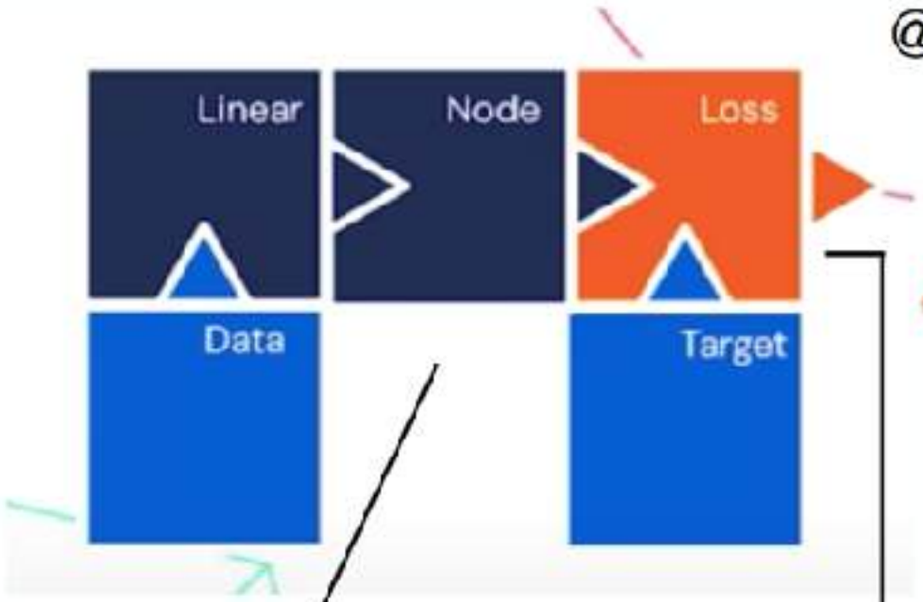
Supervised Deep Learning: CNNs, TFs, GNNs

A. Boucaud, M. Huertas-Company

Ecole d’été Rodolphe Clédassou - Session 2024 - Cycle2

RECAP Cycle 1: NNs as universal approximators

(deepmind
@Czarnecki)

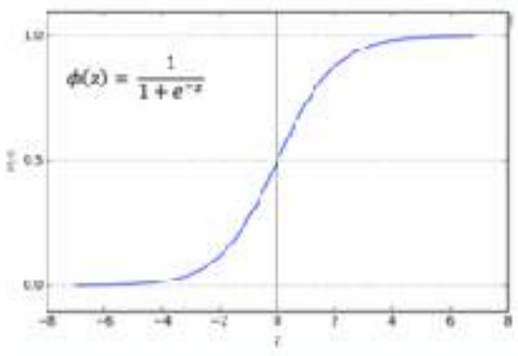


$$f_w(X) = \theta$$

$$\vdots$$

$$g_3(W_3 g_2(W_2 g_1(W_1 \vec{x_0})))$$

activation function

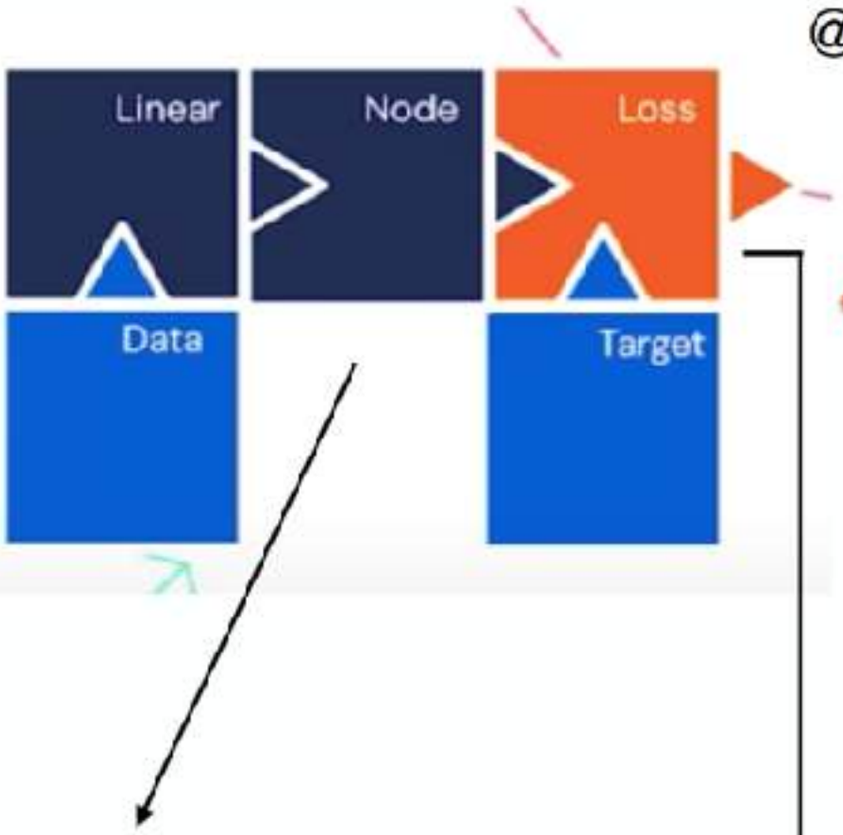


$$\frac{1}{N}\sum_{i=1}^{N}(y_i - f_w(x))^2$$

Minimized through gradient descent (backpropagation)

RECAP Cycle 1: NNs as universal approximators

(deepmind @Czarnecki)

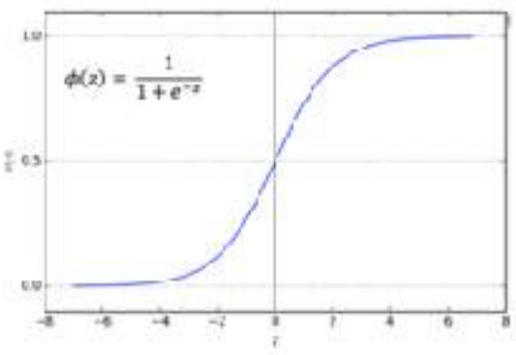


$$p_w(\theta|X)$$

$$f_w(X) = \theta$$

$$g_3(W_3 g_2(W_2 g_1(W_1 \vec{x_0})))$$

activation function

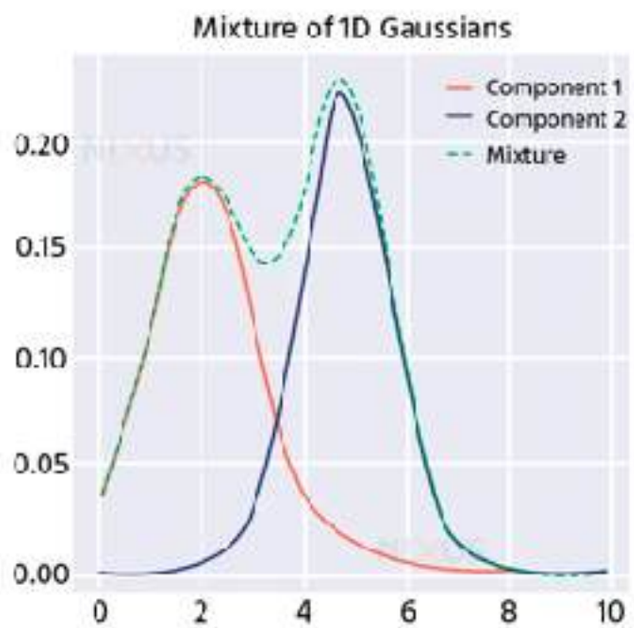


$$\frac{1}{N}\sum_{i=1}^{N}(y_i - f_w(x))^2$$

Minimized through gradient descent (backpropagation)

Cycle1：

$$p_w(\theta|\boldsymbol{X})$$



Cycle2:

$$p_w(\theta|\boldsymbol{X})$$

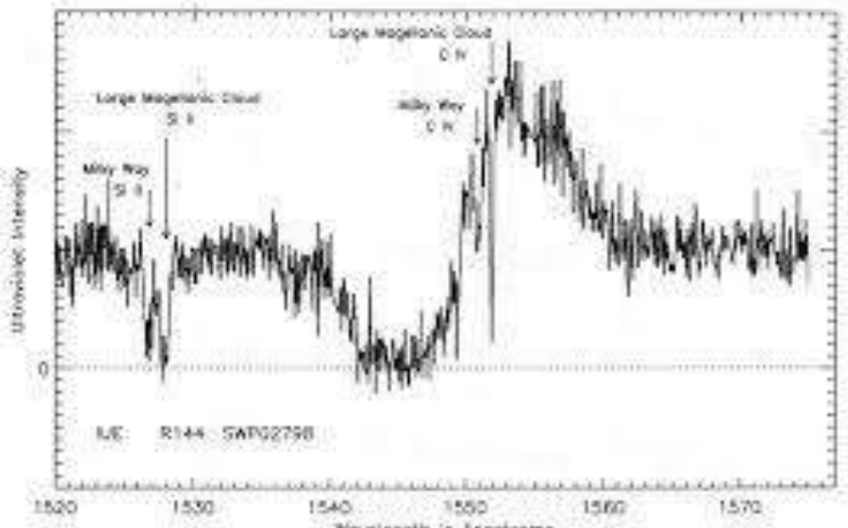
Explore different data structures

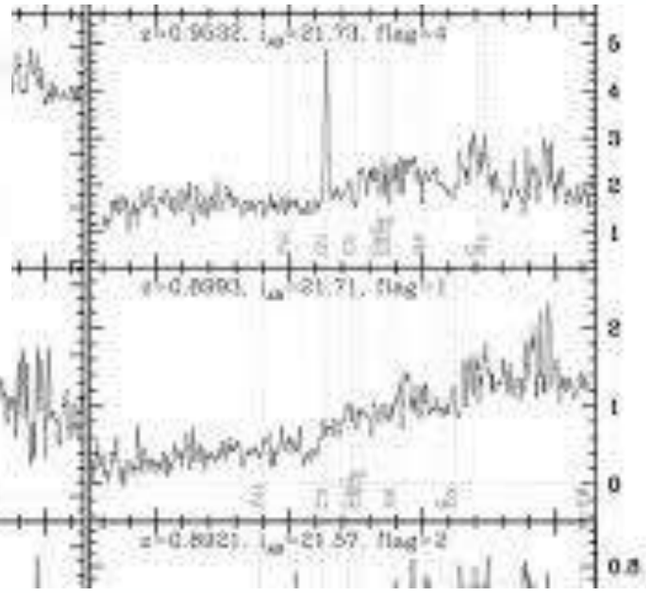
1. Neural Networks for Computer Vision

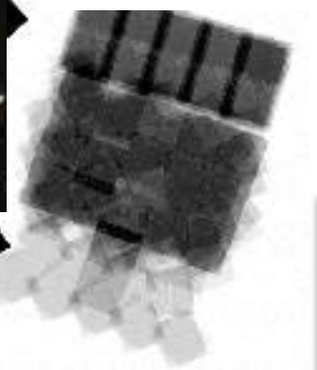
$$p_w(\theta|X)$$

X={images}

What do we put as input?

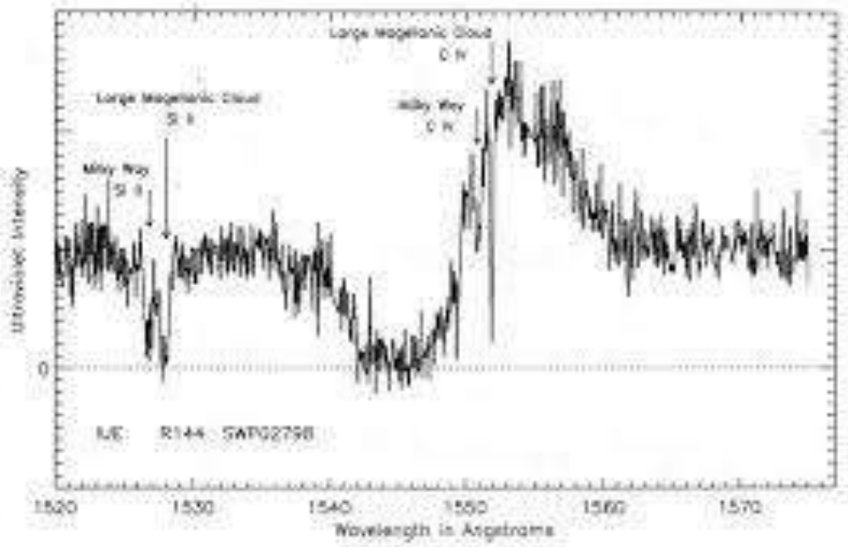


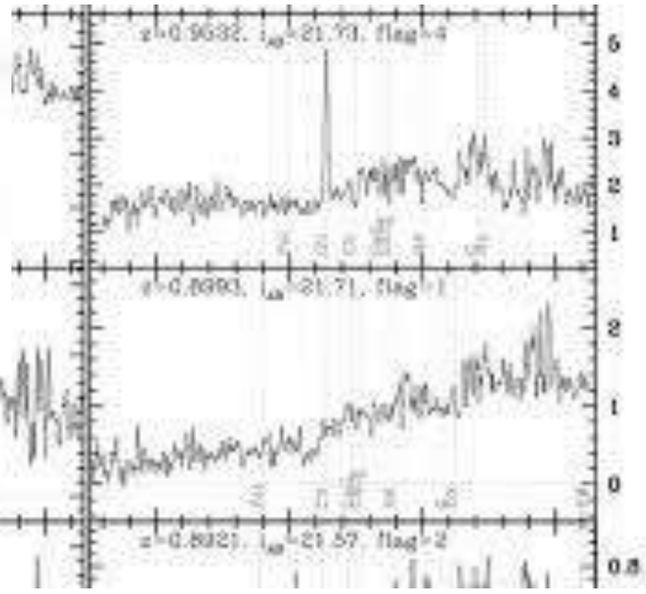




THIS IS WHAT MACHINES SEE

What do we put as input?

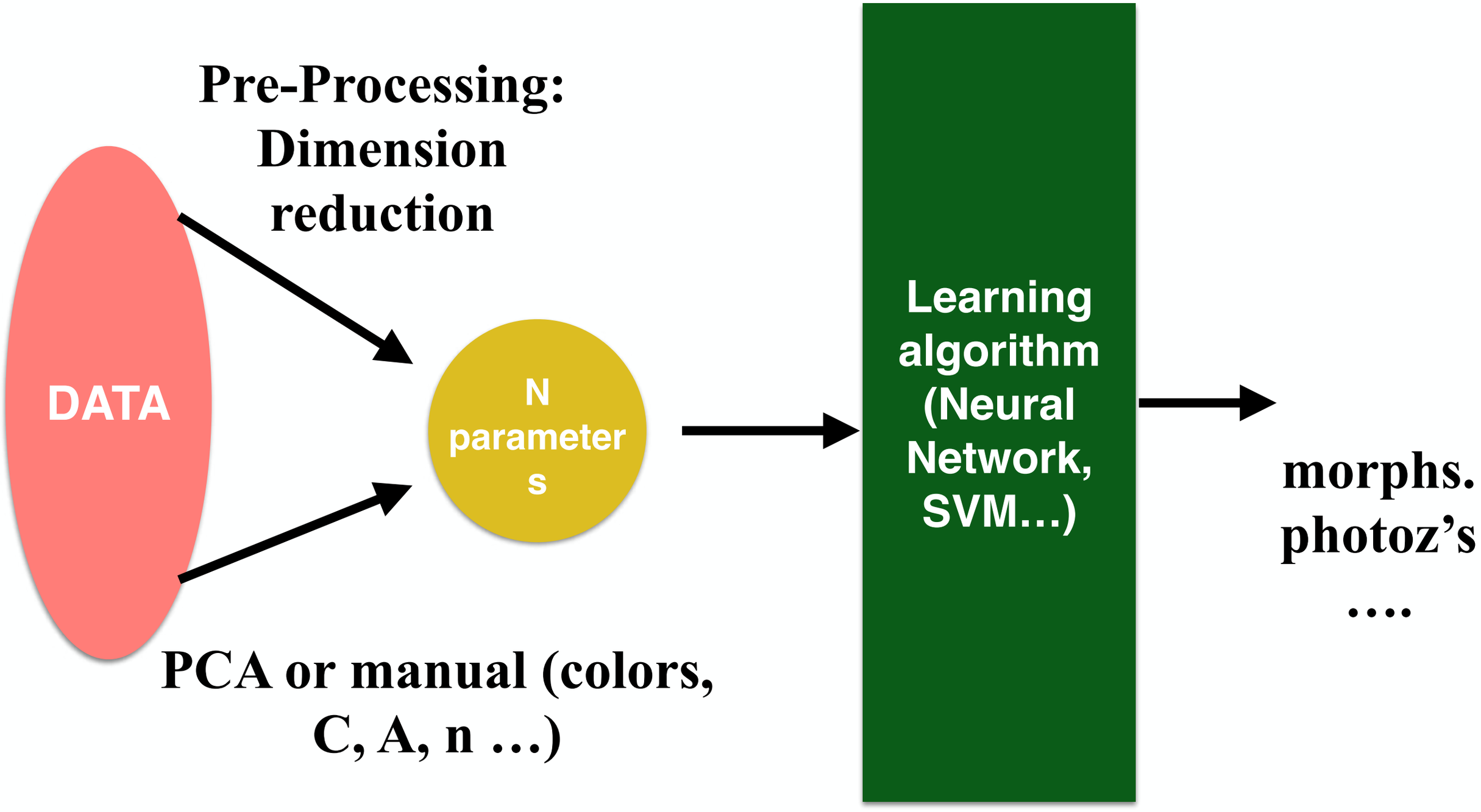




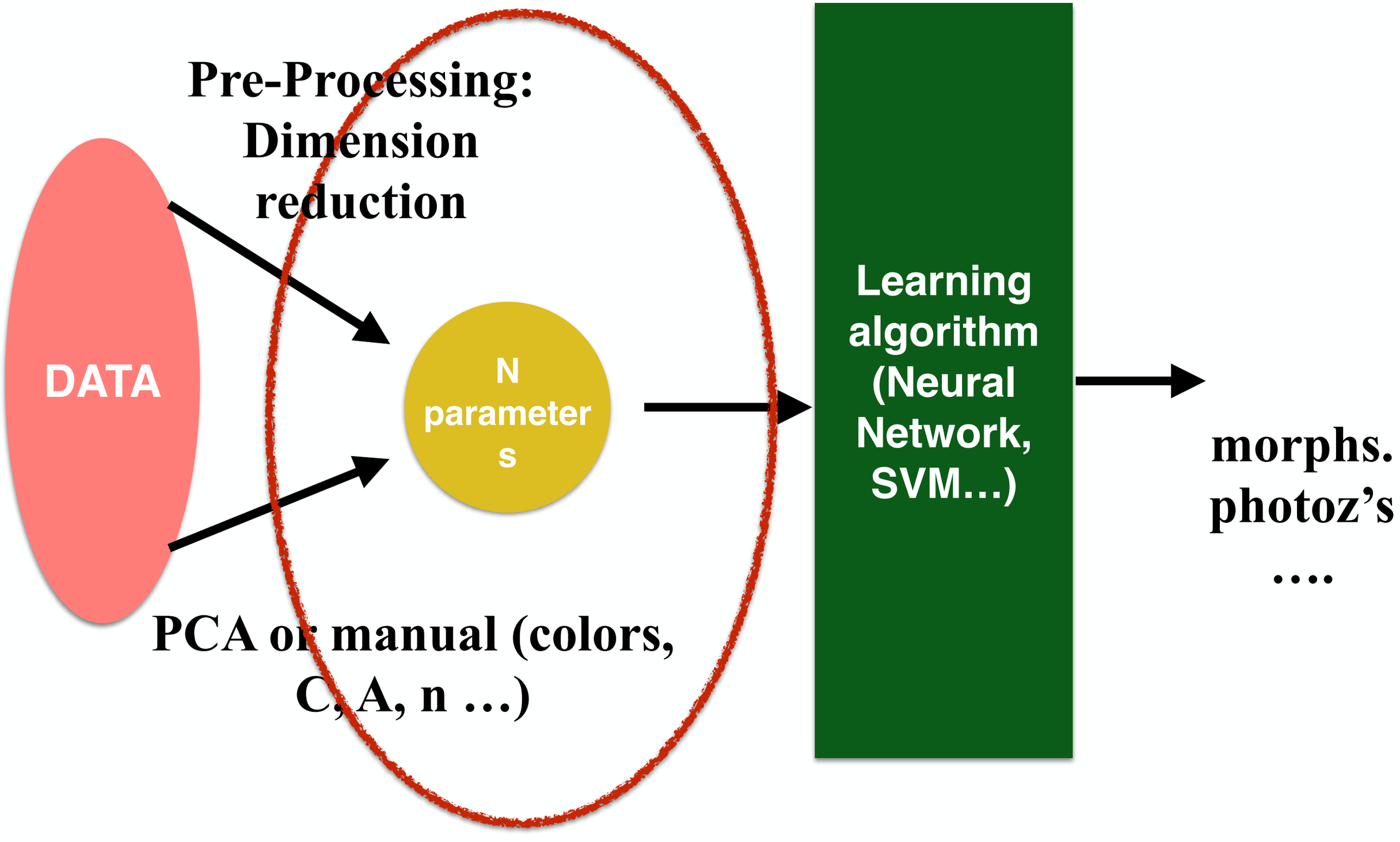
PRE-PROCESS DATA TO EXTRACT MEANINGFUL INFORMATION

THIS IS GENERALLY CALLED **FEATURE EXTRACTION**

THE "CLASSICAL" APPROACH



“CLASSICAL” MACHINE LEARNING



In Astronomy

- Colors, Fluxes
- Shape indicators
- Line ratios, spectral features
- Stellar Masses, Velocity Dispersions

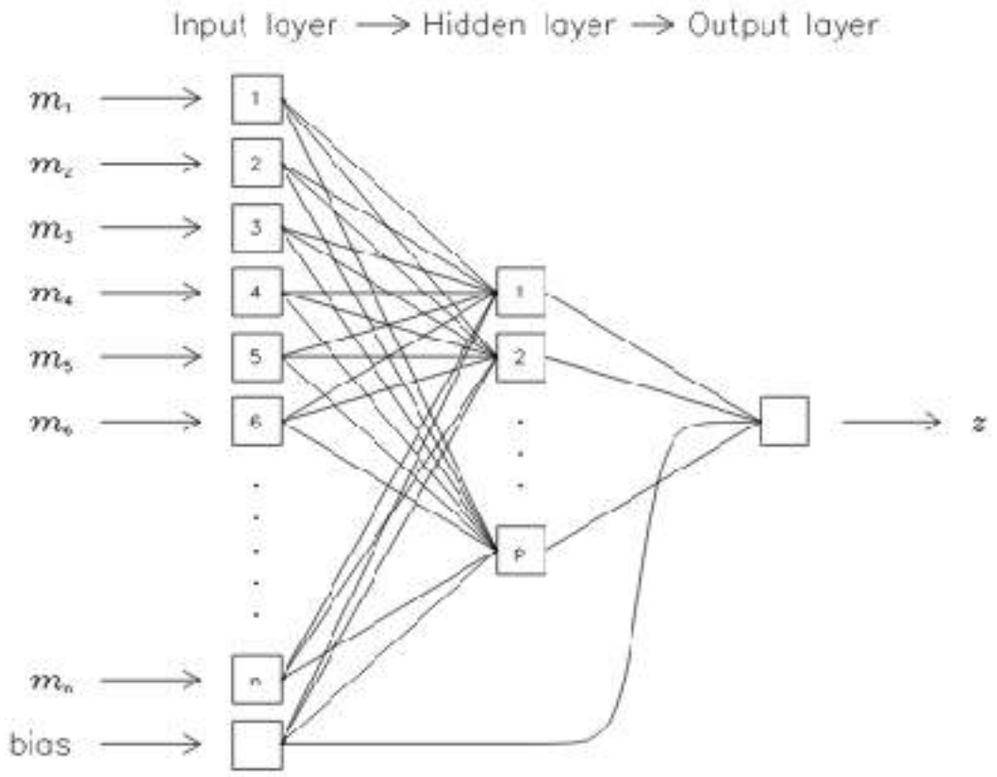
Requires specialized software before feeding the machine learning algorithm

IT IMPLIES A DIMENSIONALITY REDUCTION!

PHOTOMETRIC REDSHIFTS

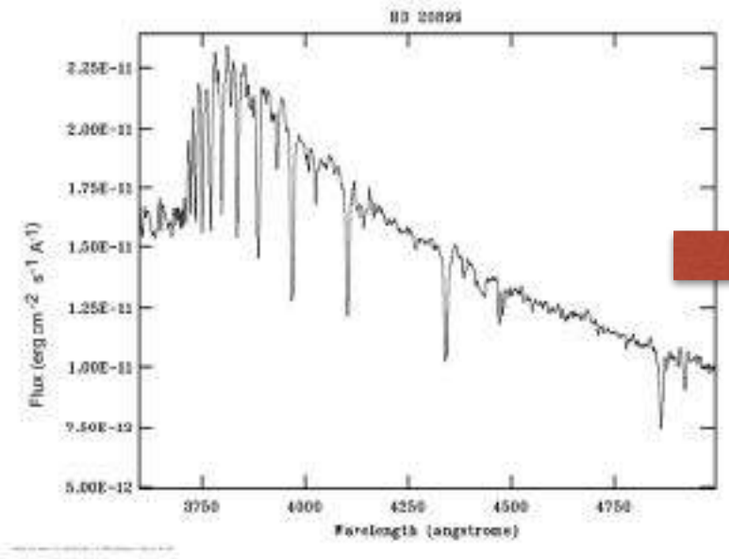
SDSS

g
r
i
z

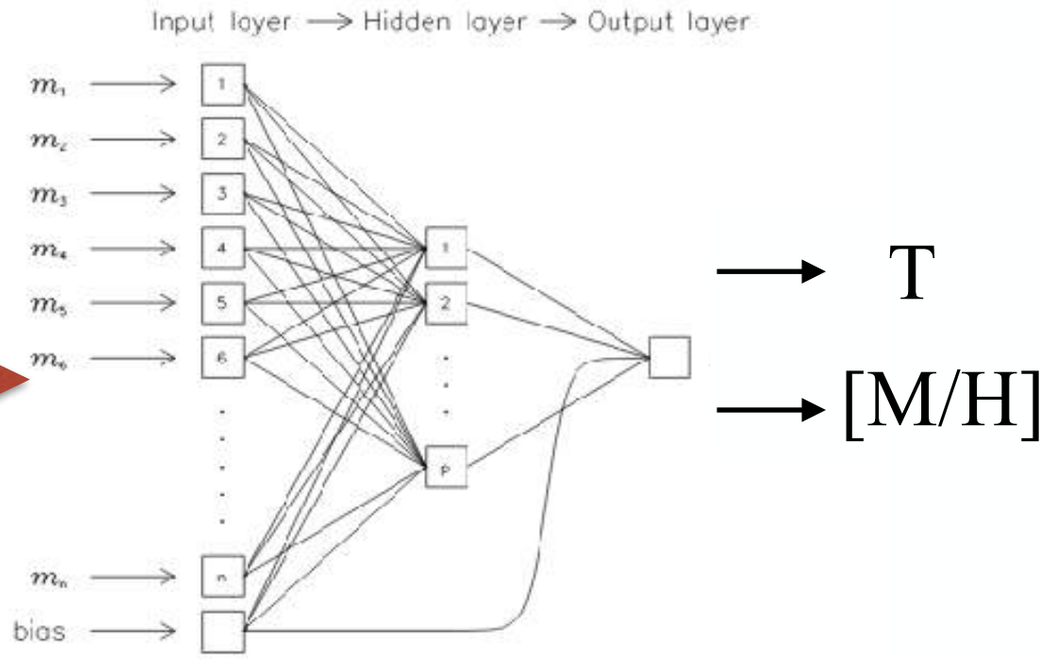


Collister+08

STELLAR PARAMETERS FROM MEDIUM BAND FILTERS



MEDIUM BAND FLUXES



→ T

→ [M/H]

Bailer-Jones+00

EVERYTHING IS IN THE FEATURES….WHAT IF I IGNORED SOME IMPORTANT FEATURES?

EVERYTHING IS IN THE FEATURES….WHAT IF I IGNORED SOME IMPORTANT FEATURES?

WHAT ABOUT USING RAW DATA?

ALL INFORMATION IS IN THE INPUT DATA

WHY REDUCING ?

LET THE NETWORK FIND THE INFO

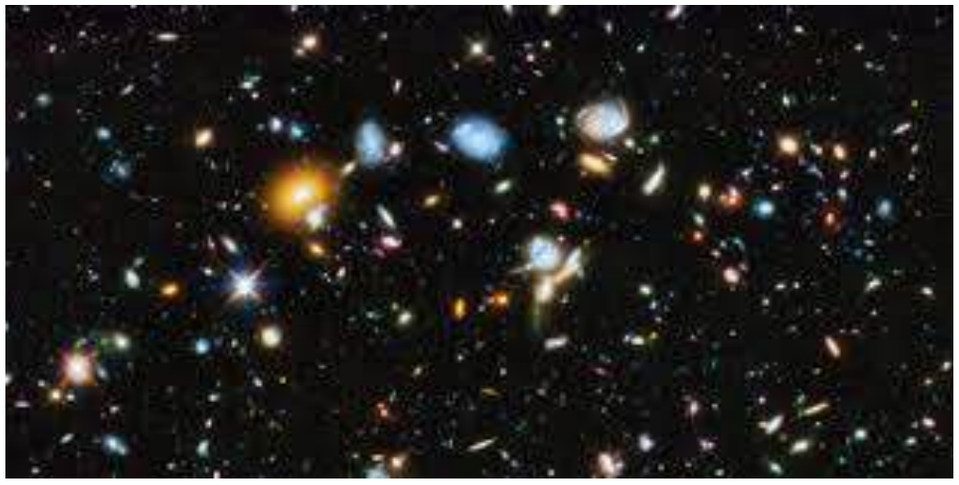
LARGE DIMENSION SIGNALS SUCH AS IMAGES OR SPECTRA WOULD REQUIRE TREMENDOUSLY LARGE MODELS

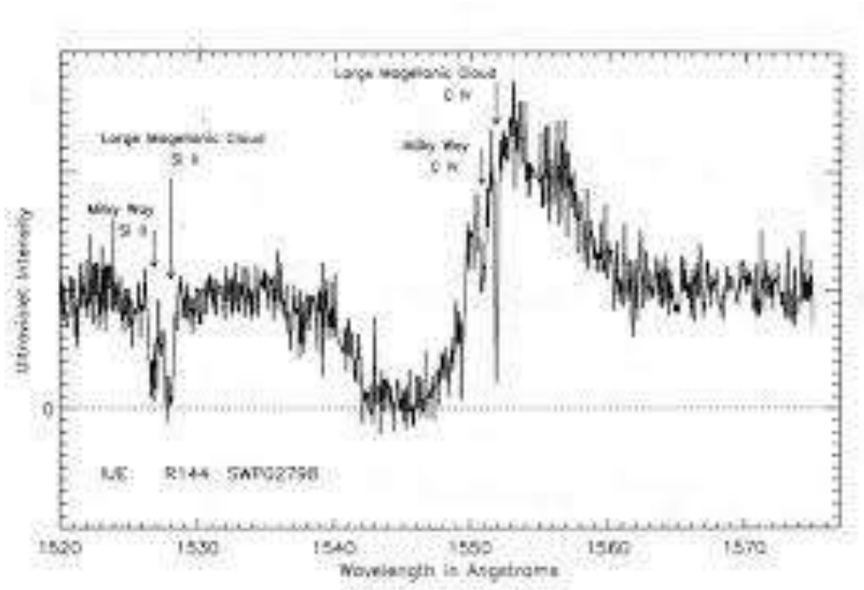
A 512x512 image as input of a fully connected layer producing output of same size:

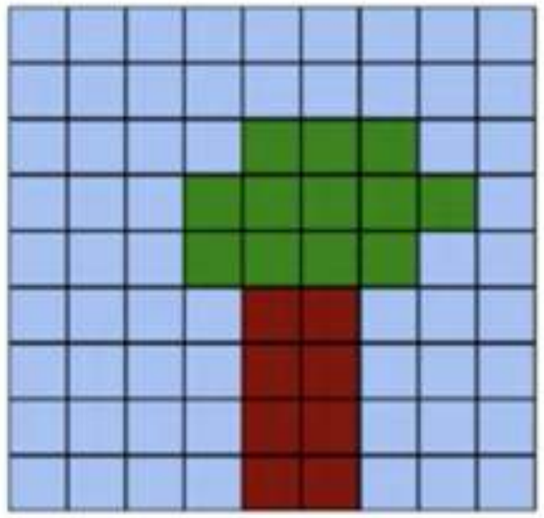
$$(512 \times 512)^2 = 7e10$$

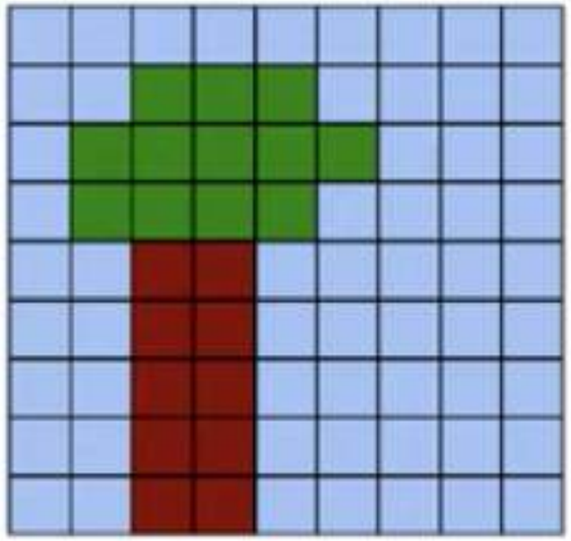
BUT

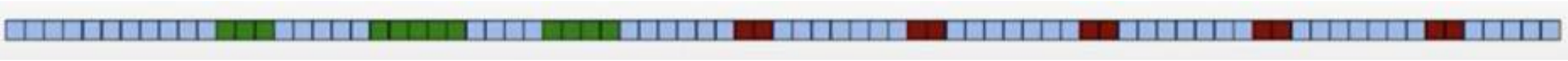
FEEDING INDIVIDUAL RESOLUTION ELEMENTS IS NOT VERY EFFICIENT SINCE IT LOOSES ALL INVARIANCE TO TRANSLATION AND IGNORES CORRELATION IN THE DATA AT ALL SCALES





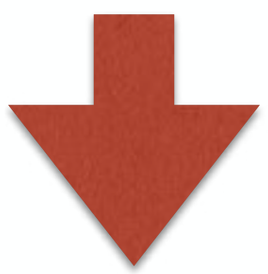






(Dielemann@Deepmind)

FEEDING INDIVIDUAL RESOLUTION ELEMENTS IS NOT VERY EFFICIENT SINCE IT LOOSES ALL INVARIANCE TO TRANSLATION



SO?

TWO BASIC PROPERTIES OF IMAGING DATA (BUT ALSO SPECTROSCOPY IN SOME SENSE) ARE **LOCALITY**

TRANSLATION INVARIANCE

locality: nearby pixels are more strongly correlated

translational invariance: meaningful patterns can appear anywhere in the image

(Dielemann@Deepmind)

Discrete Convolution

1D: [Spectra]

$$f(x) * g(x) = \sum_{k=-\infty}^{k=+\infty} f(k).g(k-x)$$

2D: [Images]

$$f(x,y) * g(x,y) = \sum_{k=-\infty}^{k=+\infty} \sum_{l=-\infty}^{l=+\infty} f(k,l).g(x-k,y-l)$$

DISCRETE CONVOLUTION

1D: [Spectra]

$$f(x) * g(x) = \sum_{k=-\infty}^{k=+\infty} f(k).g(k-x)$$

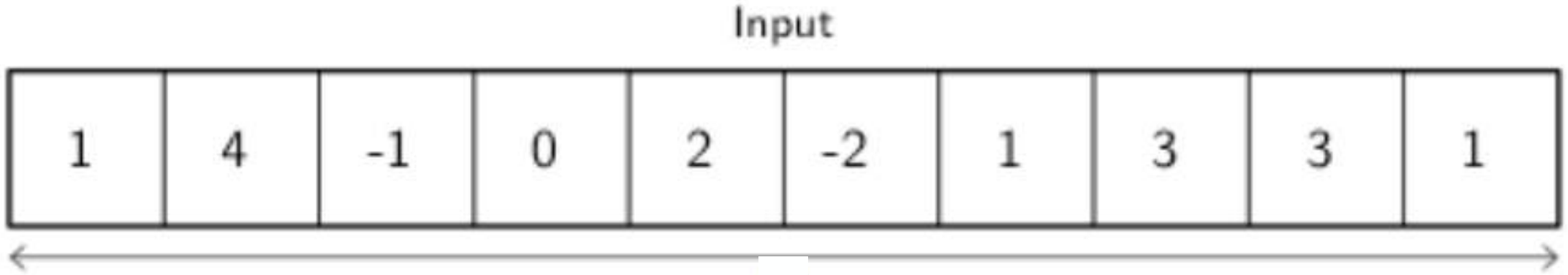
2D: [Images]

$$f(x,y) * g(x,y) = \sum_{k=-\infty}^{k=+\infty} \sum_{l=-\infty}^{l=+\infty} f(k,l).g(x-k,y-l)$$

CONVOLUTION KERNEL

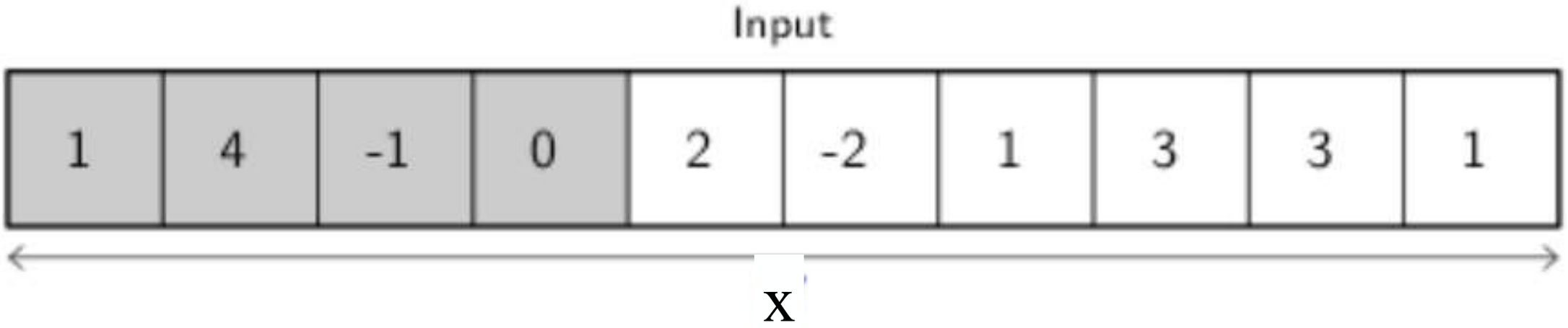
INPUT DATA

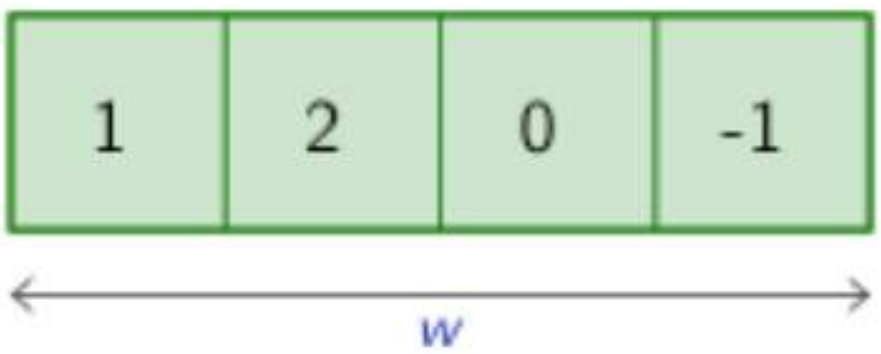
1-D CONVOLUTION



credit

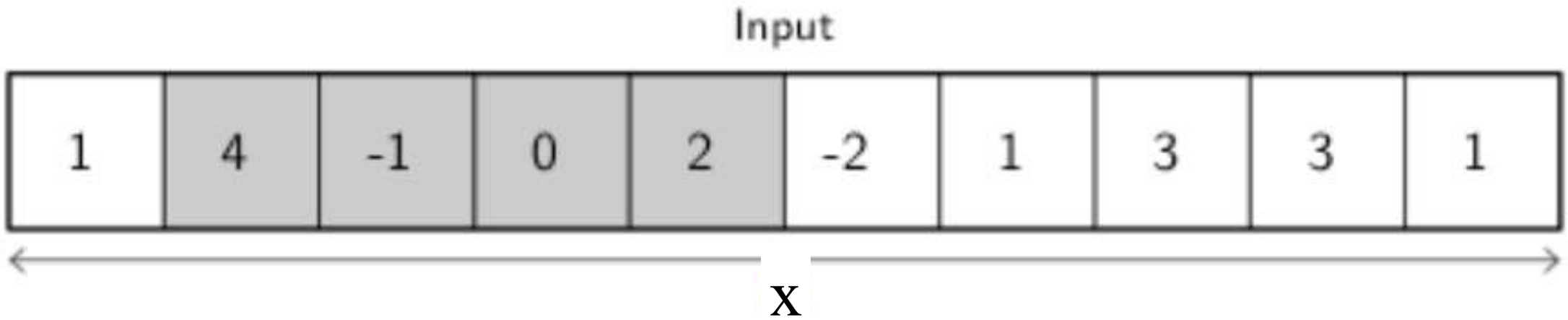
1-D CONVOLUTION





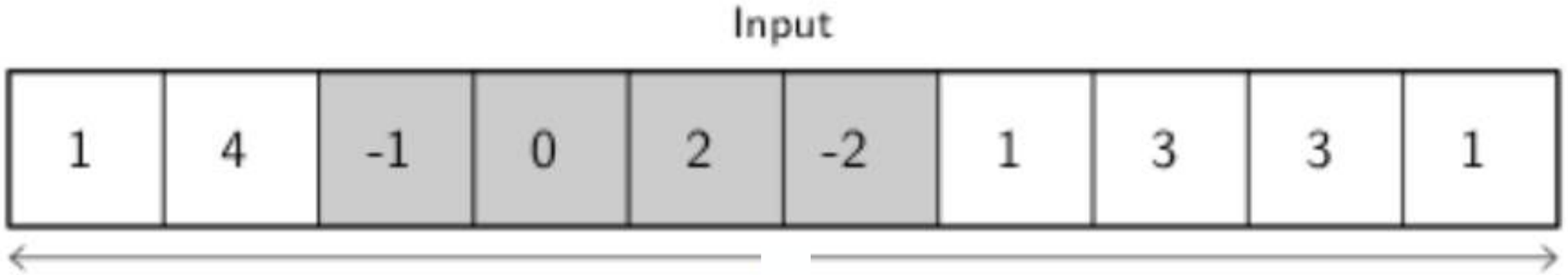
credit

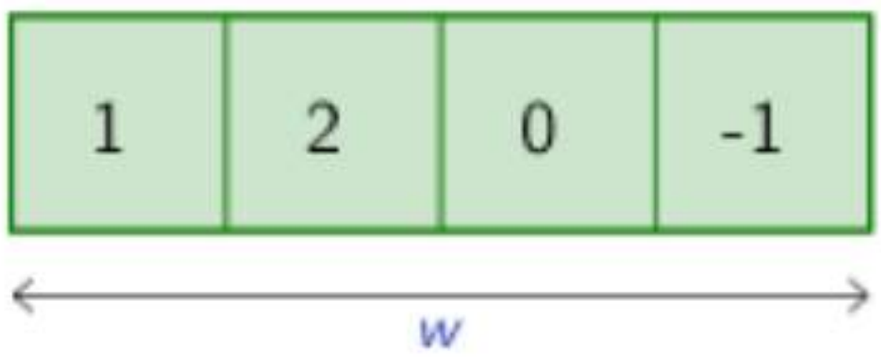
1-D CONVOLUTION



credit

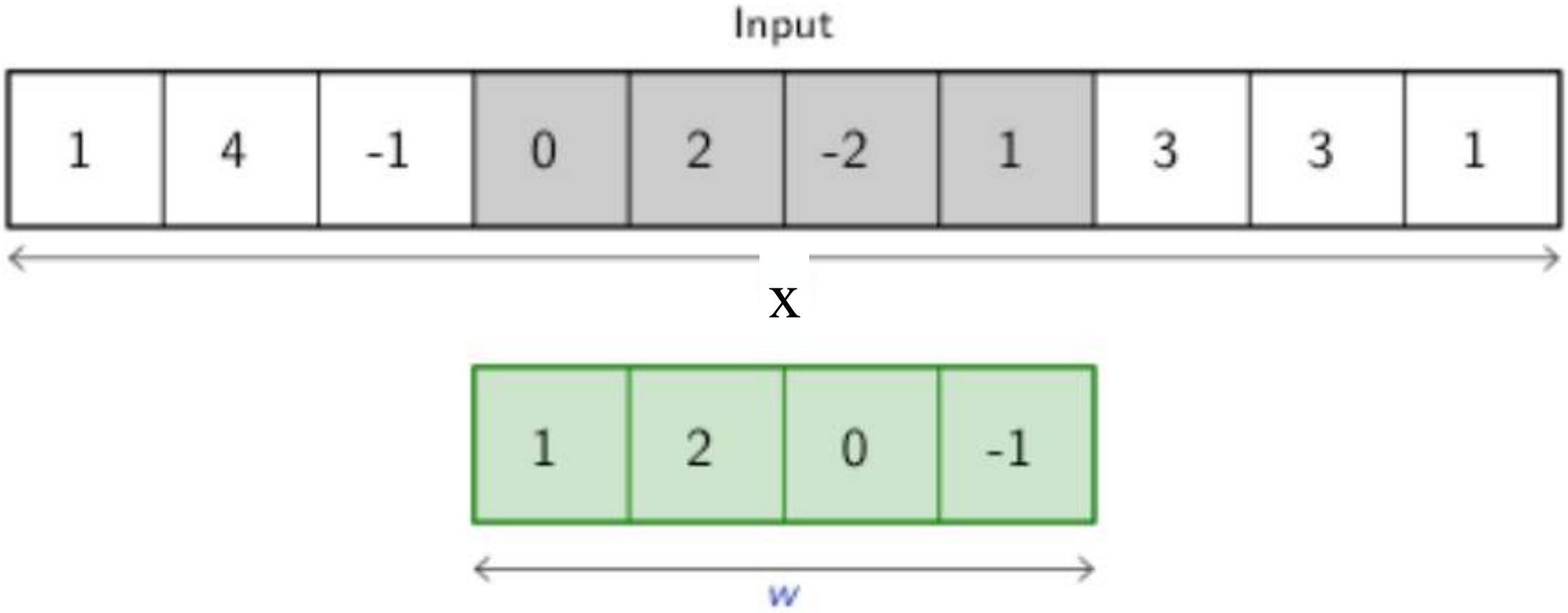
1-D CONVOLUTION





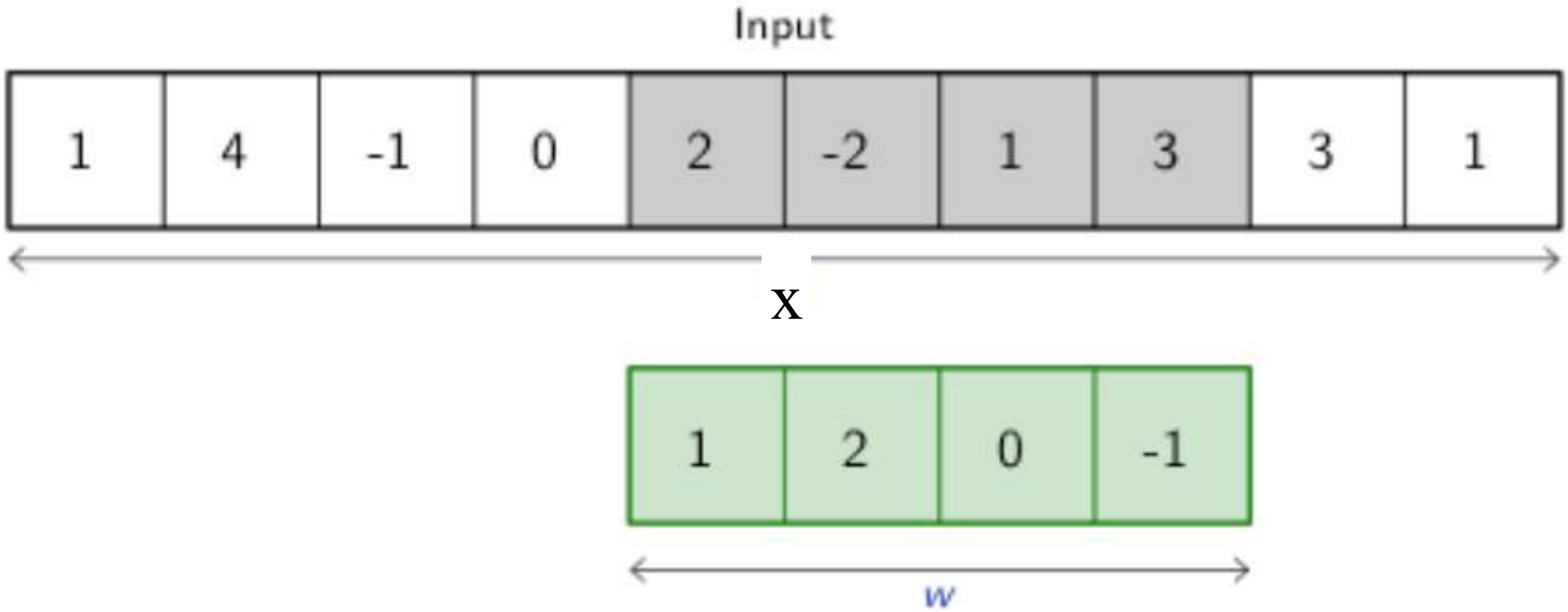
credit

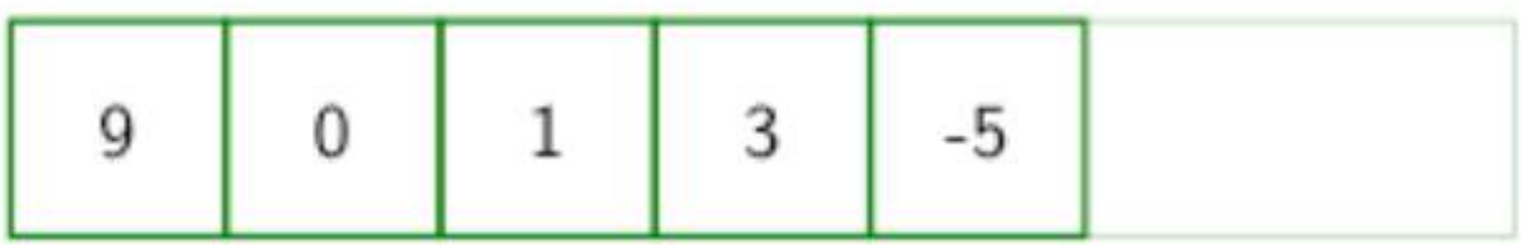
1-D CONVOLUTION



credit

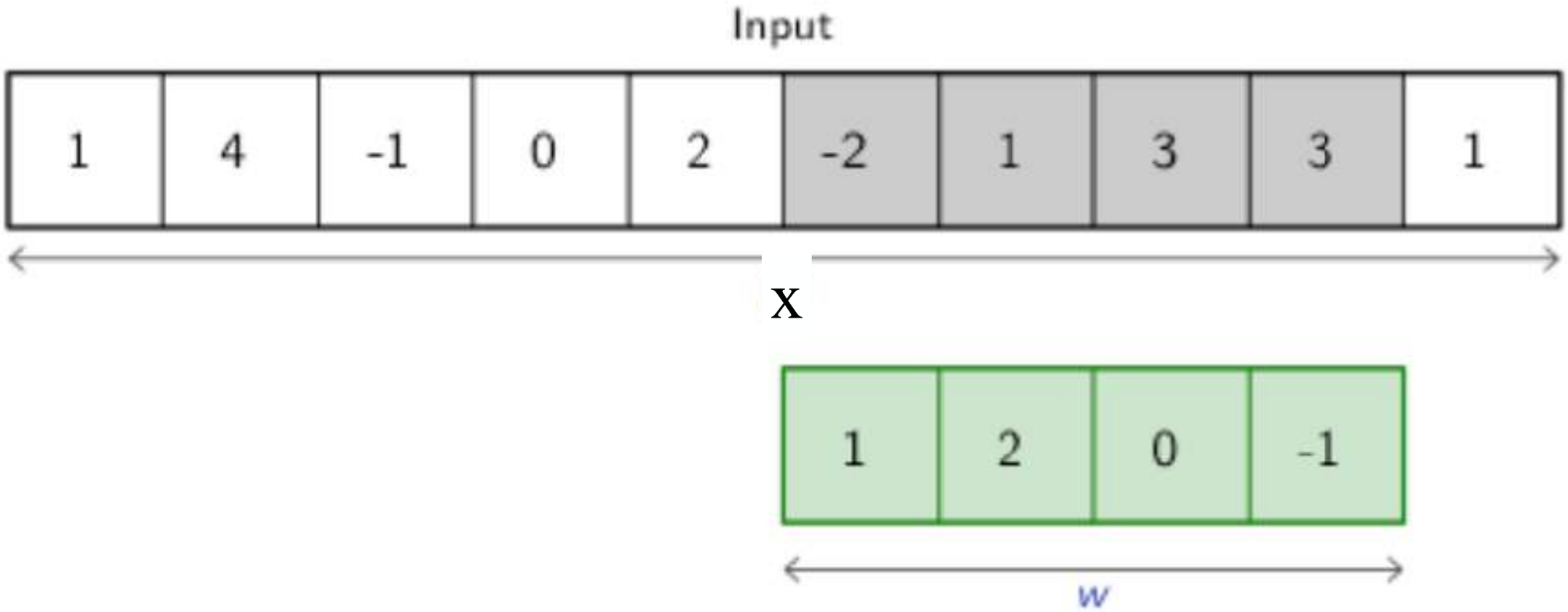
1-D CONVOLUTION





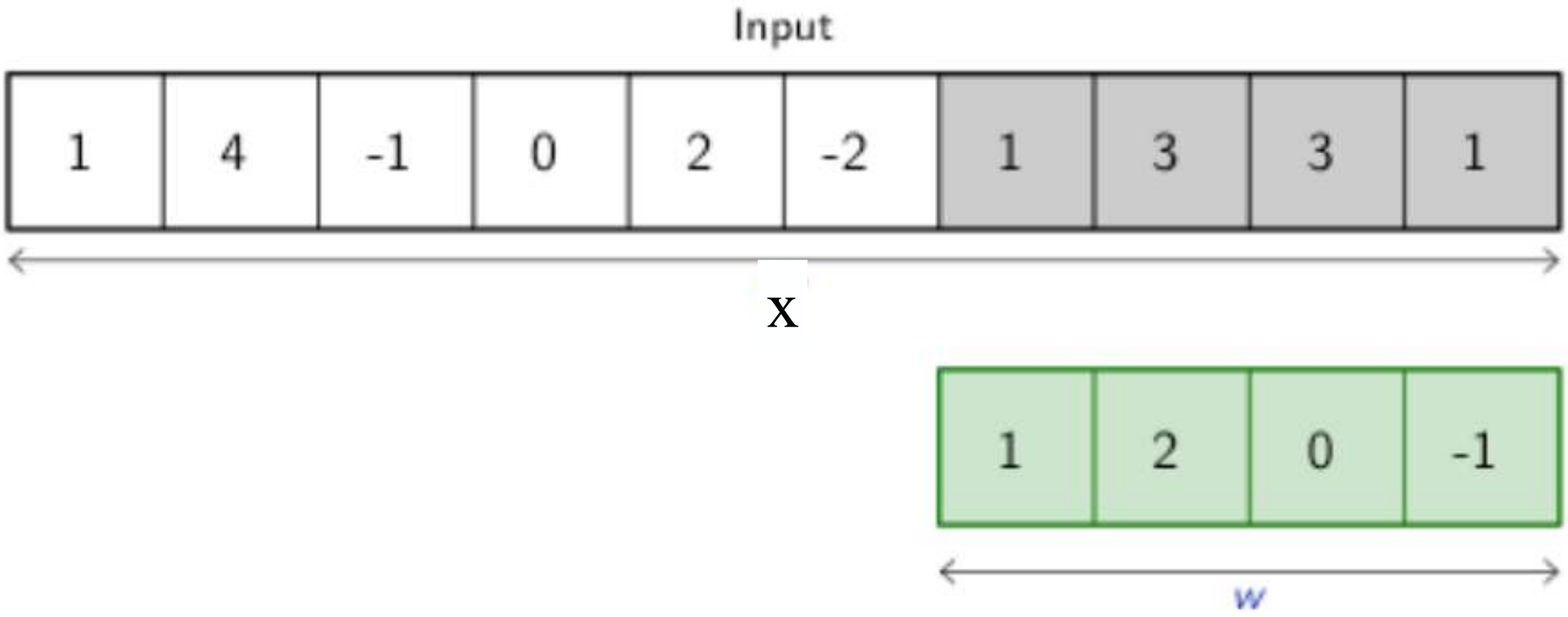
credit

1-D CONVOLUTION



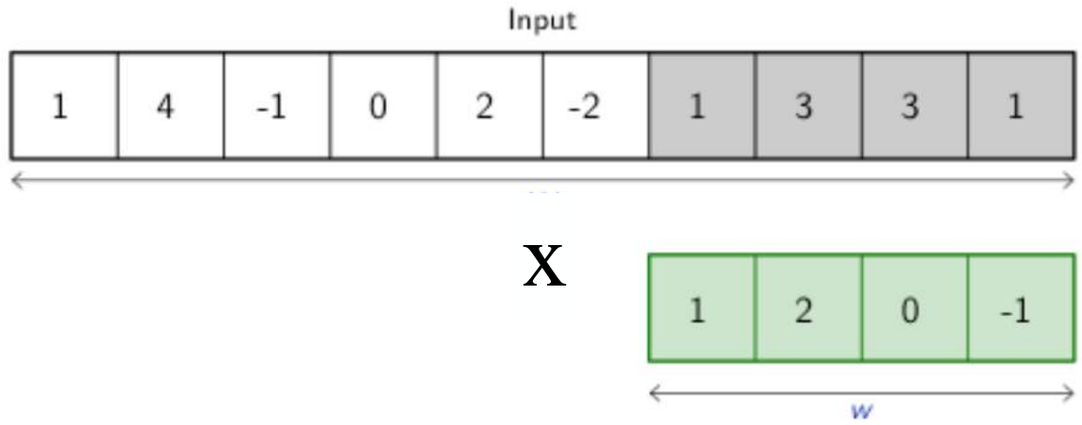
credit

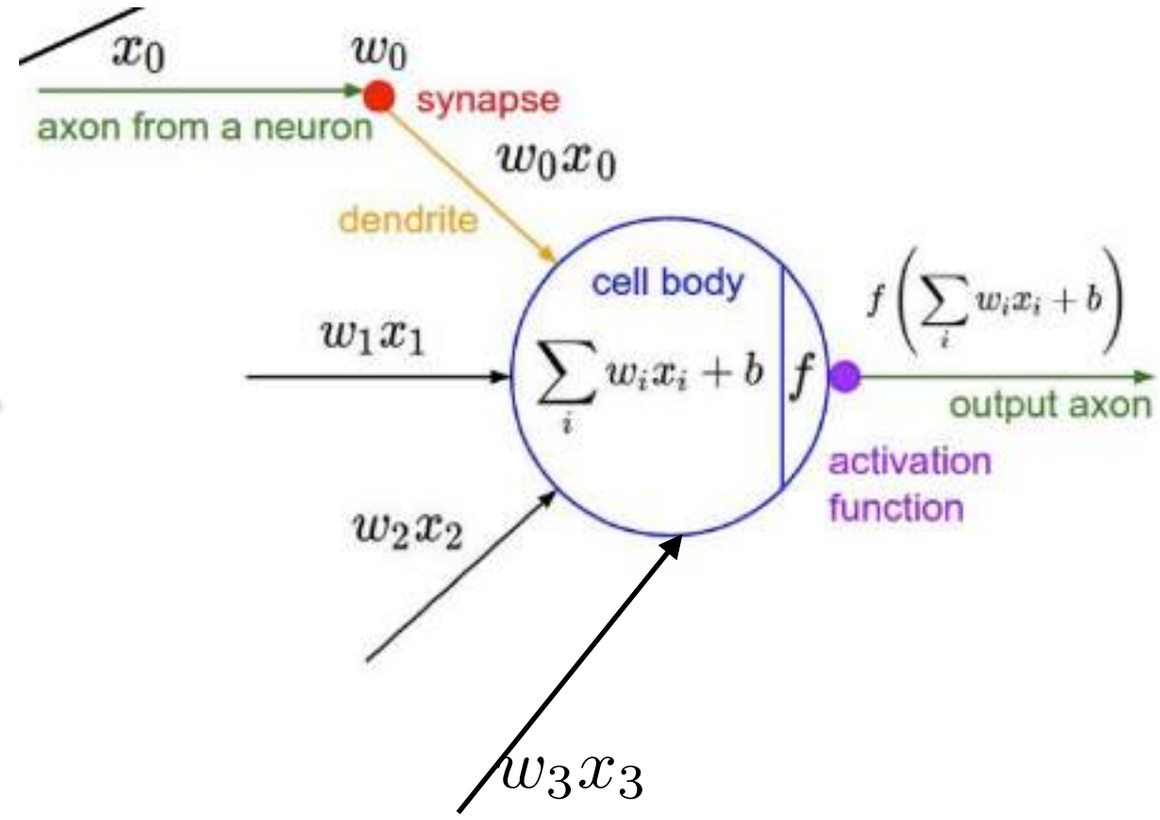
1-D CONVOLUTION



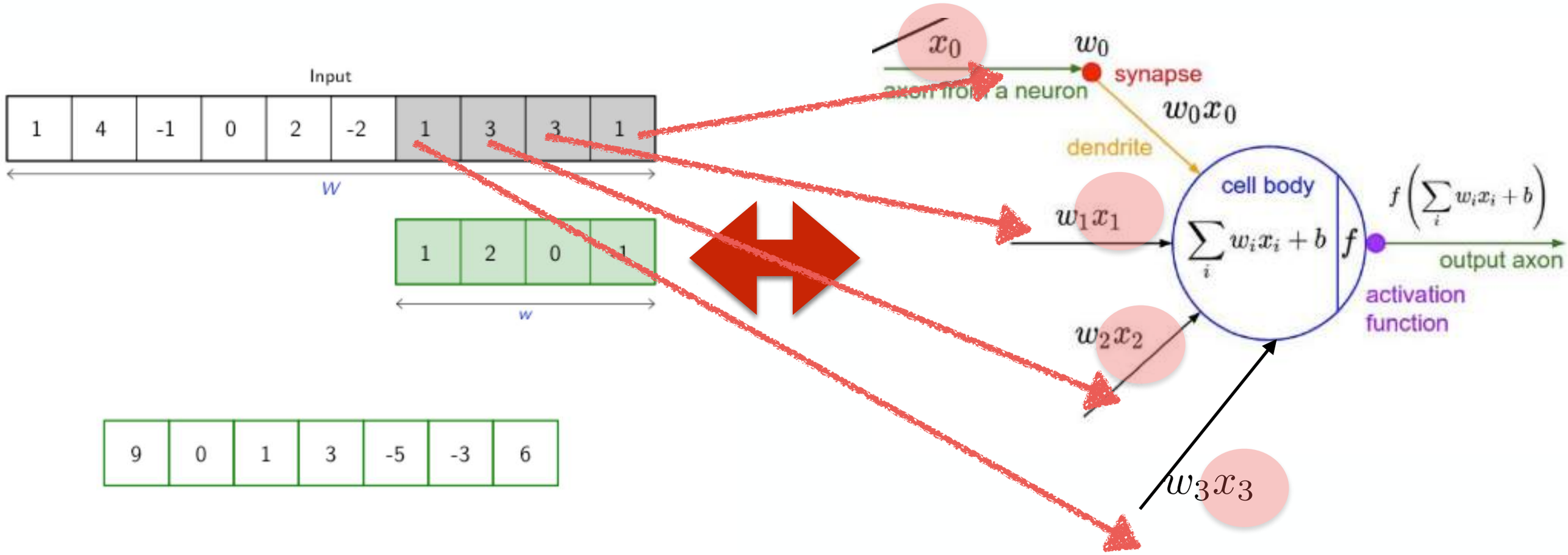
credit

THE CONVOLUTION BUILDING BLOCK OPERATION (BEFORE ACTIVATION) IS EQUIVALENT TO A NEURON WITH AS MANY INPUTS AS KERNEL ELEMENTS AND WEIGHTS EQUAL TO THE KERNEL

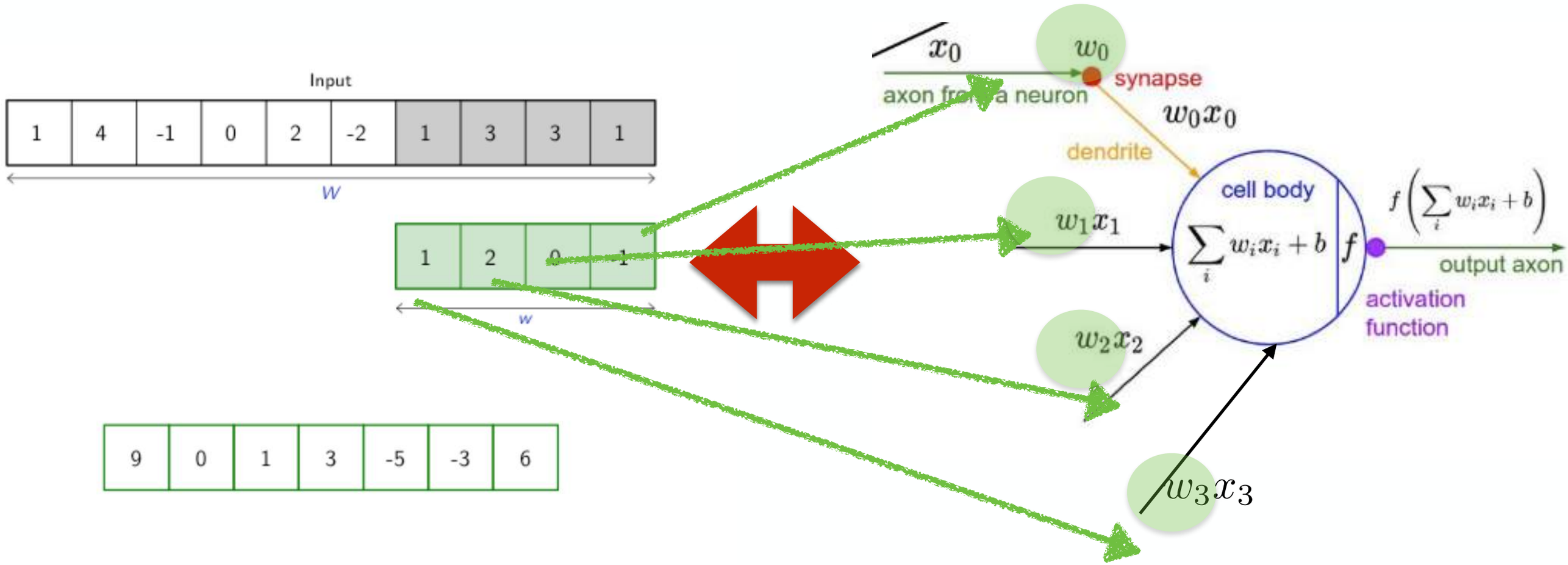




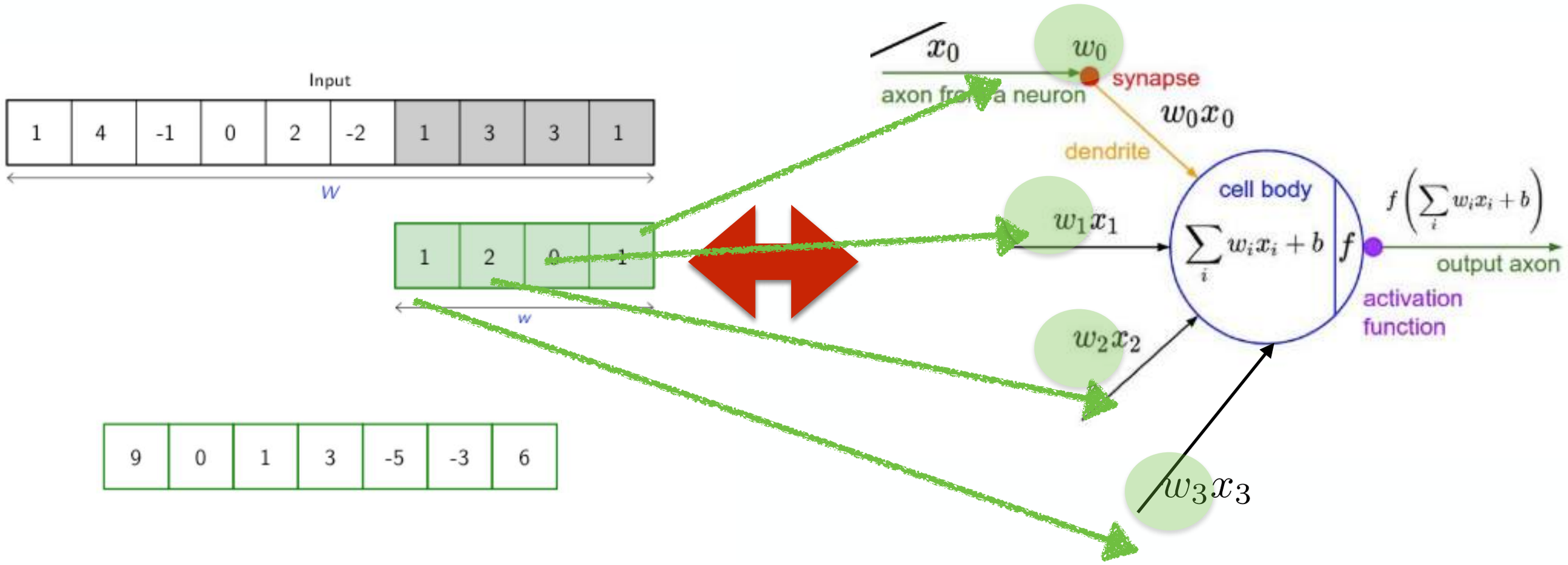
THE CONVOLUTION BUILDING BLOCK OPERATION (BEFORE ACTIVATION) IS EQUIVALENT TO A NEURON WITH AS MANY INPUTS AS KERNEL ELEMENTS AND WEIGHTS EQUAL TO THE KERNEL



THE CONVOLUTION BUILDING BLOCK OPERATION (BEFORE ACTIVATION) IS EQUIVALENT TO A NEURON WITH AS MANY INPUTS AS KERNEL ELEMENTS AND WEIGHTS EQUAL TO THE KERNEL



THE CONVOLUTION BUILDING BLOCK OPERATION (BEFORE ACTIVATION) IS EQUIVALENT TO A NEURON WITH AS MANY INPUTS AS KERNEL ELEMENTS AND WEIGHTS EQUAL TO THE KERNEL



WITH THE ADVANTAGE THAT THE SAME WEIGHTS ARE APPLIED TO ALL THE SIGNAL: TRANSLATION INVARIANCE

2-D CONVOLUTION

SAME IDEA, BUT THE KERNEL IS NOW 2D

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

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

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

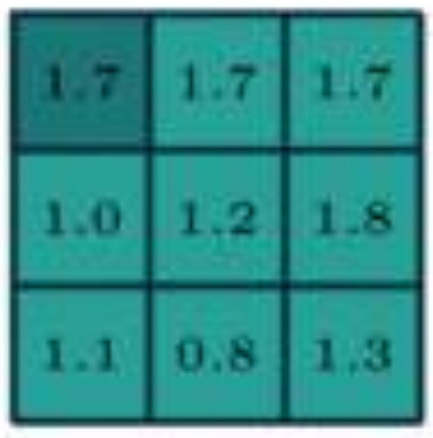
KERNEL **INPUT (IMAGE)** **OUTPUT**

Credit: animations from https://github.com/vdumoulin/conv_arithmetic

2-D CONVOLUTION

SAME IDEA, BUT THE KERNEL IS NOW 2D

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



IN THE EXAMPLE: EACH 3x3 REGION GENERATES AN OUTPUT

$$Size_{output} = Size_{input} - Size_{kernel} + 1$$

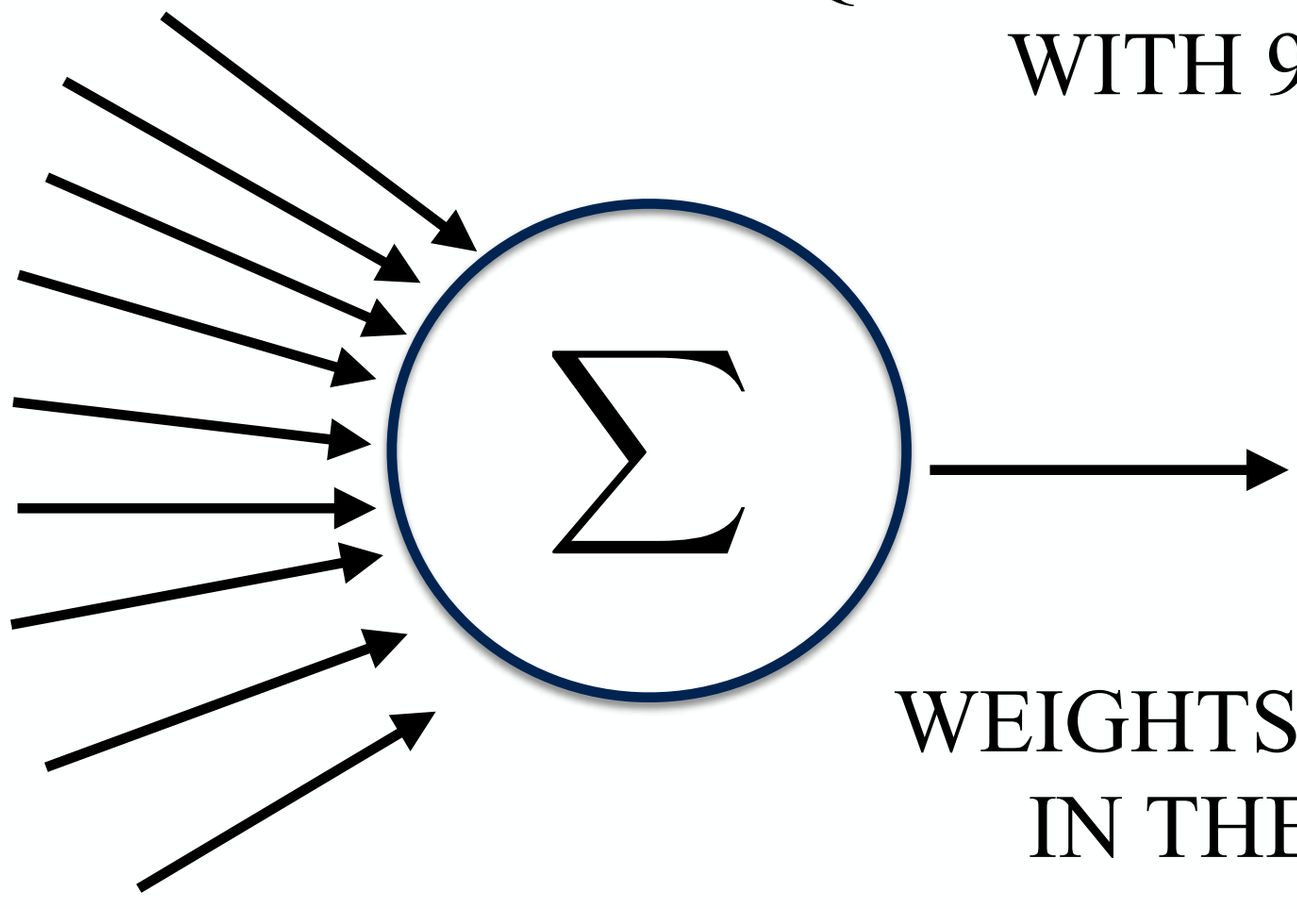
Credit: animations from https://github.com/vdumoulin/conv_arithmetic

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

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

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

EQUIVALENT TO A NEURON
WITH 9 INPUTS



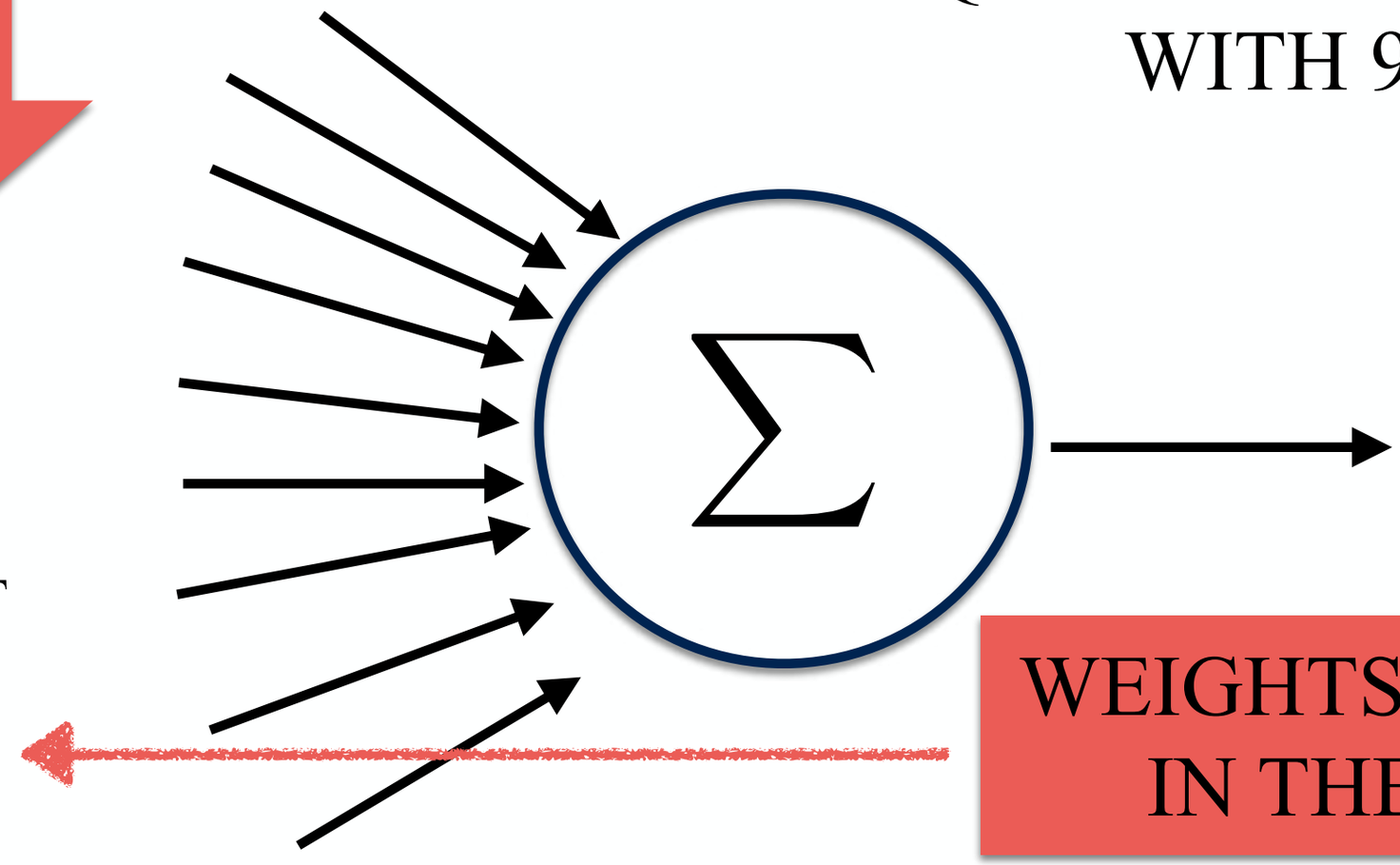
WEIGHTS ARE CODED
IN THE KERNEL

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

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

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

EQUIVALENT TO A NEURON
WITH 9 INPUTS



THIS IS WHAT
THE
NETWORK
LEARNS!

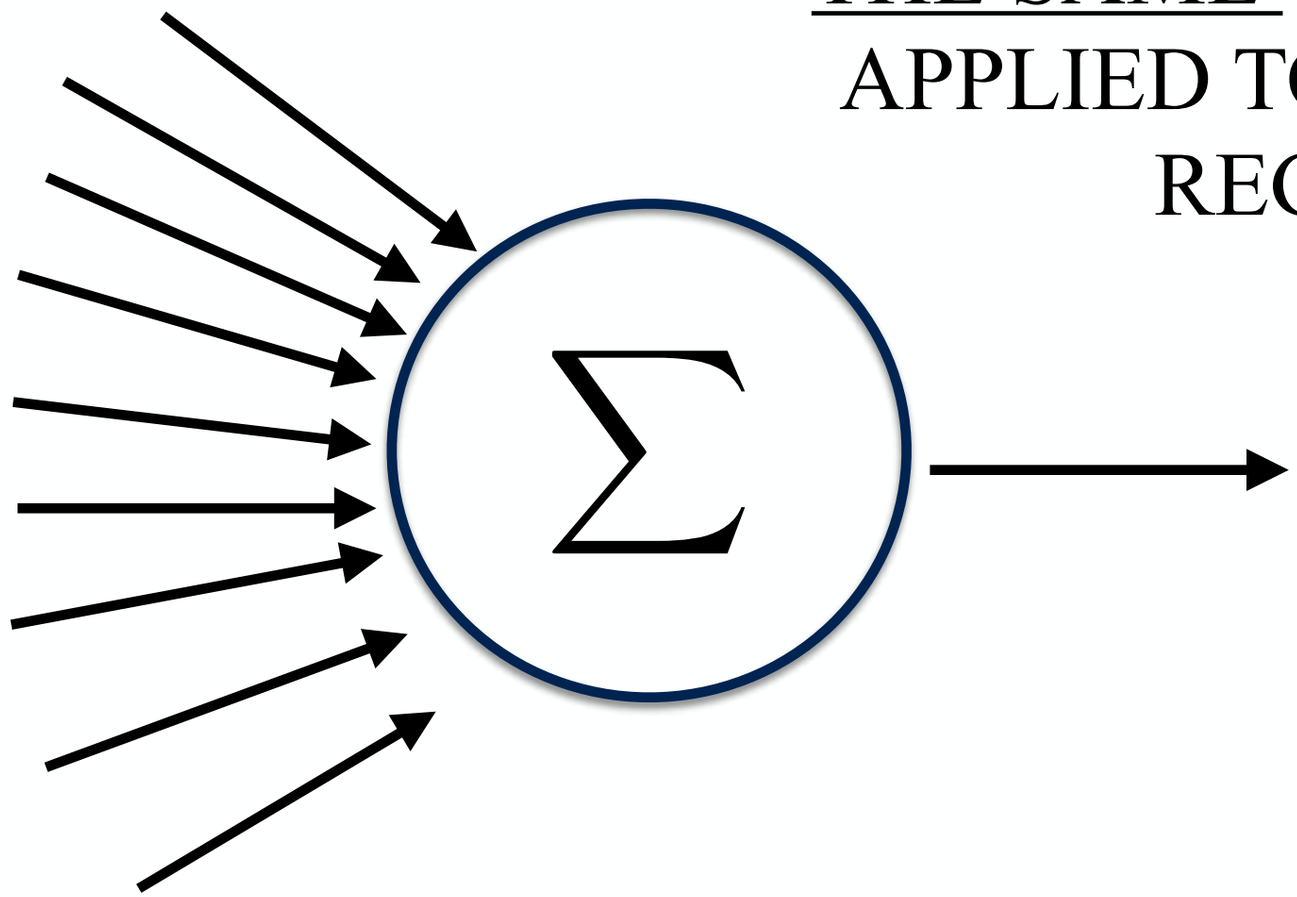
WEIGHTS ARE CODED
IN THE KERNEL

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

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

1.7	1.7	1.7
1.0	1.2	1.8
1.1	0.8	1.3

THE KEY IS AGAIN THAT <u>THE SAME</u> WEIGHTS ARE APPLIED TO ALL IMAGE REGIONS



x_i

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

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

w_i

[weights]

w_i

x_i

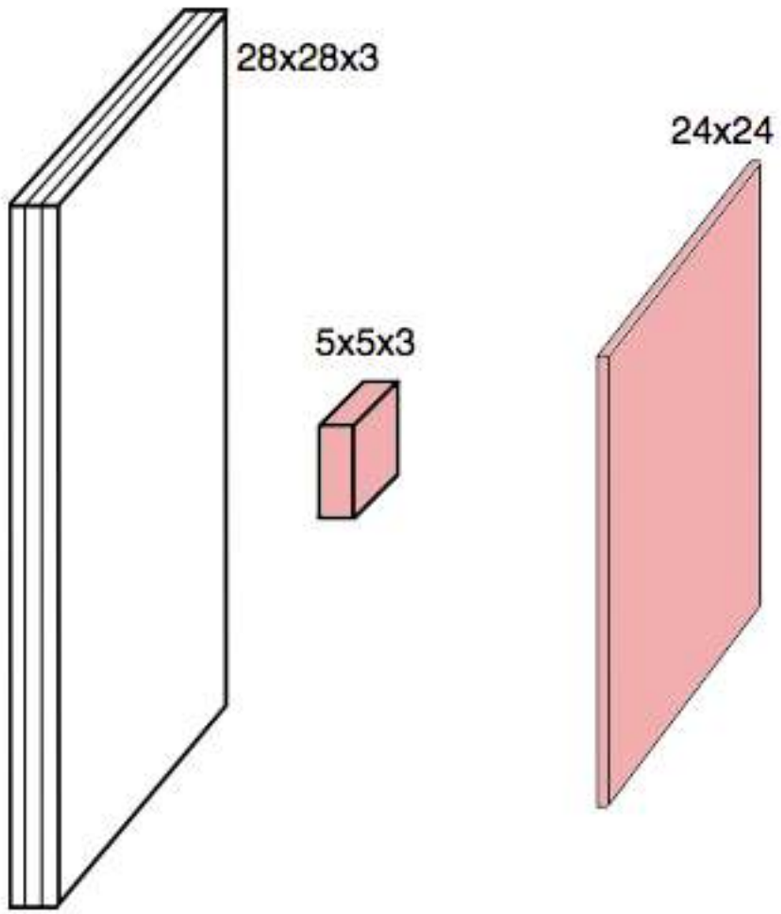
$\sum$

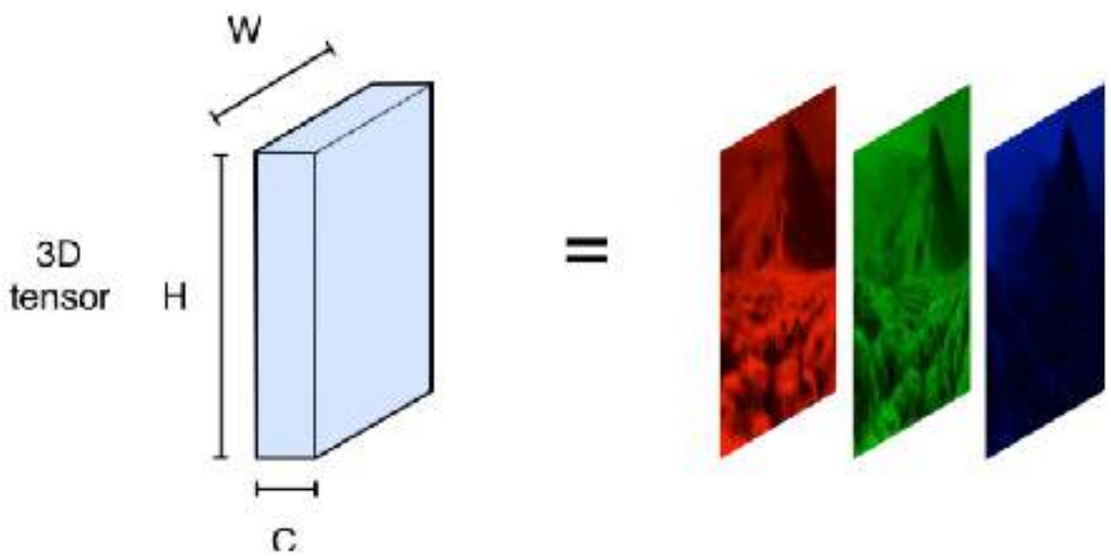
ACTIVATION FUNCTION
AT EVERY KERNEL POSITION

$$z(\mathbf{x}) = relu(w\mathbf{x} + b)$$

CONVOLUTIONS CAN ALSO BE COMPUTED ACROSS CHANNELS (OR COLORS)

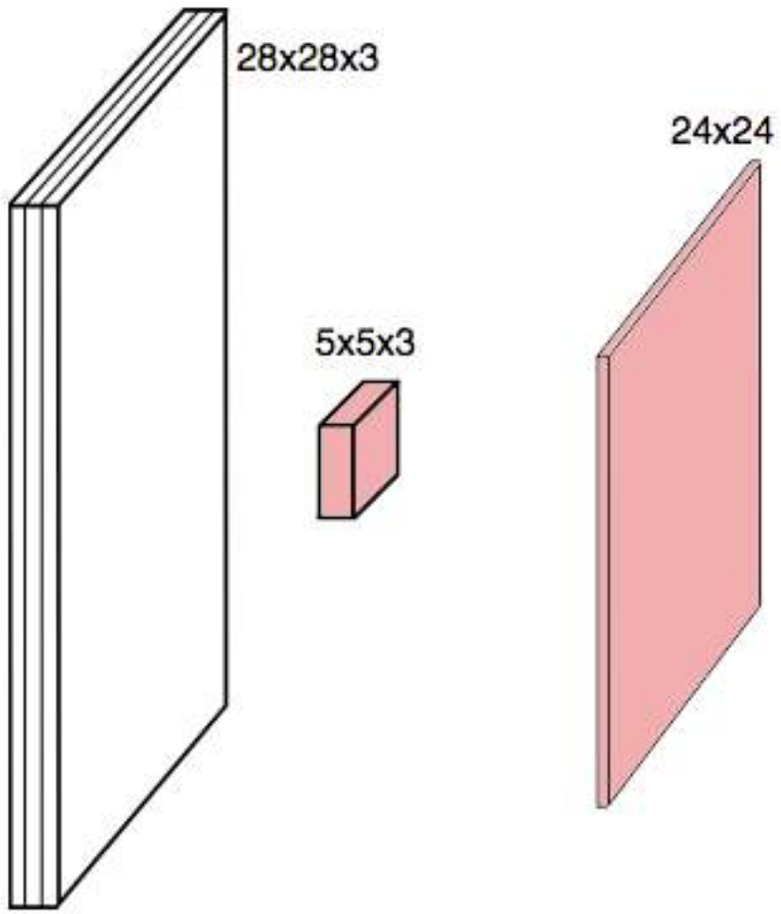
A COLOR IMAGE IS A TENSOR
OF SIZE height x width x channels

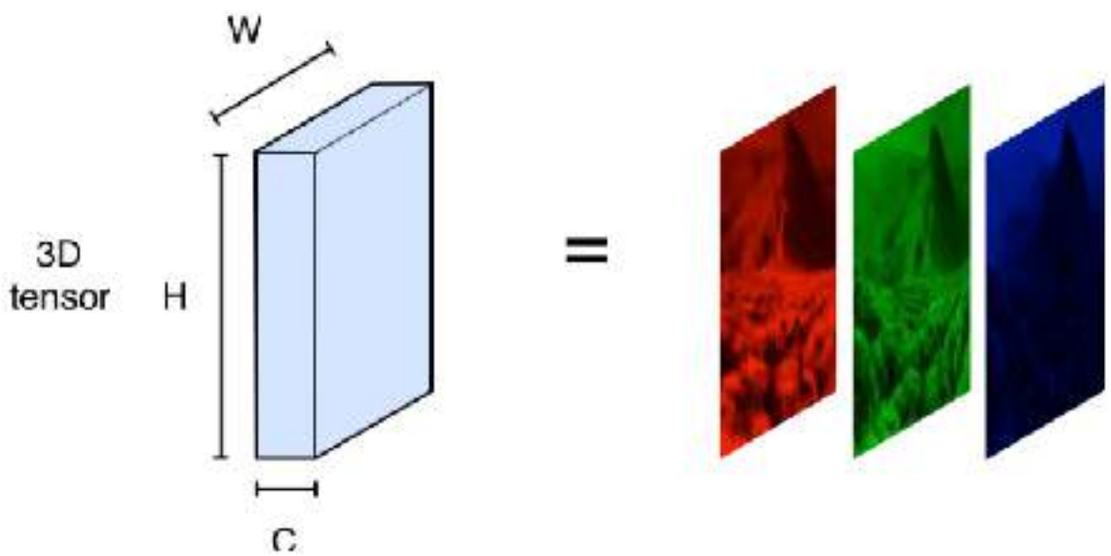




CONVOLUTIONS CAN ALSO BE COMPUTED ACROSS CHANNELS (OR COLORS)

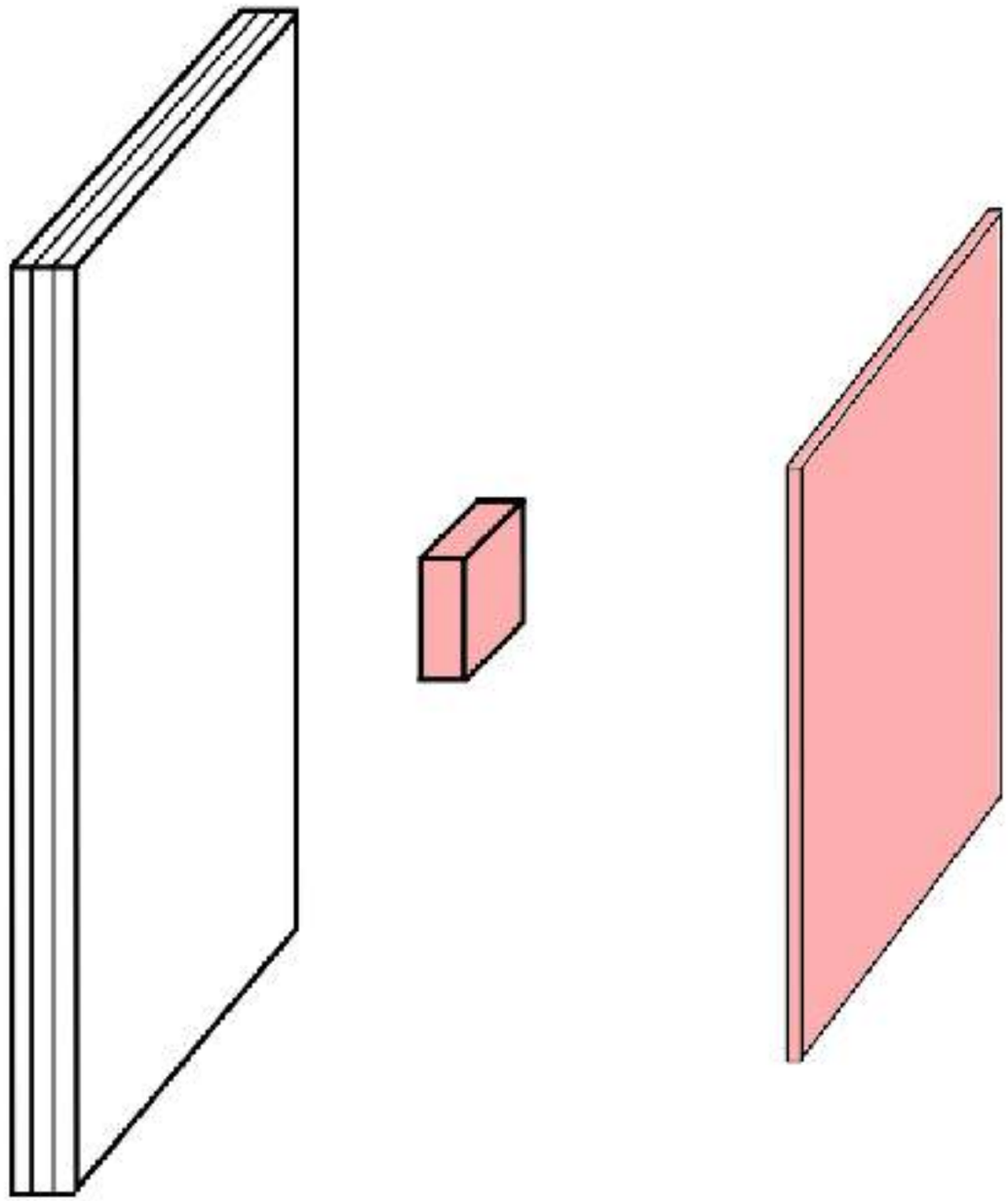
A COLOR IMAGE IS A TENSOR
OF SIZE height x width x channels





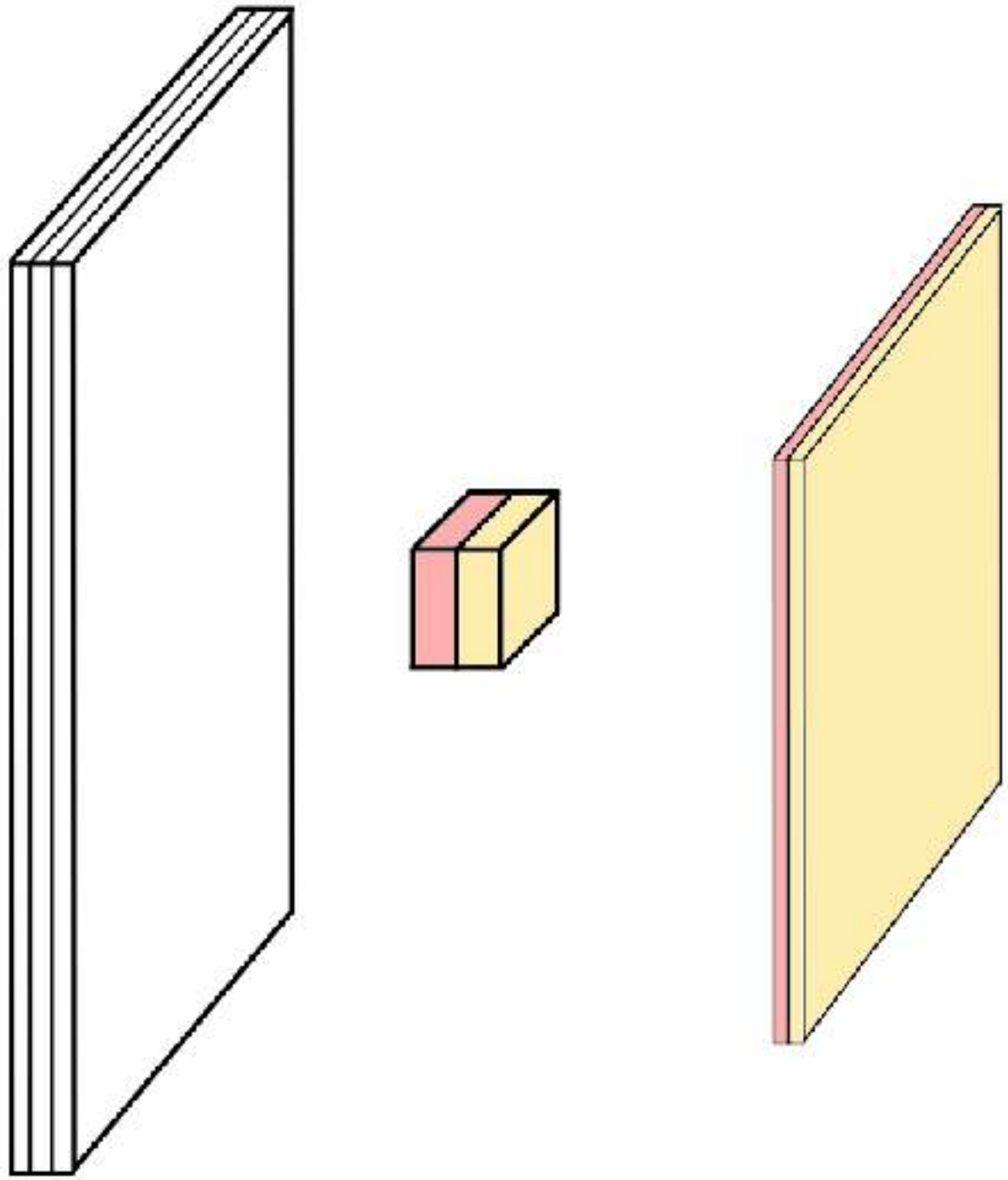
THEN THE KERNEL HAS ALSO 3 CHANNELS

MULTIPLE CONVOLUTIONS WITH DIFFERENT KERNELS CAN BE PERFORMED



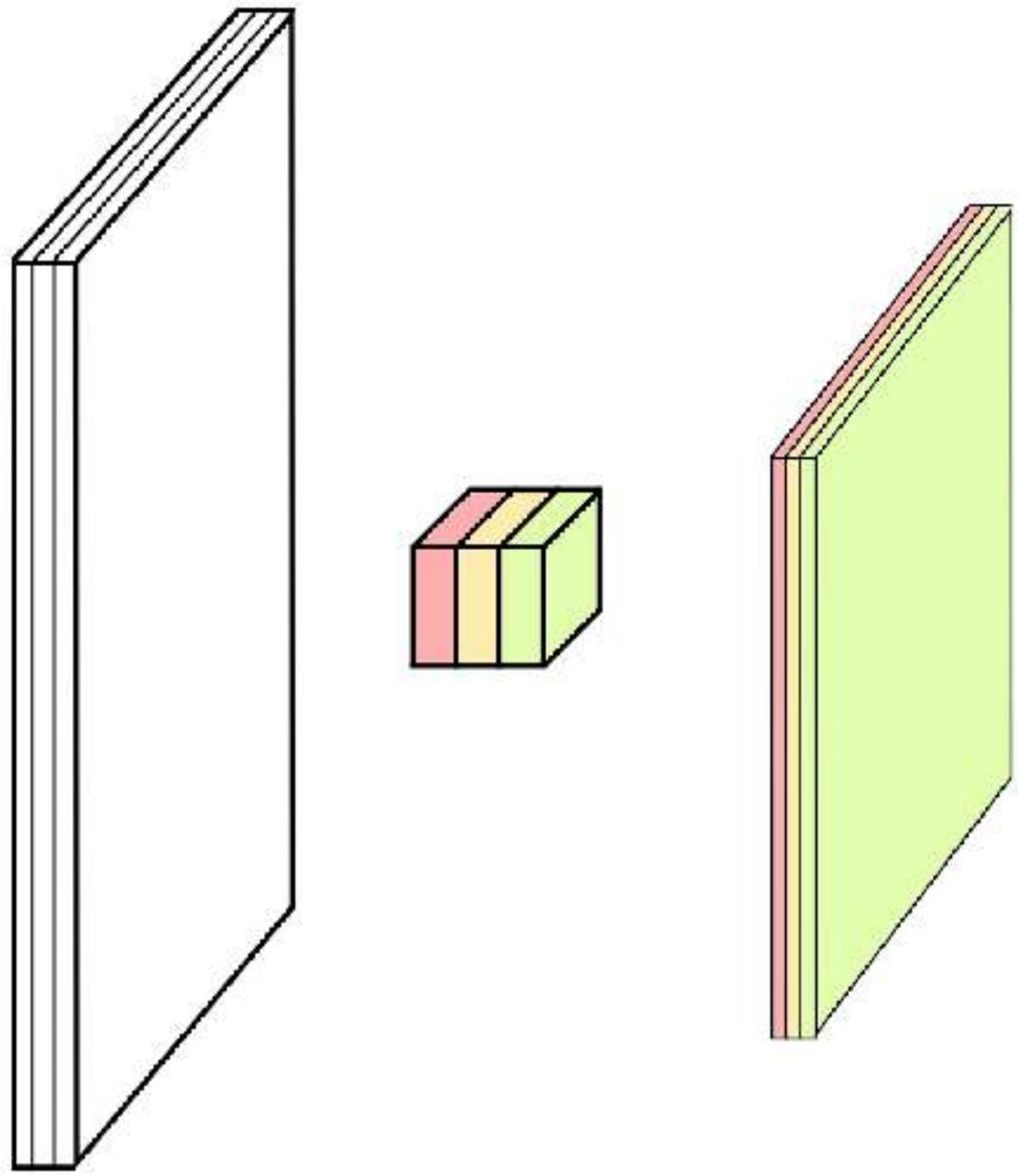
credit

MULTIPLE CONVOLUTIONS WITH DIFFERENT KERNELS CAN BE PERFORMED



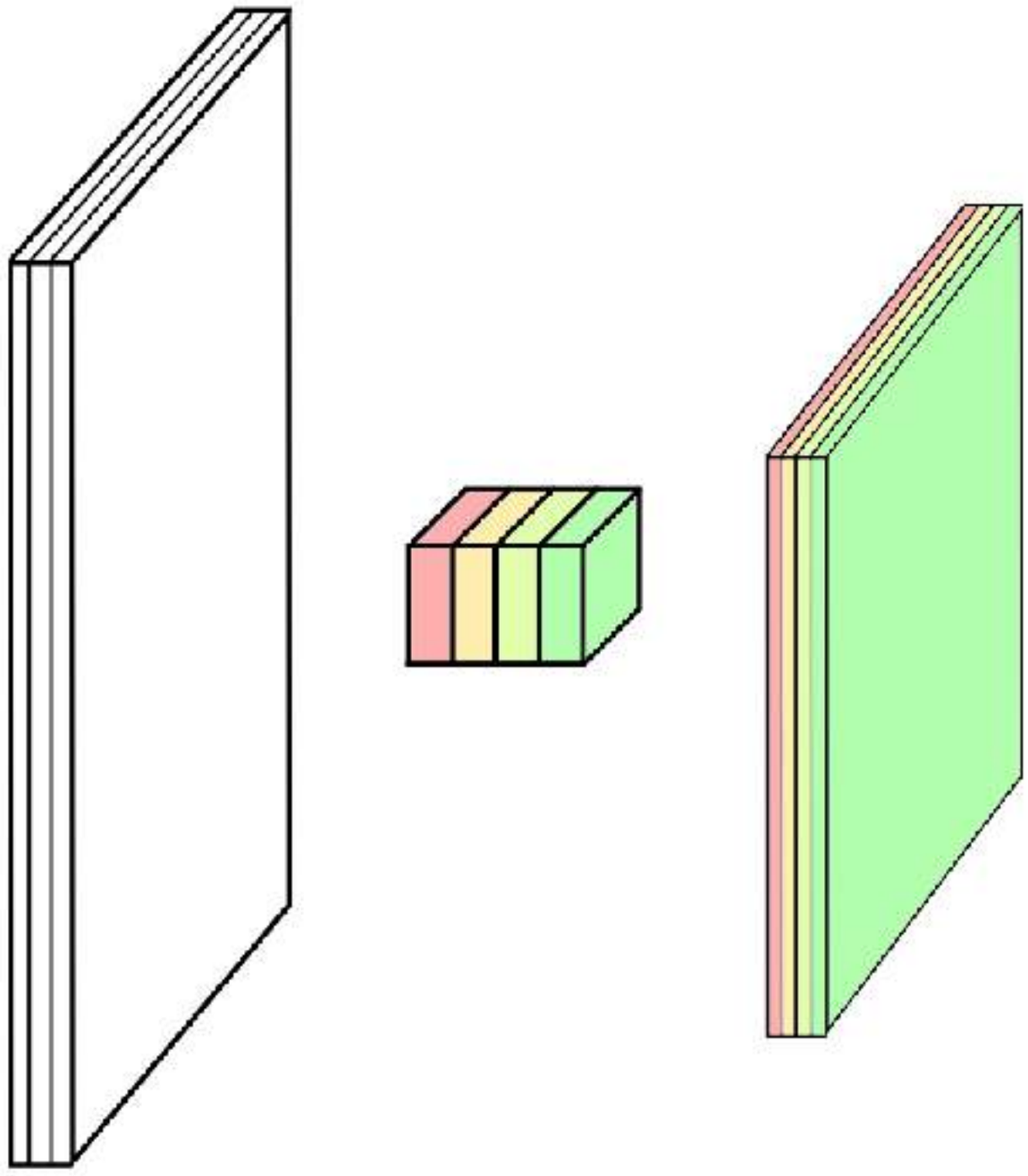
credit

MULTIPLE CONVOLUTIONS WITH DIFFERENT KERNELS CAN BE PERFORMED



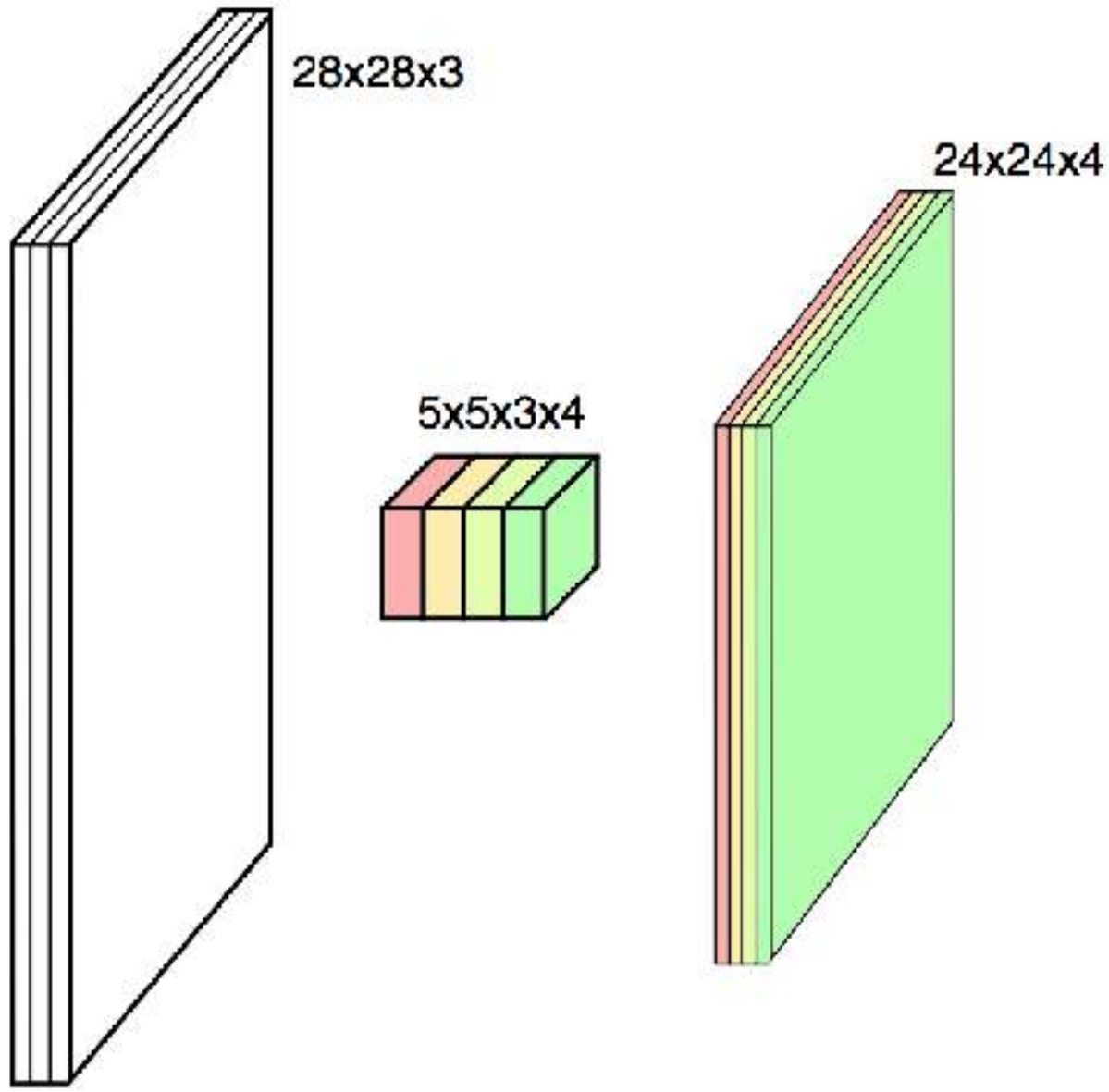
credit

MULTIPLE CONVOLUTIONS WITH DIFFERENT KERNELS CAN BE PERFORMED



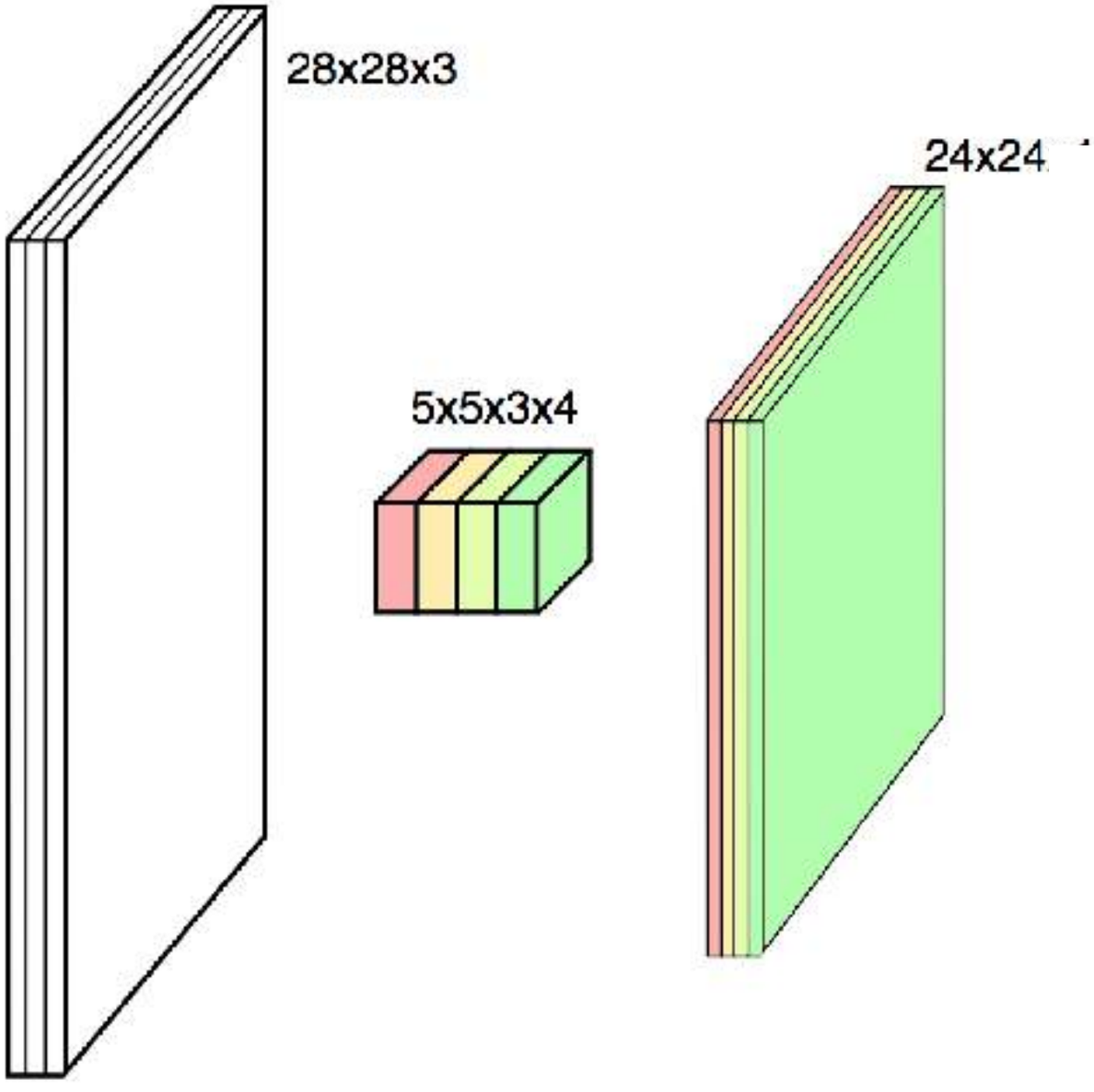
credit

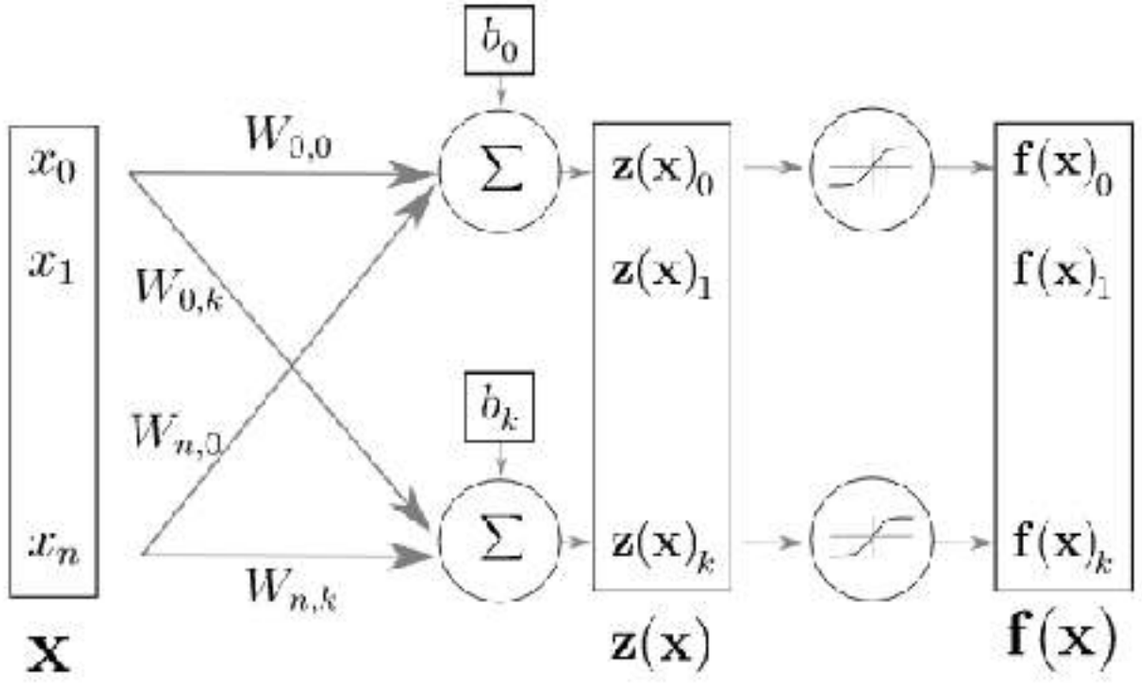
MULTIPLE CONVOLUTIONS WITH DIFFERENT KERNELS CAN BE PERFORMED



credit

MULTIPLE CONVOLUTIONS WITH DIFFERENT KERNELS CAN BE PERFORMED

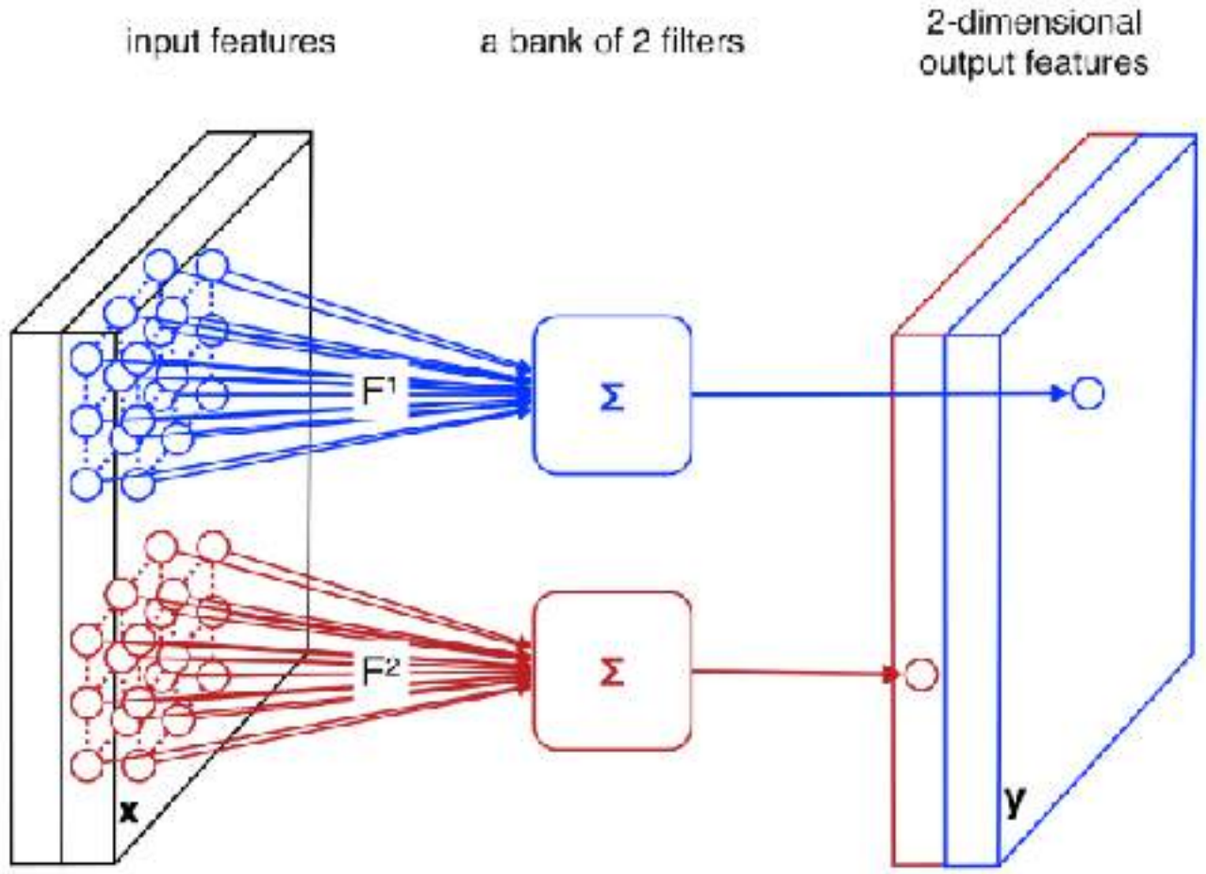




credit

SINCE CONVOLUTIONS OUTPUT ONE SCALAR< THEY CAN BE SEEN AS AN INDIVIDUAL NEURON WITH A RECEPTIVE FIELD LIMITED TO THE KERNEL DIMENSIONS

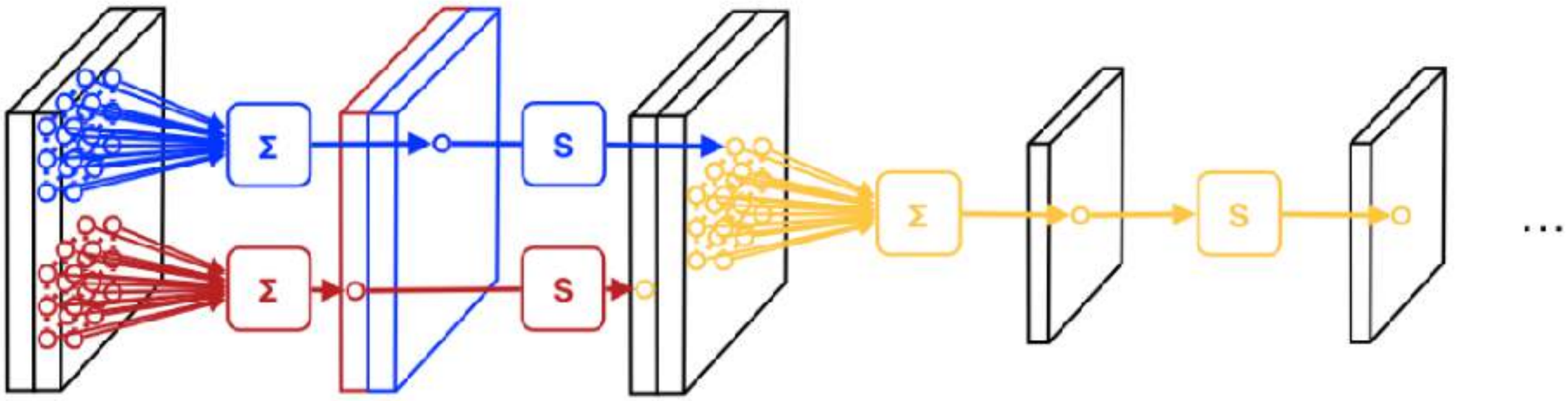
THE SAME NEURON IS FIRED WITH DIFFERENT AREAS FROM THE INPUT



Credit

DOWNSAMPLING

DOWNSAMPLING IS APPLIED TO REDUCE THE OVERALL SIZE OF TENSORS



POOLING

CONVOLUTIONS ARE OFTEN FOLLOWED BY AN OPERATION OF DOWNSAMPLING [POOLING]

VERY SIMPLE OPERATION - ONLY ONE OUT OF EVERY N PIXELS ARE KEPT

OFTEN MATCHED WITH AN INCREASE OF THE FEATURE CHANNELS

TYPES OF POOLING

SUM POOLING $$y = \sum x_{uv}$$

SQUARE SUM POOLING $$y = \sqrt{\sum x_{uv}^2}$$

MAX POOLING $$y = max(x_{uv})$$

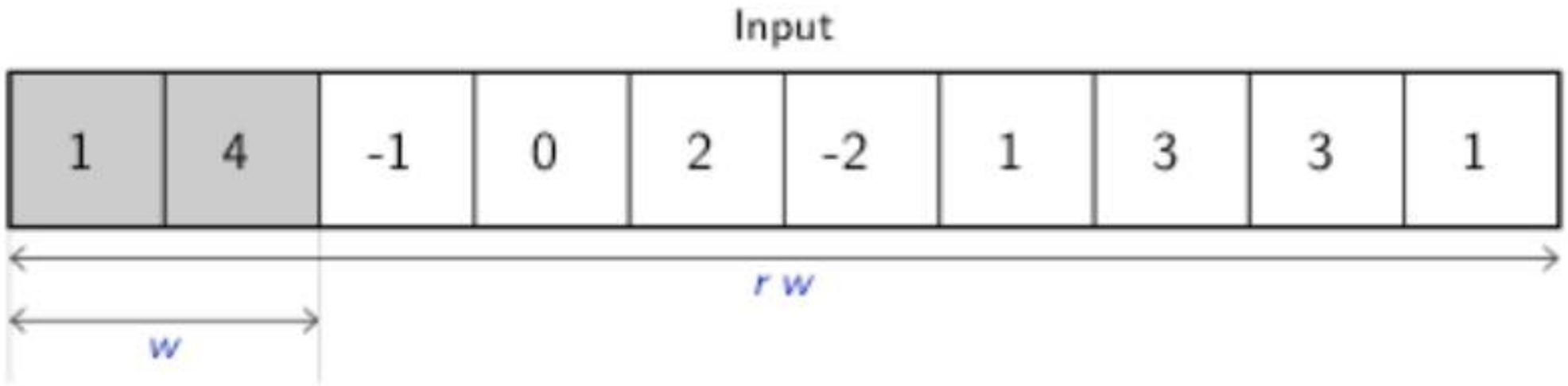
TYPES OF POOLING

SUM POOLING $$y = \sum x_{uv}$$

SQUARE SUM POOLING $$y = \sqrt{\sum x_{uv}^2}$$

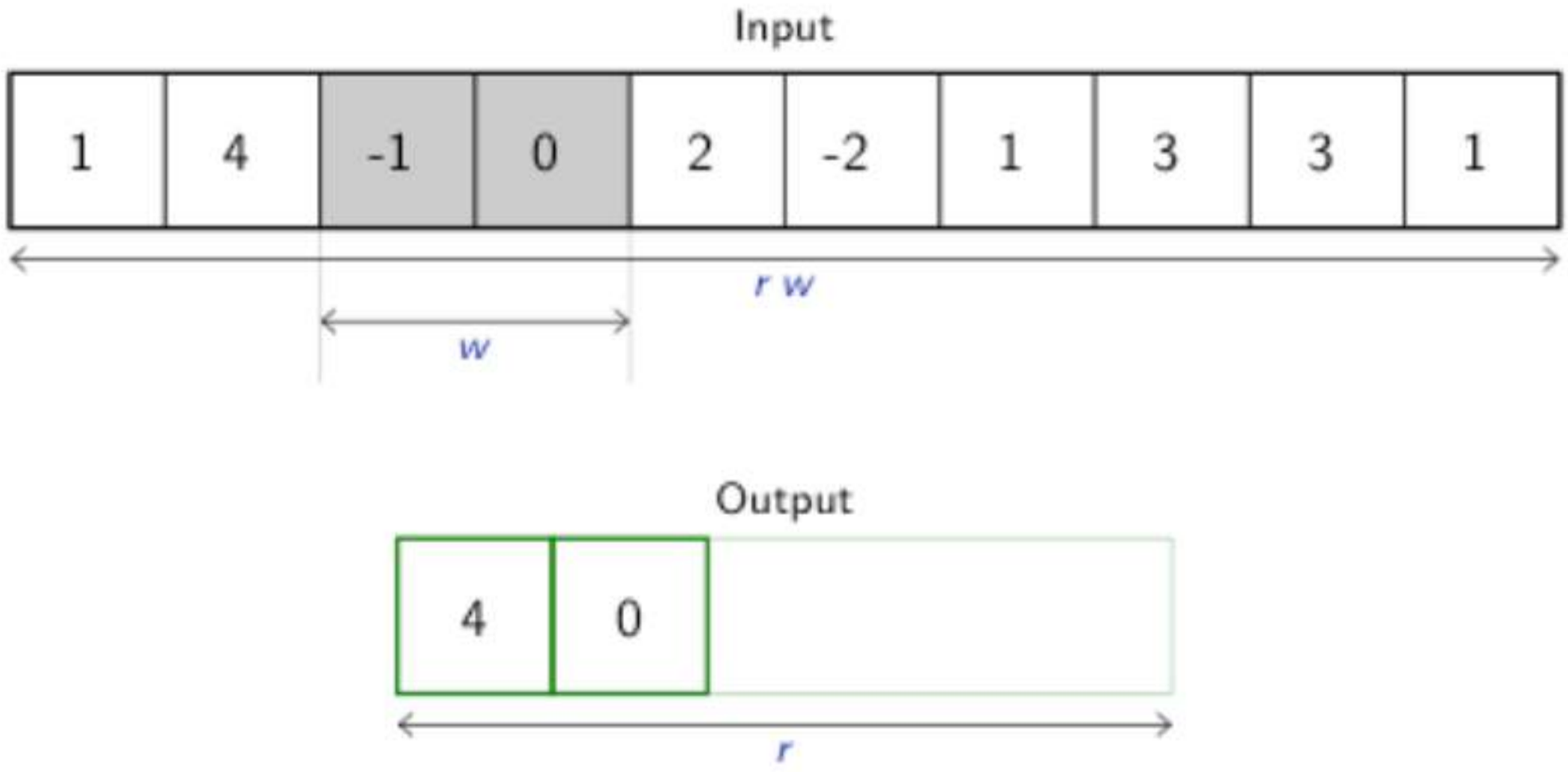
MAX POOLING $$y = max(x_{uv})$$

MAX POOLING 1D



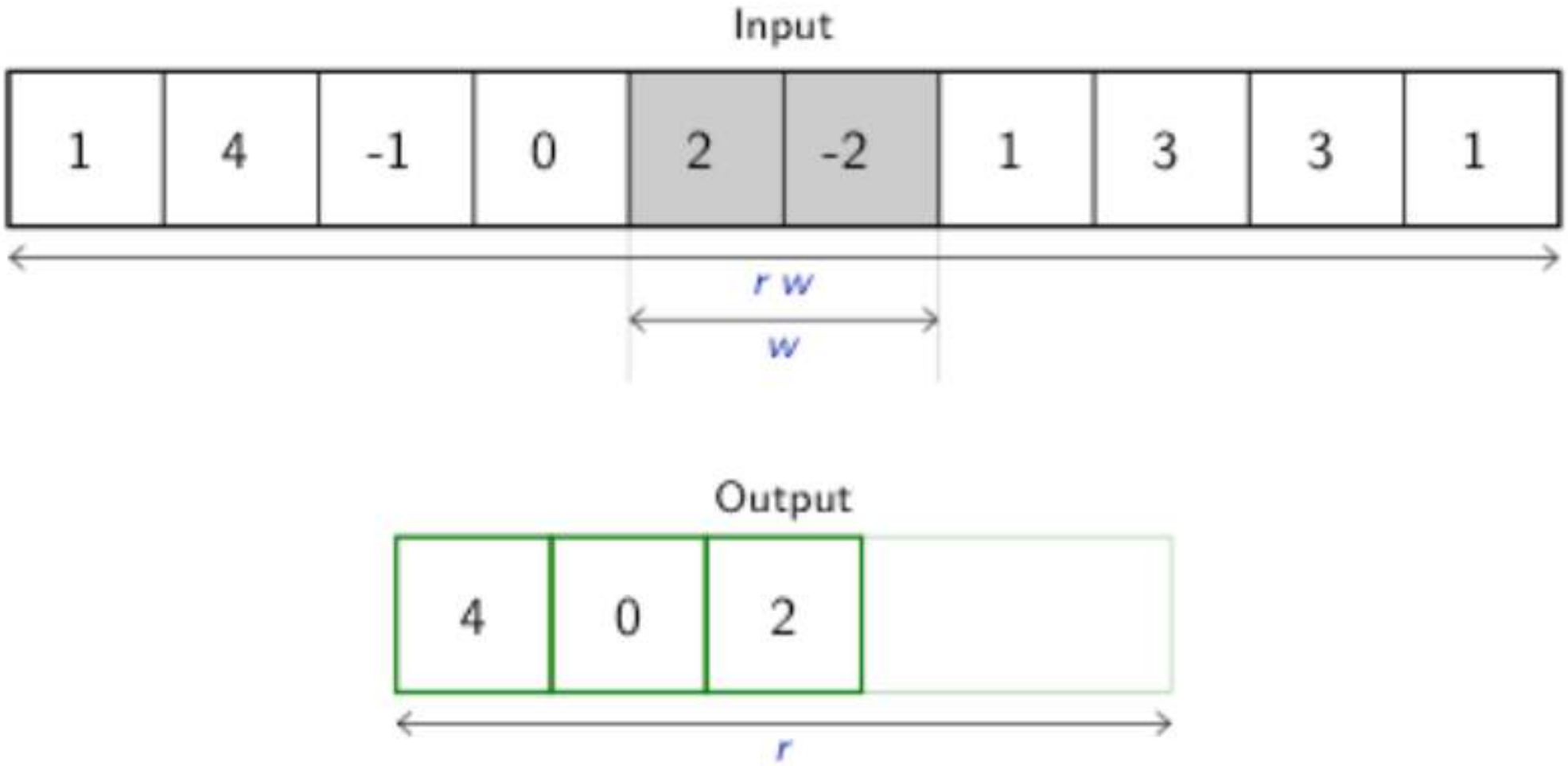
Credit: F. Fleuret

MAX POOLING 1D



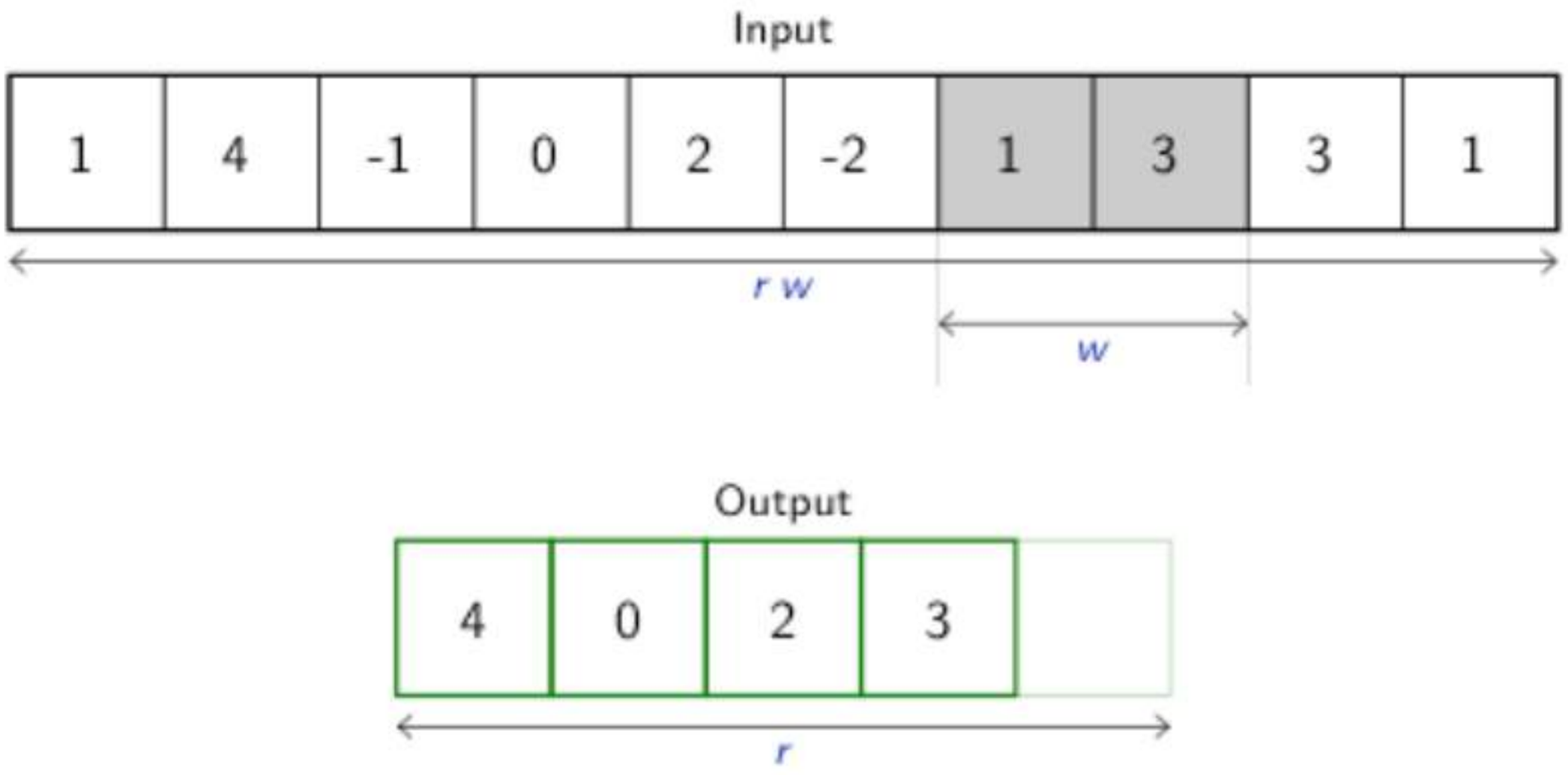
Credit: F. Fleuret

MAX POOLING 1D



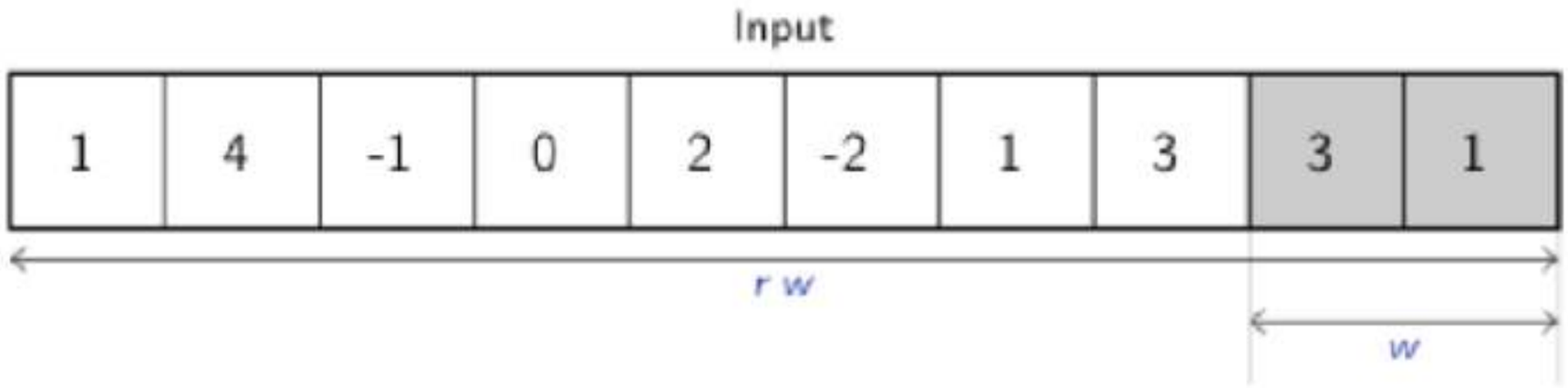
Credit: F. Fleuret

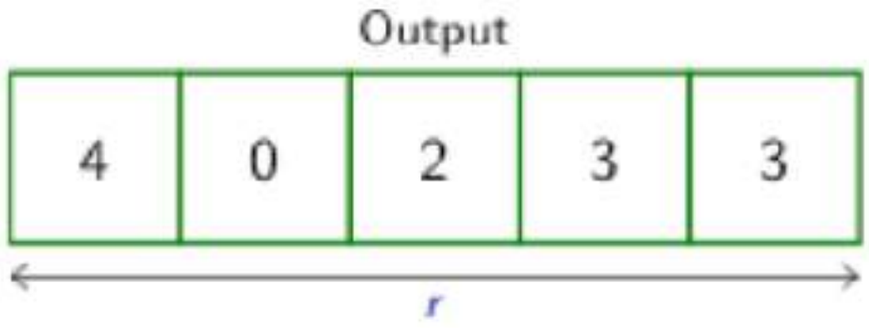
MAX POOLING 1D



Credit: F. Fleuret

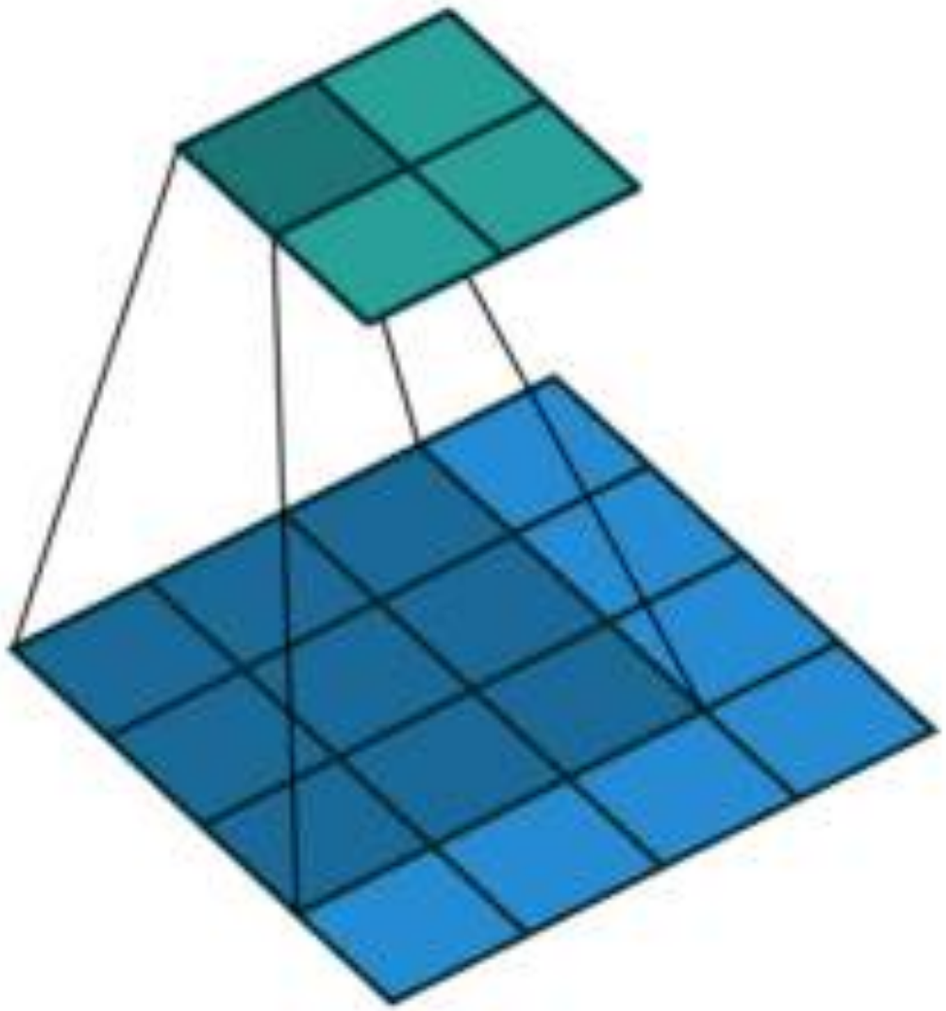
MAX POOLING 1D



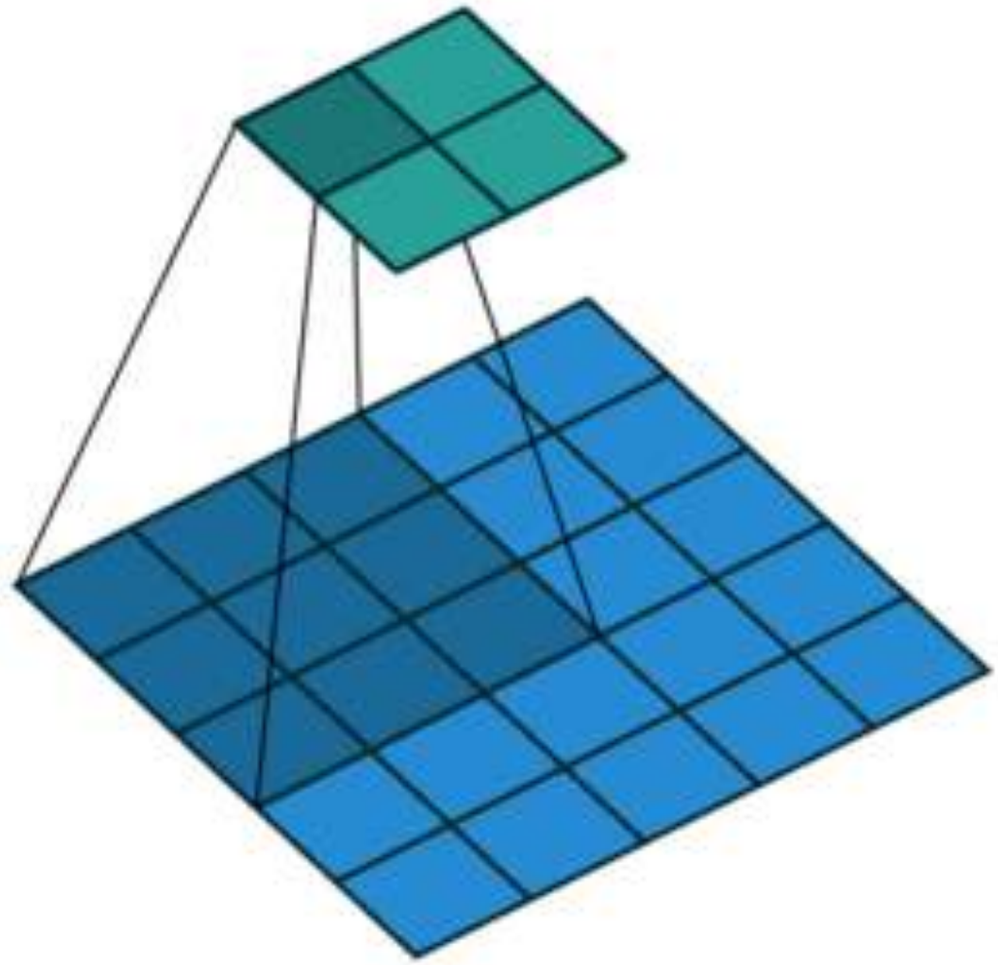


Credit: F. Fleuret

OPTIONS: **STRIDES**

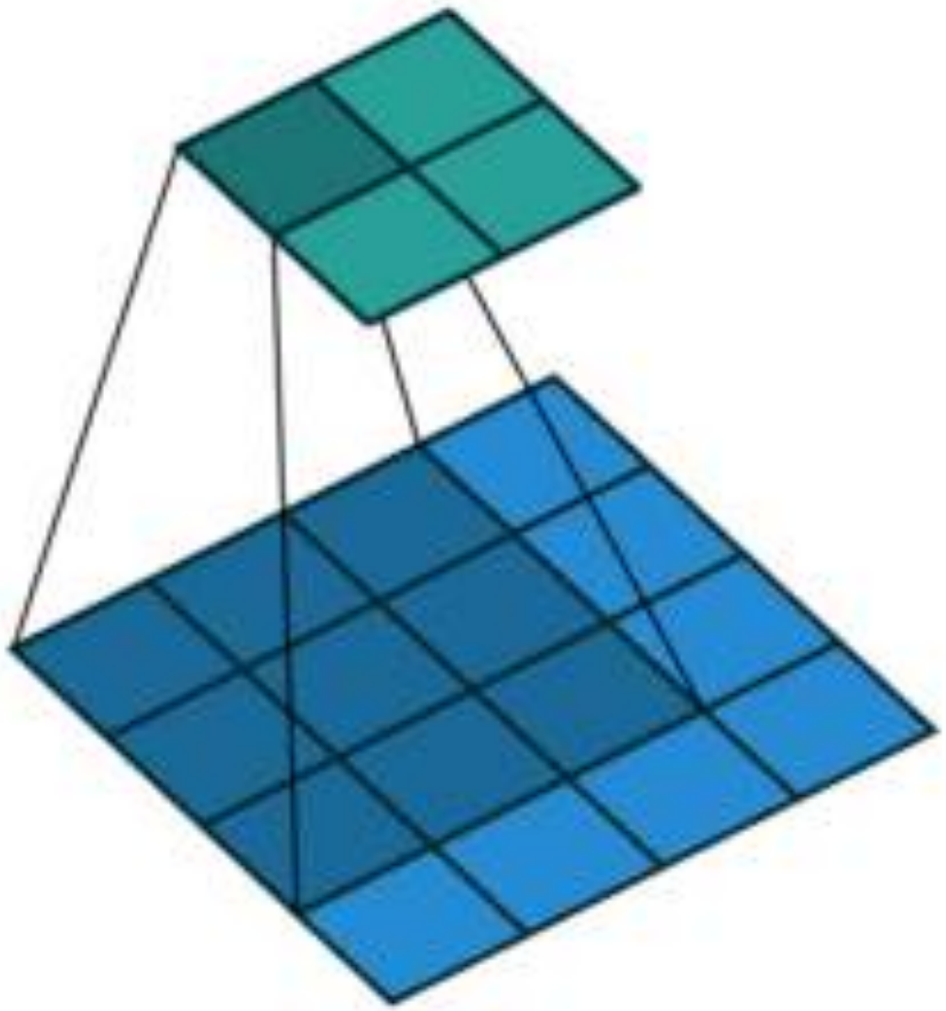


NO STRIDES

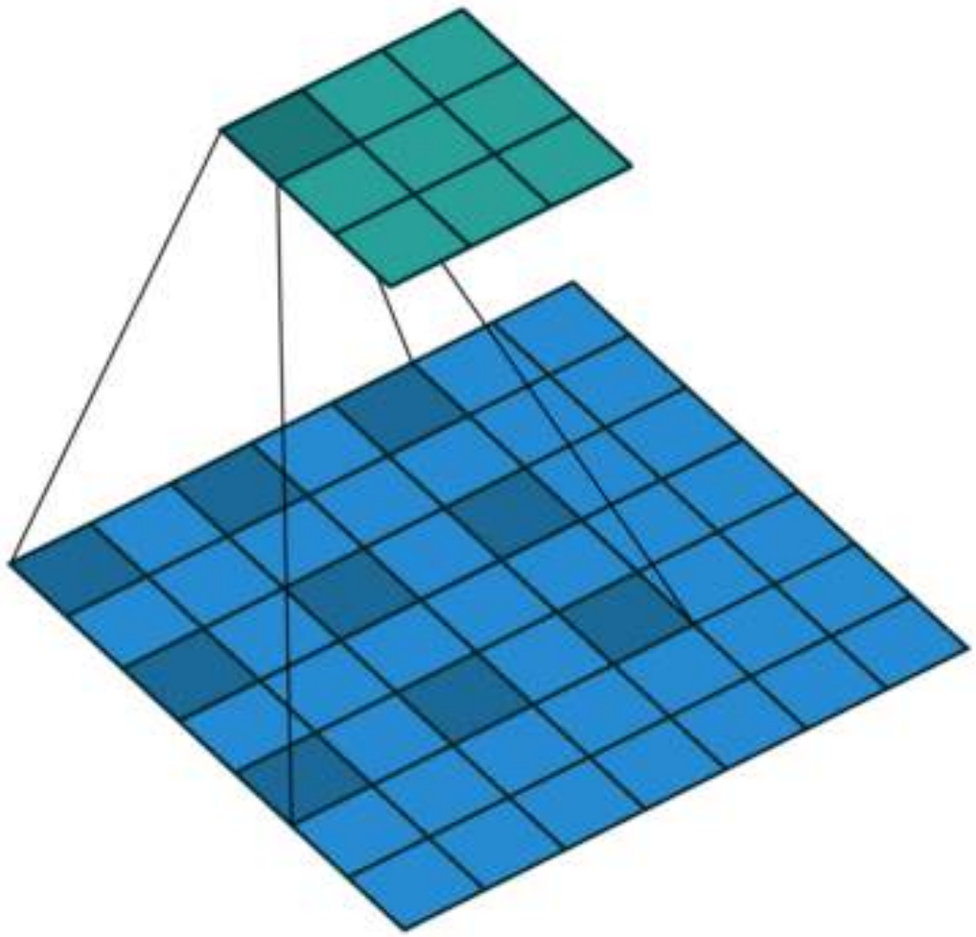


STRIDES

OPTIONS: **DILATION**

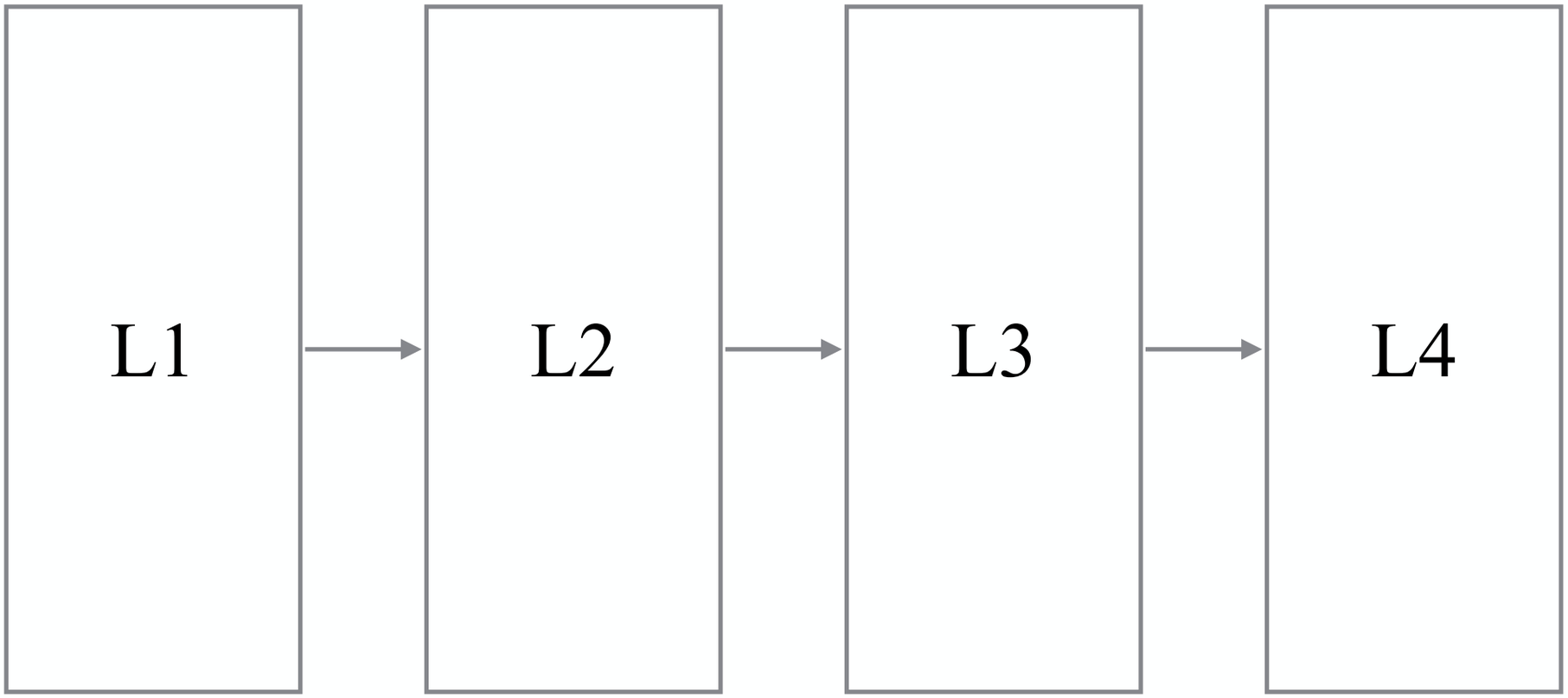


NO STRIDES



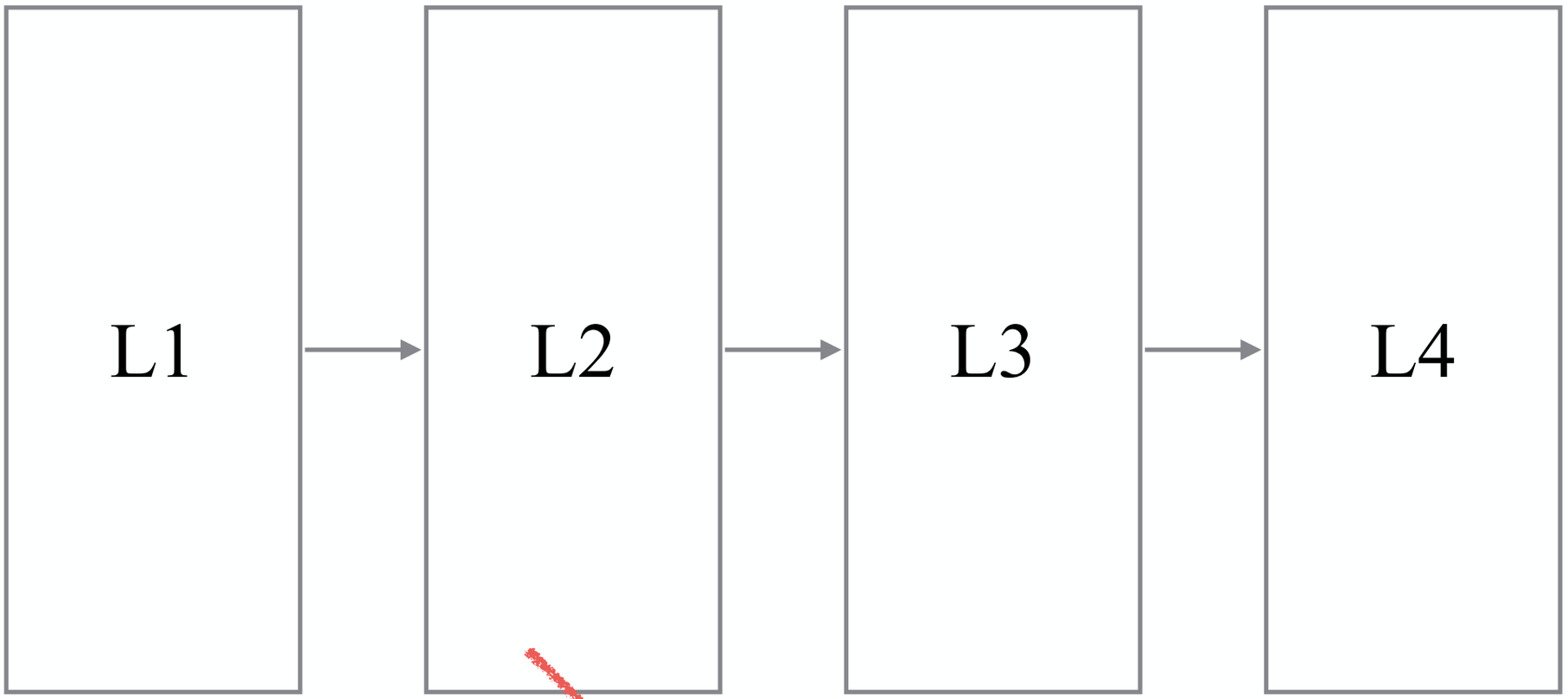
DILATION

CONVNET OR CNN



A CONCATENATION OF MULTIPLE CONVOLUTIONAL BLOCKS

CONVNET OR CNN

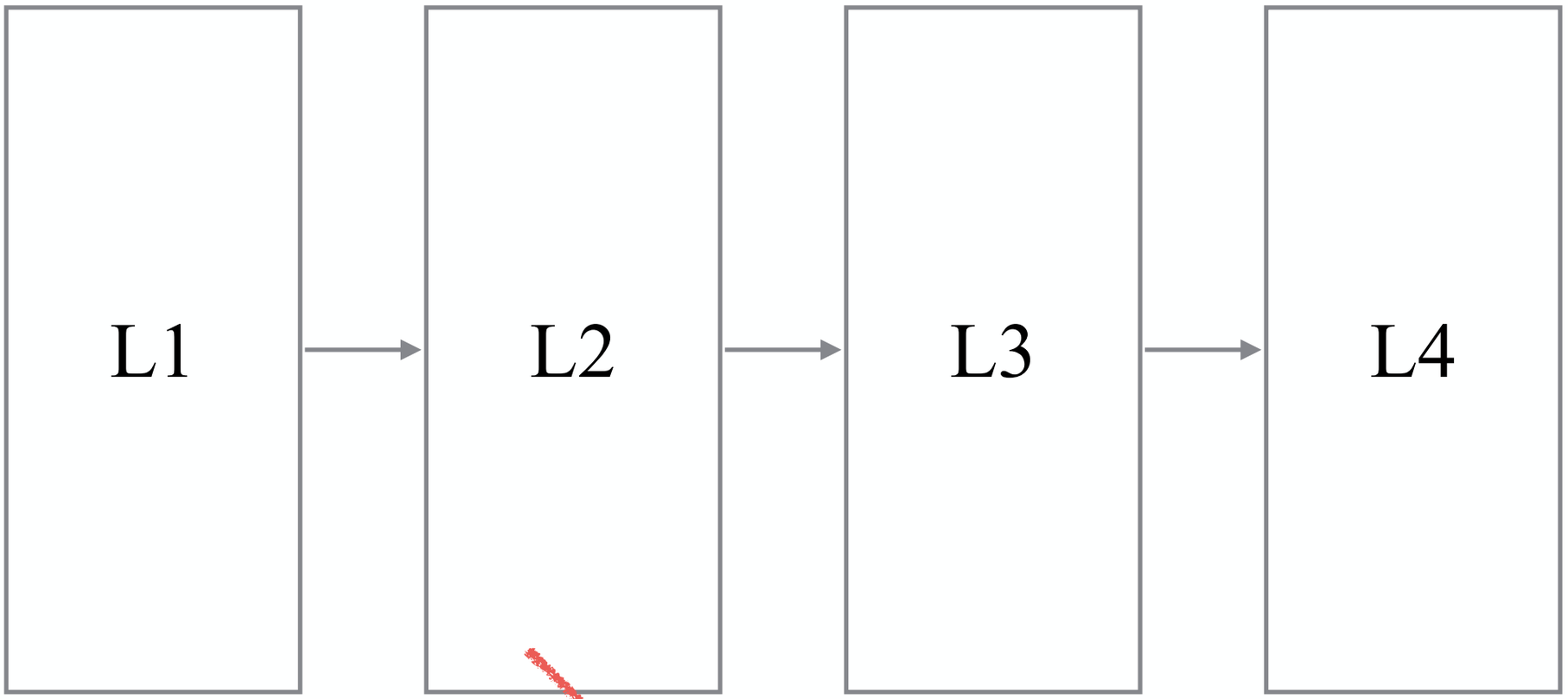


EACH BLOCK TYPICALLY MADE OF:

CONV | ACTIVATION | POOLING

(+dropout for training)

CONVNET OR CNN

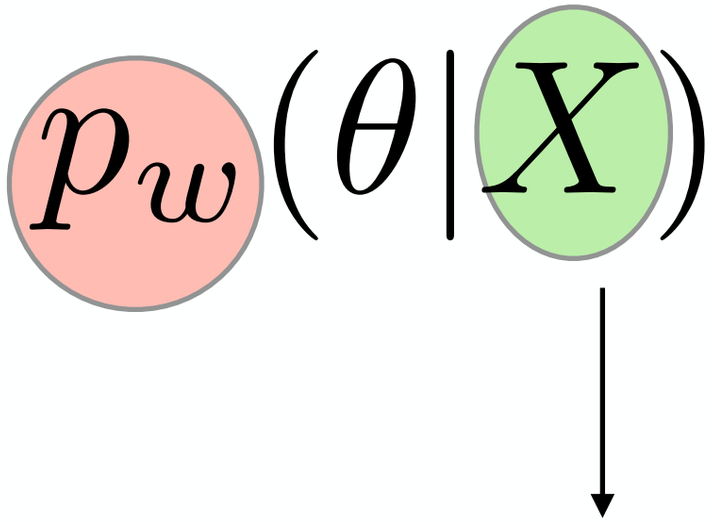


EACH BLOCK TYPICALLY MADE OF:

CONV | ACTIVATION | POOLING

(+dropout for training)

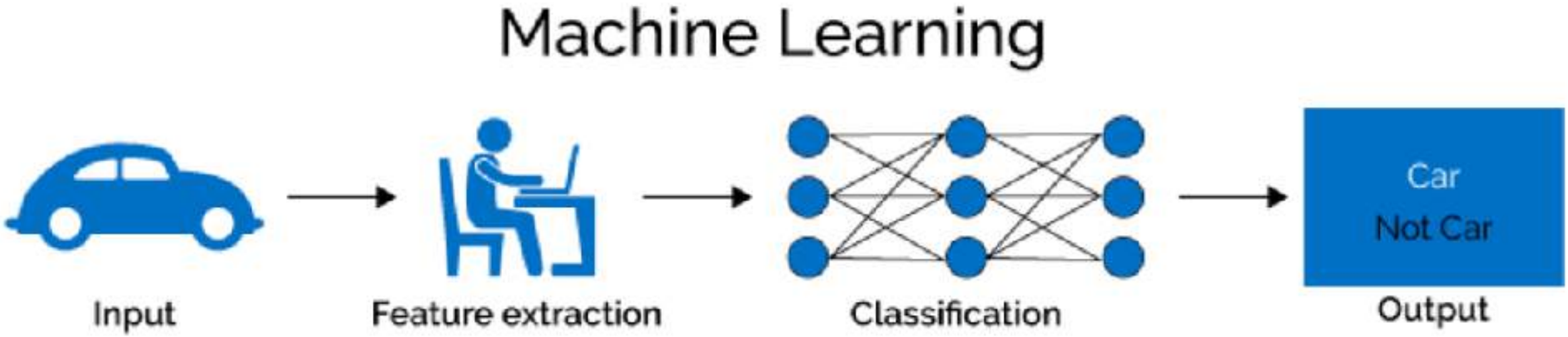
DEEP LEARNING

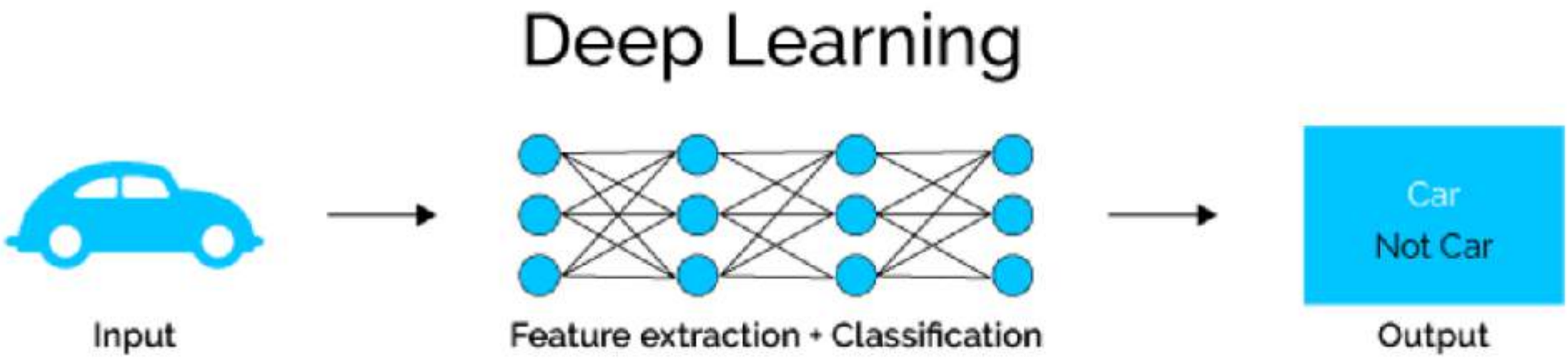


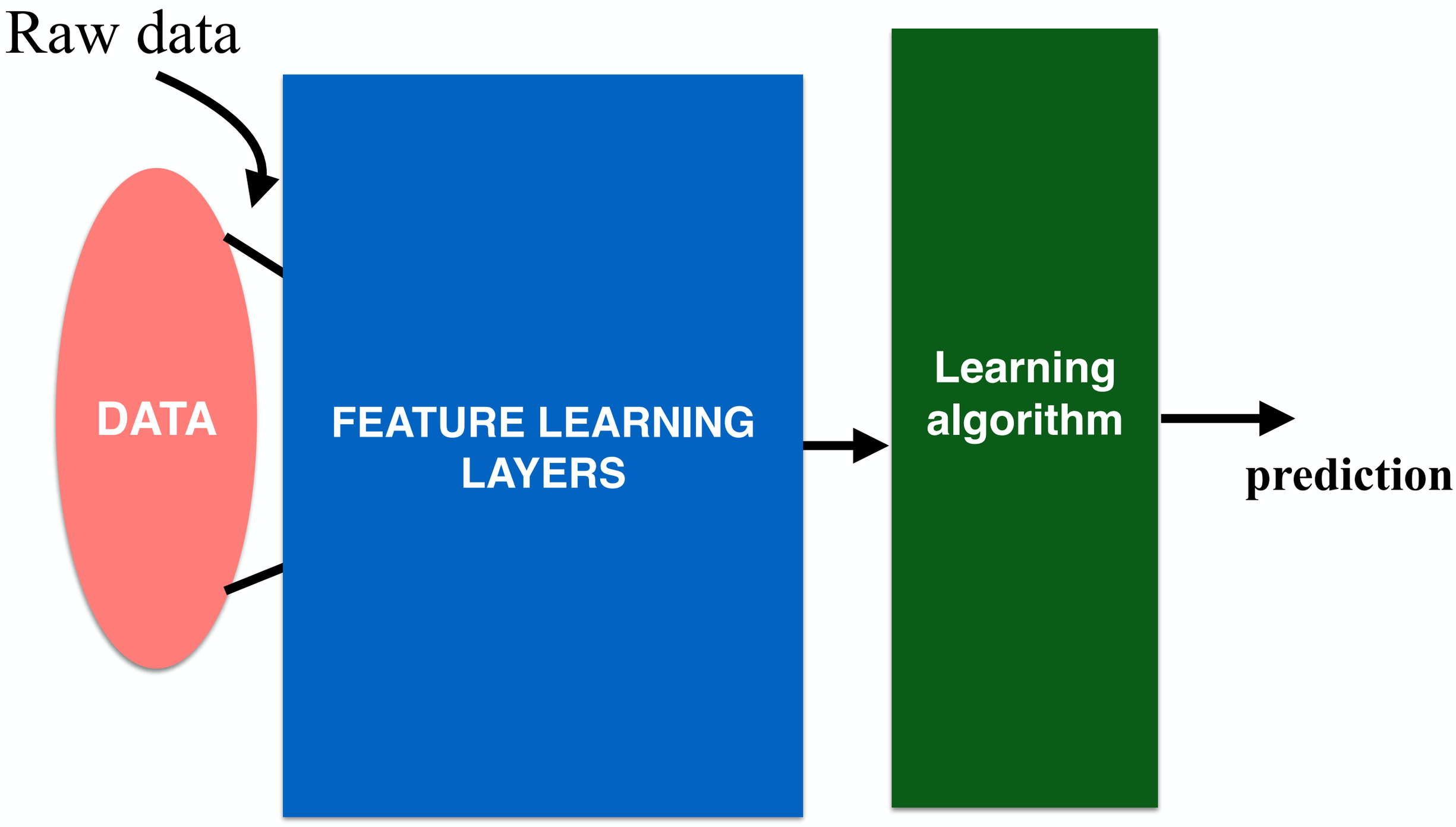
LET THE NETWORK FIGURE THIS OUT (“unsupervised feature extraction”)

LET’S GO A STEP FORWARD INTO LOOSING CONTROL….

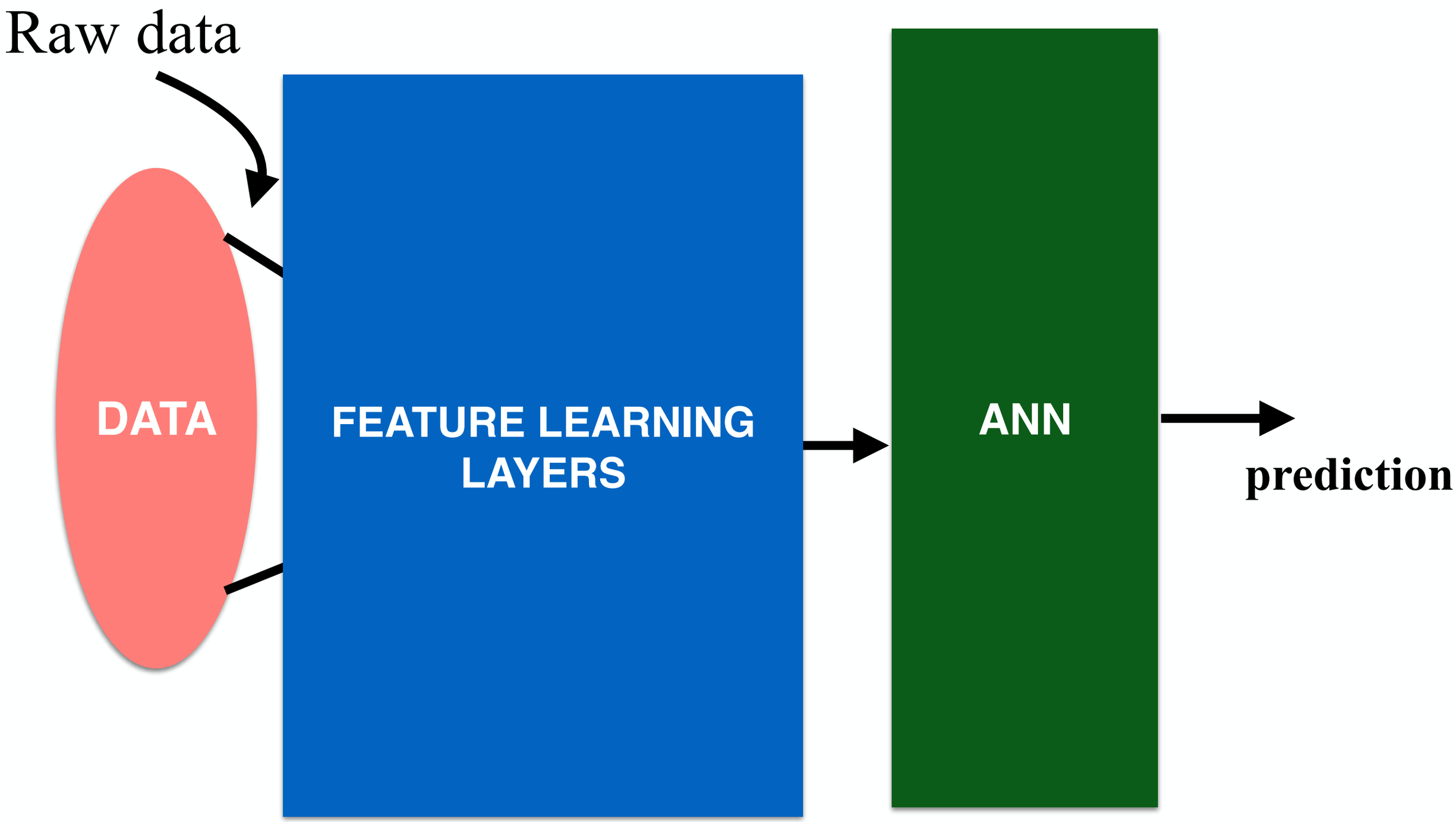
THIS IS A CHANGE OF PARADIGM!



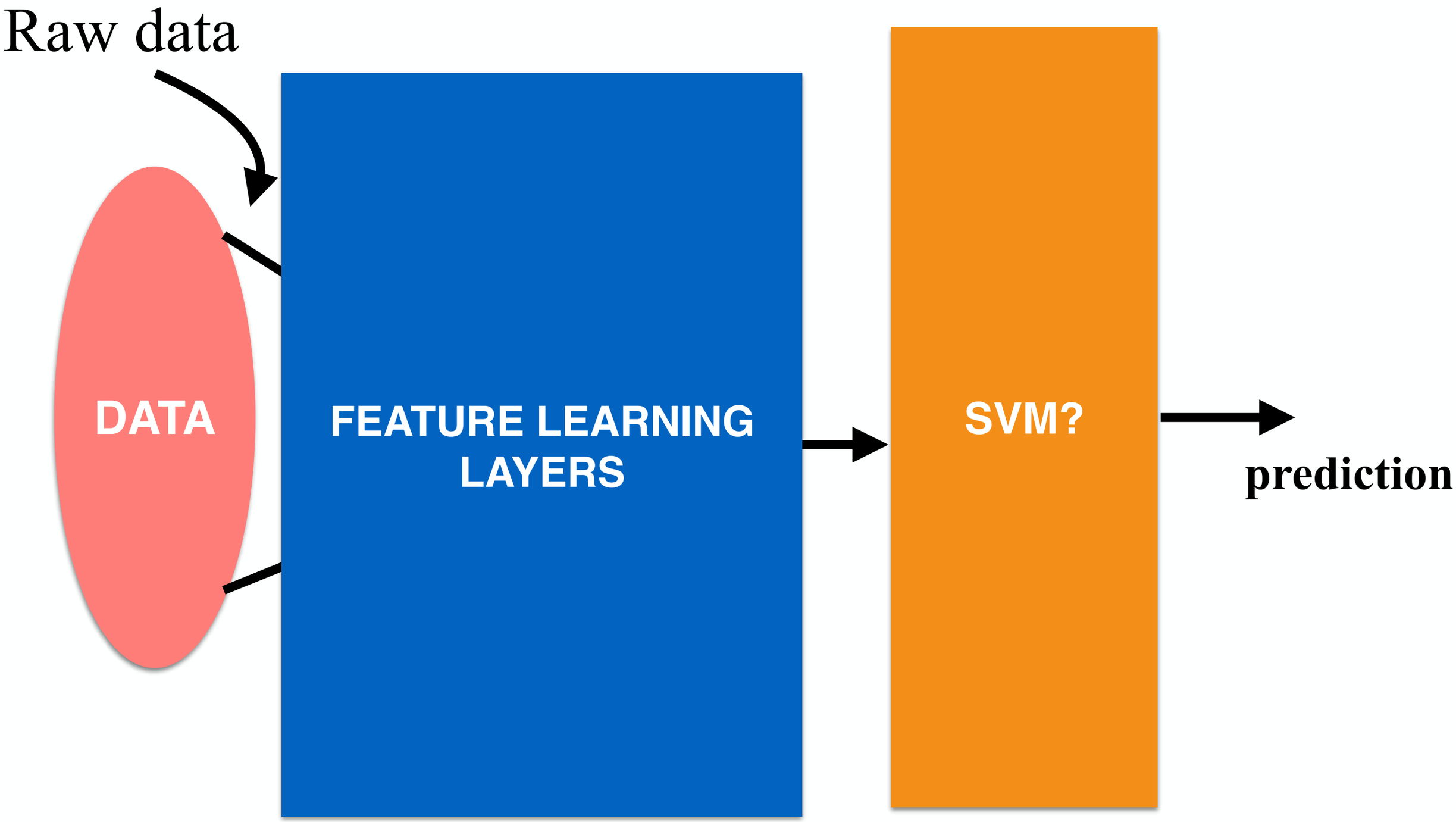



Raw data
DATA
FEATURE LEARNING LAYERS
Learning algorithm
prediction

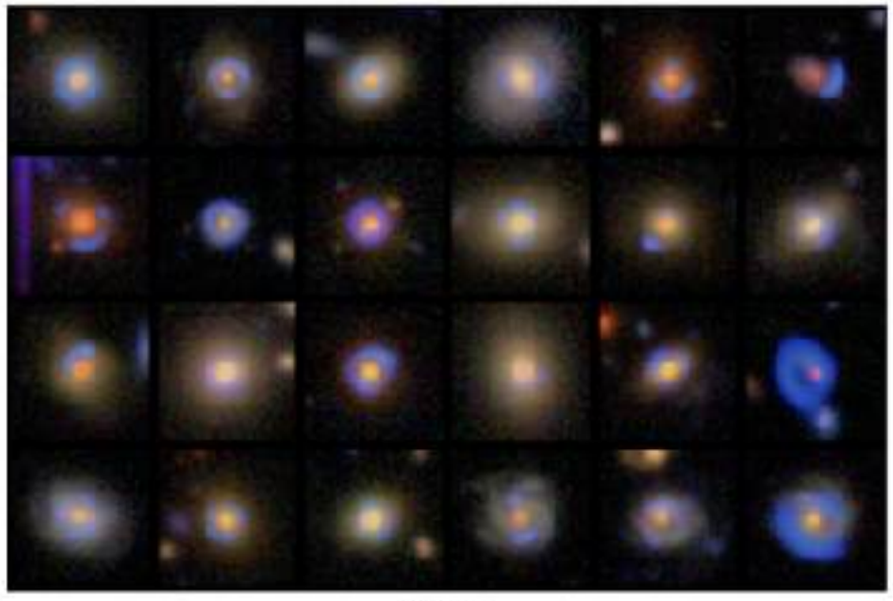
THE LEARNING ALGORITHM CAN BE CHANGED



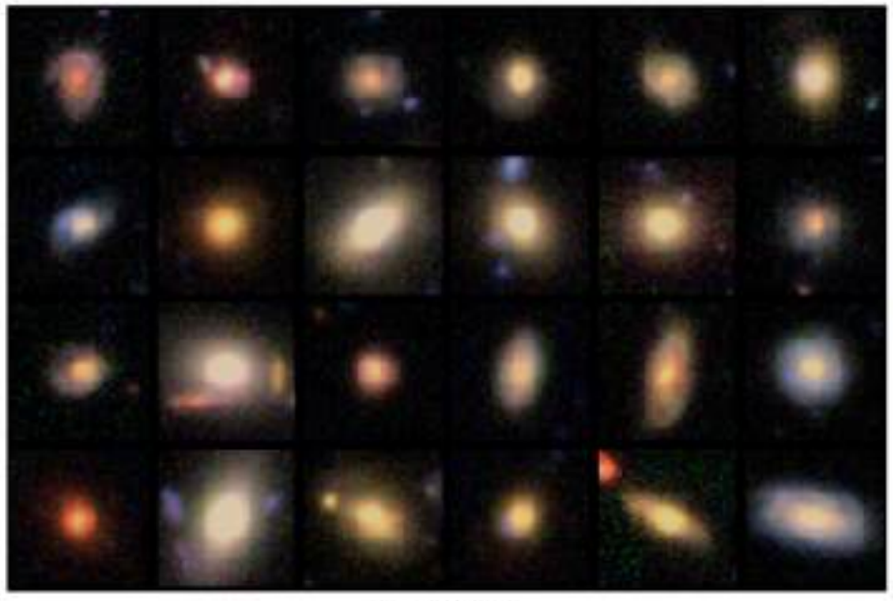
THE LEARNING ALGORITHM CAN BE CHANGED



1. Classification



LENS



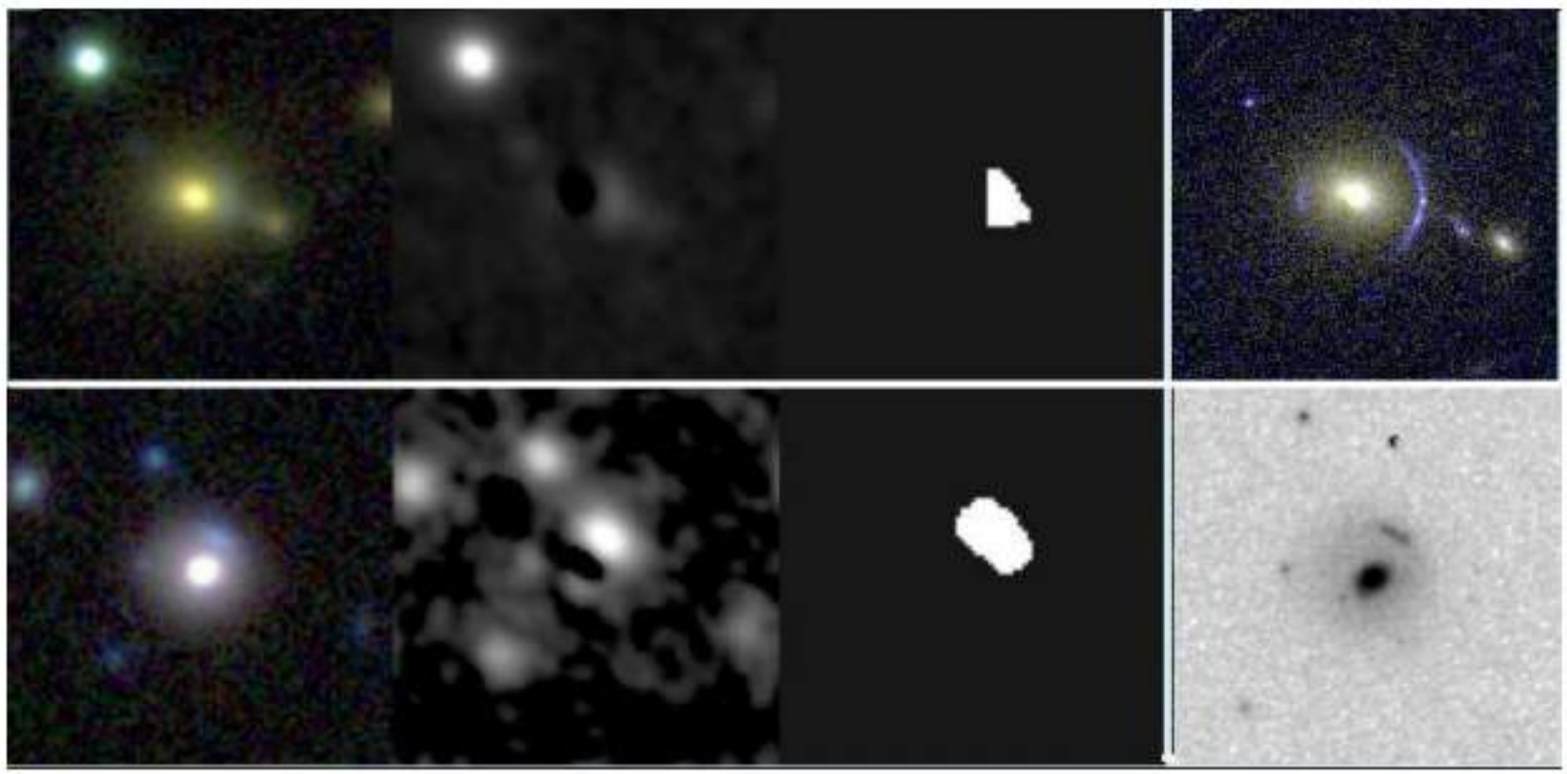
NON-LENS

Detection of Strong Lenses
Valuable information of Dark Matter properties

Future surveys will increase the samples by orders of magnitude.

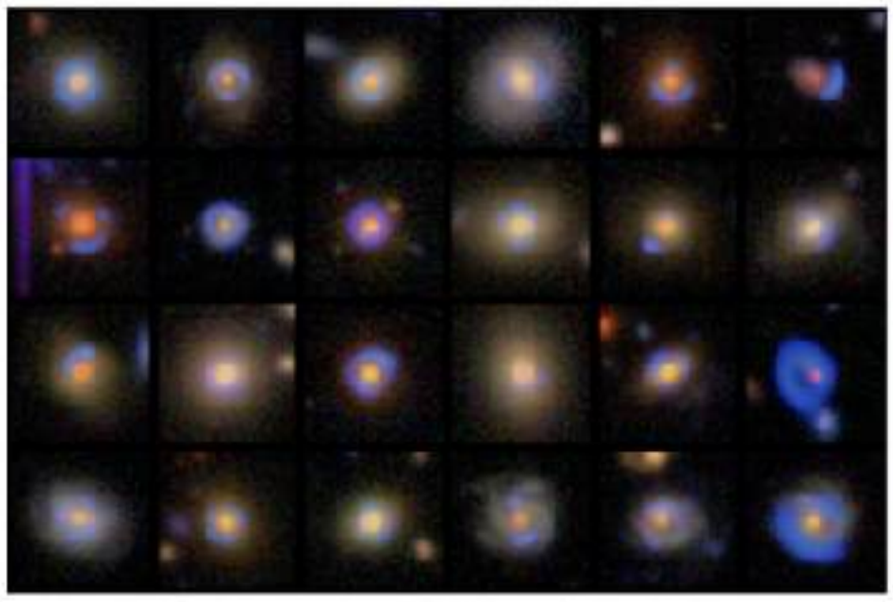
Jacobs+17

"Pre Deep Learning" Approach

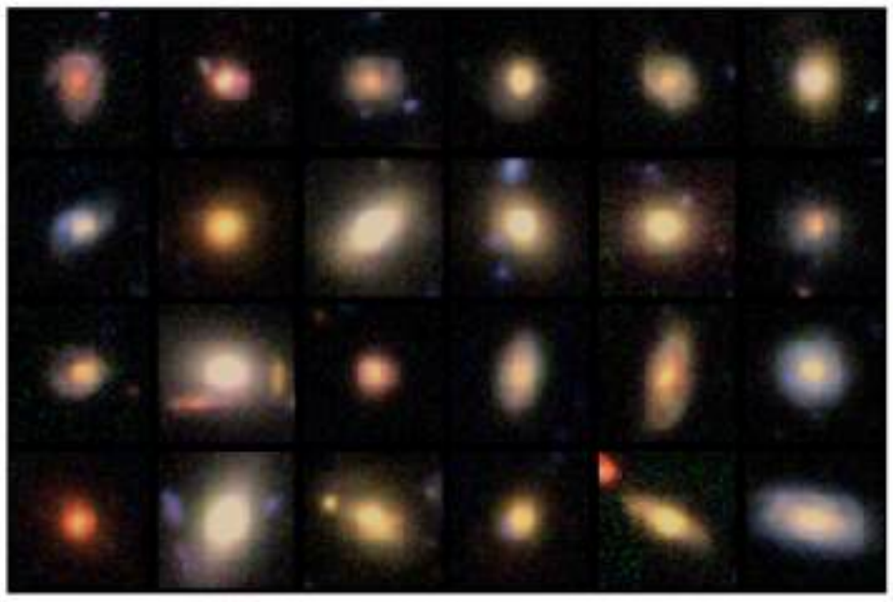


Gavazzi+17

1. Classification

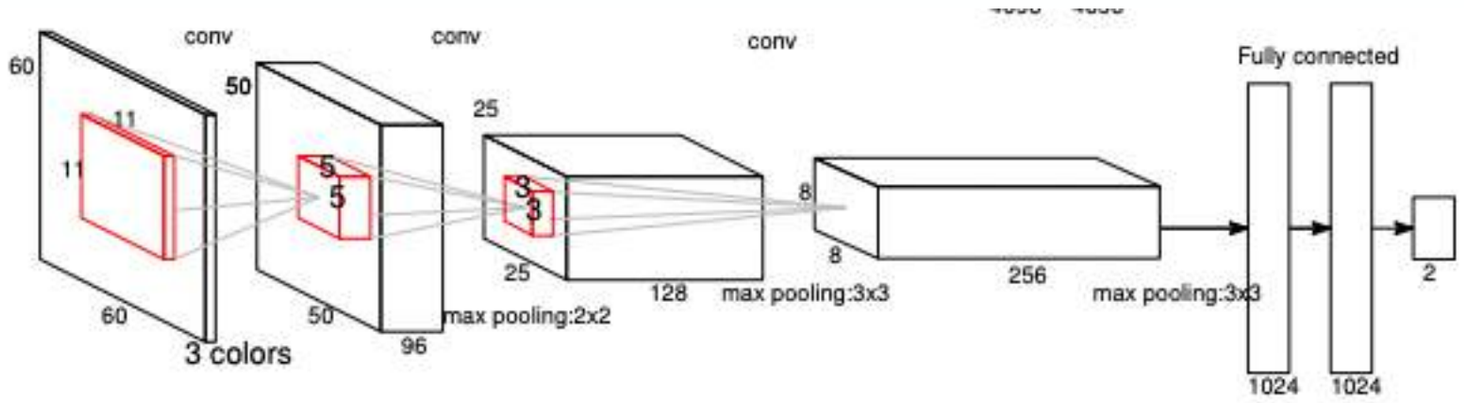


LENS

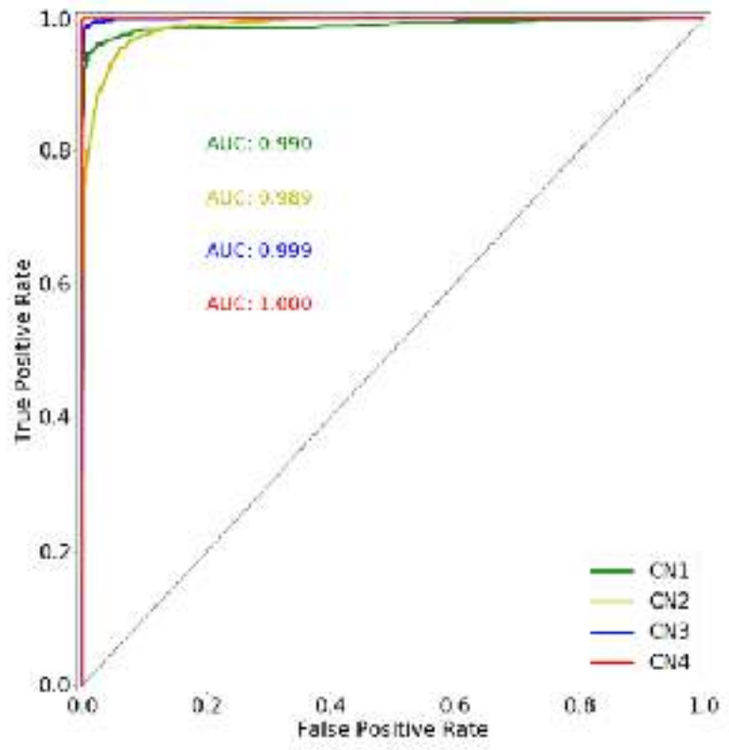


NON-LENS

Jacobs+17



It illustrates the change of paradigm from an algorithmic centric focus to a purely data driven approach to data



Dieleman+15, Huertas-Company+15, Aniyan+17, Charnock+17, Gieseke+17, Jacobs+17, Petrillo+17, Schawinski+17, Alhassan+18, Dominguez-Sanchez+18, George+18, Hinners+18, Lukic+18, Moss+18, Razzano+18, Schaefer+18, Allen+19, Burke+19, Carrasco-Davis+19, Chatterjee+19, Davies+19, Dominguez-Sanchez+19, Fusse+19, Glaser+19, Ishida+19, Jacobs+19, Katebi+19, Lanusse+19Liu+19, Lukic+19, Metcalf+19, Muthukrishna+19, Petrillo+19, Reiman+19, Boucaud+20, Chiani+20, Ghosh+20, Gomez+20, Hausen+20, Hlozek+20, Hosseinzadeh+20. Huang+20, Li+20, Moller+20, Paillassa+20, Tadaki+20, Vargas dos Santos+20, Walmsley+20, Wei+20, Allam+21, Arcelin+21, Becker+21, Bretonniere+21, Burhanudin+21, Davison+21, Donoso-Oliva+21, Jia+21, Lauritsen+21, Ono+21, Ruan+21, Sadegho+21, Tang+21, Tanoglidis+21, Vojtekova+21, Dhar+22Hausen+22, Orwat+22, Pimentel+22, Rezaei+22, Samudre+22, Shen+22, Walmsley+22

1. Classification

CNN based classifications reach unprecedented accuracy

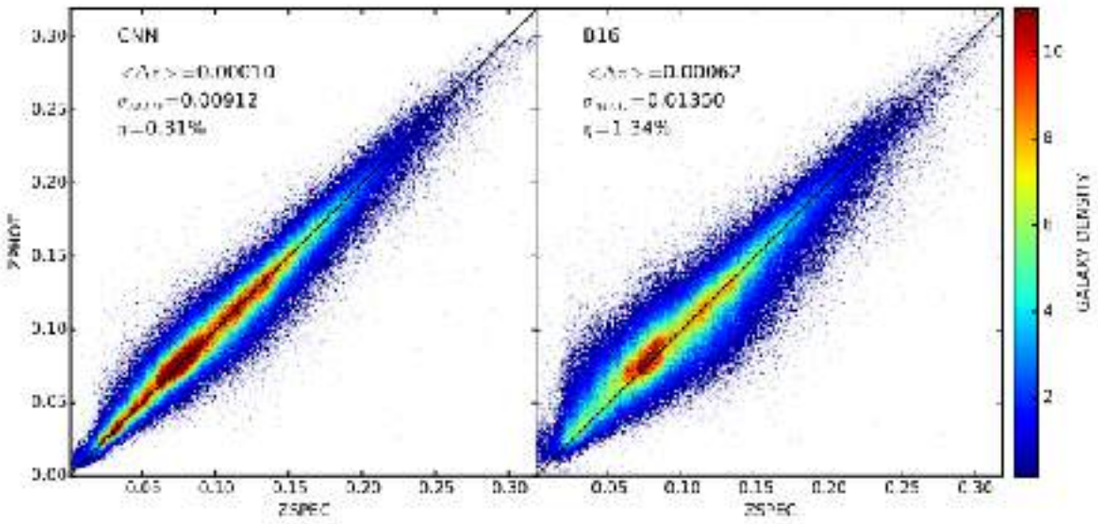
Name	type	AUROC	TPR_0	TPR_{10}	short description
CMU-DeepLens-Resnet-ground3	Ground-Based	0.98	0.09	0.45	CNN
CMU-DeepLens-Resnet-Voting	Ground-Based	0.98	0.02	0.10	CNN
LASTRO EPFL	Ground-Based	0.97	0.07	0.11	CNN
CAS Swinburne Melb	Ground-Based	0.96	0.02	0.08	CNN
AstrOmatic	Ground-Based	0.96	0.00	0.01	CNN
Manchester SVM	Ground-Based	0.93	0.22	0.35	SVM / Gabor
Manchester2	Ground-Based	0.89	0.00	0.01	Human Inspection
ALL-star	Ground-Based	0.84	0.01	0.02	edges/gradiants and Logistic Reg.
CAST	Ground-Based	0.83	0.00	0.00	CNN / SVM
YattaLensLite	Ground-Based	0.82	0.00	0.00	SExtractor
LASTRO EPFL	Space-Based	0.93	0.00	0.08	CNN
CMU-DeepLens-Resnet	Space-Based	0.92	0.22	0.29	CNN
GAMOCLASS	Space-Based	0.92	0.07	0.36	CNN
CMU-DeepLens-Resnet-Voting	Space-Based	0.91	0.00	0.01	CNN
AstrOmatic	Space-Based	0.91	0.00	0.01	CNN
CMU-DeepLens-Resnet-aug	Space-Based	0.91	0.00	0.00	CNN
Kapteyn Resnet	Space-Based	0.82	0.00	0.00	CNN
CAST	Space-Based	0.81	0.07	0.12	CNN
Manchester1	Space-Based	0.81	0.01	0.17	Human Inspection
Manchester SVM	Space-Based	0.81	0.03	0.08	SVM / Gabor
NeuralNet2	Space-Based	0.76	0.00	0.00	CNN / wavelets
YattaLensLite	Space-Based	0.76	0.00	0.00	Arcs / SExtractor
All-now	Space-Based	0.73	0.05	0.07	edges/gradiants and Logistic Reg.
GAHEC IRAP	Space-Based	0.66	0.00	0.01	arc finder

Metcalf+19

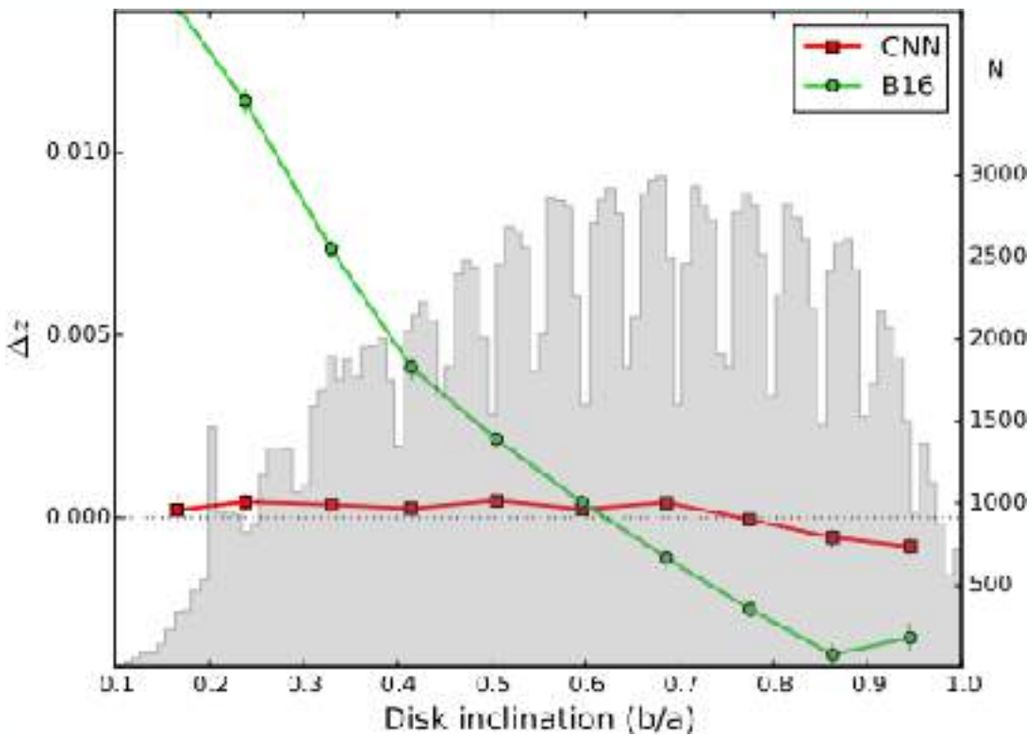
Dieleman+15, Huertas-Company+15, Aniyan+17, Charnock+17, Gieseke+17, Jacobs+17, Petrillo+17, Schawinski+17, Alhassan+18, Dominguez-Sanchez+18, George+18, Hinners+18, Lukic+18, Moss+18, Razzano+18, Schaefer+18, Allen+19, Burke+19, Carrasco-Davis+19, Chatterjee+19, Davies+19, Dominguez-Sanchez+19, Fusse+19, Glaser+19, Ishida+19, Jacobs+19, Katebi+19, Lanusse+19Liu+19, Lukic+19, Metcalf+19, Muthukrishna+19, Petrillo+19, Reiman+19, Boucaud+20, Chiani+20, Ghosh+20, Gomez+20, Hausen+20, Hlozek+20, Hosseinzadeh+20. Huang+20, Li+20, Moller+20, Paillassa+20, Tadaki+20, Vargas dos Santos+20, Walmsley+20, Wei+20, Allam+21, Arcelin+21, Becker+21, Bretonniere+21, Burhanudin+21, Davison+21, Donoso-Oliva+21, Jia+21, Lauritsen+21, Ono+21, Ruan+21, Sadegho+21, Tang+21, Tanoglidis+21, Vojtekova+21, Dhar+22Hausen+22, Orwat+22, Pimentel+22, Rezaei+22, Samudre+22, Shen+22, Walmsley+22

Photometric Redshifts

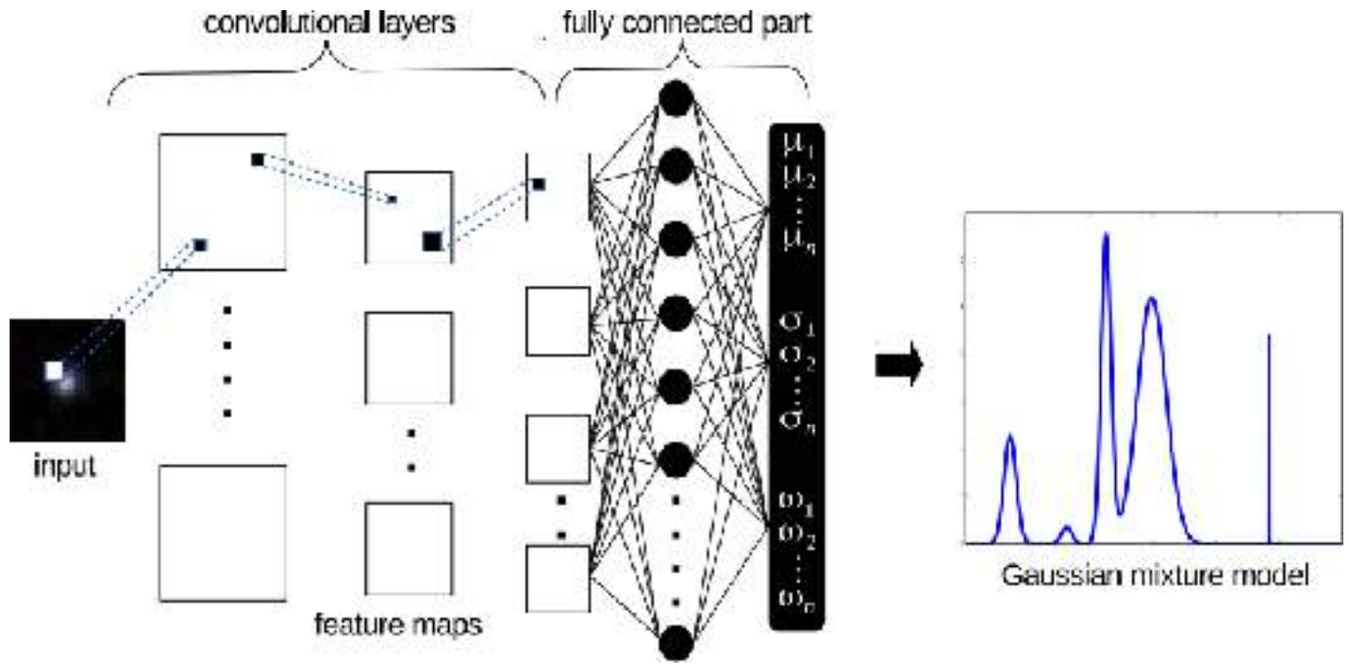
Deep Learning **Classical approach**



Geometric Effects are automatically considered (beyond photometry)



Pasquet+18

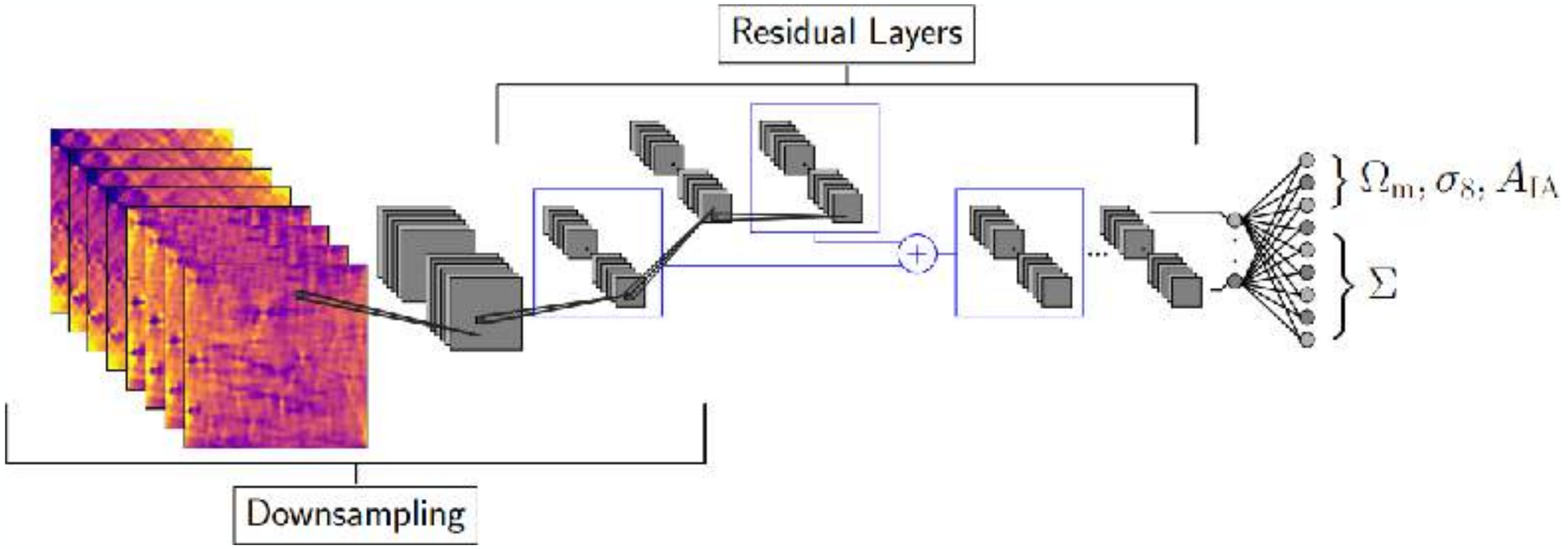


Disanto+18

Uncertainty quantification through Mixture Density Networks

Hazevah+17, Perreault-Levasseur+17, Morningstar+18, Stark+18, Tuccillo+18, Bom+19, Lovell+19, Madireddy+19, Pearson+19, Simet+19, Pasquet+19, Wu+19, Menou+19, Aragon-Calvo+20, Chianese+20, Stahl+20, Surana+20, Shuntov+20, Cabayol-Garcia+20, Campagne+20, Buck+21, Grover+21, Hayat+21, Li+21, Maresca+21, Qiu+21, Rhea+21, Schuldt+21, Tohill+21, Yao-Yu+21, Dey+21, Lee+21, Zhou+21, Henghes+21, Ansari+21

Deep Learning for Cosmological Inference



Motivation: Generalize comparison of observations with theory, beyond basic summary statistics

Neural Networks are used as efficient feature extractors

Adding additional invariances

DATA AUGMENTATION

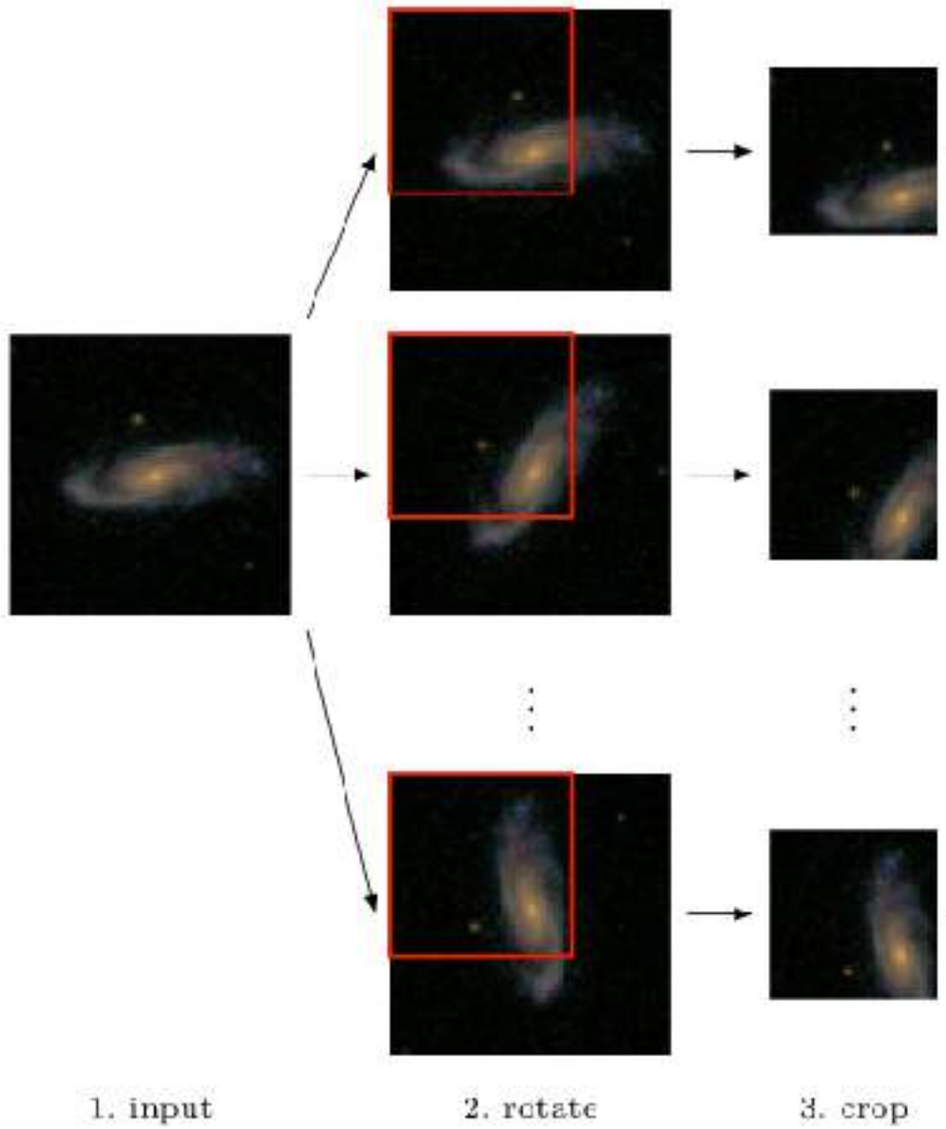
ANOTHER WAY TO REDUCE OVER-FITTING IS TO "AUGMENT" THE SIZE OF THE DATASET AVAILABLE FOR TRAINING

FOR MANY APPLICATIONS THE CLASSIFICATION SHOULD BE INDEPENDENT TO:

- TRANSALTIONS
- ROTATIONS
- SCALINGS

ETC…

DATA [illegible]TATION



Dieleman+15

FOR MANY APPLICATIONS THE CLASSIFICATION SHOULD BE INDEPENDENT TO:

- TRANSALTIONS
- ROTATIONS
- SCALINGS

ETC…

DATA AUGMENTATION

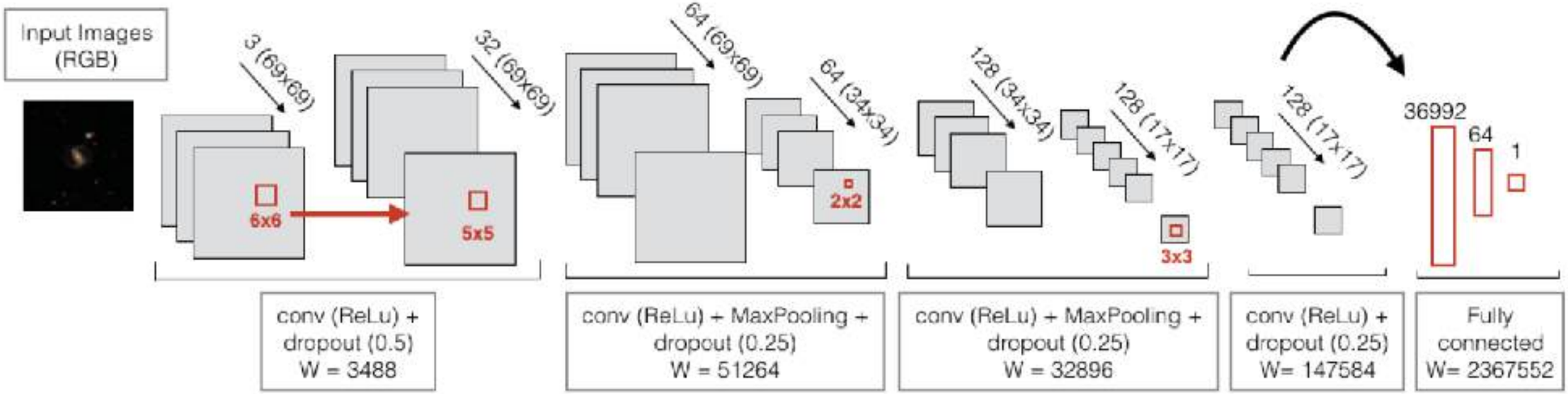
FOR MANY APPLICATIONS THE CLASSIFICATION SHOULD BE INDEPENDENT TO:

- TRANSALTIONS
- ROTATIONS
- SCALINGS

ETC…

BEYOND CLASSIFICATION: IMAGE2IMAGE NETWORKS

UP TO NOW CNNs MAP IMAGES (SIGNALS) INTO FLOATS



Dominguez-Sanchez+18

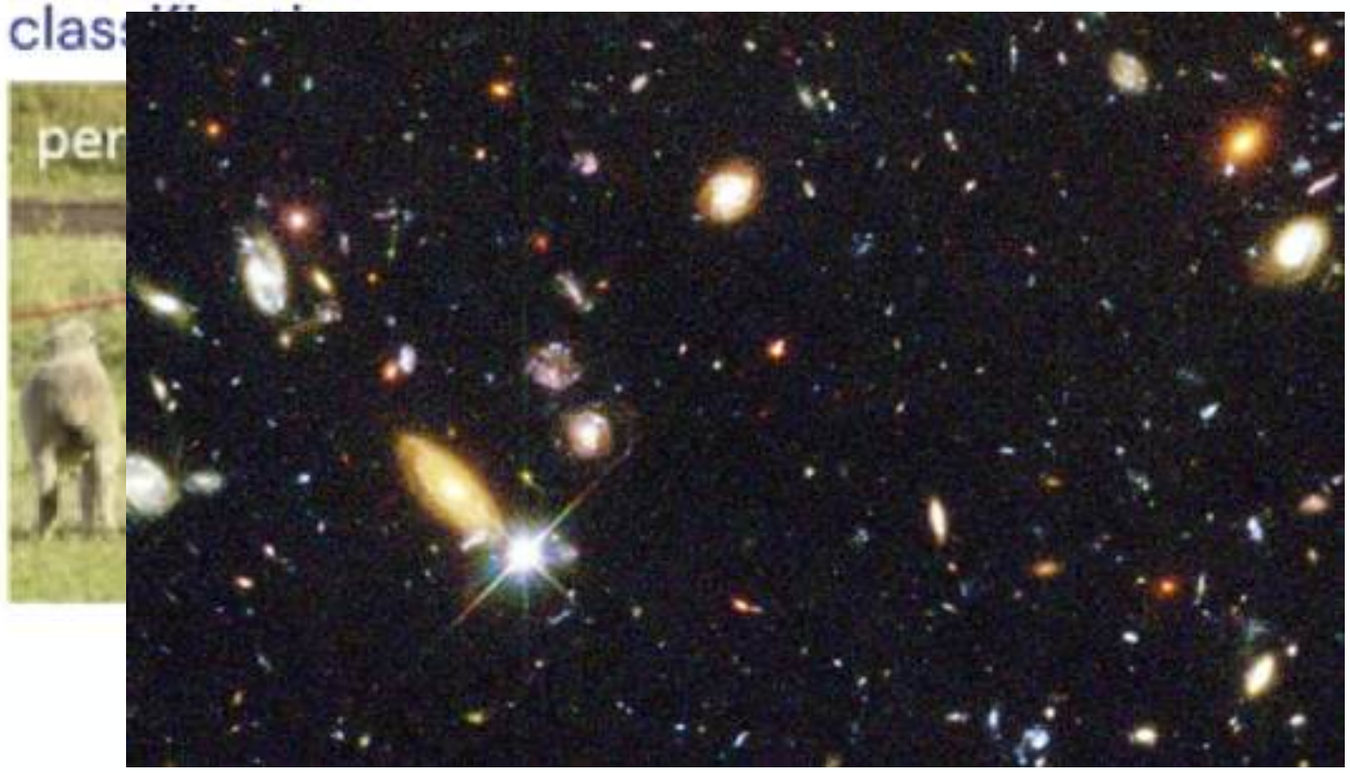
Classification has its limits ….

HOW DO I CLASSIFY THIS IMAGE?

Classification has its limits ….

HOW DO I CLASSIFY THIS IMAGE?

Classification has its limits ….

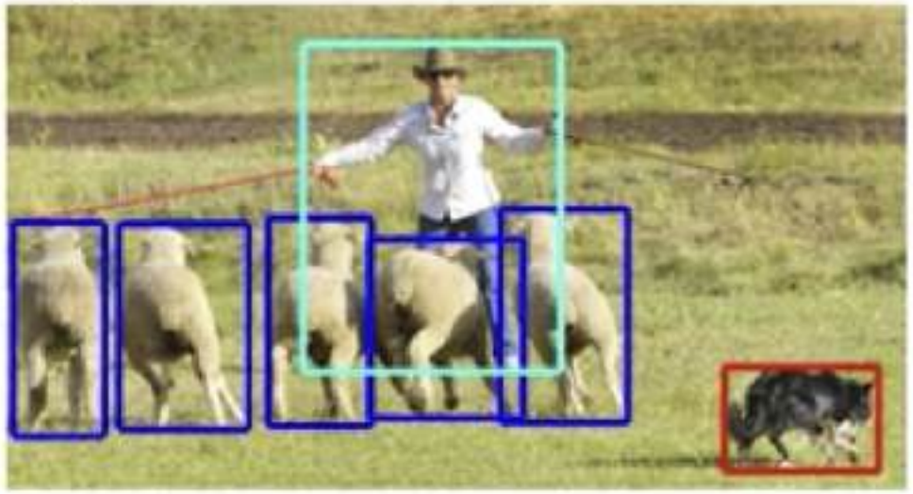


HOW DO I CLASSIFY THIS IMAGE?

Going beyond classification: increasing complexity

classification

object detection



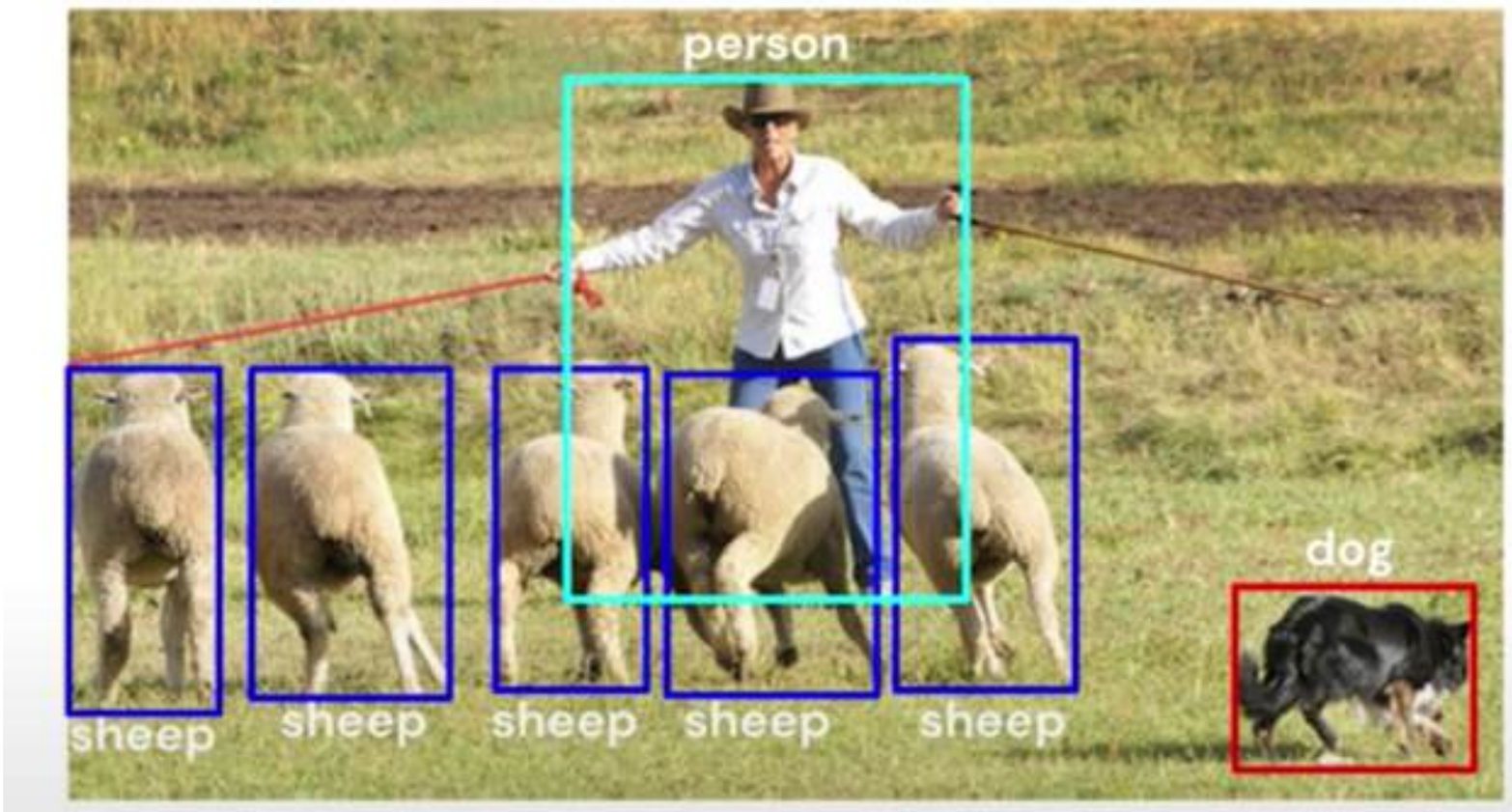
semantic segmentation

instance segmentation

@deepmind (V. Patraucean)

Object detection

First task is to find a bounding box for every object. How we do that?



@deepmind (V. Patraucean)

Inputs

- RGB image $H \times W \times 3$

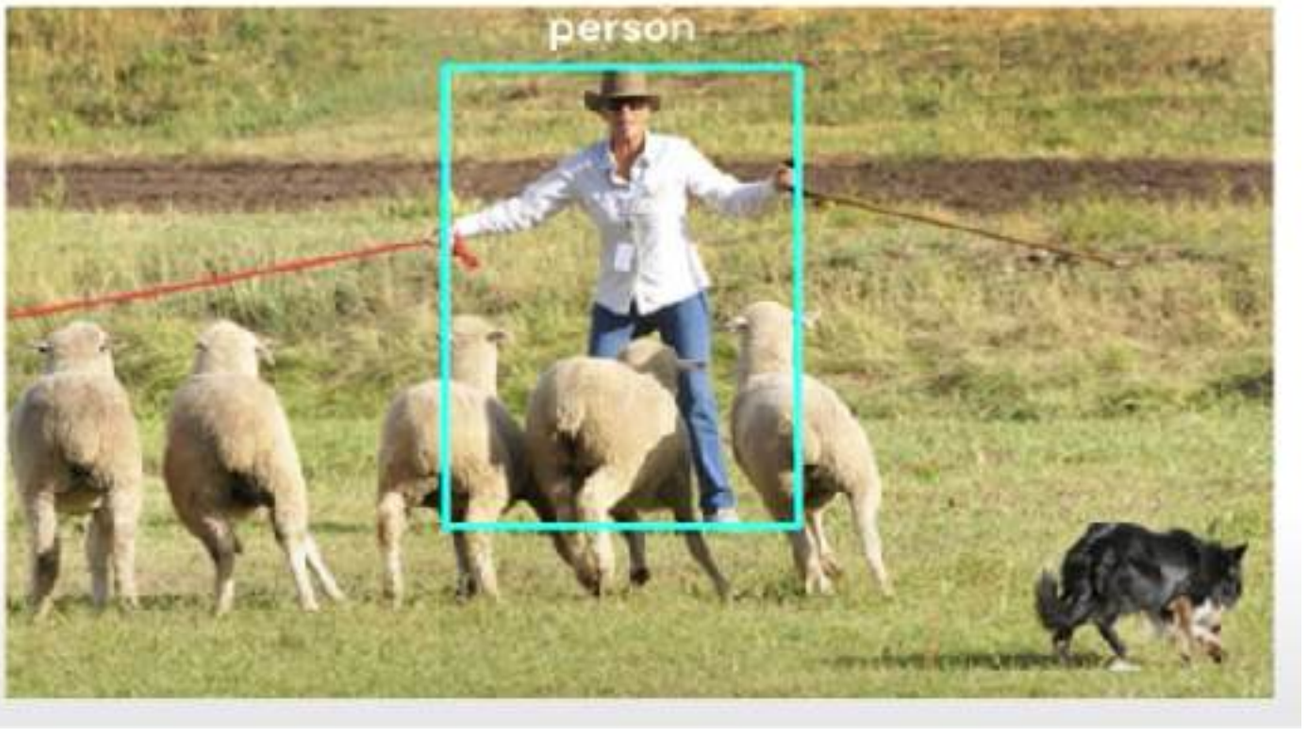
Targets

- Class label *one_hot* 0 0 0 1 0...
- Object bounding box
  (x_c, y_c, h, w)

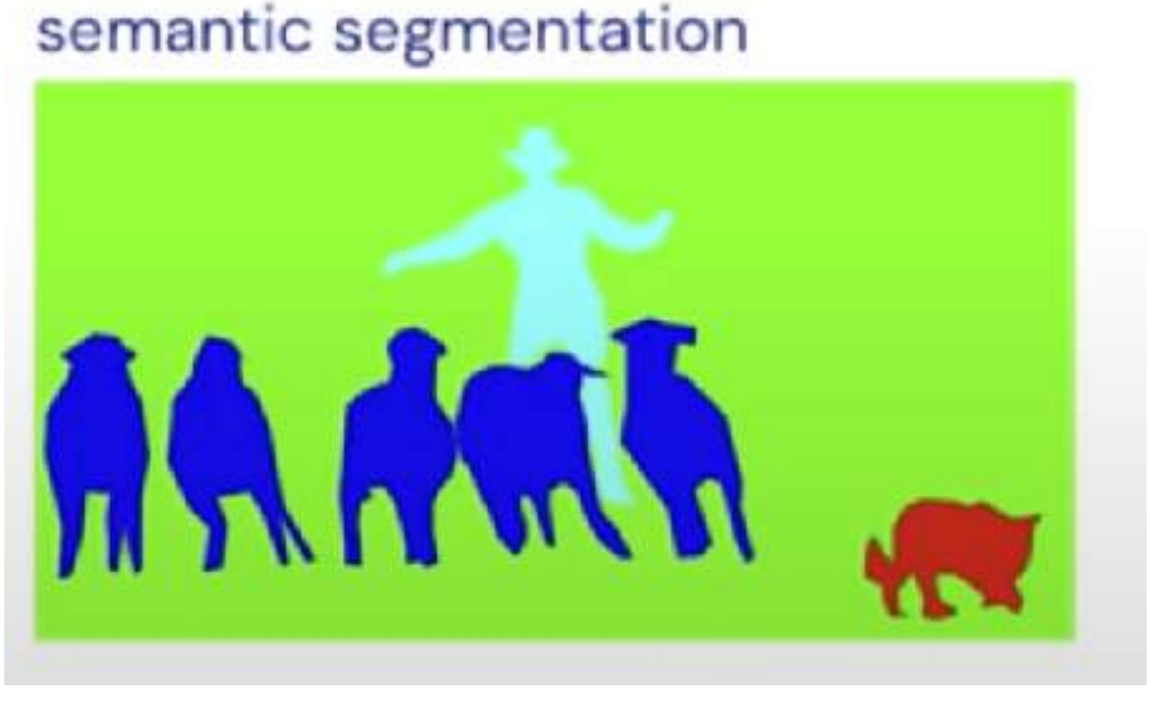
for all the objects present in the scene

@deepmind (V. Patraucean)

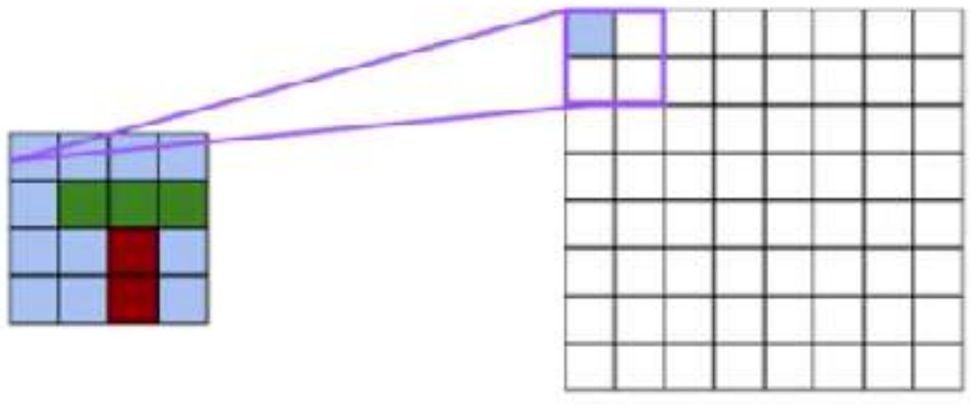
LET’S GO A STEP FURTHER INTO SEMANTIC SEGMENTATION



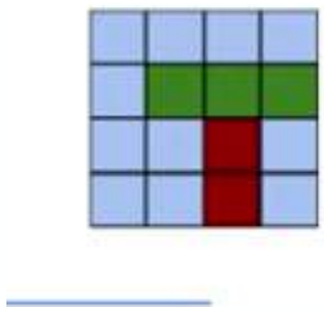
BOUNDING BOXES ARE NOT ALWAYS GOOD REPRESENTATIONS

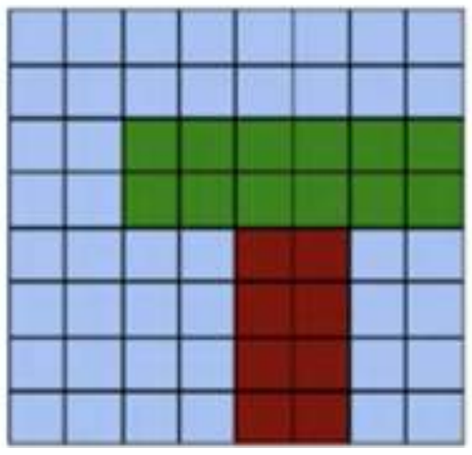


UNPOOLING OPERATION (INVERSE OF POOLING)



COPY PIXELS IN A GIVEN WINDOW

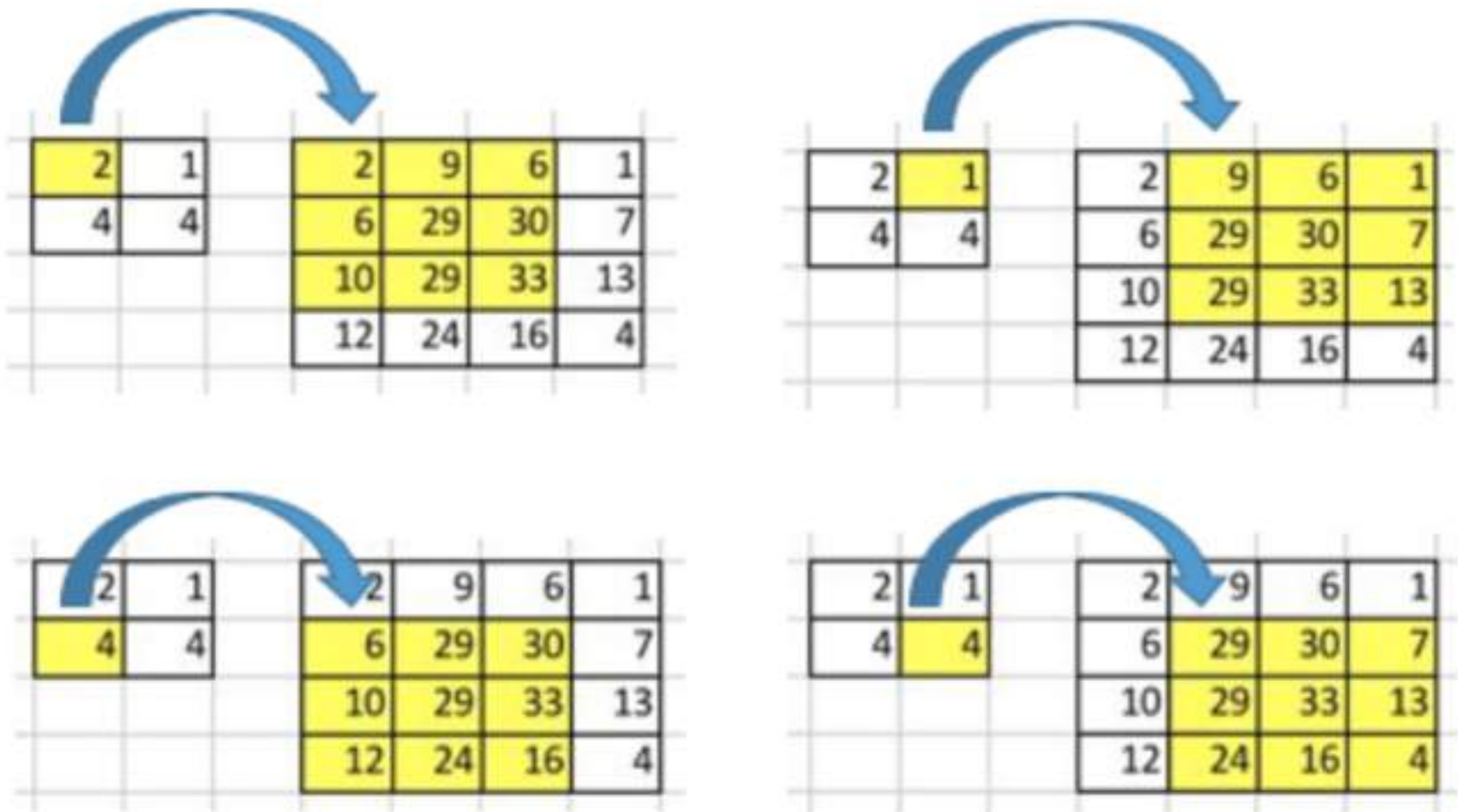




GENERATES LARGER IMAGES FROM SMALLER ONES

TRANSPOSED CONVOLUTION

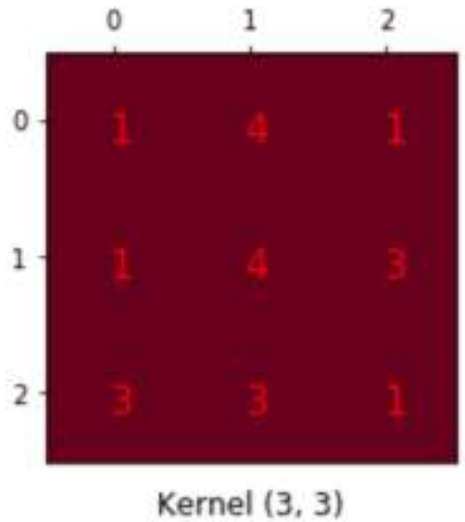
ALLOWS TO INCREASE THE SIZE



Going Backward of Convolution

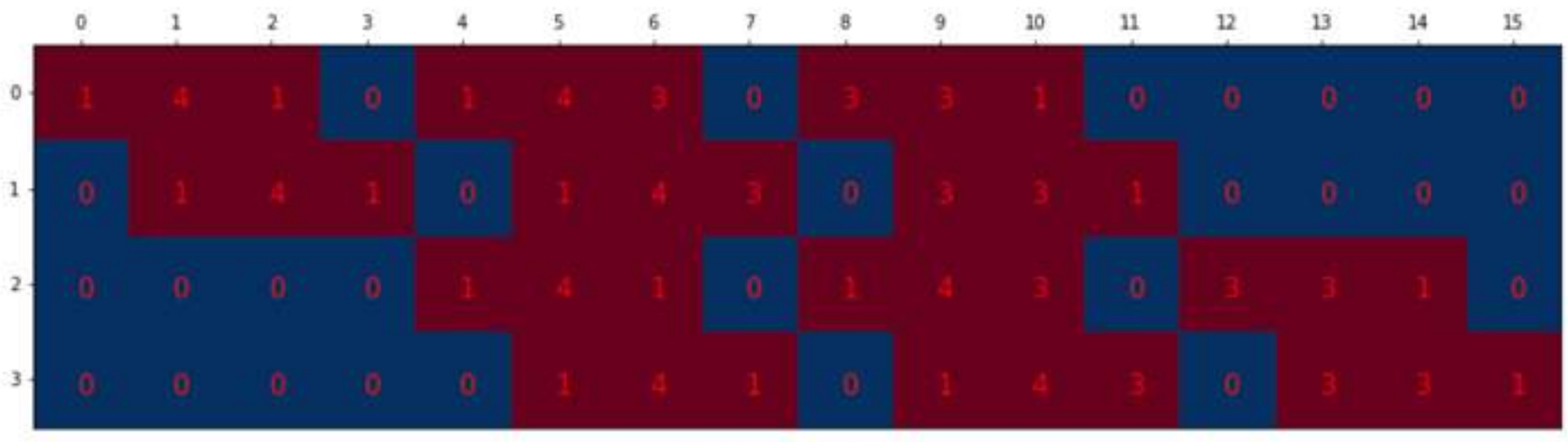
EXAMPLE TAKEN FROM HERE

CONVOLUTION MATRIX



Kernel (3, 3)

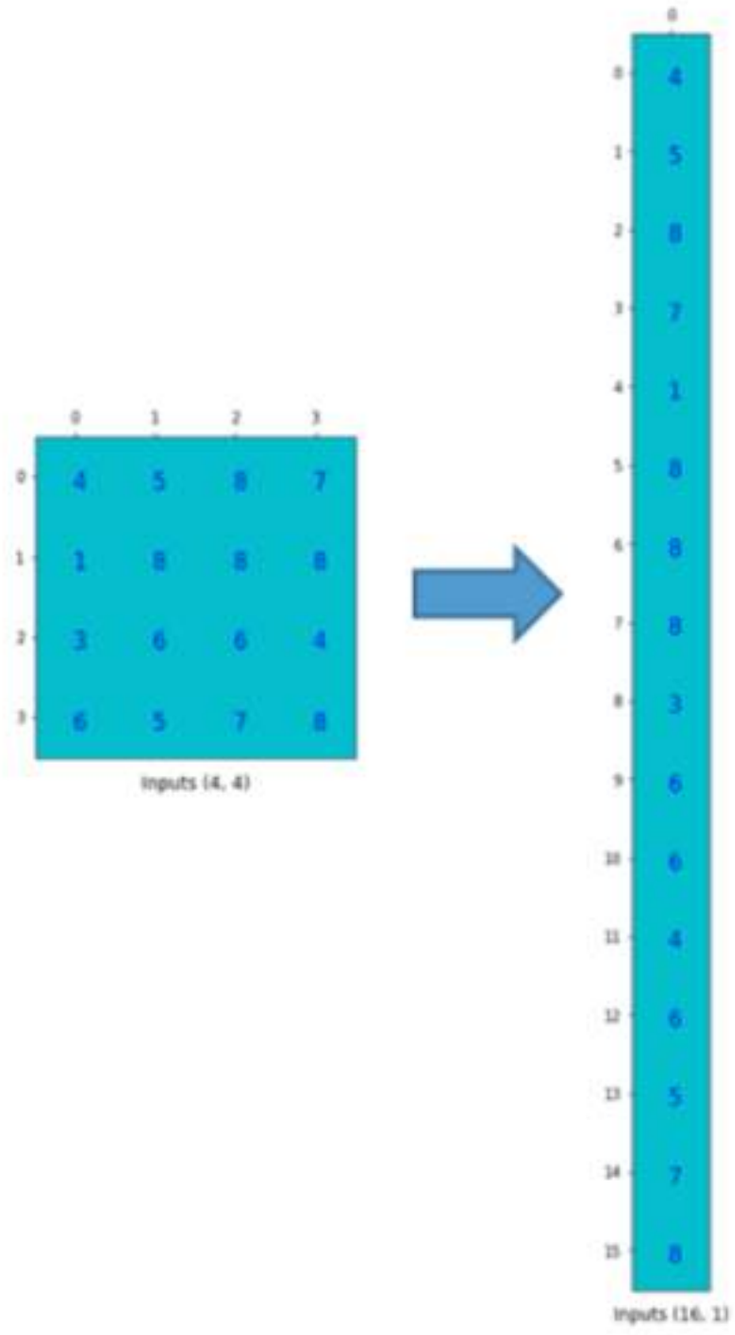
THE KERNEL CAN BE ARRANGED IN FORM OF A MATRIX:



Convolution Matrix (4, 16)

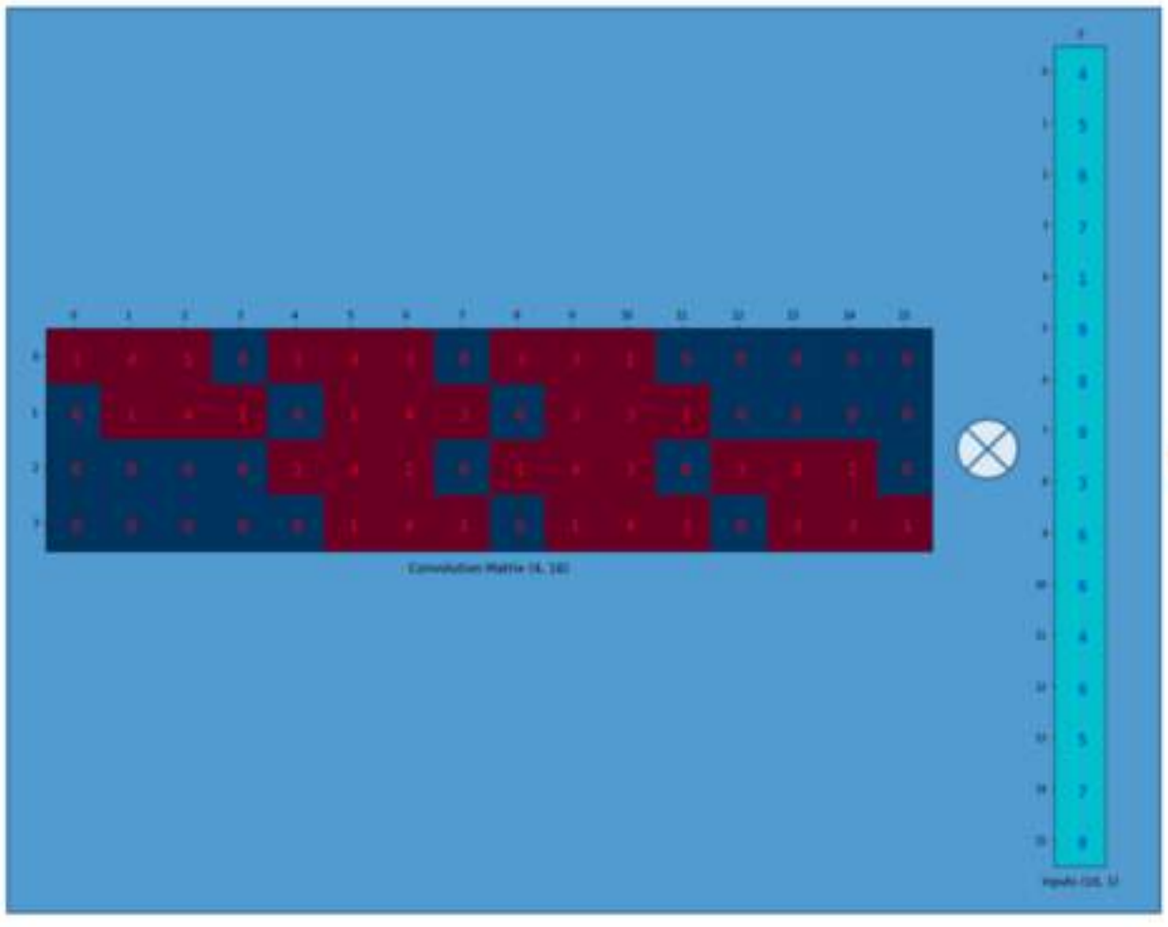
EXAMPLE TAKEN FROM HERE

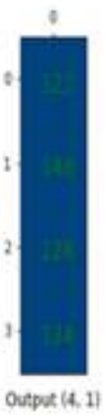
THE INPUT IS FLATTENED INTO A COLUMN VECTOR

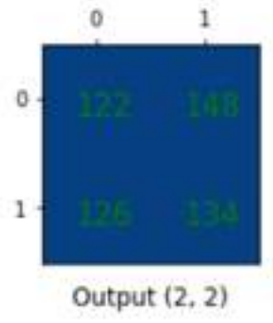


EXAMPLE TAKEN FROM HERE

THE CONVOLUTION IS TRANSFORMED INTO A PRODUCT OF MATRICES



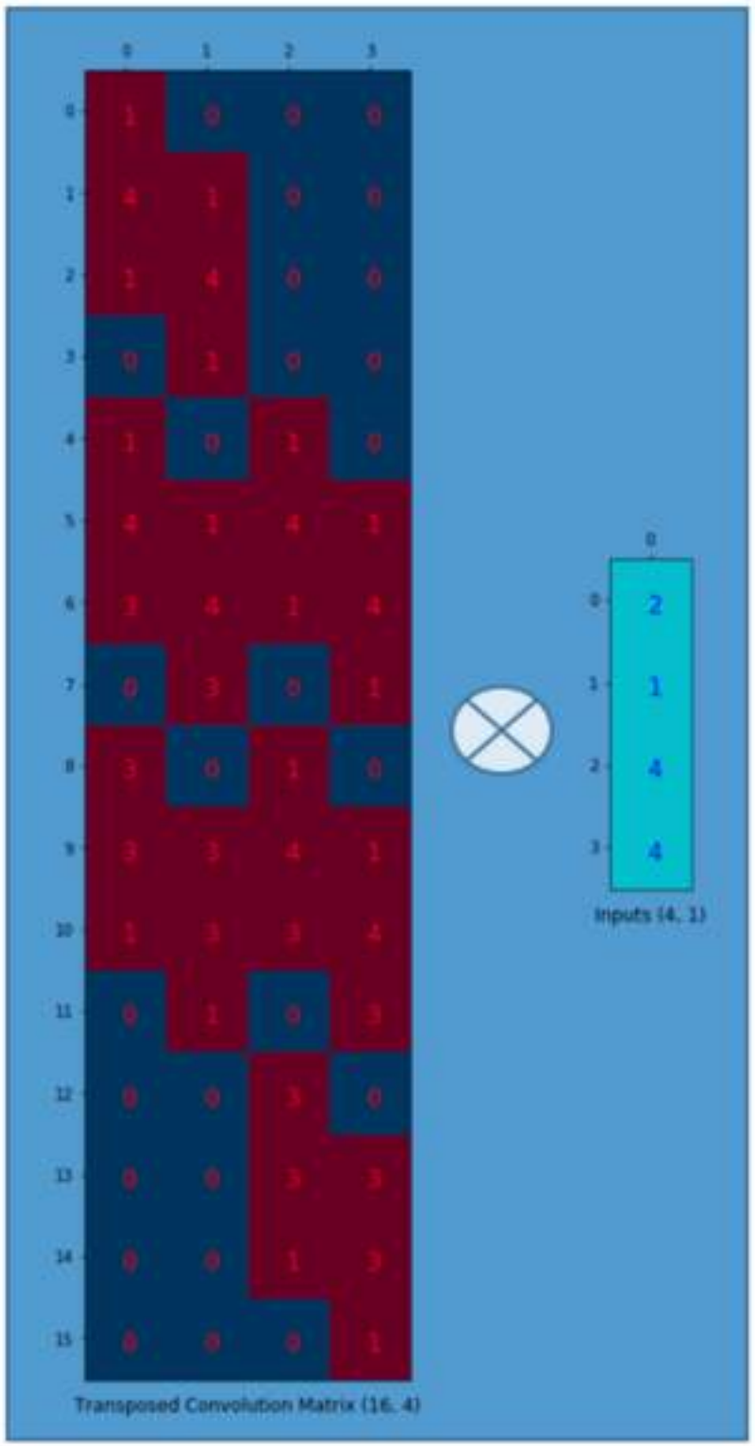


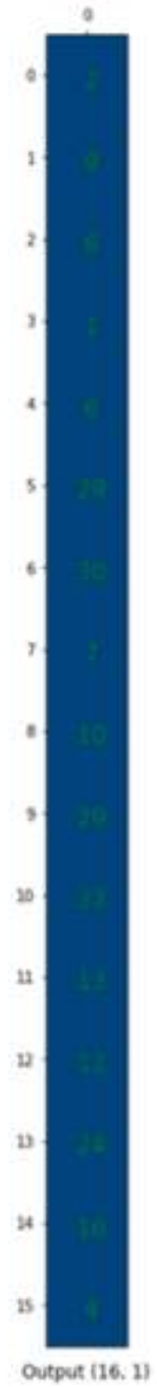


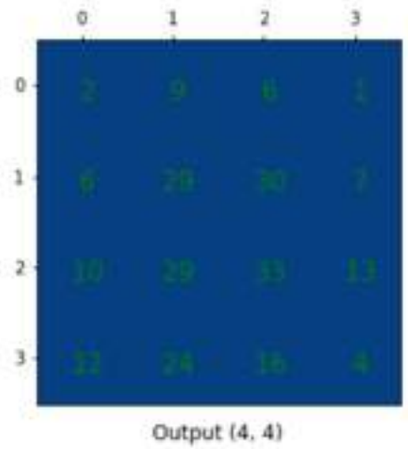
RESHAPED OUTPUT

EXAMPLE TAKEN FROM HERE

THE TRANSPOSED CONVOLUTION IS THE INVERSE OPERATION



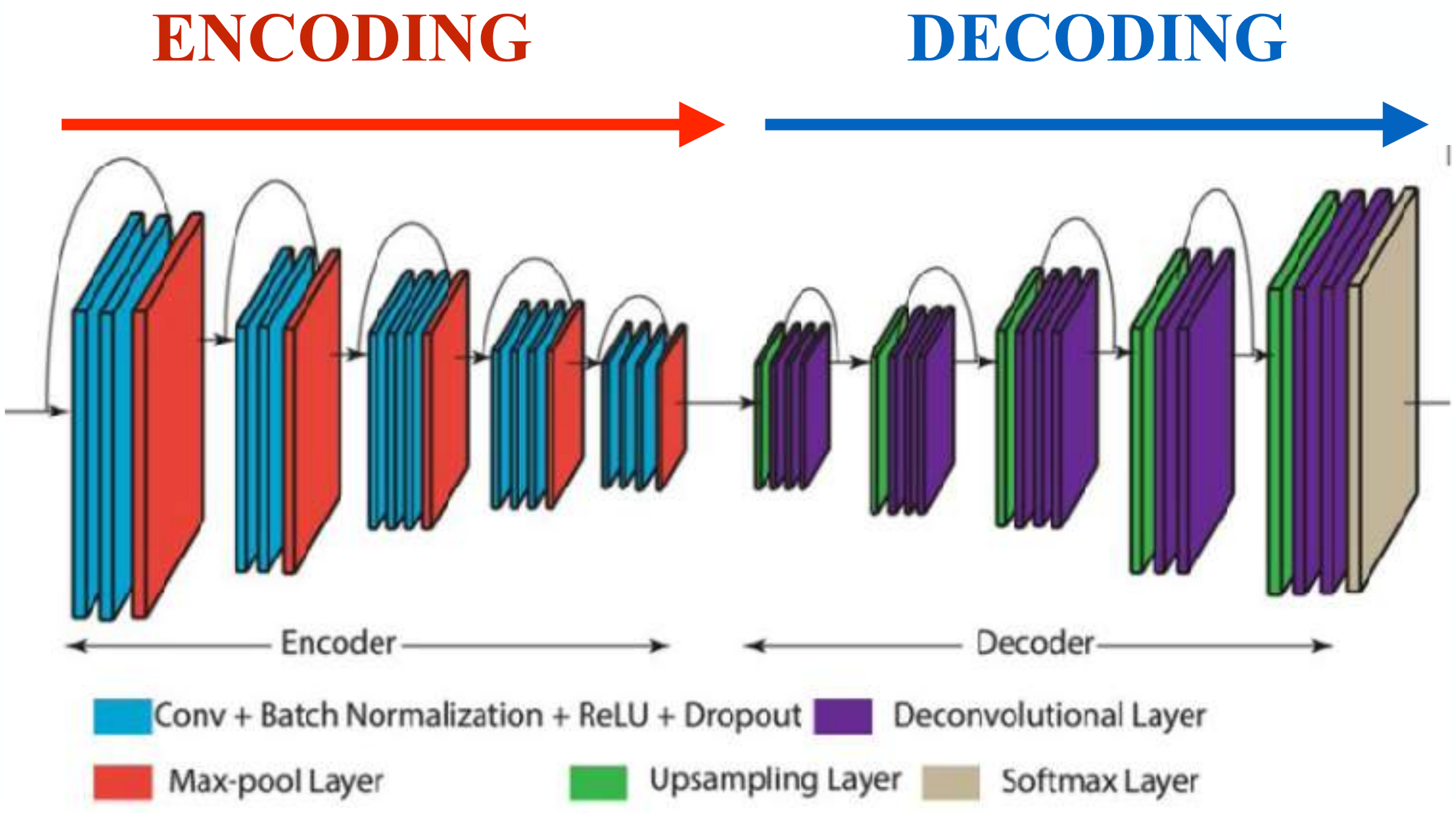




RESHAPED OUTPUT

EXAMPLE TAKEN FROM HERE

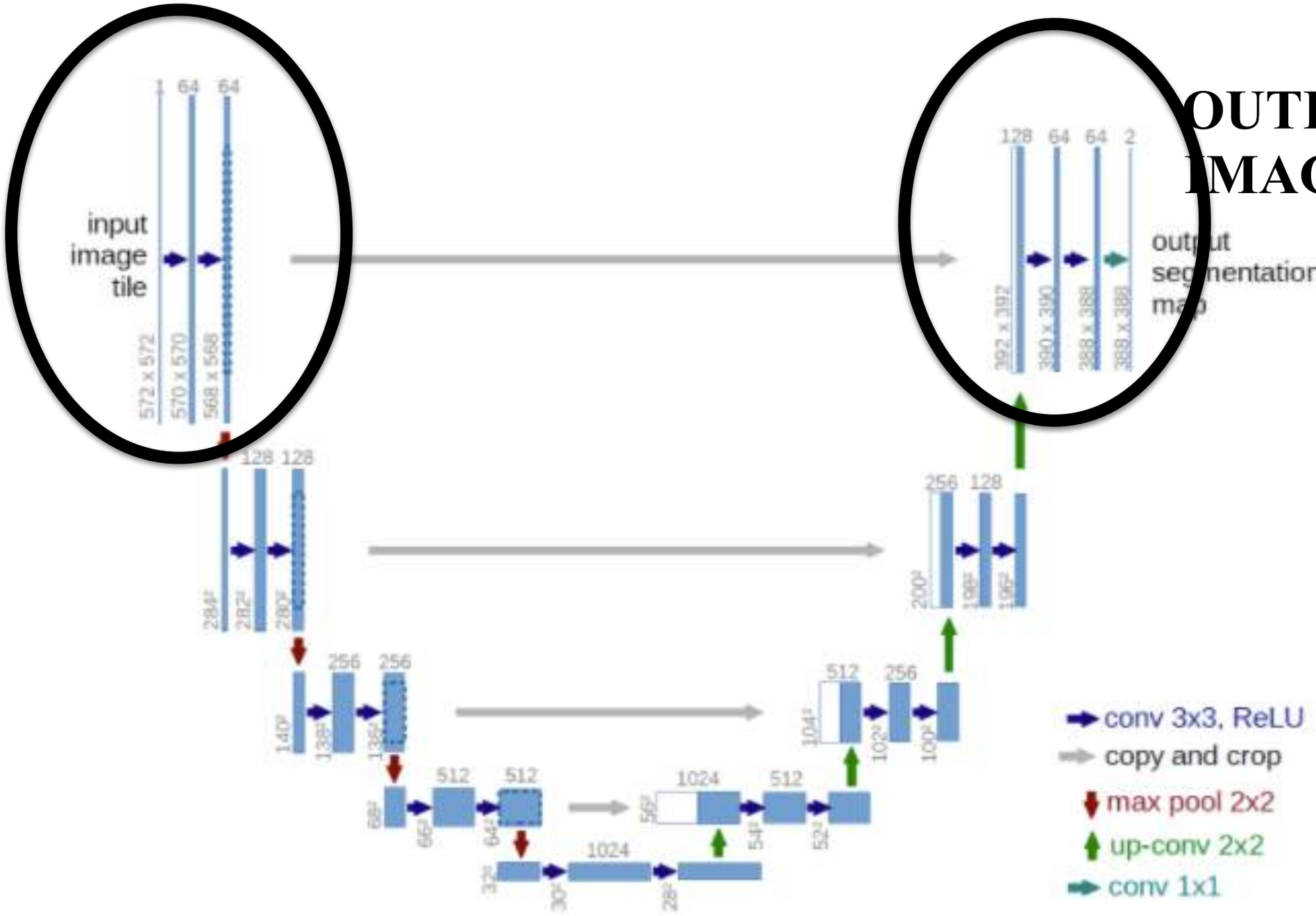
ENCODER-DECODERS GO FROM IMAGE 2 IMAGE



WE CALL THIS FULLY CONVOLUTIONAL NEURAL NETWORKS

ENCODING-DECODING TO EXTRACT IMAGE FEATURES: U-NET

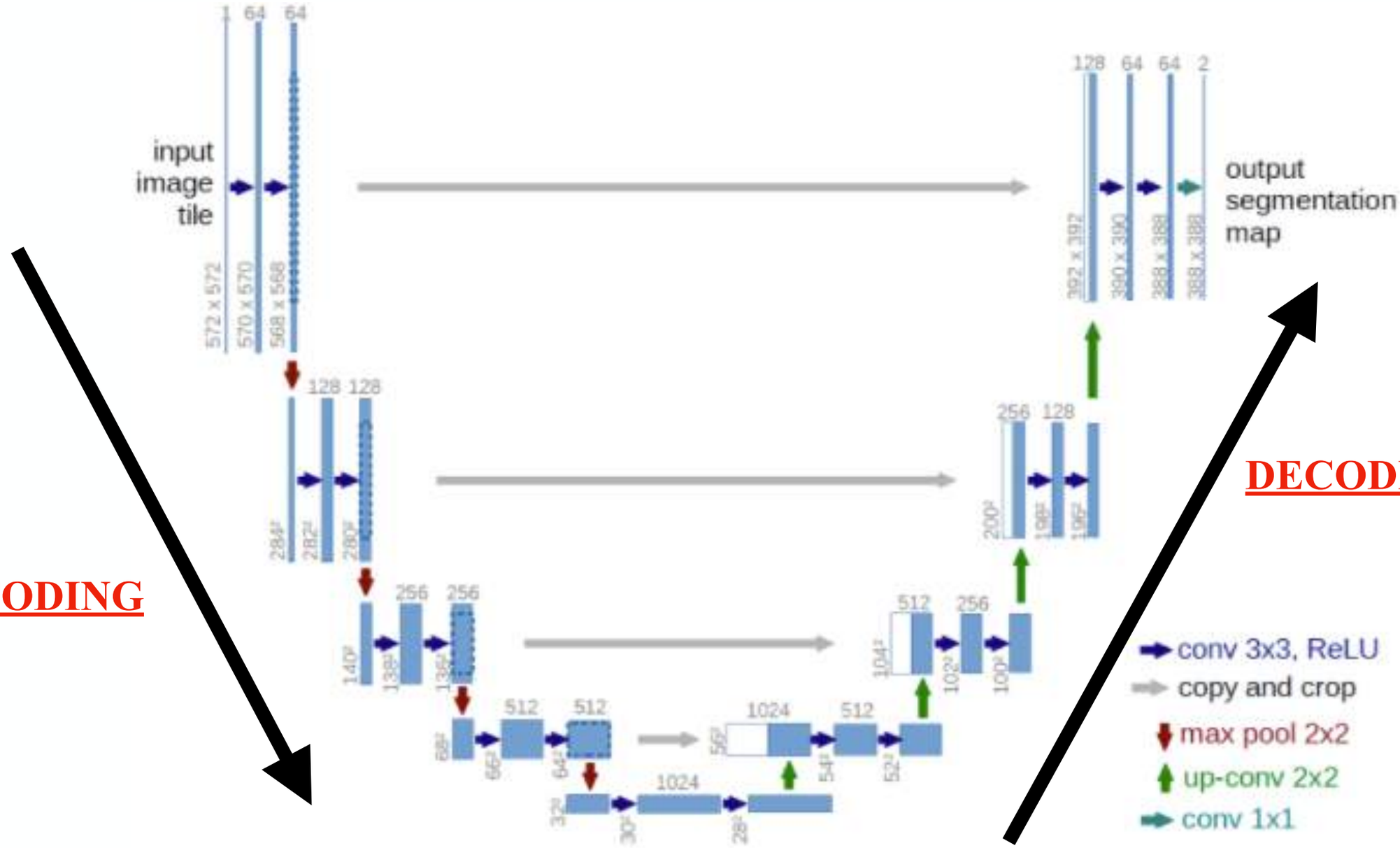
INPUT IMAGES



OUTPUT IMAGES

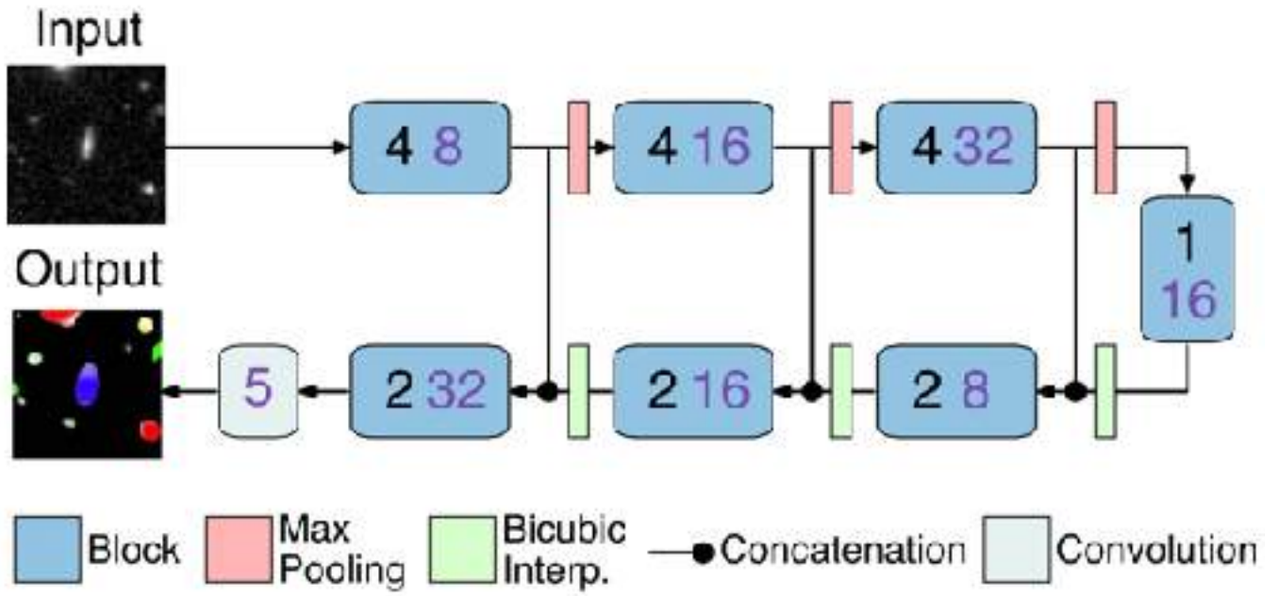
Ronnerberger+15

ENCODING-DECODING TO EXTRACT IMAGE FEATURES: THE U-NET

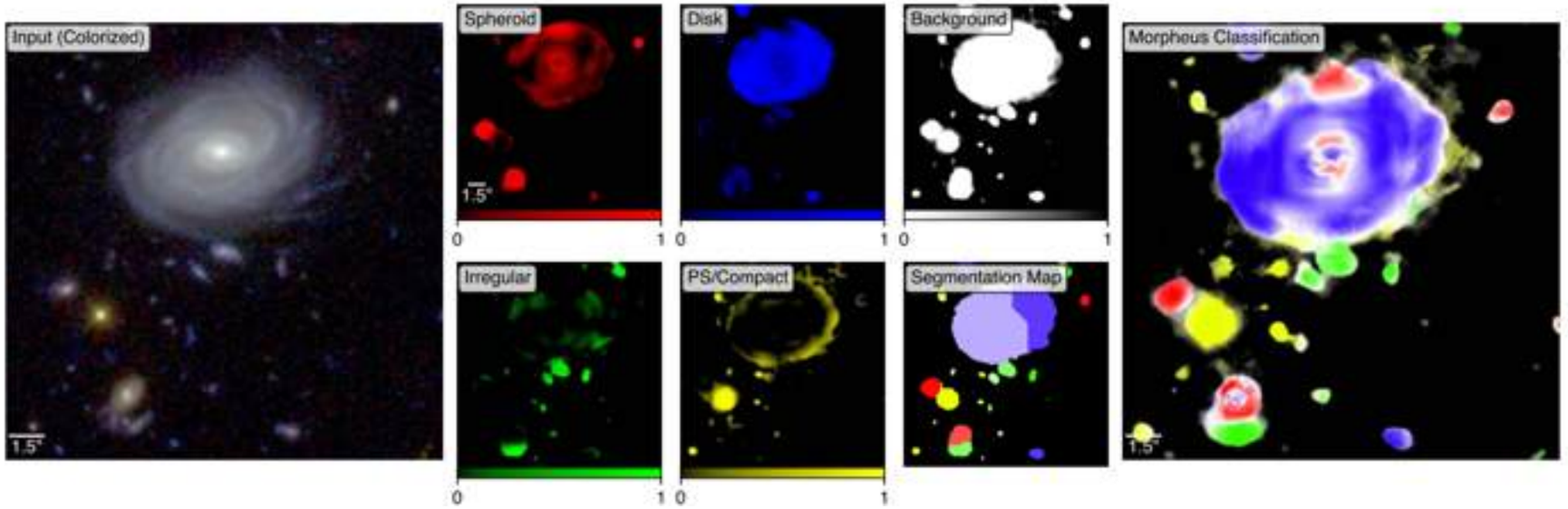


Ronnerberger+15

2. Segmentation



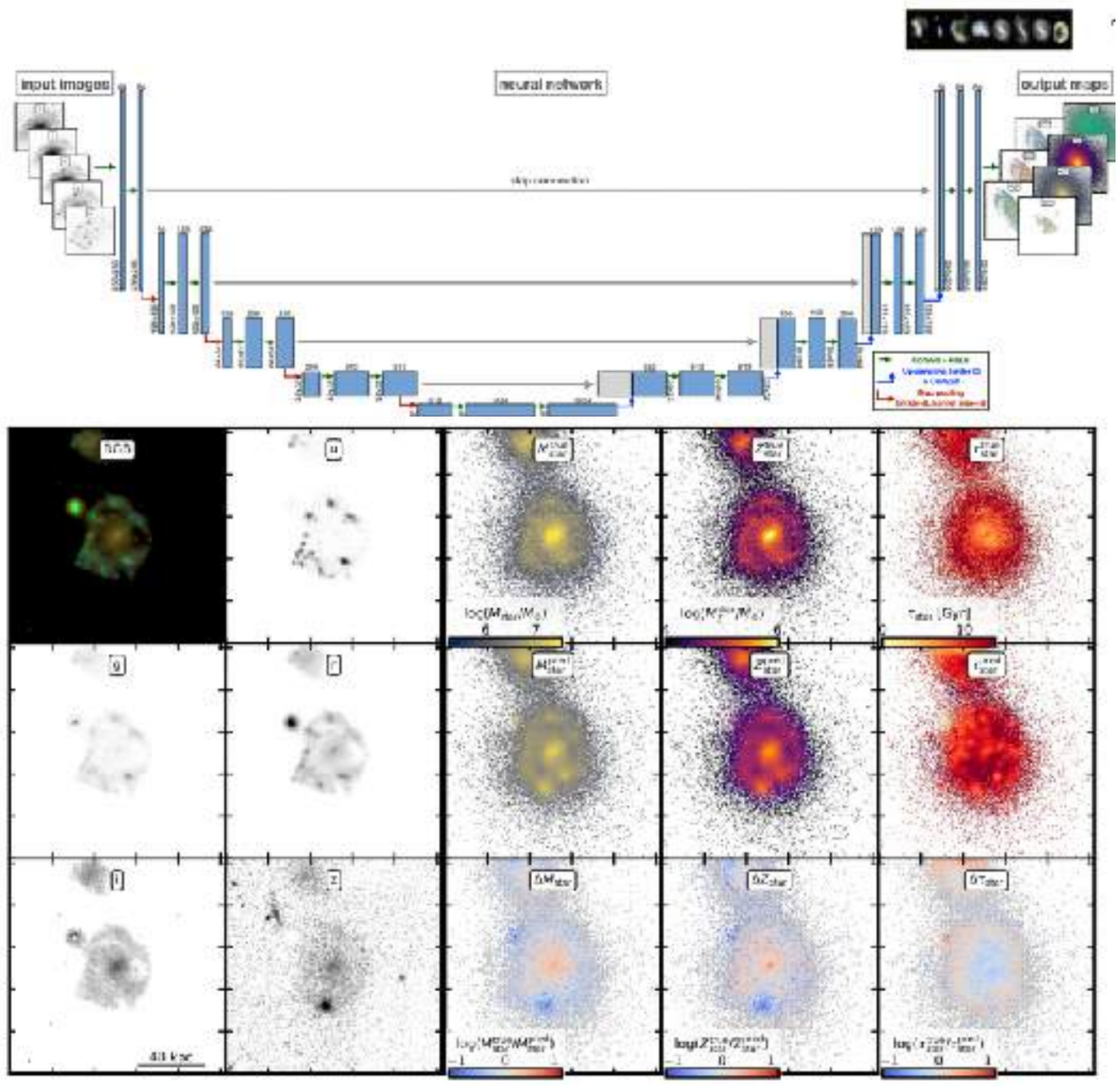
Classification at the pixel level



Hausen+20

Dieleman+15, Huertas-Company+15, Aniyan+17, Charnock+17, Gieseke+17, Jacobs+17, Petrillo+17, Schawinski+17, Alhassan+18, Dominguez-Sanchez+18, George+18, Hinners+18, Lukic+18, Moss+18, Razzano+18, Schaefer+18, Allen+19, Burke+19, Carrasco-Davis+19, Chatterjee+19, Davies+19, Dominguez-Sanchez+19, Fusse+19, Glaser+19, Ishida+19, Jacobs+19, Katebi+19, Lanusse+19Liu+19, Lukic+19, Metcalf+19, Muthukrishna+19, Petrillo+19, Reiman+19, Boucaud+20, Chiani+20, Ghosh+20, Gomez+20, Hausen+20, Hlozek+20, Hosseinzadeh+20. Huang+20, Li+20, Moller+20, Paillassa+20, Tadaki+20, Vargas dos Santos+20, Walmsley+20, Wei+20, Allam+21, Arcelin+21, Becker+21, Bretonniere+21, Burhanudin+21, Davison+21, Donoso-Oliva+21, Jia+21, Lauritsen+21, Ono+21, Ruan+21, Sadegho+21, Tang+21, Tanoglidis+21, Vojtekova+21, Dhar+22Hausen+22, Orwat+22, Pimentel+22, Rezaei+22, Samudre+22, Shen+22, Walmsley+22

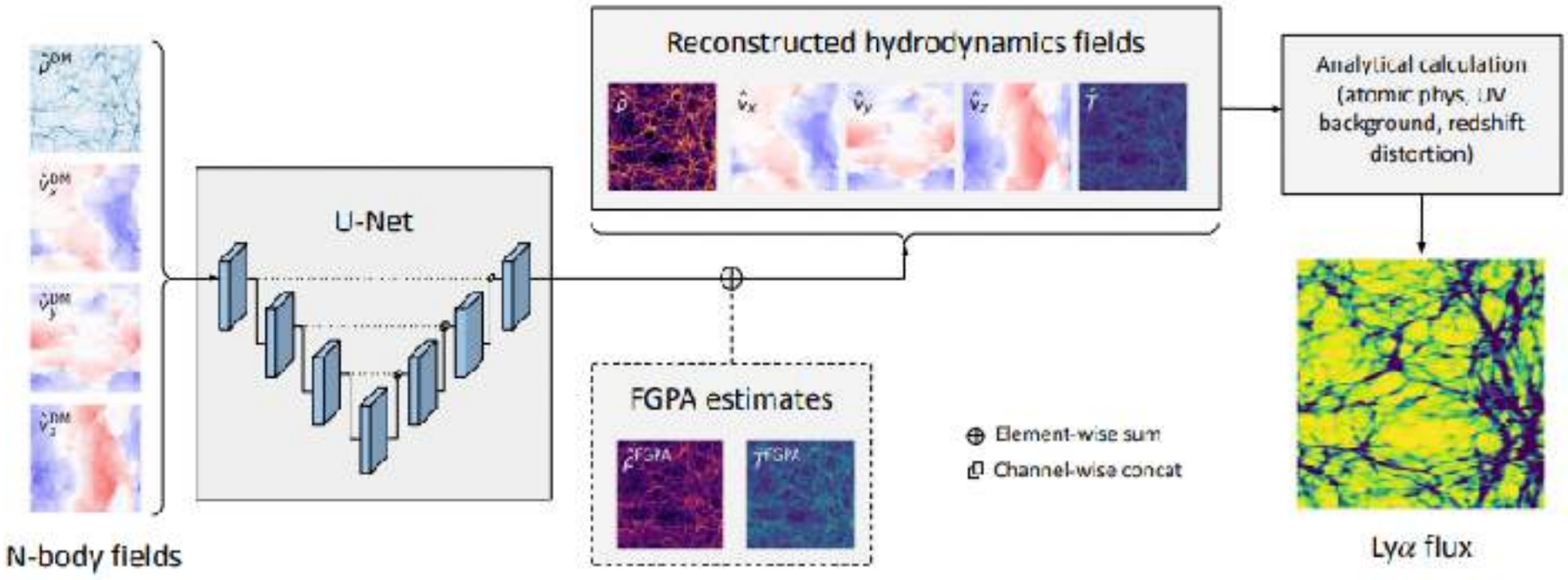
Stellar Populations



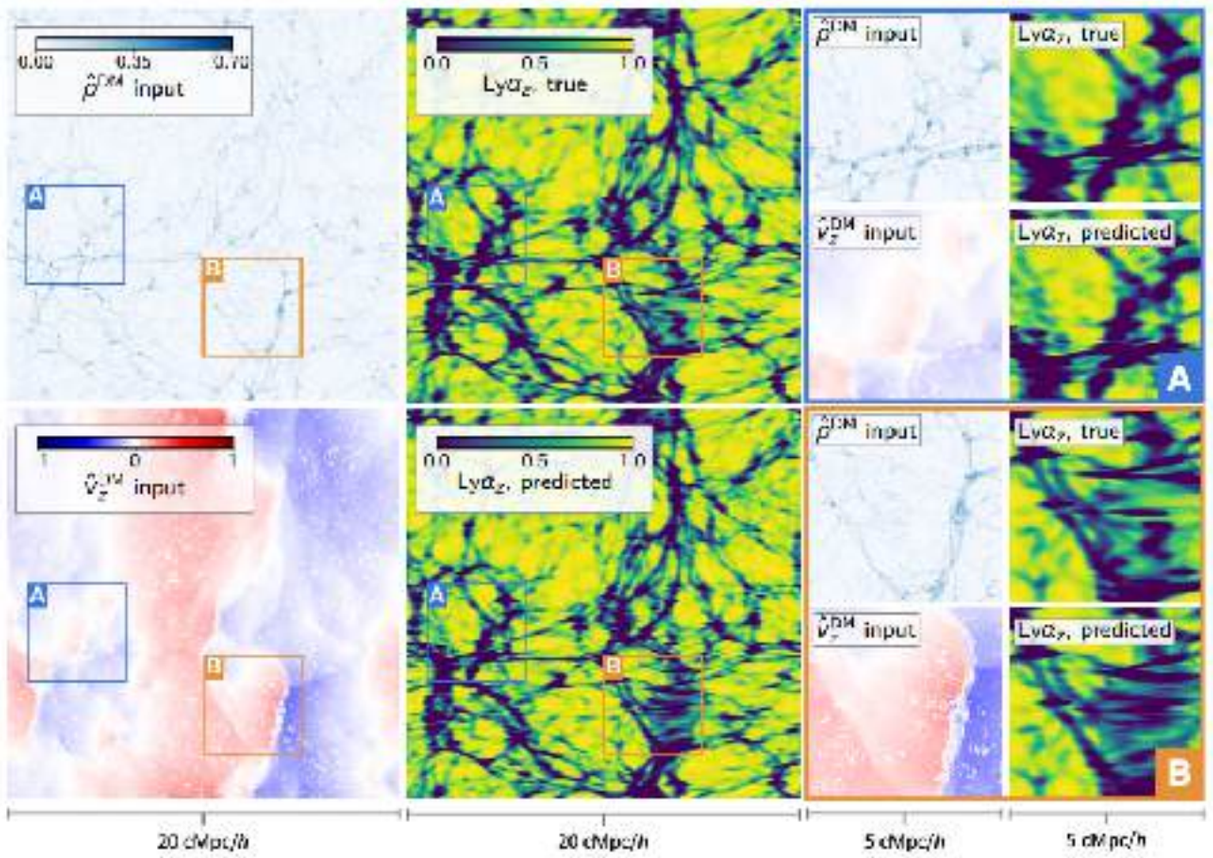
Buck+21

Hazevah+17, Perreault-Levasseur+17, Morningstar+18, Stark+18, Tuccillo+18, Bom+19, Lovell+19, Madireddy+19, Pearson+19, Simet+19, Pasquet+19, Wu+19, Menou+19, Aragon-Calvo+20, Chianese+20, Stahl+20, Surana+20, Shuntov+20, Cabayol-Garcia+20, Campagne+20, Buck+21, Grover+21, Hayat+21, Li+21, Maresca+21, Qiu+21, Rhea+21, Schuldt+21, Tohill+21, Yao-Yu+21, Dey+21, Lee+21, Zhou+21, Henghes+21, Ansari+21

Painting Baryons

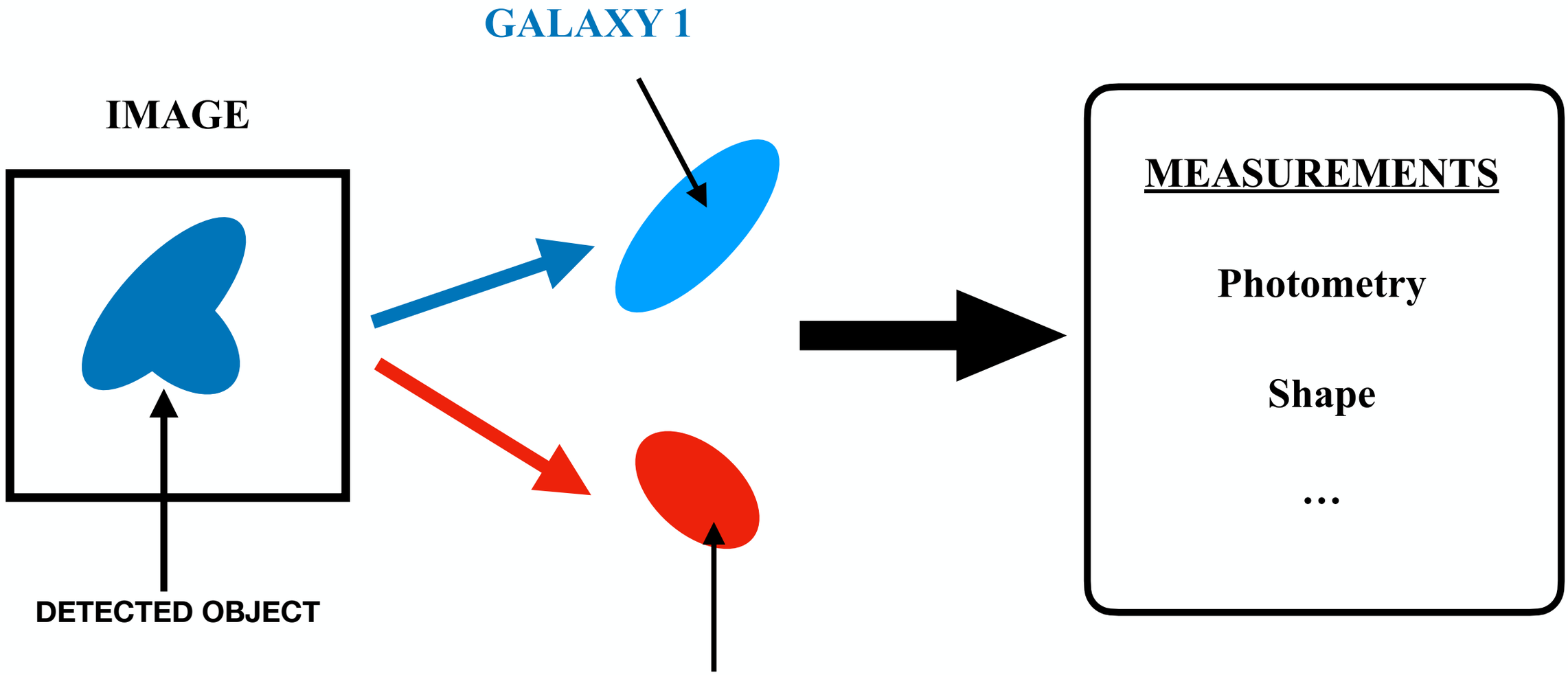


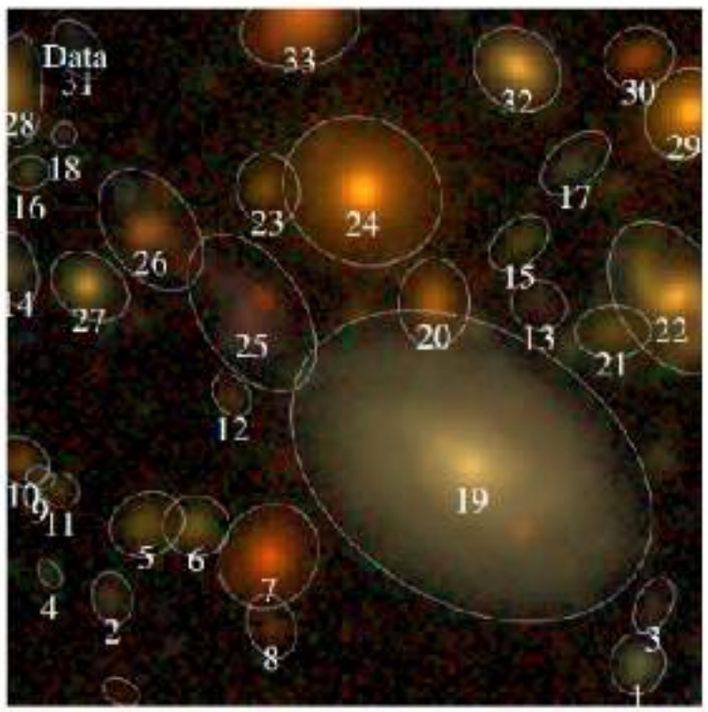
Harrington+21



Neural Networks are used to learn the non-linear mapping between cheap dark matter only simulations to expensive baryonic physics

Rodriguez+19, Modi+18, Berger+18, He+18, Zhang+19, Troster+19, Zamudio-Fernandez+19, Perraudin+19, Charnock+19, List+19, Giusarma+19, Bernardini+19, Chardin+19, Mustafa+19, Ramanah+20, Tamosiunas+20, Feder+20, Moster+20, Thiele+20, Wadekar+20, Dai+20, Li+20, Lucie-Smith+20, Kasmanoff+20, Ni+21, Rouhiainen+21, Harrington+21, Horowitz+21, Horowitz+21, Bernardini+21, Schaurecker+21, Etezad-Razavi+21, Curtis+21





Bosch+17

>50% of objects will be affected by blending in future deep surveys such as LSST

Boucaud+19

GALAXY 1

IMAGE

DETECTED OBJECT

GALAXY 2

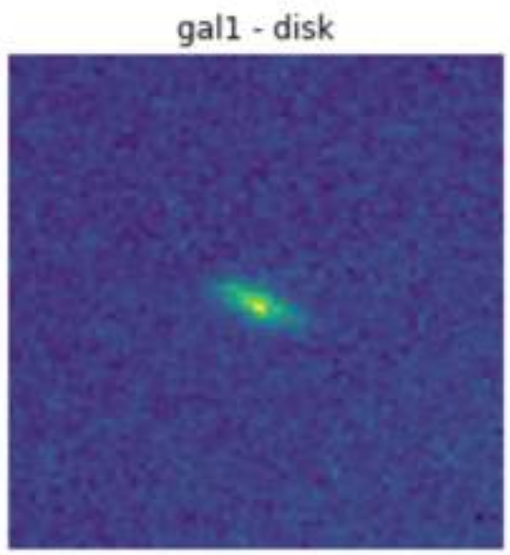
MEASUREMENTS

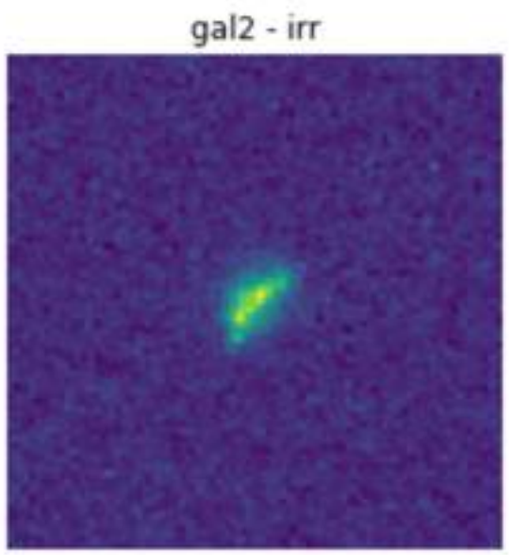
Photometry

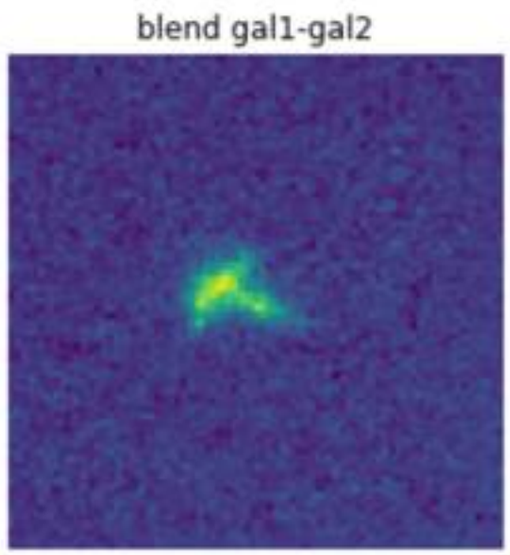
Shape

…

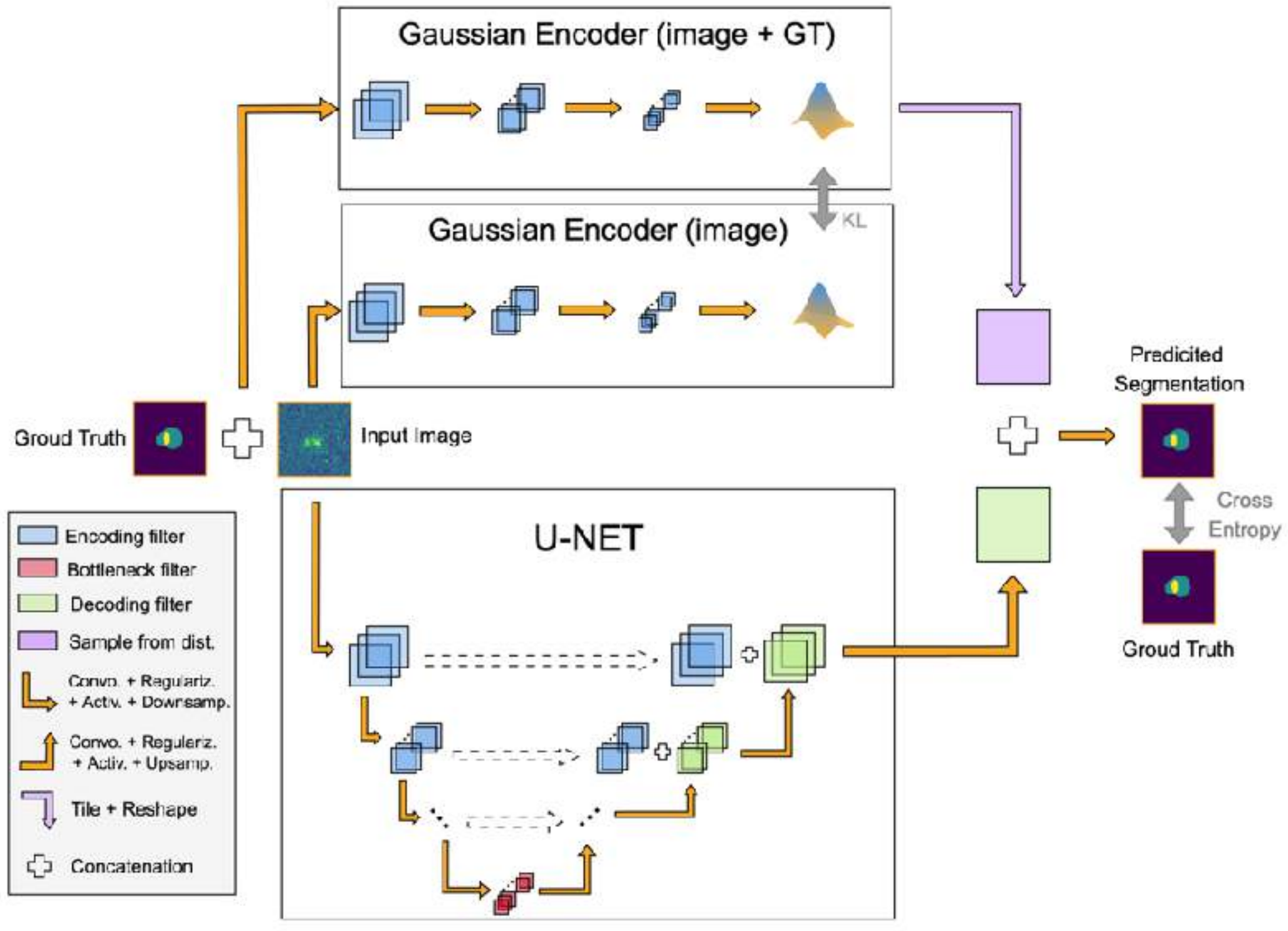
ISOLATED GALAXIES ARTIFICALLY BLENDED







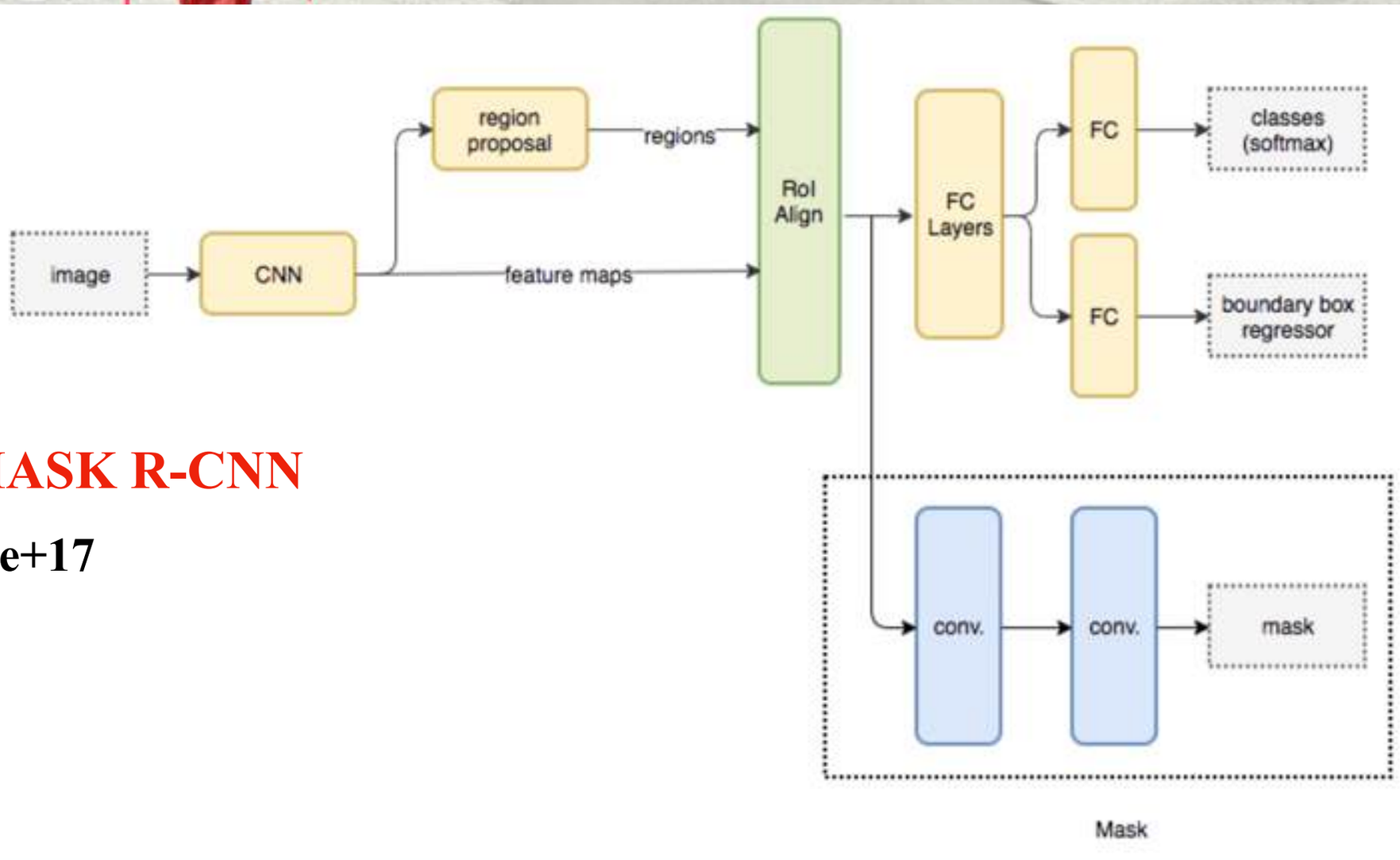
PROBABILISTIC U-NET



Bretonniere+20

instance segmentation

SIMULTANEOUS DETECTION + SEGMENTATION + CLASSIFICATION



MASK R-CNN

He+17

RECAP (yesterday):

Spatially correlated data in euclidean grid

CNN

$$p_w(\theta|X)$$

THE PRICE TO PAY?

1. LARGE NUMBER OF PARAMETERS IMPLIES LARGER DATASETS TO TRAIN

2. LOOSE EVEN MORE DEGREE OF CONTROL OF WHAT THE ALGORITHM IS DOING SINCE THE FEATURE EXTRACTION PROCESS BECOMES UNSUPERVISED

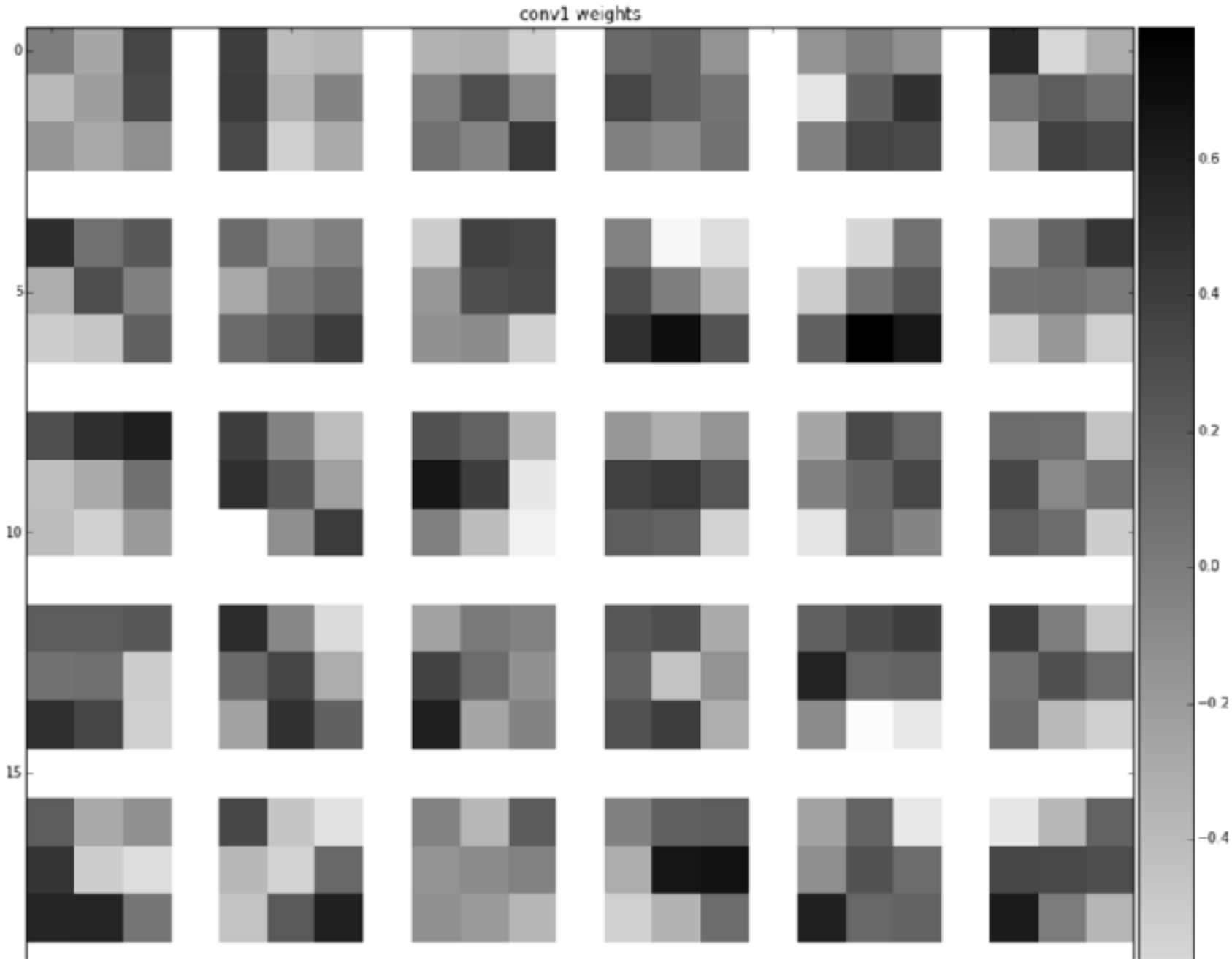
VISUALIZING CNNs

[intepreting CNN decisions]

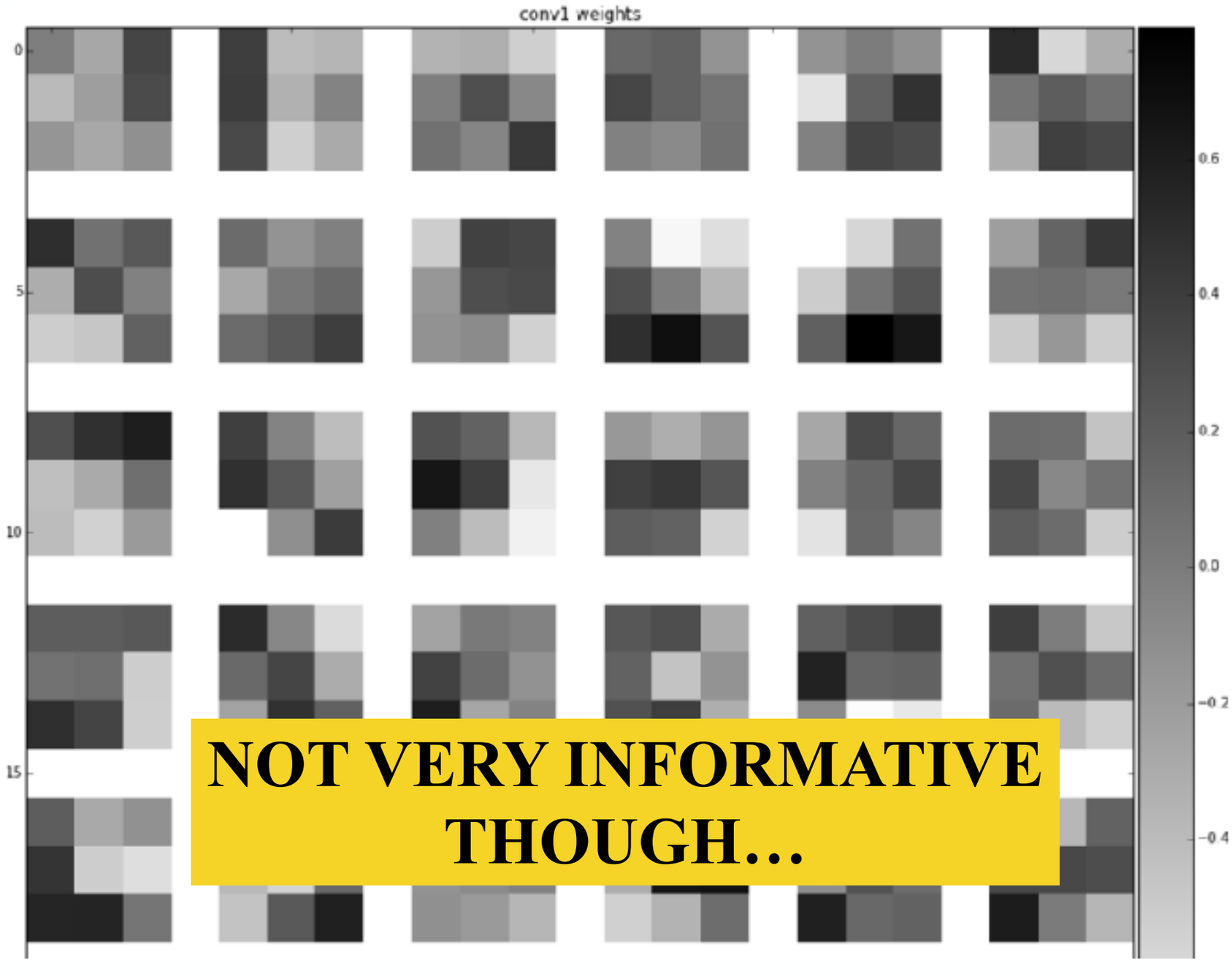
Attribution techniques

EXPLORING THE FEATURE MAPS

THE SIMPLEST APPROACH IS TO VISUALIZE THE LEARNED WEIGHTS AT INTERMEDIATE LAYERS



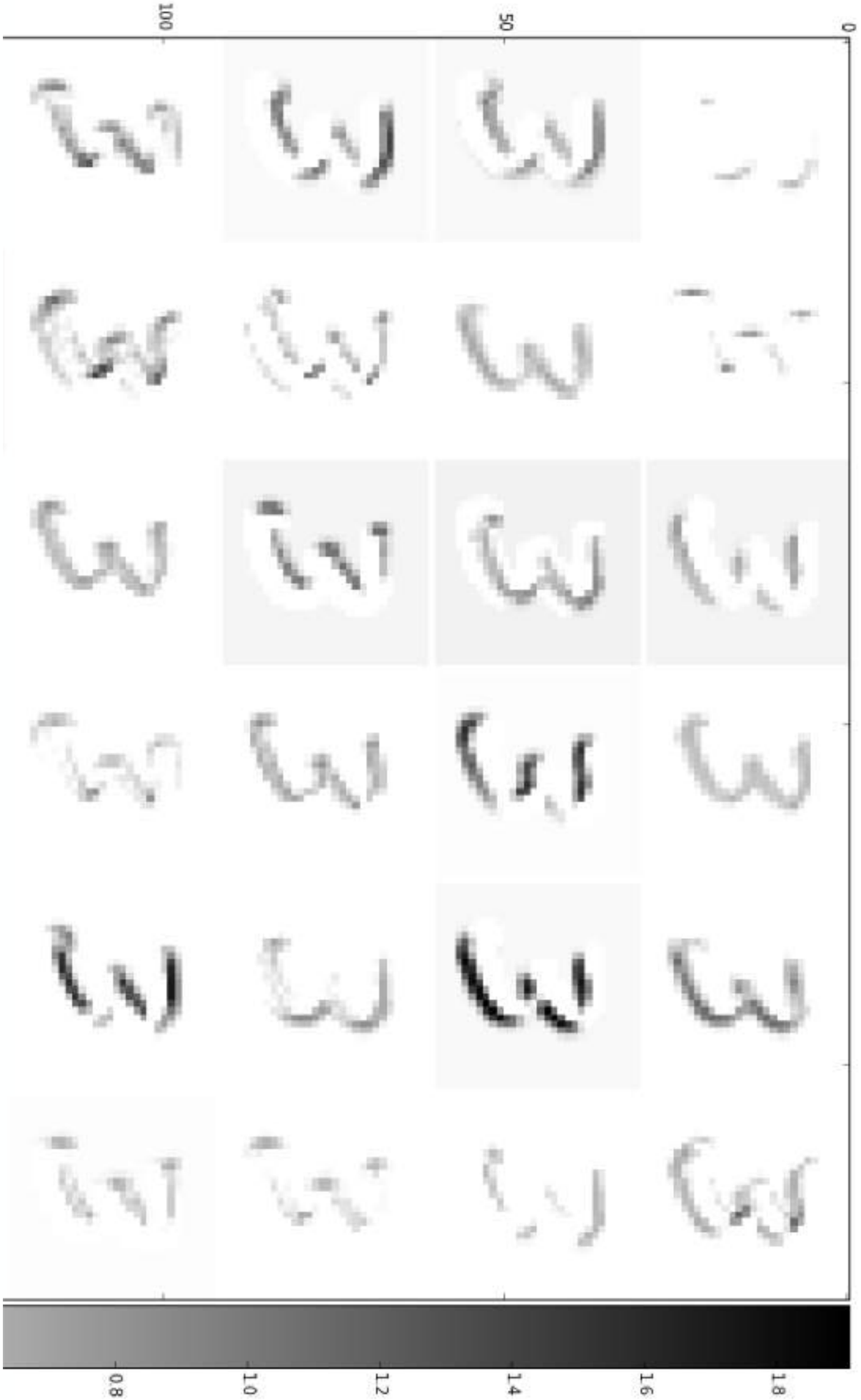
THE SIMPLEST APPROACH IS TO VISUALIZE THE LEARNED WEIGHTS AT INTERMEDIATE LAYERS



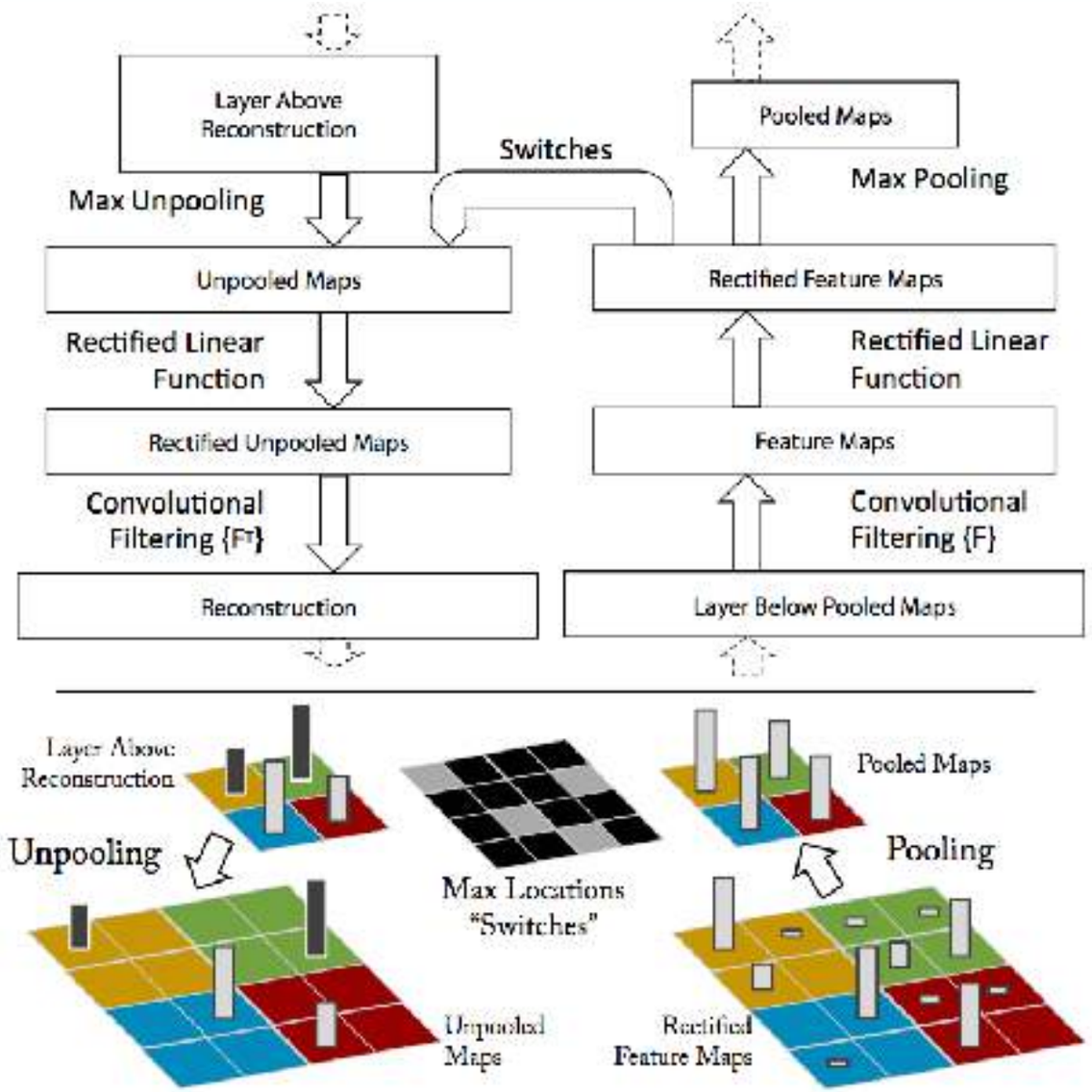
NOT VERY INFORMATIVE THOUGH…

USING THE SAME IDEA, ONE CAN ALSO VISUALIZE THE FEATURE MAPS AT INTERMEDIATE LAYERS

THIS HELPS TRACING THE FEATURES LEARNED BY THE NETWORK

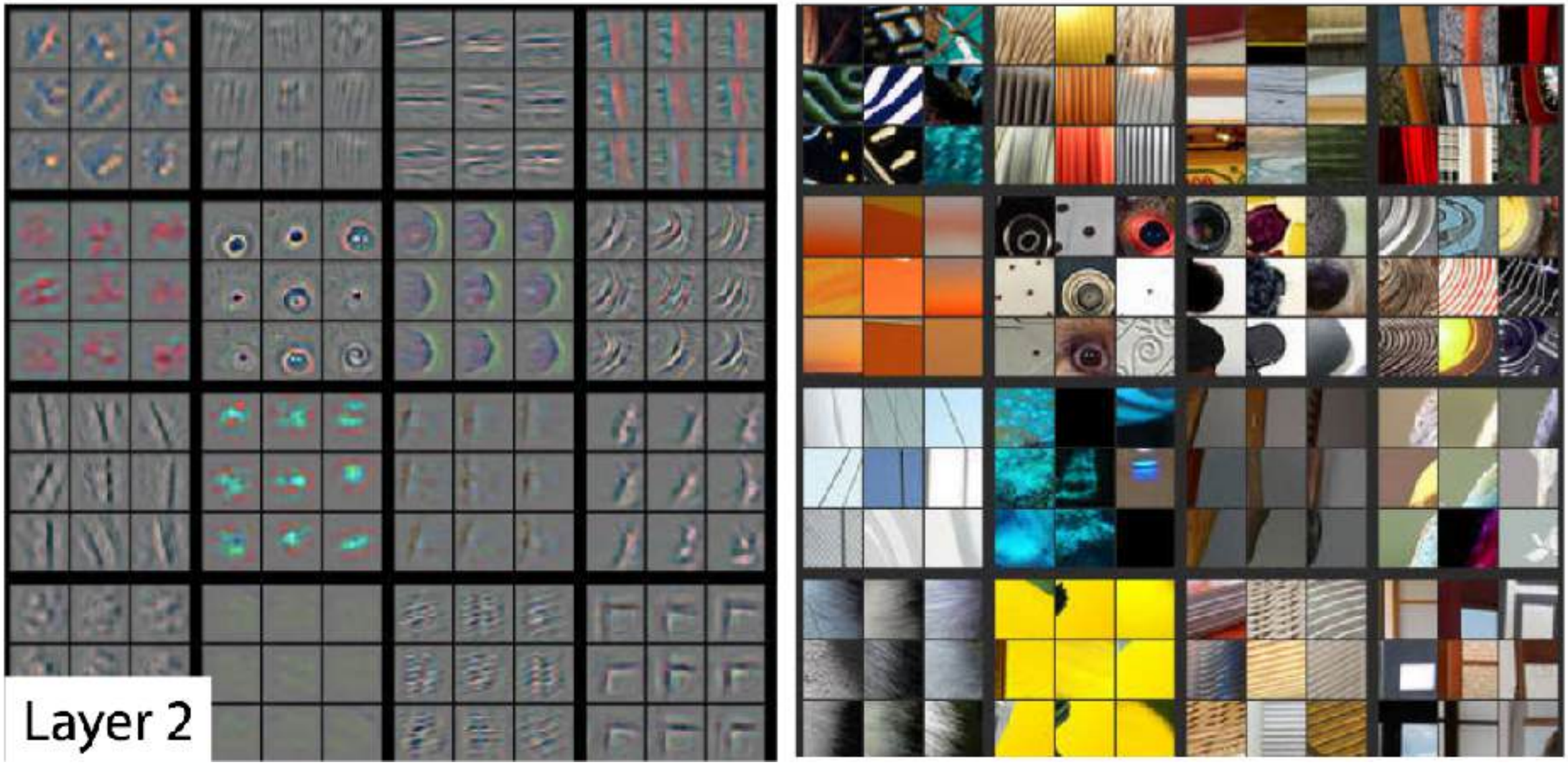

0
50
100
0.8
1.0
1.2
1.4
1.6
1.8

USE “DECONVNETS” TO MAP BACK THE FEATURE MAP INTO THE PIXEL SPACE



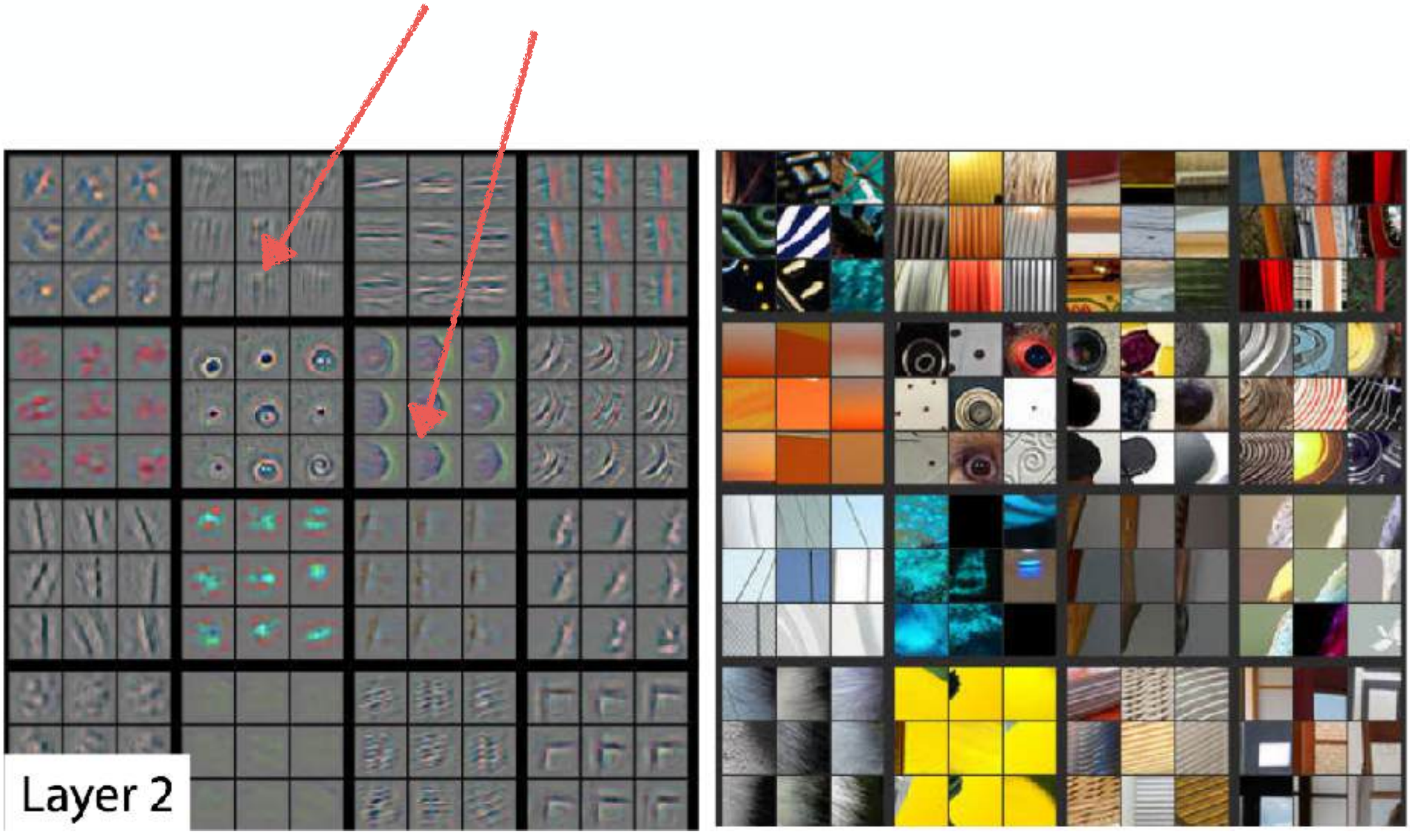
IT ALLOWS TO SEE WHICH REGIONS OF THE INPUT GENERATED A MAXIMUM RESPONSE IN A NEURON

Zeiler+14



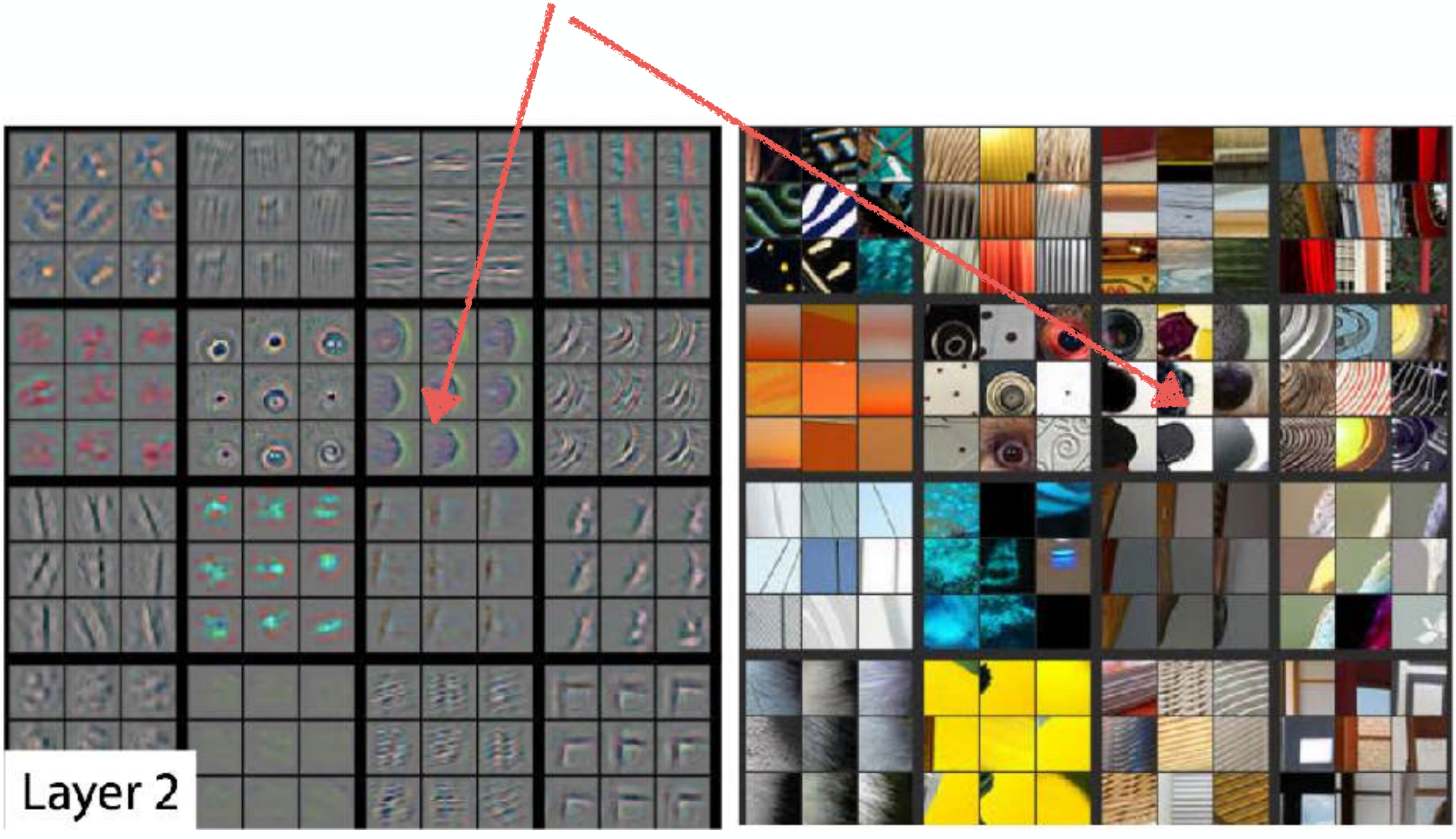
Zeiler+14

EVERY BLOCK OF 9 SHOWS
THE 9 STRONGEST RESPONSES TO A GIVEN FILTER OF LAYER2



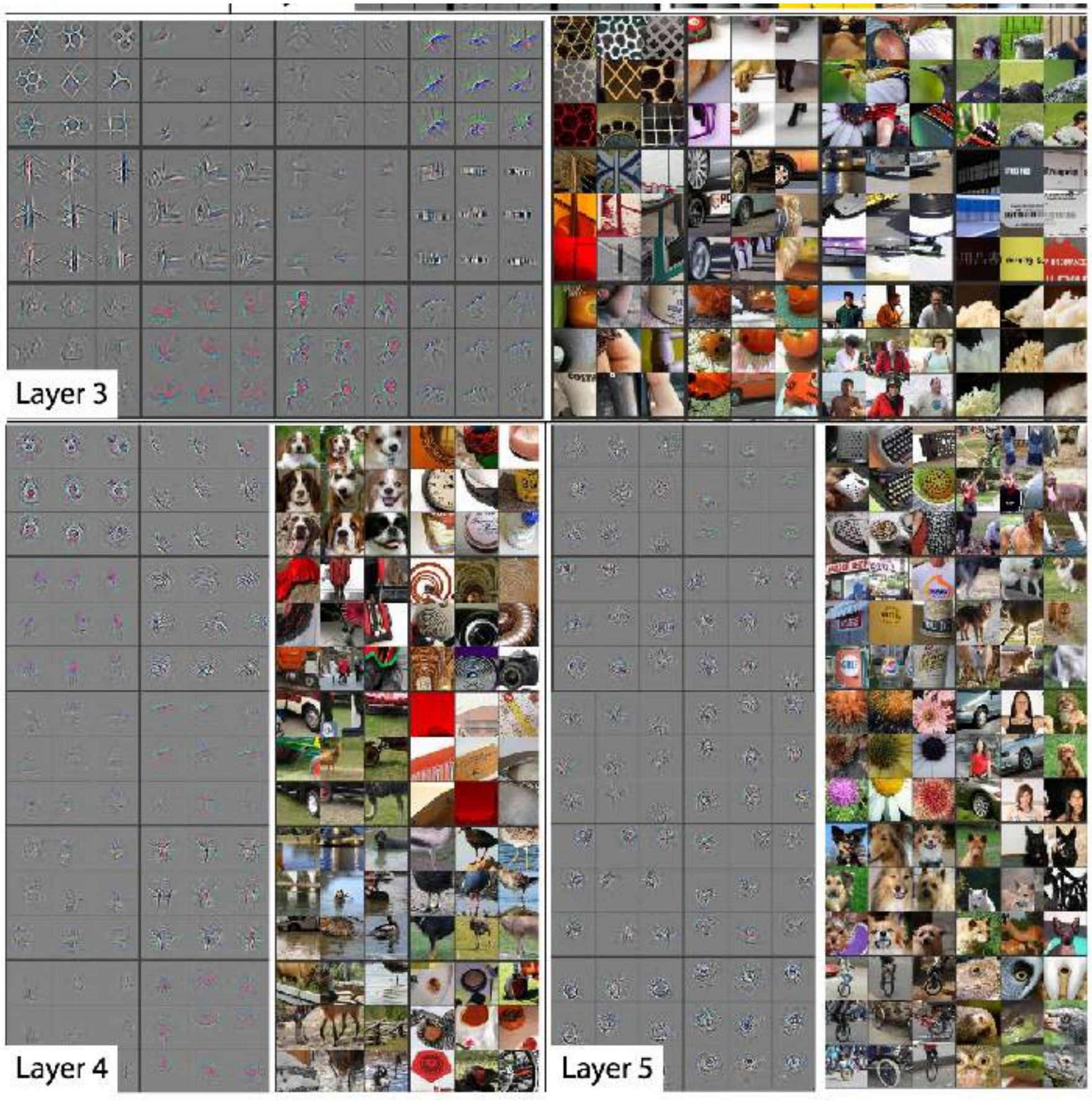
Zeiler+14

THE CORRESPONDING REGIONS OF IMAGES THAT GENERATED THE MAXIMUM RESPONSE



Zeiler+14

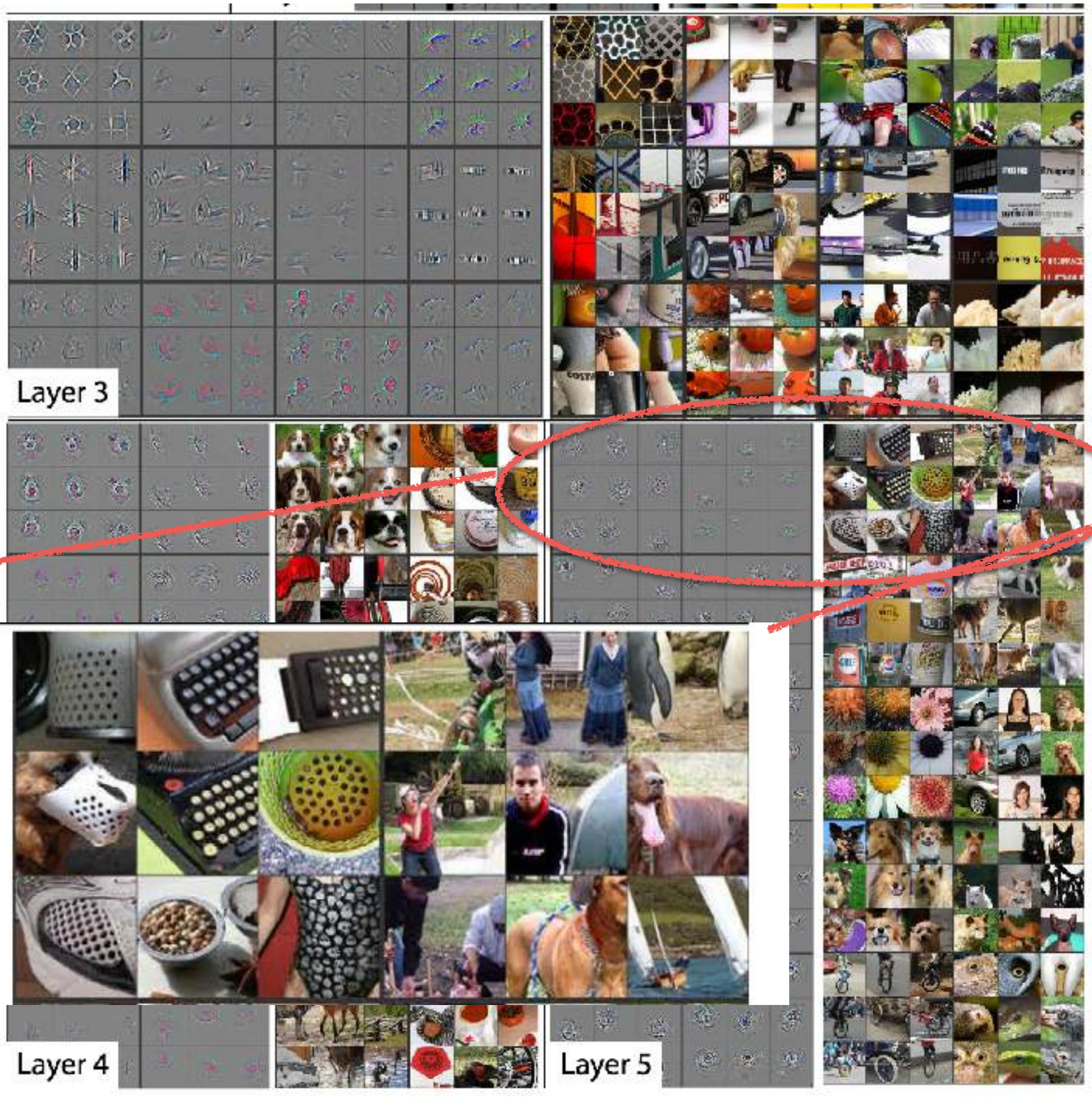
CAN BE REPEATED FOR DEEPER LAYERS ALTHOUGH IT BECOMES LESS INTUITIVE



Zeiler+14

CAN BE REPEATED FOR DEEPER LAYERS ALTHOUGH IT BECOMES LESS

Zeiler+14



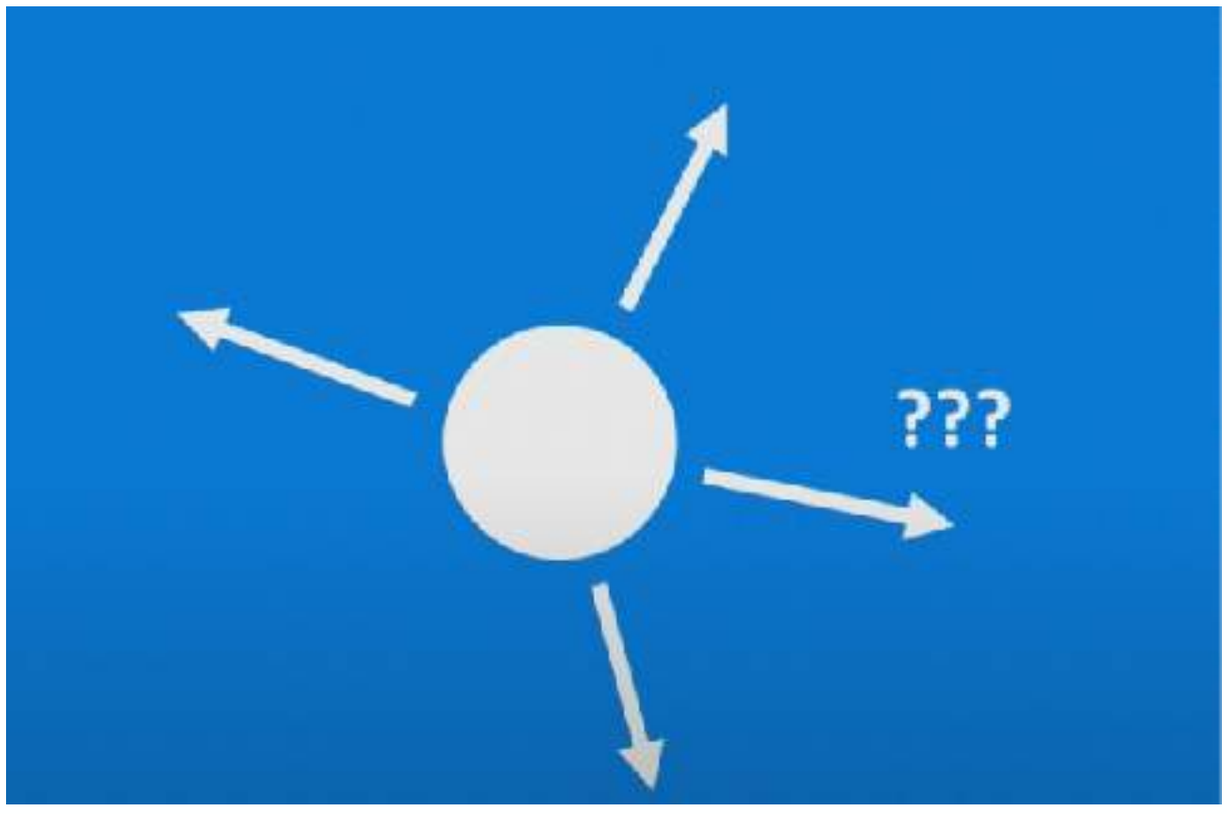
2. Neural Networks for Sequence Modelling

$$p_w(\theta|X)$$

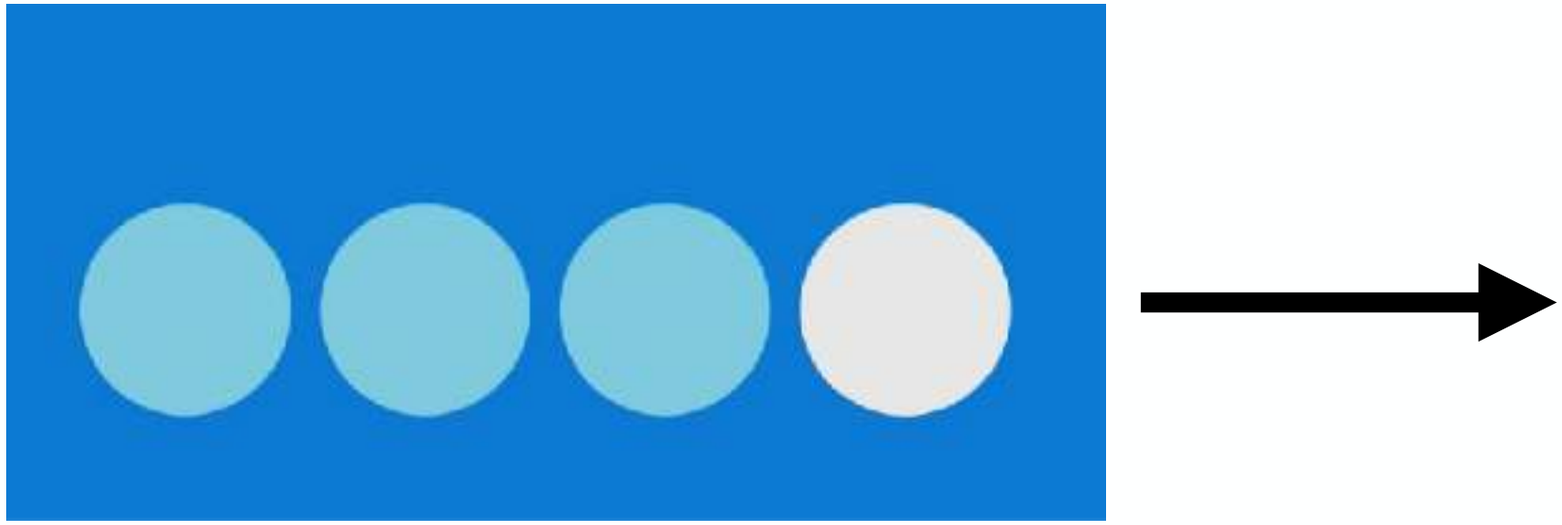
X={Time Series}

*based on MIT Lecture by Ava Soleimany

GIVEN AN IMAGE OF A BALL, CAN YOU PREDICT WHERE IT WILL GO NEXT?

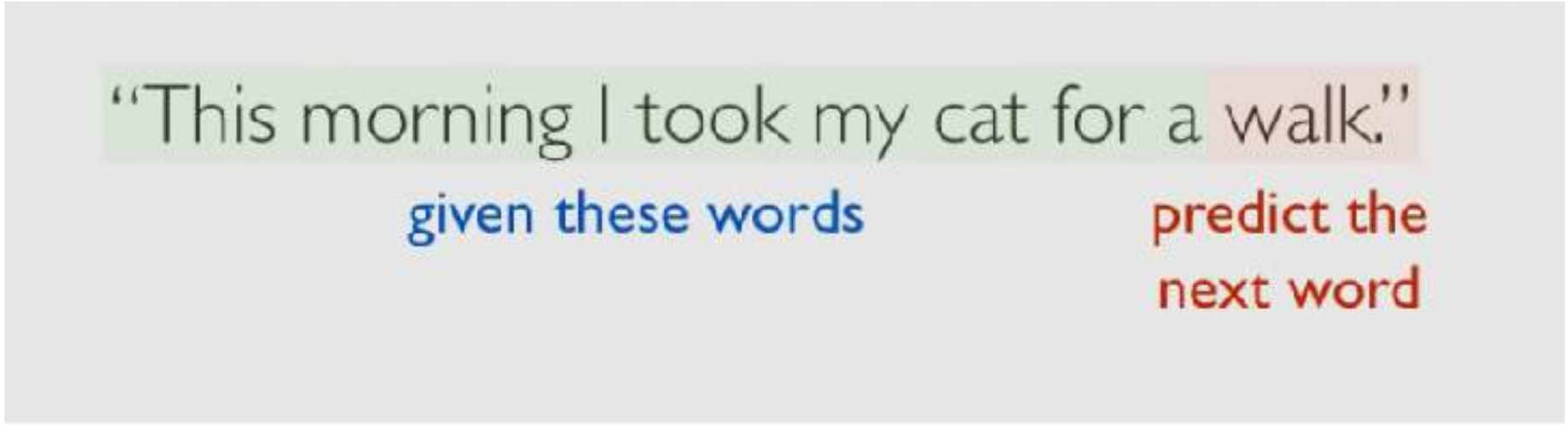


GIVEN AN IMAGE OF A BALL, CAN YOU PREDICT WHERE IT WILL GO NEXT?

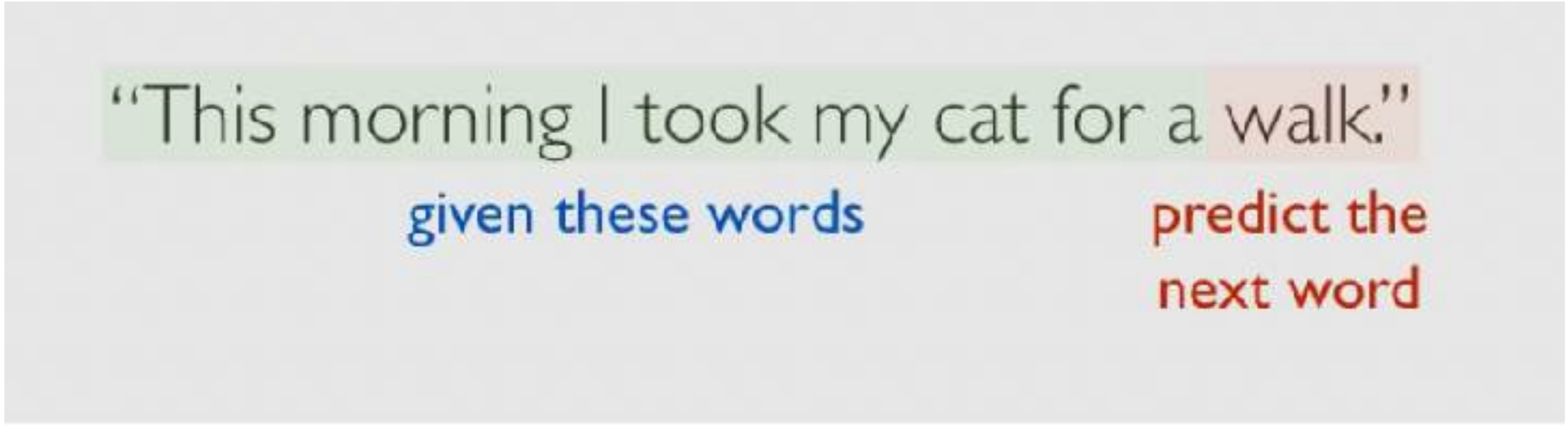


Previous positions help guessing the future

LET’S TAKE A SIMPLE EXAMPLE OF LANGUAGE MODELLING

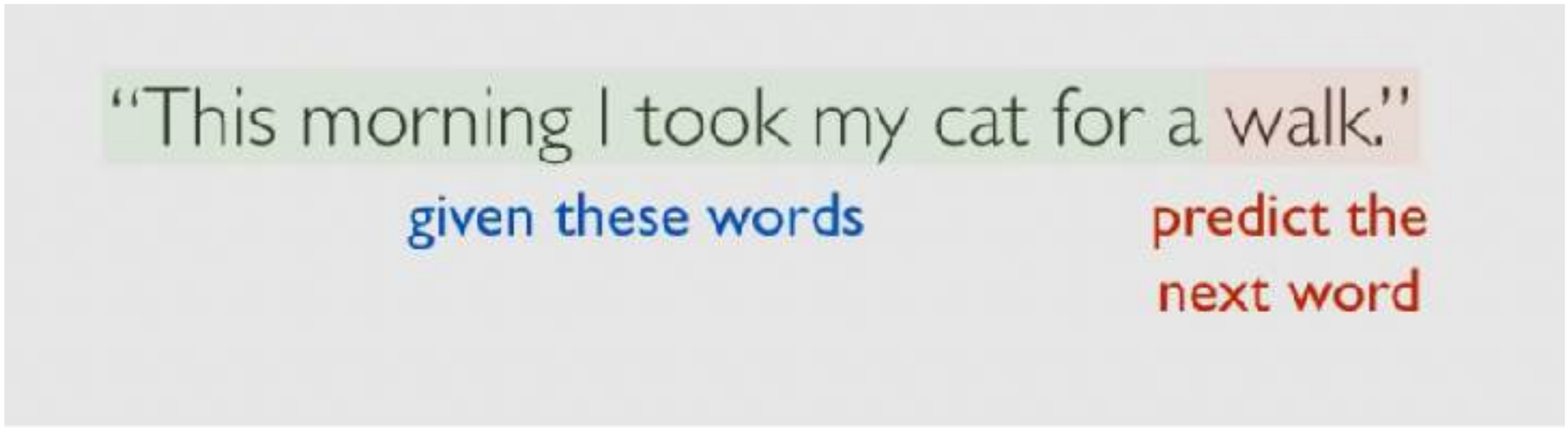


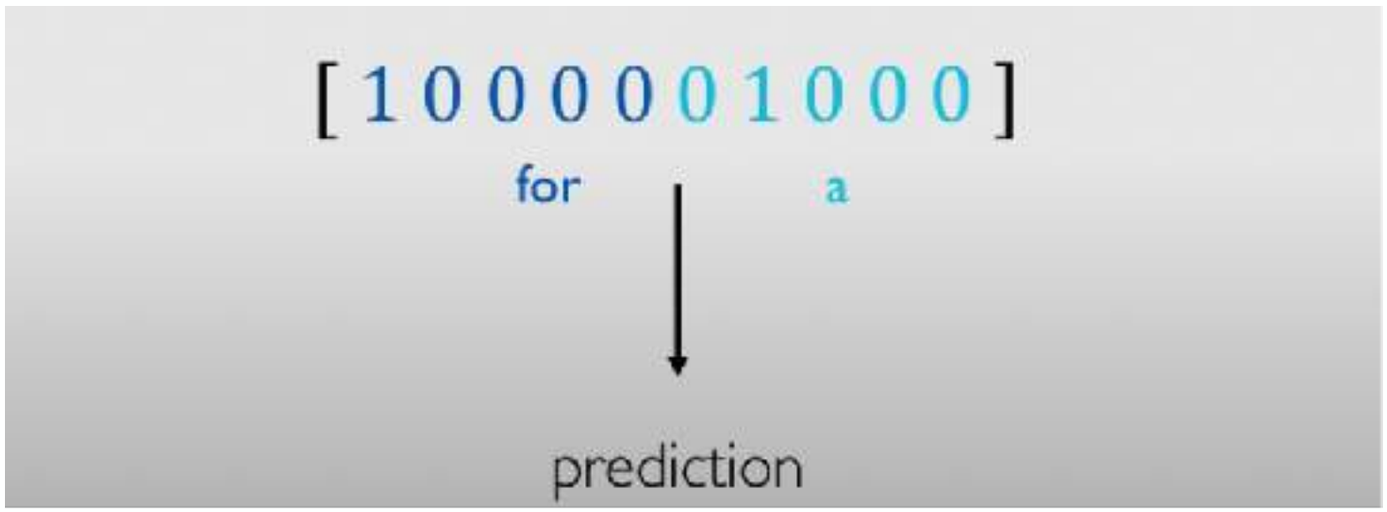
LET’S TAKE A SIMPLE EXAMPLE OF LANGUAGE MODELLING



Normal ANNs and CNNs cannot handle variable length inputs …

WE COULD SIMPLY USE A FIXED WINDOW…

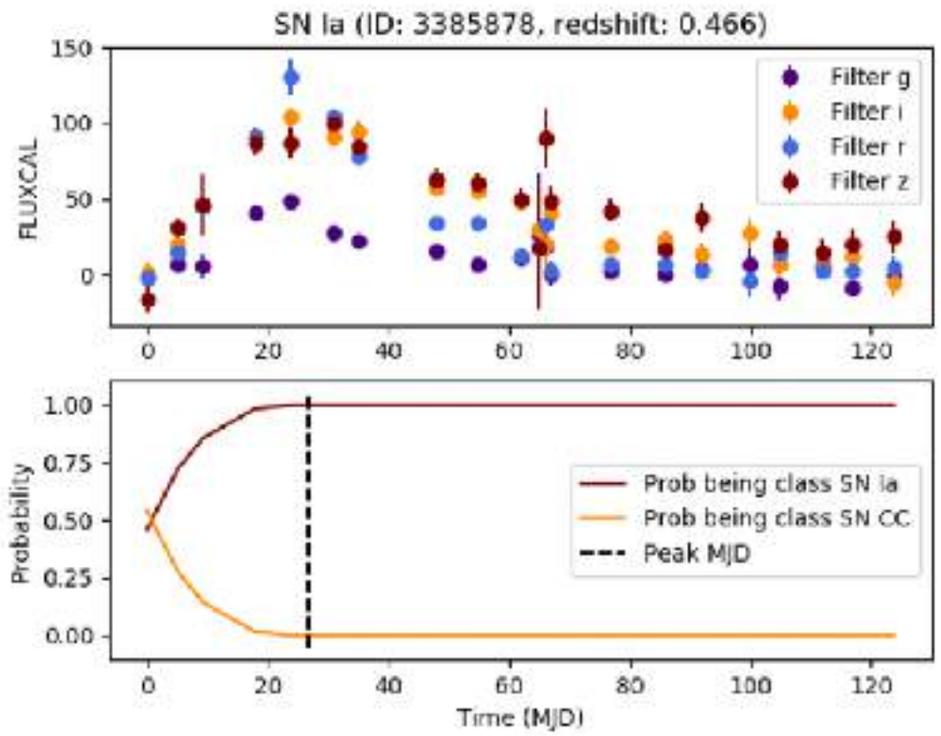




HOWEVER THIS CANNOT HANDLE LONG TERM MEMORY

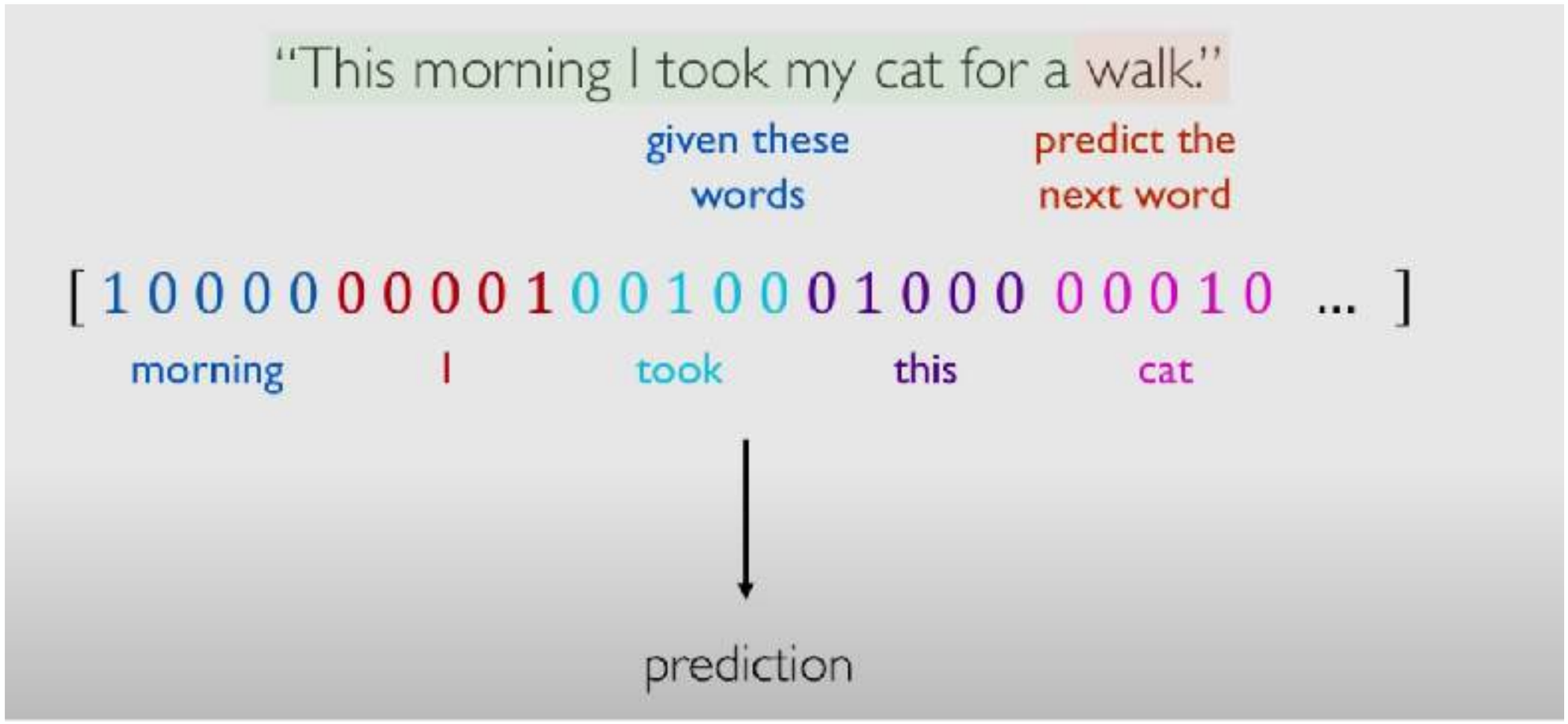
"**France** is where I grew up, but I now live in Boston. I speak fluent ___."

IN SOME CASES, INFORMATION FROM THE DISTANT PAST IS NEEDED FOR A CORRECT PREDICTION



Moller+19

ANOTHER ALTERNATIVE COULD BE TO FEED A REALLY LARGE WINDOW



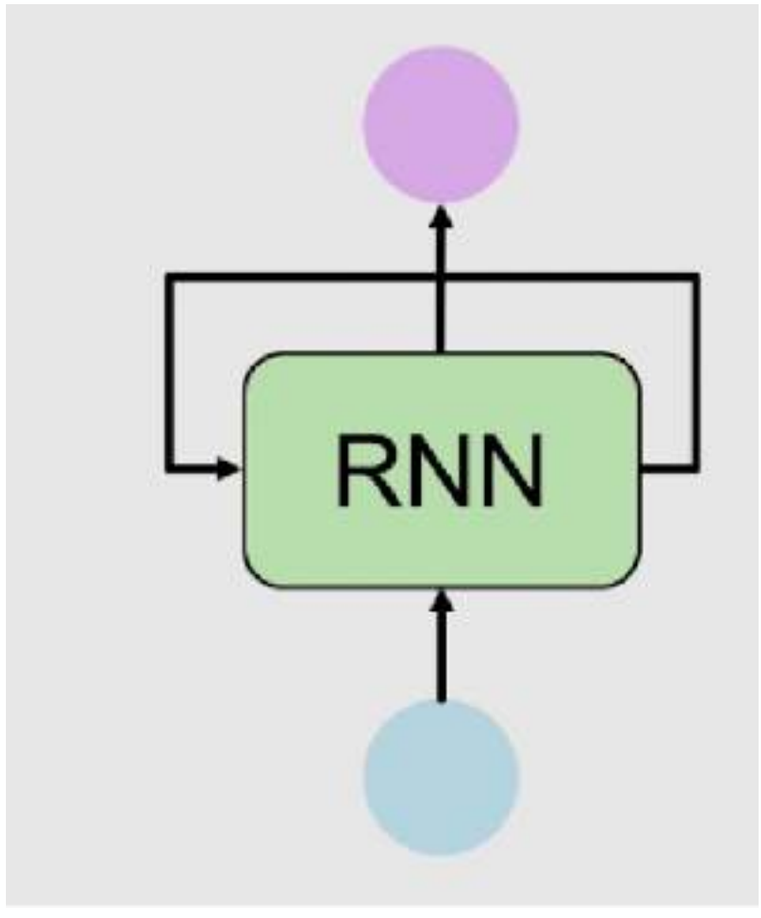
BUT WE STILL HAVE THE PROBLEM THAT THERE IS NO PARAMETER SHARING (SAME AS PIXELS)

Recurrent Neural Networks

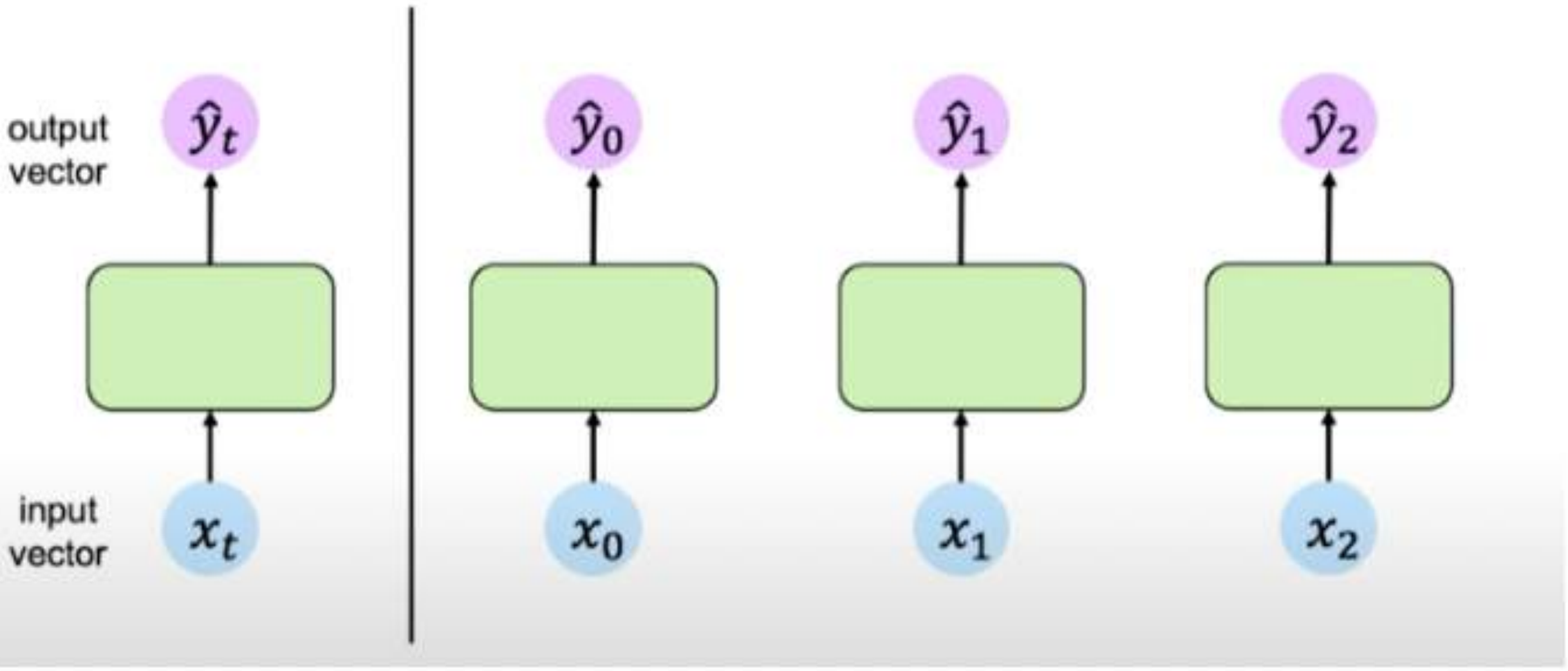
OUR WISH LIST:

1. Handle variable-length sequences
2. Track long term dependencies
3. Maintain information about order
4. Share parameters across the sequence

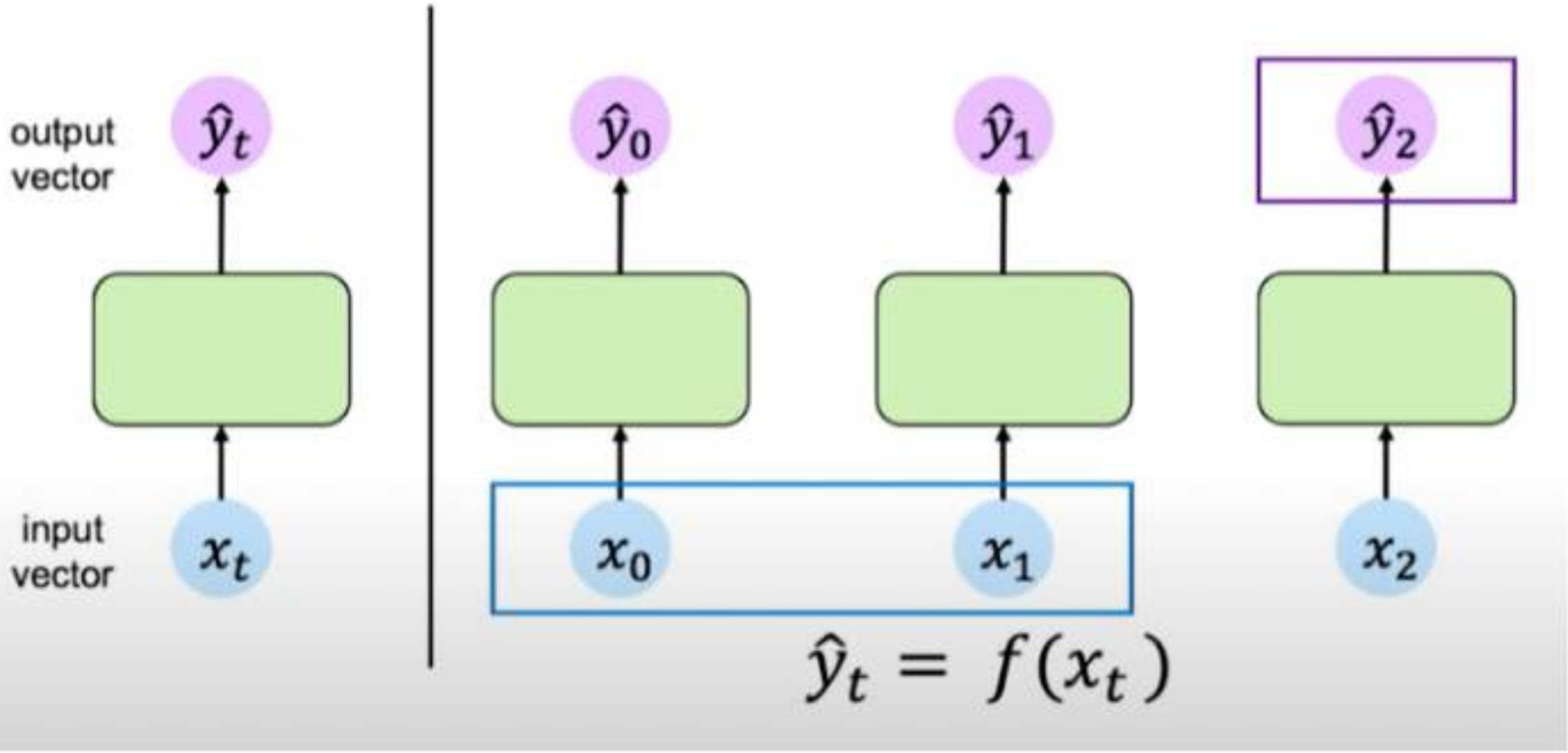
Recurrent Neural Networks
offer a first solution to this problem



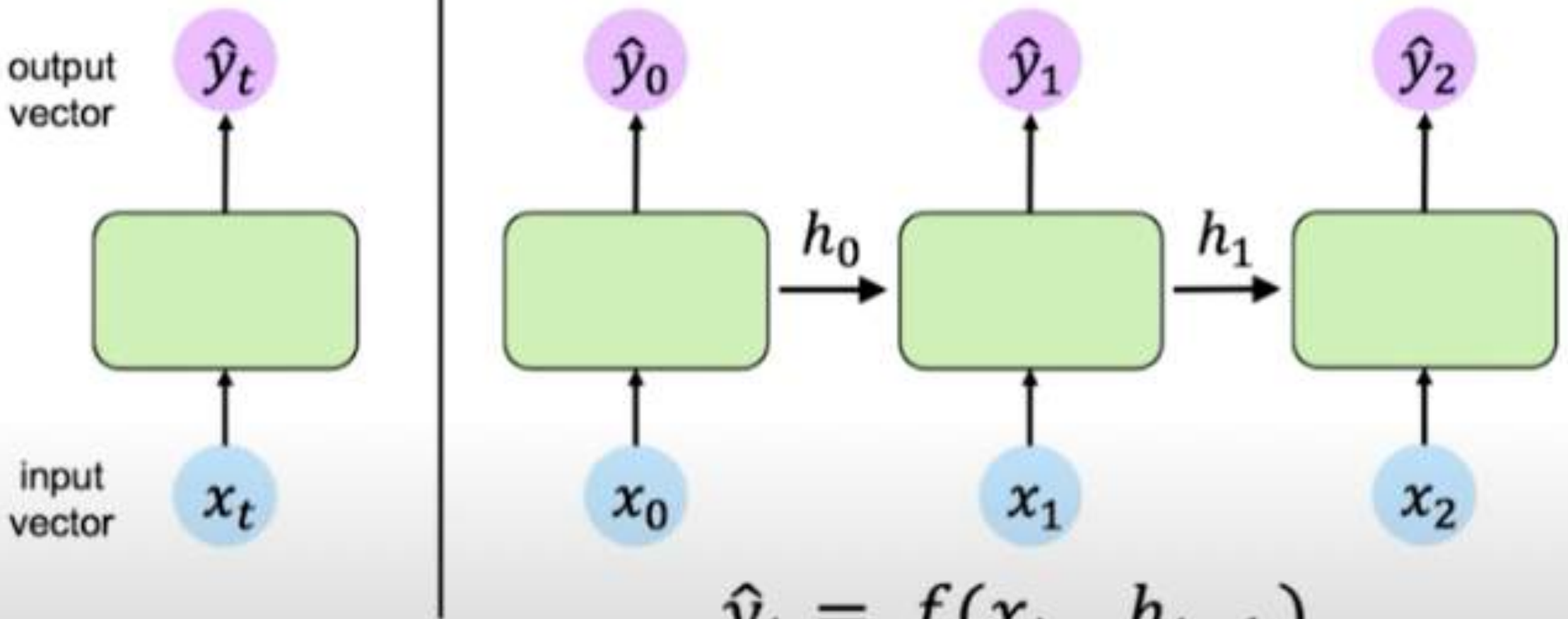
Handling Individual Time Steps



Handling Individual Time Steps

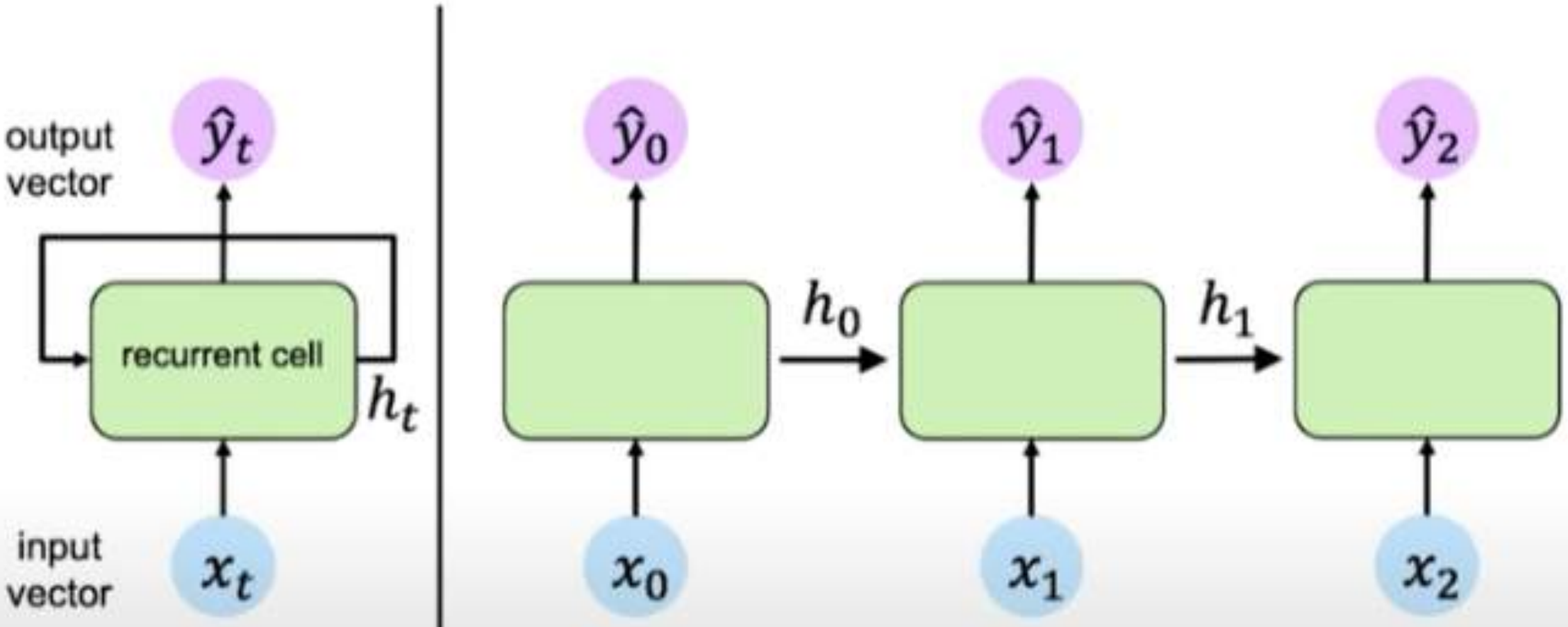


Neurons with Recurrence



$$\hat{y}_t = f(x_t, h_{t-1})$$

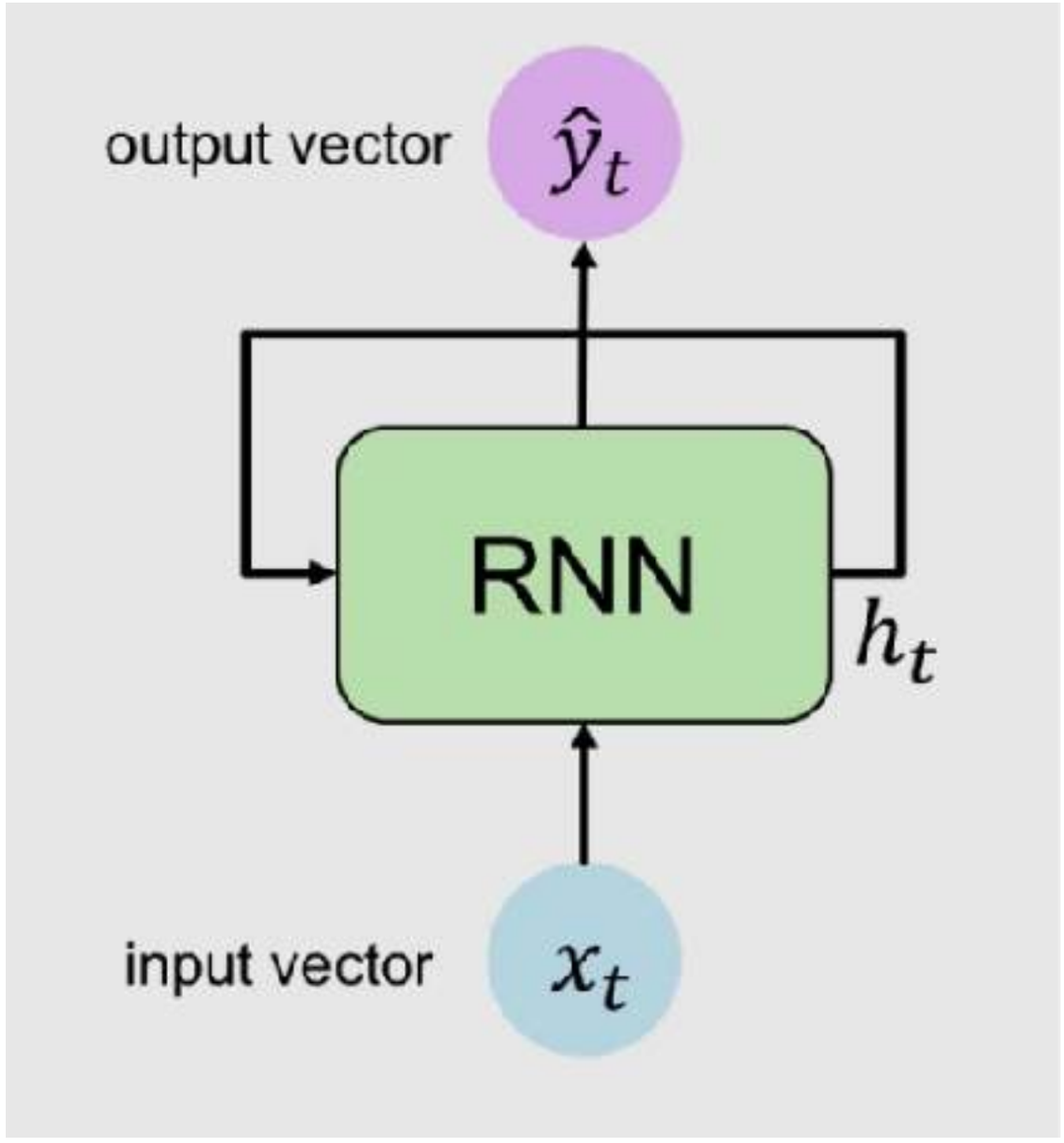
Neurons with Recurrence



$$\underline{\hat{y}_t} = f(\underline{x_t}, \underline{h_{t-1}})$$

output input past memory

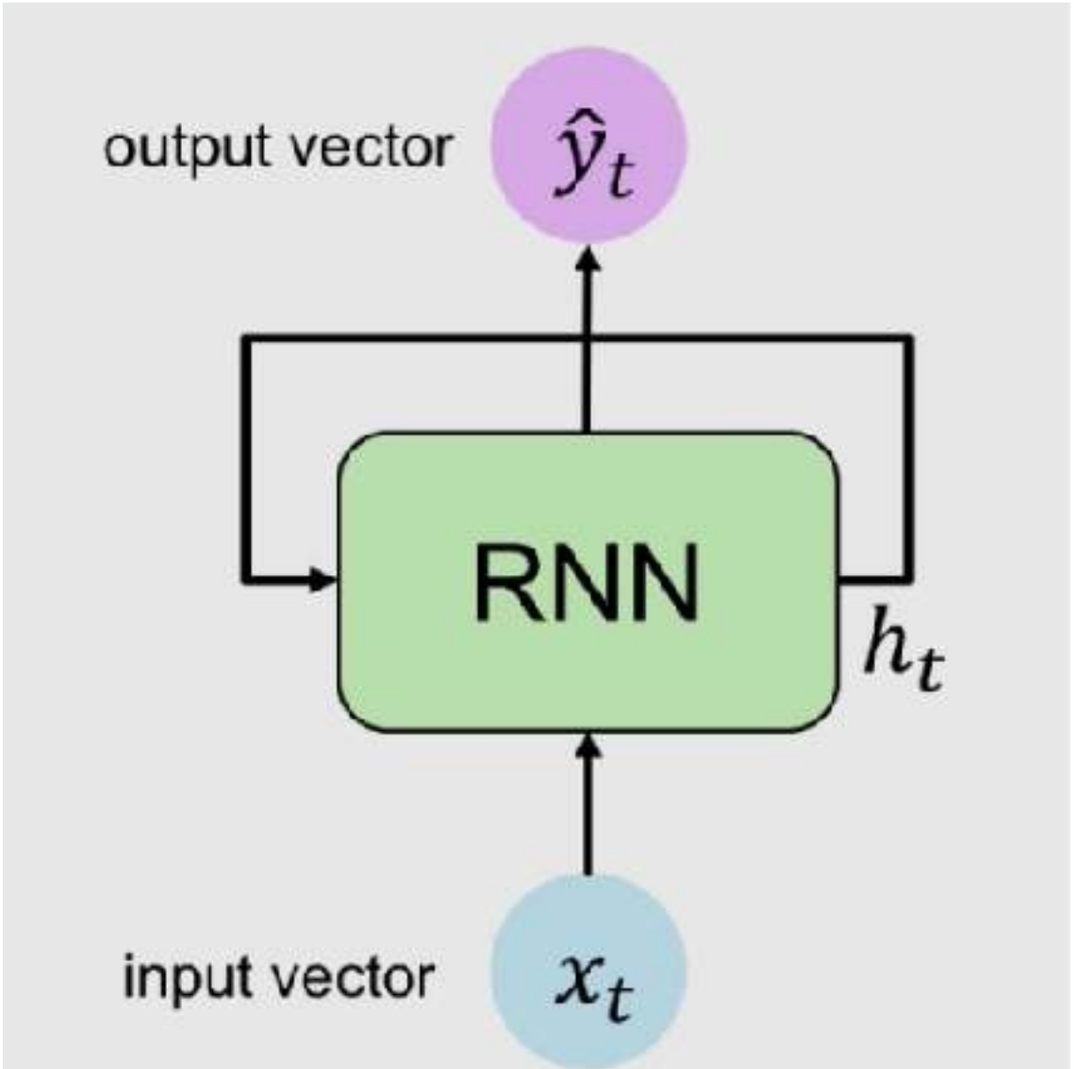
The RNN Block



$$h_t = f_W(h_{t-1}, x_t)$$

- h_t: cell state
- f_W: function parameterized by W
- h_{t-1}: old state
- x_t: input vector at time step t

The RNN Block



Output Vector

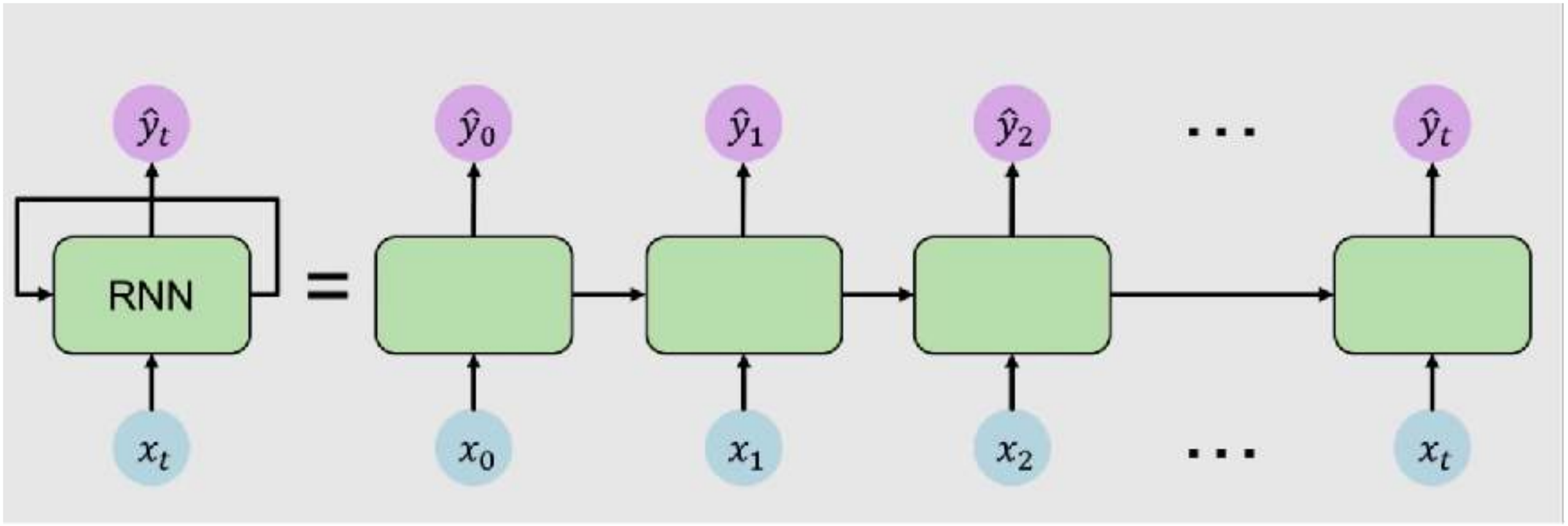
$$\hat{y}_t = \boldsymbol{W}_{\boldsymbol{hy}}^{\boldsymbol{T}} h_t$$

Update Hidden State

$$h_t = \tanh(\boldsymbol{W}_{\boldsymbol{hh}}^{\boldsymbol{T}} h_{t-1} + \boldsymbol{W}_{\boldsymbol{xh}}^{\boldsymbol{T}} x_t)$$

Input Vector

$$x_t$$


$\hat{y}_t$
RNN
=
$\hat{y}_0$
$\hat{y}_1$
$\hat{y}_2$
...
$\hat{y}_t$
x_t
x_0
x_1
x_2
...
x_t

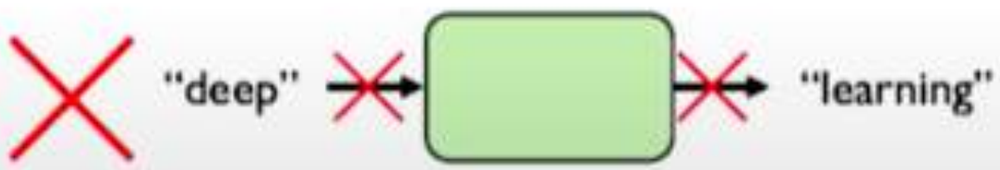
A Sequence Modeling Problem: Predict the Next Word

"This morning I took my cat for a walk."

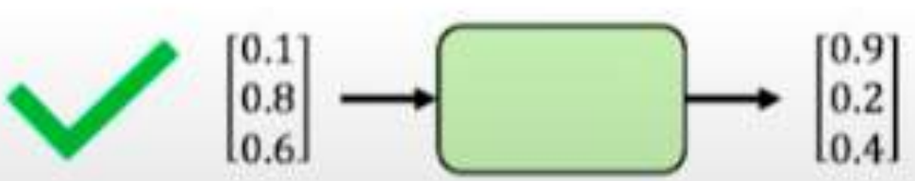
given these words

predict the next word

Representing Language to a Neural Network

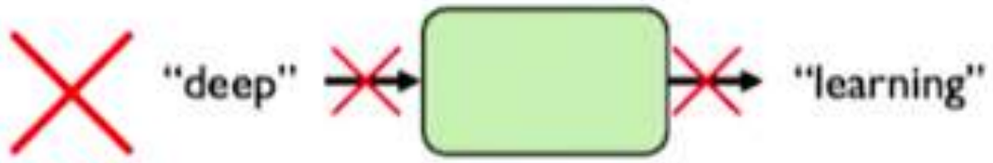


Neural networks cannot interpret words

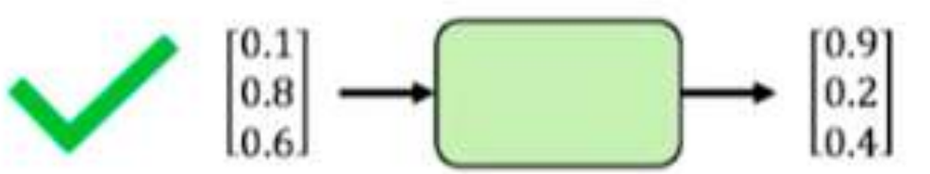


Neural networks require numerical inputs

Encoding Language for a Neural Network

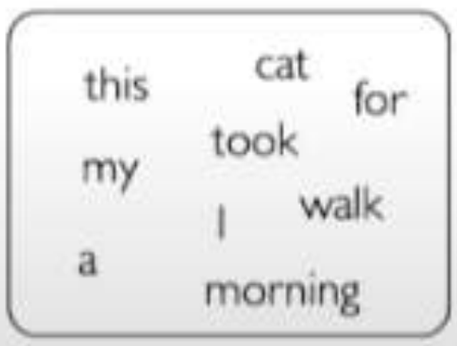


Neural networks cannot interpret words

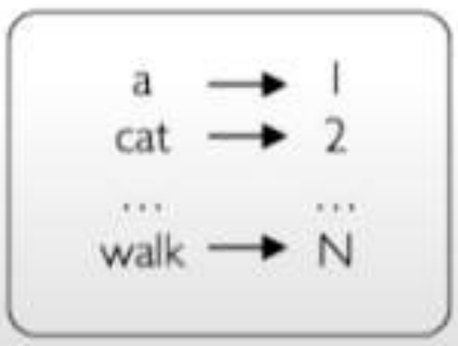


Neural networks require numerical inputs

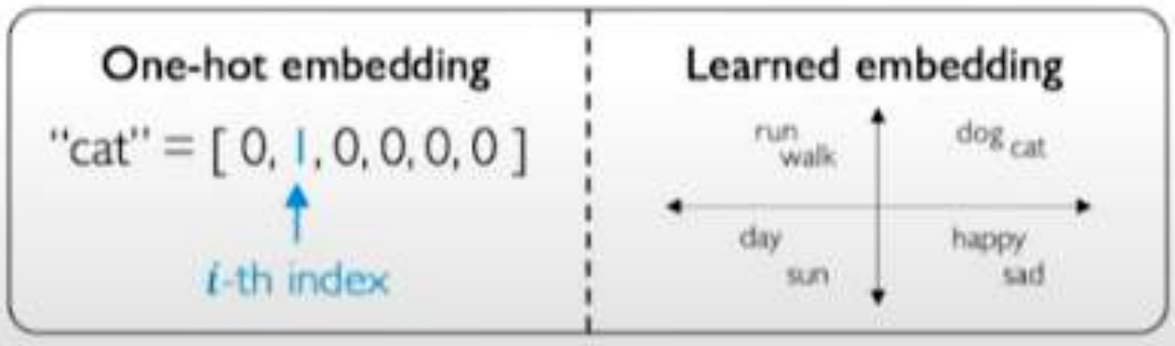
Embedding: transform indexes into a vector of fixed size.



1. Vocabulary:
Corpus of words

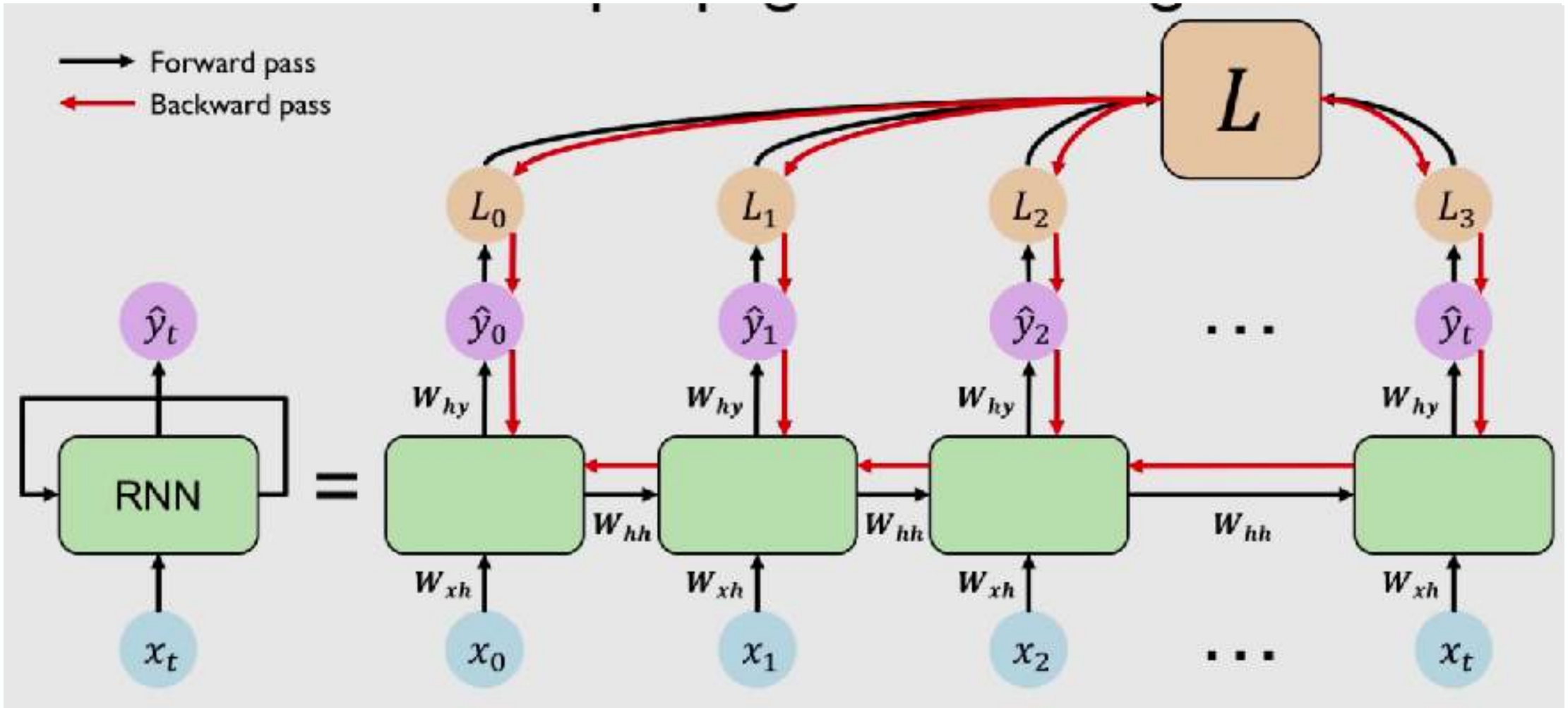


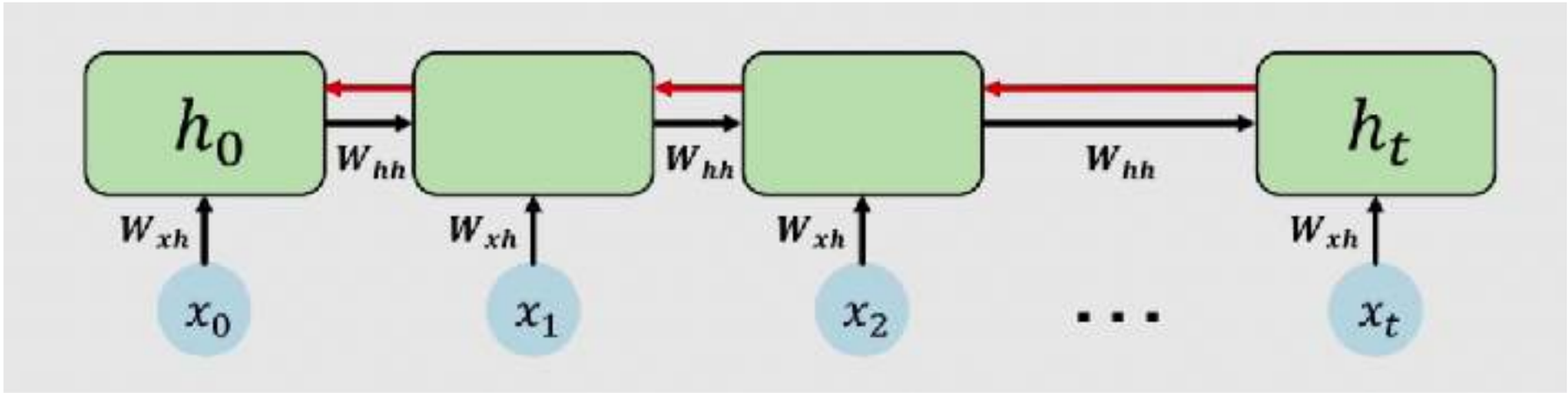
2. Indexing:
Word to index



3. Embedding:
Index to fixed-sized vector

BACKPROPAGATION THROUGH TIME (BPTT)

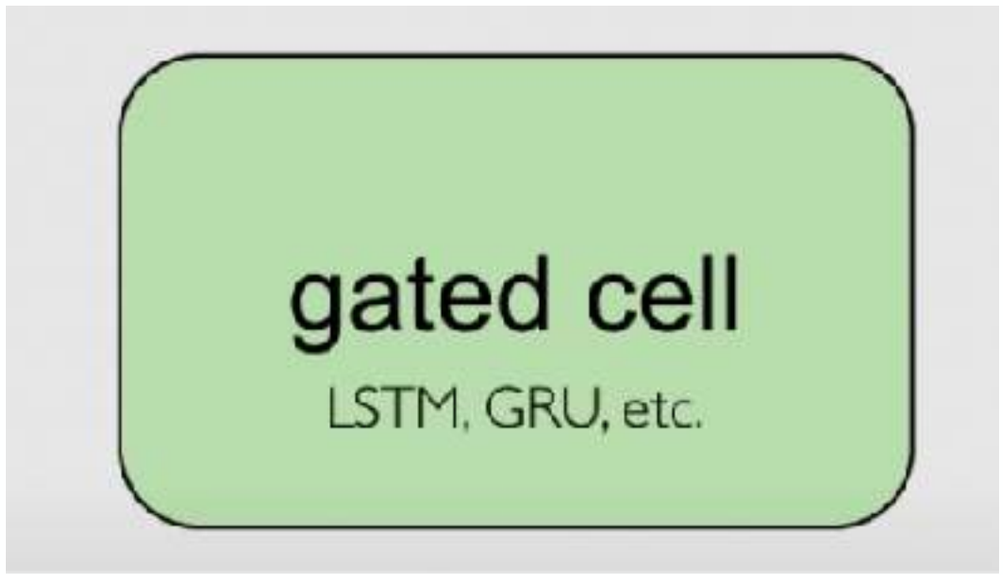




BPTT implies a large amount of weight multiplications: if we want to take into account long term memory, networks become quickly very deep and so are subjected to vanishing and exploding gradients problems (see previous lecture)

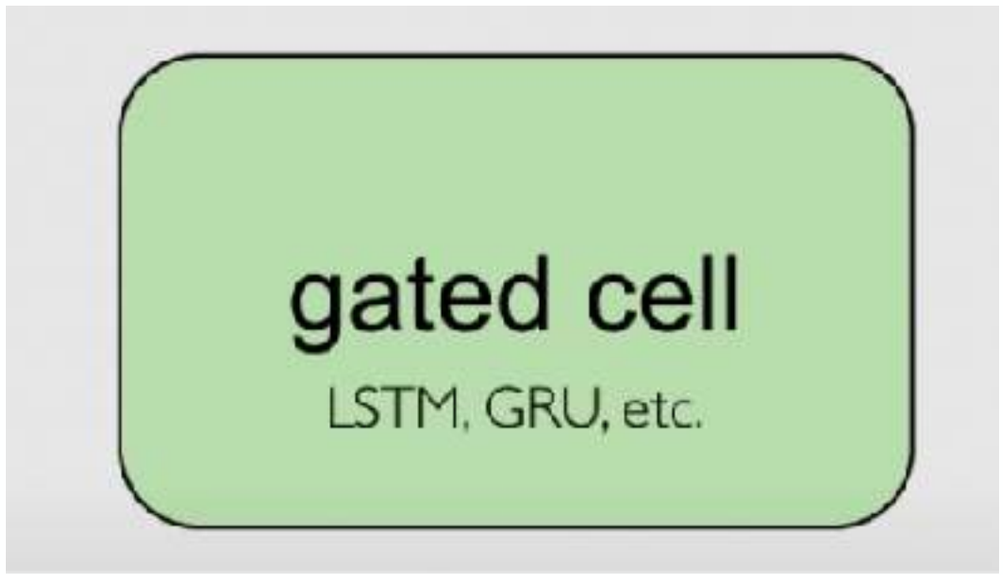
ANOTHER SPECIFIC SOLUTION: GATED CELLS

Use a more complex recurrent unit with gates to control what information is passed through…



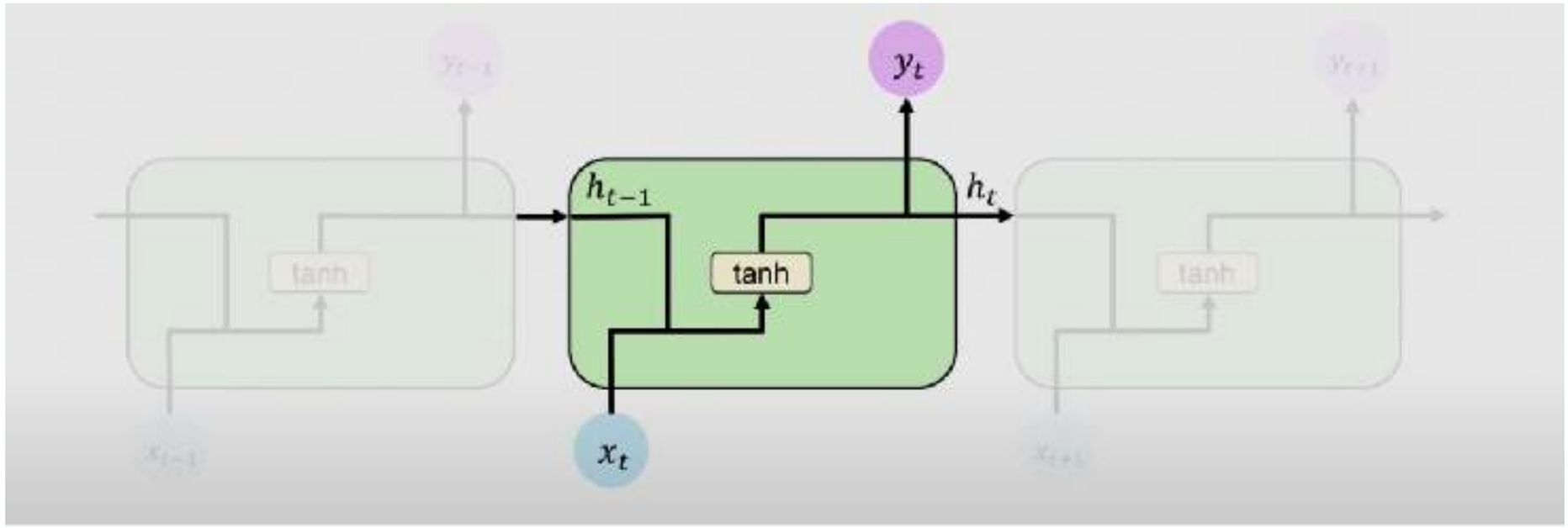
ANOTHER SPECIFIC SOLUTION: GATED CELLS

Use a more complex recurrent unit with gates to control what information is passed through…

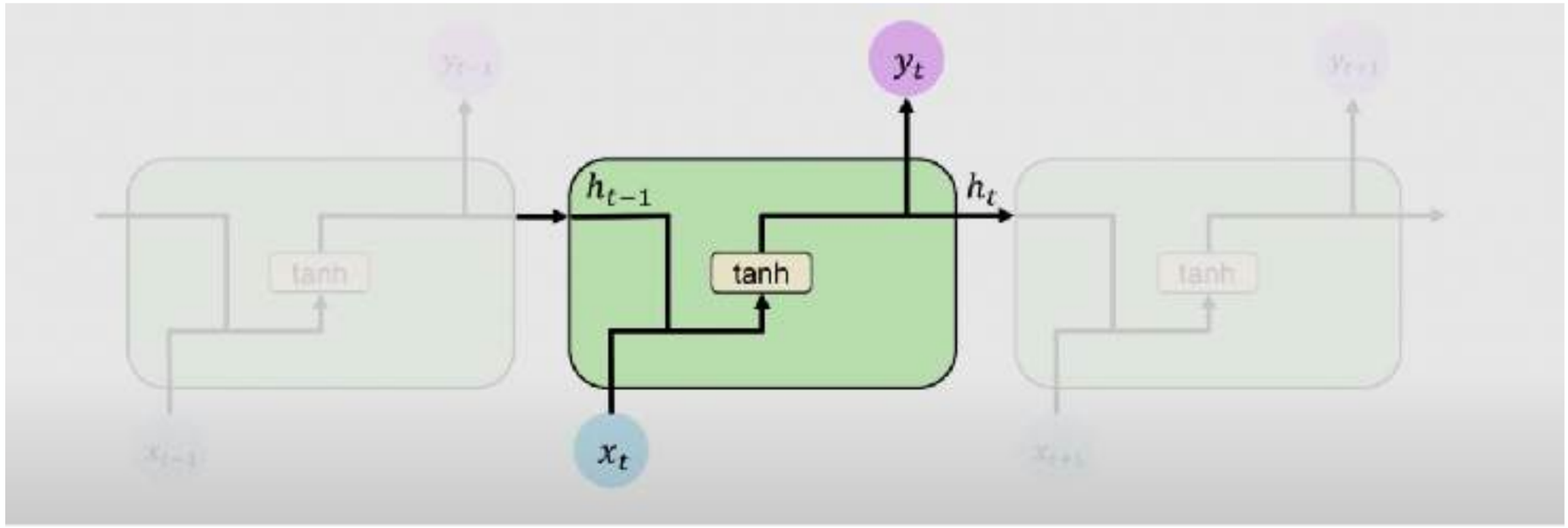


Long Short Term Memory (LSTMs) networks are an example of this

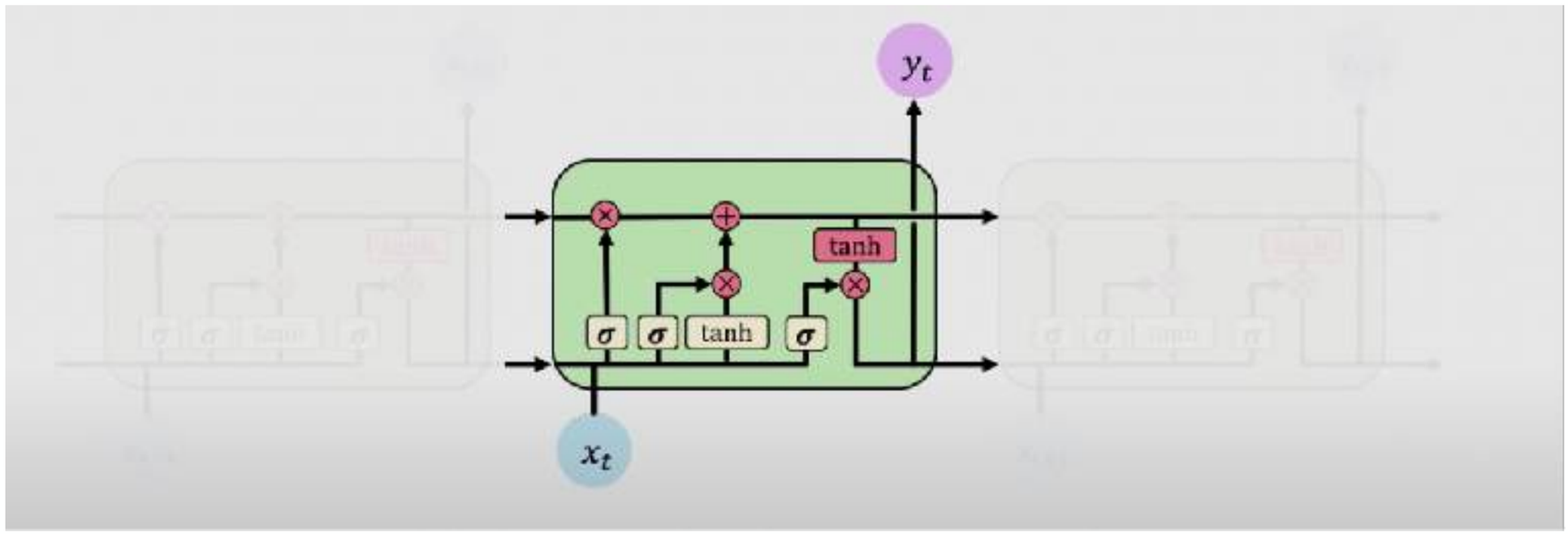
Standard RNN



Standard RNN



LSTM unit



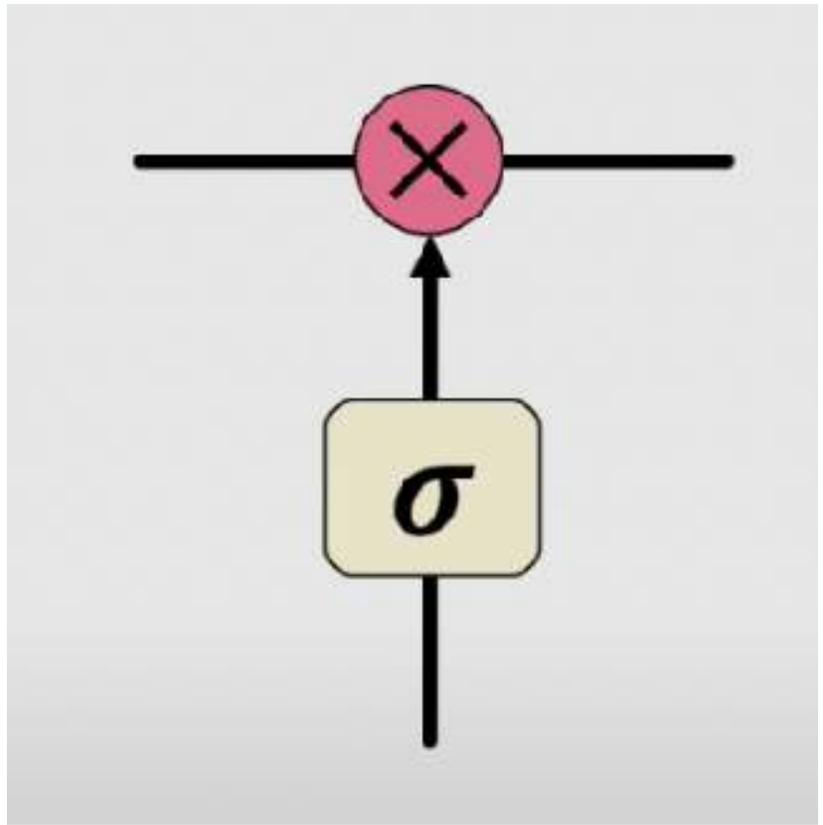
Each memory cell is equipped with an *internal state* and a number of multiplicative gates that determine whether

(i) a given input should impact the internal state (the *input gate*),

(ii) the internal state should be flushed to (the *forget gate*), and

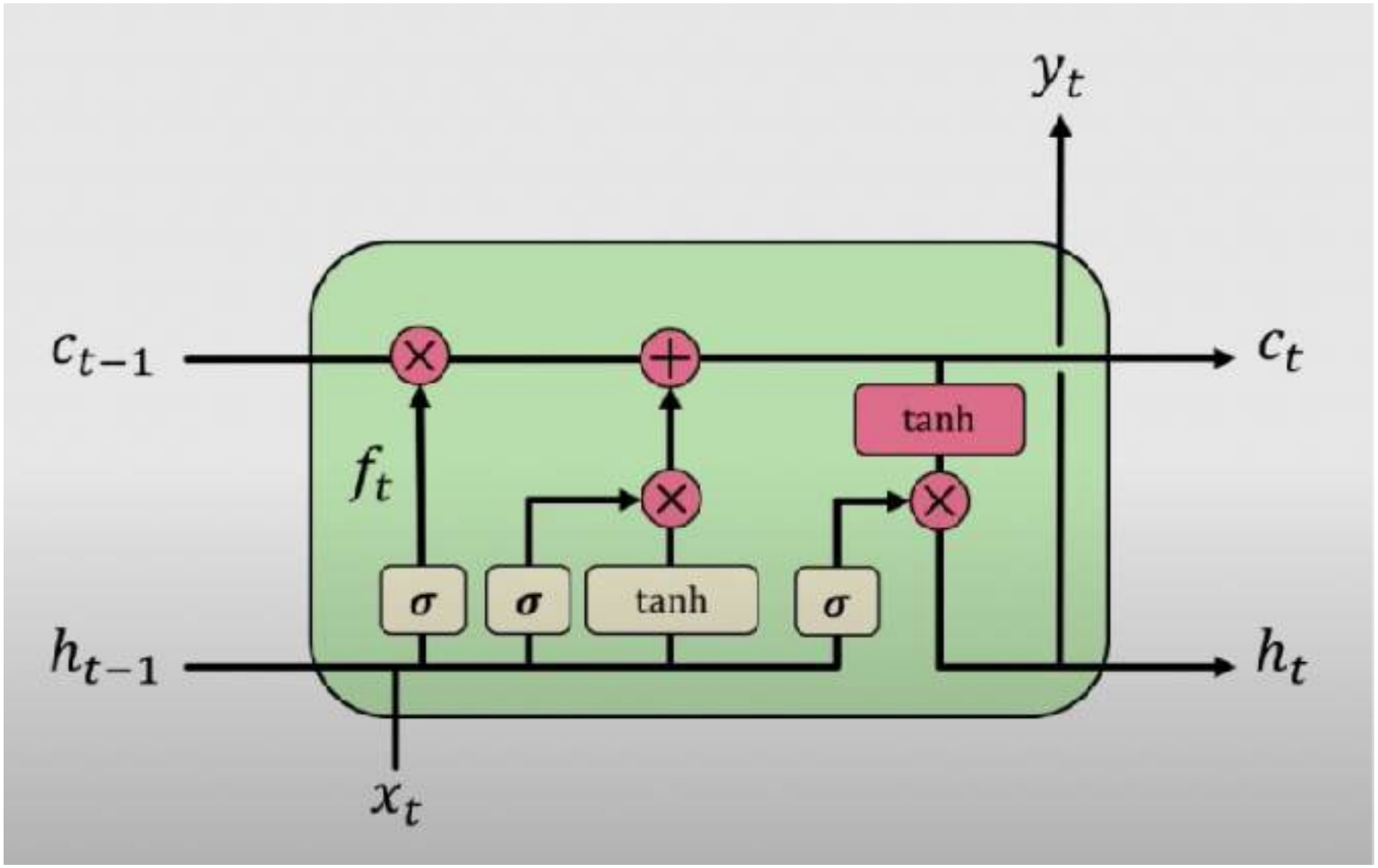
(iii) the internal state of a given neuron should be allowed to impact the cell's output (the *output* gate).

The key building block are the so-called gates



INFORMATION IS PASSED OR NOT WITH A COMBINATION OF A SIGMOID NN AND A POINTWISE MULTIPLICATION

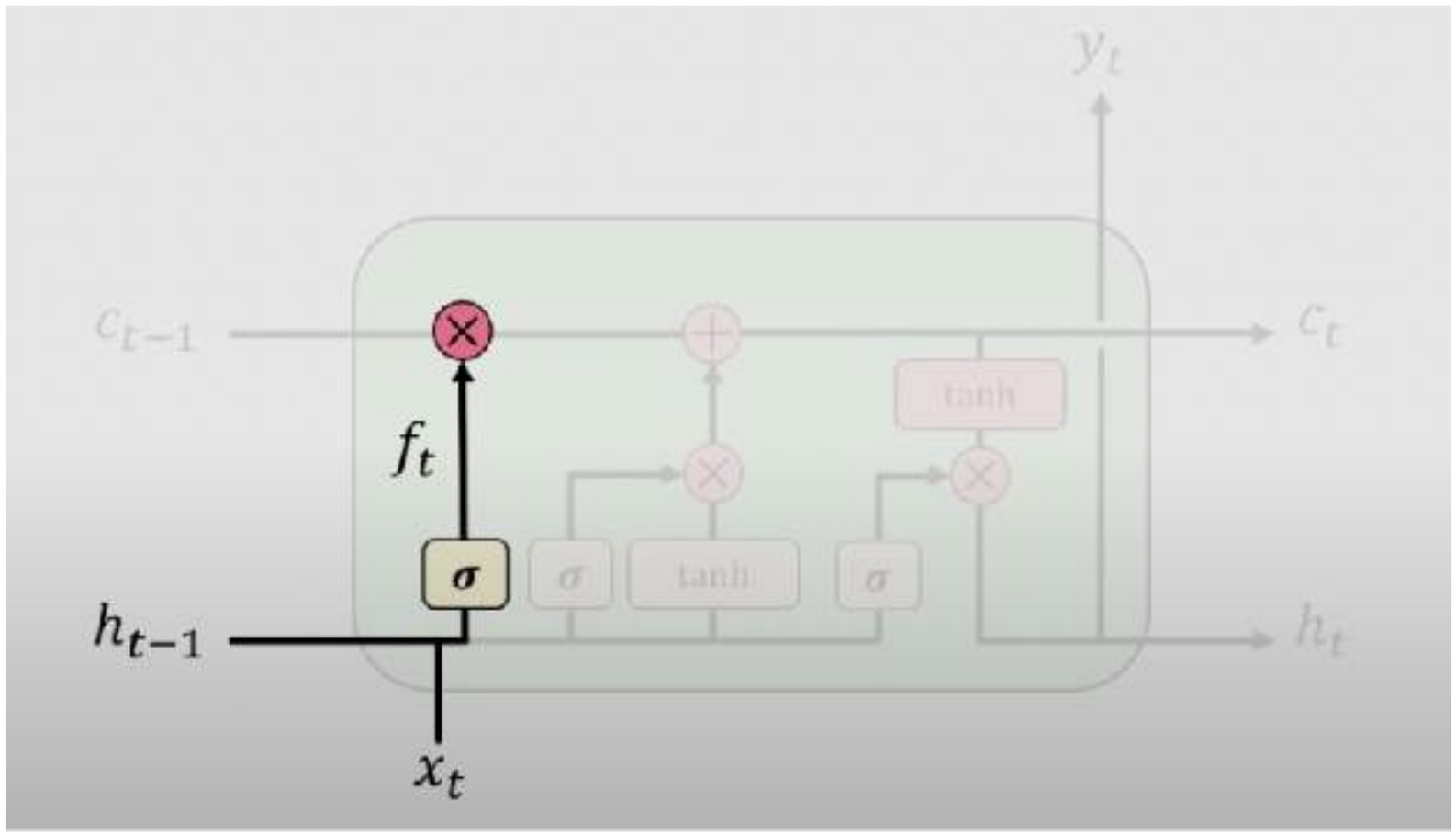
Long Short Term Memory (LSTMs)



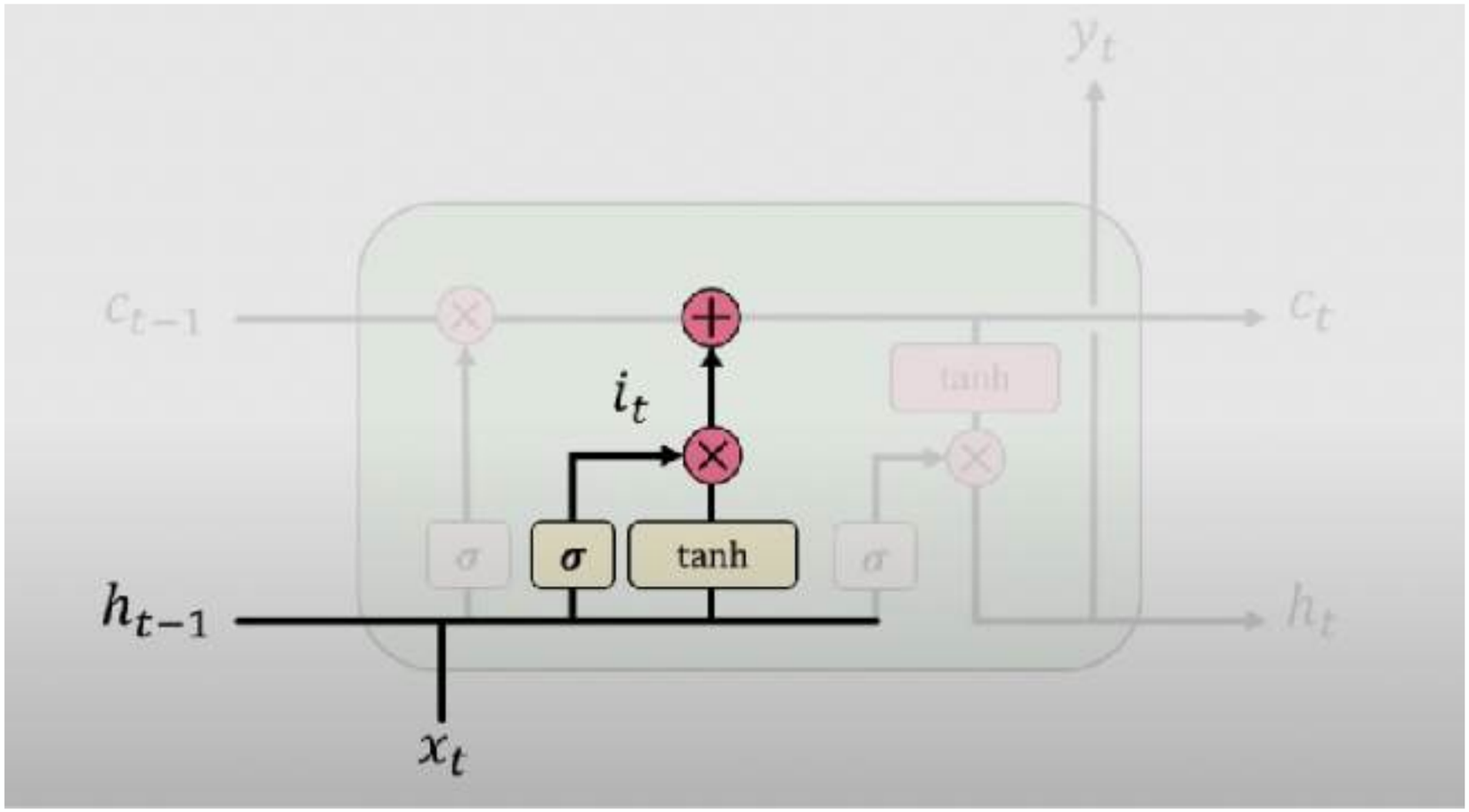
4 main steps:

1-Forget 2-Store 3-Update 4-Output

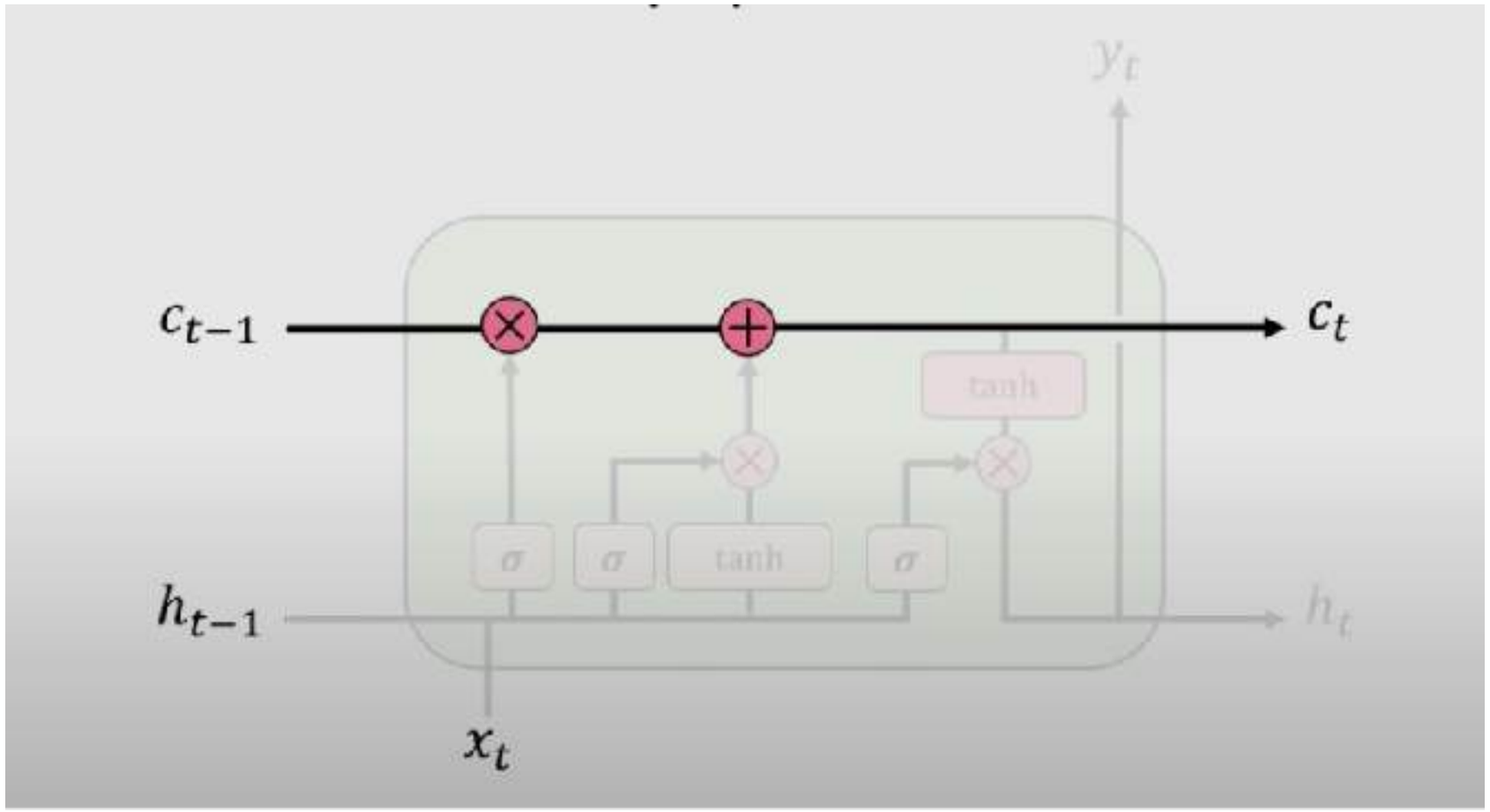
STEP 1: FORGET



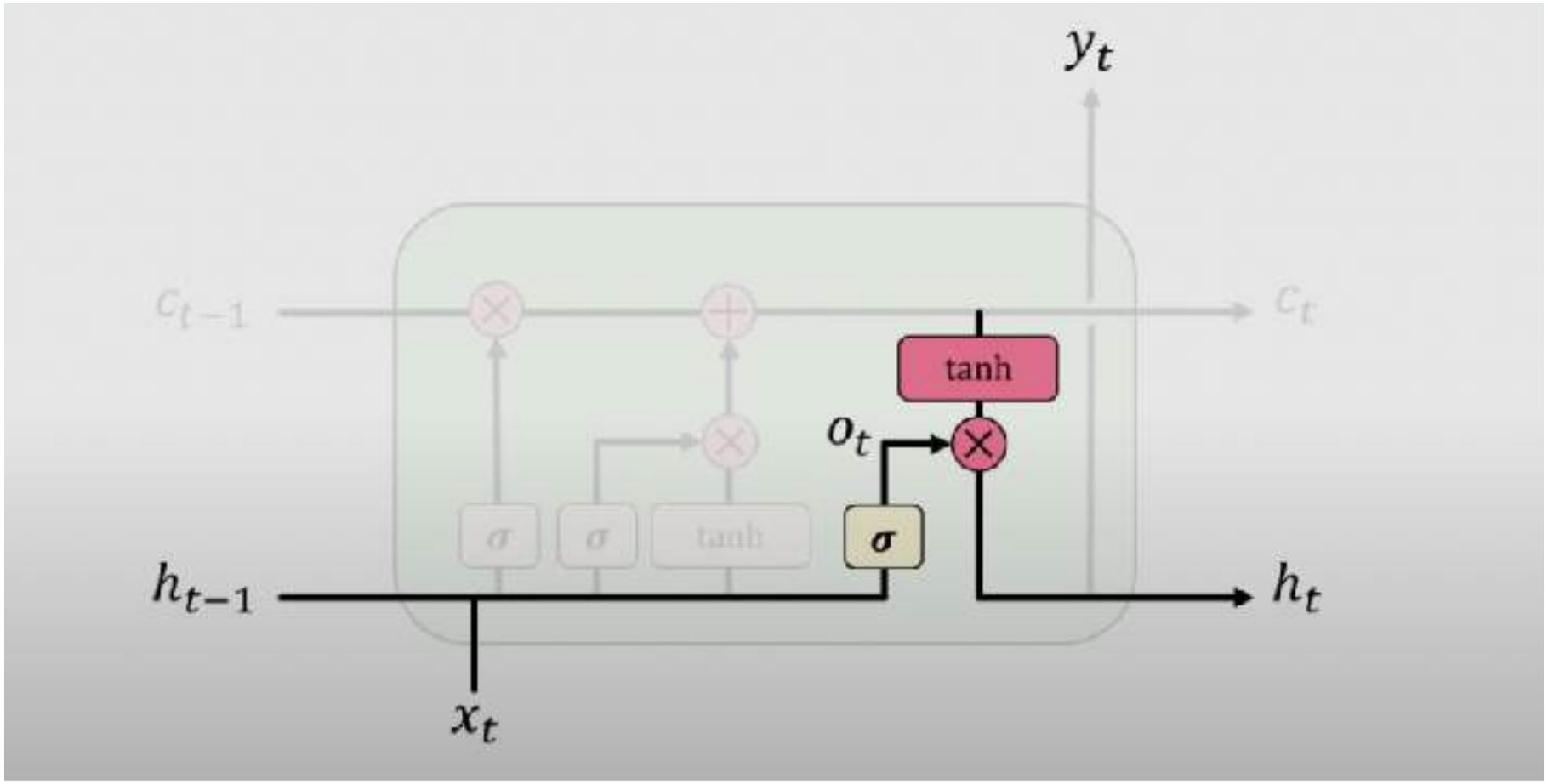
STEP 2: STORE



STEP 3: UPDATE



STEP 4: OUTPUT



LSTMs KEY CONCEPTS

1. Maintain a separate cell state from what is outputted

2. Use gates to control the flow of information

- Forget gates get rid of irrelevant information
- Store relevant information from current input
- Selectively update cell state
- Output gate returns a filtered version of the cell state

3. Backpropagation through time with uninterrupted gradient flow

Intuition Behind Self-Attention

Attending to the most important parts of an input.

Intuition Behind Self-Attention

Attending to the most important parts of an input.

1. Identify which parts to attend to
2. Extract the features with high attention

Intuition Behind Self-Attention

Attending to the most important parts of an input.

1. Identify which parts to attend to

Similar to a search problem!

2. Extract the features with high attention

Understanding Attention with Search

Query (Q)

Key (K_1)

Key (K_2)

Key (K_3)

1. **Compute attention mask:** how similar is each key to the desired query?

Understanding Attention with Search

Query (Q)

Key (K_1)

How similar is the key to the query?

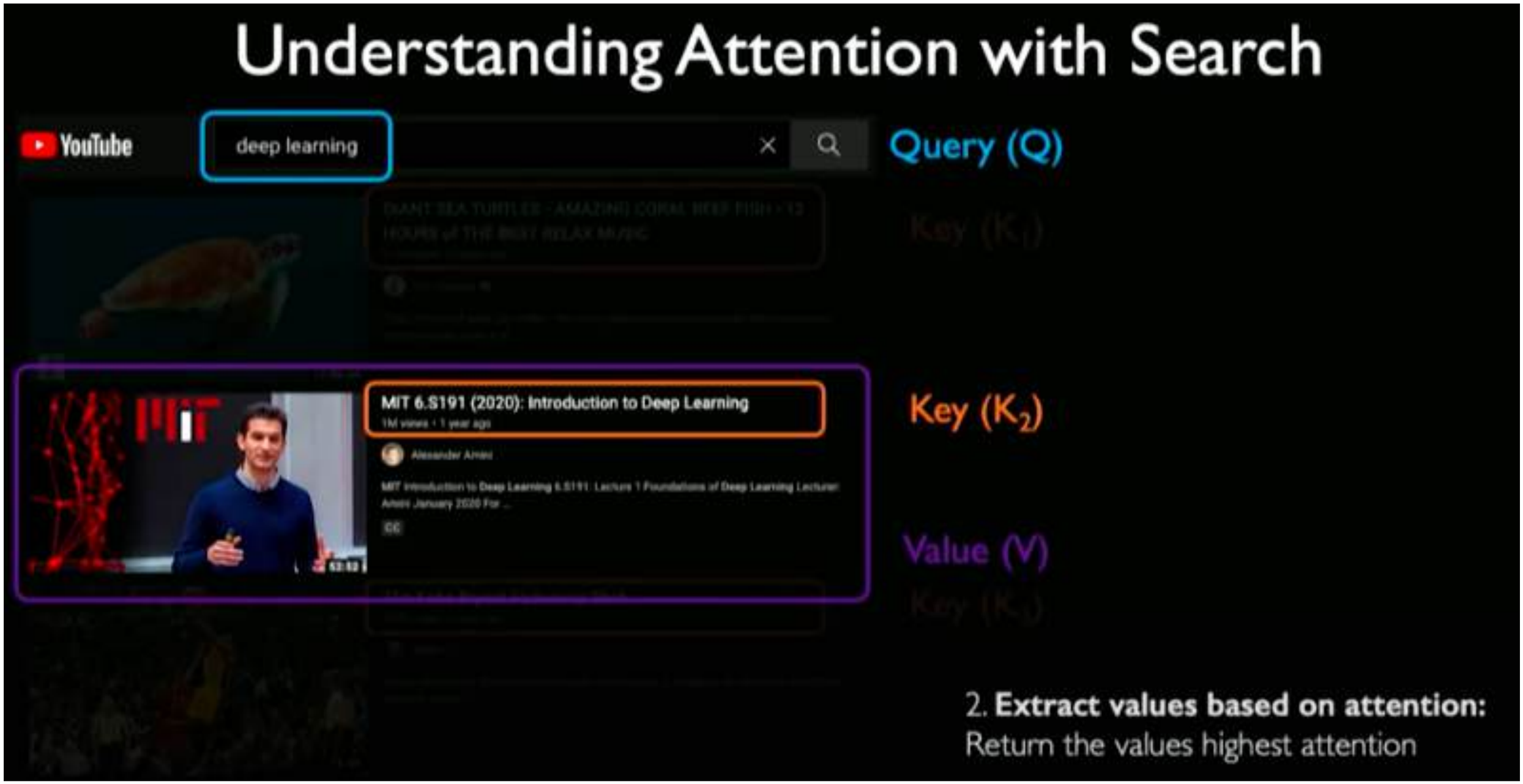
Key (K_2)

Key (K_3)

1. **Compute attention mask:** how similar is each key to the desired query?

Understanding Attention with Search

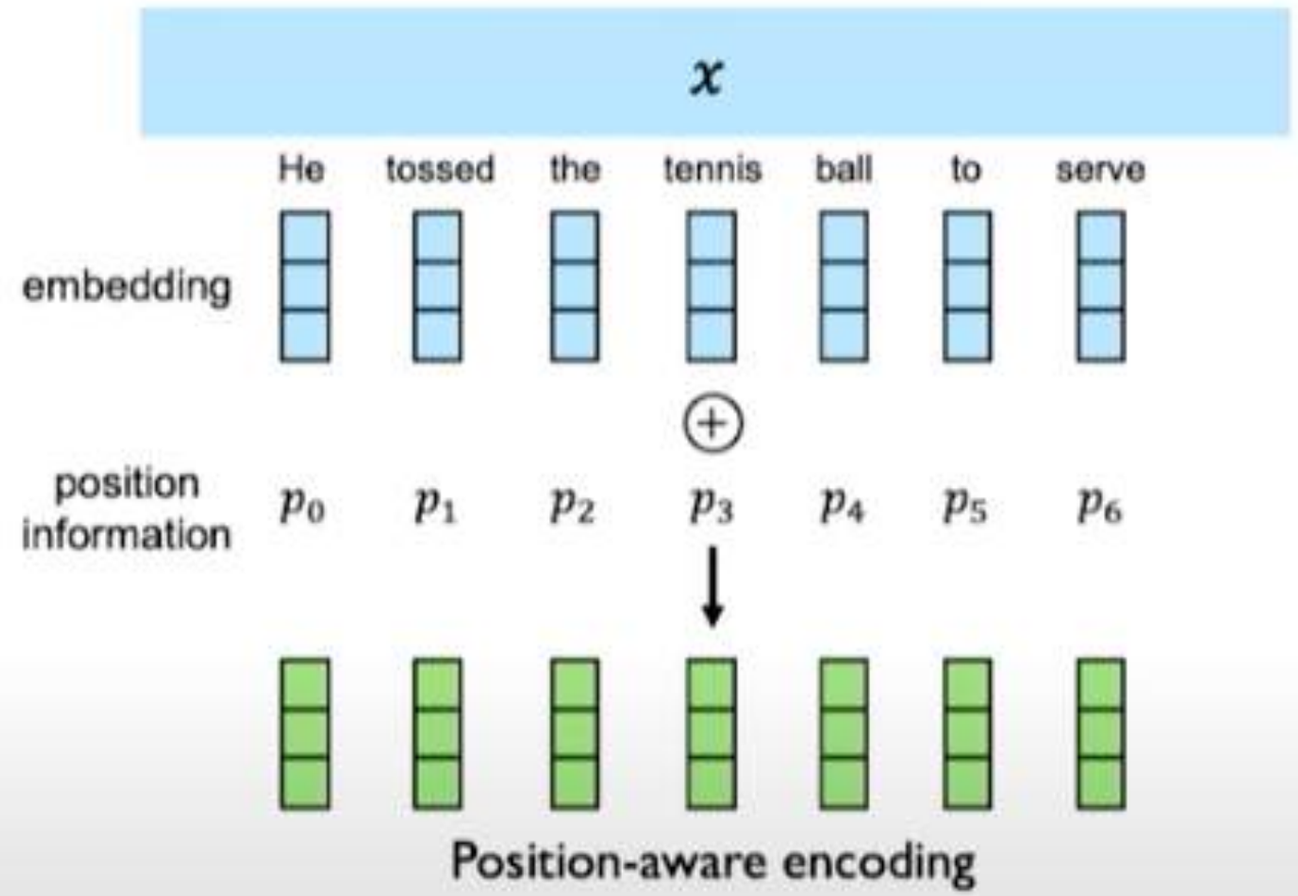
Understanding Attention with Search



Learning Self-Attention with Neural Networks

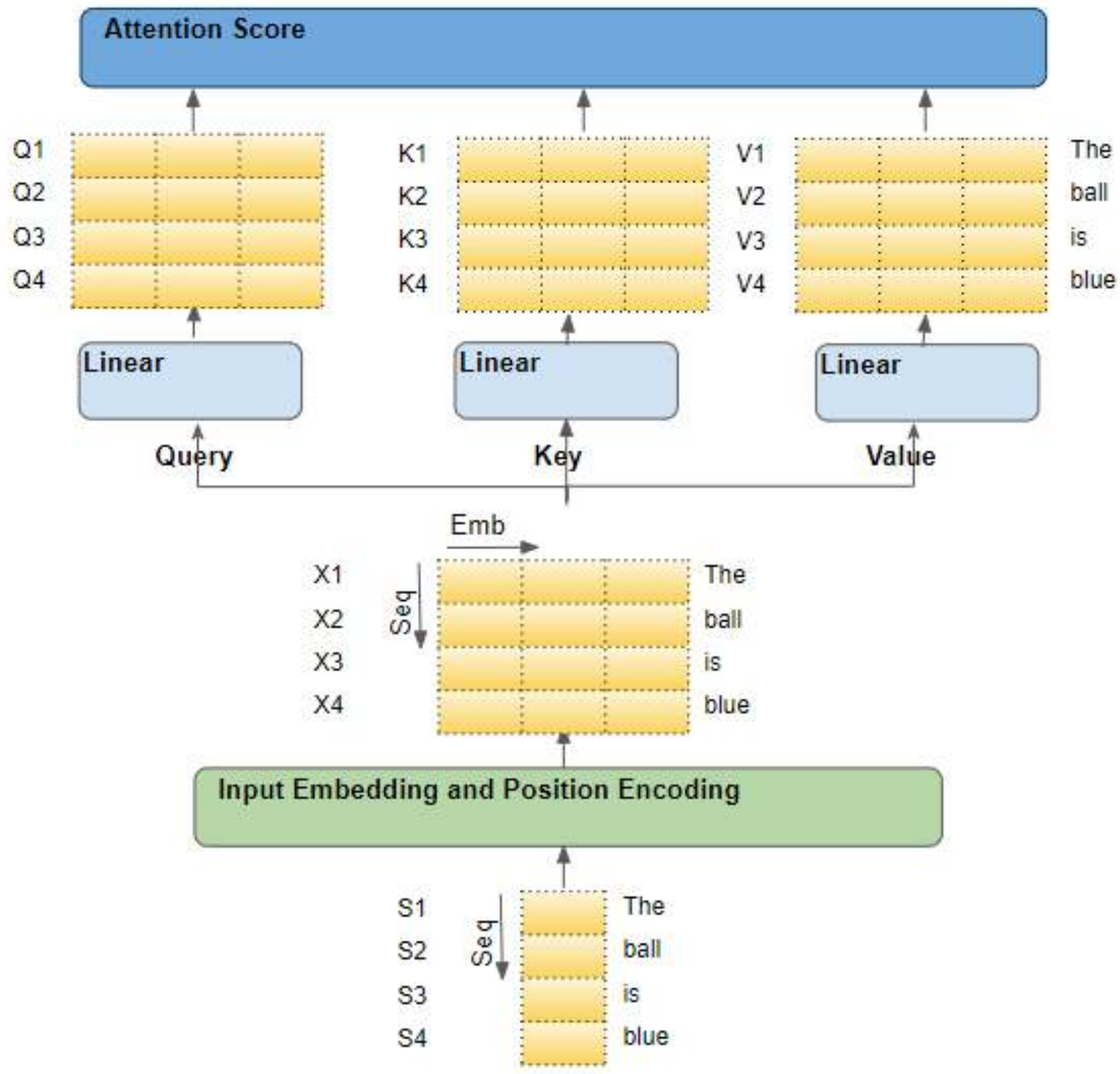
Goal: identify and attend to most important features in input.

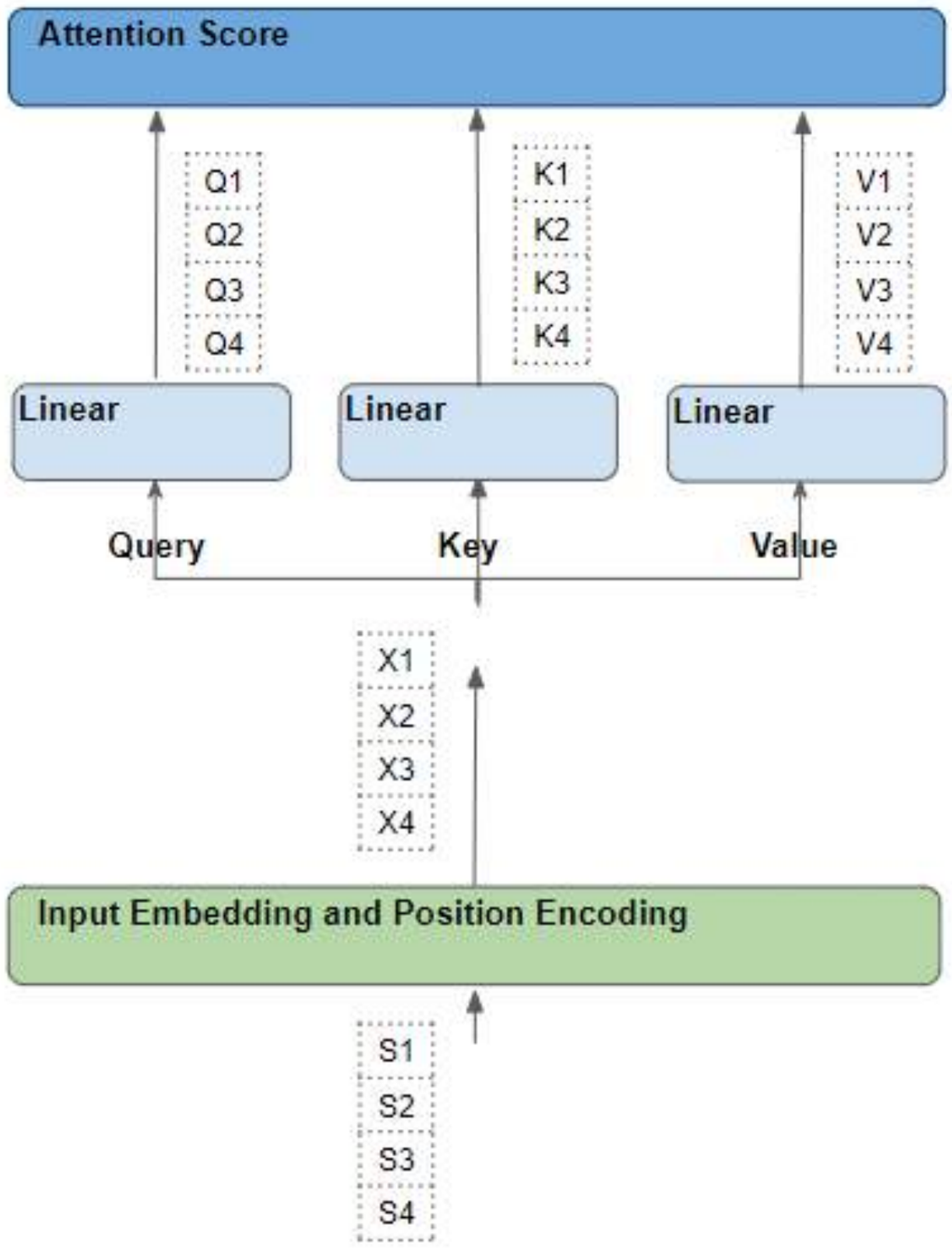
1. Encode **position** information

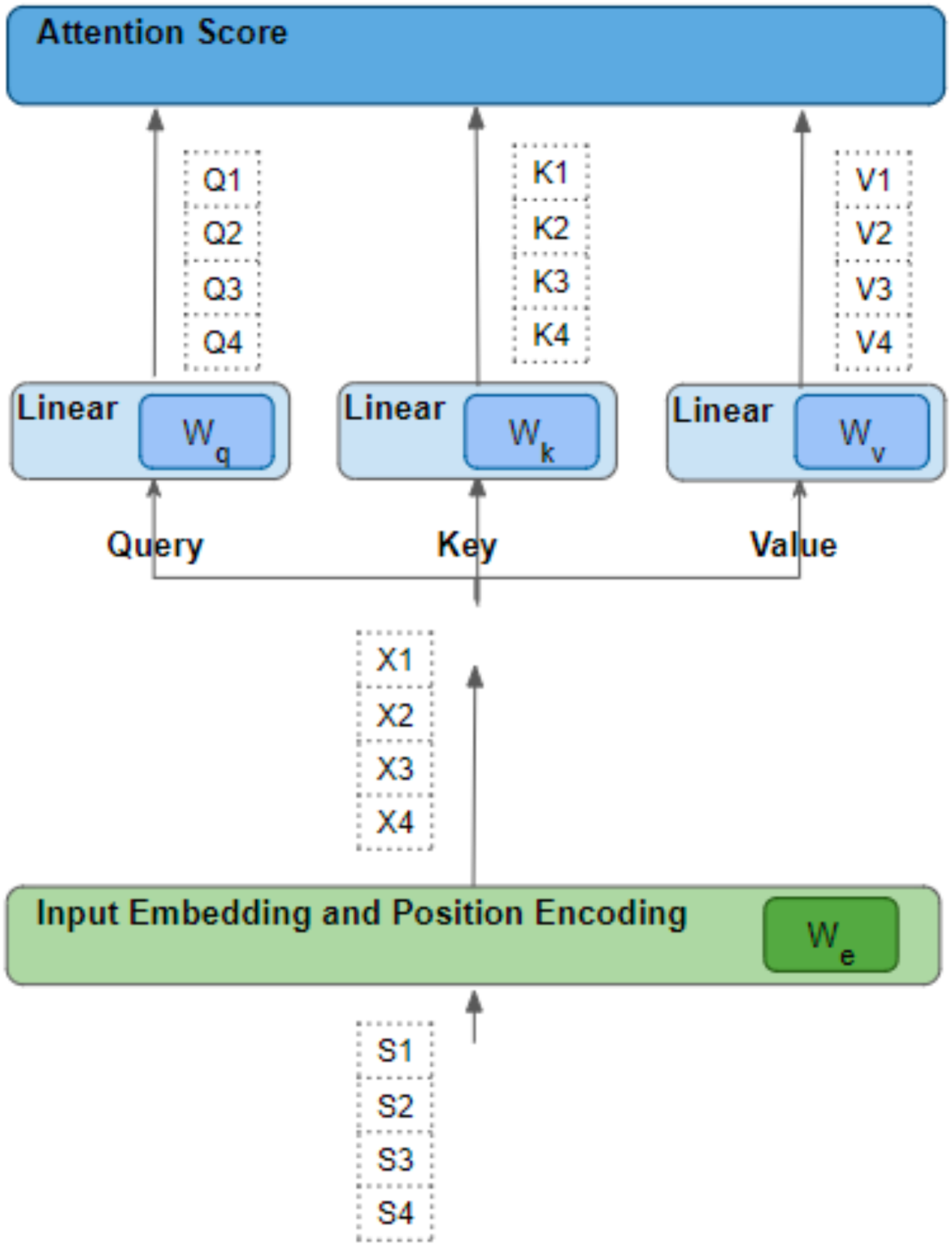


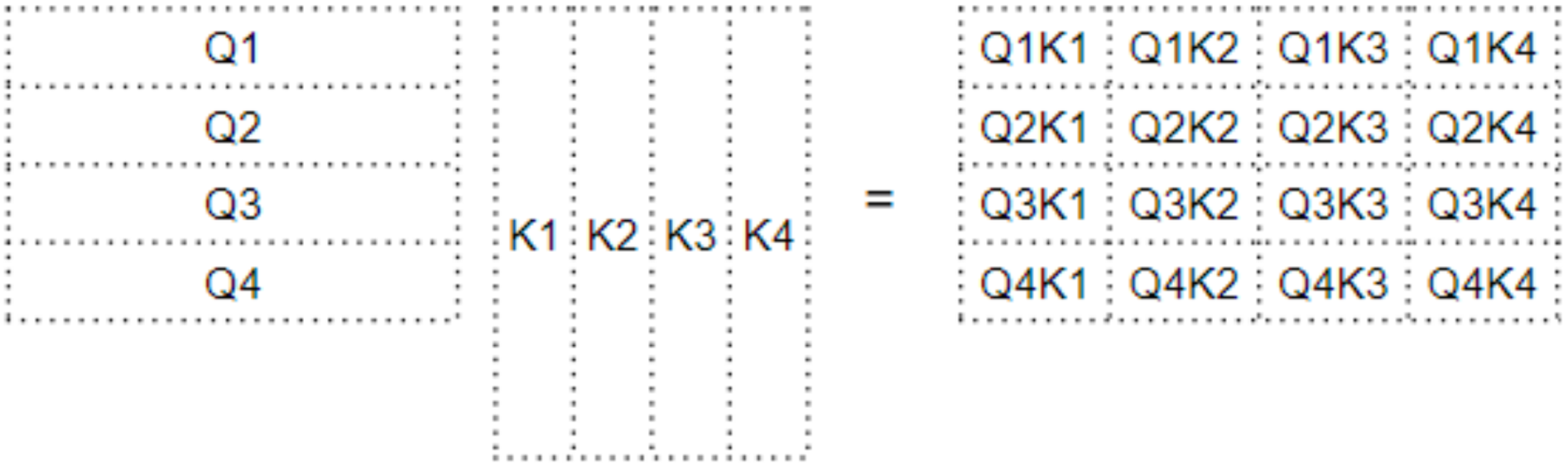
Data is fed in all at once! Need to encode position information to understand order.

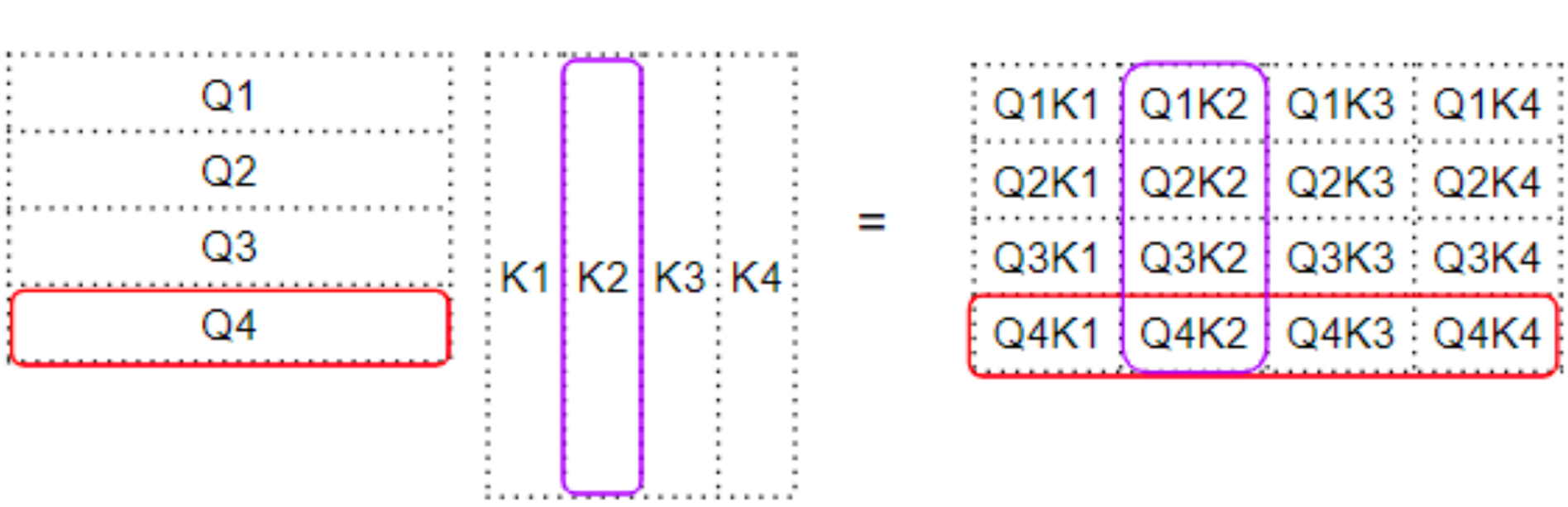
The Attention Module

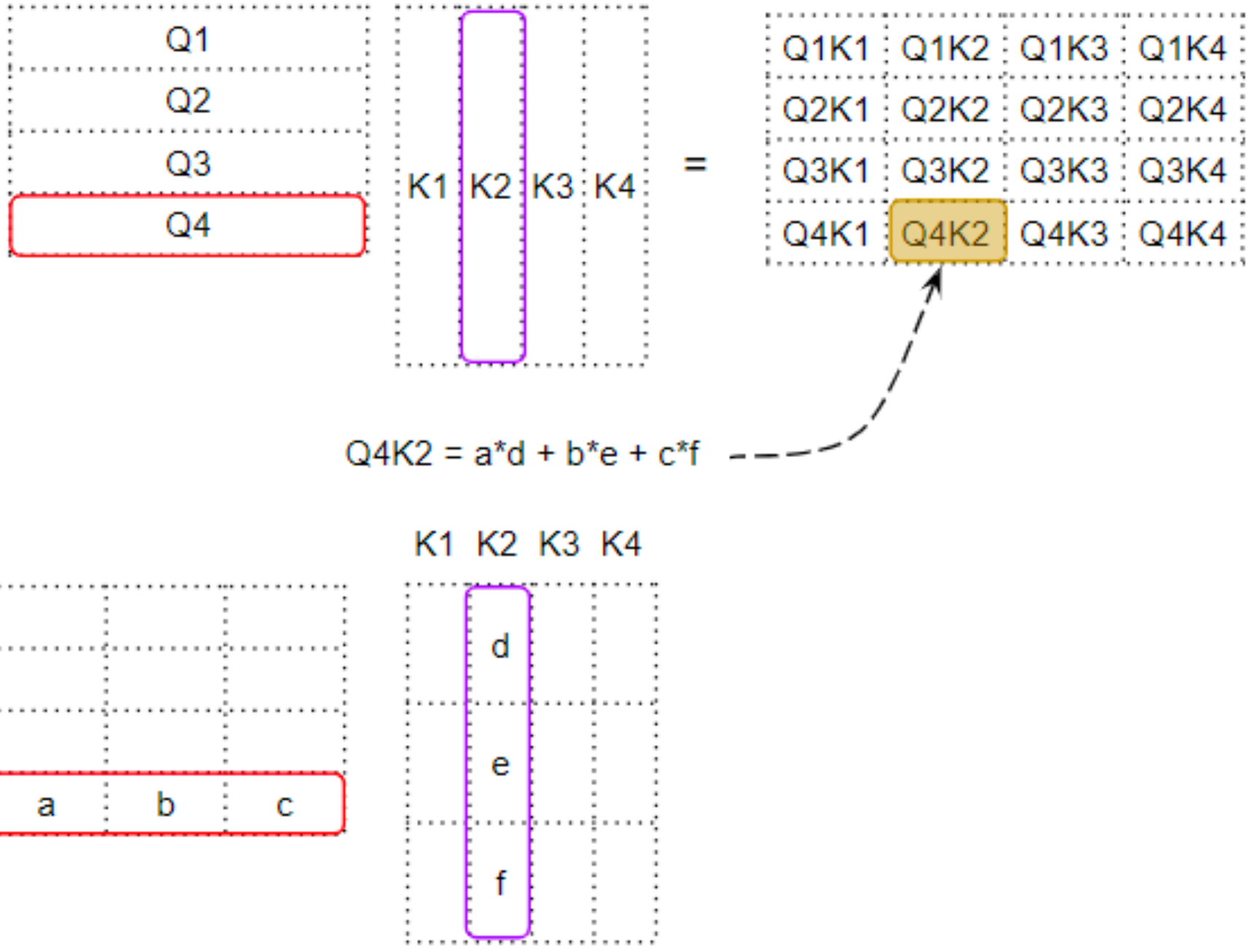



Attention Score
Q1
Q2
Q3
Q4
K1
K2
K3
K4
V1
V2
V3
V4
Linear
Linear
Linear
Query
Key
Value
X1
X2
X3
X4
Input Embedding and Position Encoding
S1
S2
S3
S4


Attention Score
Q1
Q2
Q3
Q4
K1
K2
K3
K4
V1
V2
V3
V4
Linear
W_q
Linear
W_k
Linear
W_v
Query
Key
Value
X1
X2
X3
X4
Input Embedding and Position Encoding
W_e
S1
S2
S3
S4


Q1
Q2
Q3
Q4
K1 K2 K3 K4
=
Q1K1 Q1K2 Q1K3 Q1K4
Q2K1 Q2K2 Q2K3 Q2K4
Q3K1 Q3K2 Q3K3 Q3K4
Q4K1 Q4K2 Q4K3 Q4K4


Q1
Q2
Q3
Q4
K1 K2 K3 K4
=
Q1K1 Q1K2 Q1K3 Q1K4
Q2K1 Q2K2 Q2K3 Q2K4
Q3K1 Q3K2 Q3K3 Q3K4
Q4K1 Q4K2 Q4K3 Q4K4


Q1
Q2
Q3
Q4
K1
K2
K3
K4
=
Q1K1
Q1K2
Q1K3
Q1K4
Q2K1
Q2K2
Q2K3
Q2K4
Q3K1
Q3K2
Q3K3
Q3K4
Q4K1
Q4K2
Q4K3
Q4K4
Q4K2 = a*d + b*e + c*f
Q1
Q2
Q3
Q4
a
b
c
K1
K2
K3
K4
d
e
f

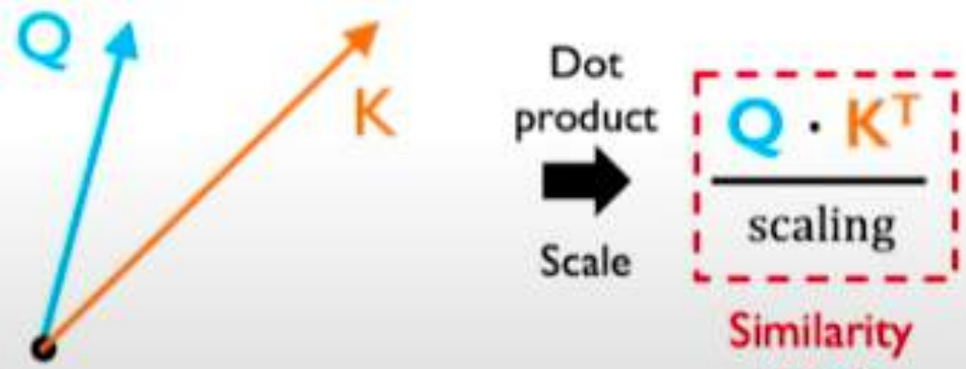
Learning Self-Attention with Neural Networks

Goal: identify and attend to most important features in input.

1. Encode **position** information
2. Extract **query, key, value** for search
3. Compute **attention weighting**

Attention score: compute pairwise similarity between each query and key

How to compute similarity between two sets of features?



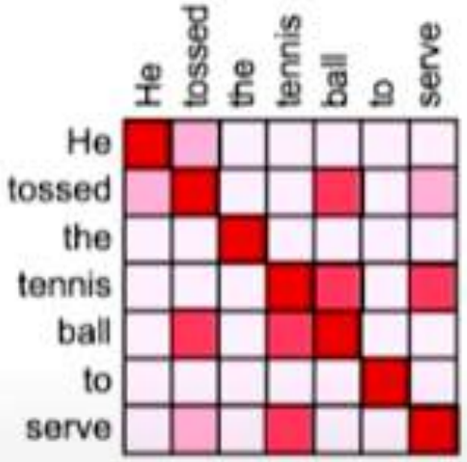
Also known as the "cosine similarity"

Learning Self-Attention with Neural Networks

Goal: identify and attend to most important features in input.

1. Encode **position** information
2. Extract **query, key, value** for search
3. Compute **attention weighting**

Attention weighting: where to attend to! How similar is the key to the query?



$$softmax\left(\frac{Q \cdot K^T}{scaling}\right)$$

Q1K1	Q1K2	Q1K3	Q1K4
Q2K1	Q2K2	Q2K3	Q2K4
Q3K1	Q3K2	Q3K3	Q3K4
Q4K1	Q4K2	Q4K3	Q4K4

V1
V2
V3
V4

=

Q1K1V1 + Q1K2V2 + Q1K3V3 + Q1K4V4
Q2K1V1 + Q2K2V2 + Q2K3V3 + Q2K4V4
Q3K1V1 + Q3K2V2 + Q3K3V3 + Q3K4V4
Q4K1V1 + Q4K2V2 + Q4K3V3 + Q4K4V4

=

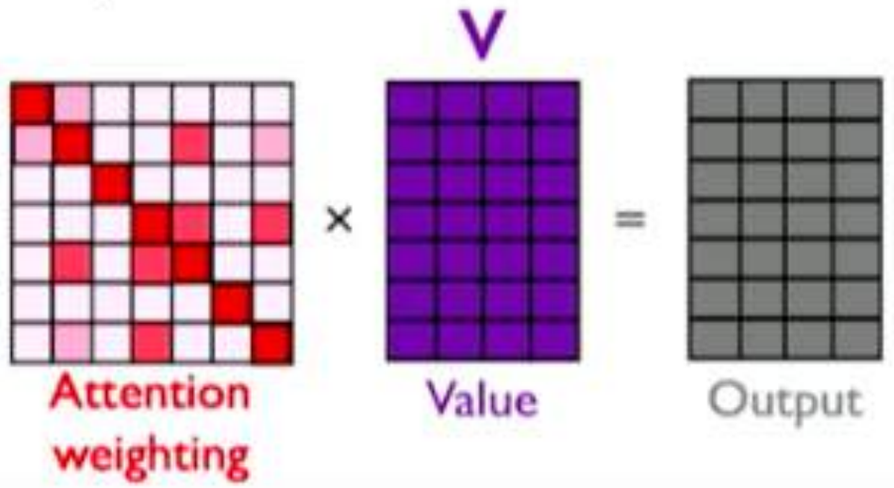
Z1
Z2
Z3
Z4

Learning Self-Attention with Neural Networks

Goal: identify and attend to most important features in input.

1. Encode **position** information
2. Extract **query, key, value** for search
3. Compute **attention weighting**
4. Extract **features with high attention**

Last step: self-attend to extract features



$$softmax\left(\frac{Q \cdot K^T}{scaling}\right) \cdot V = A(Q,K,V)$$

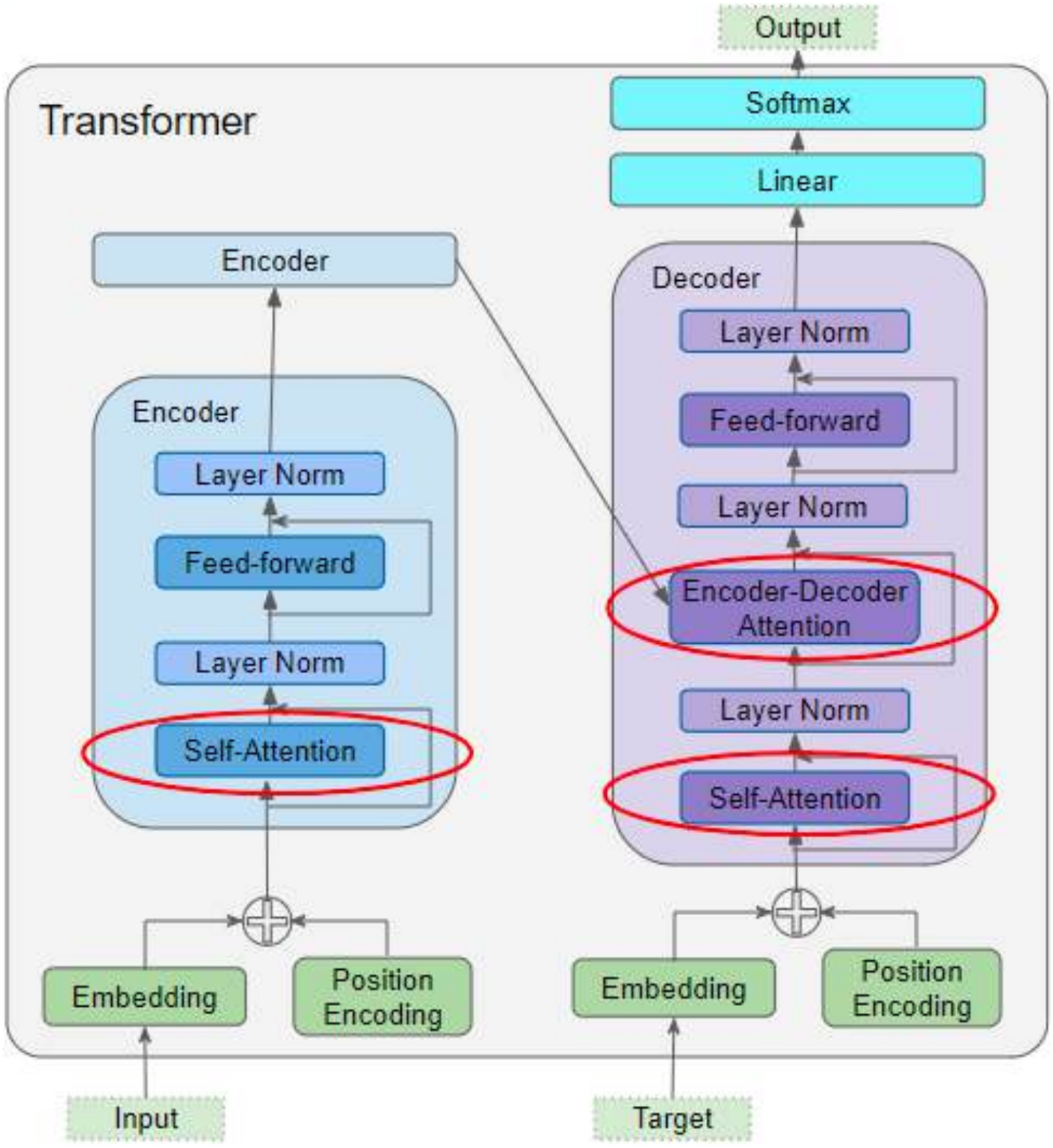
Fourth word Score

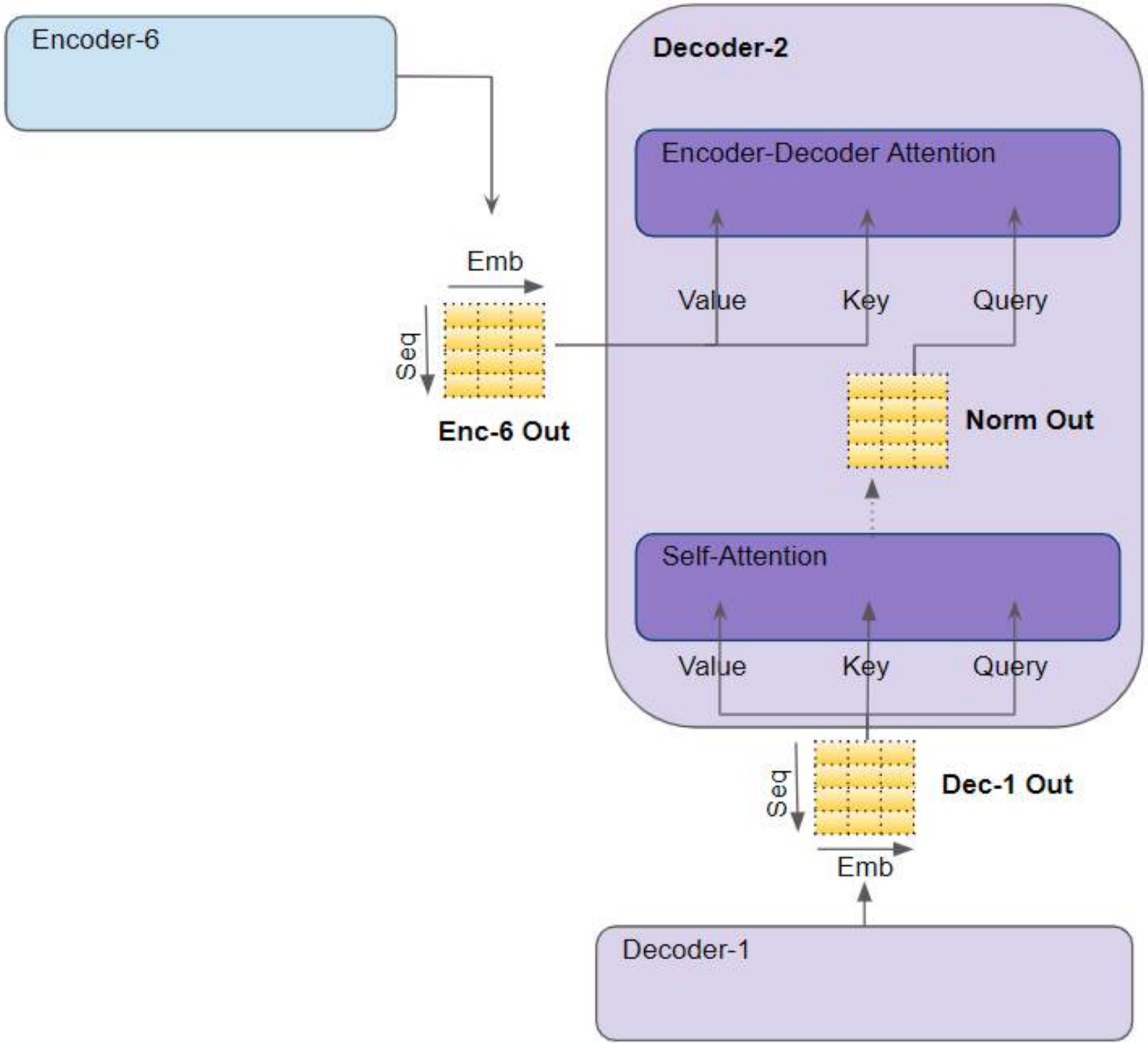
Fourth Query word * first Key word

$$Z_4 = (Q_4 K_1)\, V_1 + (Q_4 K_2)\, V_2 + (Q_4 K_3)\, V_3 + (Q_4 K_4)\, V_4$$

Fourth Query word * second Key word

$$Z = Softmax(\frac{QK^T}{\sqrt{d_k}})V$$

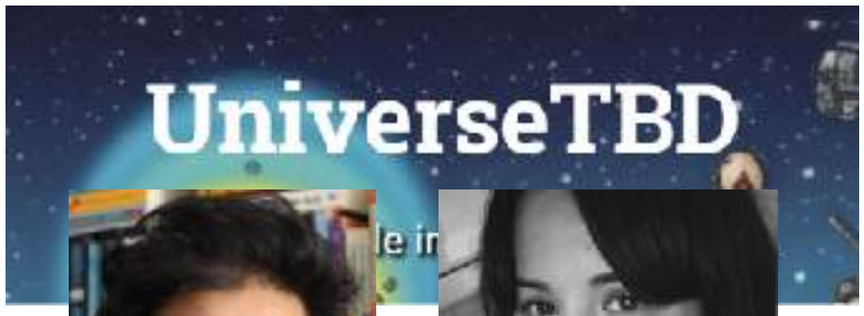

Output
Transformer
Softmax
Linear
Encoder
Decoder
Encoder
Layer Norm
Feed-forward
Layer Norm
Encoder-Decoder Attention
Layer Norm
Self-Attention
Layer Norm
Feed-forward
Layer Norm
Self-Attention
Embedding
Position Encoding
Embedding
Position Encoding
Input
Target


Encoder-6
Decoder-2
Encoder-Decoder Attention
Emb
Seq
Value
Key
Query
Enc-6 Out
Norm Out
Self-Attention
Value
Key
Query
Seq
Dec-1 Out
Emb
Decoder-1

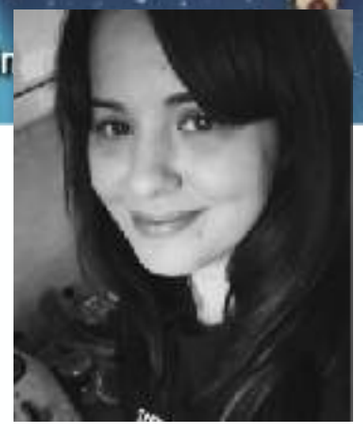
Pathfinder

LLM enabled literature search

huggingface.co



K. Iyer



I. Ciuca

Ask me anything:

how are stellar bars formed

To understand how stellar bars are formed in galaxies, we can break down the process into several key steps:

1. **Galactic Structure**: Stellar bars are non-axisymmetric structures that typically form in the disks of spiral galaxies. They are characterized by a linear arrangement of stars that extends across the center of the galaxy.

2. **Conditions for Formation**: The formation of stellar bars is often linked to the dynamics of the stellar disk. A crucial factor is the stability of the disk. If the disk is sufficiently cold (i.e., has low velocity dispersion), it can become dynamically unstable, leading to the formation of a bar. This is referred to as "bar instability."

3. **Role of Dark Matter Halos**: The presence of a mildly triaxial cold dark matter (CDM) halo can influence the formation of bars. The gravitational potential created by the dark matter can help stabilize the disk and facilitate the conditions necessary for bar formation.

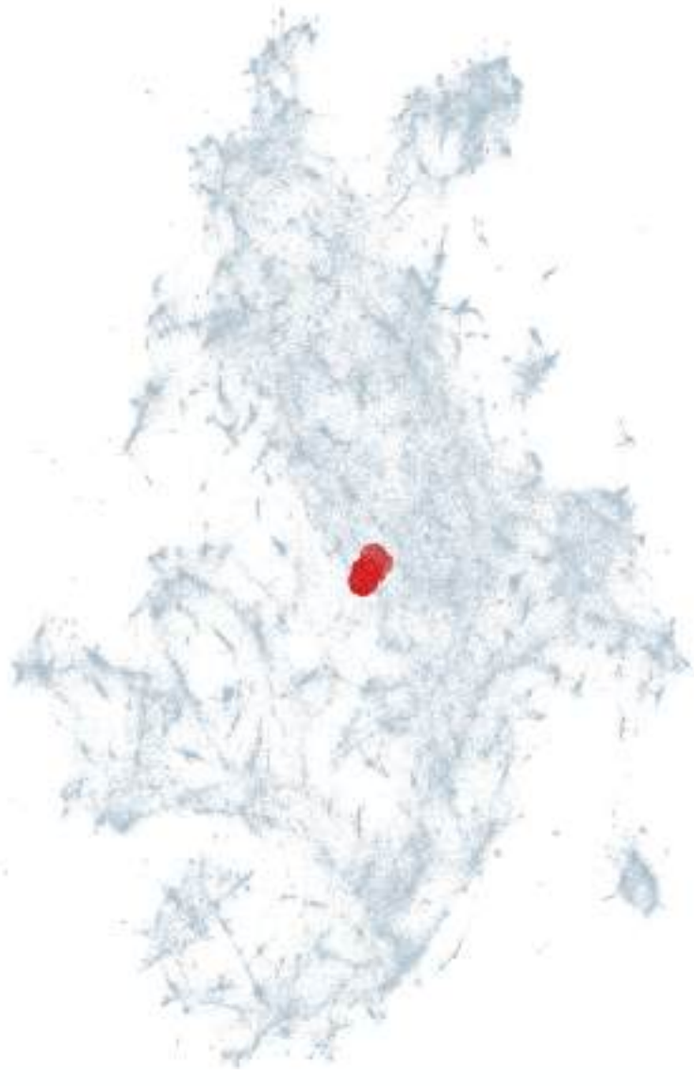
4. **Evolution of Bars**: Once formed, stellar bars can evolve over time. They tend to increase in both linear and vertical extents, which can affect the overall dynamics of the galaxy. The properties of the bars can show recurrent behavior, meaning they may undergo cycles of growth and decay.

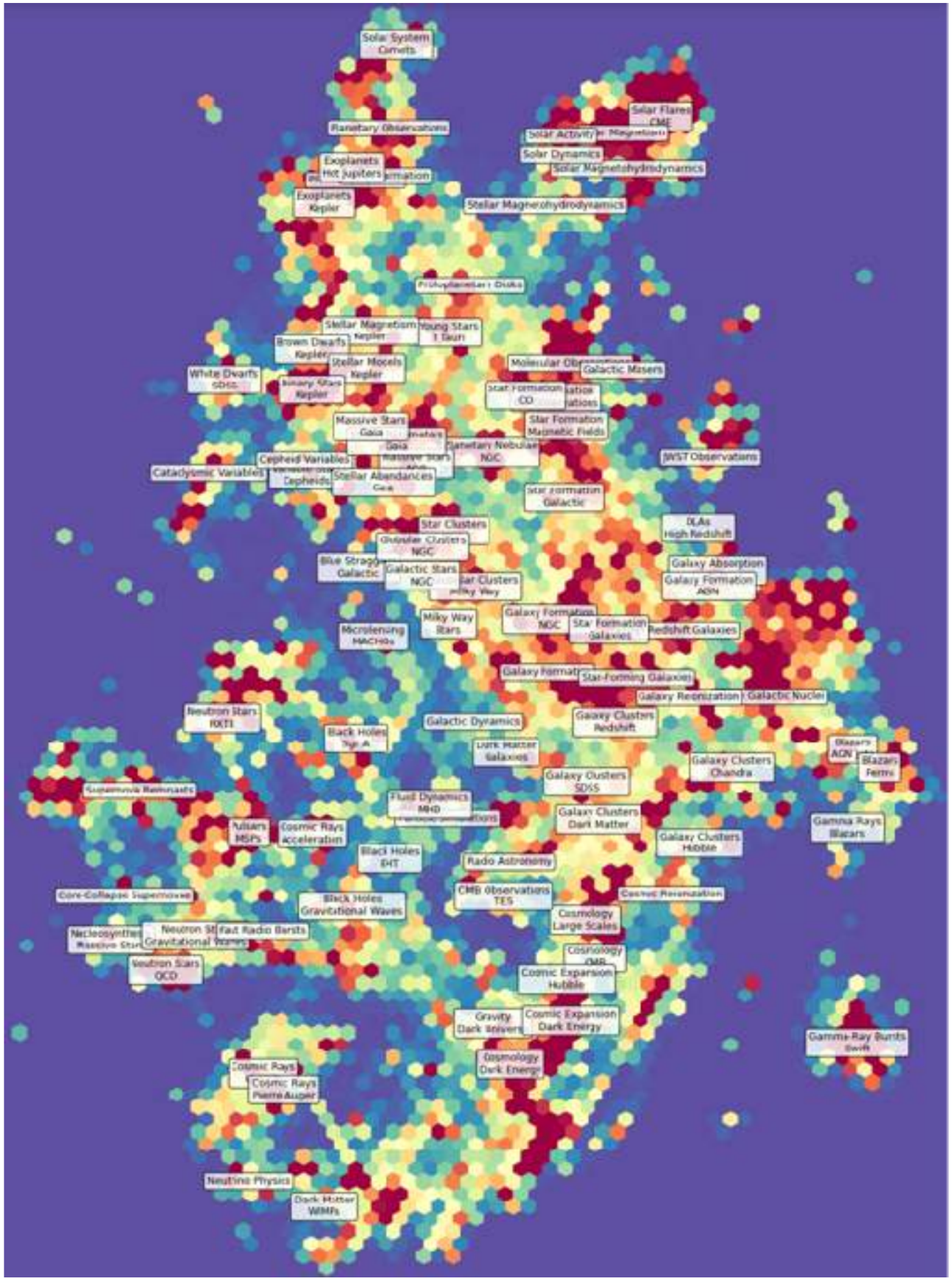
5. **Mass Dependency**: The likelihood of bar formation can depend on the mass of the galaxy. Lower mass galaxies may not form bars as readily because they can be dynamically hotter, meaning their stars have higher velocity dispersions, which can inhibit the conditions necessary for bar instability.

In summary, stellar bars form through a combination of dynamic instability in a cold stellar disk, influenced by the gravitational effects of dark matter halos, and are subject to evolutionary changes over time. The mass of the galaxy plays a significant role in whether or not bars can form.

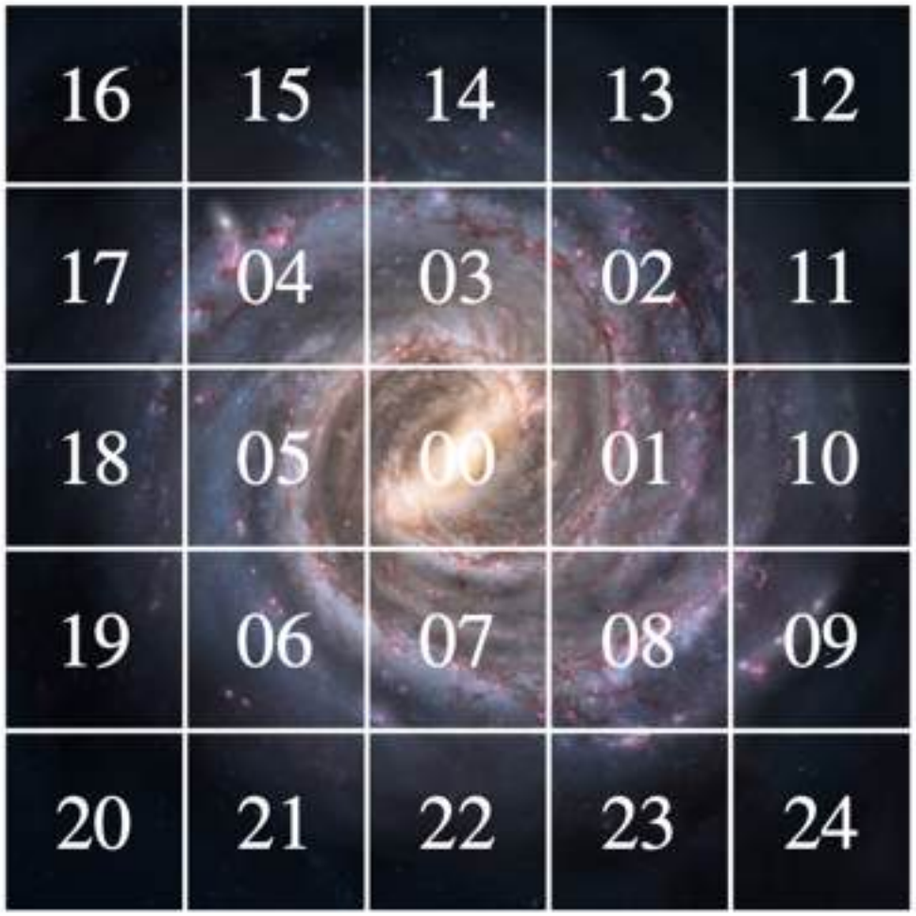
Relevant papers

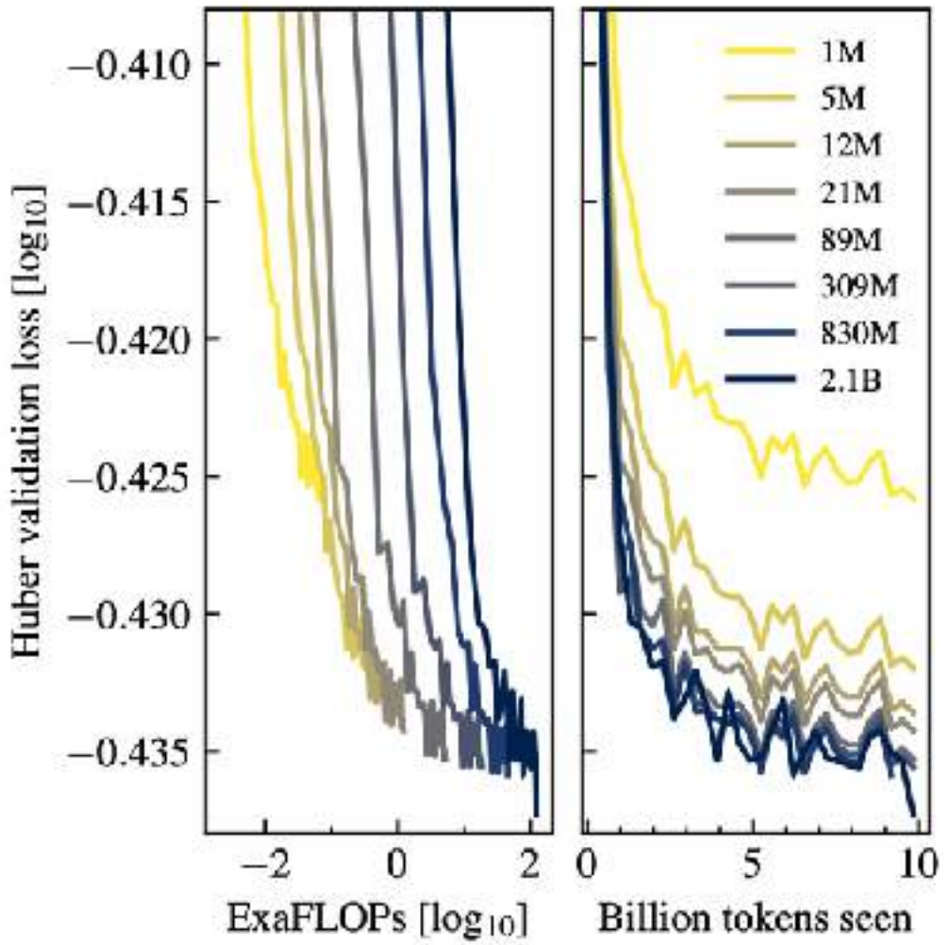
	Title	Relevance	Year	ADS Link	Citations
0	nning dark matter haloes promote bar formation	0.9893	2013	2013MNRAS.434.1287S	82
1	red Galaxies: Intrinsic or Extrinsic?	0.9887	1996	1996ApJ...469..605N	73
2	ects of Gas on Formation and Evolution of Stellar Bars and Nuclear Rings in Disk Ga	0.9833	2019	2019ApJ...872....5S	63
3	k Evolution and Bar Triggering Driven by Interactions with Dark Matter Substructu	0.9812	2008	2008ApJ...687L..13R	73
4	llar Bar Evolution in Cuspy and Flat-cored Triaxial CDM Halos	0.98	2006	2006ApJ...637..582B	65
5	lution of Stellar Bars in Live Axisymmetric Halos: Recurrent Buckling and Secular	0.972	2006	2006ApJ...637..214M	367
6	-induced evolution of dark matter cusps	0.9702	2005	2005MNRAS.363..991H	113
7	-driven evolution and quenching of spiral galaxies in cosmological simulations	0.9701			
8	: Disks and Delayed Bar Formation	0.9694			
9	formation and evolution of bars in low surface brightness galaxies with cold dark	0.9679			



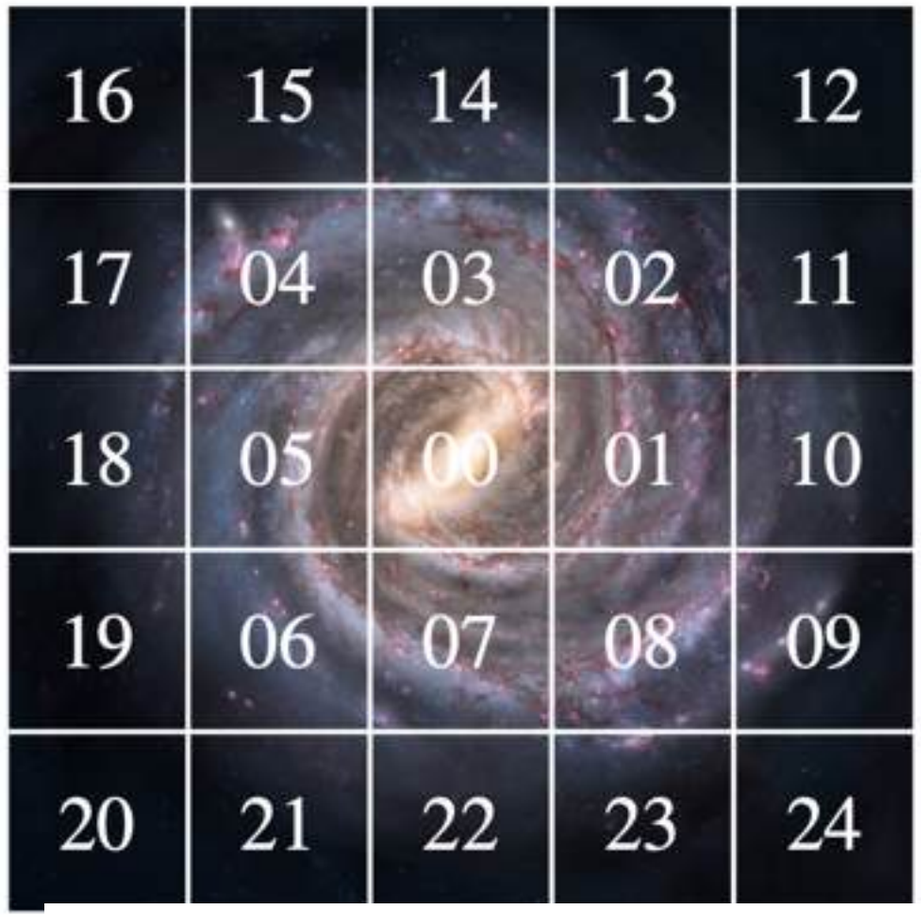


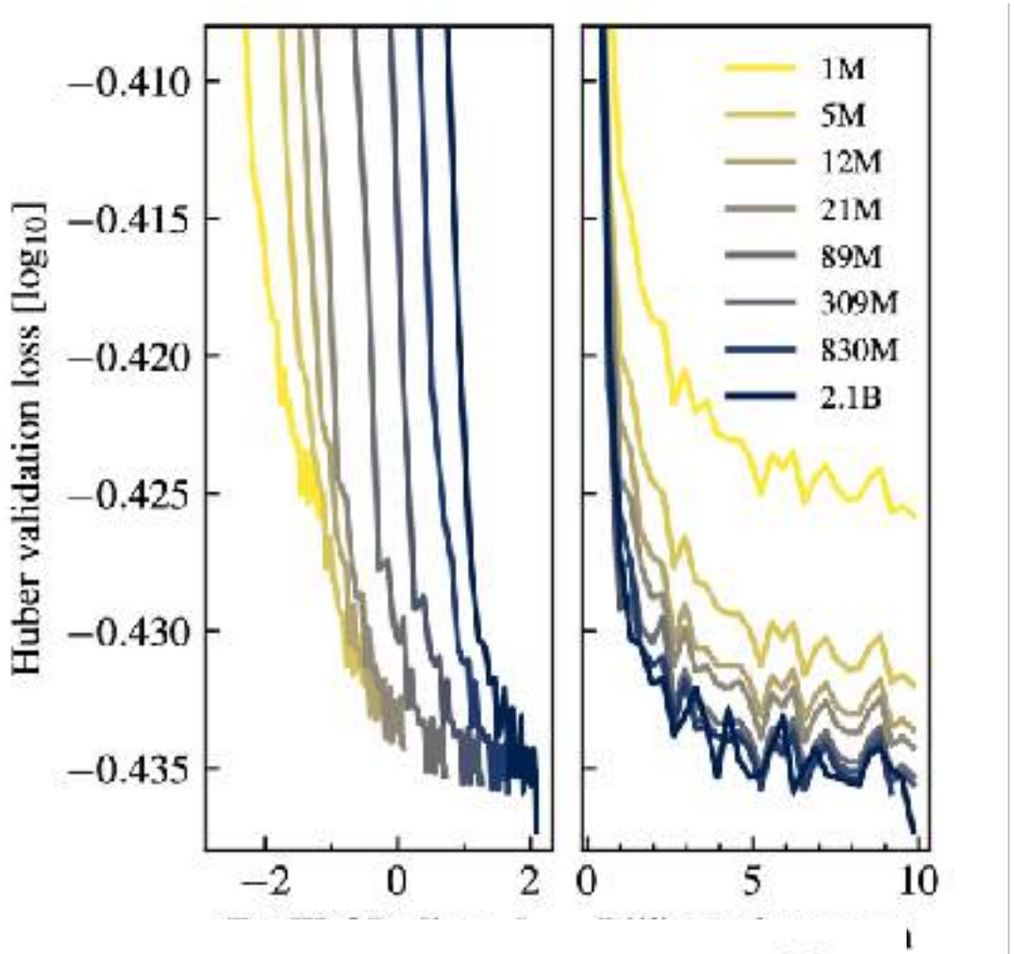
Iyer+24 (uTBD collaboration)


16
15
14
13
12
17
04
03
02
11
18
05
00
01
10
19
06
07
08
09
20
21
22
23
24


Huber validation loss [log10]
−0.410
−0.415
−0.420
−0.425
−0.430
−0.435
−2
0
2
ExaFLOPs [log10]
0
5
10
Billion tokens seen
1M
5M
12M
21M
89M
309M
830M
2.1B

Smith+24

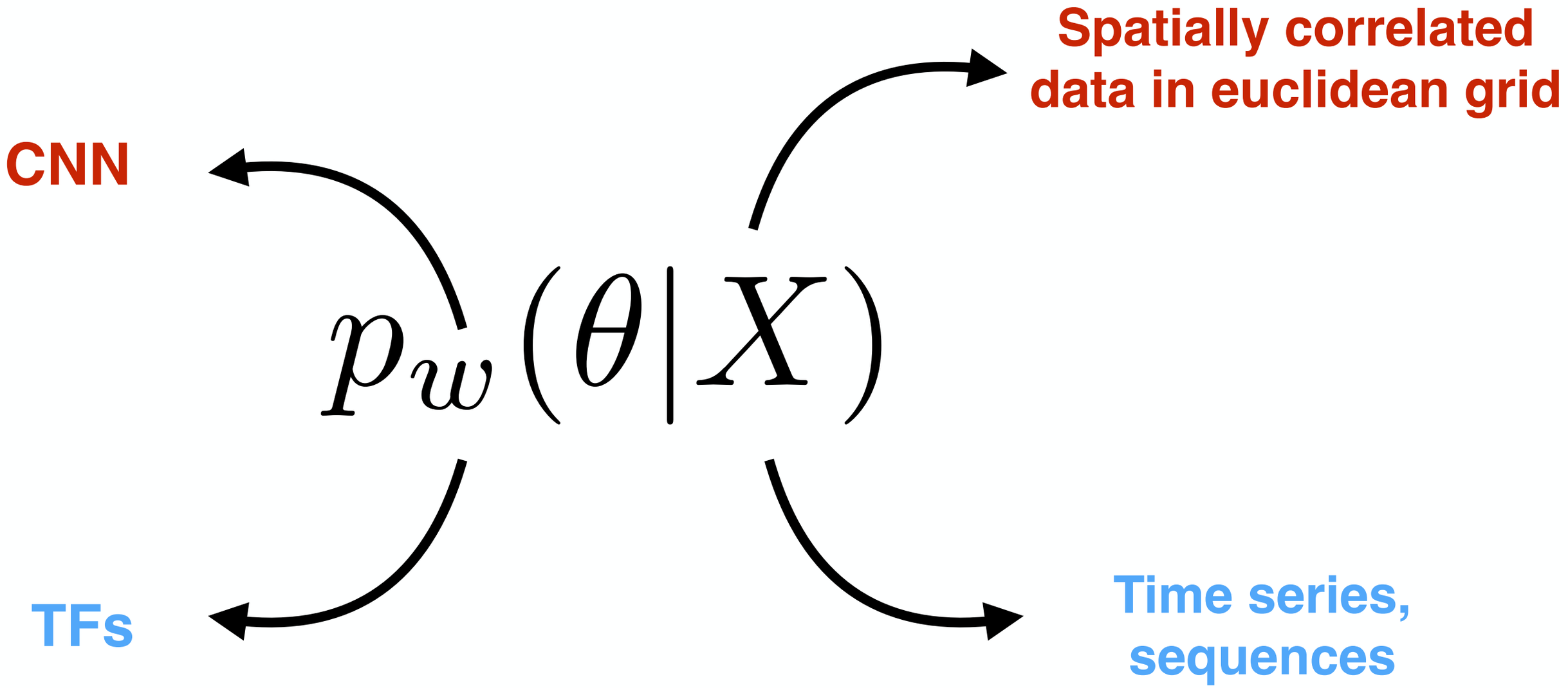




Emerging Properties

Smith+24

RECAP:

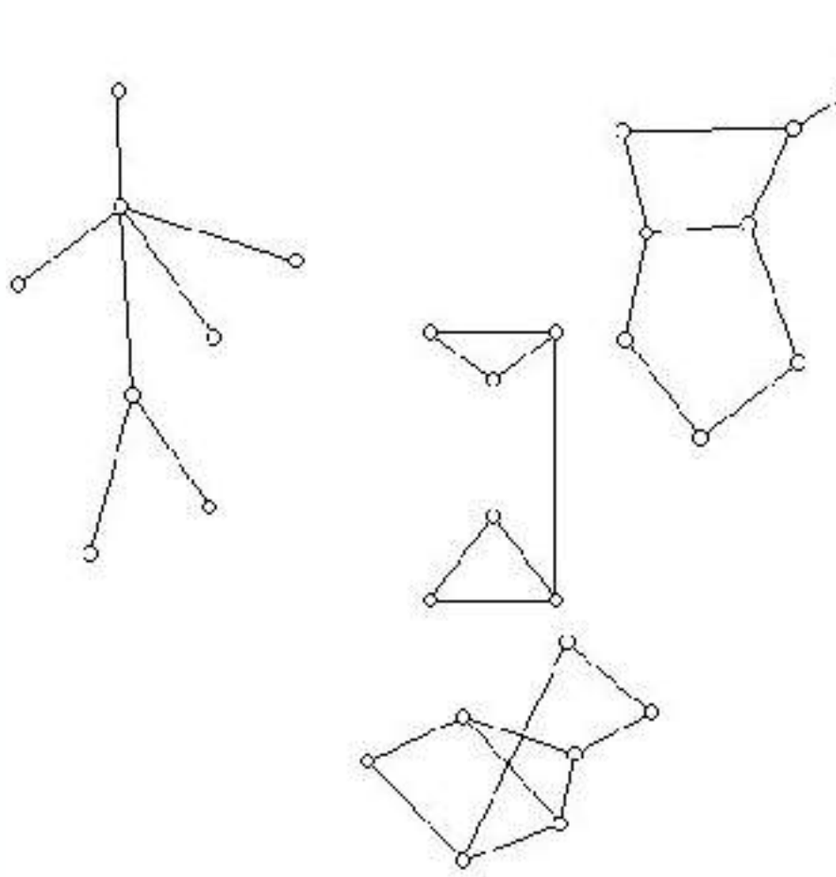

Spatially correlated
data in euclidean grid
CNN
$p_w(\theta|X)$
TFs
Time series,
sequences

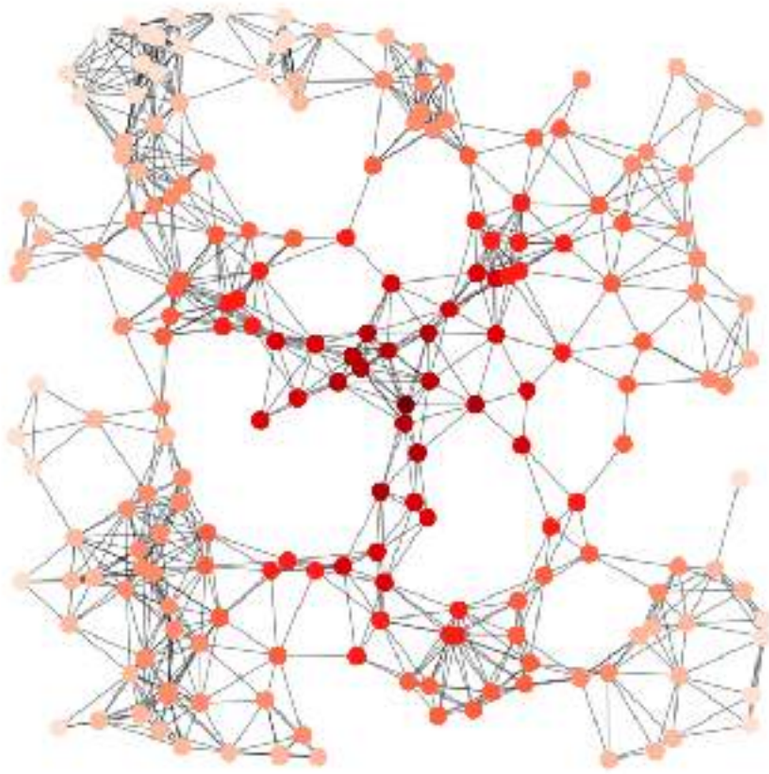
3. A brief incursion into GNNs

$$p_w(\theta|X)$$

X={Whatever}

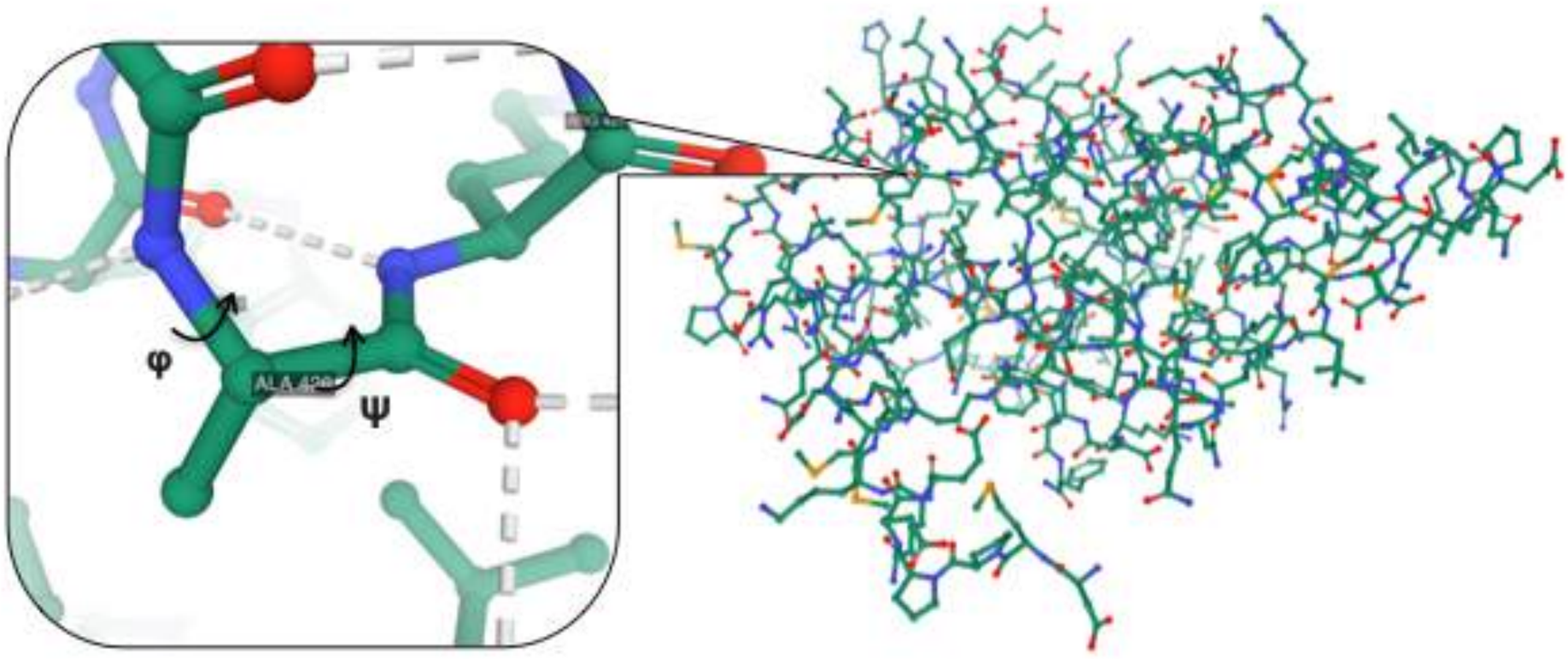
What is a graph?



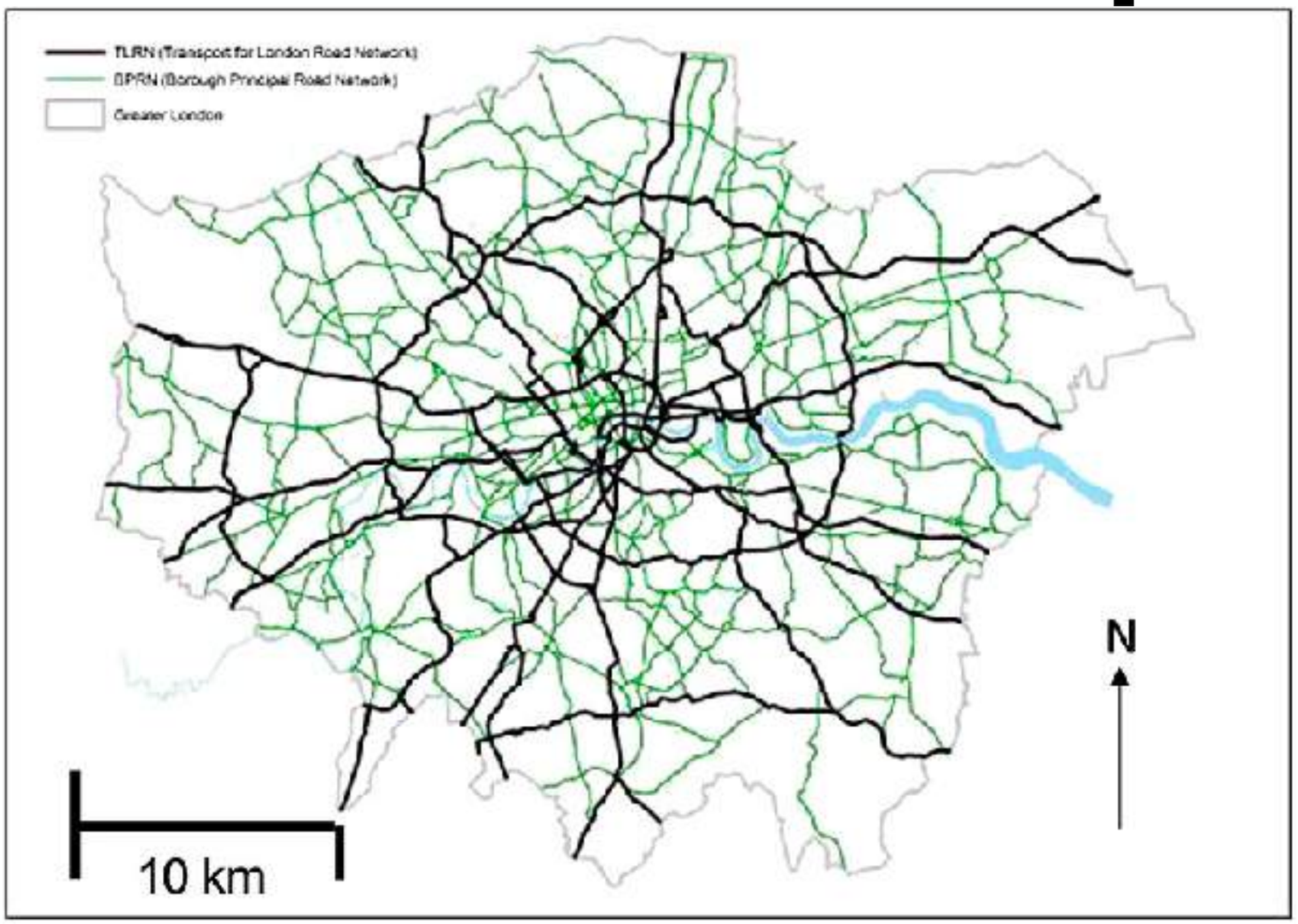


!!!

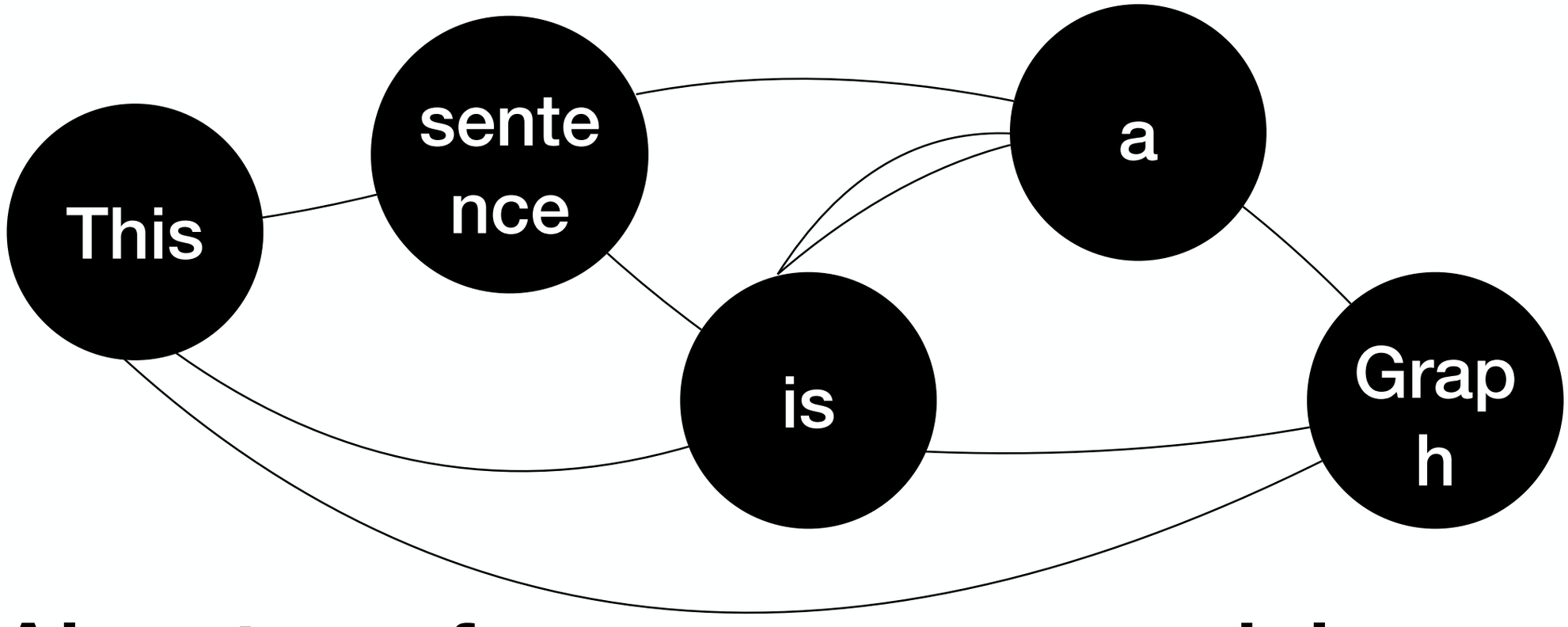
Molecules are Graphs



Traffic is a Graph



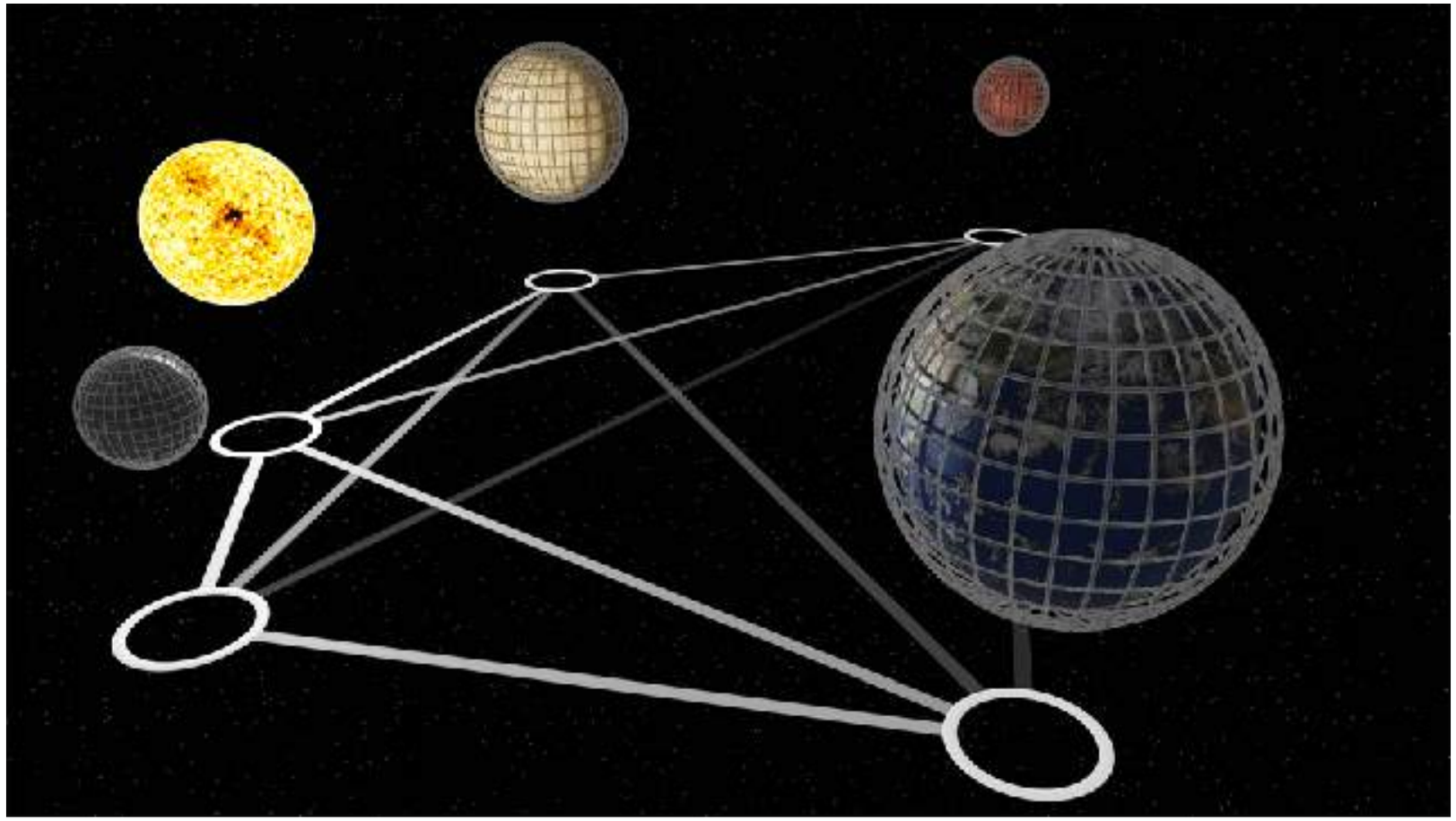
This sentence is a Graph



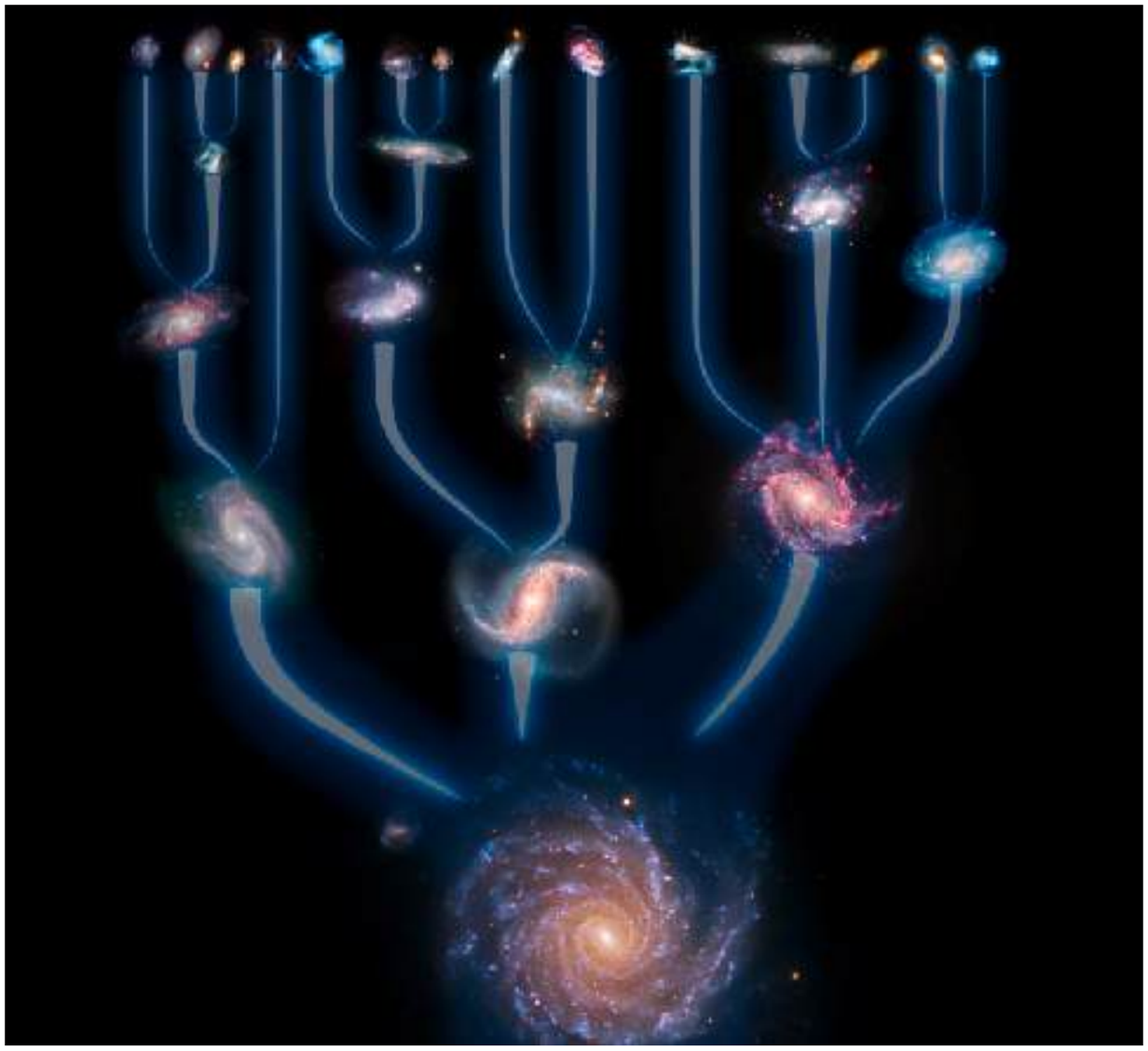
Also, transformers are a special case of GNNs!!

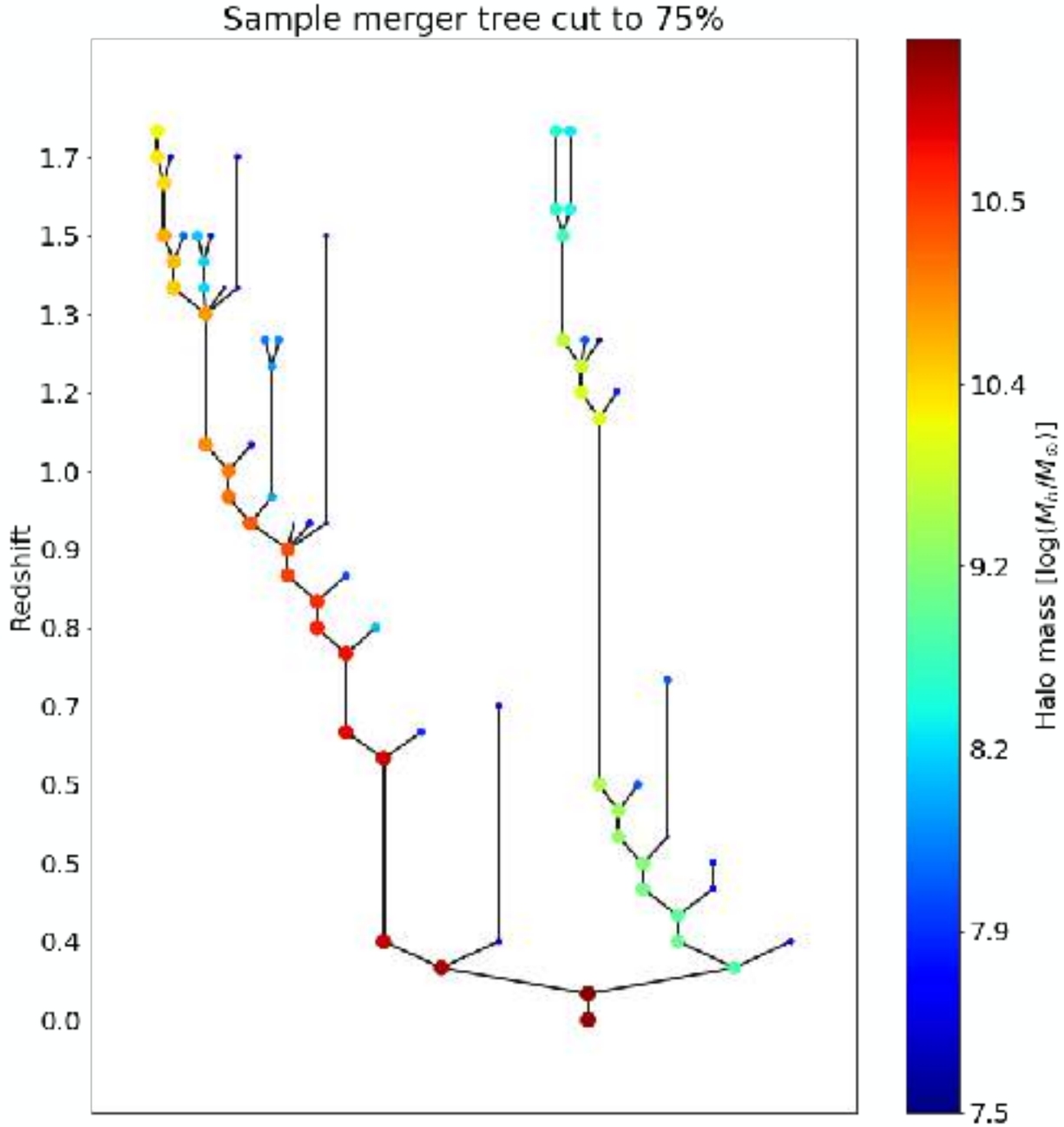
The Solar System is a Graph

Or at least a natural abstraction

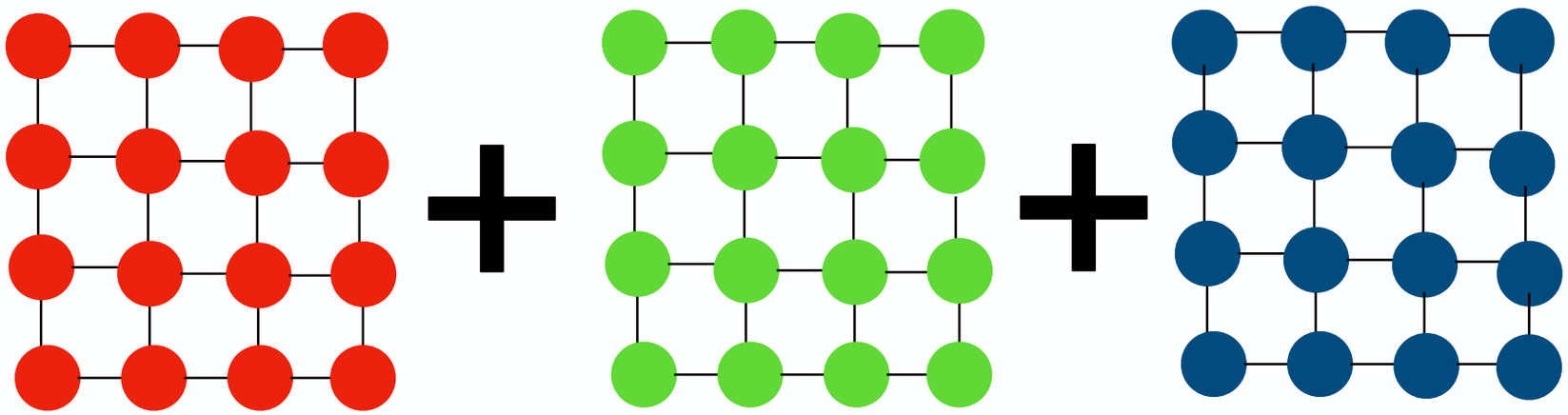


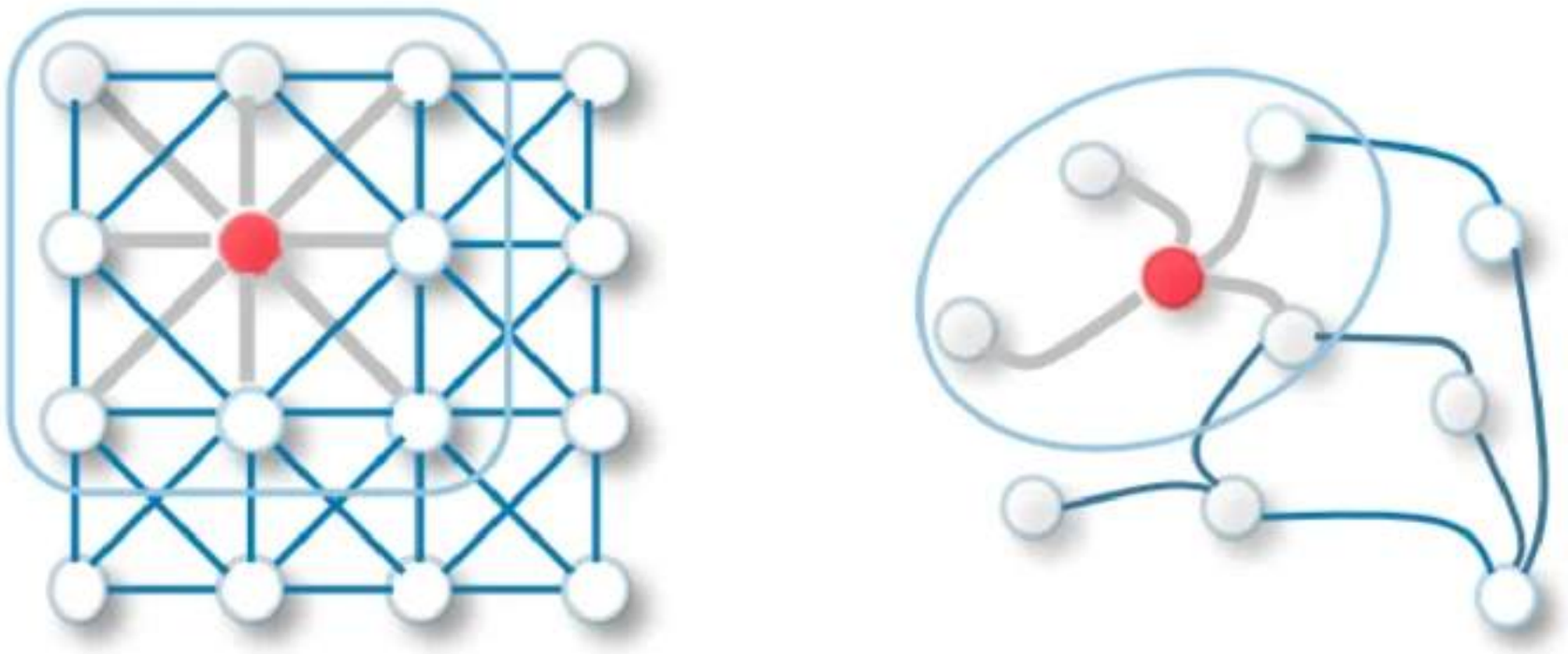
Galaxy Merger trees are Graphs




Sample merger tree cut to 75%
Redshift
1.7
1.5
1.3
1.2
1.0
0.9
0.8
0.7
0.5
0.5
0.4
0.0
Halo mass [log($M_h/M_\odot$)]
10.5
10.4
9.2
8.2
7.9
7.5

Pictures are Graphs



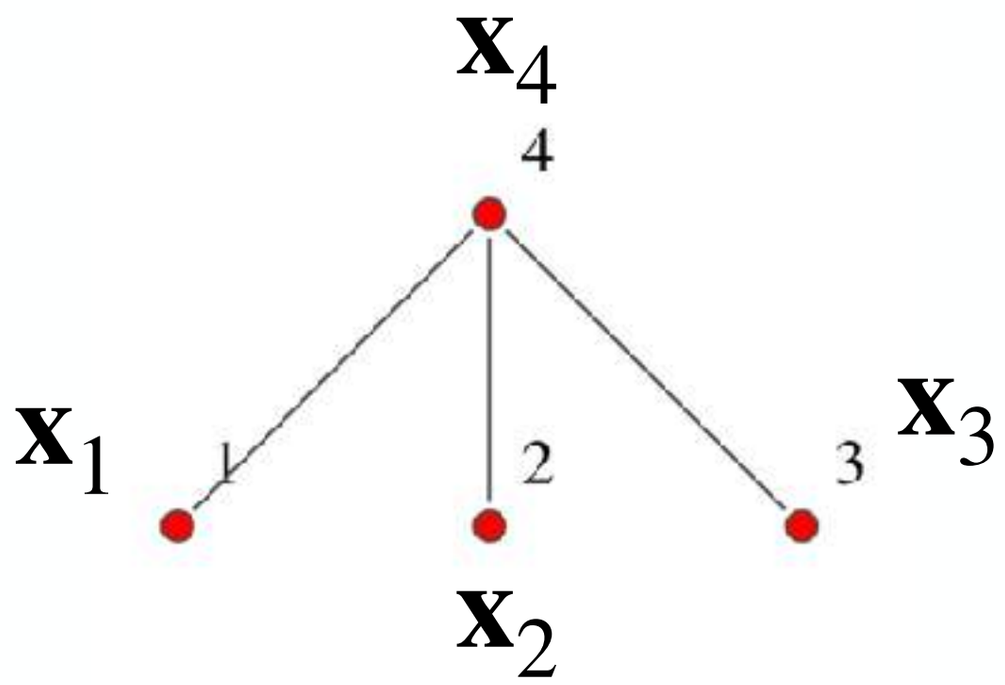


The main difference compared to images is in the definition of neighbours

The harder question is: What is not a graph?

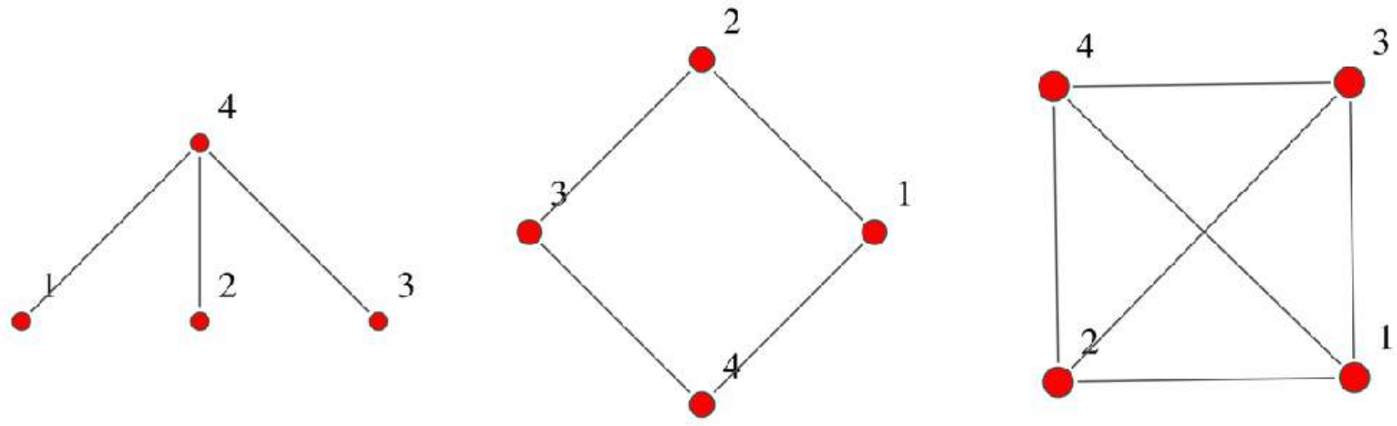
What makes up a graph?

Nodes and node features



What makes up a graph?

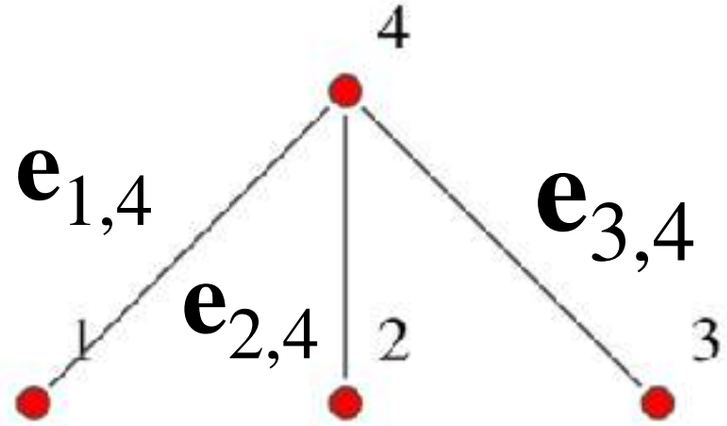
Nodes and node features



$$\mathbf{X} = \left[\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\right]^\top \rightarrow \quad \mathbf{X} = \left[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|\mathcal{V}|}\right]^\top \rightarrow \quad \mathbf{X} = \left\{\mathbf{x}_i \mid i \in N_v\right\} \in \mathbb{R}^{N_v \times d_v}$$

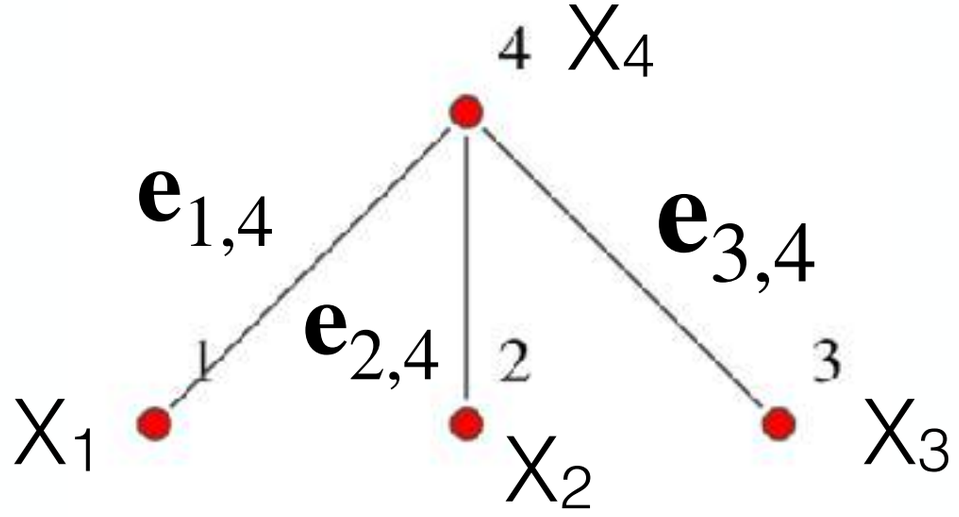
What makes up a graph?

Edges and edge features



What makes up a graph?

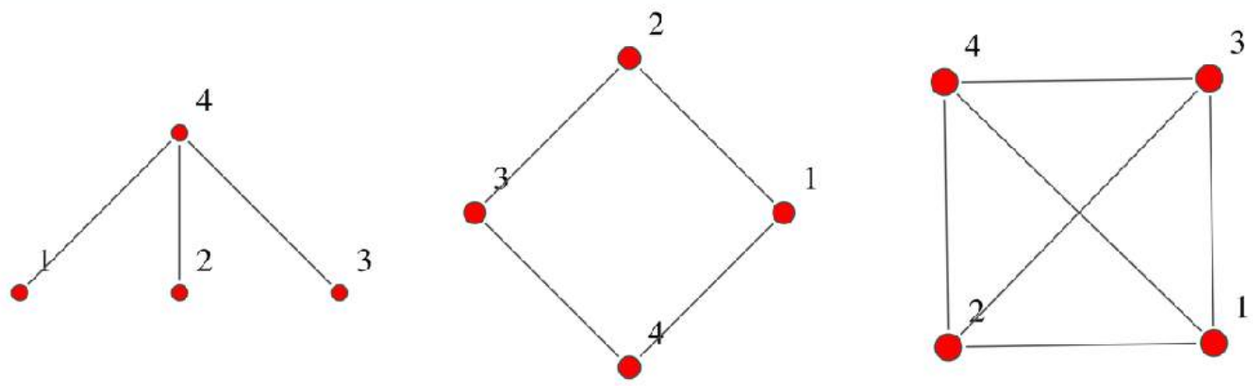
Edges and edge features



$$\mathbf{E} = \left[\mathbf{e}_{12}, \mathbf{e}_{13}, \mathbf{e}_{14}\right]^\top \rightarrow \quad \mathbf{E} = \left[\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{|\mathscr{E}|}\right]^\top \rightarrow \quad \mathbf{E} = \left\{\mathbf{e}_i \mid i \in N_e\right\} \in \mathbb{R}^{N_e \times d_e}$$

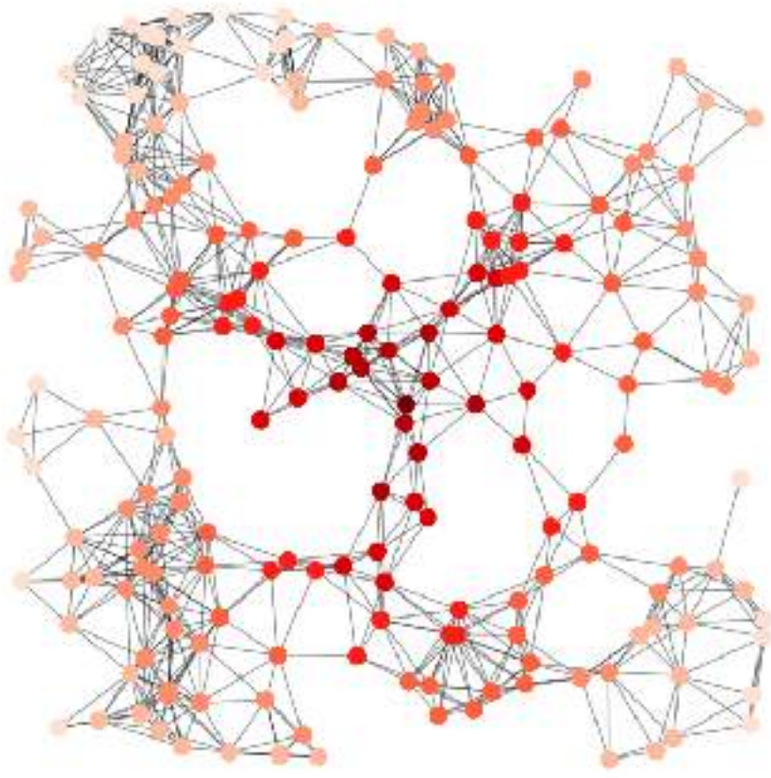
What makes up a graph?

The Adjacency Matrix



$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Global Features



Could be physical properties e.g.

$$\mathbf{U} = [\Omega_m, \sigma_8]$$

Statistical properties,
e.g. manually inserted

$$\mathbf{U} = \left[\mu_{\mathbf{x}}, \sigma_{\mathbf{x}}, max(\mathbf{x}), min(\mathbf{x}) \ldots P_n(\mathbf{x})\right]$$

Or completely learnable

Summarizing…

- Graph defined as $G(V, E, U)$ with G being the graph, V being the set of nodes, E being the set of edges and U being the global feature vector

- Adjacency matrix (A) encodes the structure of the graph and is a binary matrix

Locality constraints

The edges of a graph naturally define the concept of a local **neighborhood**

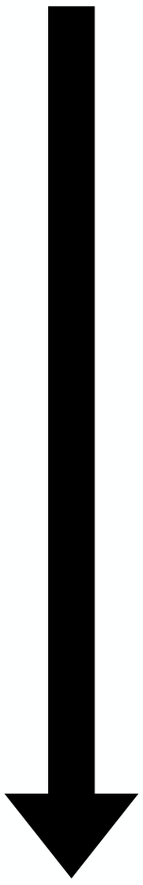
$$\mathcal{N}_u = \{v \mid (u, v) \in \mathbf{E} \vee (v, u) \in \mathbf{E}\}$$

Which allows us to allow us to define local functions, which work on a given node and its neighborhood.

$$\phi\left(\mathbf{x}_u, \mathbf{X}_{\mathcal{N}_u}\right)$$

Graph Neural Network Layers

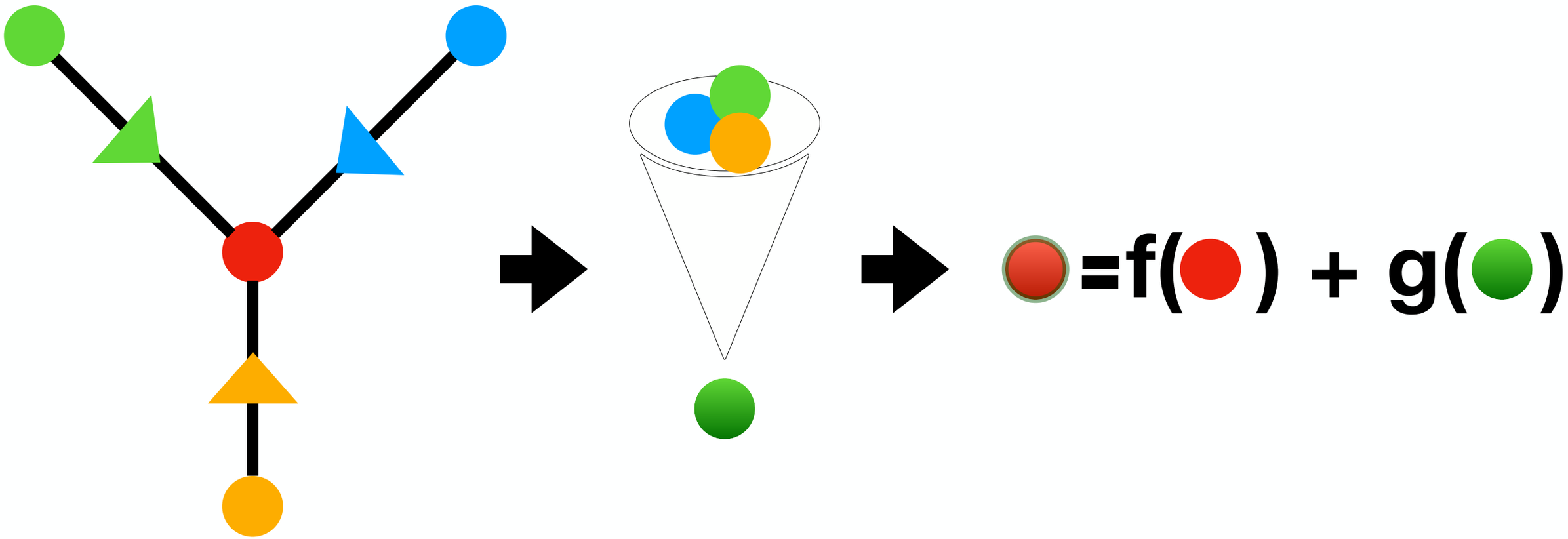
Increasing complexity/ expressivity/ training time



- Convolutional Layer $\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} c_{vu}\psi\left(\mathbf{x}_v\right)\right)$

- Attentional Layer $\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} a(\mathbf{x}_u, \mathbf{x}_v, \mathbf{e}_{uv})\psi\left(\mathbf{x}_v\right)\right)$

Looks complicated, but!



Looks complicated, but!

Essentially, these are all variants of:

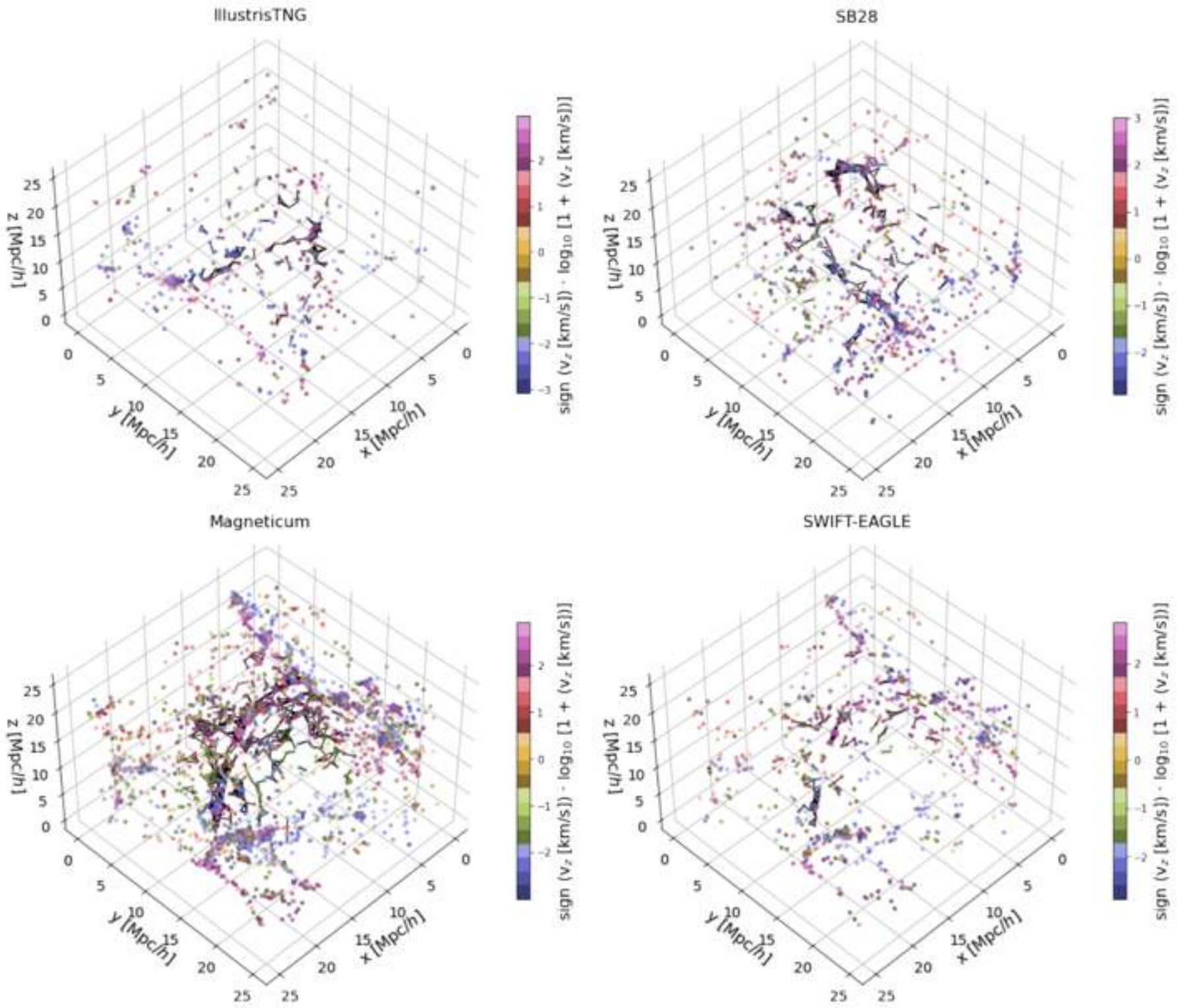
1. Define a node function f and a neighborhood function g (f and g can be the same function)
2. Act with f on the node and with g on the neighborhood

Graphs are good representations of physical systems

In modern physics … a central theme will be a Geometric Principle: The laws of physics must all be expressible as geometric (coordinate-independent and reference frame-independent) relationships between geometric objects (scalars, vectors, tensors, ...) that represent physical entities.

Physics on Graphs

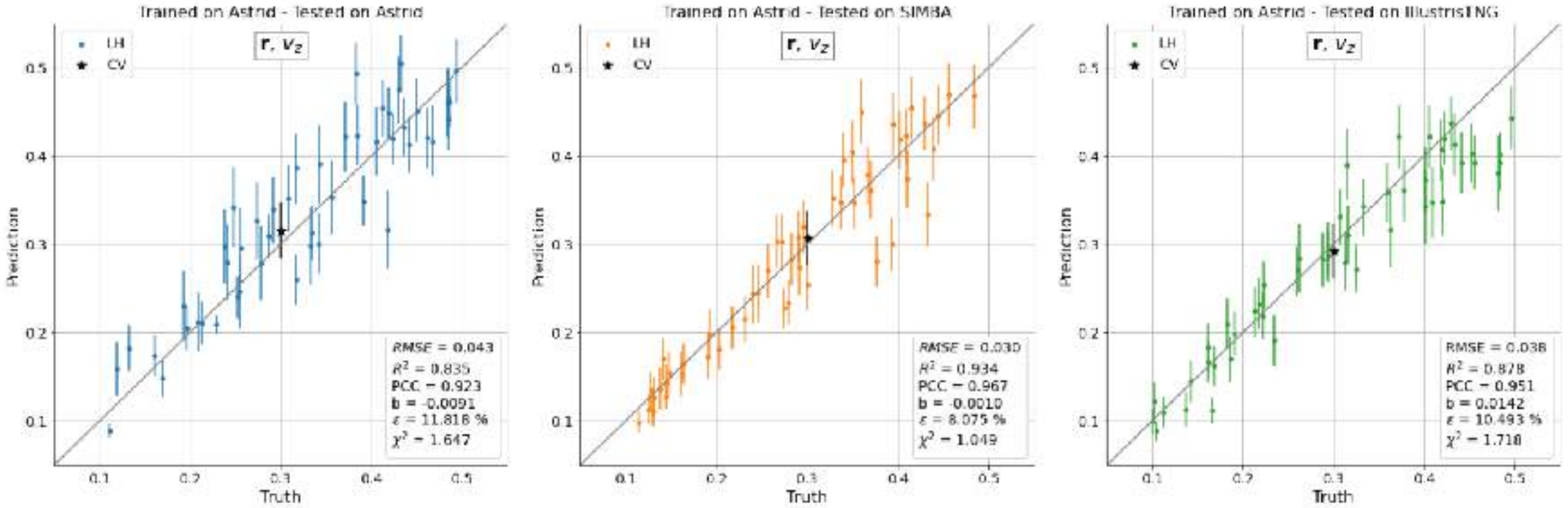
- A natural abstraction of physical systems and inherently local -> easier physical interpretations/separability
- Embeds inductive biases easily by restructuring the graph -> more efficient learning and no need to learn things we already know
- Can embed permutational, rotational, translational and reflectional symmetries



De Santi+24

Field Level Cosmological with Galaxies

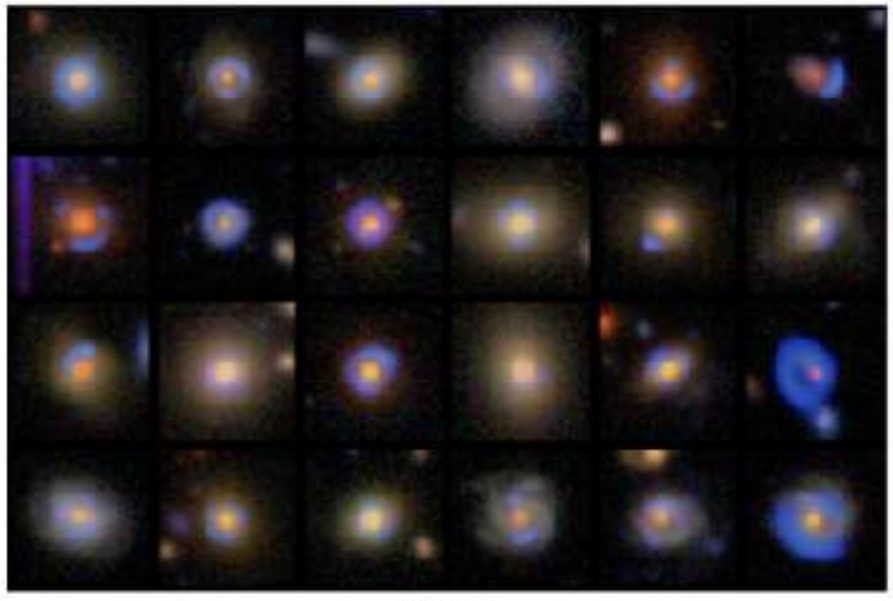
Field Level Cosmological with Galaxies



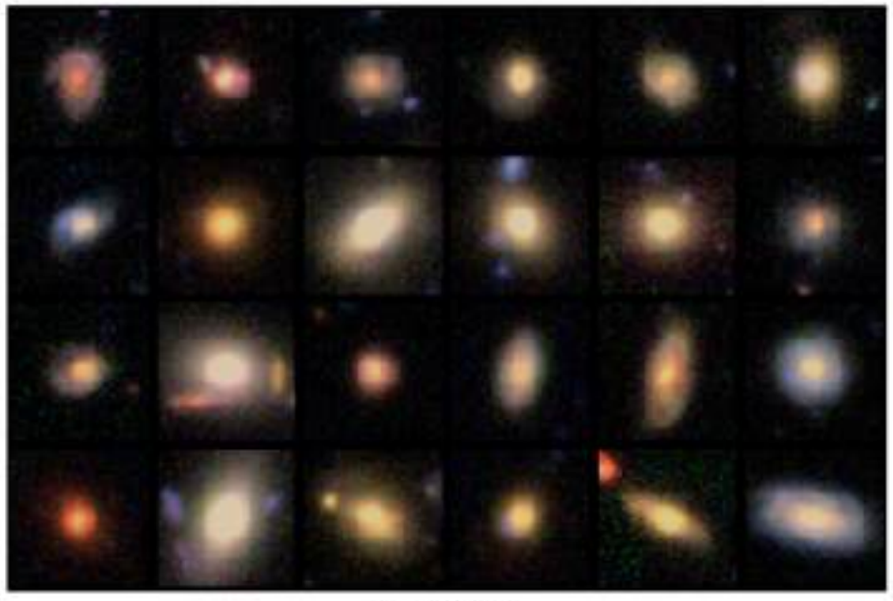
De Santi+24

EXTRA MATERIAL: CNNs

1. Classification



LENS



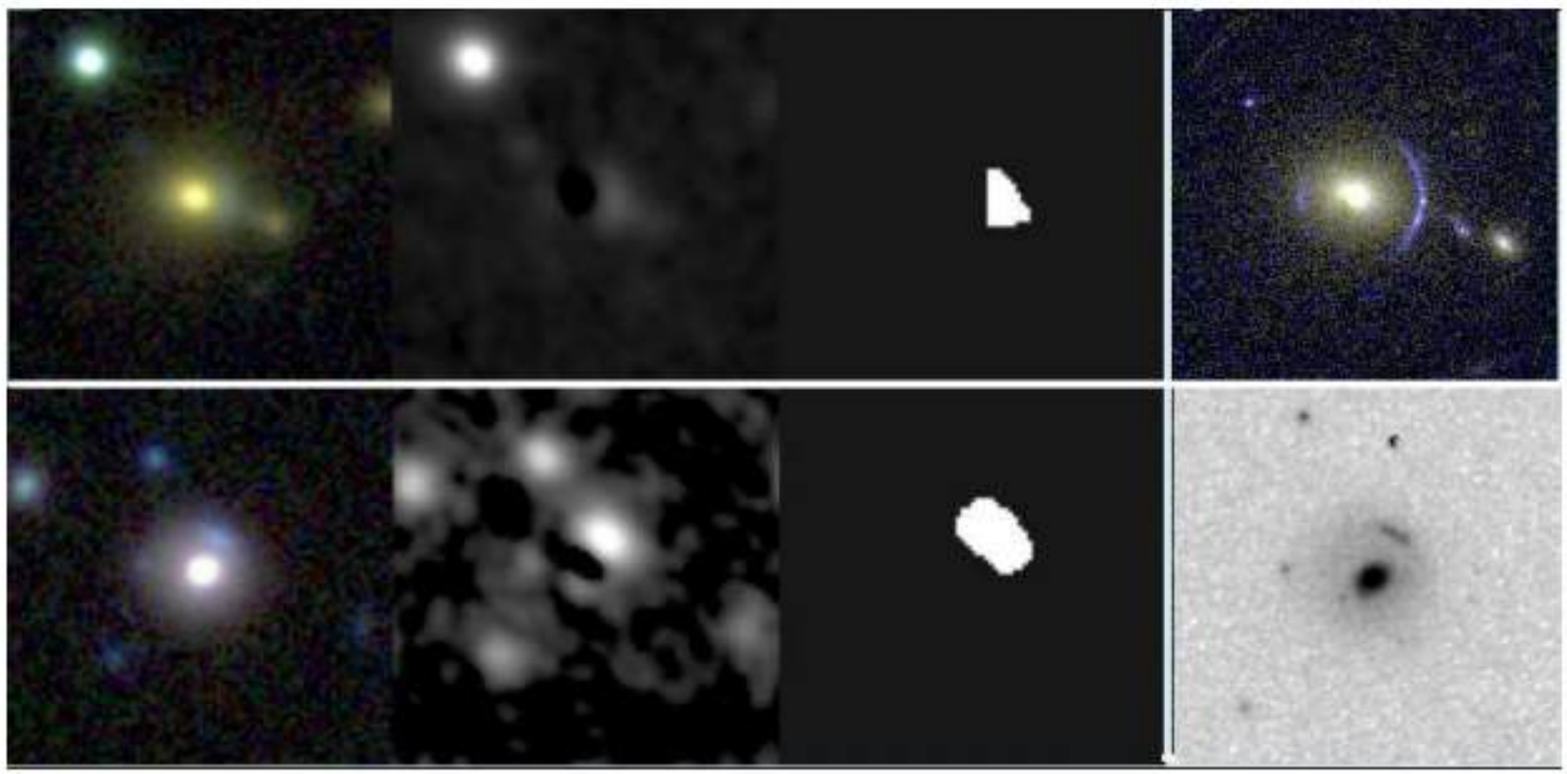
NON-LENS

Detection of Strong Lenses
Valuable information of Dark Matter properties

Future surveys will increase the samples by orders of magnitude.

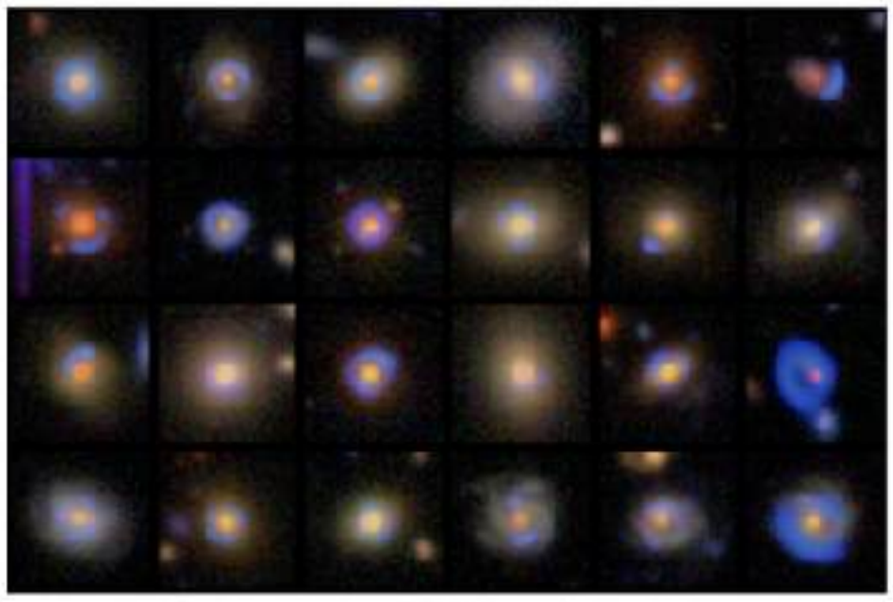
Jacobs+17

"Pre Deep Learning" Approach

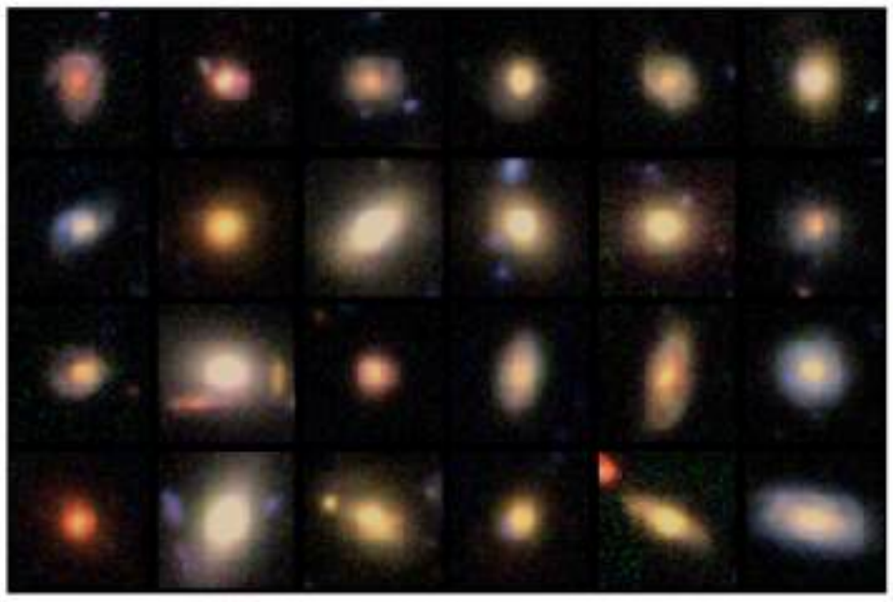


Gavazzi+17

1. Classification

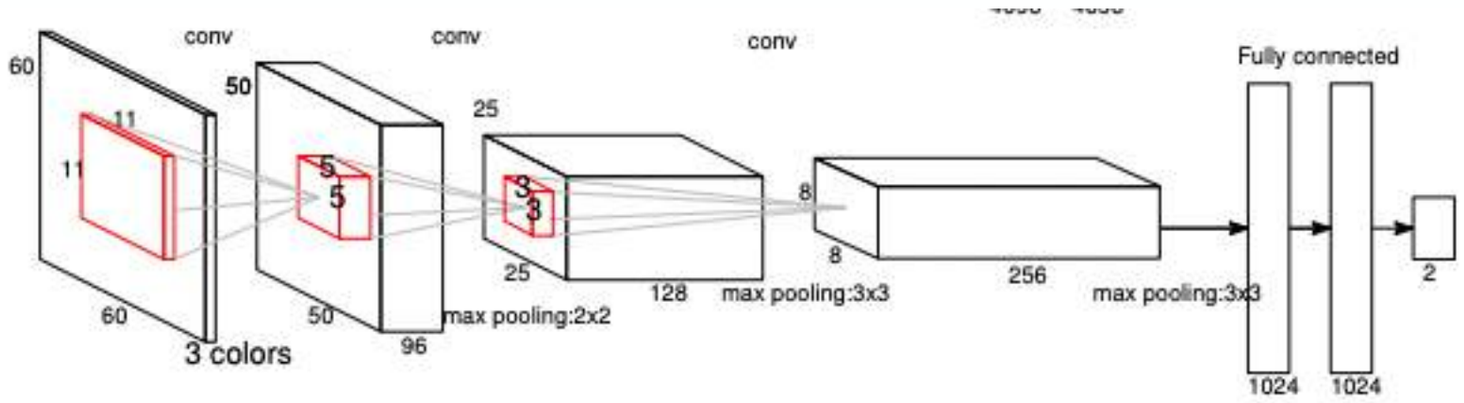


LENS

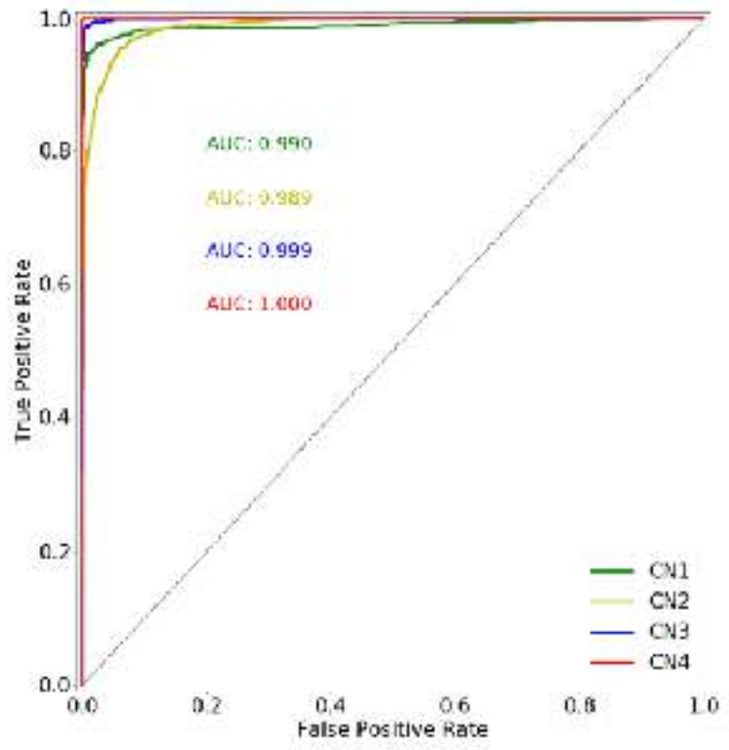


NON-LENS

Jacobs+17



It illustrates the change of paradigm from an algorithmic centric focus to a purely data driven approach to data



Dieleman+15, Huertas-Company+15, Aniyan+17, Charnock+17, Gieseke+17, Jacobs+17, Petrillo+17, Schawinski+17, Alhassan+18, Dominguez-Sanchez+18, George+18, Hinners+18, Lukic+18, Moss+18, Razzano+18, Schaefer+18, Allen+19, Burke+19, Carrasco-Davis+19, Chatterjee+19, Davies+19, Dominguez-Sanchez+19, Fusse+19, Glaser+19, Ishida+19, Jacobs+19, Katebi+19, Lanusse+19Liu+19, Lukic+19, Metcalf+19, Muthukrishna+19, Petrillo+19, Reiman+19, Boucaud+20, Chiani+20, Ghosh+20, Gomez+20, Hausen+20, Hlozek+20, Hosseinzadeh+20. Huang+20, Li+20, Moller+20, Paillassa+20, Tadaki+20, Vargas dos Santos+20, Walmsley+20, Wei+20, Allam+21, Arcelin+21, Becker+21, Bretonniere+21, Burhanudin+21, Davison+21, Donoso-Oliva+21, Jia+21, Lauritsen+21, Ono+21, Ruan+21, Sadegho+21, Tang+21, Tanoglidis+21, Vojtekova+21, Dhar+22Hausen+22, Orwat+22, Pimentel+22, Rezaei+22, Samudre+22, Shen+22, Walmsley+22

1. Classification

CNN based classifications reach unprecedented accuracy

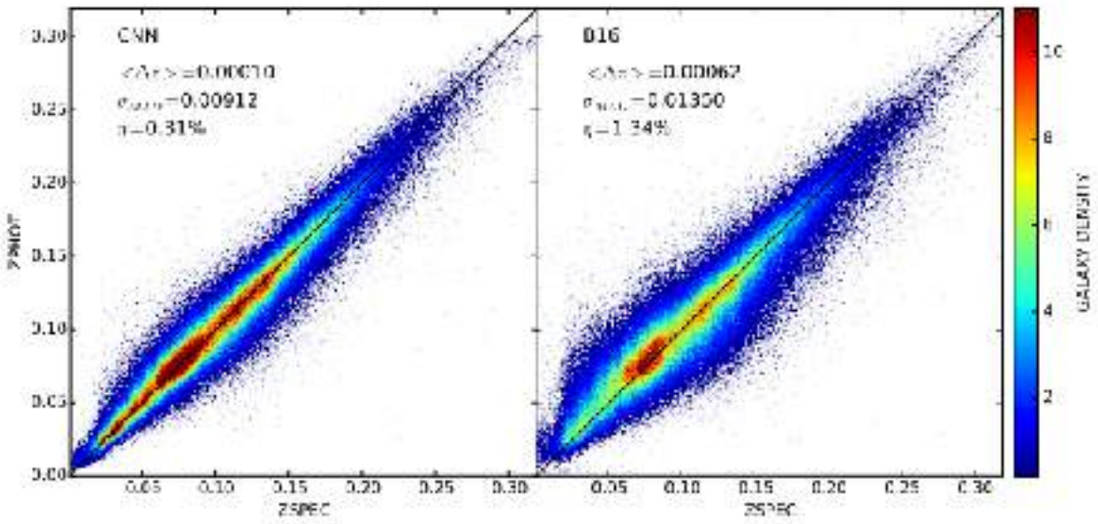
Name	type	AUROC	TPR_0	TPR_{10}	short description
CMU-DeepLens-Resnet-ground3	Ground-Based	0.98	0.09	0.45	CNN
CMU-DeepLens-Resnet-Voting	Ground-Based	0.98	0.02	0.10	CNN
LASTRO EPFL	Ground-Based	0.97	0.07	0.11	CNN
CAS Swinburne Melb	Ground-Based	0.96	0.02	0.08	CNN
AstrOmatic	Ground-Based	0.96	0.00	0.01	CNN
Manchester SVM	Ground-Based	0.93	0.22	0.35	SVM / Gabor
Manchester2	Ground-Based	0.89	0.00	0.01	Human Inspection
ALL-star	Ground-Based	0.84	0.01	0.02	edges/gradiants and Logistic Reg.
CAST	Ground-Based	0.83	0.00	0.00	CNN / SVM
YattaLensLite	Ground-Based	0.82	0.00	0.00	SExtractor
LASTRO EPFL	Space-Based	0.93	0.00	0.08	CNN
CMU-DeepLens-Resnet	Space-Based	0.92	0.22	0.29	CNN
GAMOCLASS	Space-Based	0.92	0.07	0.36	CNN
CMU-DeepLens-Resnet-Voting	Space-Based	0.91	0.00	0.01	CNN
AstrOmatic	Space-Based	0.91	0.00	0.01	CNN
CMU-DeepLens-Resnet-aug	Space-Based	0.91	0.00	0.00	CNN
Kapteyn Resnet	Space-Based	0.82	0.00	0.00	CNN
CAST	Space-Based	0.81	0.07	0.12	CNN
Manchester1	Space-Based	0.81	0.01	0.17	Human Inspection
Manchester SVM	Space-Based	0.81	0.03	0.08	SVM / Gabor
NeuralNet2	Space-Based	0.76	0.00	0.00	CNN / wavelets
YattaLensLite	Space-Based	0.76	0.00	0.00	Arcs / SExtractor
All-now	Space-Based	0.73	0.05	0.07	edges/gradiants and Logistic Reg.
GAHEC IRAP	Space-Based	0.66	0.00	0.01	arc finder

Metcalf+19

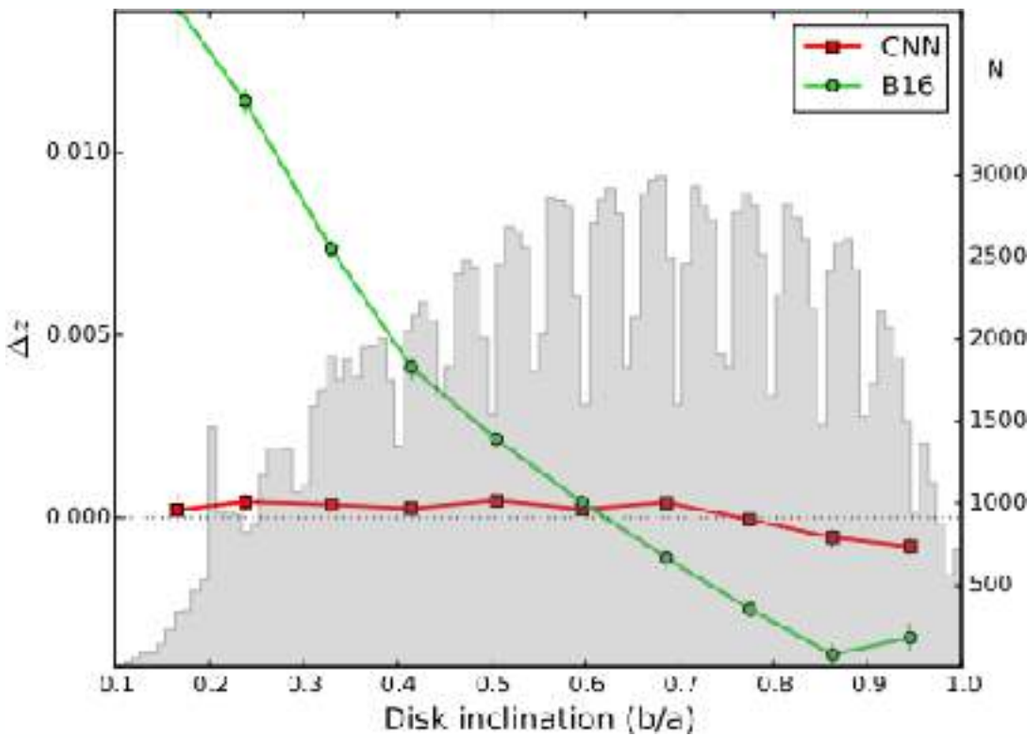
Dieleman+15, Huertas-Company+15, Aniyan+17, Charnock+17, Gieseke+17, Jacobs+17, Petrillo+17, Schawinski+17, Alhassan+18, Dominguez-Sanchez+18, George+18, Hinners+18, Lukic+18, Moss+18, Razzano+18, Schaefer+18, Allen+19, Burke+19, Carrasco-Davis+19, Chatterjee+19, Davies+19, Dominguez-Sanchez+19, Fusse+19, Glaser+19, Ishida+19, Jacobs+19, Katebi+19, Lanusse+19Liu+19, Lukic+19, Metcalf+19, Muthukrishna+19, Petrillo+19, Reiman+19, Boucaud+20, Chiani+20, Ghosh+20, Gomez+20, Hausen+20, Hlozek+20, Hosseinzadeh+20. Huang+20, Li+20, Moller+20, Paillassa+20, Tadaki+20, Vargas dos Santos+20, Walmsley+20, Wei+20, Allam+21, Arcelin+21, Becker+21, Bretonniere+21, Burhanudin+21, Davison+21, Donoso-Oliva+21, Jia+21, Lauritsen+21, Ono+21, Ruan+21, Sadegho+21, Tang+21, Tanoglidis+21, Vojtekova+21, Dhar+22Hausen+22, Orwat+22, Pimentel+22, Rezaei+22, Samudre+22, Shen+22, Walmsley+22

Photometric Redshifts

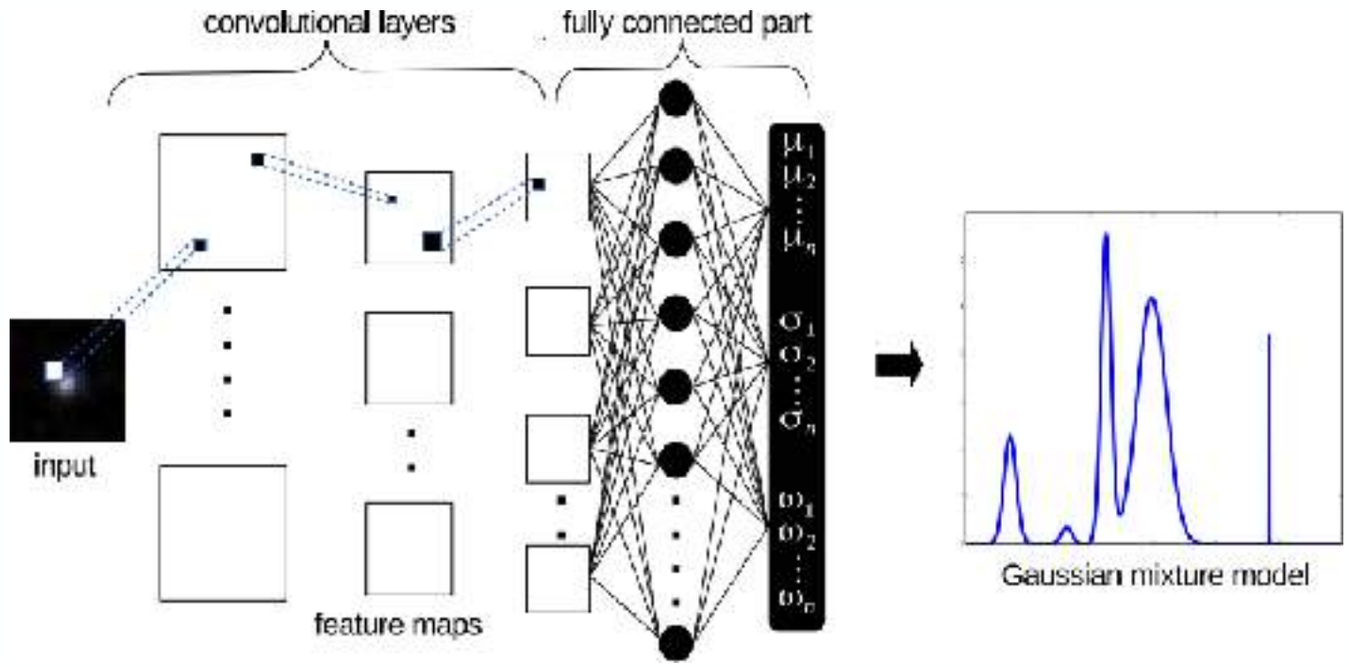
Deep Learning **Classical approach**



Geometric Effects are automatically considered (beyond photometry)



Pasquet+18

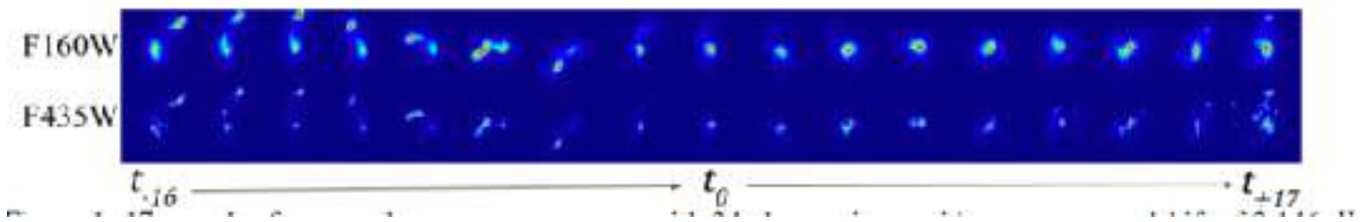


Disanto+18

Uncertainty quantification through Mixture Density Networks

Hazevah+17, Perreault-Levasseur+17, Morningstar+18, Stark+18, Tuccillo+18, Bom+19, Lovell+19, Madireddy+19, Pearson+19, Simet+19, Pasquet+19, Wu+19, Menou+19, Aragon-Calvo+20, Chianese+20, Stahl+20, Surana+20, Shuntov+20, Cabayol-Garcia+20, Campagne+20, Buck+21, Grover+21, Hayat+21, Li+21, Maresca+21, Qiu+21, Rhea+21, Schuldt+21, Tohill+21, Yao-Yu+21, Dey+21, Lee+21, Zhou+21, Henghes+21, Ansari+21

Mergers of Galaxies

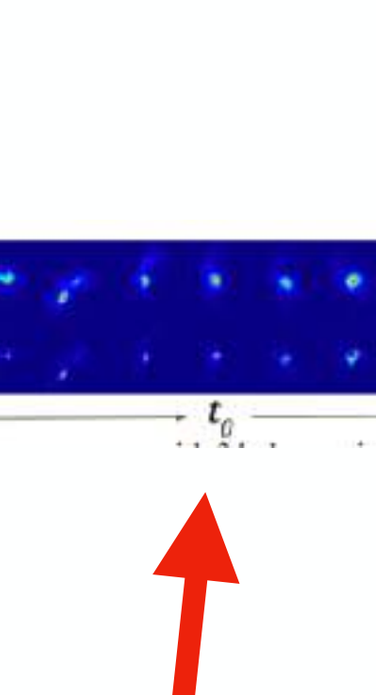


Merger of galaxies sequence from cosmological simulations

?

?

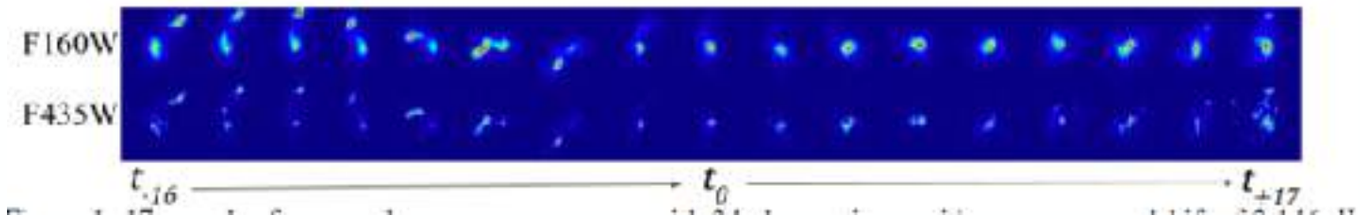
Merger stage?



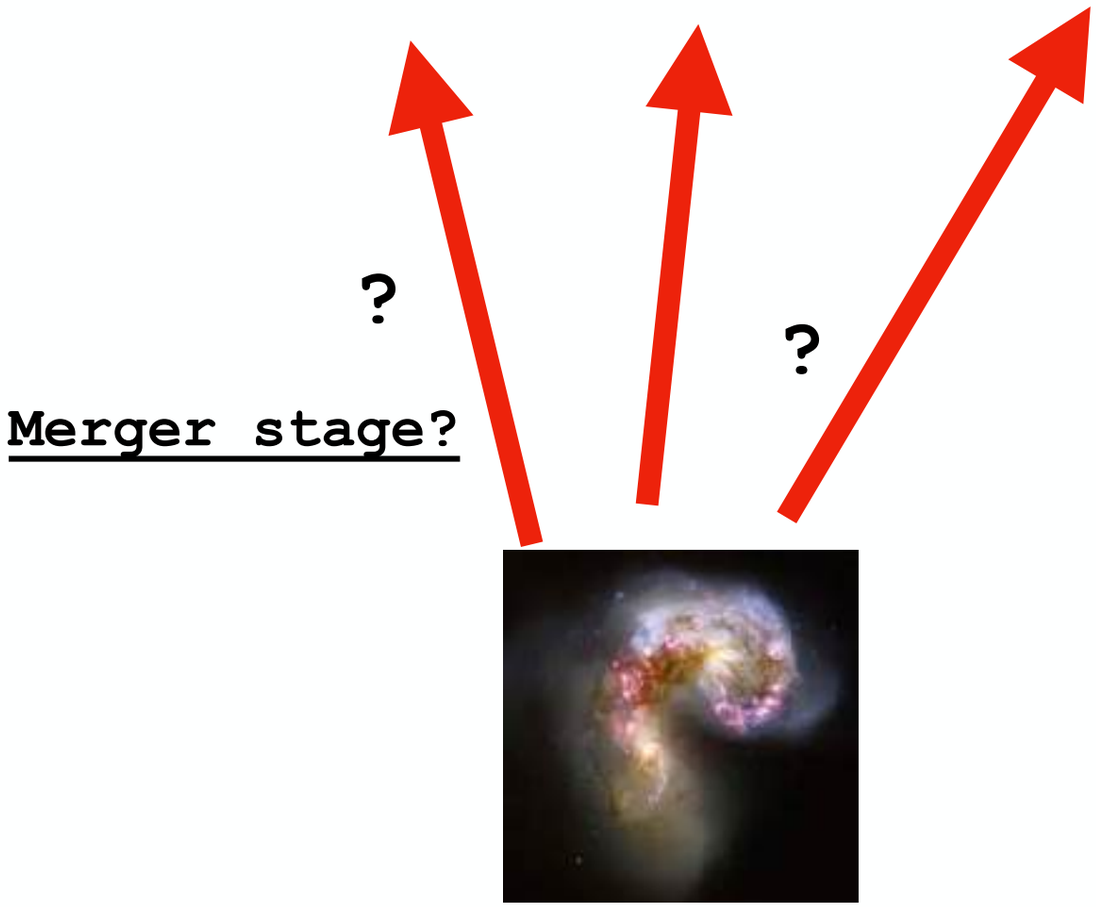
Neural Networks to find relations between observables and physical processes

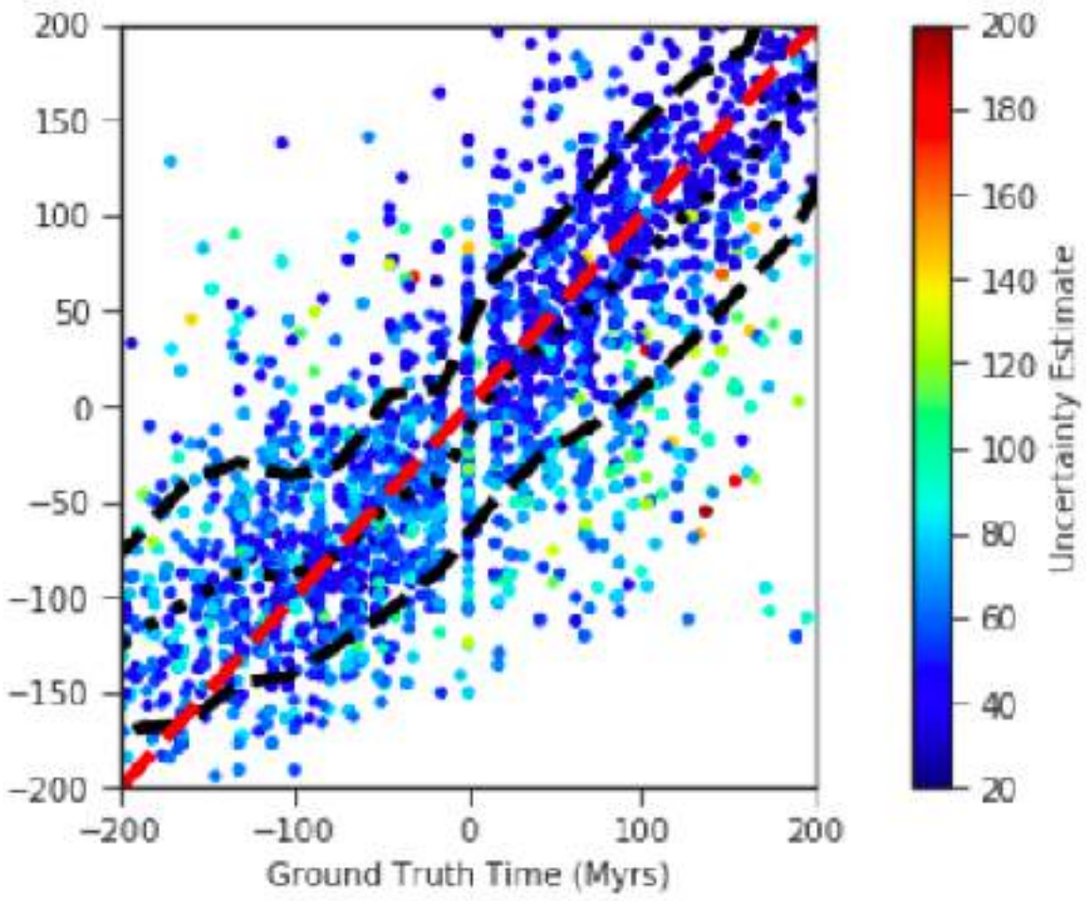
Ntampaka+16, Huertas-Company+18, Schawinski+18, Armitage+19, Bottrell+19, Calderon+19, Diaz+19, Ho+19, Pearson+19, Cai+20, Ferreira+20, Hansen+20, Kodi Ramanah+20, Wu+20, Yan+20, Gupta+20, de Andres+21, Bickley+21, Ciprijanovic+21, Ho+21, Koppula+21, von Marttens+21, Shao+21, Villanueva-Domingo+21, Whithney+21, Holwerda+21

Mergers of Galaxies



Merger of galaxies sequence from cosmological simulations



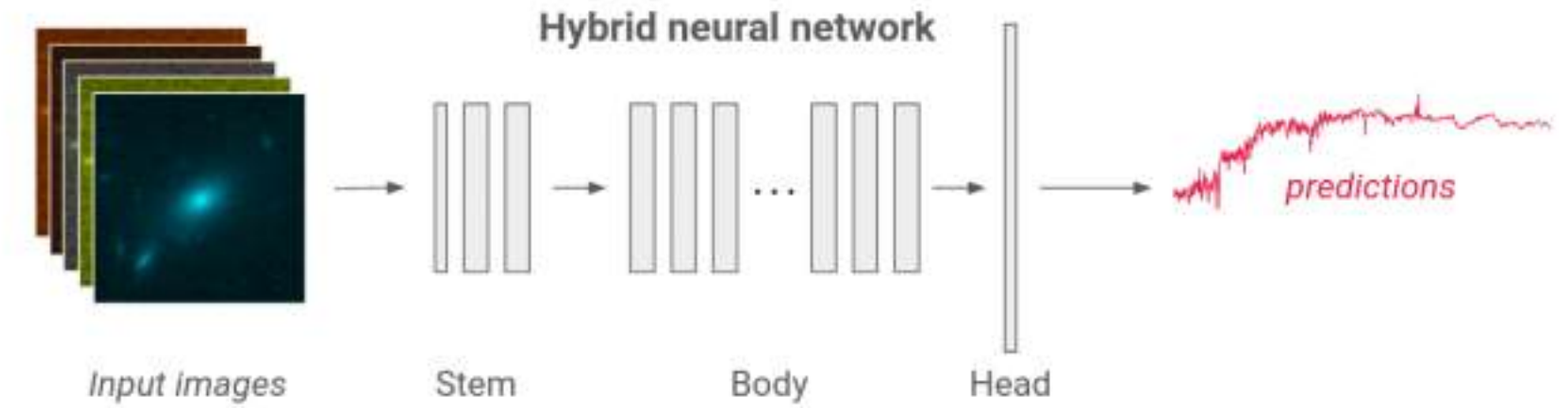


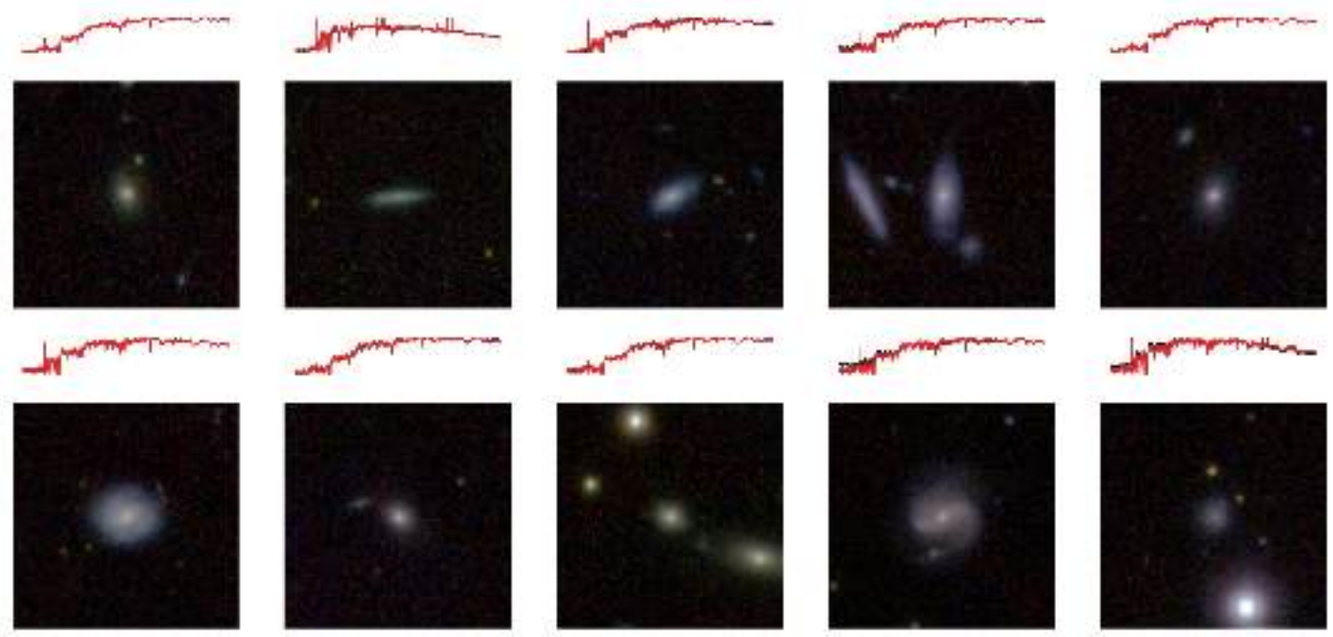
Neural Networks to find relations between observables and physical processes

Koppula+21

Ntampaka+16, Huertas-Company+18, Schawinski+18, Armitage+19, Bottrell+19, Calderon+19, Diaz+19, Ho+19, Pearson+19, Cai+20, Ferreira+20, Hansen+20, Kodi Ramanah+20, Wu+20, Yan+20, Gupta+20, de Andres+21, Bickley+21, Ciprijanovic+21, Ho+21, Koppula+21, von Marttens+21, Shao+21, Villanueva-Domingo+21, Whithney+21, Holwerda+21

Virtual Observatory



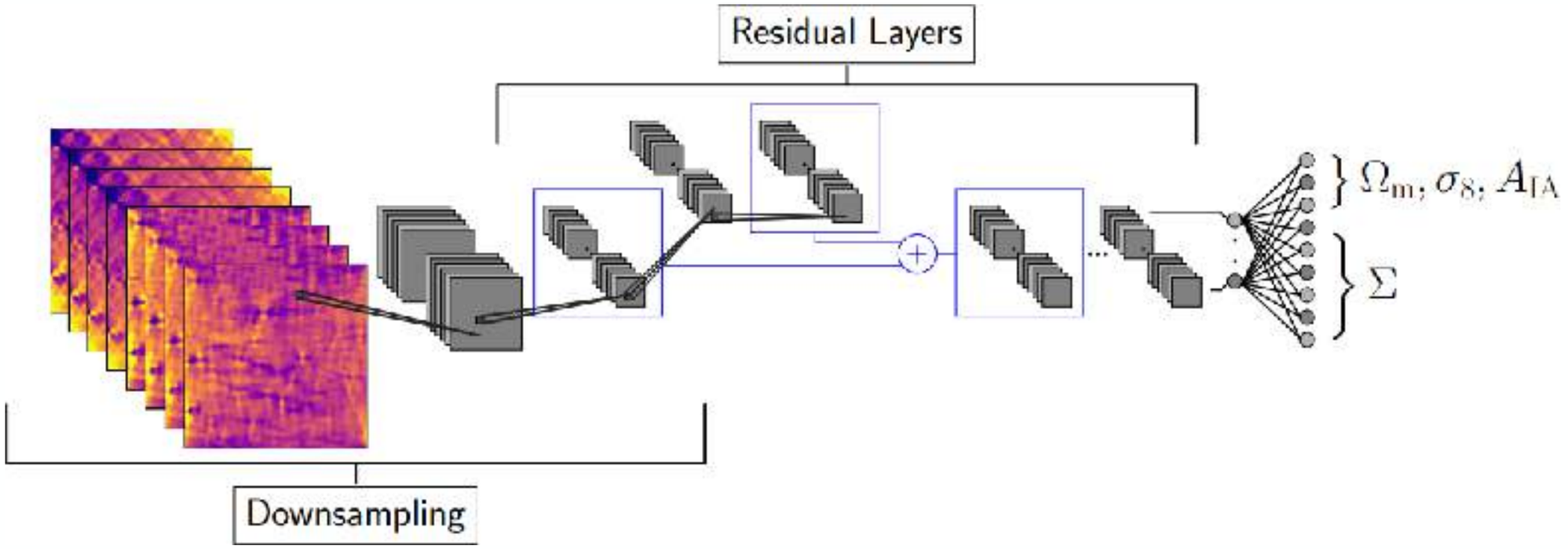


Neural Networks to learn complex mapping between observables

Wu+21

Ntampaka+16, Huertas-Company+18, Schawinski+18, Armitage+19, Bottrell+19, Calderon+19, Diaz+19, Ho+19, Pearson+19, Cai+20, Ferreira+20, Hansen+20, Kodi Ramanah+20, Wu+20, Yan+20, Gupta+20, de Andres+21, Bickley+21, Ciprijanovic+21, Ho+21, Koppula+21, von Marttens+21, Shao+21, Villanueva-Domingo+21, Whithney+21, Holwerda+21

Deep Learning for Cosmological Inference



Motivation: Generalize comparison of observations with theory, beyond basic summary statistics

Neural Networks are used as efficient feature extractors

Adding additional invariances

DATA AUGMENTATION

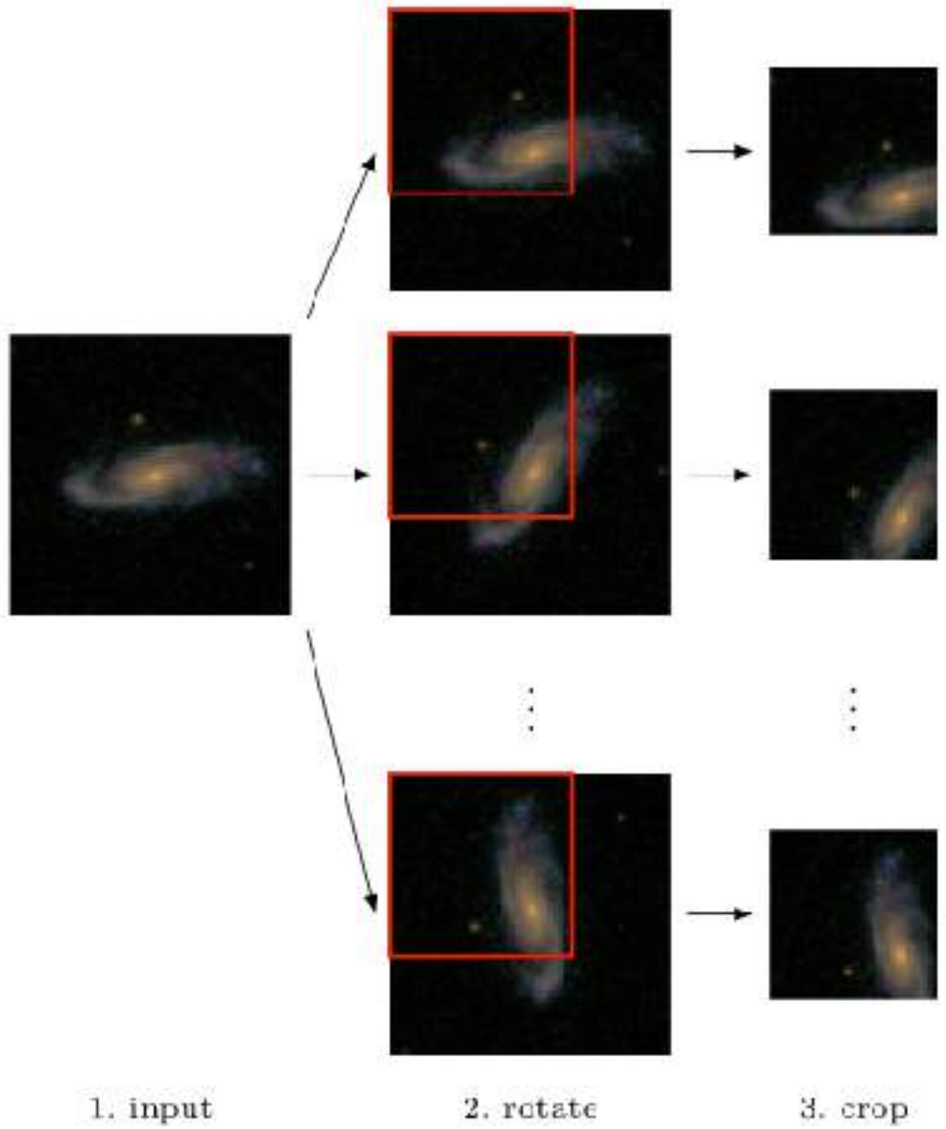
ANOTHER WAY TO REDUCE OVER-FITTING IS TO "AUGMENT" THE SIZE OF THE DATASET AVAILABLE FOR TRAINING

FOR MANY APPLICATIONS THE CLASSIFICATION SHOULD BE INDEPENDENT TO:

- TRANSALTIONS
- ROTATIONS
- SCALINGS

ETC…

DATA [illegible]TATION



Dieleman+15

FOR MANY APPLICATIONS THE CLASSIFICATION SHOULD BE INDEPENDENT TO:

- TRANSALTIONS
- ROTATIONS
- SCALINGS

ETC…

DATA AUGMENTATION

FOR MANY APPLICATIONS THE CLASSIFICATION SHOULD BE INDEPENDENT TO:

- TRANSALTIONS
- ROTATIONS
- SCALINGS

ETC…

THE PRICE TO PAY?

1. LARGE NUMBER OF PARAMETERS IMPLIES LARGE DATASETS TO TRAIN

2. LOOSE EVEN MORE DEGREE OF CONTROL OF WHAT THE ALGORITHM IS DOING SINCE THE FEATURE EXTRACTION PROCESS BECOMES UNSUPERVISED

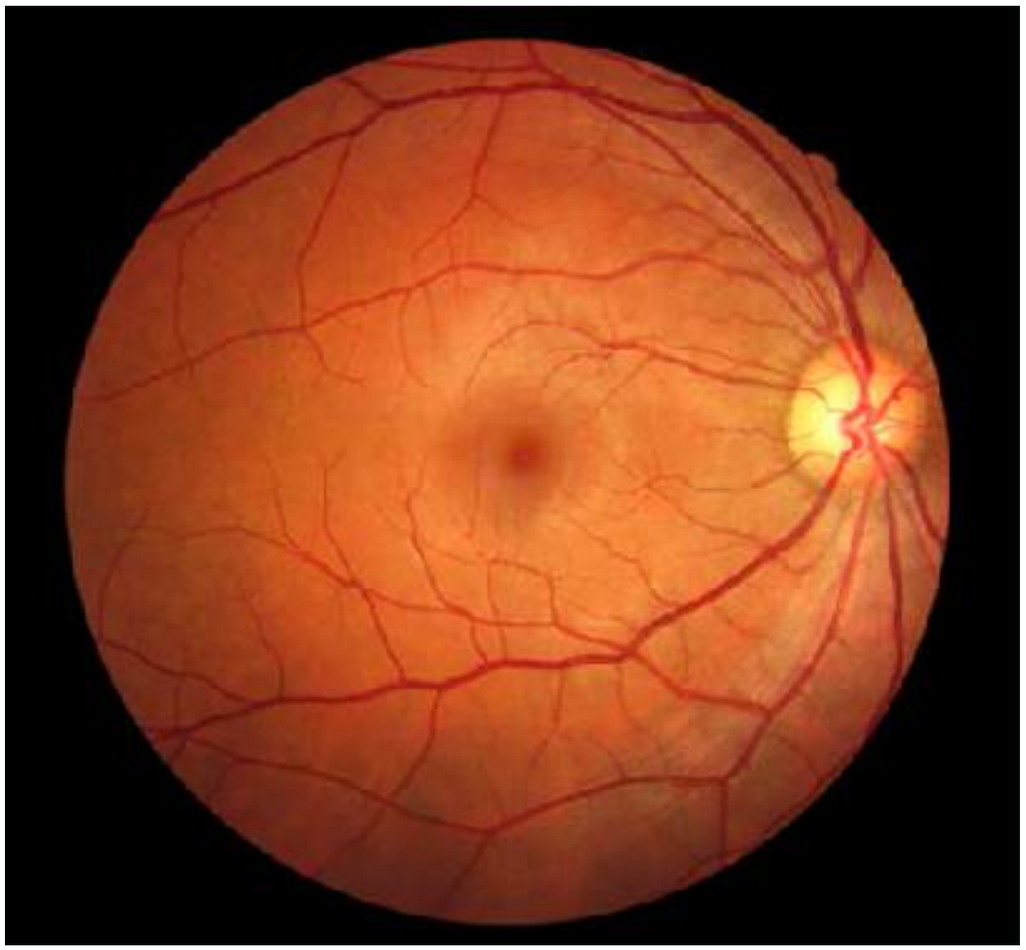


IMAGE OF THE BACK OF THE EYE

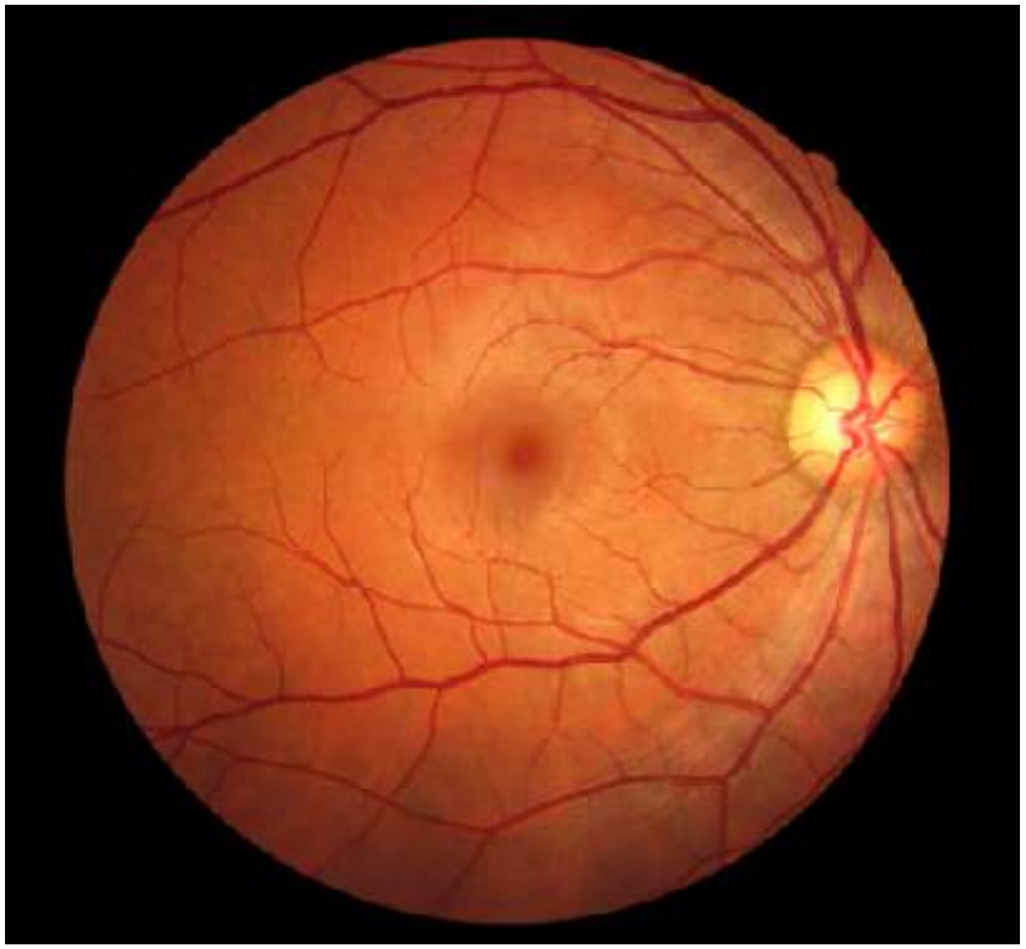


IMAGE OF THE BACK OF THE EYE

DEEP LEARNING CAN IDENTIFY THE PATIENT'S GENDER WITH 95% ACCURACY

VISUALIZING CNNs

[intepreting CNN decisions]

Attribution techniques

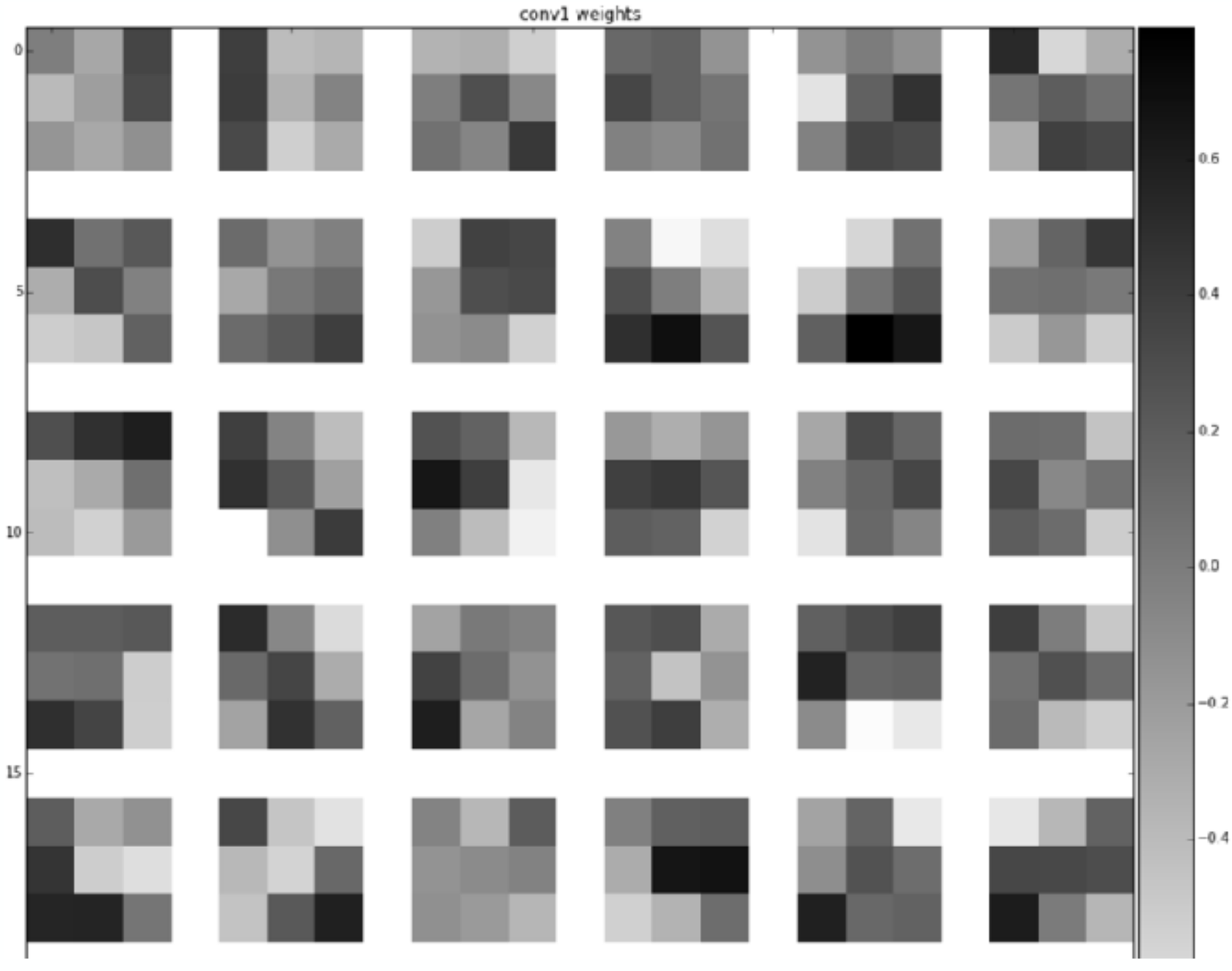
DEEP NETWORKS ARE “BLACK BOXES”?

INTERPRETING THE RESULTS IS
EXTREMELY DIFFICULT

THIS IS TRUE BUT A LOT OF WORK
IS DONE TO UNVEIL THEIR BEHAVIOR

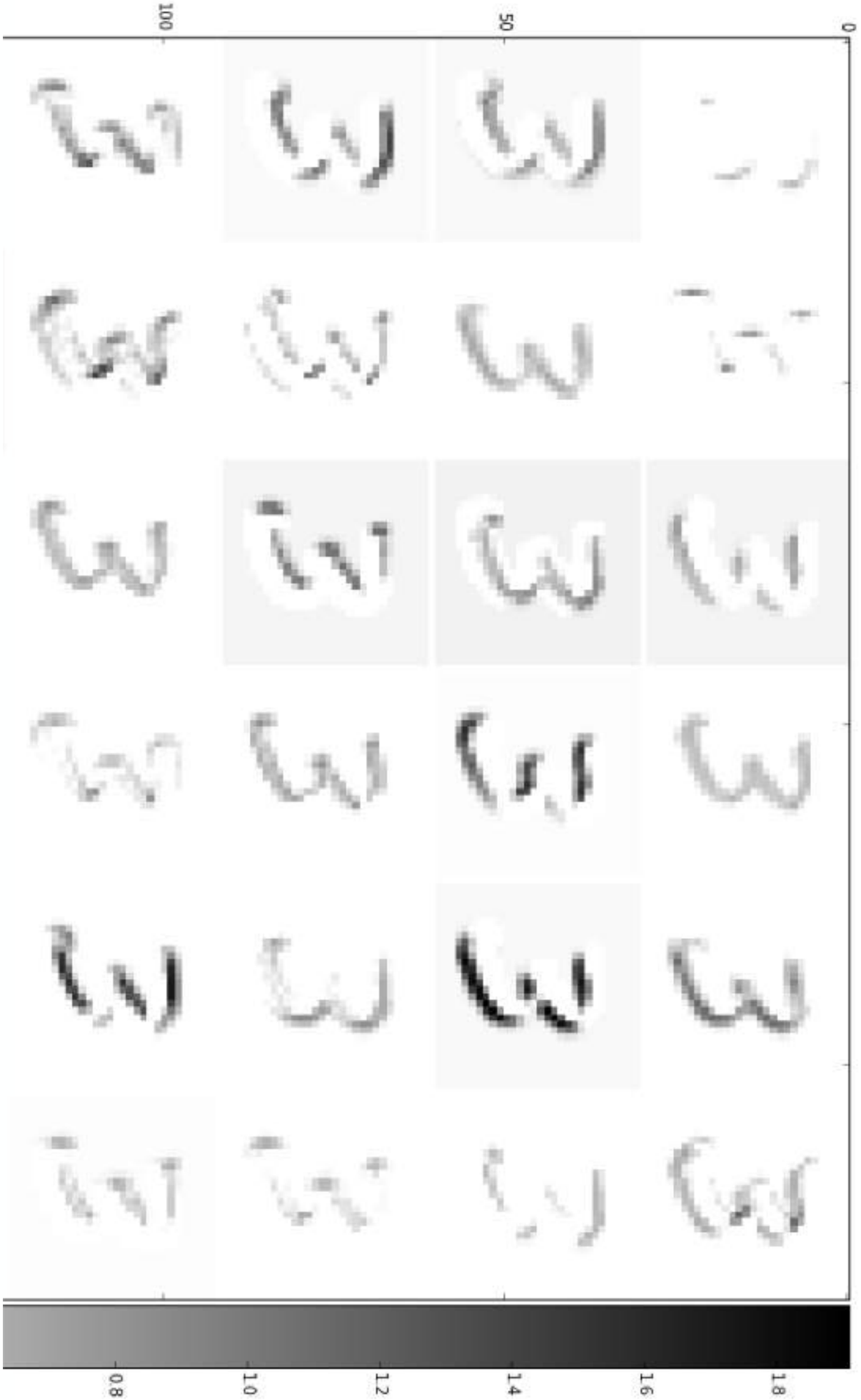
EXPLORING THE FEATURE MAPS

THE SIMPLEST APPROACH IS TO VISUALIZE THE LEARNED WEIGHTS AT INTERMEDIATE LAYERS

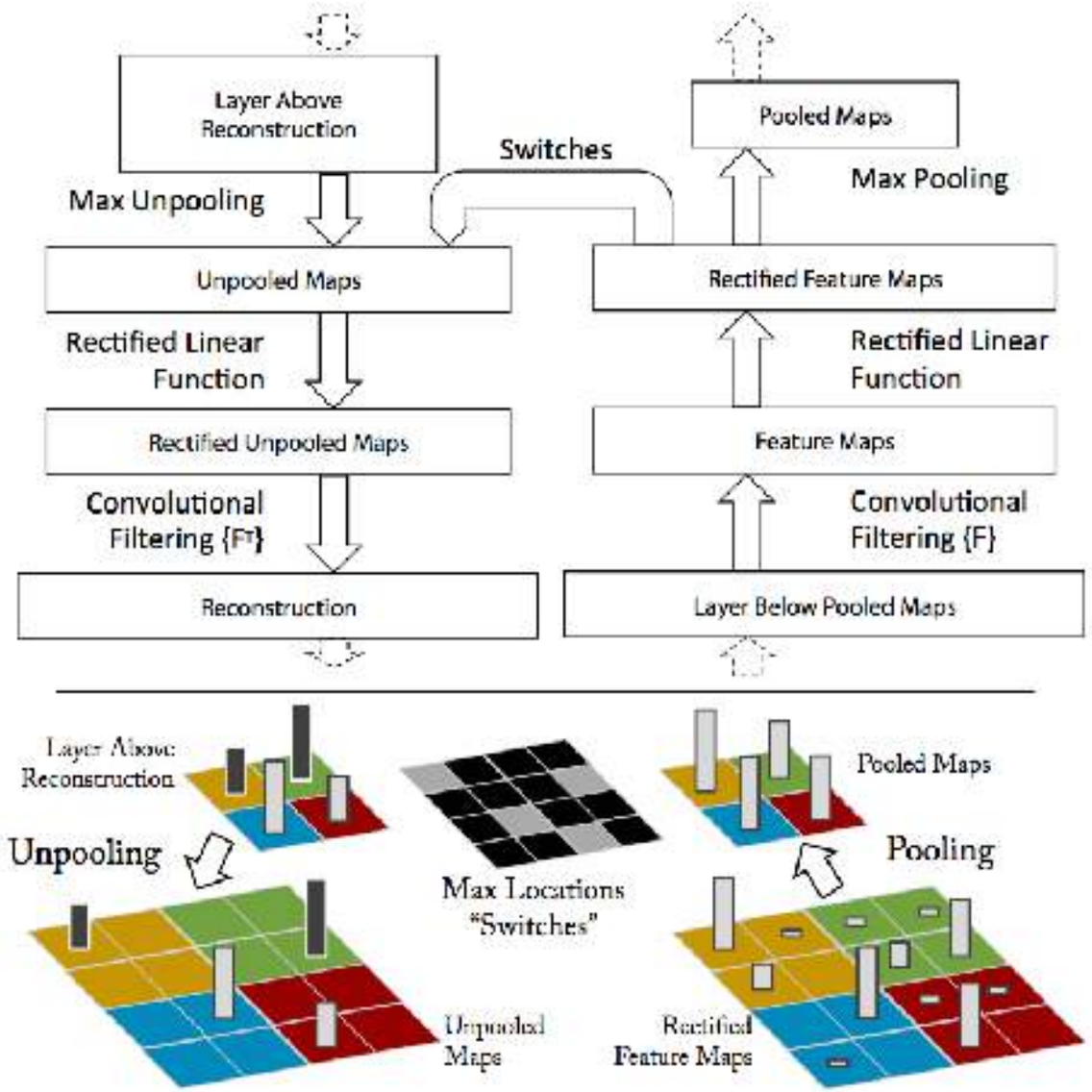


USING THE SAME IDEA, ONE CAN ALSO VISUALIZE THE FEATURE MAPS AT INTERMEDIATE LAYERS

THIS HELPS TRACING THE FEATURES LEARNED BY THE NETWORK

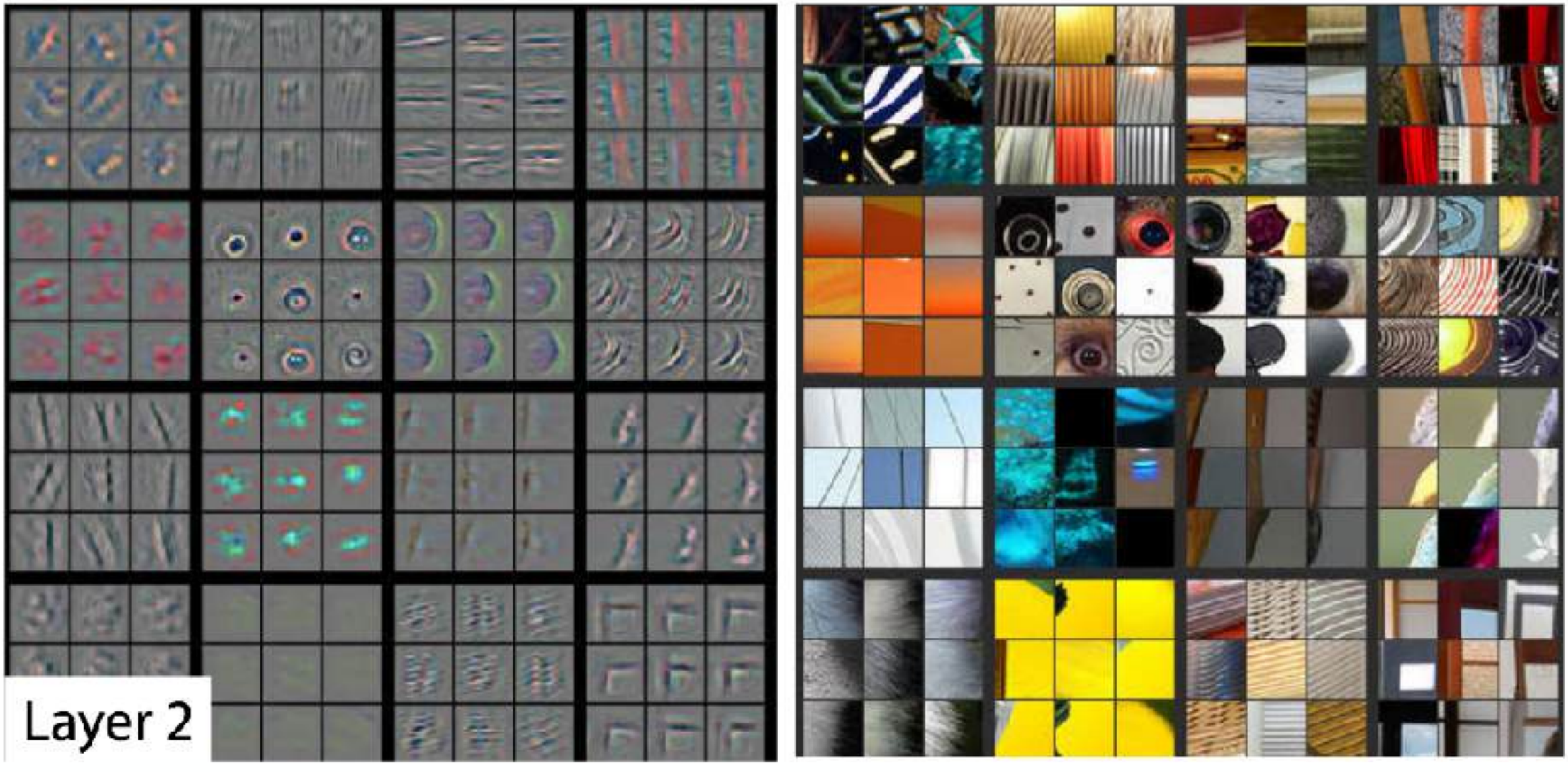

0
50
100
0.8
1.0
1.2
1.4
1.6
1.8

USE “DECONVNETS” TO MAP BACK THE FEATURE MAP INTO THE PIXEL SPACE



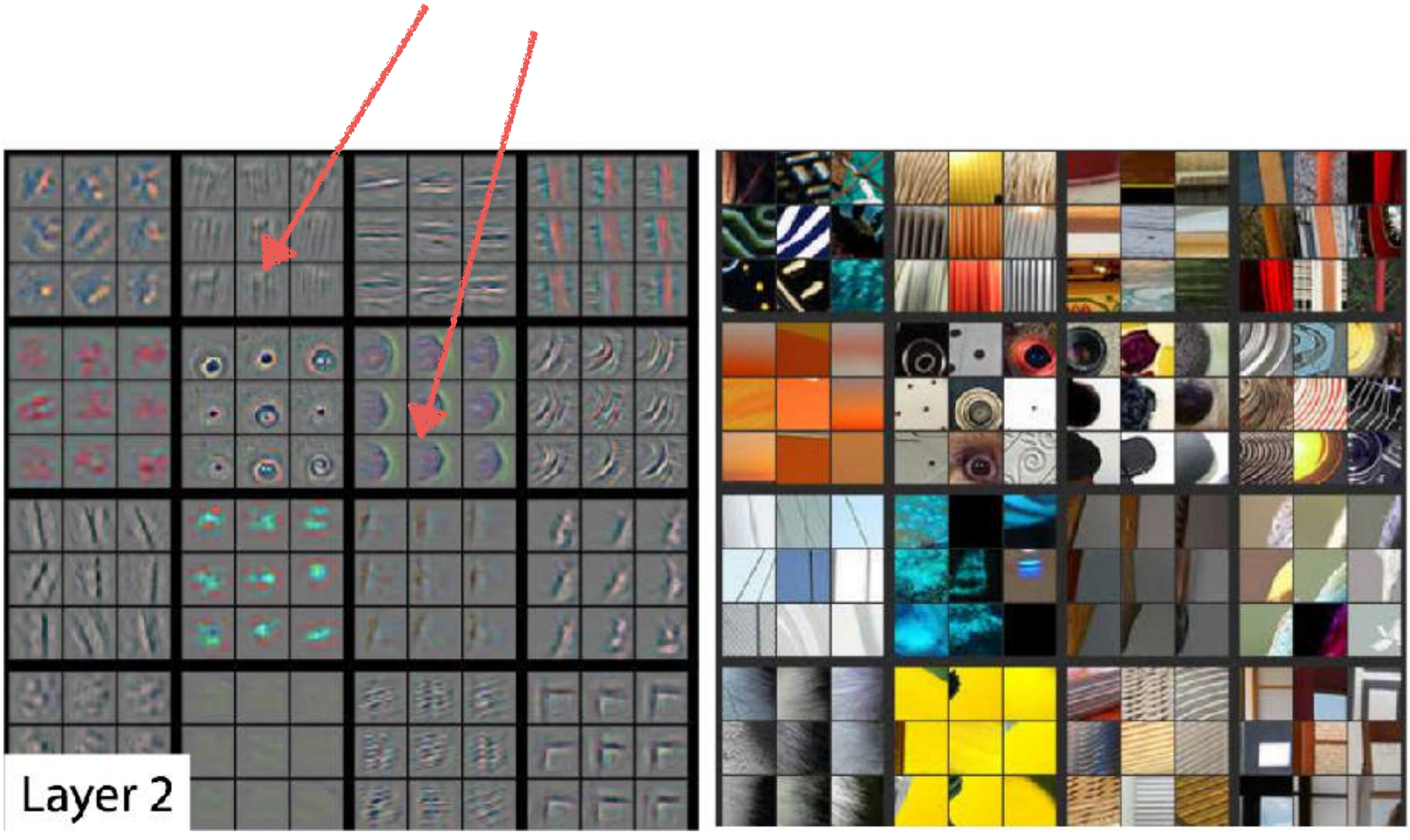
IT ALLOWS TO SEE WHICH REGIONS OF THE INPUT GENERATED A MAXIMUM RESPONSE IN A NEURON

Zeiler+14



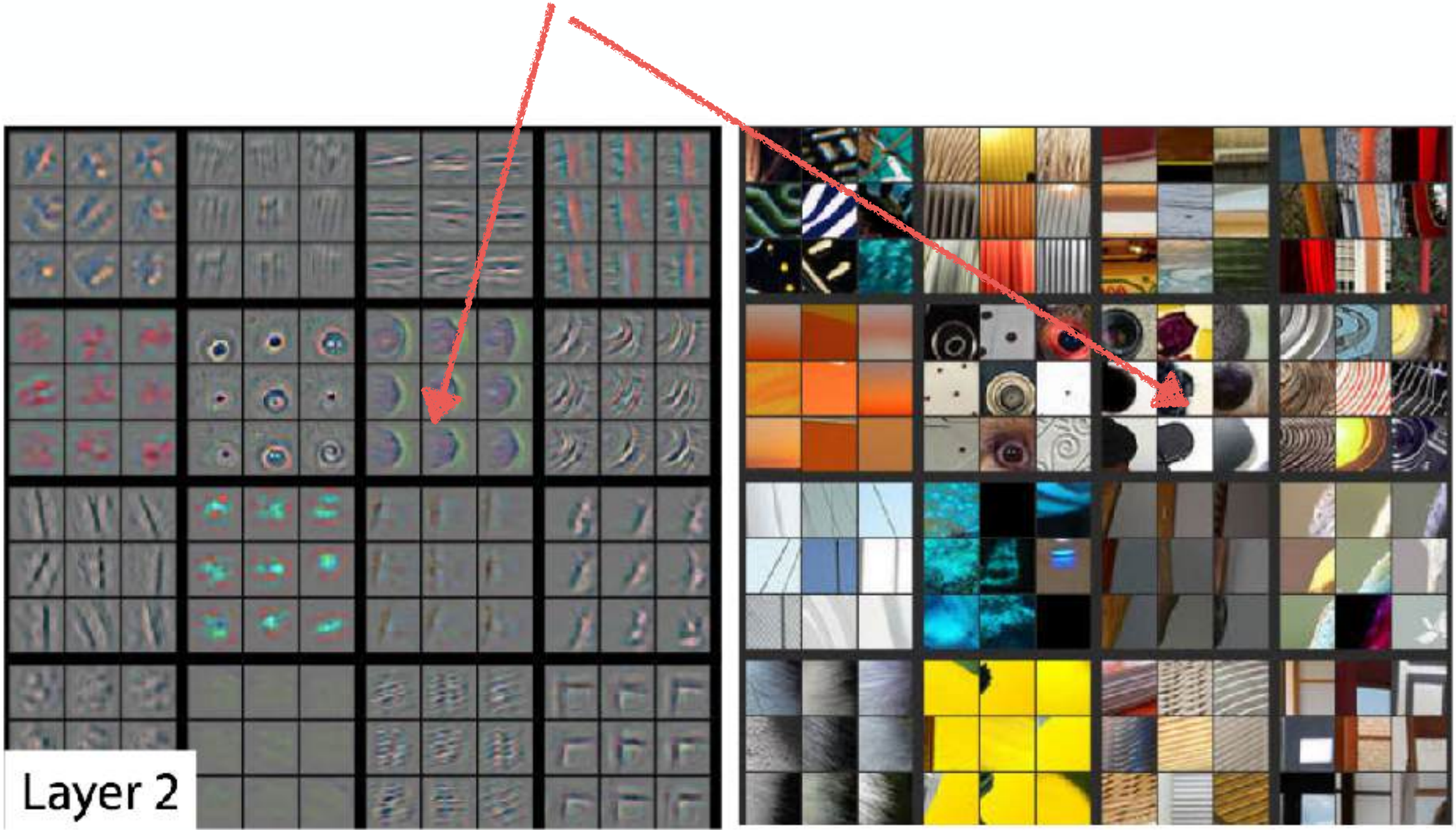
Zeiler+14

EVERY BLOCK OF 9 SHOWS
THE 9 STRONGEST RESPONSES TO A GIVEN FILTER OF LAYER2



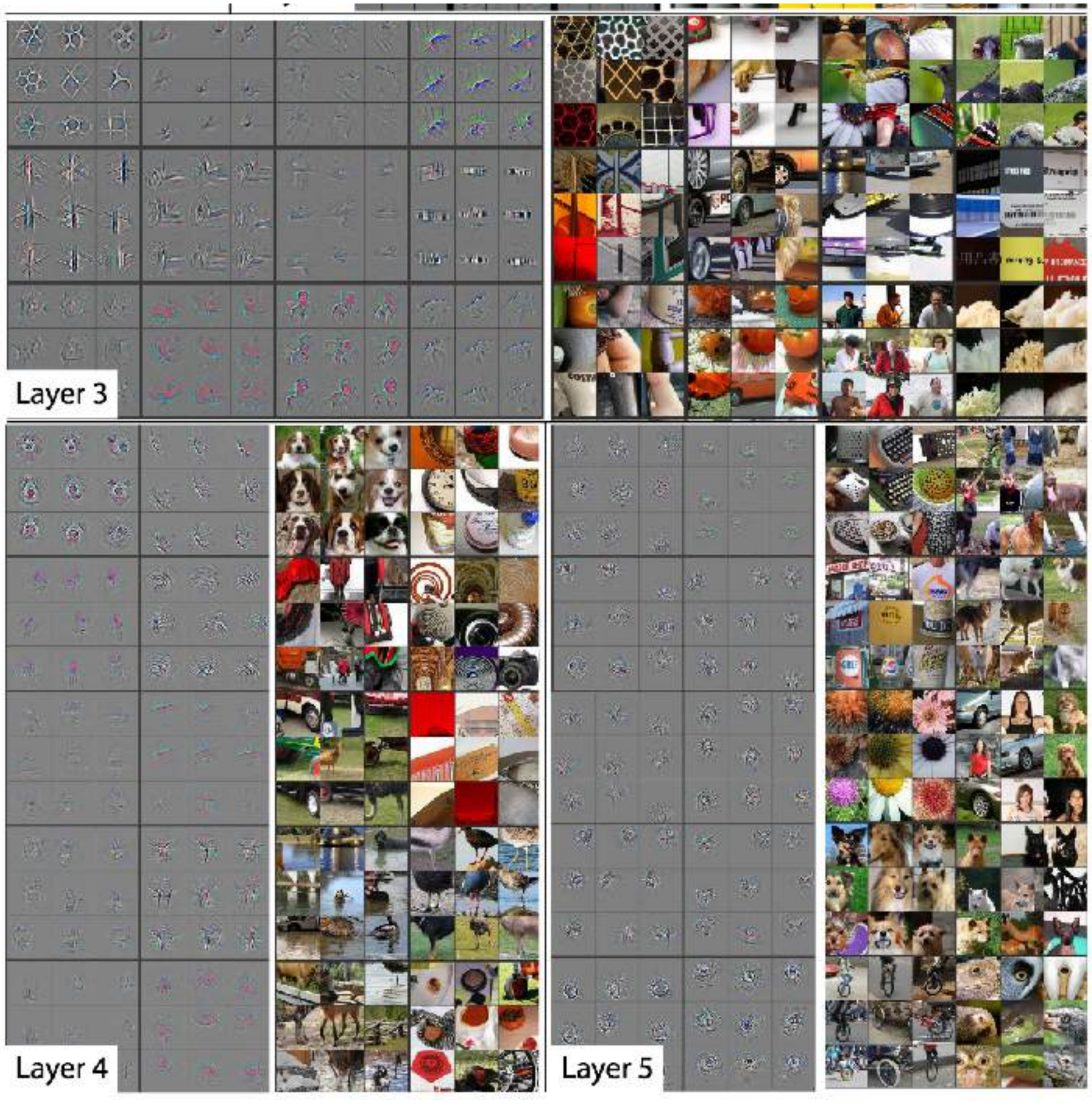
Zeiler+14

THE CORRESPONDING REGIONS OF IMAGES THAT GENERATED THE MAXIMUM RESPONSE



Zeiler+14

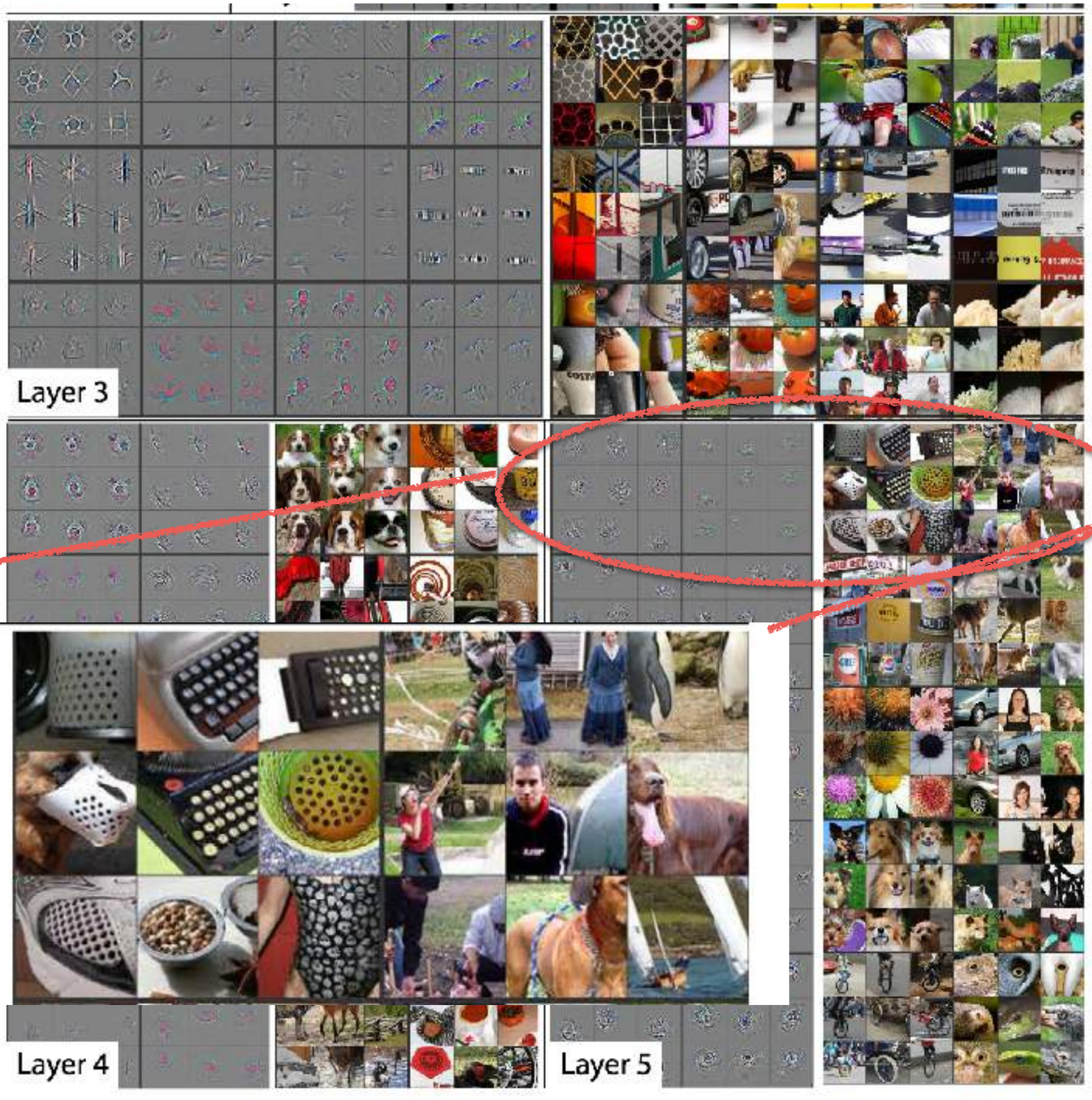
CAN BE REPEATED FOR DEEPER LAYERS ALTHOUGH IT BECOMES LESS INTUITIVE



Zeiler+14

CAN BE REPEATED FOR DEEPER LAYERS ALTHOUGH IT BECOMES LESS

Zeiler+14



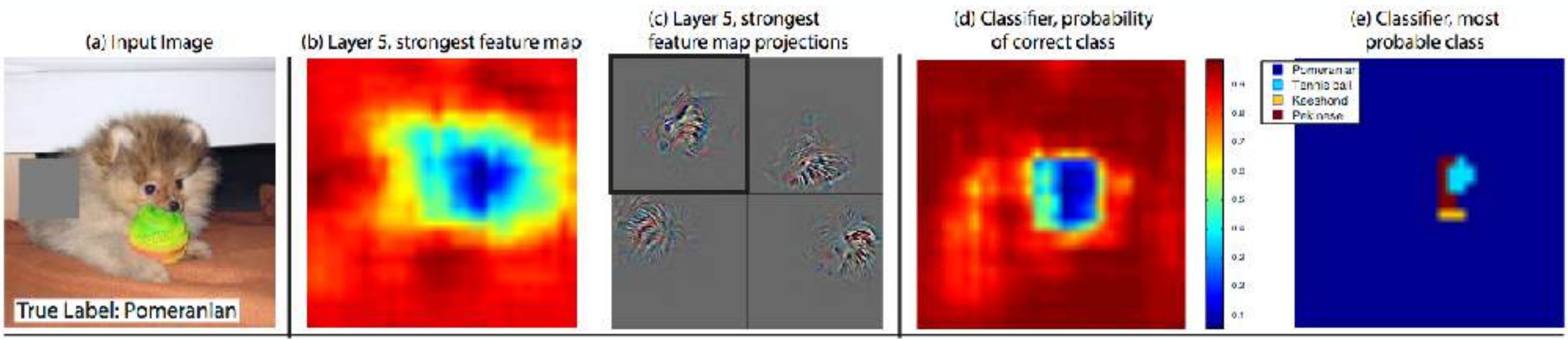
OCCLUSION SENSITIVITY

THE BASIC IDEA IS TO
PERTURB / MODIFY AN INPUT
IMAGE AND SEE THE EFFECT ON
THE PREDICTIONS

OCCLUSION SENSITIVITY TRIES ALSO TO FIND THE REGION OF THE IMAGE THAT TRIGGERED THE NETWORK DECISION BY MASKING DIFFERENT REGIONS OF THE INPUT IMAGE AND ANALYZING THE NETWORK OUPUT

IT ALLOWS TO IF THE NETWORK IS TAKING THE DECISIONS BASED ON THE EXPECTED FEATURES

VERY TIME CONSUMING!



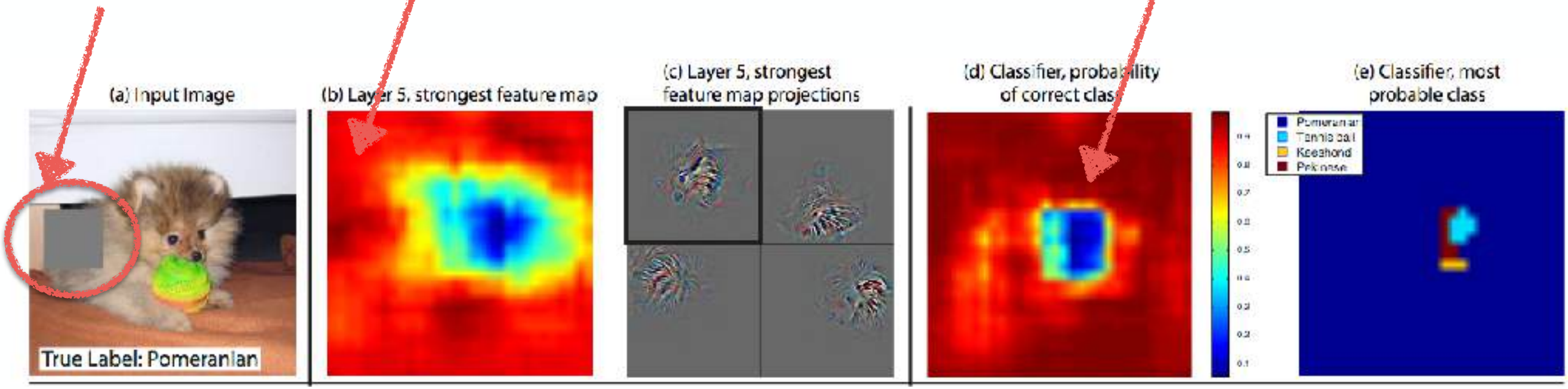
Zeiler+14

OCCLUSION SENSITIVITY TRIES ALSO TO FIND THE REGION OF THE IMAGE THAT TRIGGERED THE NETWORK DECISION BY MASKING DIFFERENT REGIONS OF THE INPUT IMAGE AND ANALYZING THE NETWORK OUPUT

for every position of the square the maximum response of a given layer is averaged

the output probability as a function of the occluding square position

occluded region



Zeiler+14

“INCEPTIONISM” TECHNIQUES

THE IDEA BEHIND INCEPTIONISM TECHNIQUES
IS TO INVERT THE NETWORK TO GENERATE AN IMAGE
THAT MAXIMIZES THE OUTPUT SCORE

image I

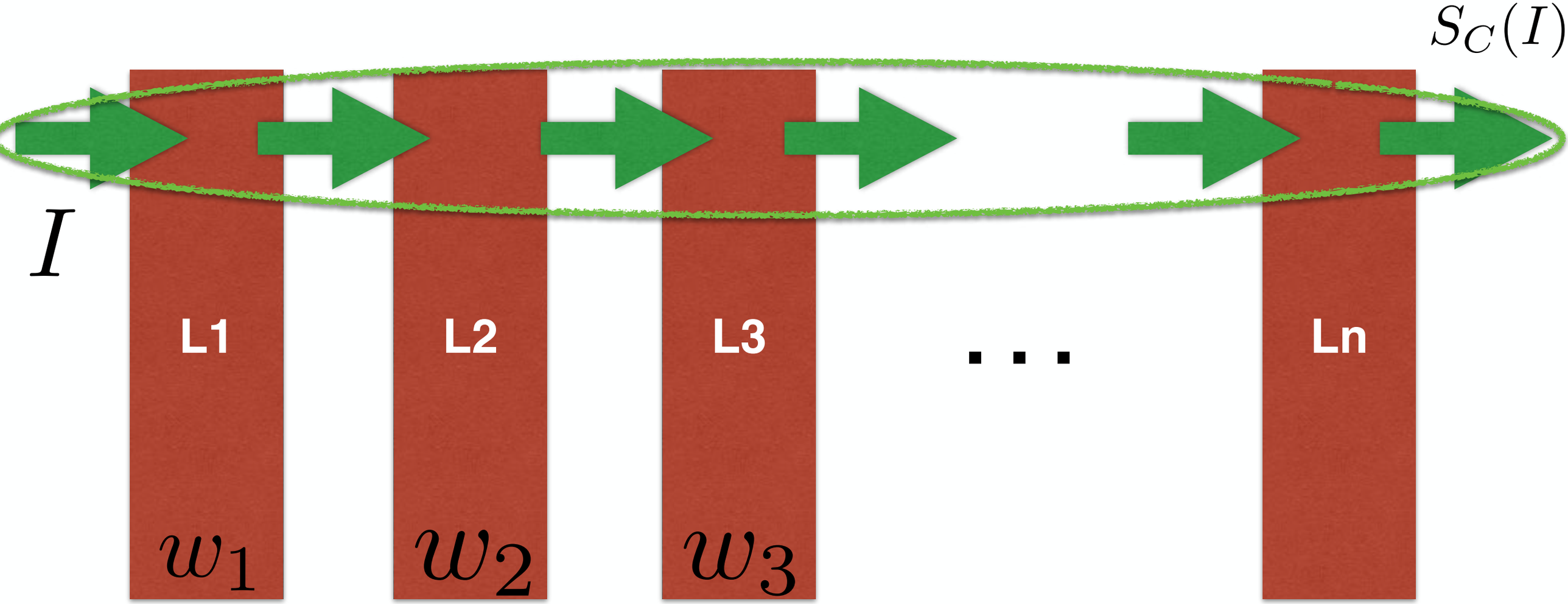
Score of class c for image I

$$arg\,\max_{\mathrm{I}} S_c(I) - \lambda ||I||_2^2$$

TRY TO FIND AN IMAGE THAT GENERATES A
HIGH SCORE FOR A GIVEN CLASS

Simonyan+14

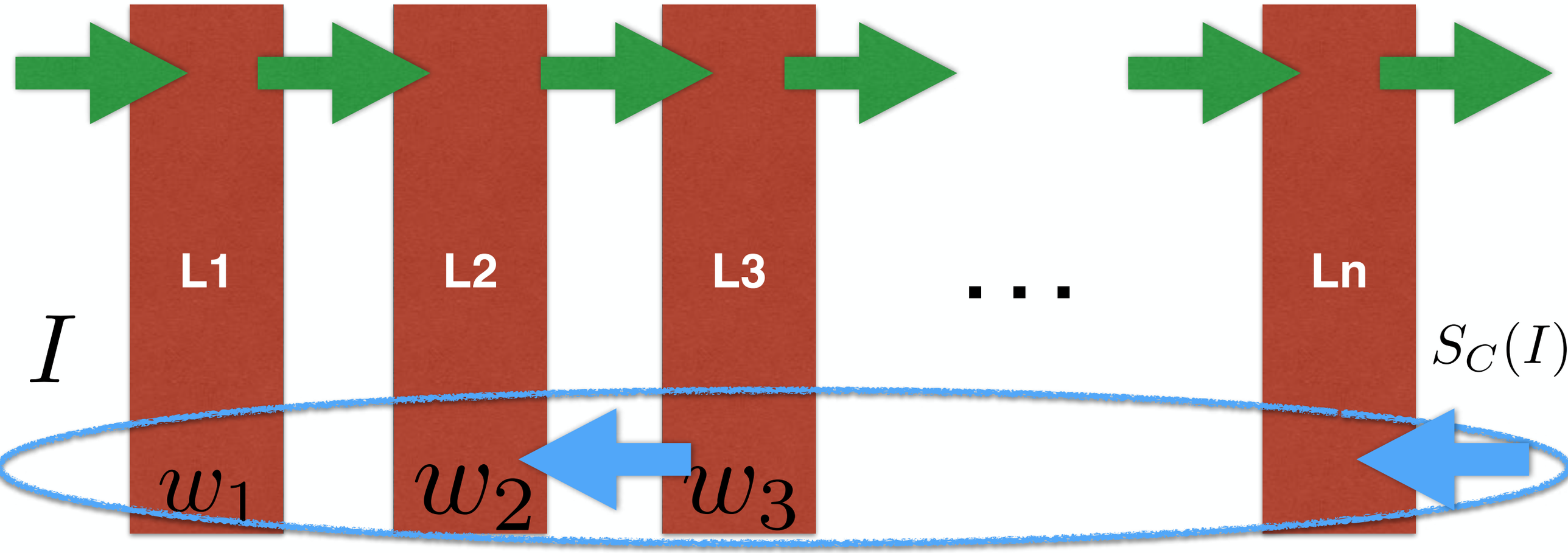
INCEPTIONISM - DEEP DREAM



DURING THE TRAINING PHASE THE WEIGHTS ARE LEARNED TO MAP I INTO Sc

Simonyan+14

INCEPTIONISM - DEEP DREAM

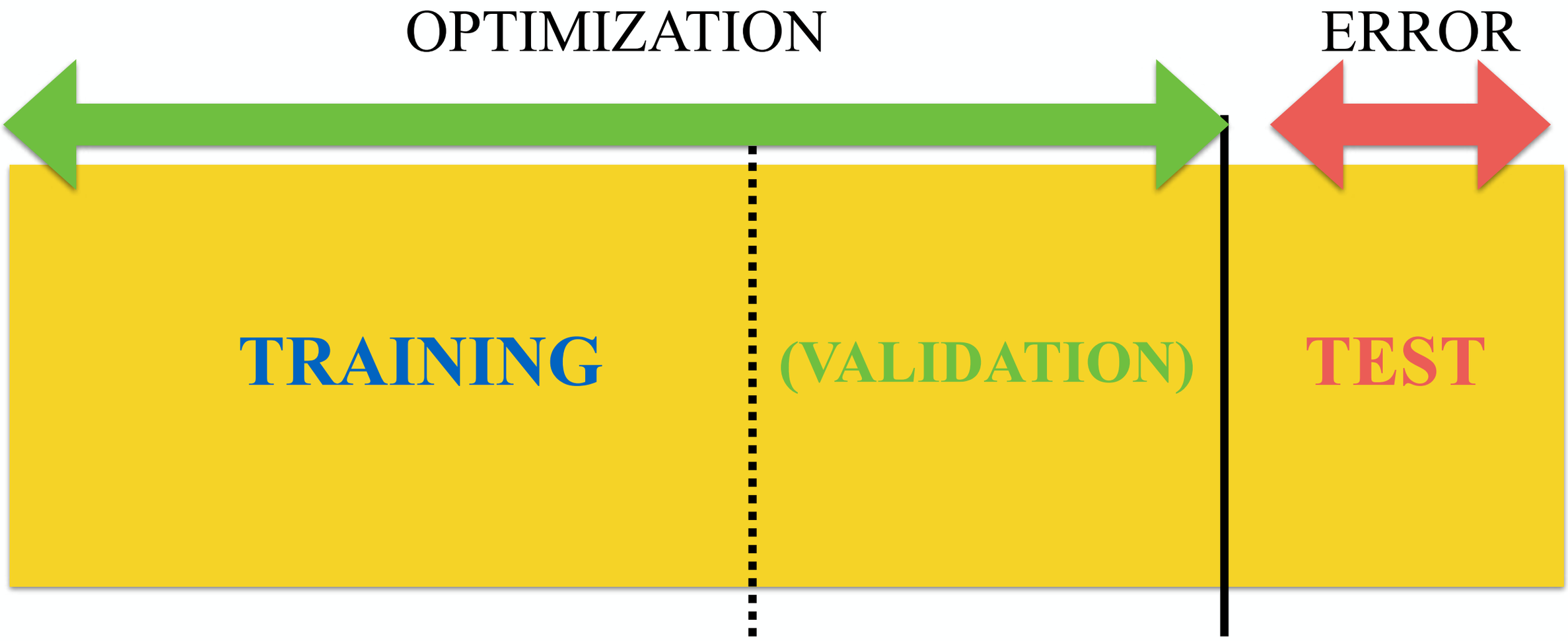


DURING THE RECONSTRUCTION PHASE, I IS LEARNT TRHOUG BACKPROPAGATION KEEPING THE WEIGHTS FIXED

Simonyan+14

Training deeper networks

IN PRACTICE

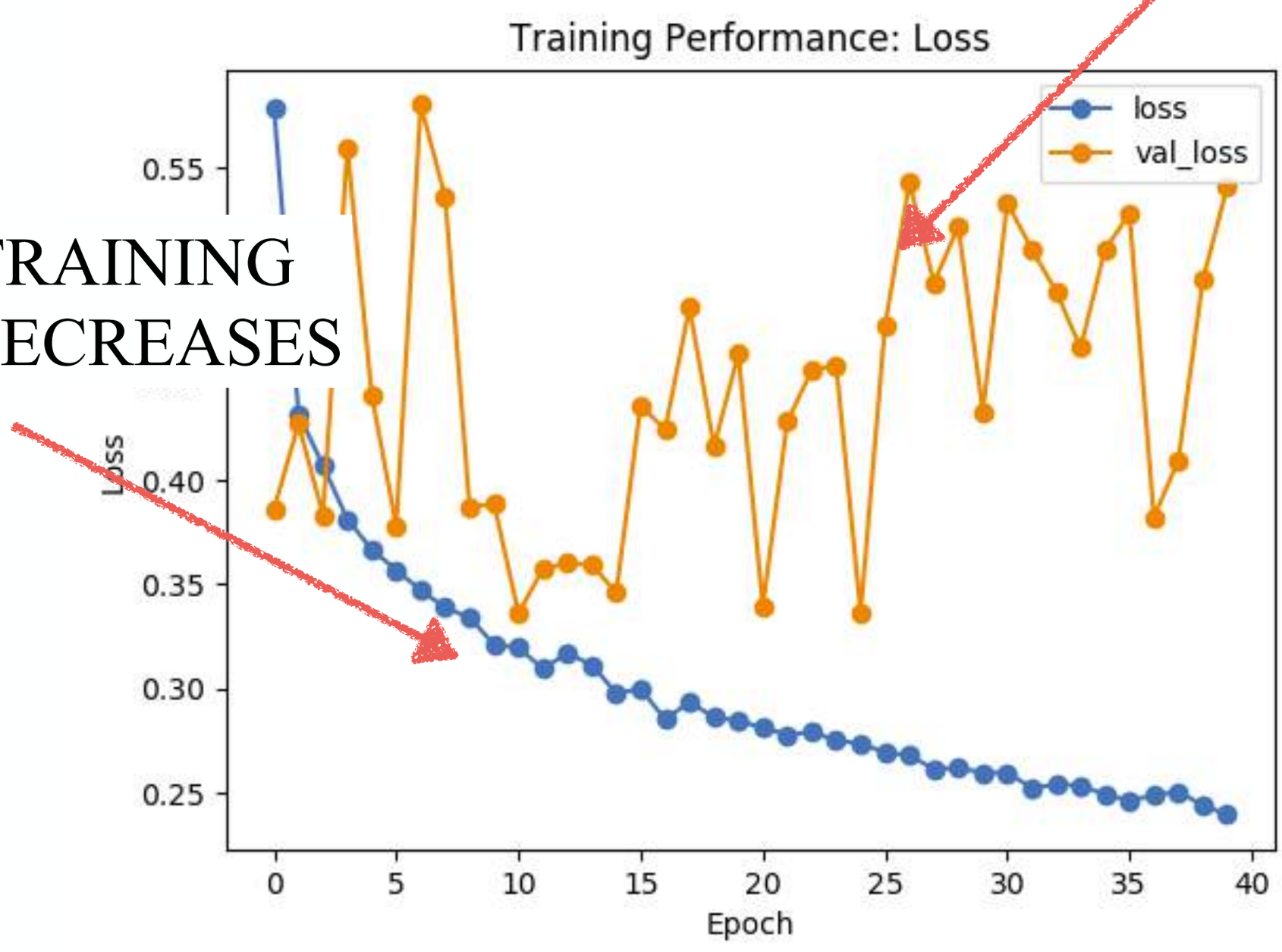


training set: use to train the classifier
validation set: use to monitor performance in real time - check for overfitting
test set: use to train the classifier

OVER-FITTING

THE TEST STAYS CONSTANT
OR INCREASES

THE TRAINING
LOSS DECREASES

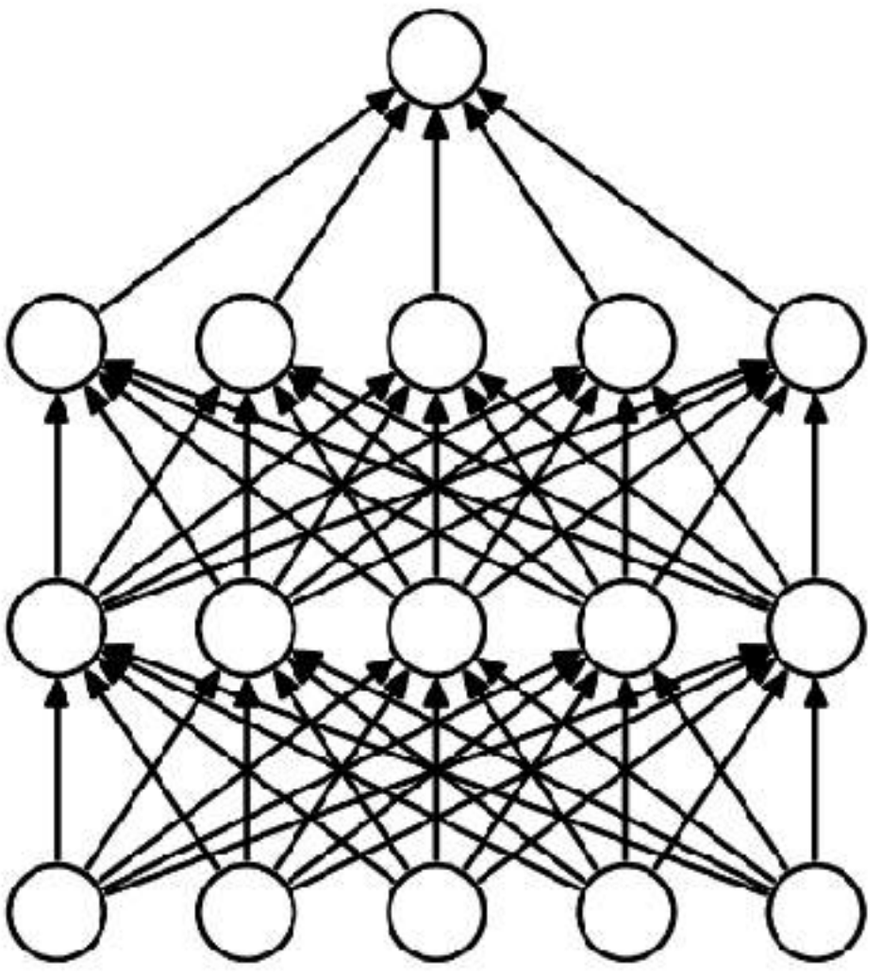


DROPOUT

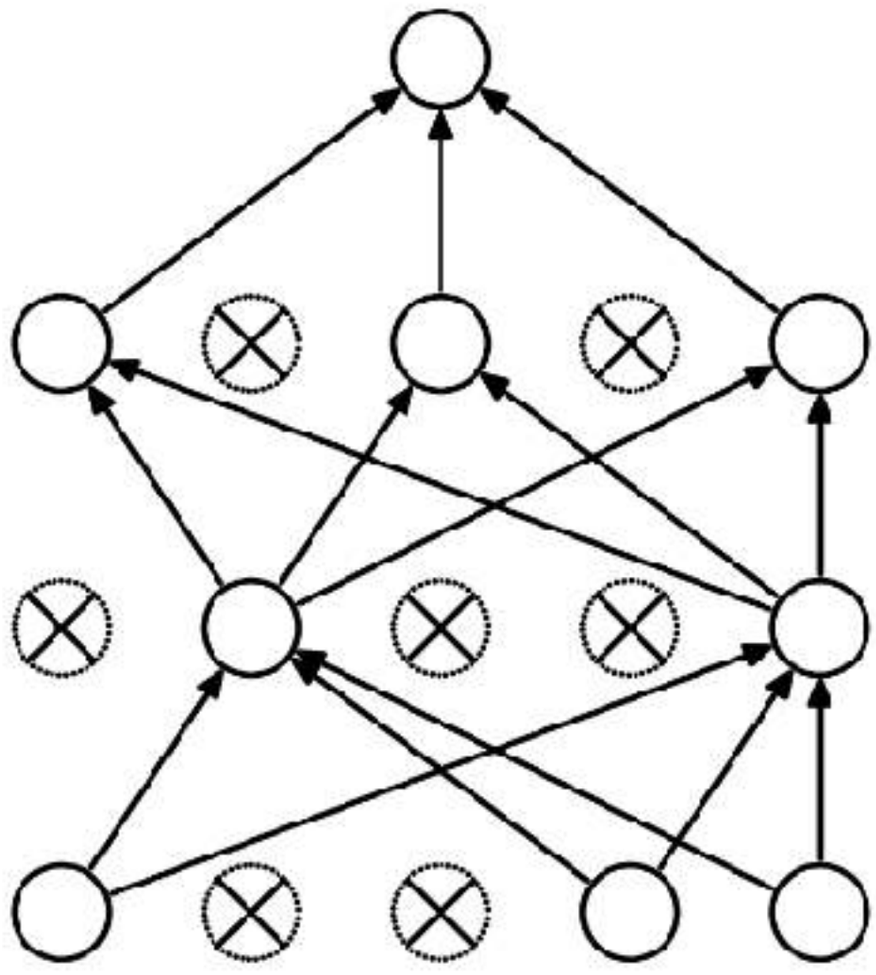
[Hinton+12]

- THE IDEA IS TO REMOVE NEURONS RANDOMLY DURING THE TRAINING

- ALL NEURONS ARE PUT BACK DURING THE TEST PHASE



(a) Standard Neural Net



(b) After applying dropout.

DROPOUT

WHY DOES IT WORK?

1. SINCE NEURONS ARE REMOVED RANDOMLY, IT AVOIDS CO-ADAPTATION AMONG THEMSELVES

2. DIFFERENT SETS OF NEURONS WHICH ARE SWITCHED OFF, REPRESENT A DIFFERENT ARCHITECTURE AND ALL THESE DIFFERENT ARCHITECTURES ARE TRAINED IN PARALLEL. FOR N NEURONS ATTACHED TO DROPOUT, THE NUMBER OF SUBSET ARCHITECTURES FORMED IS 2^N. SO IT AMOUNTS TO PREDICTION BEING AVERAGED OVER THESE ENSEMBLES OF MODELS.

VANISHING / EXPLODING GRADIENT PROBLEM

<u>REMEMBER THAT:</u>

$$y_{i+1} = \sigma[\sum w_i y_i]$$

output layer i+1

activation function

weights

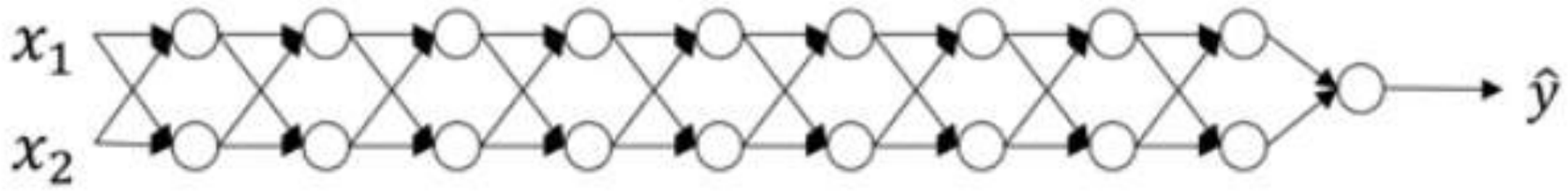
output layer i

VANISHING / EXPLODING GRADIENT PROBLEM

<u>WITH MANY LAYERS:</u>

$$y_n = \sigma\Big(...\sigma\Big(...\sigma\Big(\sum w_0 x\Big)\Big)\Big)$$

VANISHING/EXPLODING GRADIENT PROBLEM



$$w_i = \begin{pmatrix} w_i^0 & 0 \\ 0 & w_i^1 \end{pmatrix} \qquad \hat{y} = x \prod_n w_i$$

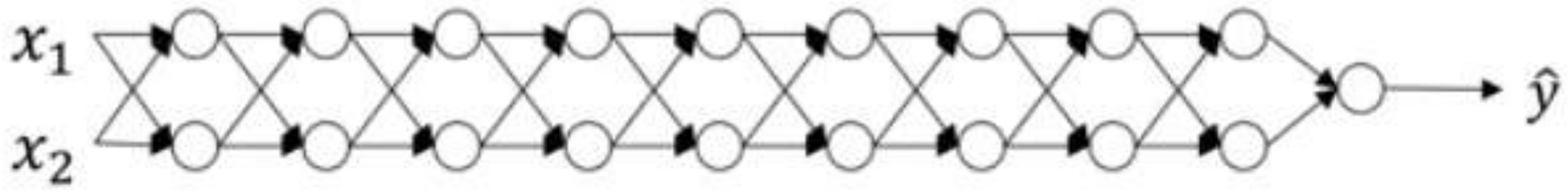
$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

IF WEIGHTS ARE ALL INITIALIZED TO VALUES <<1:

$$\hat{y} \to 0$$

VANISHING GRADIENT

VANISHING/EXPLODING GRADIENT PROBLEM



$$w_i = \begin{pmatrix} w_i^0 & 0 \\ 0 & w_i^1 \end{pmatrix} \qquad \hat{y} = x \prod_n w_i$$

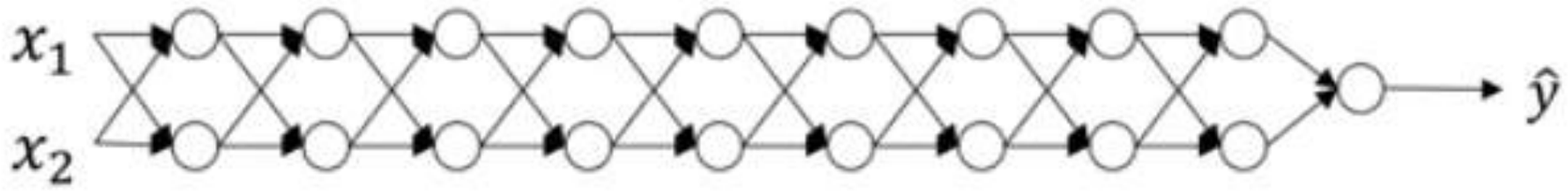
$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

IF WEIGHTS ARE ALL INITIALIZED TO VALUES >1:

$$\hat{y} \to \infty$$

EXPLODING GRADIENT

VANISHING/EXPLODING GRADIENT PROBLEM



$$w_i = \begin{pmatrix} w_i^0 & 0 \\ 0 & w_i^1 \end{pmatrix} \qquad \hat{y} = x \prod_n w_i$$

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

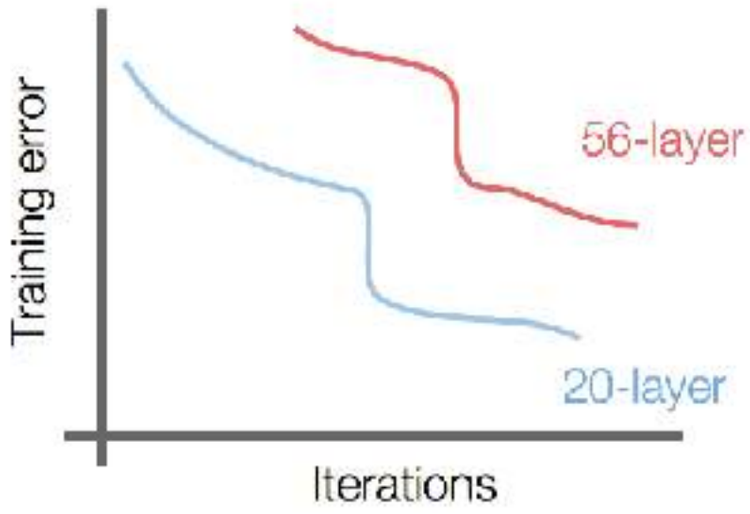
IF WEIGHTS ARE ALL INITIALIZED TO VALUES >1:

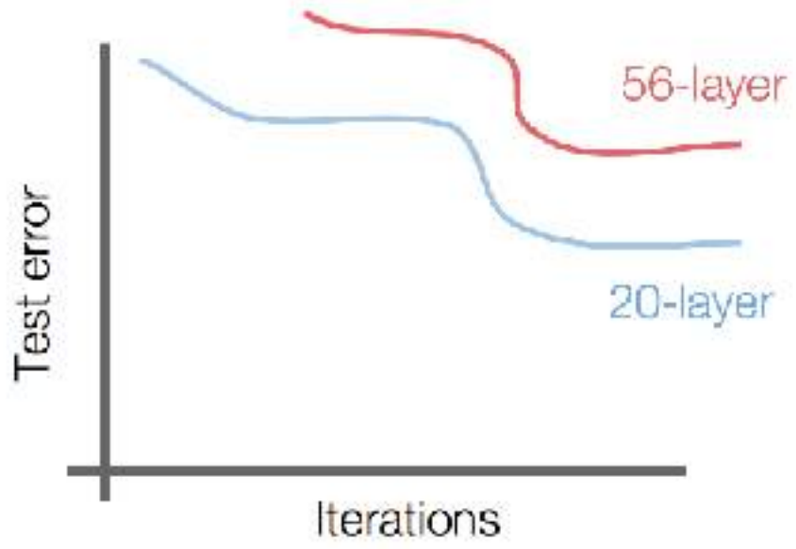
$$w_i^L \rightarrow \infty$$

EXPLODING GRADIENT

VANISHING/EXPLODING GRADIENT PROBLEM

TRAINING BECOMES UNSTABLE
VERY SLOW OR NO CONVERGENCE





BATCH NORMALIZATION

[SZEGEDY+15]

A SOLUTION TO KEEP REASONABLE VALUES OF THE ACTIVATIONS IN DEEP NETWORKS

BATCH NORMALIZATION PREVENTS LOW OR LARGE VALUES BY RE-NORMALIZING THE VALUES BEFORE ACTIVATION FOR EVERY BATCH

$$\hat{y}_i = \gamma \frac{y_i - E(y_i)}{\sigma(y_i)} + \beta$$

INPUT

NORMALIZED INPUT

SCATTER

BATCH NORMALIZATION

[SZEGEDY+15]

BATCH NORMALIZATION SPEEDS UP AND STABILIZES TRAINING

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

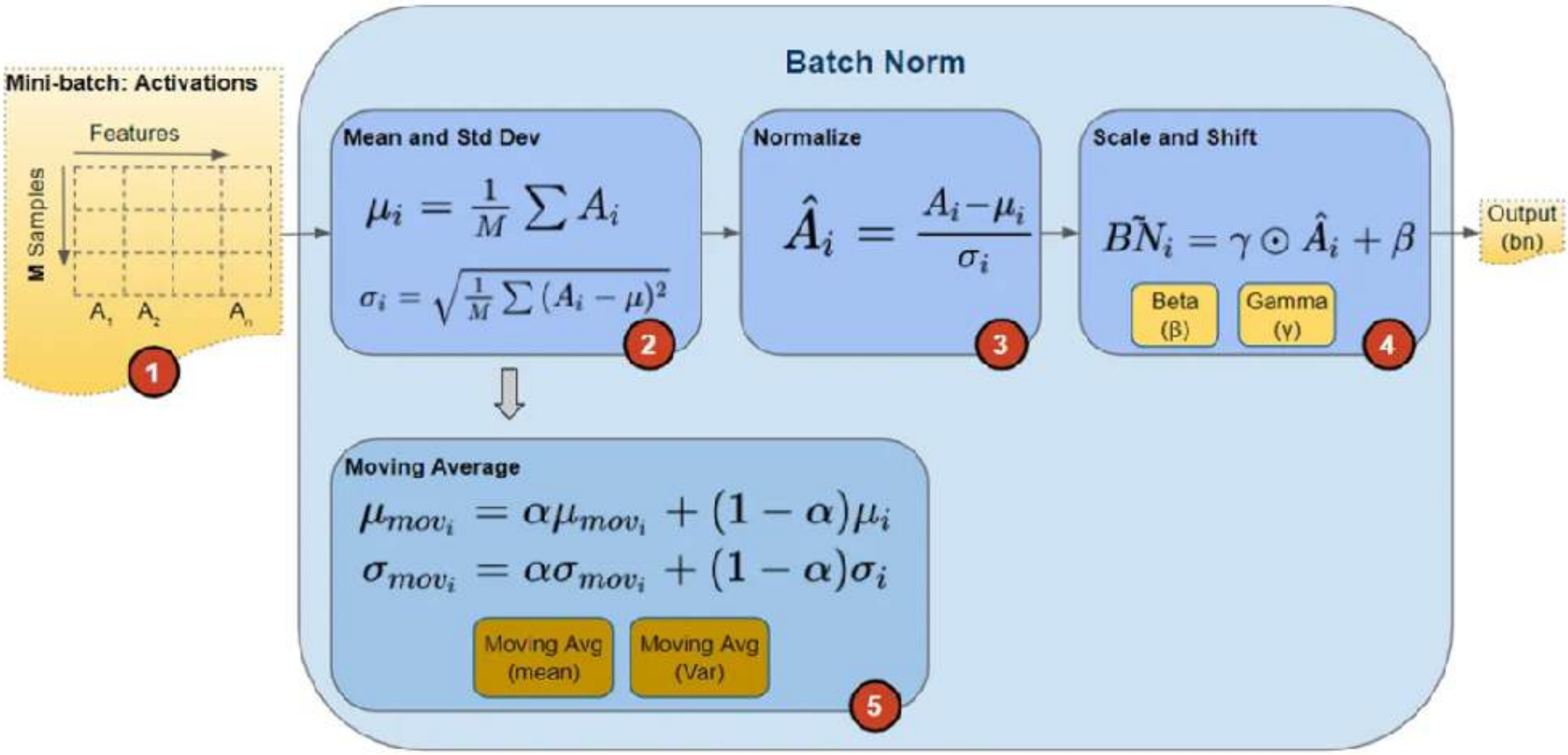
$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

BATCH NORMALIZATION

[SZEGEDY+15]

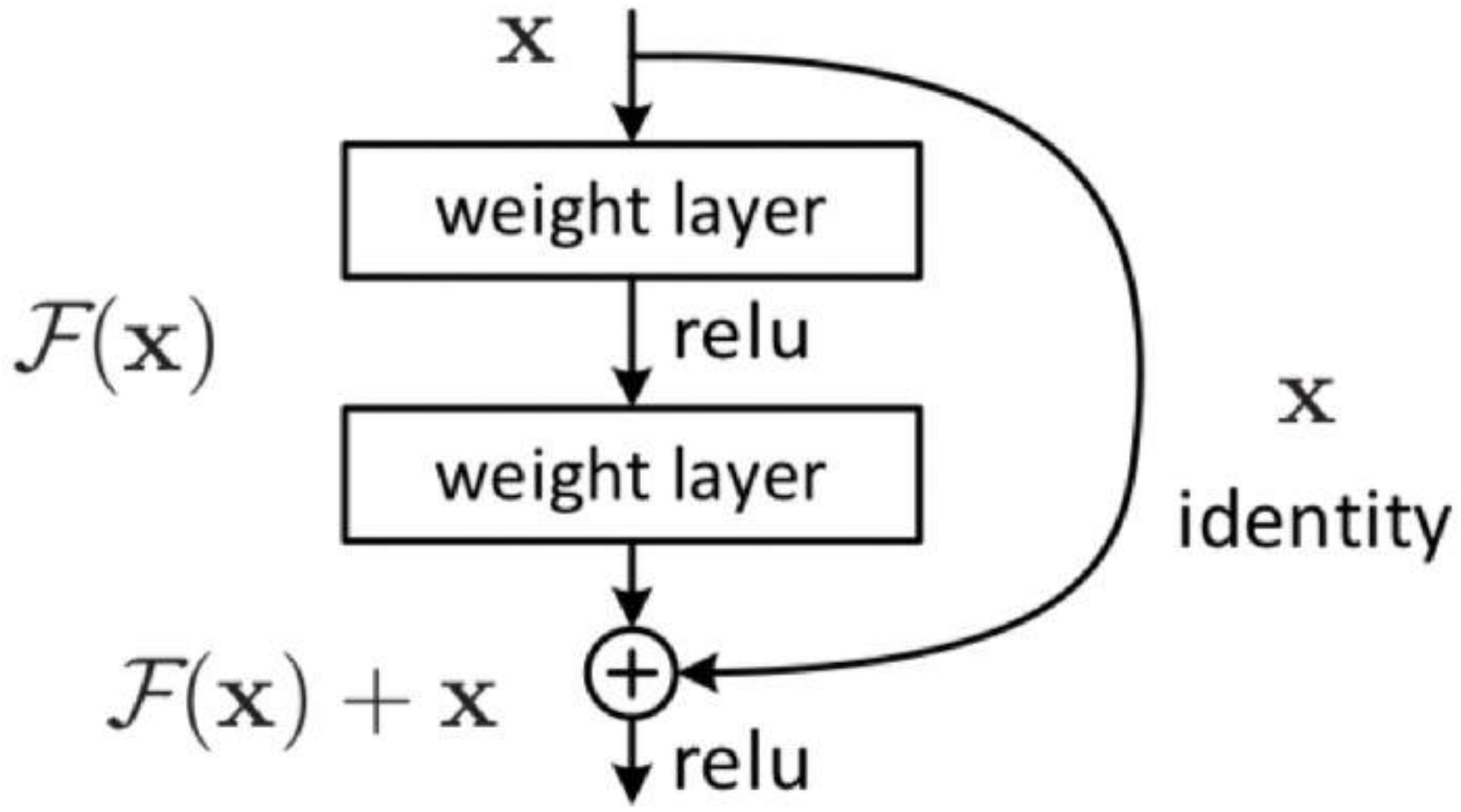


BATCH NORMALIZATION

[SZEGEDY+15]

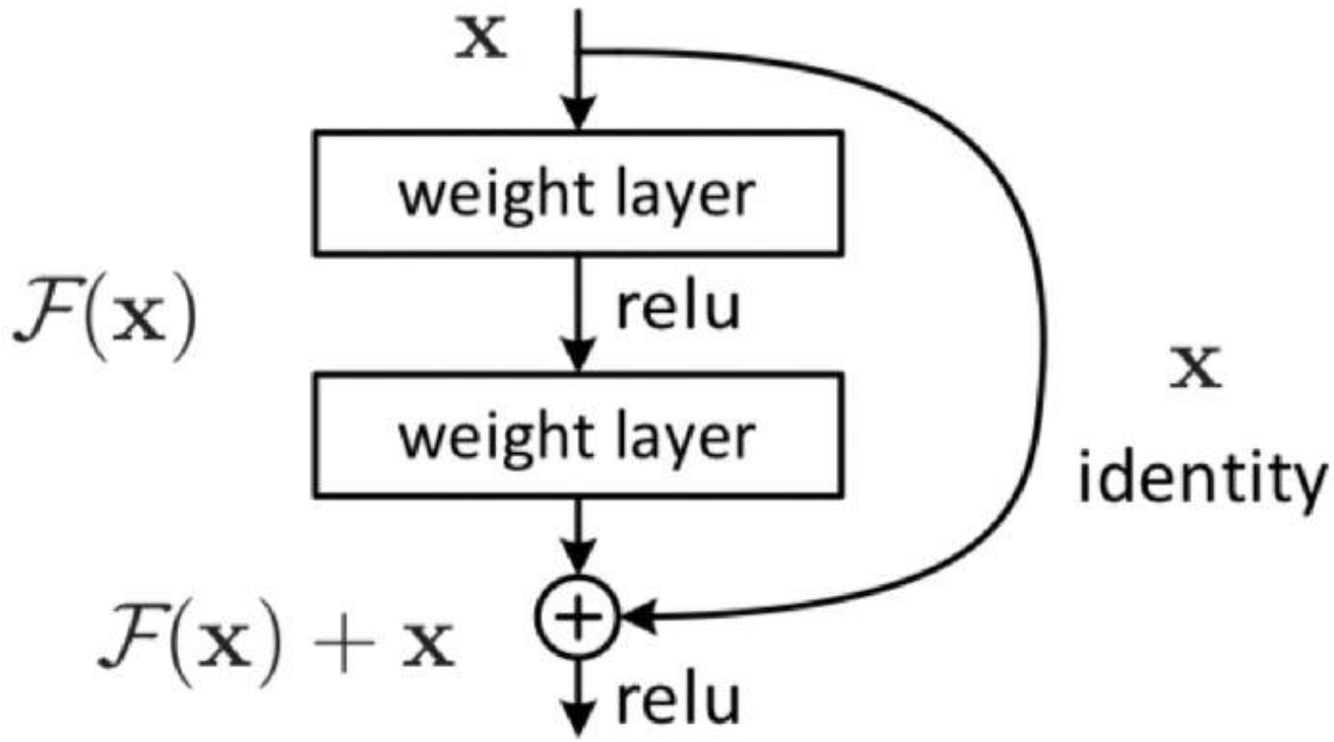
- Speeds up training
- Reduces internal covariate shift of the network
- Regularizes the network, prevents over fitting

RESIDUAL NETWORKS



Kaiming et al. 2015

RESIDUAL NETWORKS



- Adding additional / new layers would not hurt the model's performance as regularisation will skip over them if those layers were not useful.

- If the additional / new layers were useful, even with the presence of regularisation, the weights or kernels of the layers will be non-zero and model performance could increase slightly.

Kaiming et al. 2015

Dealing with a lack of labelled data

Transfer learning

THE CONVOLUTIONAL PART OF A CNN IS
A FEATURE EXTRACTOR ….

Transfer learning

THE CONVOLUTIONAL PART OF A CNN IS
A FEATURE EXTRACTOR

IN THAT RESPECT, THEY ARE VERY FLEXIBLE …

Transfer Learning)

EVEN IF OUR TRAINING SET IS NOT SO LARGE …

WE CAN USE A CNN PRE-TRAINED ON A LARGER SAMPLE

DEPENDING ON HOW SIMILAR BOTH DATASETS ARE, WE CAN:

- RECYCLE THE SAME FEATURES
- FINE-TUNING THE WEIGHTS

DATA FROM NEW SURVEY

How robust to different datasets? Do we always need a big training set?

DEEP-LEARNING BASED MACHINE

"Improved" Galaxy ZOO like classifications for for the entire sample

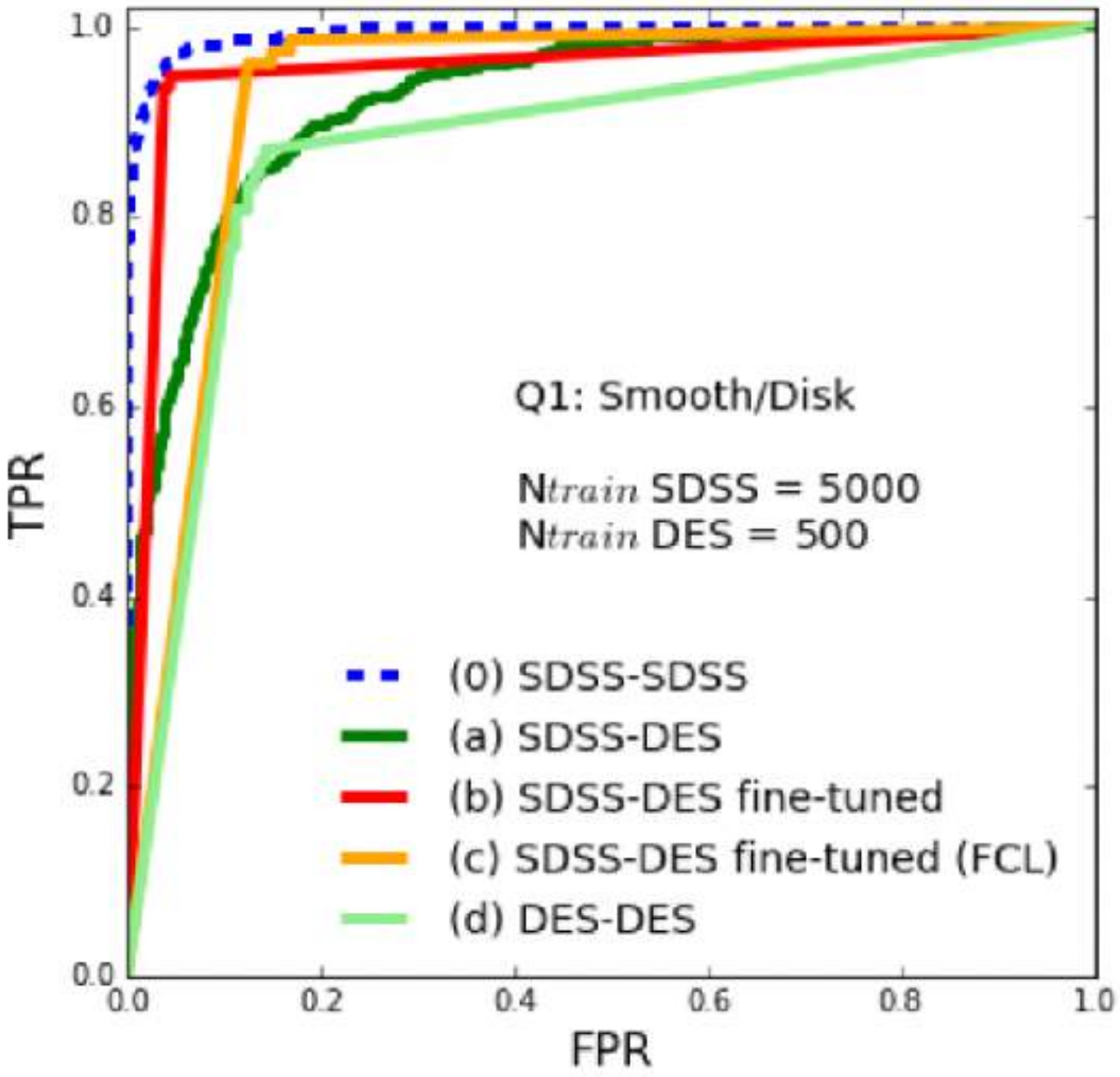
Transfer knowledge?

Human classifications from existing survey

Dominguez-Sanchez+18b

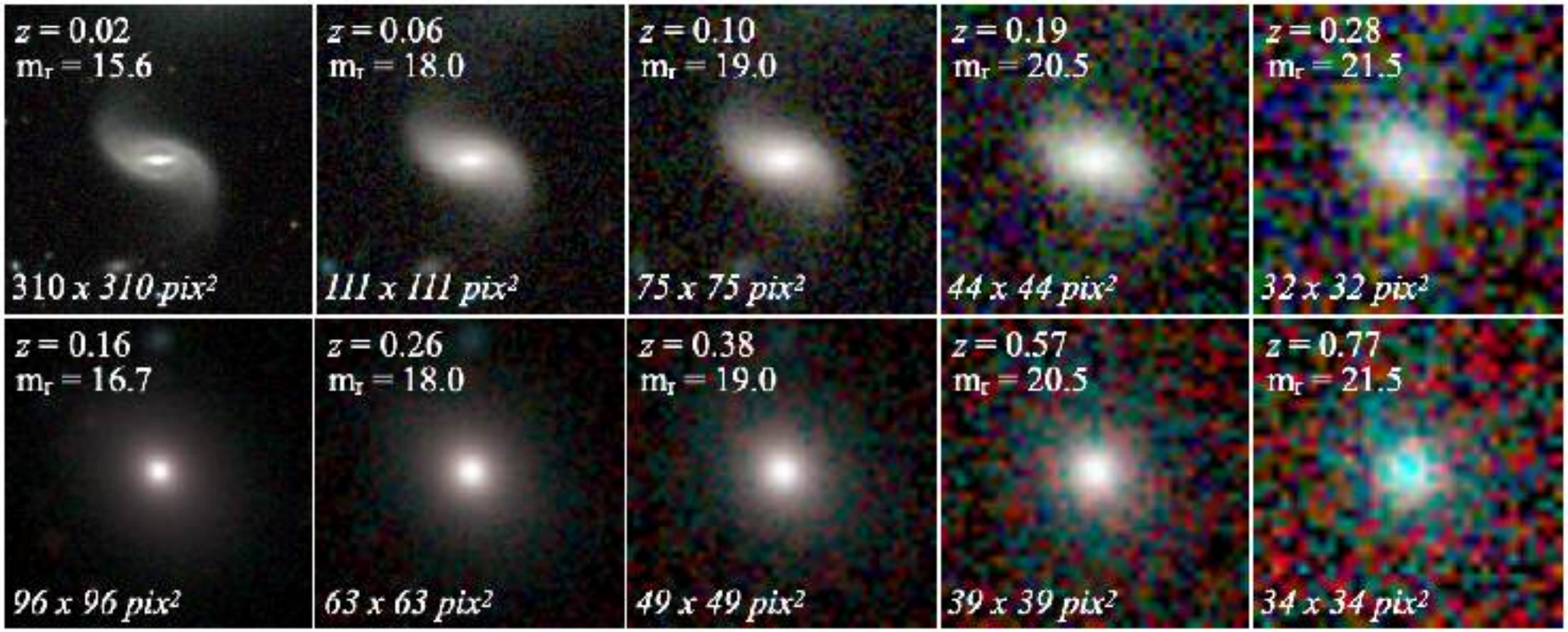
SDSS **DES**

Dominguez-Sanchez+18b

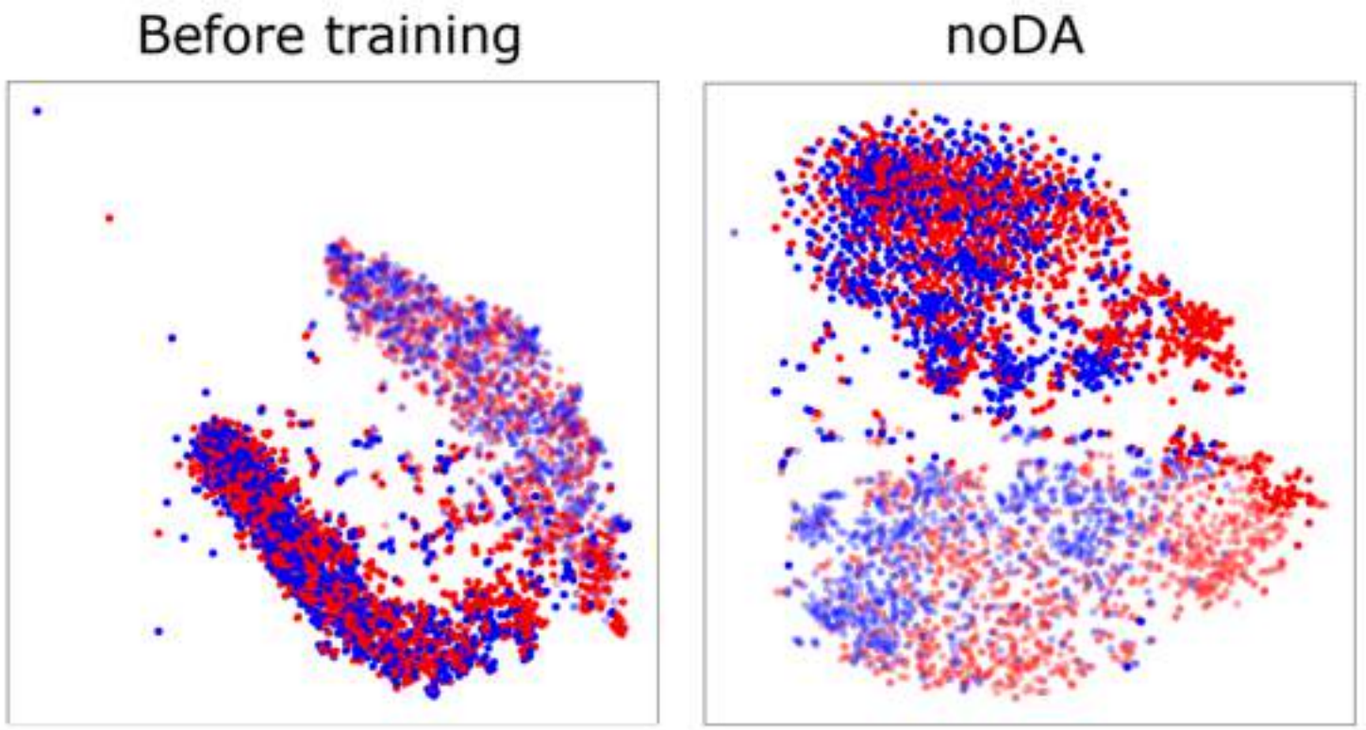


Dominguez-Sanchez+18b

OR YOU CAN ALSO TRAIN ON SIMULATIONS

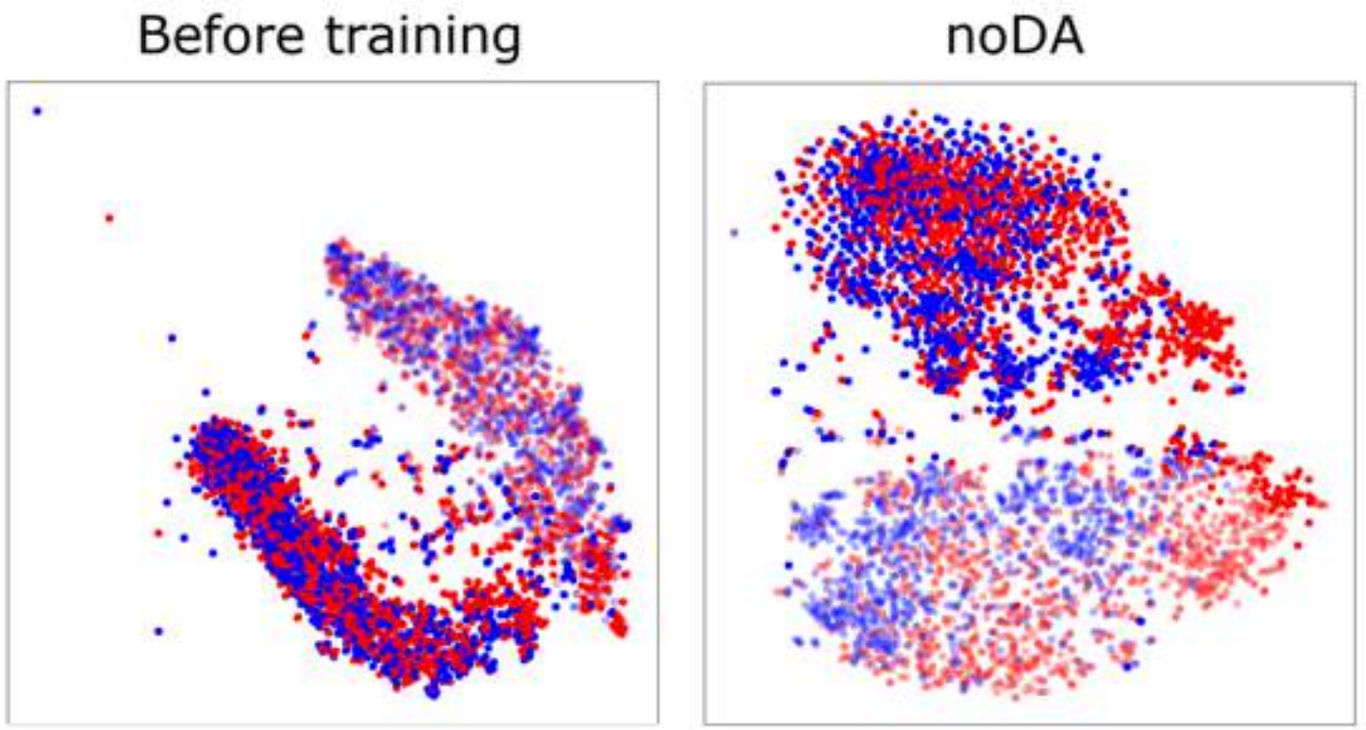


THIS IS WHERE DOMAIN KNOWLEDGE COMES INTO PLAY



Layers	Properties	Stride	Padding	Output Shape	Parameters
Input	$3 \times 75 \times 75$[a]	-	-	(3, 75, 75)	0
Convolution (2D)	Filters: 8	1	2	(8, 75, 75)	608
	Kernel: 5×5	-	-	-	-
	Activation: ReLU	-	-	-	-
Batch Normalization	-	-	-	(8, 75, 75)	16
MaxPooling	Kernel: 2×2	2	0	(8, 37, 37)	0
Convolution (2D)	Filters: 16	1	1	(16, 37, 37)	1168
	Kernel: 3×3	-	-	-	-
	Activation: ReLU	-	-	-	-
Batch Normalization	-	-	-	(16, 37, 37)	32
MaxPooling	Kernel: 2×2	2	0	(16, 18, 18)	0
Convolution (2D)	Filters: 32	1	1	(32, 18, 18)	4640
	Kernel: 3×3	-	-	-	-
	Activation: ReLU	-	-	-	-
Batch Normalization	-	-	-	(32, 18, 18)	64
MaxPooling	Kernel: 2×2	2	0	(32, 9, 9)	0
Flatten	-	-	-	(2592)	-
Fully connected	Activation: ReLU	-	-	(64)	165952
Fully connected	Activation: ReLU	-	-	(32)	2080
Fully connected	Activation: Softmax	-	-	(2)	66

Ciprijanovic+21



Layers	Properties	Stride	Padding	Output Shape	Parameters
Input	$3 \times 75 \times 75$[a]	-	-	(3, 75, 75)	0
Convolution (2D)	Filters: 8	1	2	(8, 75, 75)	608
	Kernel: 5×5	-	-	-	-
	Activation: ReLU	-	-	-	-
Batch Normalization	-	-	-	(8, 75, 75)	16
MaxPooling	Kernel: 2×2	2	0	(8, 37, 37)	0
Convolution (2D)	Filters: 16	1	1	(16, 37, 37)	1168
	Kernel: 3×3	-	-	-	-
	Activation: ReLU	-	-	-	-
Batch Normalization	-	-	-	(16, 37, 37)	32
MaxPooling	Kernel: 2×2	2	0	(16, 18, 18)	0
Convolution (2D)	Filters: 32	1	1	(32, 18, 18)	4640
	Kernel: 3×3	-	-	-	-
	Activation: ReLU	-	-	-	-
Batch Normalization	-	-	-	(32, 18, 18)	64
MaxPooling	Kernel: 2×2	2	0	(32, 9, 9)	0
Flatten	-	-	-	(2592)	-
Fully connected	Activation: ReLU	-	-	(64)	165952
Fully connected	Activation: ReLU	-	-	(32)	2080
Fully connected	Activation: Softmax	-	-	(2)	66

Ciprijanovic+21

Let's focus first on discriminative models, where X is of high dimension, i.e. images, spectra, time series, sequences …