Convolutional Neural Nets

Exploiting stationarity, locality, and compositionality of natural data

Input layer / samples

$$\mathcal{X} = \{\boldsymbol{x}^{(p)} \in \mathbb{R}^n \mid \boldsymbol{x}^{(p)} \text{ is a data sample}\}_{p=1}^{P}$$ input samples

$$\mathcal{X} = \{\boldsymbol{x}^{(p)} : \Omega \to \mathbb{R}^c, \boldsymbol{\omega} \mapsto \boldsymbol{x}^{(p)}(\boldsymbol{\omega})\}_{p=1}^{P}$$

domain, channels

$$\Omega = \{1, 2, \cdots, T/\Delta t\} \subset \mathbb{N}, \quad c \in \{1, 2, 5+1, \cdots\}$$

total time / sampling interval; stereo; mono; Dolby 5.1

$$\Omega = \{1, \cdots, h\} \times \{1, \cdots, w\} \subset \mathbb{N}^2, \quad c \in \{1, 3, 20, \cdots\}$$

height, width; grey scale; colour; hyperspectral

$$\Omega = \mathbb{R}^4 \times \mathbb{R}^4, \quad c = 1$$

four-momentum; space-time; Hamiltonian

$$\boldsymbol{x}(\omega_1, \omega_2) = \begin{pmatrix} r(\omega_1, \omega_2) \\ g(\omega_1, \omega_2) \\ b(\omega_1, \omega_2) \end{pmatrix}$$

Signals can be represented as vectors

$$\boldsymbol{x} = [x_1 \;\; x_2 \;\; x_3 \;\; \ldots \;\; x_t \;\; \ldots]^\top$$

x_t are waveform heights

$$\boldsymbol{x} = [x_{11} \;\; x_{12} \;\; \ldots \;\; x_{1n} \;\; x_{21} \;\; x_{22} \;\; \ldots]^\top$$

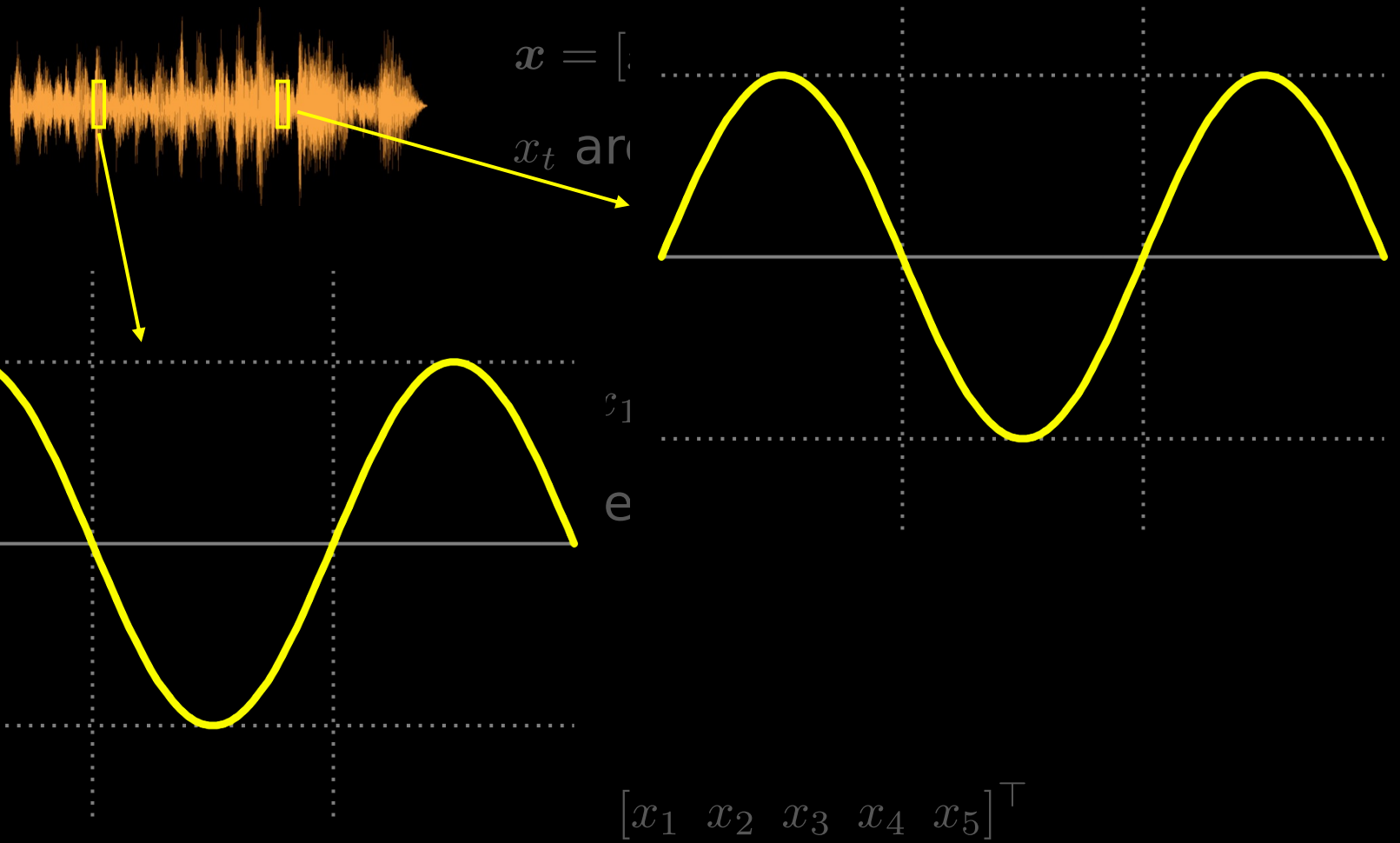
x_{ij} are pixel values

"John picked up the apple"

$$\boldsymbol{x} = [x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5]^\top$$

x_t are one-hot vectors

Signals can be represented as vectors



$\boldsymbol{x} = [$

x_t ar

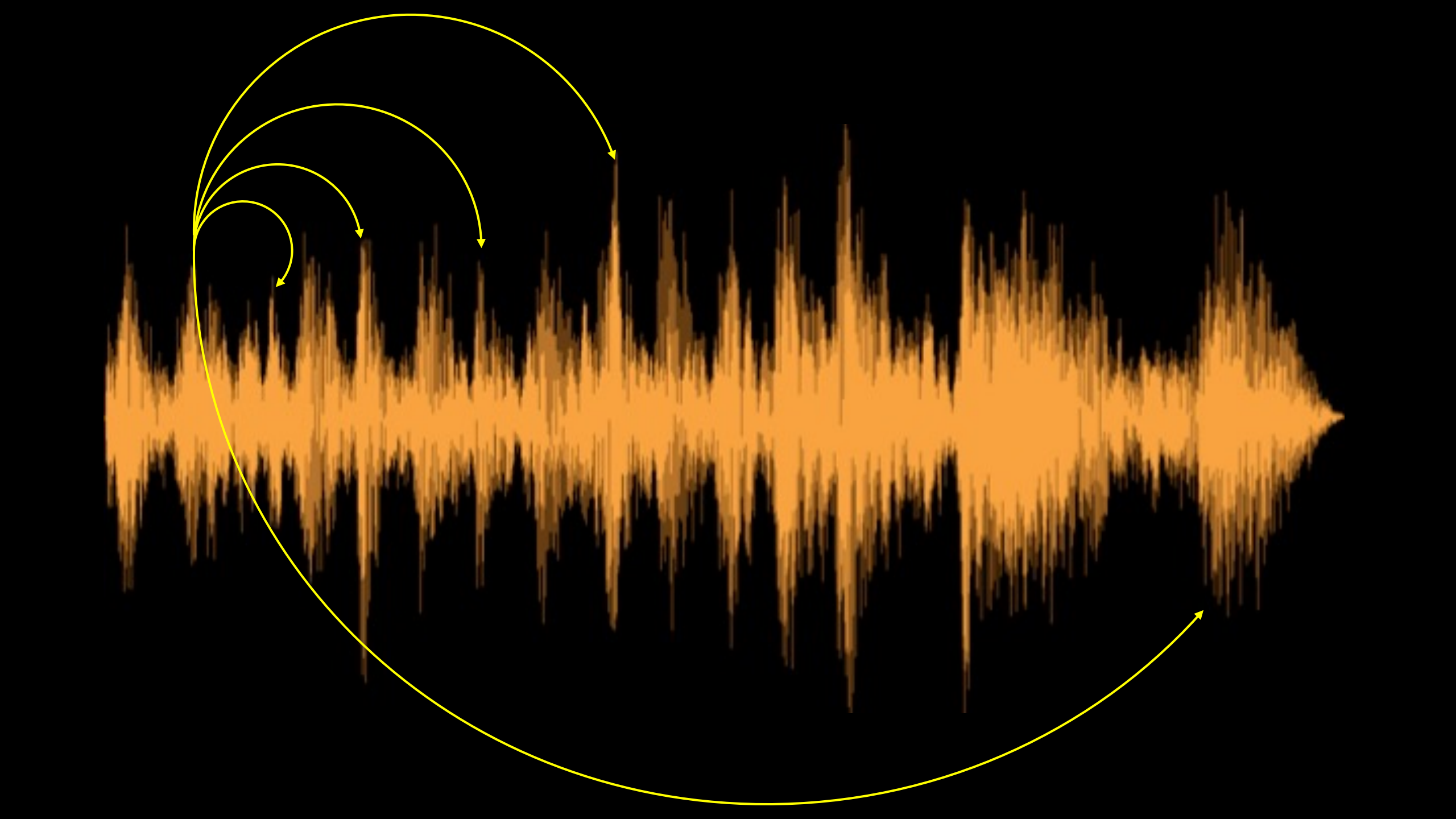
x_1

e

$[x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5]^\top$

"John picked up the apple"

x_t are one-hot vectors



Signals can be represented as vectors

$$\boldsymbol{x} = [x_1 \;\; x_2 \;\; x_3 \;\; \ldots \;\; x_t \;\; \ldots]^\top$$

x_t are waveform heights

$$\boldsymbol{x} = [x_{11} \;\; x_{12} \;\; \ldots \;\; x_{1n} \;\; x_{21} \;\; x_{22} \;\; \ldots]^\top$$

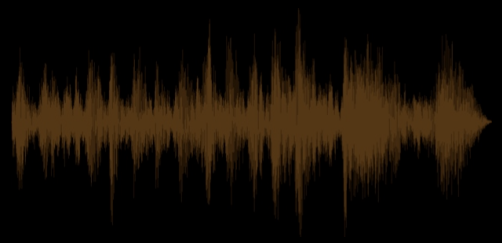
x_{ij} are pixel values

"John picked up the apple"

$$\boldsymbol{x} = [x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5]^\top$$

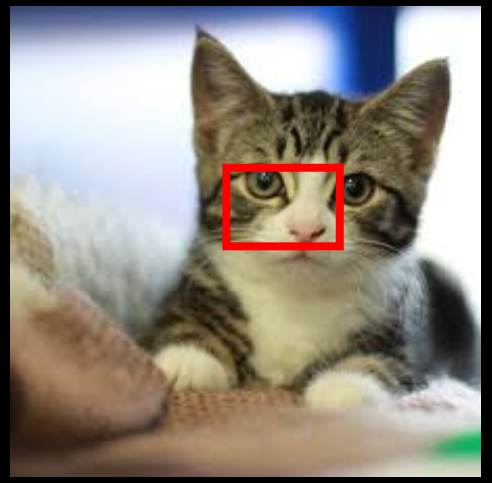
x_t are one-hot vectors

Signals can be represented as vectors



$$\boldsymbol{x} = [x_1 \;\; x_2 \;\; x_3 \;\; \ldots \;\; x_t \;\; \ldots]^\top$$

x_t are waveform heights



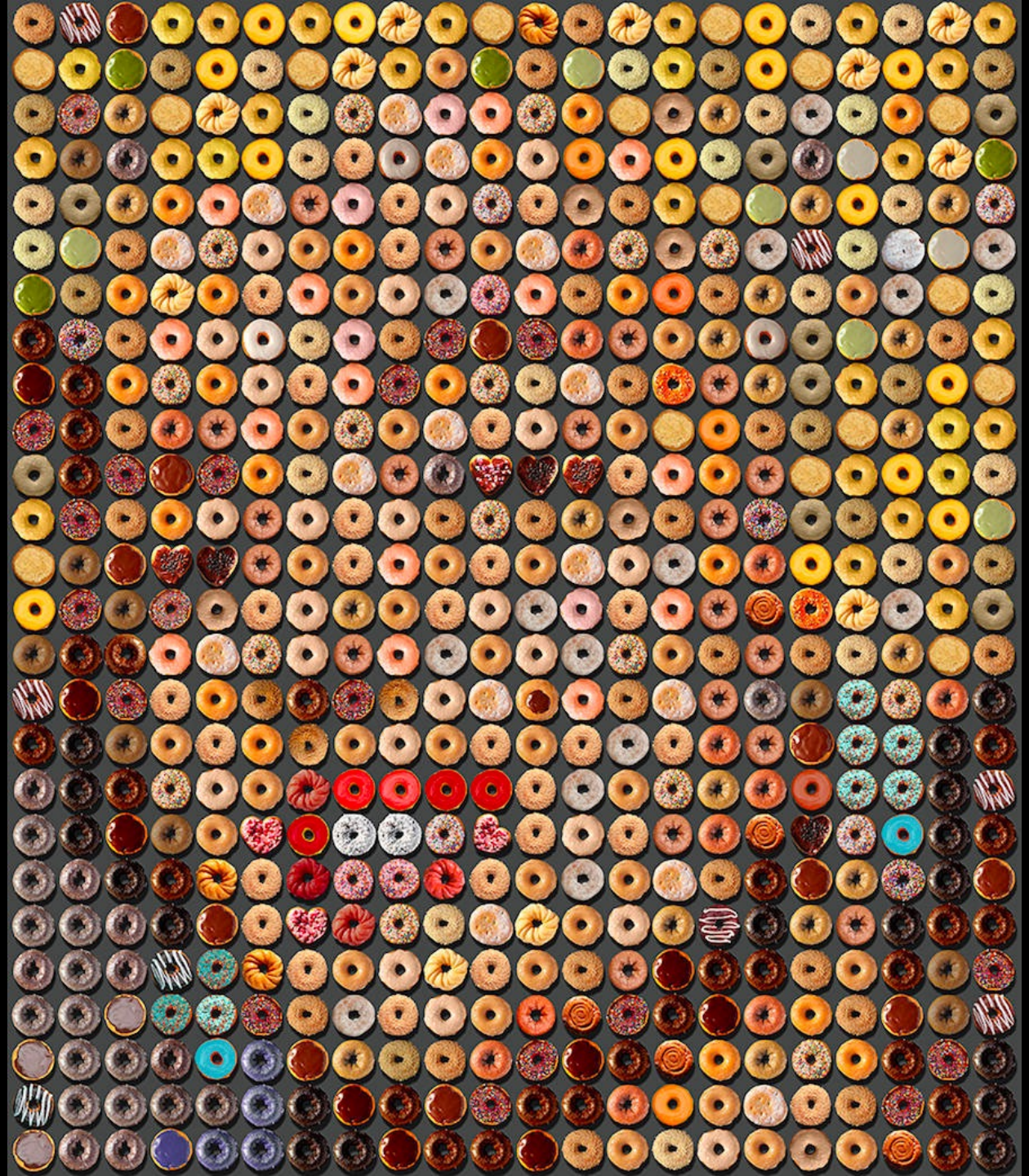
$$\boldsymbol{x} = [x_{11} \;\; x_{12} \;\; \ldots \;\; x_{1n} \;\; x_{21} \;\; x_{22} \;\; \ldots]^\top$$

x_{ij} are pixel values

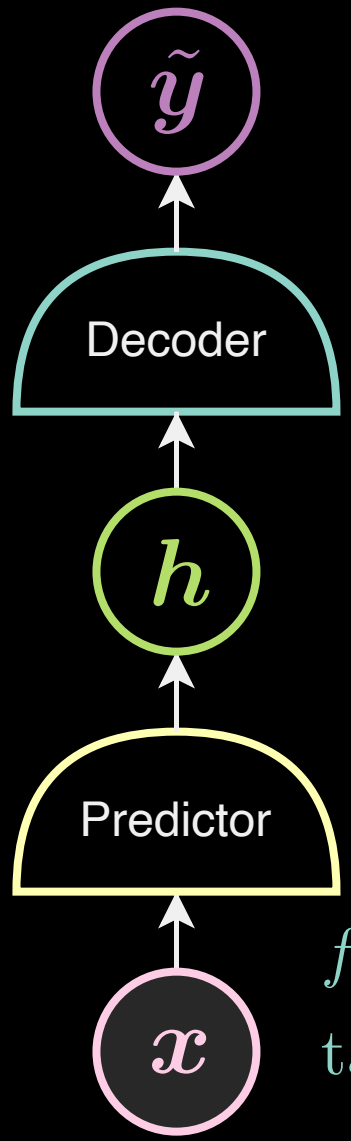
"John picked up the apple"

$$\boldsymbol{x} = [x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5]^\top$$

x_t are one-hot vectors

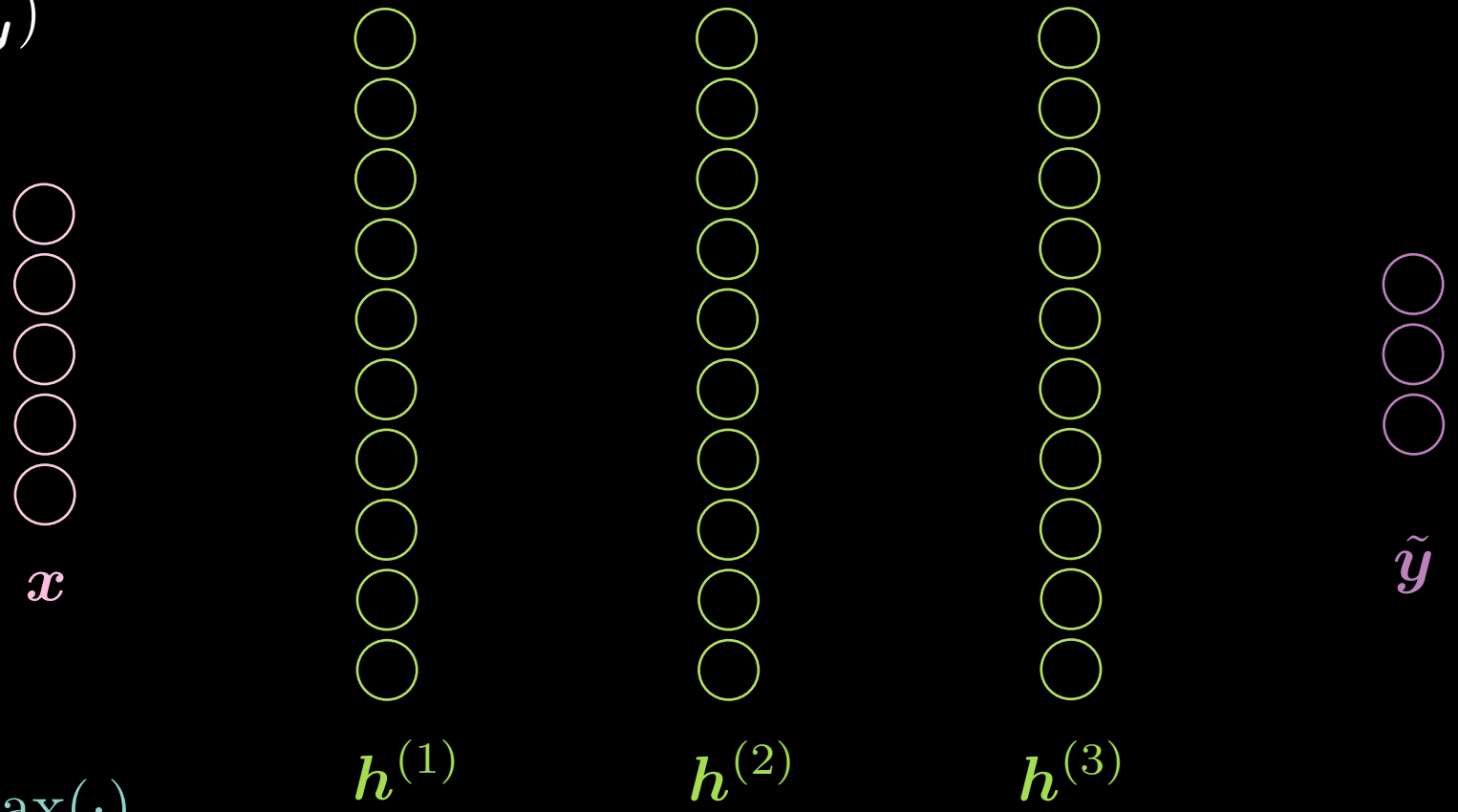


Fully connected (FC) layer



$$h = f(W_h x + b_h)$$
$$\tilde{y} = g(W_y h + b_y)$$

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



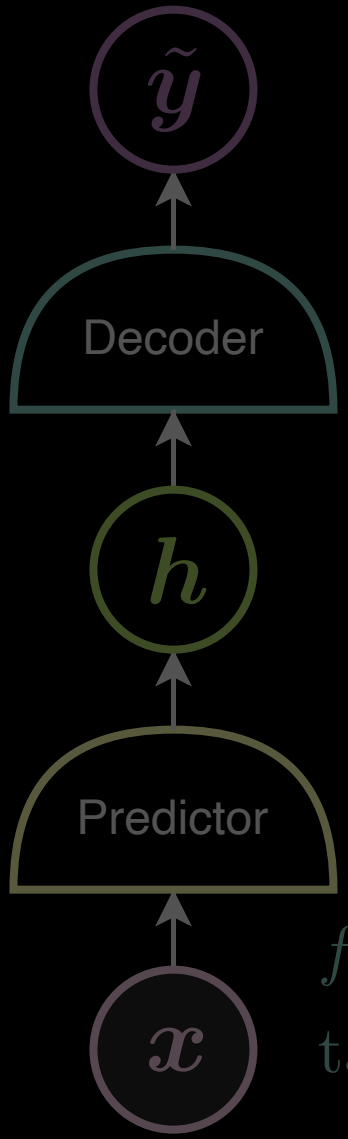
Fully connected (FC) layer

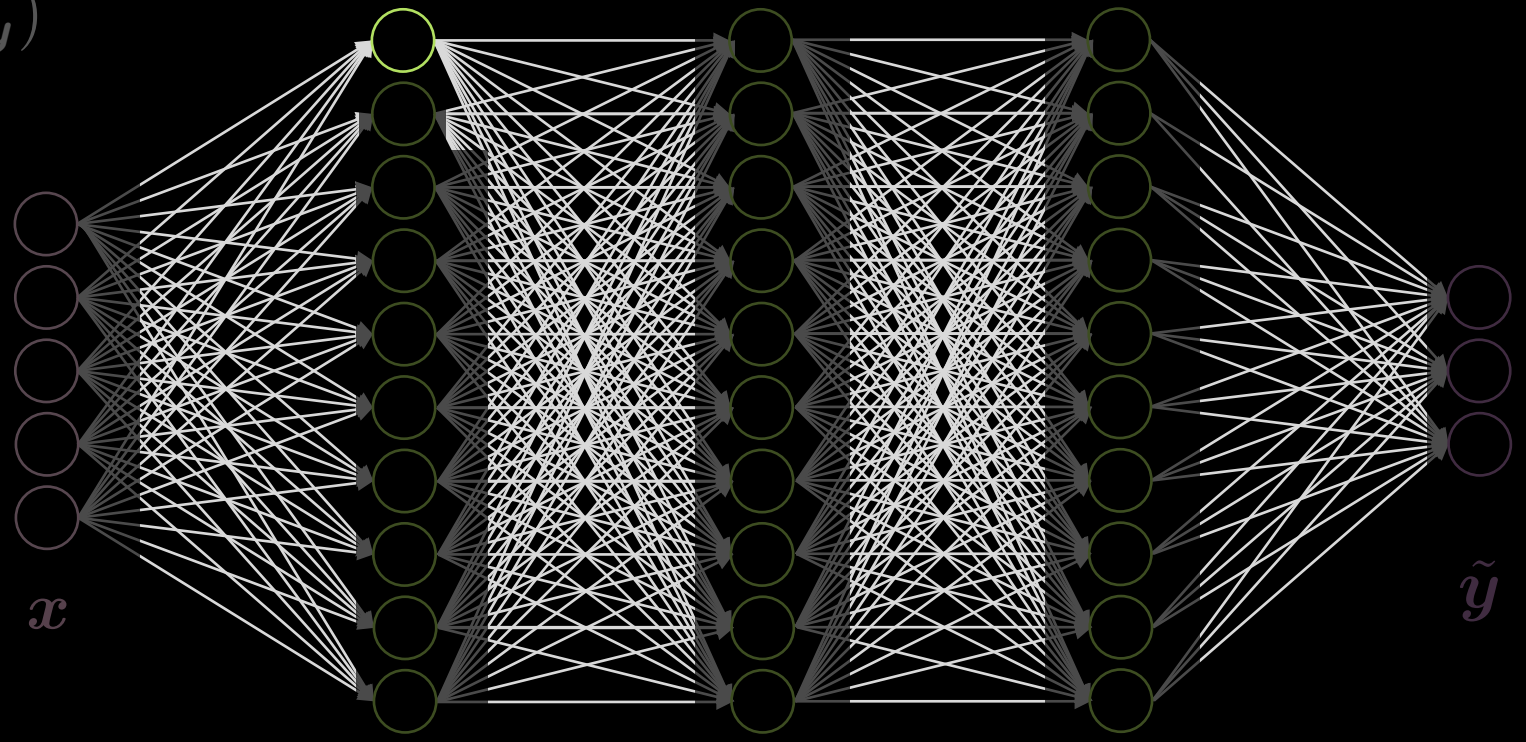
j-th row of $W^{(1)}$

$$h_j^{(1)} = f(\boldsymbol{w}^{(j)}\boldsymbol{x} + b_j) = f\left[\left(\sum_{i=1}^{n} w_i^{(j)} x_i\right) + b_j\right]$$

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

$$\tilde{\boldsymbol{y}} = g(\boldsymbol{W_y}\boldsymbol{h} + \boldsymbol{b_y})$$

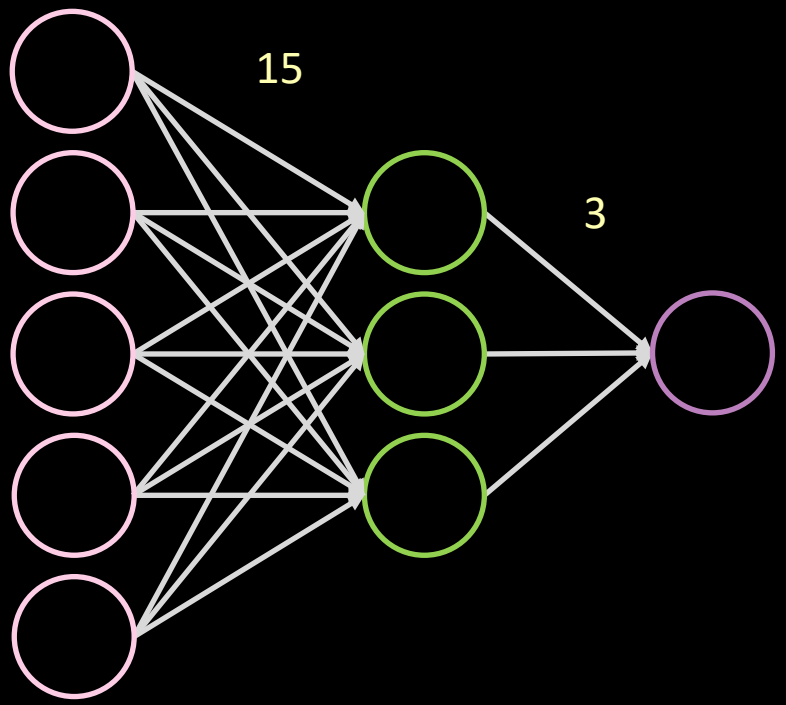


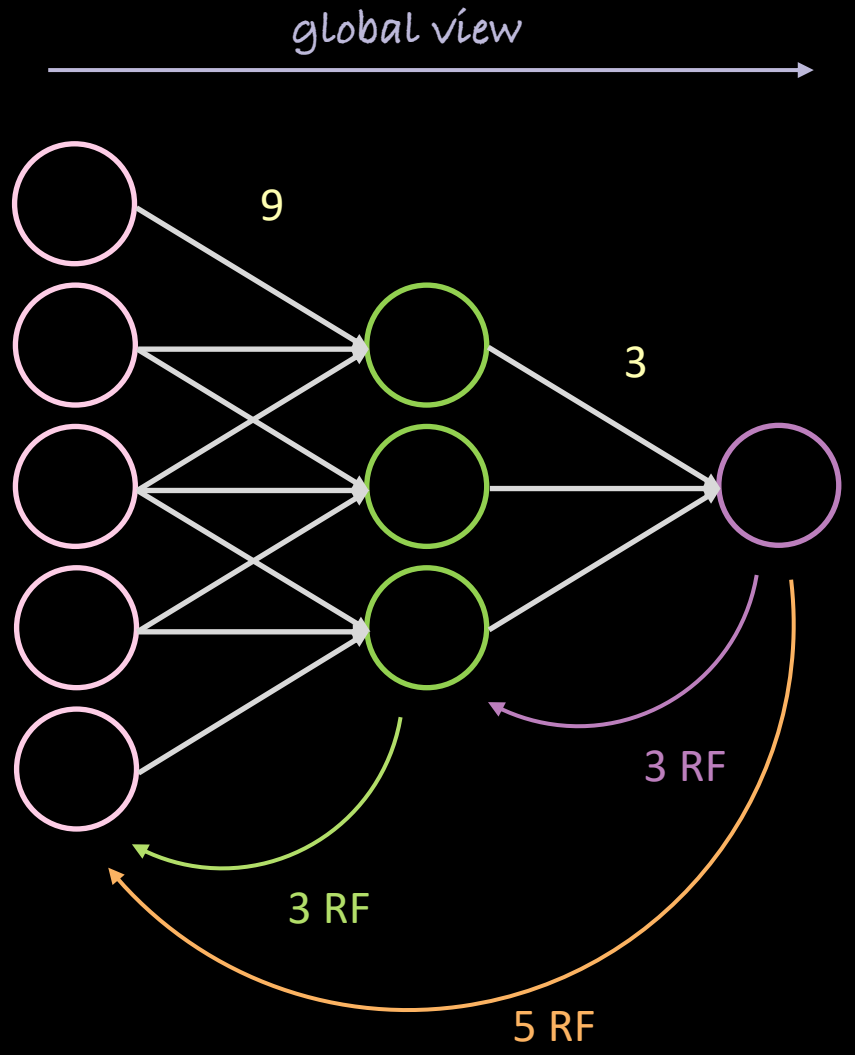


$f, g = (\cdot)^+, \sigma(\cdot),$
$\tanh(\cdot), \text{soft}(arg)\max(\cdot)$

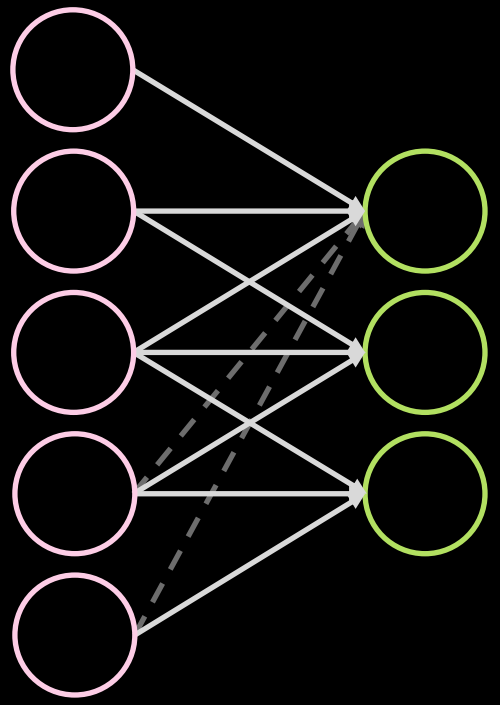
RF: receptive field

Locality ⇒ sparsity





Stationarity ⇒ parameters sharing

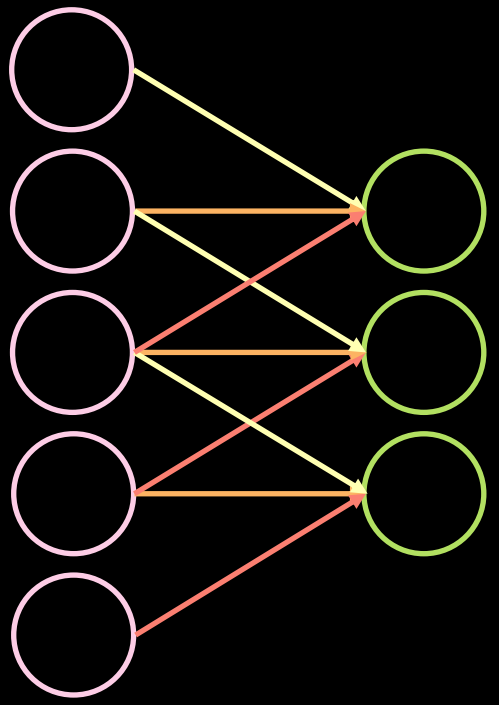


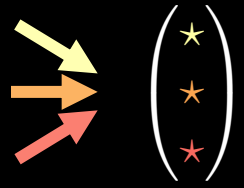
Parameters sharing

- faster convergence
- better generalisation
- not constrained to input size
- kernel independence
  ⇒ high parallelisation

Connection sparsity

- reduced amount of computation

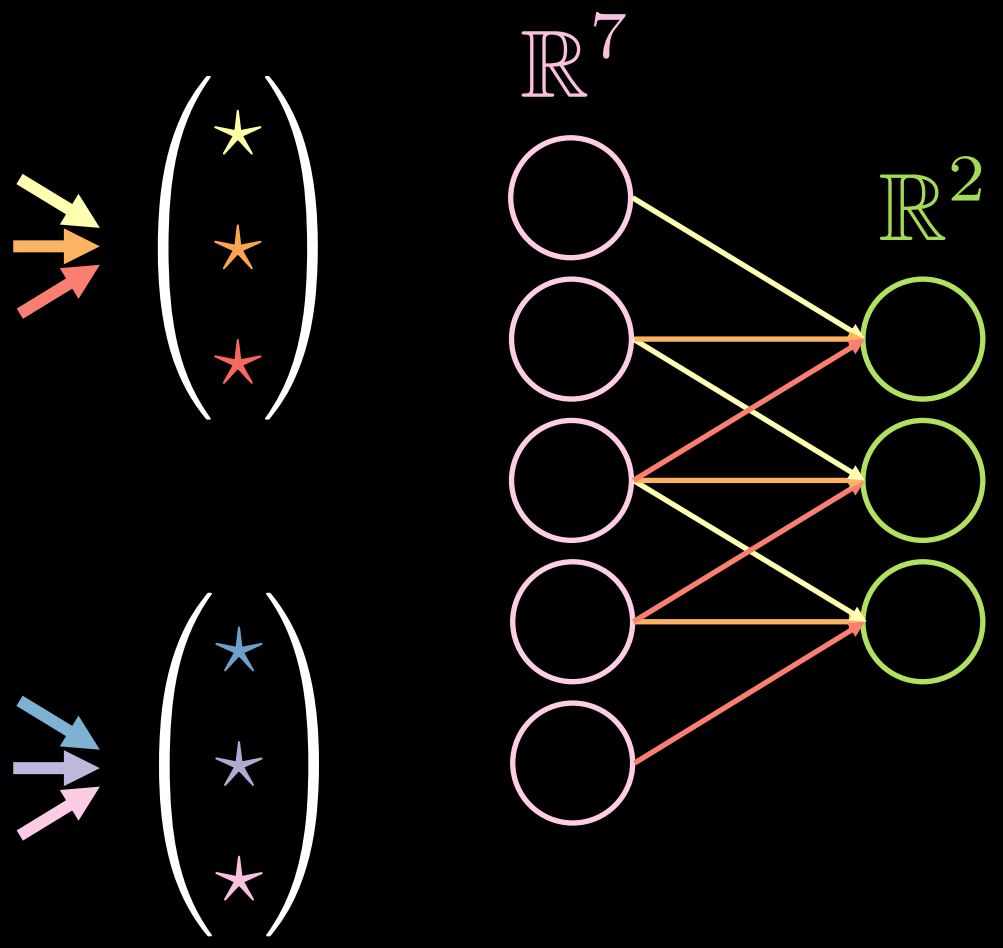




Kernels – 1D data

kernel size: $2 \times 7 \times 3$

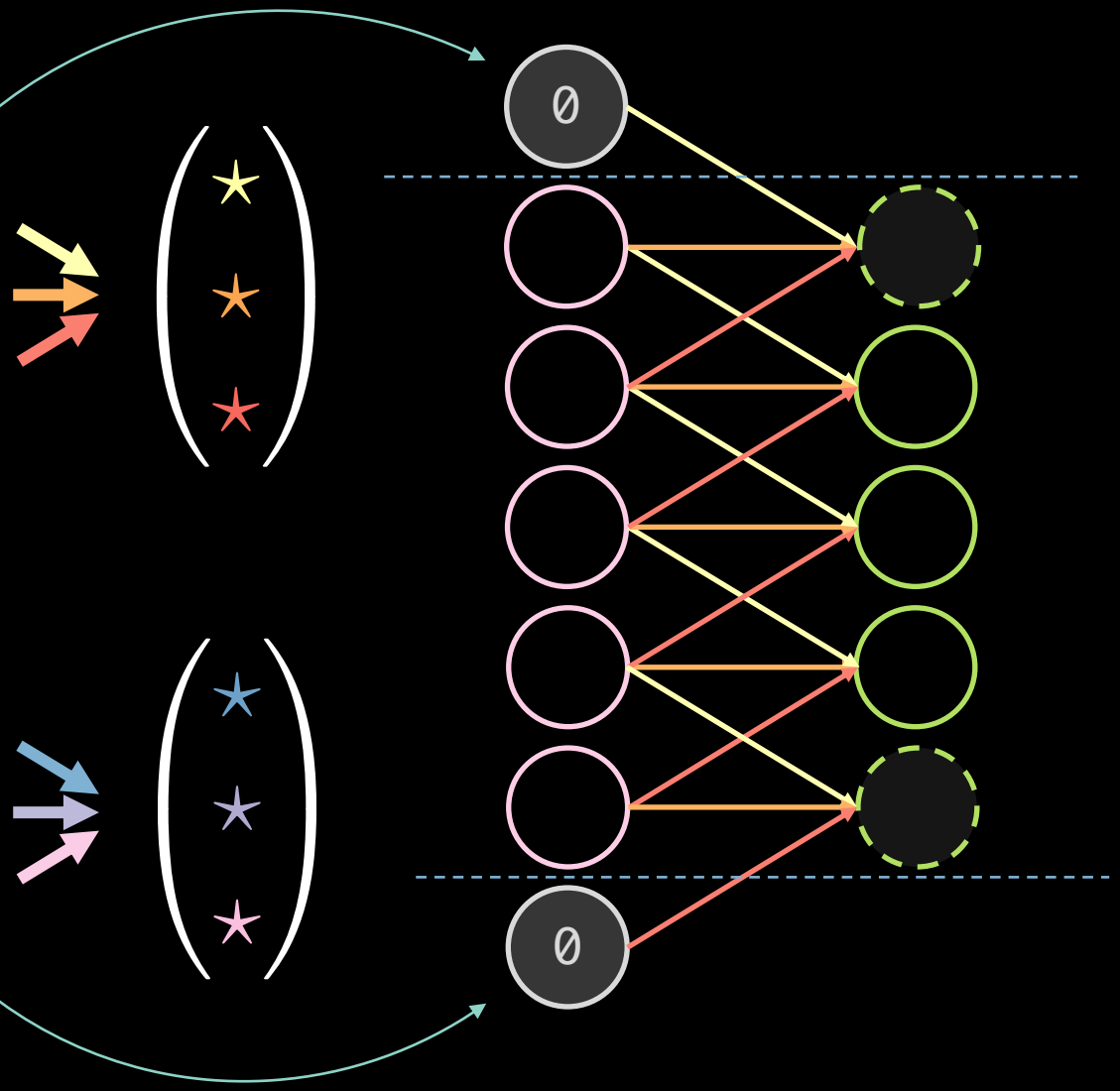
1D data uses 3D kernels-collection!



Padding – 1D data

kernel size: $2 \times 7 \times \boxed{3}$

zero padding: $(\boxed{3} - 1)/2 = \boxed{1}$



Standard spatial CNN

- Multiple layers
  - Convolution
  - Non-linearity (ReLU and Leaky)
  - Pooling
  - Batch normalisation
- Residual bypass connection