Foundations of DL

Deep Learning

ALF

Alfredo Canziani, Ritchie Ng
@alfcnz, @RitchieNg

Convolutional Neural Nets

Exploiting stationarity, locality, and compositionality of natural data

Signals can be represented as vectors

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

x_t are waveform heights

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

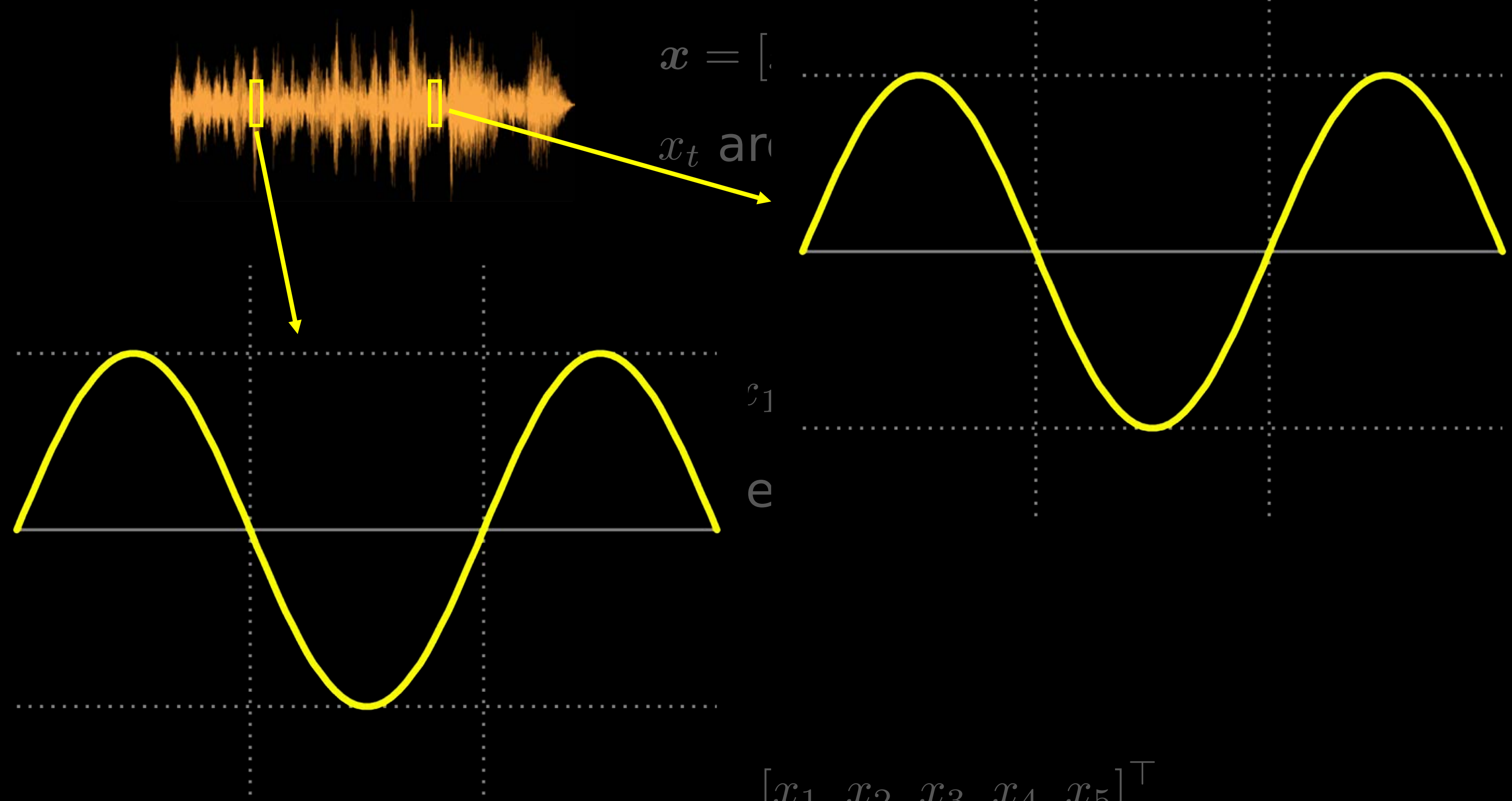
x_{ij} are pixel values

"John picked up the apple"

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

x_t are one-hot vectors

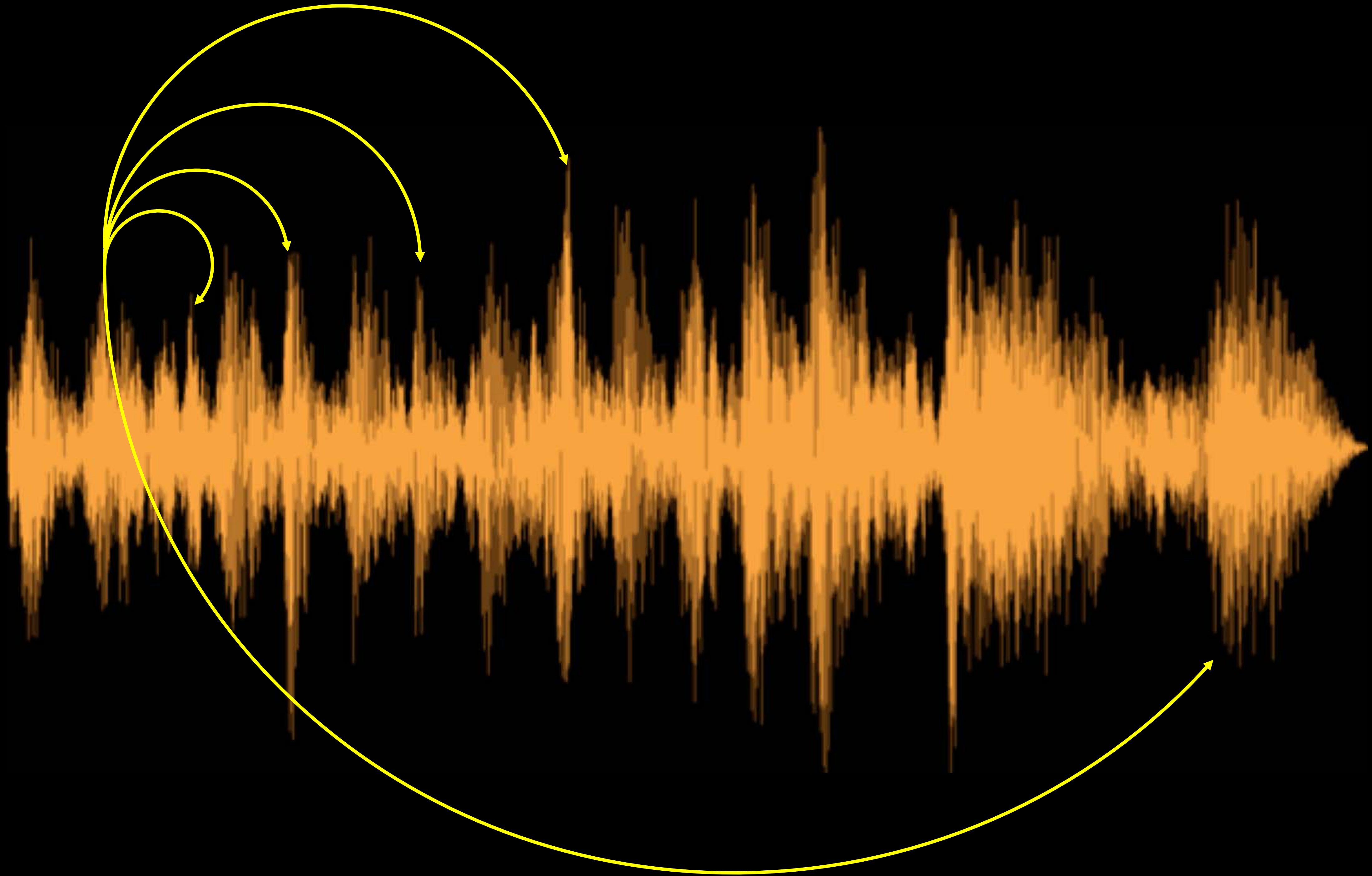
Signals can be represented as vectors



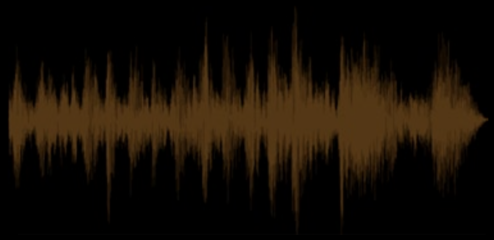
"John picked up the apple"

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

x_t are one-hot vectors

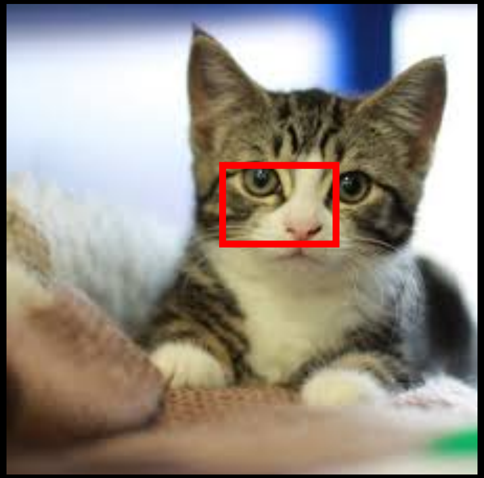


Signals can be represented as vectors



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

x_t are waveform heights



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

x_{ij} are pixel values

“John picked up the apple”

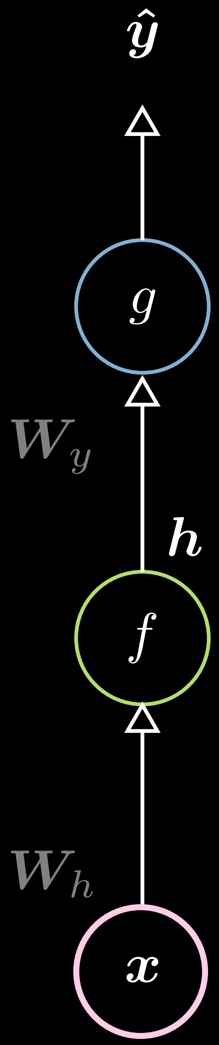
$$\boldsymbol{x} = [x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5]^\top$$

x_t are one-hot vectors

Fully connected (FC) layer

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

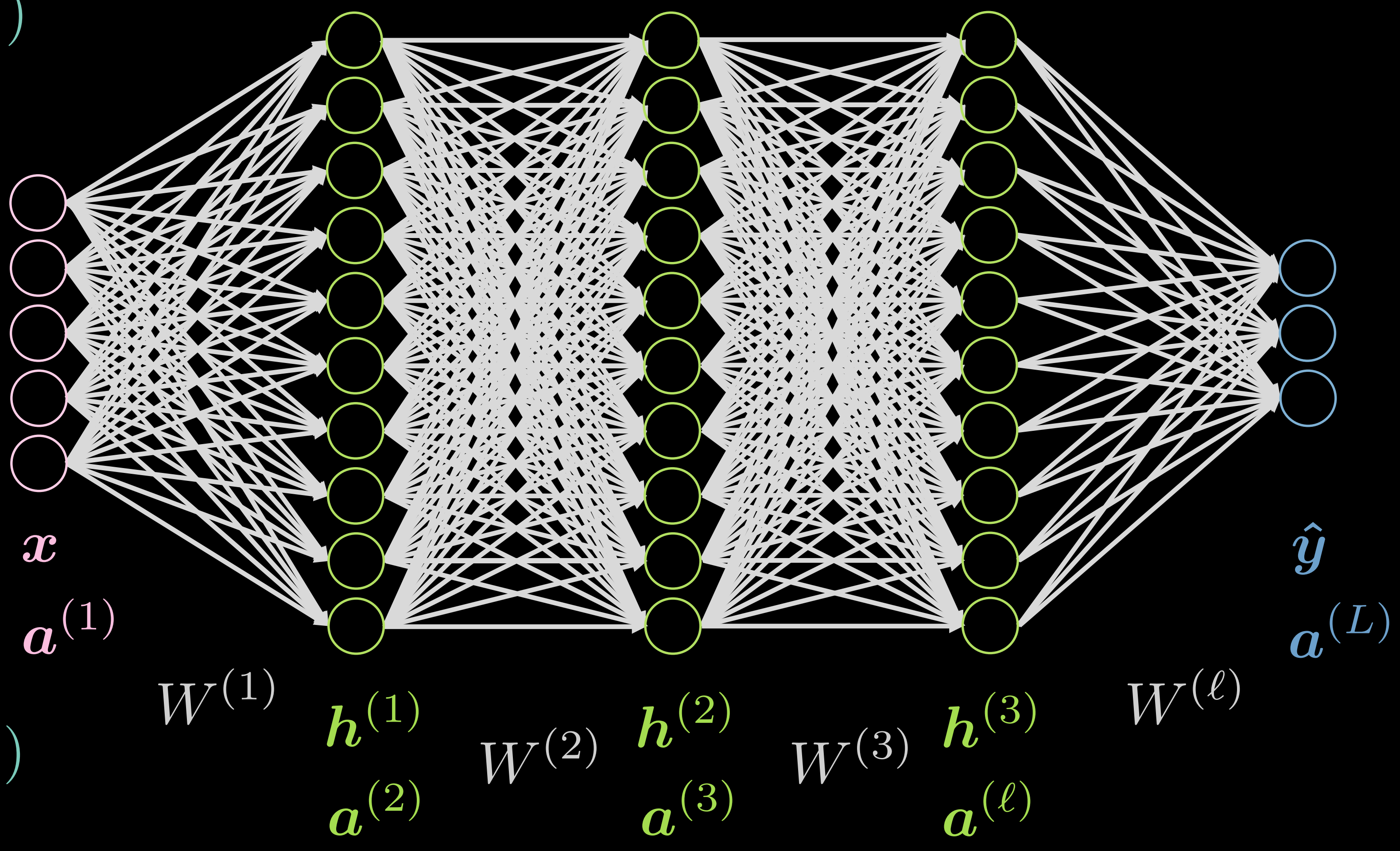
$$a_j^{(2)} = f(\boxed{\boldsymbol{w}^{(j)}}\boldsymbol{x} + b_j) = f\left(\left(\sum_{i=1}^{n} w_i^{(j)} x_i\right) + b_j\right)$$



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

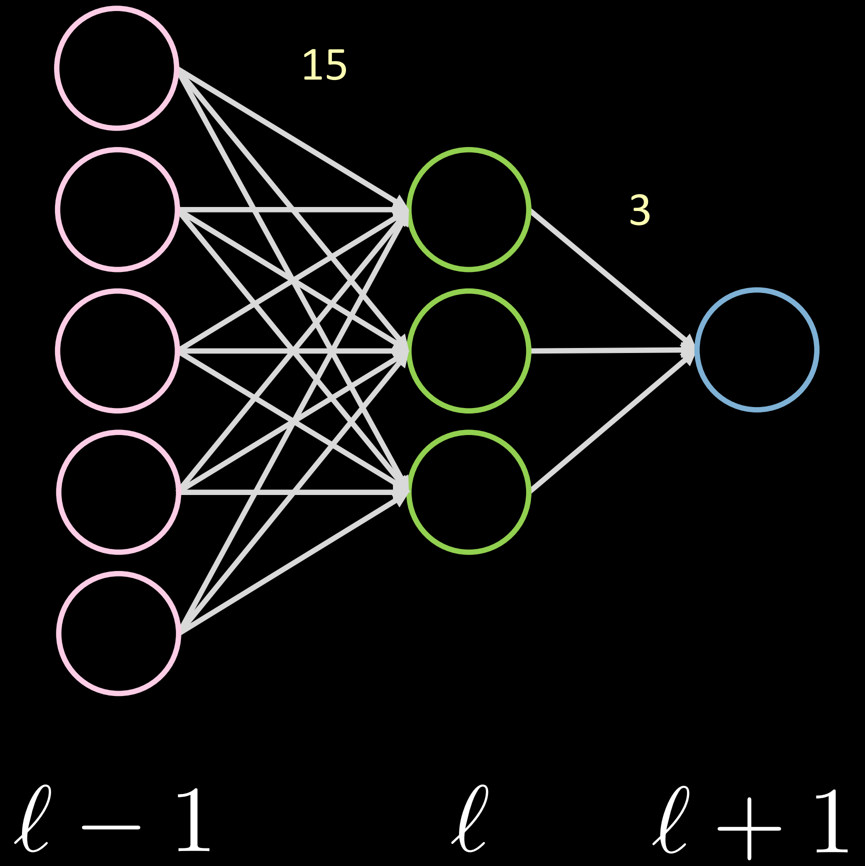
$$\hat{\boldsymbol{y}} = g(\boldsymbol{W_y h} + \boldsymbol{b_y})$$

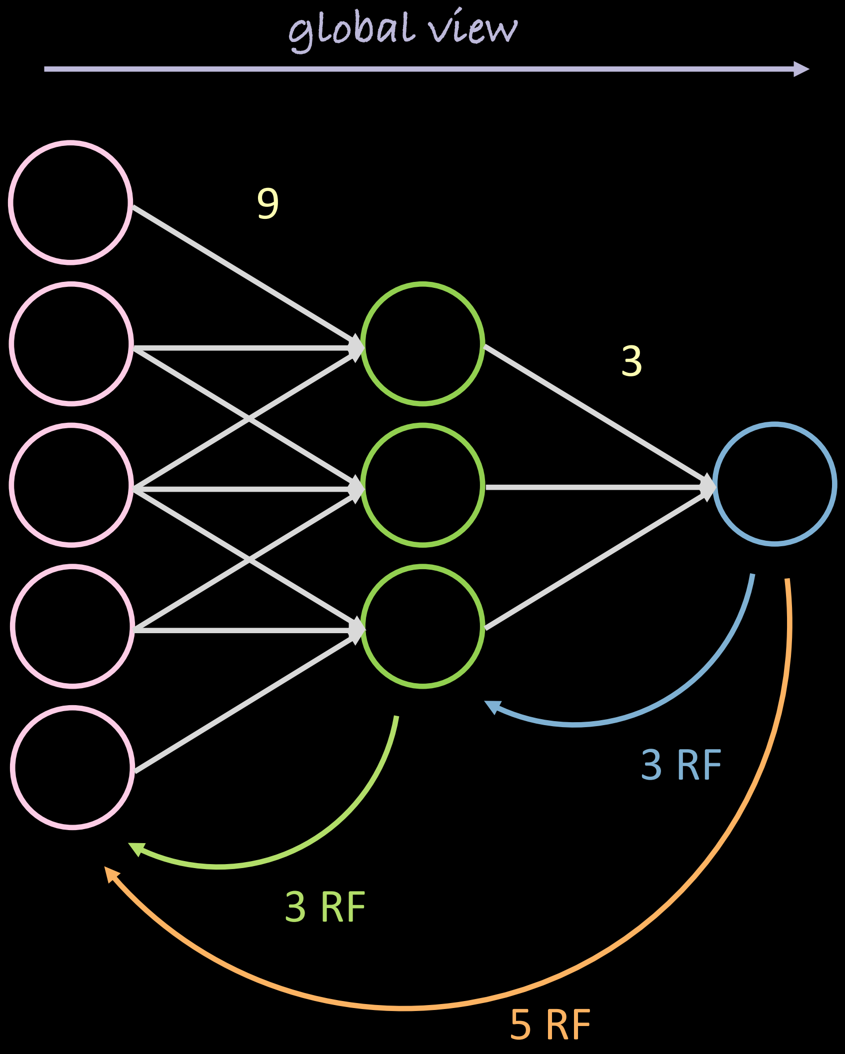
$$f, g = (\cdot)^+, \sigma(\cdot), \tanh(\cdot), \mathrm{softmax}(\cdot)$$



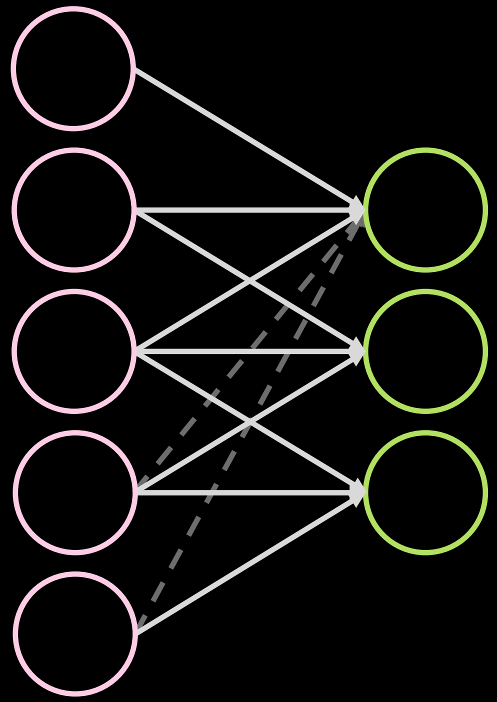
RF: receptive field

Locality ⇒ sparsity





Stationarity ⇒ parameters sharing

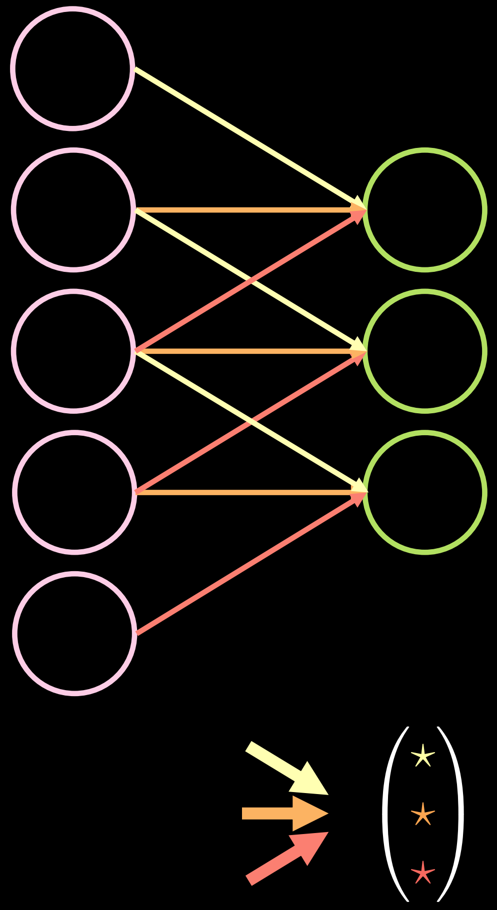


Parameters sharing

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

Connection sparsity

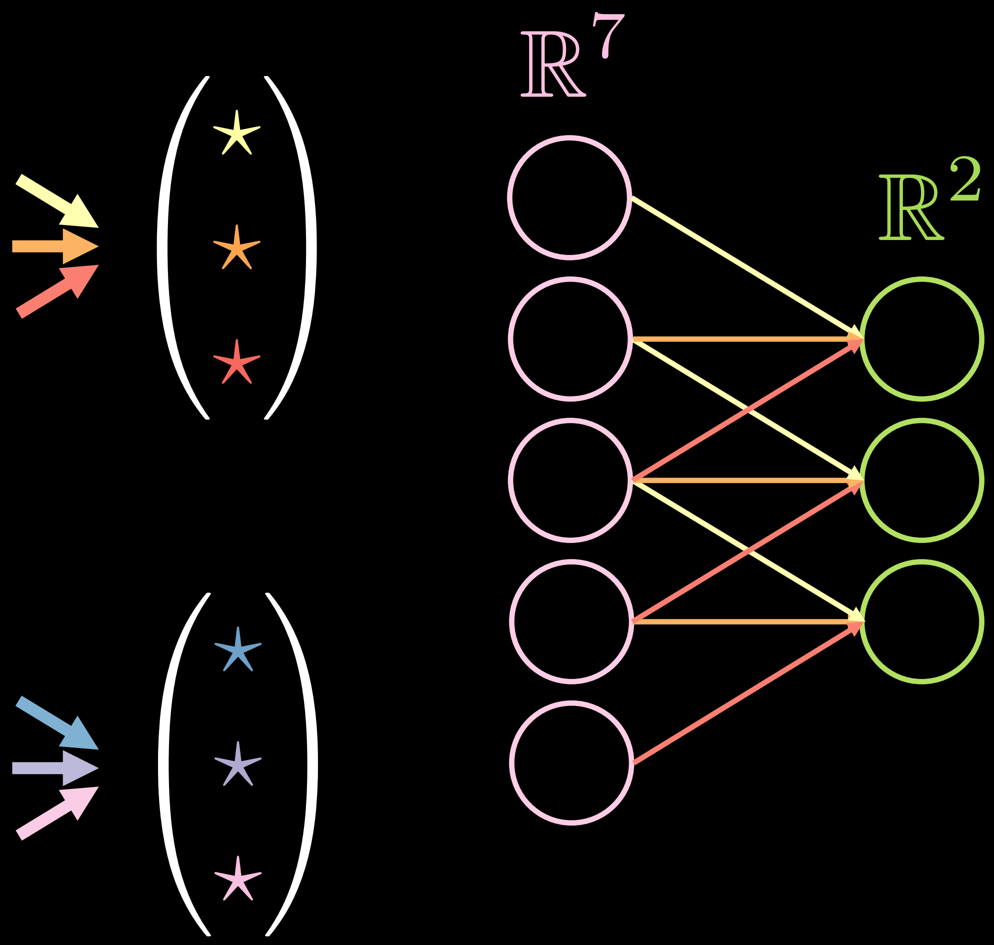
- reduced amount of computation



Kernels – 1D data

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

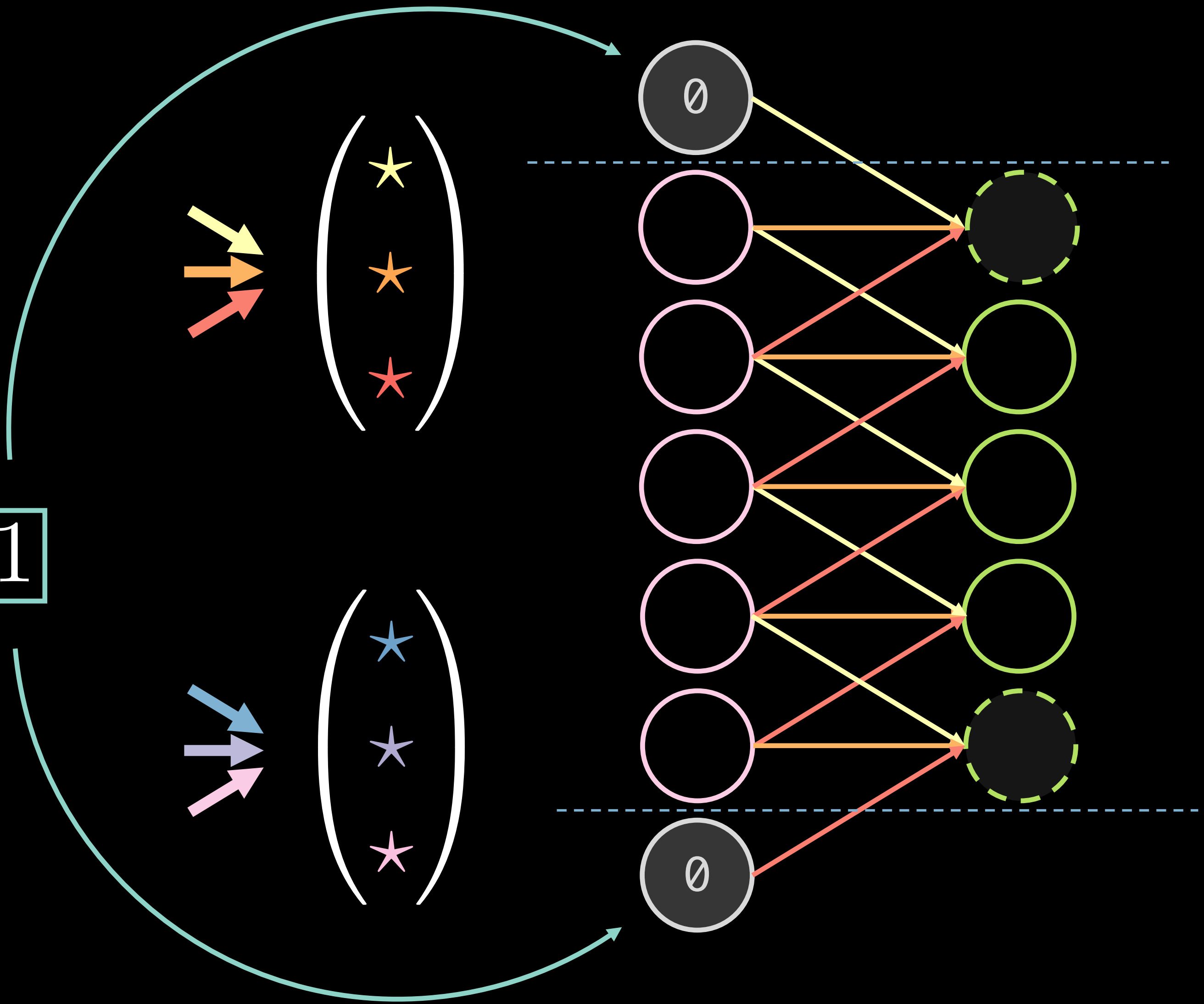
1D data uses 3D kernels!



Padding – 1D data

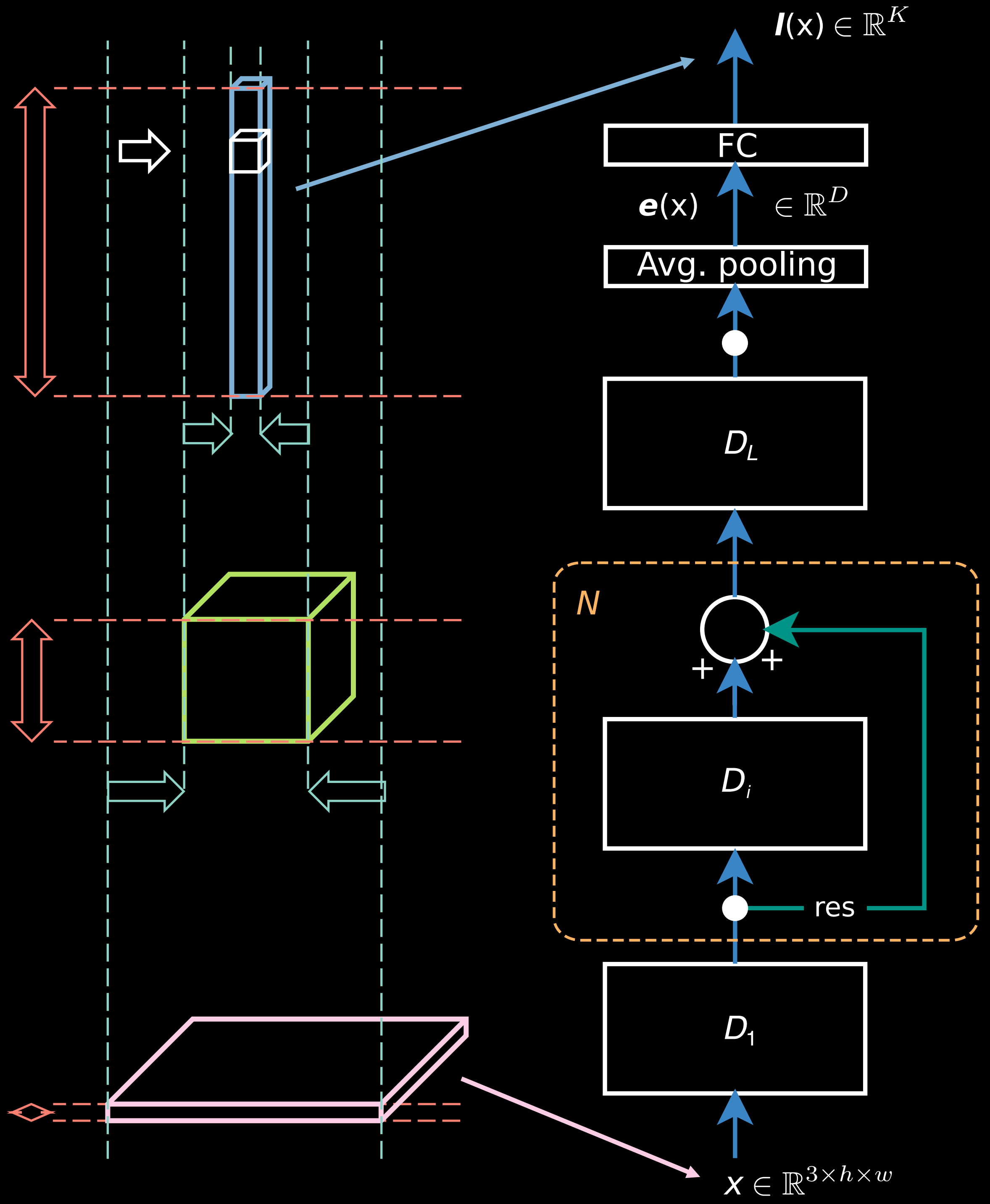
kernel size: $2 \times 7 \times \boxed{3}$

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



Standard spatial CNN

- Multiple layers
  - Convolution
  - Non-linearity
  - Pooling
  - Batch normalisation
- Residual bypass connection



Pooling

$$\|\underline{x}\|_p := \left(\sum_i |x_i|^p \right)^{1/p}$$

$$\|\underline{x}\|_p \to \max(\underline{x}), \quad p \to +\infty$$