Deep learning for medical imaging school 2022

Hands-on session

Autoencoders

By

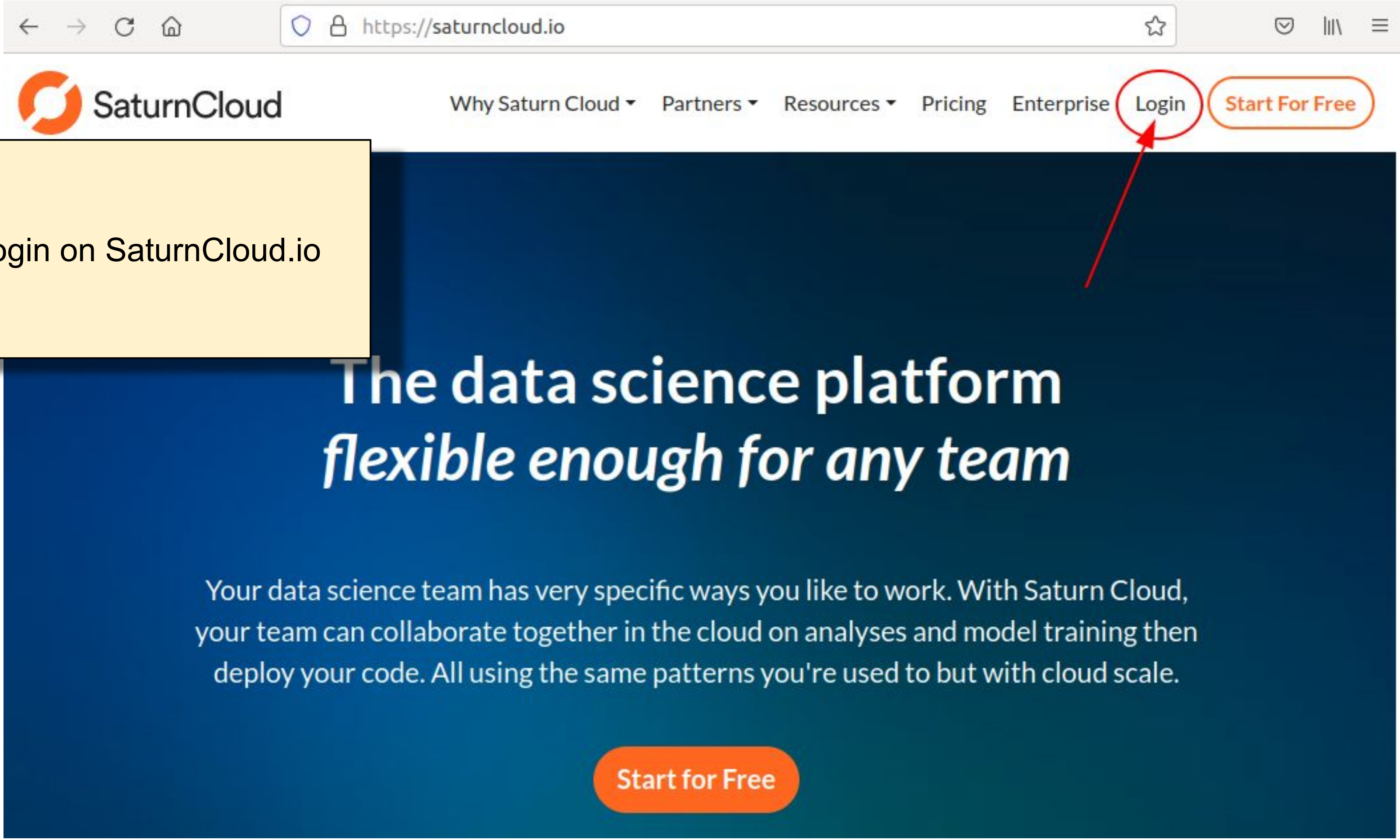
Pierre-Marc Jodoin, Nathan Painchaud

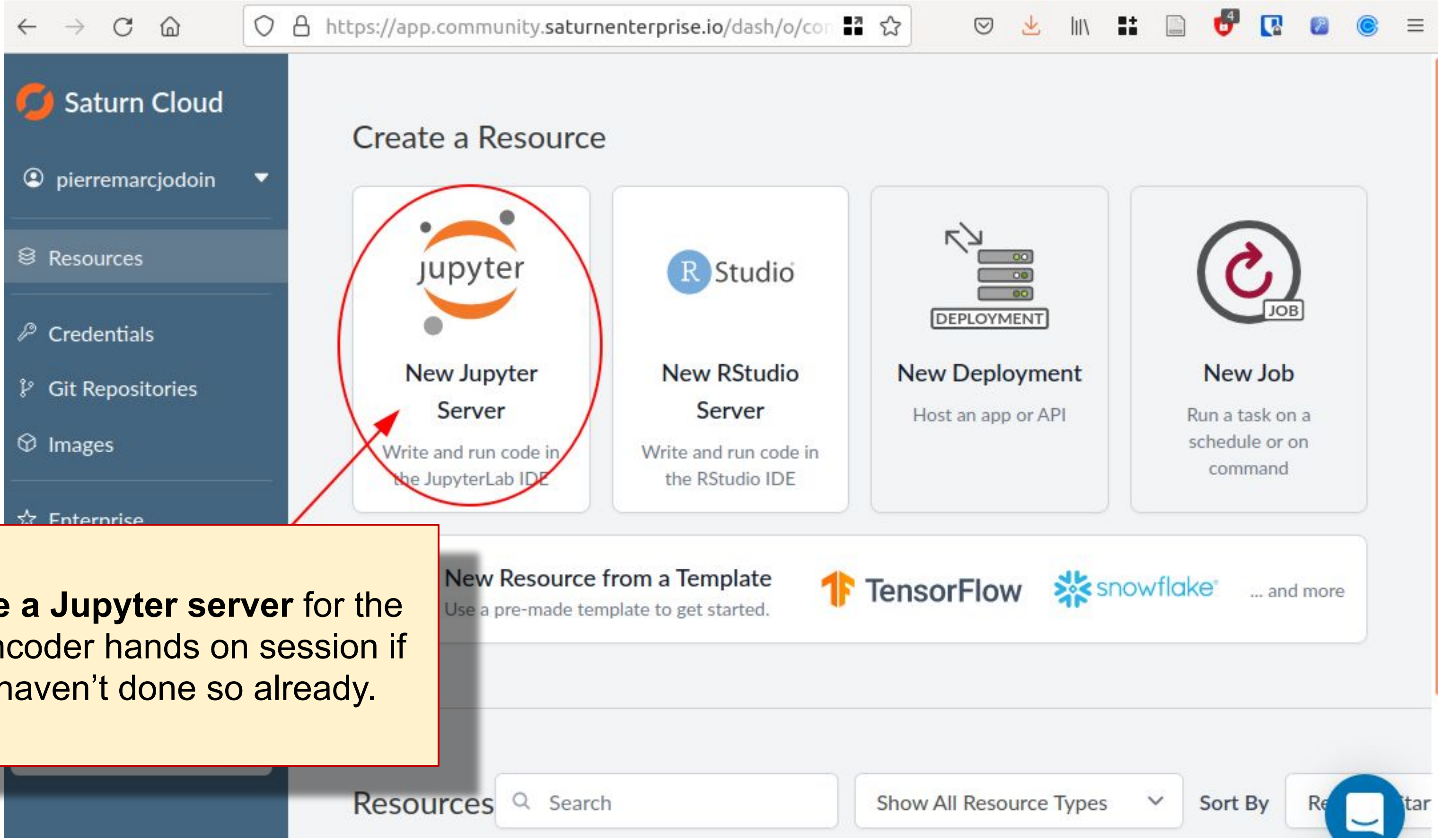
With the support of

Thomas Grenier

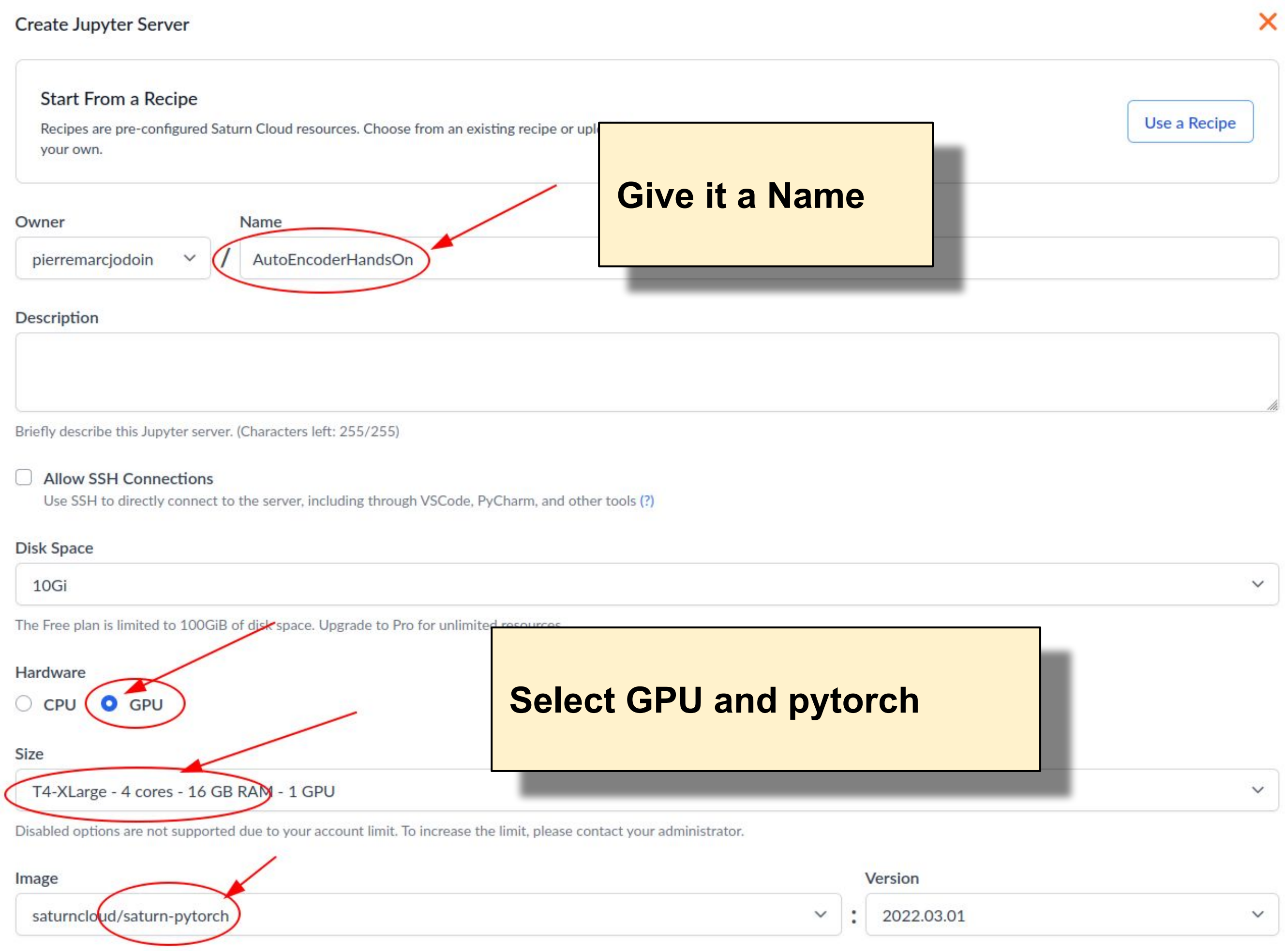
Setup your **SaturnCloud** environment

First, login on SaturnCloud.io





Create a Jupyter server for the autoencoder hands on session if you haven't done so already.


Create Jupyter Server
Start From a Recipe
Recipes are pre-configured Saturn Cloud resources. Choose from an existing recipe or upl
your own.
Use a Recipe
Give it a Name
Owner
Name
pierremarcjodoin
AutoEncoderHandsOn
Description
Briefly describe this Jupyter server. (Characters left: 255/255)
Allow SSH Connections
Use SSH to directly connect to the server, including through VSCode, PyCharm, and other tools (?)
Disk Space
10Gi
The Free plan is limited to 100GiB of disk space. Upgrade to Pro for unlimited resources.
Hardware
CPU
GPU
Select GPU and pytorch
Size
T4-XLarge - 4 cores - 16 GB RAM - 1 GPU
Disabled options are not supported due to your account limit. To increase the limit, please contact your administrator.
Image
saturncloud/saturn-pytorch
Version
2022.03.01

Image

saturncloud/saturn-pytorch

Version

2022.03.01

Spot Instance

This requests the server of specified size as Spot Instance. Spot Instances are less expensive, but may be shut down at any time (with a two-minute warning). (?)

Working Directory

/home/jovyan/workspace

Write opencv-python torchvision==0.11.2 for pip install

Extra Packages

Extra packages are installed every tim

If you find yourself adding the same packages to lots of resources, you may want to permanently add packages to a custom image instead. (?)

Conda Install | Pip Install | Apt Packages

```
opencv-python torchvision==0.11.2
```

The packages together will run the following script:

```
apt-get install htop zip unzip python3-opencv
pip install opencv-python torchvision==0.11.2
```

Shutoff After

1 hour

Disabled options are not supported in the Free plan. Upgrade to Pro for unlimited resources.

Advanced Settings (optional)

Save Cancel

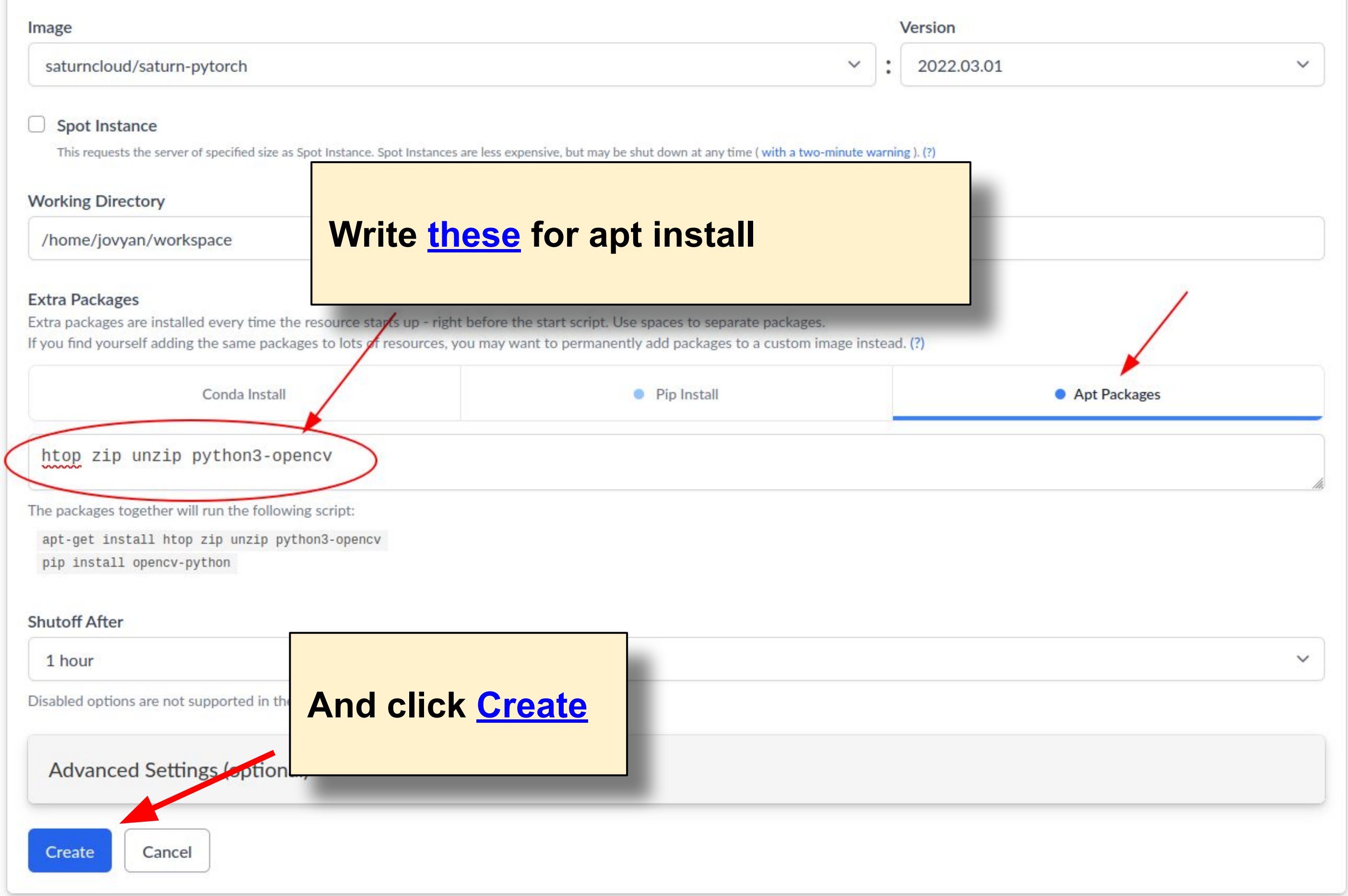
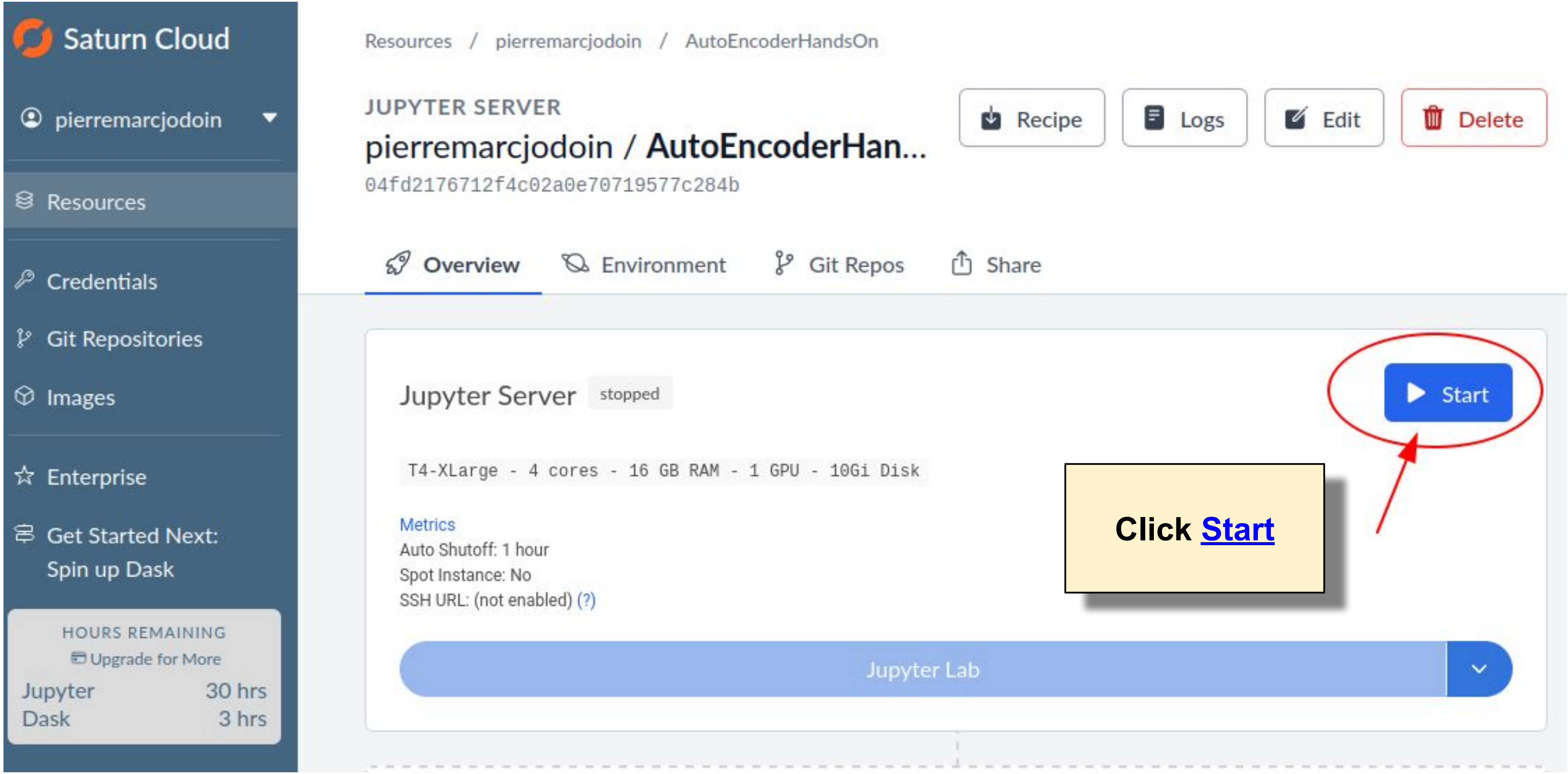
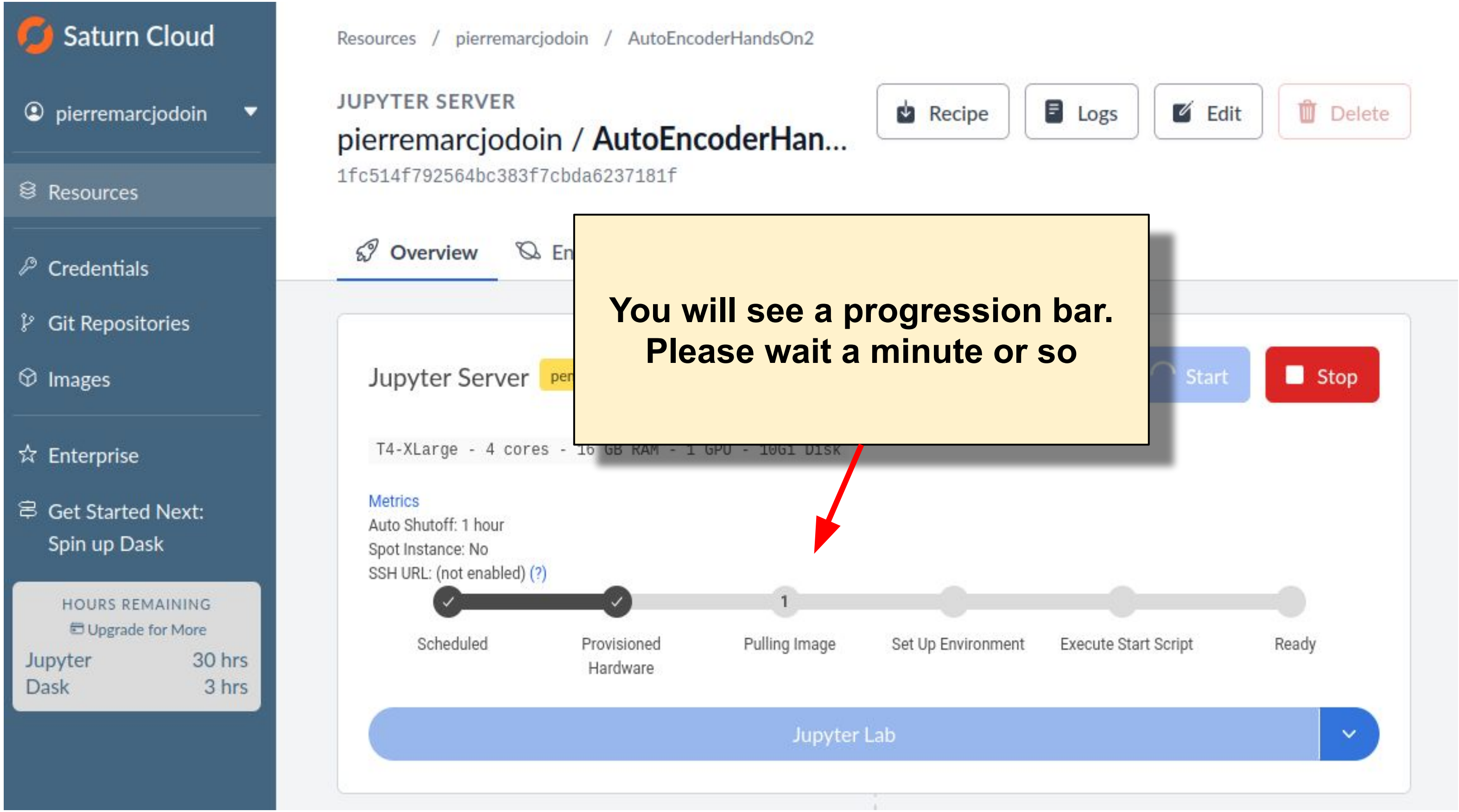
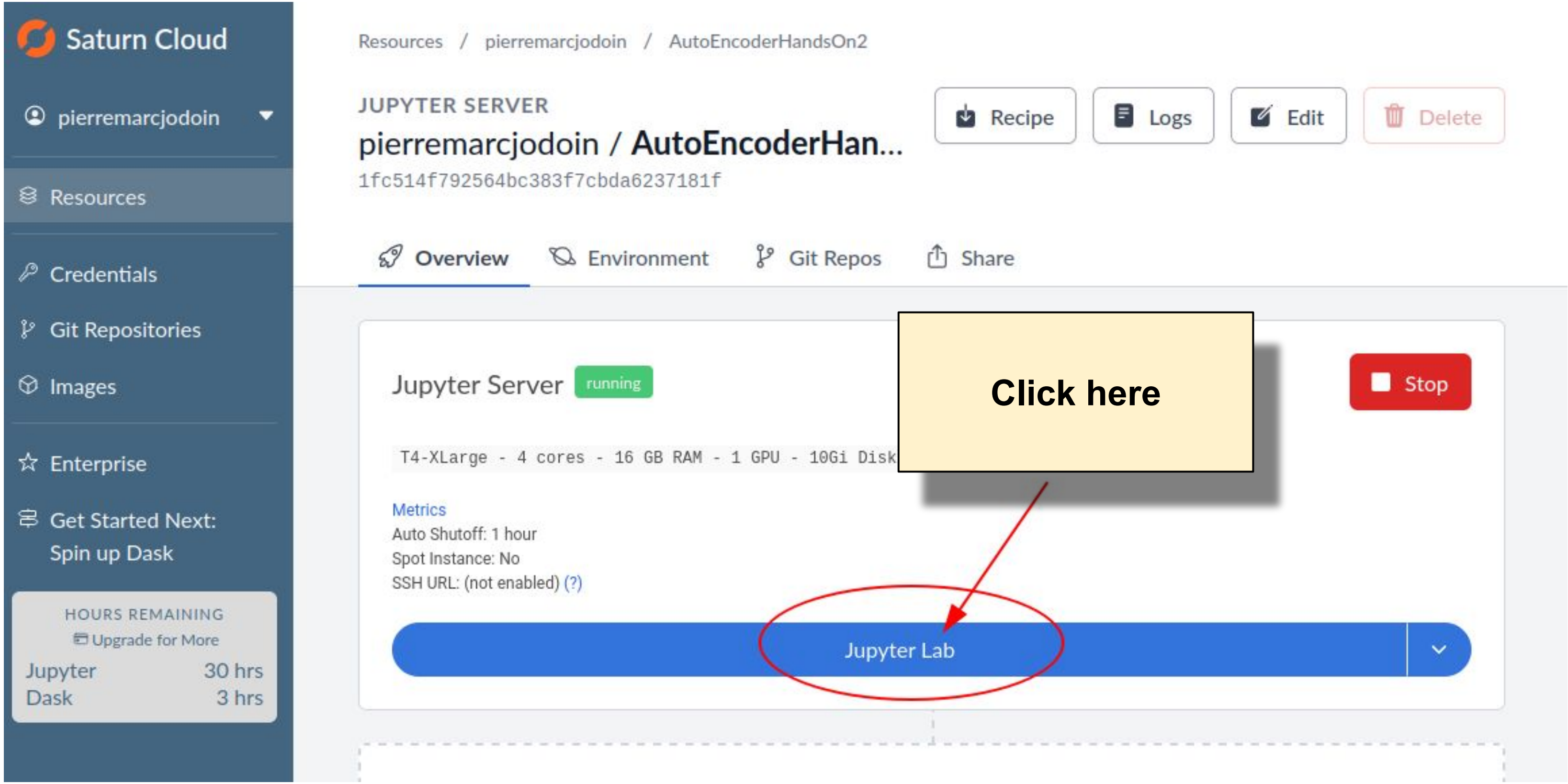
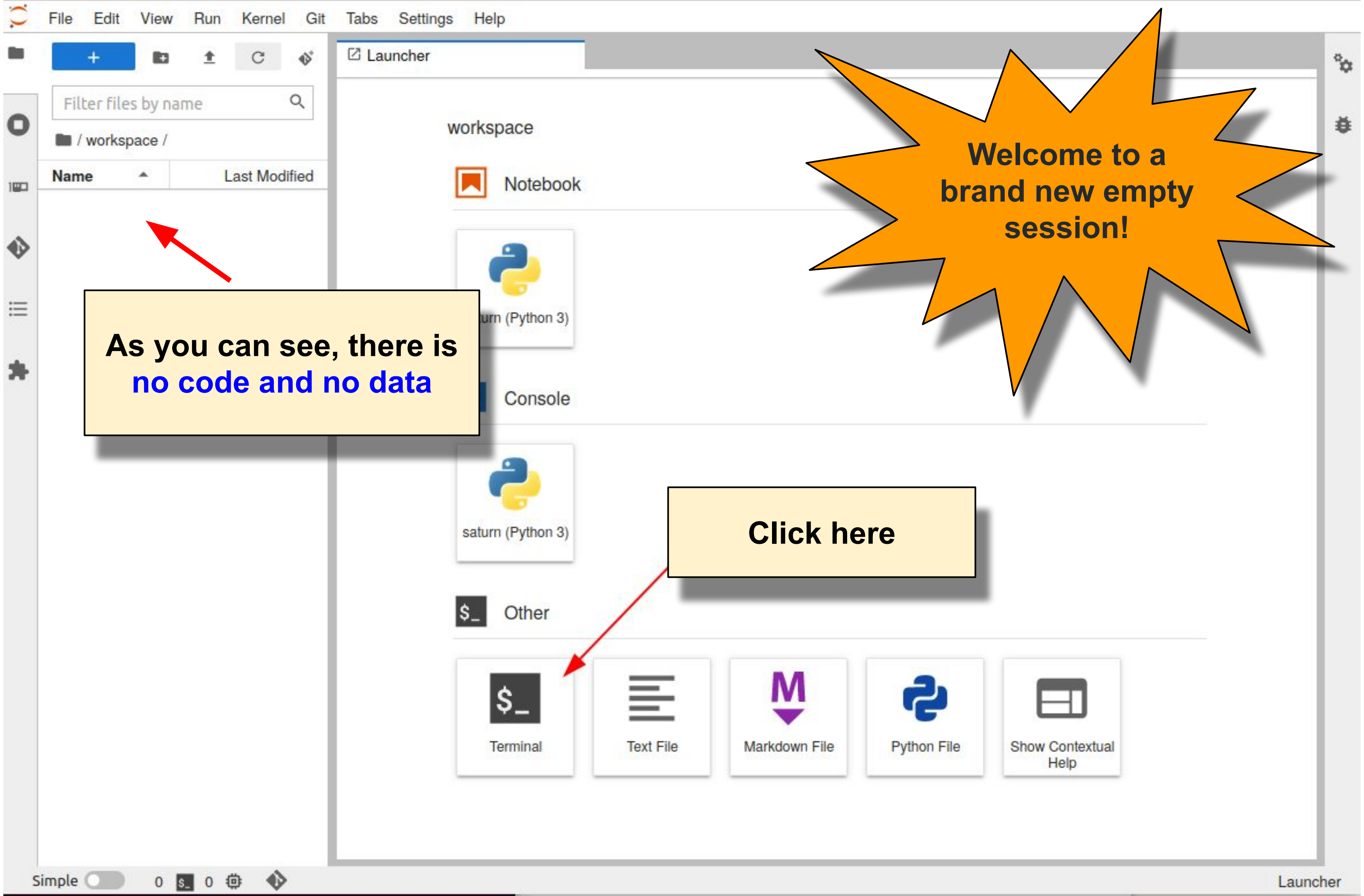


Image
saturncloud/saturn-pytorch
Version
2022.03.01
Spot Instance
This requests the server of specified size as Spot Instance. Spot Instances are less expensive, but may be shut down at any time (with a two-minute warning). (?)
Working Directory
/home/jovyan/workspace
Write these for apt install
Extra Packages
Extra packages are installed every time the resource starts up - right before the start script. Use spaces to separate packages.
If you find yourself adding the same packages to lots of resources, you may want to permanently add packages to a custom image instead. (?)
Conda Install
Pip Install
Apt Packages
htop zip unzip python3-opencv
The packages together will run the following script:
apt-get install htop zip unzip python3-opencv
pip install opencv-python
Shutoff After
1 hour
Disabled options are not supported in th
And click Create
Advanced Settings (option
Create
Cancel


Saturn Cloud
pierremarcjodoin
Resources
Credentials
Git Repositories
Images
Enterprise
Get Started Next: Spin up Dask
HOURS REMAINING
Upgrade for More
Jupyter 30 hrs
Dask 3 hrs
Resources / pierremarcjodoin / AutoEncoderHandsOn
JUPYTER SERVER
pierremarcjodoin / AutoEncoderHan...
04fd2176712f4c02a0e70719577c284b
Recipe
Logs
Edit
Delete
Overview
Environment
Git Repos
Share
Jupyter Server
stopped
Start
T4-XLarge - 4 cores - 16 GB RAM - 1 GPU - 10Gi Disk
Metrics
Auto Shutoff: 1 hour
Spot Instance: No
SSH URL: (not enabled) (?)
Jupyter Lab
Click Start


Saturn Cloud
pierremarcjodoin
Resources
Credentials
Git Repositories
Images
Enterprise
Get Started Next: Spin up Dask
HOURS REMAINING
Upgrade for More
Jupyter 30 hrs
Dask 3 hrs
Resources / pierremarcjodoin / AutoEncoderHandsOn2
JUPYTER SERVER
pierremarcjodoin / AutoEncoderHan...
1fc514f792564bc383f7cbda6237181f
Recipe
Logs
Edit
Delete
Overview
Jupyter Server
Start
Stop
Metrics
Auto Shutoff: 1 hour
Spot Instance: No
SSH URL: (not enabled) (?)
Scheduled
Provisioned Hardware
Pulling Image
Set Up Environment
Execute Start Script
Ready
Jupyter Lab
You will see a progression bar.
Please wait a minute or so


Saturn Cloud
pierremarcjodoin
Resources
Credentials
Git Repositories
Images
Enterprise
Get Started Next: Spin up Dask
HOURS REMAINING
Upgrade for More
Jupyter 30 hrs
Dask 3 hrs
Resources / pierremarcjodoin / AutoEncoderHandsOn2
JUPYTER SERVER
pierremarcjodoin / AutoEncoderHan...
1fc514f792564bc383f7cbda6237181f
Recipe
Logs
Edit
Delete
Overview
Environment
Git Repos
Share
Jupyter Server
running
Stop
T4-XLarge - 4 cores - 16 GB RAM - 1 GPU - 10Gi Disk
Metrics
Auto Shutoff: 1 hour
Spot Instance: No
SSH URL: (not enabled) (?)
Jupyter Lab
Click here


File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ workspace /
Name
Last Modified
Launcher
workspace
Notebook
urn (Python 3)
Console
saturn (Python 3)
Other
Terminal
Text File
Markdown File
Python File
Show Contextual Help
Simple
Launcher
Welcome to a brand new empty session!
As you can see, there is no code and no data
Click here

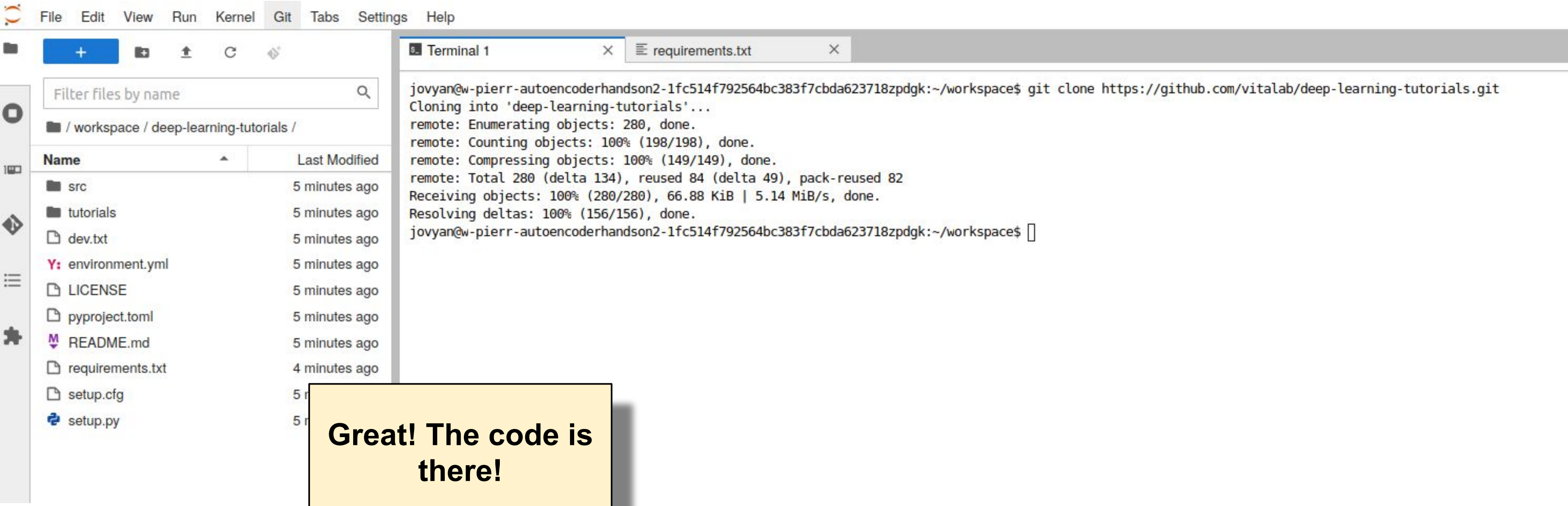
Lets download the code

In the terminal, type

```
git clone https://github.com/vitalab/deep-learning-tutorials.git
```

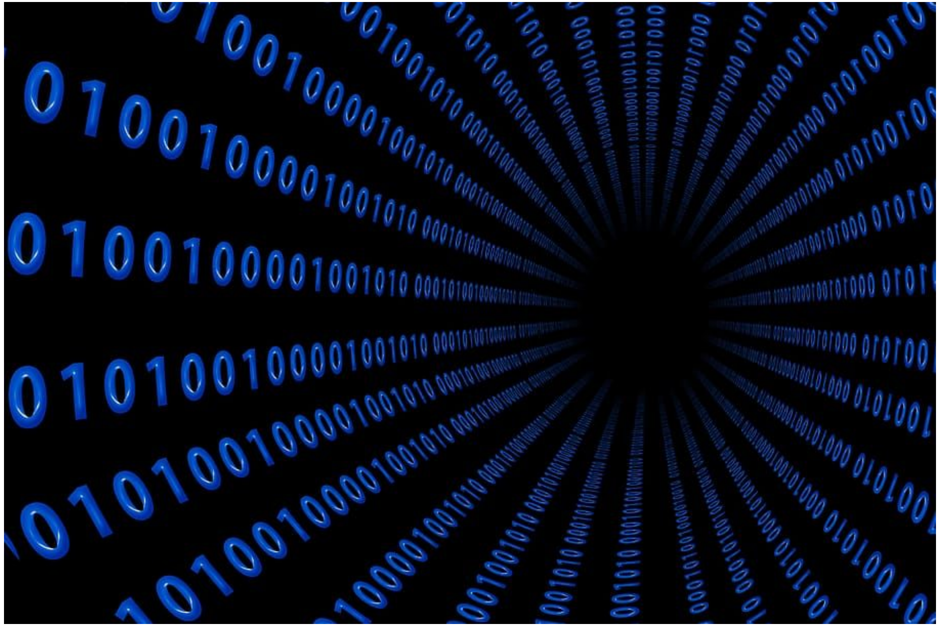
This command allows to **download the code** from a public **gitHub repository**

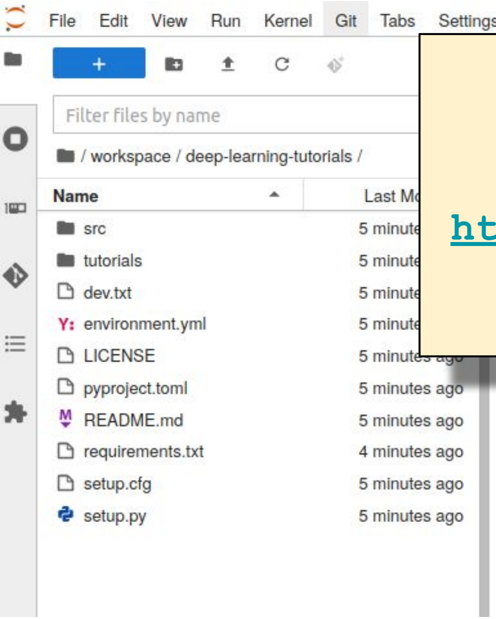
Click here



Great! The code is there!

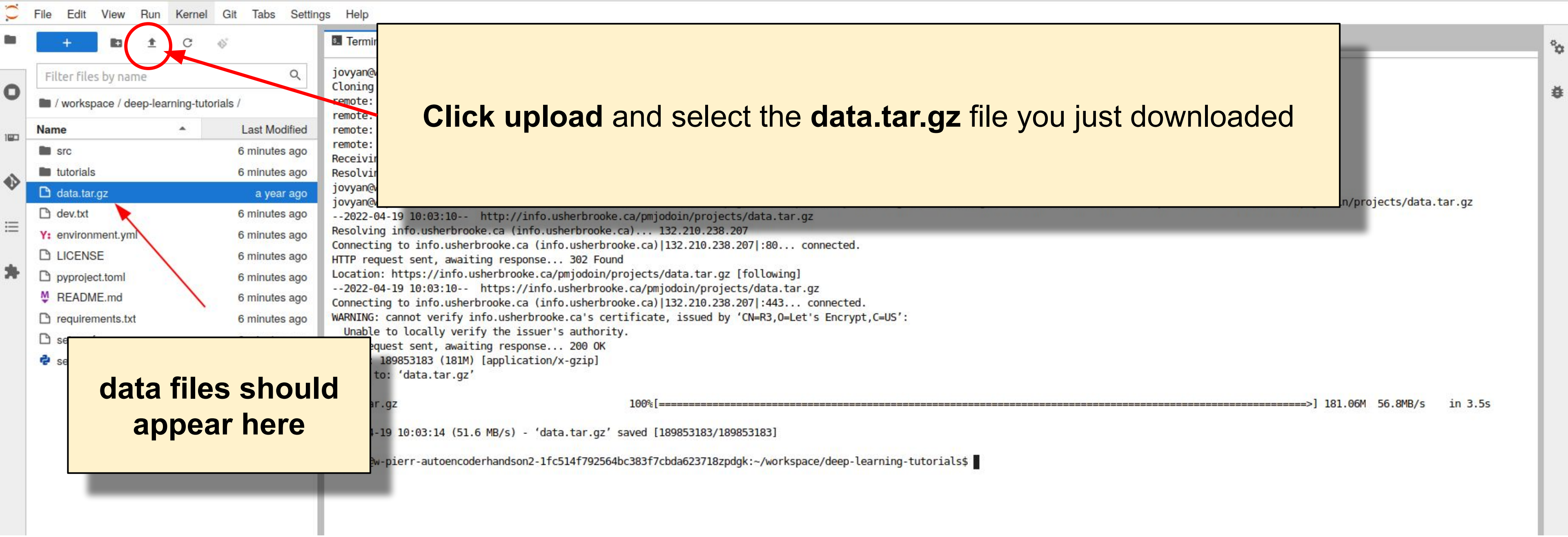
Lets download the data

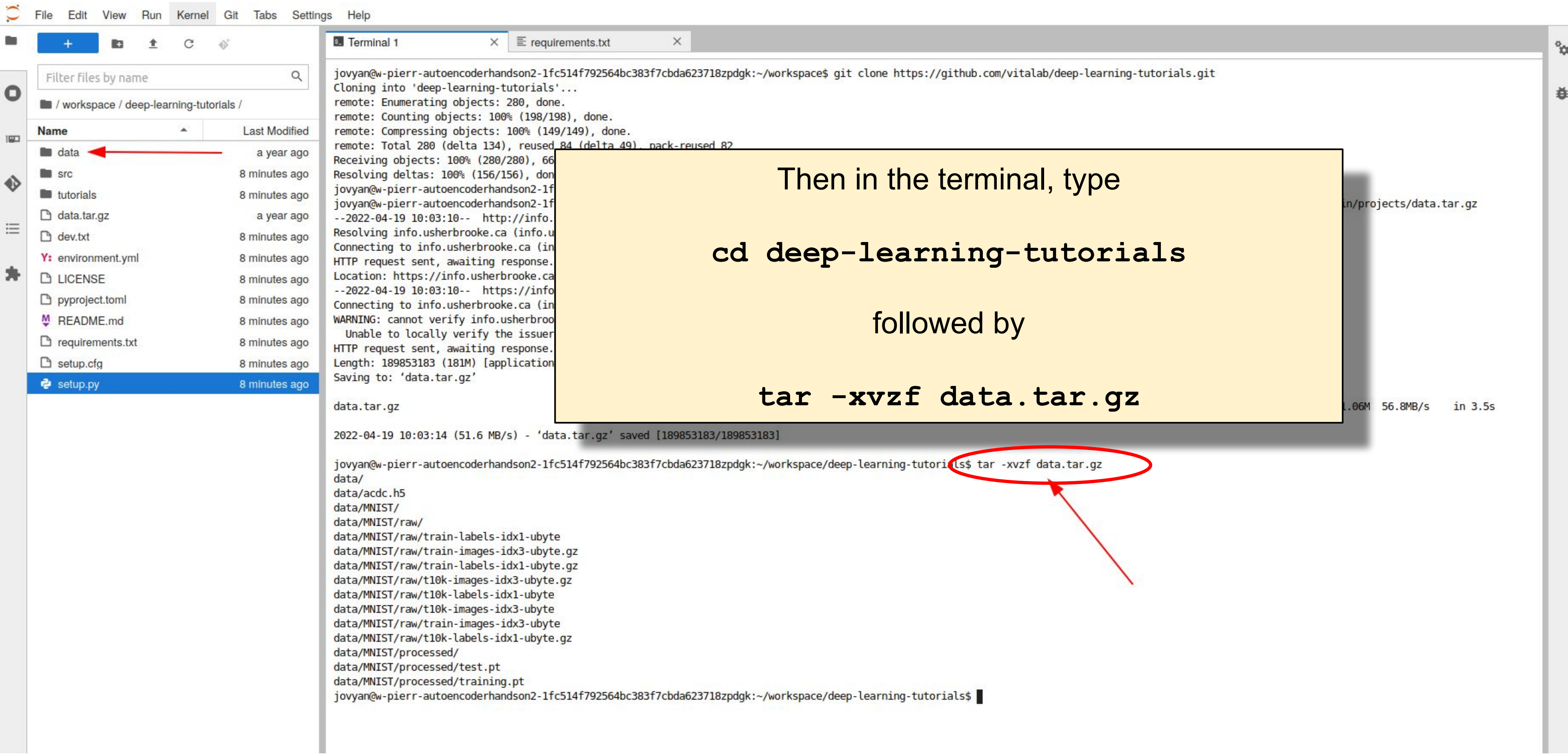


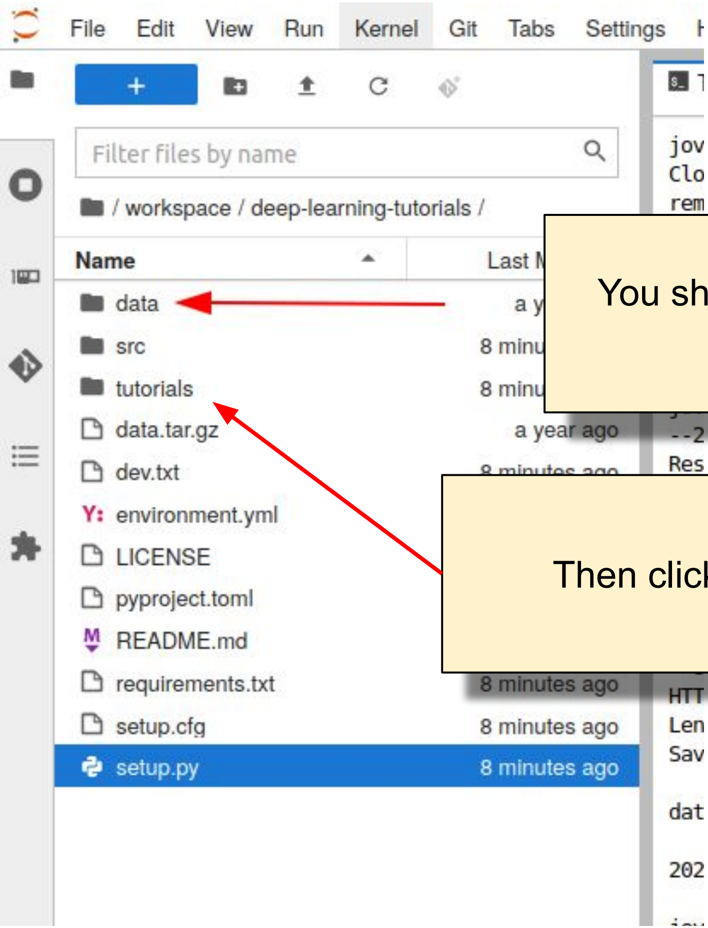


Download the following file on your computer

http://info.usherbrooke.ca/pmjodoin/projects/data.tar.gz

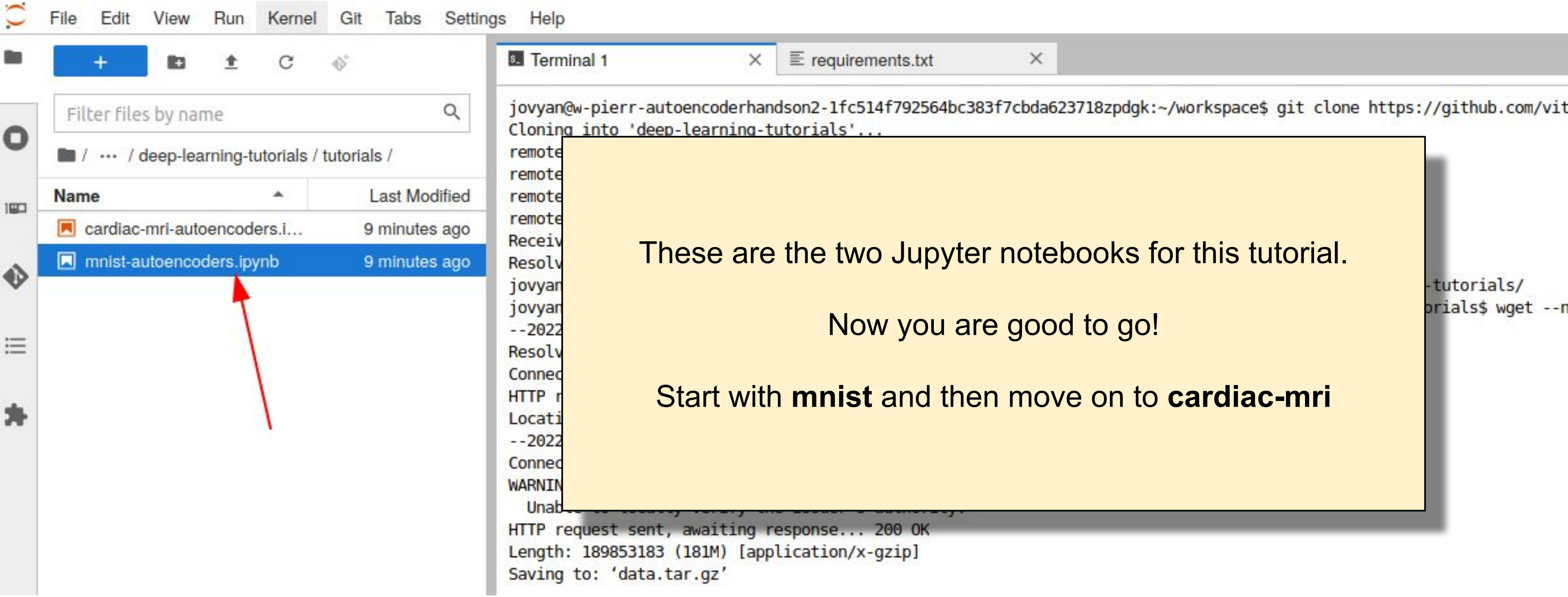

Click upload and select the data.tar.gz file you just downloaded
data files should appear here
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ workspace / deep-learning-tutorials /
Name
Last Modified
src 6 minutes ago
tutorials 6 minutes ago
data.tar.gz a year ago
dev.txt 6 minutes ago
environment.yml 6 minutes ago
LICENSE 6 minutes ago
pyproject.toml 6 minutes ago
README.md 6 minutes ago
requirements.txt 6 minutes ago
--2022-04-19 10:03:10-- http://info.usherbrooke.ca/pmjodoin/projects/data.tar.gz
Resolving info.usherbrooke.ca (info.usherbrooke.ca)... 132.210.238.207
Connecting to info.usherbrooke.ca (info.usherbrooke.ca)|132.210.238.207|:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://info.usherbrooke.ca/pmjodoin/projects/data.tar.gz [following]
--2022-04-19 10:03:10-- https://info.usherbrooke.ca/pmjodoin/projects/data.tar.gz
Connecting to info.usherbrooke.ca (info.usherbrooke.ca)|132.210.238.207|:443... connected.
WARNING: cannot verify info.usherbrooke.ca's certificate, issued by 'CN=R3,O=Let's Encrypt,C=US':
Unable to locally verify the issuer's authority.


Then in the terminal, type
cd deep-learning-tutorials
followed by
tar -xvzf data.tar.gz
File Edit View Run Kernel Git Tabs Settings Help
Terminal 1
requirements.txt
Filter files by name
/ workspace / deep-learning-tutorials /
Name
Last Modified
data a year ago
src 8 minutes ago
tutorials 8 minutes ago
data.tar.gz a year ago
dev.txt 8 minutes ago
environment.yml 8 minutes ago
LICENSE 8 minutes ago
pyproject.toml 8 minutes ago
README.md 8 minutes ago
requirements.txt 8 minutes ago
setup.cfg 8 minutes ago
setup.py 8 minutes ago
jovyan@w-pierr-autoencoderhandson2-1fc514f792564bc383f7cbda623718zpdgk:~/workspace$ git clone https://github.com/vitalab/deep-learning-tutorials.git
Cloning into 'deep-learning-tutorials'...
remote: Enumerating objects: 280, done.
remote: Counting objects: 100% (198/198), done.
remote: Compressing objects: 100% (149/149), done.
2022-04-19 10:03:14 (51.6 MB/s) - ‘data.tar.gz’ saved [189853183/189853183]
jovyan@w-pierr-autoencoderhandson2-1fc514f792564bc383f7cbda623718zpdgk:~/workspace/deep-learning-tutorials$ tar -xvzf data.tar.gz
data/
data/acdc.h5
data/MNIST/
data/MNIST/raw/
data/MNIST/raw/train-labels-idx1-ubyte
data/MNIST/raw/train-images-idx3-ubyte.gz
data/MNIST/raw/train-labels-idx1-ubyte.gz
data/MNIST/raw/t10k-images-idx3-ubyte.gz
data/MNIST/raw/t10k-labels-idx1-ubyte
data/MNIST/raw/t10k-images-idx3-ubyte
data/MNIST/raw/train-images-idx3-ubyte
data/MNIST/raw/t10k-labels-idx1-ubyte.gz
data/MNIST/processed/
data/MNIST/processed/test.pt
data/MNIST/processed/training.pt
jovyan@w-pierr-autoencoderhandson2-1fc514f792564bc383f7cbda623718zpdgk:~/workspace/deep-learning-tutorials$

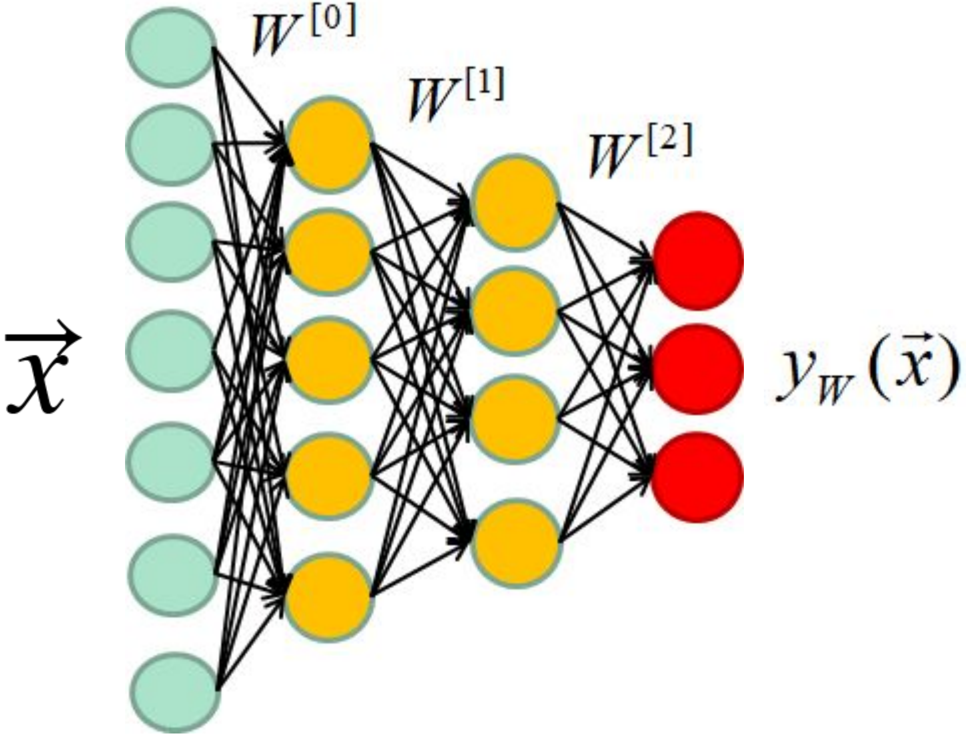


You should have the following **data folder**

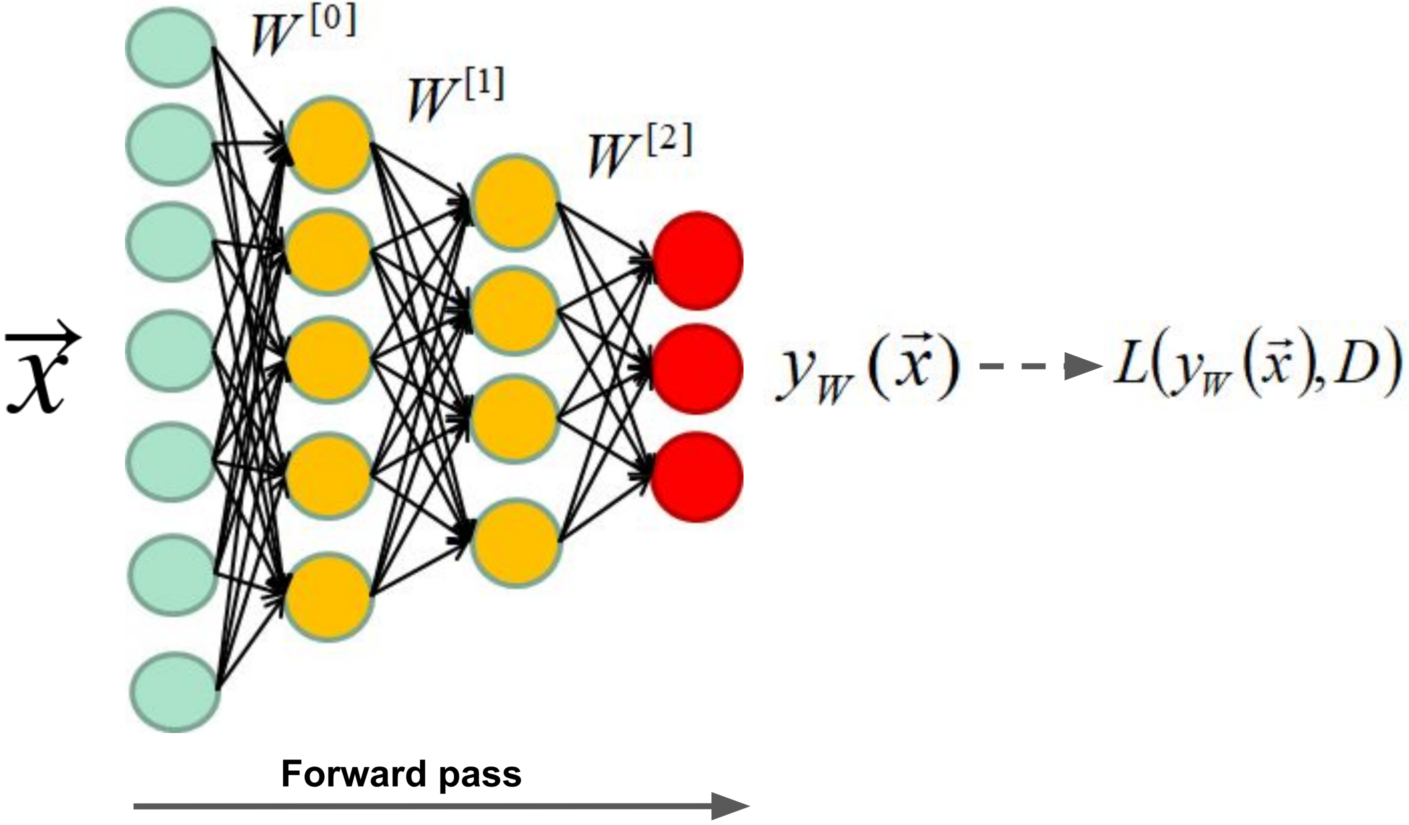
Then click on **tutorials**


File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ ··· / deep-learning-tutorials / tutorials /
Name
Last Modified
cardiac-mri-autoencoders.i...
9 minutes ago
mnist-autoencoders.ipynb
9 minutes ago
Terminal 1
requirements.txt
jovyan@w-pierr-autoencoderhandson2-1fc514f792564bc383f7cbda623718zpdgk:~/workspace$ git clone https://github.com/vit
Cloning into 'deep-learning-tutorials'...
HTTP request sent, awaiting response... 200 OK
Length: 189853183 (181M) [application/x-gzip]
Saving to: 'data.tar.gz'
These are the two Jupyter notebooks for this tutorial.
Now you are good to go!
Start with **mnist** and then move on to **cardiac-mri**

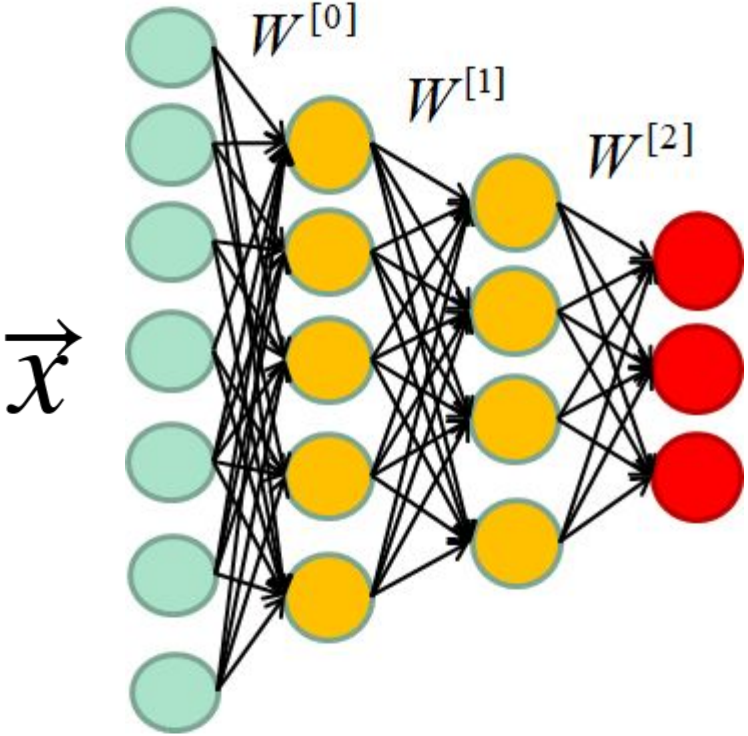
Autoencoders
(recap)


$W^{[0]}$
$W^{[1]}$
$W^{[2]}$
$\vec{x}$
$y_W(\vec{x})$

Annotated dataset: $D = \{(\vec{x}_1, t_1), (\vec{x}_2, t_2), \ldots, (\vec{x}_N, t_N)\}$



Annotated dataset: $D = \{(\vec{x}_1, t_1), (\vec{x}_2, t_2), \ldots, (\vec{x}_N, t_N)\}$

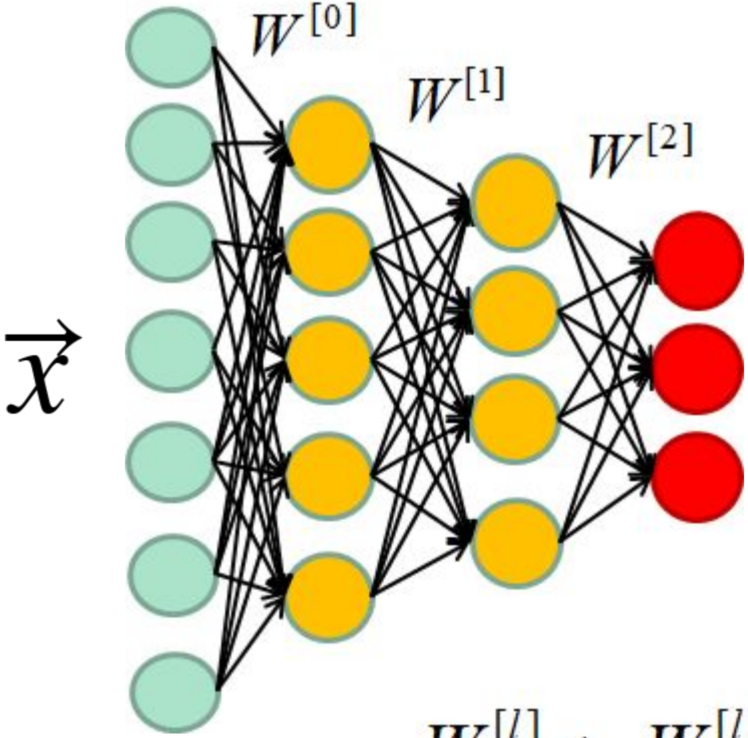


$y_W(\vec{x}) \dashrightarrow L(y_W(\vec{x}), D)$

$$\frac{\partial (L(y_W(\vec{x})))}{\partial W_j^{[l]}}$$

Backward pass

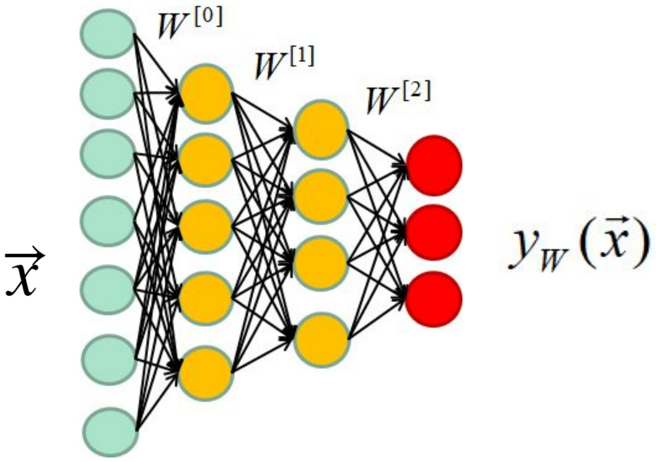
Annotated dataset: $D = \{(\vec{x}_1, t_1), (\vec{x}_2, t_2), \ldots, (\vec{x}_N, t_N)\}$

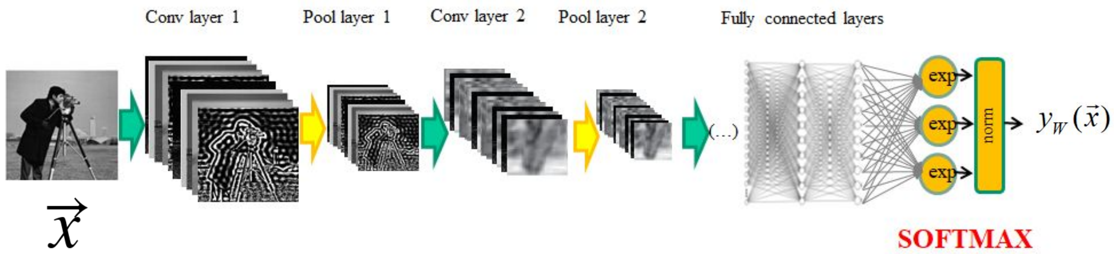


$y_W(\vec{x}) \dashrightarrow L(y_W(\vec{x}), D)$

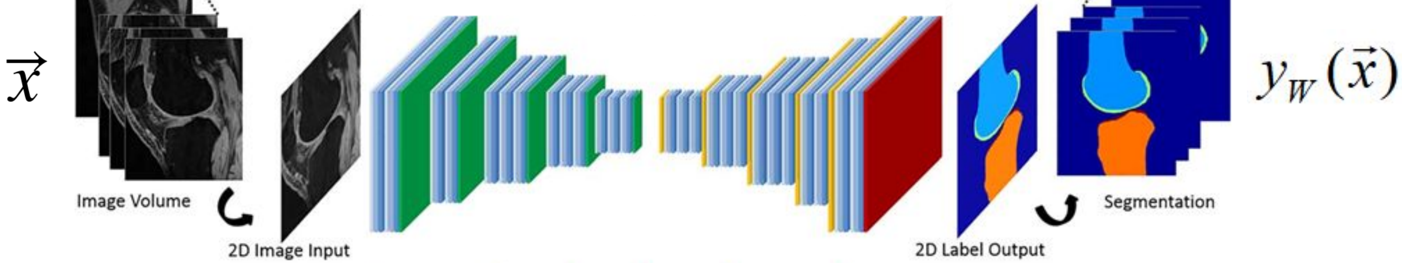
Gradient descent

$$W_j^{[l]} \leftarrow W_j^{[l]} - \eta \frac{\partial(L(y_W(\vec{x})))}{\partial W_j^{[l]}}$$





Segmentation CNN

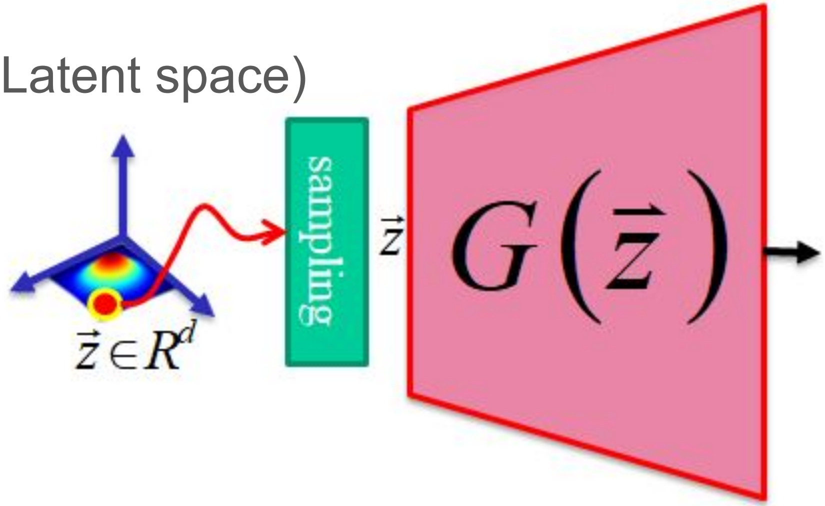


Fang Liu, etal., Deep convolutional neural network and 3D deformable approach for tissue segmentation in musculoskeletal magnetic resonance imaging. in Magnetic resonance in medicine 2018 DOI:10.1002/mrm.26841

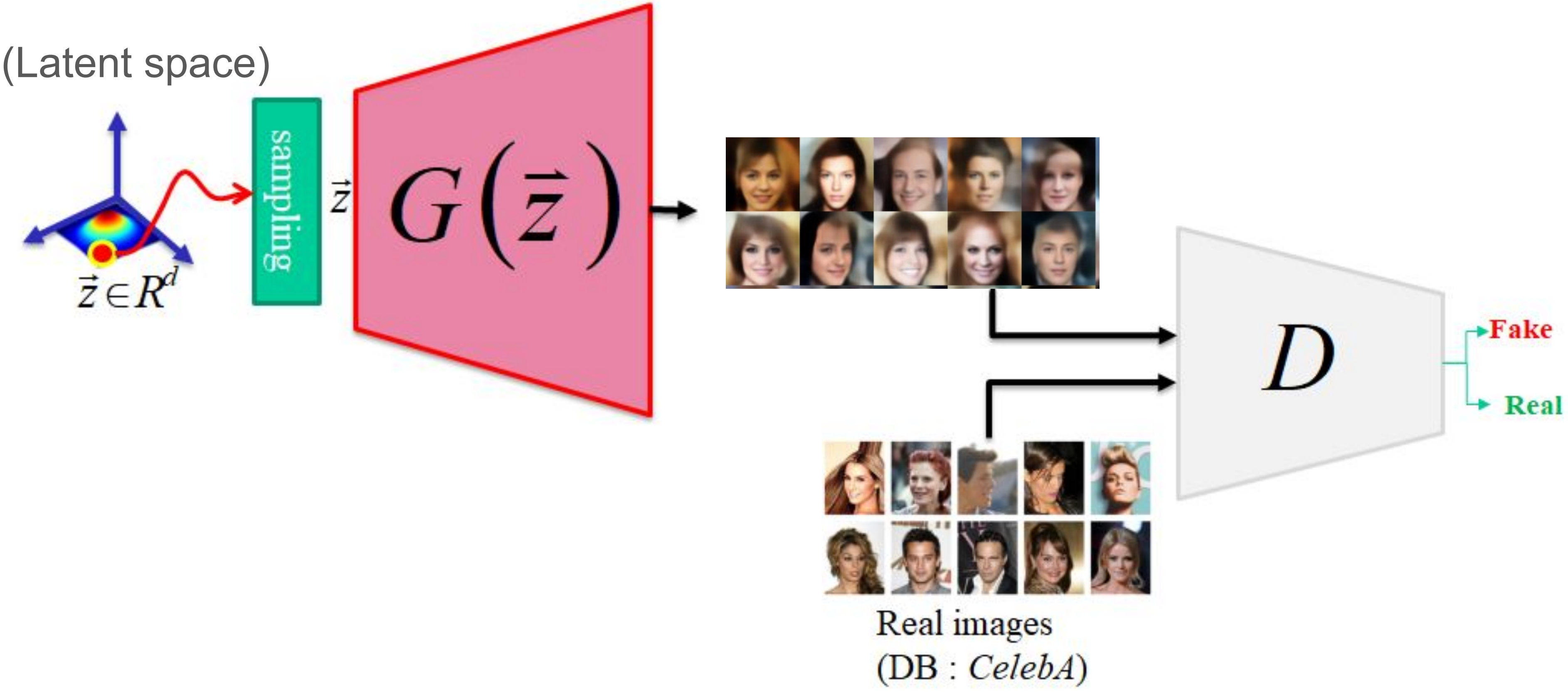
GAN

Loss?
Annotated dataset?

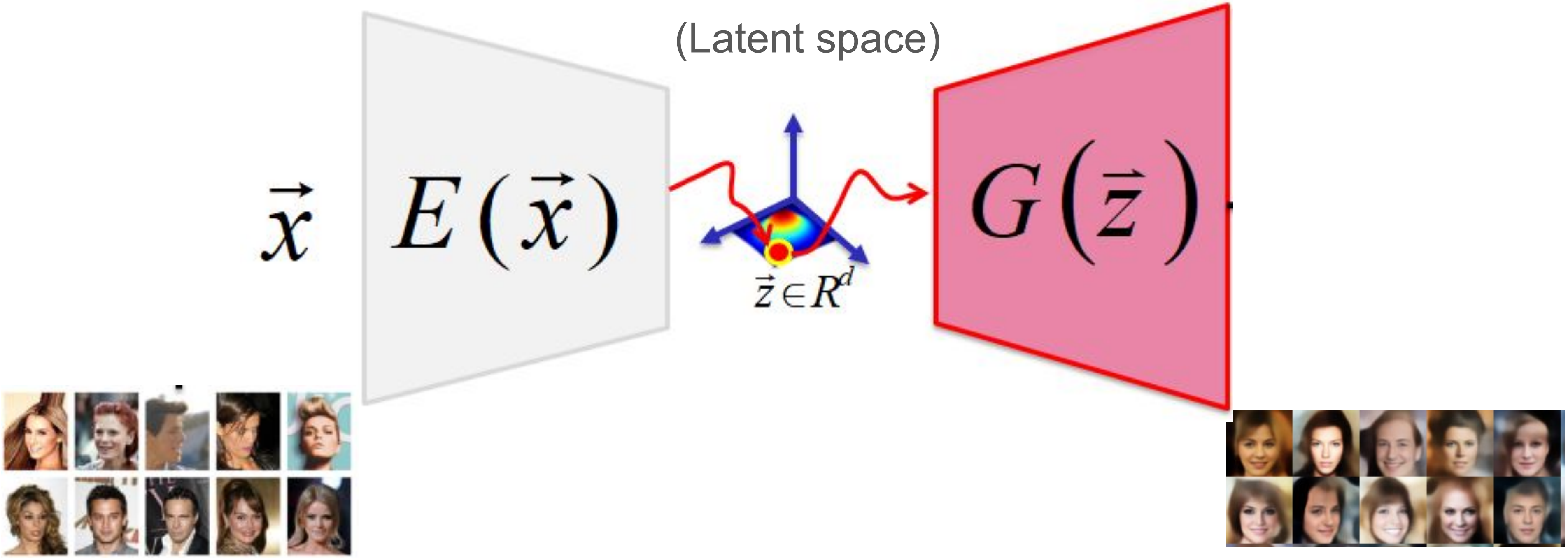
(Latent space)



GAN

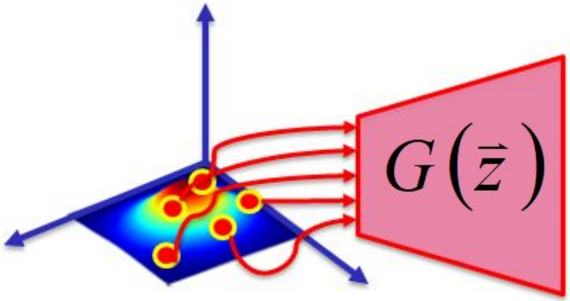

(Latent space)
$\vec{z} \in R^d$
sampling
$\vec{z}$
$G(\vec{z})$
D
Fake
Real
Real images
(DB : *CelebA*)

Autoencoders



Autoencoders (once training is over)

(Latent space)



Summary

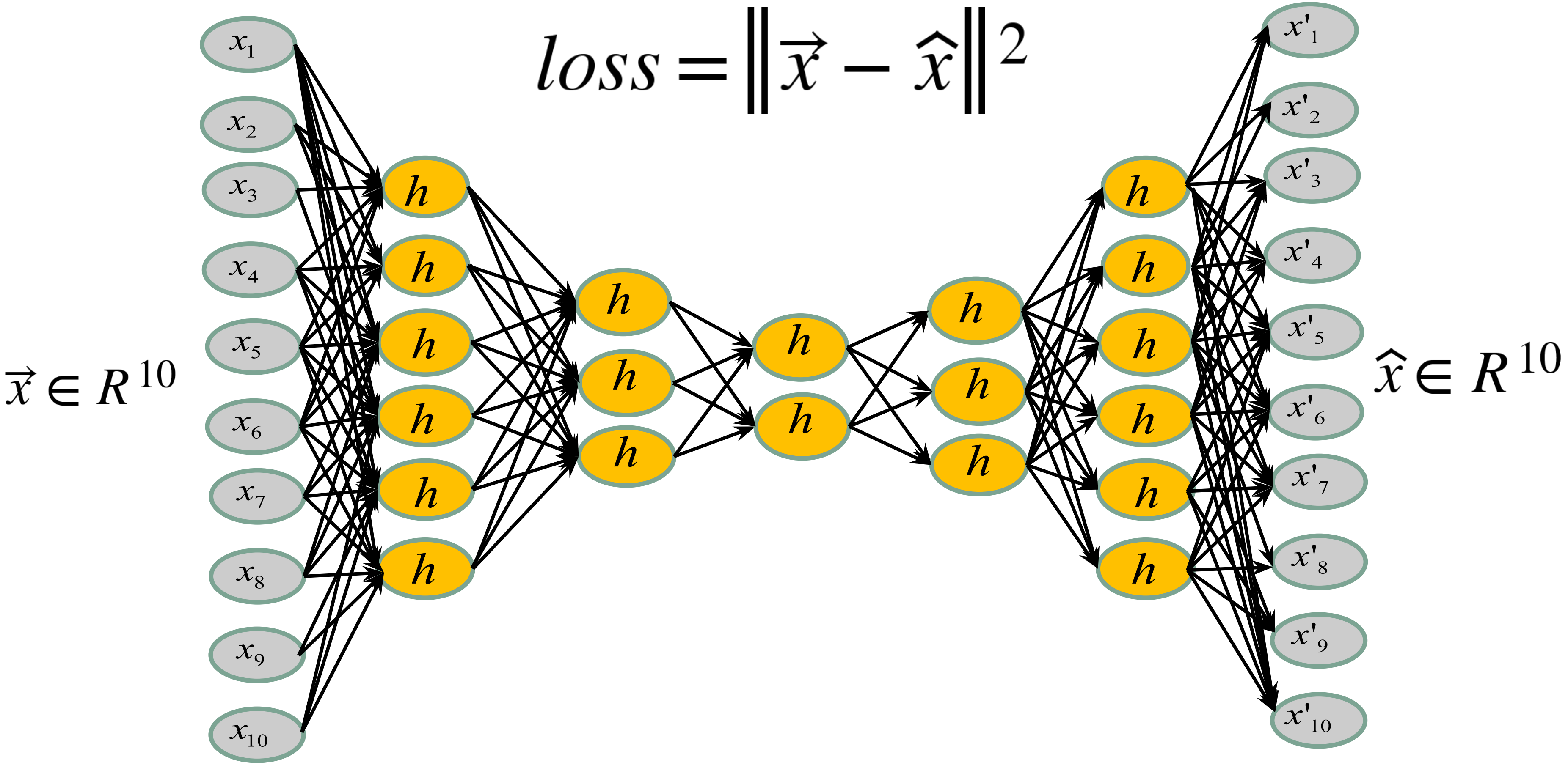
1. What are autoencoders and variational autoencoders?
2. How do they apply to MNIST (grayscale images)?
3. How do they apply to segmentation maps (ACDC cardiac labels)?

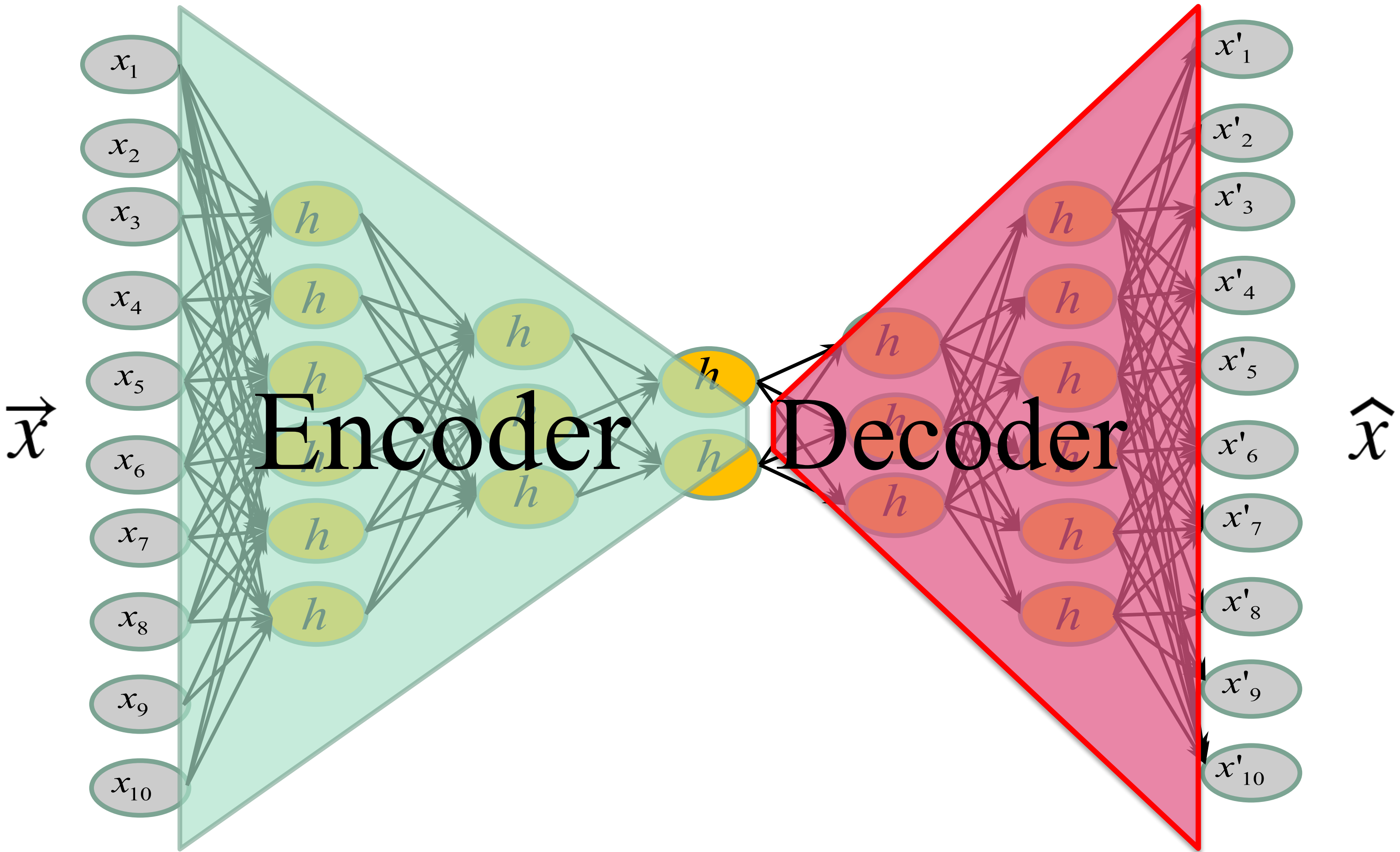
Goal : learn the latent representation of a set of data

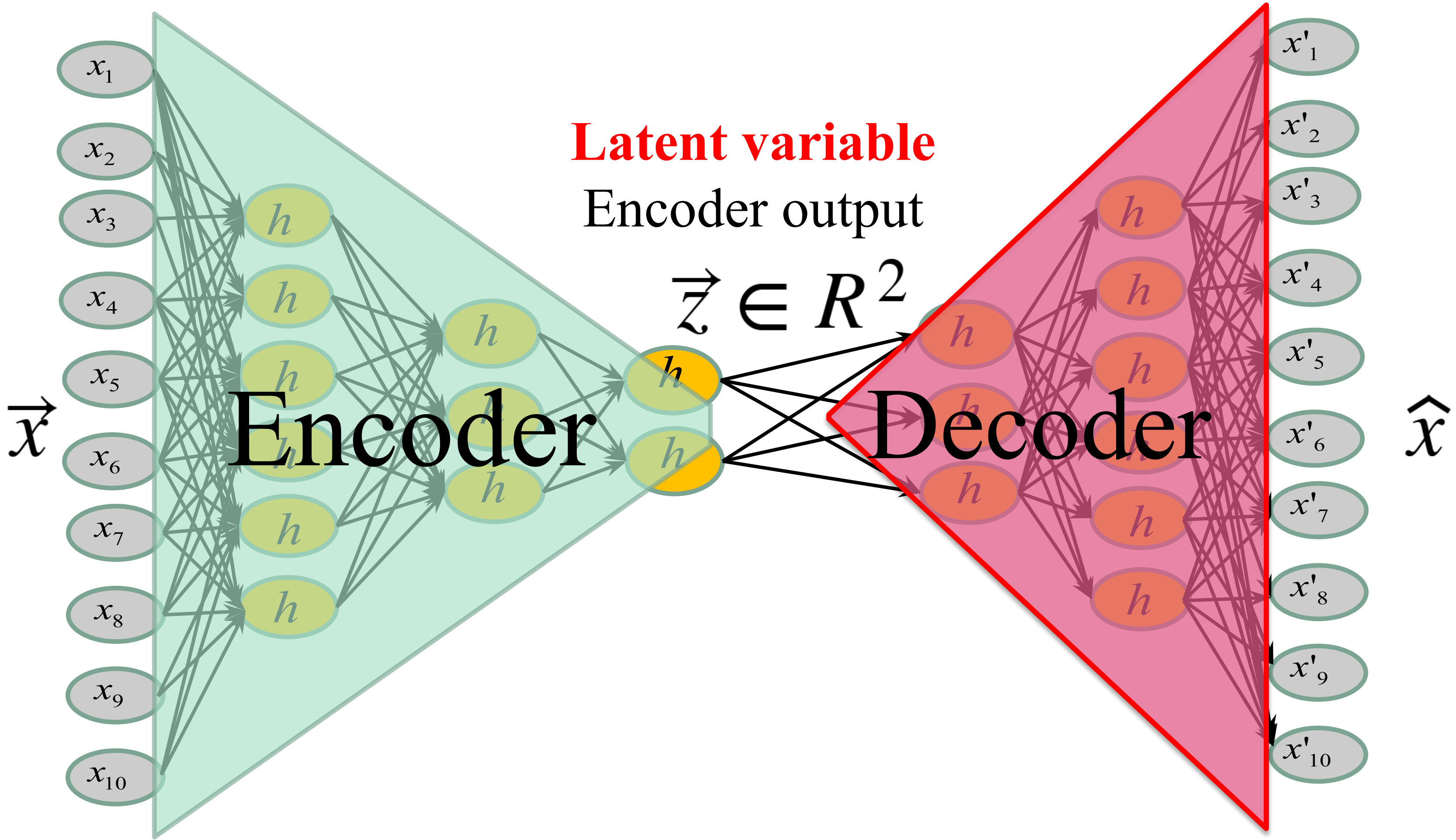
How : by training a Neural Net to output its own … input!

Note : if you are familiar with AE and VAE, you may skip the next couple of slides.

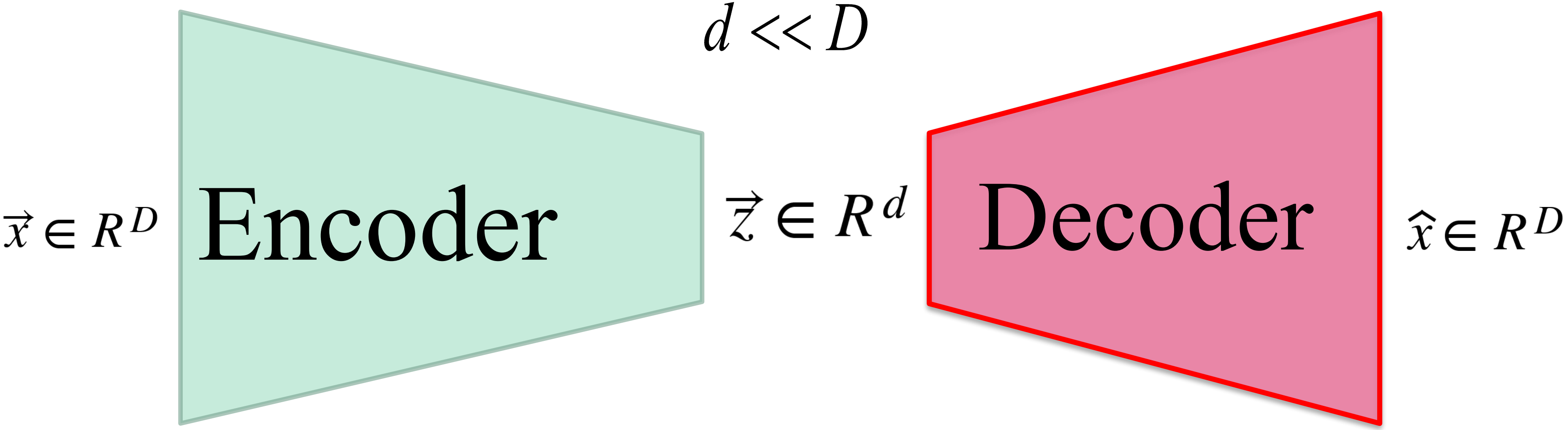
$$loss = \left\| \vec{x} - \hat{x} \right\|^2$$




$\vec{x}$
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_{10}
Encoder
Decoder
x'_1
x'_2
x'_3
x'_4
x'_5
x'_6
x'_7
x'_8
x'_9
x'_{10}
$\hat{x}$

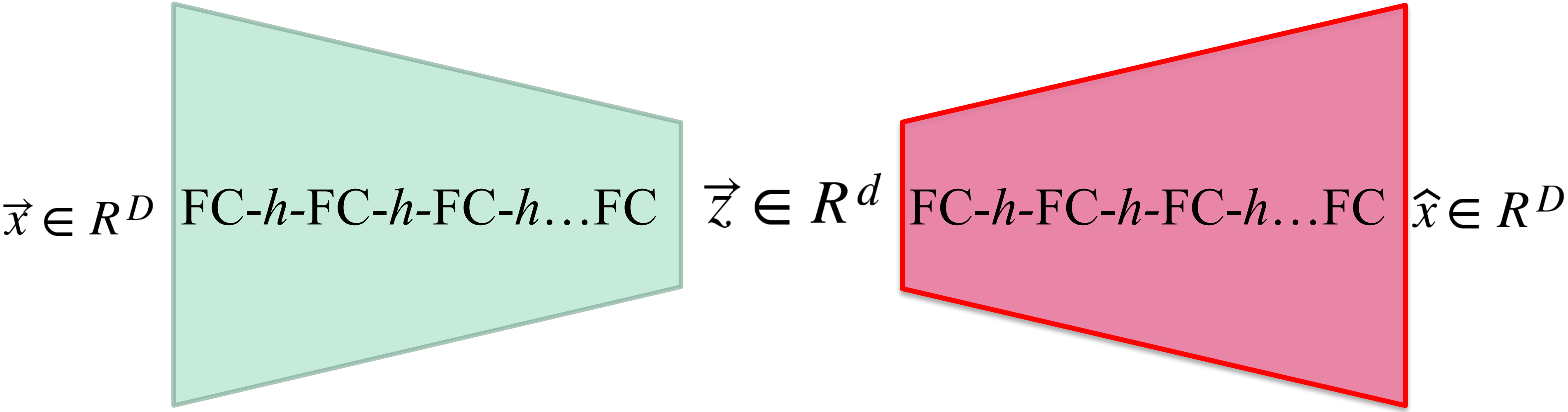


Latent variable
Encoder output
$\vec{z} \in R^2$
$\vec{x}$
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_{10}
Encoder
Decoder
x'_1
x'_2
x'_3
x'_4
x'_5
x'_6
x'_7
x'_8
x'_9
x'_{10}
$\hat{x}$

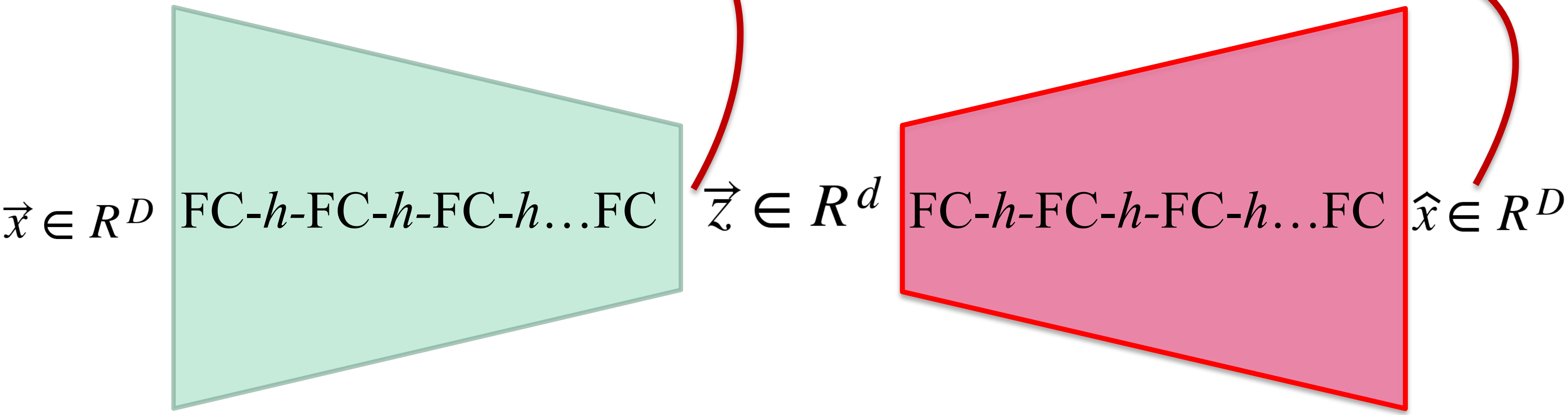

$d << D$
$\vec{x} \in R^D$
Encoder
$\vec{z} \in R^d$
Decoder
$\hat{x} \in R^D$

Fully-Connected Layers

h : activation function

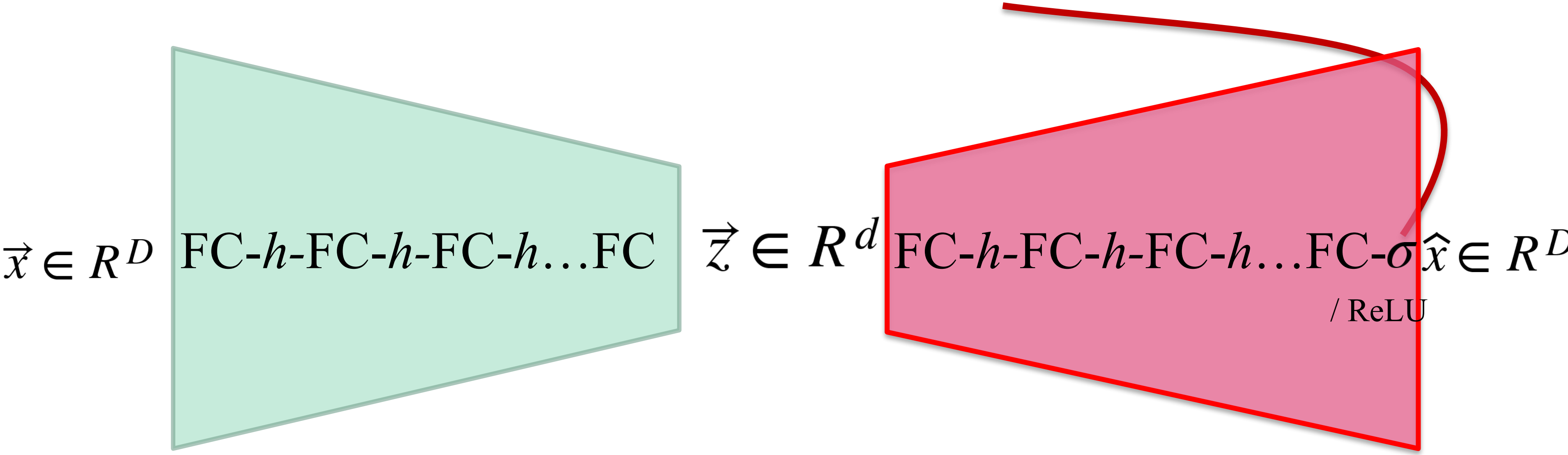


Very often...
No activation function at the output
of the encoder and the decoder



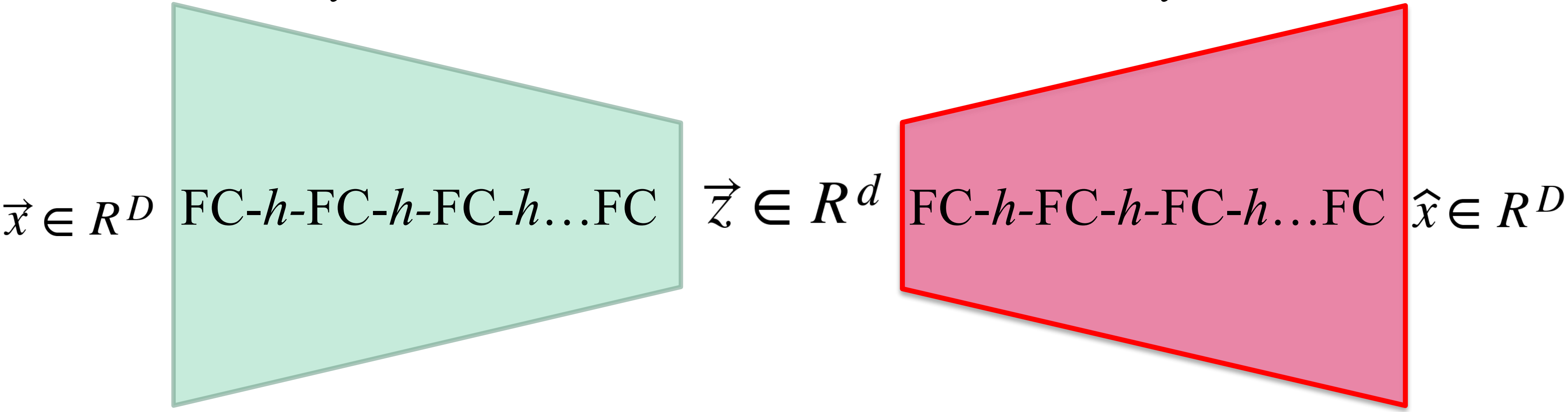
See **why?**

Sometimes **sigmoid** to predict pixel values between 0 and 1 or **ReLU** when the pixel values can be large but never negative.



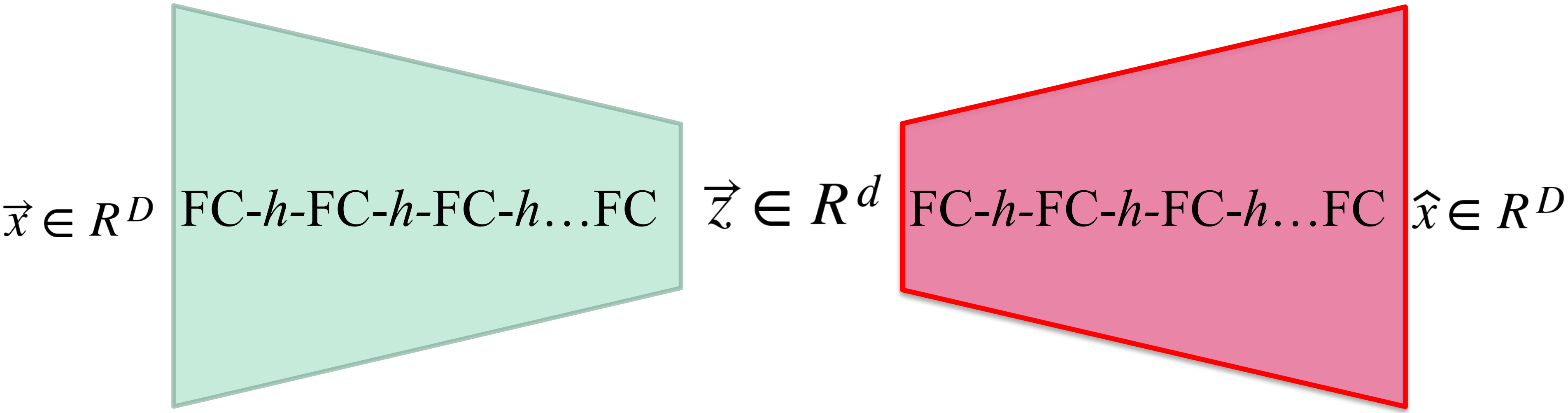
The number of neurons **Decrease (or stay the same)** from a layer to another

The number of neurons **Increase (or stay the same)** from a layer to another

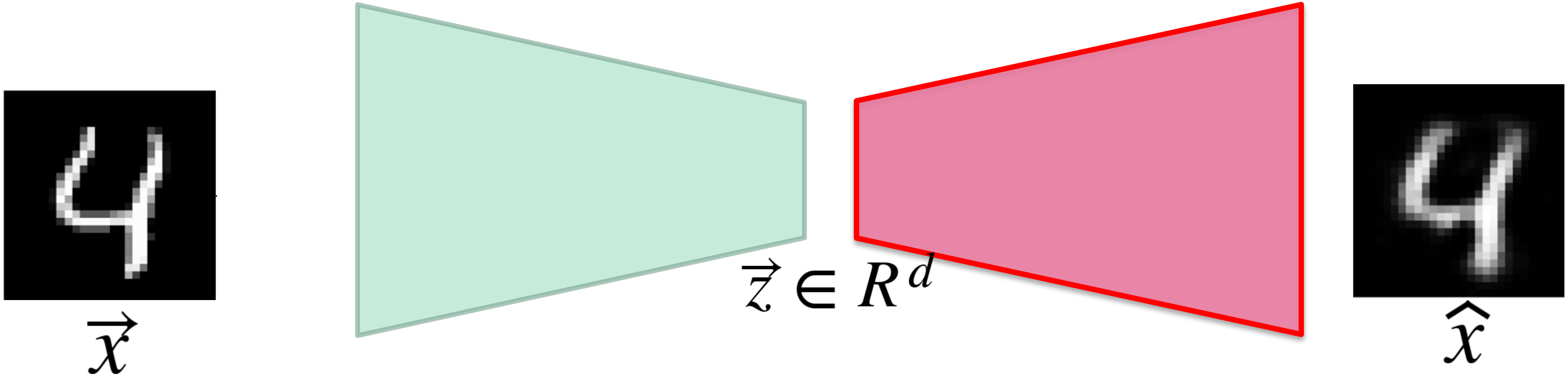


Very often...

The structure of the encoder is the **dual** of that of the decoder



Basic image-based autoencoder



Simple MNIST Autoencoder

```
class autoencoder(nn.Module):
    def __init__(self):
        super(autoencoder, self).__init__()
        self.encoder = nn.Sequential(
            nn.Linear(28 * 28, 128), nn.ReLU(True),
            nn.Linear(128, 64), nn.ReLU(True),
            nn.Linear(64, 12), nn.ReLU(True),
            nn.Linear(12, 2))
        self.decoder = nn.Sequential(
            nn.Linear(2, 12), nn.ReLU(True),
            nn.Linear(12, 64), nn.ReLU(True),
            nn.Linear(64, 128), nn.ReLU(True),
            nn.Linear(128, 28 * 28))

    def forward(self, x):
        z = self.encoder(x)
        x_prime = self.decoder(z)
        return x_prime
```

Latent space 2D

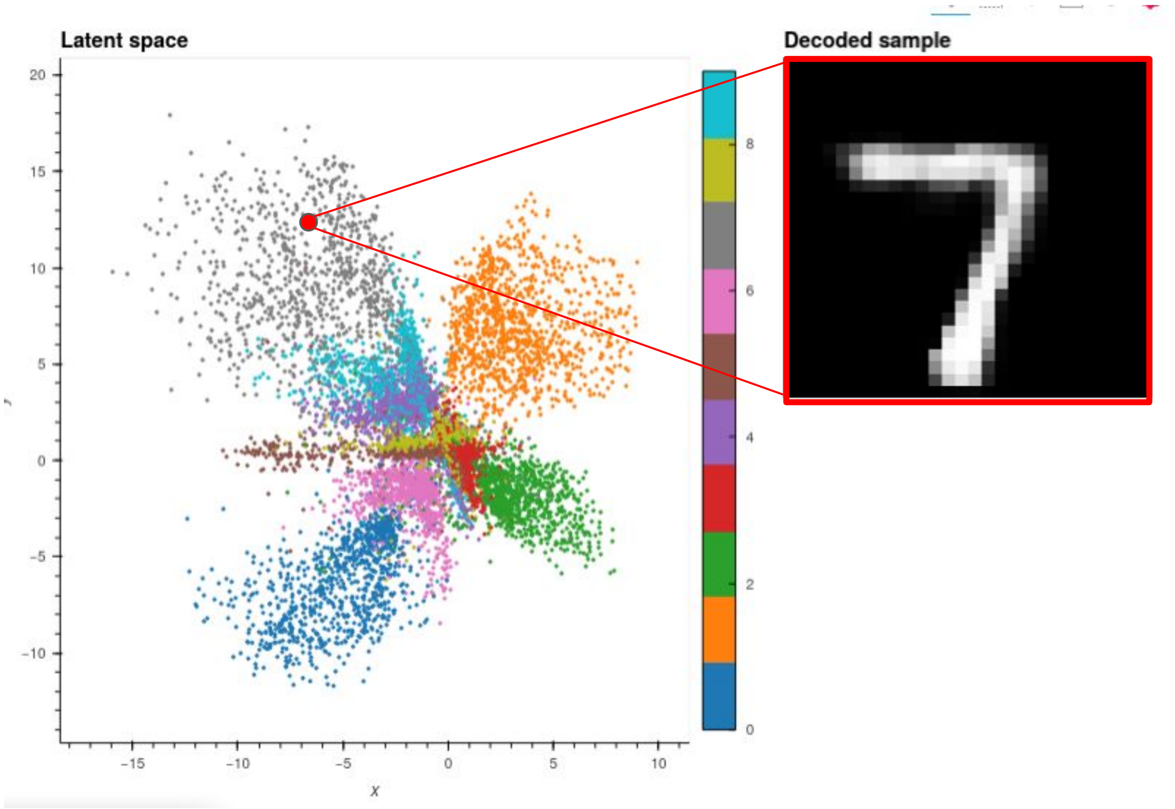
Simple MNIST Autoencoder

symmetry

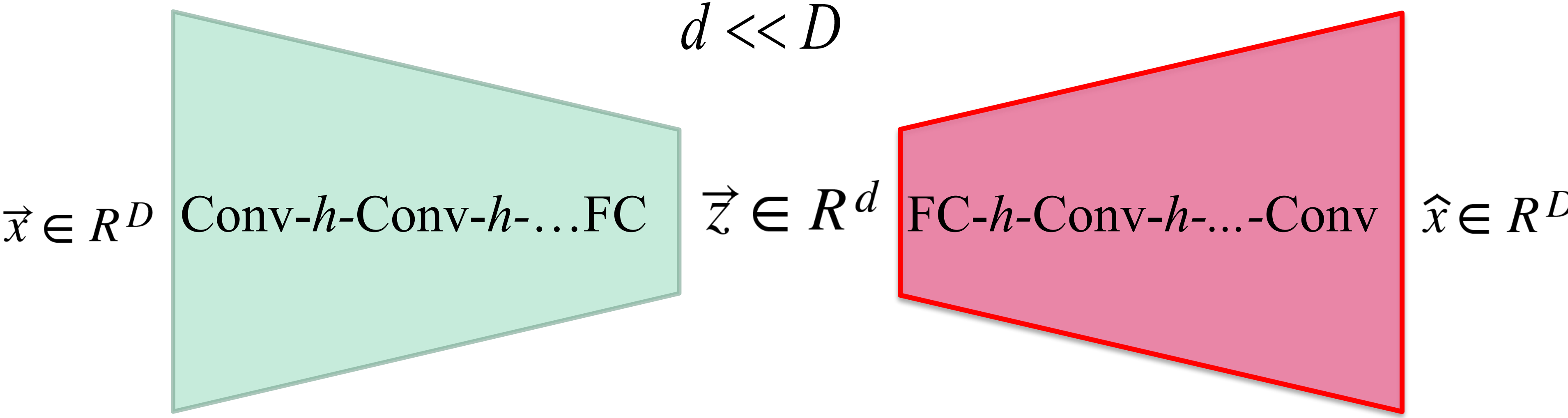
```python
class autoencoder(nn.Module):
    def __init__(self):
        super(autoencoder, self).__init__()
        self.encoder = nn.Sequential(
            nn.Linear(28 * 28, 128), nn.ReLU(True),
            nn.Linear(128, 64), nn.ReLU(True),
            nn.Linear(64, 12), nn.ReLU(True),
            nn.Linear(12, 2))
        self.decoder = nn.Sequential(
            nn.Linear(2, 12), nn.ReLU(True),
            nn.Linear(12, 64), nn.ReLU(True),
            nn.Linear(64, 128), nn.ReLU(True),
            nn.Linear(128, 28 * 28))

    def forward(self, x):
        z = self.encoder(x)
        x_prime = self.decoder(z)
        return x_prime
```

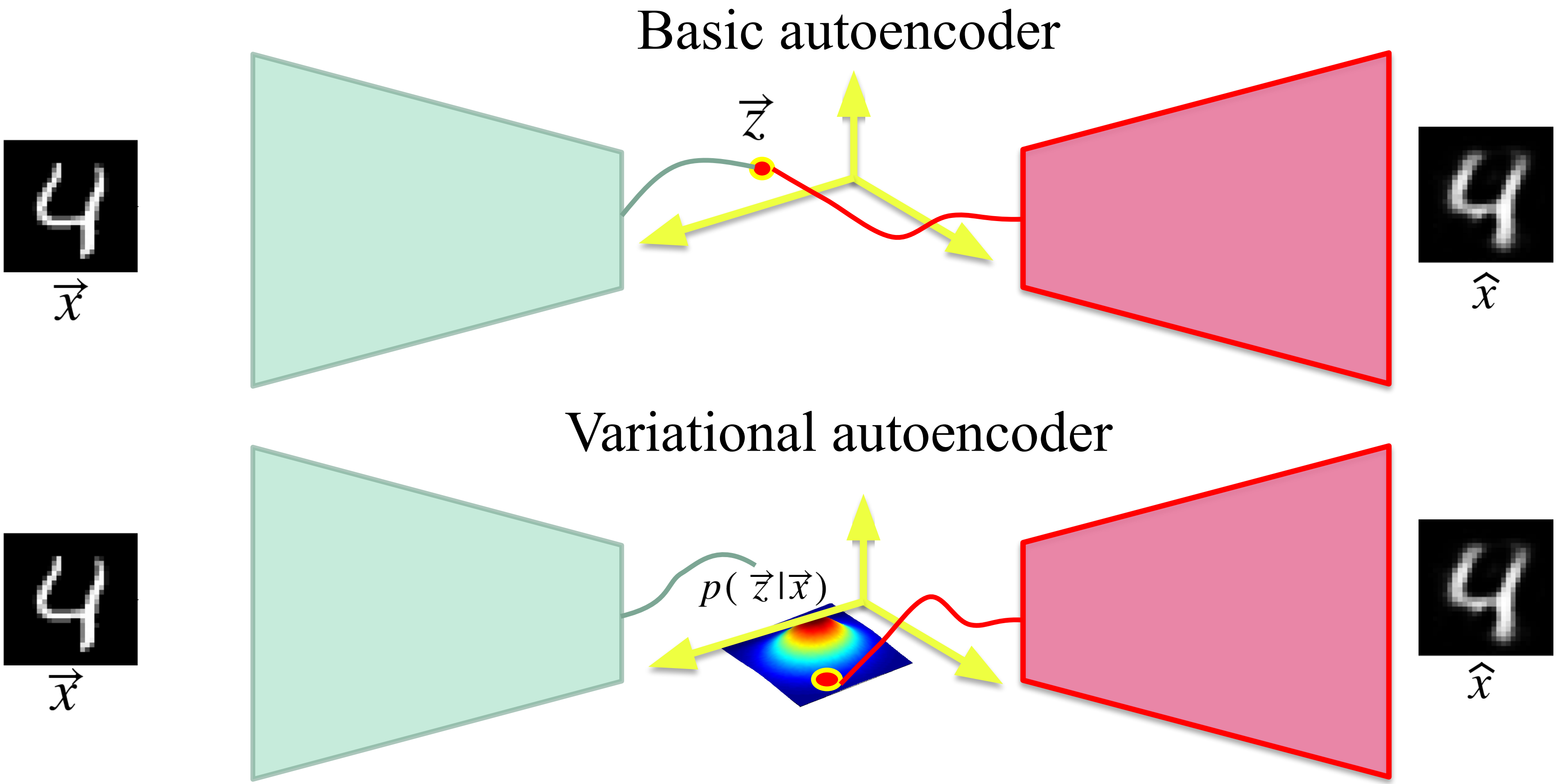
MNIST latent space (for 1000 images)

Each 2D point corresponds to an image



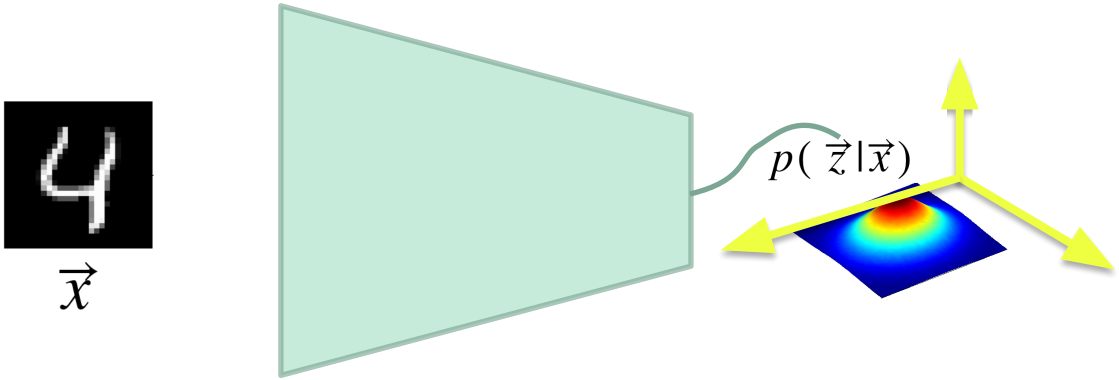
Conv layers


$d << D$
$\vec{x} \in R^D$
Conv-*h*-Conv-*h*-…FC
$\vec{z} \in R^d$
FC-*h*-Conv-*h*-...-Conv
$\hat{x} \in R^D$



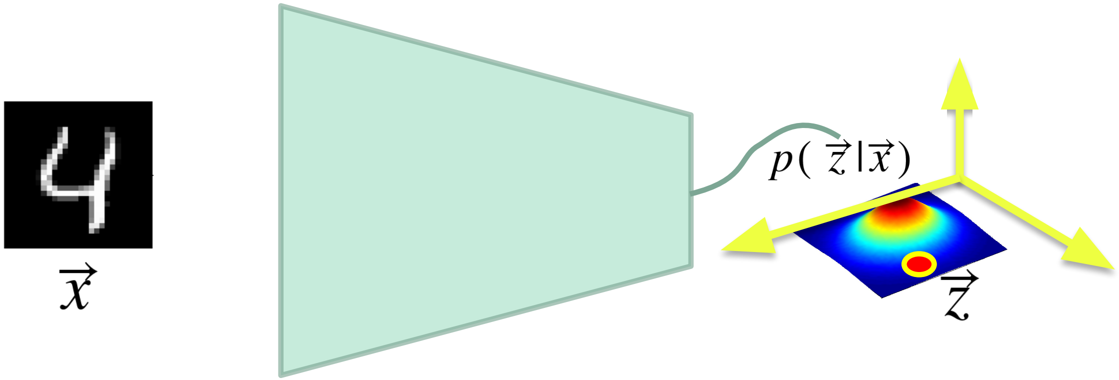
Basic autoencoder
$\vec{z}$
$\vec{x}$
$\hat{x}$
Variational autoencoder
$p(\vec{z}|\vec{x})$
$\vec{x}$
$\hat{x}$

Variational autoencoder



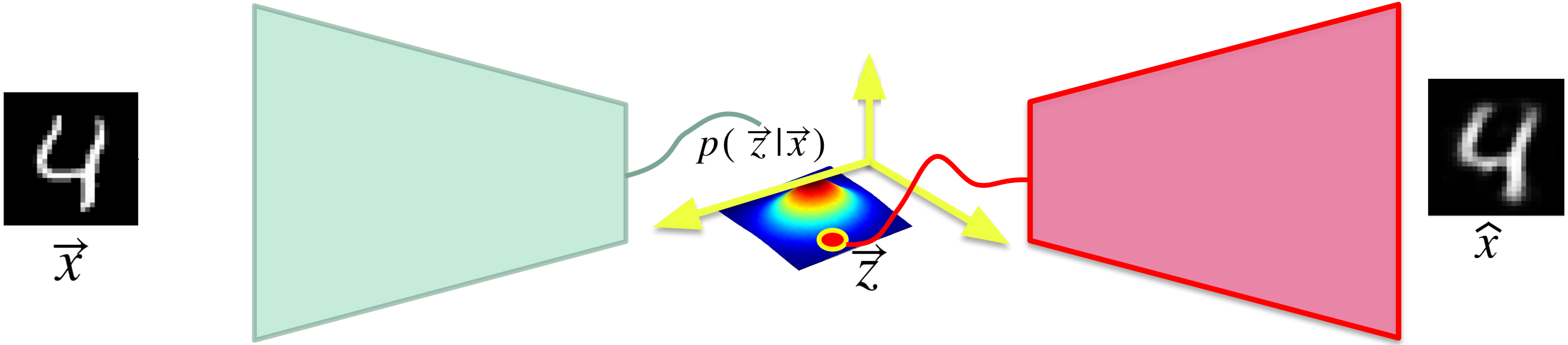
The encoder outputs a **distribution** $p(\vec{z}|\vec{x})$ and not just a **vector** $\vec{z}$

Variational autoencoder



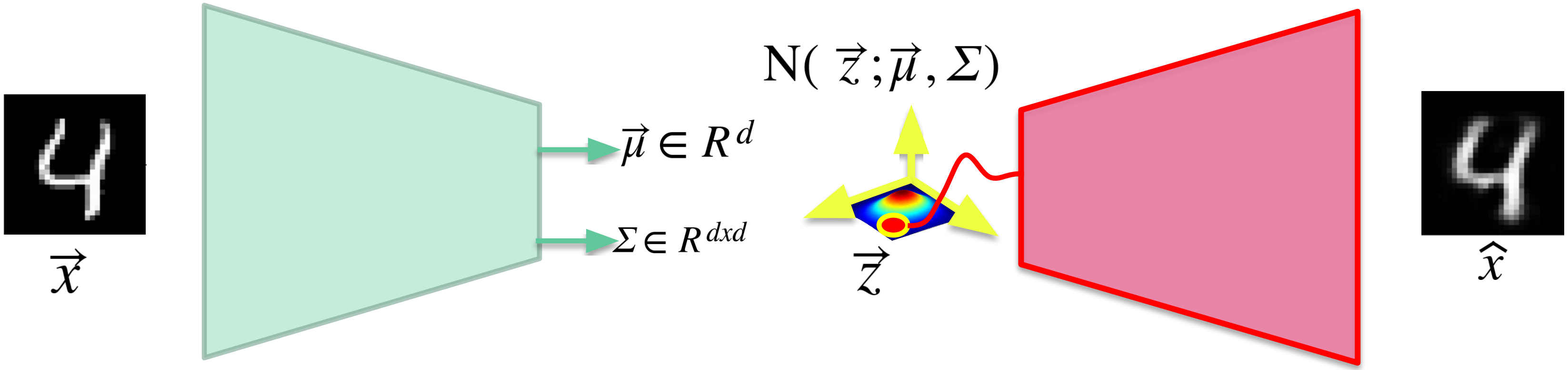
Random sample $\vec{z} \sim P(\vec{z}|\vec{x})$

Variational autoencoder

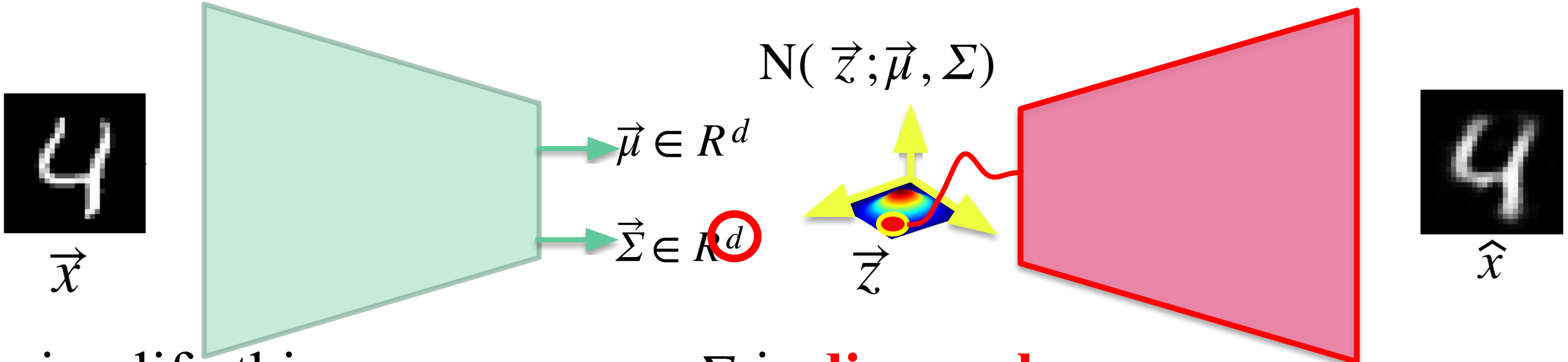


...and rebuild $\hat{x}$

$$p(\vec{z}|\vec{x}) \sim Gaussian$$



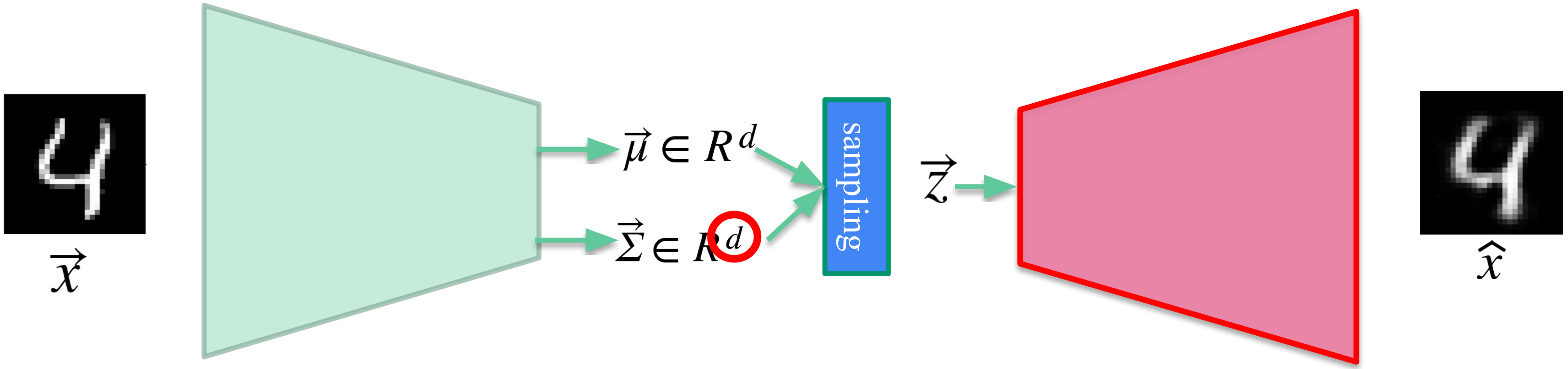
$$p(\vec{z}|\vec{x}) \sim Gaussian$$



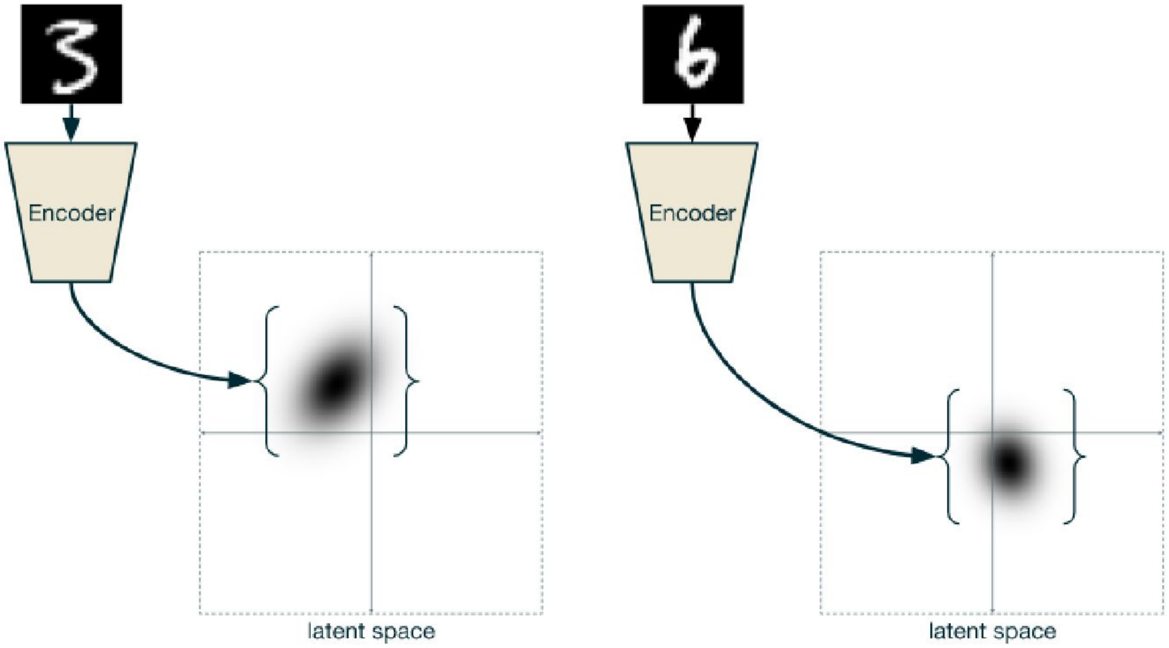
To simplify things, we assume Σ is **diagonal**

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_1^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma_d^2 \end{pmatrix}$$

Variational autoencoder

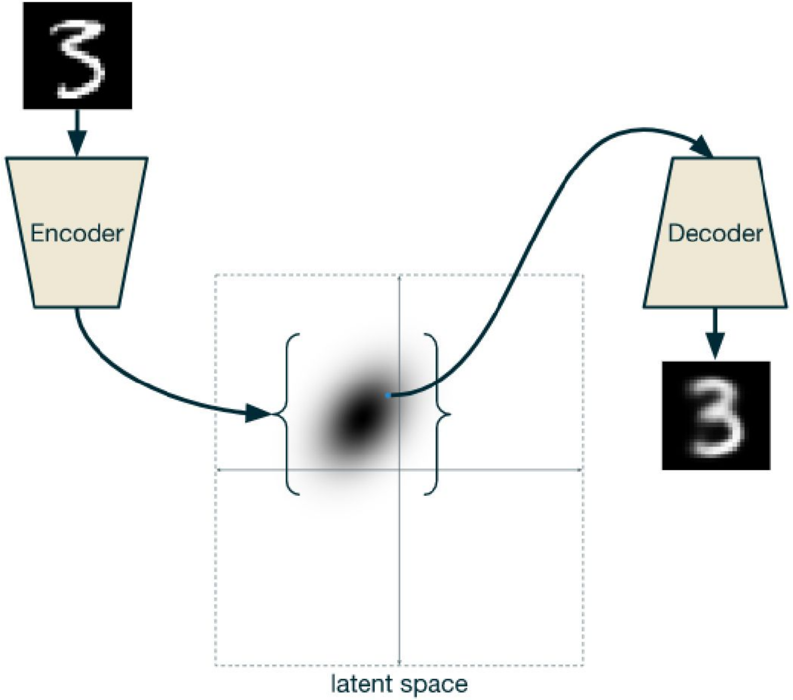


Another way of seeing things…



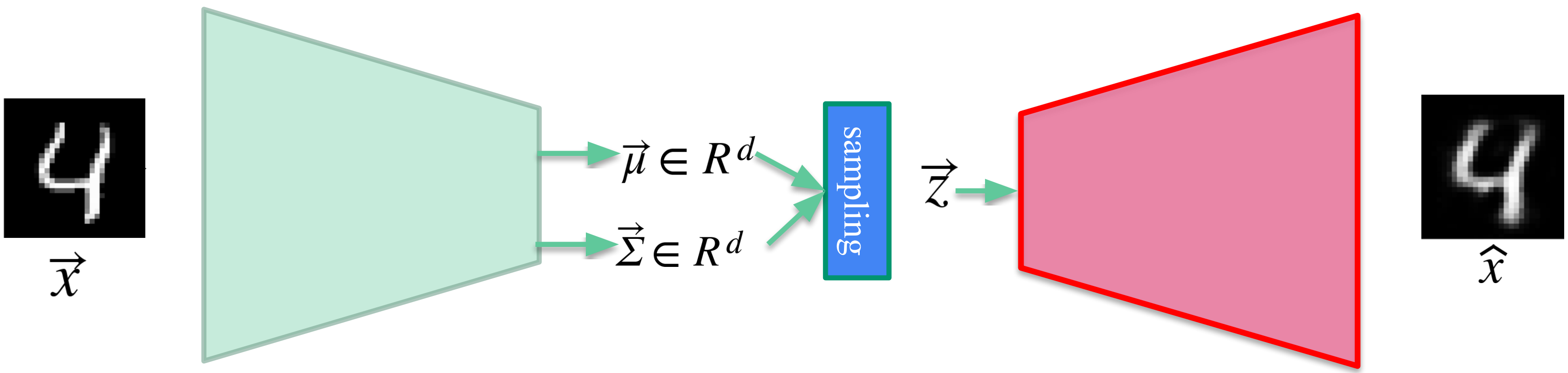
https://ijdykeman.github.io/ml/2016/12/21/cvae.html

Another way of seeing things…



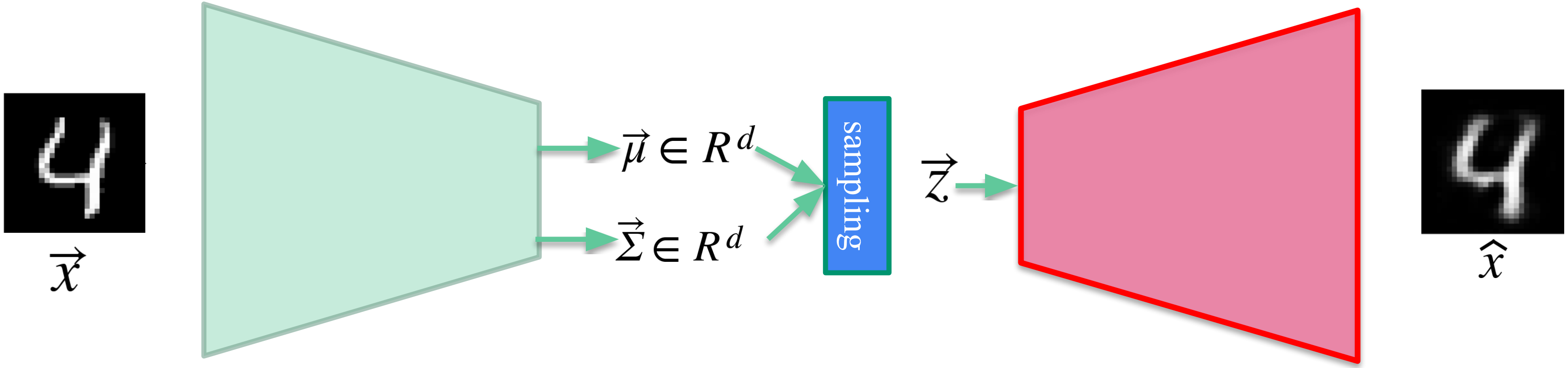
https://ijdykeman.github.io/ml/2016/12/21/cvae.html

ELBO loss : Evidence Lower Bound loss



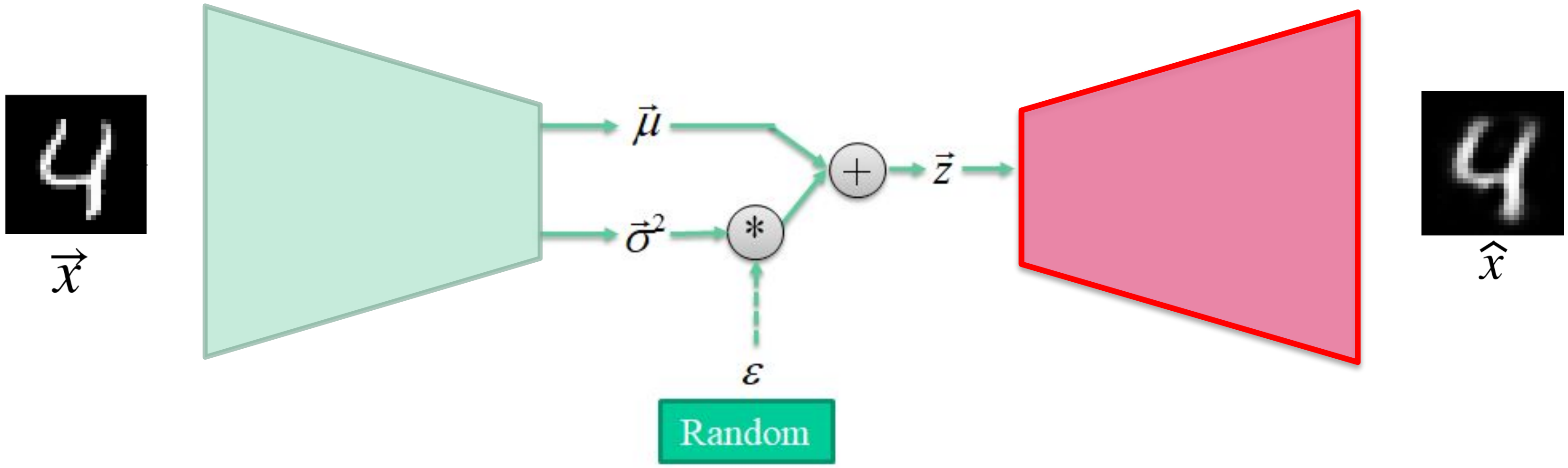
$$Loss = \underbrace{(\vec{x} - \hat{x})^2}_{\text{Loss decoder}} + \underbrace{\lambda KL\left(N(\vec{z}; \vec{0}, \vec{1}), N(\vec{z}; \vec{\mu}, \vec{\Sigma})\right)}_{\text{Loss encoder}}$$

Other loss (in case the output is binary)

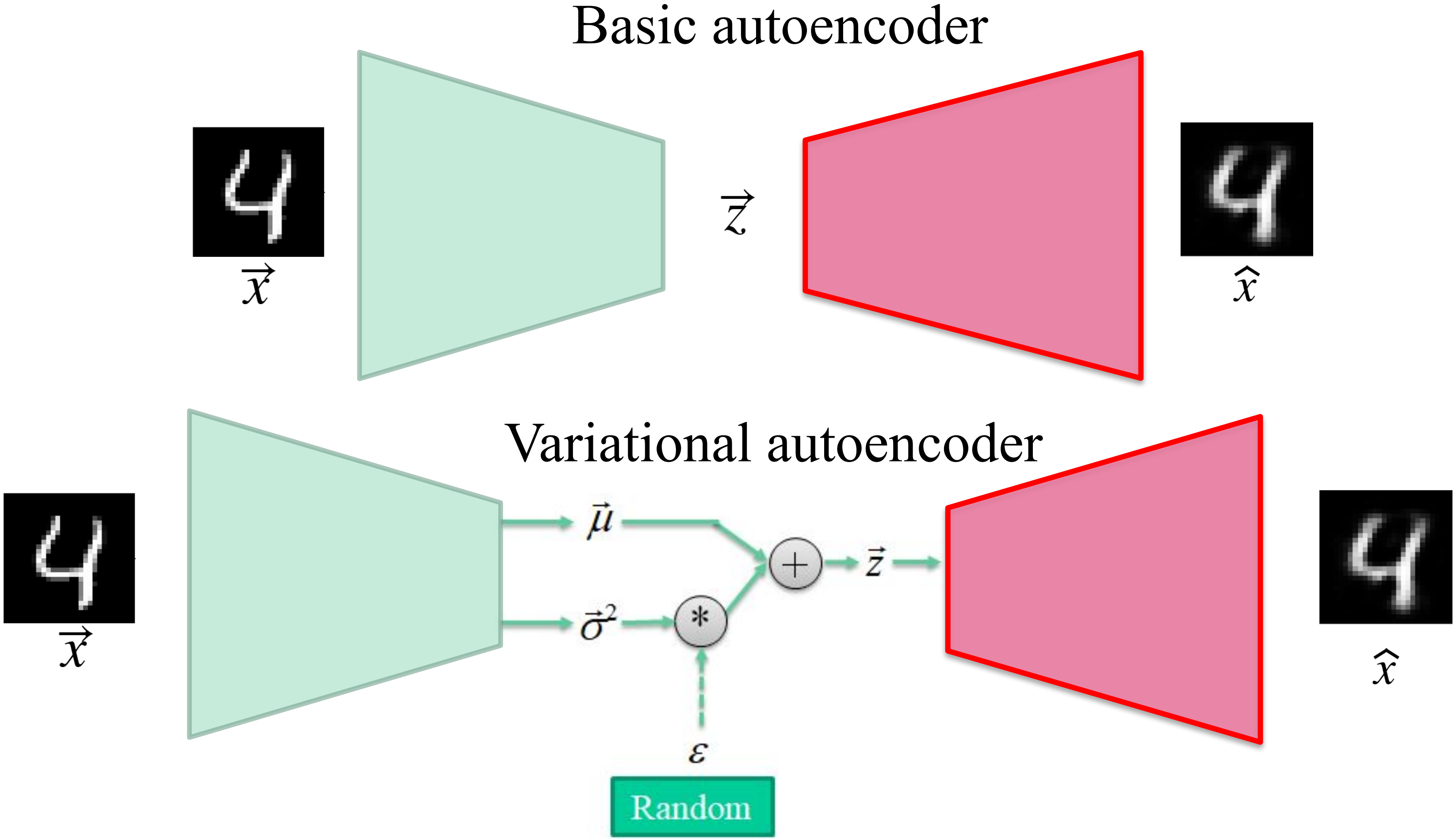


$$\underbrace{Loss = CrossEntropy(\vec{x}, \hat{x})}_{\text{Loss decoder}} + \underbrace{\lambda KL(\mathrm{N}(\vec{z}; \vec{0}, \vec{1}), \mathrm{N}(\vec{z}; \vec{\mu}, \vec{\Sigma}))}_{\text{Loss encoder}}$$

Reparametrization trick instead of sampling

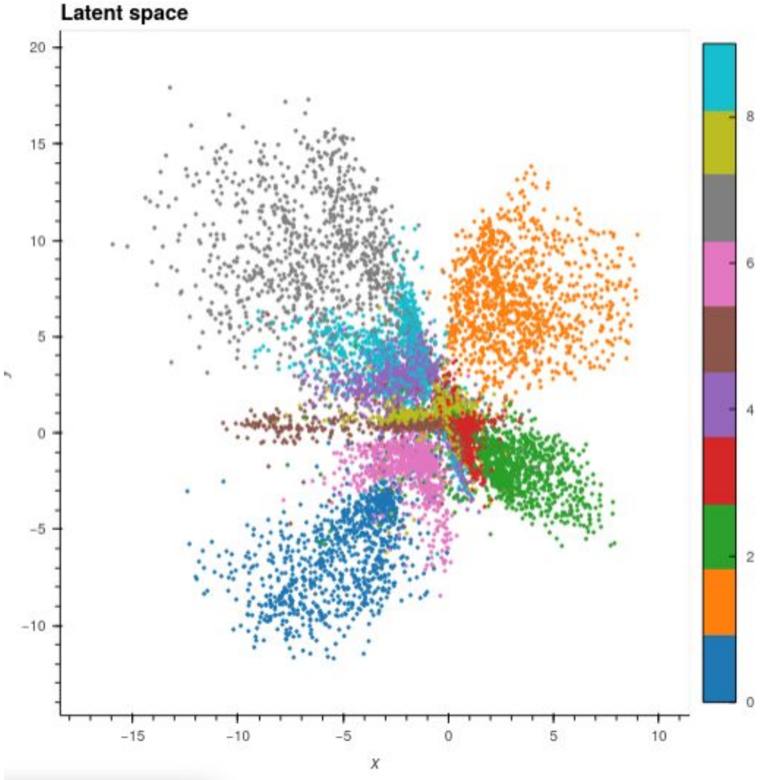


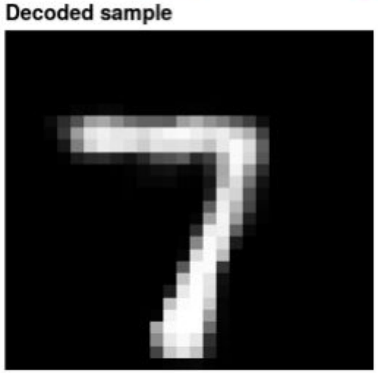
MNIST


Basic autoencoder
$\vec{x}$
$\vec{z}$
$\hat{x}$
Variational autoencoder
$\vec{x}$
$\vec{\mu}$
$\vec{\sigma}^2$
$+$
$*$
$\vec{z}$
ε
Random
$\hat{x}$

MNIST

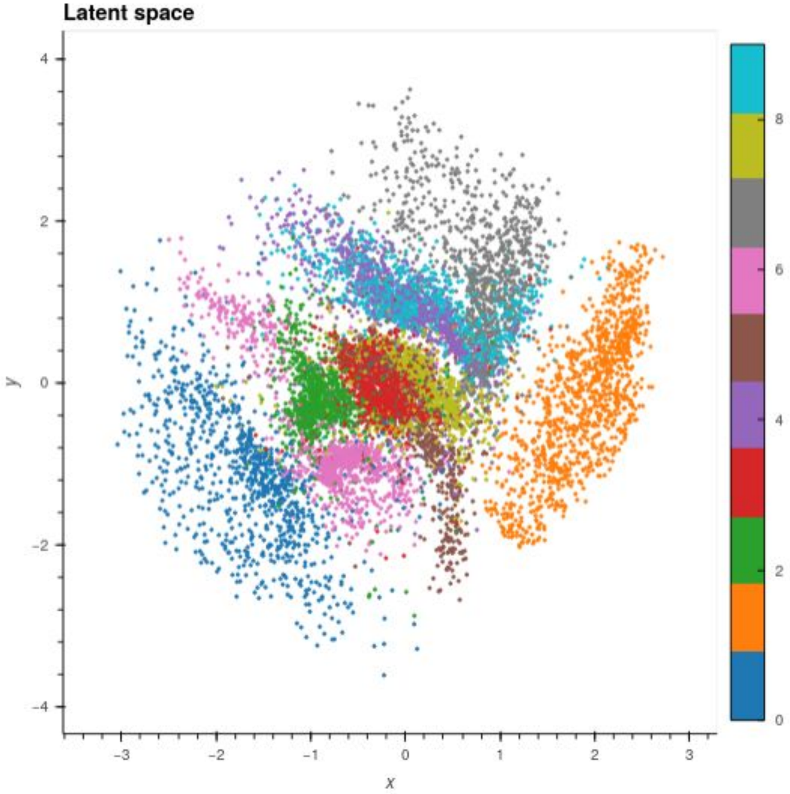
Visualize the latent space (autoencoder)

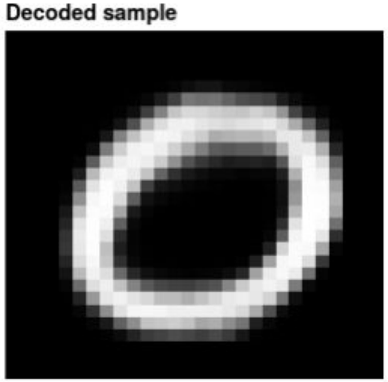




MNIST

Visualize the latent space (variational autoencoder)





MNIST

Code is in the file:

`mnist-autoencoders.ipynb`

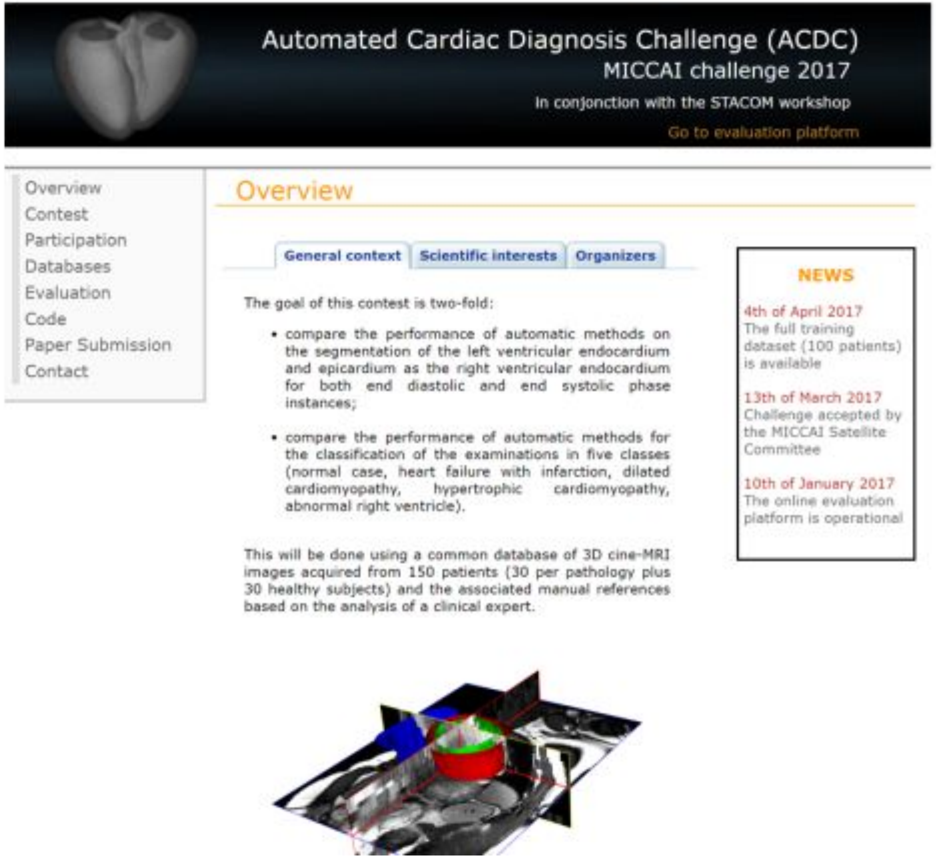
MICCAI 2017

Automated Cardiac Diagnosis Challenge

150 exams (all from different patients) divided into 5 groups:

- dilated cardiomyopathy (DCM),
- hypertrophic cardiomyopathy (HCM),
- myocardial infarction (MINF),
- abnormal right ventricle (RV)
- patients without cardiac disease (NOR).

website: ***creatis.insa-lyon.fr/Challenge/acdc***



Automated Cardiac Diagnosis Challenge (ACDC)
MICCAI challenge 2017
In conjonction with the STACOM workshop
Go to evaluation platform

- Overview
- Contest
- Participation
- Databases
- Evaluation
- Code
- Paper Submission
- Contact

Overview

General context | Scientific interests | Organizers

The goal of this contest is two-fold:

- compare the performance of automatic methods on the segmentation of the left ventricular endocardium and epicardium as the right ventricular endocardium for both end diastolic and end systolic phase instances;
- compare the performance of automatic methods for the classification of the examinations in five classes (normal case, heart failure with infarction, dilated cardiomyopathy, hypertrophic cardiomyopathy, abnormal right ventricle).

This will be done using a common database of 3D cine-MRI images acquired from 150 patients (30 per pathology plus 30 healthy subjects) and the associated manual references based on the analysis of a clinical expert.

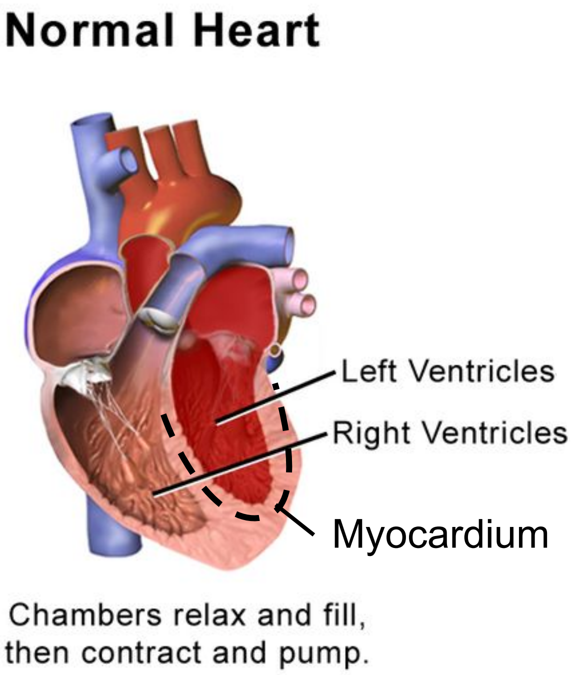
NEWS

4th of April 2017
The full training dataset (100 patients) is available

13th of March 2017
Challenge accepted by the MICCAI Satellite Committee

10th of January 2017
The online evaluation platform is operational

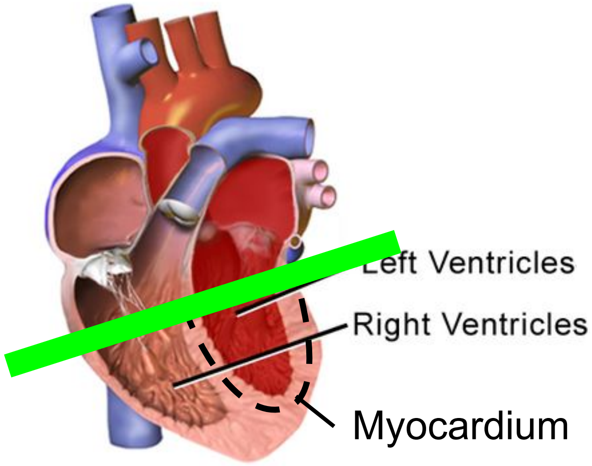
Cardiac anatomy



https://commons.wikimedia.org/

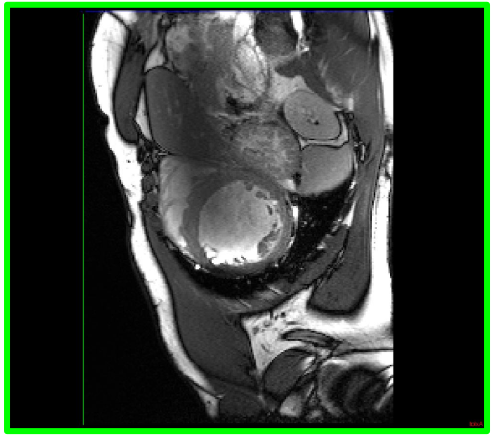
Cardiac anatomy

Normal Heart



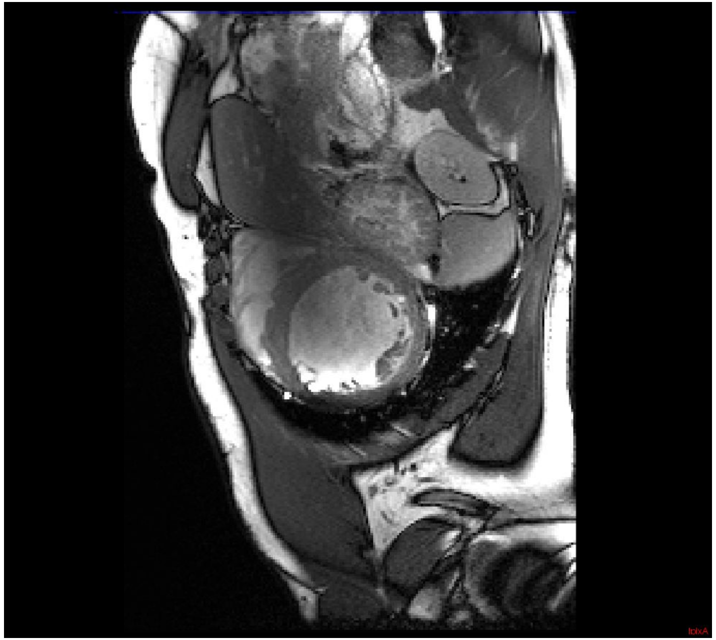
Chambers relax and fill,
then contract and pump.

Cross-section : **short axis view**

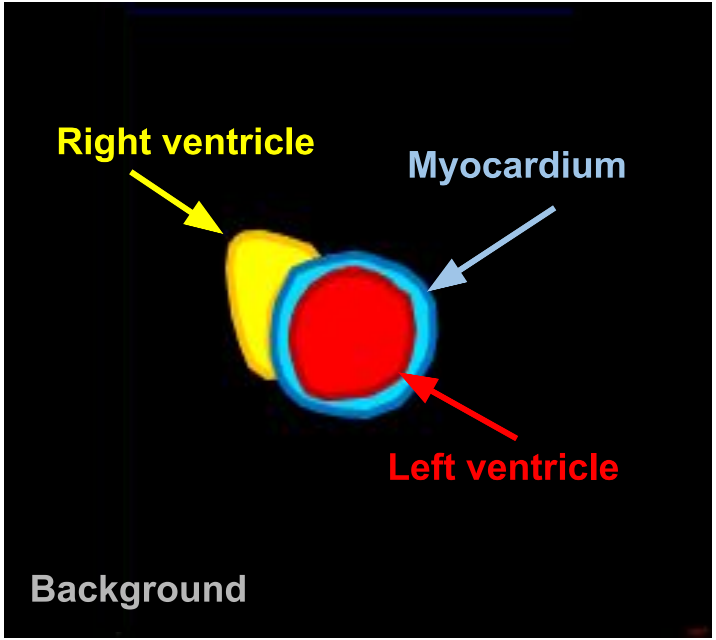


https://commons.wikimedia.org/

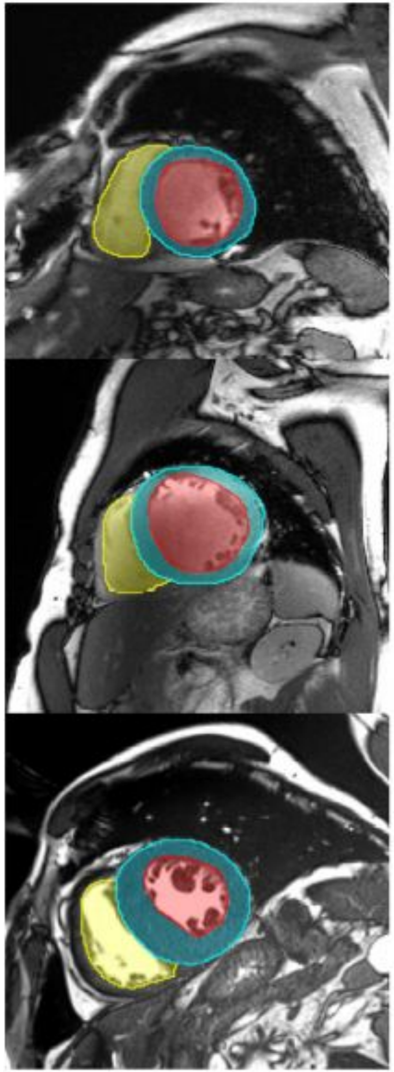
Short axis cardiac MRI



Short axis segmentation map

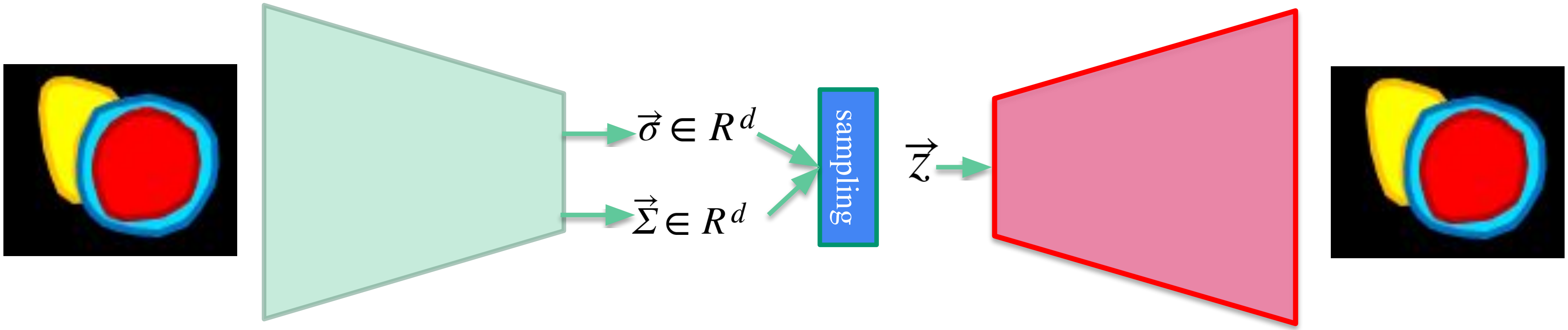


Other examples:



ACDC

Convolutional Variational autoencoder



ACDC

Code is in the file:

`cardiac-mri-autoencoders.ipynb`

Thank you