CREATIVE PROGRAMMING AND COMPUTING

Lab 7:

Time-aware generative models and Generative Adversarial Networks (GANs)

Luca Comanducci

luca.comanducci@polimi.it

CONTENTS

- **Part I**
  - Music generation using Recurrent Neural Networks (RNNs)
- **Part II**
  - Image-to-Image Translation with Conditional Adversarial Networks (Pix2Pix)

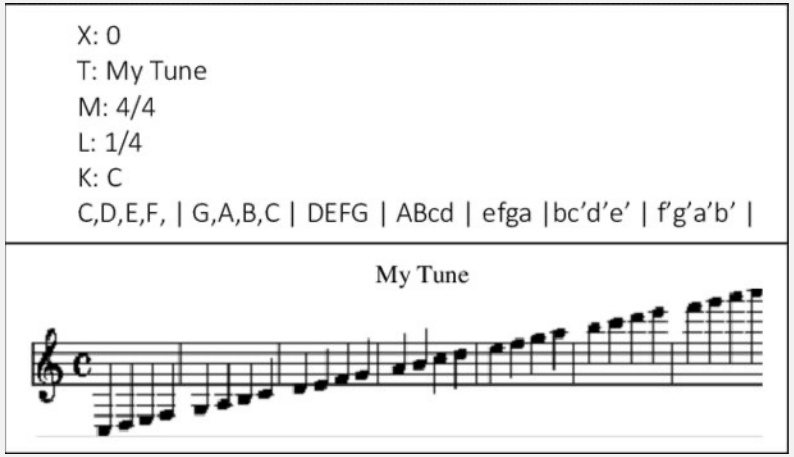
PART I: MUSIC GENERATION WITH RECURRENT NEURAL NETWORKS

- You can find the code in the file:

‘’ Ex1 Music Generation with RNNs.ipynb”

EXERCISE 1: MUSIC GENERATION WITH RECURRENT NEURAL NETWORKS

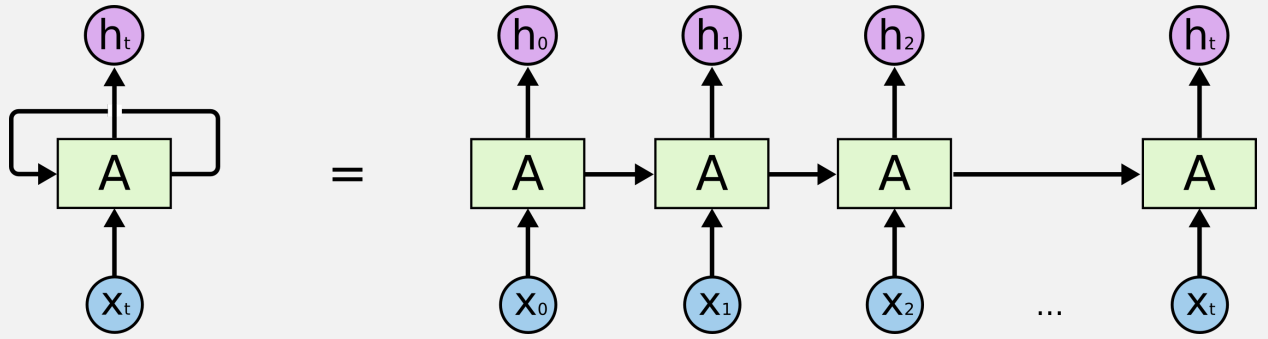
- *Objective*: apply Recurrent Neural Networks (RNNs) for music generation
- *How*: we will train a model to learn patterns contained in raw sheet music in ABC notation, treating it is as a sequence of characters.
- We will consider a dataset made of Irish Folk songs



EXERCISE 1: MUSIC GENERATION WITH RECURRENT NEURAL NETWORKS

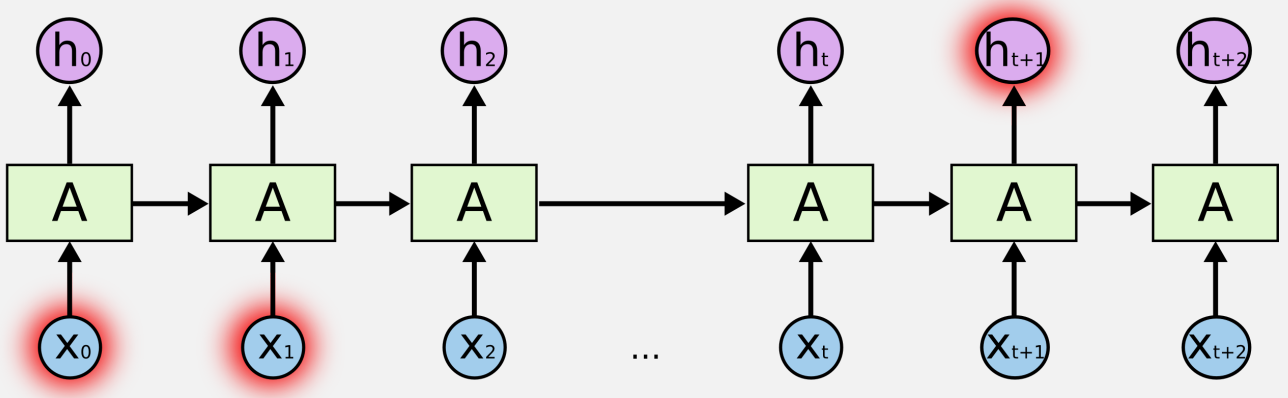
- We will use Long Short Term Memory network Cells in order to model long term dependencies

- RNNs are good at connecting previous information to the current task, but what about long-term dependencies?



EXERCISE I: MUSIC GENERATION WITH RECURRENT NEURAL NETWORKS

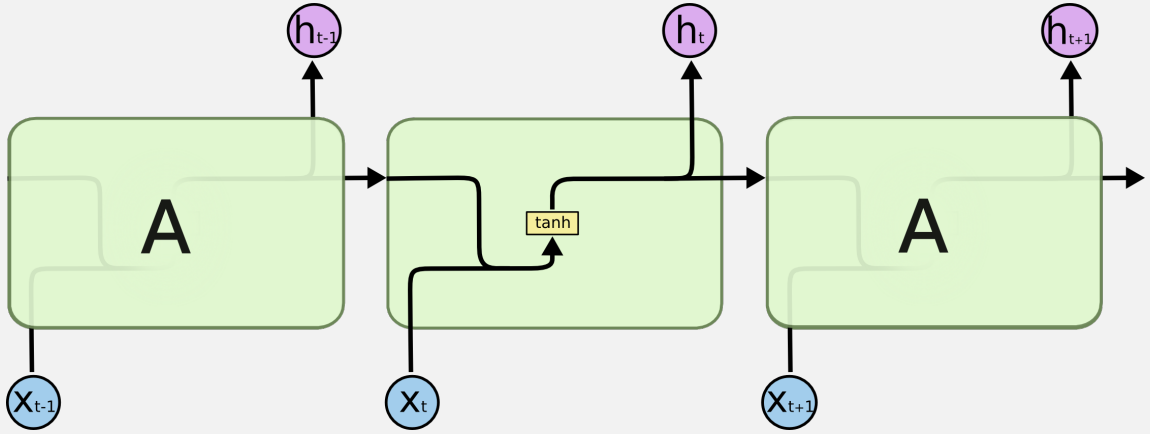
- E.g. imagine you want to predict the last word "French" in a long text: ''I grew up in France, … I speak fluent French"
- Recent info suggests: the word is probably the name of a language
- Long-term info suggests: French
- As the gap grows bigger RNNs struggle to connect the information



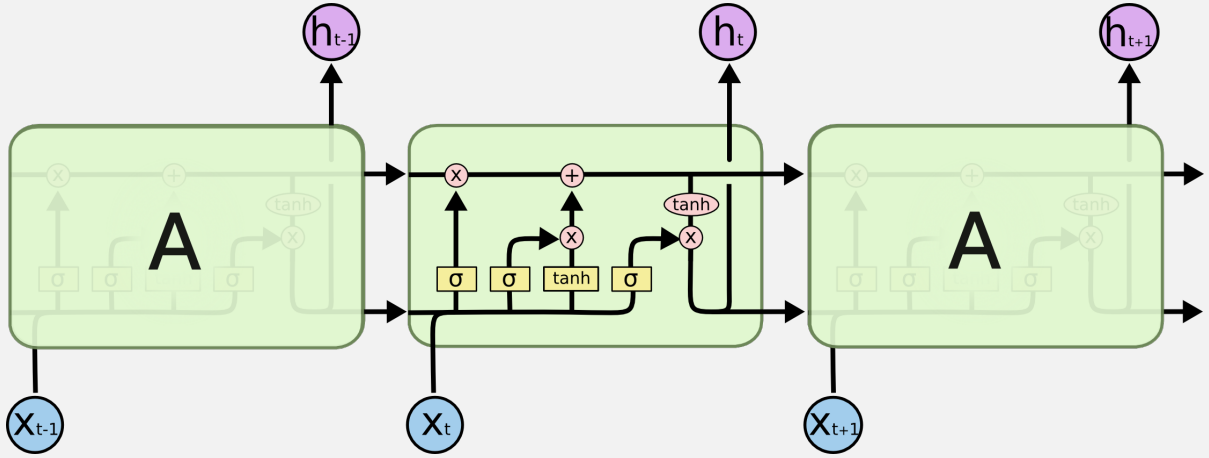
EXERCISE I: MUSIC GENERATION WITH RECURRENT NEURAL NETWORKS

- This is why Long Short Term Memory (LSTM) networks were intoduced
- The fundamental idea is that LSTM are able to select which info must pass through at each time instant

RNN

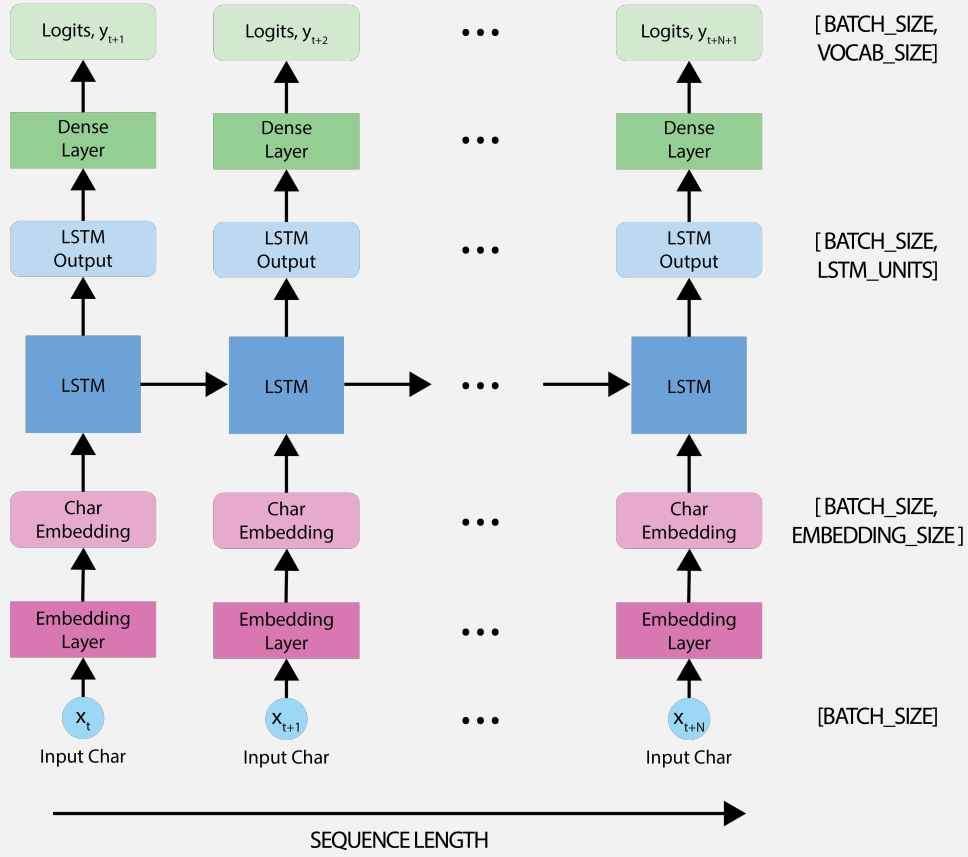


LSTM



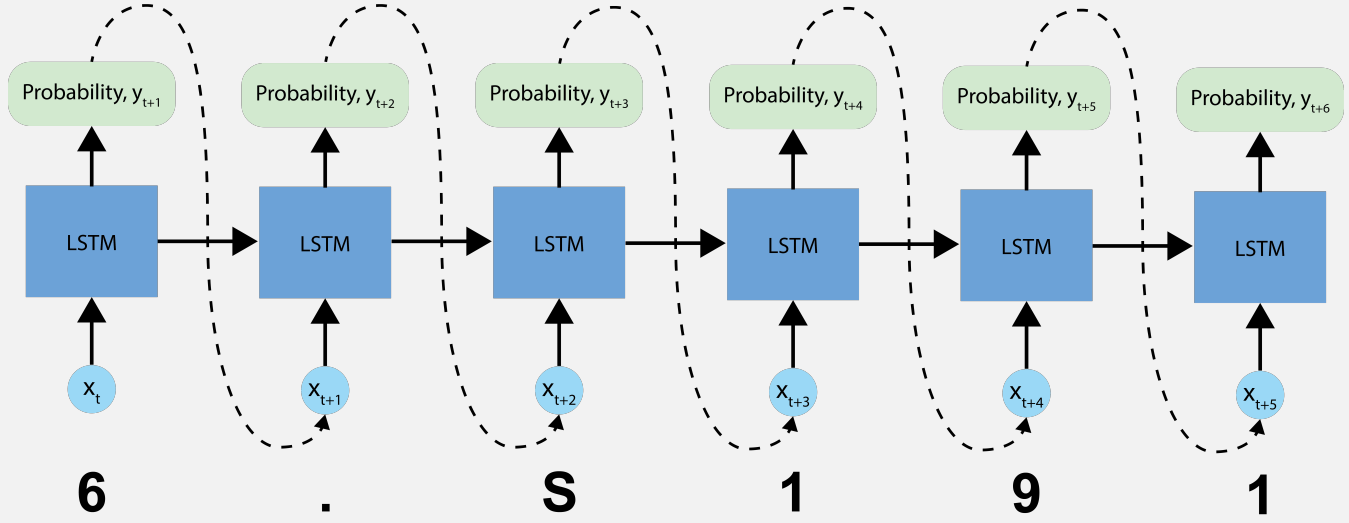
EXERCISE 1: MUSIC GENERATION WITH RECURRENT NEURAL NETWORKS

- How does the model work?
  - One-to-one mapping between ABC notation and numbers
  - Network gets a sequence of characters and outputs the probability of each character in being the next one
  - This means that we are generating an output distribution corresponding to a categorical distribution (Generalized Bernoulli Distribution), from which we can sample the next character (i.e. note or other symbol in ABC notation)



EXERCISE I: MUSIC GENERATION WITH RECURRENT NEURAL NETWORKS

- How does the prediction procedure work?
  - Initialize a "seed" start string to obtain probability distribution over next predicted character.
  - Sample from the multinomial distribution to calculate next predicted character, which will be next input to model.
  - At each time step updated RNN state is fed back into the model, learning sequence dependencies.



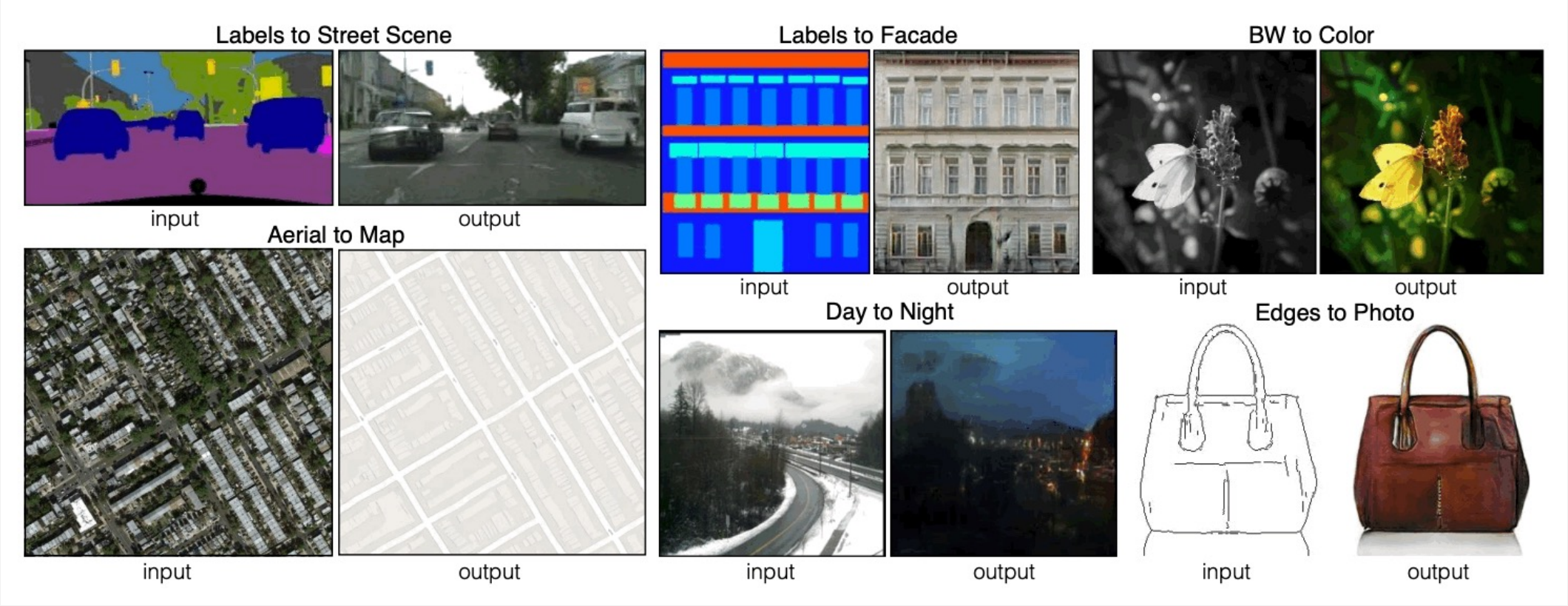
PART II: IMAGE-TO-IMAGE TRANSLATION (PIX2PIX)

- You can find the code in the file:

'' ExII Image-to-Image translation pix2pix.ipynb"

PIX2PIX

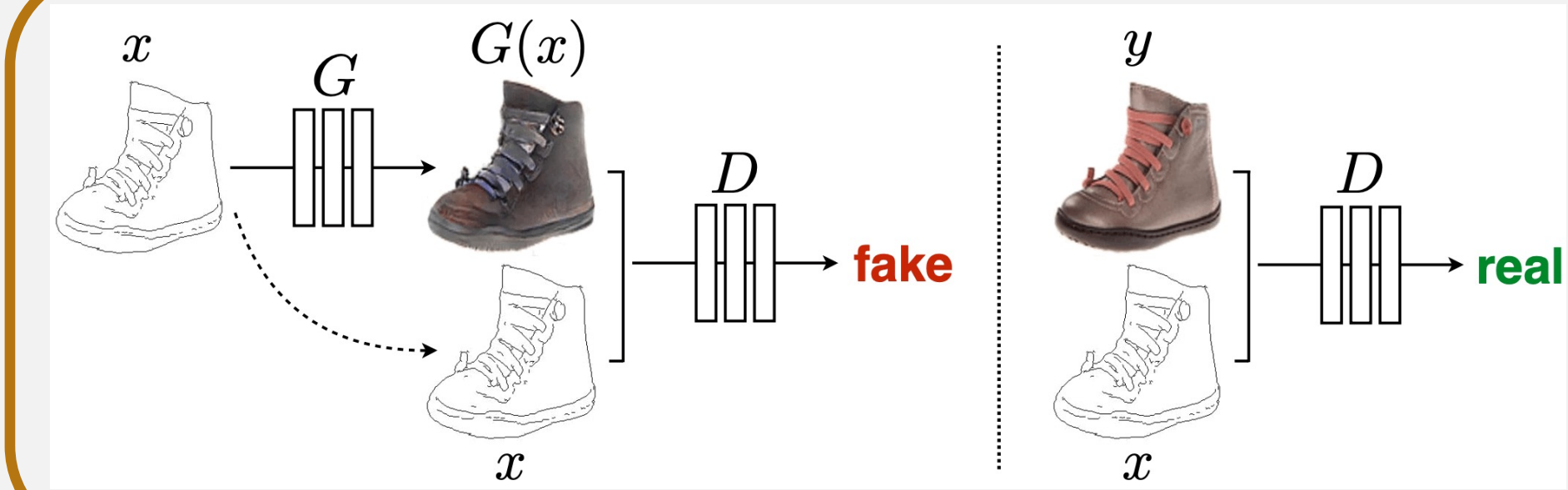
- Image-to-Image Translation with **Conditional Adversarial Networks**
  - Convert an image into another one
  - GAN is conditioned on the input image



PIX2PIX

- A naive approach such as CNN + Euclidean distance as Loss would produce blurry results
- GANs do the following:
  - Set a high level goal: ''make output indistinguishable from reality''
  - Automatically learn the loss function to do it

Training Procedure



PIX2PIX

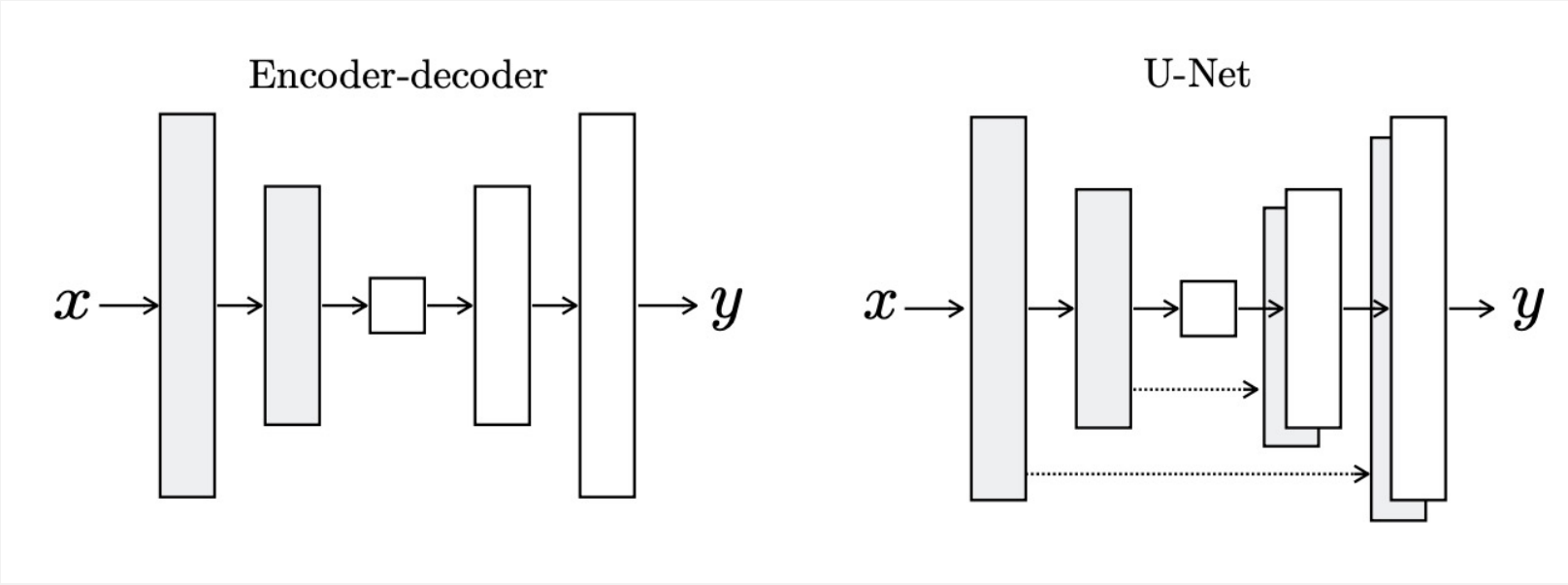
- Objective of a conditional GAN:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]$$

- The conditioning is given by the fact that the discriminator observes both input and desired image

PIX2PIX

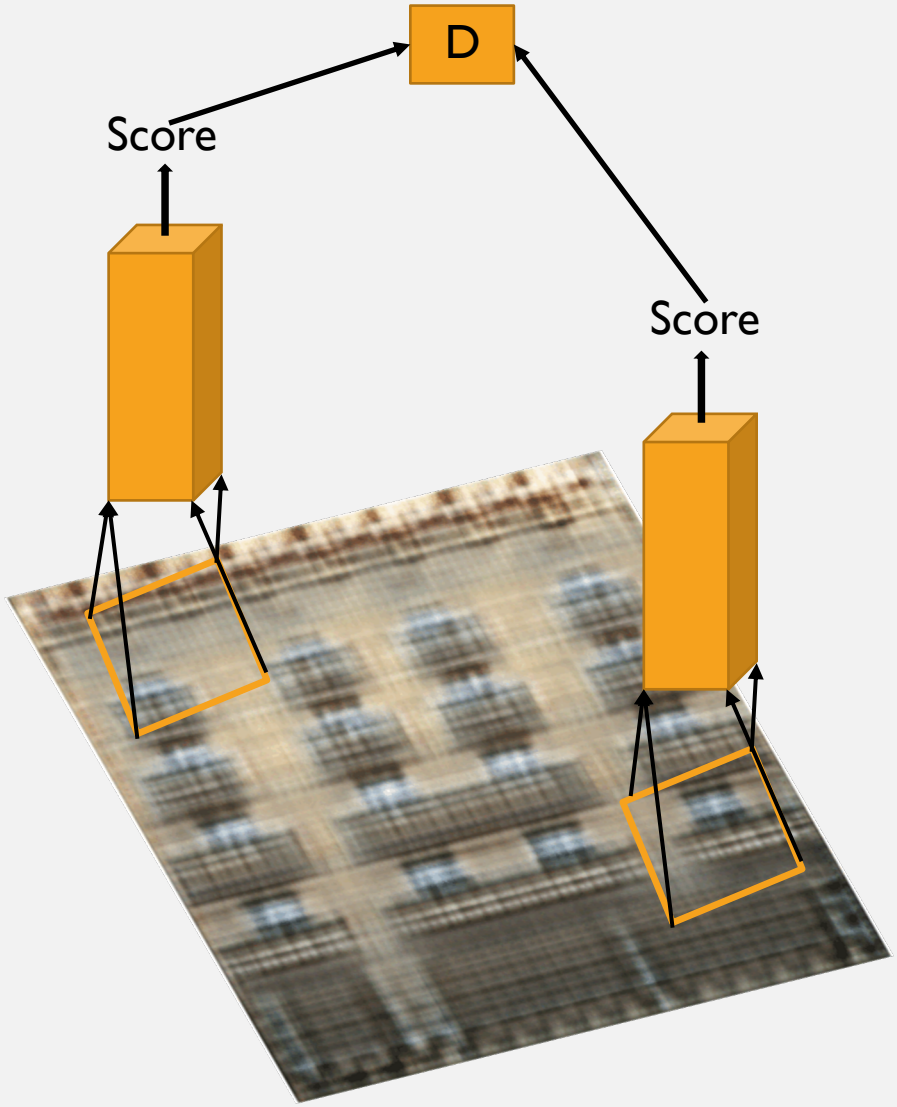
- **Generator**:
  - **U-Net** architecture
  - Autoencoder with skipped connections
  - both input and output are renderings of the same underlying structure



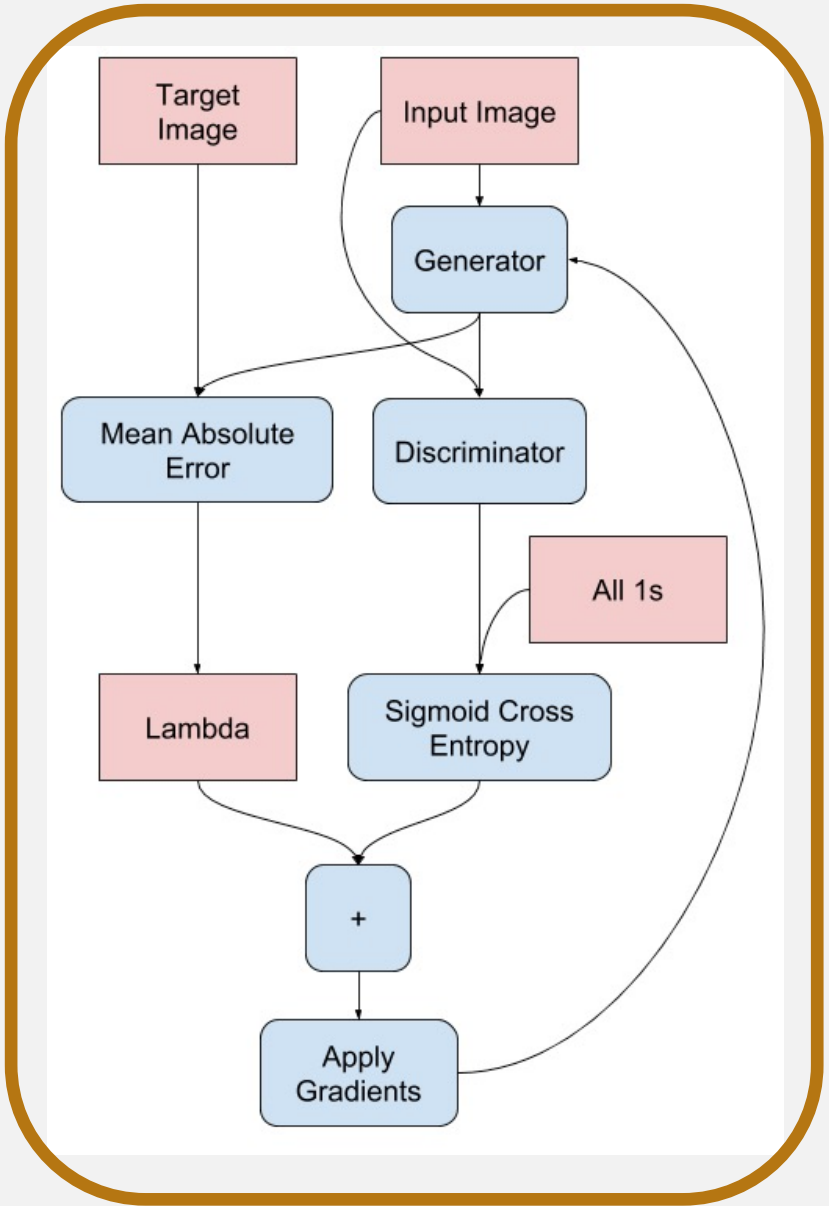
- In many image translation problems a lot of low-level information is shared between input and output
- Skip connections between each layer i and $n - i$ cirmunvents this issue

PIX2PIX

- **Discriminator**:
  - Patch GAN
    - Penalizes structure at the scale of patches
    - Tries to classify if each $N \times N$ patch in an image is real or fake
    - Run across the image, averaging all responses to provide ultimate output of the Discriminator
    - Can be understood as a form of texture/style loss

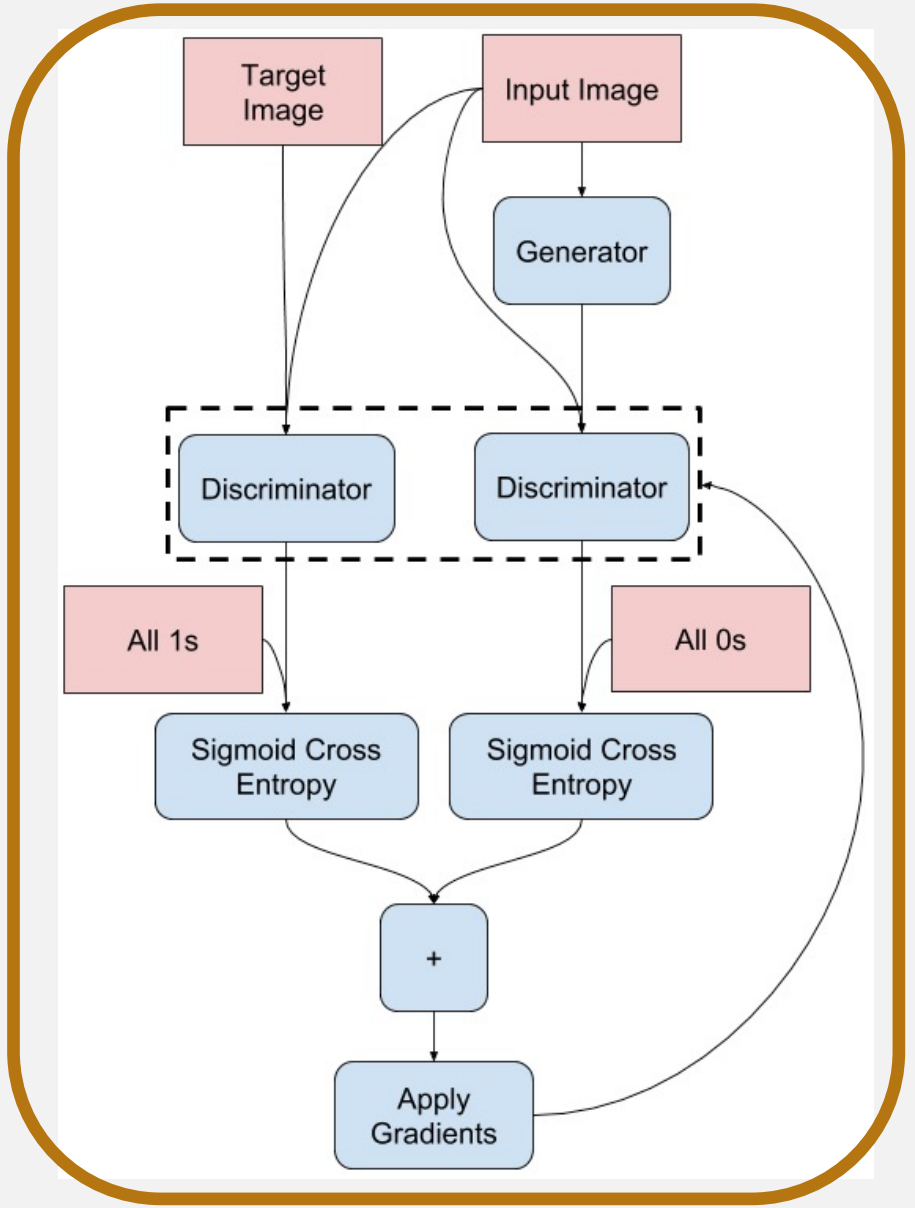


PIX2PIX



Generator training

Discriminator training



PIX2PIX

PIX2PIX

Background removal

by Kaihu Chen

Palette generation

by Jack Qiao

Sketch→ Portrait

by Mario Klingemann

Sketch → Pokemon

by Bertrand Gondouin

"Do as I do"

by Brannon Dorsey

#fotogenerator

sketch by Yann LeCun

REFERENCES

- **Part I**
  - **Music generation using Recurrent Neural Networks (RNNs)**
    - MIT 6.S191 Introduction to Deep Learning – Lab 1
  - Music Generation using RNNs (MIDI)
  - Performance RNN
  - Music Transformer
- **Sketch RNN -** Original paper - Blog post- Demo- Sketch RNN and tensorflow.js
- **LSTM** – blog post

REFERENCES

- **Part II**
  - **Pix2Pix**
    - Original paper - Tensorflow tutorial - Image-to-image Demo
  - GANSynth
  - Original paper - Blog post - Audio Examples