Department of Medical Physics and Biomedical Engineering
Centre for Medical Image Computing (CMIC)
Wellcome / EPSRC Centre for Interventional and Surgical Sciences (WEISS)

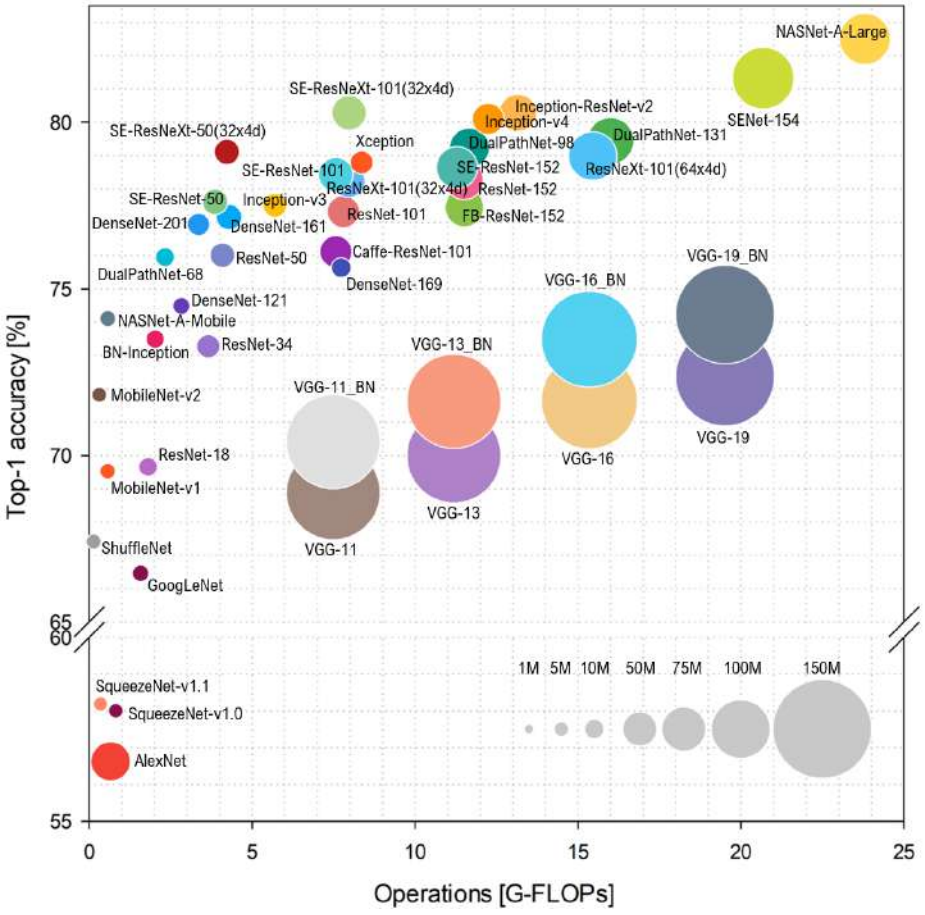
Deep Learning

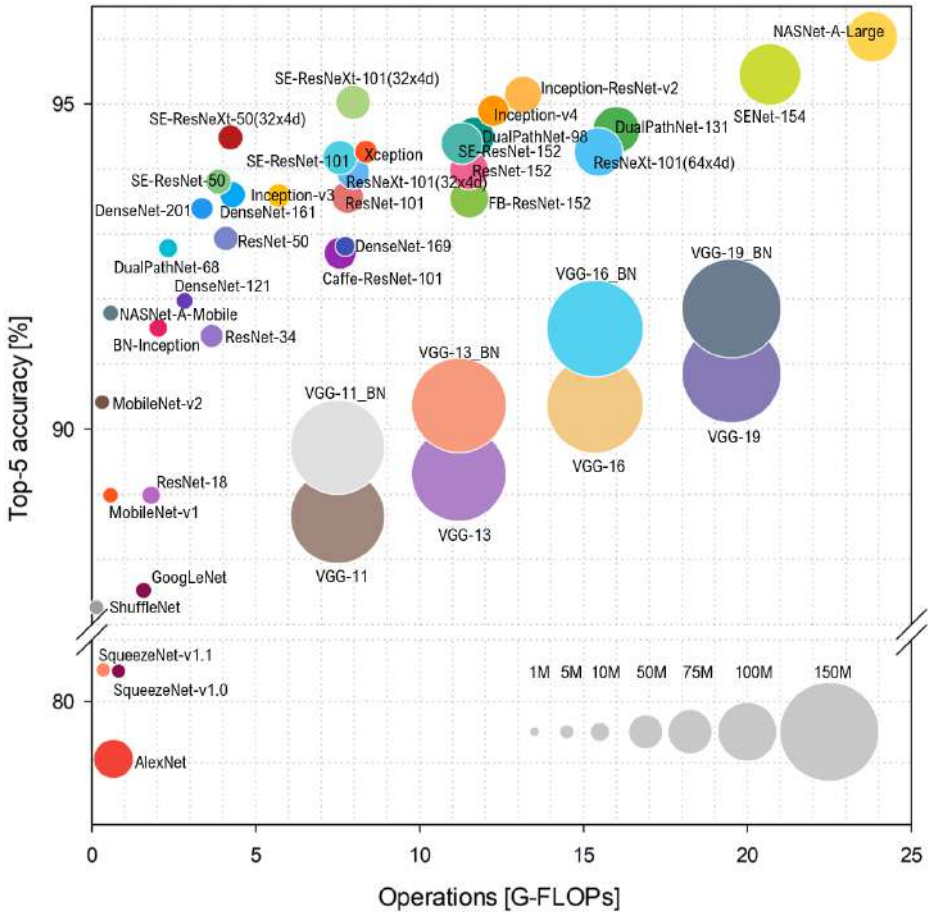
MPHY0041 Machine Learning in Medical Imaging

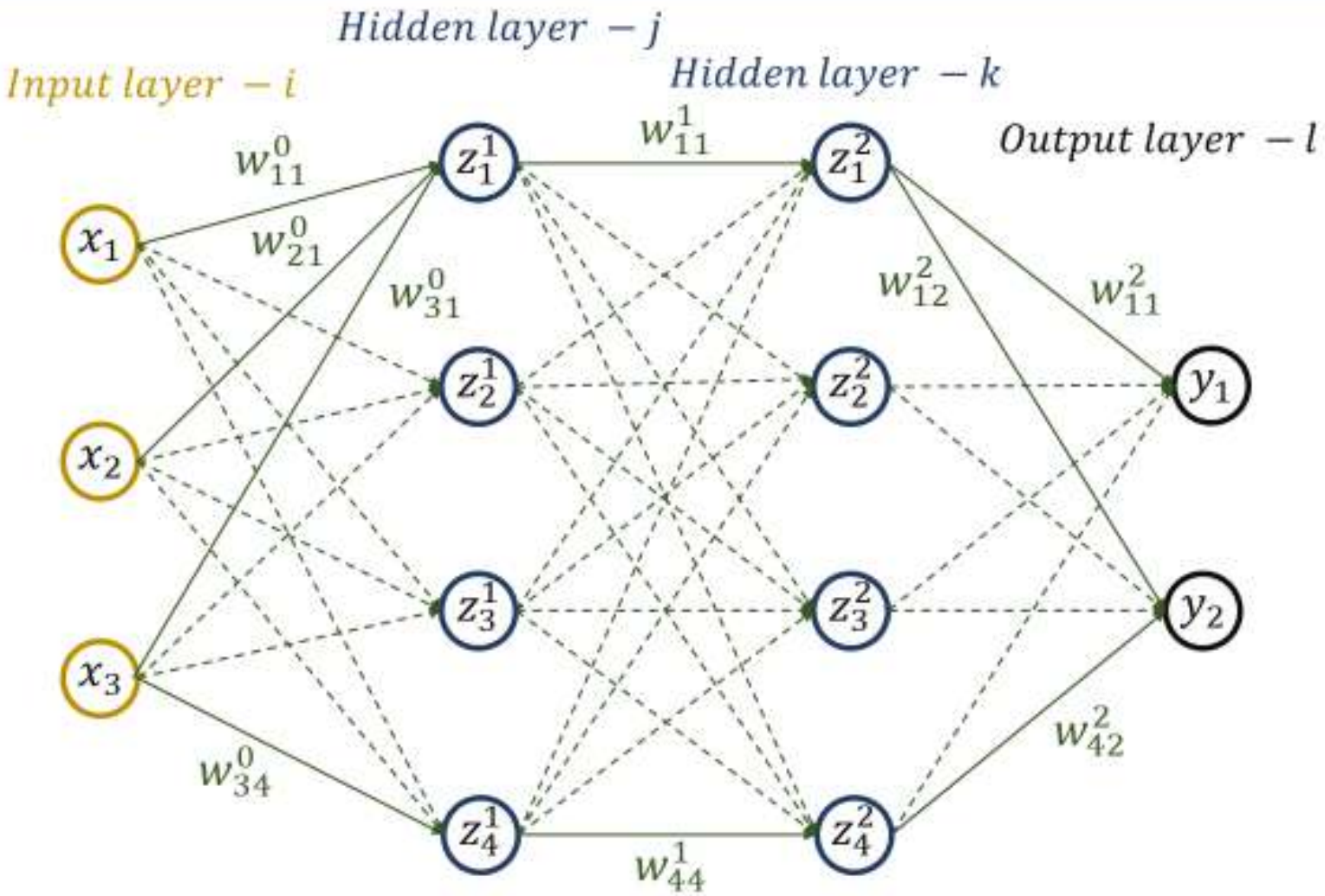
Yipeng Hu
yipeng.hu@ucl.ac.uk

Network Architecture | Width and Depth

Network Architecture | Width and Depth







Output layer $-\ l$

Input → H1: i → j

$$z_j^1 = g_j^1\left[\left(\mathbf{w}_j^0\right)^{\mathrm{T}}\mathbf{x} + b_j^1\right]$$

H1 → H2: j → k

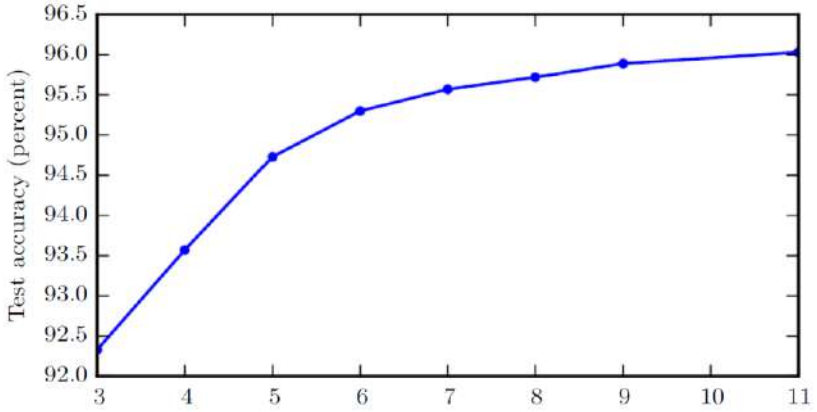
$$z_k^2 = g_k^1\left[\left(\mathbf{w}_k^1\right)^{\mathrm{T}}\mathbf{z}^1 + b_k^1\right]$$

H2 → Output: k → l

$$y_k = g_l^1\left[\left(\mathbf{w}_l^2\right)^{\mathrm{T}}\mathbf{z}^2 + b_l^2\right]$$

Universal Approximation Theorem

Any continuous functions on bounded and closed functions or between finite dimensional discrete space

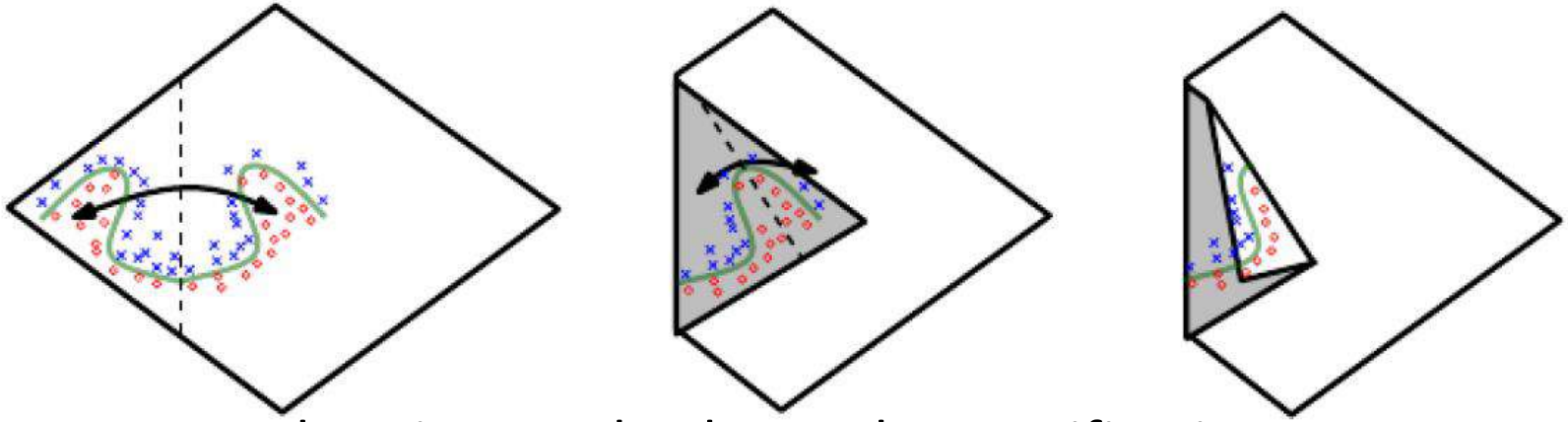


Width and Depth

Efficiency

Hierarchical representation learning

Empirical results

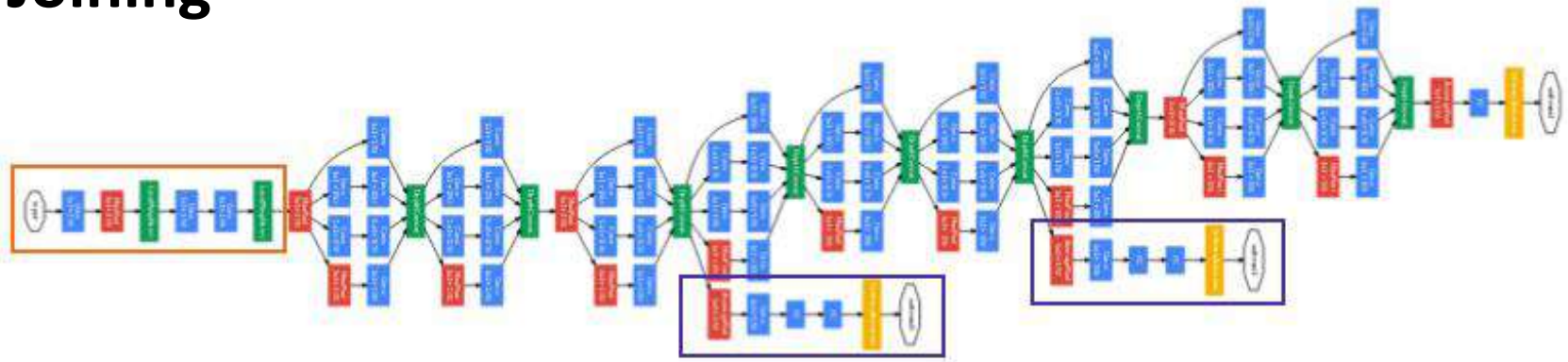


An example using an absolute value rectification

Network Architecture | Branching, Joining and Skipping

Branching and Joining

"Inception"



Multi-stream

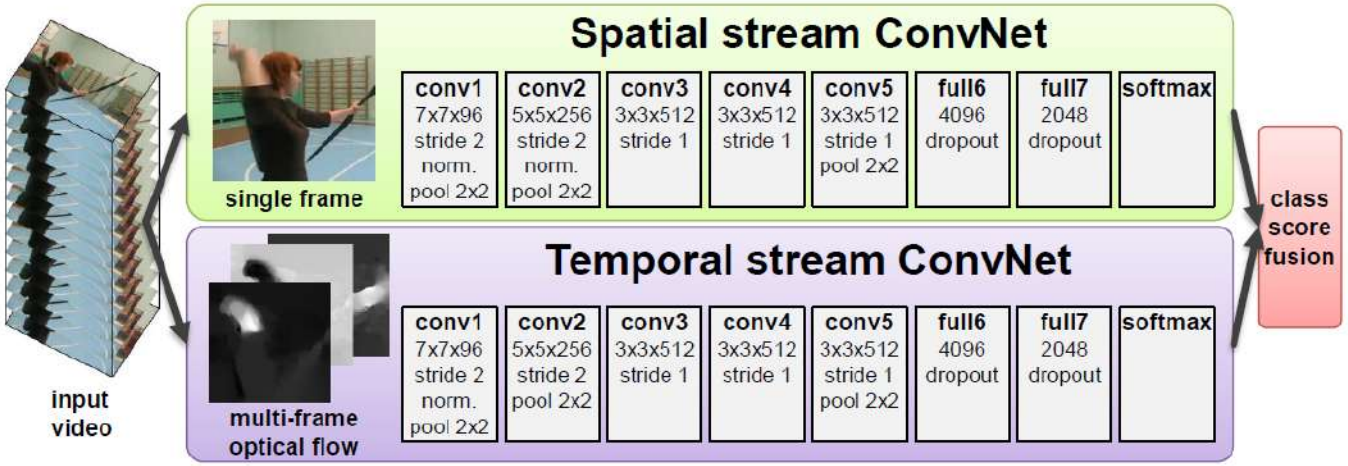
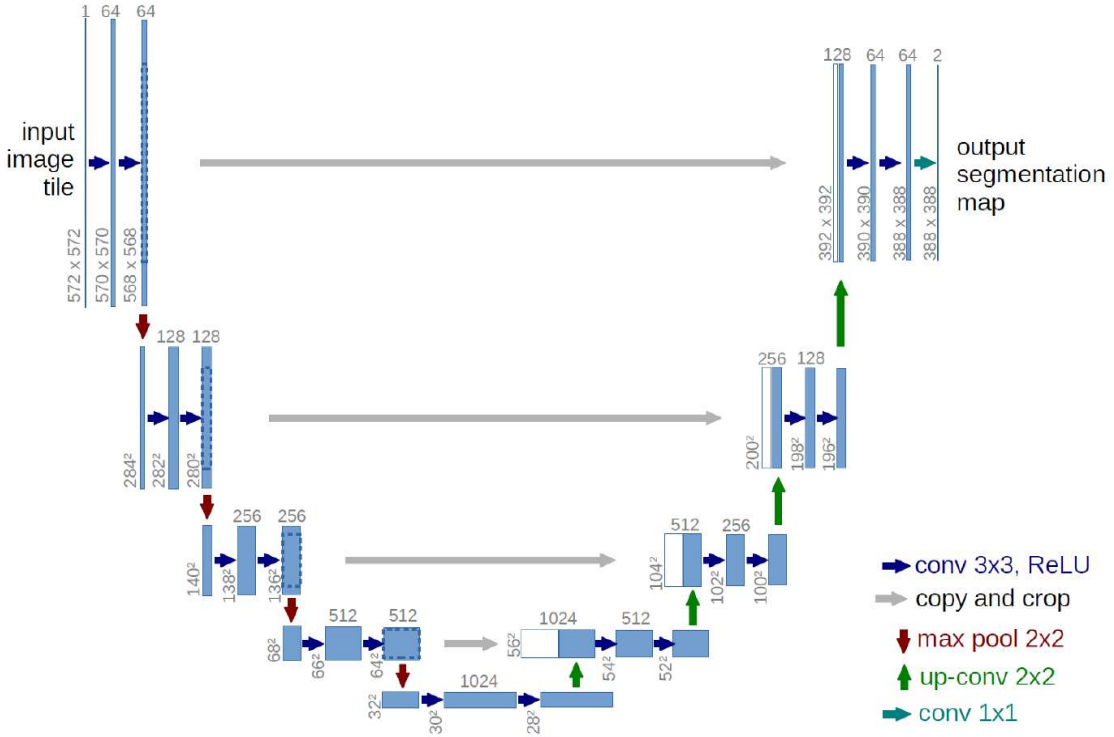
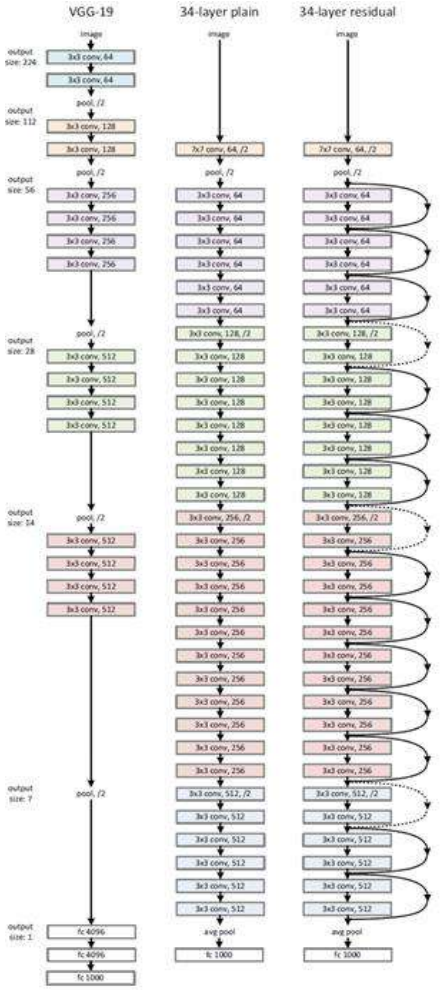


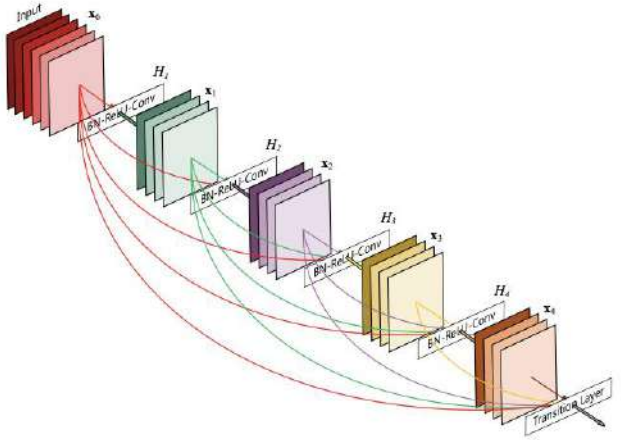
Figure 1: **Two-stream architecture for video classification.**

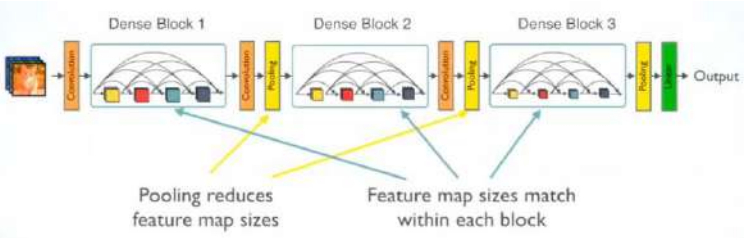
Multi-task learning*

Skipping (shortcuts, skip layers and residual connections)

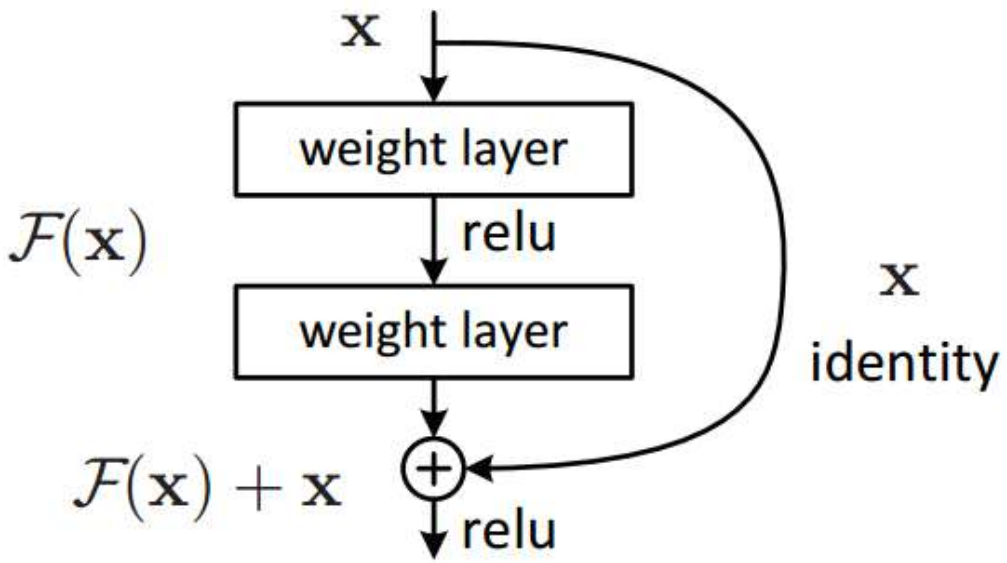








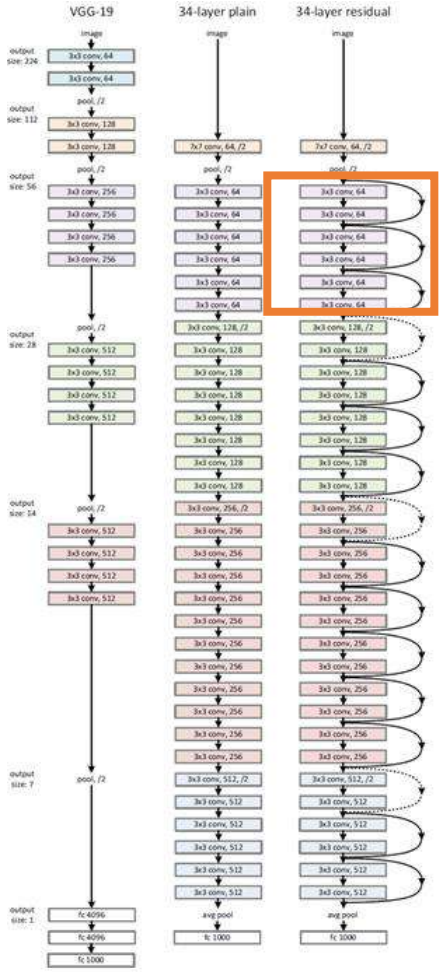
The ResNet

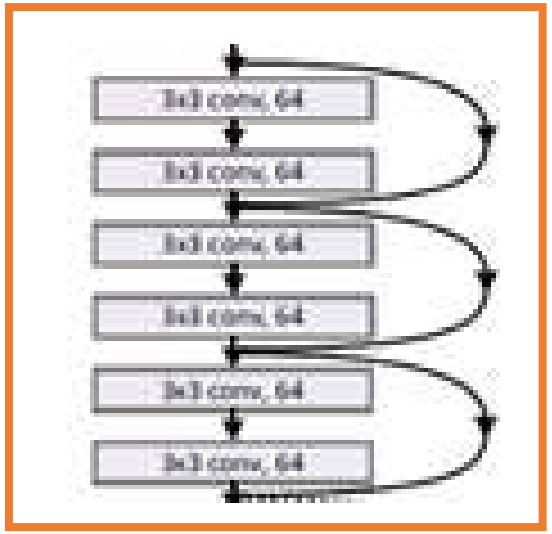


$\mathcal{F}(\mathbf{x})$

$\mathcal{F}(\mathbf{x}) + \mathbf{x}$

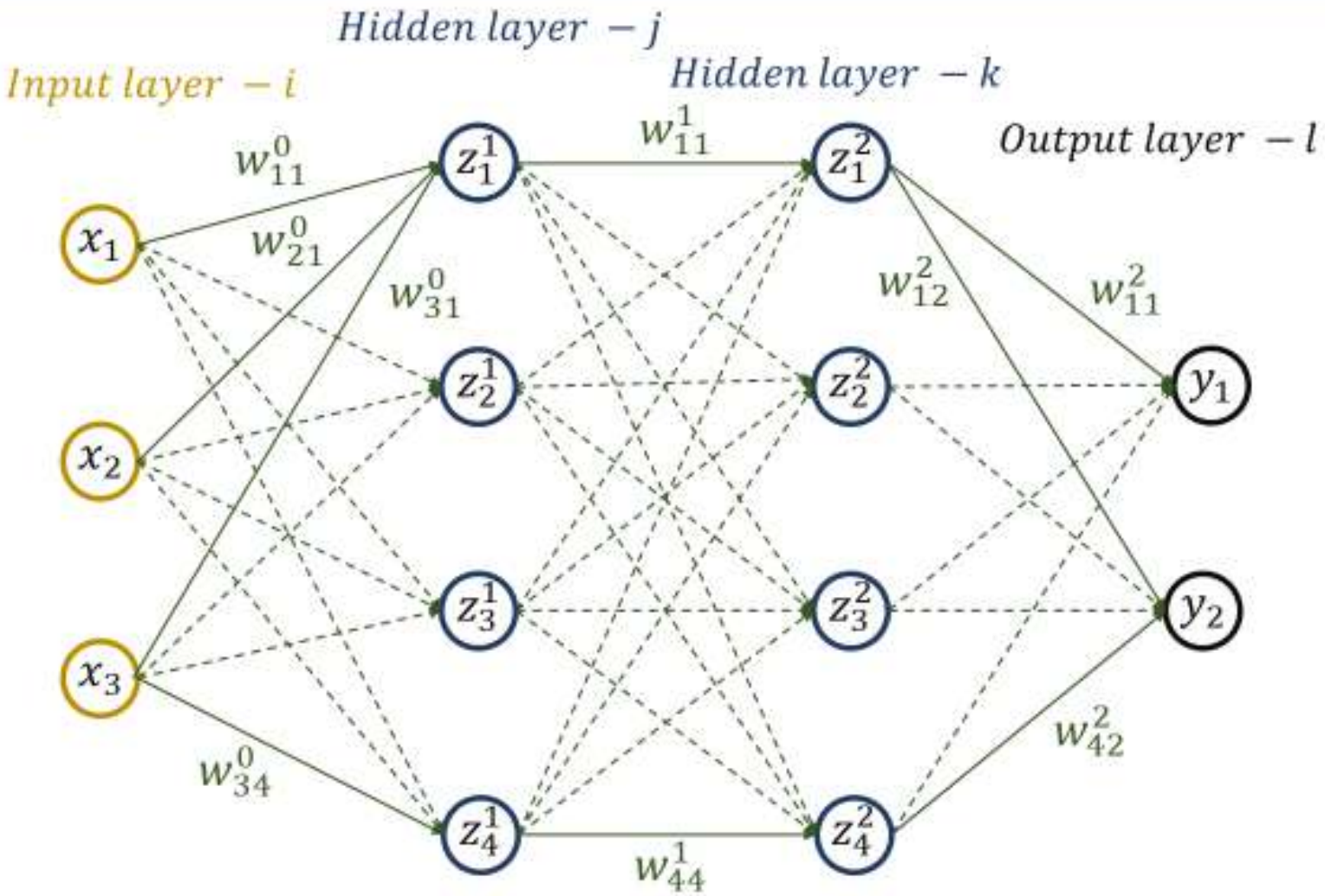
$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}.$$





Network Architecture | Convolutional Neural Networks

Neural Networks | Multilayer Perceptron



Output layer − l

Input → H1: i → j

$$z_j^1 = g_j^1\left[\left(\mathbf{w}_j^0\right)^{\mathrm{T}}\mathbf{x} + b_j^1\right]$$

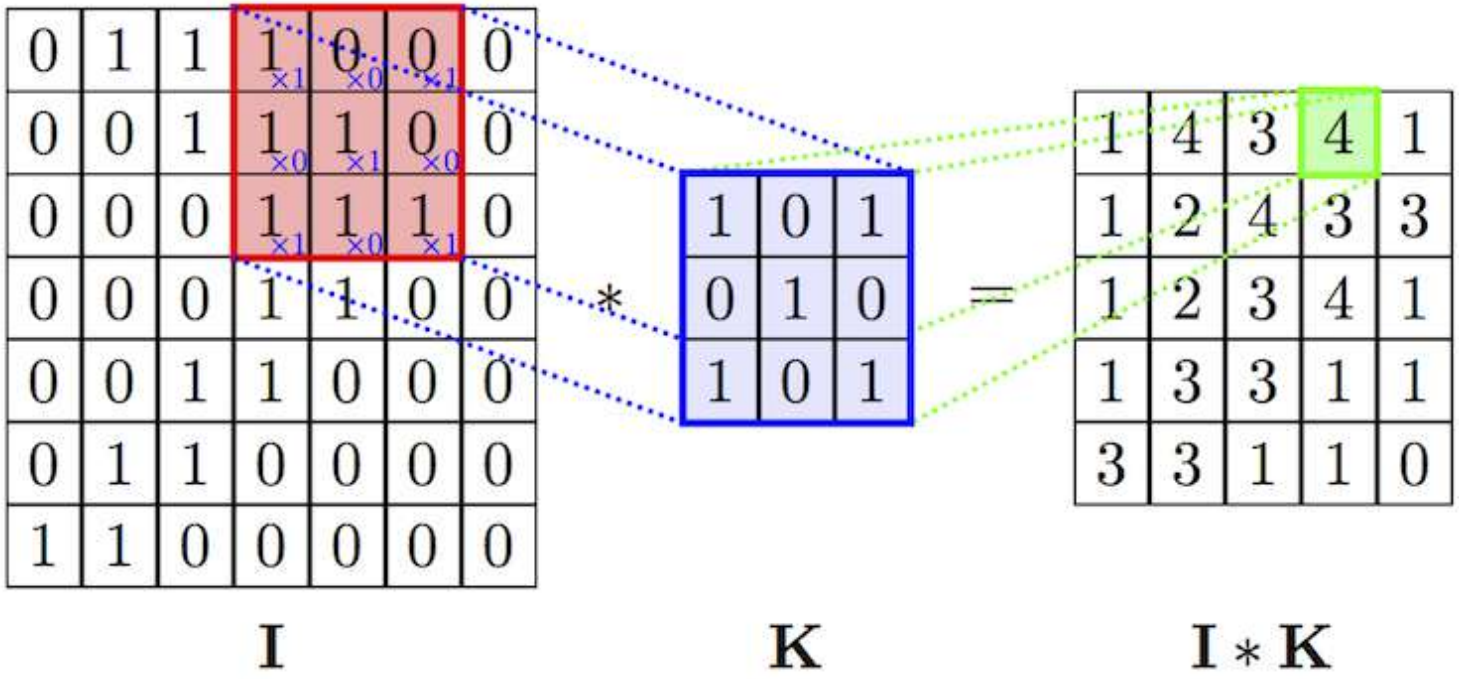
H1 → H2: j → k

$$z_k^2 = g_k^1\left[\left(\mathbf{w}_k^1\right)^{\mathrm{T}}\mathbf{z}^1 + b_k^1\right]$$

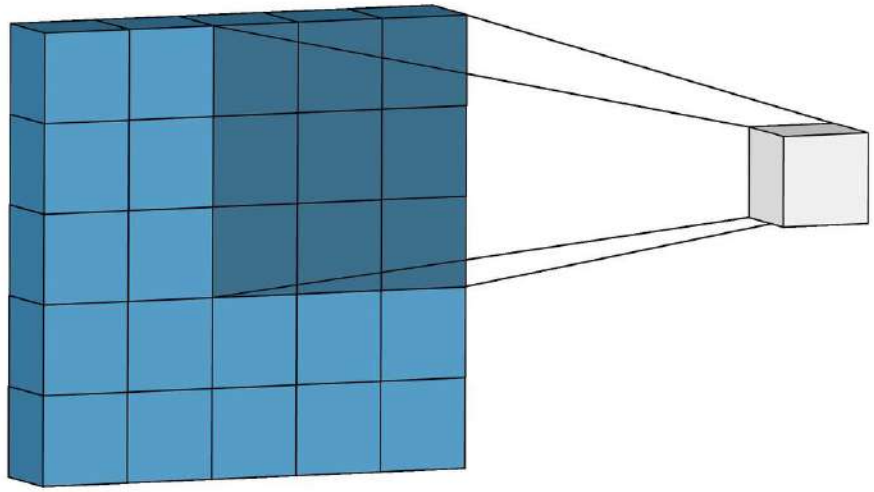
H2 → Output: k → l

$$y_k = g_l^1\left[\left(\mathbf{w}_l^2\right)^{\mathrm{T}}\mathbf{z}^2 + b_l^2\right]$$

Convolutional Neural Networks



Convolutional Layer



Convolutional Neural Networks

$$\begin{pmatrix} x1 & x2 & x3 \\ x4 & x5 & x6 \\ x7 & x8 & x9 \end{pmatrix} \quad \begin{pmatrix} k1 & k2 \\ k3 & k4 \end{pmatrix} = \begin{pmatrix} k1\,x1 + k2\,x2 + k3\,x4 + k4\,x5 \\ k1\,x2 + k2\,x3 + k3\,x5 + k4\,x6 \\ k1\,x4 + k2\,x5 + k3\,x7 + k4\,x8 \\ k1\,x5 + k2\,x6 + k3\,x8 + k4\,x9 \end{pmatrix}$$

$$= \begin{pmatrix} k1 & k2 & 0 & k3 & k4 & 0 & 0 & 0 & 0 \\ 0 & k1 & k2 & 0 & k3 & k4 & 0 & 0 & 0 \\ 0 & 0 & 0 & k1 & k2 & 0 & k3 & k4 & 0 \\ 0 & 0 & 0 & 0 & k1 & k2 & 0 & k3 & k4 \end{pmatrix} \cdot \begin{pmatrix} x1 \\ x2 \\ x3 \\ x4 \\ x5 \\ x6 \\ x7 \\ x8 \\ x9 \end{pmatrix}$$

Convolutional Layer

Input → H1: i → j

$$z_j^1 = \sigma_j^1 \left[\left(\mathbf{w}_j^0 \right)^{\mathrm{T}} \mathbf{x} + b_j^1 \right]$$

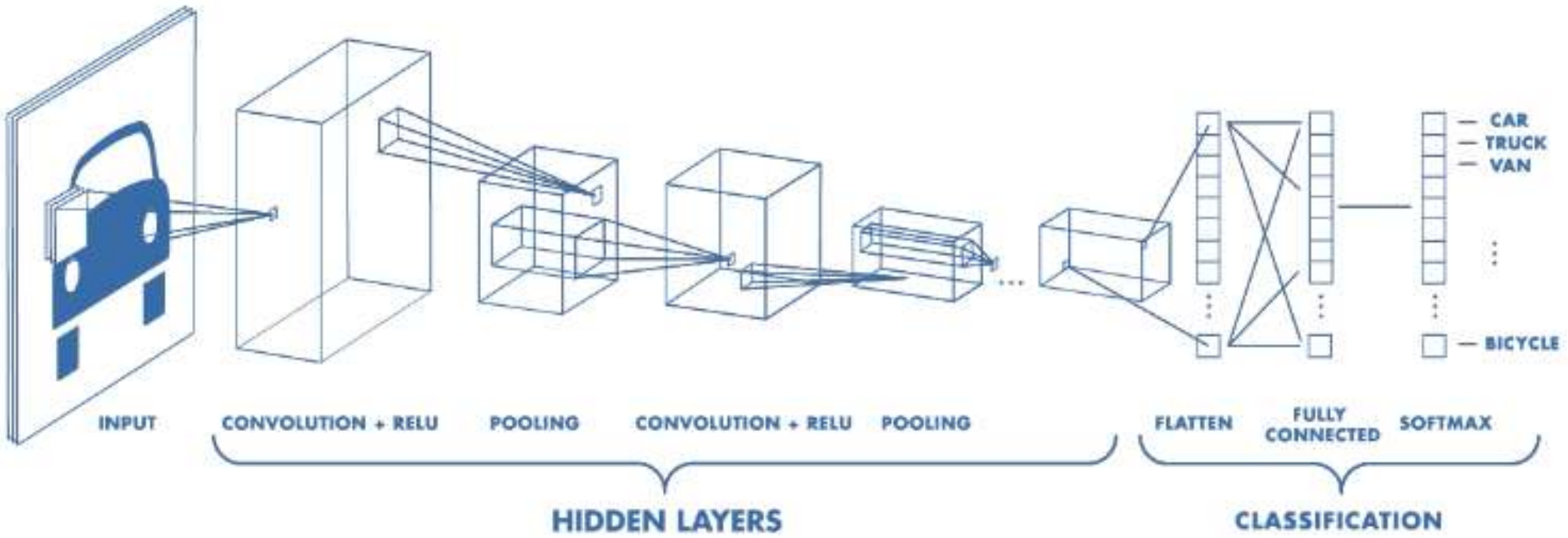
H1 → H2: j → k

$$z_k^2 = \sigma_k^1 \left[\left(\mathbf{w}_k^1 \right)^{\mathrm{T}} \mathbf{z}^1 + b_k^1 \right]$$

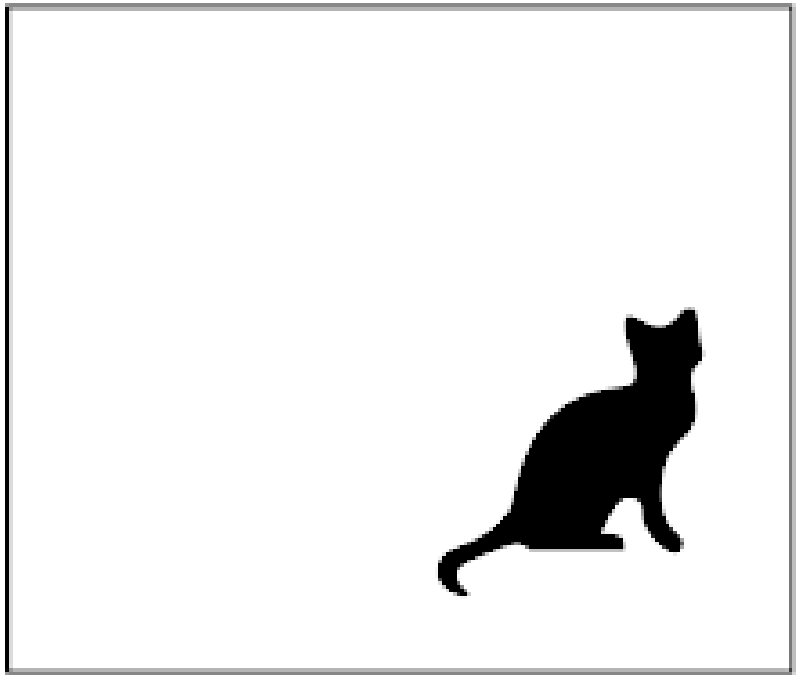
H2 → Output: k → l

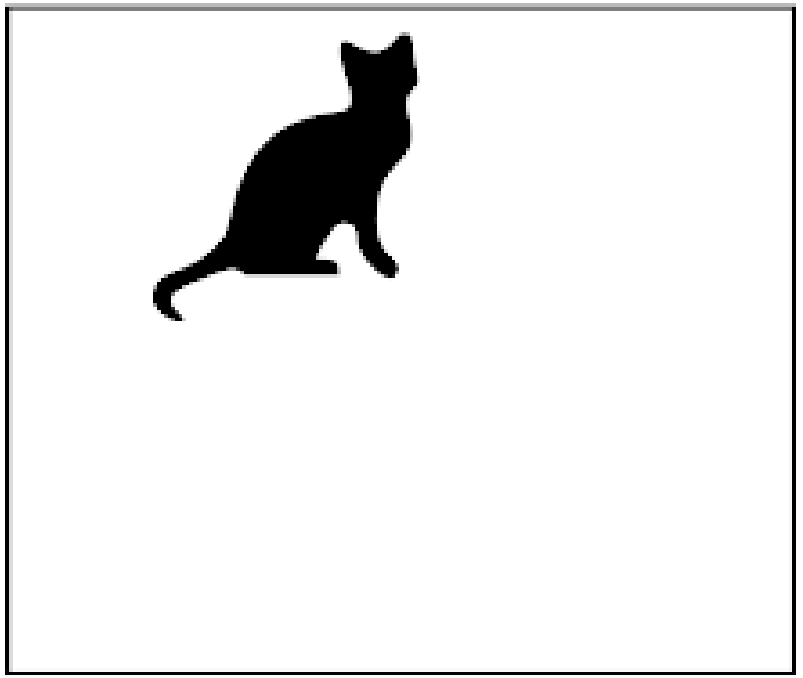
$$y_k = \sigma_l^1 \left[\left(\mathbf{w}_l^2 \right)^{\mathrm{T}} \mathbf{z}^2 + b_l^2 \right]$$

Convolutional Layers + Fully Connected Layers

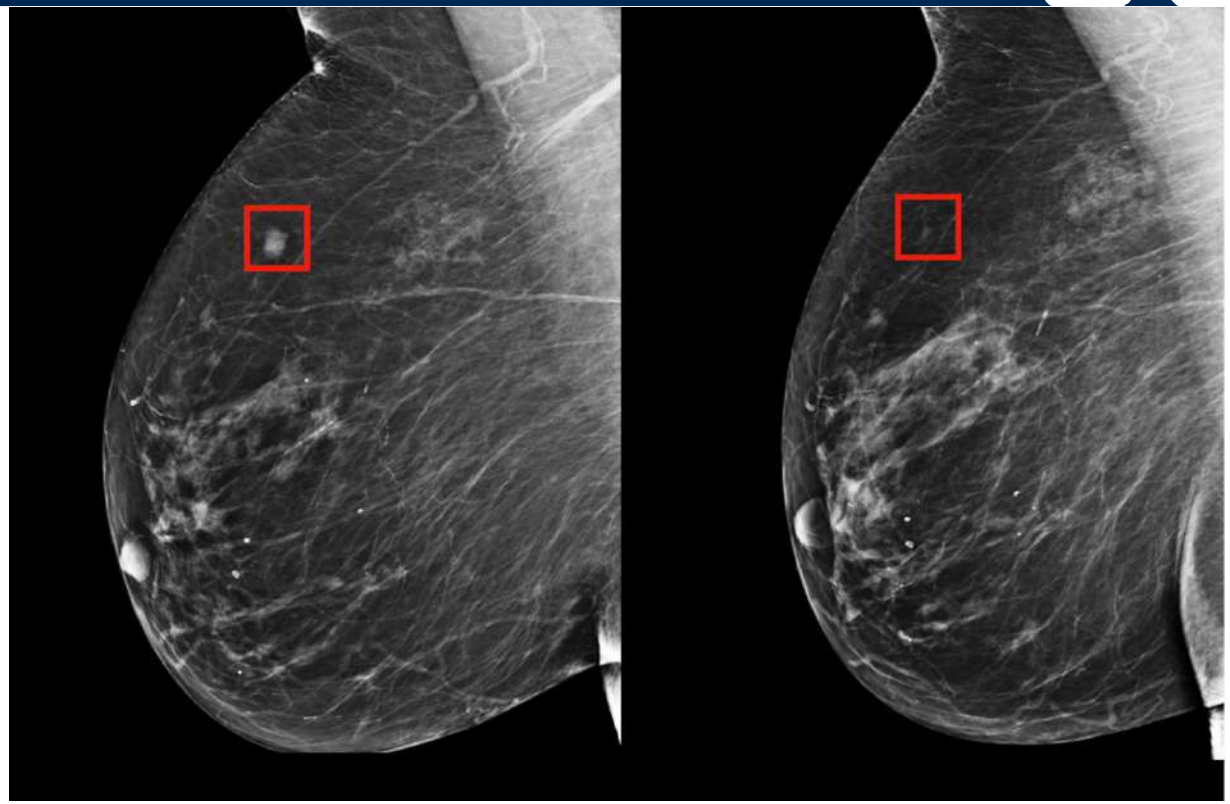


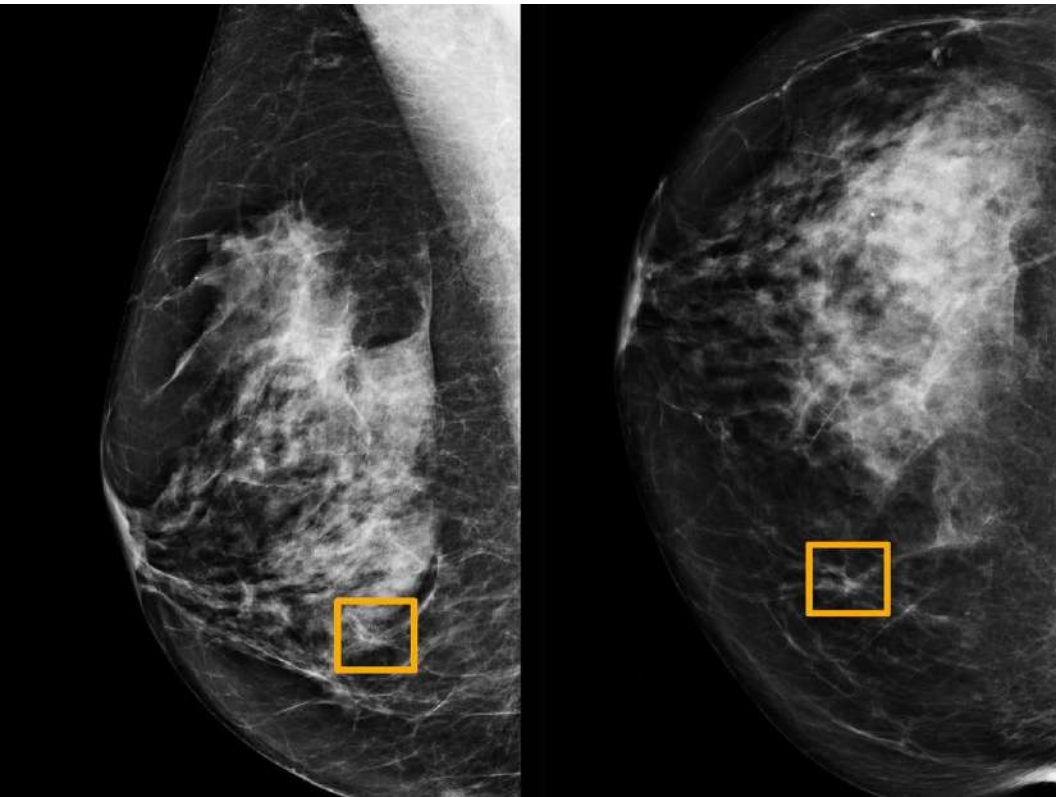
$Convolutional\ Layers + Fully\ Connected\ Layers$





Network Architecture | Convolutional Neural Networks

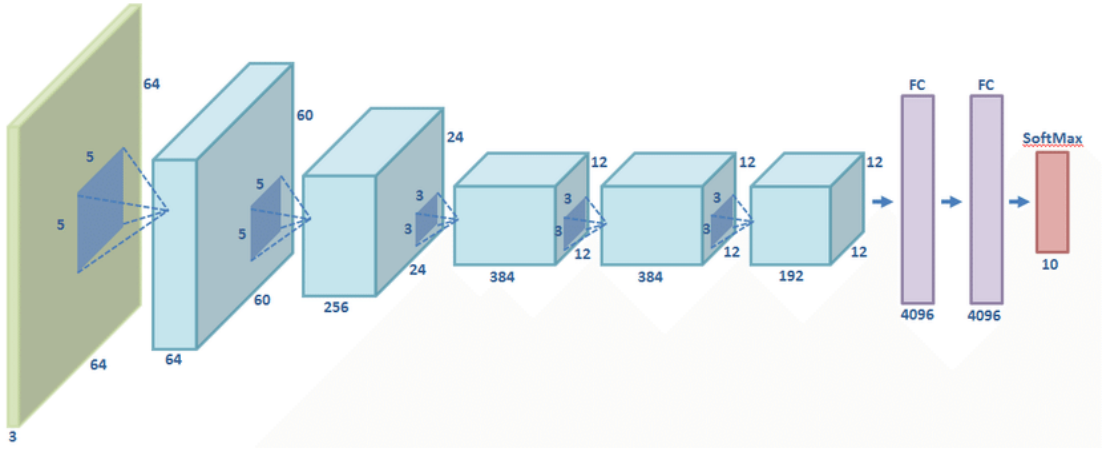




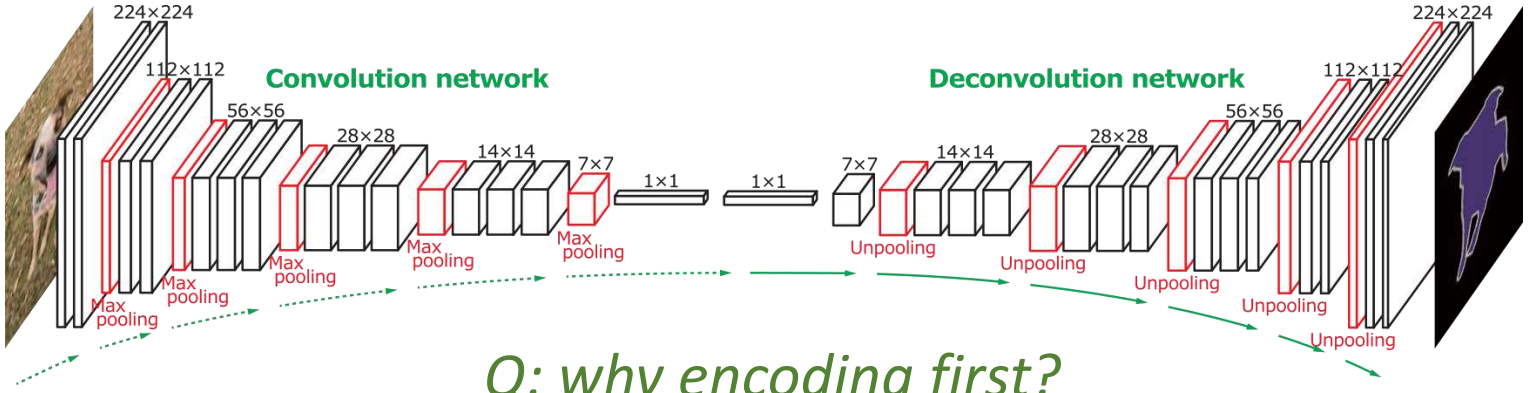
Network Architecture | Sampling

Down-sampling and up-sampling (Encoding and Decoding)

AlexNet / VGG
Fully convolutional
Global and local pooling

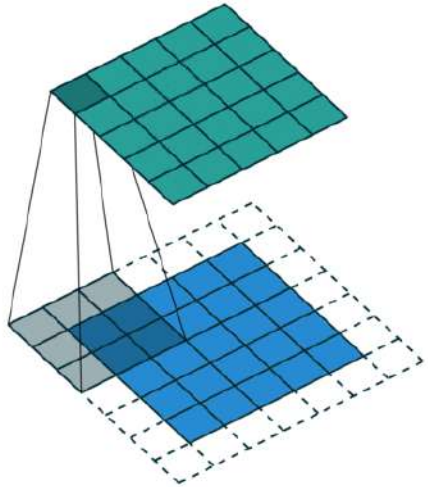


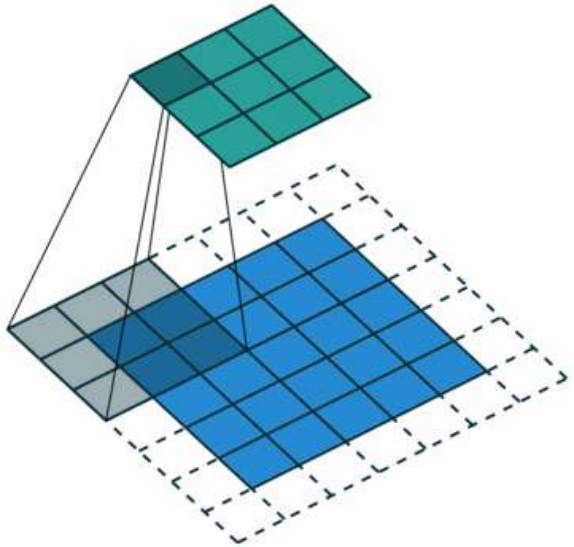
Dense prediction

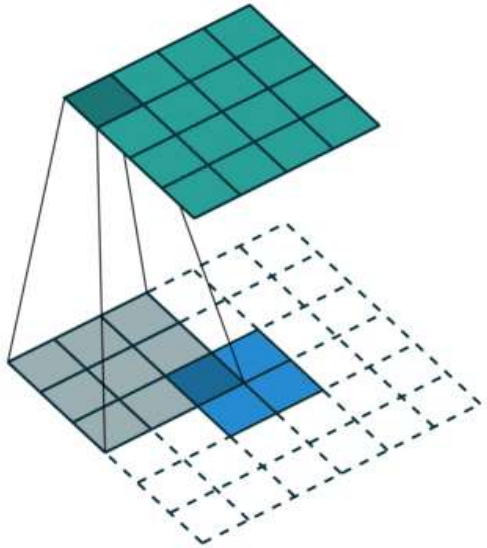


Q: why encoding first?

Down-sampling and up-sampling 2D/3D feature maps



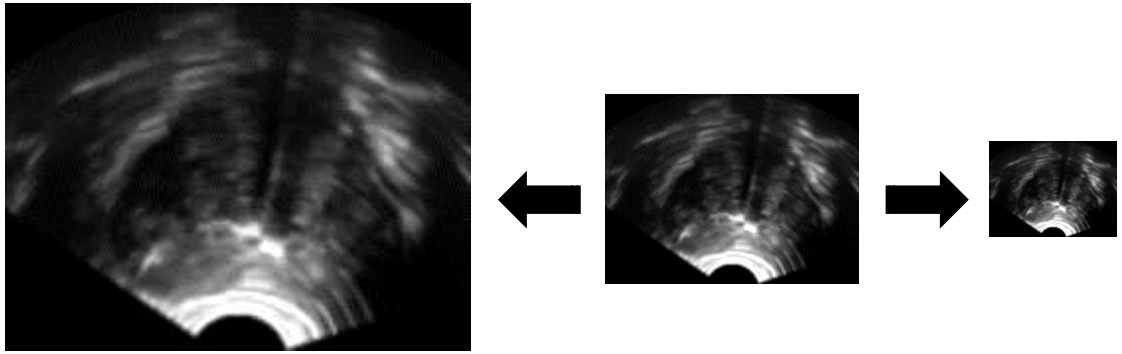




Convolution and pooling with strides

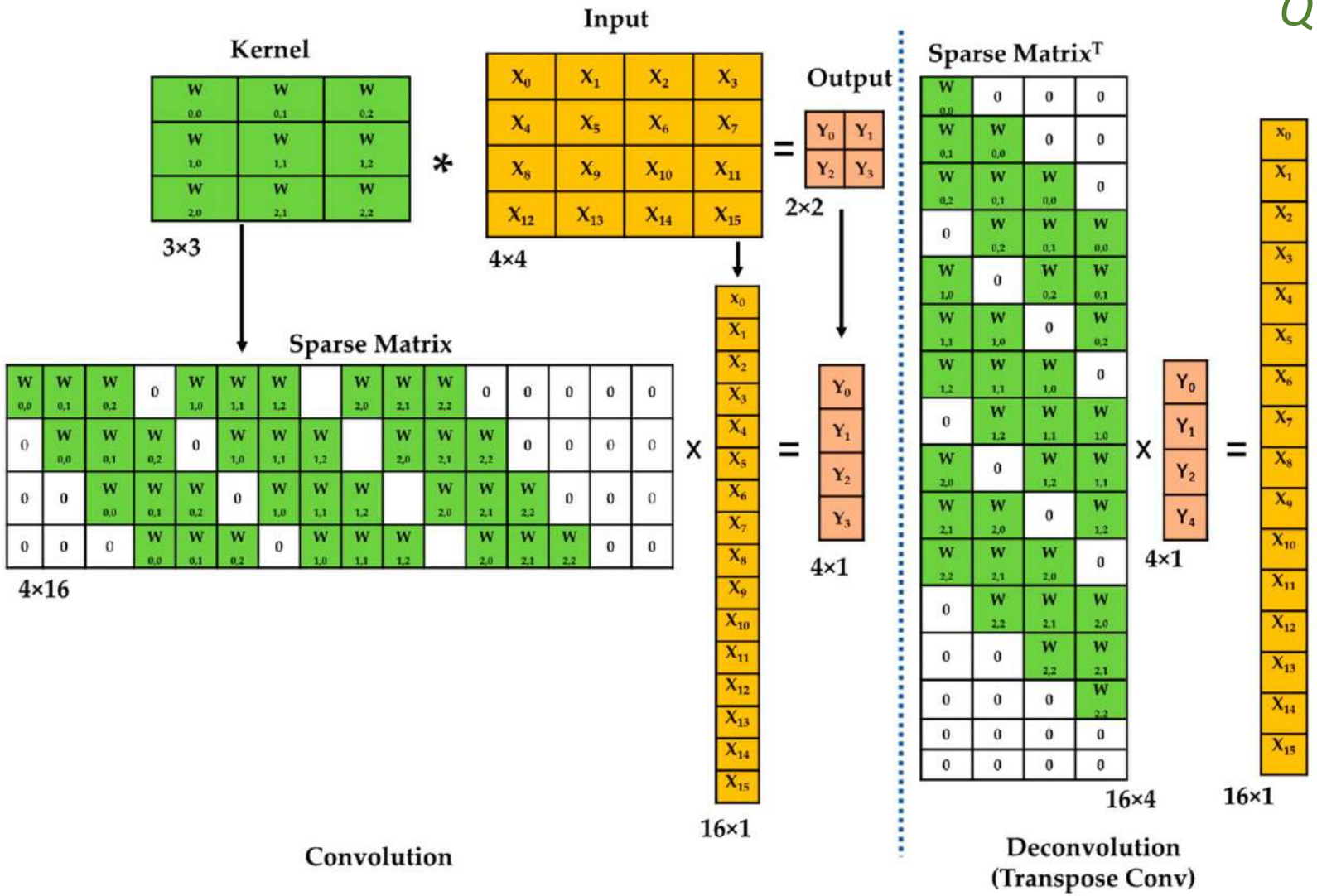
Transpose convolution and “un-pooling”

Linear sampling



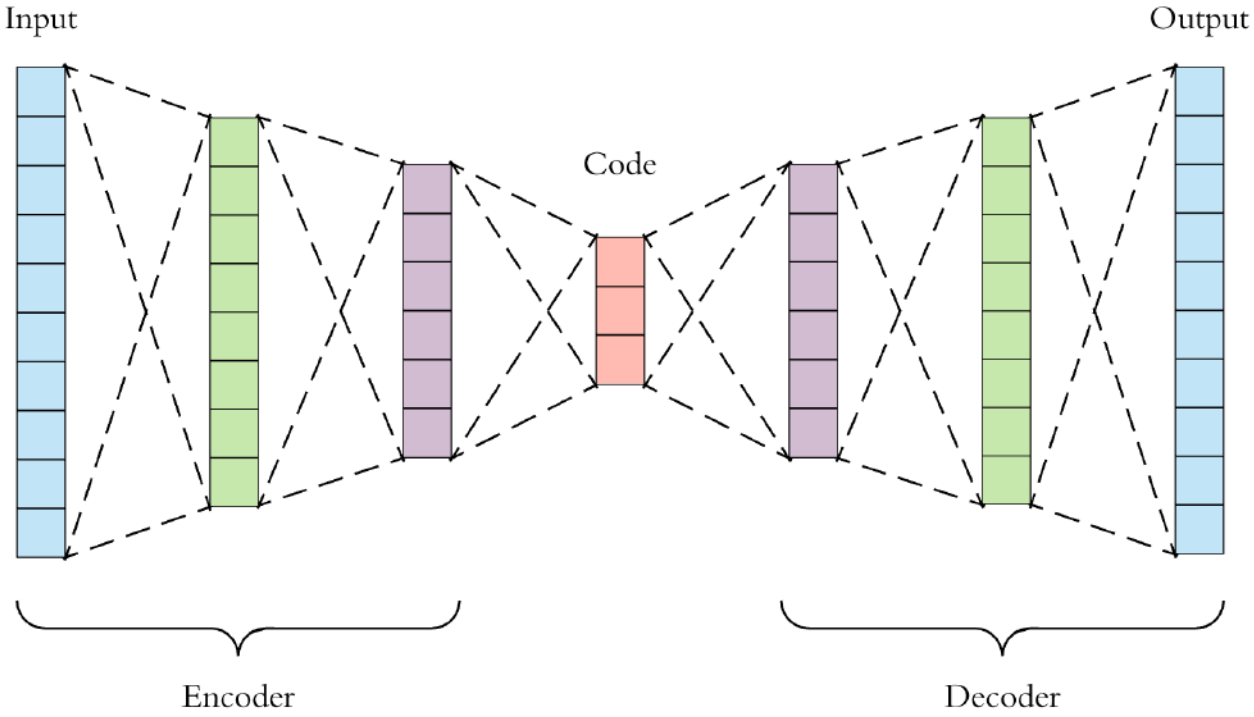
Transpose convolution

Q: is transpose convolution necessary?

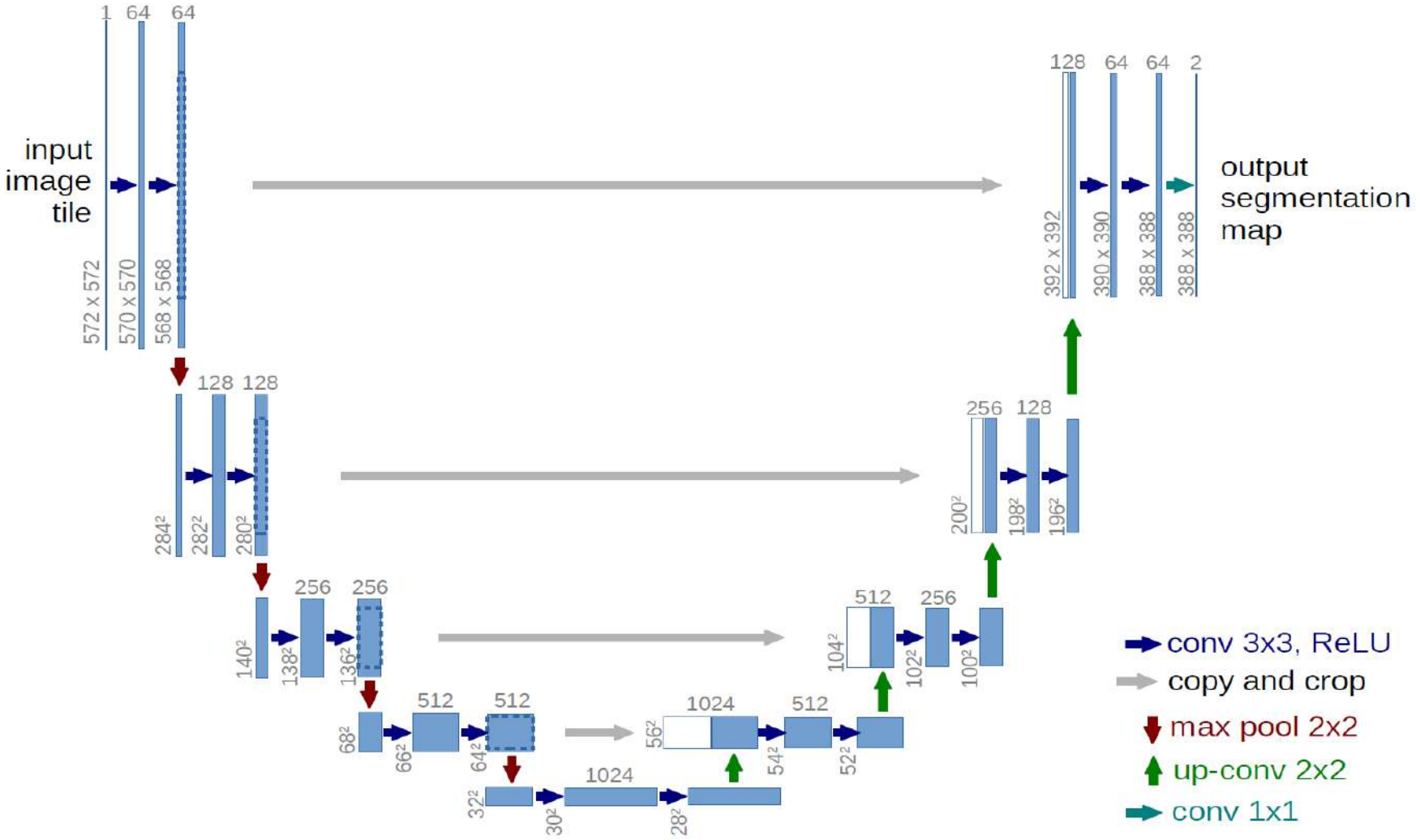


Network Architecture | Examples

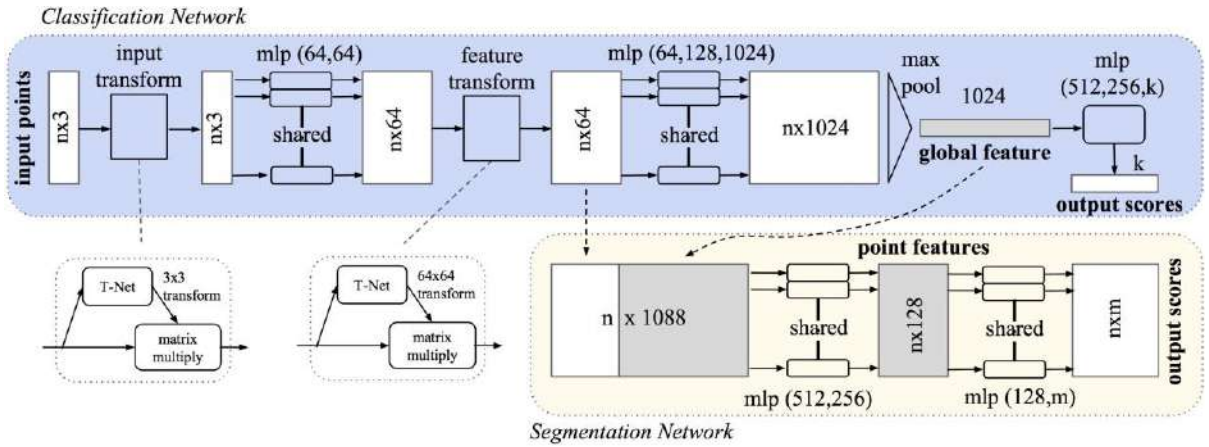
Autoencoder

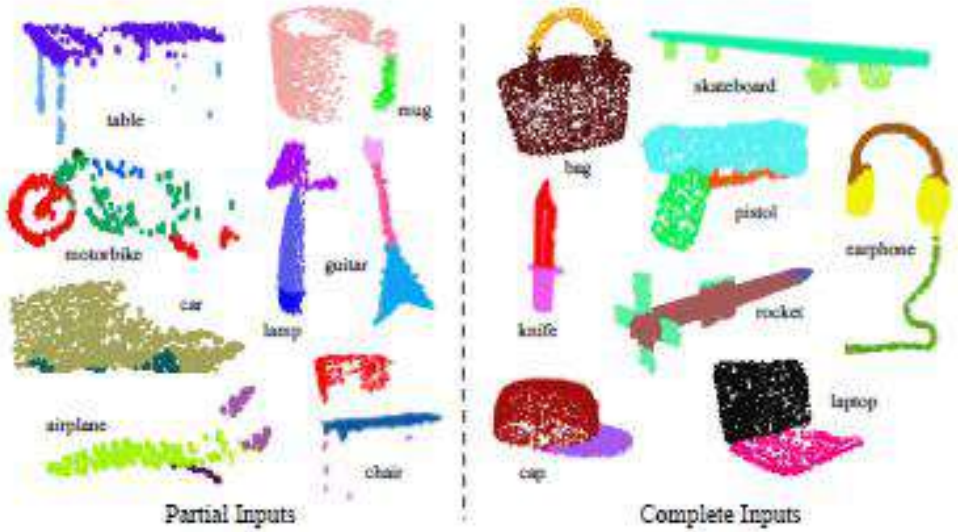


UNet


input image tile
572 x 572
570 x 570
568 x 568
1 64 64
128 128
284²
282²
280²
256 256
140²
138²
136²
512 512
68²
66²
64²
1024
32²
30²
28²
1024 512
56²
54²
52²
512 256
104²
102²
100²
256 128
200²
198²
196²
128 64 64 2
392 x 392
390 x 390
388 x 388
388 x 388
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

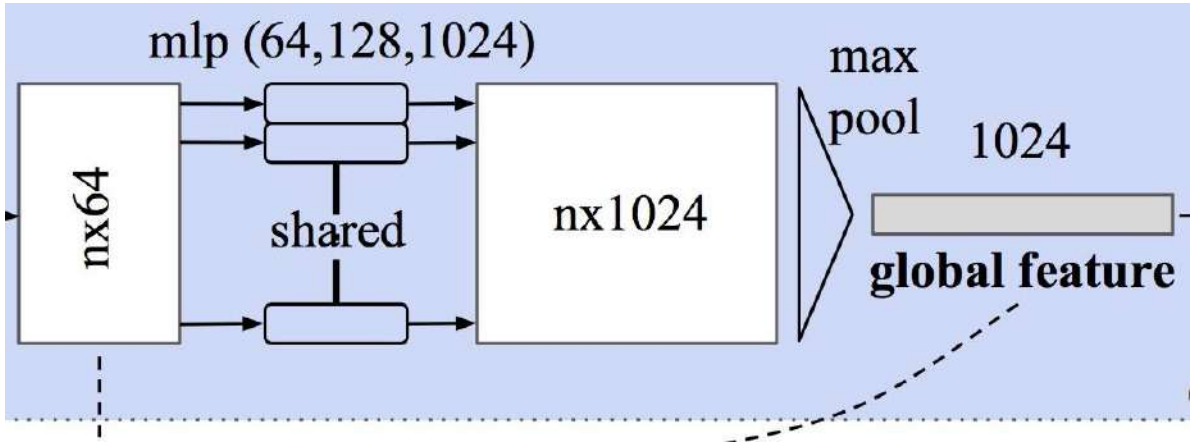
PointNet



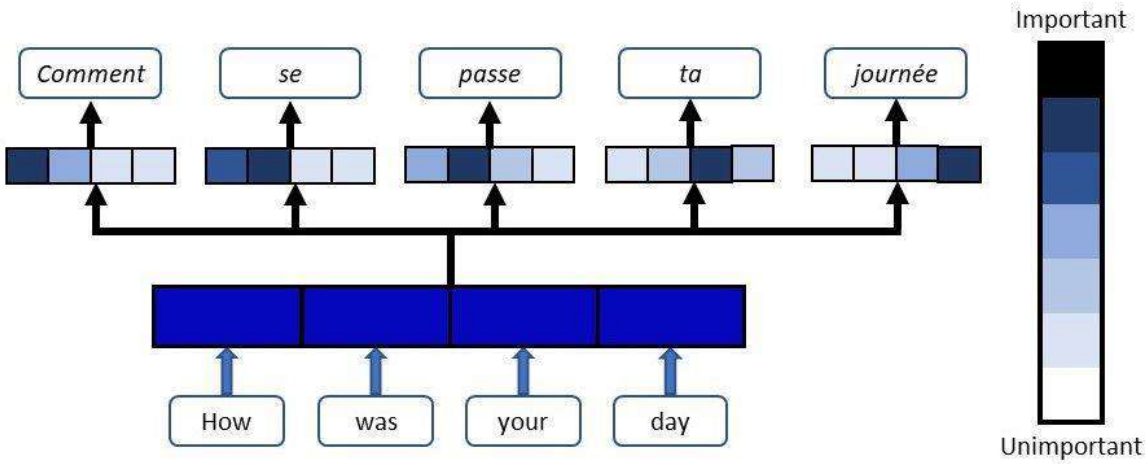


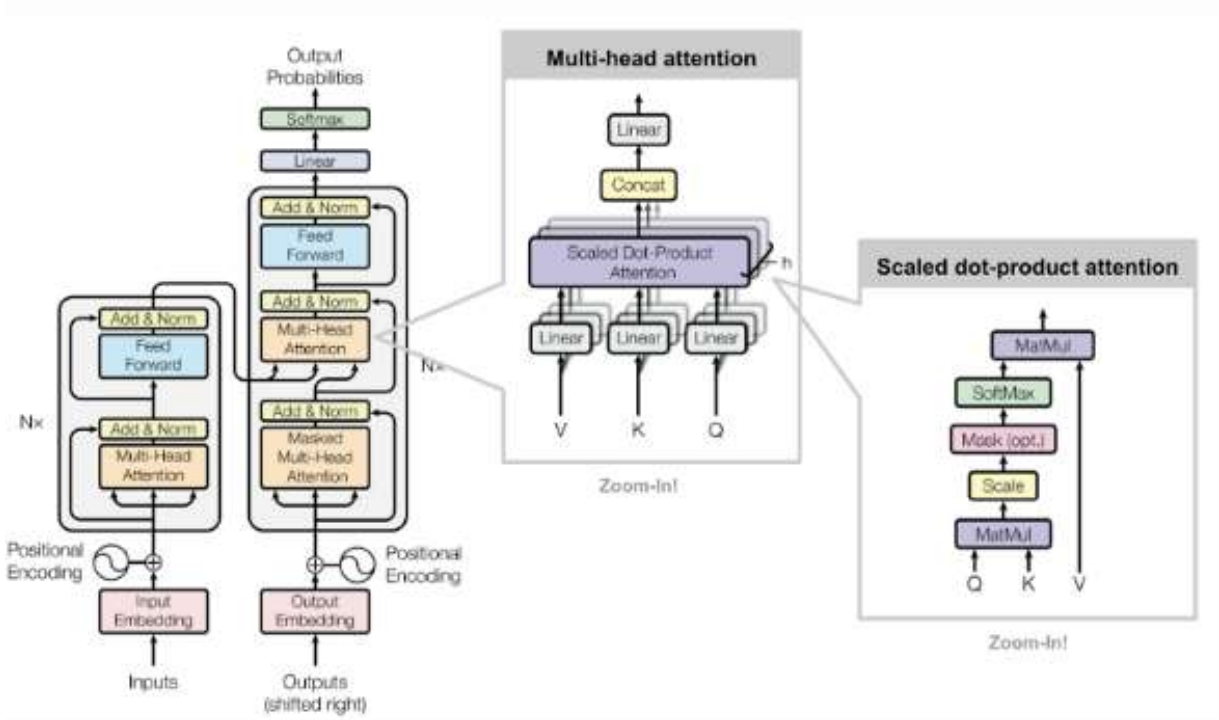
$$f(\{x_1, \ldots, x_n\}) \approx g(h(x_1), \ldots, h(x_n))$$

Permutation-Invariant Functions

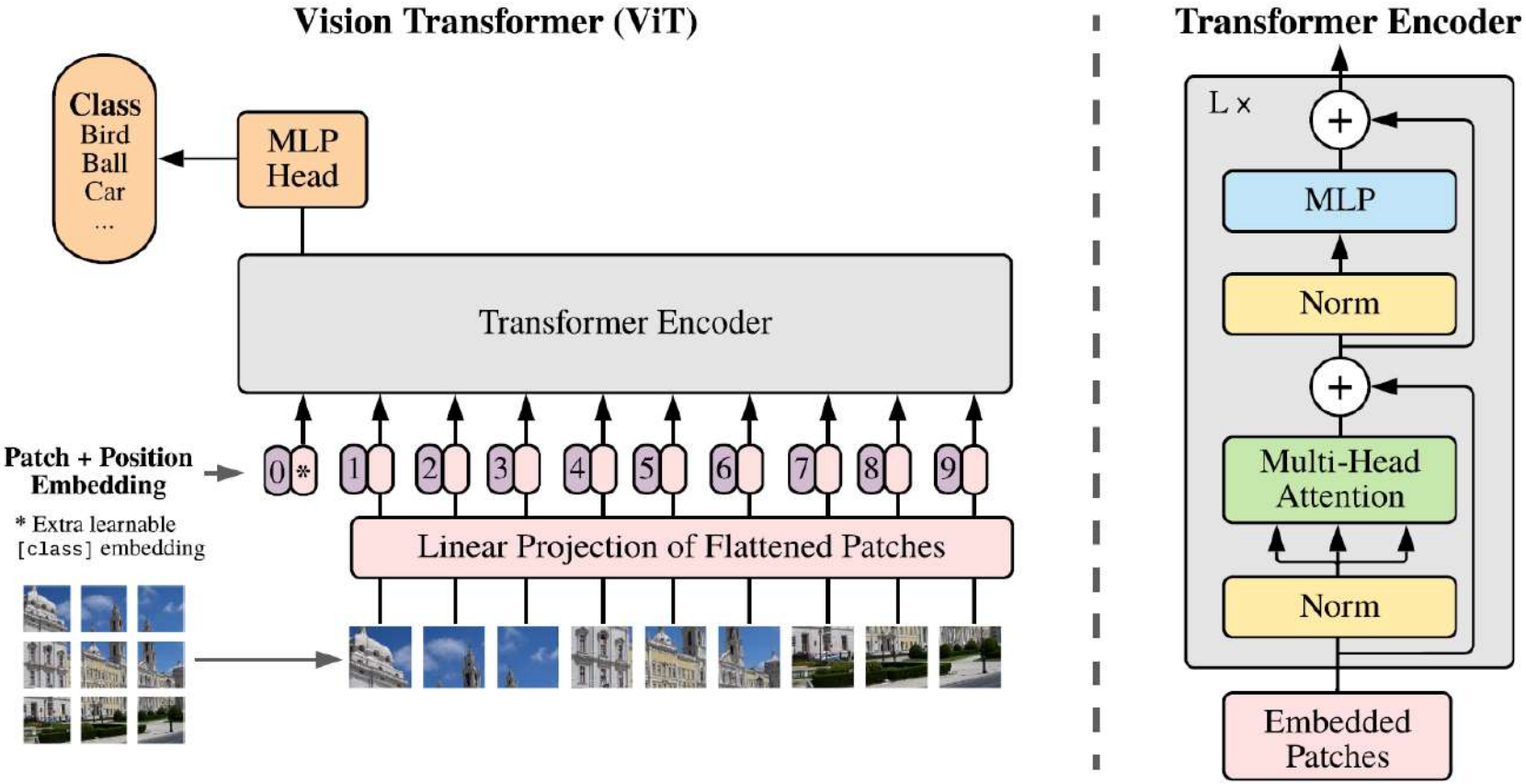


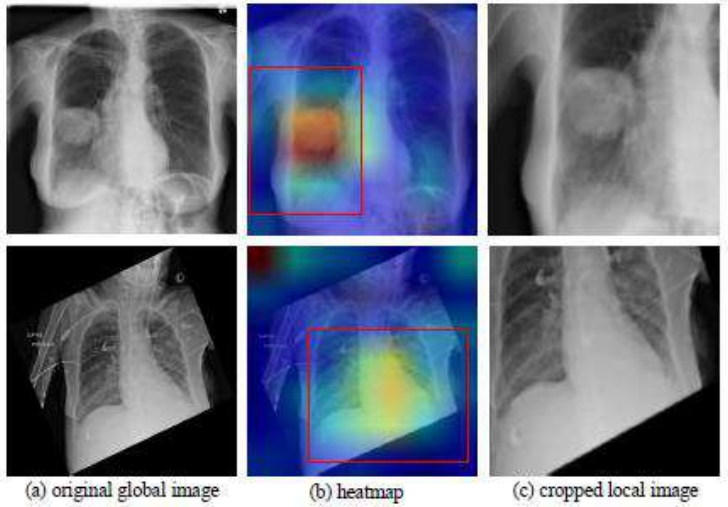
Attention Mechanism





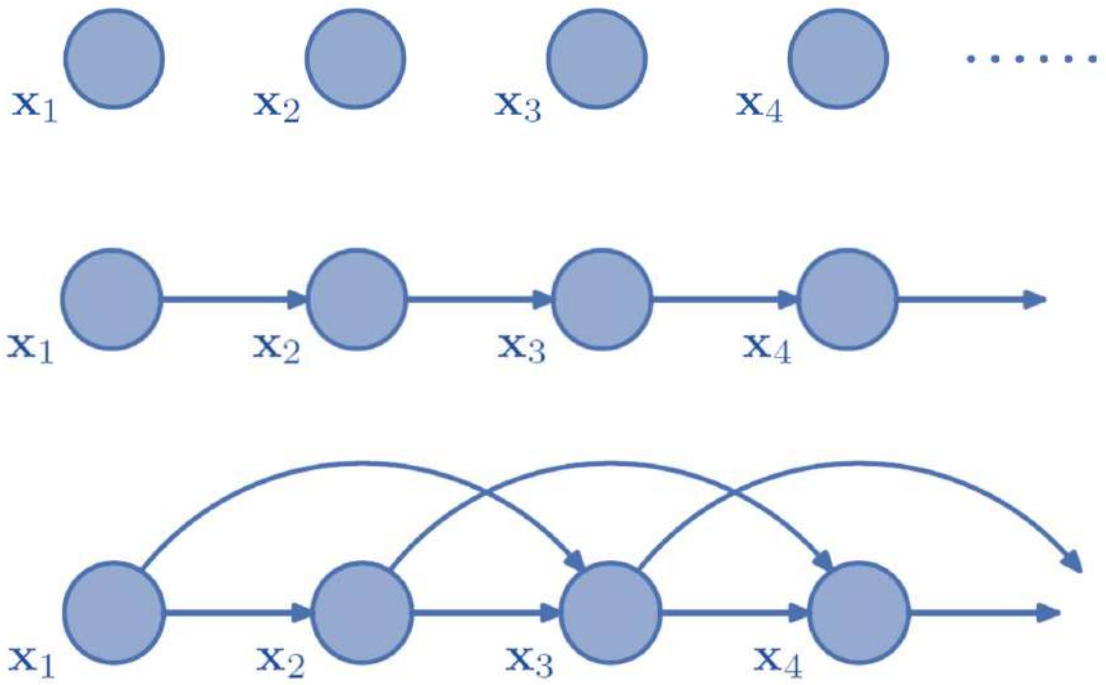
Vision Transformer





Network Architecture | Recurrent Neural Networks

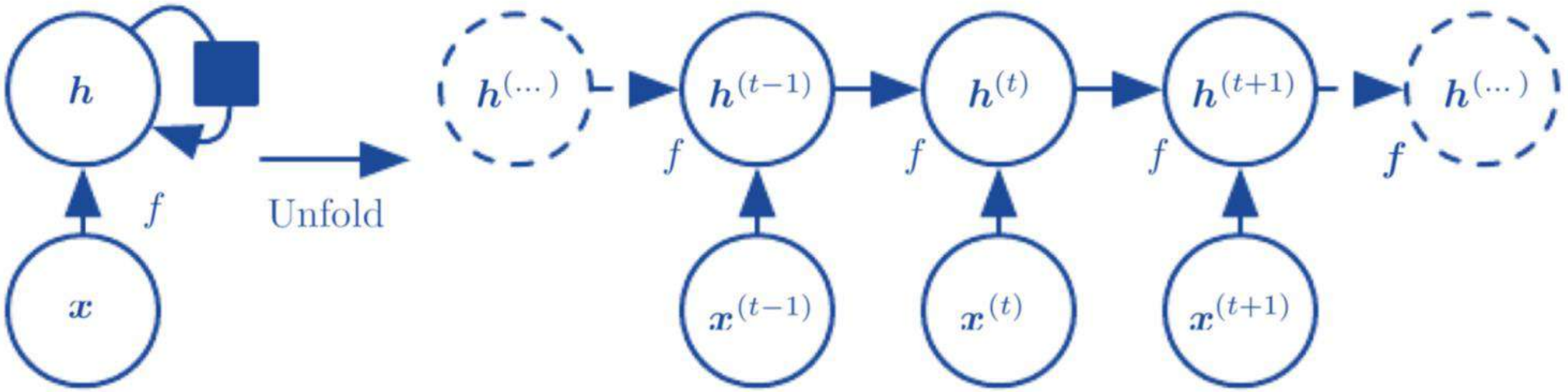
Sequential data



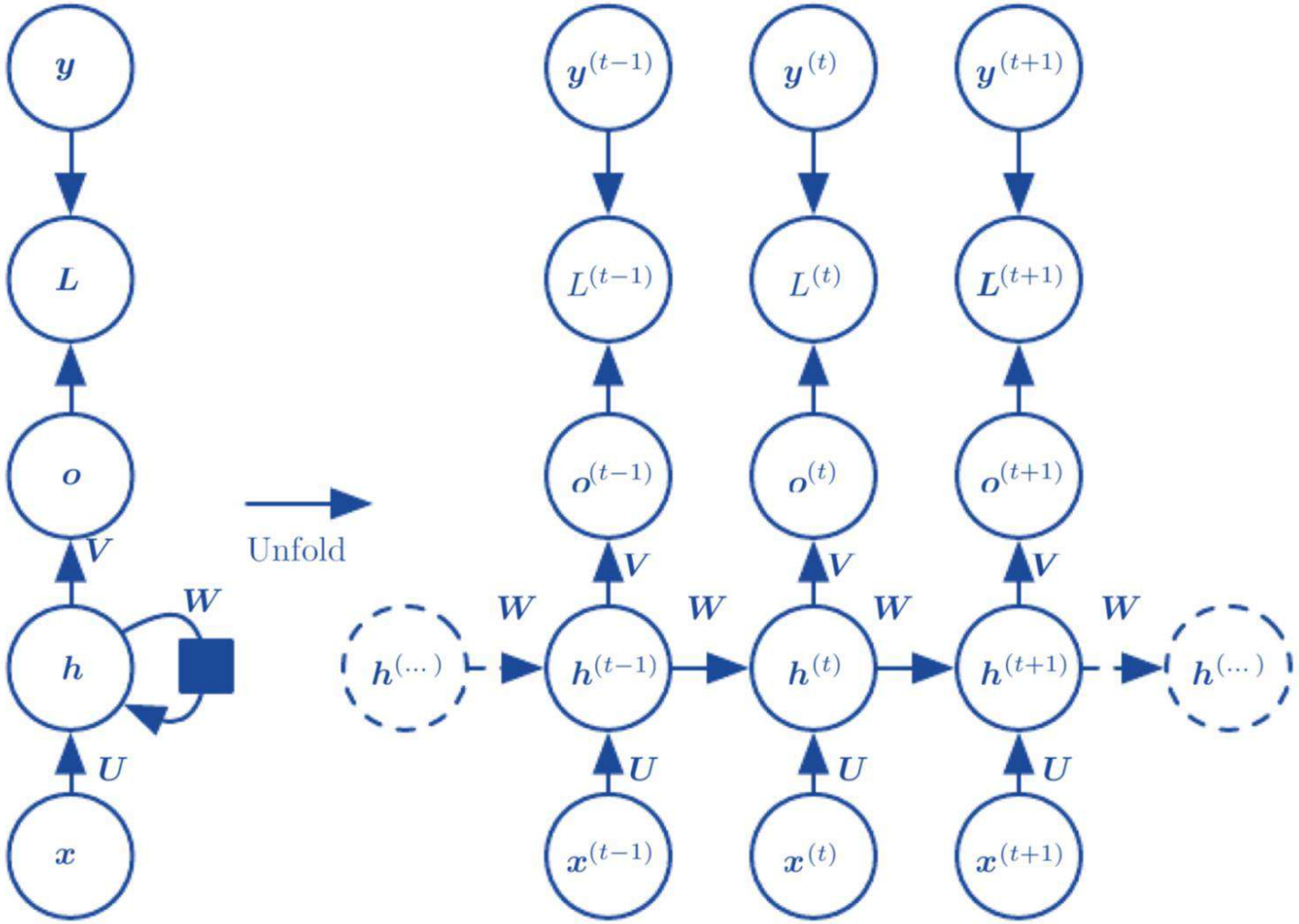
$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \prod_{n=1}^{N} p(\mathbf{x}_n | \mathbf{x}_1, \ldots, \mathbf{x}_{n-1})$$

Q: examples in medical imaging?

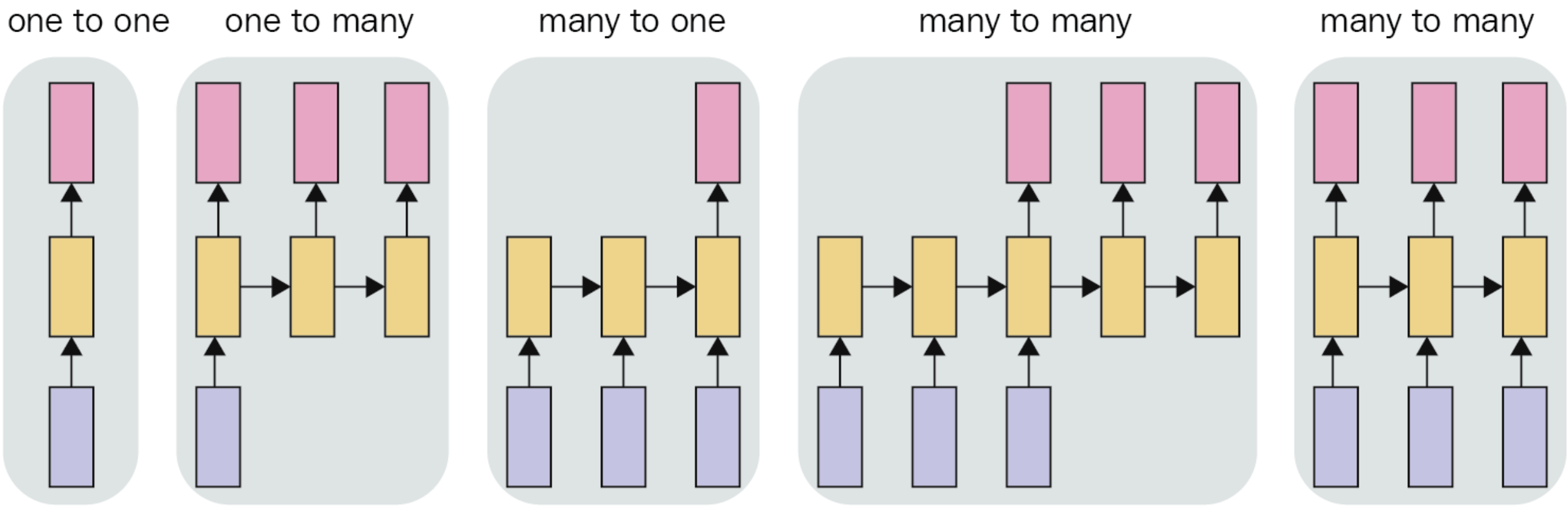
Unfolding neural networks



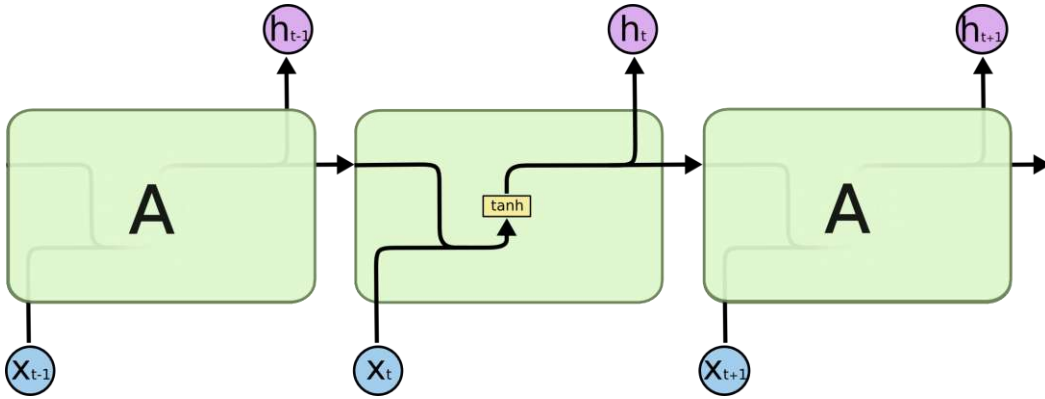
Unfolding neural networks



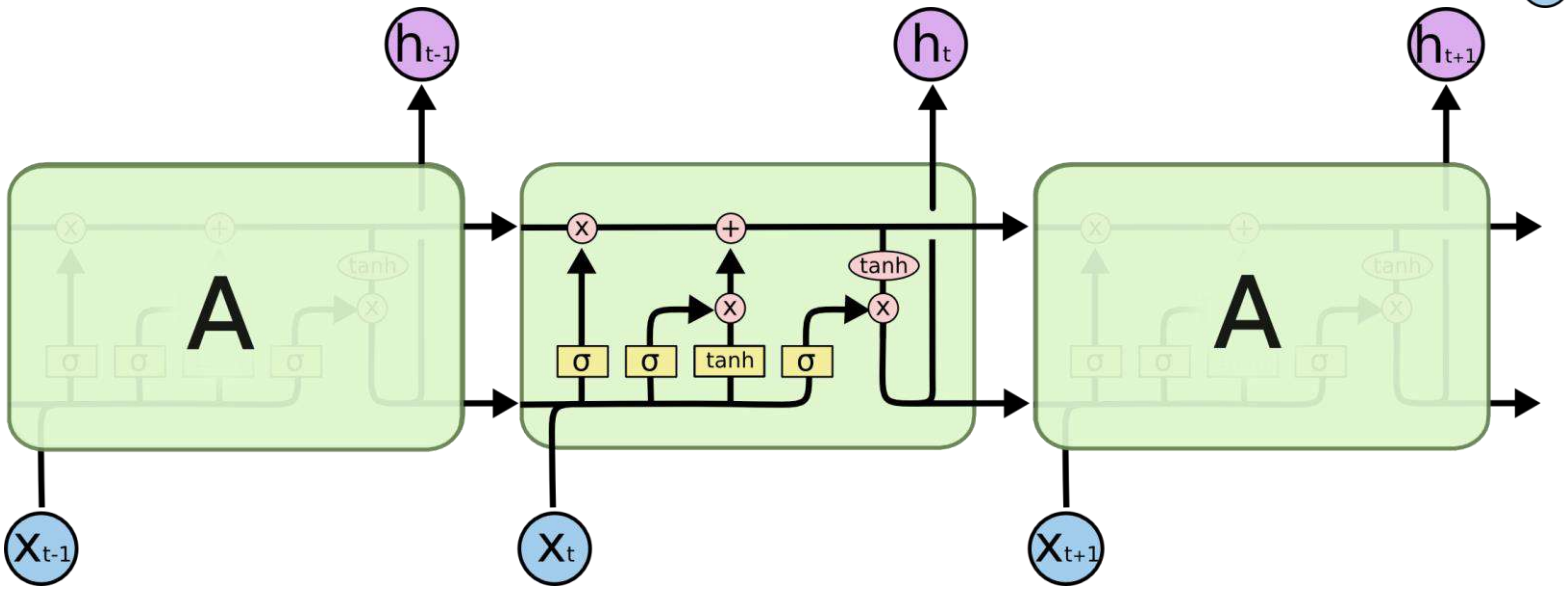
(Truncated) backpropagation through time (BPTT)


one to one
one to many
many to one
many to many
many to many

Long short-term memory (LSTM)



Standard RNN



LSTM (gated methods and cell state, “a belt for information flow”)

How to design and choose

Application requirements

- Input / output dimensionalities
- Data format, feature vectors, images, time series

Practice

- Baseline and ablation experiments
- “Do no harm” principles
- Complicated relationship with data and training strategies
- An art?