An introduction to Variational Autoencoders

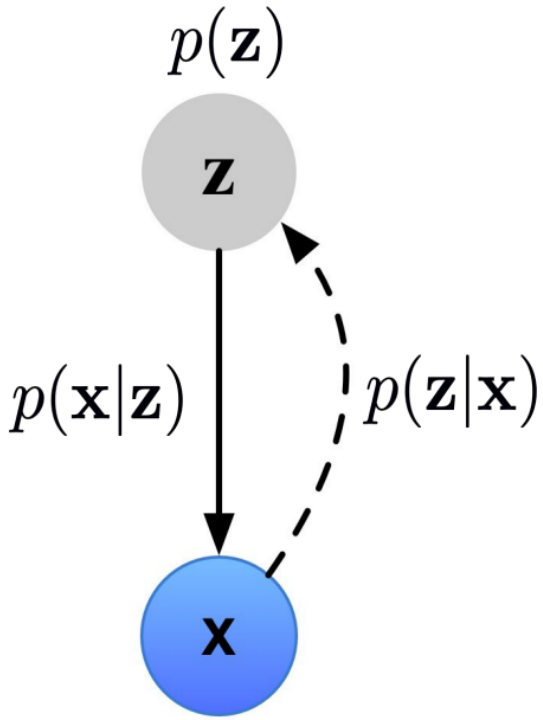
G. Catalani, J. Morlier

2024/2025 ISAE-Supaero

Latent Variable Model

A latent variable model defines a joint distribution over observed variables $\mathbf{x}$ and unobserved (latent) variables $\mathbf{z}$

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$



Latent Variable Model

Latent Variable Model

A latent variable model defines a joint distribution over observed variables $\mathbf{x}$ and unobserved (latent) variables $\mathbf{z}$

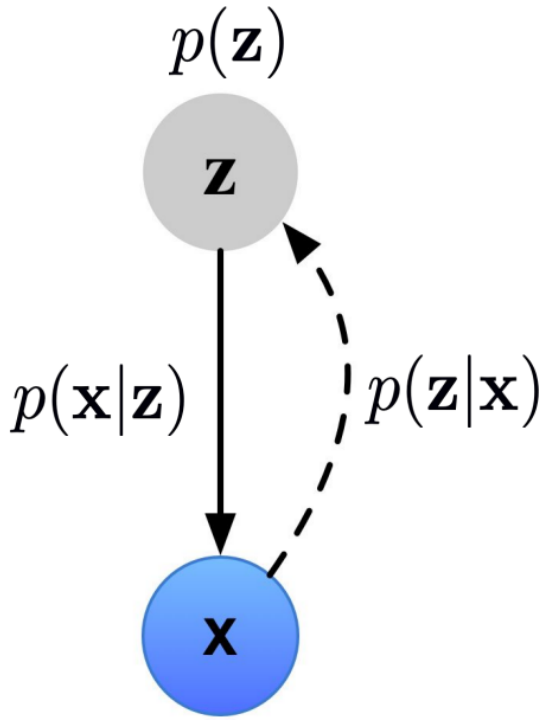
$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$

- **Marginal Likelihood** is the probability of observing the data that we want to maximize.

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

- For a dataset of N observations:

$$\log p(X) = \sum_{i=1}^{N} \log p(\mathbf{x}^{(i)}) = \sum_{i=1}^{N} \log \int p(\mathbf{x}^{(i)}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$



Latent Variable Model

Latent Variable Model

A latent variable model defines a joint distribution over observed variables $\mathbf{x}$ and unobserved (latent) variables $\mathbf{z}$

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$

- **Marginal Likelihood** is the probability of observing the data that we want to maximize.

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

- For a dataset of N observations:

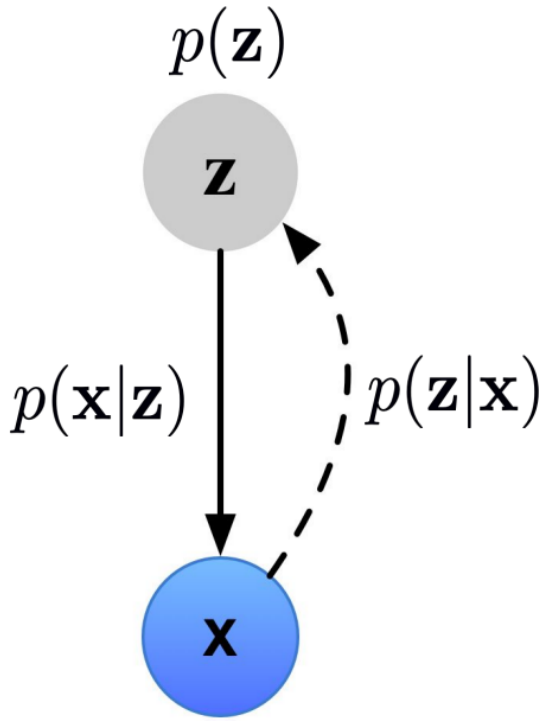
$$\log p(X) = \sum_{i=1}^{N} \log p(\mathbf{x}^{(i)}) = \sum_{i=1}^{N} \log \int p(\mathbf{x}^{(i)}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

Typically the observations comes from a high dimensional distribution (ex. images, text,...) and the latent variables are assumed to be lower dimensional and from a tractable distribution for applications like:

- Dimensionality Reduction.
- Generative modeling (by sampling from the latent distribution):

$$\mathbf{z} \sim p(\mathbf{x})$$

$$\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$



Latent Variable Model

Latent Variable Model - Challenge

Challenge:

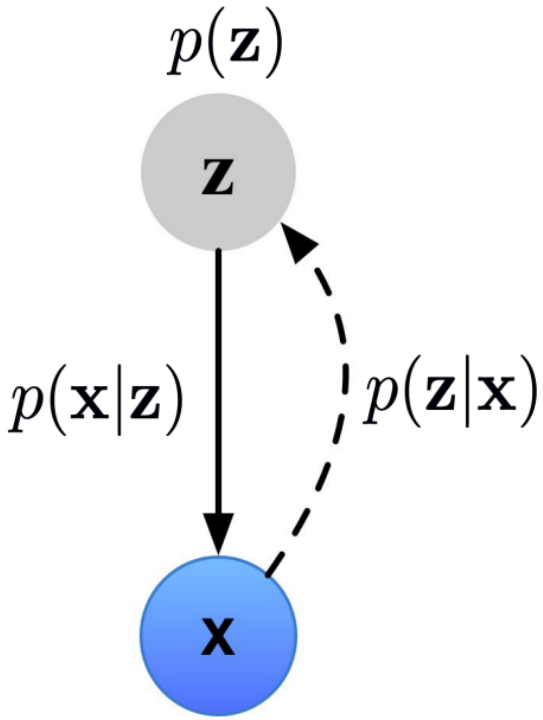
- Maximizing directly the marginal likelihood requires estimating the posterior distribution $p(\mathbf{z}|\mathbf{x})$

$$\nabla_\theta \log p_\theta(\mathbf{x}) = \frac{\nabla_\theta p_\theta(\mathbf{x})}{p_\theta(\mathbf{x})} = \frac{\int \nabla_\theta p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z}}{p_\theta(\mathbf{x})} \quad (1)$$

$$= \frac{\int p_\theta(\mathbf{x}, \mathbf{z}) \nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z}}{p_\theta(\mathbf{x})} \quad (2)$$

$$= \int p_\theta(\mathbf{z}|\mathbf{x}) \nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z} \quad (3)$$

Using the identity: $\nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{z}) = \frac{\nabla_\theta p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{x}, \mathbf{z})}$



Latent Variable Model

Latent Variable Model - Challenge

Challenge:

- Maximizing directly the marginal likelihood requires estimating the posterior distribution $p(\mathbf{z}|\mathbf{x})$

$$\nabla_\theta \log p_\theta(\mathbf{x}) = \frac{\nabla_\theta p_\theta(\mathbf{x})}{p_\theta(\mathbf{x})} = \frac{\int \nabla_\theta p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z}}{p_\theta(\mathbf{x})} \tag{4}$$

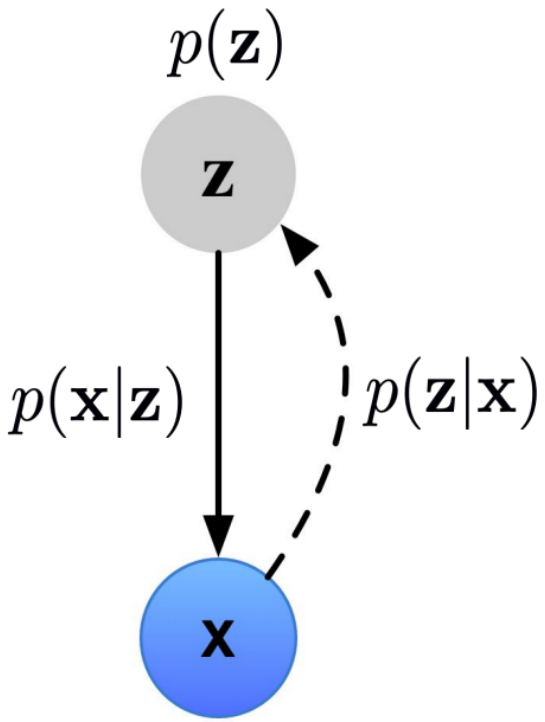
$$= \frac{\int p_\theta(\mathbf{x}, \mathbf{z}) \nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z}}{p_\theta(\mathbf{x})} \tag{5}$$

$$= \int p_\theta(\mathbf{z}|\mathbf{x}) \nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z} \tag{6}$$

Using the identity: $\nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{z}) = \frac{\nabla_\theta p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{x}, \mathbf{z})}$

- The process of estimating the posterior is called inference and it is hard in general as it requires the computation of the inegral of the joint distribution over an infinite number of configurations of $\mathbf{z}$

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})} = \frac{p(\mathbf{x}, \mathbf{z})}{\int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}} \tag{7}$$



Latent Variable Model

Variational Inference

Variational inference (VI) turns the task of finding the posterior distribution into an optimization problem: $q_\phi(\mathbf{z}|\mathbf{x}) \approx p_\theta(\mathbf{z}|\mathbf{x})$

- The idea is to approximate the exact posterior with an approximate posterior q_ϕ
- It should be easy to sample from the approx. posterior and to optimize wrt the parameters ϕ
- What is the **optimization objective**? It should be related to the original marginal likelihood

Variational Inference: the Evidence Lower Bound

For any $q_\phi(\mathbf{z})$ it holds the following (using Jensen Inequality):

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z} \tag{8}$$

$$= \log \int q_\phi(\mathbf{z}) \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z})} d\mathbf{z} \tag{9}$$

$$\geq \int q_\phi(\mathbf{z}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z})} d\mathbf{z} \tag{10}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z})} \right] \tag{11}$$

Using the variational posterior $q_\phi(\mathbf{z})$, we obtain the **ELBO**:

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right]$$

Rewriting the ELBO

The ELBO can be rewritten as the difference of two tractable terms:

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x})p_\theta(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \tag{12}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x})] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \tag{13}$$

$$= \log p_\theta(\mathbf{x}) - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})) \tag{14}$$

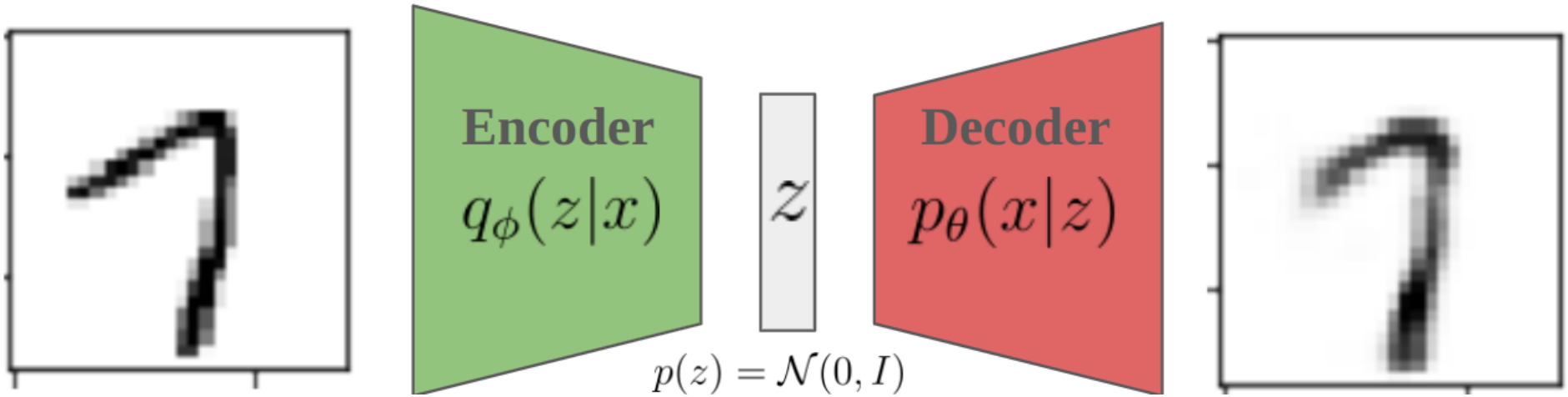
The **Kullback–Leibler divergence** between two distributions is defined as:

$$\mathrm{KL}(q(\mathbf{z})||p(\mathbf{z})) = \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{q(\mathbf{z})}{p(\mathbf{z})}\right]$$

Maximizing the ELBO wrt to the variational parameters is equivalent to fitting the variational posterior to the true posterior

Variational Auto-Encoders

A Variational Autoencoder parametrizes the approximate posterior with an **encoder** neural network and the likelihood with a **decoder** neural network



VAE Architecture: Input → Encoder → Latent Space → Decoder → Output

The following formulation is used to train a VAE by maximizing the ELBO:

$$\mathcal{L}_{\theta,\phi} = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

For a dataset of N observations the loss becomes:

$$\sum_{i=1}^{N} \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}[\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p(\mathbf{z}))$$

Variational Auto-Encoders - Implementation

In practice:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu_\phi(\mathbf{x}), \text{diag}(\sigma^2_\phi(\mathbf{x})))$$

$$\text{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) = -\frac{1}{2}\sum_{j=1}^{d}(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2)$$

$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[-\frac{1}{2\sigma^2}||\mathbf{x} - \mu_\theta(\mathbf{z})||^2 - \frac{D}{2}\log(2\pi\sigma^2)\right]$$

It is not possible to sample directly from $q_\phi(\mathbf{z}|\mathbf{x})$, so we use the **reparametrization trick**:

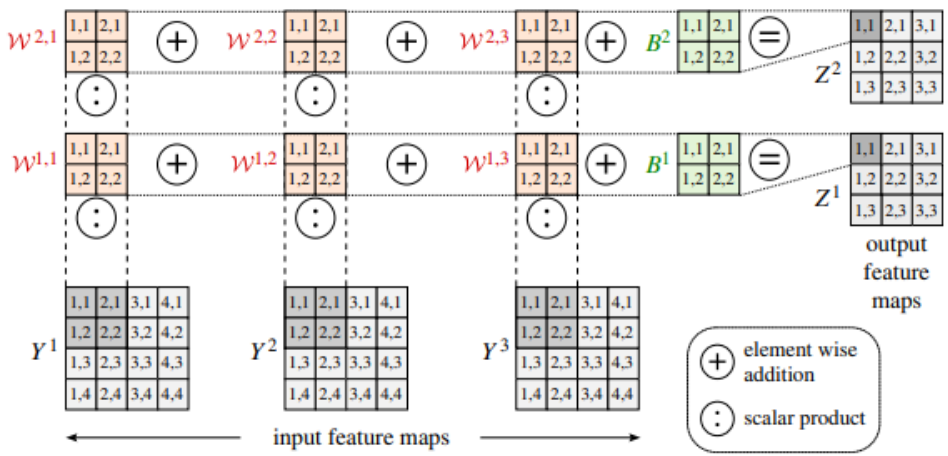
$$\mathbf{z} = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$$

Full Derivation: `https://stats.stackexchange.com/questions/318748/deriving-the-kl-divergence-loss-for-vaes/370048#370048`

Convolution Operation

$$Z_{i,j}^{m} = \sigma\left(\sum_{l=1}^{C_{in}}\sum_{\alpha=1}^{k_x}\sum_{\beta=1}^{k_y} W_{\alpha,\beta}^{m,l} Y_{i+\alpha,j+\beta}^{c} + B_{\alpha,\beta}^{m}\right)$$

for $1 \leq m \leq C_{out}$, $1 \leq i \leq W_{out}$, $1 \leq j \leq H_{out}$

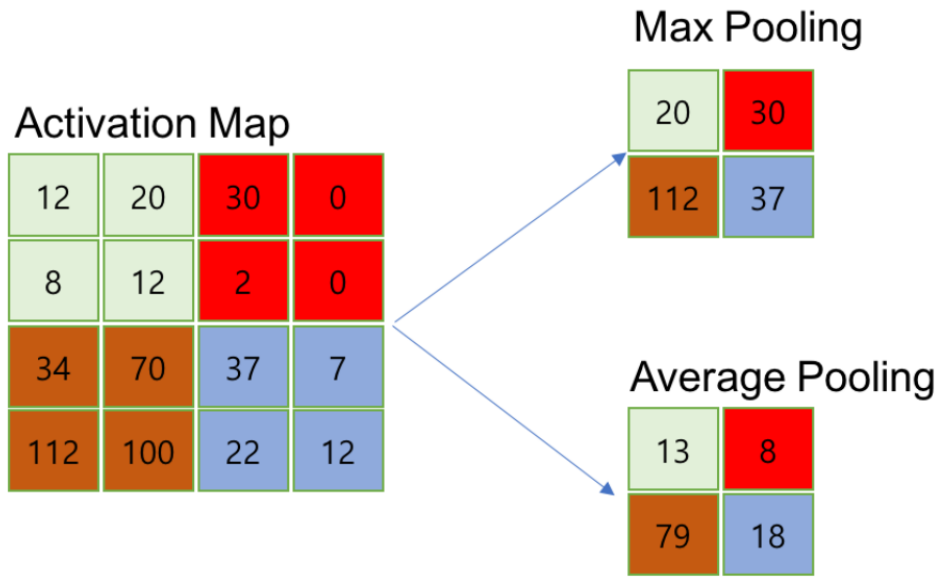


Convolution Operation with Filter Kernel

A CNN is a sequence of convolutional layers and non linear activation functions
The convolution operation applies filters (kernels) across the input to produce feature maps, with each filter detecting specific patterns or features in the data.

Pooling in CNNs

Pooling is used in CNN to reduce the feature map dimension.
Max Pooling: Takes the maximum value in each pooling window
Average Pooling: Takes the average value in each pooling window
A Convolutional **Encoder** Network can be designed using Convolution and Pooling operators (and flattening).

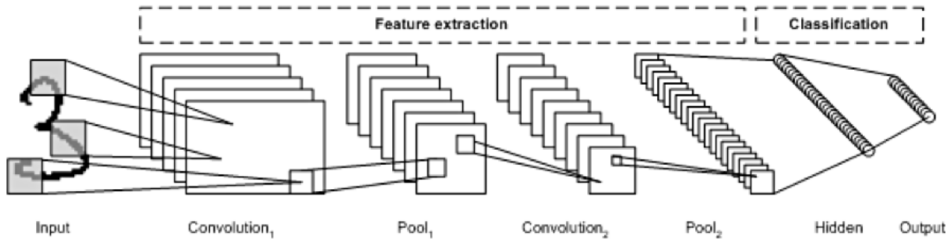


Max Pooling vs Average Pooling

Typical CNN architecture progression:

Input → CONV → ReLU → POOL → CONV → ReLU → POOL → ... → FC

The combination of convolution and pooling layers allows the network to learn hierarchical features while reducing spatial dimensions.



Convolutional Encoder Network

Tutorial in PyTorch: VAE for Topology Optimization

In this tutorial you will learn:

- How to implement a simple VAE in PyTorch to process images and reconstruct different topologies.
- How to implement a Convolutional Neural Net in PyTorch in the form of a VAE.
- How to use VAE for generative modeling and data exploration.

You can find the Notebooks in the Github repository: `https://github.com/jomorlier/IA4SM`

References


Kingma, D. P., & Welling, M. (2013). *Auto-encoding variational bayes*. arXiv preprint arXiv:1312.6114.


DeepMind UCL Course (2020). *Deep Learning Lecture 11: Variational Autoencoders*. https://storage.googleapis.com/deepmind-media/UCLxDeepMind_2020/L11%20-%20UCLxDeepMind%20DL2020.pdf


Stanford CS231n (2021). *Convolutional Neural Networks for Visual Recognition*. https://cs231n.stanford.edu/2021/slides/2021/lecture_5.pdf


Cross Validated Stack Exchange (2018). *Deriving the KL divergence loss for VAEs*. https://stats.stackexchange.com/questions/318748/deriving-the-kl-divergence-loss-for-vaes/370048#370048