Artificial Models for Music Creativity

Lesson 3 - Introduction to Audio Deep Learning 1/2

Alessandro Anatrini - 13.12.2024

WHAT IS AUDIO DEEP LEARNING FOR?

1. Audio classification
2. Audio separation and segmentation
3. Music genre classification and tagging
4. Unconditioned and conditioned audio generation
5. Music generation and transcription (symbolic)
6. Voice recognition (NLP)
7. Speech2Text and Text2Speech
8. Automatic device programming

AUDIO GENERATION

The evolution of music is tightly connected to technological advances

New synthesis method

Innovative control systems

Machine learning

Proof of concept for raw waveform generation:

- WaveNet
- SampleRNN

AUDIO GENERATION ISSUES

Hard to integrate inside creative frameworks

Quality: 16kHz - 8 bit mono signals

Temporal coherence: small processing window (300ms - 1.5s)

Computational complexity: long training time, data greedy, long sequential generation mechanism

Interaction: lack of control over generation process

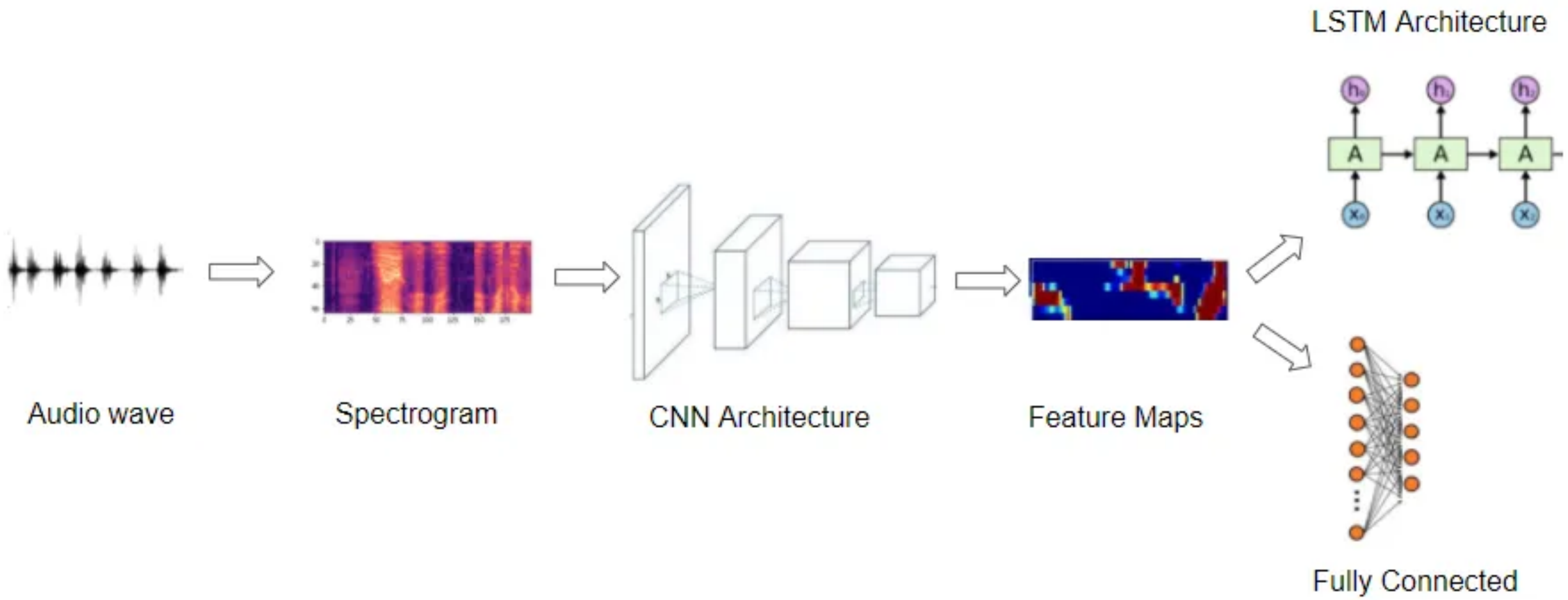
AUDIO GENERATION ISSUES

High dimensionality

Complex phase structure: differences in waveform might not be perceptually relevant, hard to define objectives

Spectral representations: easier to understand, lossy reconstruction

Audio Deep Learning Pipeline



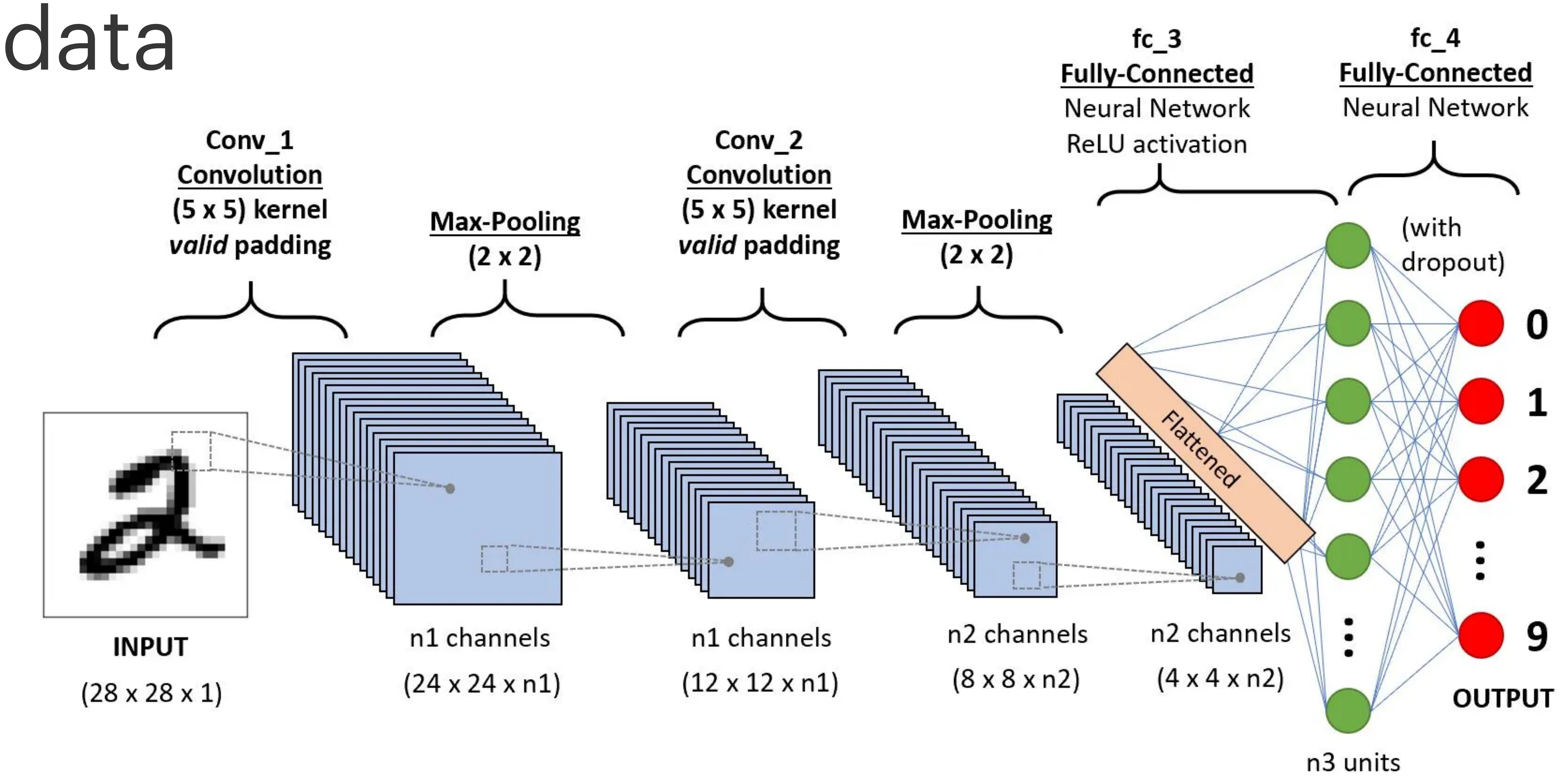
Typical pipeline used by audio deep learning models

AUDIO DEEP LEARNING PIPELINE

1. Start with raw audio data in the form of a wave file
2. Convert the audio data into its corresponding spectrogram
3. Augment dataset with simple DSP techniques: e.g. random crop
4. Pass the data to a CNN architecture to process them and extract feature maps that are encoded representation of the spectrogram
5. Generate output predictions from the encoded representation, depending on the problem we want to solve

Convolutional Neural Networks (CNN)

Convolution: passes a filter (kernel) over the input data (such as to produce a feature map. This process allows the network to learn local patterns in the data



A CNN sequence to classify handwritten digits

Convolutional Neural Networks (CNN)

Convolutional layers: apply a set of filters to the input data, each filter is able to detect a specific type of feature, such as an edge in an image or a particular frequency in an audio signal

Pooling layers: reduce the spatial size (i.e., the width and height) of the input, which helps to make the network invariant to small translations and reduces computational complexity

Fully connected layers: perform high-level reasoning on the features extracted by the previous layers, The last FCL uses a softmax activation function to output the network's prediction

CNN TRAINING

Training a CNN involves feeding it input data (such as audio spectrograms), and adjusting the network's weights to minimize the difference between the network's predictions and the actual values. This is typically done using a method called backpropagation, along with an optimization algorithm such as stochastic gradient descent

CNN'S AUDIO TASKS

Audio classification: i.e. speech, music, environmental sounds

Speech recognition: i.e. spoken language to written text

MIR: genre classification, mood detection, instrument recognition

Sound event detection: CNNs can be used to detect and classify different sound events in an audio signal

CNN’S PROS AND CONS

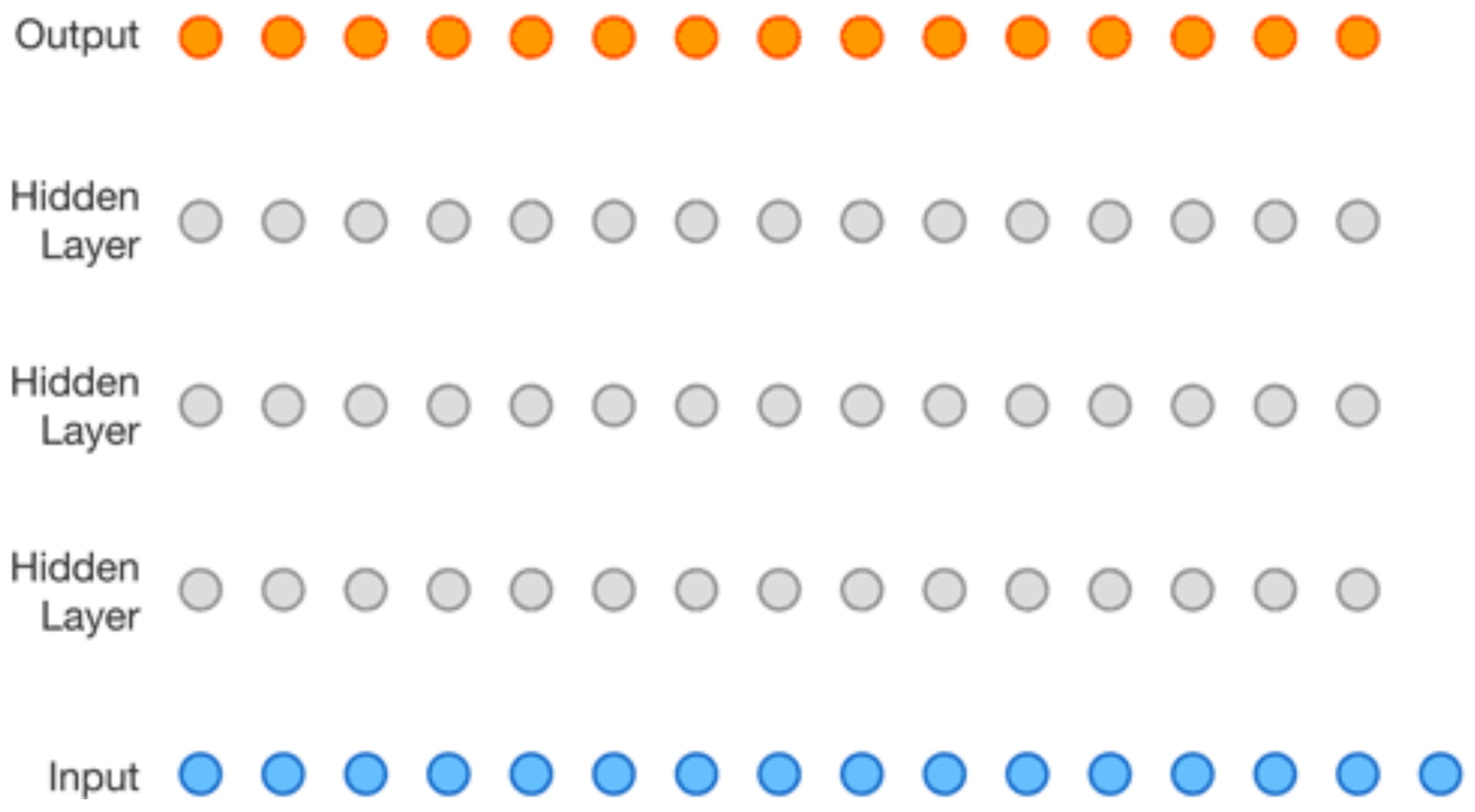
Pros: can model local patterns and hierarchies, making them well-suited to tasks involving grid-like data

Cons: they may struggle with long-term dependencies or global patterns, which are often important in audio signals

Autoregressive Models (ARM)

Regressive neural networks, also known as autoregressive models, are designed to predict future values based on past values in a sequence, making them particularly suitable for tasks involving time-series data

Like an RNN, an autoregressive model's output depends from the previous output; unlike RNN the previous output are not provided via hidden state: it is given as just another input to the model



Autoregressive Models (ARM)

RNNs typically consist of multiple layers of neurons, with each layer feeding its output to the next layer, the last layer produces the output of the network

The architecture can vary widely depending on the specific task at hand: i.e. in the case of audio synthesis, architectures like WaveNet have been used, which are fully probabilistic, autoregressive, and causal, thus each sample generated depends only on the previously observed samples

ARM's AUDIO TASKS

Audio synthesis: i.e. WaveNet for raw audio generation

Audio restoration: i.e. through diffusion models

Symbolic generation: predict the next note in a sequence based on the previous one

MIR: genre classification

Speech recognition: phonemes identification

ARM'S PROS AND CONS

Pros: can model complex temporal dependencies, making them well-suited to tasks involving time-series data

Cons: can be computationally intensive to train, particularly for long sequences, and they may require large amounts of training data to perform well

Autoencoders (AE)

An autoencoder is a type of neural network that is trained to copy its input to its output, it has two main parts:

encoder: this part of the network compresses the input into a latent-space representation

decoder: this part reconstructs the input from the latent space representation

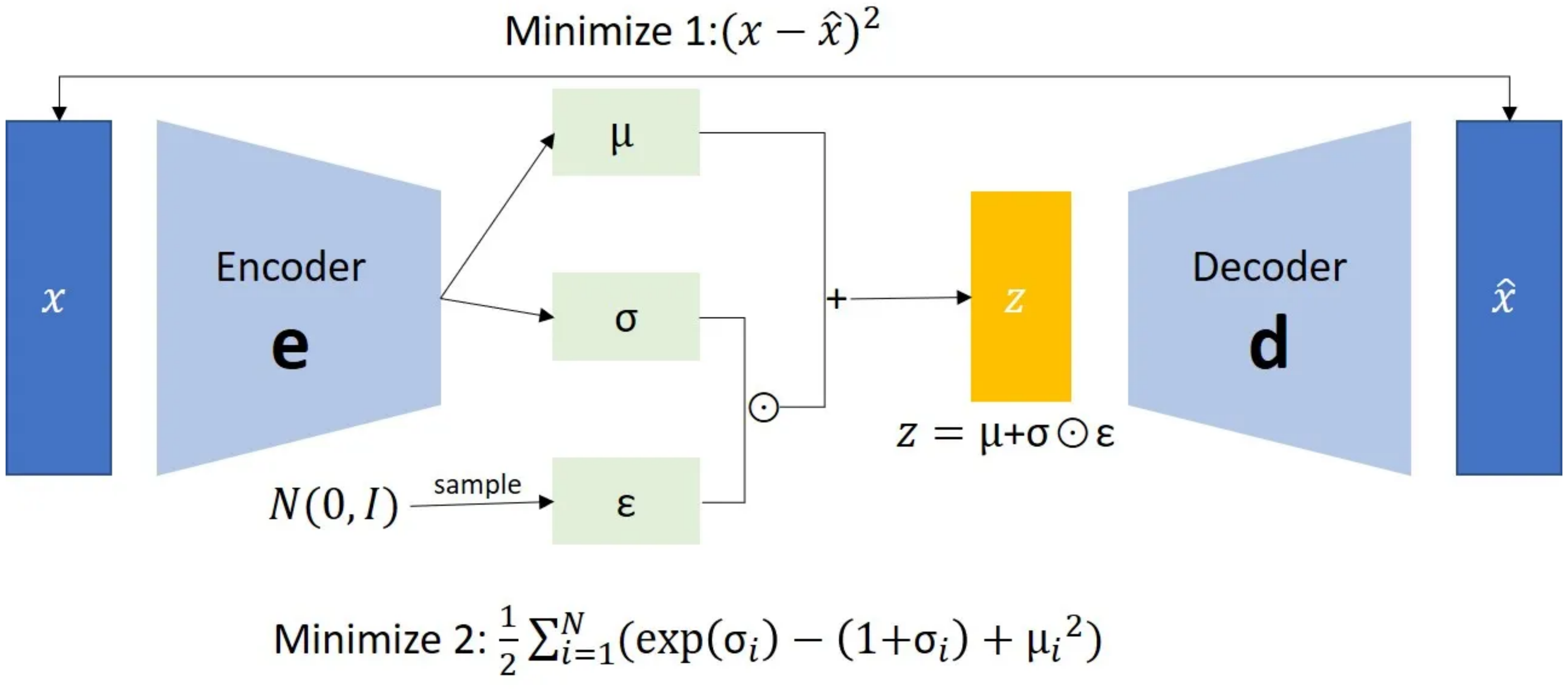
Variational Autoencoders (VAE)

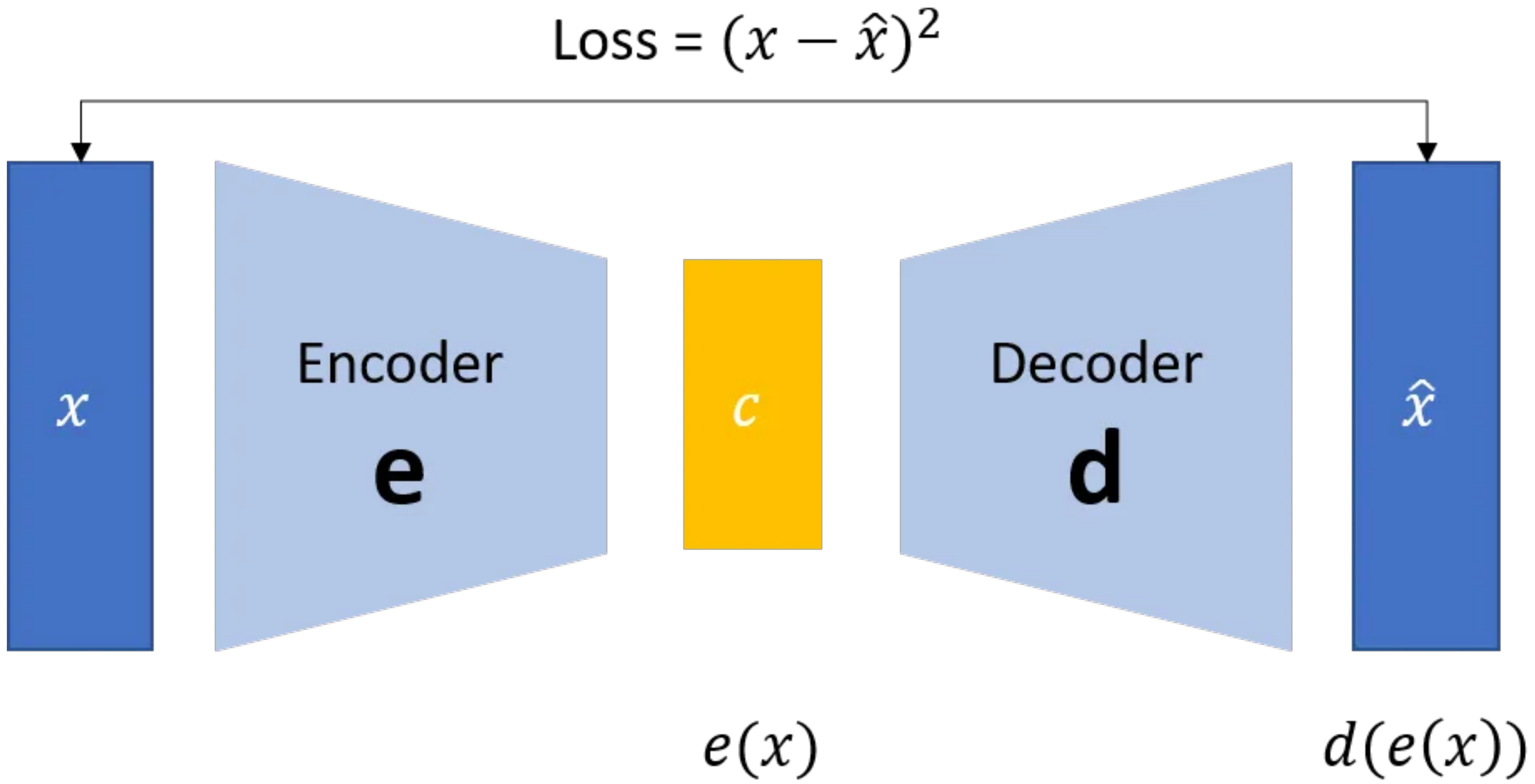
VAEs are a type of autoencoder with added constraints on the encoded representations being learned; more specifically, they are designed to force the encoded data to approximate a standard Gaussian distribution

This constraint helps to avoid overfitting and ensures that similar inputs have similar encoding

VAE'S AUDIO TASKS

In the domain of audio deep learning audio VAE are mainly used for audio synthesis


Minimize 1:$(x-\hat{x})^2$
x
Encoder
e
μ
σ
ε
$N(0,I)$
sample
⊙
+
z
$z=$ μ+σ⊙ε
Decoder
d
$\hat{x}$
Minimize 2: $\frac{1}{2}\sum_{i=1}^{N}(\exp(\sigma_i)-(1+\sigma_i)+\mu_i^{2})$



Above a VAE architecture

On the left an AE architecture

VAE’S PROS AND CONS

Pros: can learn efficient representations of data, making them well-suited to tasks involving high-dimensional data

Cons: they may struggle with complex data distributions or when the data has underlying factors of variation that are not apparent in the input