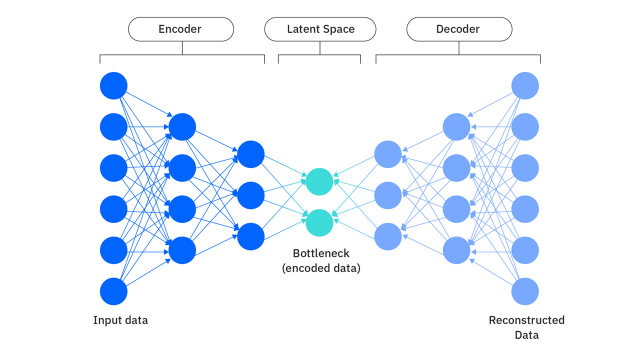
***Auto-encoder-***

* a [neural network](https://www.ibm.com/topics/neural-networks" \t "https://www.ibm.com/think/topics/_self) architecture typically used in [deep learning](https://www.ibm.com/topics/deep-learning" \t "https://www.ibm.com/think/topics/_self)
* it is part of the families of [probabilistic graphical models](https://en.wikipedia.org/wiki/Graphical_model" \o "Graphical model) and [variational Bayesian methods](https://en.wikipedia.org/wiki/Variational_Bayesian_methods" \o "Variational Bayesian methods).
* for tasks such as
* data compression,
* image denoising,
* anomaly detection
* And facial recognition.
* are [self-supervised](https://www.ibm.com/topics/self-supervised-learning" \t "https://www.ibm.com/think/topics/_self) systems.
* goal(training): to compress (or *encode*) input data through dimensionality reduction and then accurately reconstruct (or *decode*) their original input by using that compressed representation.
* function of an autoencoder is to effectively extract the data’s most salient information—its latent variables—and discard irrelevant noise.

*Uses of Auto-encoders*:-

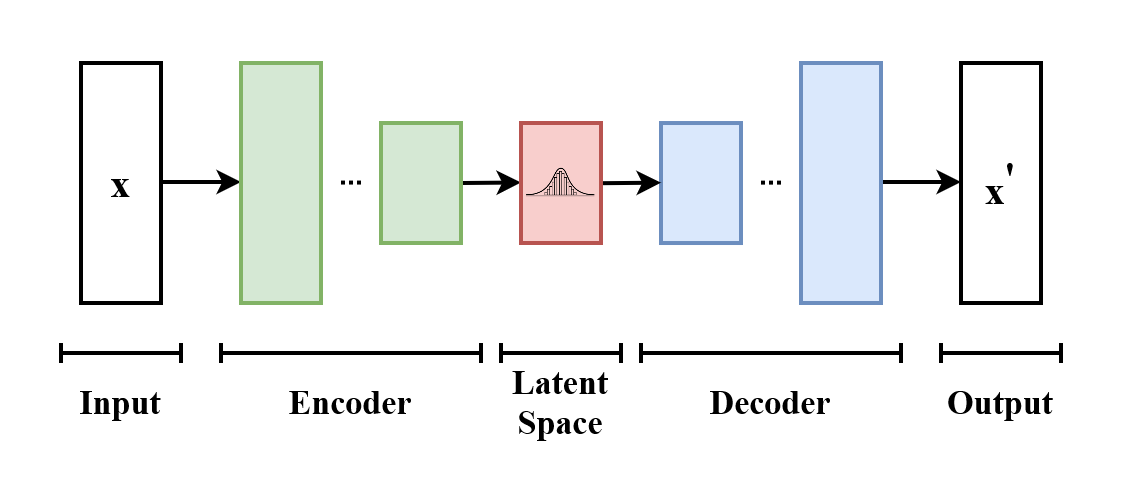
* restore corrupted audio files,
* colorize grayscale images
* or detect anomalies (such as those resulting from fraud) that would otherwise be invisible to the naked eye.



Visual depiction of the architecture of a standard autoencoder neural network.

What distinguishes different types of autoencoders from one another?

- is the specific strategy they employ to extract that information and the use cases to which their respective strategy is best suited.



* Basic representation of variationalautoencoder
* Model receives x as i/p
* Encoder compresses it into latent space
* Decoder produces x’as similar as possible to x

**Autoencoder structure**

* ****encoder**** extracts latent variables of input data *x* and outputs them in the form of a vector representing latent space *z.*

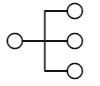


* ****bottleneck****, or ****"code,"**** is both the output layer of the encoder network and the input layer of the decoder network.

bottleneck is necessary to help ensure that the decoder cannot simply copy or memorize the input data, which would nominally satisfy its training task but prevent the autoencoder from learning.



* ****decoder**** uses that latent representation to reconstruct the original input by essentially reversing the encoder



**latent variables** are underlying variables of data that inform the way the data is distributed but are often not directly observable.

**latent space***-*  collective *latent variables* of a specific set of input data.

*In a machine learning (ML)* *context*, mathematical dimensions correspond not to the familiar spatial dimensions of the physical world, but to features of data.

***Variational auto-encoder model***

Variational autoencoders (VAEs) are [deep learning](https://www.ibm.com/topics/deep-learning" \t "https://www.ibm.com/think/topics/_self) models composed of an encoder that learns to isolate the important latent variables from training data and a decoder that then uses those latent variables to reconstruct the input data.

* VAEs are probalistics models
* autoencoder architectures encode a *discrete*, fixed representation of latent variables
* VAEs encode a *continuous* representation of that latent space.
* This enables a VAE to not only accurately reconstruct the exact original input, but also use variational inference to generate new data samples that resemble the original input data.
* In variational inference, the generative process of synthesizing new data points, this prior distribution is used to calculate the posterior distribution, [mathematical representaionp(z|x)].

**TRAINING:**

-the encoder network passes input data from the training data set through a "bottleneck" before it reaches the decoder.

-The decoder network, in turn, is then responsible for reconstructing the original input by using only the vector of latent variables.

-After each training, optimization algorithms such as [gradient descent](https://www.ibm.com/topics/gradient-descent" \t "https://www.ibm.com/think/topics/_self) are used to adjust model weights in a way that minimizes the difference between the original data input and the decoder’s output.

-Eventually, the encoder learns to allow through the information most conducive to accurate reconstruction and the decoder learns to effectively reconstruct it.

**How do variational autoencoders work?**

For each latent attribute of training data, VAEs encode two different latent vectors:

1. a vector of means, “μ,”
2. and a vector of standard deviations, “σ.”

In essence, these two vectors represent the range of possibilities for each latent variable and the expected variance within each range of possibilities.

By randomly sampling from within this range of encoded possibilities, VAEs can synthesize new data samples that, while unique and original unto themselves, resemble the original training data.

Though relatively intuitive in principle, this methodology requires further adaptations to standard autoencoder methodology to be put into practice.

**Concepts**:-

1. *Reconstruction loss/reconstruction erorr/primary loss function in training*

* measures the difference between the original input data and the reconstructed version of that data
* Multiple algorithms, including cross-entropy loss or mean-squared error (MSE), can be used as the reconstruction loss function.
* Training the VAE:- During training, the VAE optimizes this total loss using backpropagation. The encoder and decoder parameters are updated to minimize the reconstruction loss while also ensuring the latent space distribution remains close to the prior.
* Reconstruction loss alone is sufficient to optimize most autoencoders, whose sole goal is a learning compressed representation of input data that’s conducive to accurate reconstruction.

1. *Evidence lower bound (ELBO)*

* Calculating the posterior distribution is intractable
* Implying it would take a theoretically infinite amount of time to compute directly.
* ELBO acts as the objective function during the training of VAEs
* By maximizing the ELBO, we ensure that model learns to reconstruct data well while also maintaining a latent space that apt.prior distribution
* ELBO consists of two components: the reconstruction term (which encourages accurate data reconstruction) and the KL divergence term (which regularizes the learned latent distribution)

1. *Kullback-Leibler (KL) divergence/KL divergence*

* quantifies the difference between two distributions P (the true distribution) and Q (the approximating distribution).
* To generate images, the decoder samples from the latent space
* For this, the latent space must exhibit two types of regularity:

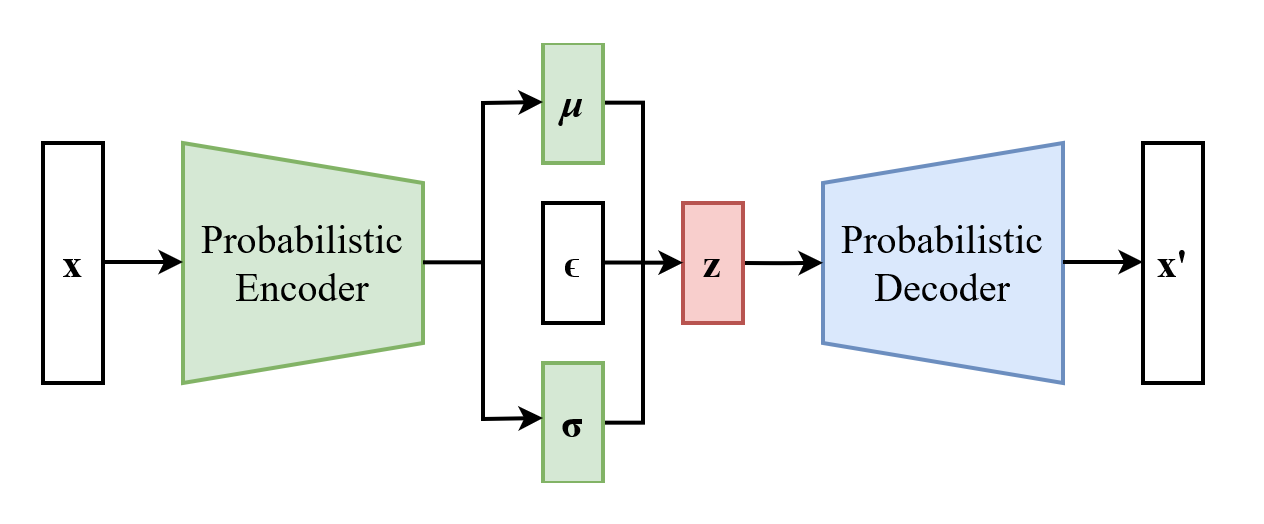
1. Continuity: Nearby points in latent space should yield similar content when decoded.
2. Completeness: Any point sampled from the latent space should yield meaningful content when decoded.

* Uses Guassian Distribution to implement continuity and completeness
* But minimizing only reconstruction loss doesn't incentivize the model to organize the latent space in any particular way, because the “in-between” space is not relevant to the accurate reconstruction of the original data points. This is where the KL divergence regularization term comes into play.



1. *The reparameterization trick*

* It allows the model to learn the parameters of the variational distribution while still being able to backpropagate through the stochastic sampling process.
* Instead of sampling z directly from q(z∣x), we express z as a deterministic function of μ and σ plus some noise: z=μ+σ⊙ϵ Where:
* ϵ\epsilone is a random variable sampled from a standard normal distribution
* ⊙ denotes element-wise multiplication.
* Deterministic Transformation: By expressing z in terms of μ, σ, and ϵ, the randomness is isolated in ϵ, which is independent of the parameters of the model. This allows the model to learn μ and σ using standard gradient descent.
* Backpropagation-Friendly: Since the transformation z=μ+σ⊙ϵ is deterministic and differentiable with respect to μ and σ, gradients can be backpropagated through the encoder and decoder networks without any issues.
* Benefits:- effective learning and stability improvement



The scheme of a variational autoencoder after the reparameterization trick.

Reference:- https://www.ibm.com/think/topics/variational-autoencoder#:~:text=Like%20all%20autoencoders%2C%20variational%20autoencoders,to%20reconstruct%20the%20input%20data.