ConvNets

→ Used for problems that have grid-like topology.

* 1D → Time series

* 2D → Image recognition

Convolution

$$s(t) = \underbrace{(x * w)(t)}_{\int x(a)w(t-a)\,da} \quad \Big\}\ \text{feature map}$$

input, kernel

$$2D \rightarrow S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)\,K(m,n)$$

Kernel Input

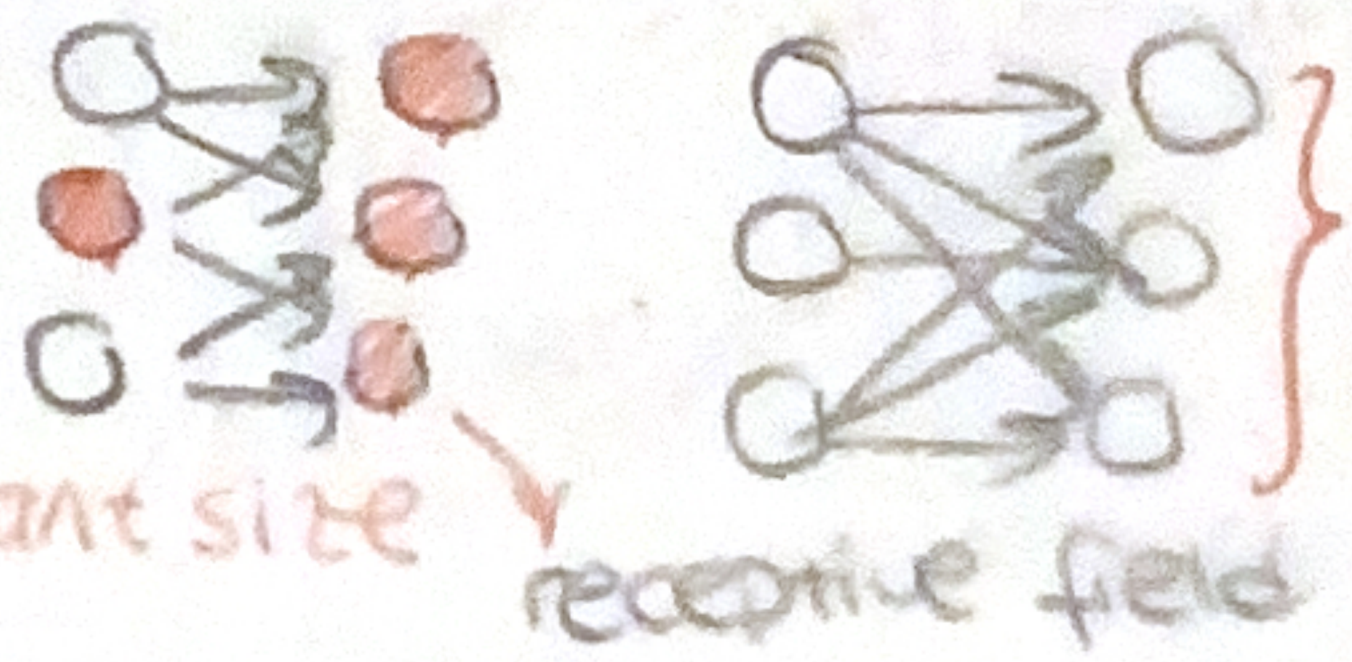
Why?

→ Sparse Interactions: Features can be detected with less params

→ Parameter Sharing

→ Equivariant Representations

→ We can work with inputs of variant size



- Sparsity: we store less info thanks to kernels.

m input, n output } m×n params, $O(m\times n)$

k → number of connections each input can have

$O(k\times n)$ → sparsely connected approach

- Parameter Sharing: Using same parameter for more than one function in a model.

ex// Dense Nets → Each element of weight matrix is multiplied for once with input

CNNs → Has tied weights, value of the weight applied to one input is tied to another value elsewhere. Each element in kernel is used at every position of the input. (Except for boundary pixels)