

Backpropagation

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.

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

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

Neuron equations (symbolic).

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

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

Definition :

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

Derivative :

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

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.

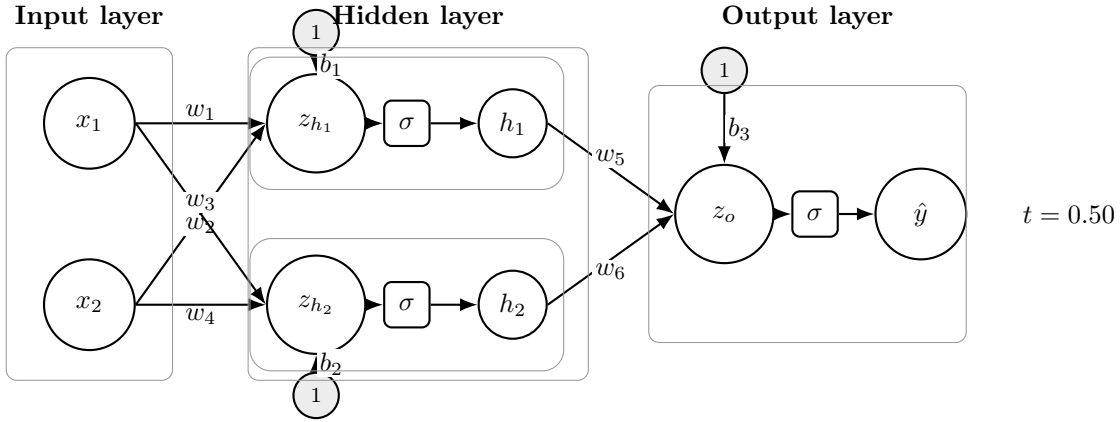
Chain rule pattern at each neuron. If $a = \sigma(z)$ and $z = (\text{linear sum})$, then for any upstream scalar E ,

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

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

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

Network diagram.



2. Forward Pass (Step by Step,)

2.1 Hidden layer

$$\begin{aligned} z_{h1} &= 0.15(0.05) + 0.20(0.10) + 0.35 = 0.378, & h_1 &= \sigma(0.378) = 0.593, \\ z_{h2} &= 0.25(0.05) + 0.30(0.10) + 0.35 = 0.393, & h_2 &= \sigma(0.393) = 0.597. \end{aligned}$$

2.2 Output layer

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

3. Backpropagation: Exact Chain-Rule Equations

3.1 Output layer

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

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:

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

Sigmoid local slopes:

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

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

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

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

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

4.2 Hidden-layer sensitivities

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

Local sigmoid slopes :

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

Thus

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

4.3 Input→hidden parameter gradients

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

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

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

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

Numerically :

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

5. Gradient Summary (Rounded to Three Decimals)

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

(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 $\theta \leftarrow \theta - \eta \partial E / \partial \theta$:

$$\begin{aligned}w'_5 &= 0.40 - 0.50(0.028) = \boxed{0.386}, & w'_6 &= 0.45 - 0.50(0.028) = \boxed{0.436}, & b'_3 &= 0.60 - 0.50(0.047) = \boxed{0.577}, \\ w'_1 &= 0.15 - 0.50(0.000) \approx \boxed{0.150}, & w'_2 &= 0.20 - 0.50(0.001) \approx \boxed{0.200}, & b'_1 &= 0.35 - 0.50(0.005) = \boxed{0.348}, \\ w'_3 &= 0.25 - 0.50(0.000) \approx \boxed{0.250}, & w'_4 &= 0.30 - 0.50(0.001) \approx \boxed{0.300}, & b'_2 &= 0.35 - 0.50(0.005) = \boxed{0.348}.\end{aligned}$$

(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:

$$\begin{aligned}z'_{h_1} &= 0.150(0.05) + 0.200(0.10) + 0.348 = 0.376, & h'_1 &= \sigma(0.376) = 0.593, \\ z'_{h_2} &= 0.250(0.05) + 0.300(0.10) + 0.348 = 0.390, & h'_2 &= \sigma(0.390) = 0.596.\end{aligned}$$

Then

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

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

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

Common pitfalls.

- Always use $\sigma'(z) = \sigma(z)(1 - \sigma(z))$, 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.