

Foundations of Deep Learning



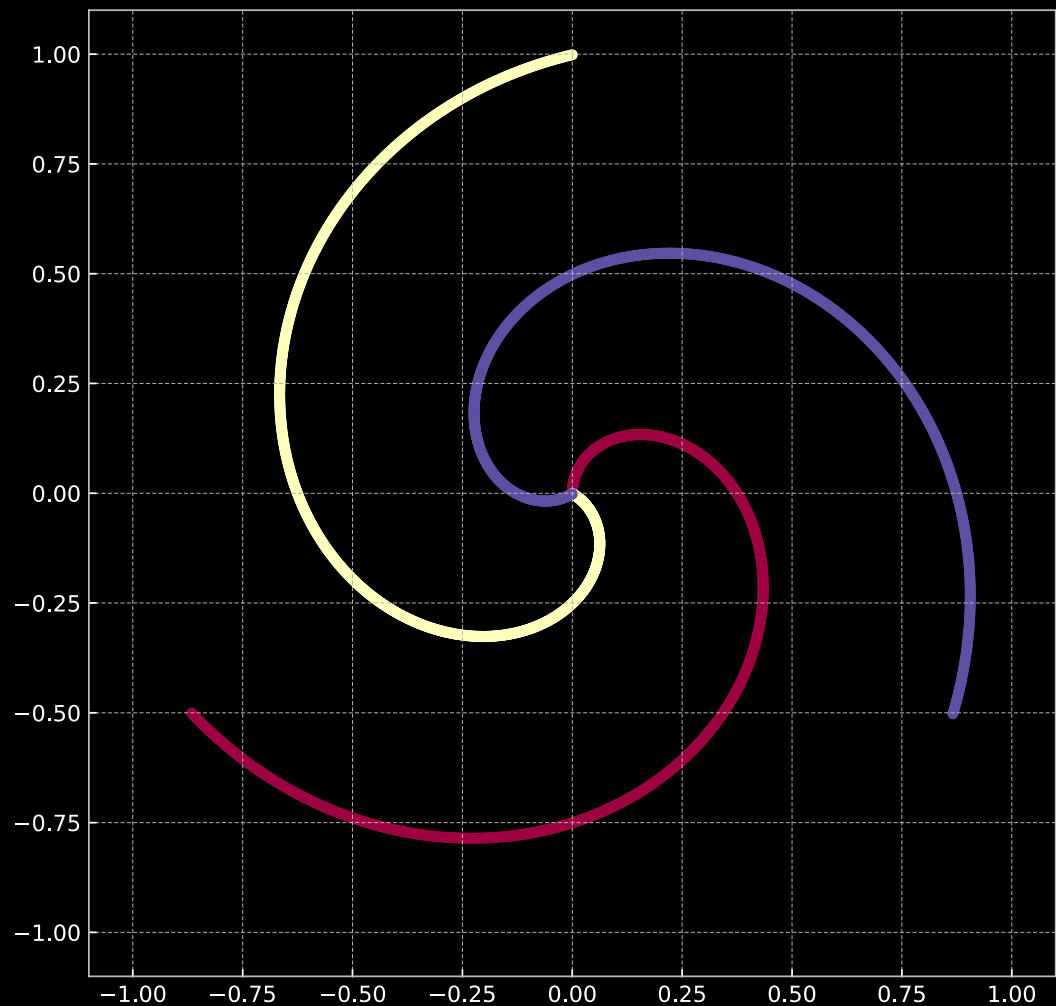
ALF

Alfredo Canziani

 @alfcnz

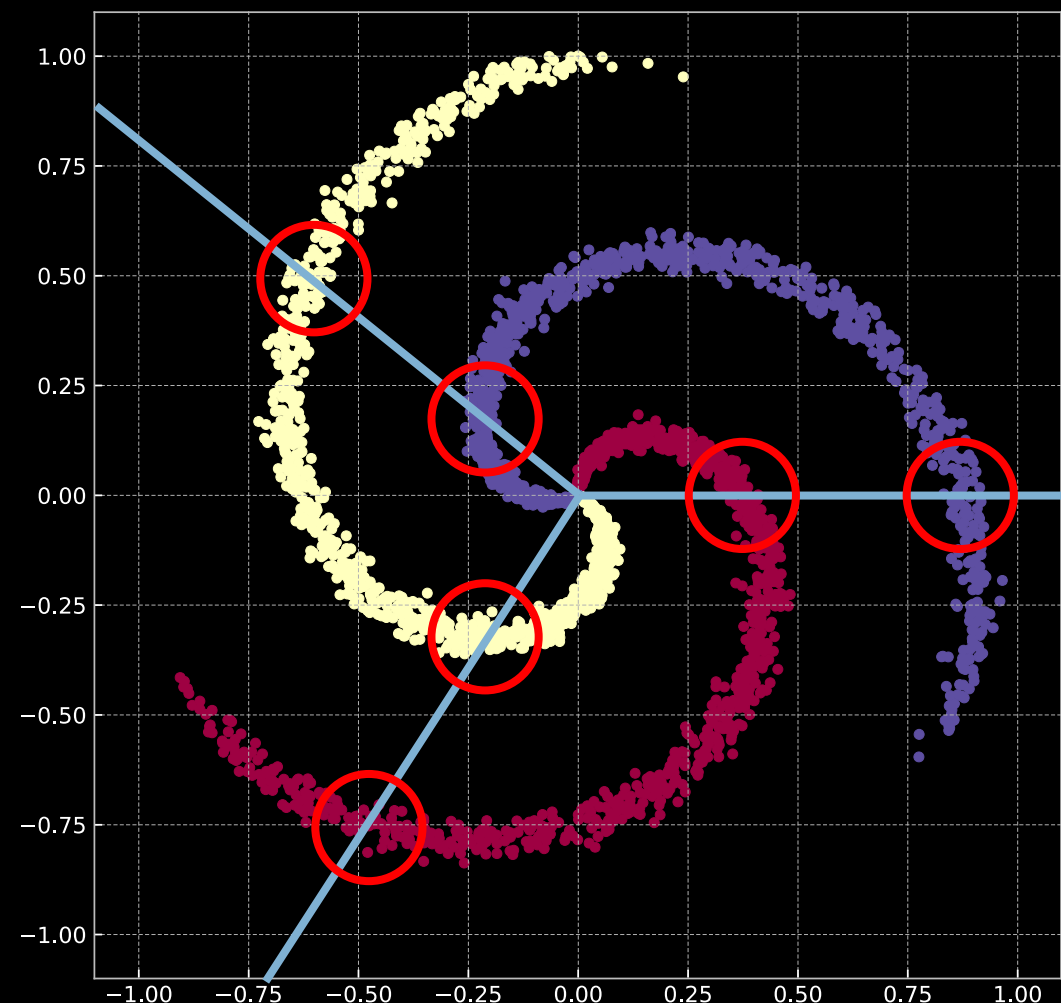
ANN – supervised learning

Classification

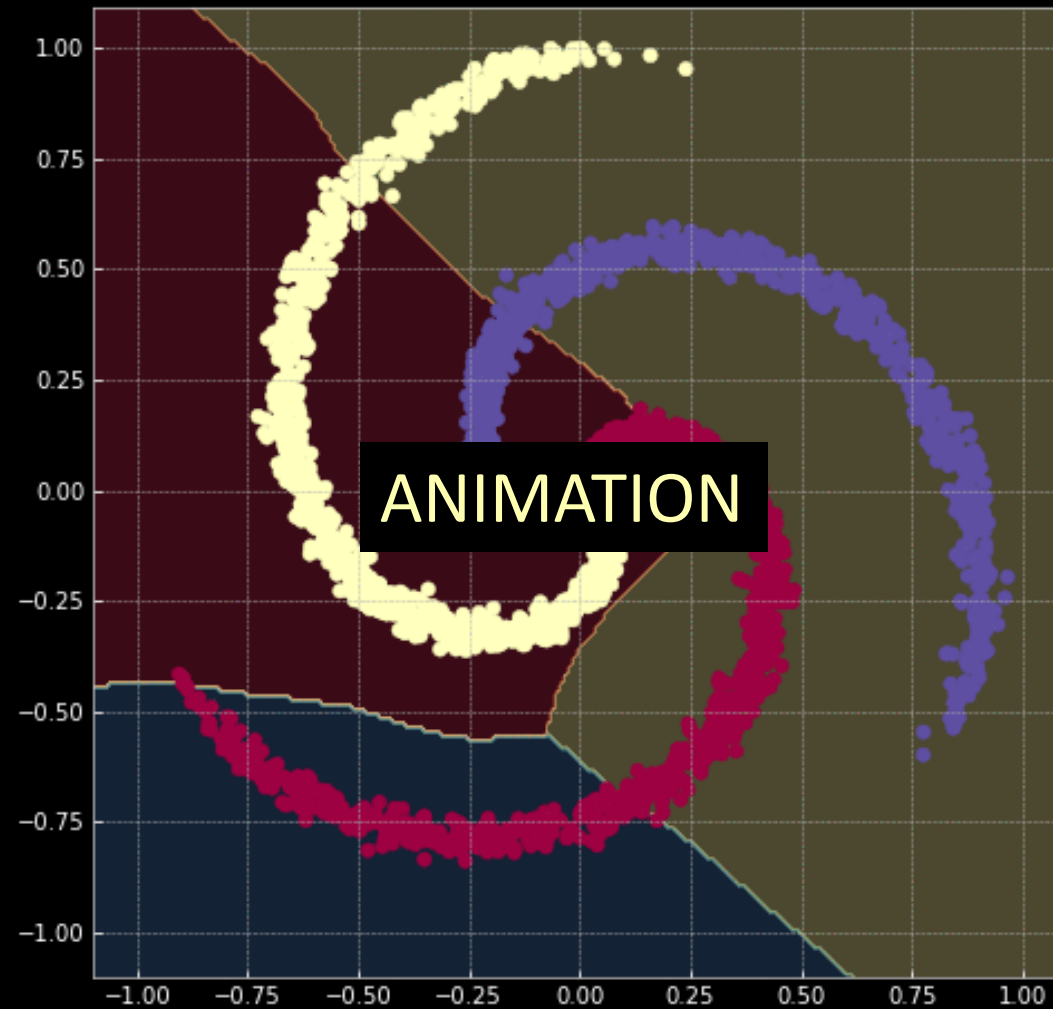
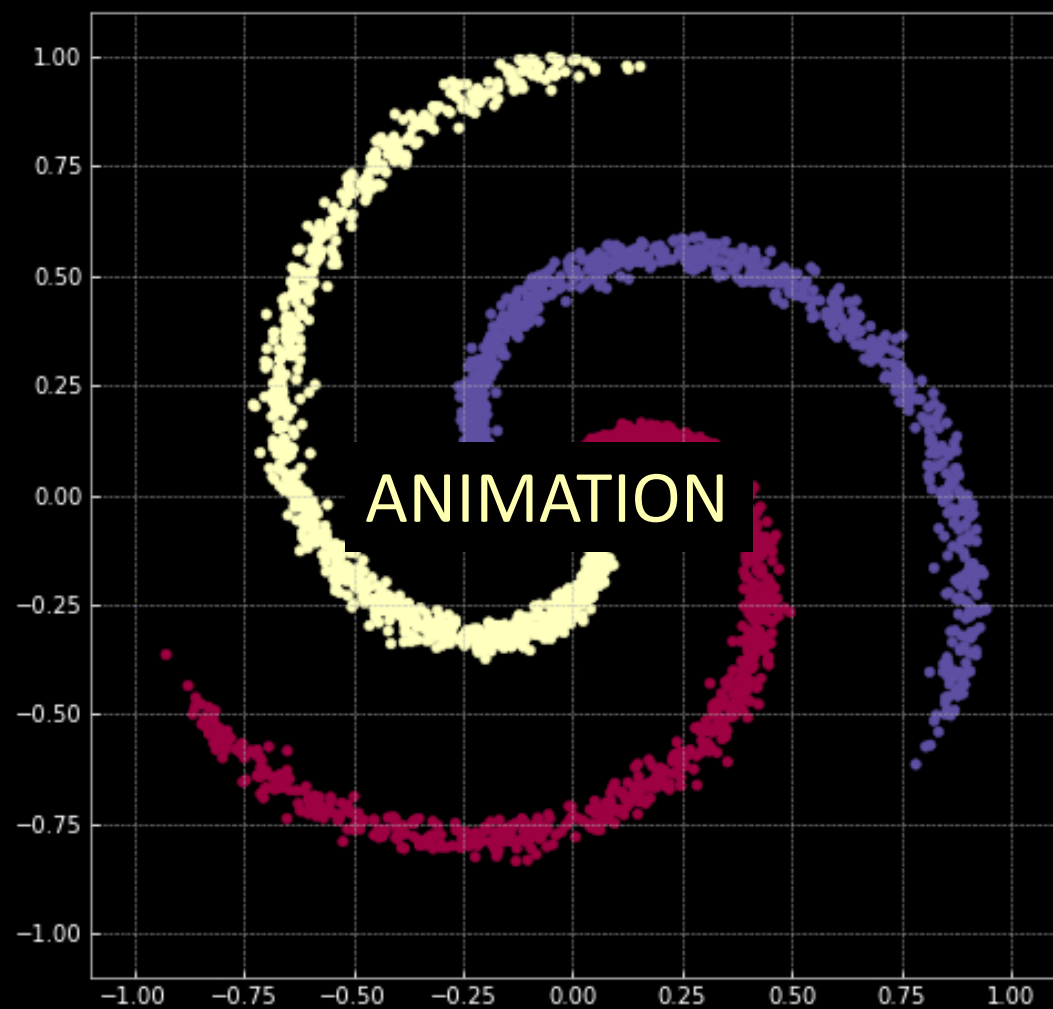


$$X_k(t) = t \begin{pmatrix} \sin \left[\frac{2\pi}{K} (2t + k - 1) \right] \\ \cos \left[\frac{2\pi}{K} (2t + k - 1) \right] \end{pmatrix}$$

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



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



Train data

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

1-hot encoding

$$\mathbf{X} = \begin{bmatrix} \text{---} \mathbf{x}^{(1)} \text{---} \\ \text{---} \mathbf{x}^{(2)} \text{---} \\ \vdots \\ \text{---} \mathbf{x}^{(m)} \text{---} \end{bmatrix}$$

Diagram showing matrix \mathbf{X} with dimensions n (width) and m (height). The matrix contains input vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)}$.

$$\mathbf{Y} = \begin{bmatrix} \text{---} \mathbf{y}^{(1)} \text{---} \\ \text{---} \mathbf{y}^{(2)} \text{---} \\ \vdots \\ \text{---} \mathbf{y}^{(m)} \text{---} \end{bmatrix}$$

Diagram showing matrix \mathbf{Y} with dimensions K (width) and m (height). The matrix contains output vectors $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(m)}$.

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

Diagram showing vector \mathbf{c} with dimension m (height). The vector contains class labels c_1, c_2, \dots, c_m .

$$\mathbf{x}^{(i)} \in \mathbb{R}^n$$

$$\mathbf{y}^{(i)} \in \{0, 1\}^K$$

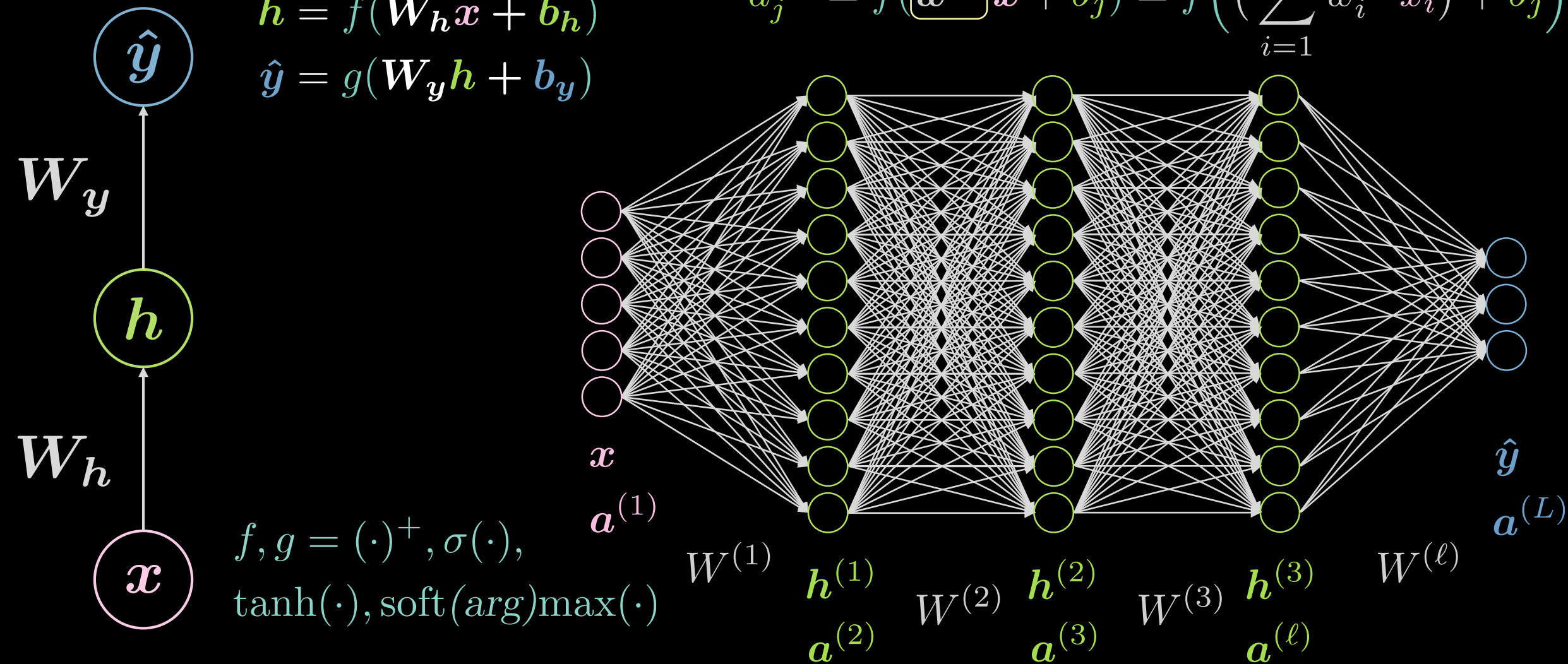
$$c_i \in \{1, 2, \dots, K\}$$

Fully connected (FC) layer

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

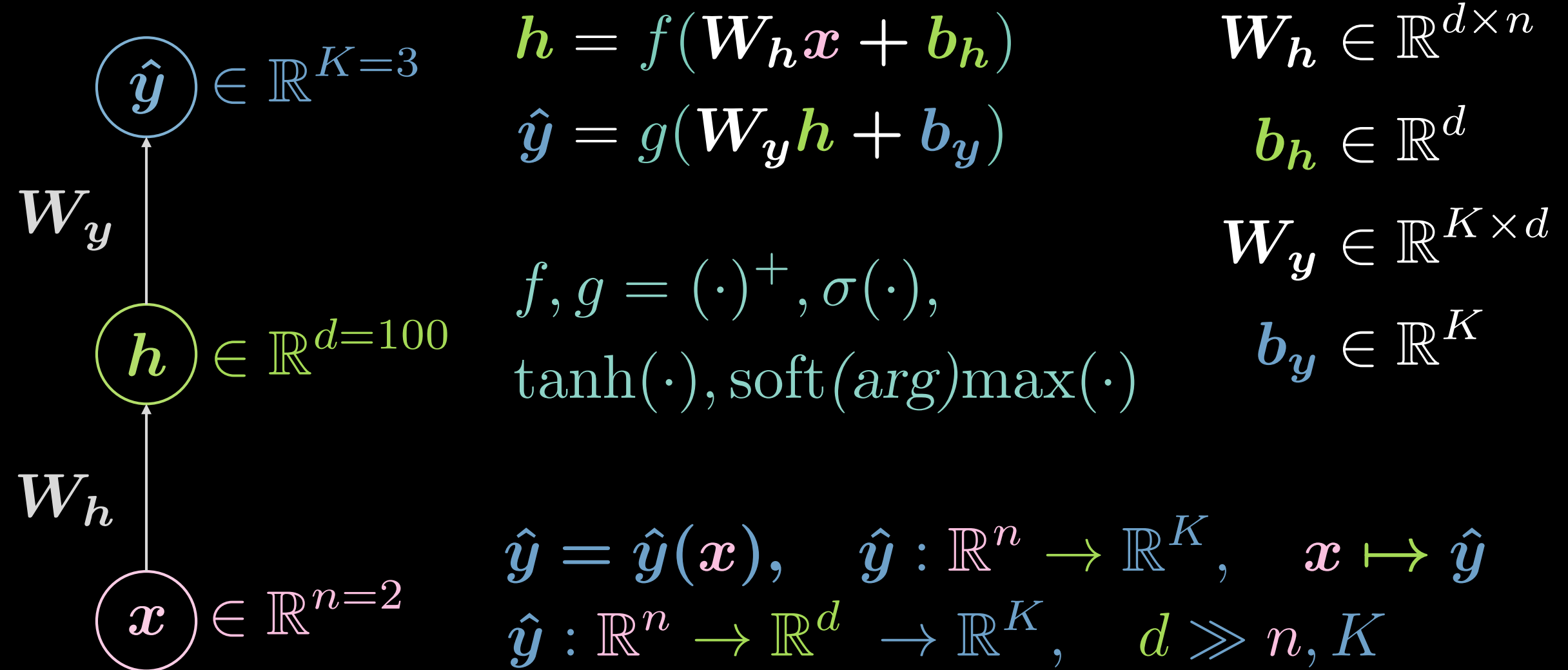
$$\hat{\mathbf{y}} = g(\mathbf{W}_y \mathbf{h} + \mathbf{b}_y)$$

$$a_j^{(2)} = f(\boxed{w^{(j)}} \mathbf{x} + b_j) = f\left(\left(\sum_{i=1}^n w_i^{(j)} x_i\right) + b_j\right)$$



$f, g = (\cdot)^+, \sigma(\cdot), \tanh(\cdot), \text{soft}(\text{arg})\text{max}(\cdot)$

Neural network (inference)



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

$$\hat{y} = g(\mathbf{W}_y h + b_y)$$

Neural network (training I)

logits: output of final layer

$$\text{soft}(\text{arg})\max(\mathbf{l})[c] \doteq \frac{\exp(\mathbf{l}[c])}{\sum_{k=1}^K \exp(\mathbf{l}[k])} \in (0, 1)$$

$$\mathcal{L}(\hat{\mathbf{Y}}, \mathbf{c}) \doteq \frac{1}{m} \sum_{i=1}^m \ell(\hat{\mathbf{y}}^{(i)}, \mathbf{c}_i), \quad \ell(\hat{\mathbf{y}}, \mathbf{c}) \doteq -\log(\hat{\mathbf{y}}[c])$$

cross entropy / negative log-likelihood

$$\mathbf{x}, \quad \mathbf{c} = 1 \quad \Rightarrow \quad \mathbf{y} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\hat{\mathbf{y}}(\mathbf{x}) = \begin{pmatrix} \sim 1 \\ \sim 0 \\ \sim 0 \end{pmatrix} \Rightarrow \ell \left(\begin{pmatrix} \sim 1 \\ \sim 0 \\ \sim 0 \end{pmatrix}, \mathbf{1} \right) \rightarrow 0^+$$

$$\hat{\mathbf{y}}(\mathbf{x}) = \begin{pmatrix} \sim 0 \\ \sim 1 \\ \sim 0 \end{pmatrix} \Rightarrow \ell \left(\begin{pmatrix} \sim 0 \\ \sim 1 \\ \sim 0 \end{pmatrix}, \mathbf{1} \right) \rightarrow +\infty$$

Neural network (training II)

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

$$J(\Theta) \doteq \mathcal{L}(\hat{Y}(\Theta), \mathbf{c}) \in \mathbb{R}^+$$

$$\frac{\partial J(\Theta)}{\partial \mathbf{W}_y} = \frac{\partial J(\Theta)}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{W}_y}$$

$$\frac{\partial J(\Theta)}{\partial \mathbf{W}_h} = \frac{\partial J(\Theta)}{\partial \hat{\mathbf{y}}} \frac{\partial \hat{\mathbf{y}}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{W}_h}$$

back-propagation

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

$$\hat{\mathbf{y}} = g(\mathbf{W}_y \mathbf{h} + \mathbf{b}_y)$$

