

07MIAR – Redes neuronales y deep learning



Universidad
Internacional
de Valencia

Práctica Observacional: Desarrollo de un autoencoder convolucional para la reconstrucción de placas RX

01

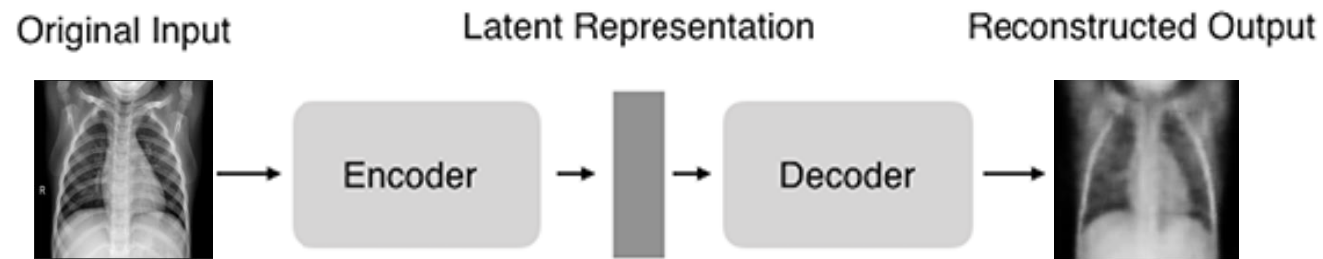
Introducción

Tareas avanzadas de *computer vision* empleando aprendizaje profundo

06/11/2021

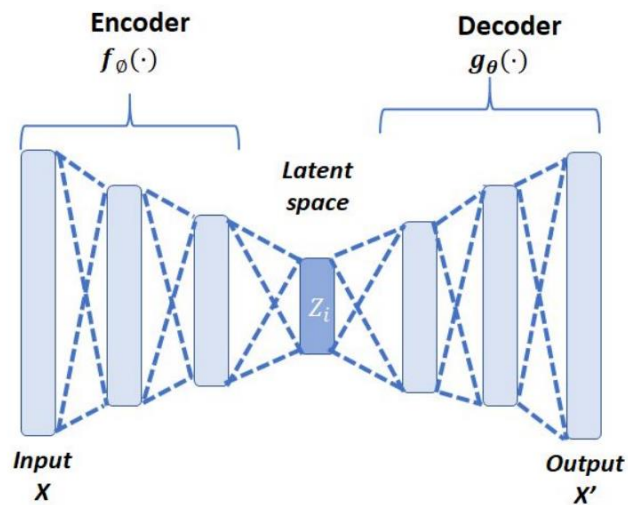
Contexto

- Llevar a cabo el **proceso de reconstrucción** de imagen mediante un **autoencoder** para **posteriormente** dar solución a **tareas de computer vision** como *denoising*, compresión de datos, detección de anomalías o sistemas *content based image retrieval* es de vital **importancia en multitud de sectores**.
- Uno de estos sectores es al **ámbito sanitario**. Tras la pandemia provocada por la **covid19**, son muchos los sets de **datos públicos** que existen para llevar a cabo la **detección** de la **enfermedad** mediante una placa RX.
- Es por ello que esta práctica observacional tiene como **objetivo** realizar la **reconstrucción** de **imágenes RX** mediante **autoencoders convolucionales** para facilitar el **desarrollo** de un **sistema se ayuda al diagnóstico clínico** en un paso posterior.

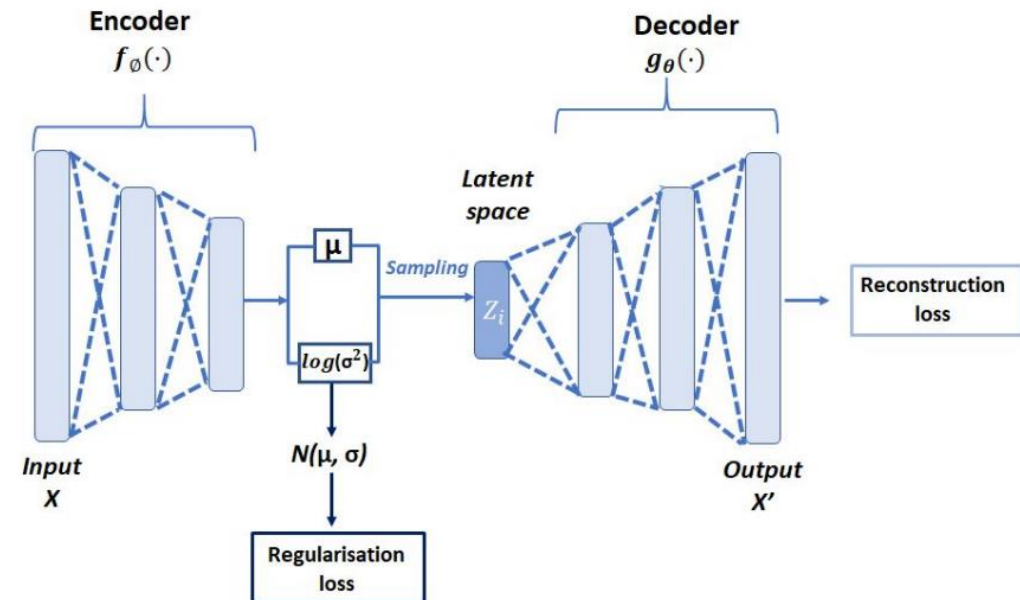


Contexto

- Para ello, vamos a llevar a cabo el desarrollo de un autoencoder convolucional en sus dos versiones: la versión básica y la versión variacional



Autoencoder clásico



Autoencoder variacional

Desarrollos a llevar a cabo

- Una **clase data generator** que nos ofrezca total versatilidad a la hora de realizar la carga de imágenes
- La **clase convolutional autoencoder** en la que desarrollaremos ambas versiones del autoencoder convolucional
- Un **fichero principal** en el que implementaremos la **reconstrucción de imágenes RX** provenientes de un dataset covid19
- Se hará **uso** de una **pequeña librería** propia de **utilidades** para el desarrollo de soluciones basadas en aprendizaje profundo



02

Clase DataGenerator

Práctica Observacional: Desarrollo de un autoencoder convolucional para la reconstrucción de placas RX

my_data_generator.py

- Desarrollo de la clase DataGenerator en un fichero denominado **my_data_generator.py**. A continuación se listan las funcionalidades a desarrollar en dicha clase:

1. **def __init__**
2. **def __len__**
3. **def __getitem__**
4. **def on_epoch_end**
5. **def get_sample**
6. **def norm**



03

Clase ConvAutoencoder

Práctica Observacional: Desarrollo de un autoencoder convolucional para la reconstrucción de placas RX

convolutional_autoencoder.py

- Desarrollo de la clase ConvAutoencoder en un fichero denominado **convolutional_autoencoder.py**. A continuación se listan las funcionalidades a desarrollar en dicha clase:

1. def __init__
2. def load_model
3. def build
4. def sampling
5. def compile
6. def train_with_generator
7. def save_model



04

Main training_CAE

Práctica Observacional: Desarrollo de un autoencoder convolucional para la reconstrucción de placas RX

training_CAE.ipynb

- Desarrollo de un fichero main training_CAE.ipynb. A continuación se listan las funcionalidades a desarrollar en dicha clase:
 1. Carga de datos
 2. Instanciar modelo
 3. Compilación y entrenamiento
 4. Visualización e interpretación entrenamiento
 5. Evaluación en test



viu

Universidad
Internacional
de Valencia