SAISON 22/23

Formation

Introduction au Deep Learning

Variational Autoencoder (VAE)
ou comment jouer avec les espaces latents ;-)

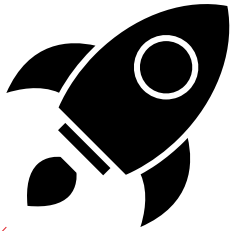
FIDLE

MIAI Grenoble Alpes — cnrs — UGA Université Grenoble Alpes

DEVLOG RÉSEAU — Resinfo — sari — ANITI — GRICAD — SiMaP

EFELIA - MIAI
ÉCOLE FRANÇAISE DE L'INTELLIGENCE ARTIFICIELLE

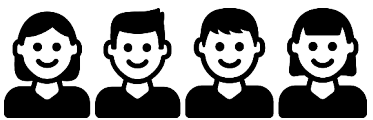
Annonce !



« Journée Deep Learning pour la Science ! »

Mai 2023, Orsay

In the real world !



Resources

https://fidle.cnrs.fr

Powered by CNRS CRIC, and UGA DGDSI
of Grenoble, Thanks !

Course materials (pdf)

Practical work environment*

Corrected notebooks

Videos (YouTube)

(*) Procedure via Docket or pip
Remember to get the latest version !

Resources

You can also subscribe to :

http://fidle.cnrs.fr/listeinfo
Fidle information list

https://listes.services.cnrs.fr/wws/info/devlog
List of ESR* « Software developers » group

https://listes.math.cnrs.fr/wws/info/calcul
List of ESR* « Calcul » group

(*) ESR is Enseignement Supérieur et Recherche, french universities and public academic research organizations

Program

20 Séquences
du 17 novembre
au 14 mai 2023

1

History, Fundamental Concepts


Basic Regression
DNN


Basic Classification
DNN

2 3 ✓

Hight Dimensionnal Data
CNN

4 ✓

Demystify **mathematics** for neural networks.

5 ✓

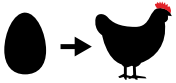
Training strategies
Evaluation

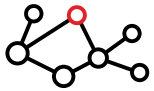

Sparse data (text)
Embedding

6 ✓

Sequences data
RNN

7 ✓
PyTorch
A small detour with **PyTorch**.

8 ✓

«Attention is All You Need»
Transformers

9 ✓ New !

Graph Neural Network
GNN

10 ✓

Autoencoder networks
AE

11

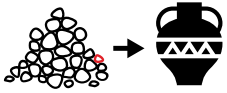
Variational Antoencoder
VAE

12

Project session
«My project in 180 s»

13

Generative Adversarial Networks
GAN

14 New !

Diffusion Model
Text to image

15

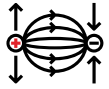
AI, Law, Society and **Ethics**

16 New !

Model and training optimization
Resource efficiency

17

Jean-Zay
GPU acceleration

18 New !

Physics-Informed Neural Networks
PINNS

19

Deep Reinforcement Learning
RL

20

What will be tomorrow's AI
Review & perspectives !

FIDLE

SAISON 22/23

Roadmap

11

Variational Antoencoder

VAE

11.1 **Variational Autoencoder network**

- Concepts and architecture
- Gaussian / probabilistic projections
- Kullback-Leibler divergence
- Morphing in latent space

11.2 **Example 1 : VAE1/3**

- Deeper in Keras !
- VAE using Functional API (MNIST)
- VAE using Model subclass (MNIST)

Roadmap

11

Variational Antoencoder

VAE

11.1 **Variational Autoencoder network**

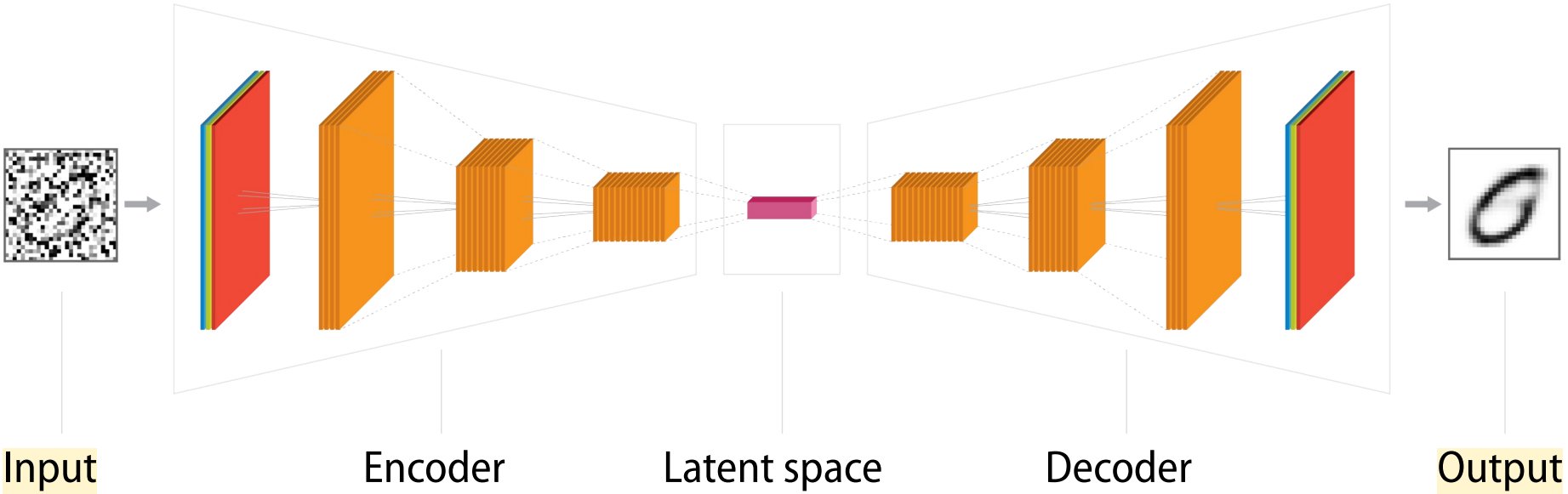
- Concepts and architecture
- Gaussian / probabilistic projections
- Kullback-Leibler divergence
- Morphing in latent space

11.2 **Example 1 : VAE1/3**

- Deeper in Keras !
- VAE using Functional API (MNIST)
- VAE using Model subclass (MNIST)

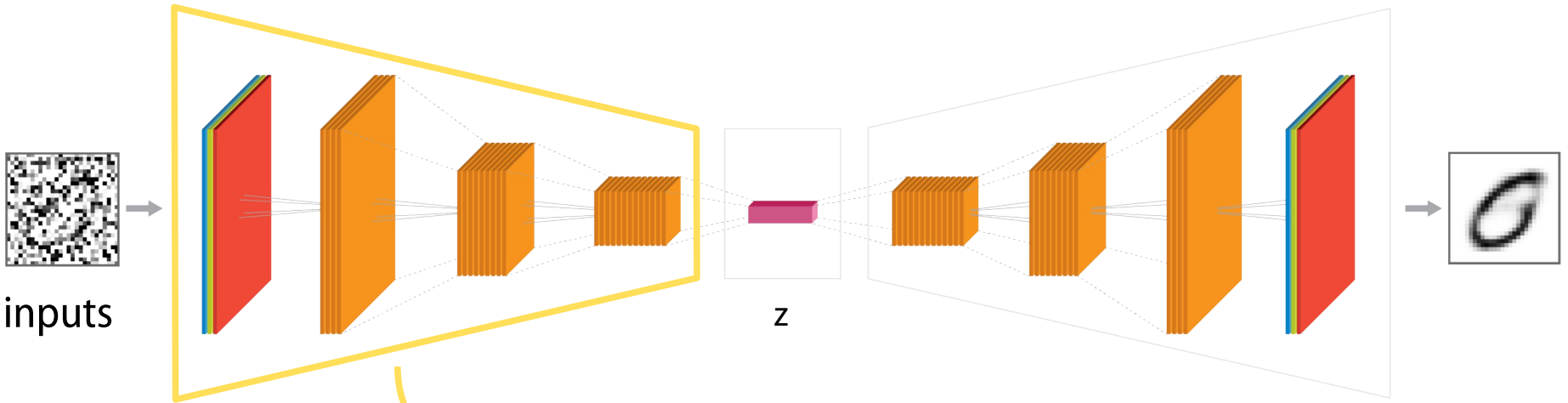
Concepts & architecture

We have seen that an autoencoder network is trained to minimize a reconstruction error :



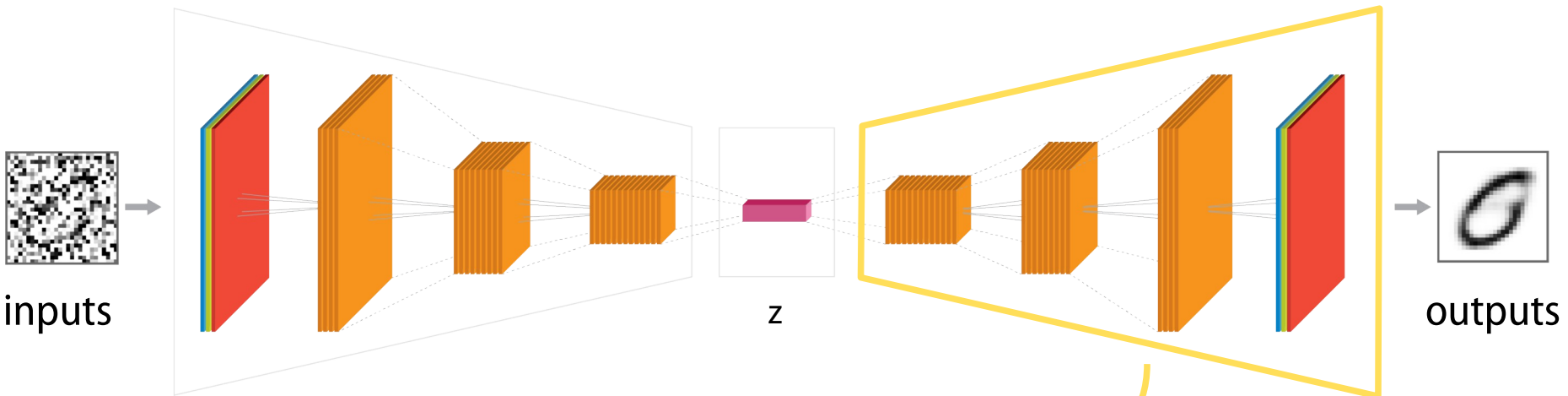
Autoencoder

An autoencoders network have 2 parts :



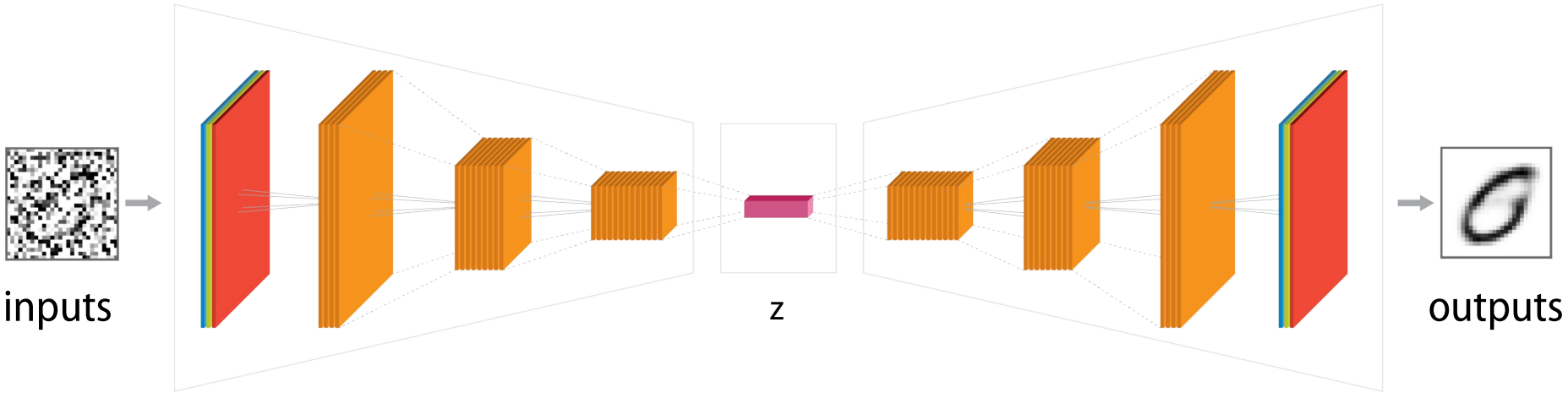
Autoencoder

An autoencoders network have 2 parts :



Autoencoder

An autoencoders network have 2 parts :



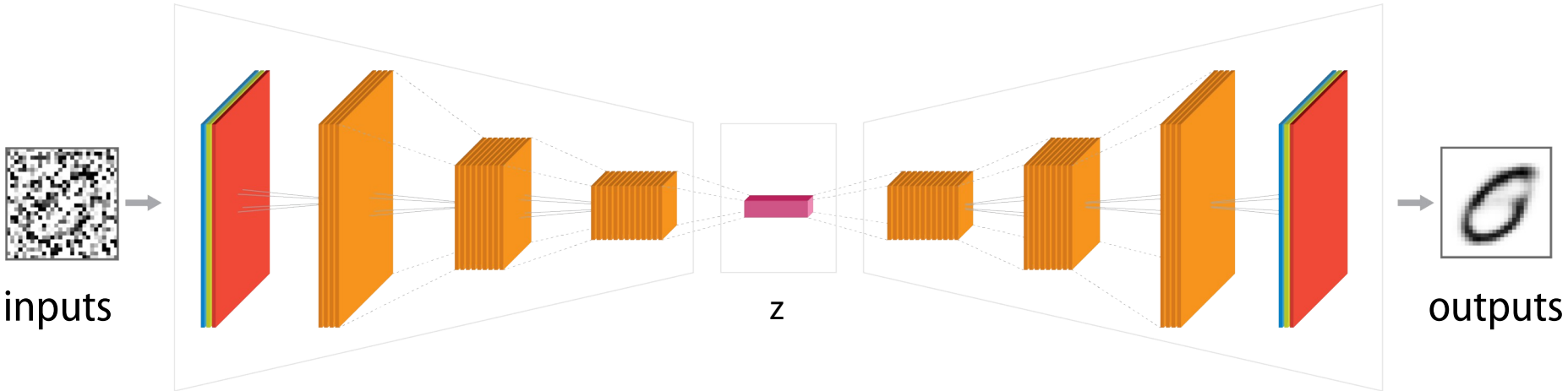
z = encoder(inputs)

outputs = decoder(z)

ae = keras.Model(inputs, outputs)

Autoencoder

An autoencoders network have 2 parts :



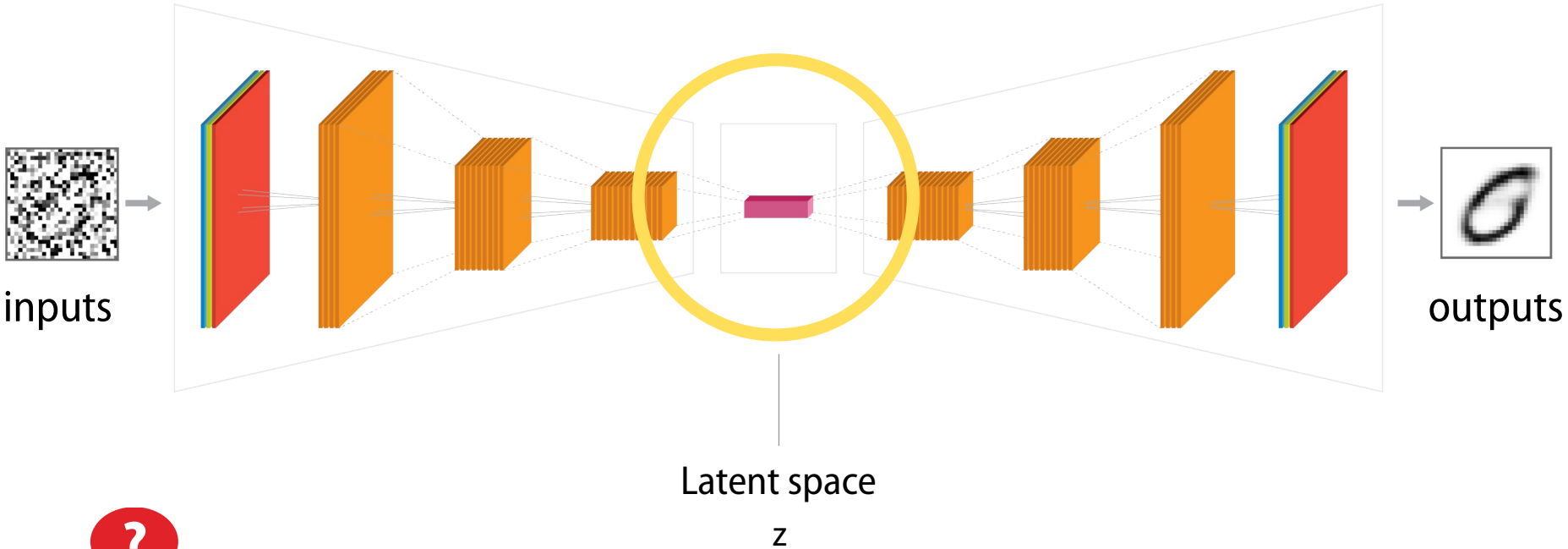

```
z = encoder(inputs)

outputs = decoder(z)
```

```
ae = keras.Model(inputs, outputs)
```

Autoencoder

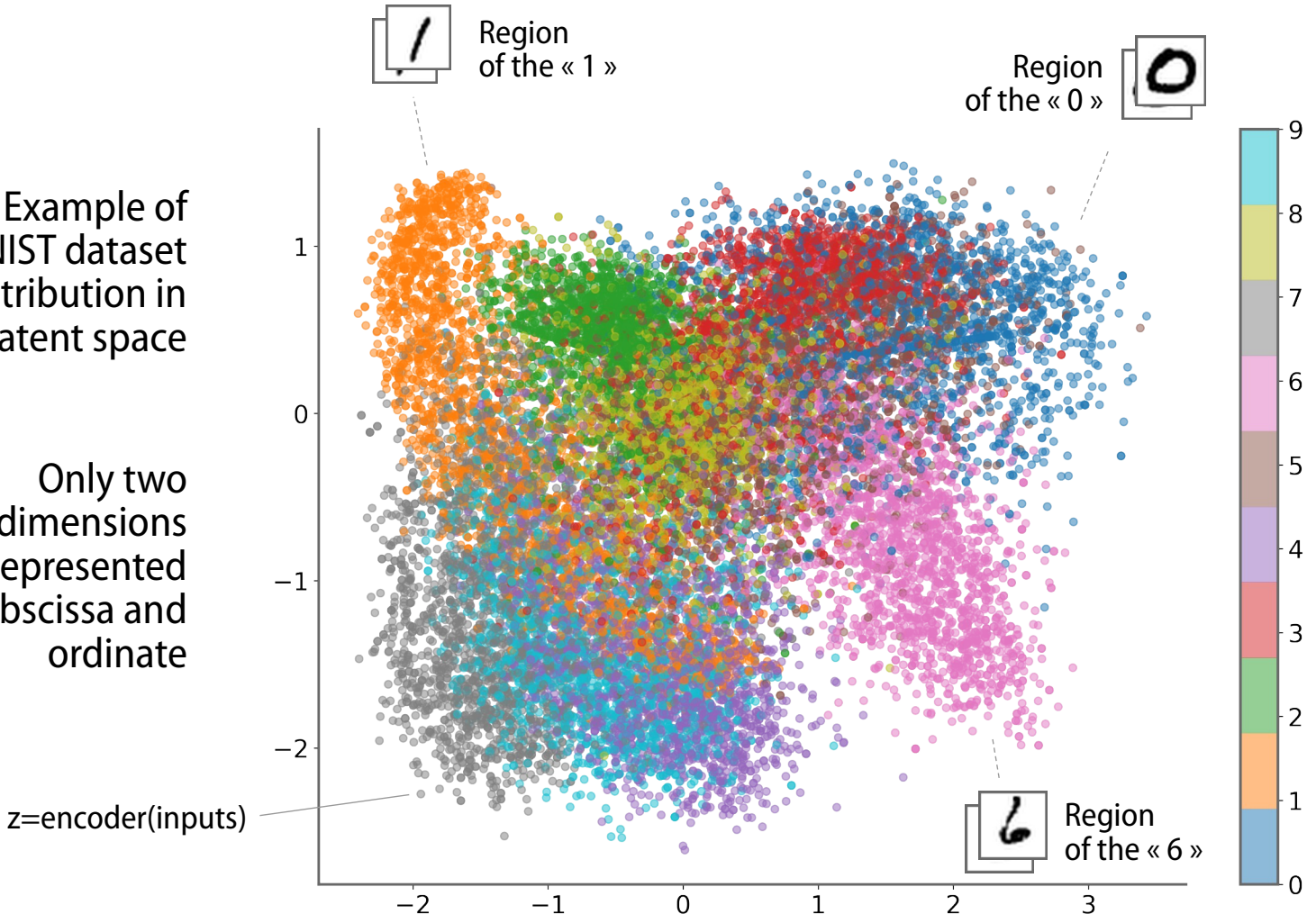
An autoencoders network have ~~2~~ 3 parts :



Autoencoder

Example of MNIST dataset distribution in its latent space

Only two dimensions are represented in abscissa and ordinate



Clusters appear, but many of them are nested or very spread out

How can we make our network better separate the different clusters?

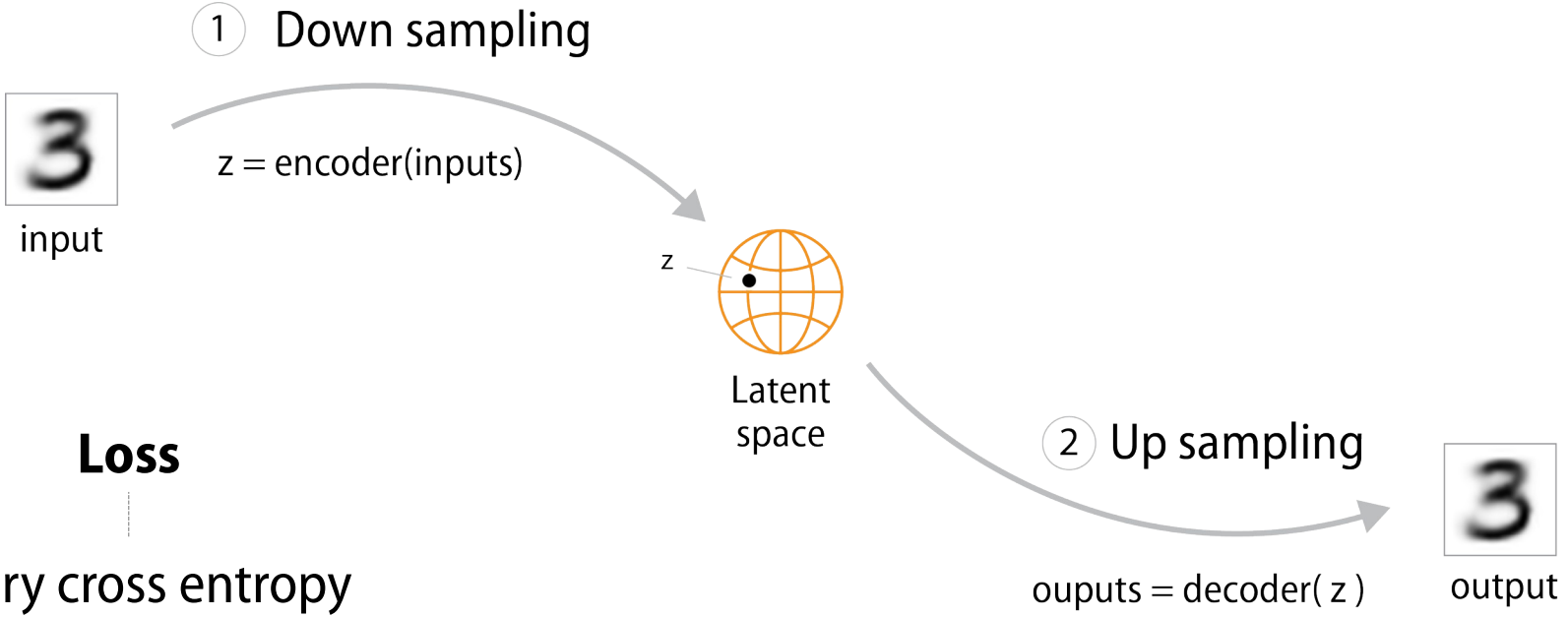
➔ Just train it !

See : notebook [AE3]

Autoencoder (AE)

z vector in latent space

> An autoencoder performs a direct projection into the latent space and an upsampling from this latent space.



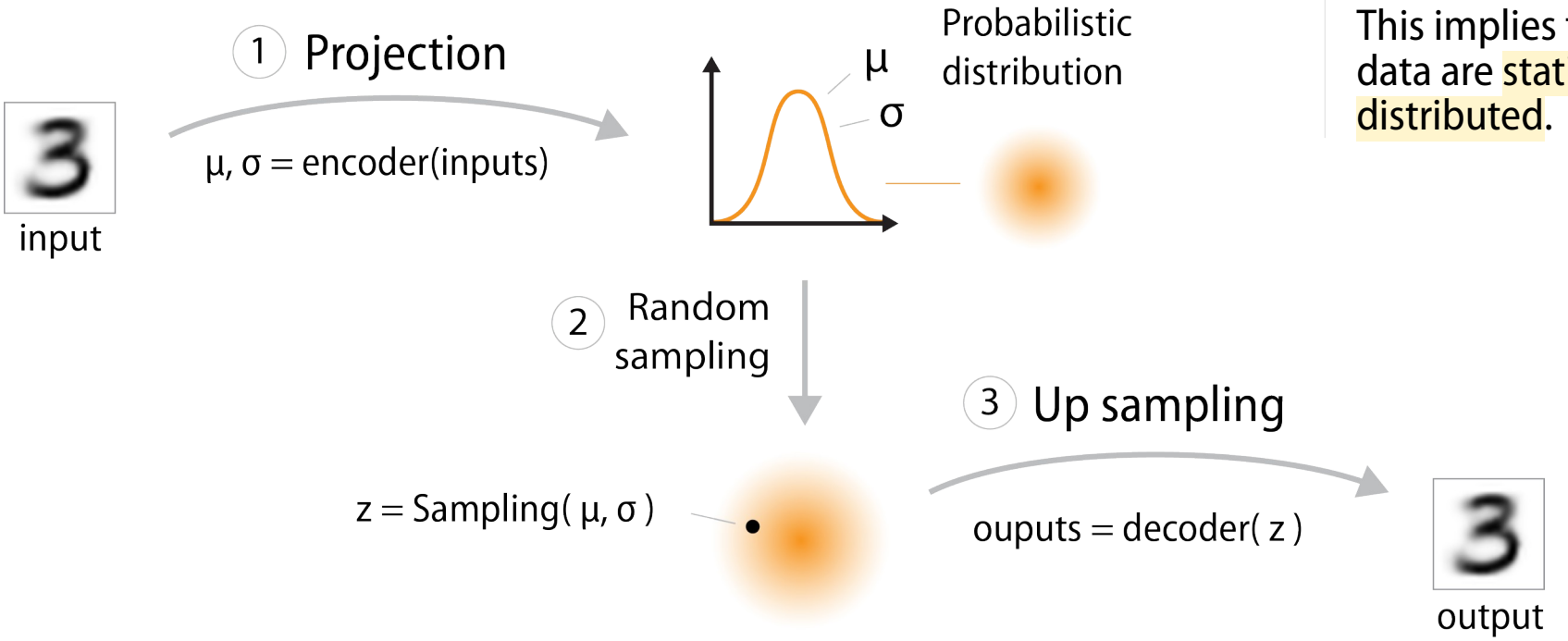
Loss

Binary cross entropy

Measures the difference between input and output

Variational Autoencoder (VAE)

μ is a mean
σ is a variance
z vector in latent space

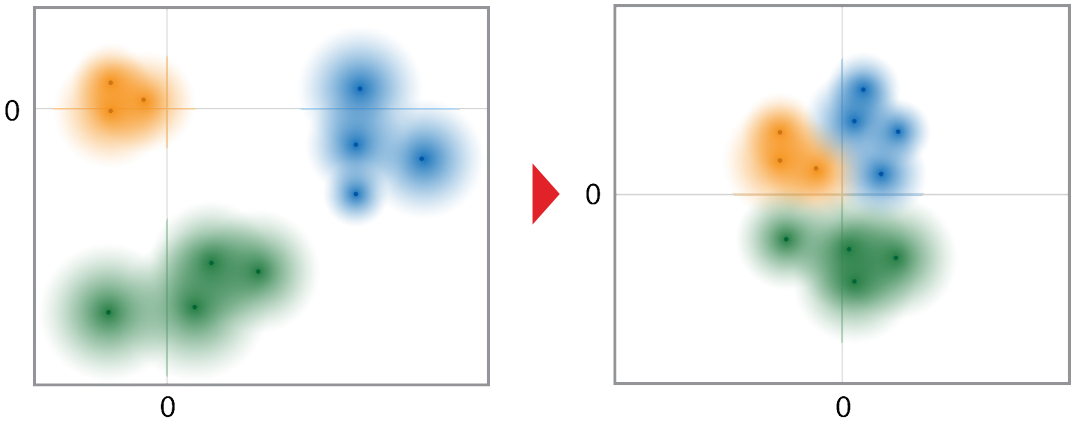


The stochastic approach encourages the network to generate relevant output throughout the distribution space.

This implies that the input data are statistically distributed.

Diederik P Kingma; Welling, Max (2013). "Auto-Encoding Variational Bayes".
https://arxiv.org/abs/1312.6114

Variational Autoencoder (VAE)



Kullback-Leibler divergence*

$$D_{\mathrm{KL}} = \frac{1}{2}\sum_{i=1}^{k}\left(\sigma_i^2 + \mu_i^2 - 1 - \ln\left(\sigma_i^2\right)\right).$$

KL divergence between two (probability) distributions measures how much they diverge from each other

KL loss

Reconstruction loss

Binary cross entropy

Measures the difference between input and output

Total loss

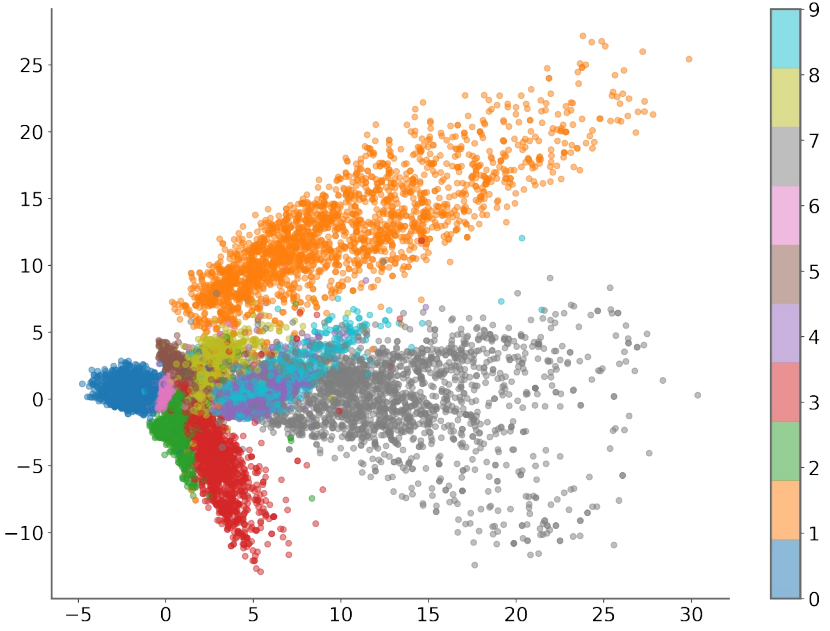
$$\text{total}_{\text{loss}} = k_1.r_{\text{loss}} + k_2.kl_{\text{loss}}$$

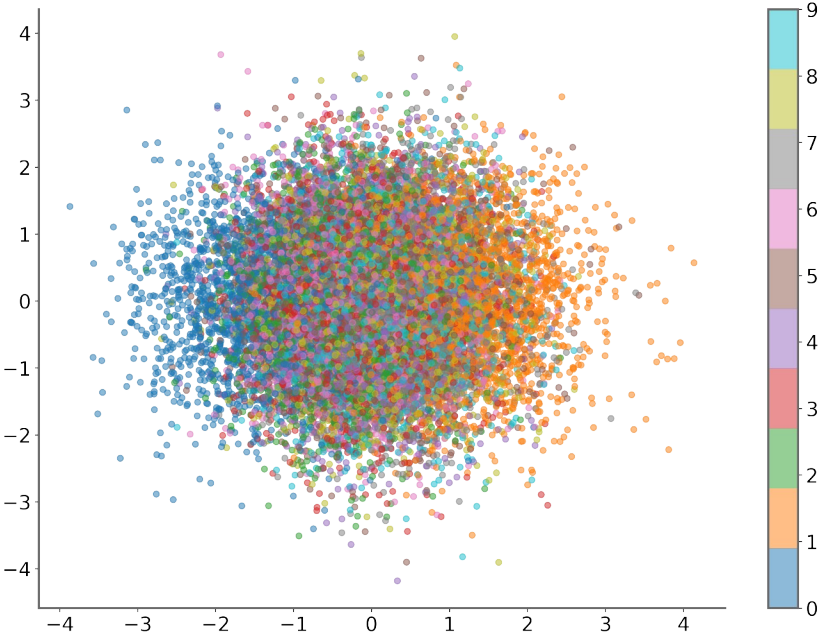
The trick is to find a nice compromise between these two components of the loss function.

(*) Special case for a standard normal distribution

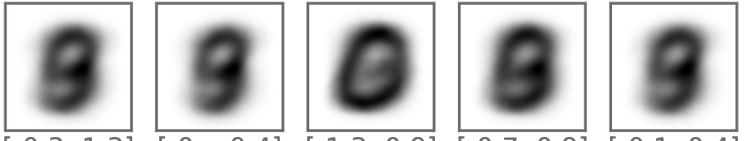
Variational Autoencoder (VAE)

$$total_{loss} = k_1.r_{loss} + k_2.kl_{loss}$$





[4.3 -1.2] [4.1 0.1] [-0.2 0.4] [4.5 0.6] [19.3 1.]



[-0.3 1.3] [-0. -0.4] [-1.3 0.9] [-0.7 0.9] [-0.1 0.4]

Variational Autoencoder (VAE)

Easy !

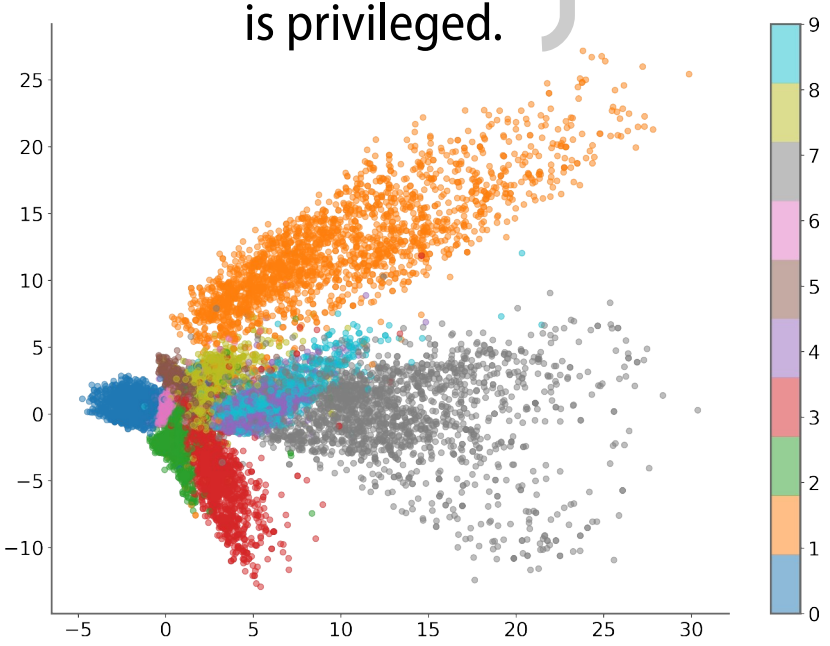
k_1 is too **high**

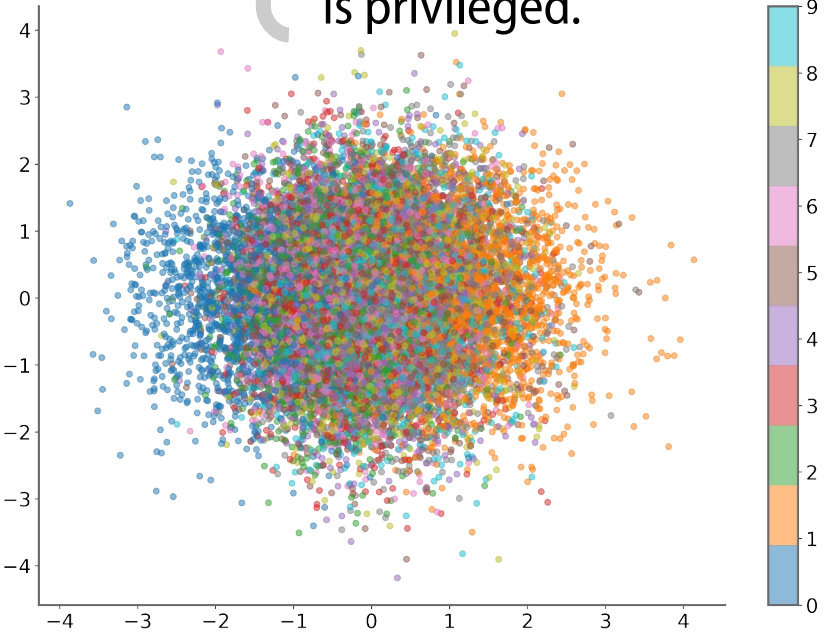
Reconstruction is privileged.

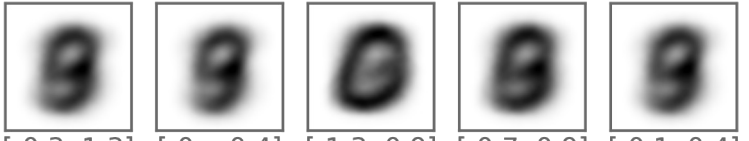
$$total_{loss} = k_1.r_{loss} + k_2.kl_{loss}$$

k_2 is too **high**

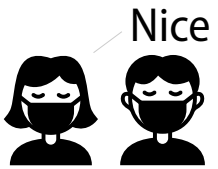
Agglomeration is privileged.







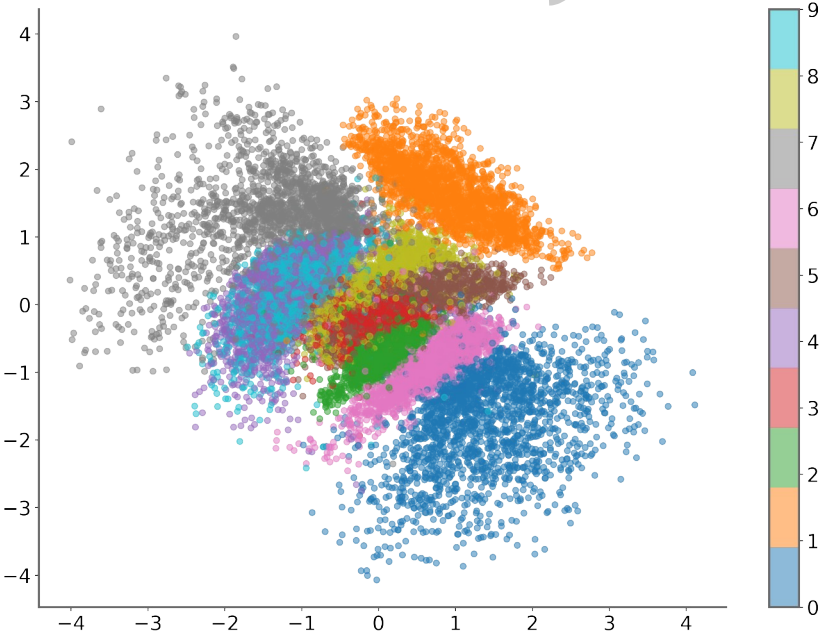
Variational Autoencoder (VAE)

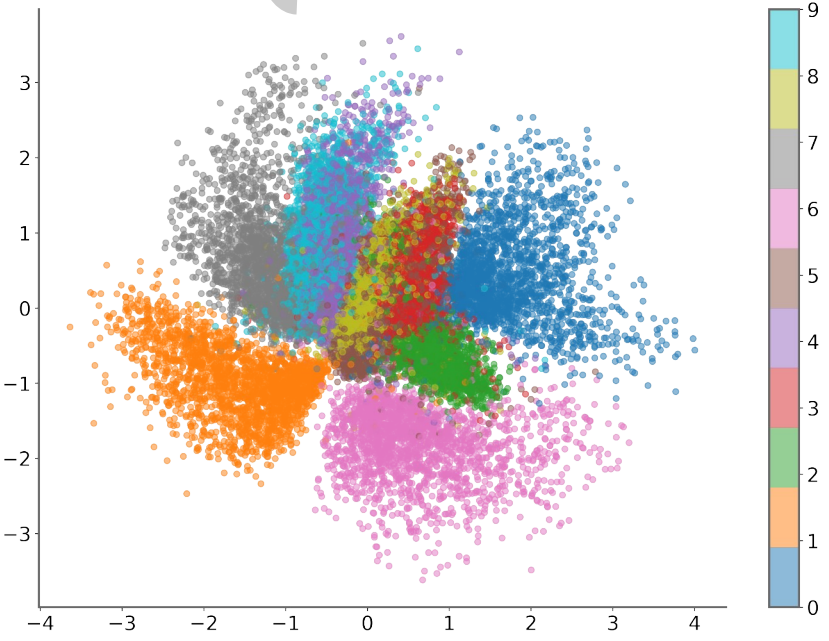


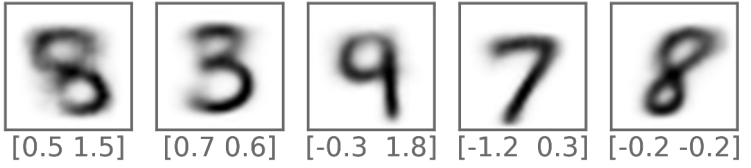
VAE2, with :
$k_1, k_2 = [1, 5e\text{-}4]$

$$total_{loss} = k_1.r_{loss} + k_2.kl_{loss}$$

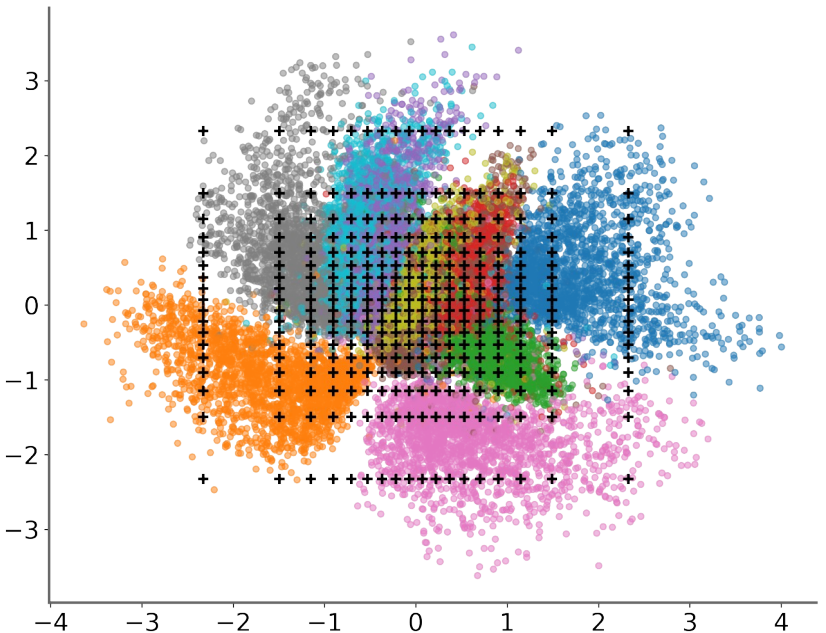
VAE2, with :
$k_1, k_2 = [1, 1e\text{-}3]$

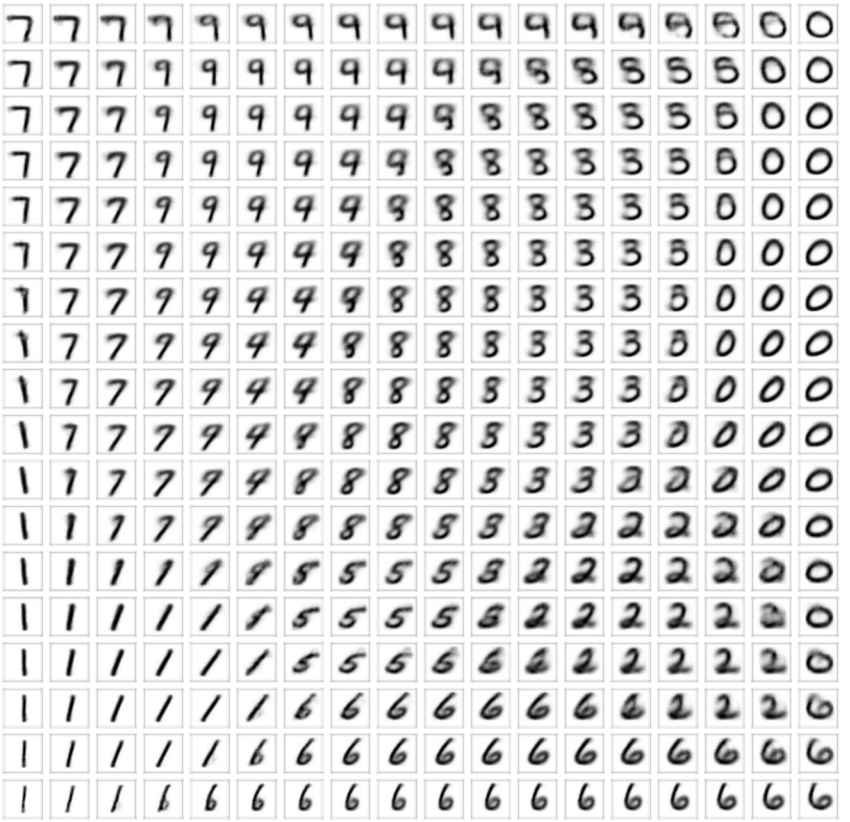






Variational Autoencoder (VAE)





Tha's great ! We can generate data in profusion !!!

Roadmap

11

Variational Antoencoder

VAE

11.1 **Variational Autoencoder network**

- Concepts and architecture
- Gaussian / probabilistic projections
- Kullback-Leibler divergence
- Morphing in latent space

11.2 **Example 1 : VAE1/3**

- Deeper in Keras !
- VAE using Functional API (MNIST)
- VAE using Model subclass (MNIST)

Roadmap

11

Variational Antoencoder

VAE

11.1 **Variational Autoencoder network**

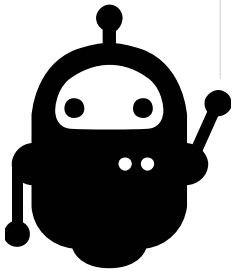
- Concepts and architecture
- Gaussian / probabilistic projections
- Kullback-Leibler divergence
- Morphing in latent space

11.2 **Example 1 : VAE1/3**

- Deeper in Keras !
- VAE using Functional API (MNIST)
- VAE using Model subclass (MNIST)

Our first VAE

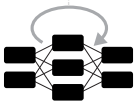
Notebook : [VAE1/3]



Objectives :
implementating a VAE, using Keras functionel API and model subclass, using real Python !

Dataset :
MNIST

20' on a CPU
With scale=1



Import and init

Retrieve the dataset

Train

START

Step 1

Step 2

Step 3

Step 4

Step 5

Step 6

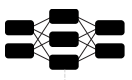
END

Parameters

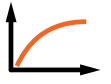
```
latent_dim    = 2
loss_weights = [1,.001]

scale         = .1
seed          = 123
```

Build models

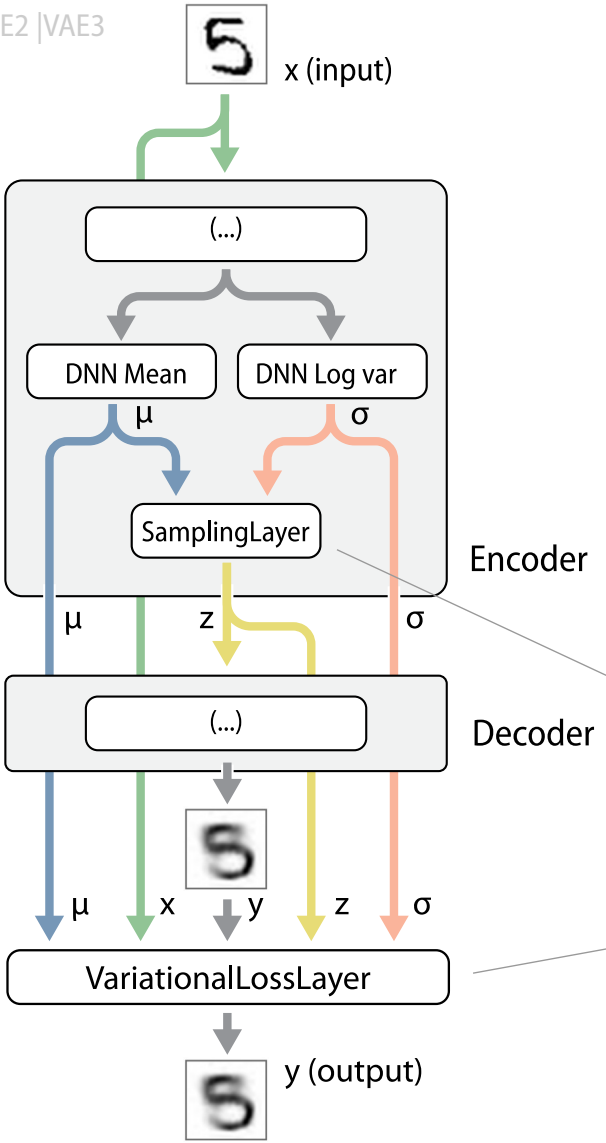


Training review



Objectives :

VAE using Keras functional API and MNIST dataset



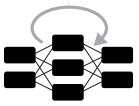
The classical loss functions use only the inputs and the outputs tensors.

The calculation of the loss is here more complex and will be done via a specific layer.

So we have two custom layers in this example :

- SamplingLayer
- VariationalLossLayer

20 ' on a CPU
With scale=1



Import and init

Retrieve the dataset

Train

START

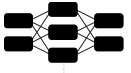
END

Parameters

```
latent_dim    = 2
loss_weights = [1,.001]

scale         = .1
seed          = 123
```

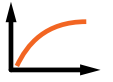
Build models



SamplingLayer
module

VariationalLossLayer
module

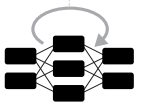
Training review



Objectives :

VAE using Keras functional API and MNIST dataset

20' on a CPU
With scale=1



Import and init

Retrieve the dataset

Train

Training review

START

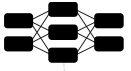
END

Parameters

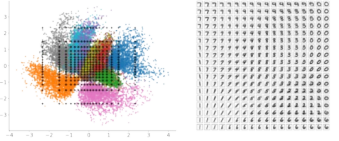
```
latent_dim    = 2
loss_weights = [1,.001]

scale         = .1
seed          = 123
```

Build model

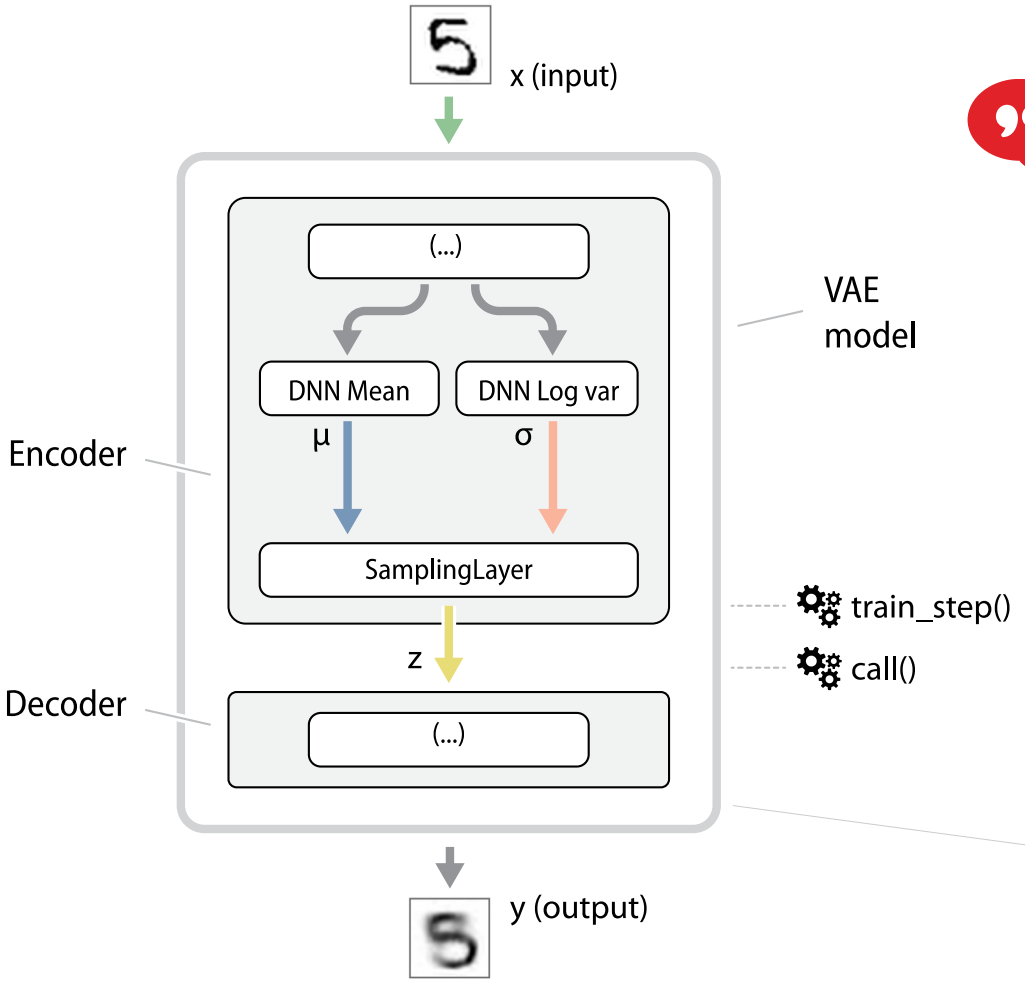


Model evaluation



Objectives :

VAE using Keras subclass API and MNIST dataset



The encoder and decoder are always built via the functional API and a custom layer.

The model is defined by a custom class in real Python :-)

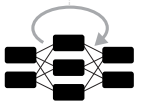
The gradient descent and the the loss function are defined via a method.

We have one custom layer and one custom model class :

SamplingLayer

VAE

20' on a CPU
With scale=1



Import and init

Retrieve the dataset

Train

Training review

START

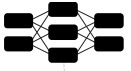
Step 1 → Step 2 → Step 3 → Step 4 → Step 5 → Step 6 → Step 7 → END

Parameters

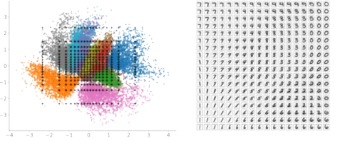
```
latent_dim    = 2
loss_weights = [1,.001]

scale         = .1
seed          = 123
```

Build model



Model evaluation



Objectives :

VAE using Keras subclass API and MNIST dataset

Few seconds !

MNIST module

MNIST

START

Step 1

Import and init

Parameters

```
scale = .1
seed  = 123
```

Step 2

Retrieve the dataset

Step 3

Reload model

SamplingLayer module

VAE module

Step 4

Images reconstruction

Step 5

Visualizing latent space

Step 6

Generate fom latent space

END

Objectives :

Reload a saved model and visualize latent space

Little things and concepts to keep in mind

- Unsupervised learning is great!
- The use cases of the VAE are very wide
  - Anomaly detection
  - Clustering
  - Semi-supervised classification
  - ...
- VAEs are often coupled with other tools
- Balancing the loss function is not always easy to find
- GPUs are great!
- Other (and better?) generative networks exist
- ...

Next, on Fidle :

13

Generative Adversarial Networks

GAN

Jeudi 9 mars,

Épisode 13 :

Generative Adversarial Networks (GAN)

Principes et architecture d'un GAN
Generator - Discriminator - Apprentissage
WGAN et WGAN-GP
Exemple proposé : Dessine moi un mouton !

Durée : 2h

« Journée Deep Learning pour la Science ! »

Mai 2023, Orsay

Mon projet en 180 secondes !

Next, on Fidle :

13

Generative Adversarial Networks

GAN

Jeudi 9 mars,

Épisode 13 :

Generative Adversarial Networks (GAN)

Principes et architecture d'un GAN
Generator - Discriminator - Apprentissage
WGAN et WGAN-GP
Exemple proposé : Dessine moi un mouton !

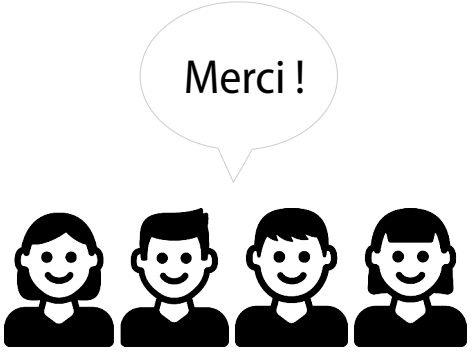
Durée : 2h

Next on Fidle :

Jeudi 9 mars,

Séquence 13 :

Generative Adversarial Networks (GAN)
Principes et architecture d'un GAN



To be continued...

Contact@fidle.cnrs.fr
FIDLE https://fidle.cnrs.fr
YouTube https://fidle.cnrs.fr/youtube

FIDLE