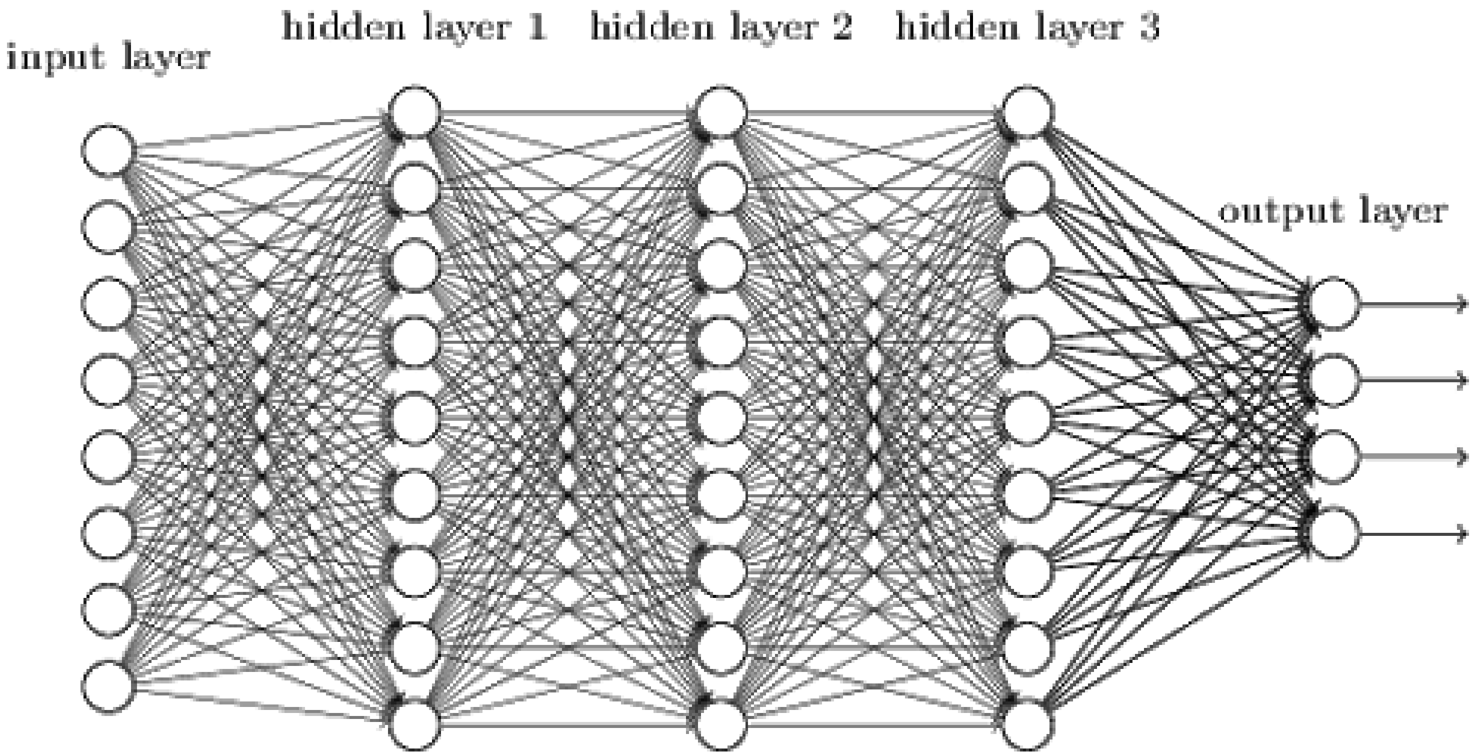
LUND UNIVERSITY

Lecture 4 – Machine Learning Models

Limitations of ”regular” Neural Nets

Fully connected network

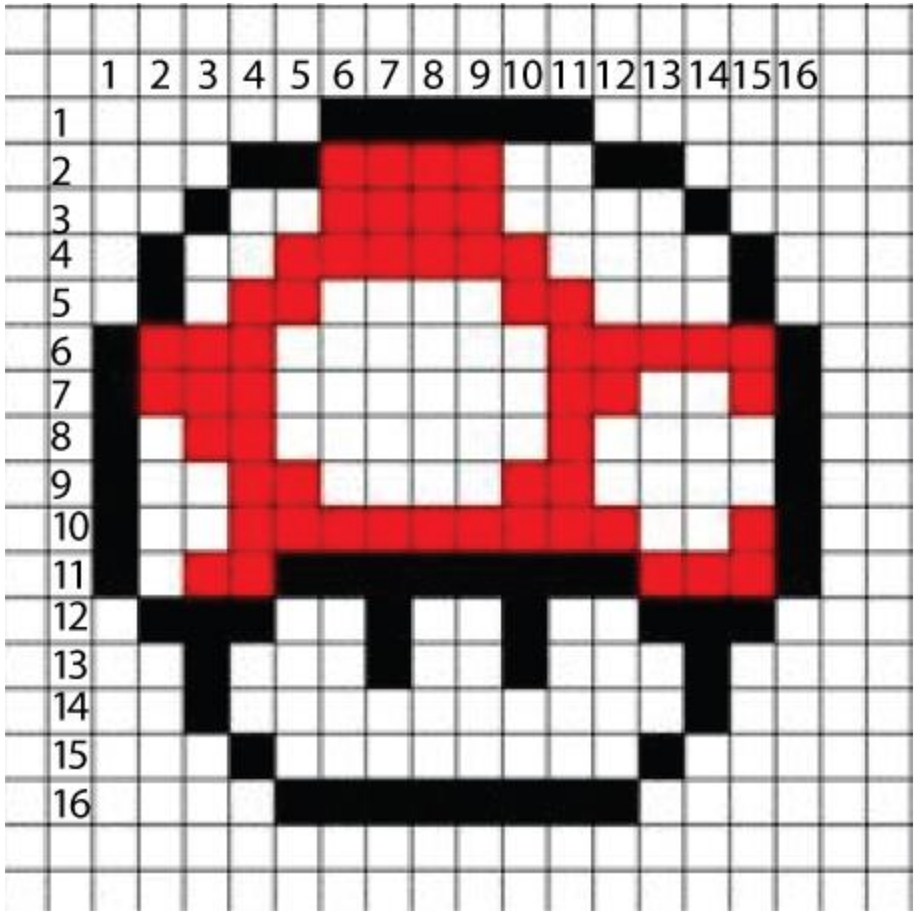


Outline – Lecture 4

<u>Other types of neural network models</u>

- Convolutional Neural Networks
- Recurrent Neural Networks (RNNs)
- Locally Connected Networks (LCNs)

Spatially structured data



An image is more than the sum of its pixels

i.e. the relative position of data points contain information, not just the pixel colors

How can we build an ANN that takes advantage of this?

The convolution operation

0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	0	1	0
0	1	0	0	1	0
0	1	0	0	1	0
0	0	1	1	0	0

image

0	0	1
0	1	0
1	0	0

“Filter”

0	0	1
0	1	0
1	0	0

*

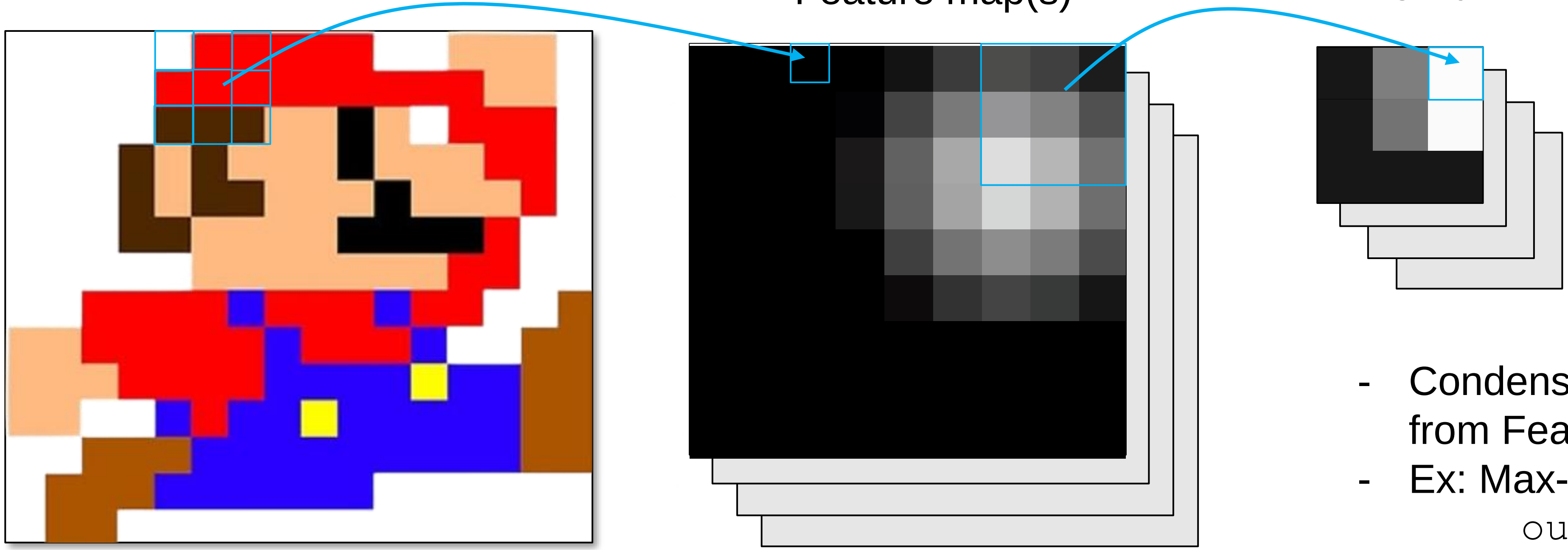
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	0	1	0
0	1	0	0	1	0
0	1	0	0	1	0
0	0	1	1	0	0

= 0*0 + 0*0 + 1*1 + 0*0 +
+ 1*1 + 0*0 + 0*1 + 1*0 + 0*0 = 2

2	2	0	1
1	1	1	1
1	1	1	1
1	0	2	2

Feature map

Convolution in an ANN



Shared weights + bias
in whole feature map

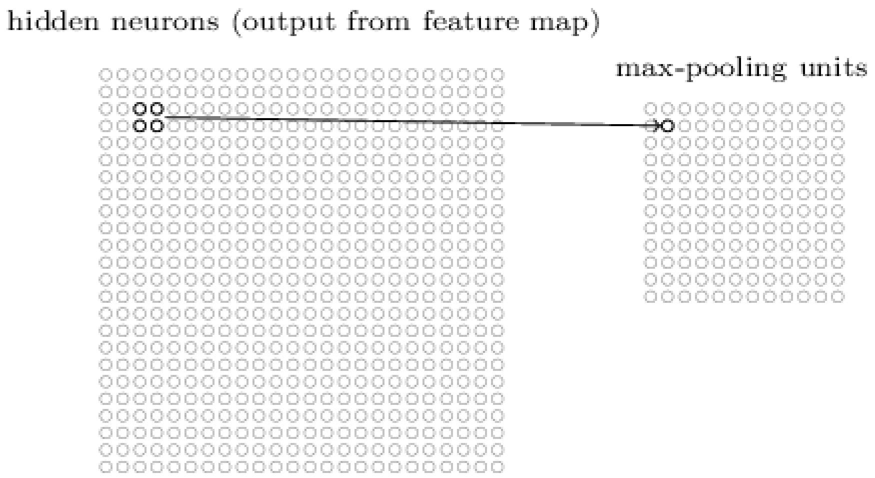
- Condenses information from Feature map
- Ex: Max-pooling

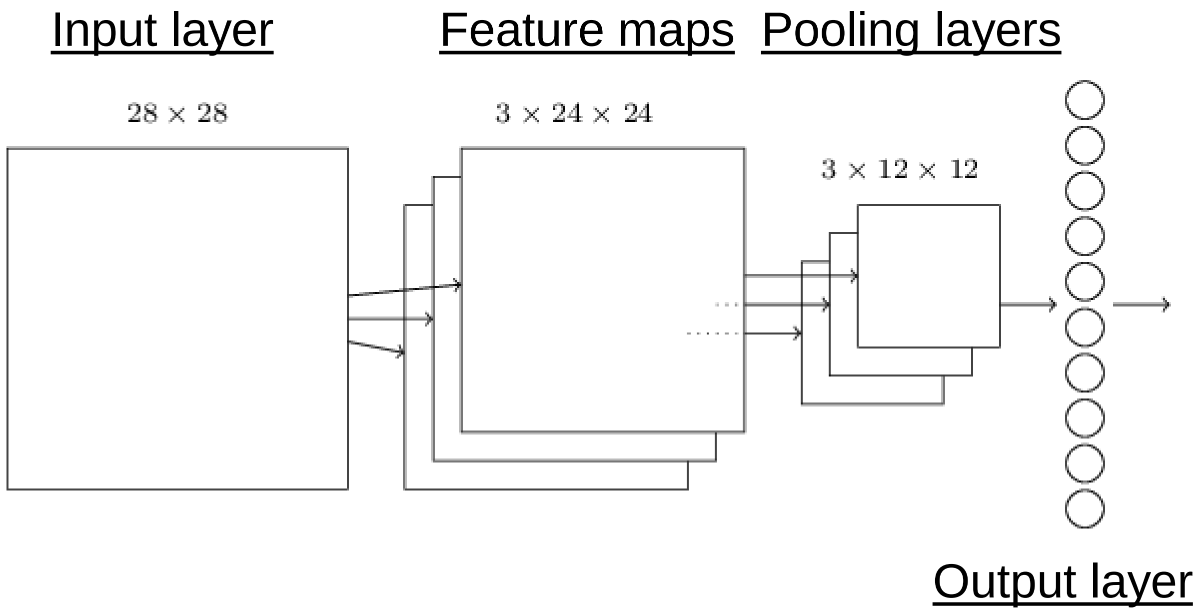
  `out = max(in)`

https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Convolutional Neural Network (CNN)

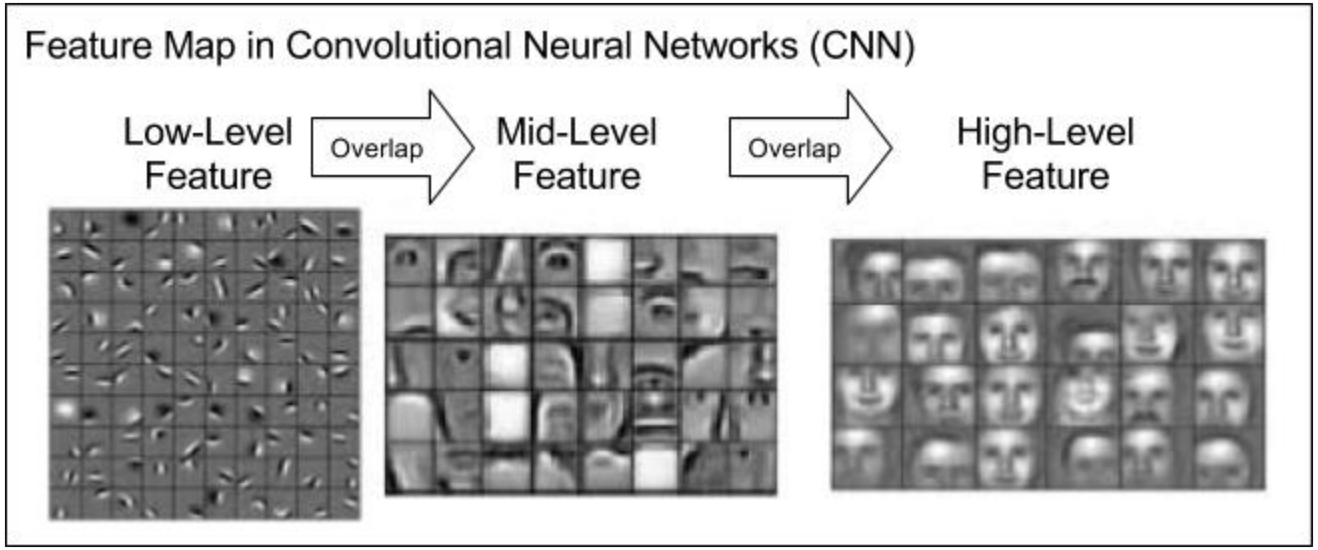
- A multitude of feature maps
- One pooling layer per feature map
  - L2 pooling: root of the sum of squares
- The Output layer is a normal completely connected layer to all neurons in the pooling layers

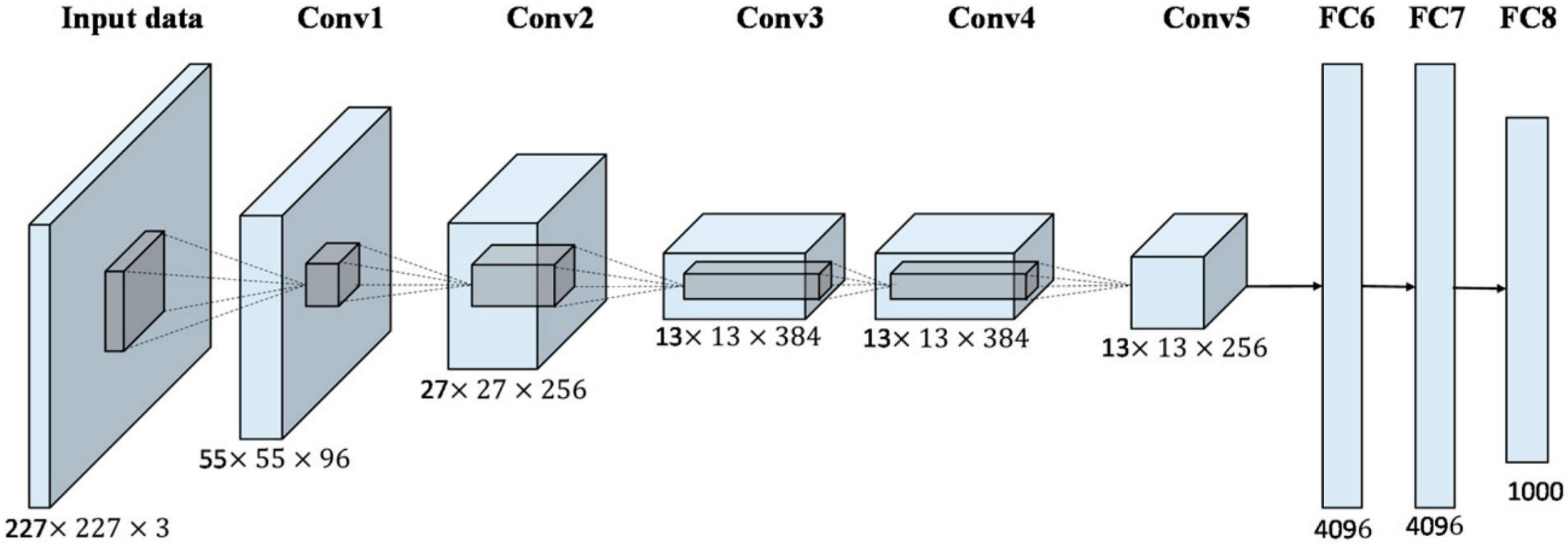




A full CNN

- Multiple convolutional layers for various levels of features.
- Data size reduced between layers
- For classification: Finish off with fully-connected layers





Han et al. Remote Sens. **2017**, *9*(8), 848, https://www.quora.com/How-does-a-convolutional-neural-network-recognize-an-occluded-face

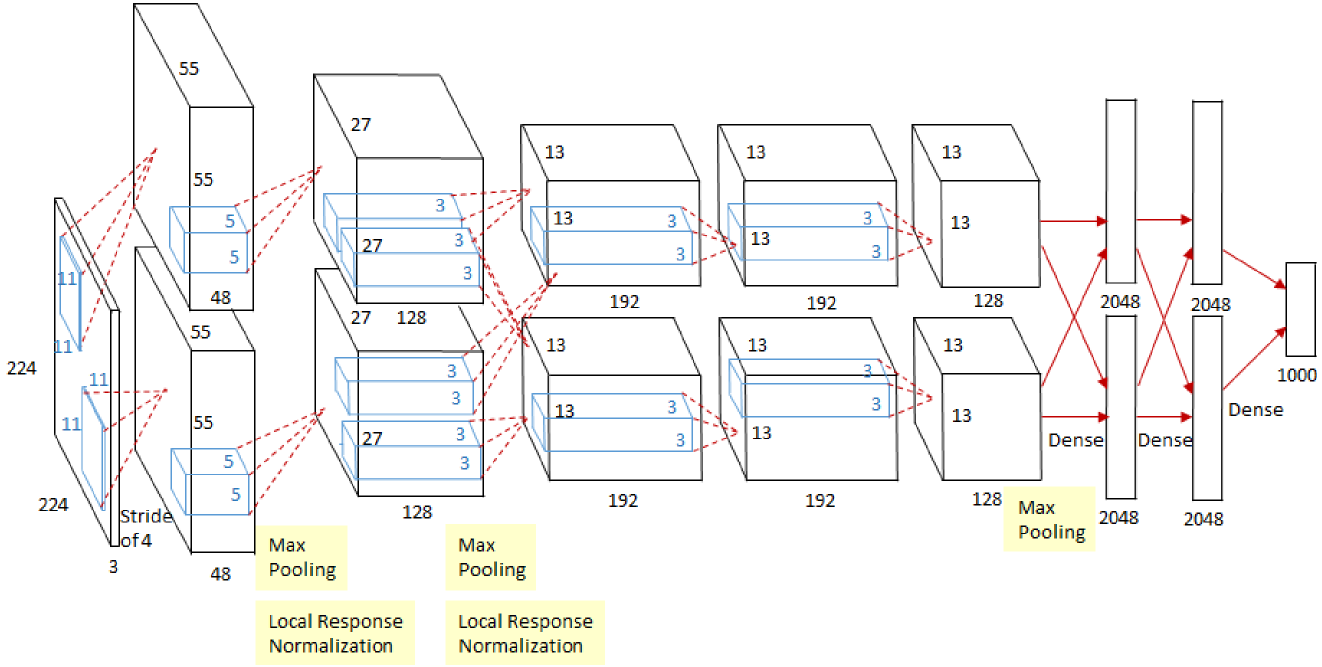
Classification Example

- AlexNet: Pioneer in CNNs, now standard for image classification
- First to use ReLU on CNNs
- 5 conv. Layers + 2 FCLs → 62.3 million parameters (!)
- 6 days to train

→ Won the 2012 ImageNet competition by a large margin
→ Top 1 accuracy of 62.5% (Top 5: 83%)

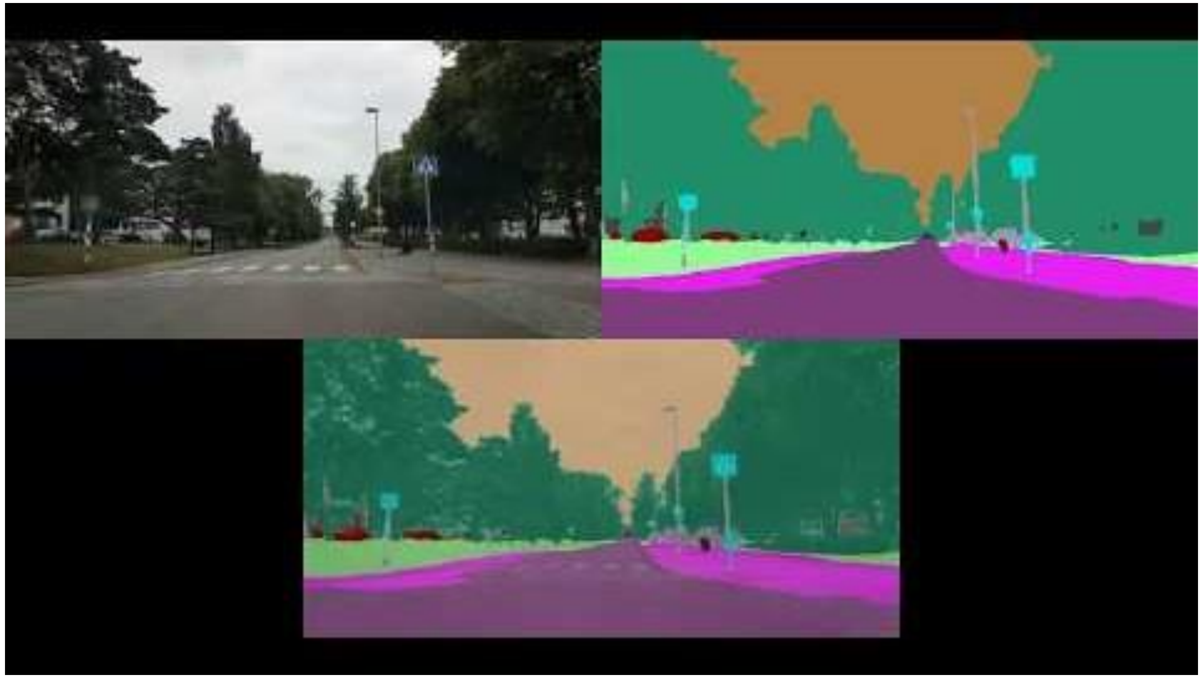
ImageNet

15 million high-res images in ~22000 categories

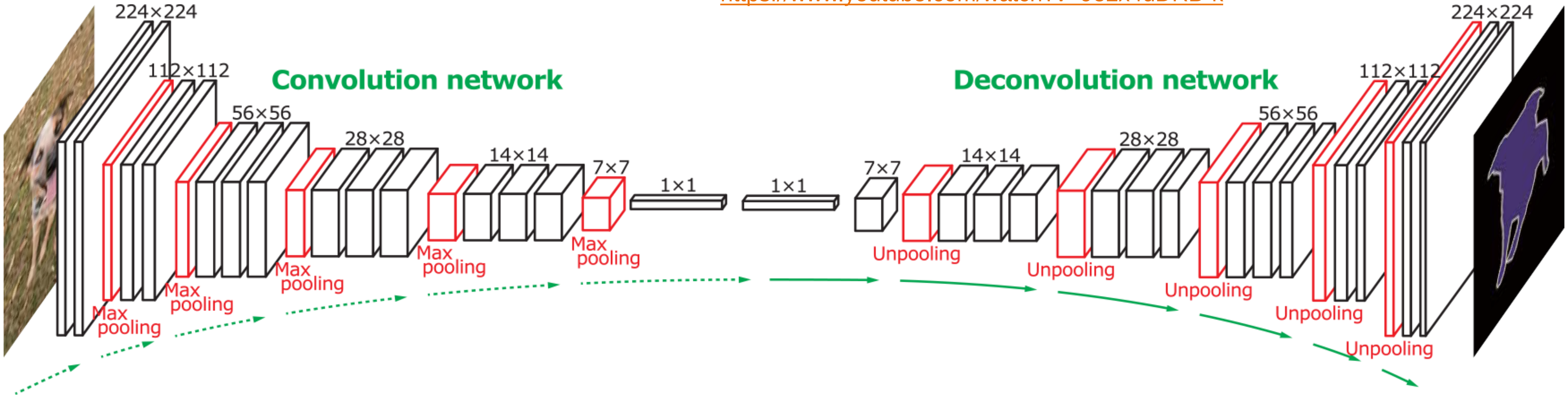


Other example

- Image segmentation
- Upscale features by “deconvolution”



https://www.youtube.com/watch?v=9e2x4dDRB-k



https://medium.com/@wilburdes/semantic-segmentation-using-fully-convolutional-neural-networks-86e45336f99b

Sequences

- CNNs see spatial correlations
- But what about if the order of data matters?
- **Q: Examples?**

The pony ate the hat of my father

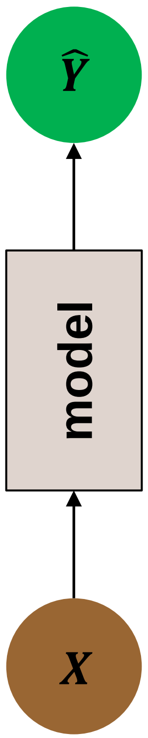
The hat of my father ate the pony

The of hat pony ate the father my

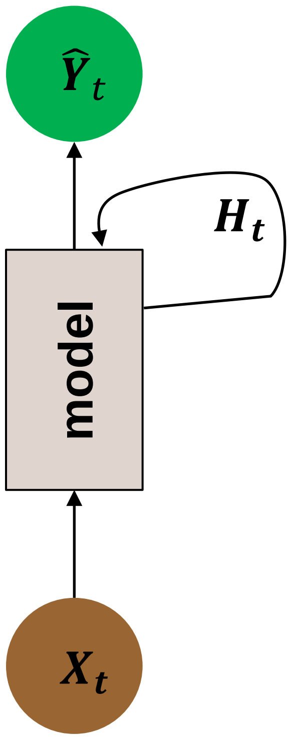
How can one predict what comes next, given an existing sequence of data?

Recurrent Neural Networks

Standard "feed-forward" network



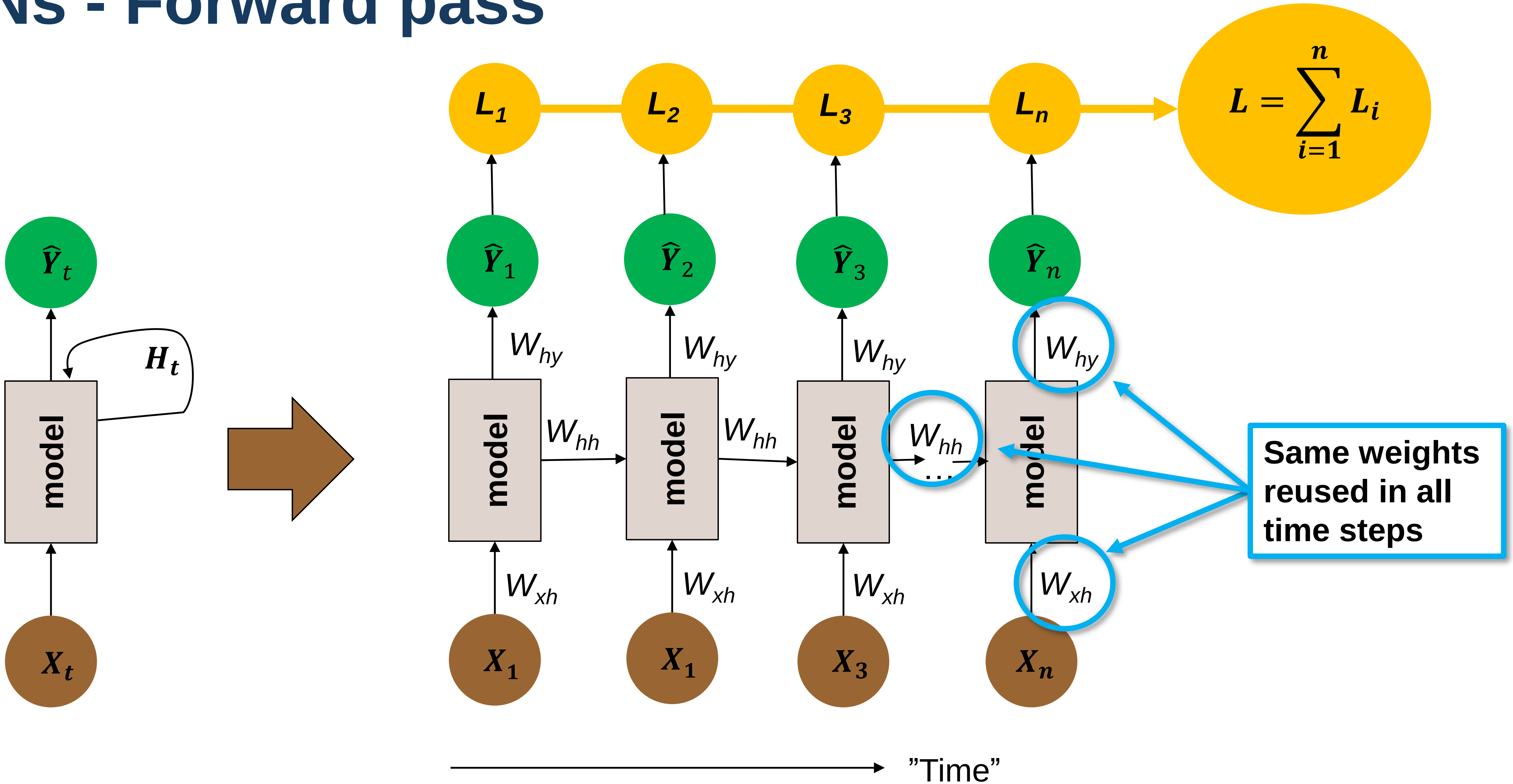
Recurrent network



→ Internal state

$$H_t = f_W(X_i, H_{t-1})$$

RNNs - Forward pass

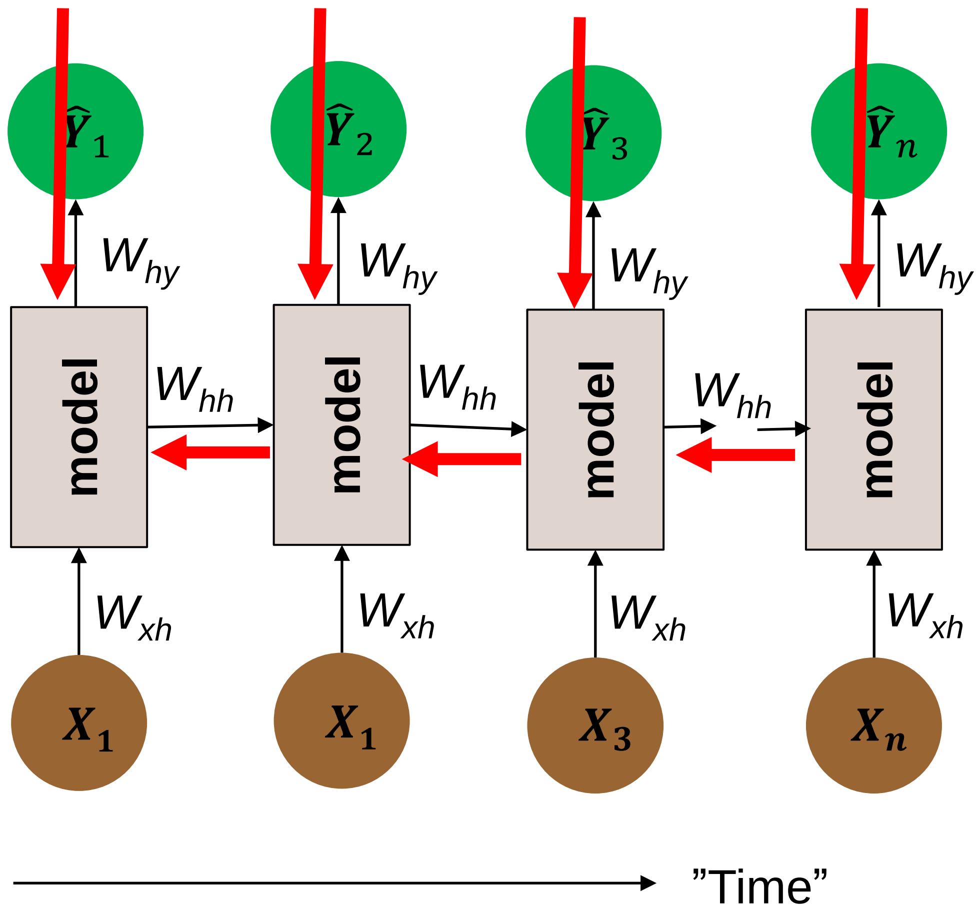


RNNs - Backpropagation through time

- Training by backpropagation involves propagating gradients backwards through each time step.

- Each step consists of a matrix multiplication with W_{hh} and gradient of activation function.

- **Q: Potential issues?**



Vanishing gradient problem

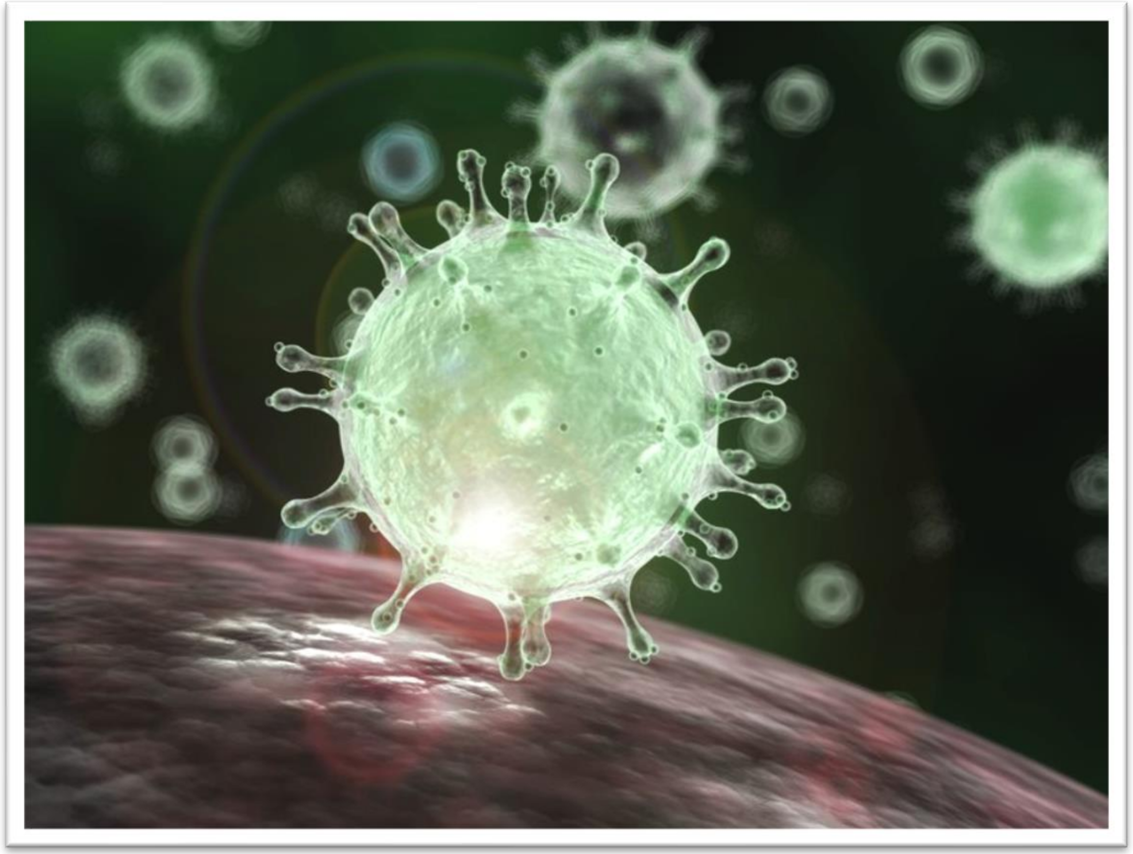
- Vanishing gradient → biasing to capture recent (short-term) dependencies

Short-term

The virus gave me the ____.

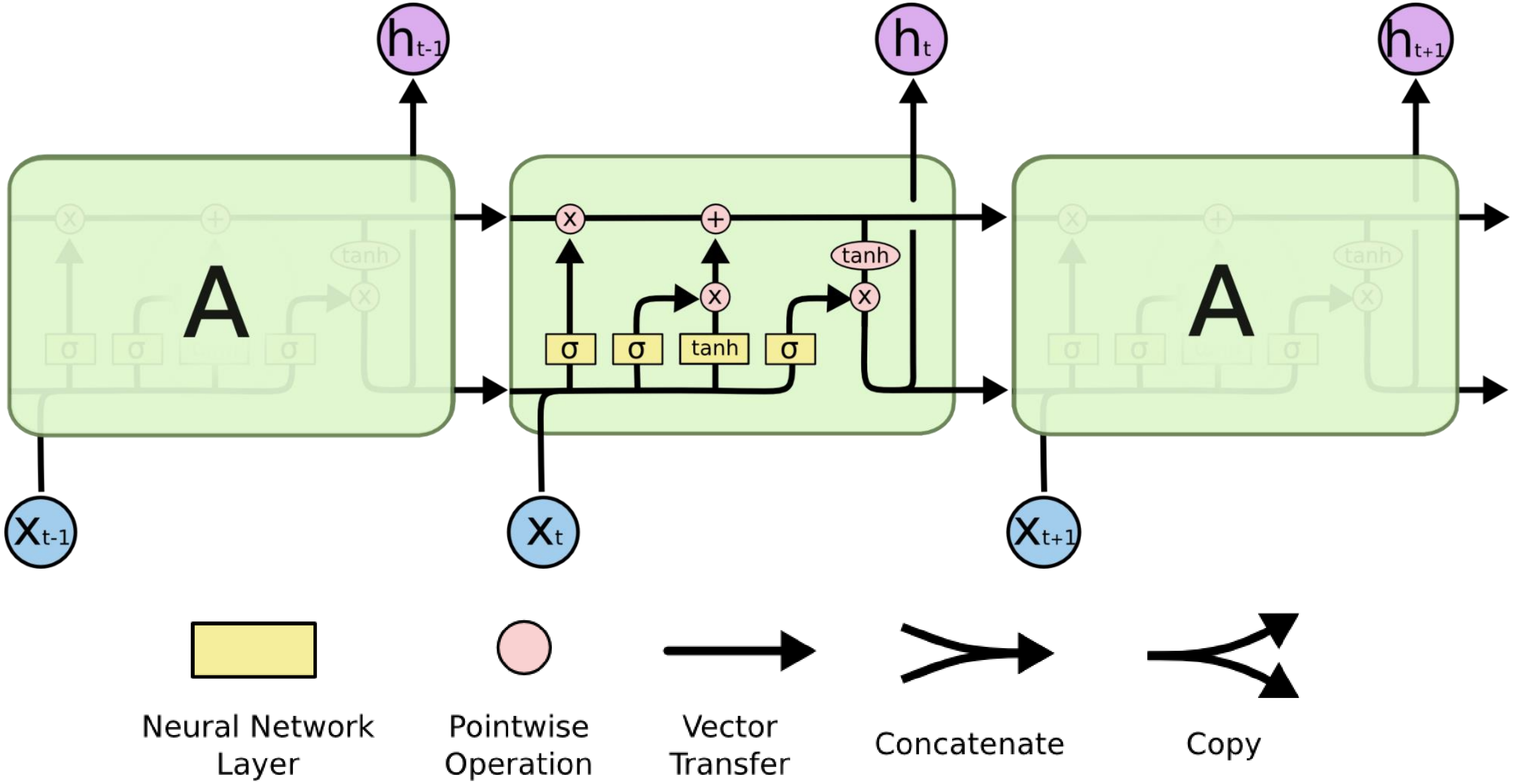
Long-term

A virus is an infective agent that typically consists of a nucleic acid molecule in a protein coat and can give a person the ____.



Long Short Term Memory Networks

- LSTM: Recurrent units that are good for tracking long-term dependencies

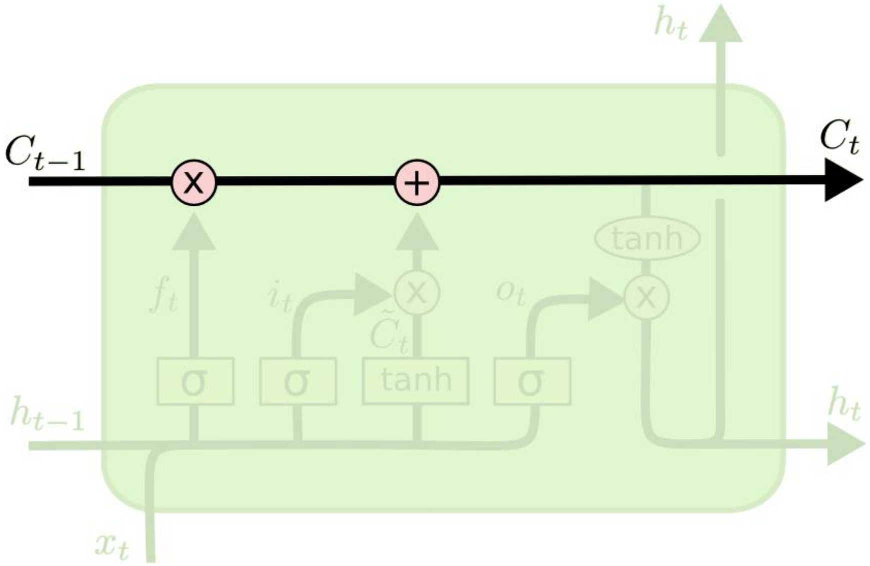


https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Important features of LSTMs

Cell state

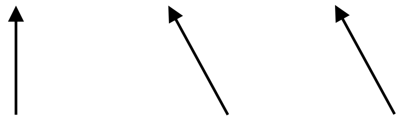
with easy information transfer through time



- Only elementwise multiplication and addition
- Avoids vanishing gradient problem!

Example:

”Mike likes his boat, but …”

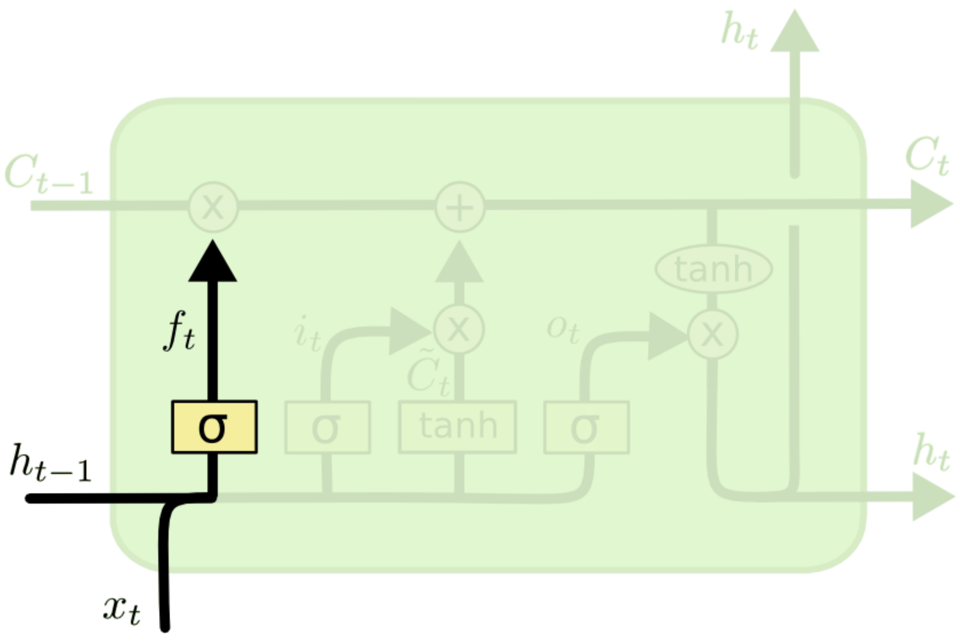


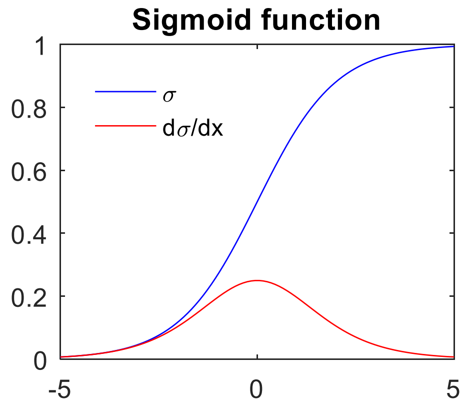
Subject
Gender
Singular/plural

Verb
Tense
Inflection

Object
Type
Singular/plural

1. What to forget





$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

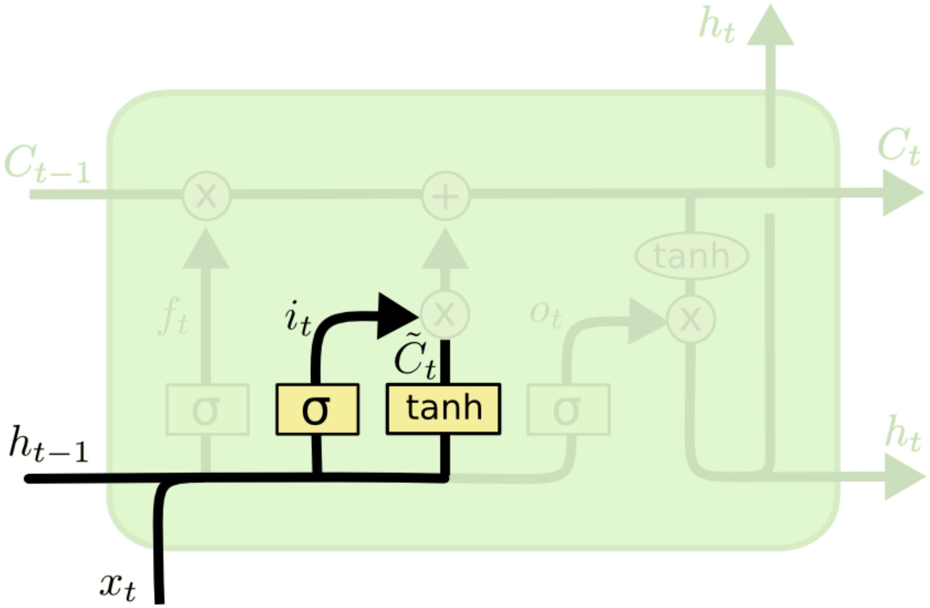
x_t

”Mike likes his boat, but Maria…”

New subject,
Forget Mike?

- f_t outputs 0…1 for each element in C_{t-1}, based on h_{t-1} and x_t
- $f_t = 0$ → completely forget
- $f_t = 1$ → completely remember

2. What to update



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

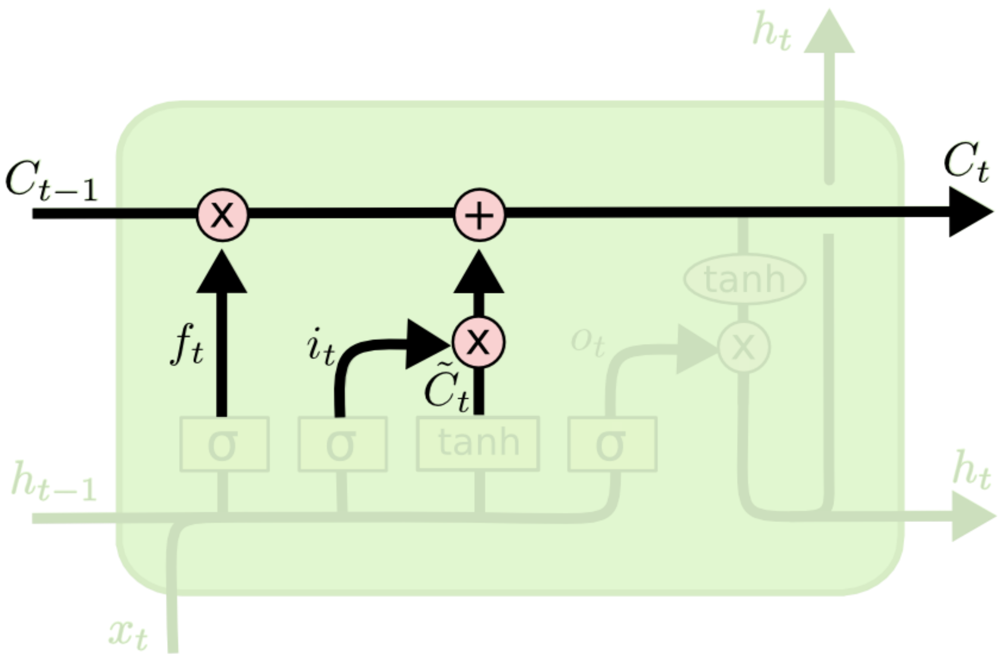
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

"Mike likes his boat, but Maria…"

Candidate for new Subject

- tanh-layer creates new candidates for cell state C (like in regular RNN)
- σ-layer decides which of these to update (factor 0..1)

Updating cell state



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

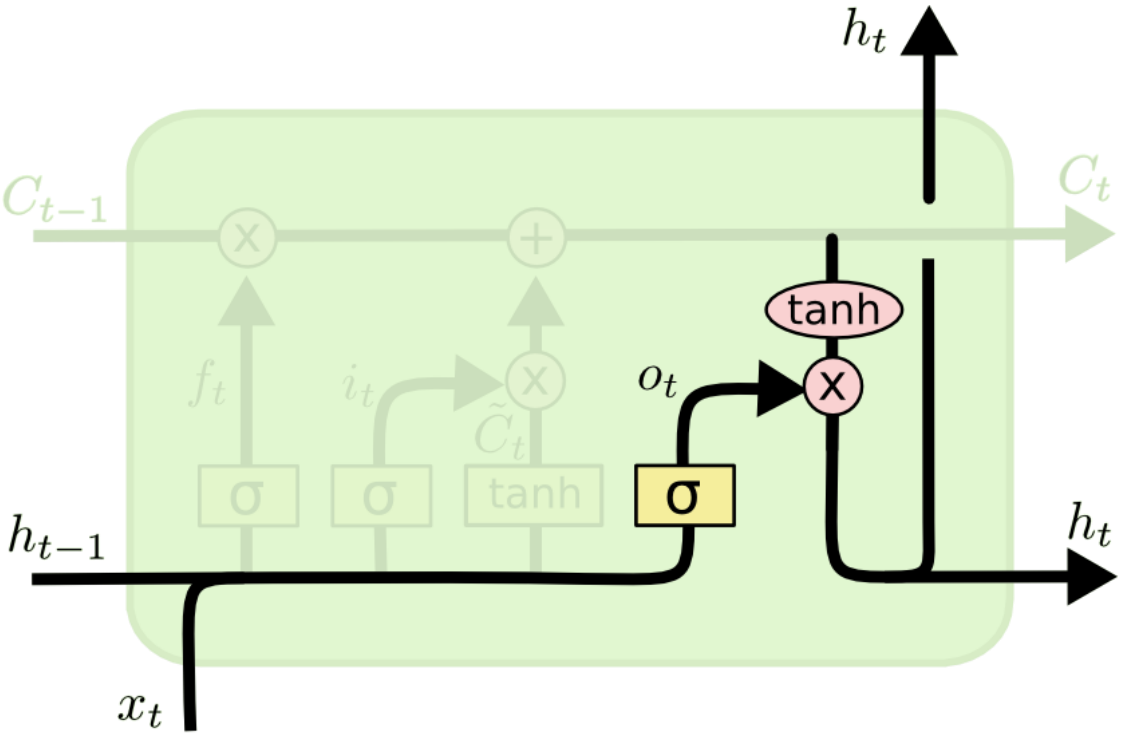
"Mike likes his boat, but Maria…"

Update C by forgetting Mike

Update C with new Subject

- Cell state is updated elementwise! (computationally cheap)

Outputting to next timestep



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

- tanh-function bounds output between -1 and 1
- σ-layer decides what to output (factor 0..1)
  - Relevant to what should come next (verb etc)

opposition

”Mike likes his boat, but Maria…”

Gender, singular

Transport, aquatic, …

positive, sentiment, …

2 min Exercise – What could come next?

opposition

”Mike likes his boat, but Maria…”

Gender, **singular**

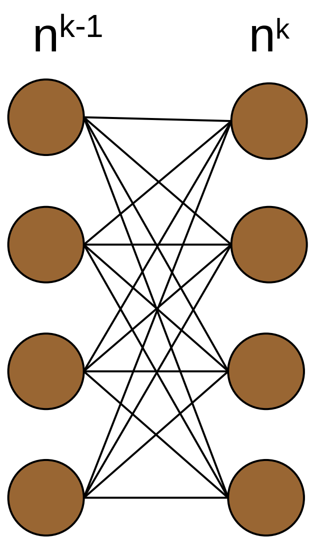
Transport, aquatic, …

positive, sentiment, …

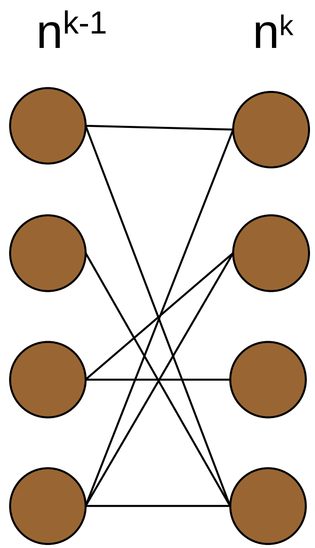
Example – Music generation

Sparse and Locally connected networks

- Typical parameter size in Neural Network: 1-10 million
  - Work-intensive
  - Risk of overfitting
  - Scalability issues!



$n^{k-1} \times n^k$ synapses



$O(n^{k-1} + n^k)$ synapses

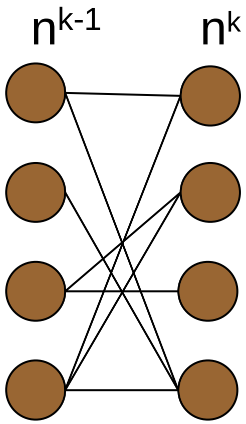
- Can layers with $O(n^{k-1} + n^k)$ connections perform as good?
- What type of connectivity would then be ideal?

Sparse Network topologies

Erdos-Renyi network
(random graph)

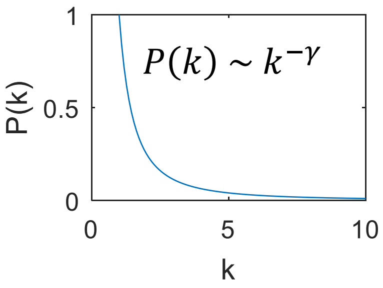
$$P\left(W_{ij}^k\right) = \frac{\varepsilon\left(n^k + n^{k-1}\right)}{n^k n^{k-1}}, \varepsilon \in \mathbb{R}^+$$

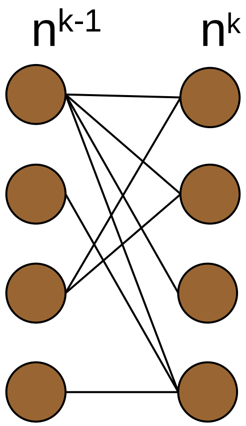
→ $\left|W^k\right| = \varepsilon(n^k + n^{k-1})$ if $\varepsilon \ll n^k, n^{k-1}$



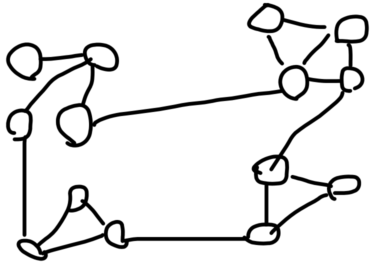
$O(n^{k-1} + n^k)$ synapses

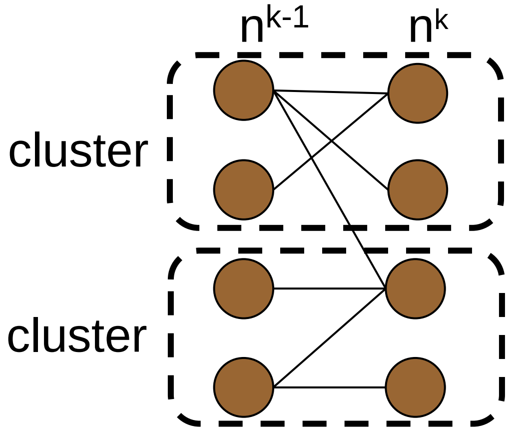
Scale-free network
(few nodes with many connections)





Small world network
(most nodes are not neighbors, but connected in a few steps)





Example – Adaptive sparse connectivity

- Sparse-Evolutionary-Training (SET) algorithm

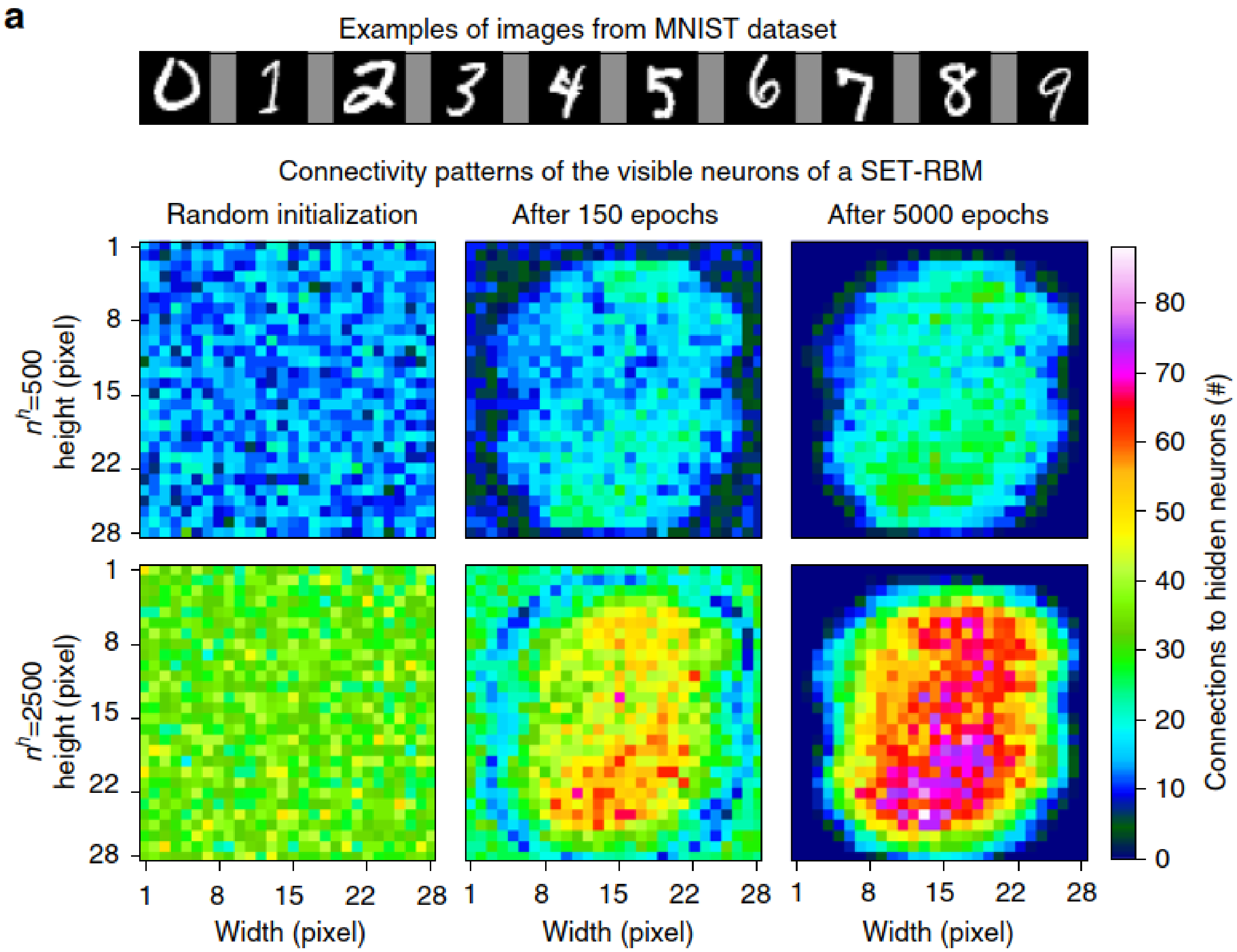
Algorithm 1: SET pseudocode

```
%Initialization;                                                                    1.
initialize ANN model;
set ε and ζ;
for each bipartite fully-connected (FC) layer of the ANN do
    replace FC with a Sparse Connected (SC) layer having a Erdős-Rényi topology given by ε and Eq.1;
end
initialize training algorithm parameters;
%Training;
for each training epoch e do                          2.
    perform standard training procedure;
    perform weights update;
    for each bipartite SC layer of the ANN do                   3.
        remove a fraction ζ of the smallest positive weights;                4.
        remove a fraction ζ of the largest negative weights;
        if e is not the last training epoch then
            add randomly new weights (connections) in the same amount as the ones removed previously;
        end
    end
end
```

Clustering of connections

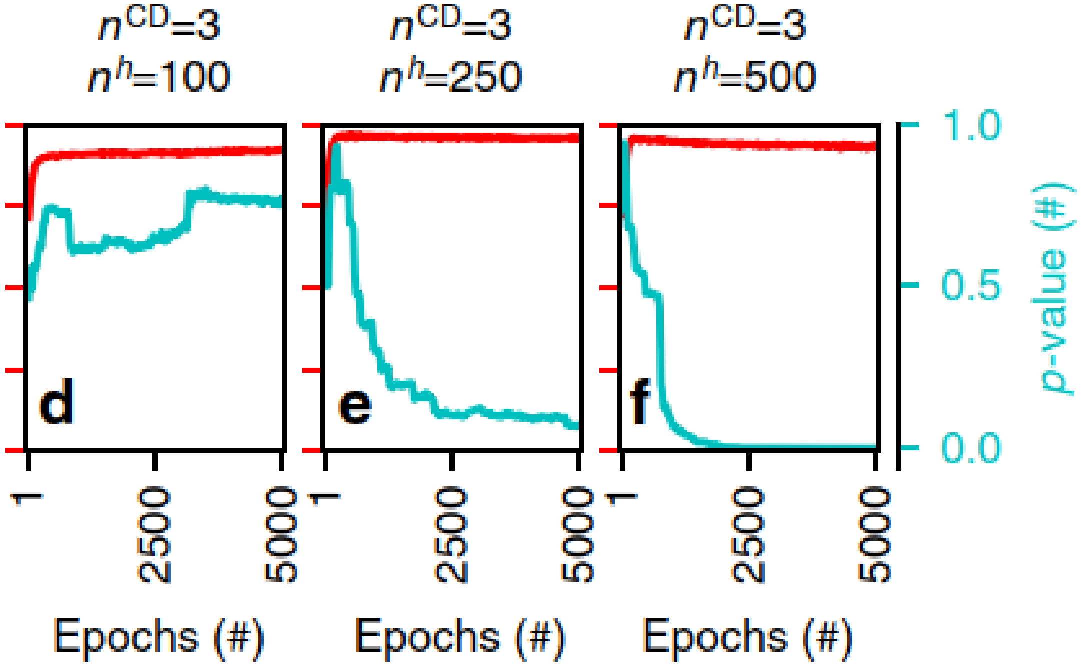
- Towards end of training:
  - Pixels in the center of the images (MNIST) have many connections
  - Pixels on the edges have few or zero connections

→ Natural data-dependent clustering (scale-free network?)



Evolution towards scale-free network

- Indeed: Adaptive sparsity evolves towards a scale-free network
- Starts from a binomal probability distribution (Erdos-Renyi)
- With enough neurons in a layer → scale-free ($p < 0.05$)



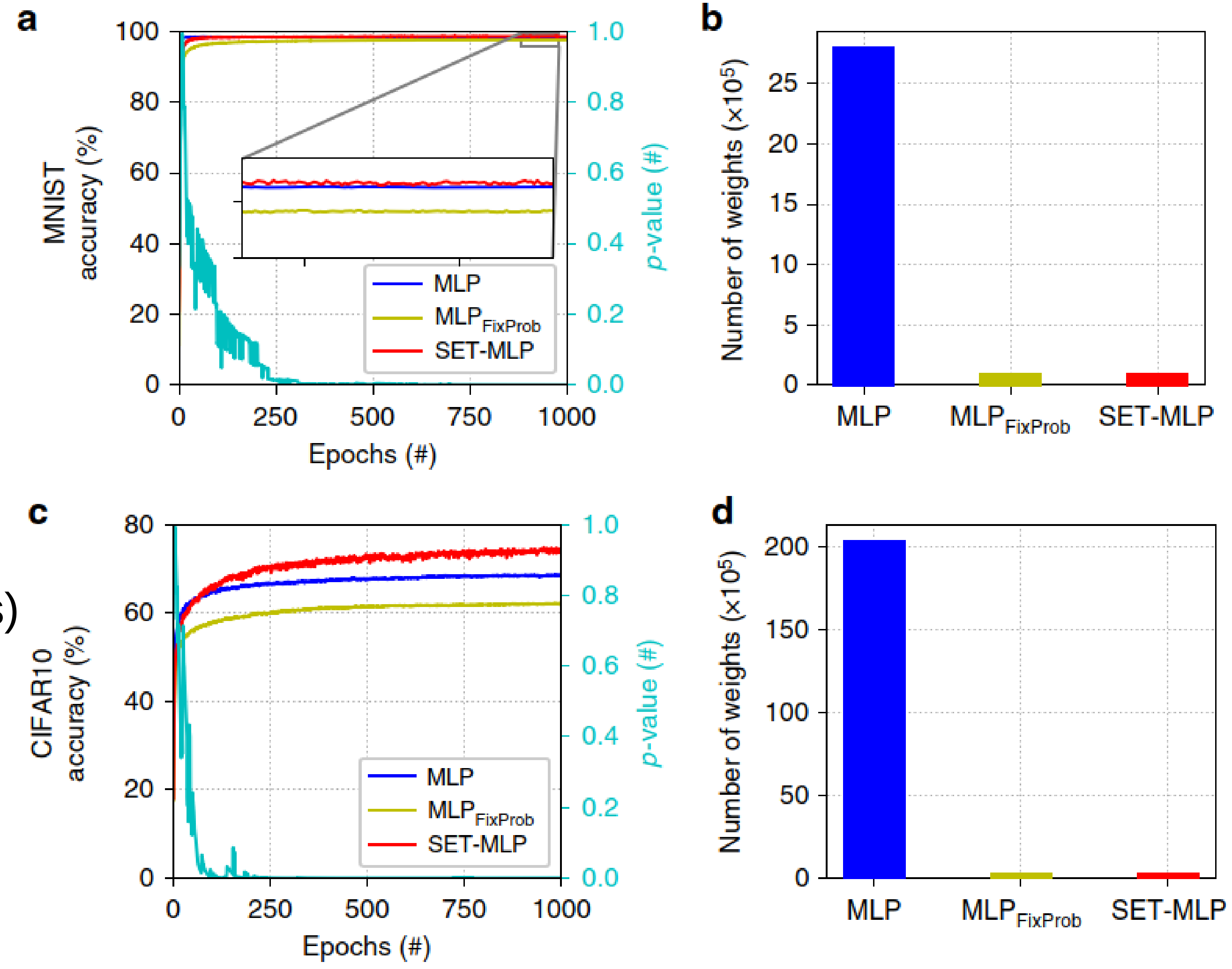
Mocanu et al Nat Comm 2018

Results compared to fully connected networks

- $MLP_{FixProb}$: Fixed sparse network (E-R topology)
- MLP: Fully connected network
- SET-MLP: Adaptively sparse network

- Fixed sparsity performs worst
  - 62% on CIFAR10
  - But with only 1.4% as many weights
- Adaptive sparsity outperforms all!
  - 75% on CIFAR10 (278K params)
  - 2nd best in literature has 74% (31M params)

	MLP	$MLP_{FixProb}$	SET-MLP
MNIST	98.55%	97.68%	98.74%
CIFAR10	68.70%	62.19%	74.84%

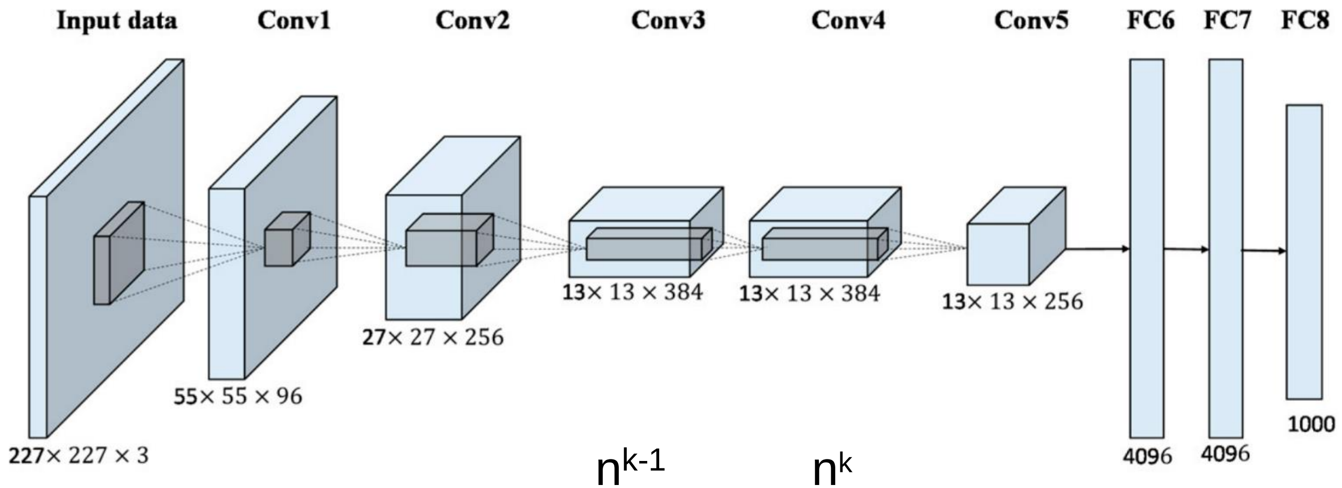


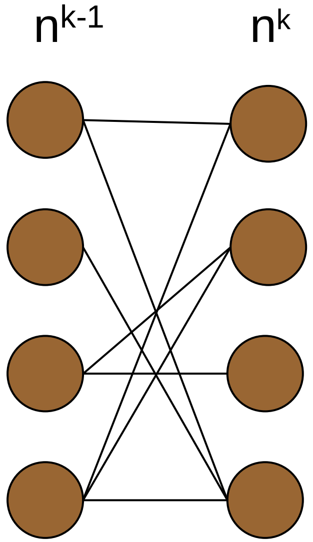
Mocanu et al Nat Comm 2018

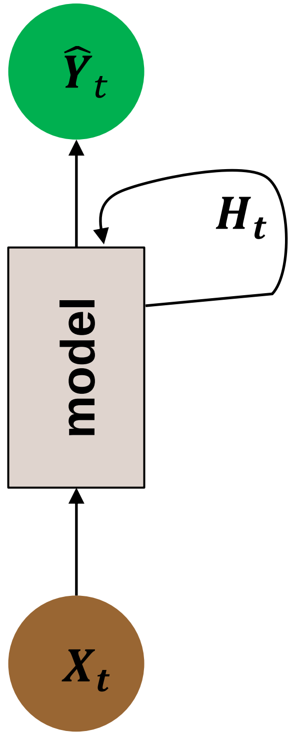
Possibilities with LCNs

- The sparsity enables:
  - MUCH larger models (1 billion neurons…) on supercomputers
  - Powerful tiny models trained directly on wearables
- Problem: GPU optimized for dense matrices, not for sparse..
  - More on GPU next lecture…

Summary


Input data
Conv1
Conv2
Conv3
Conv4
Conv5
FC6
FC7
FC8
227× 227 × 3
55× 55 × 96
27× 27 × 256
13× 13 × 384
13× 13 × 384
13× 13 × 256
4096
4096
1000


n^{k-1}
n^{k}


$\hat{Y}_t$
H_t
model
X_t