Convolutional LSTM for image processing

Armando Vieira

Summary

what is a recurrent network and why it matters

What is a LSTM and a CNN

The convolutional LSTM

How to implement it in Keras & Tensorflow

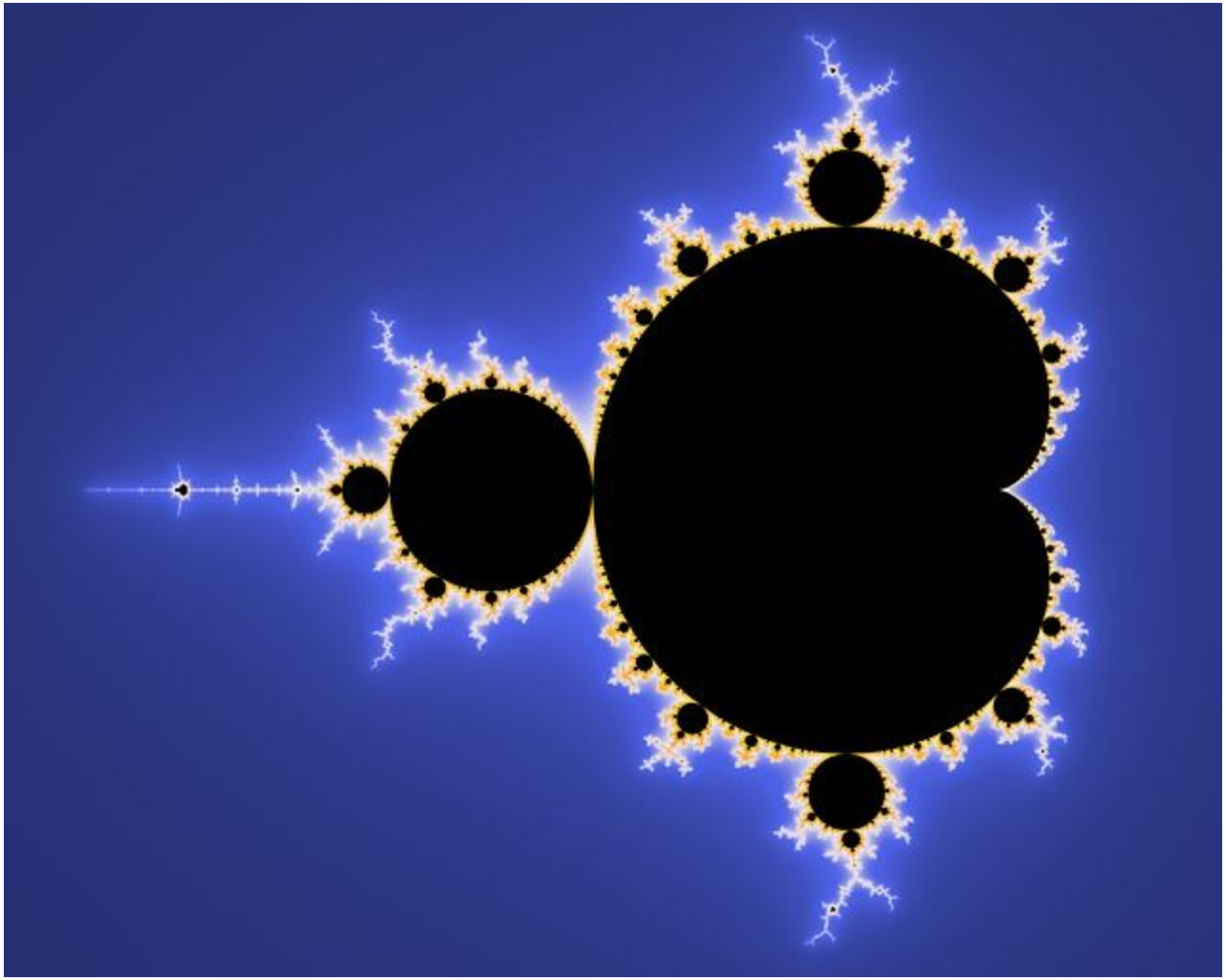
Supervised and unsupervised mode

Applications: Weather forecasting, Self-driving cars, Image segmentation

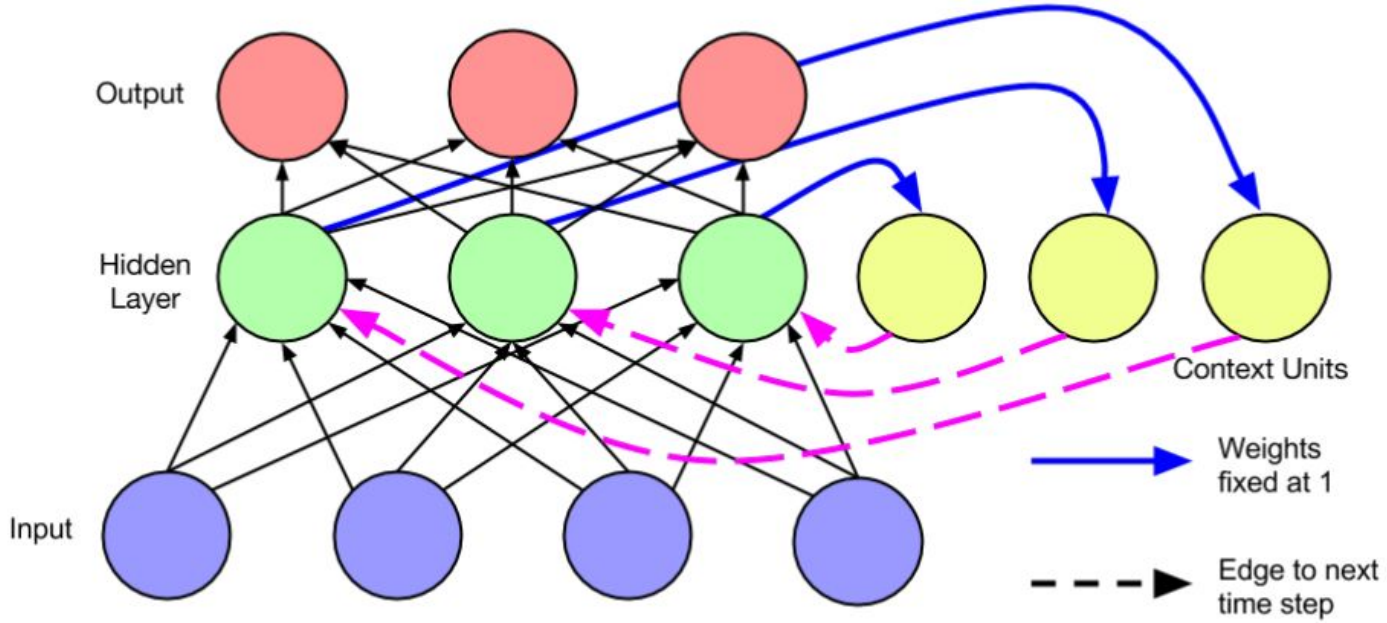
Why recursivity matters and why it’s hard to deal?

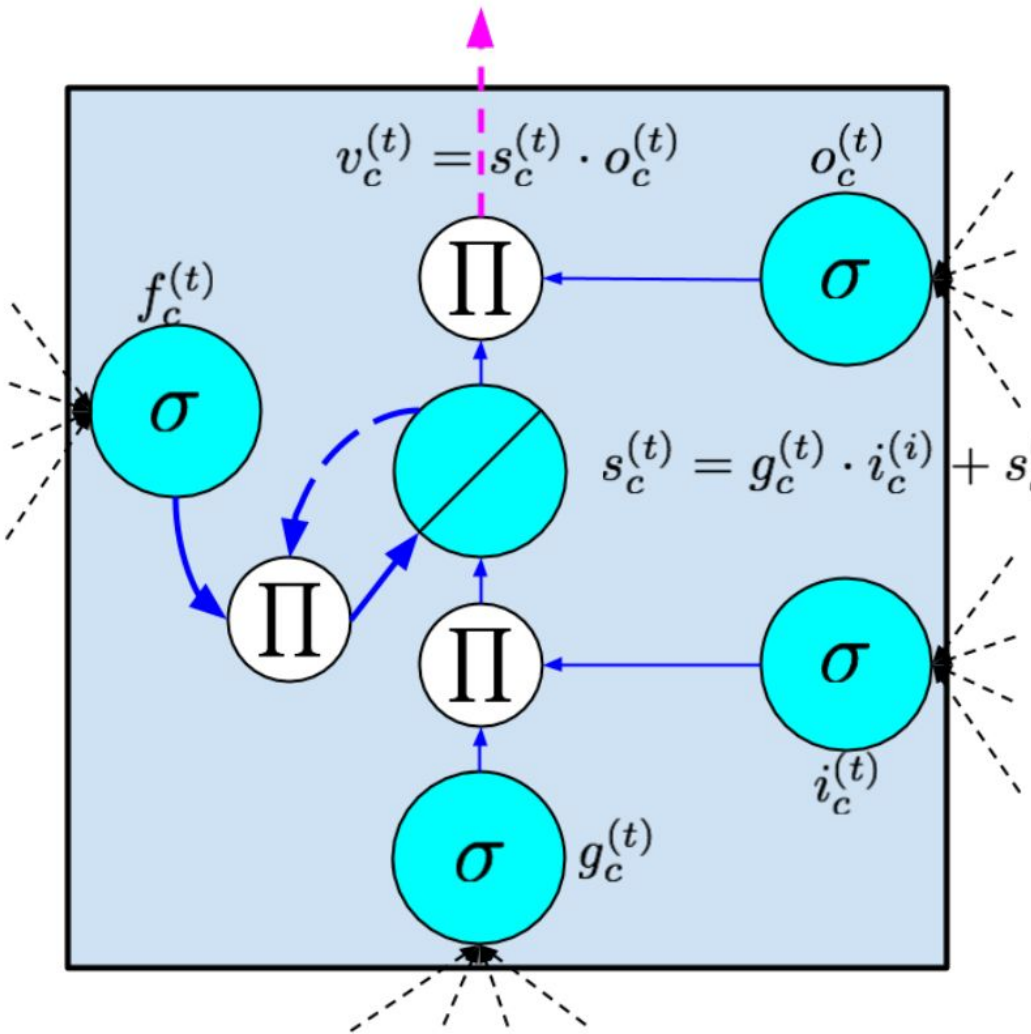
$$f(x) = 3.5\ x\ (1 - x)$$

$$f(f(f(f(x)))) = ?$$



What is a Recurrent Network?

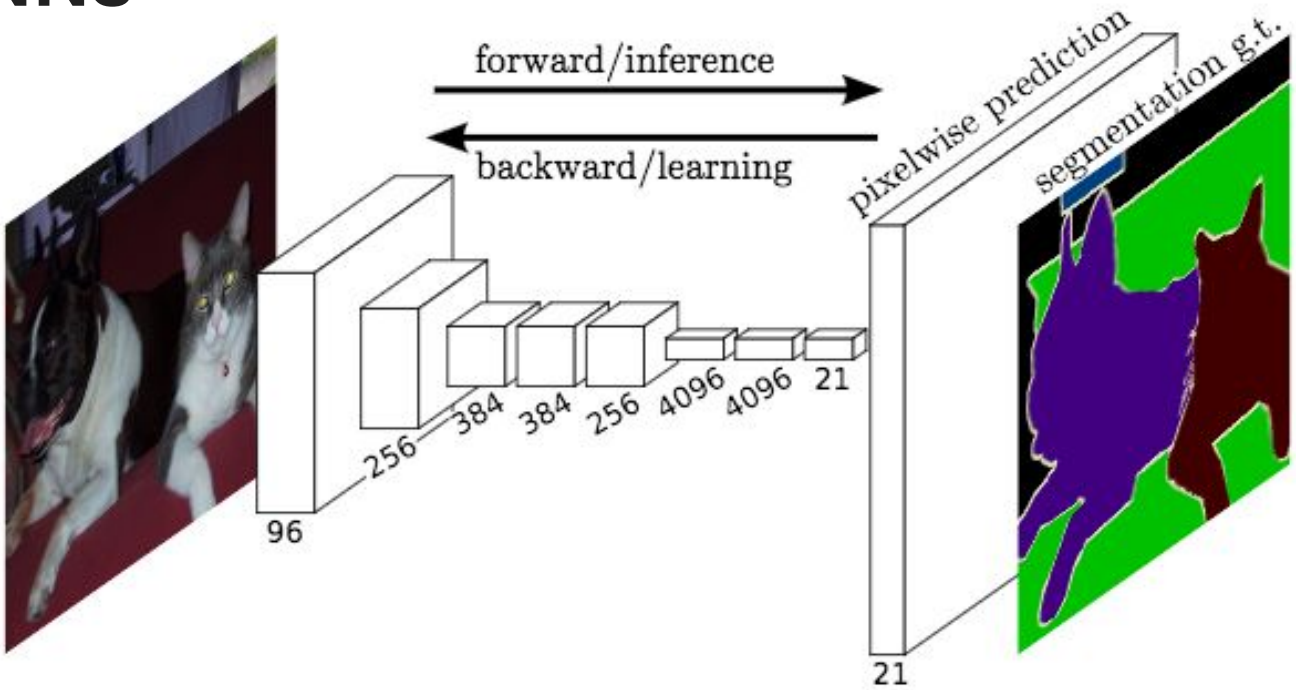


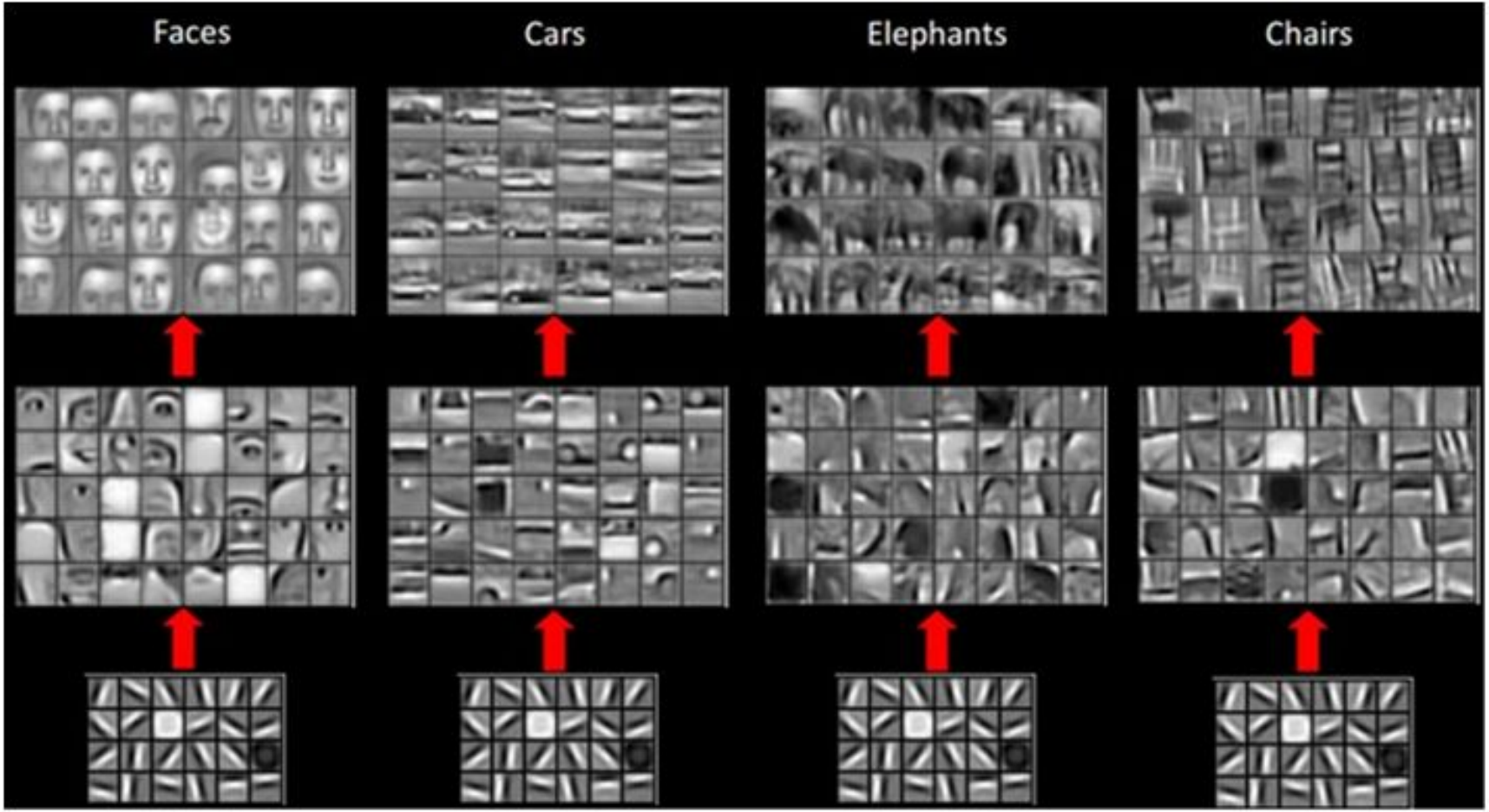


Schmidhuber 1997

LSTM

FCNNs


forward/inference
backward/learning
pixelwise prediction
segmentation g.t.
96
256
384
384
256
4096
4096
21
21


Faces
Cars
Elephants
Chairs

Convolutional LSTM + FCN

Main idea: FCN Conv. Layers are replaced with ConvLSTM layers

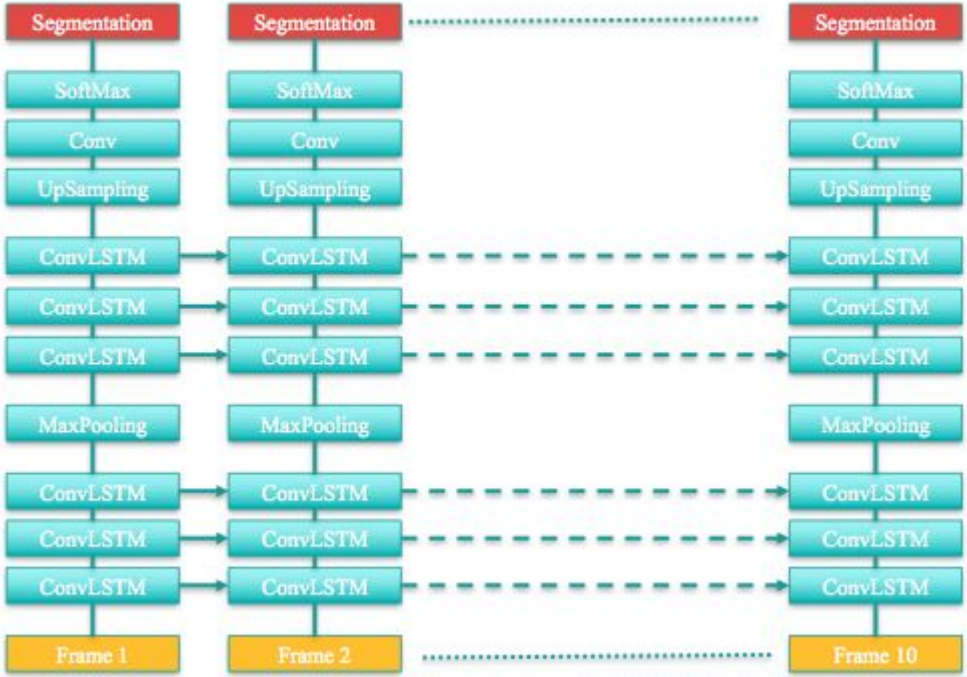
ConvLSTM is similar to fully connected LSTM

Network architecture

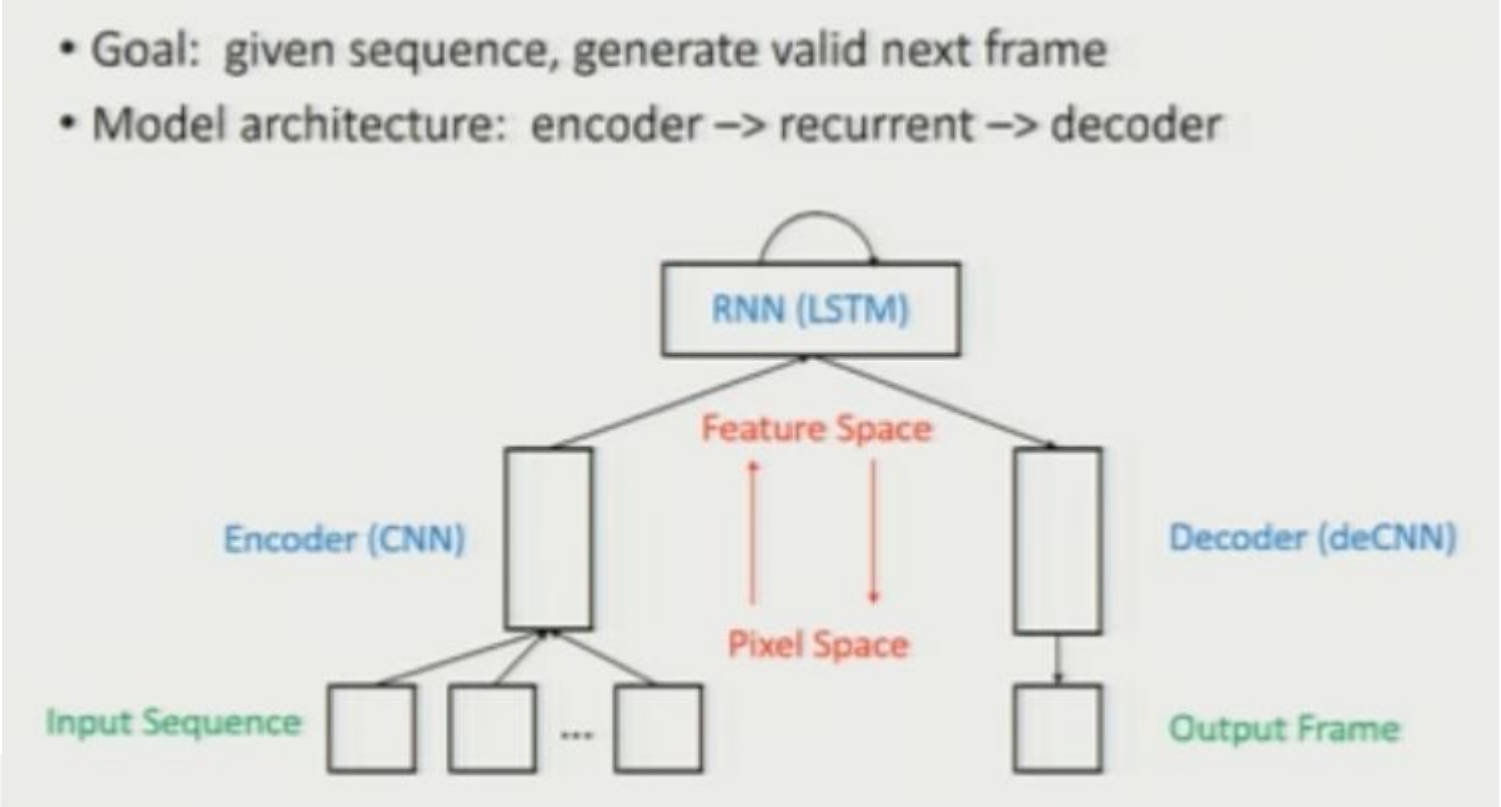
Temoral dynamic model at different spatial scales:

Recurrent convolutional LSTM blocks forward outputs and inner states to future predictions.

Network architecture



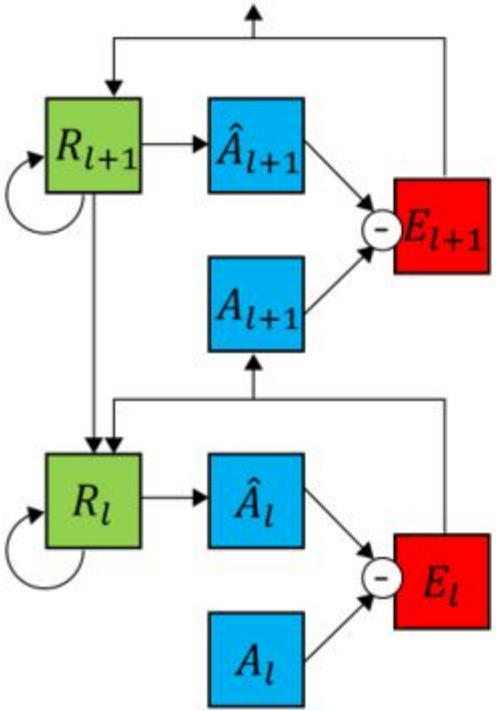
Unsupervised: learning representation of the world

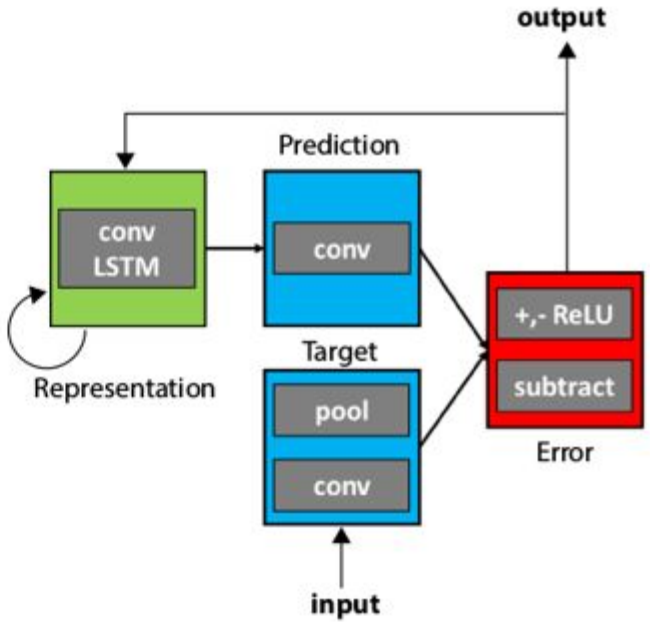


Two models

Non supervised (next frame prediction)

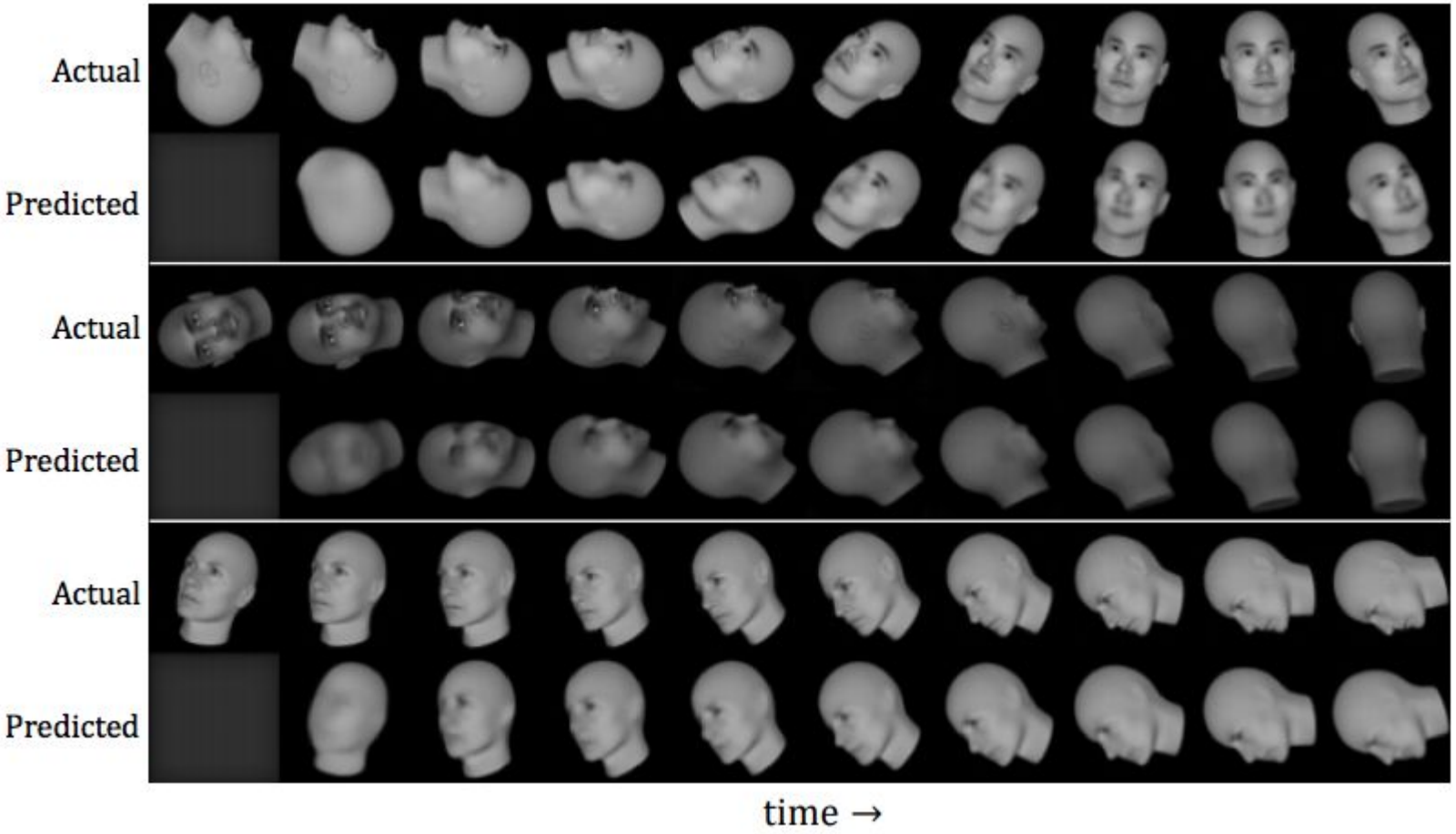
Supervised (sequence to sequence)





Code (Keras)

Supervised / Unsupervised


Actual
Predicted
Actual
Predicted
Actual
Predicted
time →

Self driving cars: Kitti dataset

https://coxlab.github.io/prednet/
https://coxlab.github.io/prednet/prednet_animation.html

Bouncing balls:
learning physics from scratch

https://www.youtube.com/watch?v=RjZ1VKYyHhs

Weather forecasting

$$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{dy}+w\frac{\partial u}{\partial z}\right)=$$

$$\rho g_x-\frac{\partial p}{\partial x}+\frac{\partial}{\partial x}\left[2\mu\frac{\partial u}{\partial x}+\lambda\nabla\cdot\mathbf{V}\right]+\frac{\partial}{\partial y}\left[\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right]+\frac{\partial}{\partial z}\left[\mu\left(\frac{\partial w}{\partial x}+\frac{\partial u}{\partial z}\right)\right]$$

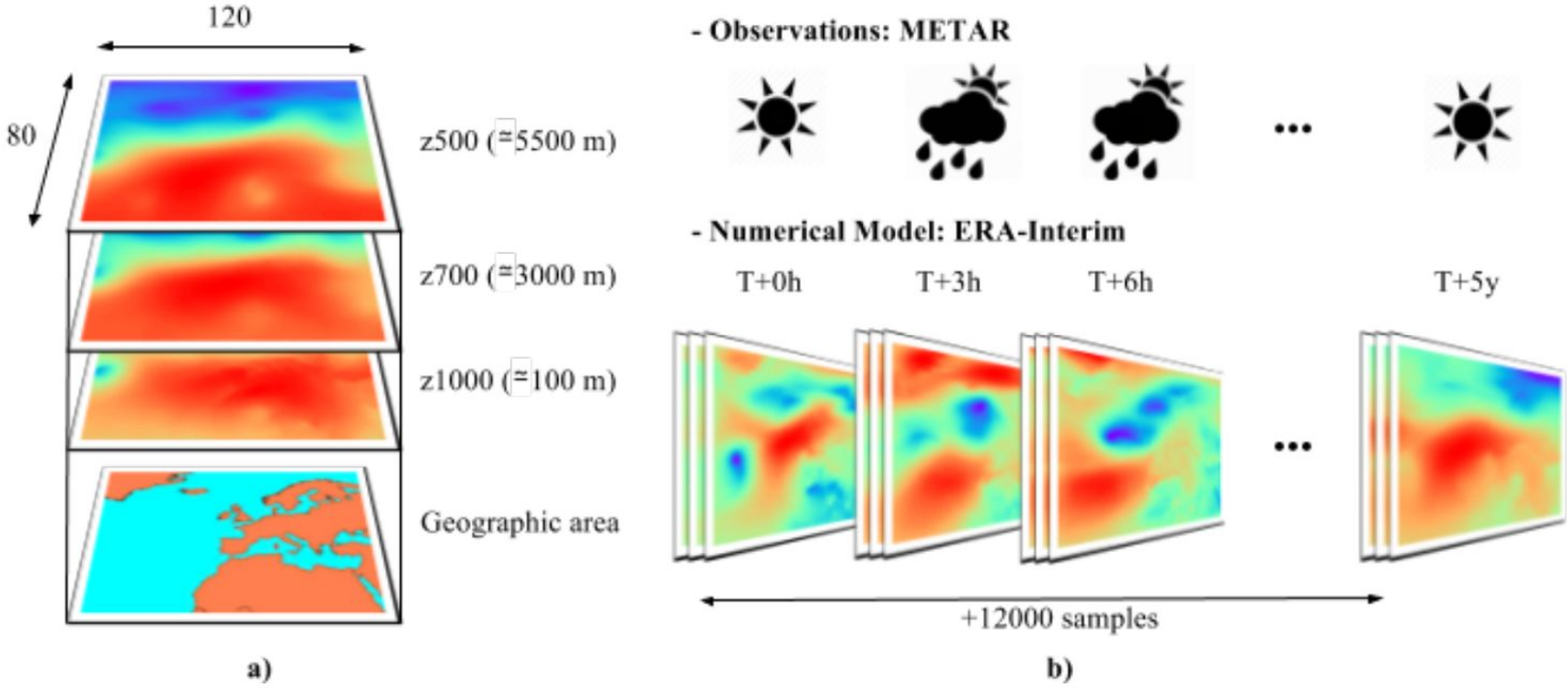
$$\rho\left(\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{dy}+w\frac{\partial v}{\partial z}\right)=$$

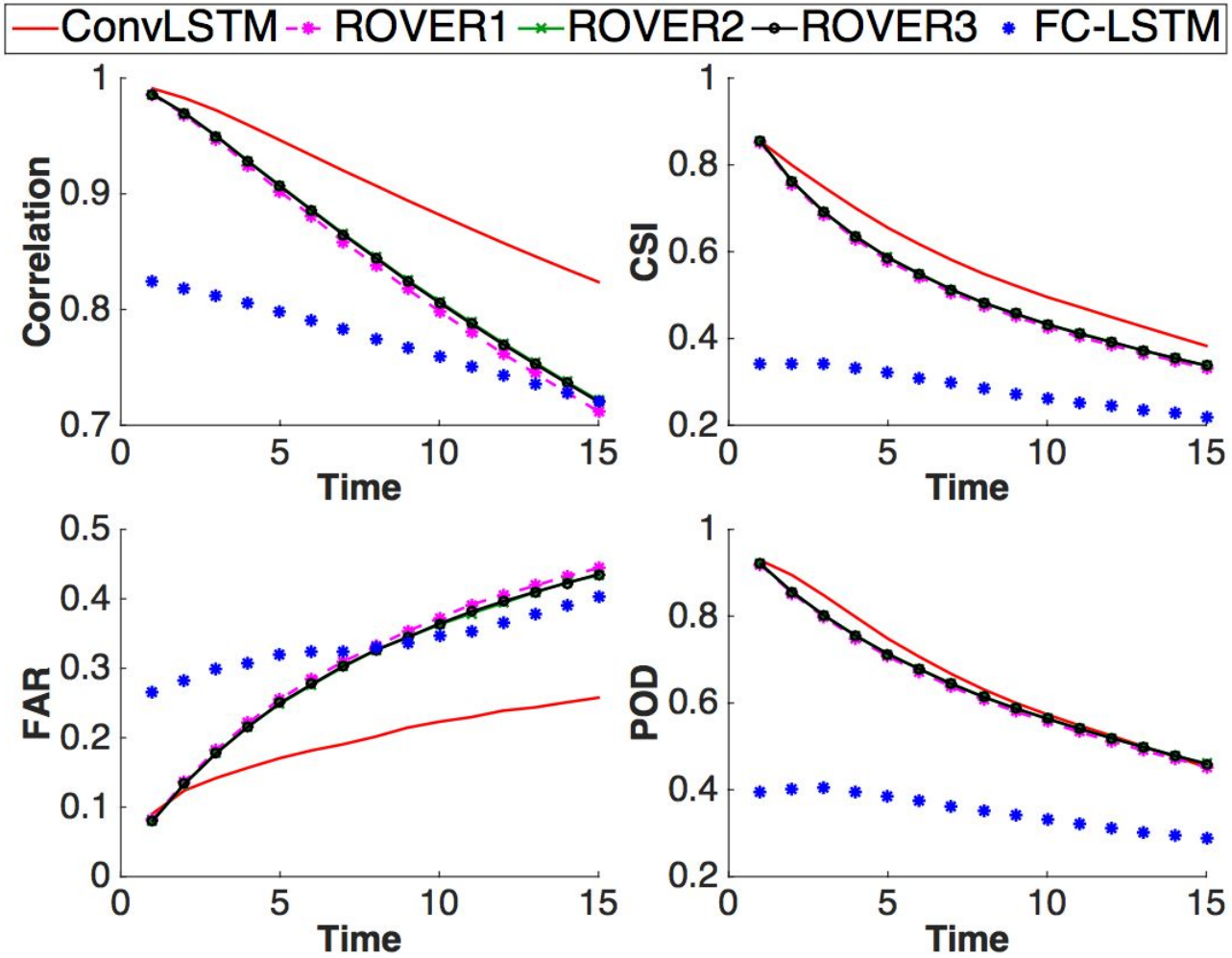
$$\rho g_y-\frac{\partial p}{\partial y}+\frac{\partial}{\partial y}\left[2\mu\frac{\partial v}{\partial y}+\lambda\nabla\cdot\mathbf{V}\right]+\frac{\partial}{\partial z}\left[\mu\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)\right]+\frac{\partial}{\partial x}\left[\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right]$$

$$\rho\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{dy}+w\frac{\partial w}{\partial z}\right)=$$

$$\rho g_z-\frac{\partial p}{\partial z}+\frac{\partial}{\partial z}\left[2\mu\frac{\partial w}{\partial z}+\lambda\nabla\cdot\mathbf{V}\right]+\frac{\partial}{\partial x}\left[\mu\left(\frac{\partial w}{\partial x}+\frac{\partial u}{\partial z}\right)\right]+\frac{\partial}{\partial y}\left[\mu\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)\right]$$

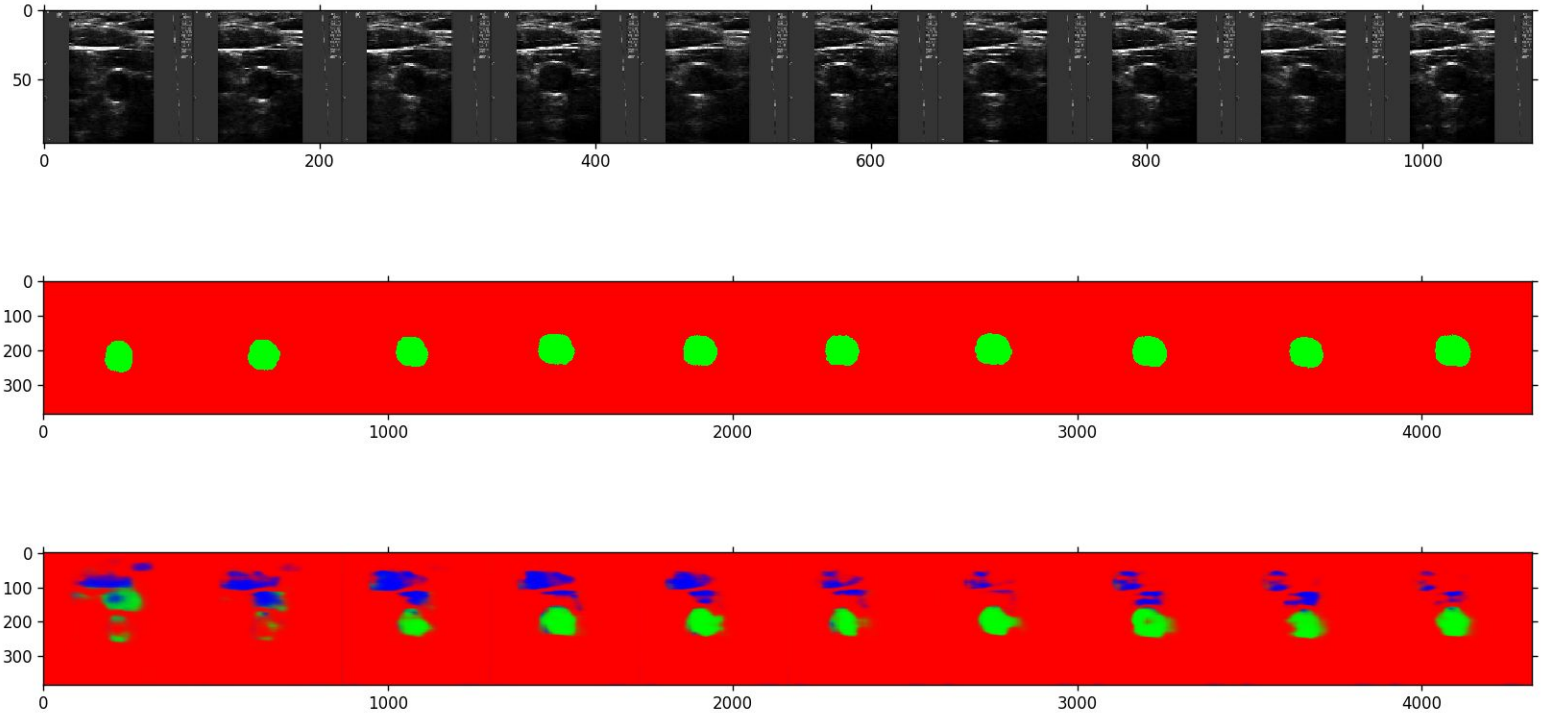
As a convolutional LSTM

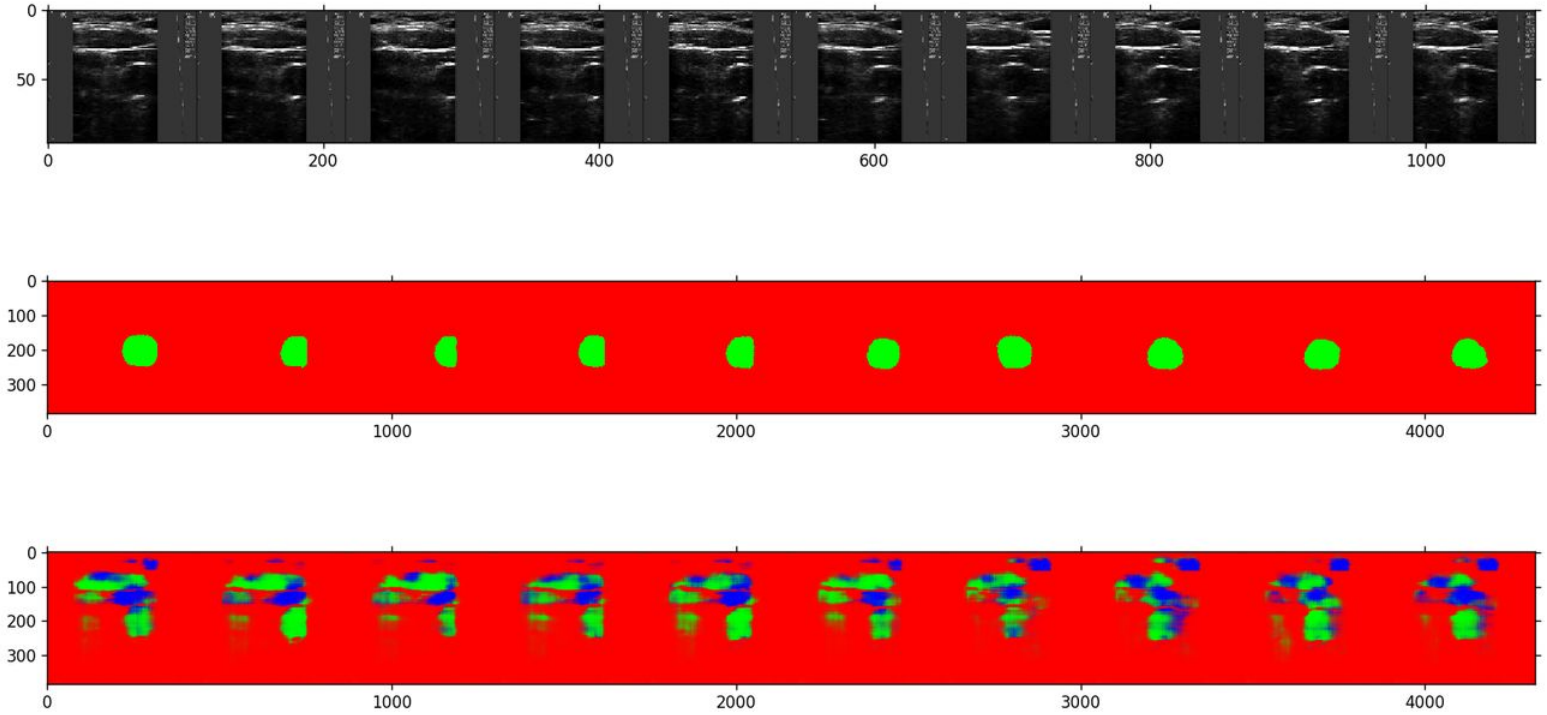




Xingjian Shi et al., 2015

Image segmentation




0
50
0
200
400
600
800
1000
0
100
200
300
0
1000
2000
3000
4000
0
100
200
300
0
1000
2000
3000
4000

Resources

Convolutional LSTM in Tensorflow

Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting

Prednet (Keras Layer) see video

Bouncing balls example

Review recurrent networks

CGRU https://arxiv.org/pdf/1511.08228.pdf (paper) https://github.com/openai/neural-gpu/tree/master/neuralgpu (code)
https://github.com/chiphuyen/stanford-tensorflow-tutorials/tree/master/examples/cgru (code II) and an example
https://ctmakro.github.io/site/on_learning/gru/gru.html

Conclusion

Suited to spatio temporal patterns (videos, wheather)

Anomaly detection

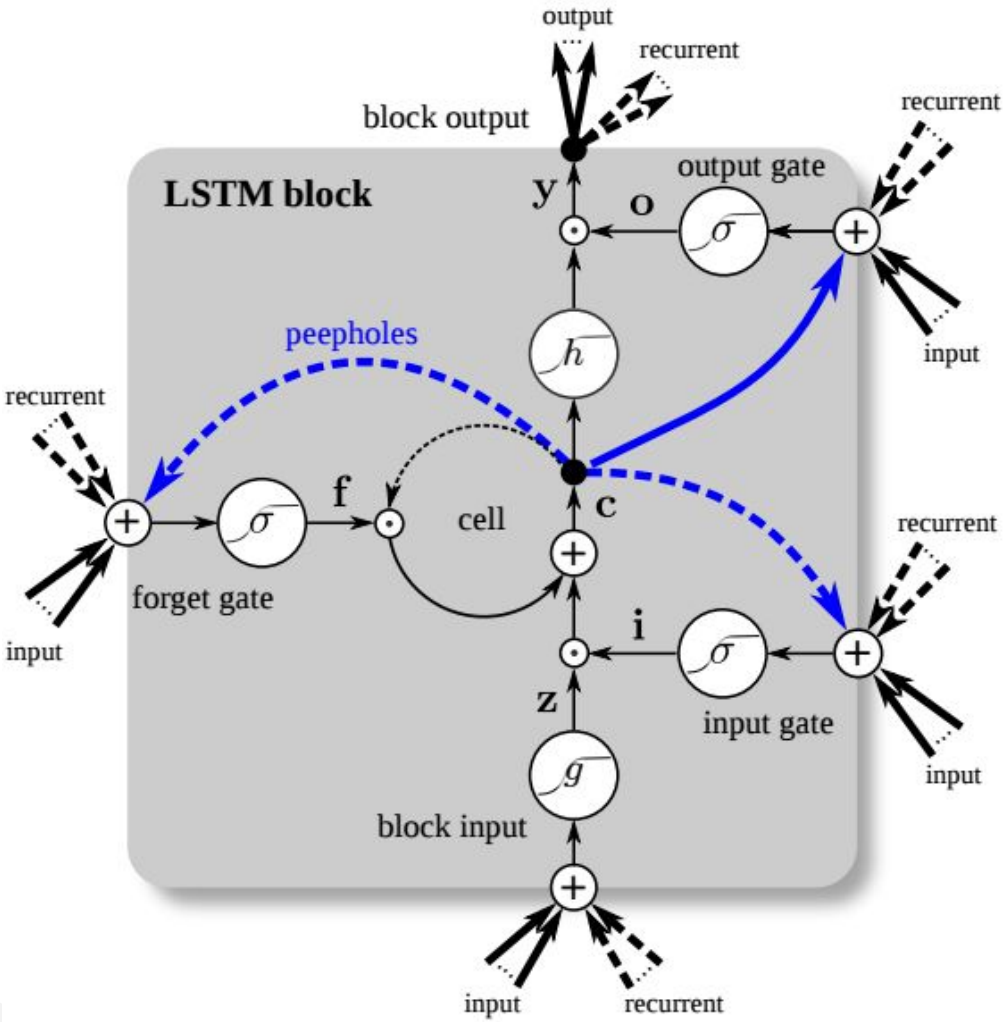
Learn model of the world

Computationally intensive

Tensorflow implementation through CGRUs

Model in Keras

```
return_sequences=True)(x)

    x = TimeDistributed(MaxPooling2D((2, 2), (2, 2)))(c2)
    x = ConvLSTM2D(nb_filter=2 * c, nb_row=3, nb_col=3, border_mode='same', return_sequences=True)(x)
    x = ConvLSTM2D(nb_filter=2 * c, nb_row=3, nb_col=3, border_mode='same', return_sequences=True)(x)
    c3 = ConvLSTM2D(nb_filter=2 * c, nb_row=3, nb_col=3, border_mode='same',
return_sequences=True)(x)

    x = TimeDistributed(UpSampling2D((2, 2)))(c3)
    x = merge([c2, x], mode='concat')
    x = TimeDistributed(Convolution2D(c, 3, 3, border_mode='same'))(x)

    x = TimeDistributed(UpSampling2D((2, 2)))(x)
    x = merge([c1, x], mode='concat')
    # x = TimeDistributed(Convolution2D(c, 3, 3, border_mode='same'))(x)

    x = TimeDistributed(Convolution2D(3, 3, 3, border_mode='same'))(x)
    x = TimeDistributed(UpSampling2D((2, 2)))(x)
    x = TimeDistributed(Convolution2D(3, 3, 3, border_mode='same'))(x)
    x = TimeDistributed(UpSampling2D((2, 2)))(x)

    output = TimeDistributed(Convolution2D(3, 3, 3, border_mode='same', activation=softmax_2d(-1)),
name='output')(x)

    model = Model(input_img, output=[output])
```



Legend

- unweighted connection
- weighted connection
- connection with time-lag
- branching point
- mutliplication
- sum over all inputs
- gate activation function (always sigmoid)
- input activation function (usually tanh)
- output activation function (usually tanh)

Parameters

- kerner size = 3x3
- Initial filter 12
- U-network
- small learning rate
- 100 k to 1 million parameters
- Adam optimizer