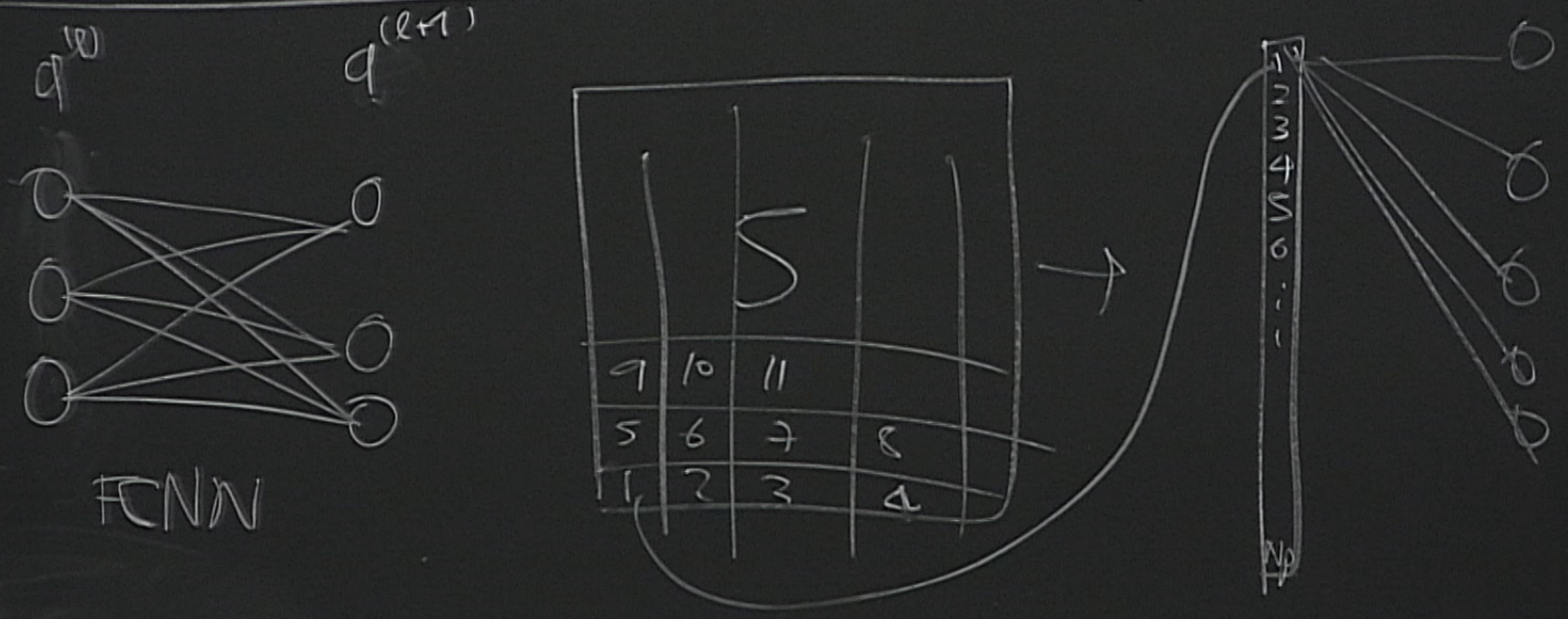
Title: PSI 2017/2018 - Machine Learning for Many Body Physics - Lecture 6

Date: Apr 16, 2018 09:00 AM

URL: http://pirsa.org/18040059

Abstract:

Convolutional neural networks



$t=0$

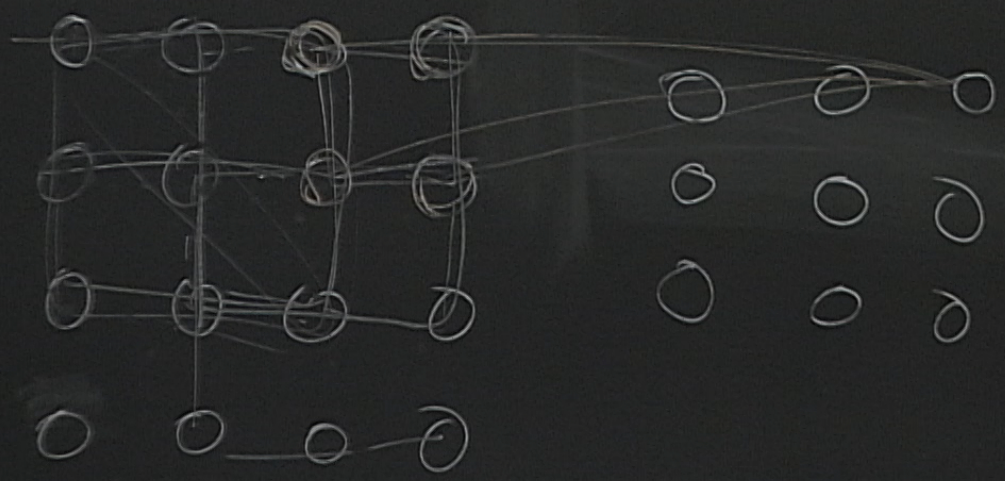
i

Ingredients

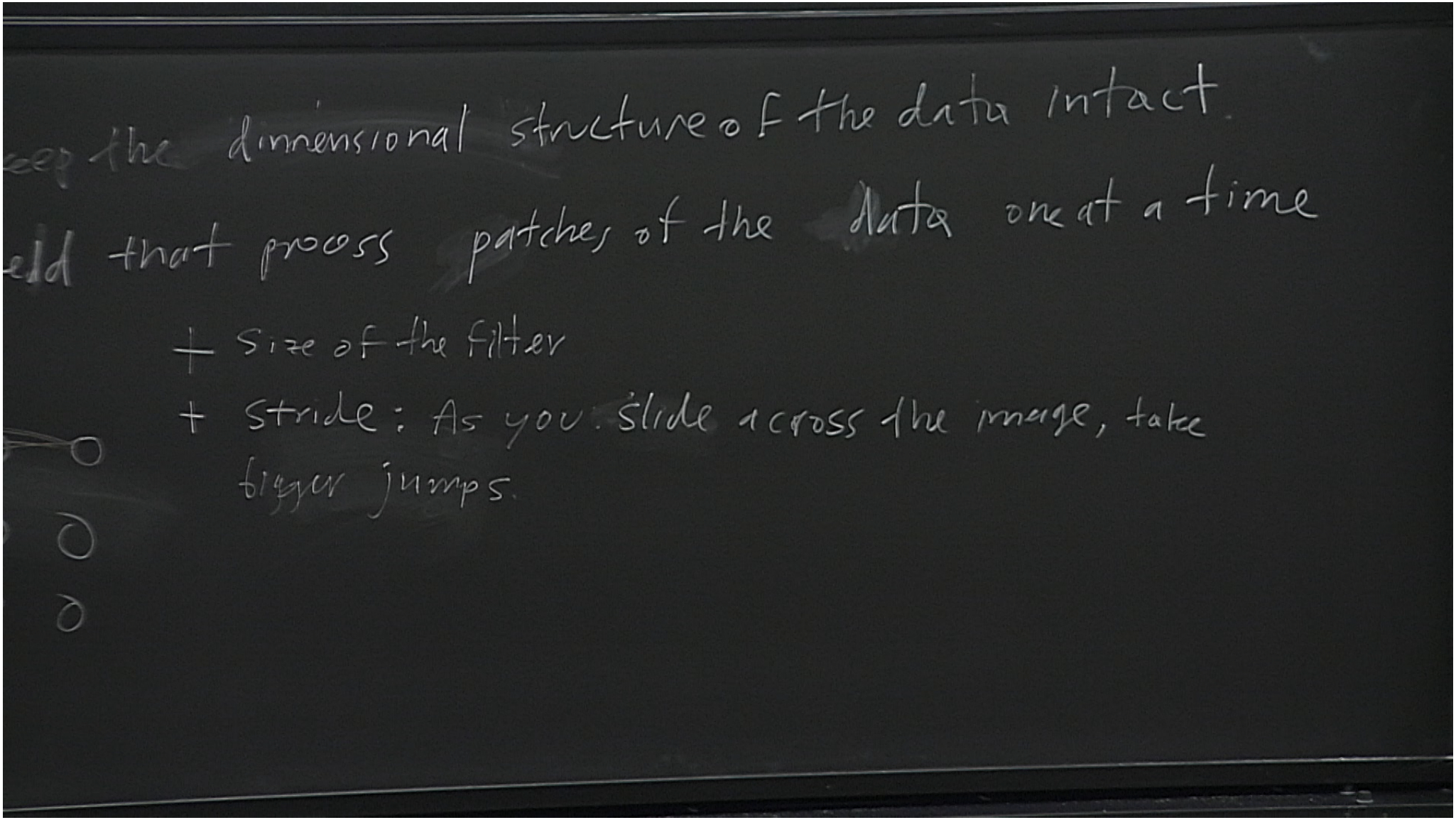
* Local receptive fields: keep the dimensional structure o

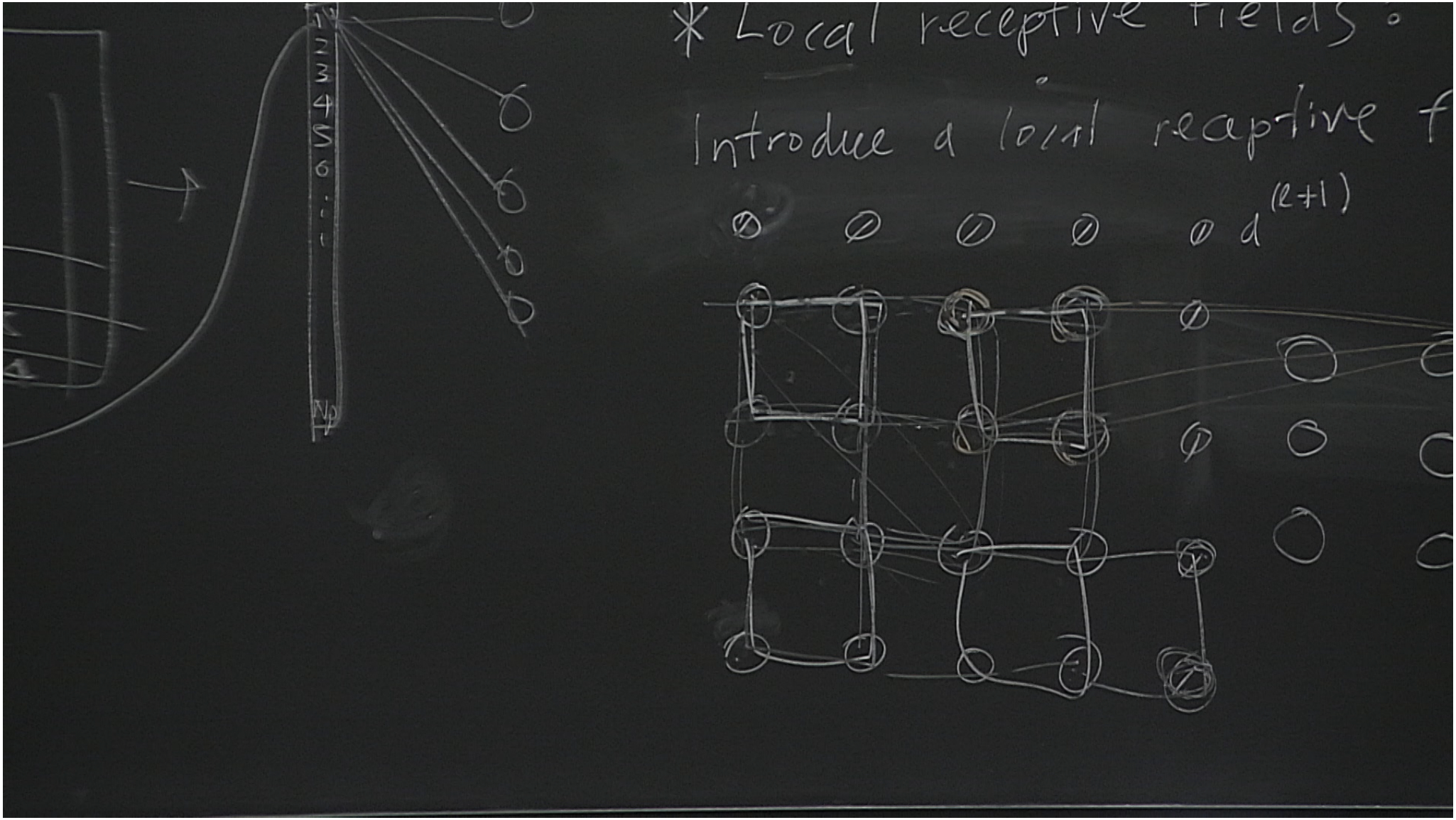
Introduce a local receptive field that process patches of th

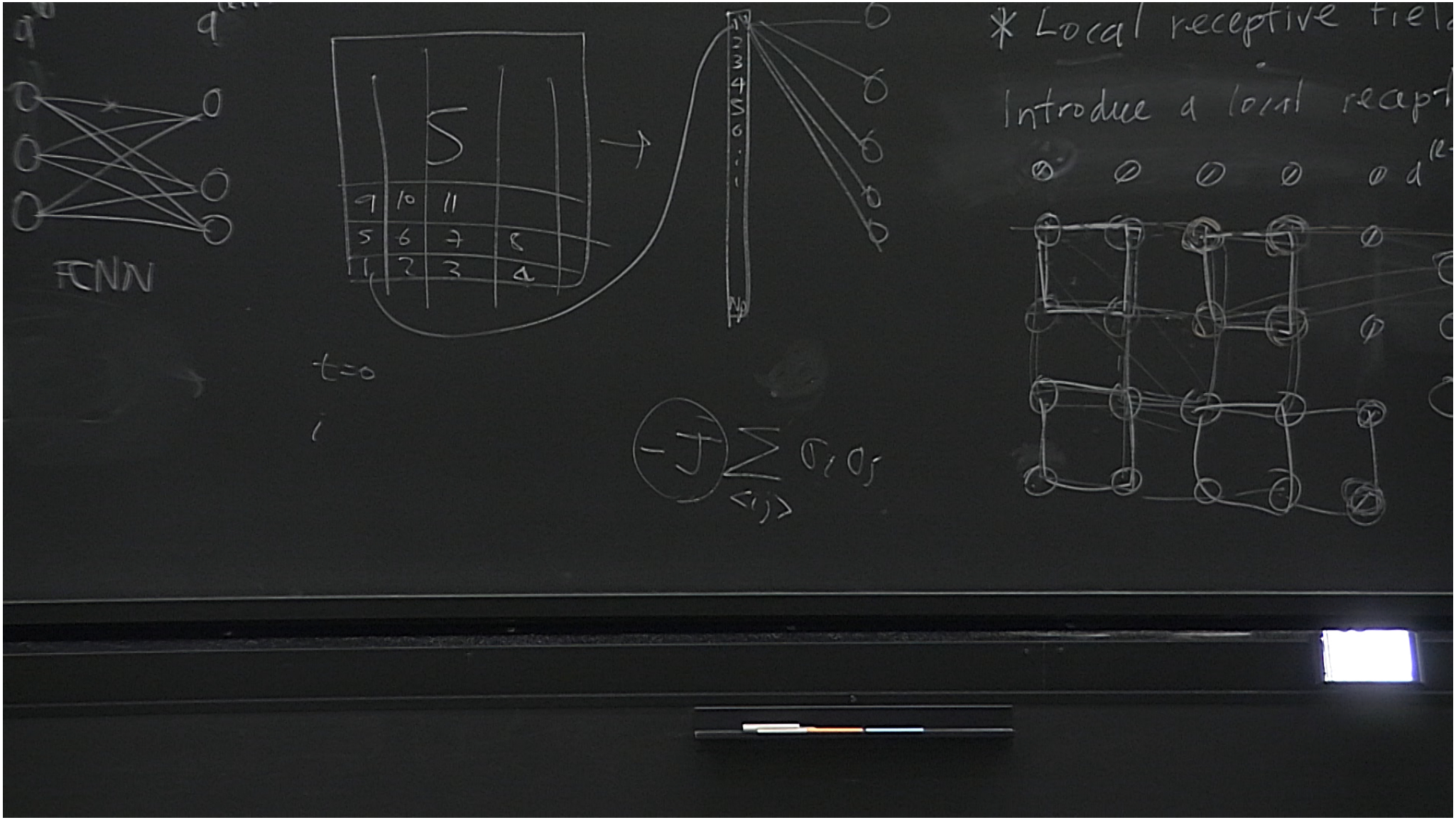
$d^{(l)}$ $d^{(l+1)}$

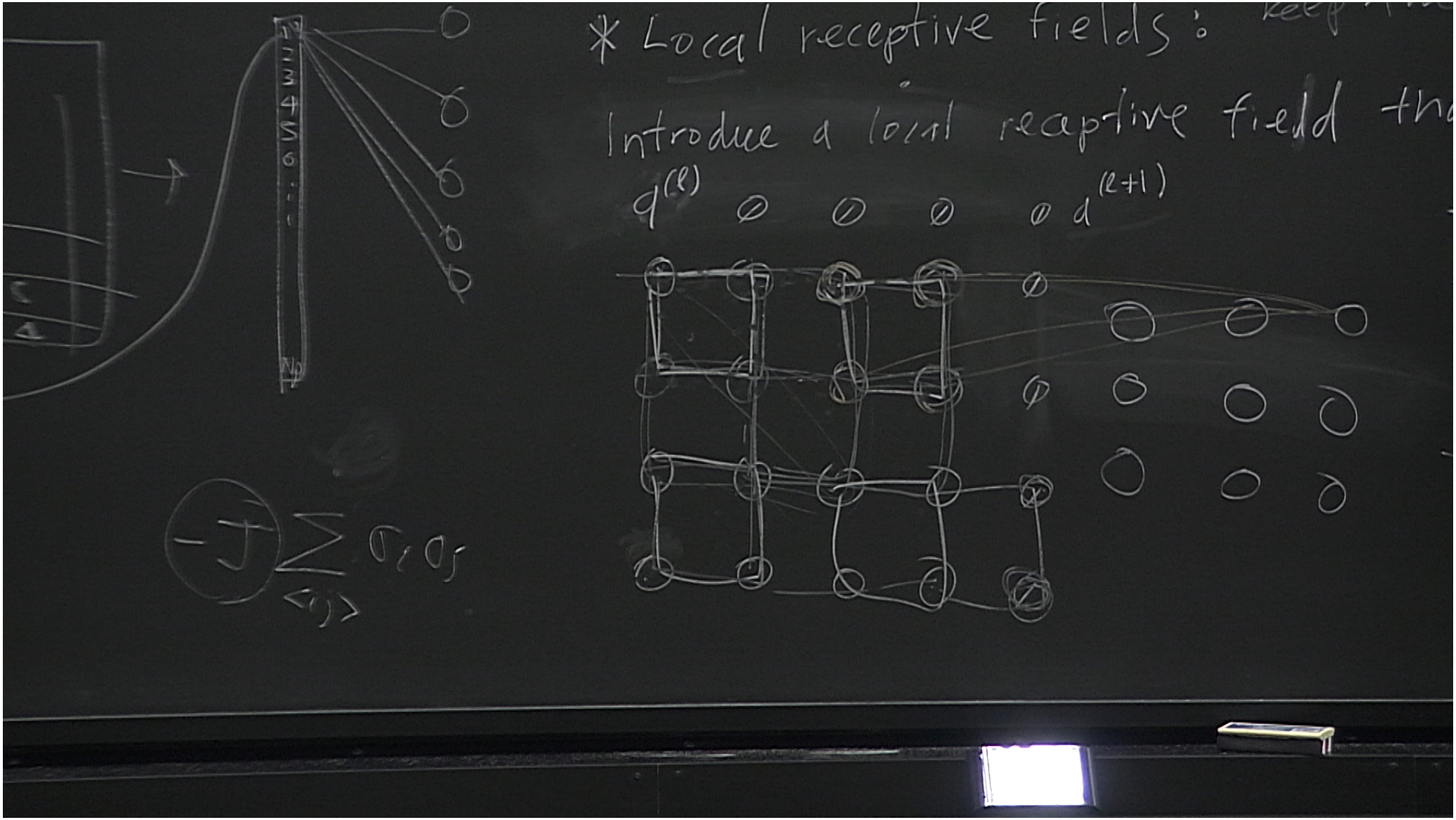


+ Size of the filter
+ Stride


keep the dimensional structure of the data intact.
eld that process patches of the data one at a time
+ Size of the filter
+ Stride: As you slide across the image, take
bigger jumps.


* Local receptive fields:
Introduce a local receptive f
$d^{(l+1)}$


FCNN
t=0
* Local receptive fiel
Introduce a local recept

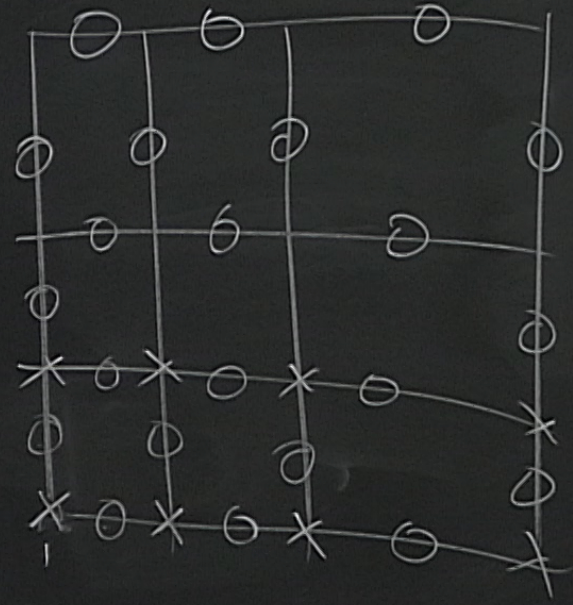

* Local receptive fields:
Introduce a local receptive field
$q^{(l)}$
$q^{(l+1)}$
$-J \sum_{\langle ij \rangle} \sigma_i \sigma_j$

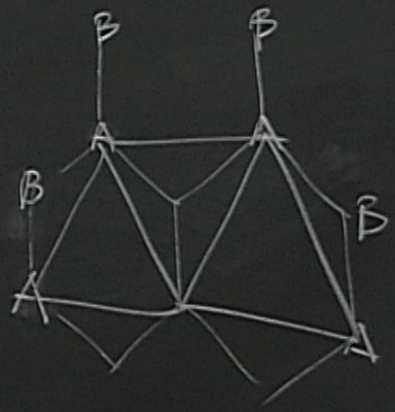
* Shared weights and biases

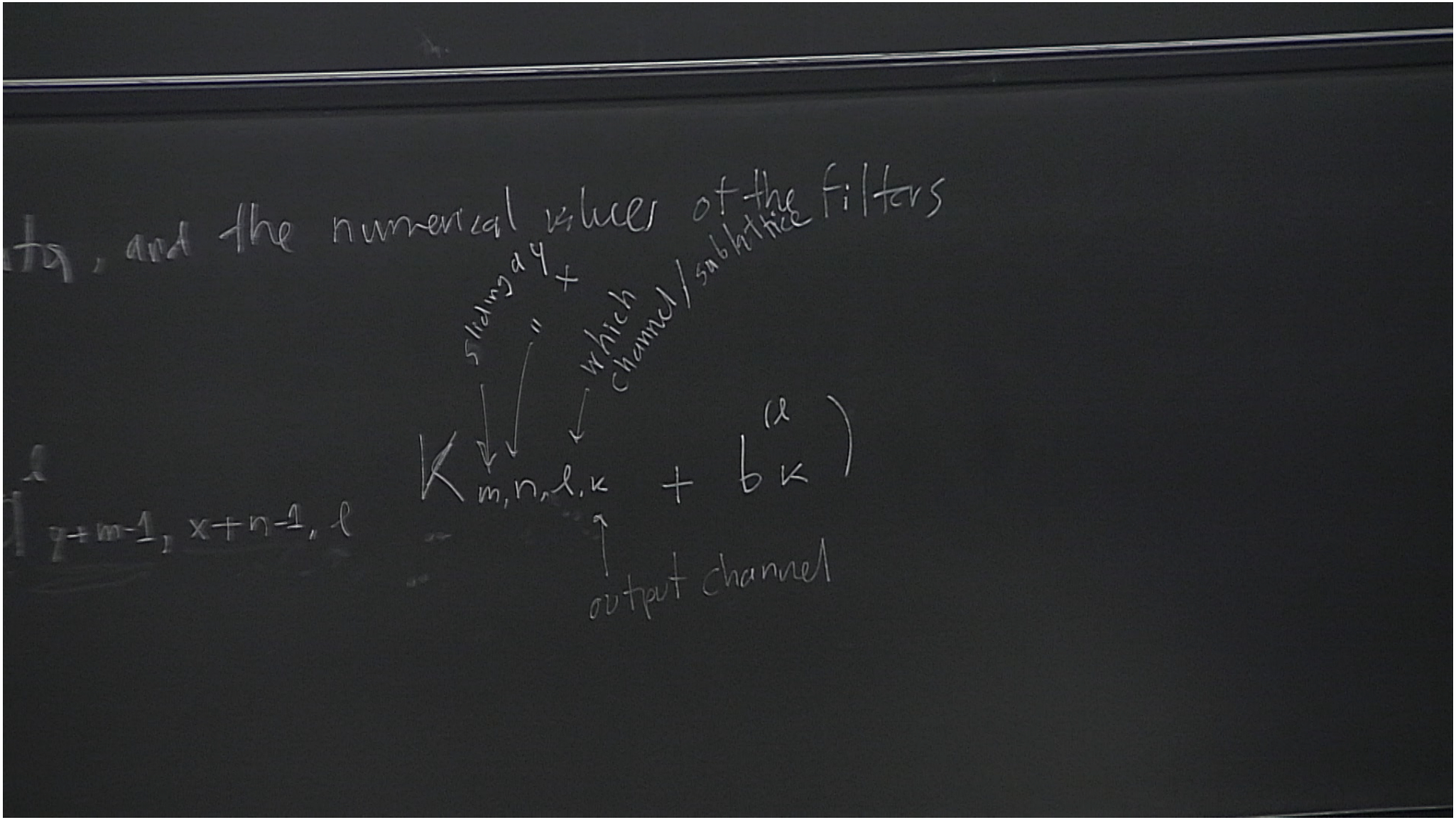
LRF is moved across the data, and the numerical values of the filters remain the same

$$a^{(l+1)}_{y,x,k} = f\left(\sum_{m,n,l} a^{l}_{y+m-1,\, x+n-1,\, l}\, K_{m,n,l,k} + b^{(l)}_{k}\right)$$

Convolutional neural networks






..., and the numerical values of the filters
sliding y, x
which channel / sublattice
$K_{m,n,\ell,k} + b_k^{(\ell)}$
$y+m-1, x+n-1, \ell$
output channel

* Shared weights and biases

LRF is moved across the data, and the numerical values of the filter remain the same

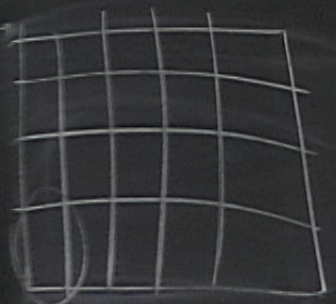
$$a^{(l+1)}_{y,x,k} = f\left(\sum_{m,n,l} a^{l}_{y+m-1,\, x+n-1,\, l}\; K_{m,n,l,k} + b^{(l)}_{k}\right)$$

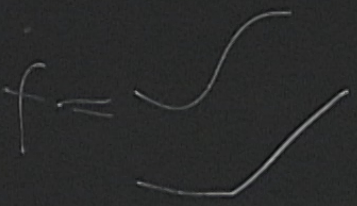
sliding y, x

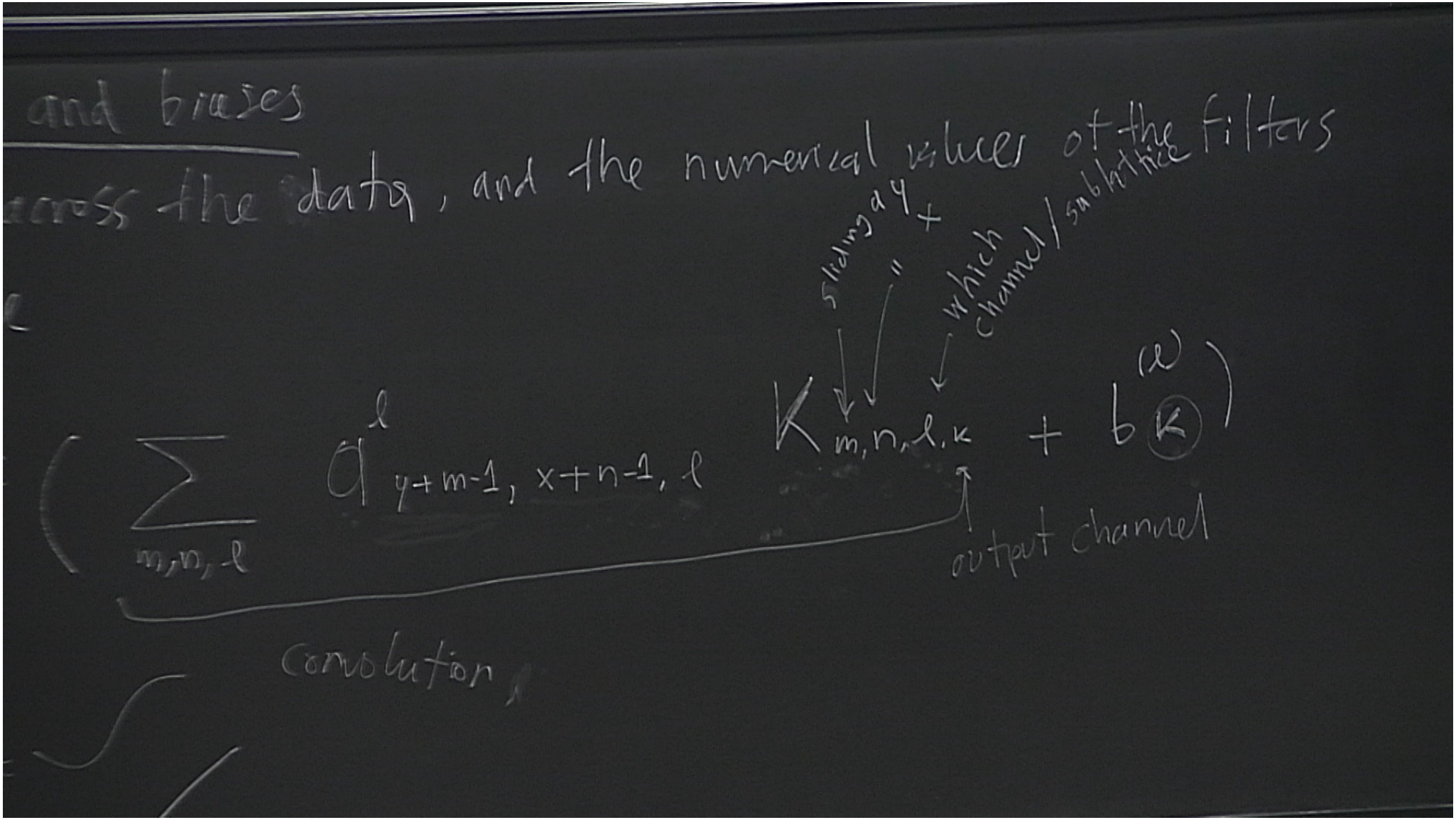
which channel / subhidden

output channel

convolution

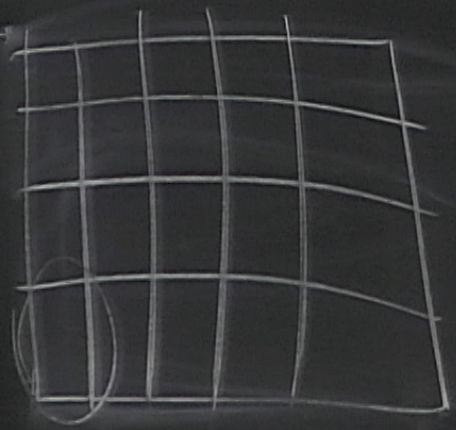


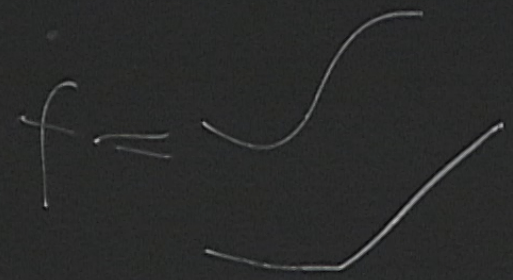



and biases
across the data, and the numerical values of the filters
sliding y
x
which channel / sublattice
$\left(\sum_{m,n,l} a^{l}_{y+m-1,\, x+n-1,\, l} K^{l}_{m,n,l,k} + b^{(l)}_{k}\right)$
output channel
convolution

remain the same

$$a^{(l+1)}_{y,x,k} = f\left(\sum_{m,n,l} a^{l}_{y+m-1,\ x+n-}\right)$$





convolution