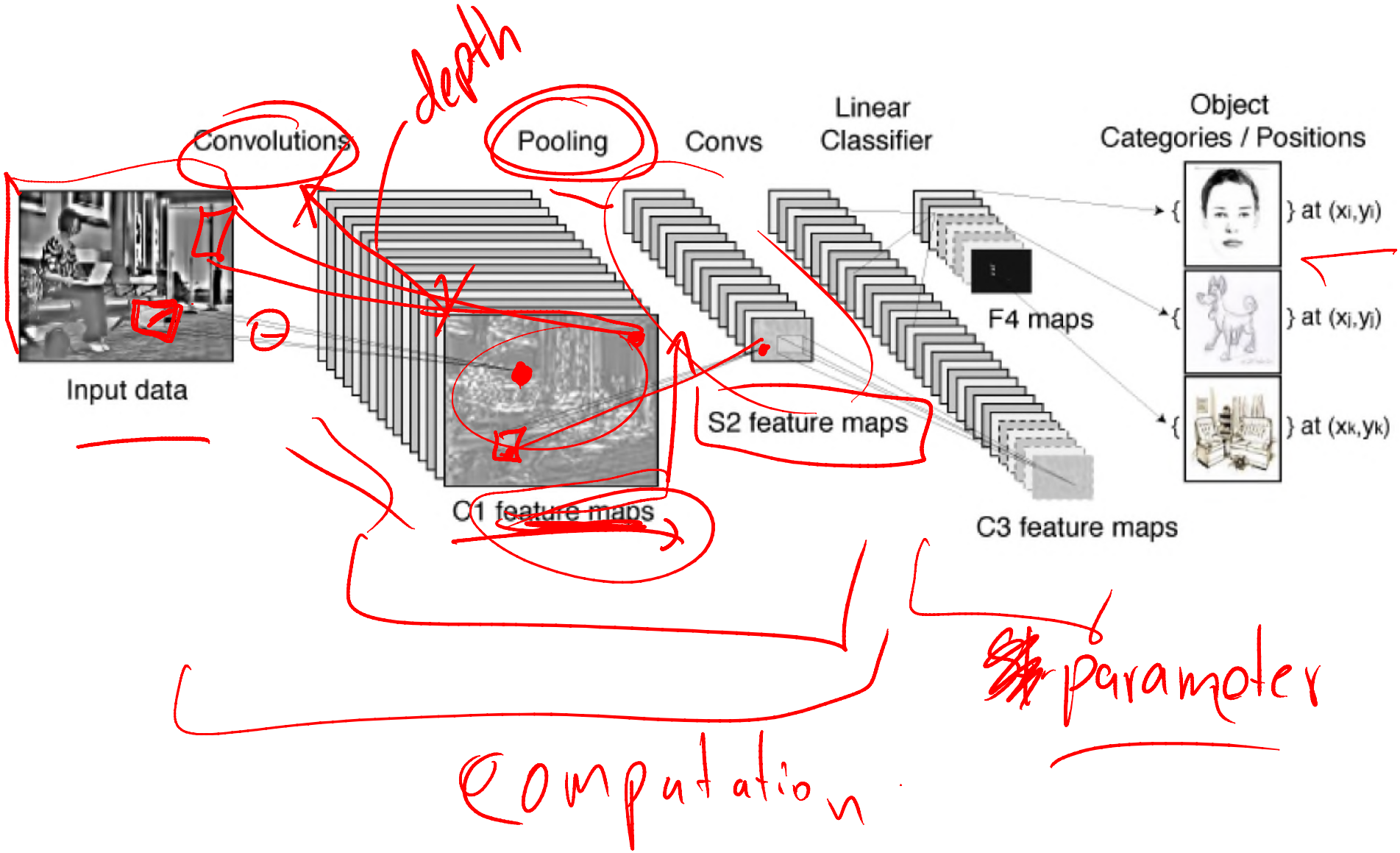

Convnets
Nando de Freitas
UNIVERSITY OF
OXFORD

Outline of the lecture

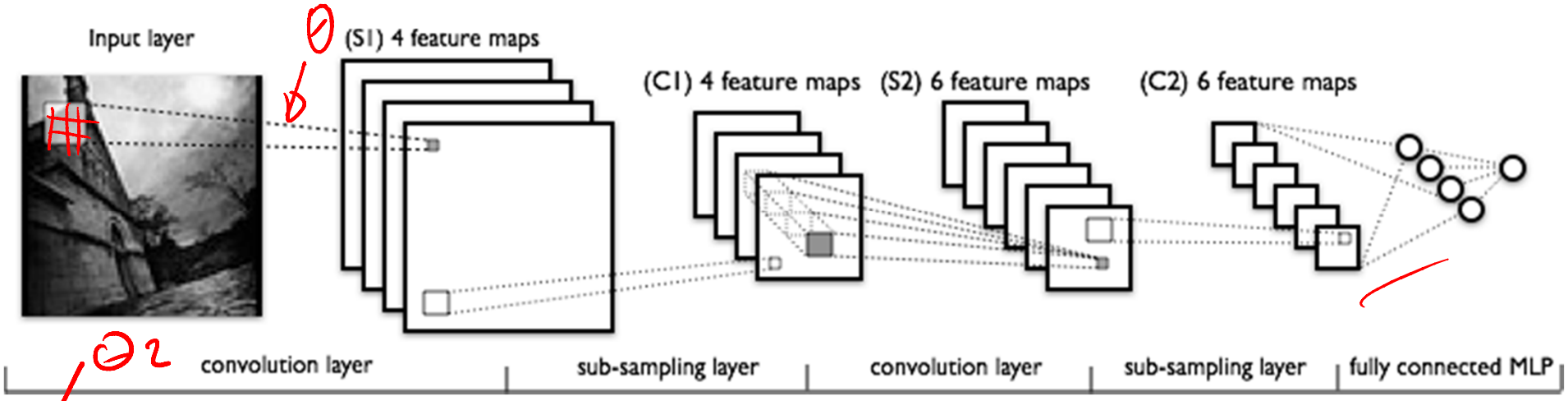
This lecture introduces you to convolutional neural networks. These models have revolutionized speech and object recognition. The goal is for you to learn

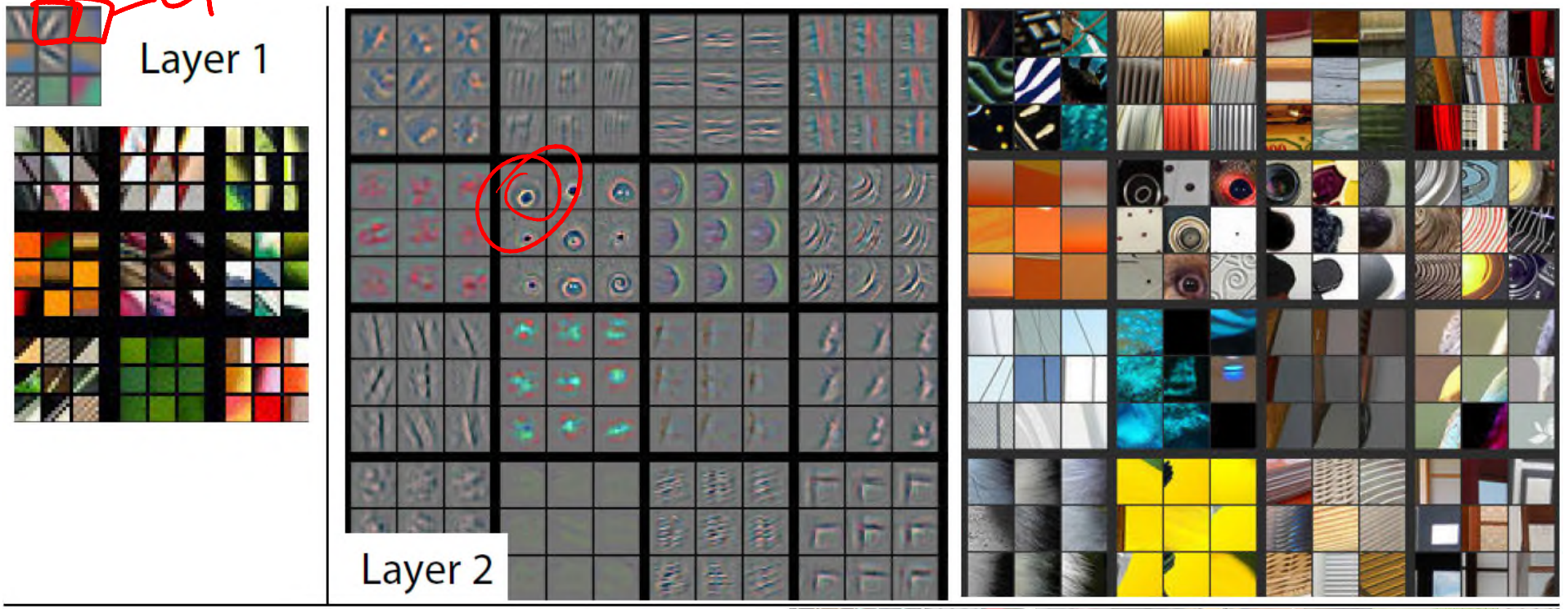
- ❑ Convnets for object recognition and language
- ❑ How to design convolutional layers
- ❑ How to design pooling layers
- ❑ How to build convnets in torch

Convnets (Fukushima, LeCun, Hinton)

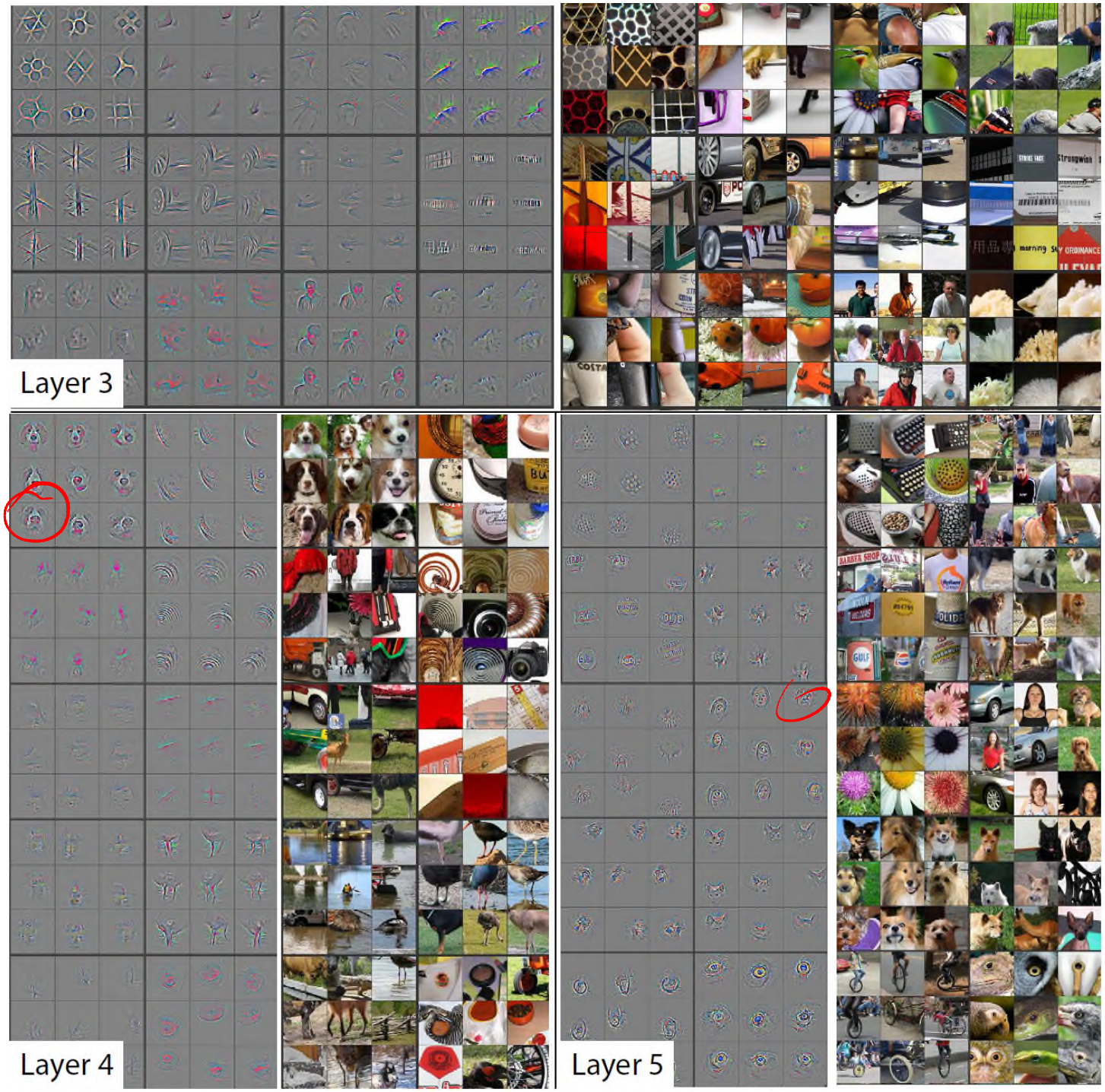


Convolutional networks

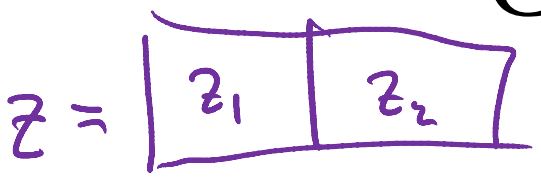


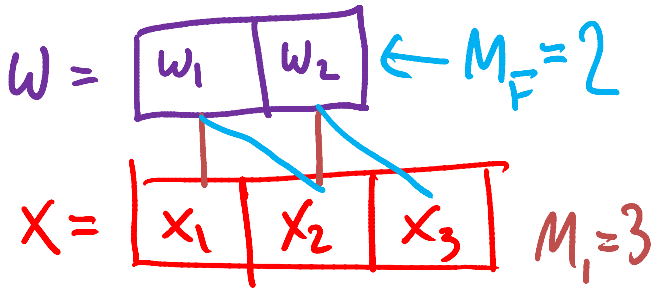


[Matthew Zeiler & Rob Fergus]

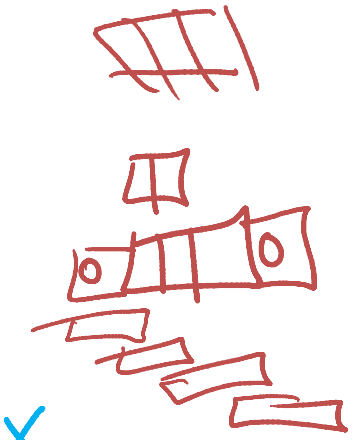

Layer 3
Layer 4
Layer 5

Convolution



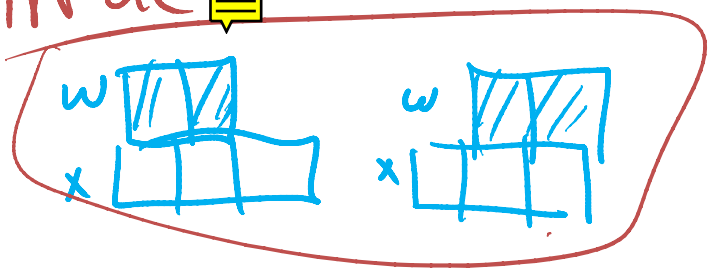


$$\begin{cases} z_1 = w_1 x_1 + w_2 x_2 \\ z_2 = w_1 x_2 + w_2 x_3 \end{cases}$$



zero padding

Stride



$$z_{i'} = \sum_{i=1}^{M_F=2} w_i x_{i'+i-1}$$

Correlation (similarity)

$$\| $$

$$z_{i'} = \sum_{i=1}^{M_F=2} x_{i'+i-1} \bar{w}_{M_F-i+1}$$

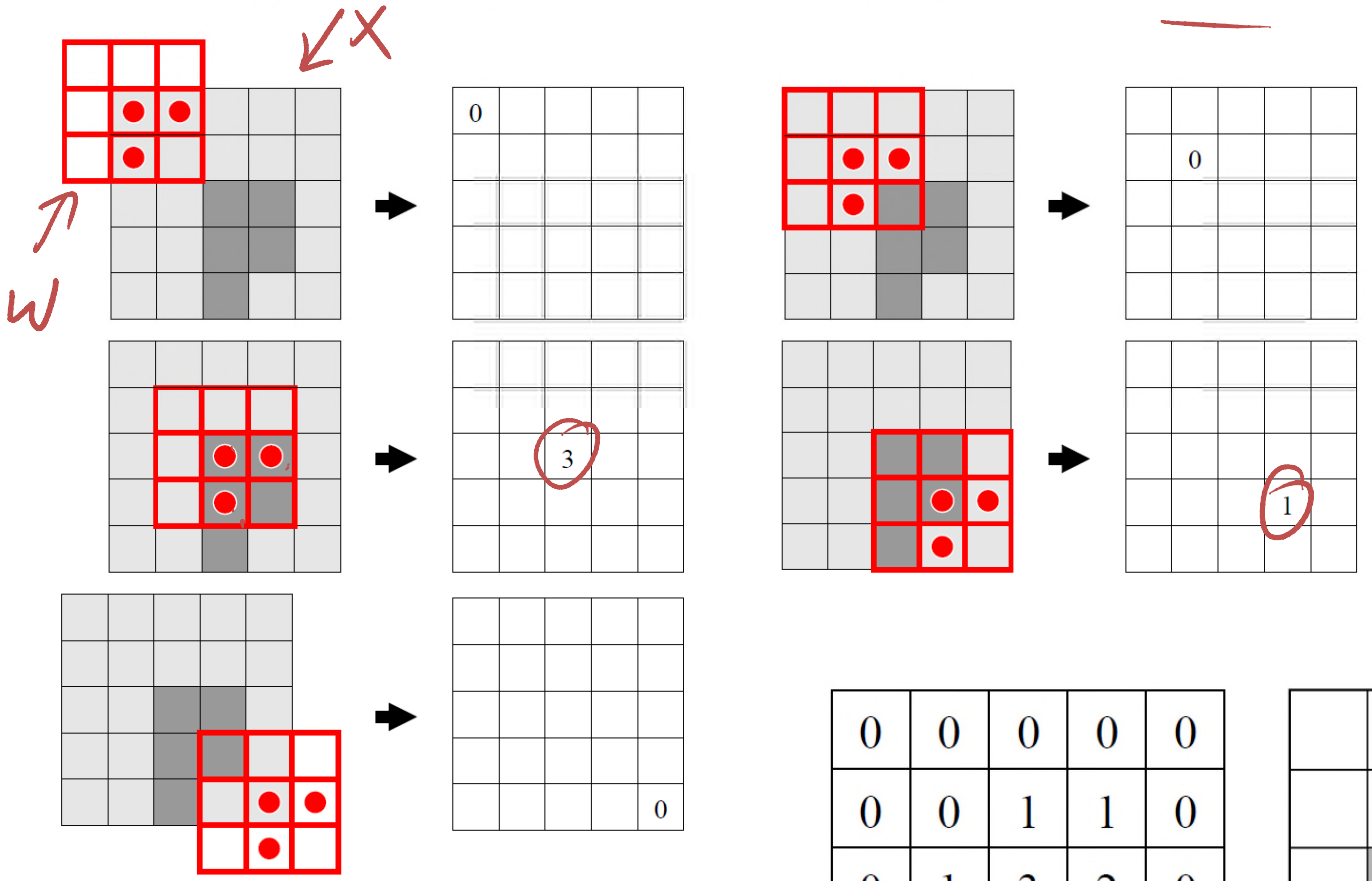
(convolution)

$\vec{w} \rightarrow$ flip $\rightarrow \bar{w}$ =

Andrej Karpathy

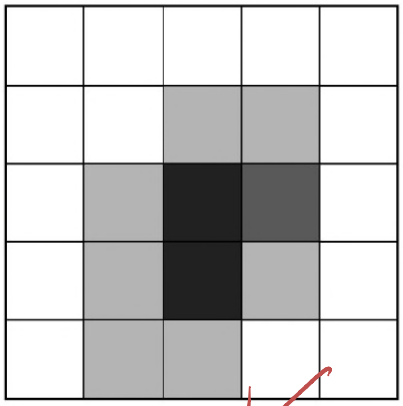
Image convolution

For the image, take dark pixel value = 1, light pixel value = 0.



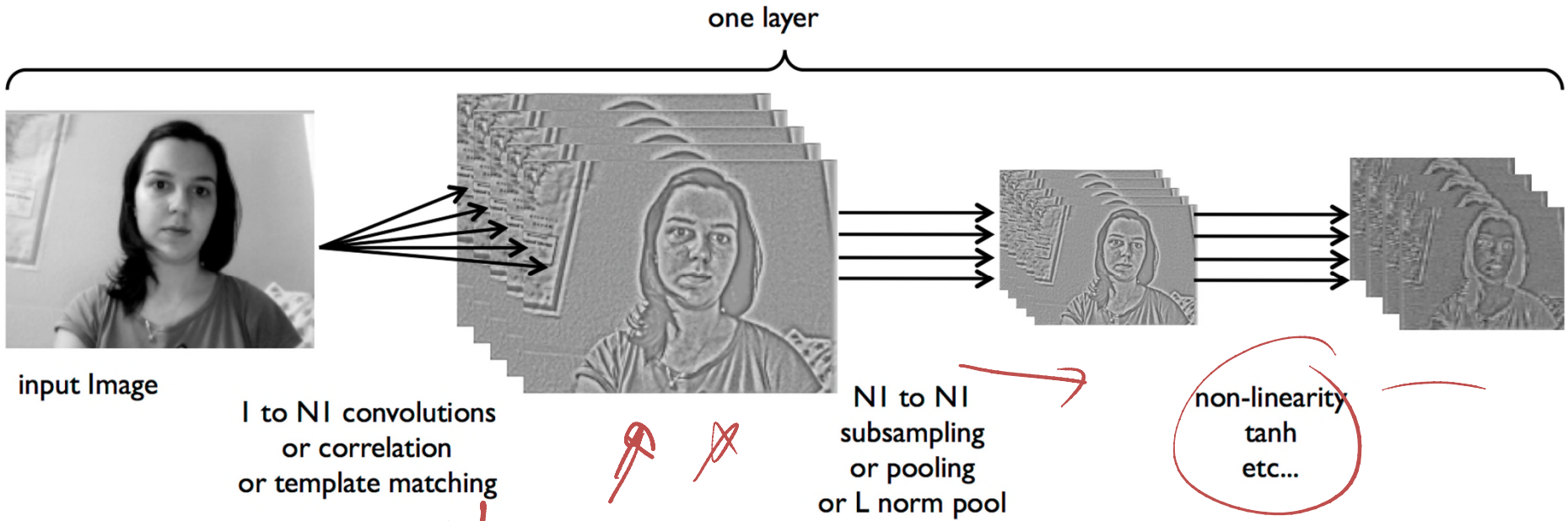
feature map

0	0	0	0	0
0	0	1	1	0
0	1	3	2	0
0	1	3	1	0
0	1	1	0	0



Leow Wee Kheng

Convnets (Fukushima, LeCun, Hinton)



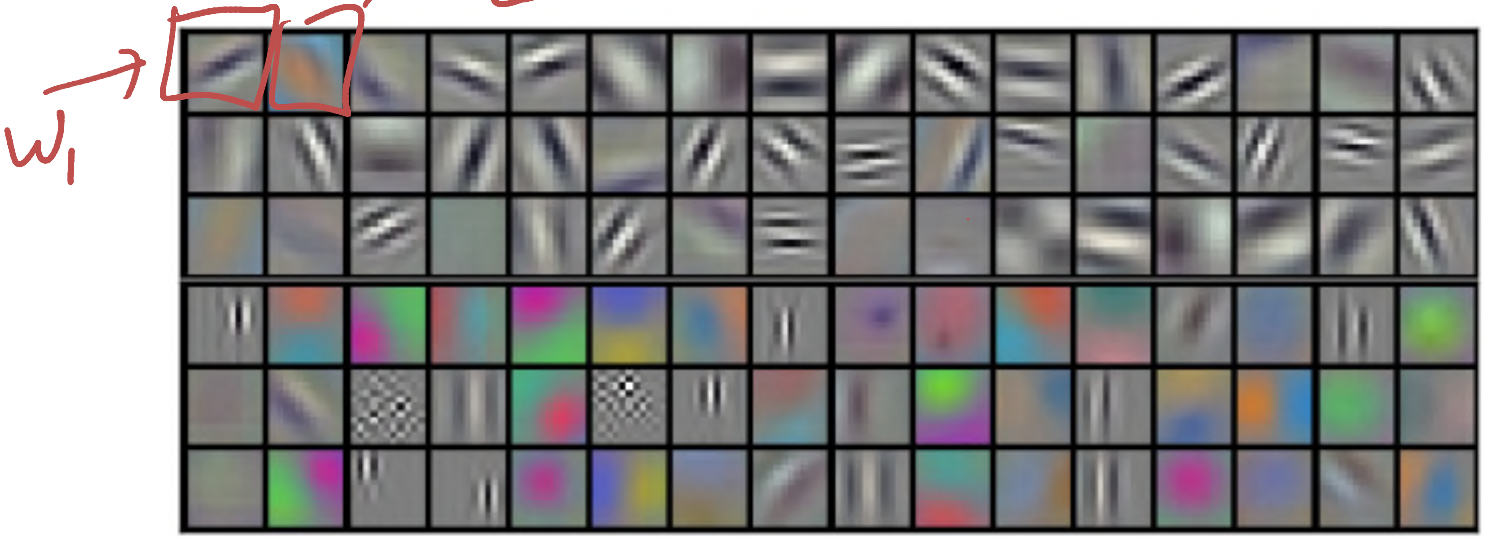
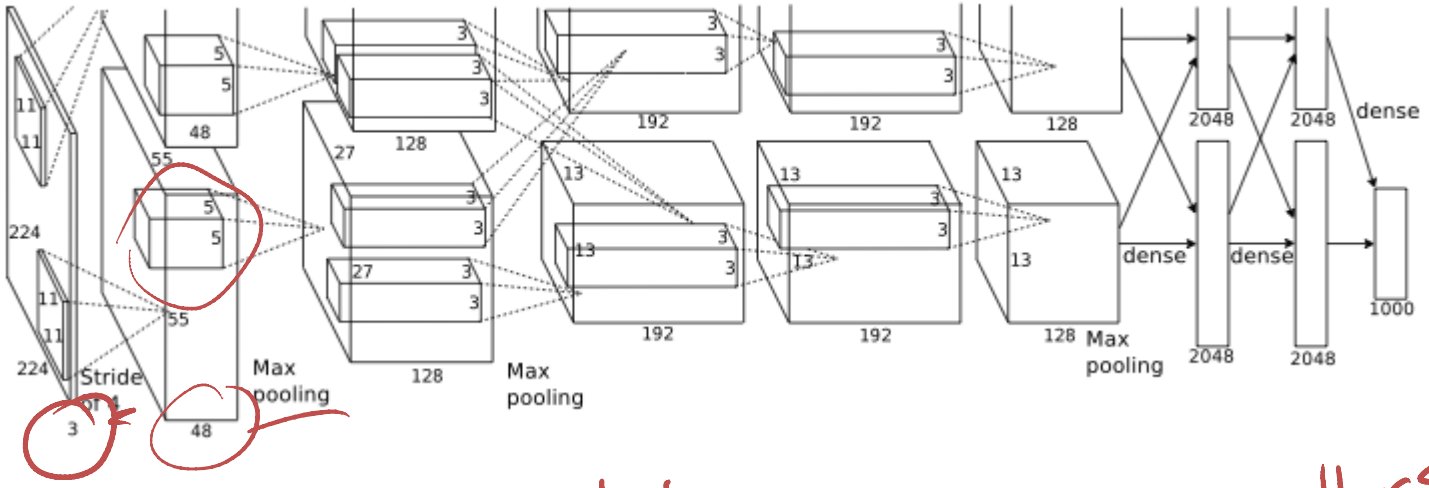


Image convolution layer



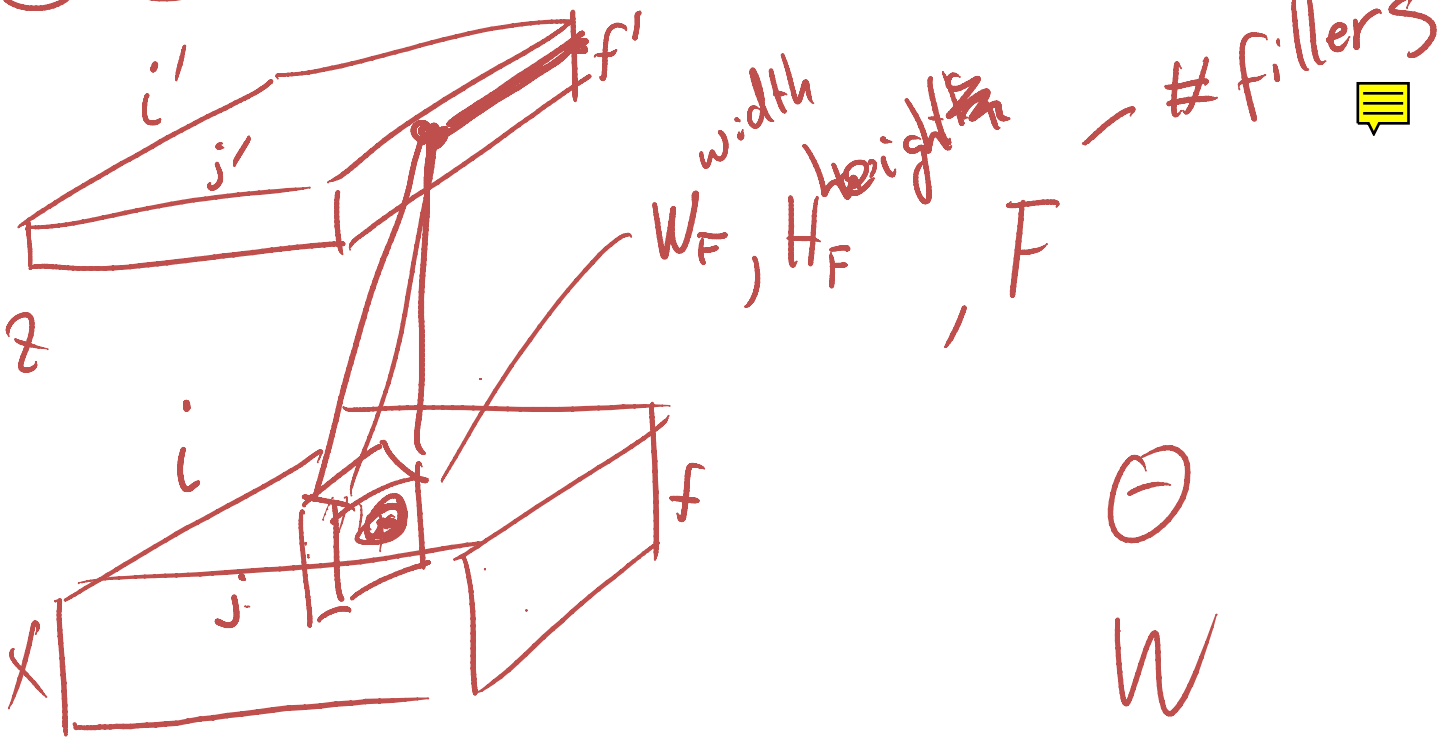
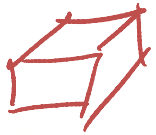
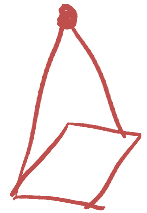


Image convolution layer





$$\mathbf{y}_{i',j',f'} = b_{f'} + \sum_{i=1}^{H_f} \sum_{j=1}^{W_f} \sum_{f=1}^{F} \mathbf{x}_{i'+i-1,j'+j-1,f} \boldsymbol{\theta}_{ijff'}$$

$$\frac{\partial E}{\partial \theta_{ijff'}} = \sum_{i'j'f'} \delta^{l+1}_{i'j'f'} \frac{\partial f_{i'j'f'}(\mathbf{x}; \boldsymbol{\theta}_{f'})}{\partial \theta_{ijff'}}$$

$$= \sum_{i'j'} \delta^{l+1}_{i'j'f'} \mathbf{x}_{i'+i-1,j'+j-1,f}$$

Image convolution layer

$$\mathbf{y}_{i',j',f'} = b_{f'} + \sum_{i''=1}^{H_f} \sum_{j''=1}^{W_f} \sum_{f''=1}^{F} \mathbf{x}_{i'+i''-1,j'+j''-1,f''} \boldsymbol{\theta}_{i''j''f''f'}$$

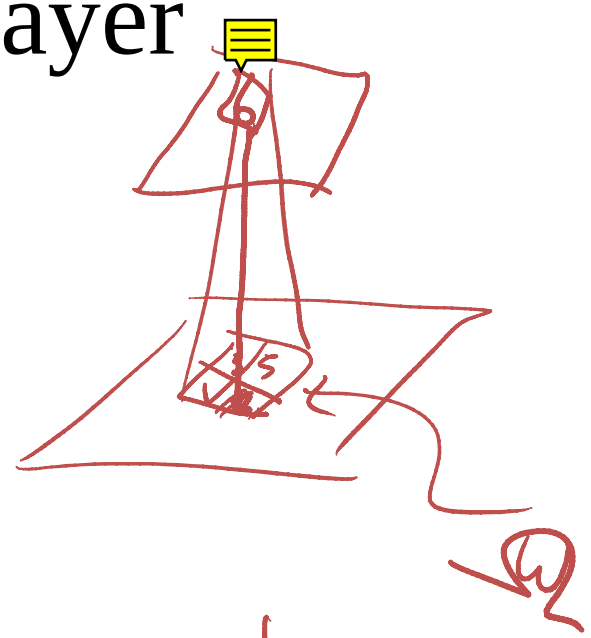
$$\delta^l_{ijf} = \sum_{i'j'f'} \delta^{l+1}_{i'j'f'} \frac{\partial f_{i'j'f'}(\mathbf{x}; \boldsymbol{\theta}_{f'})}{\partial \mathbf{x}_{ijf}}$$

$$i = i' + i'' - 1$$

$$i'' = i - i' + 1$$

$$= \sum_{i'j'f'} \delta^{l+1}_{i'j'f'} \theta_{i-i'+1,j-j'+1,f,f'}$$

Image max-pooling layer



$$\mathbf{y}_{i',j'} = \max_{ij \in \Omega(i'j')} \mathbf{x}_{ij}$$

$$\delta_{ij}^{l} = \sum_{i'j'} \delta_{i'j'}^{l+1} \frac{\partial f_{i'j'}(\mathbf{x})}{\partial \mathbf{x}_{ij}}$$

$$= \delta_{ij}^{l+1} \mathbb{I}_{ij = \arg\max_{i''j'' \in \Omega(i'j')} \mathbf{x}_{i''j''}}$$

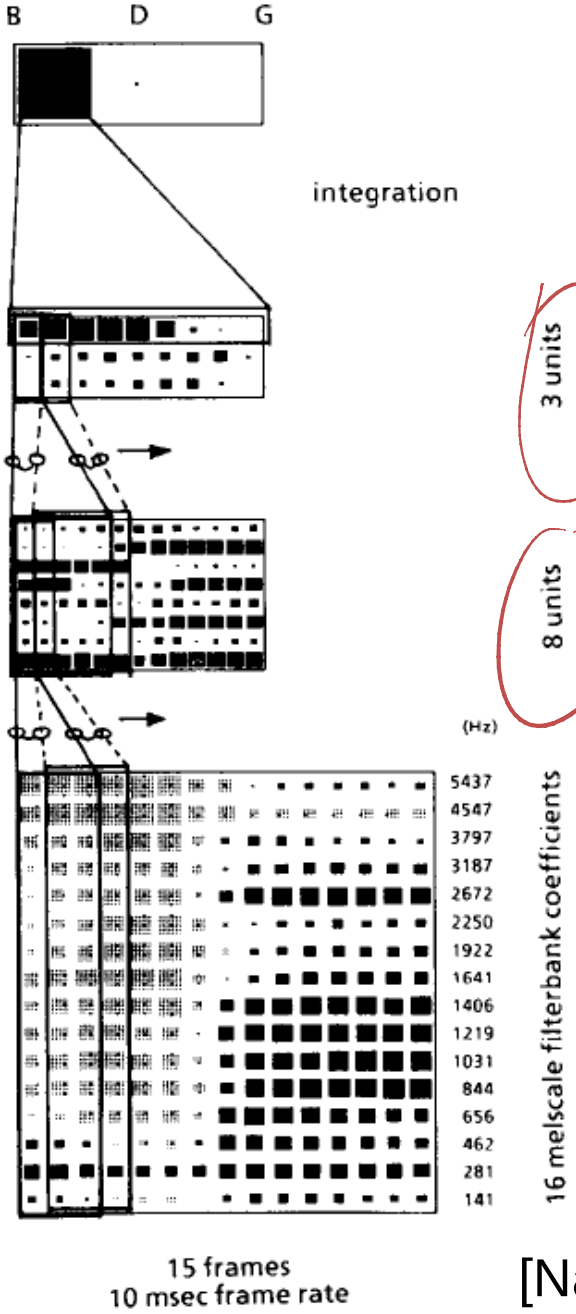
Capsules Draw

Convnets in Torch

Xavier Glorot

```
model = nn.Sequential()
model:add(nn.Reshape(1,32,32))
-- layer 1:
model:add(nn.SpatialConvolution(1, 16, 5, 5))
model:add(nn.Tanh())
model:add(nn.SpatialMaxPooling(2, 2, 2, 2))
-- layer 2:
model:add(nn.SpatialConvolution(16, 128, 5, 5))
model:add(nn.Tanh())
model:add(nn.SpatialMaxPooling(2, 2, 2, 2))
-- layer 3, a simple 2-layer neural net:
model:add(nn.Reshape(128*5*5))
model:add(nn.Linear(128*5*5, 200))
model:add(nn.Tanh())
model:add(nn.Linear(200,10))
model:add(nn.LogSoftMax())
```

ConvNets for Language



Leon Bottou

Phoneme Recognition Using Time-Delay Neural Networks

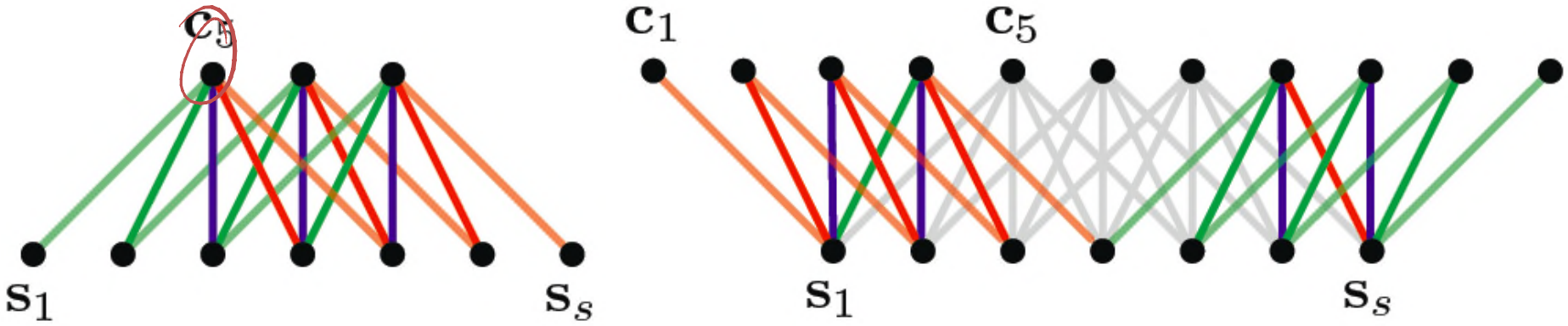
ALEXANDER WAIBEL, MEMBER, IEEE, TOSHIYUKI HANAZAWA, GEOFFREY HINTON, KIYOHIRO SHIKANO, MEMBER, IEEE, AND KEVIN J. LANG

NLP almost from scratch

→ [Ronan Collobert, Jason Weston, 2008]

[Nal Kalchbrenner, Ed Grefenstette, Phil Blunsom, 2014]

Sentence ConvNets



$$\mathbf{c}_j = \mathbf{m}^\intercal \mathbf{s}_{j-m+1:j}$$

$\mathbf{m}$ is the *filter* of the convolution

$$\mathbf{s} = \begin{bmatrix} | & | & | \\ \mathbf{w}_1 & \cdots & \mathbf{w}_s \\ | & | & | \end{bmatrix}$$

$$\mathbf{w}_i \in \mathbb{R}^d$$

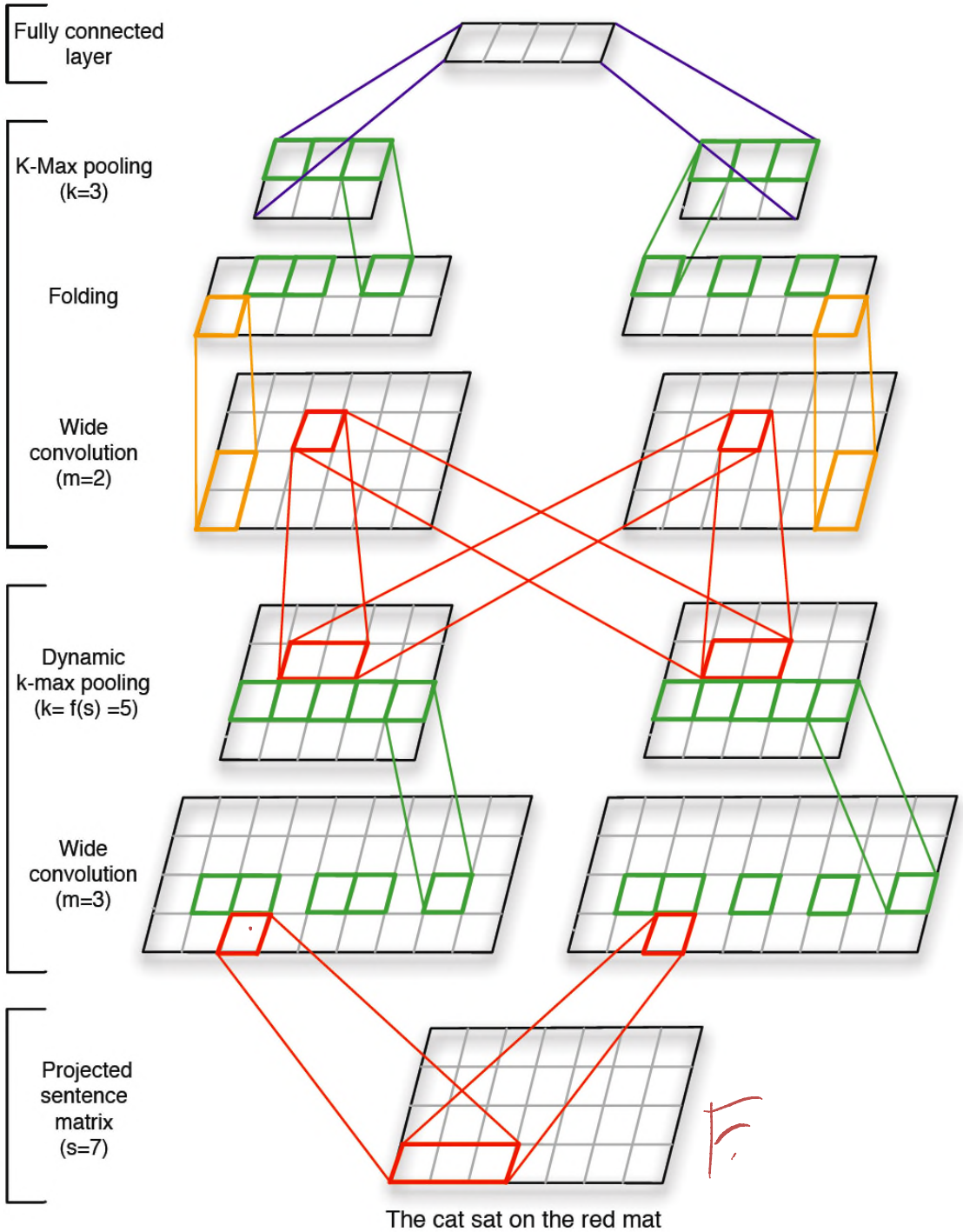
$$W_T \begin{bmatrix} w_1 & \cdots & w_s \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

← cat

← one hot encoding

[Kalchbrenner, Grefenstette, Blunsom, 2014]

Sentence DynConvNet



[Kalchbrenner et al, 2014]

Document models (Misha Denil)