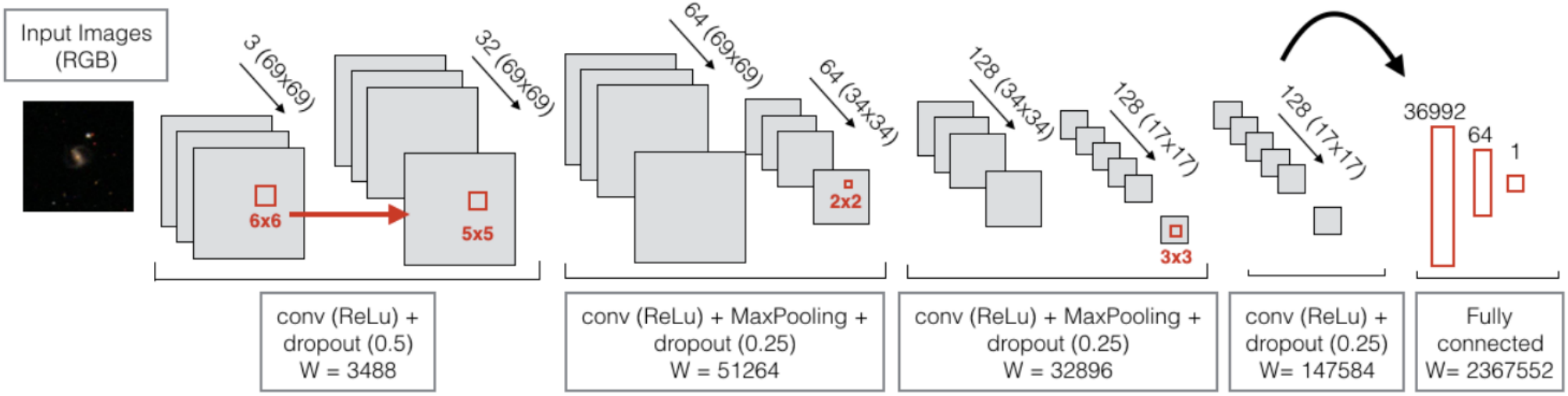
PART IV: BEYOND CLASSIFICATION: IMAGE2IMAGE NETWORKS

UP TO NOW CNNs MAP IMAGES (SIGNALS) INTO FLOATS



Dominguez-Sanchez+18

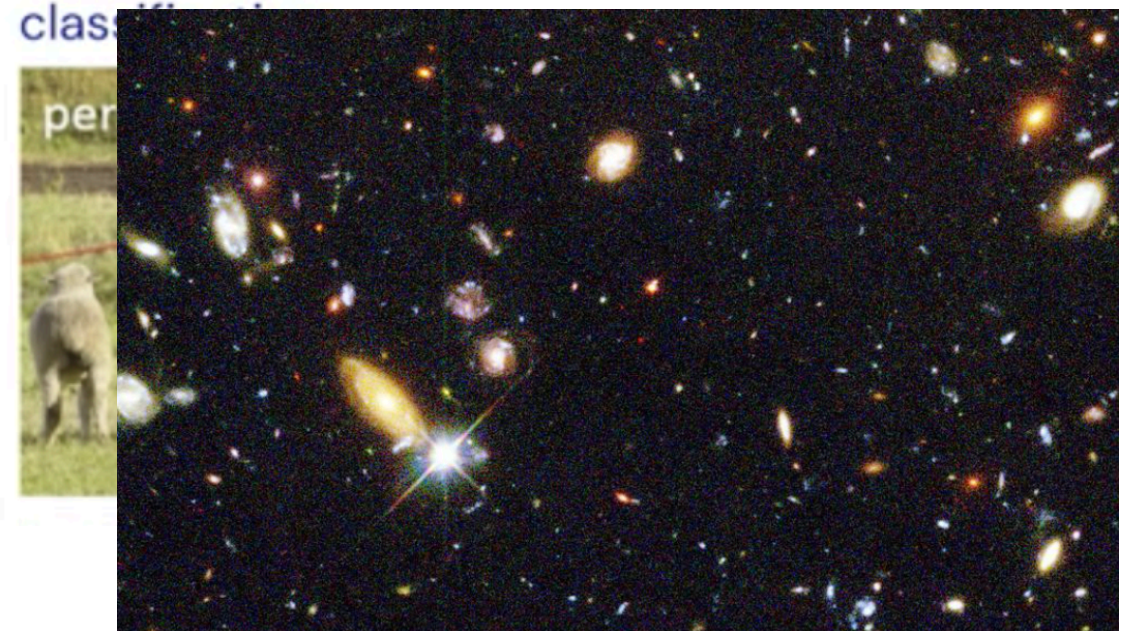
Classification has its limits ….

HOW DO I CLASSIFY THIS IMAGE?

Classification has its limits ….

HOW DO I CLASSIFY THIS IMAGE?

Classification has its limits ….

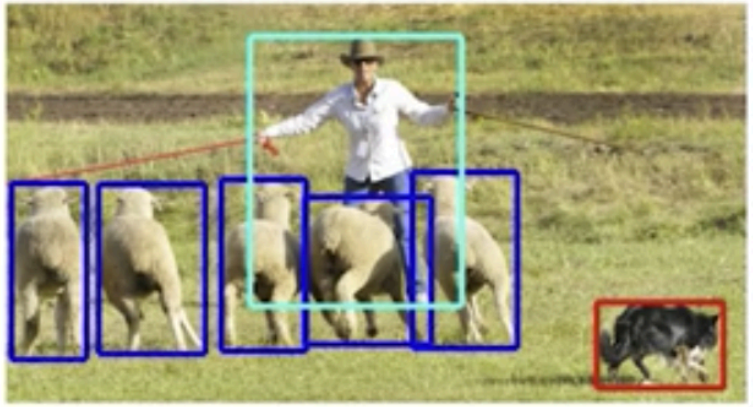


HOW DO I CLASSIFY THIS IMAGE?

oing beyond classificaiton: increasing complexity

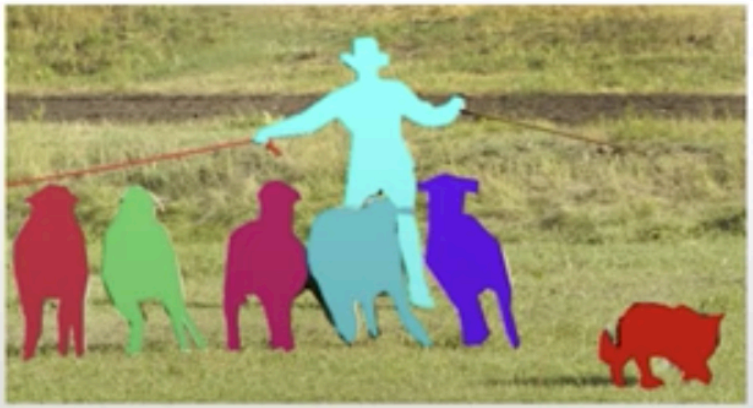
classification

object detection



semantic segmentation

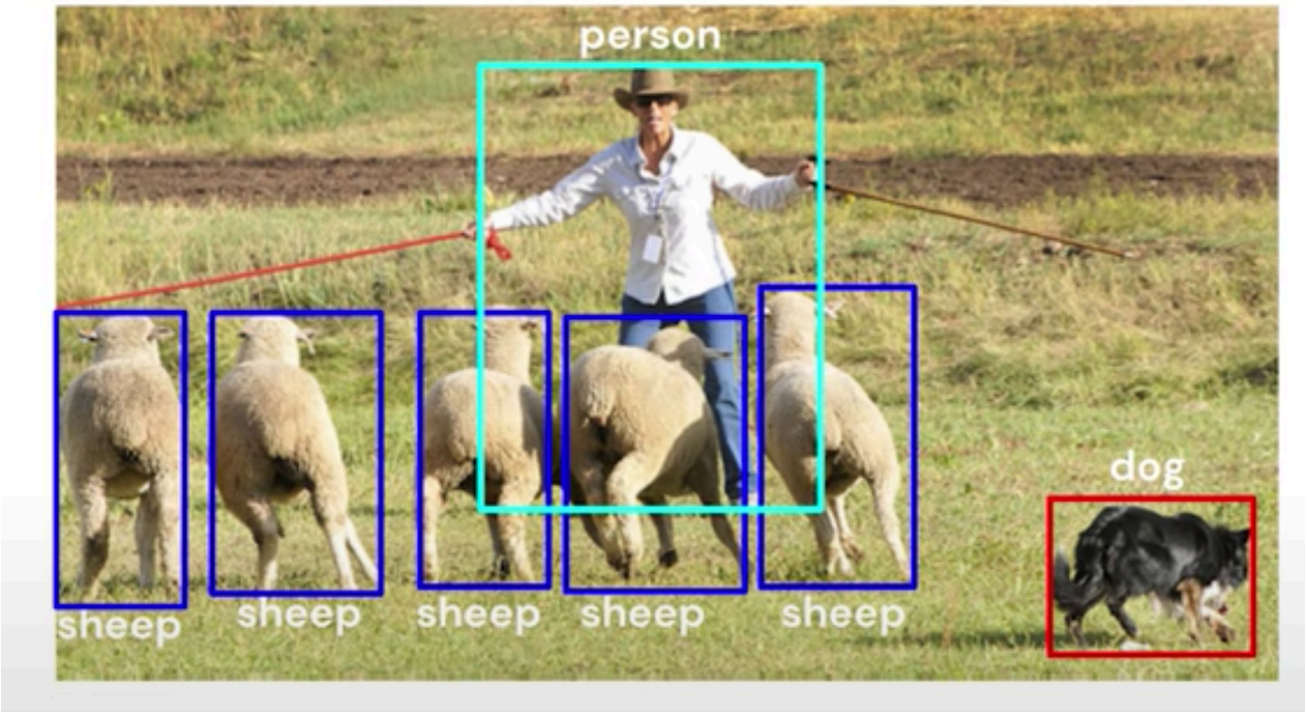
instance segmentation



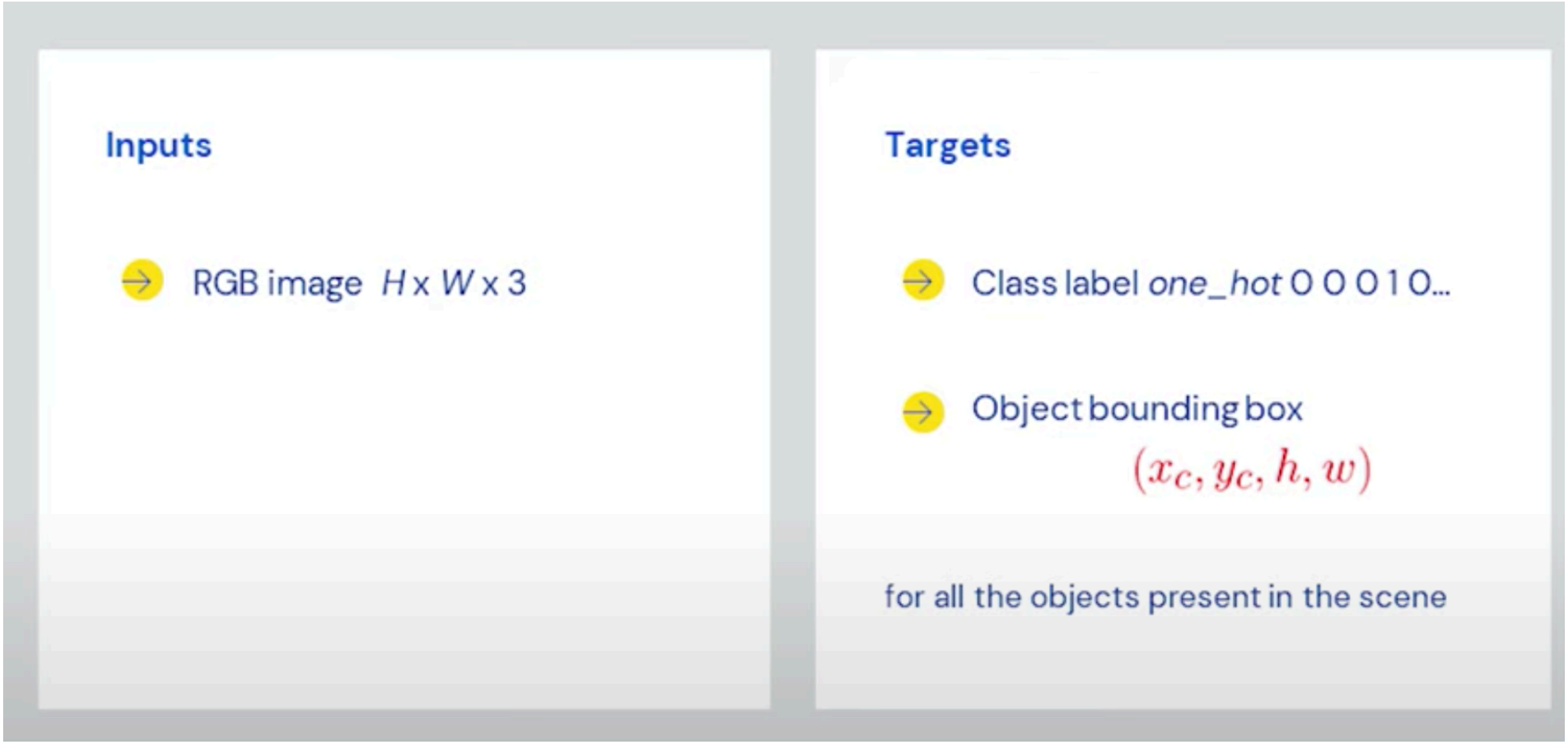
@deepmind (V. Patraucean)

Object detection

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



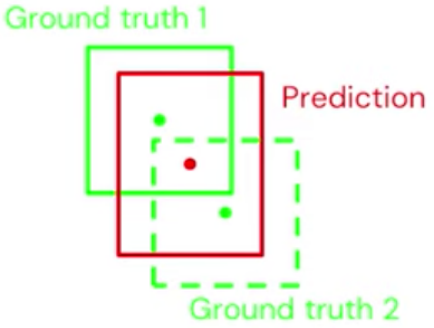
@deepmind (V. Patraucean)



@deepmind (V. Patraucean)

WHAT WOULD BE THE LOSS FUNCTION OF SUCH A PROBLEM?

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

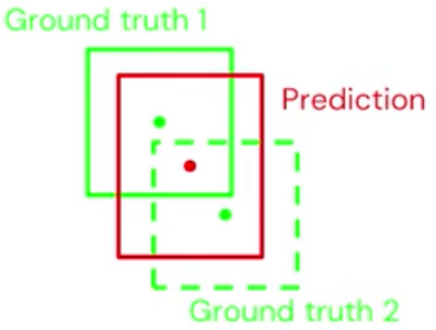


IT IS A REGRESSION WITH A SIMPLE QUADRATIC LOSS. WE TRY TO FIND THE BEST COORDINATES OF THE BOUNDING BOX.

@deepmind (V. Patraucean)

WHAT WOULD BE THE LOSS FUNCTION OF SUCH A PROBLEM?

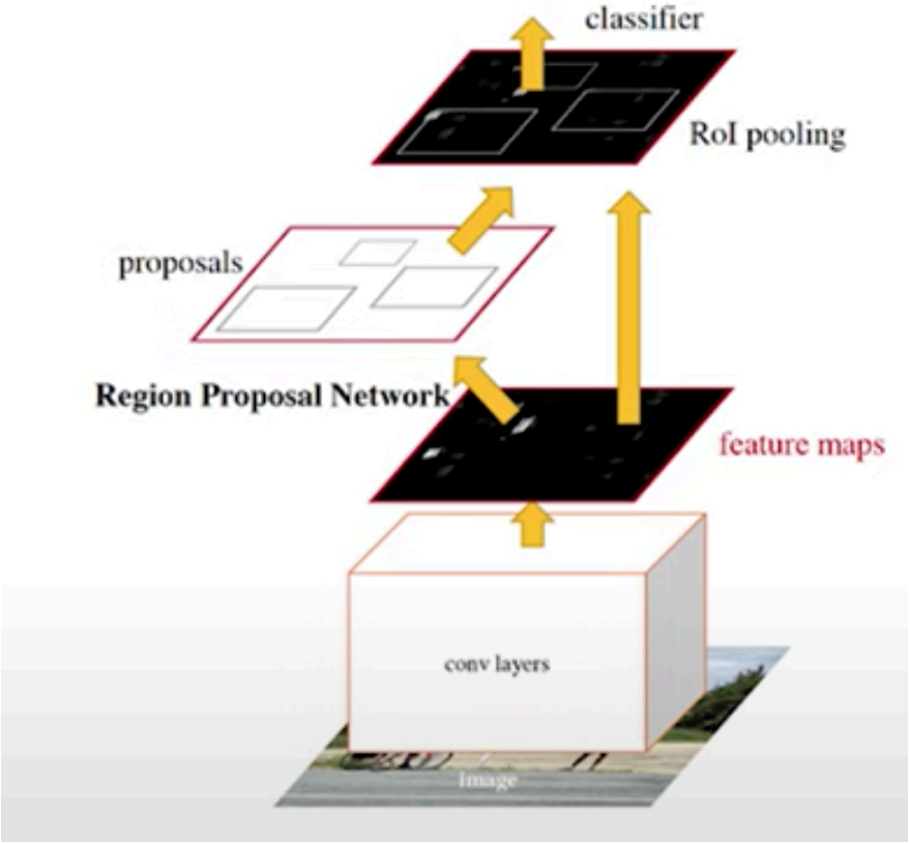
$$\frac{1}{N}\sum_{i=1}^{N}(y_i - p_i)^2$$



IT IS A REGRESSION WITH A SIMPLE QUADRATIC LOSS. WE TRY TO FIND THE BEST COORDINATES OF THE BOUNDING BOX.

IT BECOMES MESSY QUITE RAPIDLY…

XAMPLE: FASTER R-CNN

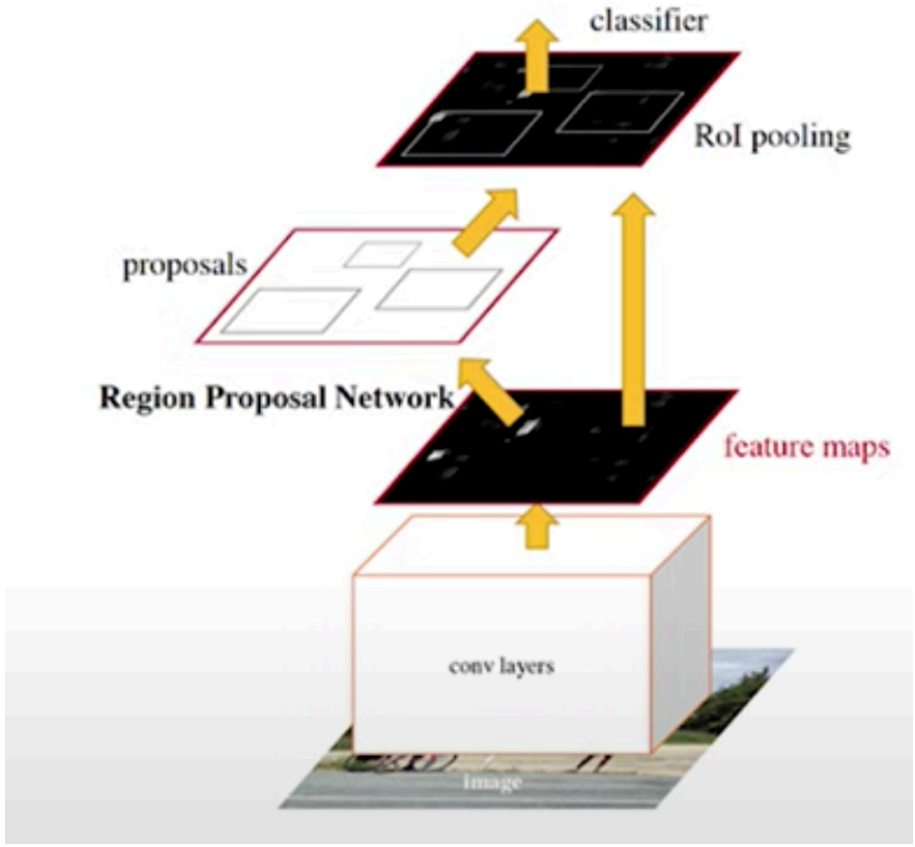


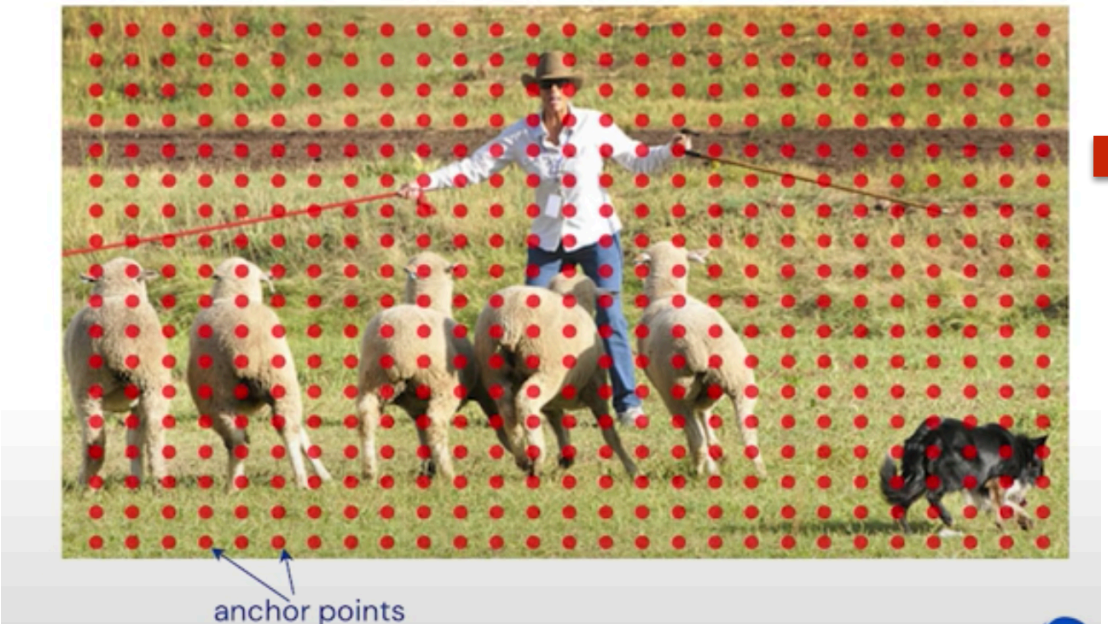
We divide the task in 2 steps:

1. Identify bounding box candidates (CLASSIFICATION)

2. Classify and Refine (REGRESSION)

Ren+16


classifier
RoI pooling
proposals
Region Proposal Network
feature maps
conv layers
image



Discretize Bounding Box Space

Choose n candidates per position

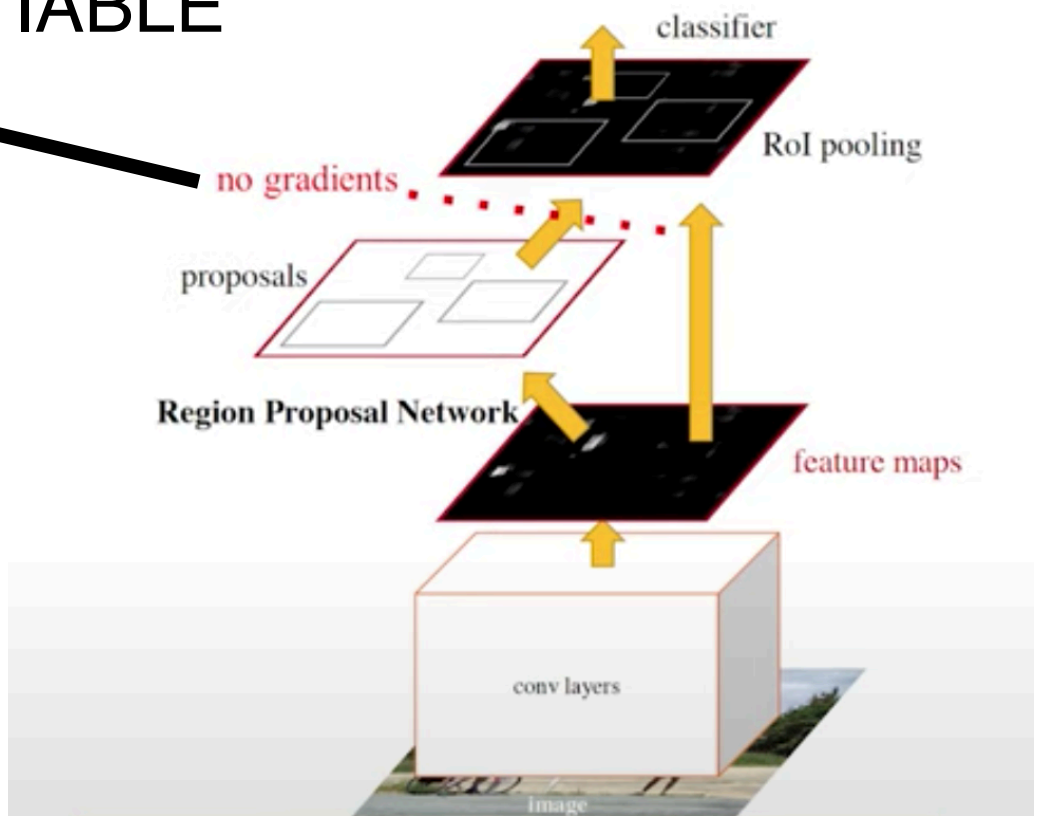
Predict objectness score (classifcation)

Sort and keep top K

@deepmind (V. Patraucean)

HIS IS NOT DIFFERENTIABLE

(FIXED)



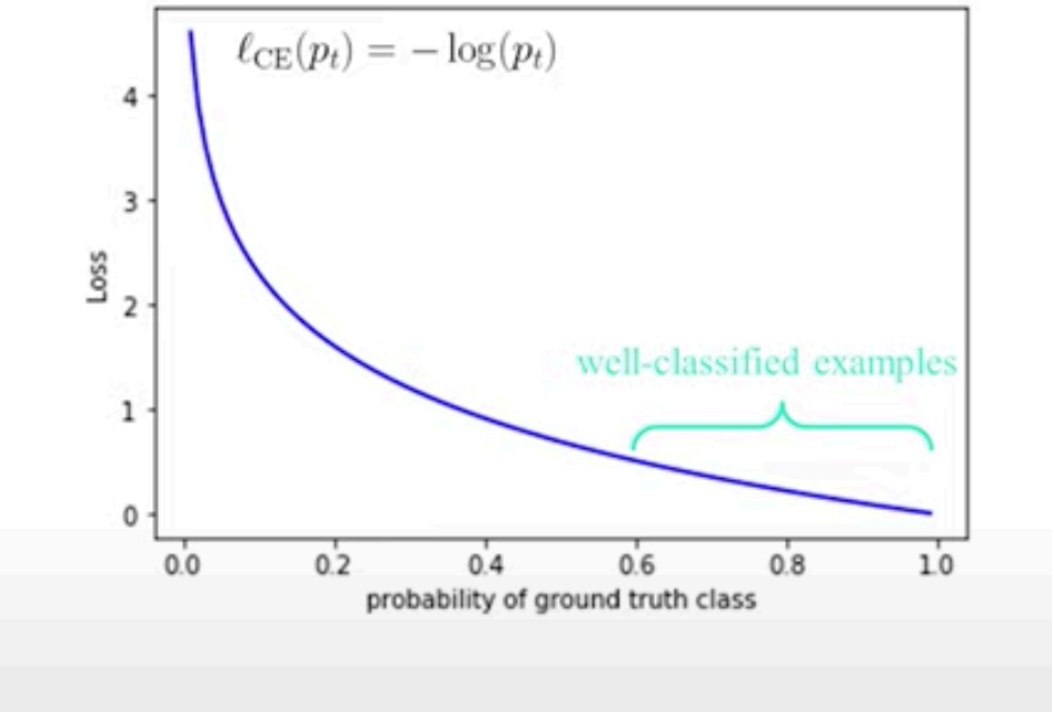
YS OF MAKING THIS DIFFERENTIABLE:

Spatial Transformer Networks (Jaderberg+18)

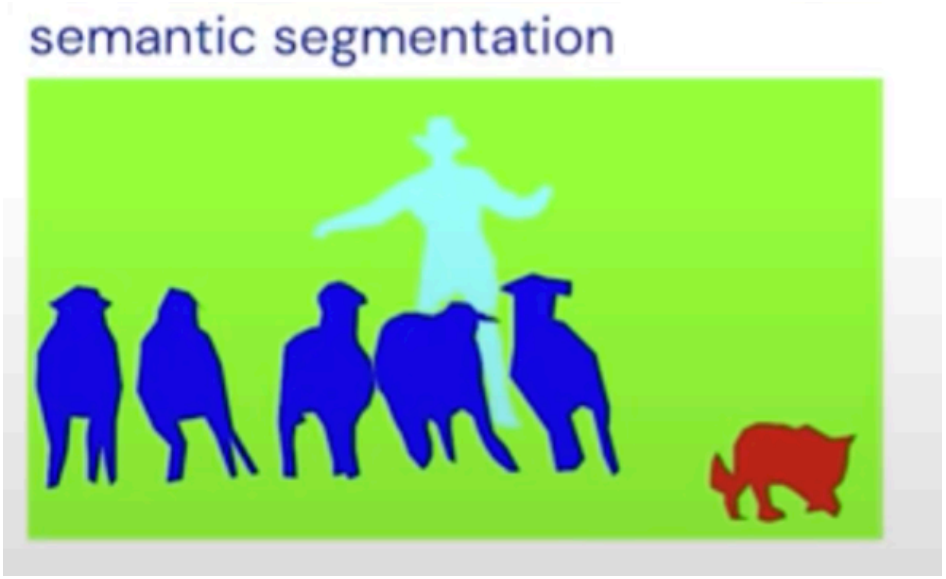
Y NOT DOING IT ONE STAGE?

MOST OF THE CANDIDATES
ARE BACKGROUND,
EASY TO IDENTIFY

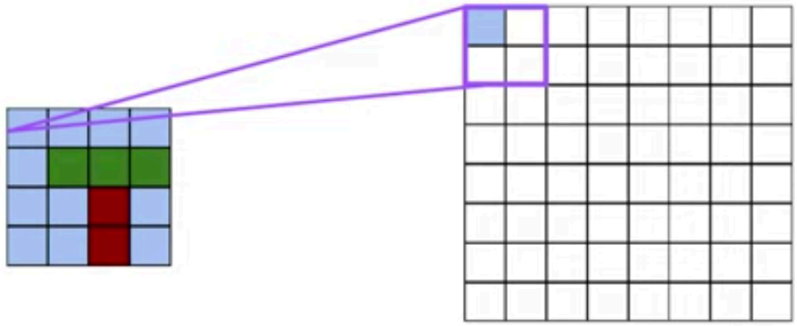
THE LOSS OF THE MANY
EASY EXAMPLES
DOMINATES OVER THE
RARE USEFUL ONES



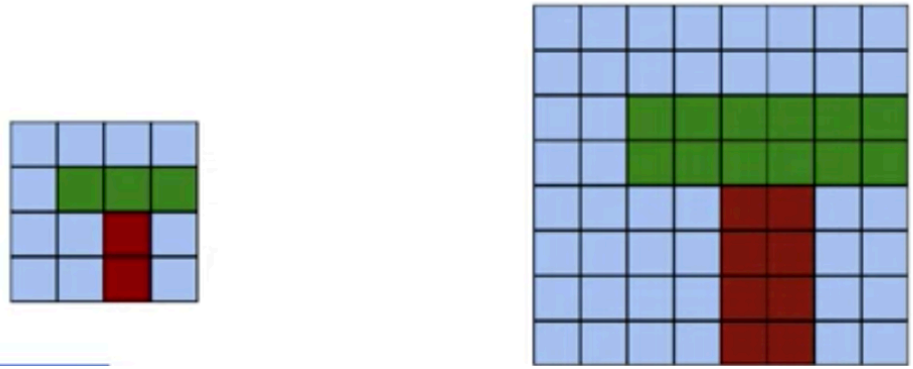
@deepmind (V. Patraucean)


semantic segmentation

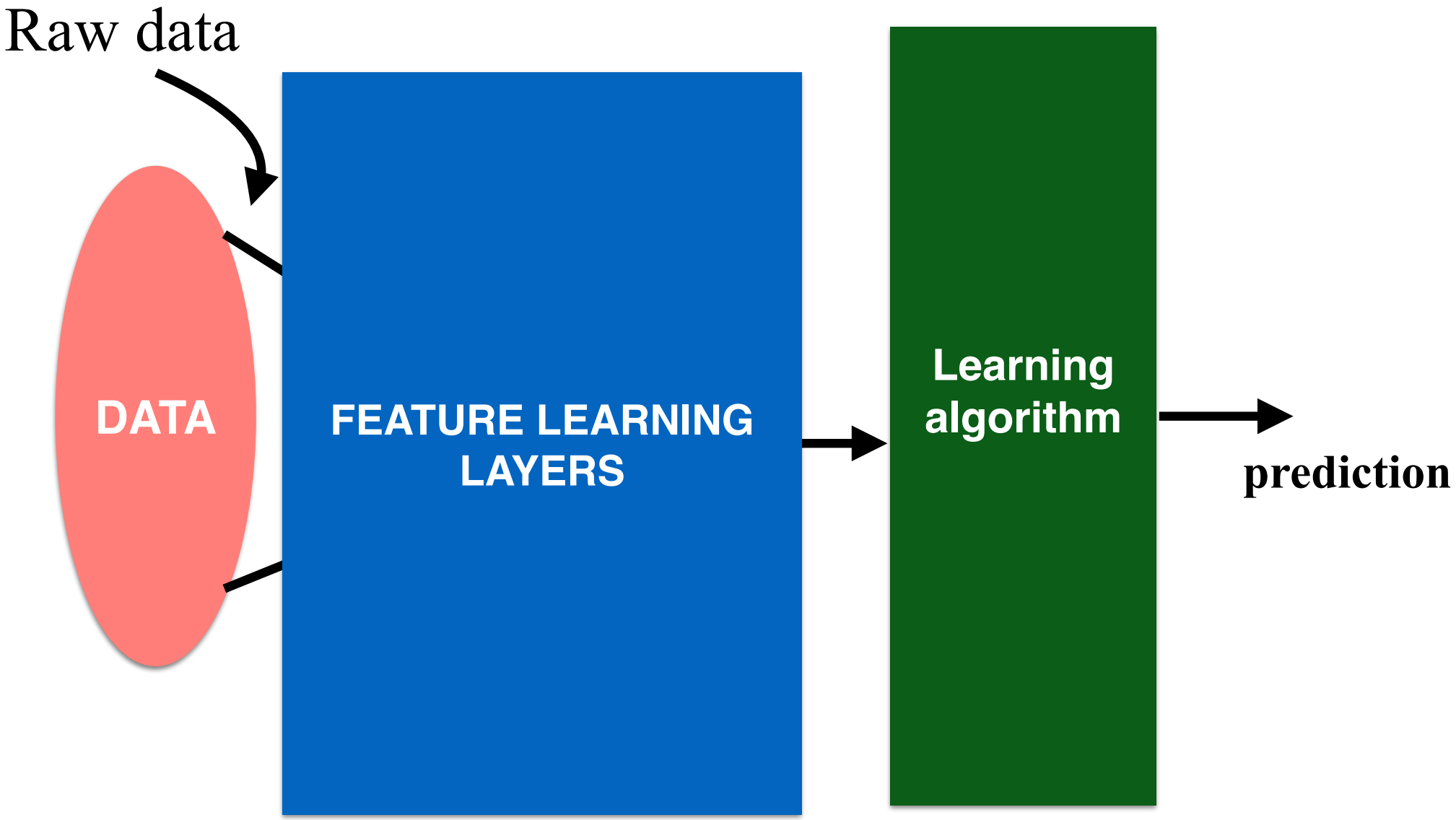
UNPOOLING OPERATION (INVERSE OF POOLING)



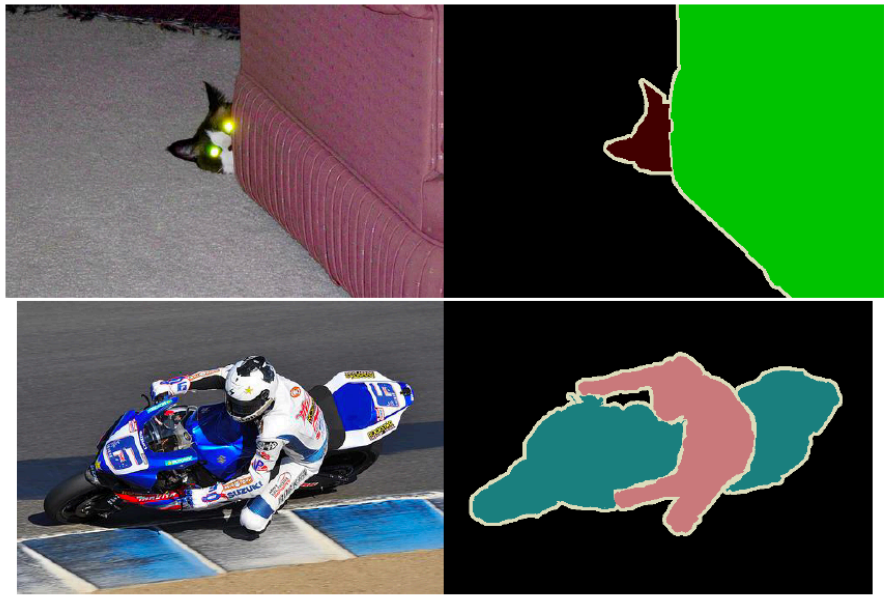
COPY PIXELS IN A GIVEN WINDOW



GENERATES LARGER IMAGES FROM SMALLER ONES

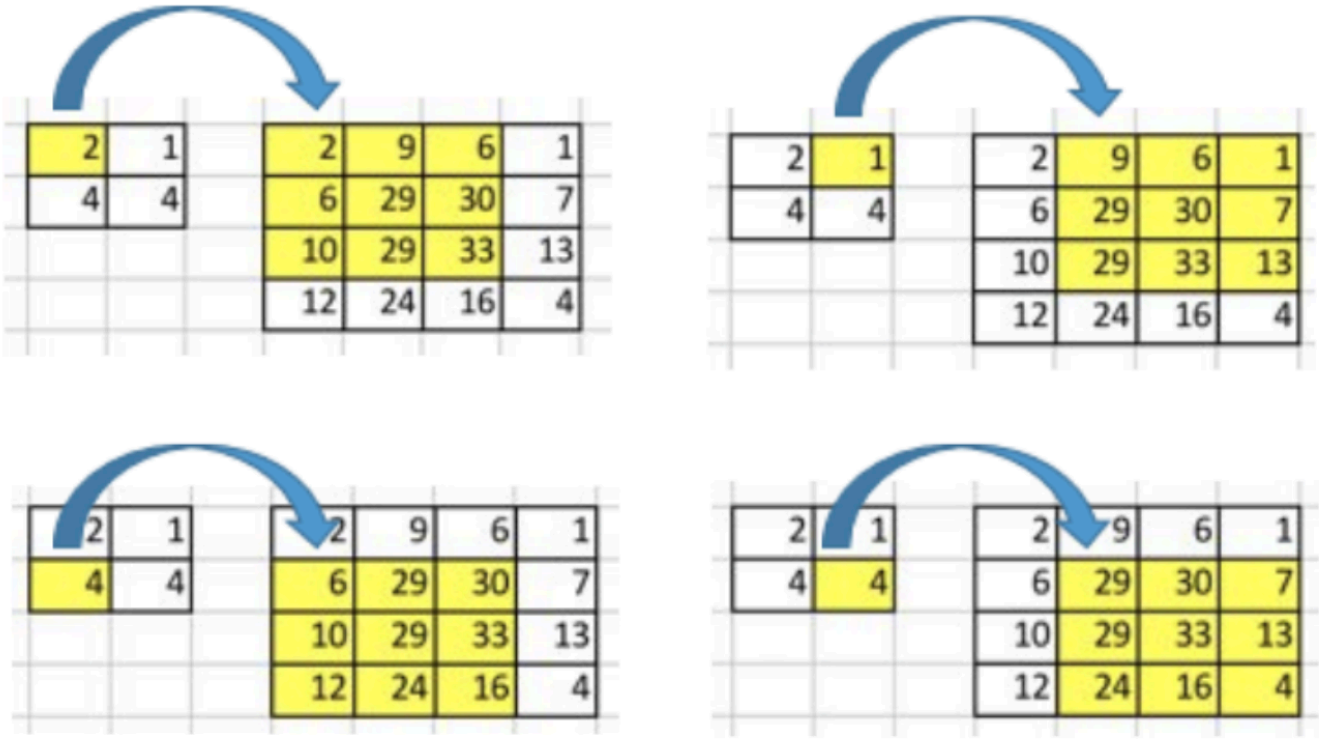

Raw data
DATA
FEATURE LEARNING LAYERS
Learning algorithm
prediction

FIRST: IMAGE SEGMENTATION WITH ENCODERS-DECODERS



TRANSPOSED CONVOLUTION

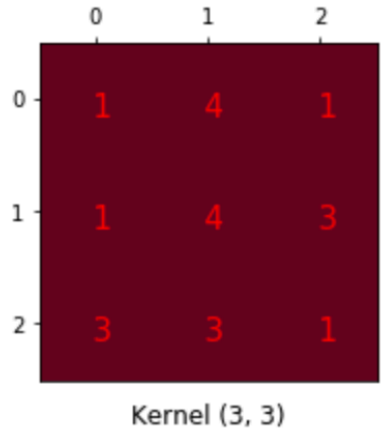
ALLOWS TO INCREASE THE SIZE



Going Backward of Convolution

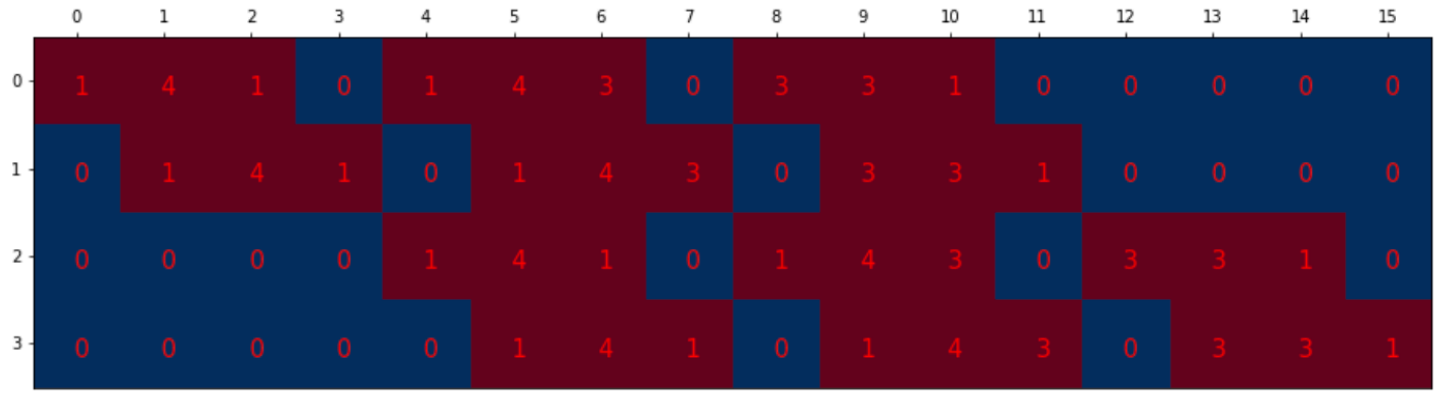
EXAMPLE TAKEN FROM HERE

CONVOLUTION MATRIX



Kernel (3, 3)

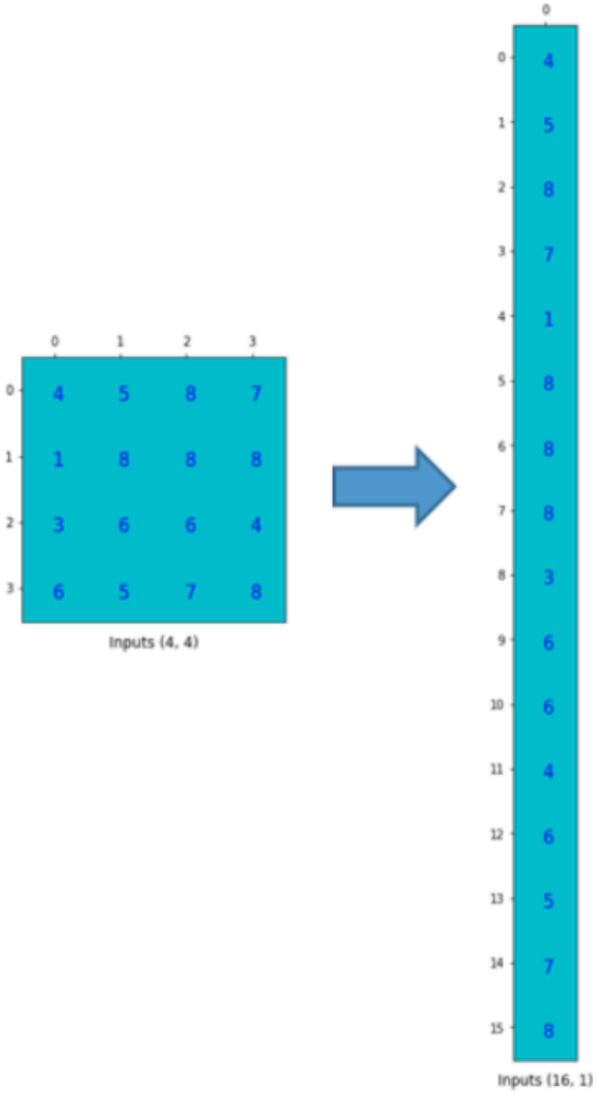
THE KERNEL CAN BE ARRANGED IN FORM OF A MATRIX:



Convolution Matrix (4, 16)

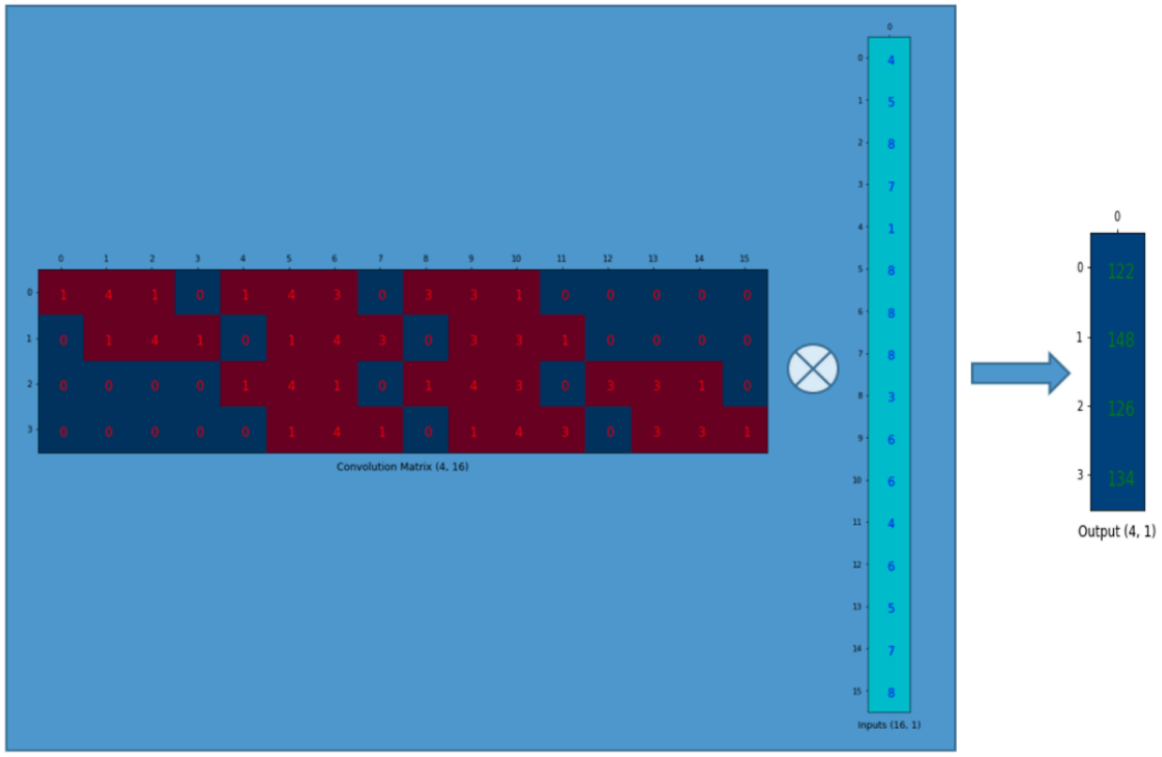
EXAMPLE TAKEN FROM HERE

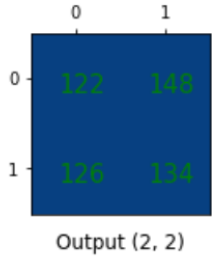
THE INPUT IS FLATTENED INTO A COLUMN VECTOR



EXAMPLE TAKEN FROM HERE

THE CONVOLUTION IS TRANSFORMED INTO A PRODUCT OF MATRICES

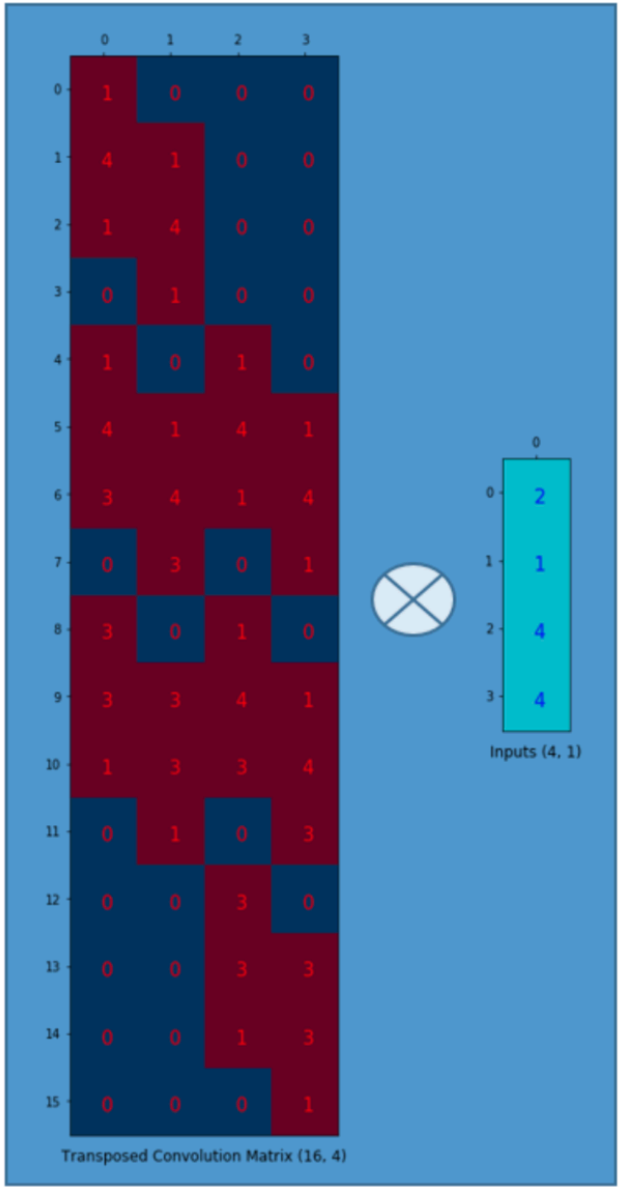


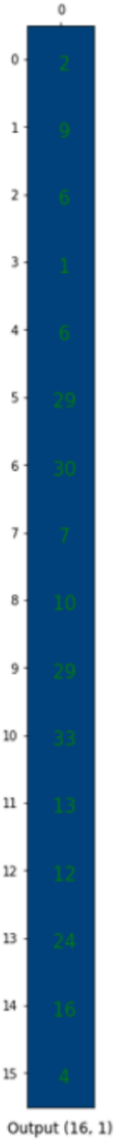


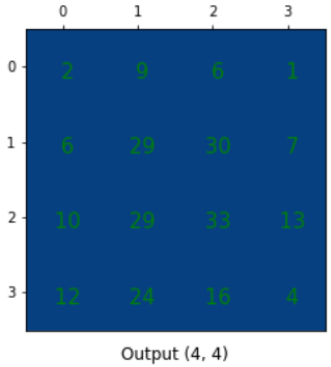
RESHAPED OUTPUT

EXAMPLE TAKEN FROM HERE

THE TRANSPOSED CONVOLUTION IS THE INVERSE OPERATION



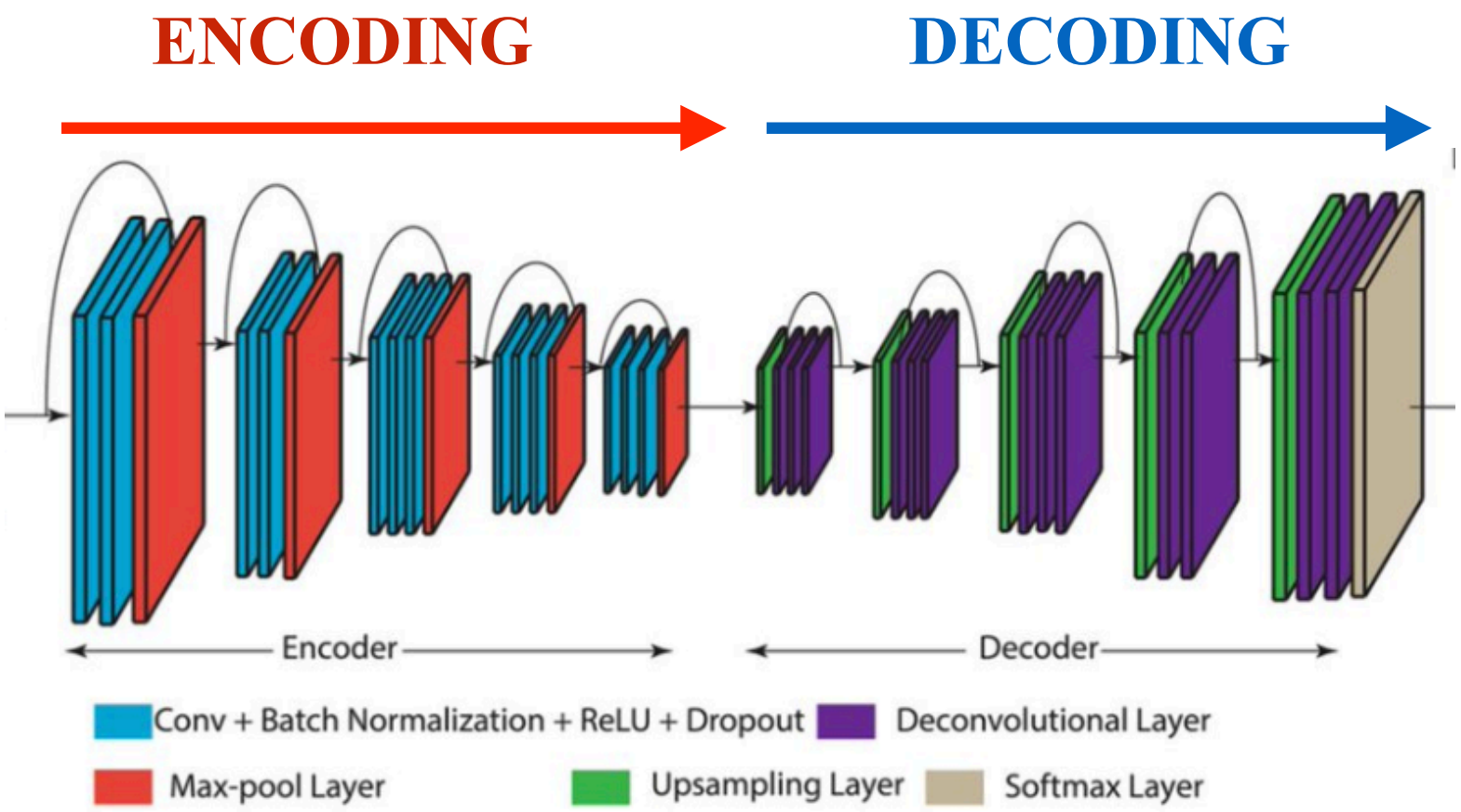




RESHAPED OUTPUT

EXAMPLE TAKEN FROM HERE

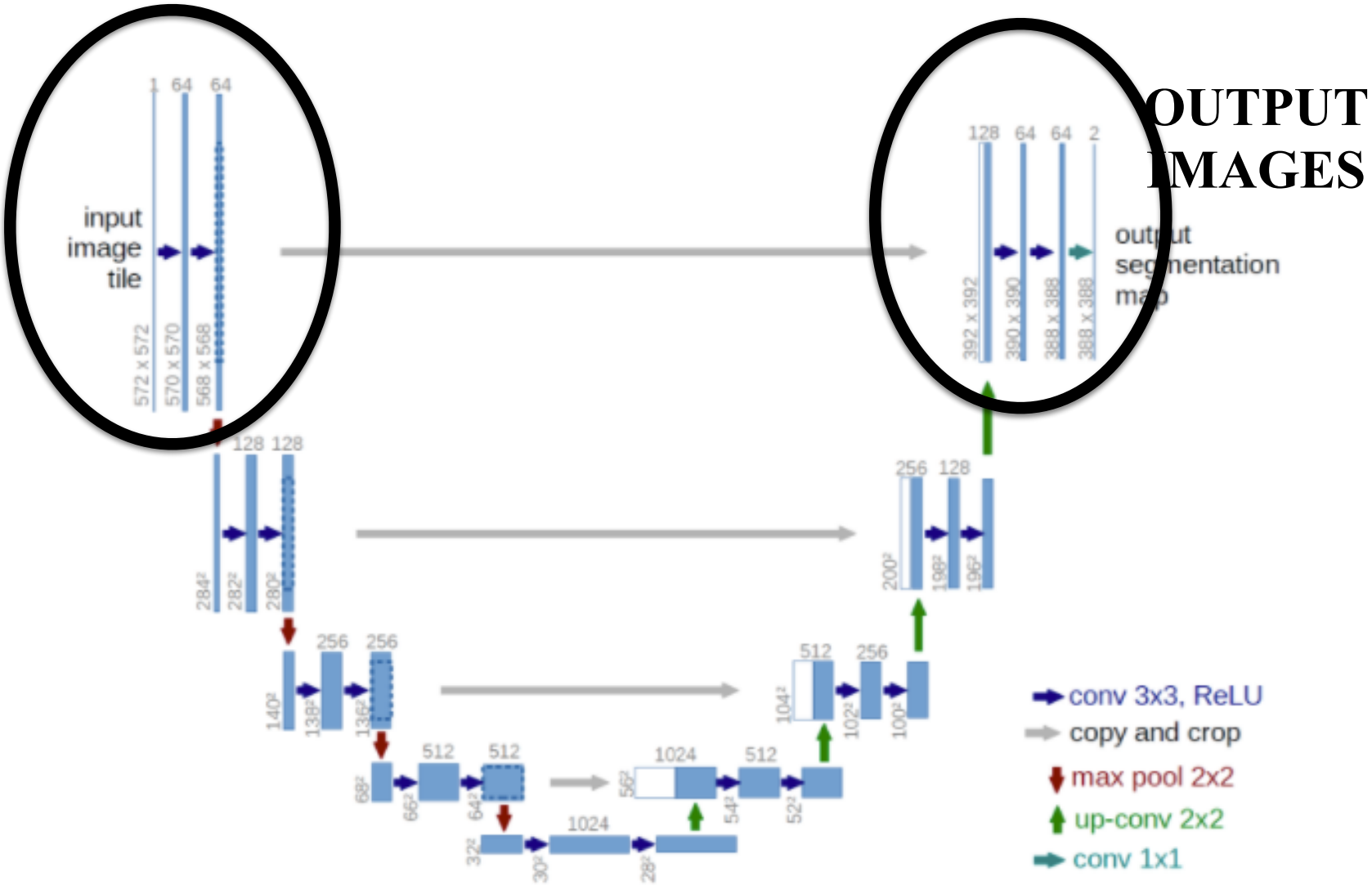
ENCODER-DECODERS GO FROM IMAGE 2 IMAGE



WE CALL THIS FULLY CONVOLUTIONAL NEURAL NETWORKS

ENCODING-DECODING TO EXTRACT IMAGE FEATURES: U-NET

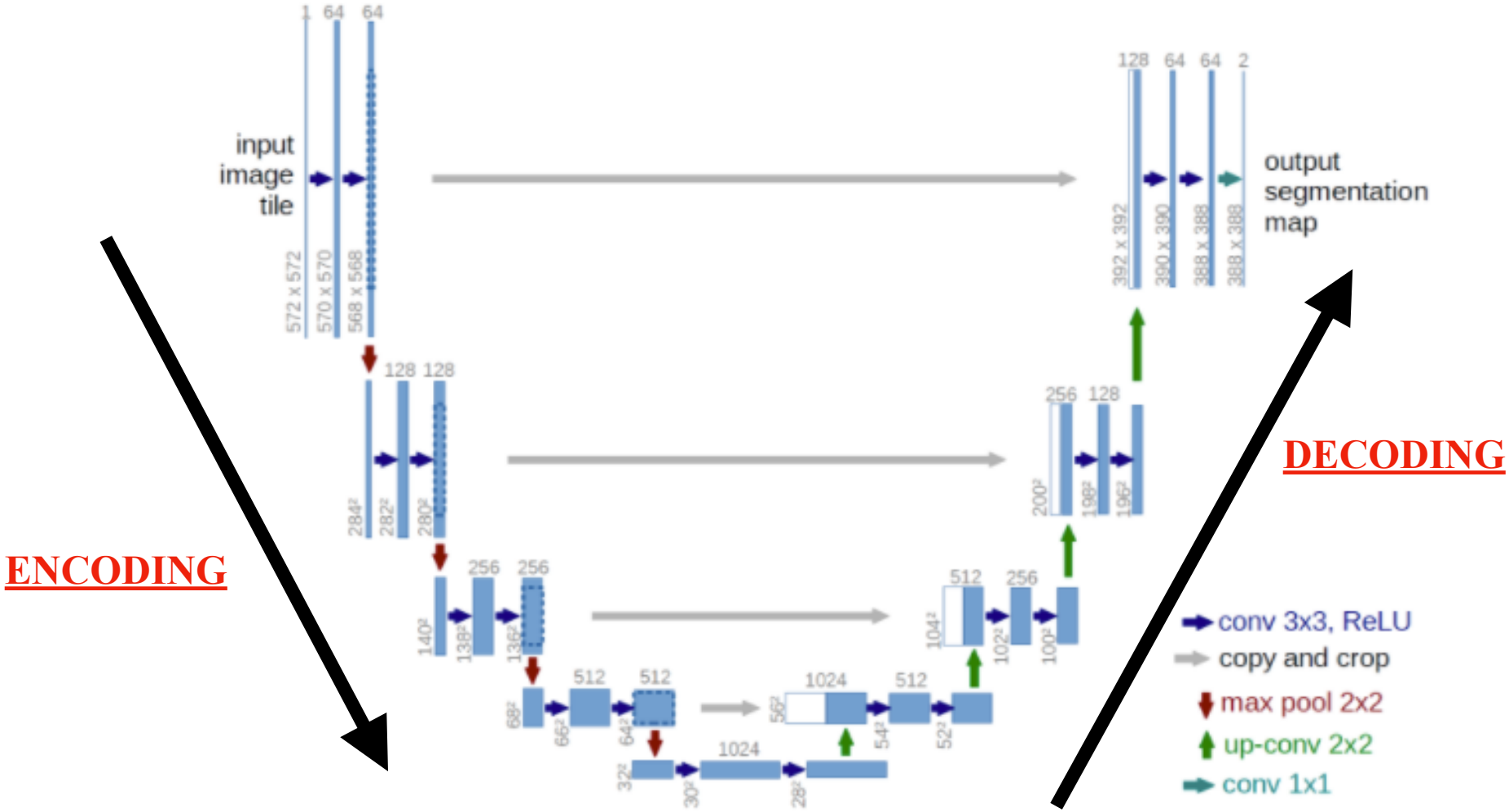
INPUT IMAGES



OUTPUT IMAGES

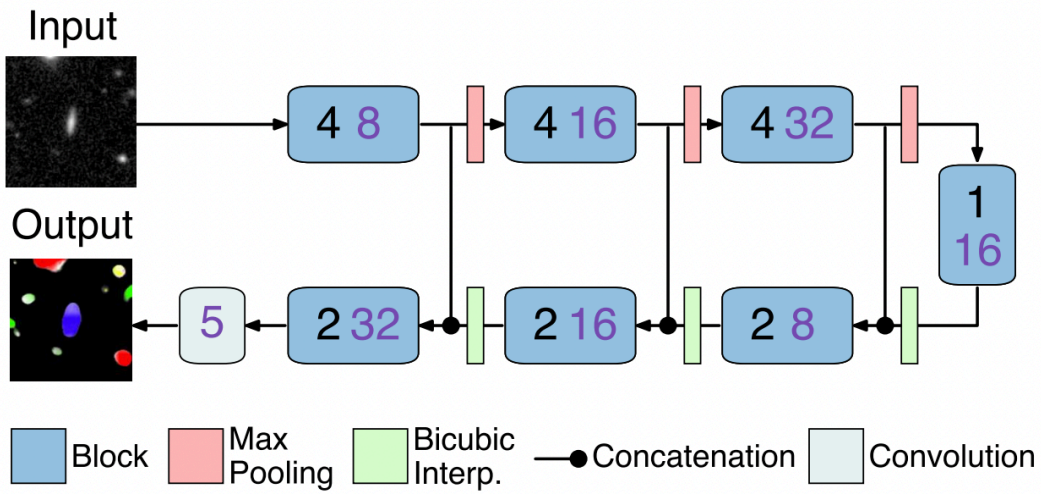
Ronnerberger+15

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

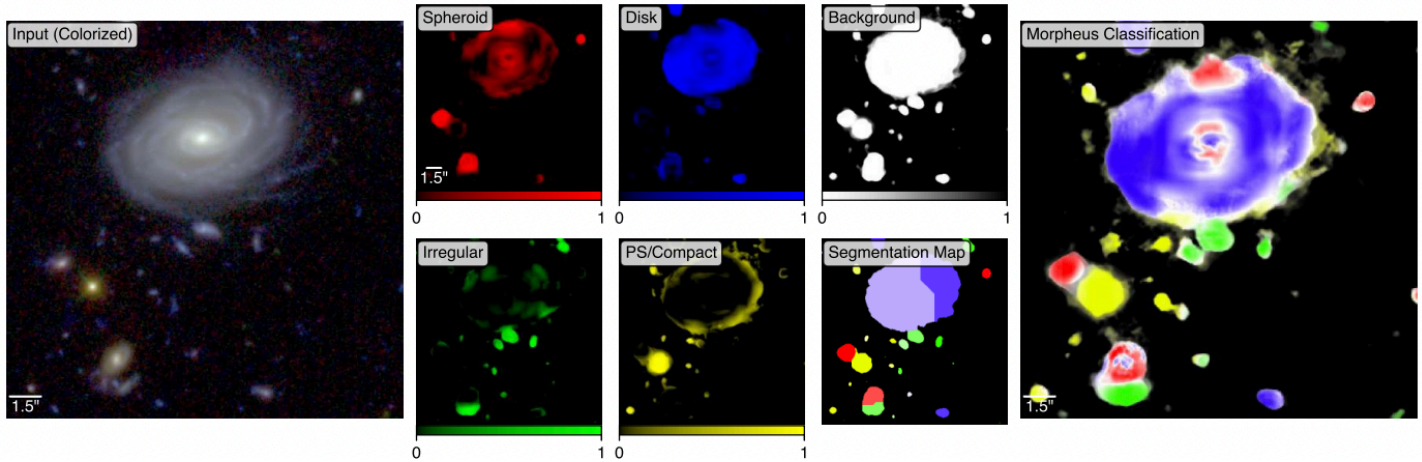


Ronnerberger+15

2. Segmentation

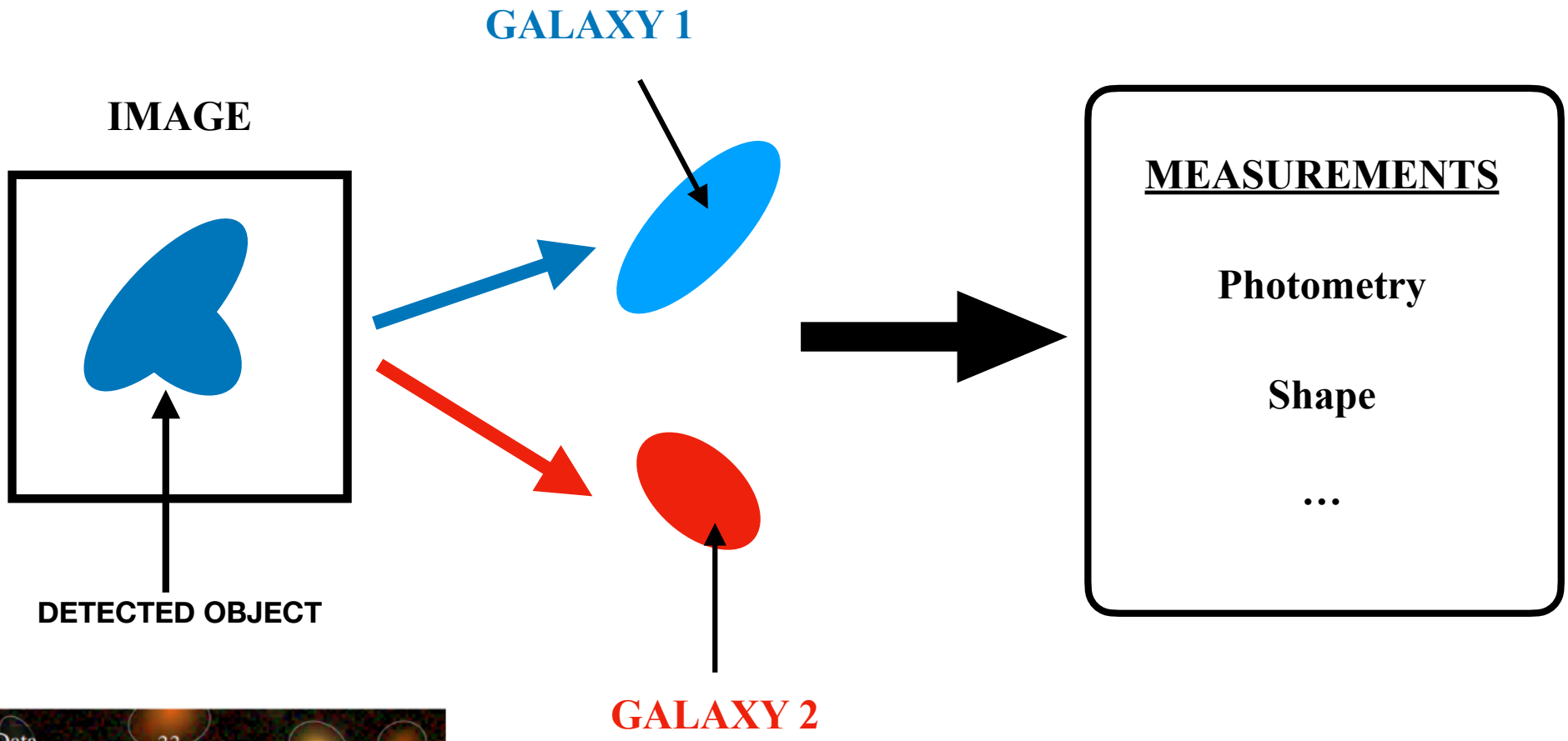


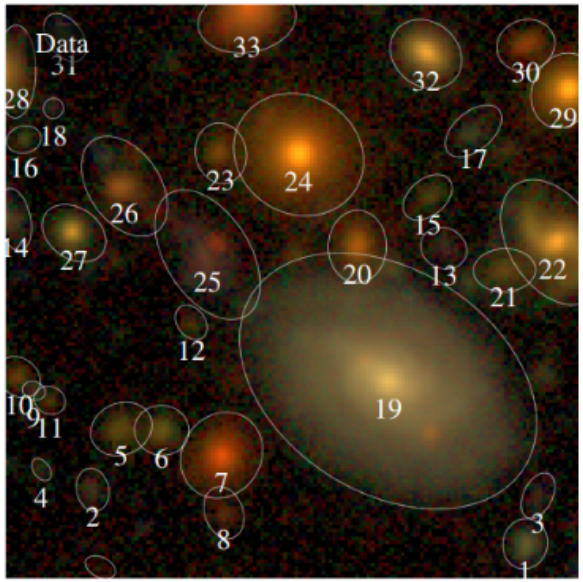
Classification at the pixel level



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





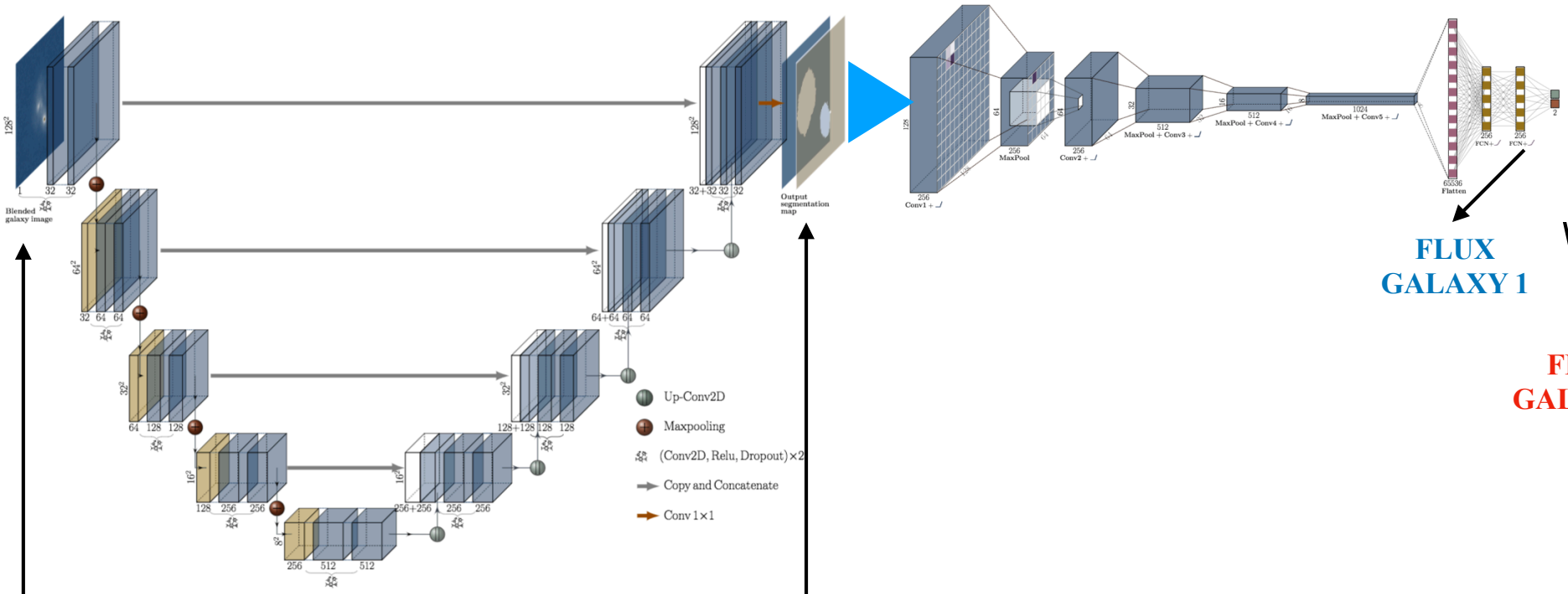
Bosch+17

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

Boucaud+19

U-NET (ENCODER DECODER)

VANILLA CNN



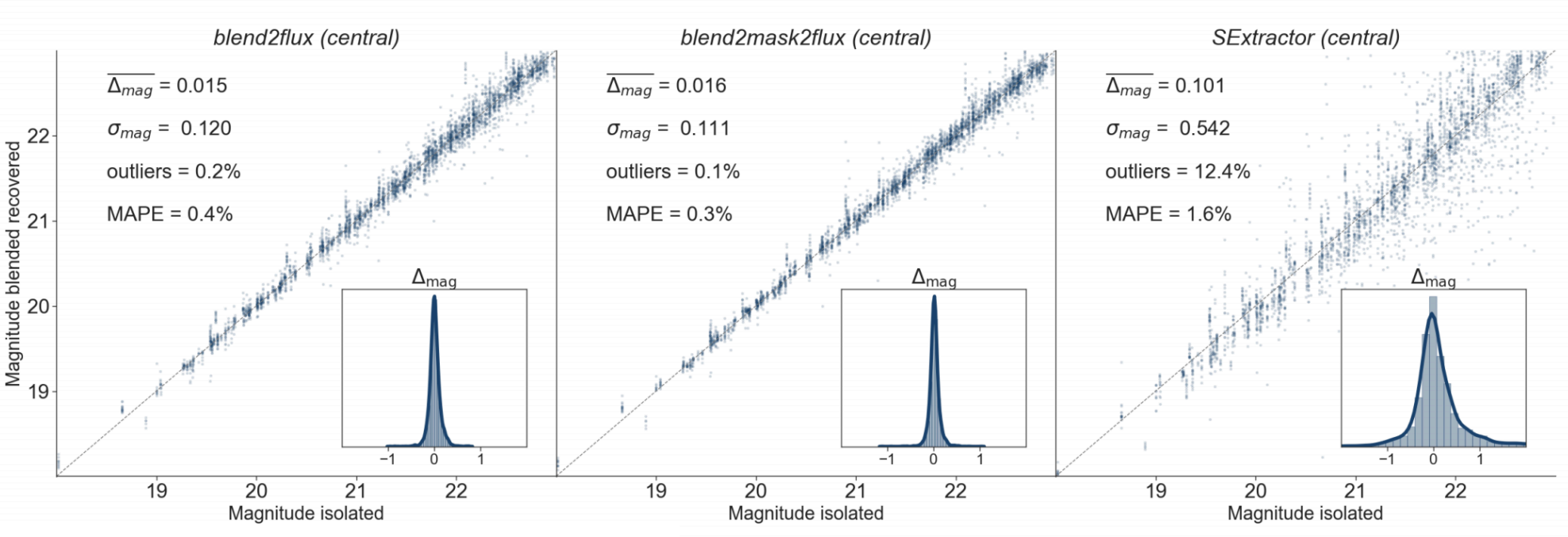
FLUX GALAXY 1

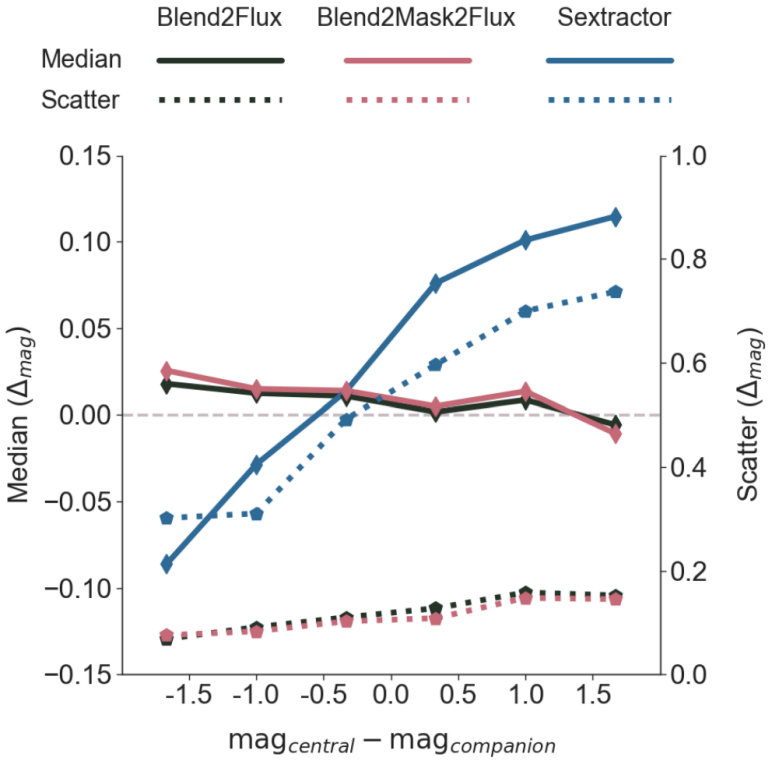
FLUX GALAXY 2

IMAGE OF BLENDED OBJECTS

OUTPUT SEGMENTATION MAP (BINARY 2 CHANNELS)

Boucaud, MHC+19



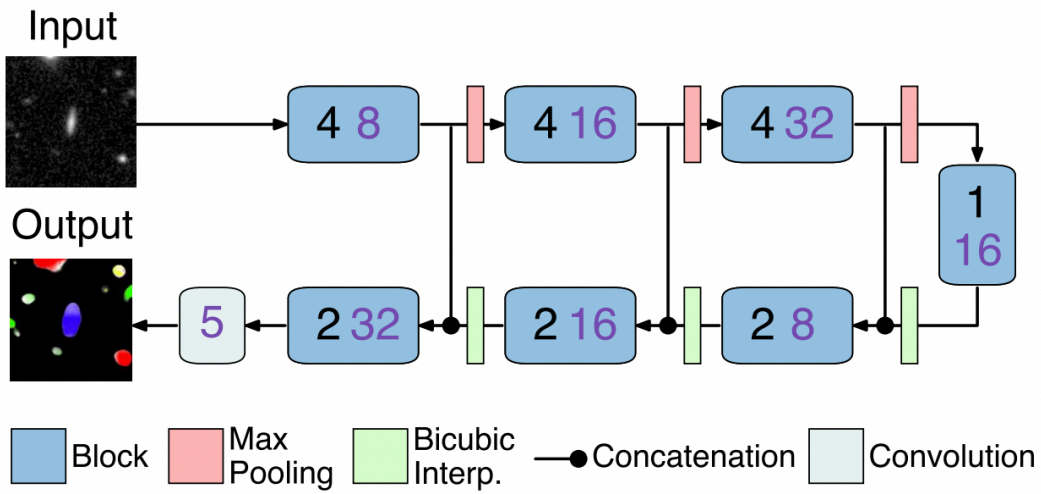


LESSONS LEARNED:

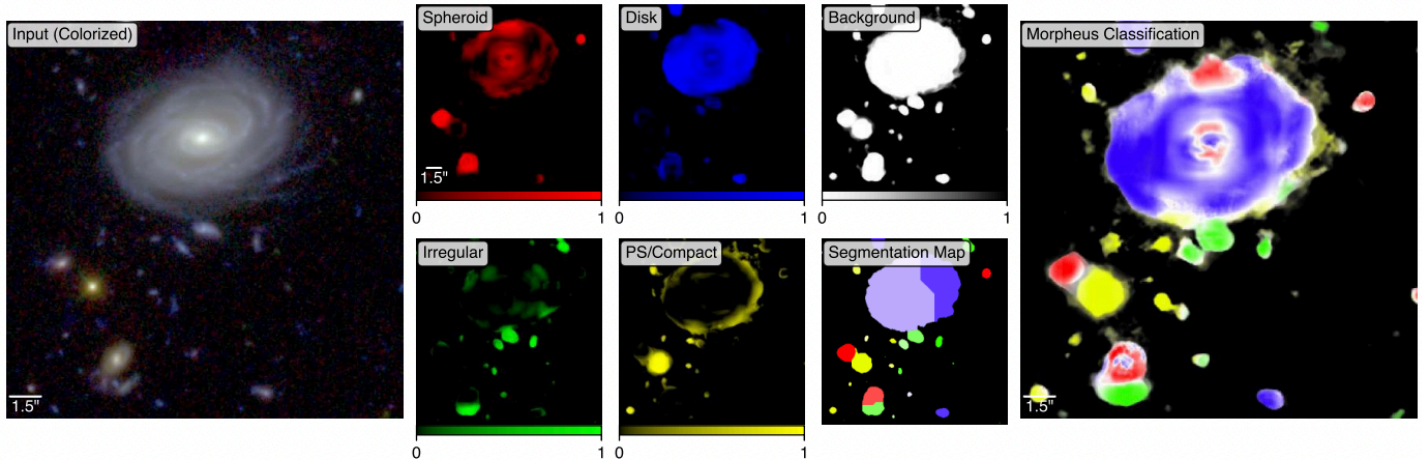
- It is easy to do better than Sextractor

- Masks do not provide additional information

2. Segmentation



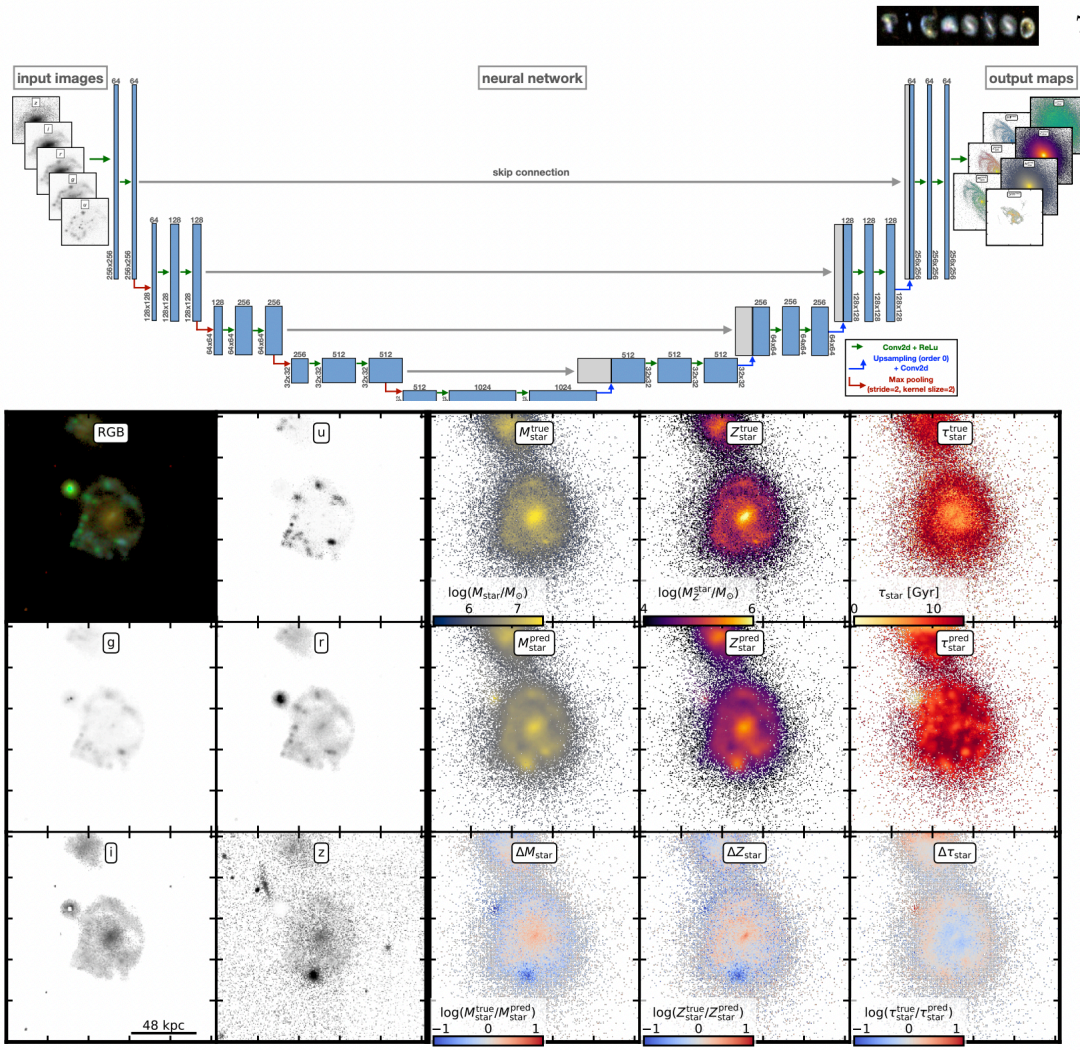
Classification at the pixel level

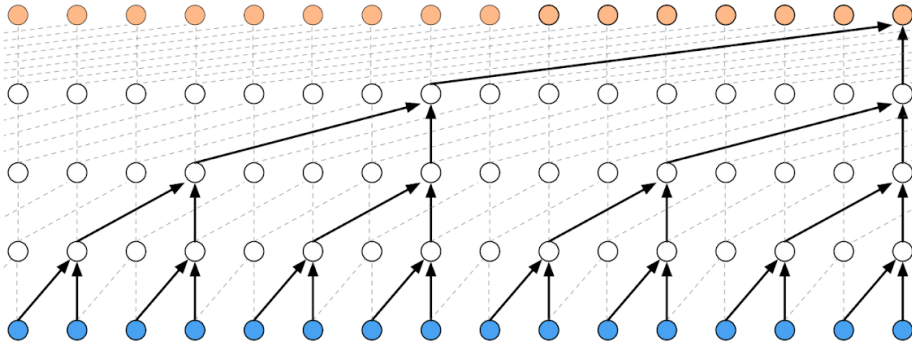


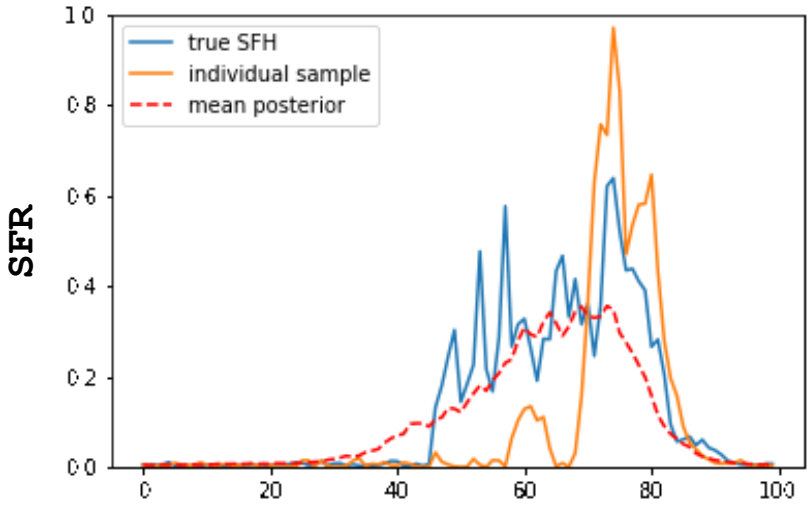
Hausen+20

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



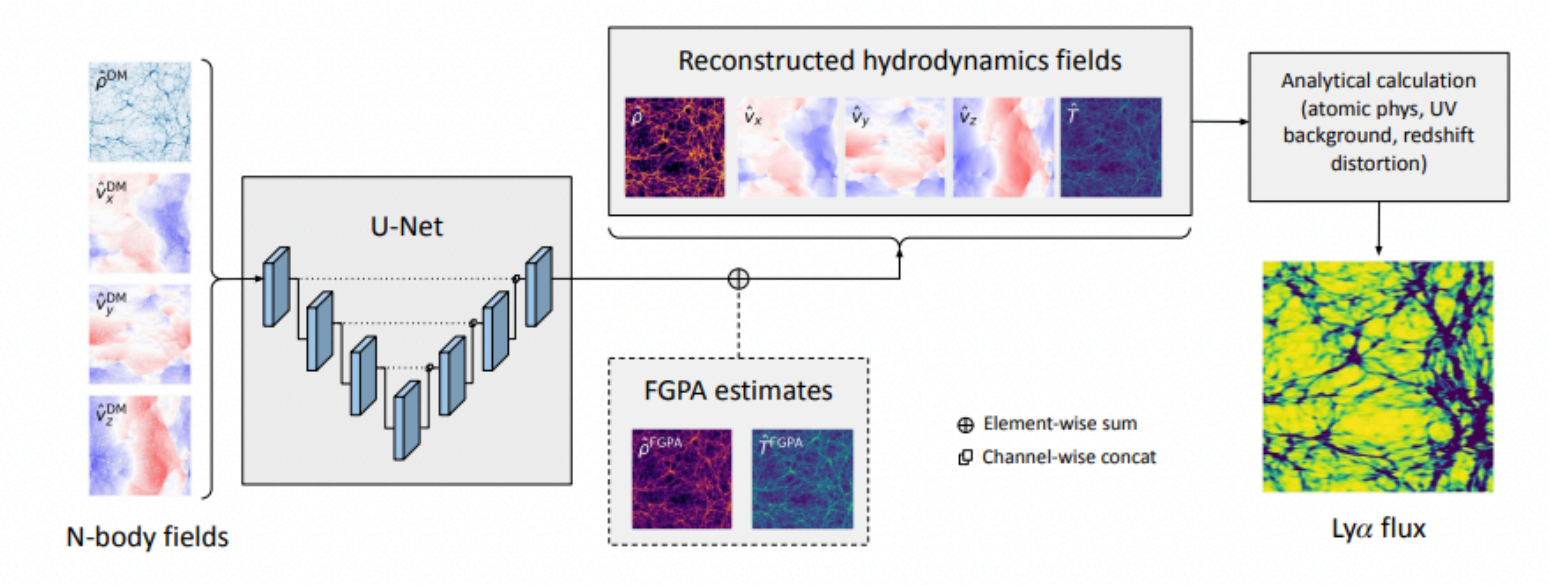




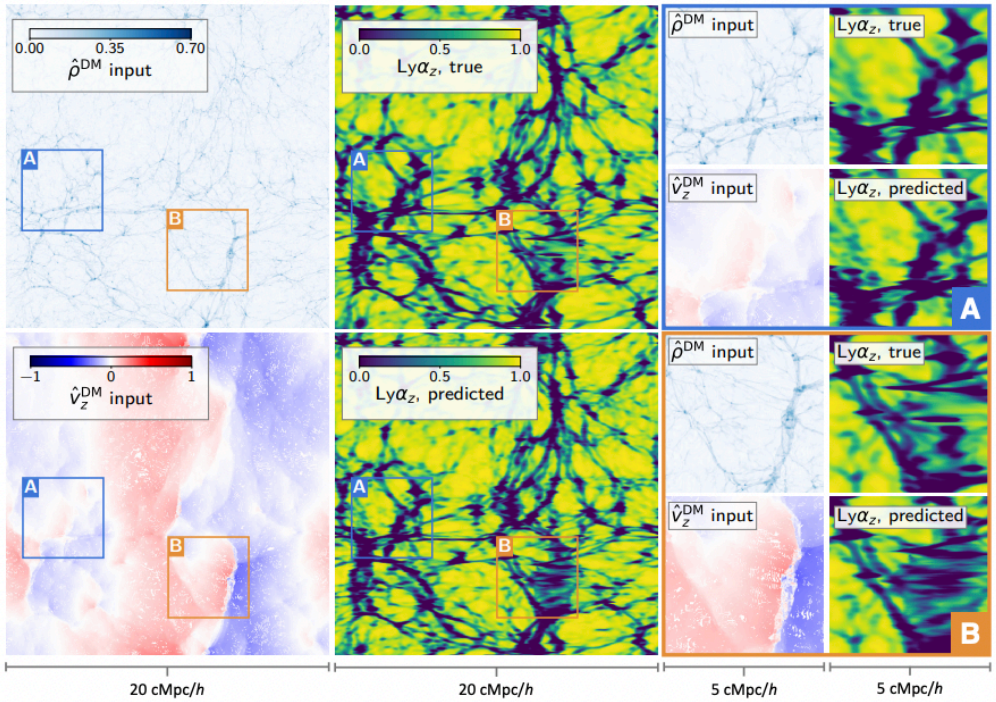
Buck+21

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



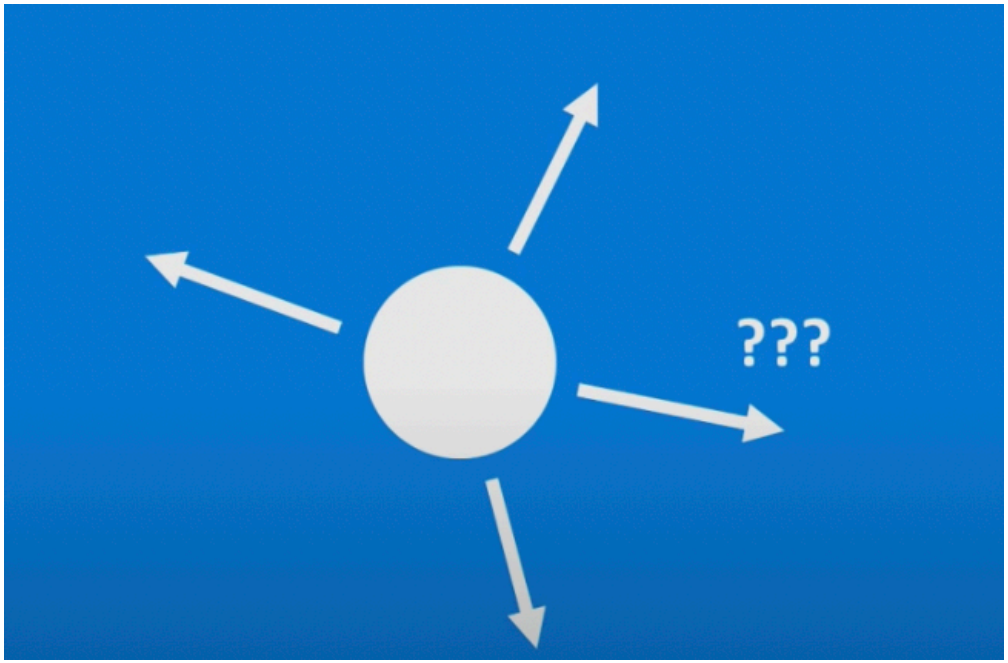
Neural Networks are used to learn the non-linear mapping between cheap dark matter only simulations to expensive baryonic physics

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

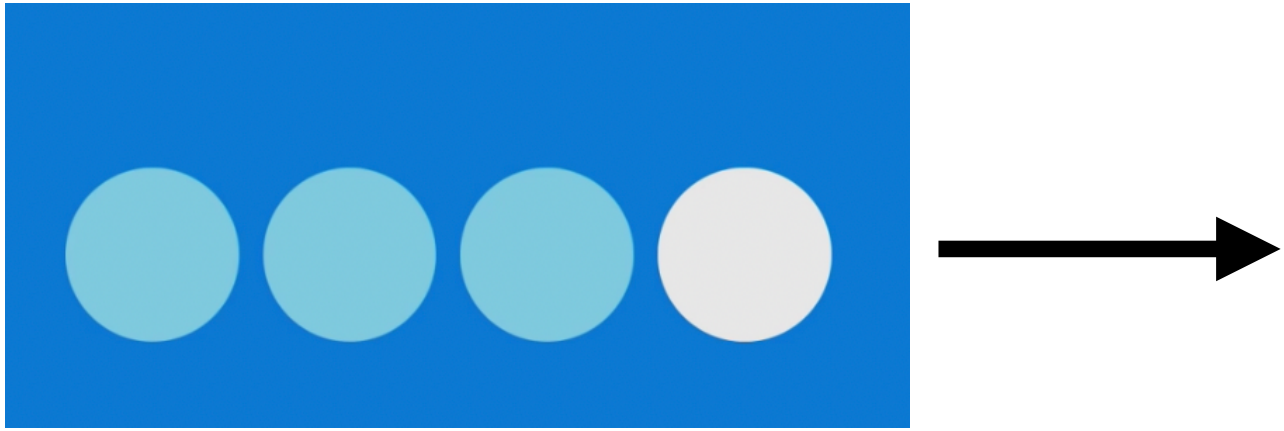
SEQUENCE MODELLING

*based on MIT Lecture by Ava Soleimany

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

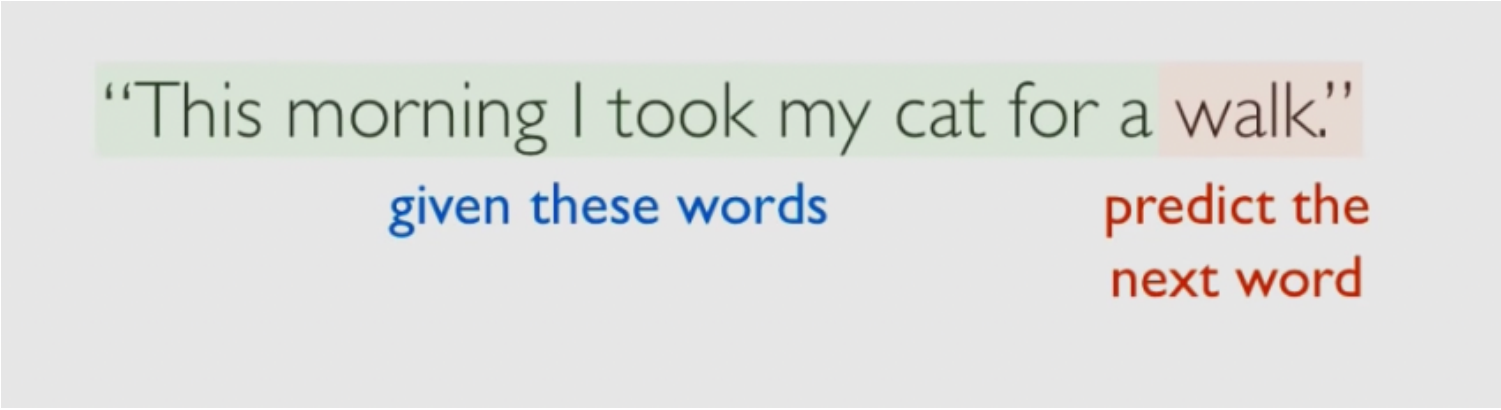


GIVEN AN IMAGE OF A BALL, CAN YOU PREDICT WHERE IT WILL GO NEXT?

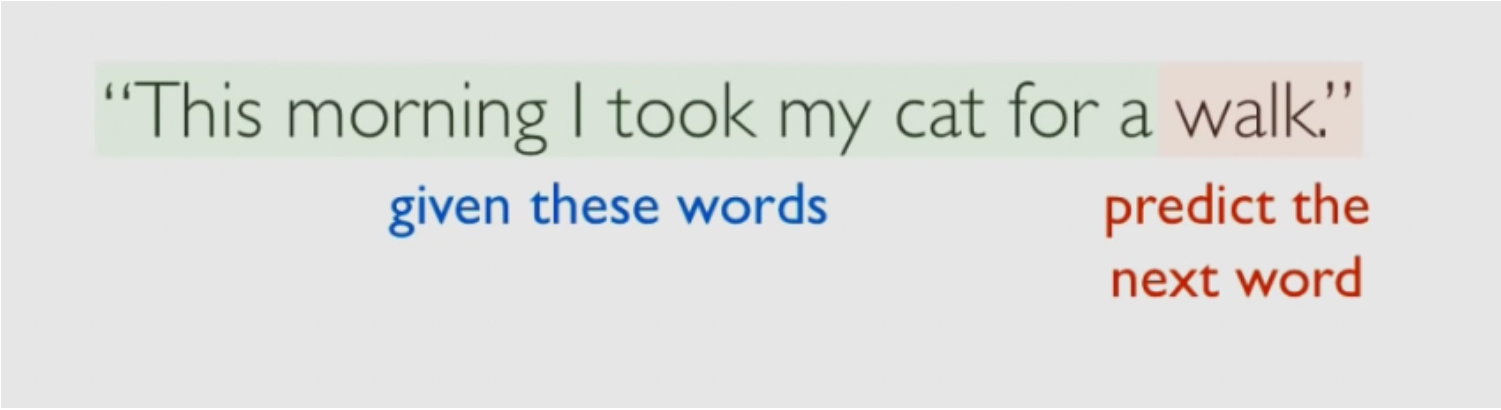


Previous positons help guessing the future

LET'S TAKE A SIMPLE EXAMPLE OF LANGUAGE MODELLING

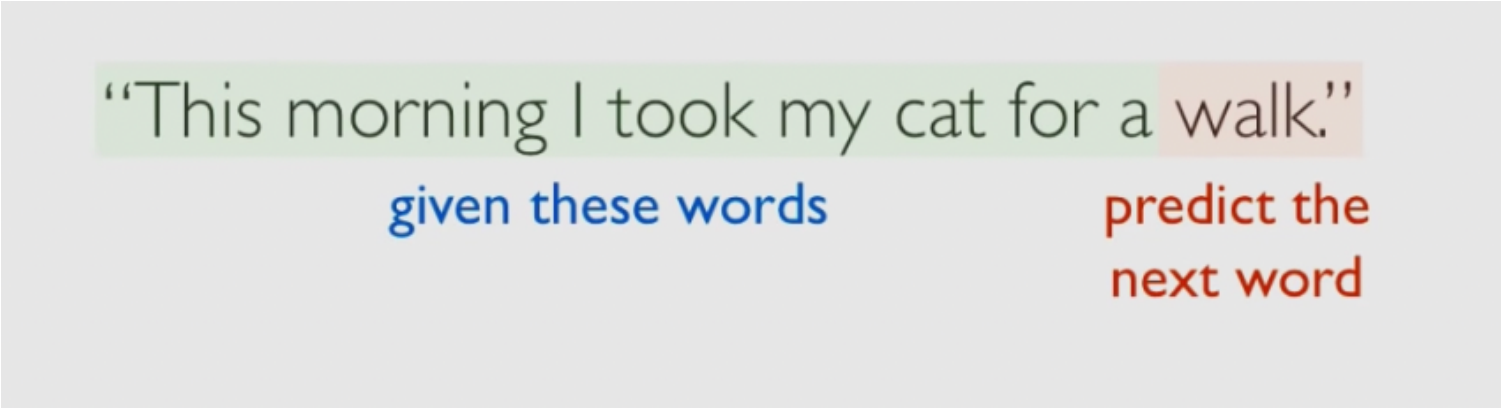


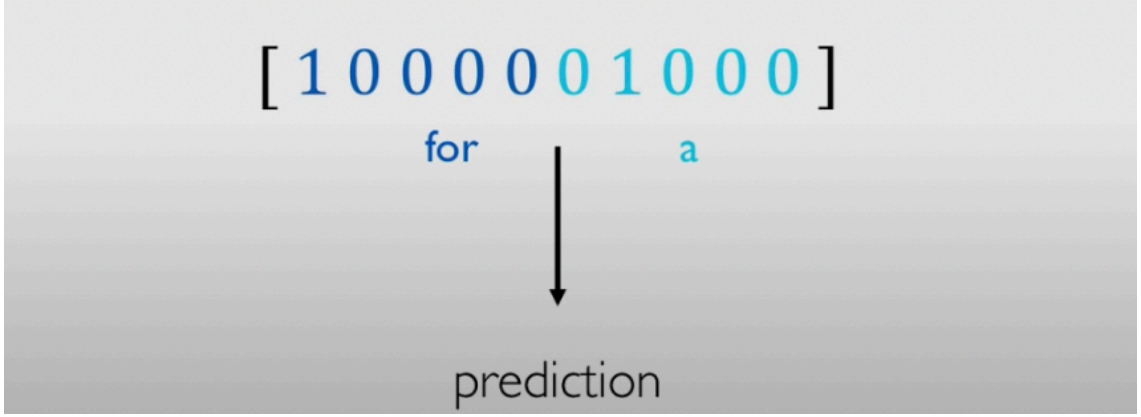
LET'S TAKE A SIMPLE EXAMPLE OF LANGUAGE MODELLING



Normal ANNs and CNNs cannot handle variable length inputs …

WE COULD SIMPLY USE A FIXED WINDOW…

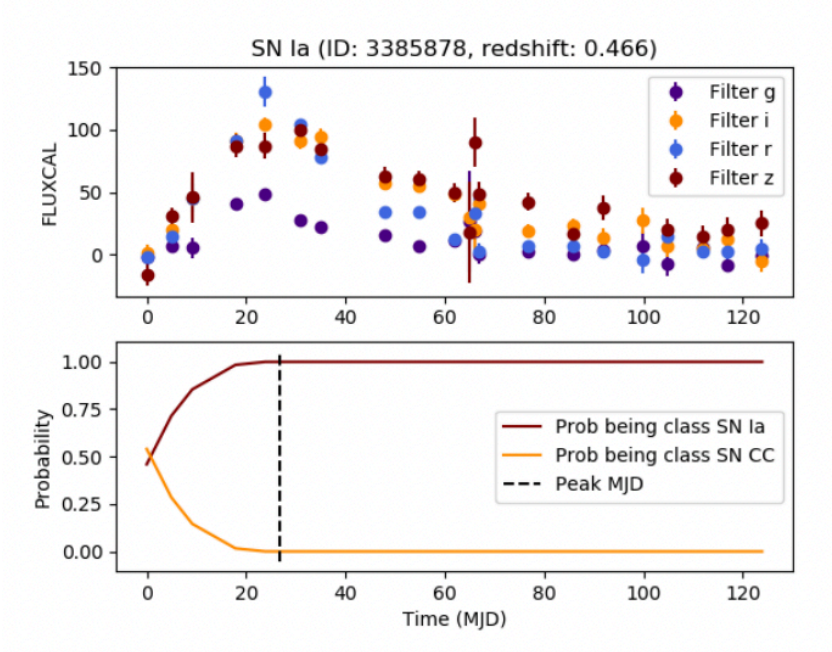




HOWEVER THIS CANNOT HANDLE LONG TERM MEMORY

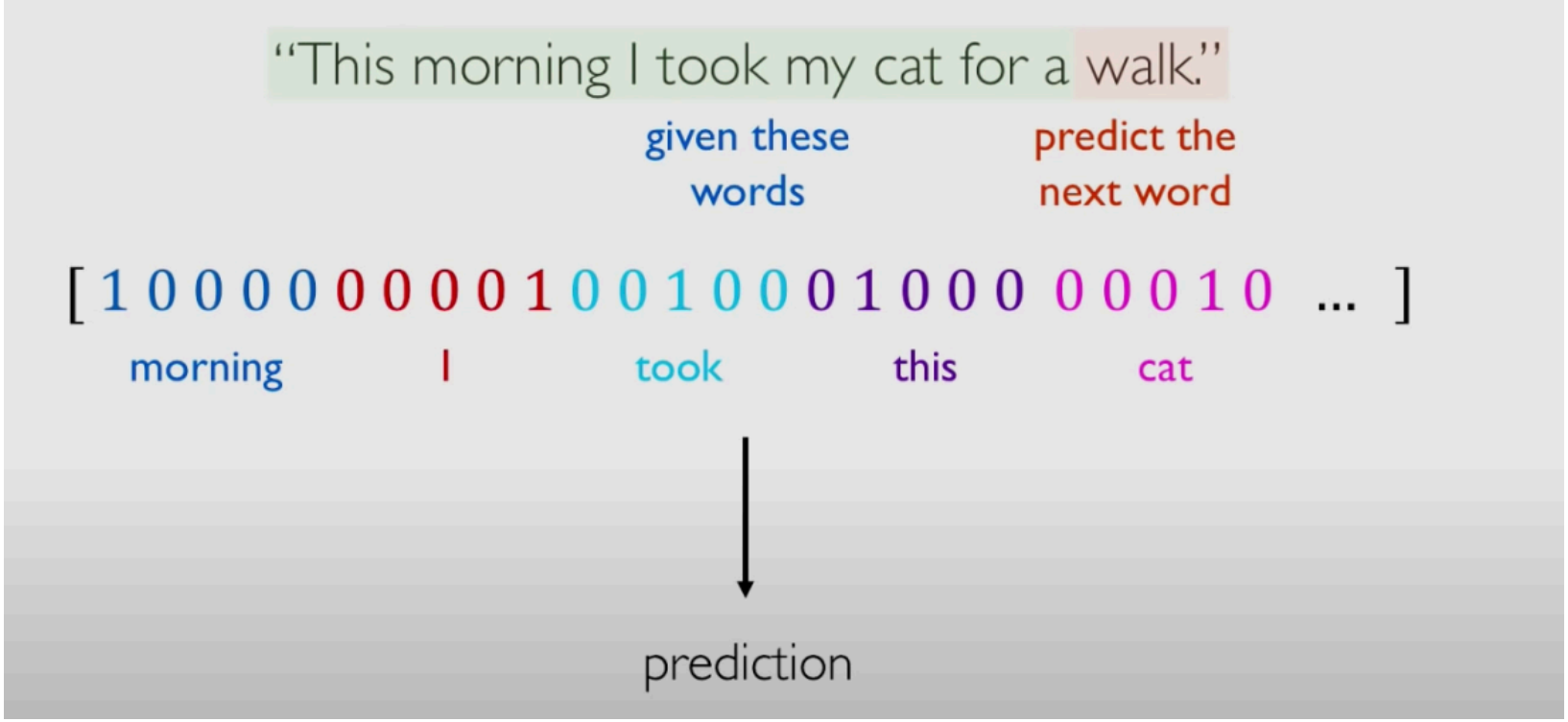
"**France** is where I grew up, but I now live in Boston. I speak fluent ____."

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 (RNNs)

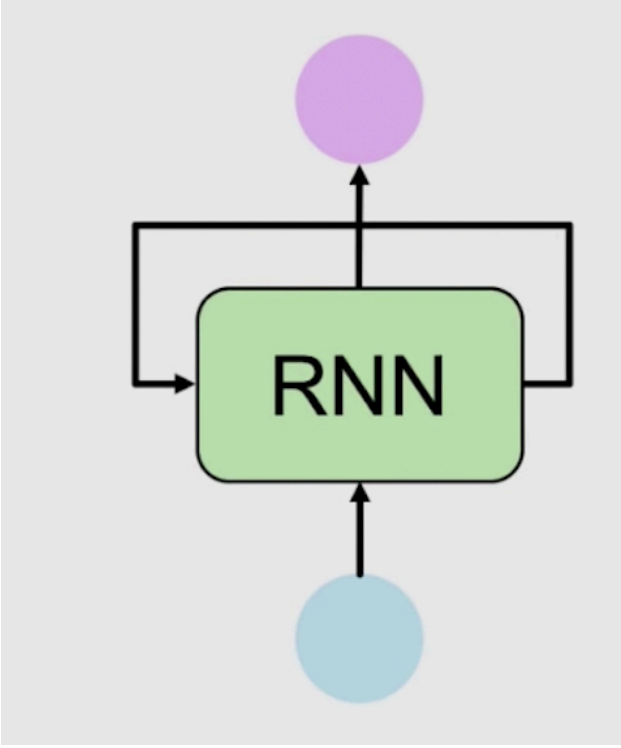
OUR WISH LIST:

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

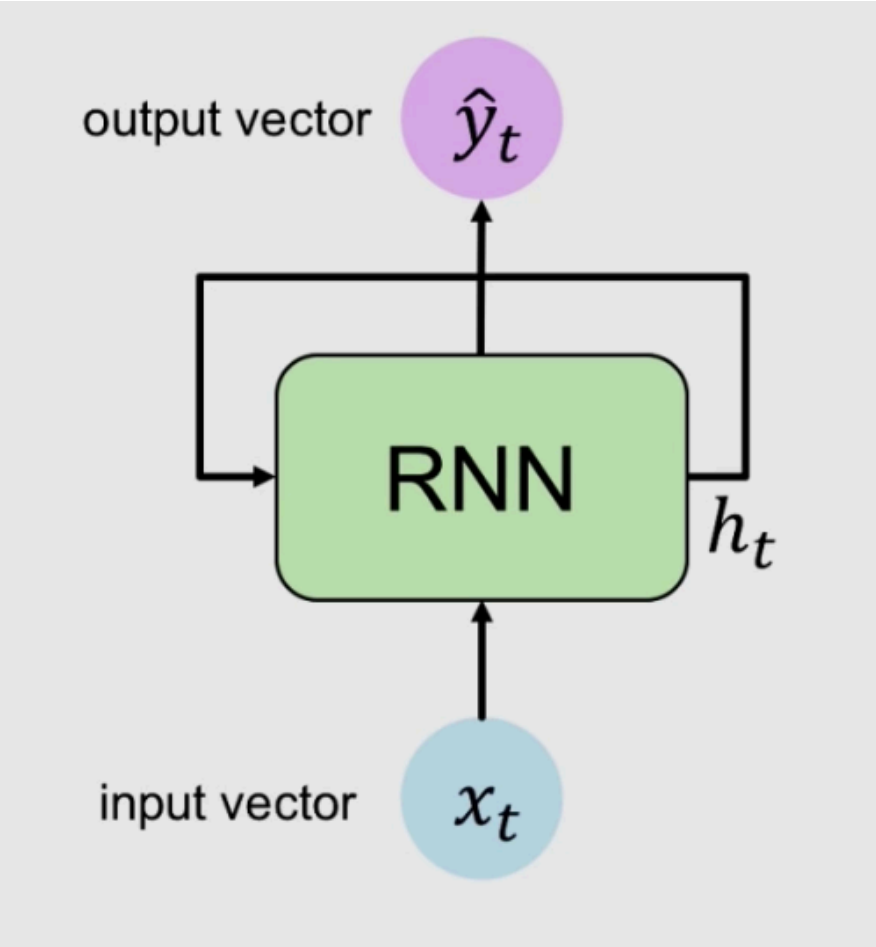
OUR WISH LIST:

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



The RNN Block



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

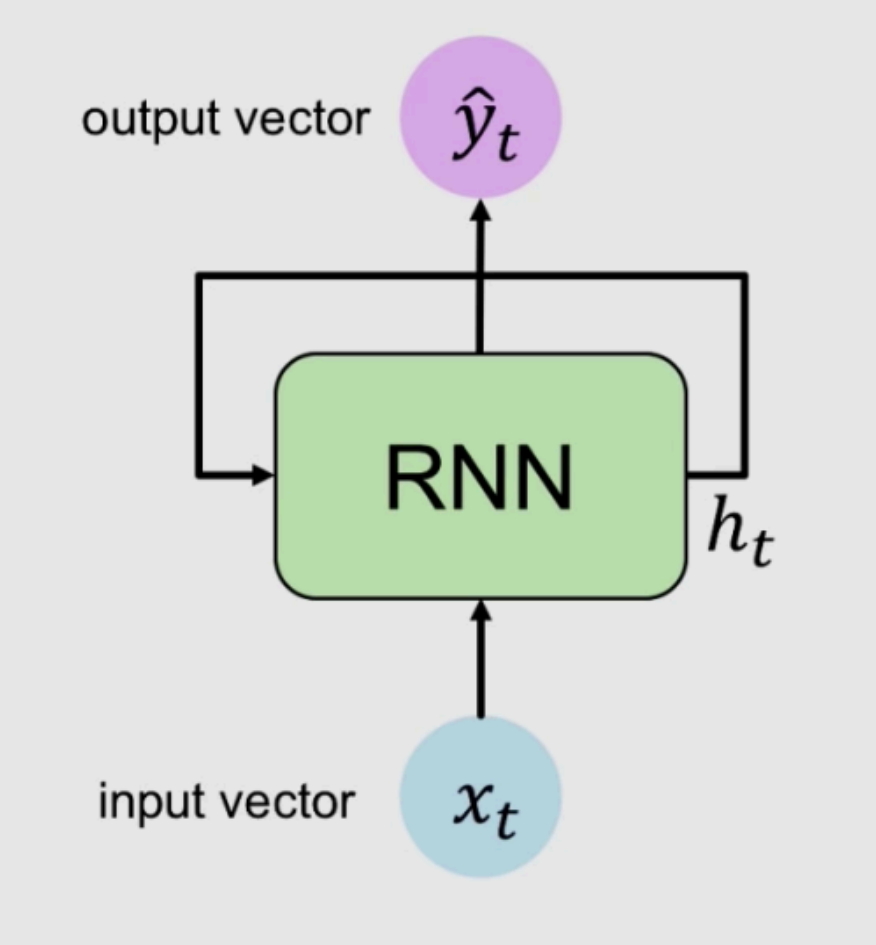
cell state

function parameterized by W

old state

input vector at time step *t*

The RNN Block



Output Vector

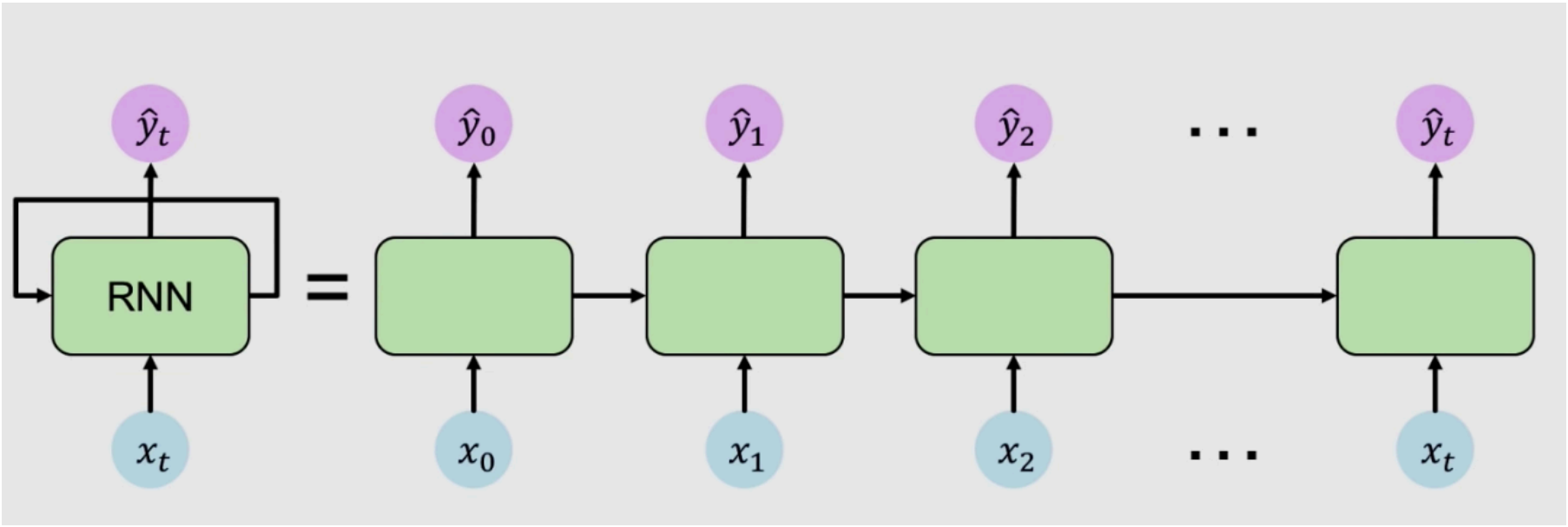
$$\hat{y}_t = \boldsymbol{W}_{\boldsymbol{hy}}^{\boldsymbol{T}} h_t$$

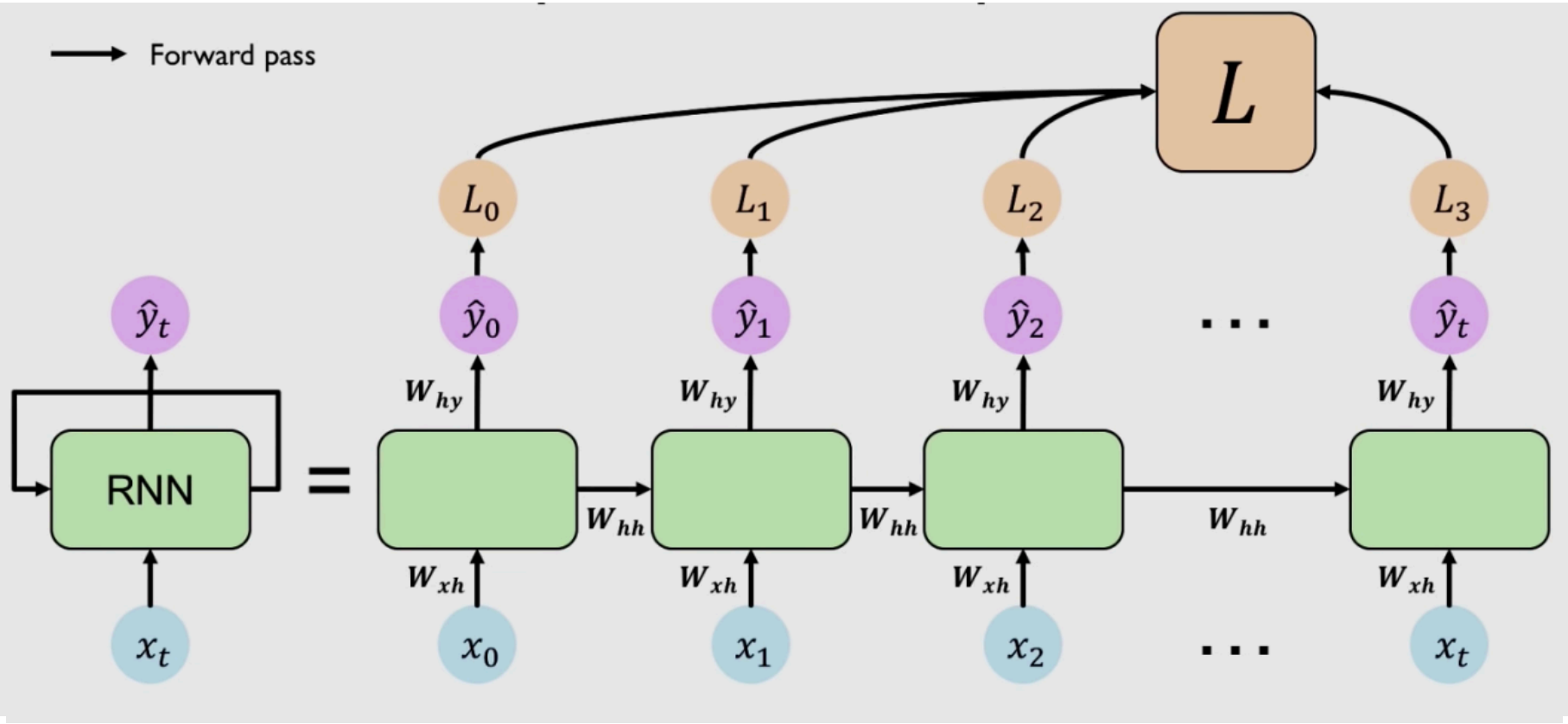
Update Hidden State

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

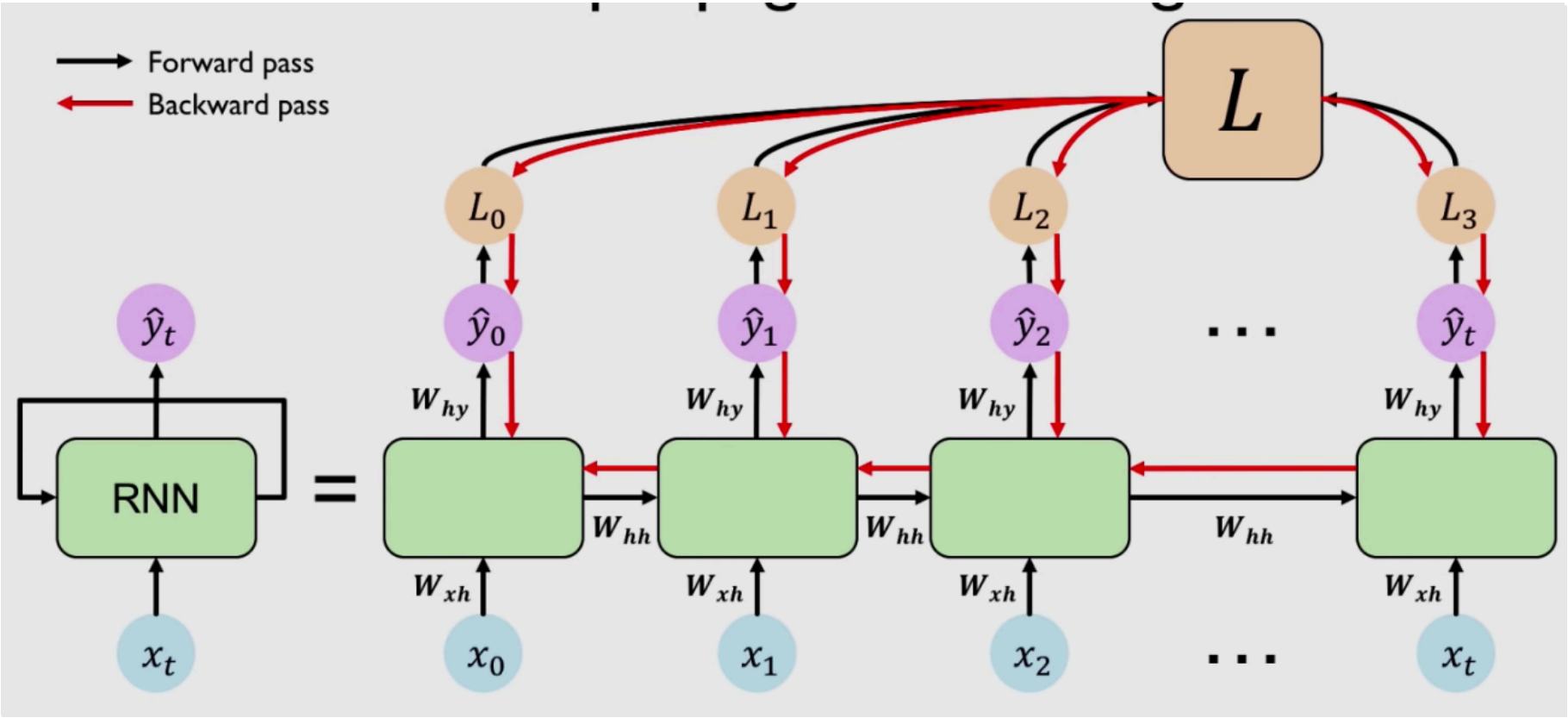
Input Vector

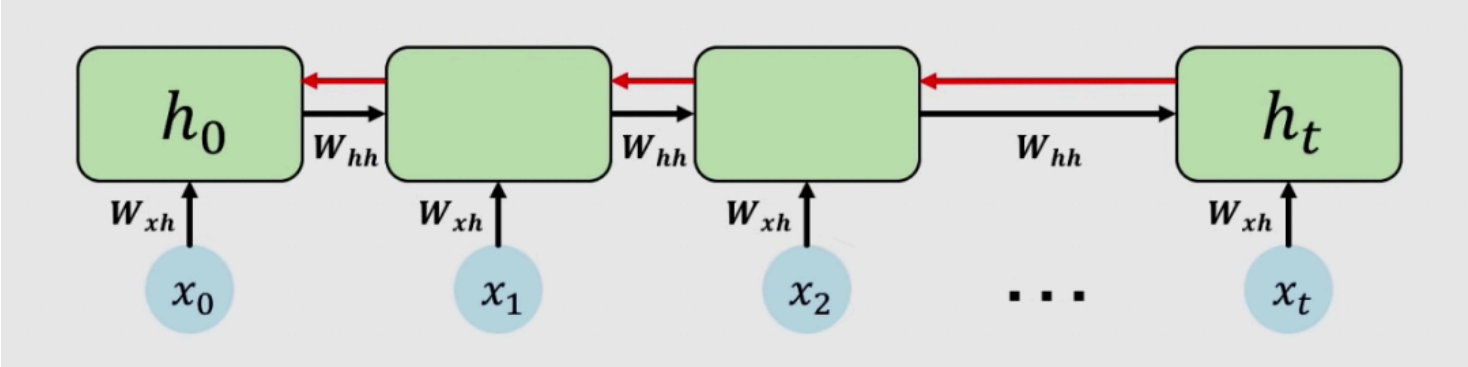
$$x_t$$


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

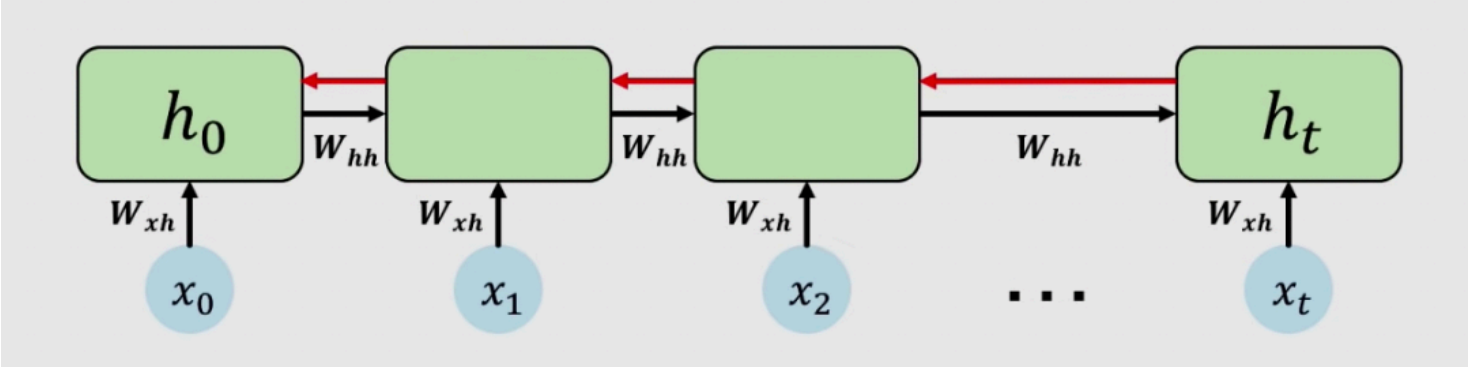

Forward pass
L
L_0
L_1
L_2
L_3
$\hat{y}_t$
$\hat{y}_0$
$\hat{y}_1$
$\hat{y}_2$
$\hat{y}_t$
RNN
=
W_{hy}
W_{hh}
W_{xh}
x_t
x_0
x_1
x_2
x_t

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)

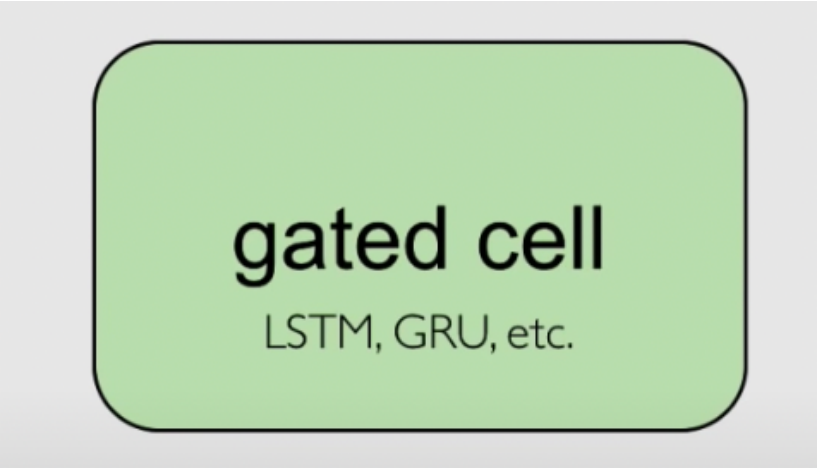


BPTT implies a large amount of weight multiplications: if we want to take into account long term memory, networks become quickly very deep and so are subjected to vanishing and exploding gradients problems (see previous lecture)

ONE CAN USE THE SAME TRICKS THAT WE DISCUSSED FOR DEEP CNNs (WEIGTH INITIALISATION, RELU ACTIVATION FUNCTION …)

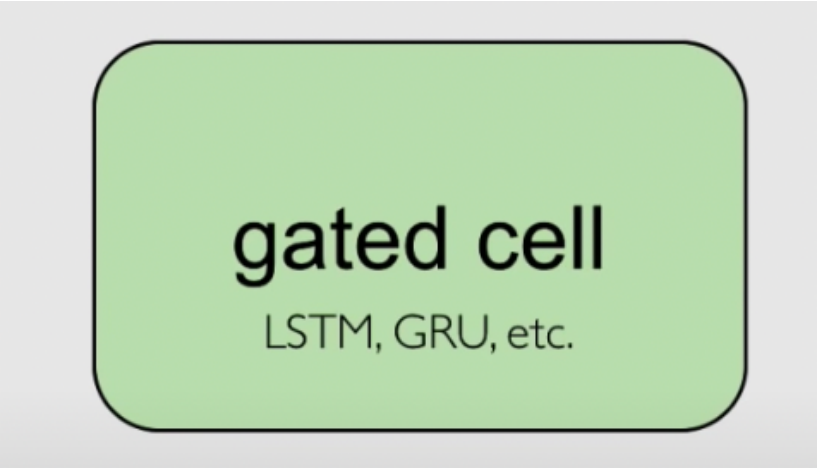
ANOTHER SPECIFIC SOLUTION: GATED CELLS

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



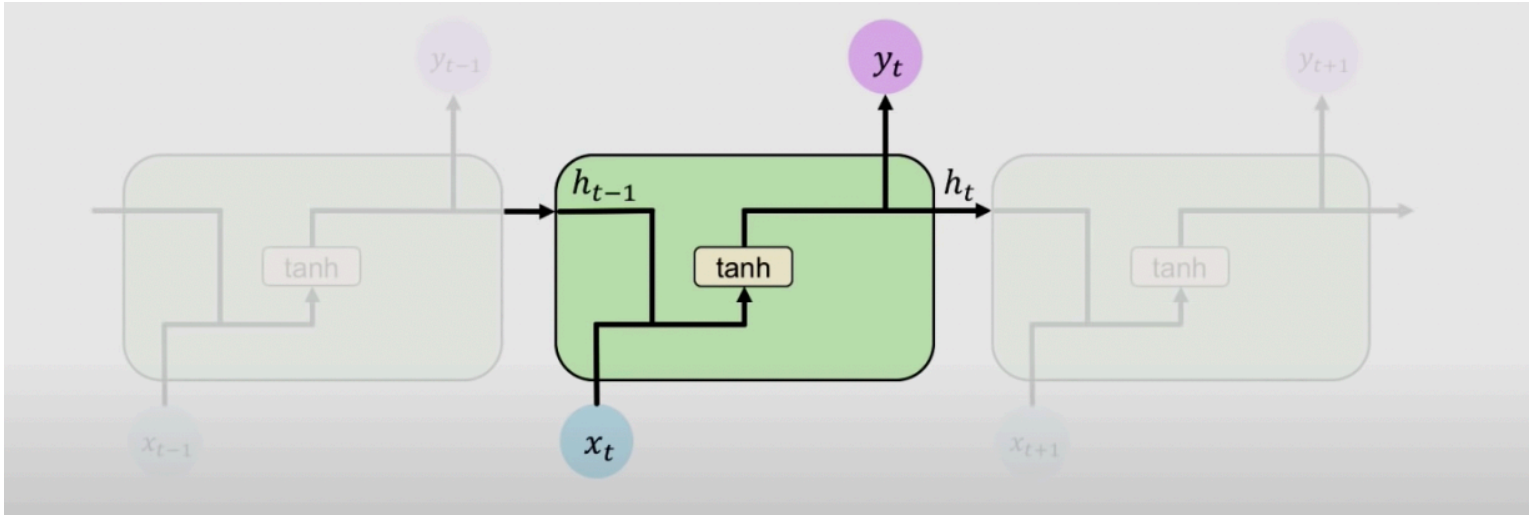
ANOTHER SPECIFIC SOLUTION: GATED CELLS

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

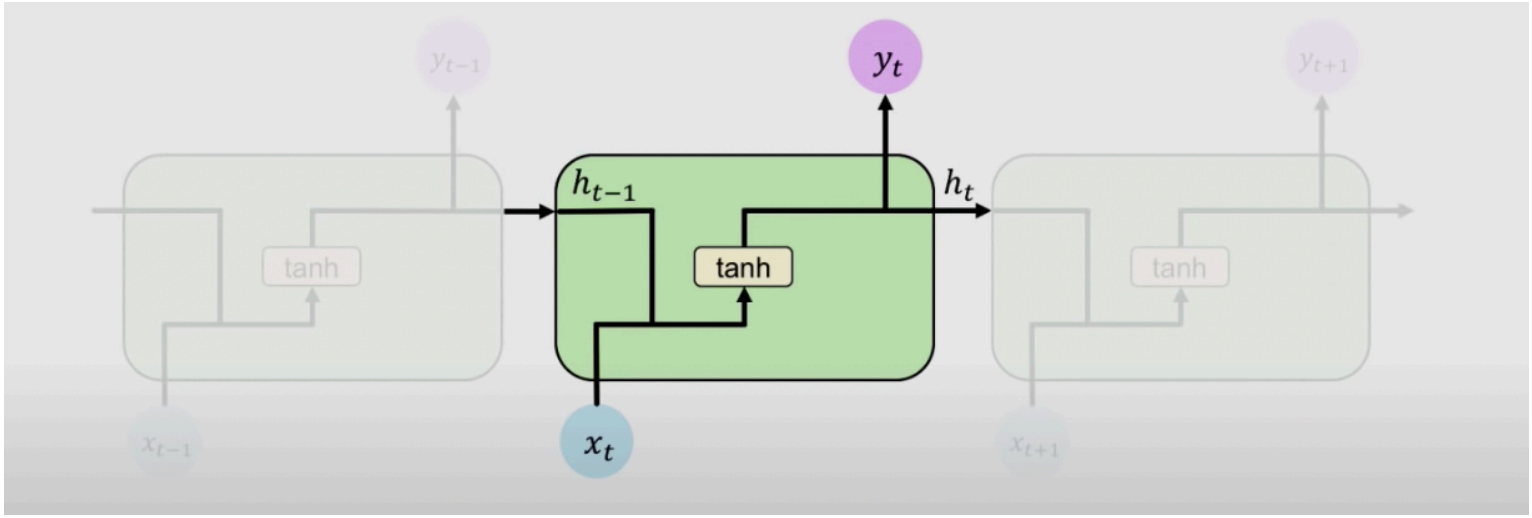


Long Short Term Memory (LSTMs) networks are an example of this

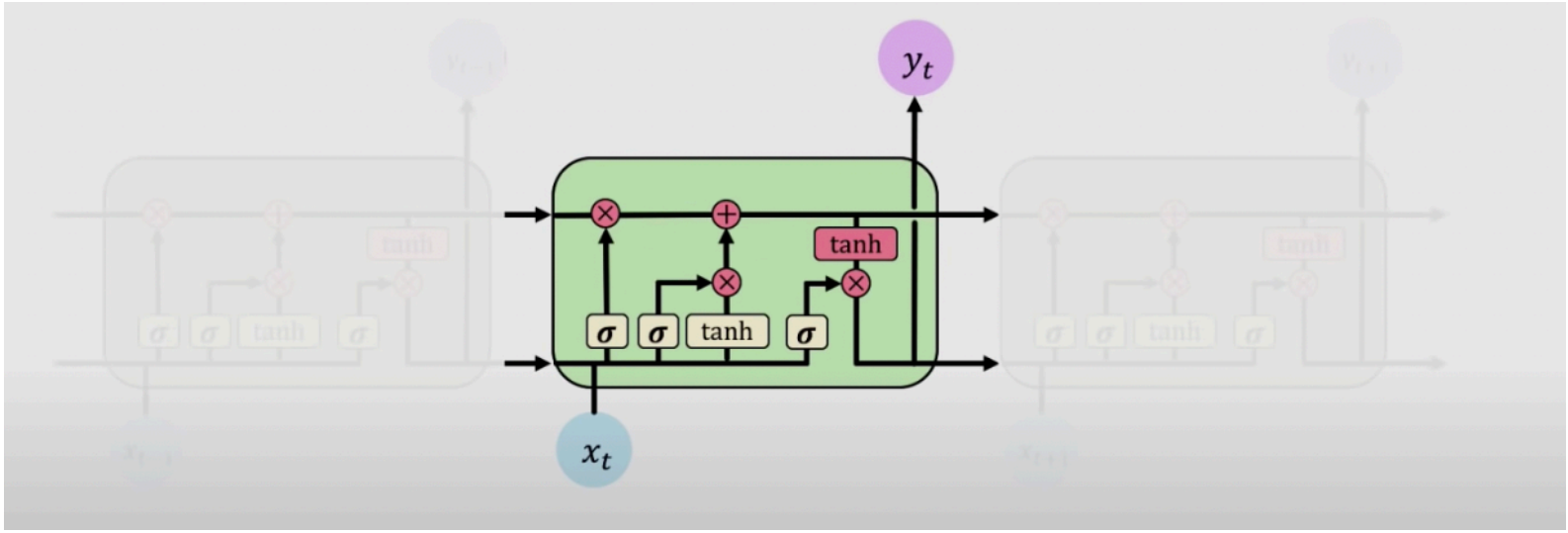
Standard RNN



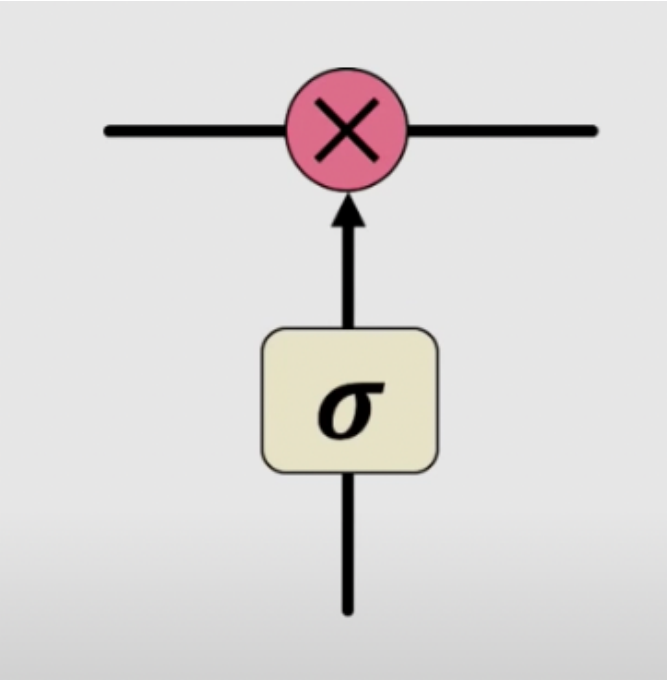
Standard RNN



LSTM unit

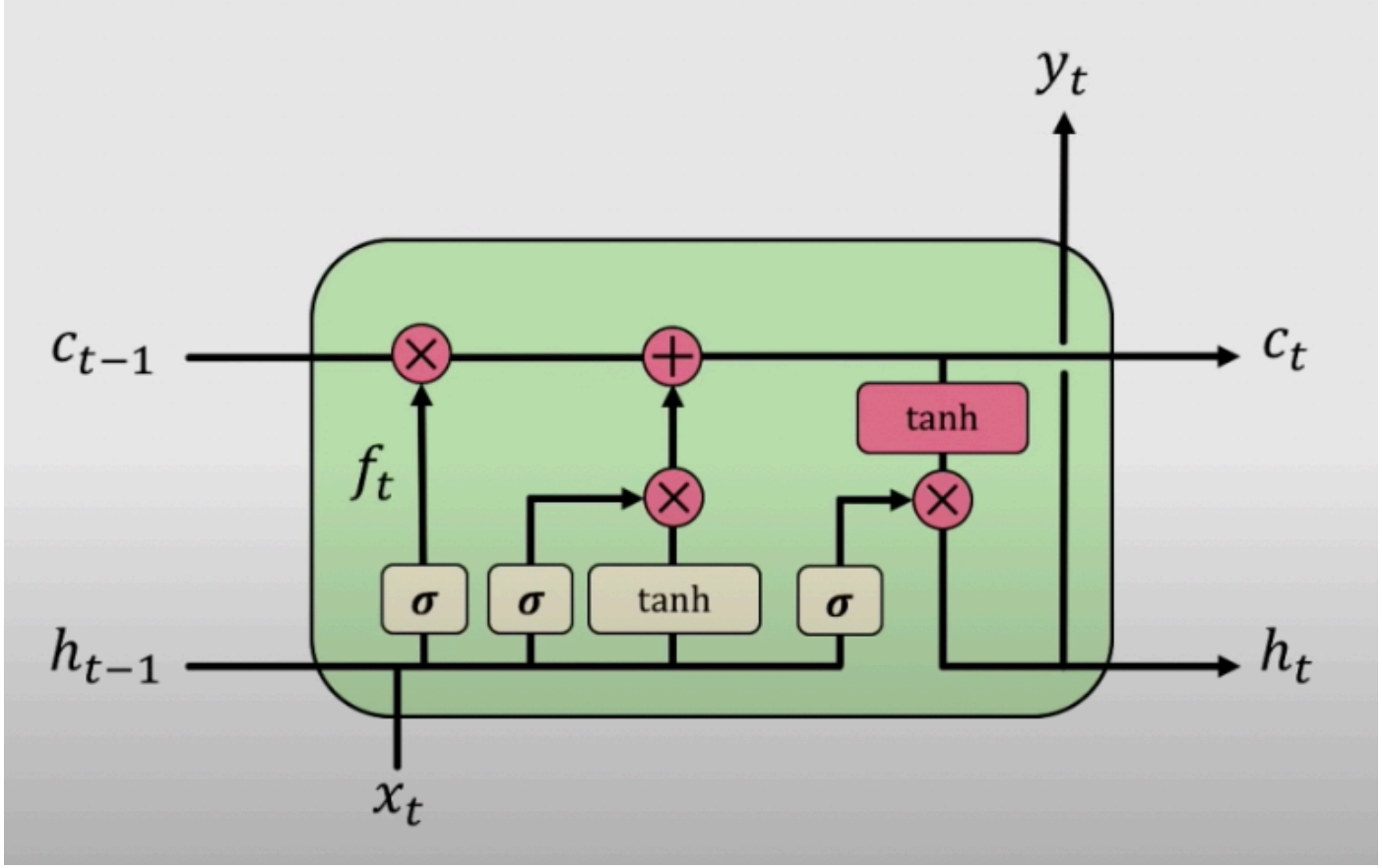


The key building block are the so-called gates



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

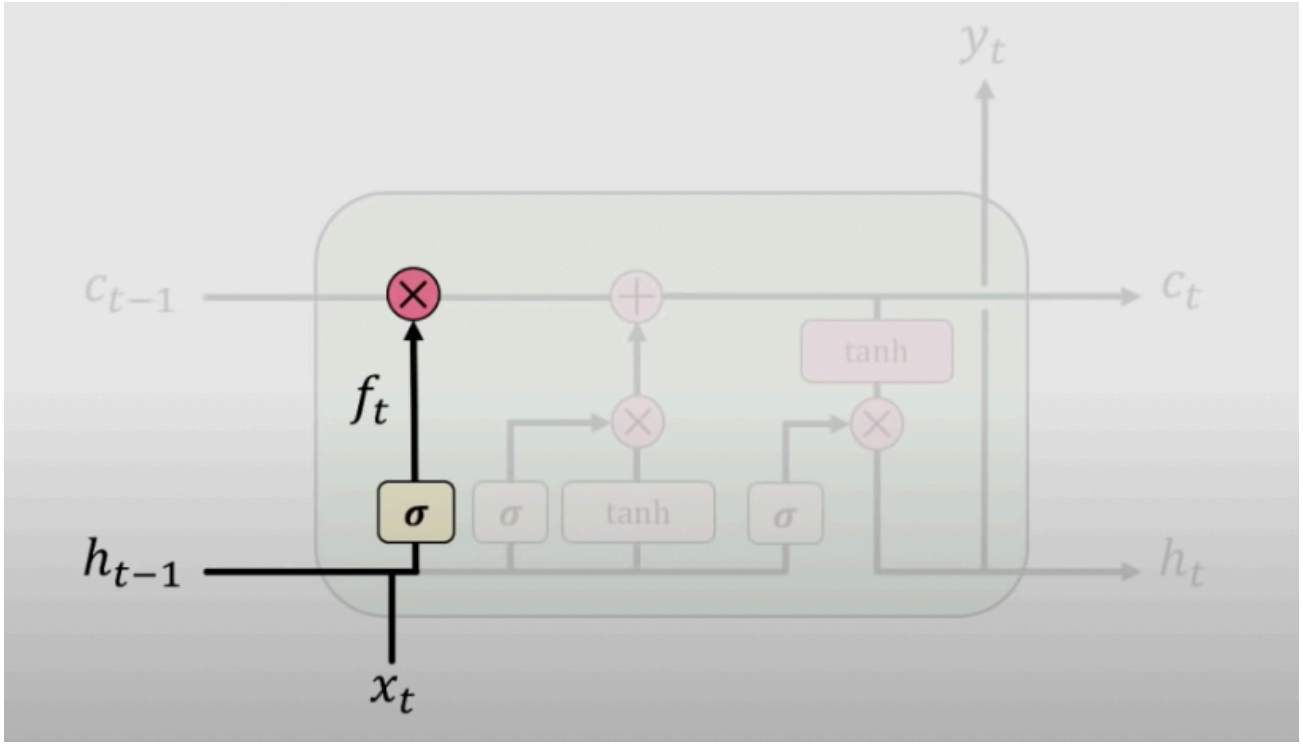
Long Short Term Memory (LSTMs)



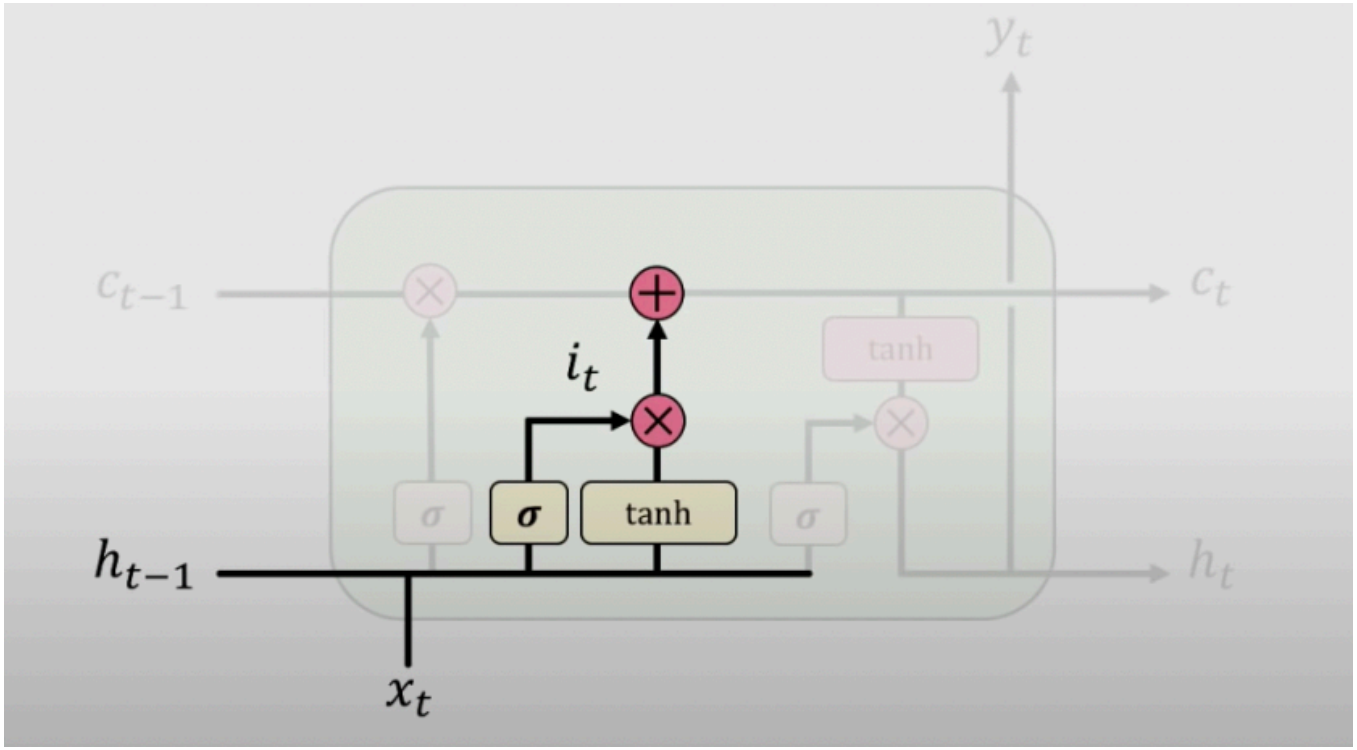
4 main steps:

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

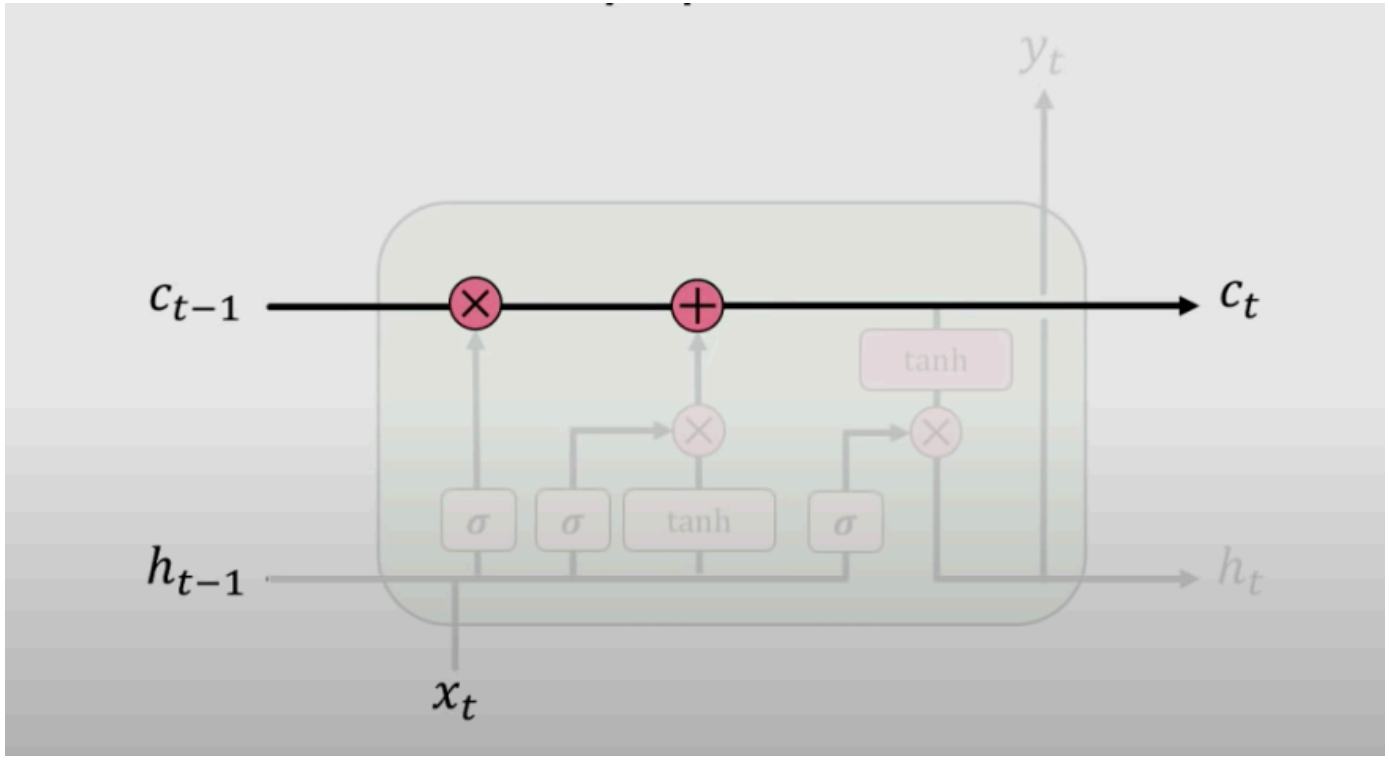
STEP 1: FORGET



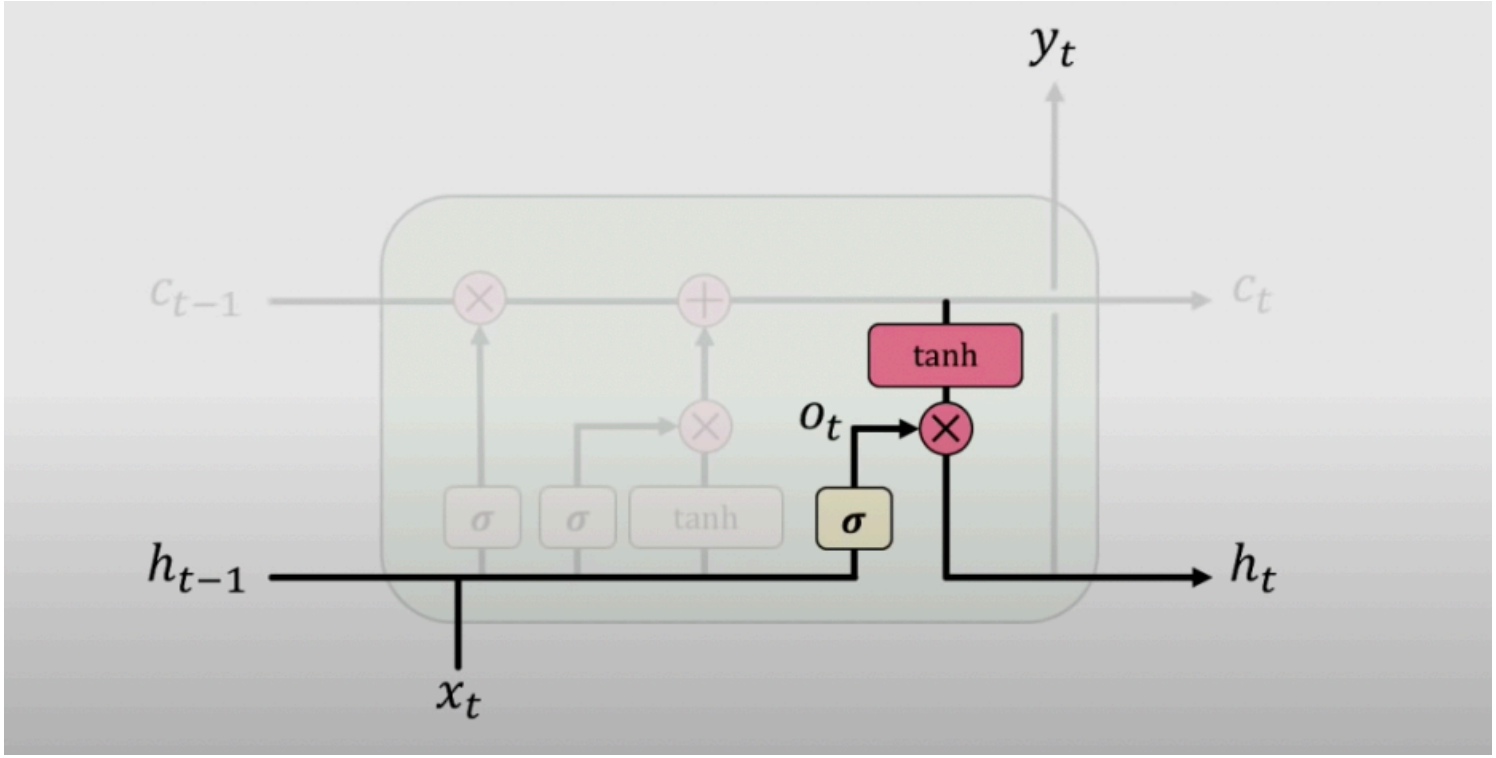
STEP 2: STORE



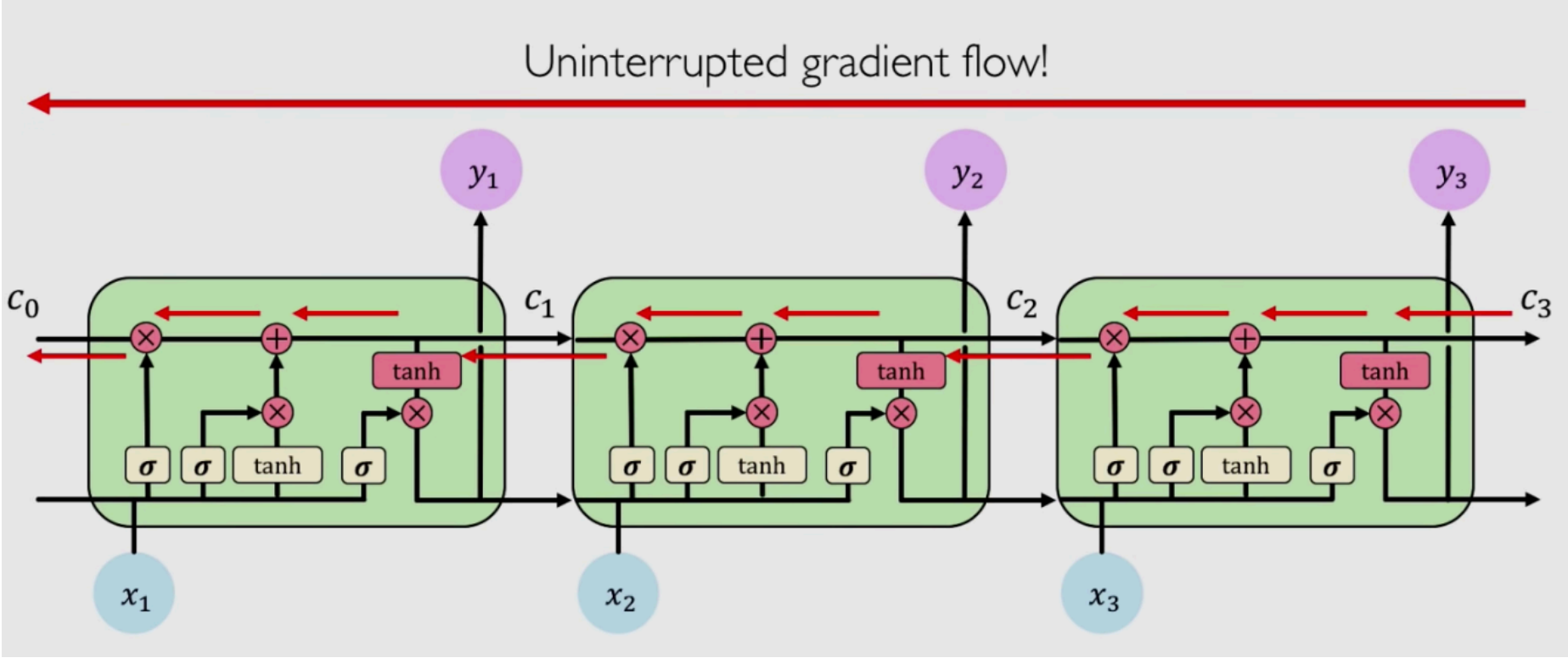
STEP 3: UPDATE



STEP 4: OUTPUT



LSTMs help mitigating the vanishing gradient problem



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. Backropagation through time with uninterrupted gradient flow