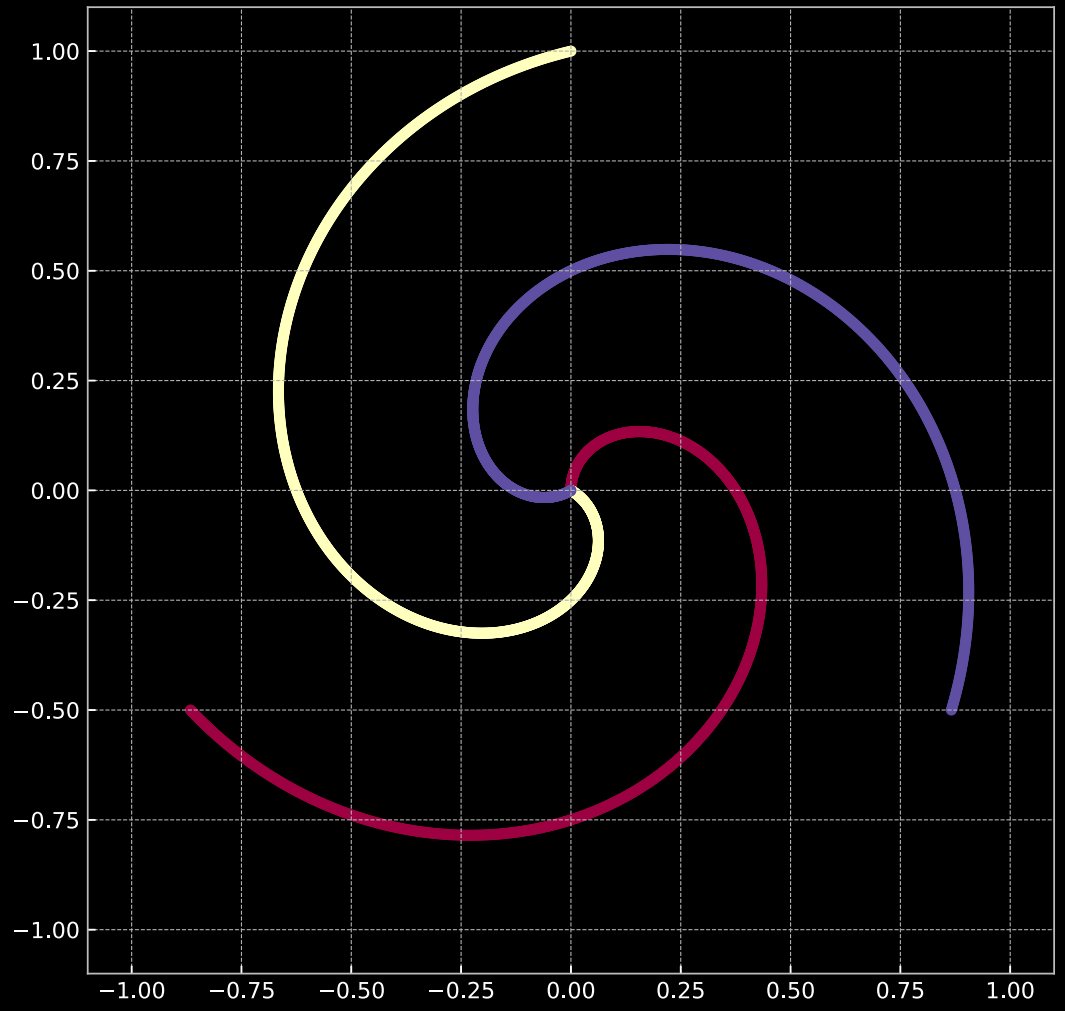
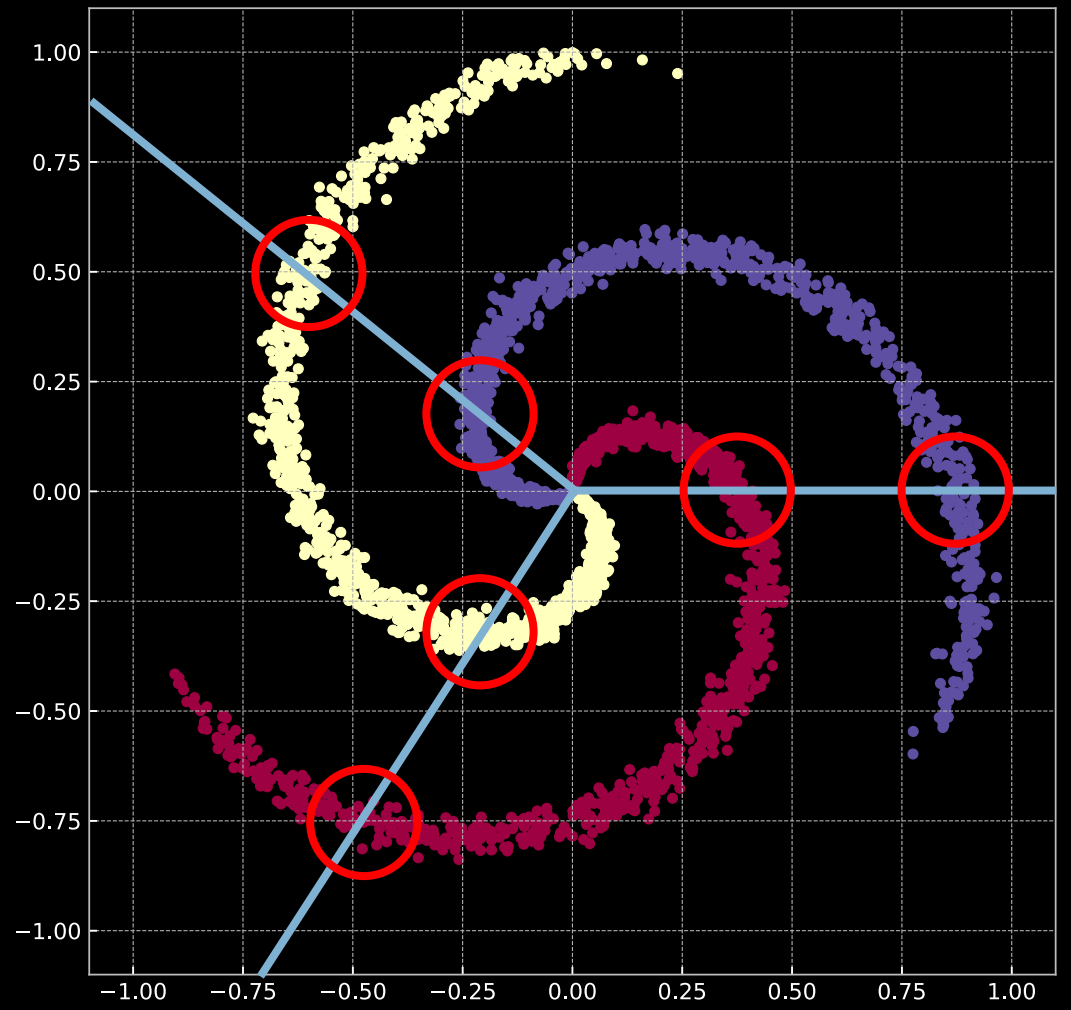
ANN – supervised learning

Classification

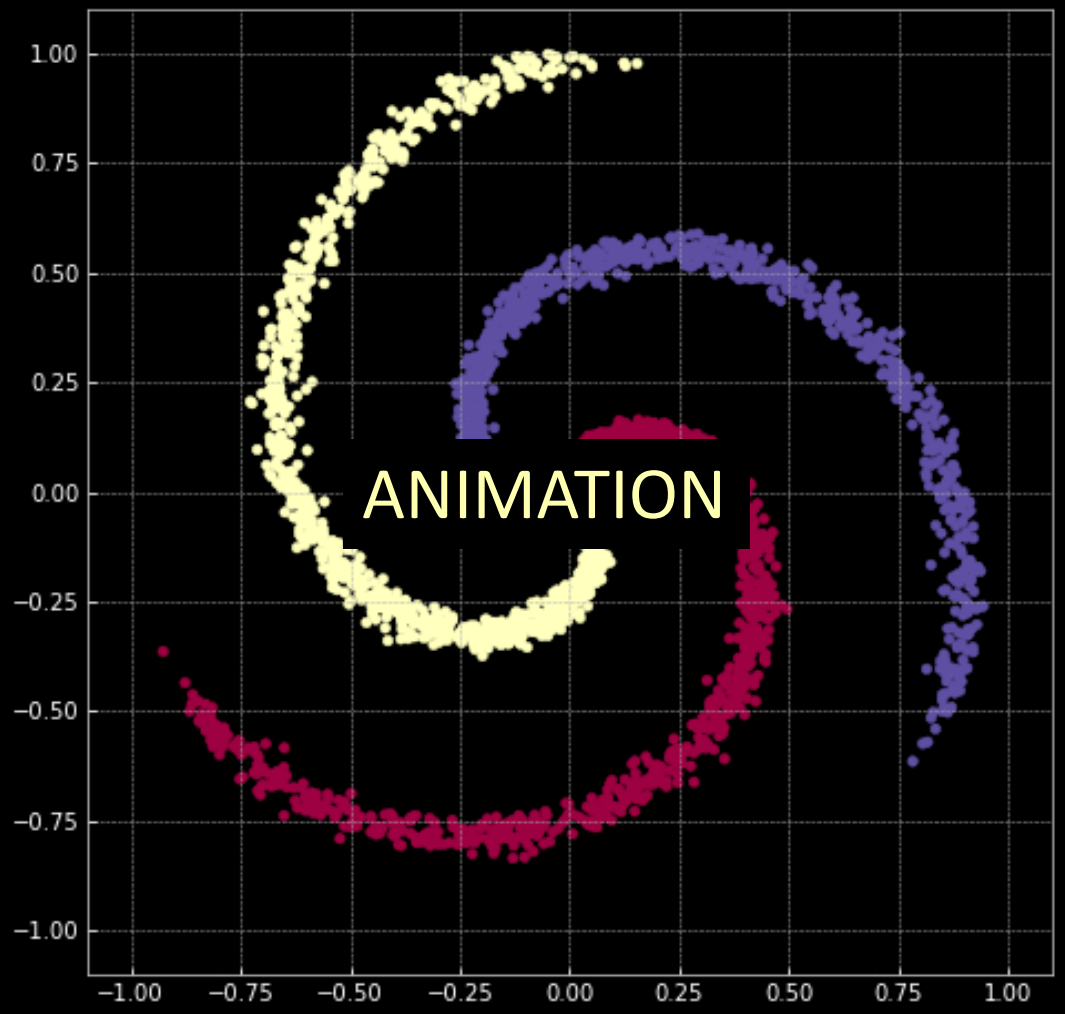


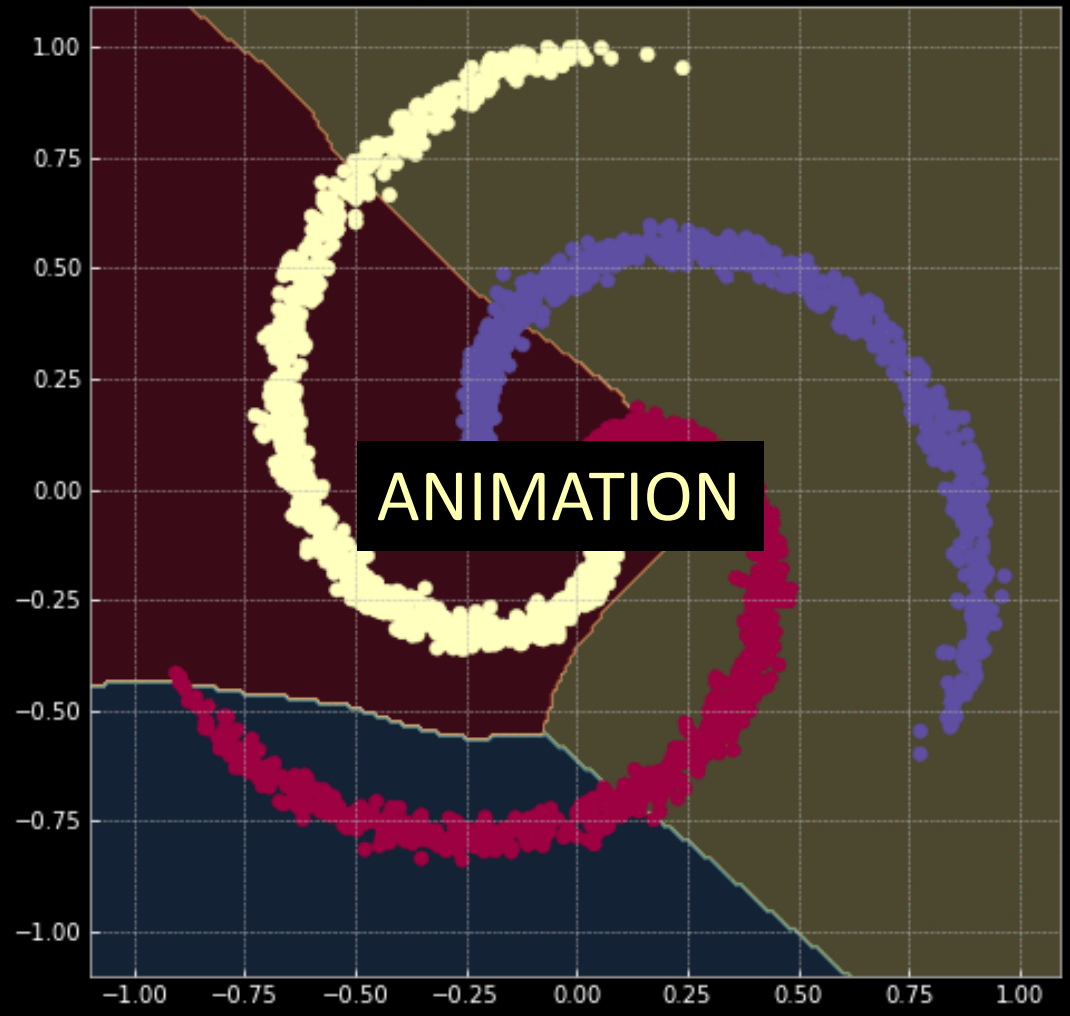


$$X_k(t) = t \begin{pmatrix} \sin\left[\frac{2\pi}{K}(2t+k-1)\right] \\ \cos\left[\frac{2\pi}{K}(2t+k-1)\right] \end{pmatrix}$$

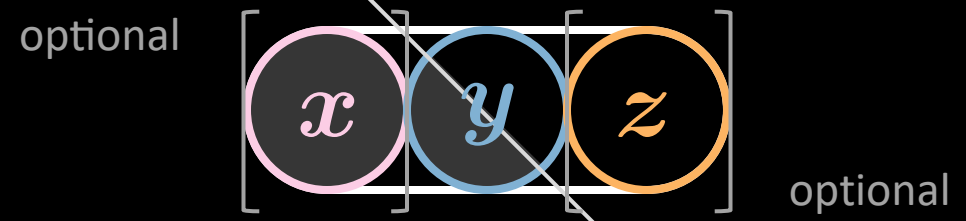
$$0 \le t \le 1, \quad k = 1, \cdots, K$$

$$X_k(t) = t \begin{pmatrix} \sin\left[\frac{2\pi}{K}(2t+k-1)\right] \\ \cos\left[\frac{2\pi}{K}(2t+k-1)\right] \end{pmatrix} + \mathcal{N}(0, \sigma^2)$$


ANIMATION
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00


ANIMATION
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00

Data



Variables' name

Input

x observed during: training ✅ — testing ✅

y observed during: training ✅ — testing ❌

z observed during: training ❌ — testing ❌

Output

h computed from the input (hidden / internal)

$\tilde{y}$ computed from the hidden (predicted y, ~ means *circa*)

Classification train data

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$
$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$
$$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

1-hot encoding

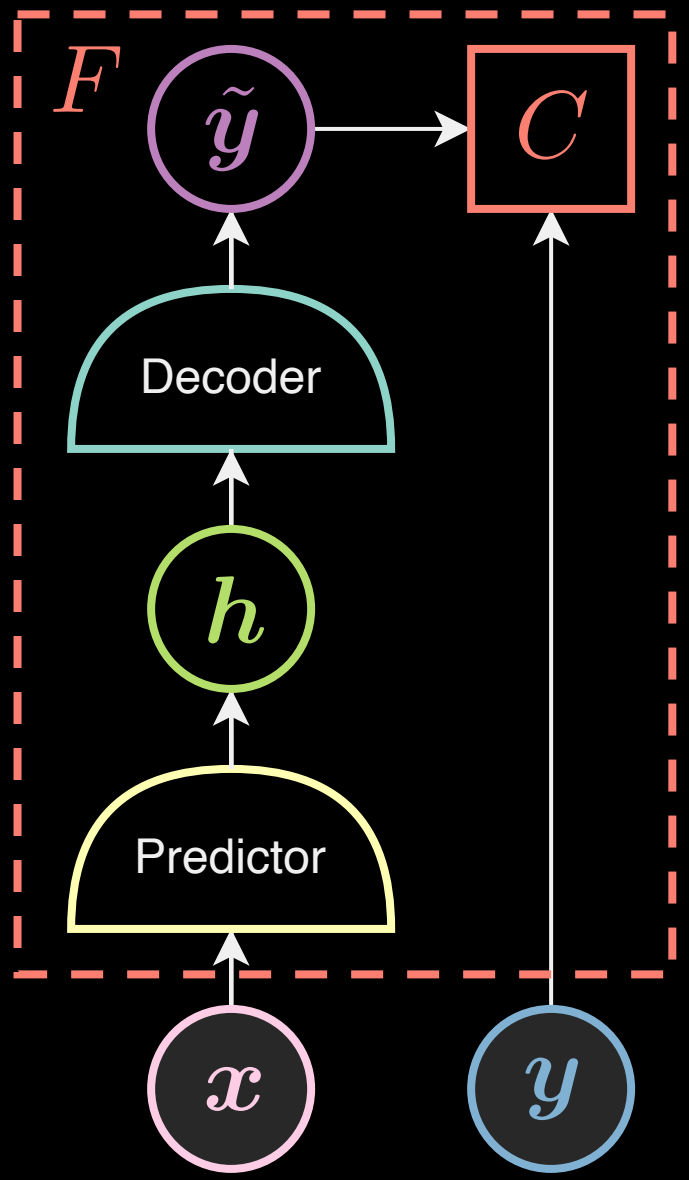
$K = 3$

$$\boldsymbol{X} = \begin{bmatrix} — \; \boldsymbol{x}^{(1)} \; — \\ — \; \boldsymbol{x}^{(2)} \; — \\ \vdots \\ — \; \boldsymbol{x}^{(P)} \; — \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \end{matrix} \overset{n}{\qquad} \; P$$

$$\boldsymbol{Y} = \begin{bmatrix} — \; \boldsymbol{y}^{(1)} \; — \\ — \; \boldsymbol{y}^{(2)} \; — \\ \vdots \\ — \; \boldsymbol{y}^{(P)} \; — \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \end{matrix} \overset{K}{\qquad} \; P$$

$$\boldsymbol{x}^{(p)} \in \mathbb{R}^n$$

$$\boldsymbol{y}^{(p)} \in \mathbb{I}_K$$



Neural network (inference)

$$h = \mathrm{Pred}(x) = f(W_h x + b_h)$$

$$\tilde{y} = \mathrm{Dec}(h) = g(W_{\tilde{y}} h + b_{\tilde{y}})$$

$$f, g = (\cdot)^+, \sigma(\cdot), \tanh(\cdot), \mathrm{softargmax}(\cdot)$$

incompatibility between ***x*** and ***y***

$$F(x, y) = C(y, \tilde{y})$$

distance between ***y*** and ***ỹ***

$$\tilde{y} = \tilde{y}(x), \quad \tilde{y} : \mathbb{R}^n \to \mathbb{R}^K, \quad x \mapsto \tilde{y}$$

$$\tilde{y} : \mathbb{R}^n \to \mathbb{R}^d \to \mathbb{R}^K, \quad d \gg n, K$$

Classification

$$C(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = -\log(\boldsymbol{y}^\top \tilde{\boldsymbol{y}})$$

cross entropy / negative log-probability

last layer linear sum

$$\tilde{\boldsymbol{y}} = \text{softargmax}_\beta(\boldsymbol{s}) \doteq \frac{\exp(\beta \boldsymbol{s})}{\sum_{k=1}^{K} \exp(\beta s_k)} \in (0,1)$$

w badness given a data set

data set

$$\mathcal{L}(\boldsymbol{w}, \mathcal{S}) \doteq \frac{1}{P} \sum_{p=1}^{P} L(\boldsymbol{w}, \boldsymbol{x}^{(p)}, \boldsymbol{y}^{(p)}), \quad L(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y}) \triangleq F(\boldsymbol{x}, \boldsymbol{y})$$

w badness given a data sample

$$= C(\boldsymbol{y}, \tilde{\boldsymbol{y}})$$

$$\boldsymbol{x}, \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\tilde{\boldsymbol{y}}(\boldsymbol{x}) = \begin{pmatrix} \sim 1 \\ \sim 0 \\ \sim 0 \end{pmatrix} \quad \Rightarrow \quad C\left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sim 1 \\ \sim 0 \\ \sim 0 \end{bmatrix}\right) \to 0^+$$

$$\tilde{\boldsymbol{y}}(\boldsymbol{x}) = \begin{pmatrix} \sim 0 \\ \sim 1 \\ \sim 0 \end{pmatrix} \quad \Rightarrow \quad C\left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sim 0 \\ \sim 1 \\ \sim 0 \end{bmatrix}\right) \to +\infty$$

Neural net training

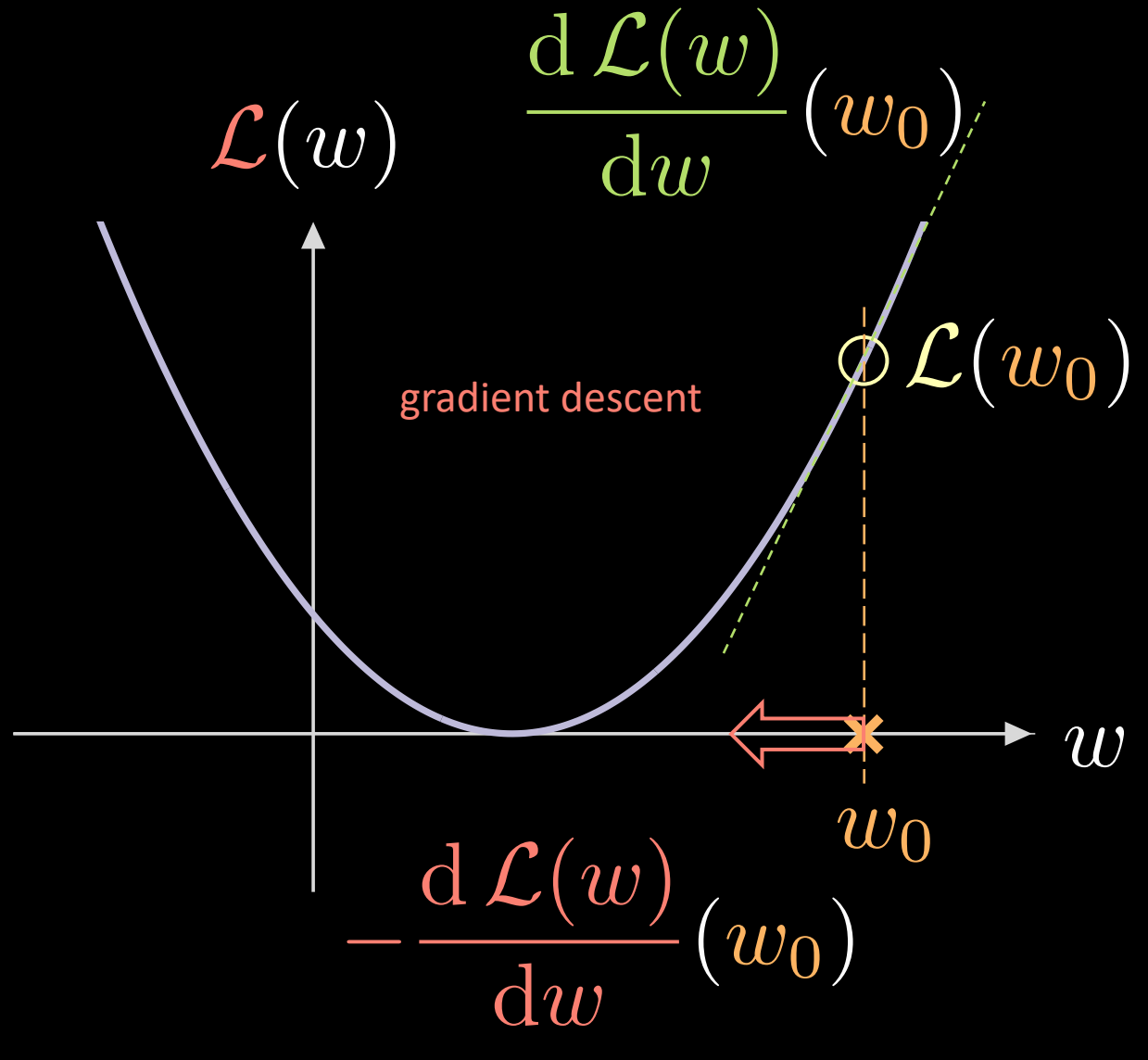
$$\boldsymbol{w} \doteq \{\boldsymbol{W_h}, \boldsymbol{b_h}, \boldsymbol{W_{\tilde{y}}}, \boldsymbol{b_{\tilde{y}}}\}$$

$$\mathcal{L}(\boldsymbol{w}, \mathcal{S}) \in \mathbb{R}^+$$

$$\frac{\partial \mathcal{L}(\boldsymbol{w})}{\partial \boldsymbol{W_{\tilde{y}}}} = \frac{\partial \mathcal{L}(\boldsymbol{w})}{\partial \tilde{\boldsymbol{y}}} \frac{\partial \tilde{\boldsymbol{y}}}{\partial \boldsymbol{W_{\tilde{y}}}}$$

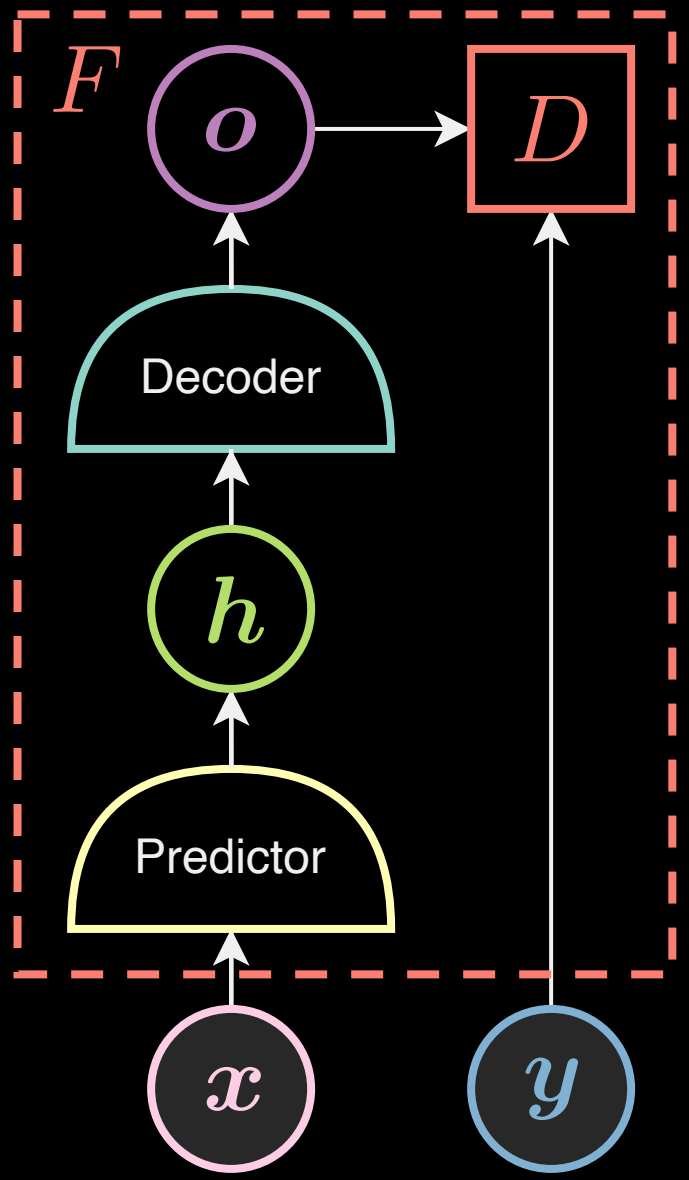
$$\frac{\partial \mathcal{L}(\boldsymbol{w})}{\partial \boldsymbol{W_h}} = \frac{\partial \mathcal{L}(\boldsymbol{w})}{\partial \tilde{\boldsymbol{y}}} \frac{\partial \tilde{\boldsymbol{y}}}{\partial \boldsymbol{h}} \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{W_h}}$$

back-propagation



Back-propagation

of the gradient



$$\boldsymbol{h} = f(\boldsymbol{W_h x} + \boldsymbol{b_h})$$

$$\boldsymbol{s} = a(\boldsymbol{h}) = \boldsymbol{W_y h} + \boldsymbol{b_y}$$

$$\boldsymbol{o} = g(\boldsymbol{s})$$

$$g = \text{logsoftargmax}$$

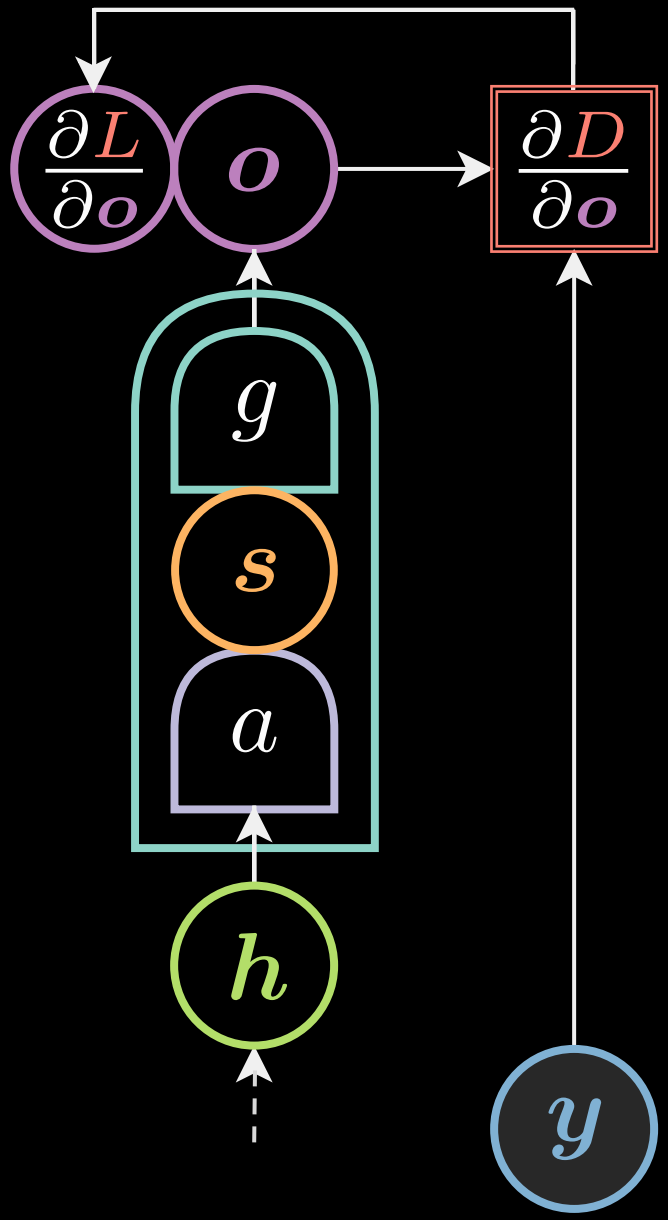
$$\boldsymbol{o} = \log(\tilde{\boldsymbol{y}})$$

$$C(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = -\log(\boldsymbol{y}^\top \tilde{\boldsymbol{y}})$$

$$L = F = D = -\boldsymbol{y}^\top \boldsymbol{o}$$

$$\frac{\partial L(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{W_y}} = \frac{\partial L(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{o}} \frac{\partial g}{\partial \boldsymbol{s}} \frac{\partial a}{\partial \boldsymbol{W_y}}$$

$$\boldsymbol{w} \doteq \{\boldsymbol{W_h}, \boldsymbol{b_h}, \boldsymbol{W_y}, \boldsymbol{b_y}\}$$



$$h = f(W_h x + b_h)$$

$$s = a(h) = W_y h + b_y$$

$$o = g(s)$$

$$g = \text{logsoftargmax}$$

$$L = F = D = -y^\top o$$

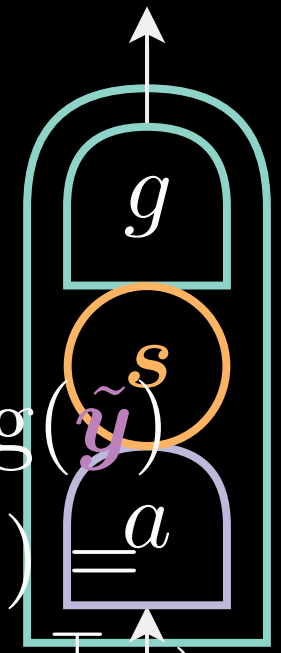
$$\frac{\partial L(w, x, y)}{\partial W_y} = \underbrace{\frac{\partial L(w, x, y)}{\partial o}}_{-y^\top} \frac{\partial g}{\partial s} \frac{\partial a}{\partial W_y}$$

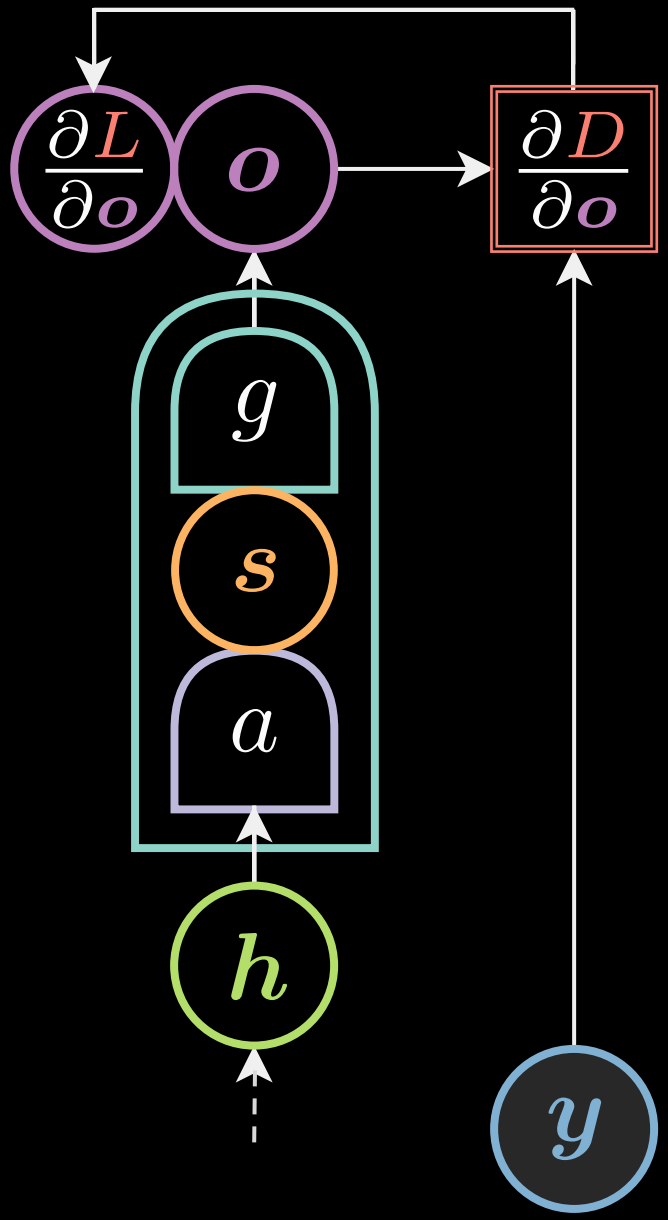
$$w \doteq \{W_h, b_h, W_y, b_y\}$$

decoder

$$o = \log(\tilde{y})$$

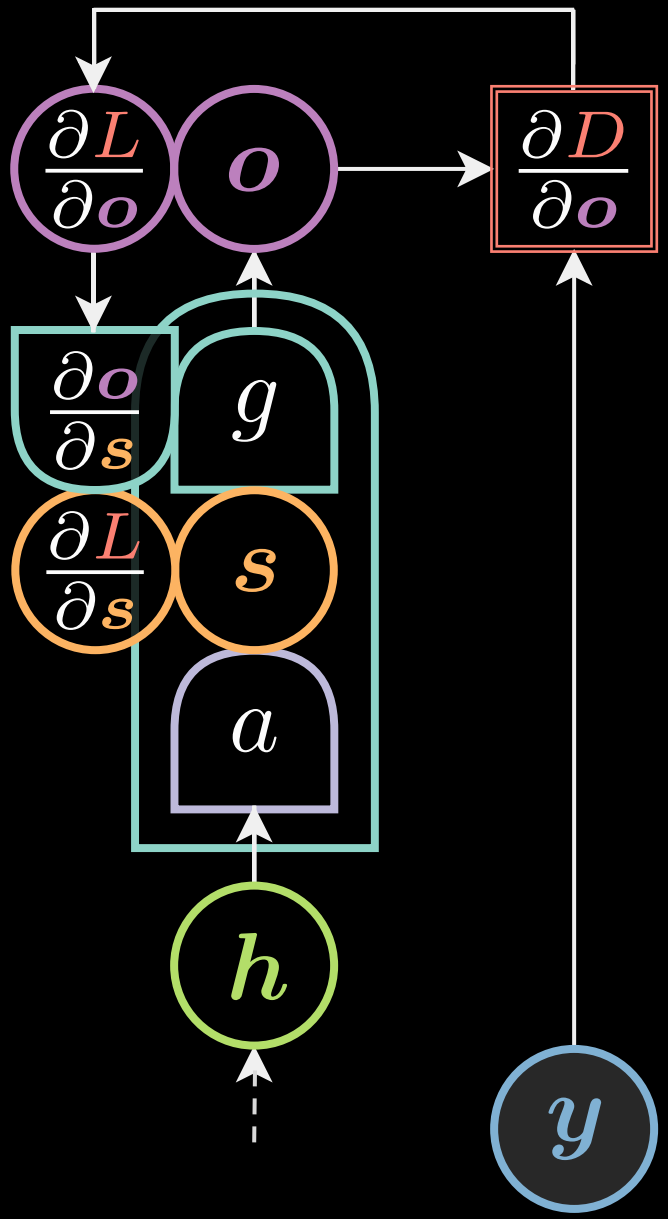
$$C(y, \tilde{y}) = -\log(y^\top \tilde{y})$$



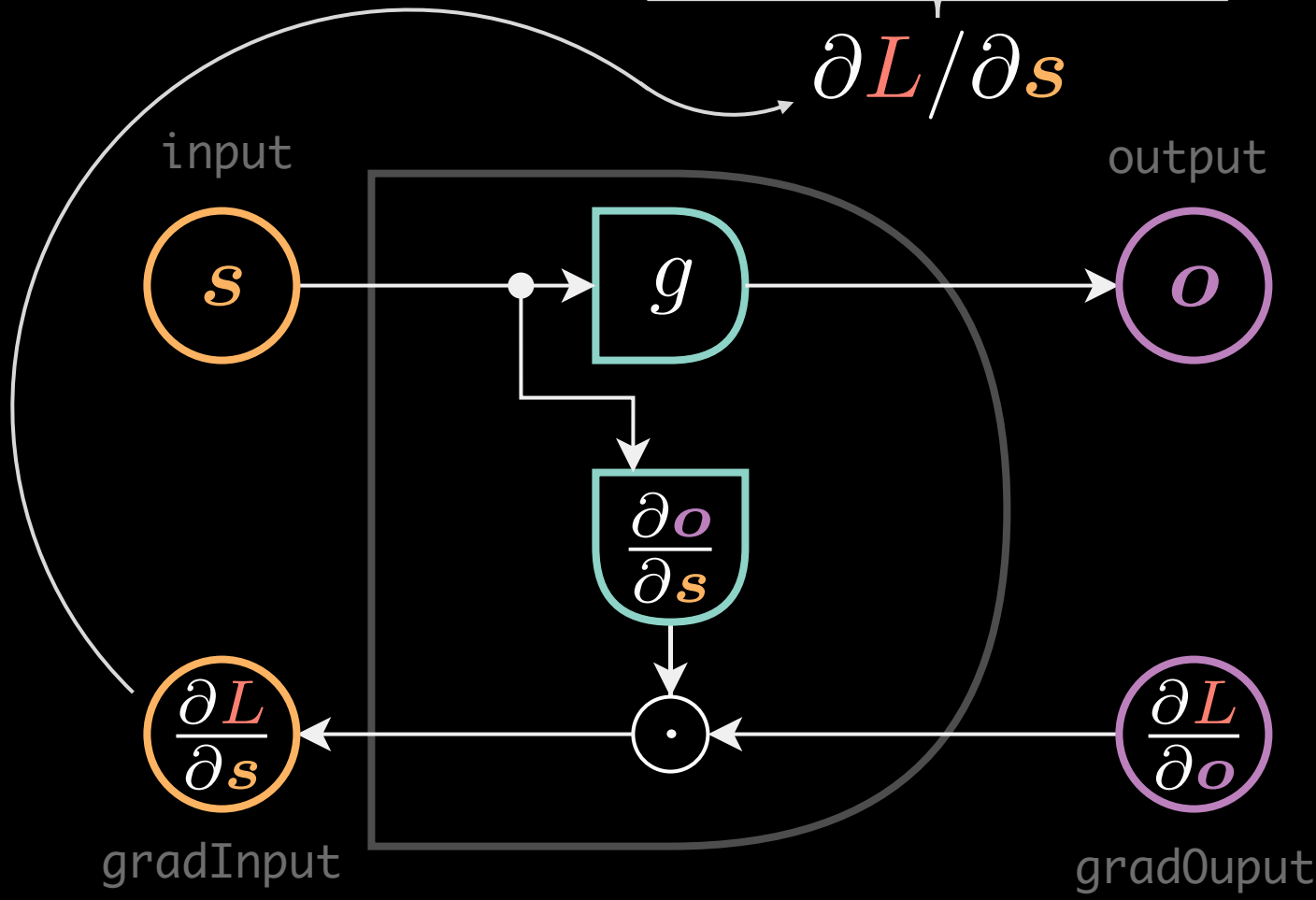


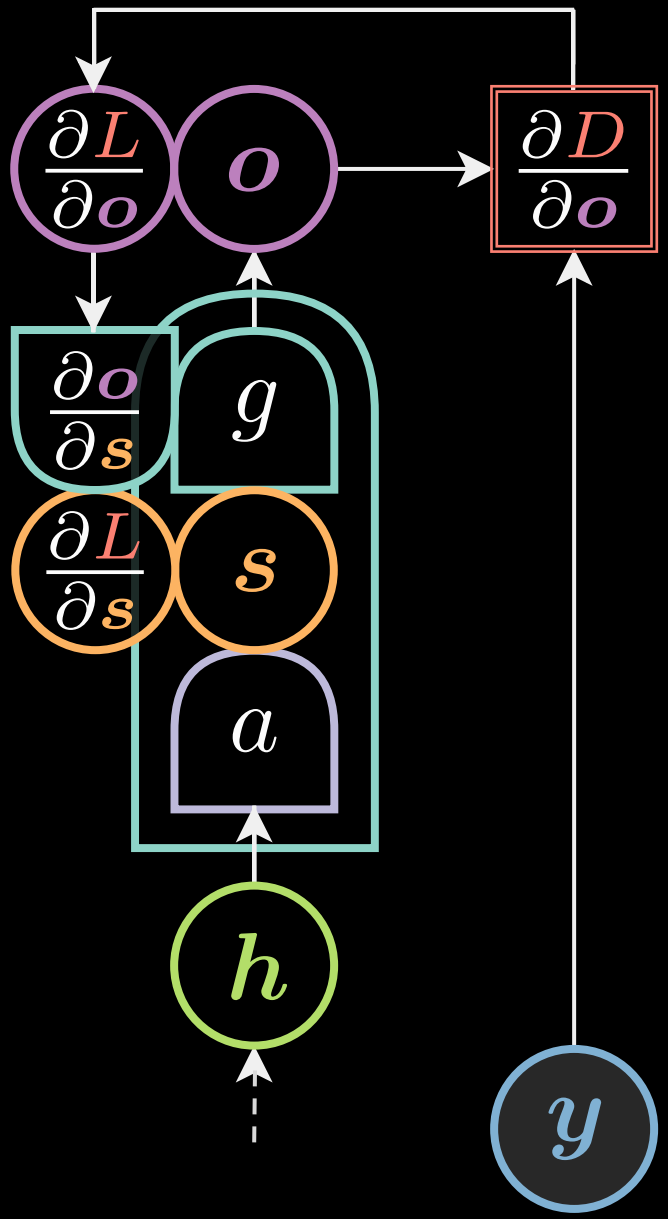
$$\frac{\partial L(w, x, y)}{\partial W_y} = \underbrace{\frac{\partial L(w, x, y)}{\partial o}}_{-y^\top} \frac{\partial g}{\partial s} \frac{\partial a}{\partial W_y}$$

$$w \doteq \{W_h, b_h, W_y, b_y\}$$

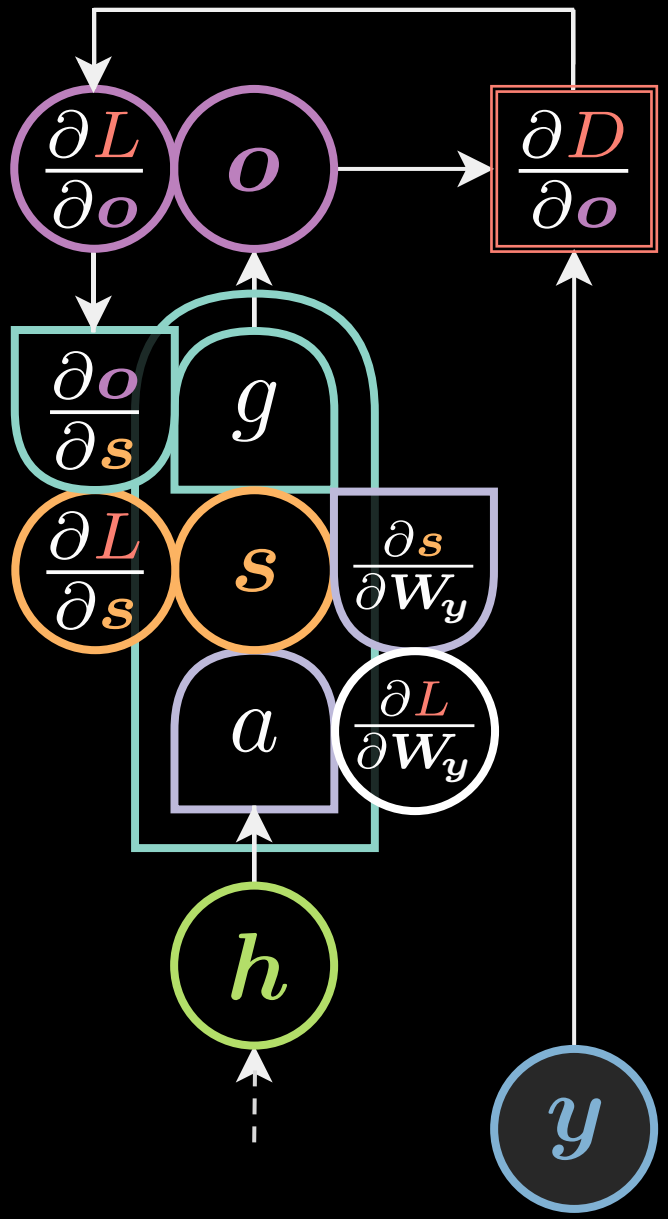


$$\frac{\partial L(w,x,y)}{\partial W_y} = \underbrace{\frac{\partial L(w,x,y)}{\partial o}\frac{\partial g}{\partial s}}_{\partial L/\partial s}\frac{\partial a}{\partial W_y}$$

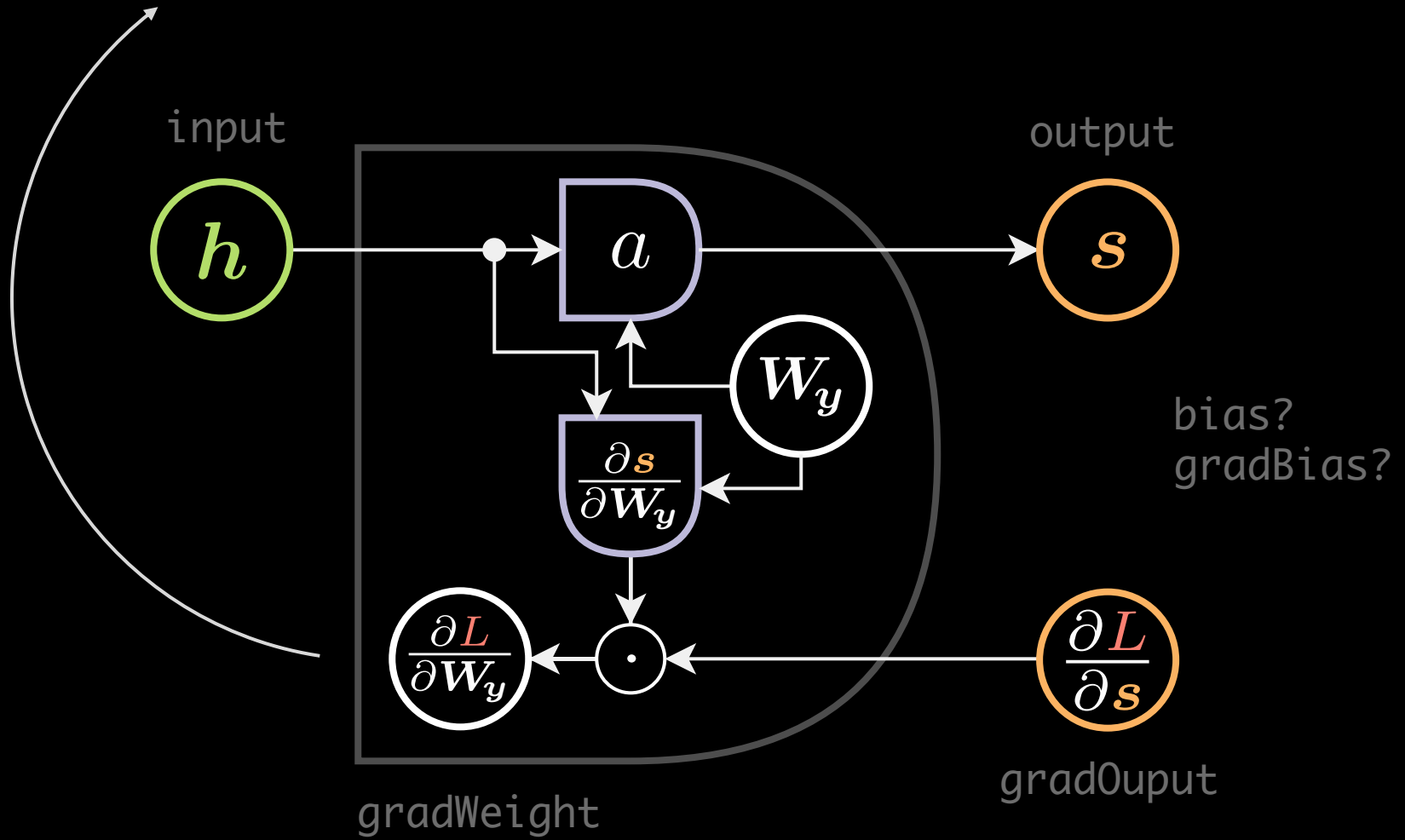


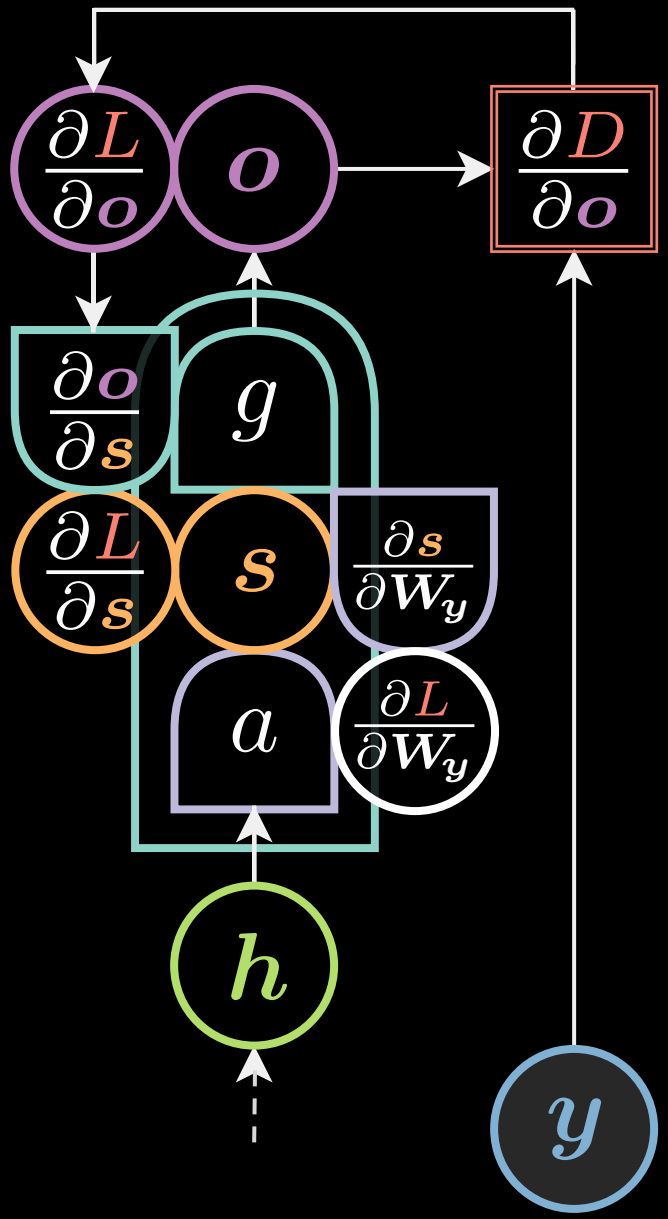


$$\frac{\partial L(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{W}_{\boldsymbol{y}}} = \underbrace{\frac{\partial L(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{o}} \frac{\partial g}{\partial \boldsymbol{s}}}_{\partial L / \partial \boldsymbol{s}} \frac{\partial a}{\partial \boldsymbol{W}_{\boldsymbol{y}}}$$

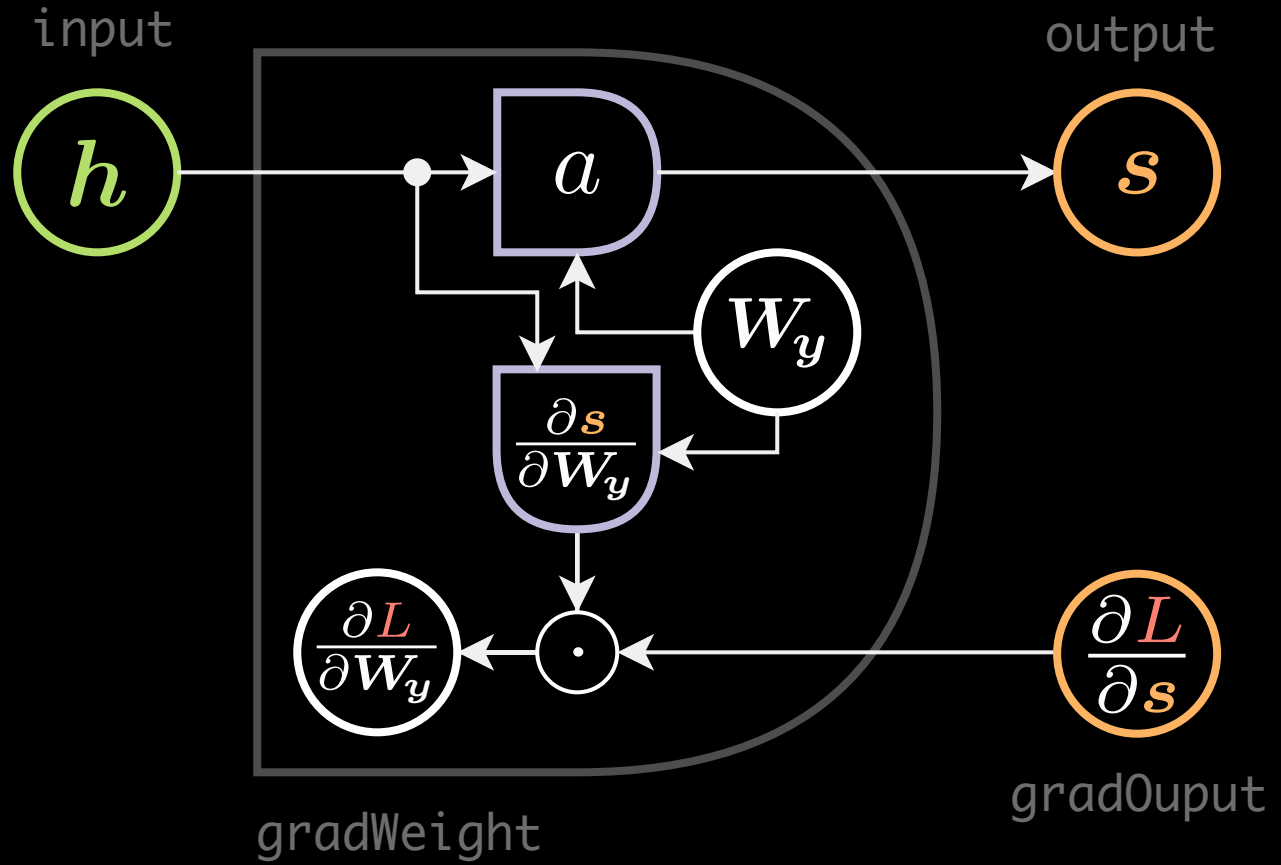


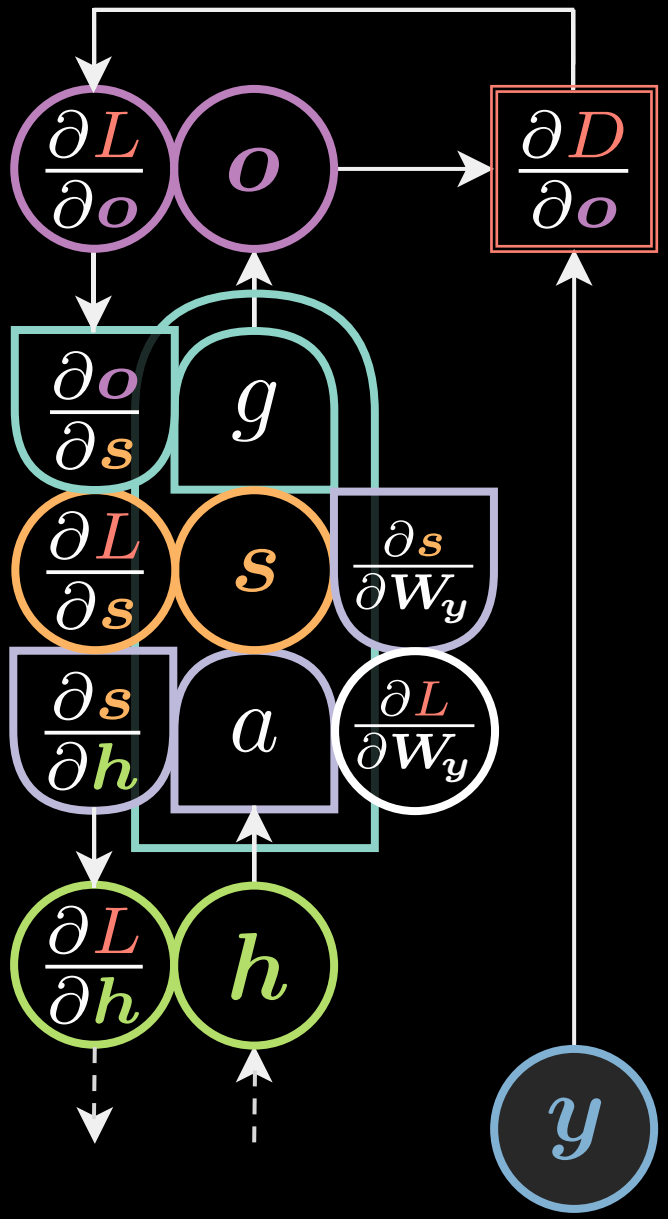
$$\frac{\partial L(w,x,y)}{\partial W_y} = \frac{\partial L(w,x,y)}{\partial s}\frac{\partial a}{\partial W_y}$$



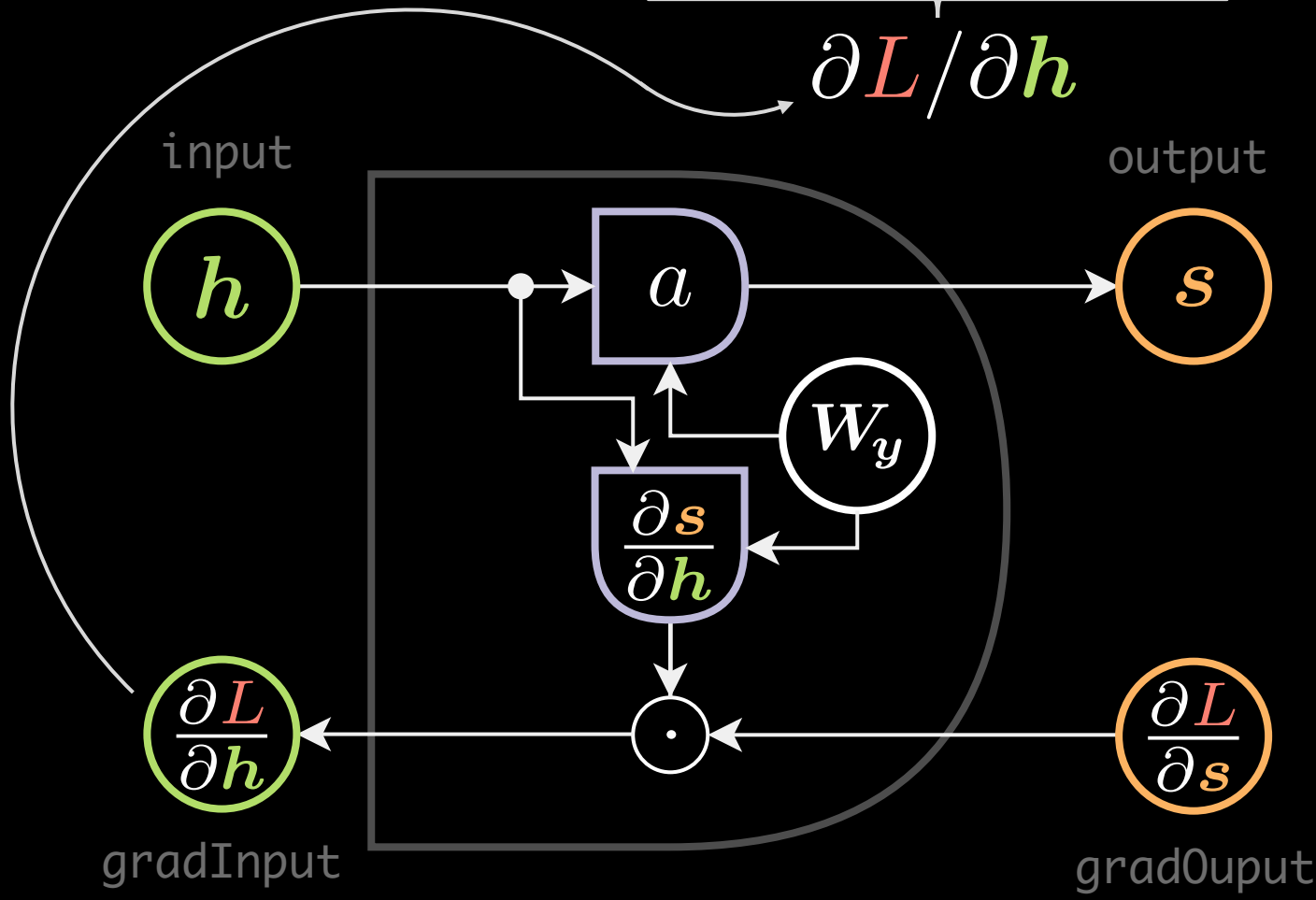


$$\frac{\partial L(w,x,y)}{\partial W_h} = \frac{\partial L(w,x,y)}{\partial s}\frac{\partial a}{\partial h}\frac{\partial h}{\partial W_h}$$





$$\frac{\partial L(w,x,y)}{\partial W_h} = \underbrace{\frac{\partial L(w,x,y)}{\partial s}\frac{\partial a}{\partial h}}_{\partial L/\partial h}\frac{\partial h}{\partial W_h}$$



Actual softmax and softmin

Fixing broken nomenclature

Softer maximum and minimum (I)

$$\boldsymbol{e} = [e_1\ e_2\ \cdots\ e_N]^\top \in \mathbb{R}^N \qquad [\text{soft}]\text{m}**_{[\beta]} : \mathbb{R}^N \to \mathbb{R}$$

optional

coldness

$$\text{softmax}_\beta(\boldsymbol{e}) \doteq \frac{1}{\beta} \log \sum_{n=1}^{N} \exp(\beta e_n) \xrightarrow{\beta \to +\infty} \max(\boldsymbol{e})$$

$$\text{softmin}_\beta(\boldsymbol{e}) \doteq -\frac{1}{\beta} \log \sum_{n=1}^{N} \exp(-\beta e_n)$$

$$= -\,\text{softmax}_\beta(-\boldsymbol{e}) \xrightarrow{\beta \to +\infty} \min(\boldsymbol{e})$$

Softer maximum and minimum (II)

optional

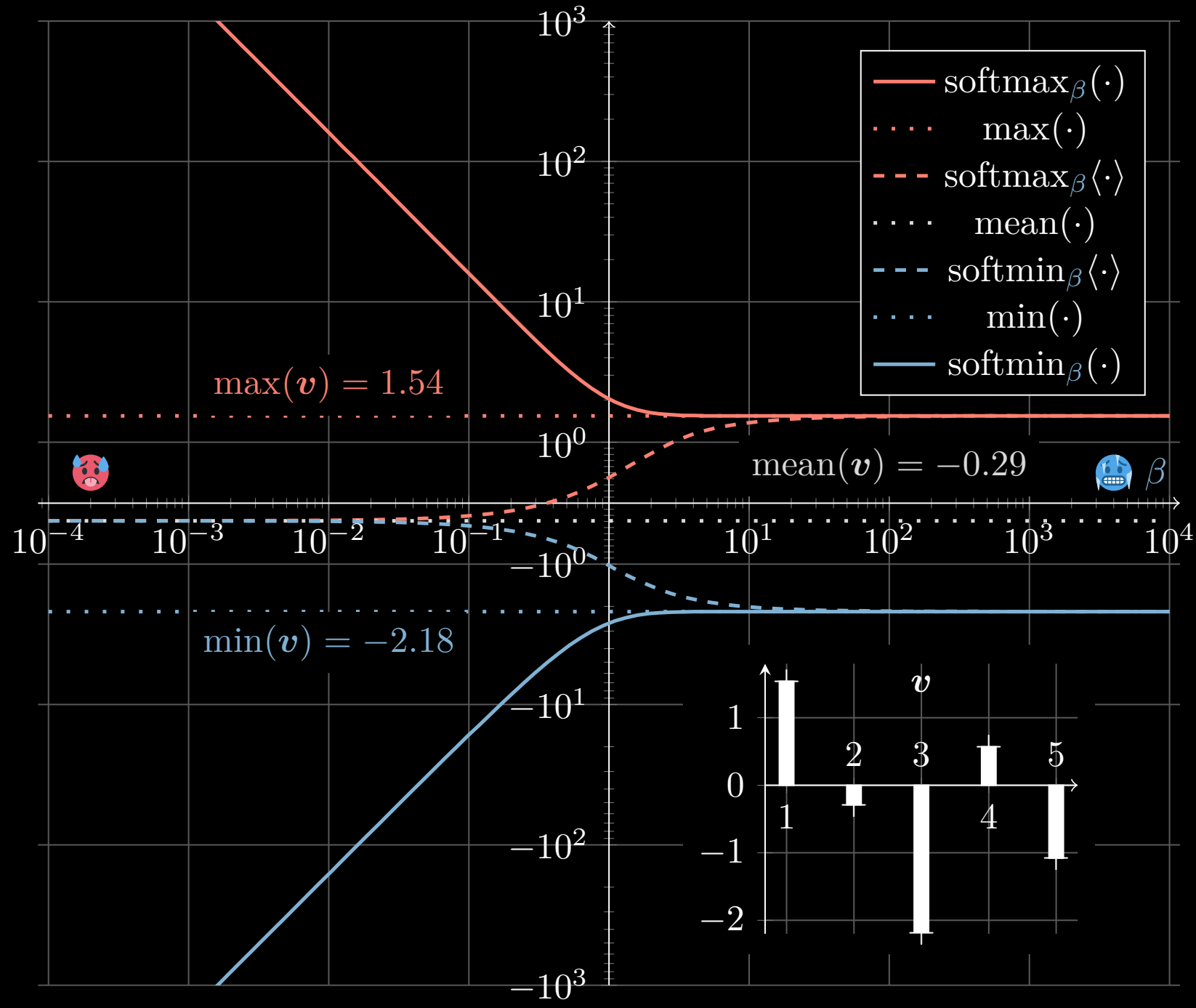
$$\boldsymbol{e} = [e_1\ e_2\ \cdots\ e_N]^\top \in \mathbb{R}^N \qquad [\text{soft}]\text{m}**_{[\beta]} : \mathbb{R}^N \to \mathbb{R}$$

coldness

$$\text{softmax}_\beta\langle \boldsymbol{e} \rangle \doteq \frac{1}{\beta} \log \frac{1}{N} \sum_{n=1}^{N} \exp(\beta e_n) \xrightarrow{\beta \to +\infty} \max(\boldsymbol{e})$$

$$\text{softmin}_\beta\langle \boldsymbol{e} \rangle \doteq -\frac{1}{\beta} \log \frac{1}{N} \sum_{n=1}^{N} \exp(-\beta e_n) \xrightarrow[\beta \to 0^+]{} \text{mean}(\boldsymbol{e})$$

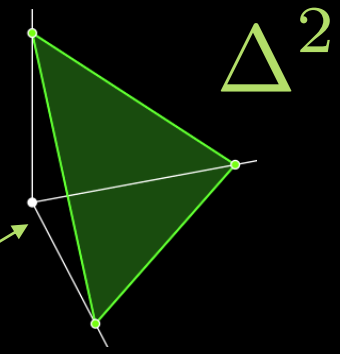
$$= -\,\text{softmax}_\beta\langle -\boldsymbol{e} \rangle \xrightarrow{\beta \to +\infty} \min(\boldsymbol{e})$$


10^3
10^2
10^1
10^0
-10^0
-10^1
-10^2
-10^3
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^1
10^2
10^3
10^4
β
$\max(\boldsymbol{v}) = 1.54$
$\min(\boldsymbol{v}) = -2.18$
$\text{mean}(\boldsymbol{v}) = -0.29$
$\text{softmax}_\beta(\cdot)$
$\max(\cdot)$
$\text{softmax}_\beta\langle\cdot\rangle$
$\text{mean}(\cdot)$
$\text{softmin}_\beta\langle\cdot\rangle$
$\min(\cdot)$
$\text{softmin}_\beta(\cdot)$
$\boldsymbol{v}$
1
2
3
4
5
0
−1
−2

Softargmax and softargmin

Yeah, too many people drop the ‘arg’ part…

Softer arg- maximum and minimum (I)



optional

coldness

$$\boldsymbol{e} \in \mathbb{R}^N \qquad [\text{soft}]\text{argm}**_{[\beta]} : \mathbb{R}^N \to \boxed{\Delta^{N-1}} \subseteq [0,1]^N$$

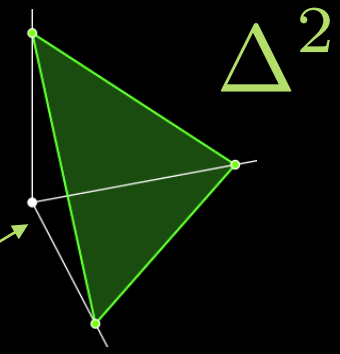
exponentiation of each component

$$\text{softargmax}_\beta(\boldsymbol{e}) \doteq \frac{\exp(\beta \boldsymbol{e})}{\sum_{n=1}^{N} \exp(\beta e_n)} \xrightarrow{\beta \to +\infty} \arg\max(\boldsymbol{e})$$

$$\text{softargmin}_\beta(\boldsymbol{e}) \doteq \frac{\exp(-\beta \boldsymbol{e})}{\sum_{n=1}^{N} \exp(-\beta e_n)} \xrightarrow[\beta \to 0^+]{\beta \to 0\times} \frac{1}{N}\mathbf{1} \in \mathbb{R}^N$$

$$= \text{softargmax}_\beta(-\boldsymbol{e}) \xrightarrow{\beta \to +\infty} \arg\min(\boldsymbol{e})$$

Softer arg- maximum and minimum (II)



optional

coldness

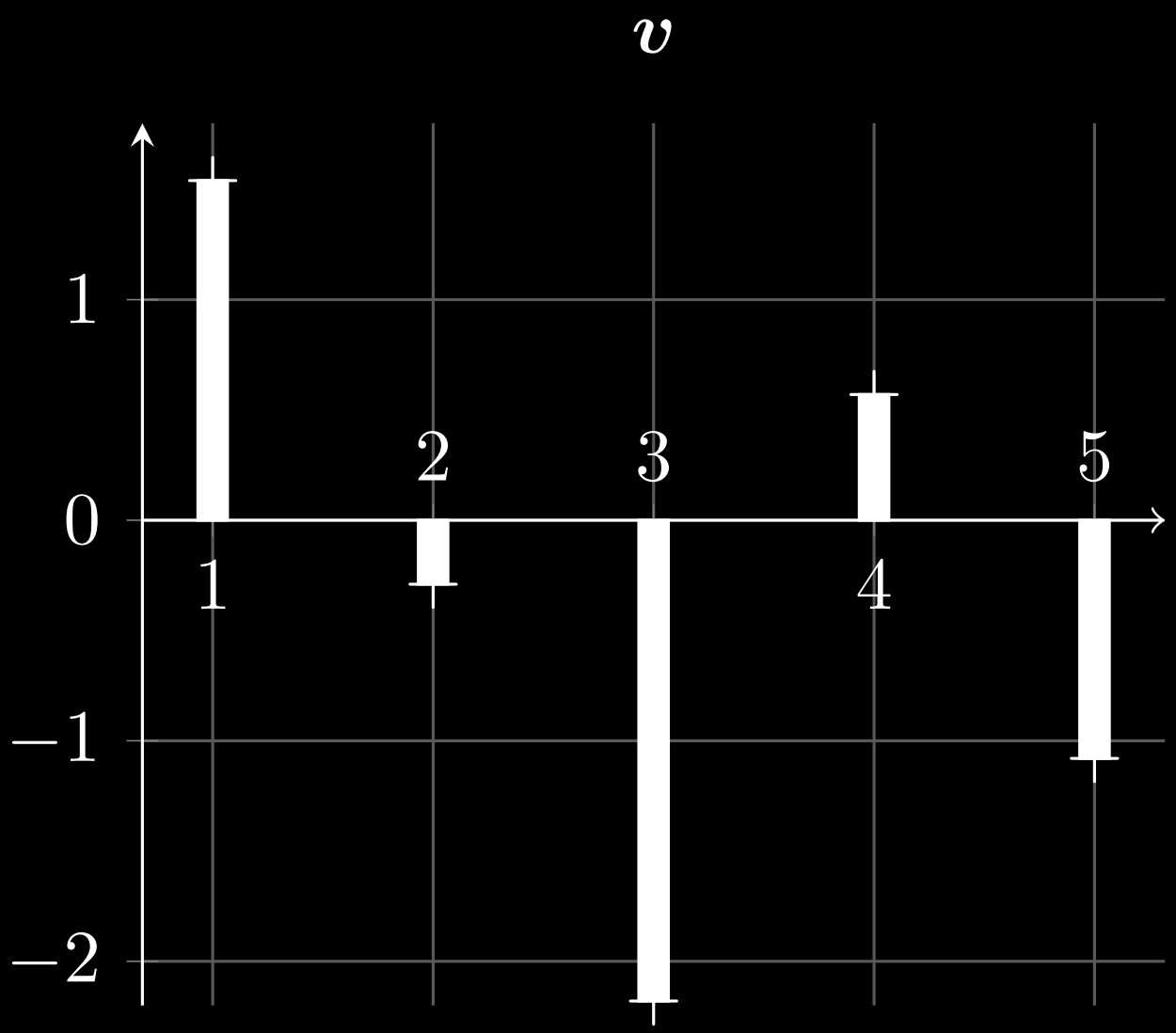
$$\boldsymbol{e} \in \mathbb{R}^N \qquad [\mathrm{soft}]\mathrm{argm}{*}{*}_{[\beta]} : \mathbb{R}^N \to \boxed{\Delta^{N-1}} \subseteq [0,1]^N$$

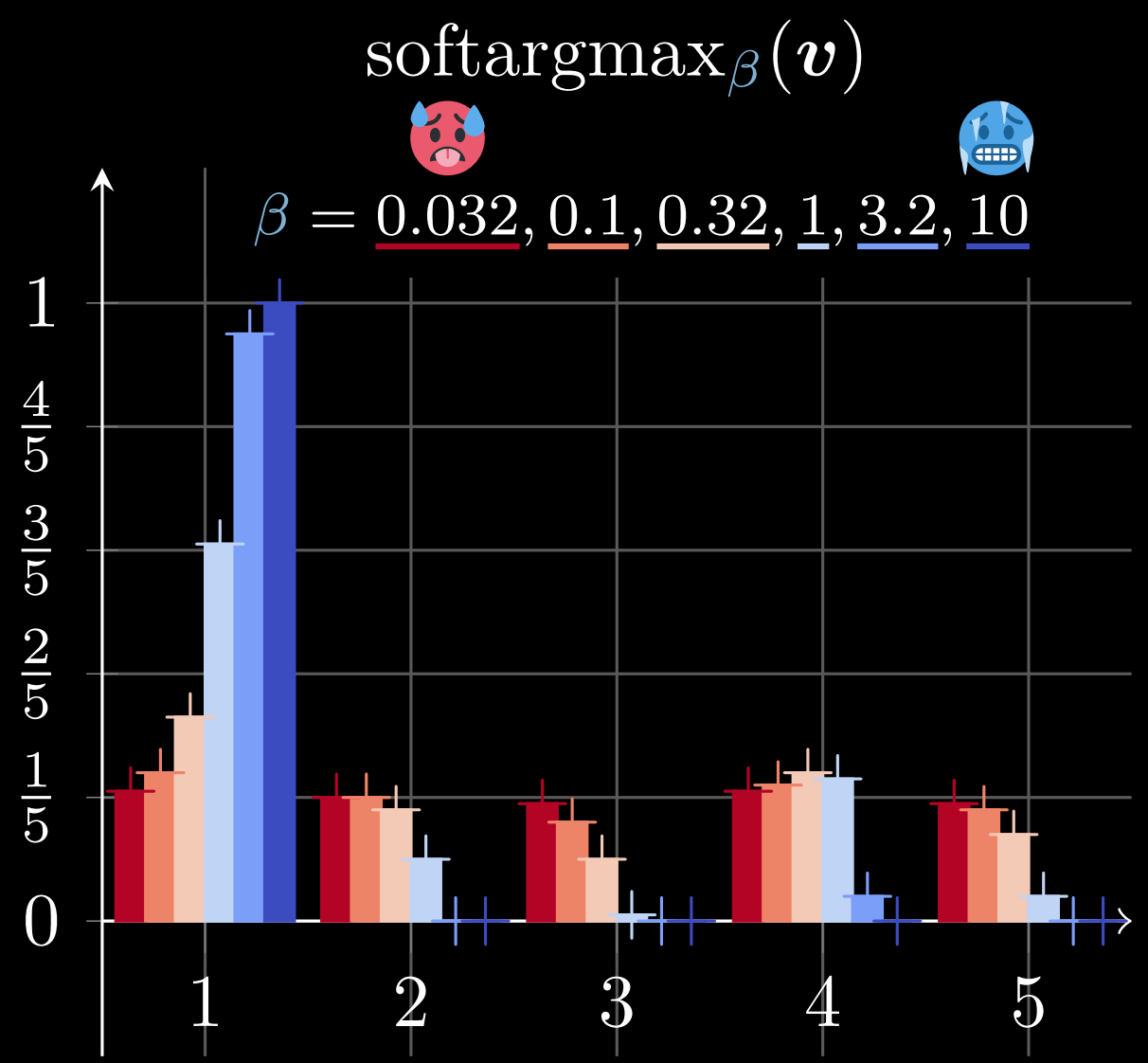
exponentiation of each component

$$\mathrm{softargmax}_\beta(\boldsymbol{e}) \doteq \frac{\exp(\beta \boldsymbol{e})}{\sum_{n=1}^{N} \exp(\beta e_n)} \xrightarrow{\beta \to +\infty} \arg\max(\boldsymbol{e})$$

$$= \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{e}} \mathrm{softmax}_\beta(\boldsymbol{e}) = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{e}} \mathrm{softmax}_\beta \langle \boldsymbol{e} \rangle$$

$$\arg\max(\boldsymbol{e}) = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{e}} \max(\boldsymbol{e})$$


$\boldsymbol{v}$
1
0
−1
−2
1
2
3
4
5


$\mathrm{softargmax}_\beta(\boldsymbol{v})$
$\beta = 0.032, 0.1, 0.32, 1, 3.2, 10$
1
$\frac{4}{5}$
$\frac{3}{5}$
$\frac{2}{5}$
$\frac{1}{5}$
0
1
2
3
4
5

Negative log of the softargmax / softargmin

$$-\frac{1}{\beta}\log[\text{softargmax}_\beta(\boldsymbol{e})] = -\boldsymbol{e} + \frac{1}{\beta}\log\sum_{n=1}^{N}\exp(\beta e_n)$$

distance to the soft maximum

scalar

$$= \text{softmax}_\beta(\boldsymbol{e}) - \boldsymbol{e}$$

N dimensional

$$-\frac{1}{\beta}\log[\text{softargmin}_\beta(\boldsymbol{e})] = \boldsymbol{e} + \frac{1}{\beta}\log\sum_{n=1}^{N}\exp(-\beta e_n)$$

distance to the soft minimum

$$= \boldsymbol{e} - \text{softmin}_\beta(\boldsymbol{e})$$

scalar

N dimensional

Interesting properties

$$\mathrm{softmax}_\beta(\boldsymbol{e} + c\mathbf{1}) = \mathrm{softmax}_\beta(\boldsymbol{e}) + c$$

$$(a + c) \oplus (b + c) = (a \oplus b) + c \qquad ac + bc = (a + b) \cdot c$$

$$(\mathbb{R} \cup \{-\infty\}), \oplus, \otimes), \quad a \oplus b = \mathrm{softmax}_\beta(\begin{bmatrix} a \\ b \end{bmatrix})$$

log addition

log semiring

$$a \otimes b = a + b$$

log multiplication

$$\mathrm{softargmax}_\beta(e + c\mathbf{1}) = \mathrm{softargmax}_\beta(e)$$