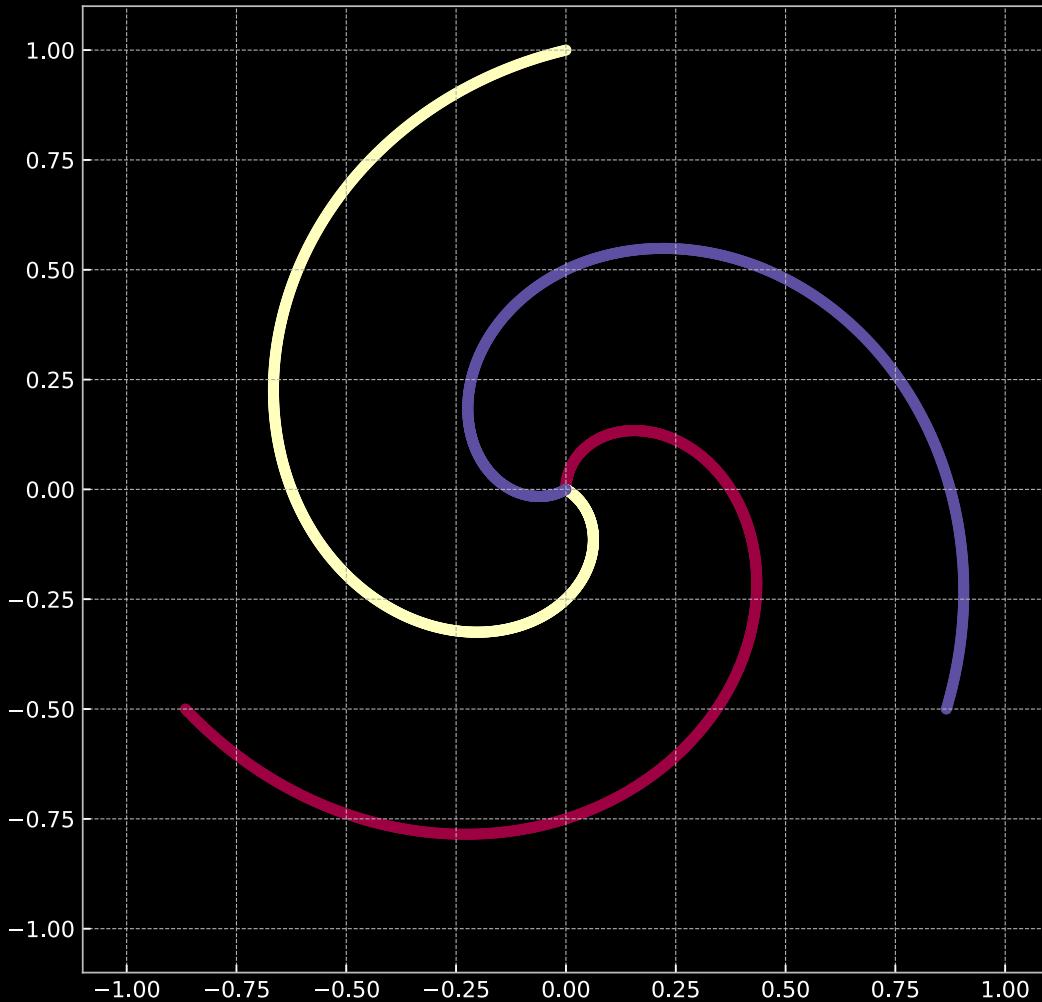


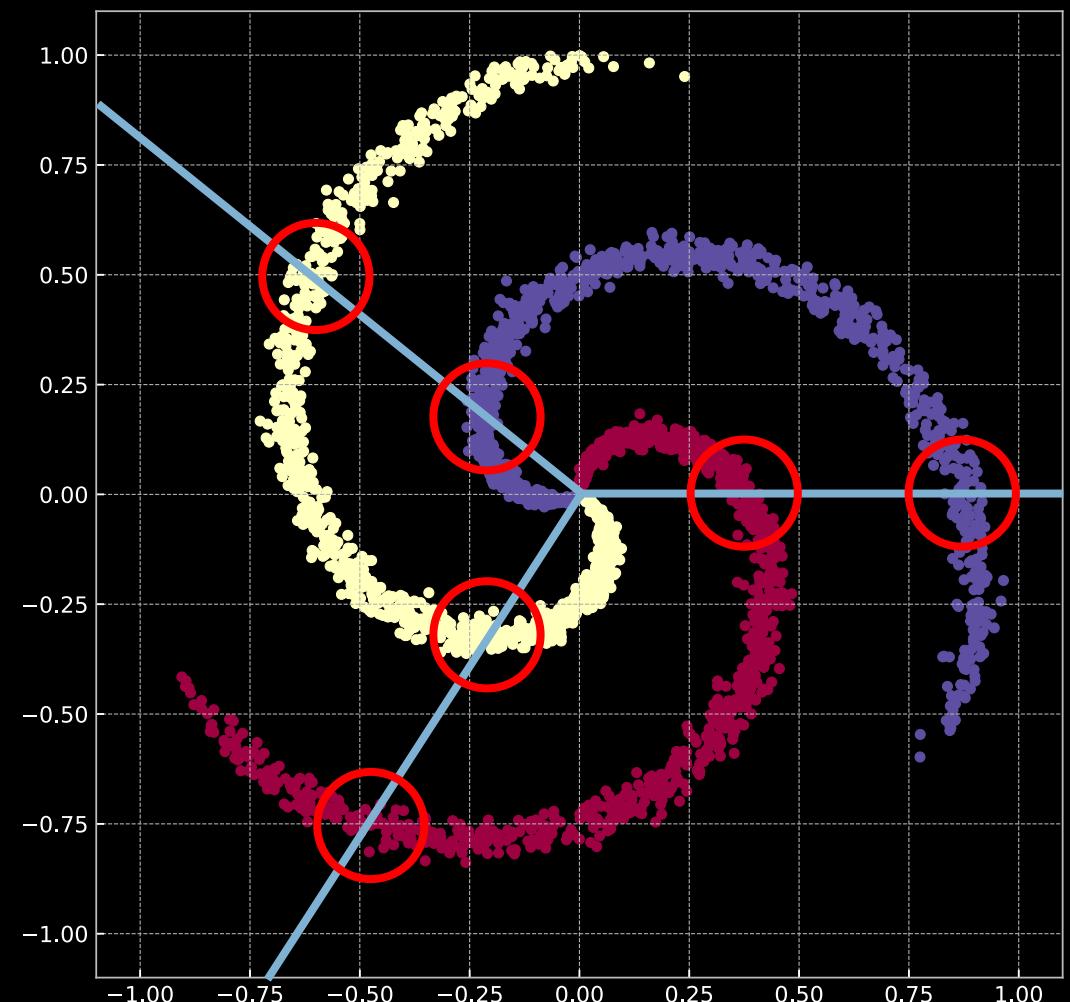
ANN – supervised learning

Classification



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

$0 \leq t \leq 1, \quad k = 1, \dots, K$



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

Classification train data

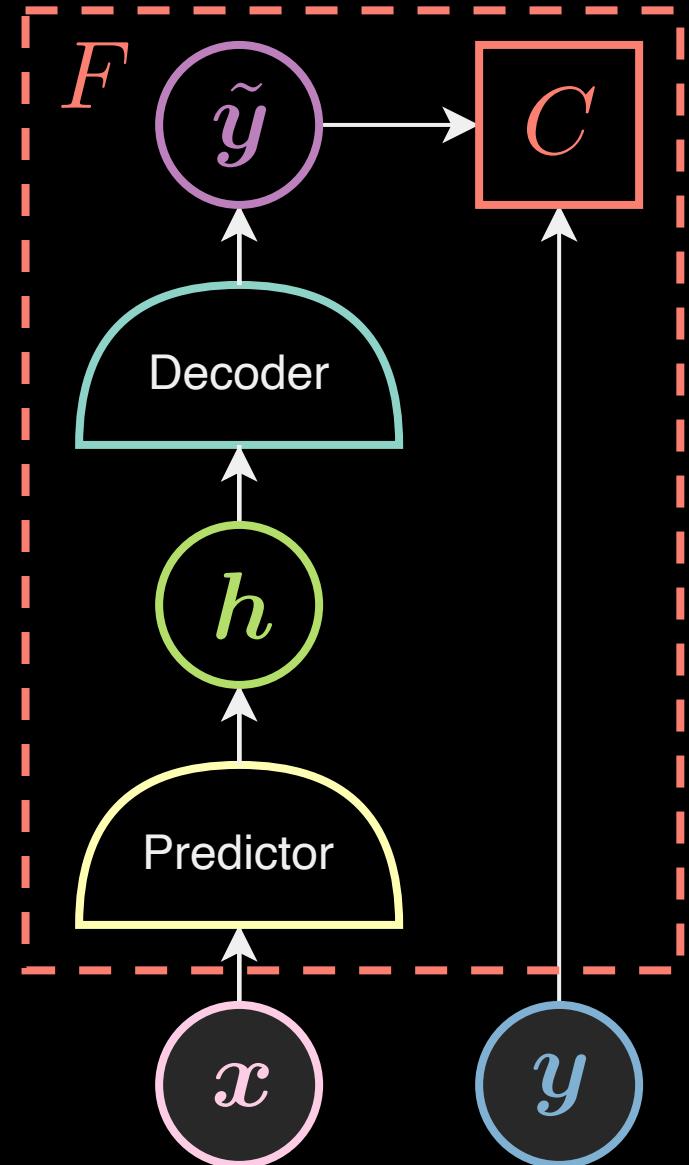
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{matrix} \text{1-hot encoding} \\ K = 3 \end{matrix}$$

$$X = \begin{bmatrix} \xrightarrow{n} & \xleftarrow{n} \\ \cdots & \cdots \\ \xrightarrow{n} & \xleftarrow{n} \end{bmatrix} \quad P$$
$$\begin{bmatrix} \xrightarrow{n} & \xleftarrow{n} \\ \cdots & \cdots \\ \xrightarrow{n} & \xleftarrow{n} \end{bmatrix} \quad P$$

$$Y = \begin{bmatrix} \xrightarrow{K} & \xleftarrow{K} \\ \cdots & \cdots \\ \xrightarrow{K} & \xleftarrow{K} \end{bmatrix} \quad P$$
$$\begin{bmatrix} \xrightarrow{K} & \xleftarrow{K} \\ \cdots & \cdots \\ \xrightarrow{K} & \xleftarrow{K} \end{bmatrix} \quad P$$

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

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



Neural network (inference)

$$\mathbf{h} = \text{Pred}(\mathbf{x}) = f(\mathbf{W}_\mathbf{h}\mathbf{x} + \mathbf{b}_\mathbf{h})$$

$$\tilde{\mathbf{y}} = \text{Dec}(\mathbf{h}) = g(\mathbf{W}_{\tilde{\mathbf{y}}}\mathbf{h} + \mathbf{b}_{\tilde{\mathbf{y}}})$$

incompatibility
between \mathbf{x} and \mathbf{y}

$$f, g = (\cdot)^+, \sigma(\cdot),$$

$$\tanh(\cdot), \text{softargmax}(\cdot)$$

$$F(\mathbf{x}, \mathbf{y}) =$$

$$C(\mathbf{y}, \tilde{\mathbf{y}})$$

distance
between \mathbf{y} and $\tilde{\mathbf{y}}$

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

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

Classification

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

↑
cross entropy / negative log-probability

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

↑
last layer linear sum

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

↑
data set

↑
badness given
a data sample

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

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

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

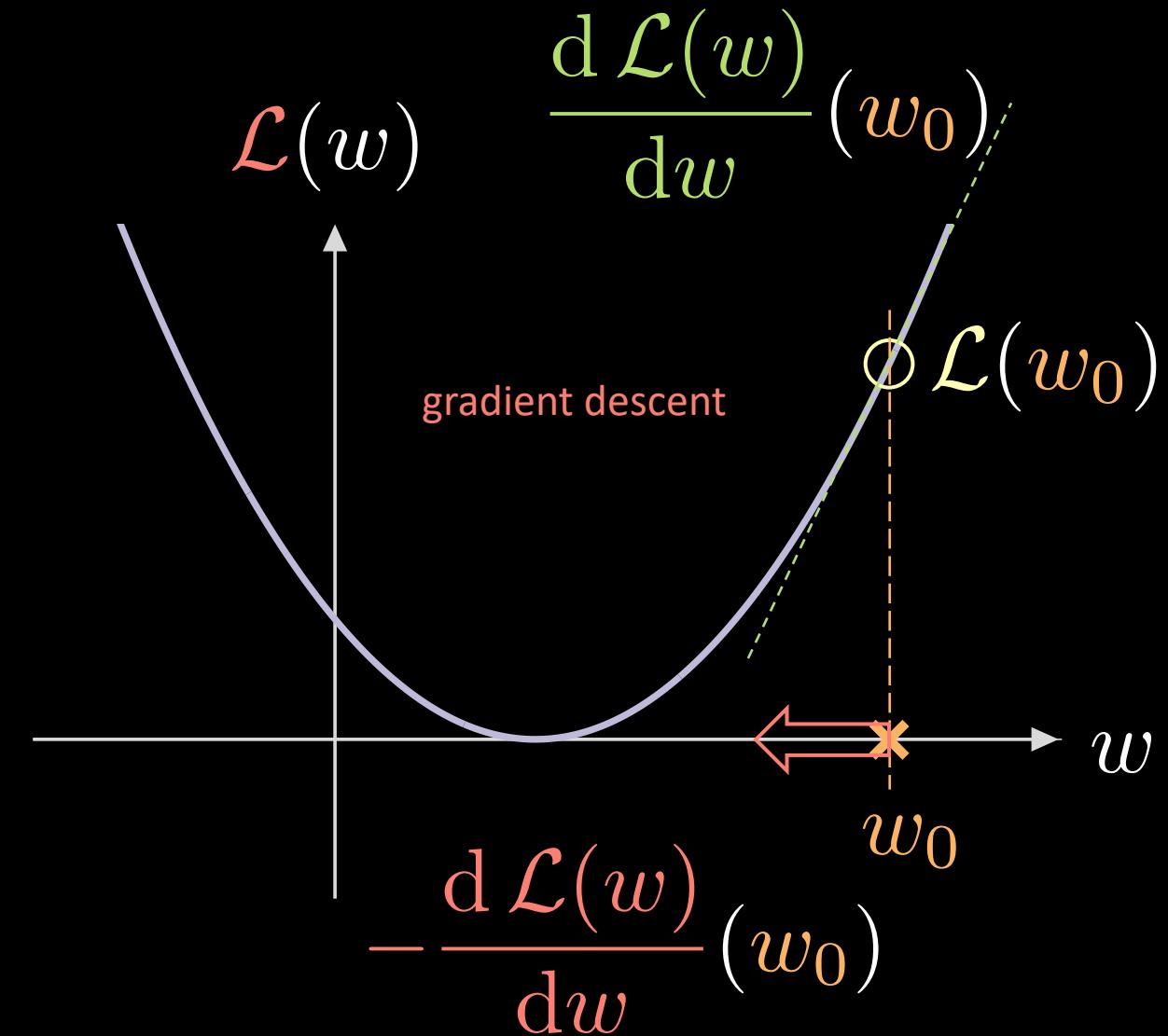
Neural net training

$$\boldsymbol{w} \doteq \{\mathbf{W}_{\mathbf{h}}, \mathbf{b}_{\mathbf{h}}, \mathbf{W}_{\tilde{\mathbf{y}}}, \mathbf{b}_{\tilde{\mathbf{y}}}\}$$

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

$$\frac{\partial \mathcal{L}(\boldsymbol{w})}{\partial \mathbf{W}_{\tilde{\mathbf{y}}}} = \frac{\partial \mathcal{L}(\boldsymbol{w})}{\partial \tilde{\mathbf{y}}} \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{W}_{\tilde{\mathbf{y}}}}$$

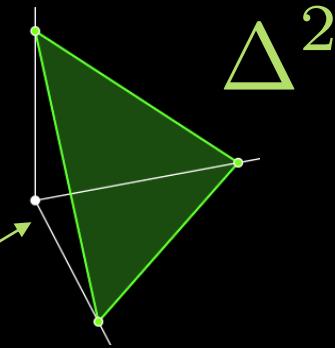
$$\frac{\partial \mathcal{L}(\boldsymbol{w})}{\partial \mathbf{W}_{\mathbf{h}}} = \frac{\partial \mathcal{L}(\boldsymbol{w})}{\partial \tilde{\mathbf{y}}} \frac{\partial \tilde{\mathbf{y}}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{W}_{\mathbf{h}}}$$



Softargmax and softargmin

Yeah, too many people drop the 'arg' part...

Softer arg- maximum and minimum (I)



$$\mathbf{e} \in \mathbb{R}^N \xrightarrow{\text{optional}} [\text{soft}] \text{argm} **_{[\beta]} : \mathbb{R}^N \rightarrow \boxed{\Delta^{N-1}} \subseteq [0, 1]^N$$

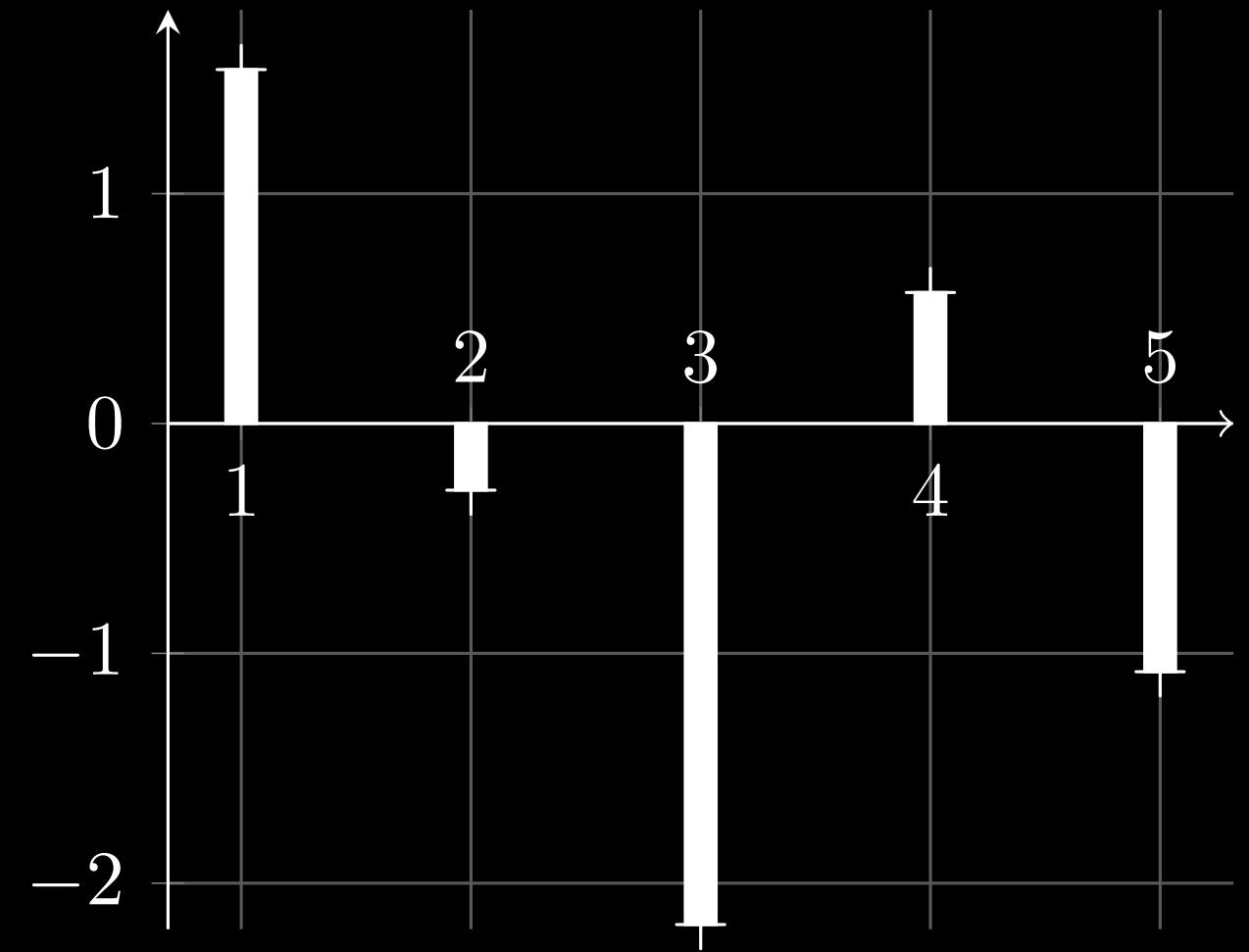
coldness

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

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

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

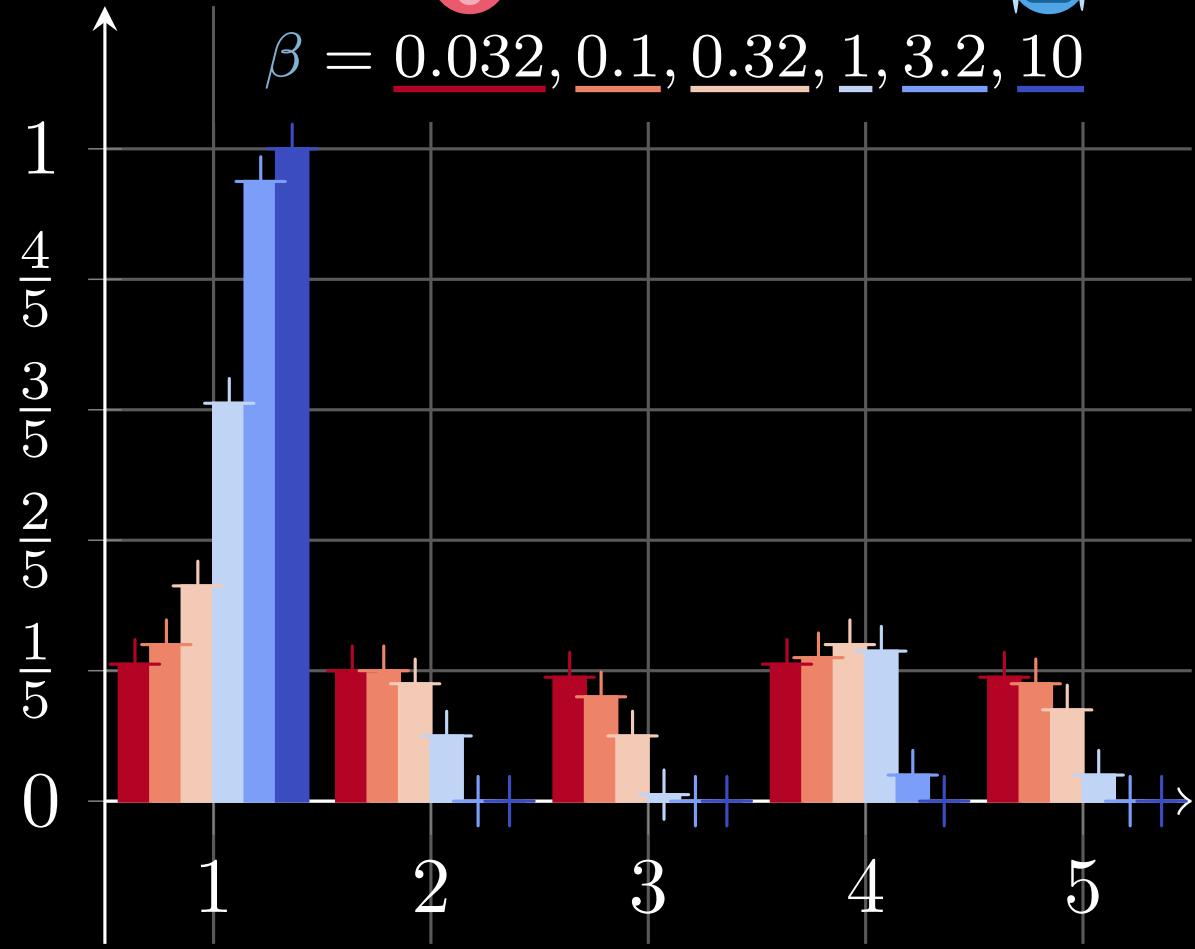
v



$\text{softargmax}_\beta(v)$

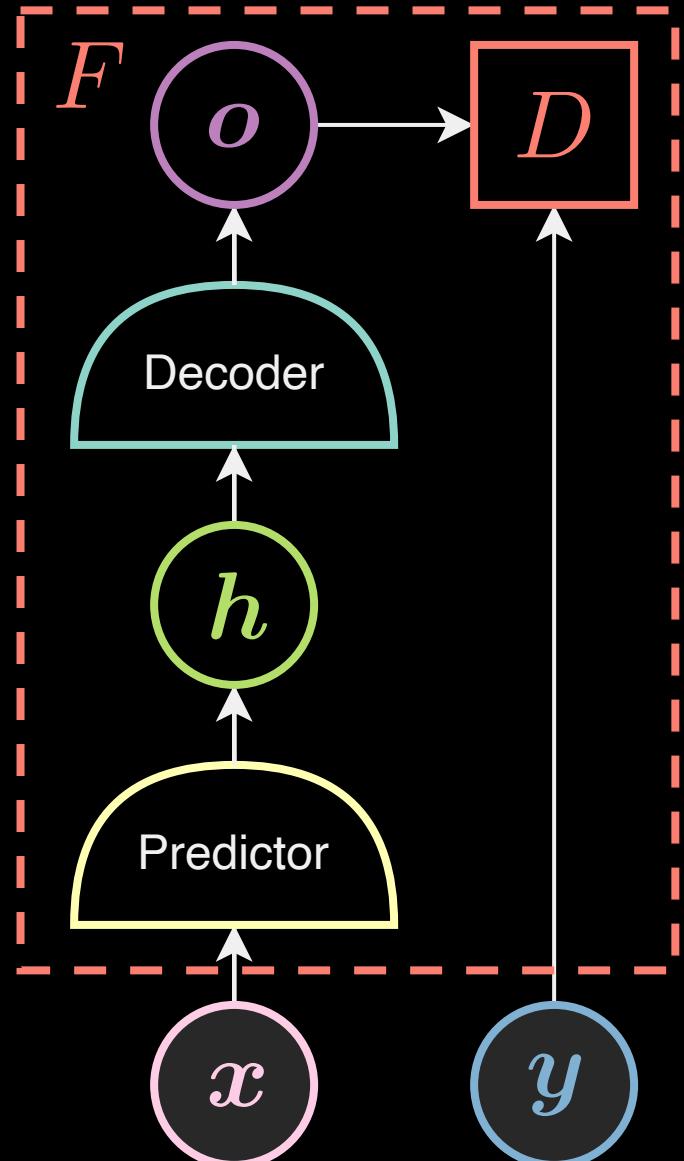


$\beta = \underline{0.032}, \underline{0.1}, \underline{0.32}, \underline{1}, \underline{3.2}, \underline{10}$



Back-propagation

of the gradient



$$\mathbf{h} = f(\mathbf{W}_h \mathbf{x} + \mathbf{b}_h)$$

$$\mathbf{s} = a(\mathbf{h}) = \mathbf{W}_y \mathbf{h} + \mathbf{b}_y$$

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

$$g = \text{logsoftmax}$$

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

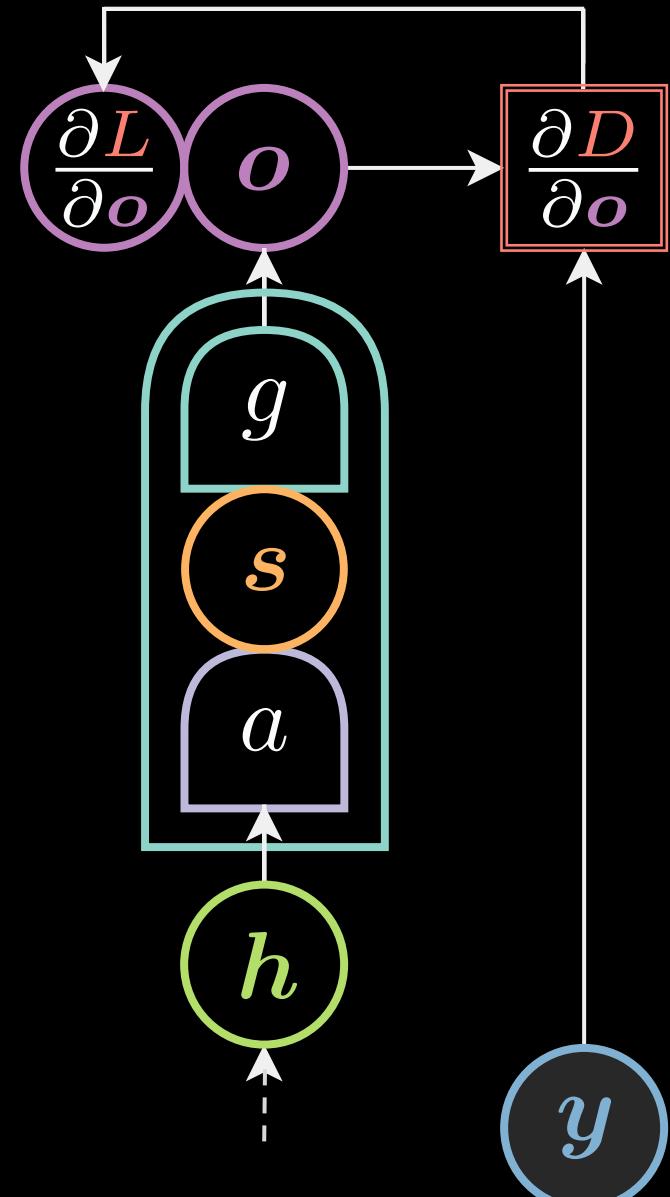
$$\frac{\partial L(\mathbf{w}, \mathbf{x}, \mathbf{y})}{\partial \mathbf{W}_y} = \frac{\partial L(\mathbf{w}, \mathbf{x}, \mathbf{y})}{\partial \mathbf{o}} \frac{\partial g}{\partial \mathbf{s}} \frac{\partial a}{\partial \mathbf{W}_y}$$

$$\mathbf{w} \doteq \{\mathbf{W}_h, \mathbf{b}_h, \mathbf{W}_y, \mathbf{b}_y\}$$

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

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

$$-\log(\mathbf{y}^\top \tilde{\mathbf{y}})$$



$$\mathbf{h} = f(\mathbf{W}_h \mathbf{x} + \mathbf{b}_h)$$

$$\mathbf{s} = a(\mathbf{h}) = \mathbf{W}_y \mathbf{h} + \mathbf{b}_y$$

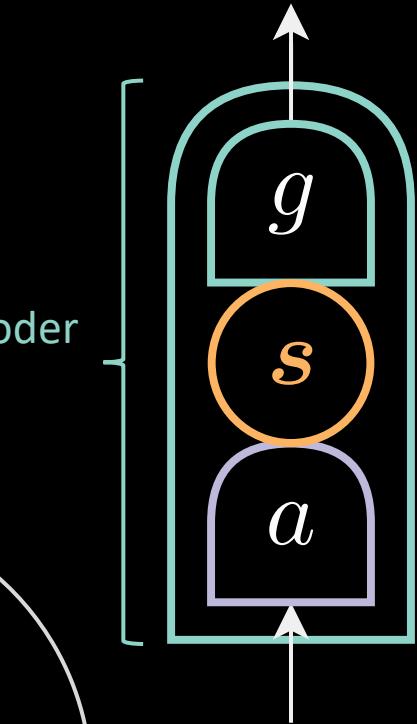
$$\mathbf{o} = g(\mathbf{s})$$

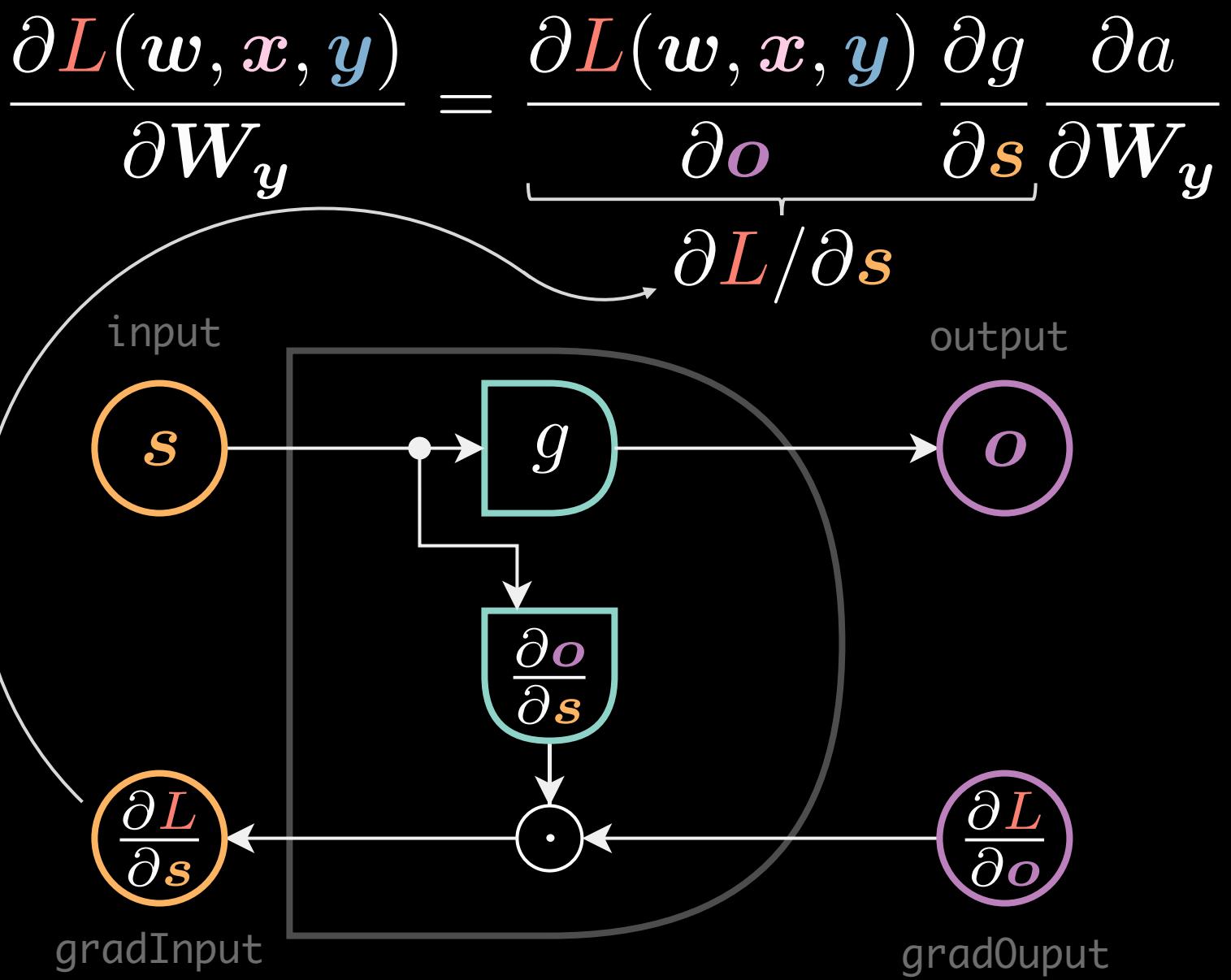
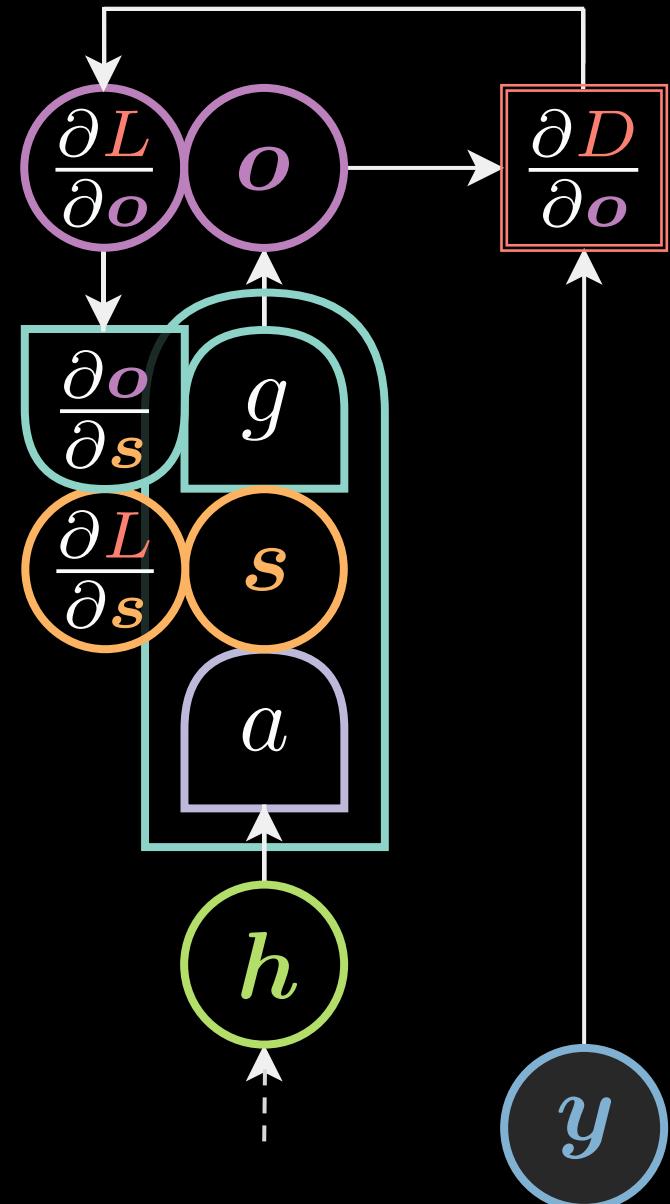
$$g = \text{logsoftmax}$$

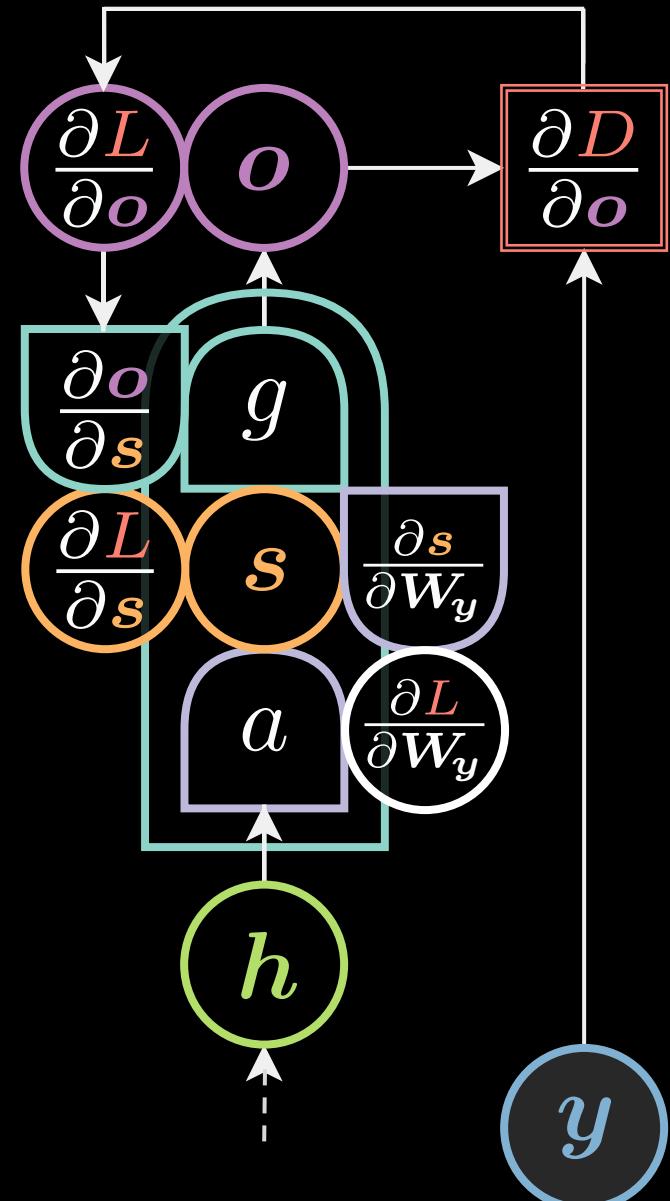
$$L = F = D = -\mathbf{y}^\top \mathbf{o}$$

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

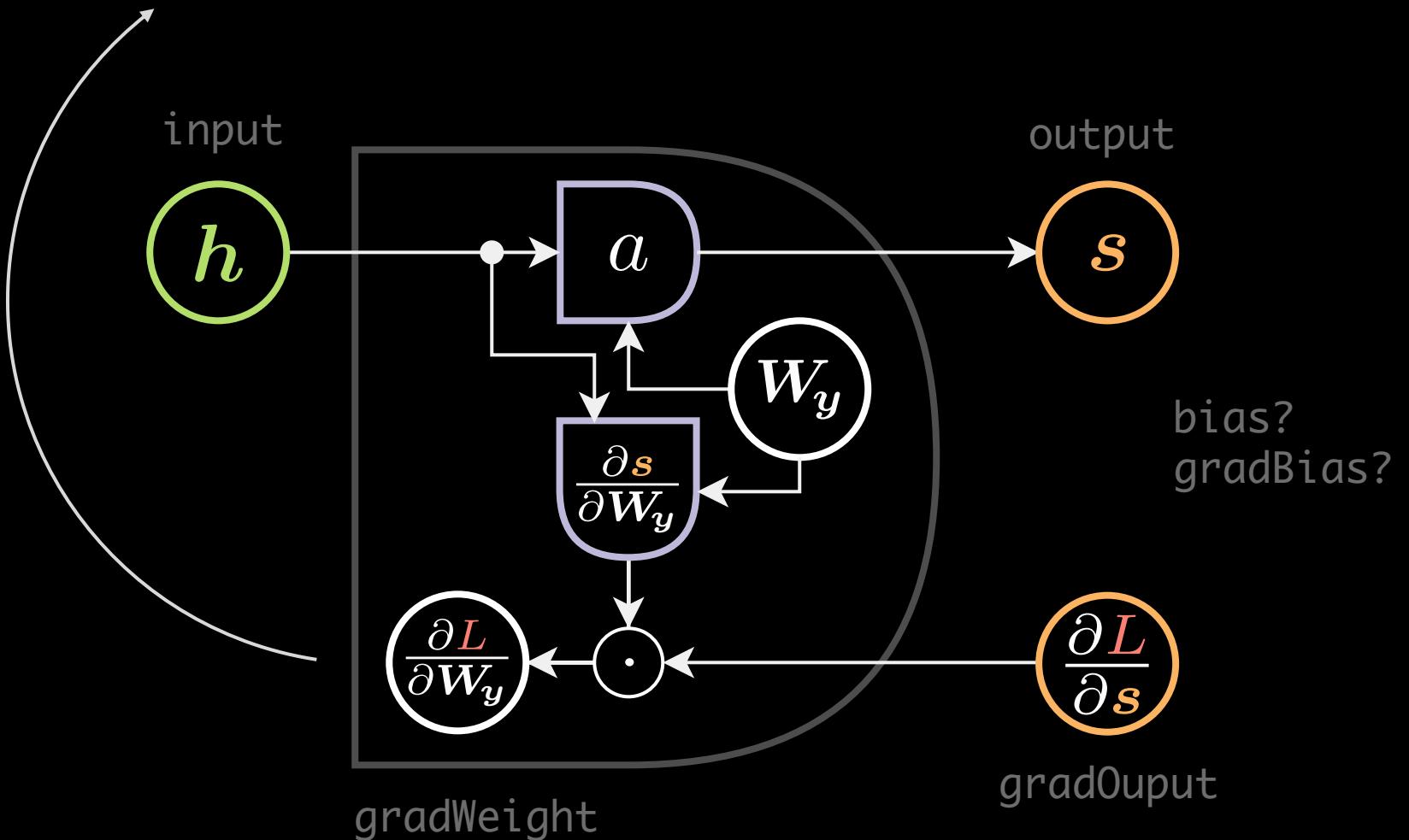
$$\mathbf{w} \doteq \{\mathbf{W}_h, \mathbf{b}_h, \mathbf{W}_y, \mathbf{b}_y\}$$

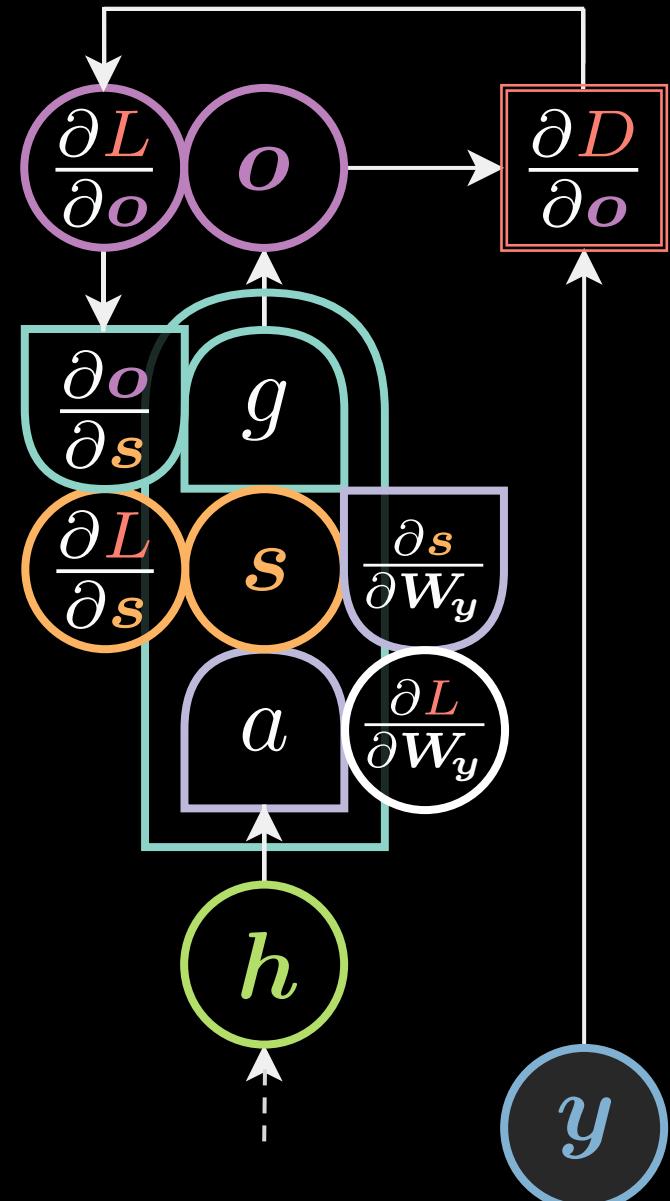




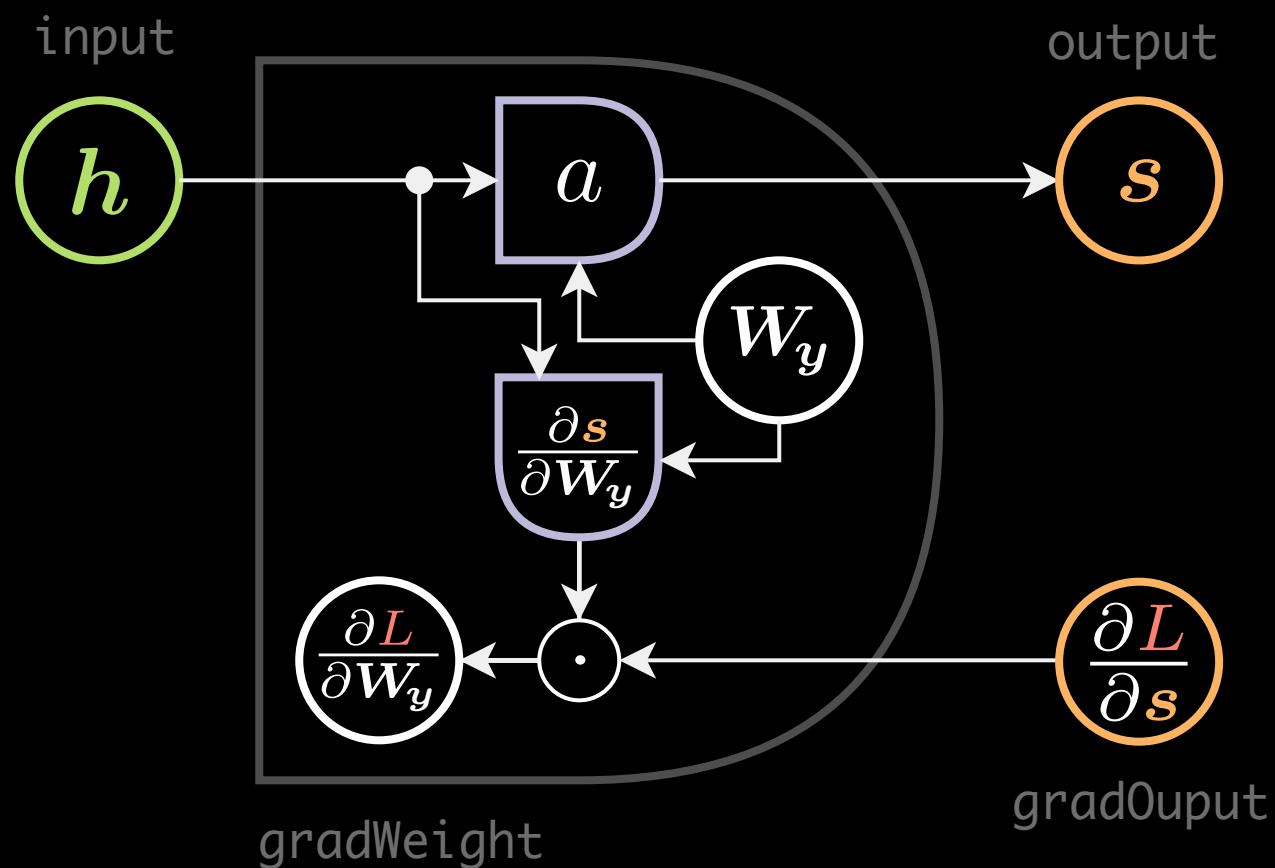


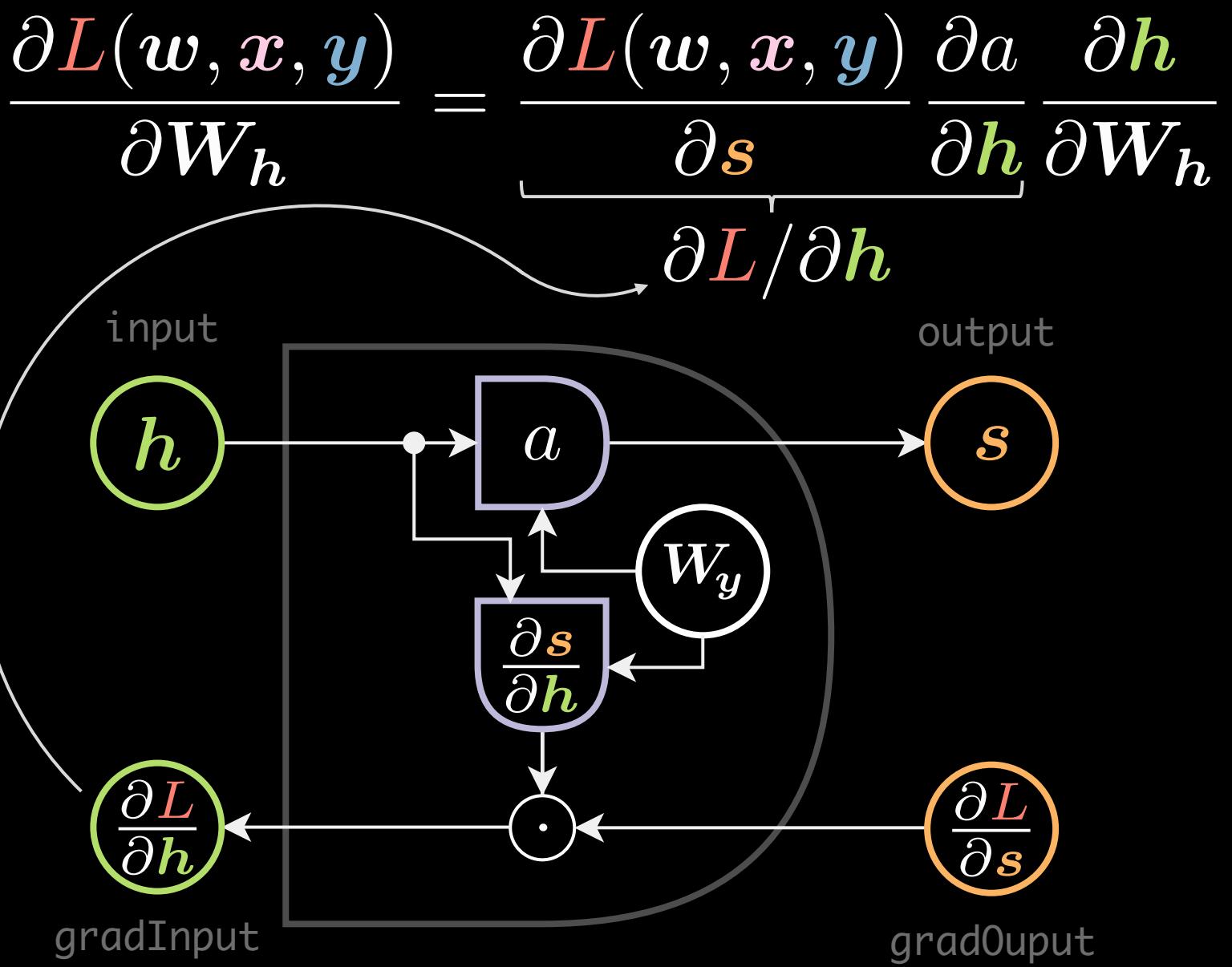
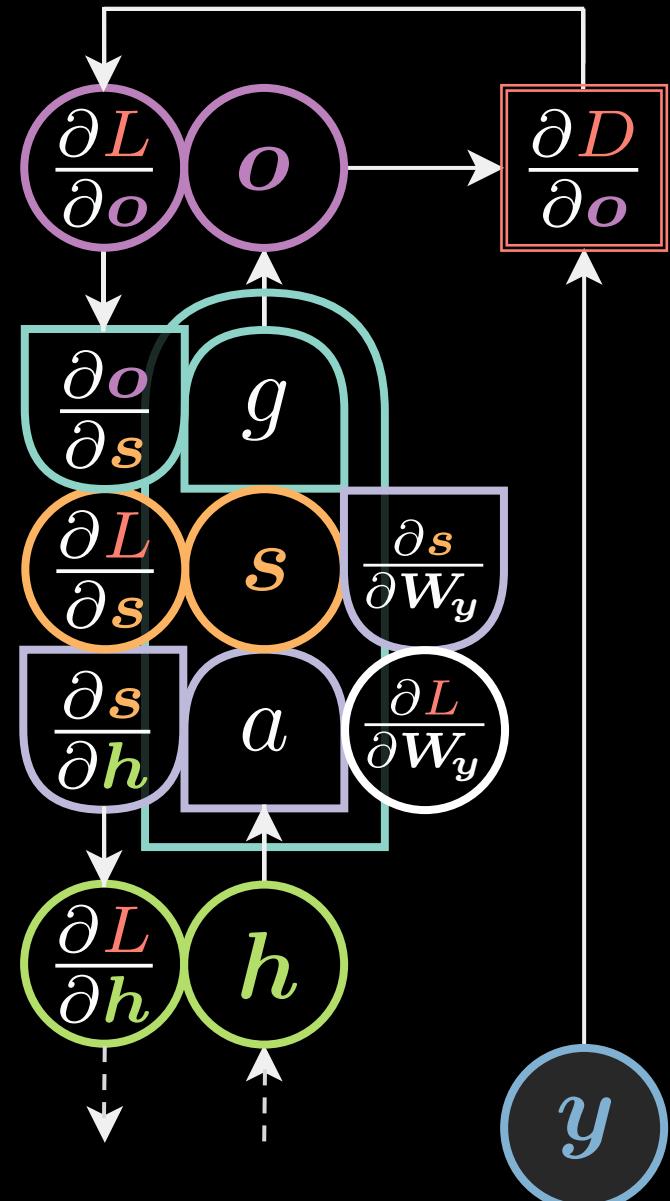
$$\frac{\partial L(\mathbf{w}, \mathbf{x}, \mathbf{y})}{\partial \mathbf{W}_y} = \frac{\partial L(\mathbf{w}, \mathbf{x}, \mathbf{y})}{\partial \mathbf{s}} \frac{\partial \mathbf{a}}{\partial \mathbf{W}_y}$$





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

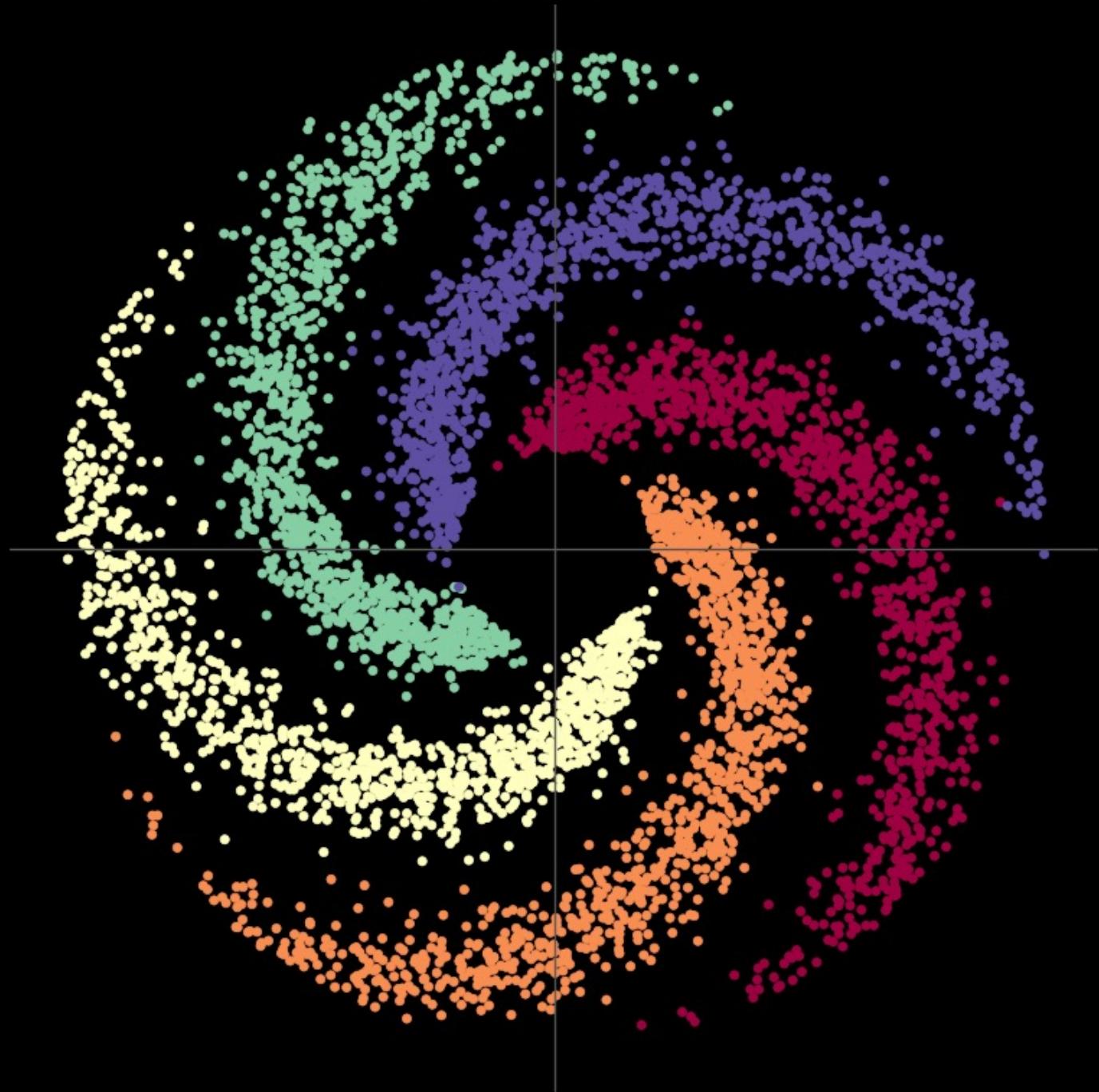




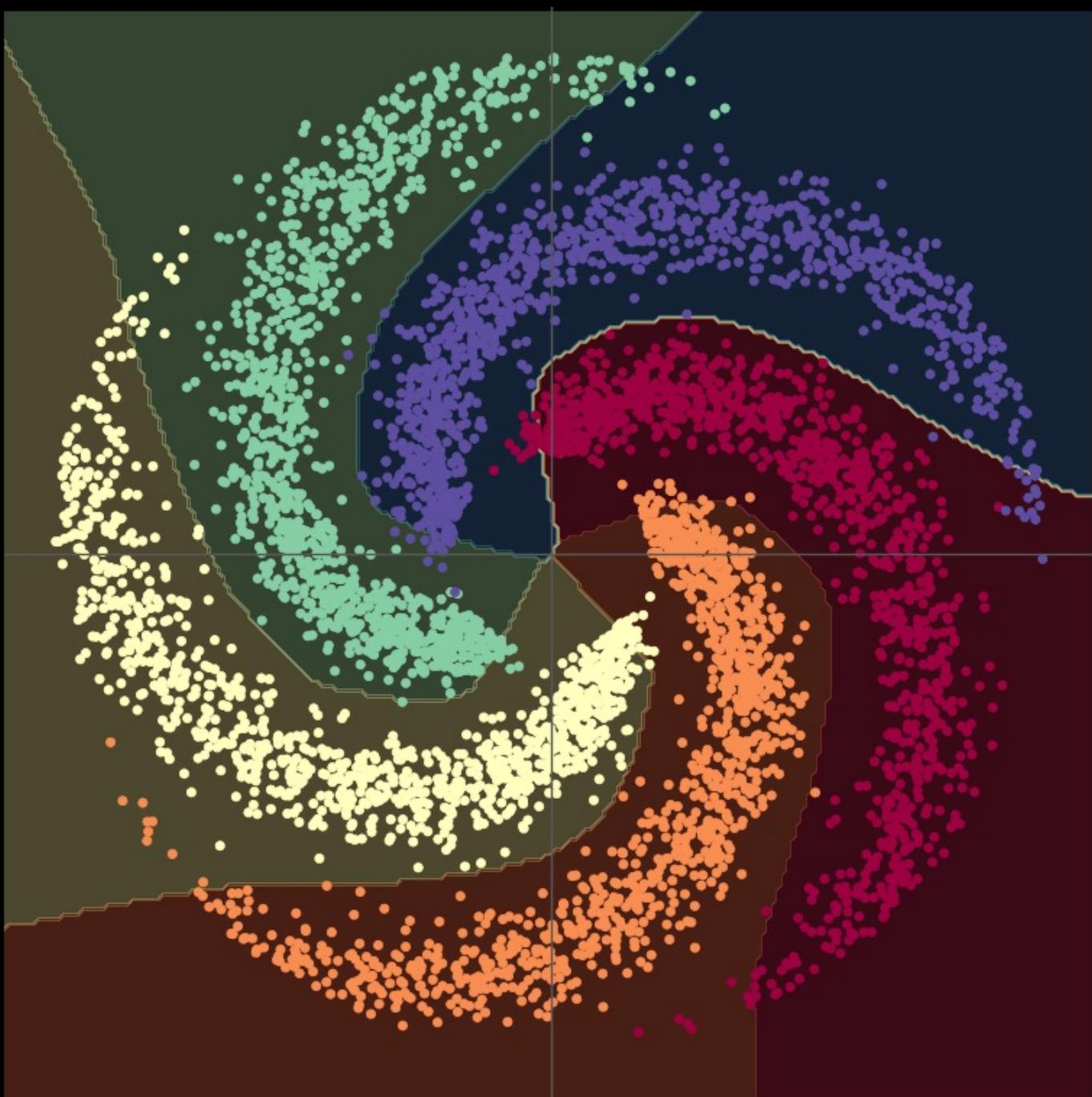
Free energy

Visuals

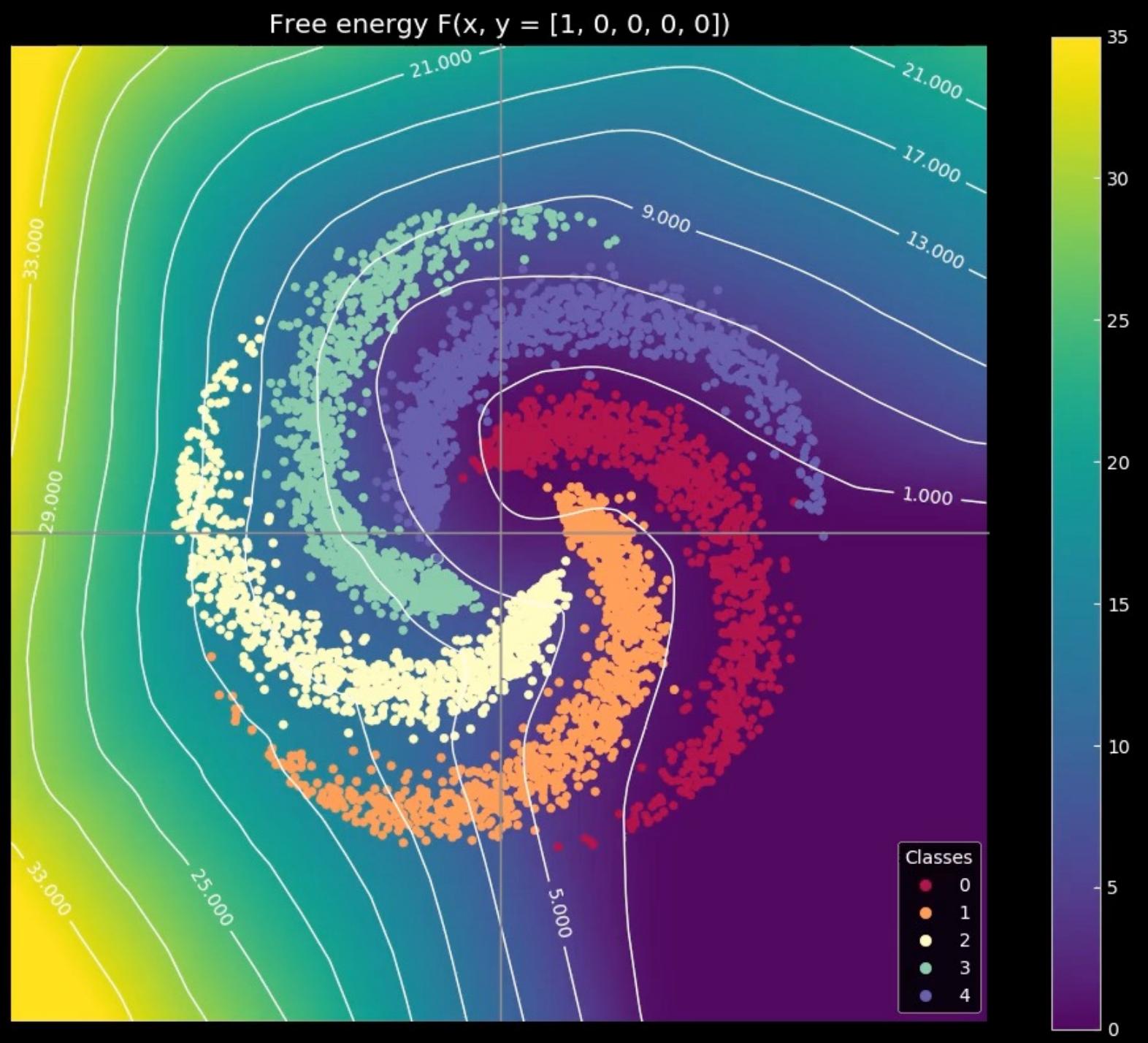
Training data (x, y)



Model decision boundaries



Cross-entropy free energy



Cross-entropy free energy

Free energy $F(x, y = [1, 0, 0, 0, 0])$

