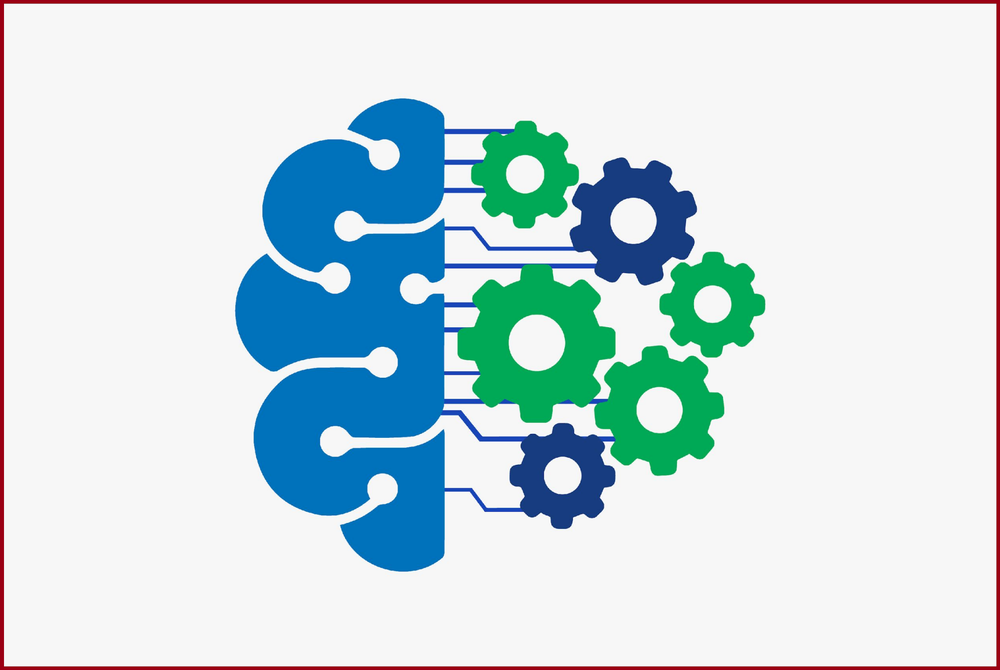
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Deep Learning: Advanced Approaches

Machine Learning 2022-23

Outline

Deep Learning: Advanced Approaches

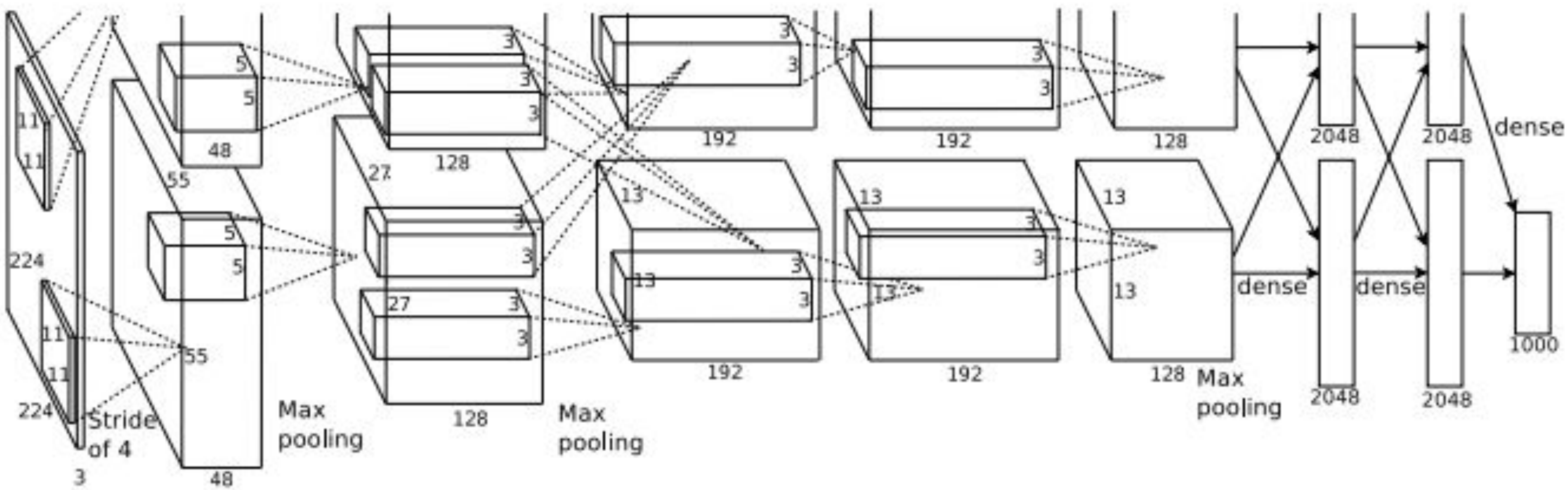
1. *Advanced CNN schemes:* Residual networks, skip connections, auto-encoders
2. *Generative models*: Generative Adversarial Networks (GAN)
3. *Modeling temporal information*: Recurrent Neural Networks (RNN) and Long-Short Term Memory (LSTM) *(not part of the course)*

Advanced CNN Models

- We'll see some relatively recent advanced architectures
- Some new concepts will be briefly introduced:
  - Residual Networks
  - Auto-Encoders
  - Skip Connections
  - Transposed Convolutions

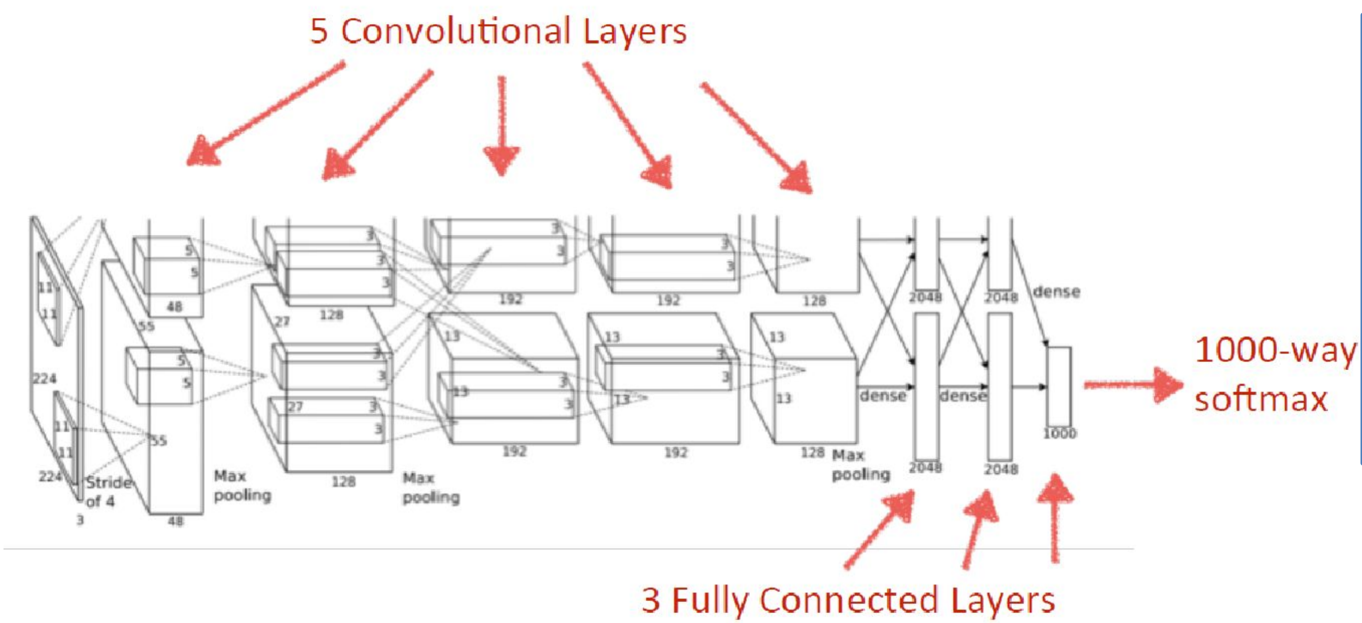
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

«Historical» Perspective: AlexNet (2012)

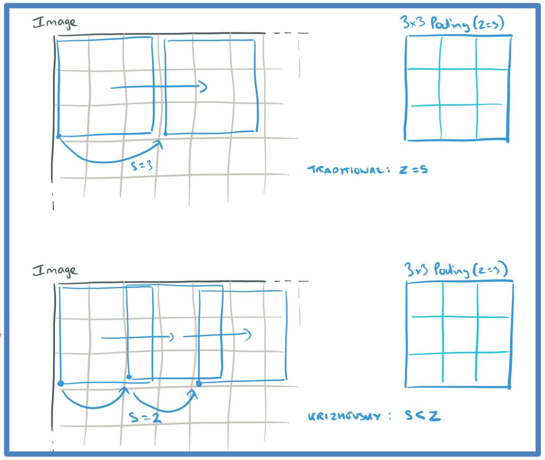


- *AlexNet* [3]: First Deep Learning approach outperforming "classic" methods (i.e., outperforming SVM or RF)
- Exploits 11x11, 5x5, 3x3, convolutions, max pooling, dropout, data augmentation, ReLU activations, SGD with momentum
- Split in 2 pipelines since it was trained with 2 GPUs (for 6 days)
  - According to Nvidia the DGX-2 server released in 2018 can train it in 18 mins!!!
- Complex but quite "standard" model

AlexNet: the Network



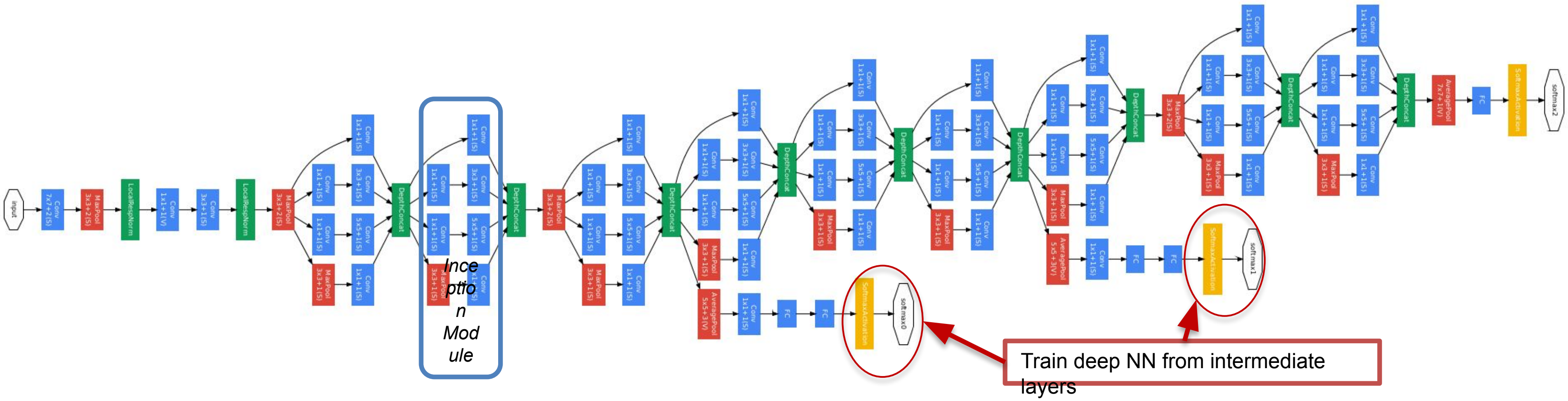
overlapping pooling



- 5 convolutional layers, 3 fully connected ones
- Many feature maps for each layer
- 650K neurons, 60M parameters
- Rectified Linear Units (ReLU) activations, overlapping pooling, dropout trick
- Training with randomly extracted 224x224 patches for more data

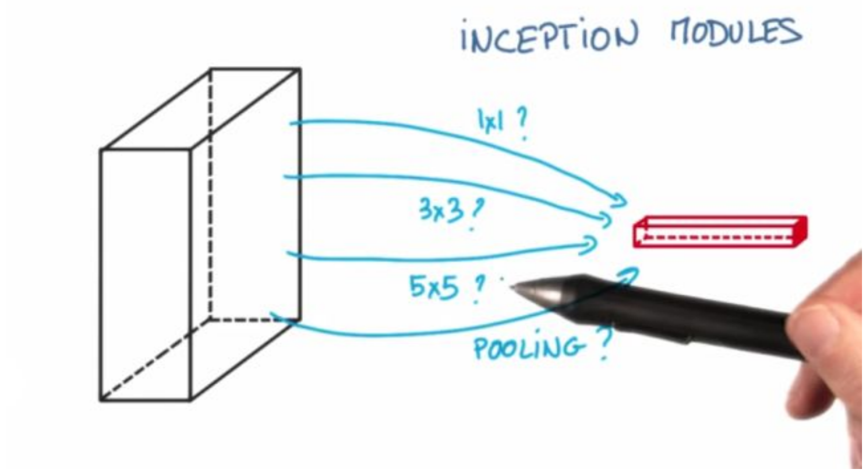
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

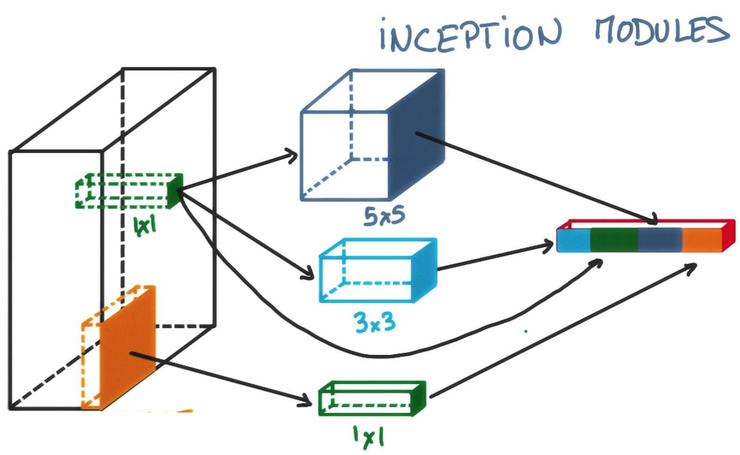
GoogleNet (Inception V1)

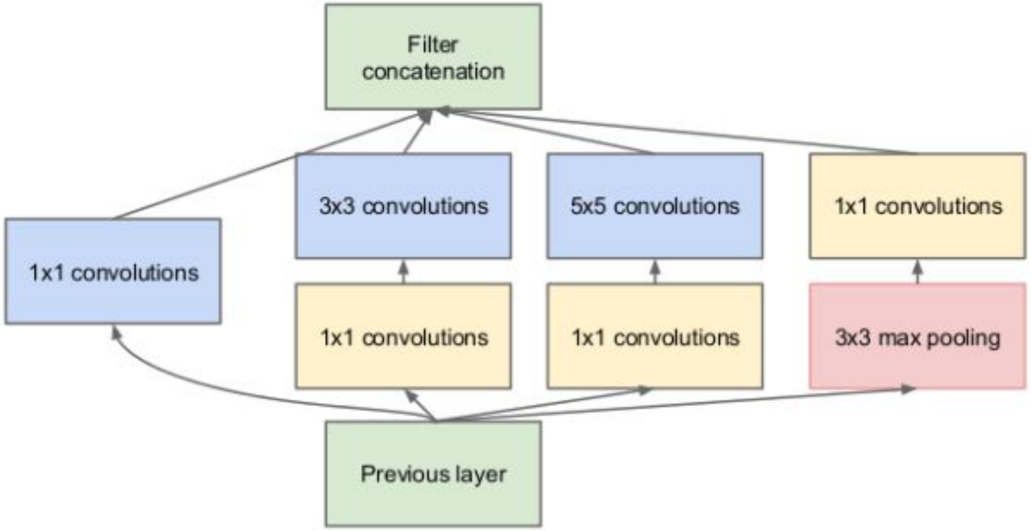


- Released in 2014, 1st method very close to human level performance
- Implemented a novel element: *the inception module*
    - This module performs multiple small convolutions with different sizes in parallel
- The networks is a 22 layers deep CNN but reduced the number of parameters from 60M of AlexNet to 4M

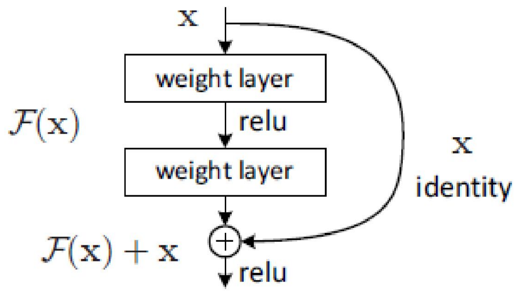
The Inception Module



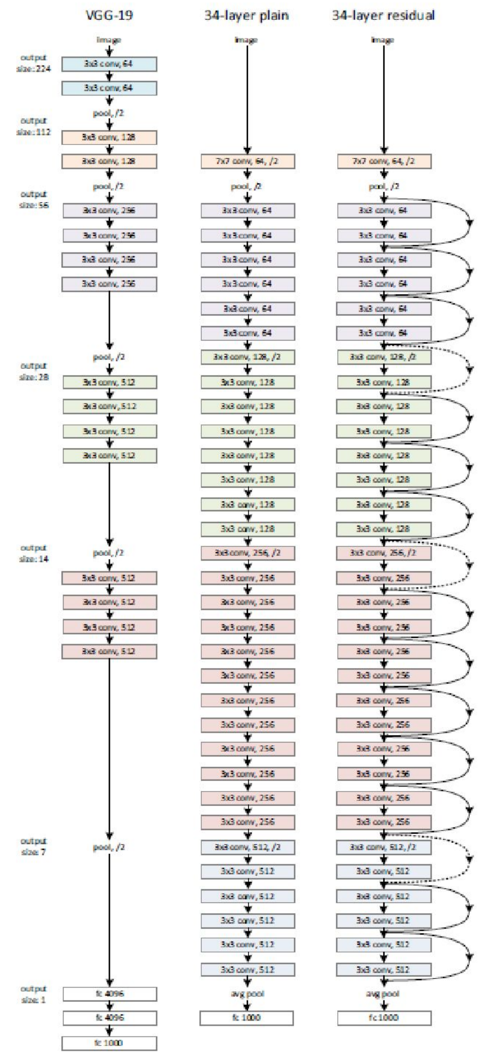




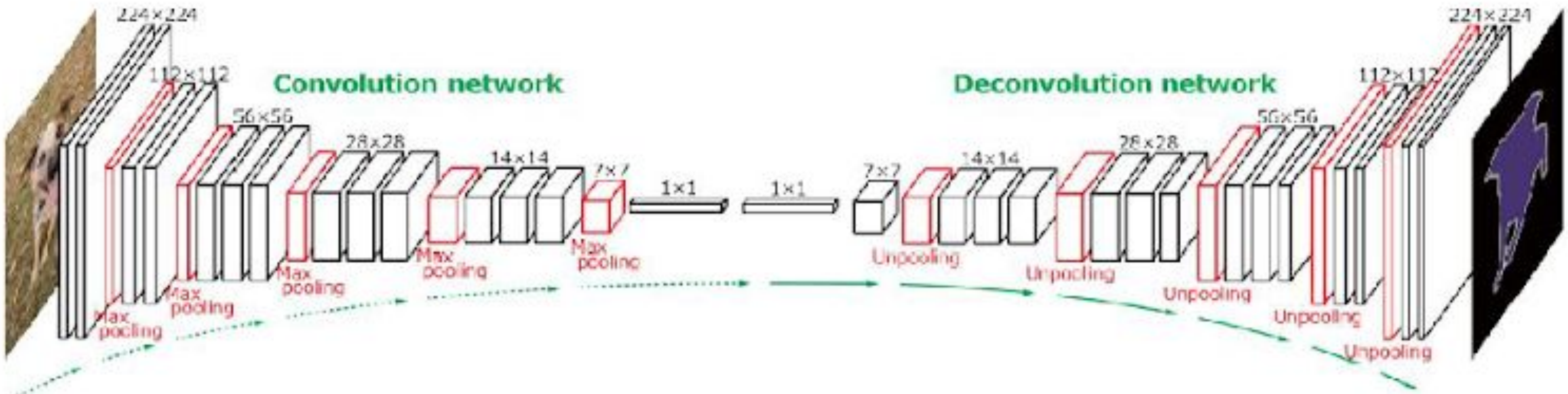
ResNet

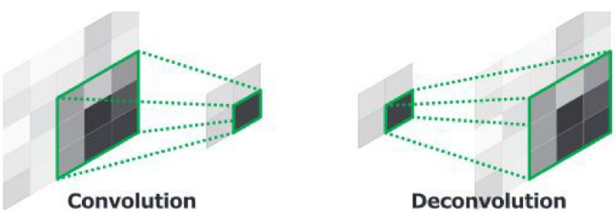


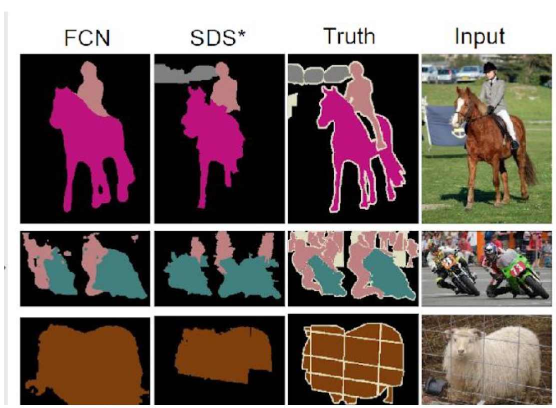
- Residual Neural Network [4] introduced in 2015 a novel architecture with "*skip connections*"
- Idea: try to estimate the residual w.r.t the previous estimation instead of the function itself
- Thanks to this technique they were able to train a NN with 152 layers with reasonable complexity
- Was able to beat human-level performance on image classification tasks



Upsampling : Transposed Convolutions



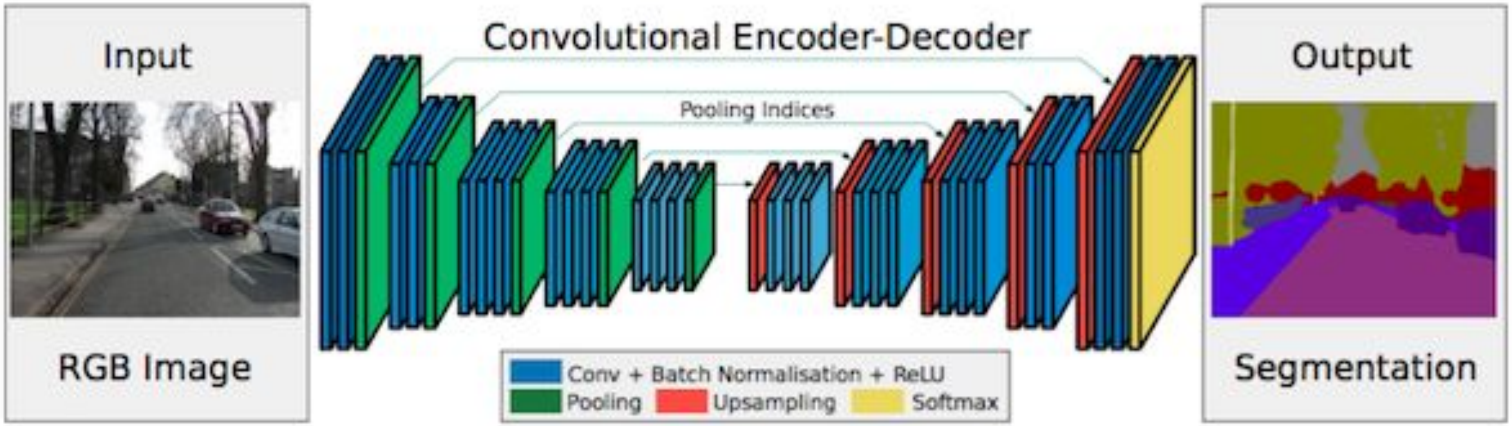




- ❑ In some applications the output has the same or even larger size than the input (e.g., semantic segmentation, denoising)
- ❑ Convolutional layers connect multiple input activations within a filter window to a single activation
- ❑ Transposed convolutions associate a single input activation with multiple outputs
- ❑ Use transposed convolutions for upsampling

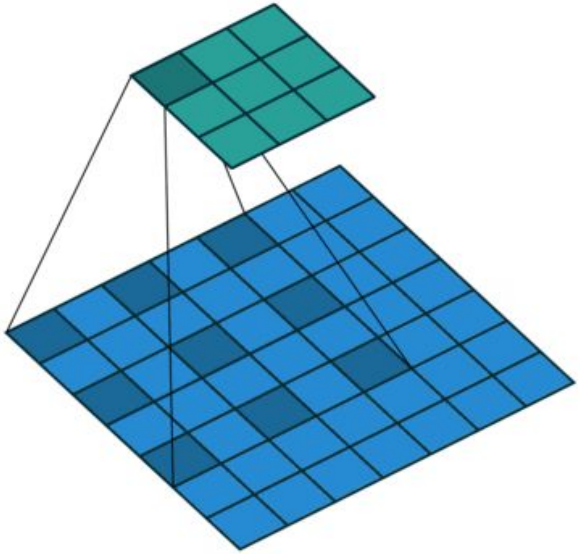
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Encoder-Decoder Architectures



- The network is made of 2 parts, an encoder and a decoder
- A "*compressed*" description of the input data is created at the middle layers by the encoder
- The decoder expands it into the final result
- Maxpooling indices can be transferred to decoder to improve the reconstruction
- FCN and SegNet are among the first encoder-decoder architectures
  - Fully Convolutional Networks for Semantic Segmentation (2014)
  - A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation (*SegNet*) (2015)

Dilated Convolutions

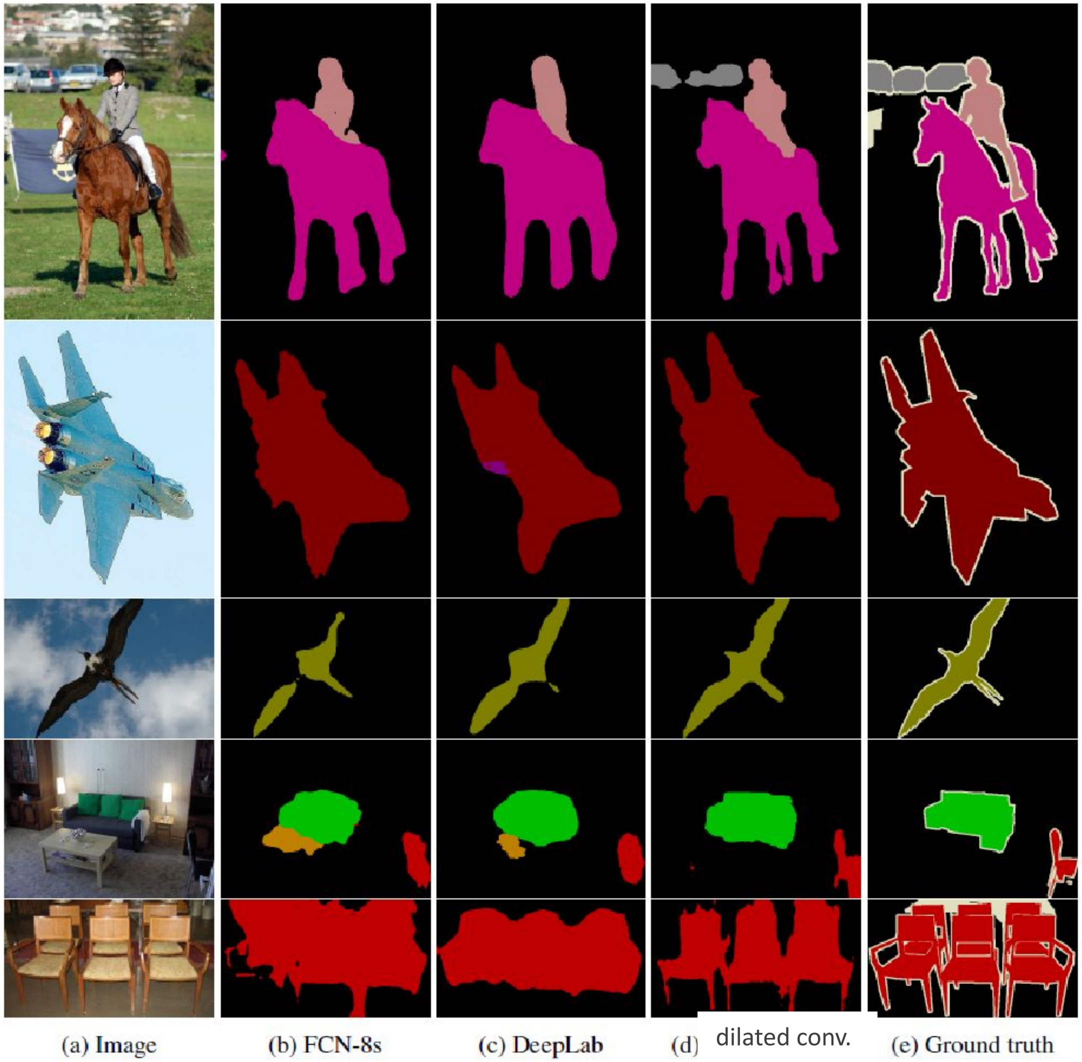


- Large convolutions have a wide receptive field but requires a lot of parameters
- Use dilated (*atrous*) convolutions, to increase the field of view without increasing the spatial dimensions
- The convolution works on samples spaced apart with a regular step instead of over each sing sample in the window.

Multi-Scale Context Aggregation by Dilated Convolutions (2015) [5]

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Examples



Generative Adversarial Networks

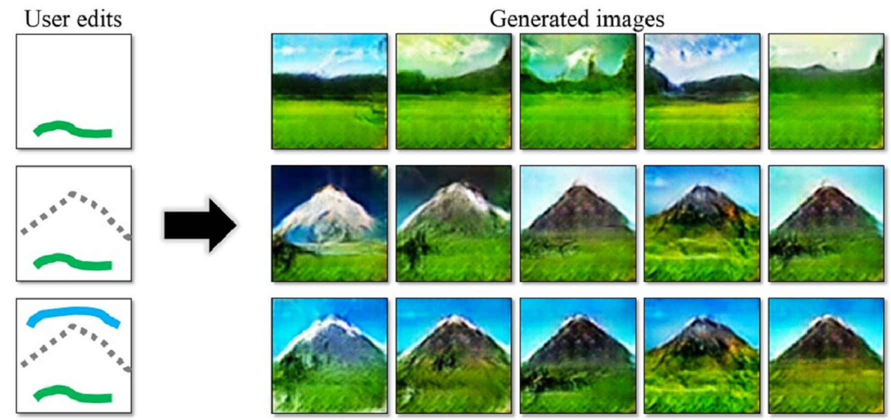
GANs

- Generative
  - Learn a generative model
- Adversarial
  - Trained in an adversarial setting
- Networks
  - Use Deep Neural Networks

Introuduced by Goodfellow et al in 2014 [2]

Generative Models

Which one is Computer generated?

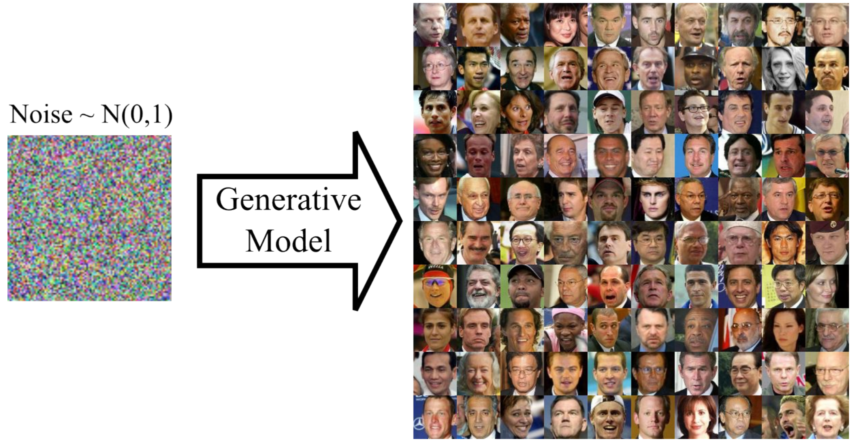


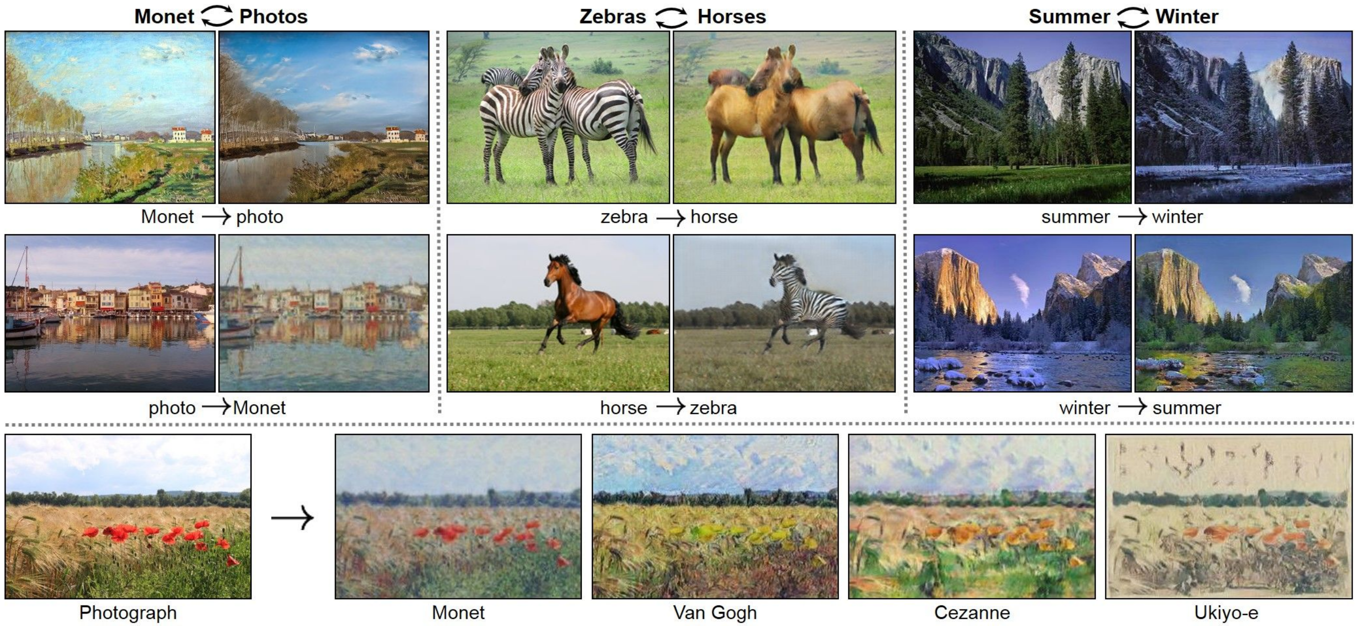
Why Generative Models?

- We have only seen discriminative models so far…
  - Given a vector ***x***, predict a label ***y*** → The model estimates *P(**y**|**x**)*
- Discriminative models can't model *P(**x**)*, i.e. the probability of seeing a certain data sample
  - Thus, can't sample from *P(**x**)*, i.e. *can't generate new data samples*
- Generative models can model *P(**x**)*
  - Can generate new data samples

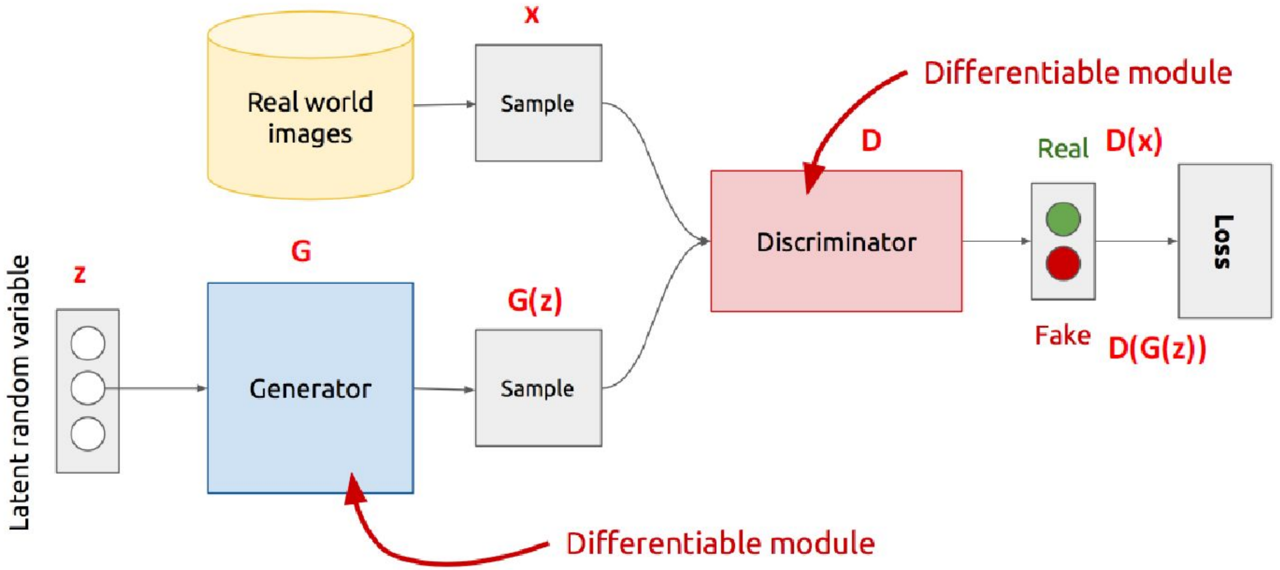
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Generative Models: Examples





Generative Adversarial Networks (GAN)

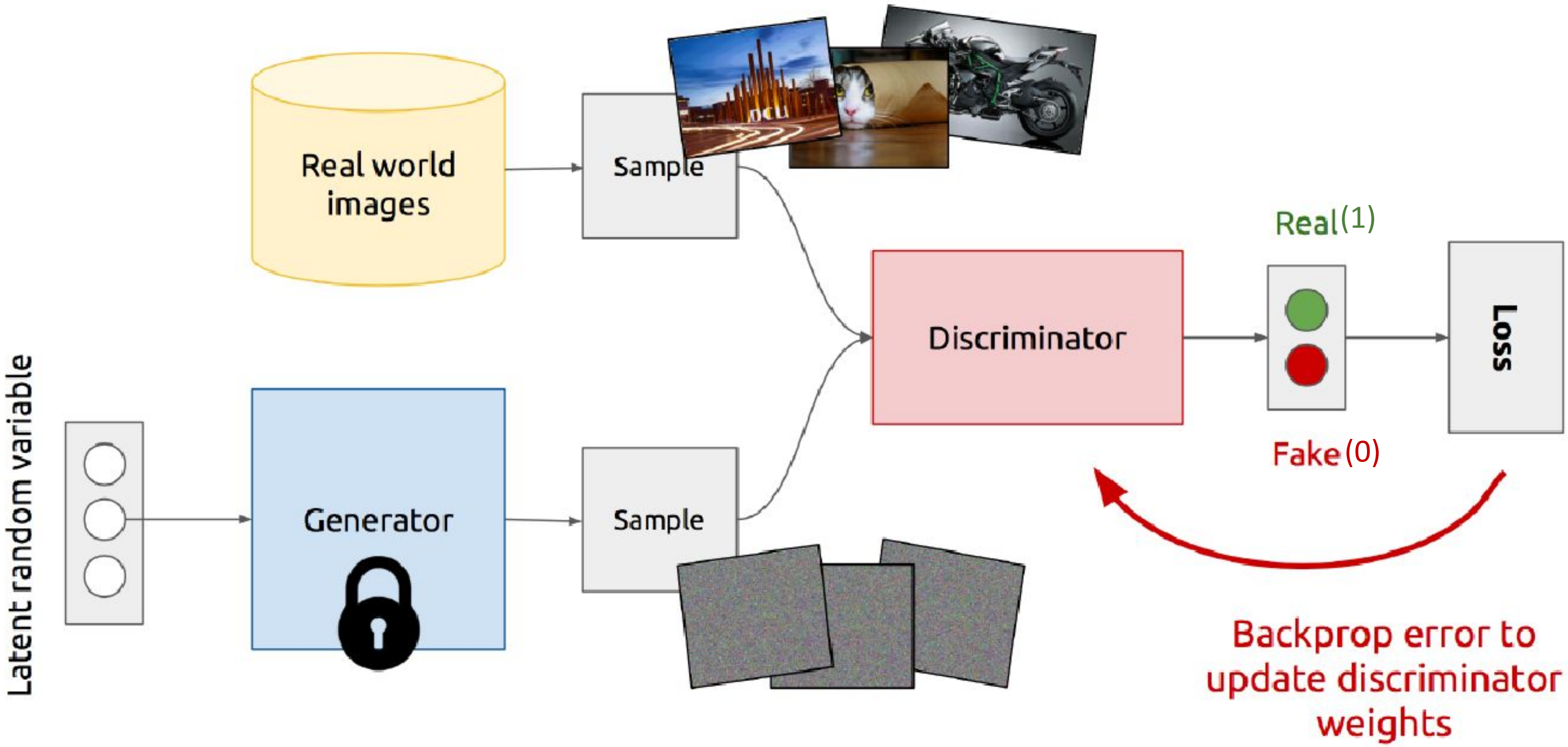


A GAN is composed of two sub-networks:

1. *Generator (G)*: generate fake samples, tries to fool the Discriminator
2. *Discriminator (D):* tries to distinguish between real and fake samples

- Train them against each other (in practice we alternate between training Generator and Discriminator)
- Repeat this and we get better Generator and Discriminator

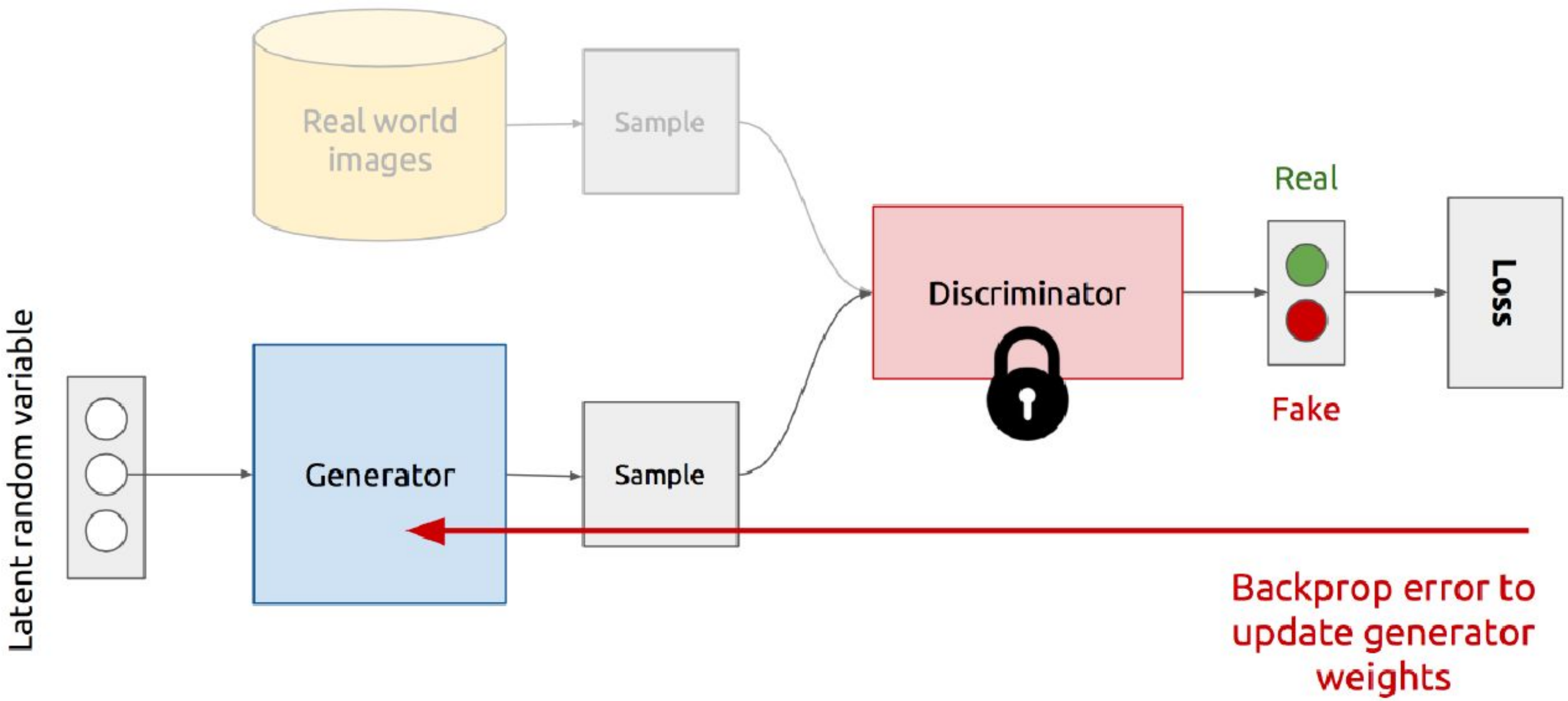
Discriminator Training



Target: Minimize discriminator loss

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Generator Training



Target: Maximize discriminator loss

Loss Function

$$\min_G \max_D V(D, G)$$

- It is formulated as a **minimax game**, where:
  - The Discriminator is trying to maximize its reward $\boldsymbol{V(D, G)}$ (or minimize its loss)
  - The Generator is trying to minimize Discriminator's reward (or maximize its loss)

$$V(D, G) = \underbrace{\boxed{\mathbb{E}_{x \sim p(x)}[\log D(x)]}}_{\text{true samples}} + \underbrace{\boxed{\mathbb{E}_{z \sim q(z)}\left[\log\left(1 - D(G(z))\right)\right]}}_{\text{fake (generated) samples}}$$

- The Nash equilibrium of this particular game is achieved at:
  - $P_{data}(x) = P_{gen}(x) \;\; \forall x$
  - $\mathrm{D}(x) = \frac{1}{2} \;\; \forall x$

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Training Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

Discriminator updates

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

end for

Generator updates

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

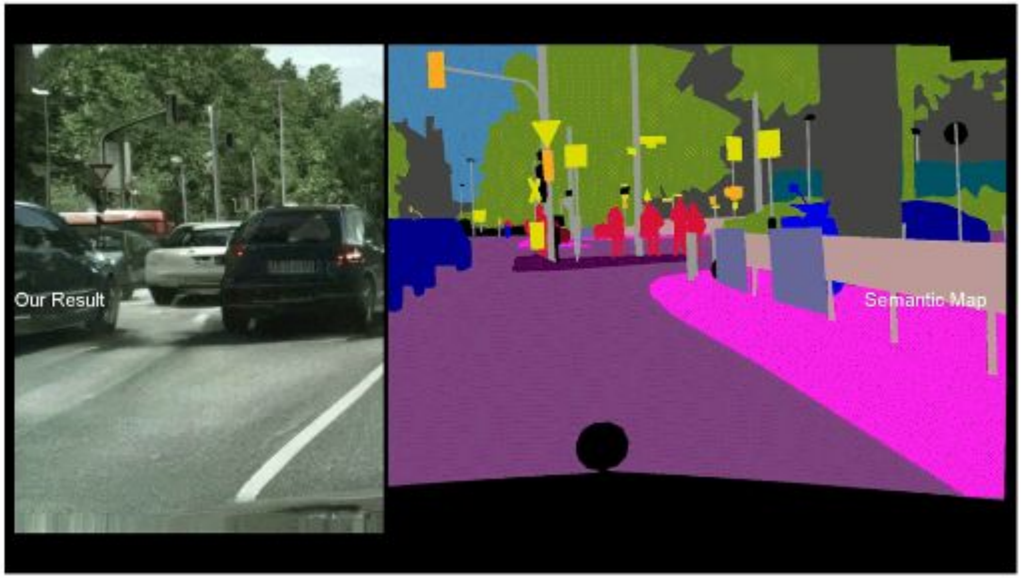
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Examples (1): Generated Images

Examples (2)

(b) Handbag images (input) & **Generated** shoe images (output)

Examples (3)



DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Many Other Approaches….

- This was just a quick overview of some relatively recent results
  - For ICT students more approaches will be presented in computer vision, neural networks and deep learning and many other courses….
- Huge amount of resources is currently spent on Deep Learning research
- Many other schemes exist, and every month there is a new one outperforming previous results

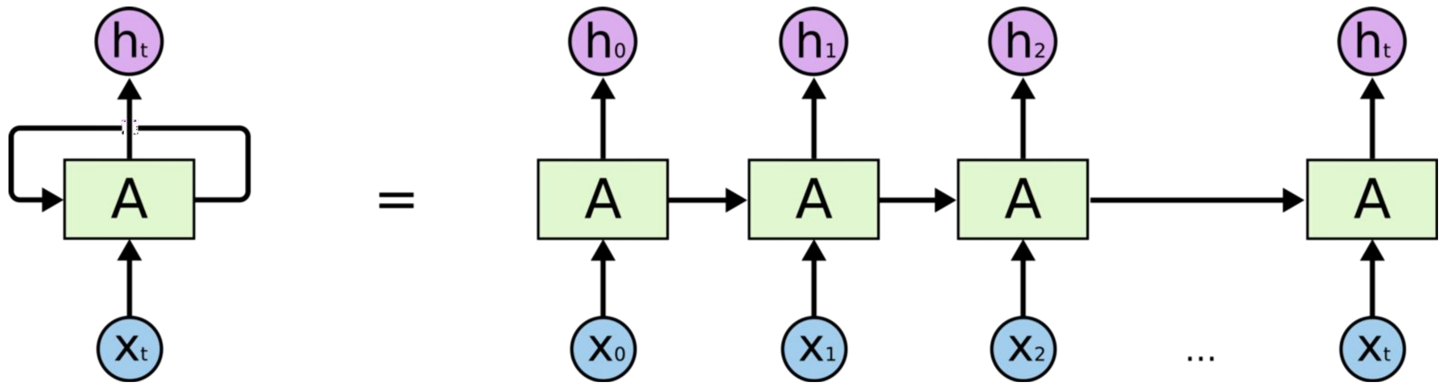
- End of the course material
- RNN/LSTM slides only for personal interest

Exploit Temporal Information

From here on not part of the course

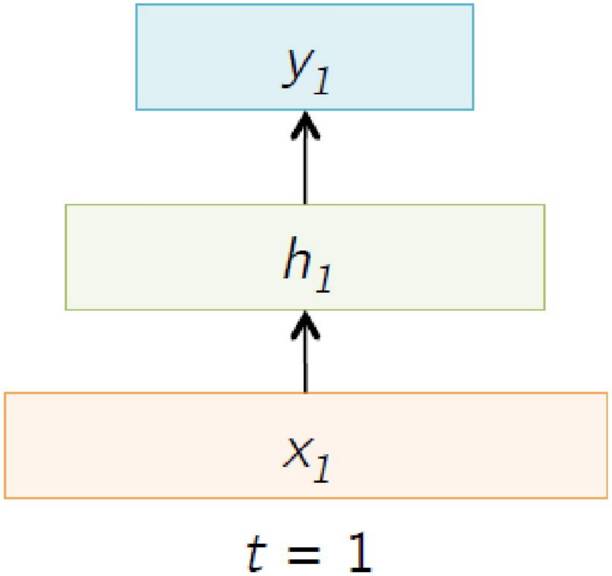
- Not all problem data can be fitted into a representation with fixed-length inputs and outputs !
- Problems such as speech recognition or time-series prediction require a system able to store and use context information
- Example:
  - Sequence of bits: output YES if the number of 1s is even, else NO
    - e.g.,"1000010101" → YES (4 ones), "100011" → NO (3 ones), …
  - Hard/Impossible to choose a fixed context window
  - There can always be a new sample longer than anything seen

Recurrent Neural Networks (RNN)

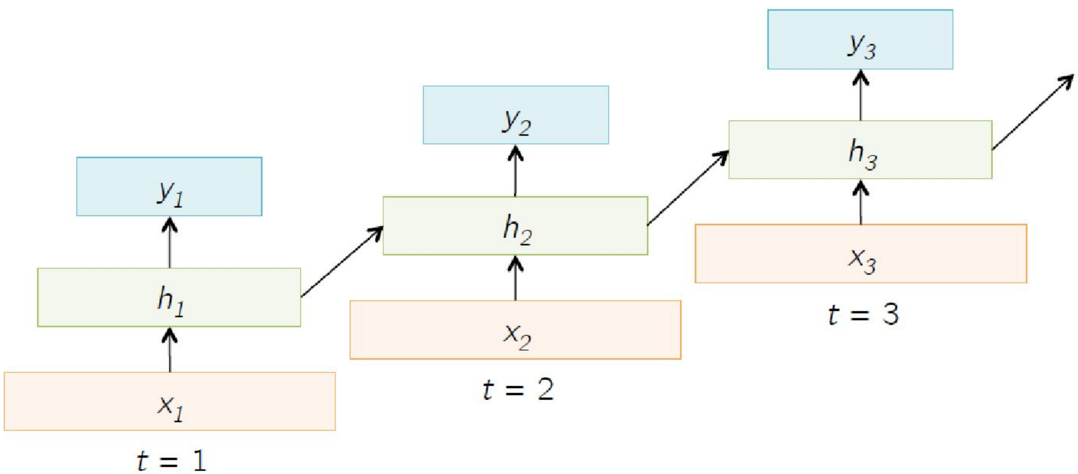


- Recurrent Neural Networks (RNN) take the previous outputs or hidden states as inputs
- The composite input at time t has some historical information about the happenings at times $t' < t$
- RNNs are useful as their intermediate values (state) can store information about past inputs for a time that is not fixed a priori

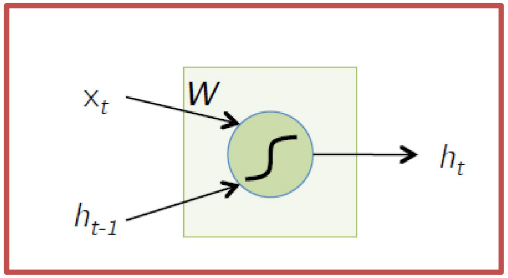
Feedforward vs Recurrent Networks



Sample Feedforward Network

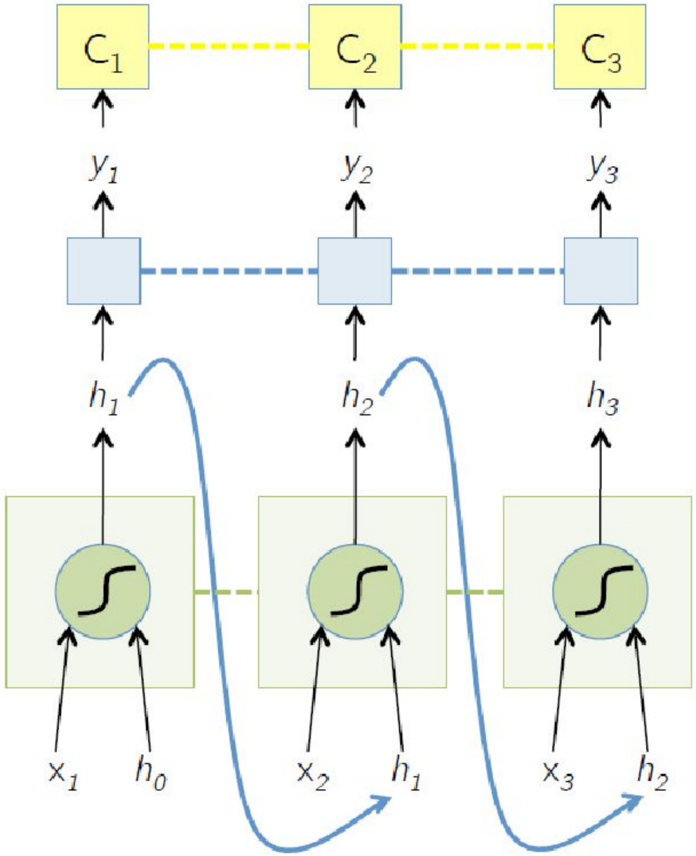


Sample Recurrent Network



Basic RNN cell

Basic RNN model



$$y_t = \mathrm{F}(h_t)$$

$$C_t = \mathrm{Loss}(y_t, \mathrm{GT}_t)$$

------ indicates shared weights

- Note that the weights are ***shared*** over time!
- Essentially, copies of the RNN cell are made over time (unrolling/unfolding), with different inputs at different time steps !

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Example: Image Captioning (1)

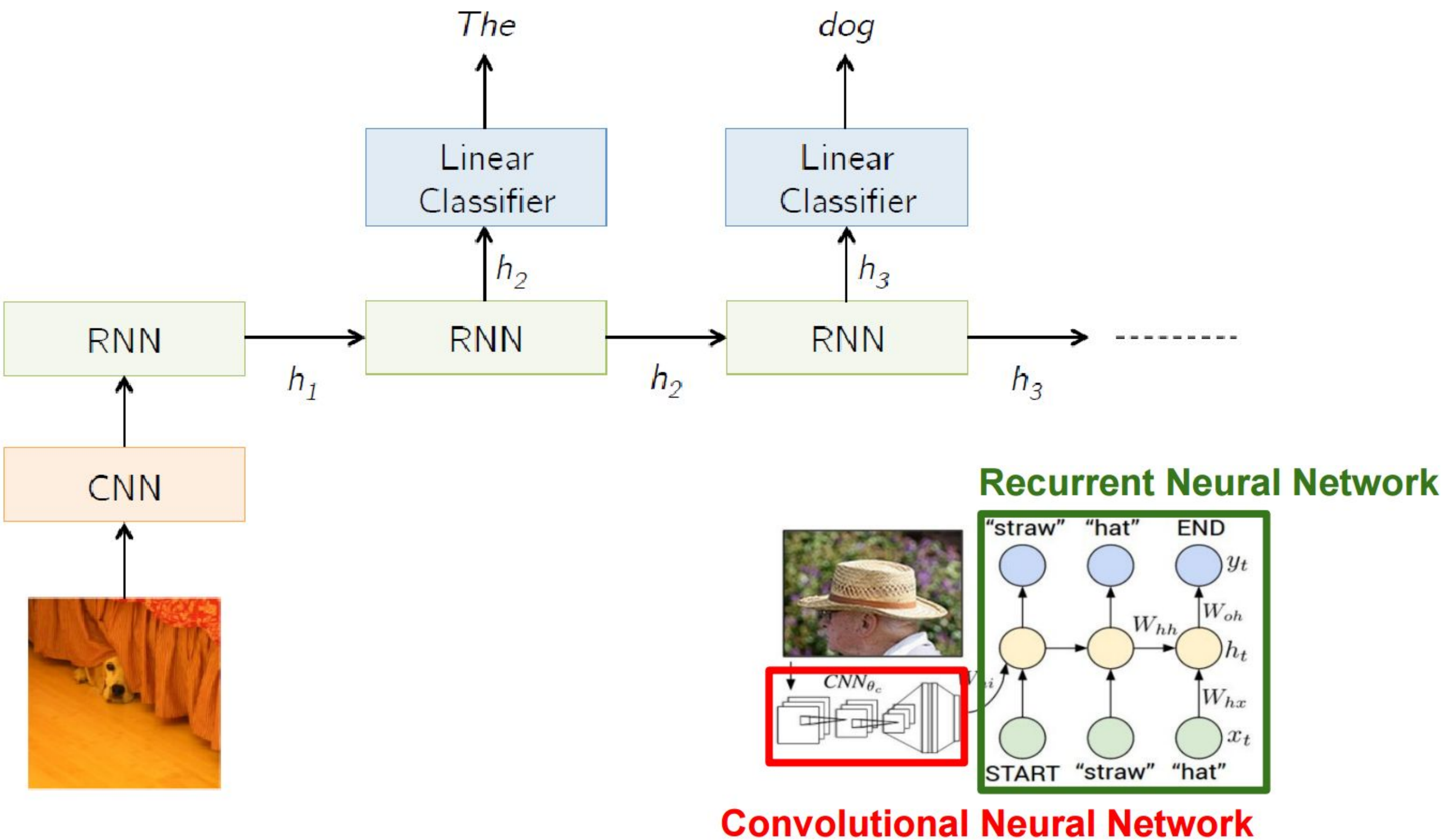
"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

- Given an image produce a sentence describing its content
- *Input:* Image features (e.g., output of a CNN)
- *Output*: Multiple words (e.g., one sentence)

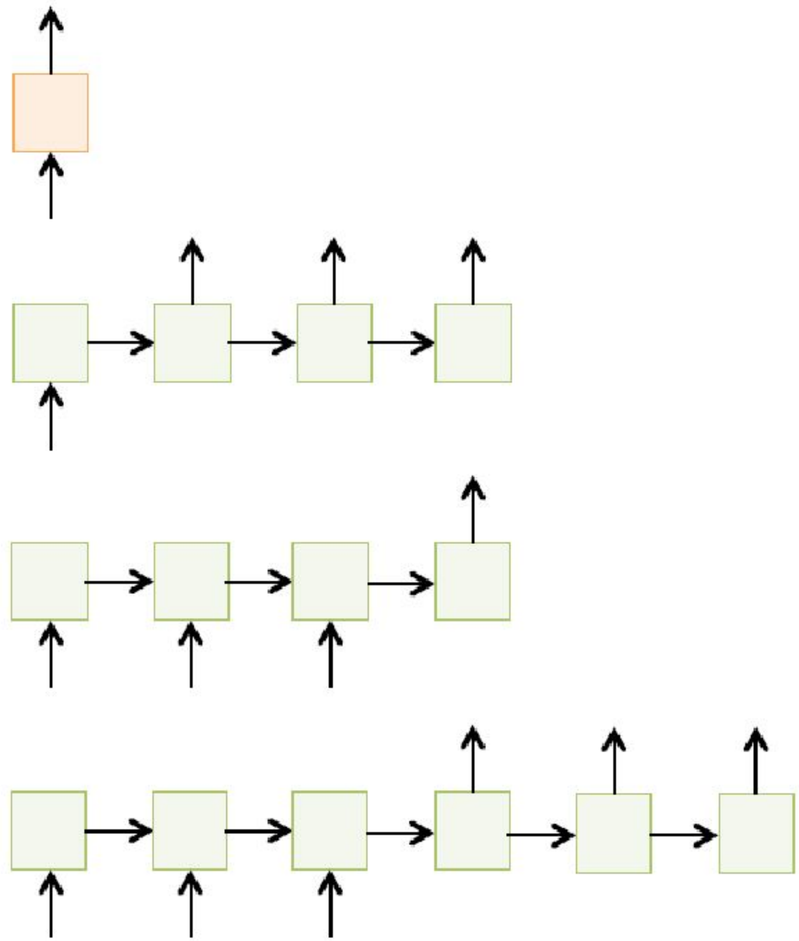
Example: Image Captioning (2)



Example: Image Captioning (3)

Show and Tell: A Neural Image Caption Generator, CVPR 15

Input-Output Scenarios



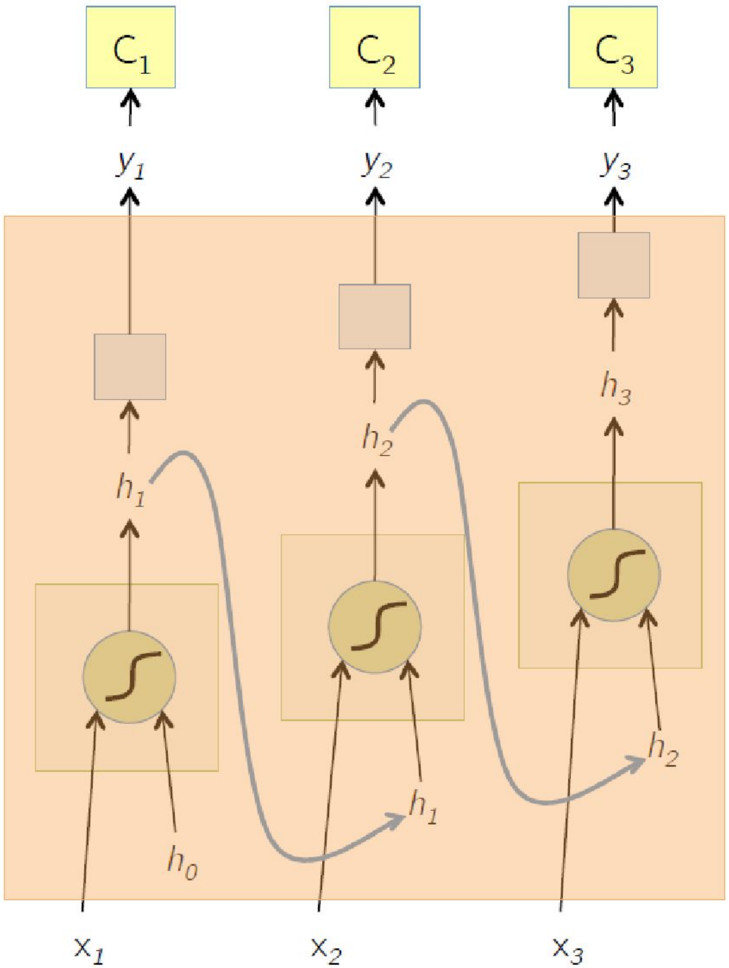
Single - Single — Feed-forward Network

Single - Multiple — Image Captioning

Multiple - Single — Sentiment Classification

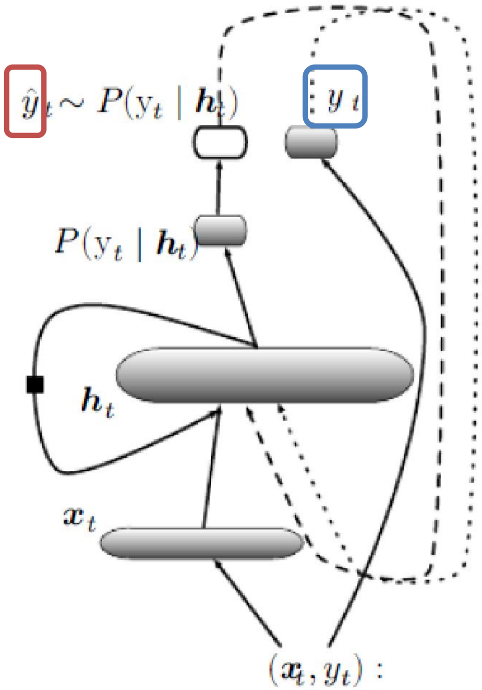
Multiple - Multiple — Translation

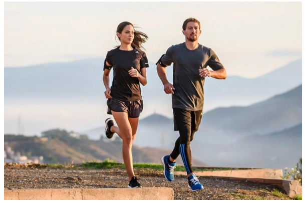
BackPropagation Through Time (BPTT)

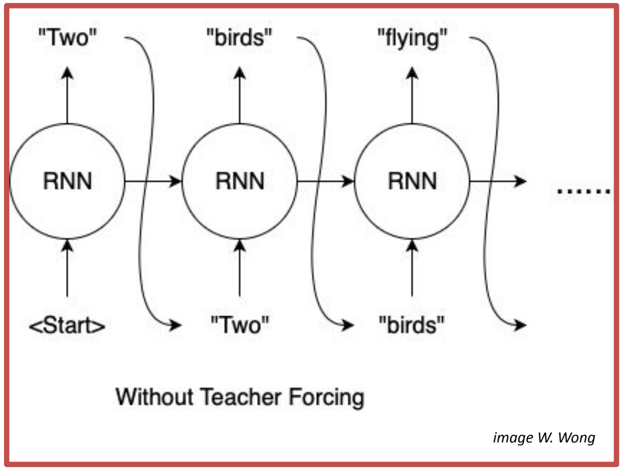


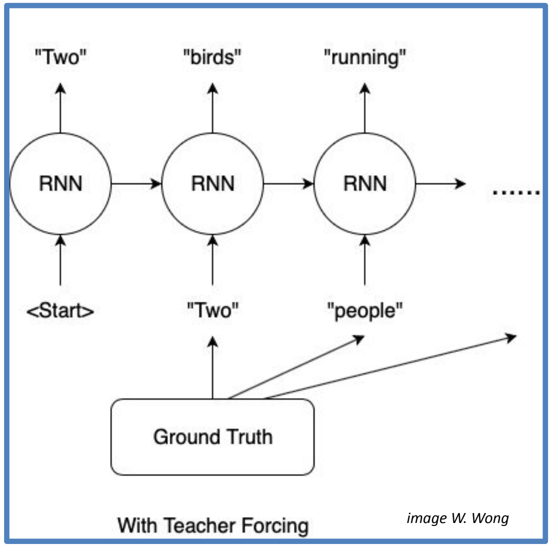
- One of the methods used to train RNNs
- The unfolded network (used during forward pass) is treated as one big feed-forward network!
- This unfolded network accepts the whole time series as input
- The weight updates are computed for each copy in the unfolded network (using standard BackPropagation), then summed (or averaged) and finally applied to the RNN weights
- Training RNN is challenging

Teacher Forcing



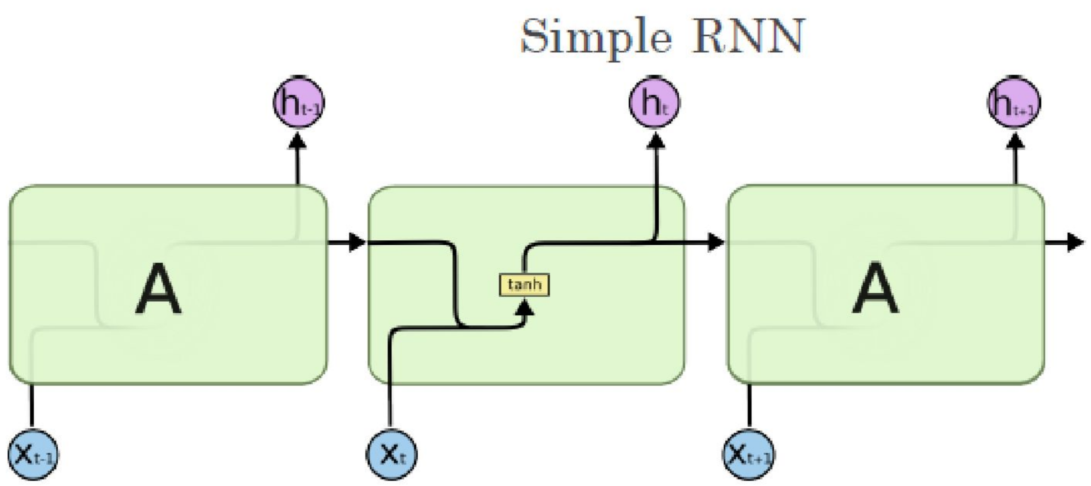






- If the output is used for the hidden state it is possible to choose if using network's output or ground truth labels
- With GT labels (teacher forcing) the model is easier to train
- But generalization properties can be poor

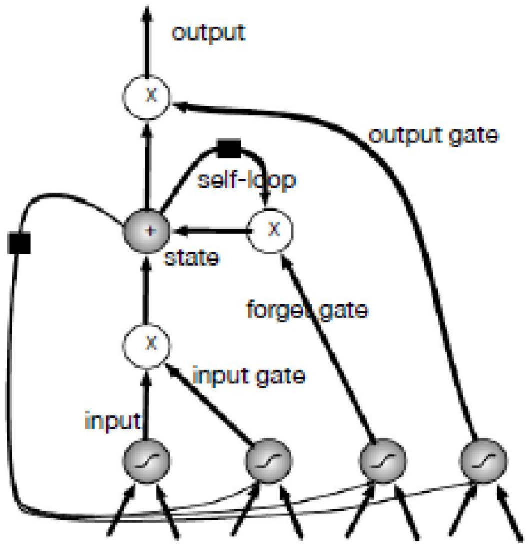
Model Long Time Temporal Relationship

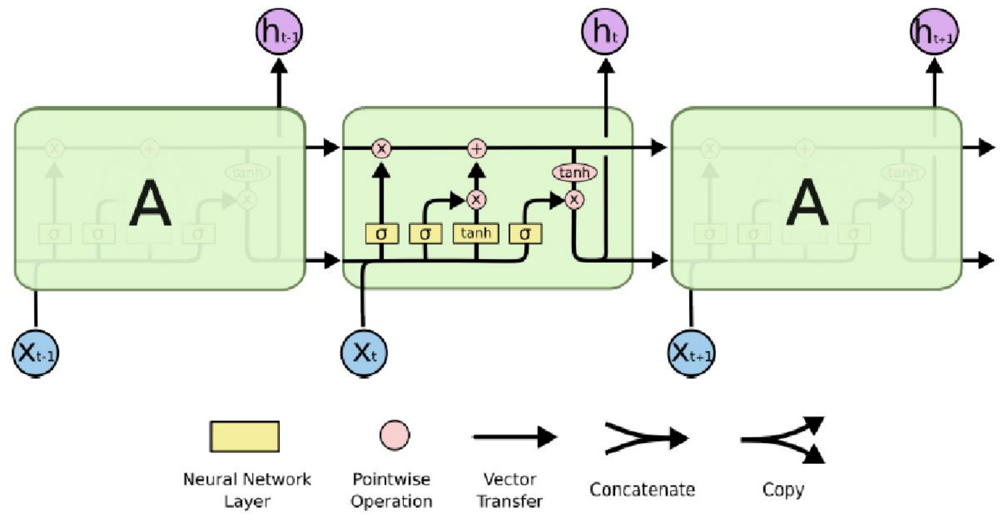


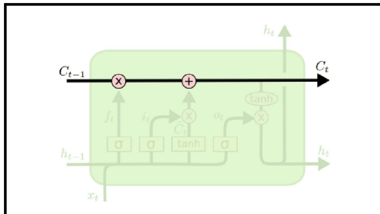
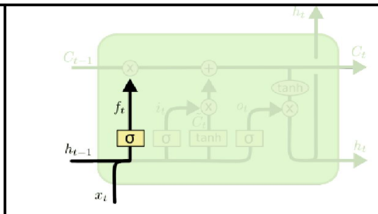
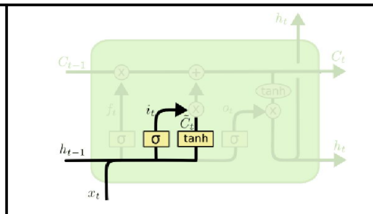
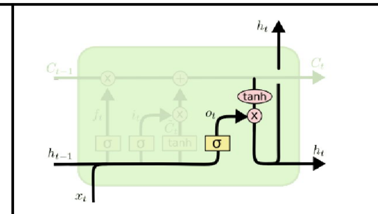
- Baseline RNNs are good for short time temporal relationships
- But they are not able to capture long-time relationships since the gradients vanish or explode
- Also in some applications (e.g., word recognition) a way of "*forgetting*" the state is needed

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Long-Short-Term-Memory (LSTM)





			
Cell state: Taken in input from previous instant and forwarded to the next	*Forget gate:* If set to 0 can delete the content of the cell state	*Input gate:* Enable contribution of new input (the gate can enable/disable the input)	*Output:* Filtered version of state controlled by output gate

Long-Short-Term-Memory (LSTM), Hochreiter & Schmidhuber (1997) [1]

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

References

[1]: S Hochreiter, J Schmidhuber, "*Long short-term memory*", Neural computation 9 (8), 1735-1780, 1997

[2]: Goodfellow, J Pouget-Abadie, M Mirza, B Xu, D Warde-Farley, S Ozair, "*Generative adversarial nets*", Advances in neural information processing systems, 2672-2680, 2014

[3] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "*Imagenet classification with deep convolutional neural network*" Advances in neural information processing systems, 2012

[4]He, K., Zhang, X., Ren, S., & Sun, J. (2016). "*Deep residual learning for image recognition*", In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[5] Yu, F., & Koltun, V. (2015). "*Multi-scale context aggregation by dilated convolutions*", arXiv preprint arXiv:1511.07122.

The papers can be downloaded from elearning