**: $n_2 = 3$ with sigmoid activation
- **Output layer**: $n_3 = 2$ with **no activation** (linear output)

**Parameters:**

$$W^{[2]} \in \mathbb{R}^{3\times 2}, \quad b^{[2]} \in \mathbb{R}^3 \tag{9}$$
$$W^{[3]} \in \mathbb{R}^{2\times 3}, \quad b^{[3]} \in \mathbb{R}^2 \tag{10}$$

### 6.2 Forward Pass

Given input $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2$:

**Layer 2 (hidden):**

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]} = \begin{pmatrix} w_{11}^{[2]} & w_{12}^{[2]} \\ w_{21}^{[2]} & w_{22}^{[2]} \\ w_{31}^{[2]} & w_{32}^{[2]} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1^{[2]} \\ b_2^{[2]} \\ b_3^{[2]} \end{pmatrix} = \begin{pmatrix} z_1^{[2]} \\ z_2^{[2]} \\ z_3^{[2]} \end{pmatrix}$$

$$a^{[2]} = \sigma(z^{[2]}) = \begin{pmatrix} \sigma(z_1^{[2]}) \\ \sigma(z_2^{[2]}) \\ \sigma(z_3^{[2]}) \end{pmatrix} = \begin{pmatrix} a_1^{[2]} \\ a_2^{[2]} \\ a_3^{[2]} \end{pmatrix}$$

where $\sigma(t) = \frac{1}{1+e^{-t}}$.

**Layer 3 (output, linear):**

$$z^{[3]} = W^{[3]}a^{[2]} + b^{[3]} = \begin{pmatrix} w_{11}^{[3]} & w_{12}^{[3]} & w_{13}^{[3]} \\ w_{21}^{[3]} & w_{22}^{[3]} & w_{23}^{[3]} \end{pmatrix} \begin{pmatrix} a_1^{[2]} \\ a_2^{[2]} \\ a_3^{[2]} \end{pmatrix} + \begin{pmatrix} b_1^{[3]} \\ b_2^{[3]} \end{pmatrix}$$

$$a^{[3]} = z^{[3]} \quad \text{(no activation)}$$

## 6.3 Cost Function

For target $y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$:
$C = \frac{1}{2}\|y - a^{[3]}\|^2 = \frac{1}{2}[(y_1 - a_1^{[3]})^2 + (y_2 - a_2^{[3]})^2]$

## 6.4 Backward Pass with Explicit Chain Rule

### Step 1: Output Layer Error $\delta^{[3]}$

Since $a^{[3]} = z^{[3]}$ (identity function), we have $\frac{\partial a_i^{[3]}}{\partial z_i^{[3]}} = 1$.

**Component-wise:**
$\delta_1^{[3]} = \frac{\partial C}{\partial z_1^{[3]}} = \frac{\partial C}{\partial a_1^{[3]}} \frac{\partial a_1^{[3]}}{\partial z_1^{[3]}} = -(y_1 - a_1^{[3]}) \cdot 1 = a_1^{[3]} - y_1$
$\delta_2^{[3]} = \frac{\partial C}{\partial z_2^{[3]}} = \frac{\partial C}{\partial a_2^{[3]}} \frac{\partial a_2^{[3]}}{\partial z_2^{[3]}} = -(y_2 - a_2^{[3]}) \cdot 1 = a_2^{[3]} - y_2$

**Vector form:**

$$\boxed{\delta^{[3]} = a^{[3]} - y = \begin{pmatrix} a_1^{[3]} - y_1 \\ a_2^{[3]} - y_2 \end{pmatrix}}$$

**Note:** For linear output, $D^{[3]} = I_2$ (identity matrix), so $\delta^{[3]} = I_2(a^{[3]} - y) = a^{[3]} - y$.

### Step 2: Hidden Layer Error $\delta^{[2]}$

**Explicit chain rule for $\delta_1^{[2]}$:**
$\delta_1^{[2]} = \frac{\partial C}{\partial z_1^{[2]}} = \frac{\partial C}{\partial a_1^{[3]}} \frac{\partial a_1^{[3]}}{\partial z_1^{[2]}} + \frac{\partial C}{\partial a_2^{[3]}} \frac{\partial a_2^{[3]}}{\partial z_1^{[2]}}$

Now:
$a_1^{[3]} = z_1^{[3]} = w_{11}^{[3]} a_1^{[2]} + w_{12}^{[3]} a_2^{[2]} + w_{13}^{[3]} a_3^{[2]} + b_1^{[3]}$

Since $a_1^{[2]} = \sigma(z_1^{[2]})$ and other $a_i^{[2]}$ don't depend on $z_1^{[2]}$:
$\frac{\partial a_1^{[3]}}{\partial z_1^{[2]}} = w_{11}^{[3]} \frac{\partial a_1^{[2]}}{\partial z_1^{[2]}} = w_{11}^{[3]} \sigma'(z_1^{[2]})$

Similarly:
$\frac{\partial a_2^{[3]}}{\partial z_1^{[2]}} = w_{21}^{[3]} \sigma'(z_1^{[2]})$

Therefore:
$\delta_1^{[2]} = \frac{\partial C}{\partial a_1^{[3]}} w_{11}^{[3]} \sigma'(z_1^{[2]}) + \frac{\partial C}{\partial a_2^{[3]}} w_{21}^{[3]} \sigma'(z_1^{[2]})$

Since $\frac{\partial C}{\partial a_i^{[3]}} = -(y_i - a_i^{[3]})$ and $\frac{\partial C}{\partial z_i^{[3]}} = \delta_i^{[3]}$:
$\boxed{\delta_1^{[2]} = \sigma'(z_1^{[2]}) [w_{11}^{[3]} \delta_1^{[3]} + w_{21}^{[3]} \delta_2^{[3]}]}$

**Similarly:**
$\delta_2^{[2]} = \sigma'(z_2^{[2]}) [w_{12}^{[3]} \delta_1^{[3]} + w_{22}^{[3]} \delta_2^{[3]}]$
$\delta_3^{[2]} = \sigma'(z_3^{[2]}) [w_{13}^{[3]} \delta_1^{[3]} + w_{23}^{[3]} \delta_2^{[3]}]$

**Matrix form:**
$\delta^{[2]} = D^{[2]} (W^{[3]})^T \delta^{[3]}$

where:

$$D^{[2]} = \begin{pmatrix} \sigma'(z_1^{[2]}) & 0 & 0 \\ 0 & \sigma'(z_2^{[2]}) & 0 \\ 0 & 0 & \sigma'(z_3^{[2]}) \end{pmatrix}$$

$$(W^{[3]})^T = \begin{pmatrix} w_{11}^{[3]} & w_{21}^{[3]} \\ w_{12}^{[3]} & w_{22}^{[3]} \\ w_{13}^{[3]} & w_{23}^{[3]} \end{pmatrix}$$

**Explicit computation:**

$$\boxed{\delta^{[2]} = \begin{pmatrix} \sigma'(z_1^{[2]}) & 0 & 0 \\ 0 & \sigma'(z_2^{[2]}) & 0 \\ 0 & 0 & \sigma'(z_3^{[2]}) \end{pmatrix} \begin{pmatrix} w_{11}^{[3]} & w_{21}^{[3]} \\ w_{12}^{[3]} & w_{22}^{[3]} \\ w_{13}^{[3]} & w_{23}^{[3]} \end{pmatrix} \begin{pmatrix} \delta_1^{[3]} \\ \delta_2^{[3]} \end{pmatrix}}$$

## 6.5 Parameter Gradients

**Layer 3 weights:**

$$\frac{\partial C}{\partial W^{[3]}} = \delta^{[3]}(a^{[2]})^T = \begin{pmatrix} \delta_1^{[3]} \\ \delta_2^{[3]} \end{pmatrix} \begin{pmatrix} a_1^{[2]} & a_2^{[2]} & a_3^{[2]} \end{pmatrix} = \begin{pmatrix} \delta_1^{[3]} a_1^{[2]} & \delta_1^{[3]} a_2^{[2]} & \delta_1^{[3]} a_3^{[2]} \\ \delta_2^{[3]} a_1^{[2]} & \delta_2^{[3]} a_2^{[2]} & \delta_2^{[3]} a_3^{[2]} \end{pmatrix}$$

**Layer 3 biases:**

$$\frac{\partial C}{\partial b^{[3]}} = \delta^{[3]} = \begin{pmatrix} \delta_1^{[3]} \\ \delta_2^{[3]} \end{pmatrix}$$

**Layer 2 weights:**

$$\frac{\partial C}{\partial W^{[2]}} = \delta^{[2]}(a^{[1]})^T = \begin{pmatrix} \delta_1^{[2]} \\ \delta_2^{[2]} \\ \delta_3^{[2]} \end{pmatrix} \begin{pmatrix} x_1 & x_2 \end{pmatrix} = \begin{pmatrix} \delta_1^{[2]} x_1 & \delta_1^{[2]} x_2 \\ \delta_2^{[2]} x_1 & \delta_2^{[2]} x_2 \\ \delta_3^{[2]} x_1 & \delta_3^{[2]} x_2 \end{pmatrix}$$

**Layer 2 biases:**

$$\frac{\partial C}{\partial b^{[2]}} = \delta^{[2]} = \begin{pmatrix} \delta_1^{[2]} \\ \delta_2^{[2]} \\ \delta_3^{[2]} \end{pmatrix}$$

## 6.6 Sigmoid Derivative

Recall $\sigma(t) = \frac{1}{1+e^{-t}}$, so $\sigma'(t) = \sigma(t)(1 - \sigma(t))$.

Thus:
$\sigma'(z_j^{[2]}) = a_j^{[2]}(1 - a_j^{[2]})$

This allows computing $D^{[2]}$ from the activations $a^{[2]}$ stored during the forward pass.