Backprojection

Assistant Lecturer: Eng. Ahmed Métwalli

Rounding policy. All displayed numeric values are rounded to **three decimals**. Full precision is used internally; values < 0.0005 display as 0.000.

1. Problem, Network, and Glossary of Symbols

We consider a minimal feedforward network: inputs $\mathbf{x} = (x_1, x_2)$, one hidden layer with two sigmoid units h_1, h_2, and a sigmoid output $\hat{y}$. One training example:

$$x_1 = 0.05, \quad x_2 = 0.10, \quad t = 0.50.$$

Parameters.

$$\begin{aligned}
\text{Input} \to \text{Hidden:}\quad & w_1 = 0.15,\ w_2 = 0.20\ (\to h_1), \quad w_3 = 0.25,\ w_4 = 0.30\ (\to h_2),\\
& b_1 = 0.35,\ b_2 = 0.35,\\
\text{Hidden} \to \text{Output:}\quad & w_5 = 0.40,\ w_6 = 0.45, \quad b_3 = 0.60.
\end{aligned}$$

Glossary (what each symbol means).

x_1, x_2	input features for this example
t	target (ground-truth) label for this example
w_i	weights (strength of a connection)
b_j	bias of a neuron (offset; connected to a constant 1)
z_{h_1}, z_{h_2}	pre-activations (weighted sums) of hidden units
h_1, h_2	hidden activations after the nonlinearity
z_o	pre-activation of the output unit
$\hat{y}$	model prediction (output activation)
E	loss: $E = \frac{1}{2}(\hat{y} - t)^2$
$\sigma(\cdot)$	sigmoid activation $\sigma(z) = \dfrac{1}{1 + e^{-z}}$

Neuron equations (symbolic).

$$\begin{aligned}
z_{h_1} &= w_1 x_1 + w_2 x_2 + b_1, \quad h_1 = \sigma(z_{h_1}),\\
z_{h_2} &= w_3 x_1 + w_4 x_2 + b_2, \quad h_2 = \sigma(z_{h_2}),\\
z_o &= w_5 h_1 + w_6 h_2 + b_3, \quad \hat{y} = \sigma(z_o).
\end{aligned}$$

1.1 Sigmoid activation and its derivative (how to write and use)

Definition :

$$\sigma(z) \;=\; \frac{1}{1+e^{-z}}.$$

Derivative :

$$\sigma'(z) \;=\; \frac{d}{dz}\left(\frac{1}{1+e^{-z}}\right) \;=\; \frac{e^{-z}}{(1+e^{-z})^2} \;=\; \sigma(z)\big(1-\sigma(z)\big).$$

The last form is the one used in backprop because at each neuron we already have the activation $a = \sigma(z)$, so $\sigma'(z) = a(1-a)$ is easy to compute.

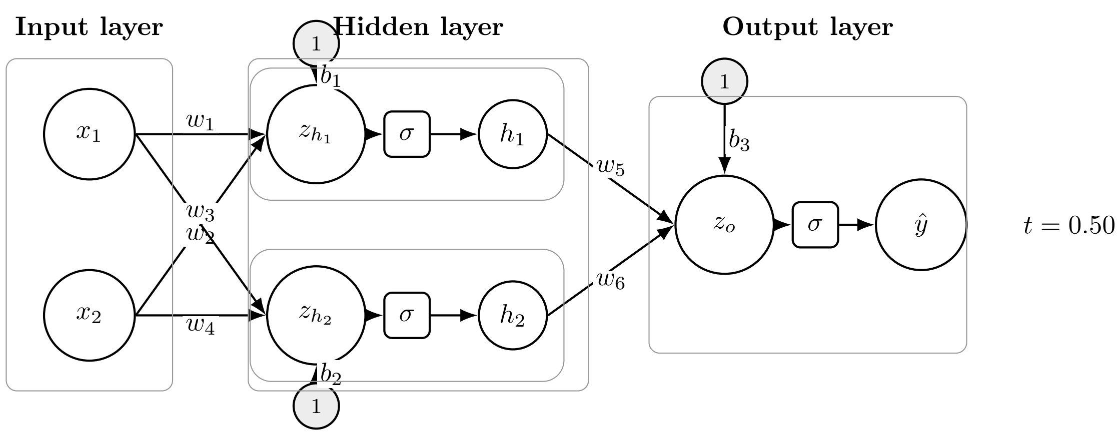
Chain rule pattern at each neuron. If $a = \sigma(z)$ and $z = $ (linear sum), then for any upstream scalar E,

$$\frac{\partial E}{\partial z} \;=\; \frac{\partial E}{\partial a} \quad \underbrace{\frac{\partial a}{\partial z}}_{\sigma'(z)=a(1-a)} \quad .$$

And for a weight w that appears linearly in $z = w \cdot (\text{input}) + \dots,$

$$\frac{\partial E}{\partial w} \;=\; \frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial w} \;=\; \big(\text{backprop sensitivity at the neuron}\big) \times \big(\text{local input}\big).$$

Network diagram.



2. Forward Pass (Step by Step,)

2.1 Hidden layer

$$z_{h_1} = 0.15(0.05) + 0.20(0.10) + 0.35 = 0.378, \quad h_1 = \sigma(0.378) = 0.593,$$
$$z_{h_2} = 0.25(0.05) + 0.30(0.10) + 0.35 = 0.393, \quad h_2 = \sigma(0.393) = 0.597.$$

2.2 Output layer

$$z_o = 0.40(0.593) + 0.45(0.597) + 0.60 = 1.106, \qquad \hat{y} = \sigma(1.106) = 0.751.$$
$$E = \tfrac{1}{2}(\hat{y} - t)^2 = \tfrac{1}{2}(0.751 - 0.50)^2 = 0.032.$$

3. Backpropagation: Exact Chain-Rule Equations

3.1 Output layer

$$\frac{\partial E}{\partial \hat{y}} = \hat{y} - t, \qquad \frac{\partial \hat{y}}{\partial z_o} = \hat{y}(1-\hat{y}), \qquad \boxed{\frac{\partial E}{\partial z_o} = (\hat{y} - t)\,\hat{y}(1-\hat{y}).}$$

Parameter gradients:

$$\boxed{\frac{\partial E}{\partial w_5} = \frac{\partial E}{\partial z_o}\,h_1}, \quad \boxed{\frac{\partial E}{\partial w_6} = \frac{\partial E}{\partial z_o}\,h_2}, \quad \boxed{\frac{\partial E}{\partial b_3} = \frac{\partial E}{\partial z_o}}.$$

3.2 Hidden layer

Backpropagated to hidden activations:

$$\frac{\partial E}{\partial h_1} = \frac{\partial E}{\partial z_o}\,w_5, \qquad \frac{\partial E}{\partial h_2} = \frac{\partial E}{\partial z_o}\,w_6.$$

Sigmoid local slopes:

$$\frac{\partial h_j}{\partial z_{h_j}} = h_j(1-h_j), \qquad \boxed{\frac{\partial E}{\partial z_{h_j}} = \frac{\partial E}{\partial h_j}\,h_j(1-h_j)}.$$

Gradients to input→hidden parameters:

$$\boxed{\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial z_{h_1}}\,x_1}, \quad \boxed{\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial z_{h_1}}\,x_2}, \quad \boxed{\frac{\partial E}{\partial b_1} = \frac{\partial E}{\partial z_{h_1}}},$$

$$\boxed{\frac{\partial E}{\partial w_3} = \frac{\partial E}{\partial z_{h_2}}\,x_1}, \quad \boxed{\frac{\partial E}{\partial w_4} = \frac{\partial E}{\partial z_{h_2}}\,x_2}, \quad \boxed{\frac{\partial E}{\partial b_2} = \frac{\partial E}{\partial z_{h_2}}}.$$

4. Backpropagation: Numeric Walkthrough

4.1 Output sensitivity and output-layer gradients

$$\hat{y} - t = 0.751 - 0.50 = 0.251, \qquad \hat{y}(1-\hat{y}) = 0.751 \cdot 0.249 = 0.187,$$

$$\boxed{\frac{\partial E}{\partial z_o} = 0.251 \times 0.187 = 0.047.}$$

$$\frac{\partial E}{\partial w_5} = 0.047 \cdot 0.593 = \boxed{0.028}, \quad \frac{\partial E}{\partial w_6} = 0.047 \cdot 0.597 = \boxed{0.028}, \quad \frac{\partial E}{\partial b_3} = \boxed{0.047}.$$

4.2 Hidden-layer sensitivities

$$\frac{\partial E}{\partial h_1} = 0.047 \cdot 0.40 = 0.019, \qquad \frac{\partial E}{\partial h_2} = 0.047 \cdot 0.45 = 0.021.$$

Local sigmoid slopes :

$$h_1(1-h_1) = 0.593(0.407) = 0.241, \qquad h_2(1-h_2) = 0.597(0.403) = 0.241.$$

Thus

$$\frac{\partial E}{\partial z_{h_1}} = 0.019 \cdot 0.241 = \boxed{0.005}, \qquad \frac{\partial E}{\partial z_{h_2}} = 0.021 \cdot 0.241 = \boxed{0.005}.$$

4.3 Input→hidden parameter gradients

$$\frac{\partial E}{\partial w_1} = 0.005 \cdot 0.05 = \boxed{0.000}, \quad \frac{\partial E}{\partial w_2} = 0.005 \cdot 0.10 = \boxed{0.001}, \quad \frac{\partial E}{\partial b_1} = \boxed{0.005},$$
$$\frac{\partial E}{\partial w_3} = 0.005 \cdot 0.05 = \boxed{0.000}, \quad \frac{\partial E}{\partial w_4} = 0.005 \cdot 0.10 = \boxed{0.001}, \quad \frac{\partial E}{\partial b_2} = \boxed{0.005}.$$

4.4 One fully expanded chain (example: $\partial E/\partial w_1$)

Follow the path $w_1 \to z_{h_1} \to h_1 \to z_o \to \hat{y} \to E$:

$$\frac{\partial E}{\partial w_1} = \underbrace{\frac{\partial E}{\partial \hat{y}}}_{\hat{y}-t} \cdot \underbrace{\frac{\partial \hat{y}}{\partial z_o}}_{\hat{y}(1-\hat{y})} \cdot \underbrace{\frac{\partial z_o}{\partial h_1}}_{w_5} \cdot \underbrace{\frac{\partial h_1}{\partial z_{h_1}}}_{h_1(1-h_1)} \cdot \underbrace{\frac{\partial z_{h_1}}{\partial w_1}}_{x_1}.$$

Numerically :

$(0.751-0.50)\cdot 0.187\cdot 0.40\cdot 0.241\cdot 0.05 = 0.251\cdot 0.187\cdot 0.40\cdot 0.241\cdot 0.05 \approx 2.26\times 10^{-4} \quad \to \quad \boxed{0.000\ (3\text{ d.p.})}$.

5. Gradient Summary (Rounded to Three Decimals)

$$\begin{aligned}
\text{Output layer:}\quad & \frac{\partial E}{\partial w_5} = 0.028, \quad \frac{\partial E}{\partial w_6} = 0.028, \quad \frac{\partial E}{\partial b_3} = 0.047;\\
\text{Hidden layer:}\quad & \frac{\partial E}{\partial w_1} \approx 0.000, \quad \frac{\partial E}{\partial w_2} \approx 0.001, \quad \frac{\partial E}{\partial b_1} = 0.005,\\
& \frac{\partial E}{\partial w_3} \approx 0.000, \quad \frac{\partial E}{\partial w_4} \approx 0.001, \quad \frac{\partial E}{\partial b_2} = 0.005.
\end{aligned}$$

(Gradients for w_1, w_3 are $\ll 0.0005$ and therefore display as 0.000 at 3 d.p.; w_2, w_4 are slightly larger and display as 0.001.)

6. One SGD Update (Learning Rate $\eta = 0.50$)

Using $\theta \leftarrow \theta - \eta\, \partial E/\partial\theta$:

$$\begin{aligned}
&w_5' = 0.40 - 0.50(0.028) = \boxed{0.386}, \quad w_6' = 0.45 - 0.50(0.028) = \boxed{0.436}, \quad b_3' = 0.60 - 0.50(0.047) = \boxed{0.577},\\
&w_1' = 0.15 - 0.50(0.000) \approx \boxed{0.150}, \quad w_2' = 0.20 - 0.50(0.001) \approx \boxed{0.200}, \quad b_1' = 0.35 - 0.50(0.005) = \boxed{0.348},\\
&w_3' = 0.25 - 0.50(0.000) \approx \boxed{0.250}, \quad w_4' = 0.30 - 0.50(0.001) \approx \boxed{0.300}, \quad b_2' = 0.35 - 0.50(0.005) = \boxed{0.348}.
\end{aligned}$$

(The small input→hidden weight updates are within rounding tolerance at 3 d.p.)

7. Post-Update Forward Pass and Loss (Recomputing Hidden,)

Recompute hidden pre-activations because b_1, b_2 changed:

$$\begin{aligned}
z_{h_1}' &= 0.150(0.05) + 0.200(0.10) + 0.348 = 0.376, \quad h_1' = \sigma(0.376) = 0.593,\\
z_{h_2}' &= 0.250(0.05) + 0.300(0.10) + 0.348 = 0.390, \quad h_2' = \sigma(0.390) = 0.596.
\end{aligned}$$

Then

$$z_o' = 0.386 \cdot 0.593 + 0.436 \cdot 0.596 + 0.577 = 1.066, \qquad \hat{y}' = \sigma(1.066) = 0.744,$$

$$E' = \tfrac{1}{2}(0.744 - 0.50)^2 = \boxed{0.030}.$$

At full precision the loss decreases from ≈ 0.03159 to ≈ 0.02972.

8. Quick Reference and Tips

Chain rule recipe at each layer (scalar view).

1. Compute **output sensitivity** (a.k.a. delta): $\delta_o = \dfrac{\partial E}{\partial z_o} = (\hat{y} - t)\,\hat{y}(1 - \hat{y})$.
2. Backprop to hidden activations: $\dfrac{\partial E}{\partial h_j} = \delta_o\, w_{j\to o}$.
3. Convert to **hidden pre-activation** sensitivities: $\delta_{h_j} = \dfrac{\partial E}{\partial z_{h_j}} = \big(\dfrac{\partial E}{\partial h_j}\big)\, h_j(1 - h_j)$.
4. Form **parameter gradients**: for any weight into a neuron, gradient = (that neuron's δ) $\times$ (local input).

Common pitfalls.

- Always use $\sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big)$, not $\sigma'(a)$ unless $a = \sigma(z)$ is clearly referenced.
- Keep track of which *local input* multiplies which weight when forming $\partial E/\partial w$.
- Rounding too early can hide tiny but real gradients; keep full precision for computations, round only for display.

Small Note: Two Outputs — Regression and Classification

The hidden layer (h_1, h_2) is unchanged. The output layer is duplicated with its own $(w_5^{(k)}, w_6^{(k)}, b_3^{(k)})$ and targets $t^{(k)}$, $k \in \{1, 2\}$:

$$z_o^{(k)} = w_5^{(k)} h_1 + w_6^{(k)} h_2 + b_3^{(k)}, \qquad \hat{y}^{(k)} = \sigma\big(z_o^{(k)}\big) \text{ (unless stated otherwise).}$$

Pick an output/loss pair (only the output delta changes).

Activation + Loss	**Loss E**	**Output delta $\delta_o^{(k)} = \dfrac{\partial E}{\partial z_o^{(k)}}$**
Sigmoid + MSE	$\frac{1}{2}\sum_{k=1}^{2}\big(\hat{y}^{(k)} - t^{(k)}\big)^2$	$\big(\hat{y}^{(k)} - t^{(k)}\big)\,\hat{y}^{(k)}\big(1 - \hat{y}^{(k)}\big)$
Sigmoid + Binary CE (multi-label)	$-\sum_{k=1}^{2}\big[t^{(k)}\ln\hat{y}^{(k)} + \big(1 - t^{(k)}\big)\ln\big(1 - \hat{y}^{(k)}\big)\big]$	$\hat{y}^{(k)} - t^{(k)}$
Softmax + CE (mutually exclusive)	$-\sum_{k=1}^{2} t^{(k)}\ln\hat{y}^{(k)}$, $\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{z}_o)$	$\hat{y}^{(k)} - t^{(k)}$

Output-layer parameter gradients (component-wise).

$$\frac{\partial E}{\partial w_5^{(k)}} = \delta_o^{(k)}\, h_1, \qquad \frac{\partial E}{\partial w_6^{(k)}} = \delta_o^{(k)}\, h_2, \qquad \frac{\partial E}{\partial b_3^{(k)}} = \delta_o^{(k)}.$$

Backprop to hidden activations (sum both outputs).

$$\frac{\partial E}{\partial h_1} = \sum_{k=1}^{2} \delta_o^{(k)}\, w_5^{(k)}, \qquad \frac{\partial E}{\partial h_2} = \sum_{k=1}^{2} \delta_o^{(k)}\, w_6^{(k)}.$$

Hidden pre-activation sensitivities and earlier gradients (unchanged).

$$\frac{\partial E}{\partial z_{h_j}} = \left(\frac{\partial E}{\partial h_j}\right) h_j(1-h_j) \quad (j=1,2),$$

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial z_{h_1}}\, x_1, \quad \frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial z_{h_1}}\, x_2, \quad \frac{\partial E}{\partial b_1} = \frac{\partial E}{\partial z_{h_1}},$$

$$\frac{\partial E}{\partial w_3} = \frac{\partial E}{\partial z_{h_2}}\, x_1, \quad \frac{\partial E}{\partial w_4} = \frac{\partial E}{\partial z_{h_2}}\, x_2, \quad \frac{\partial E}{\partial b_2} = \frac{\partial E}{\partial z_{h_2}}.$$

Unbounded regression (if needed). For real-valued targets without $[0,1]$ bounds, identity outputs are used:

$$\hat{y}^{(k)} = z_o^{(k)}, \qquad \delta_o^{(k)} = \hat{y}^{(k)} - t^{(k)}.$$

All downstream formulas remain identical.

Takeaway. No new backprop rules are introduced; only the output delta depends on the activation/loss choice. All contributions from the two outputs are simply summed before applying the usual hidden-layer and input$\to$hidden formulas.