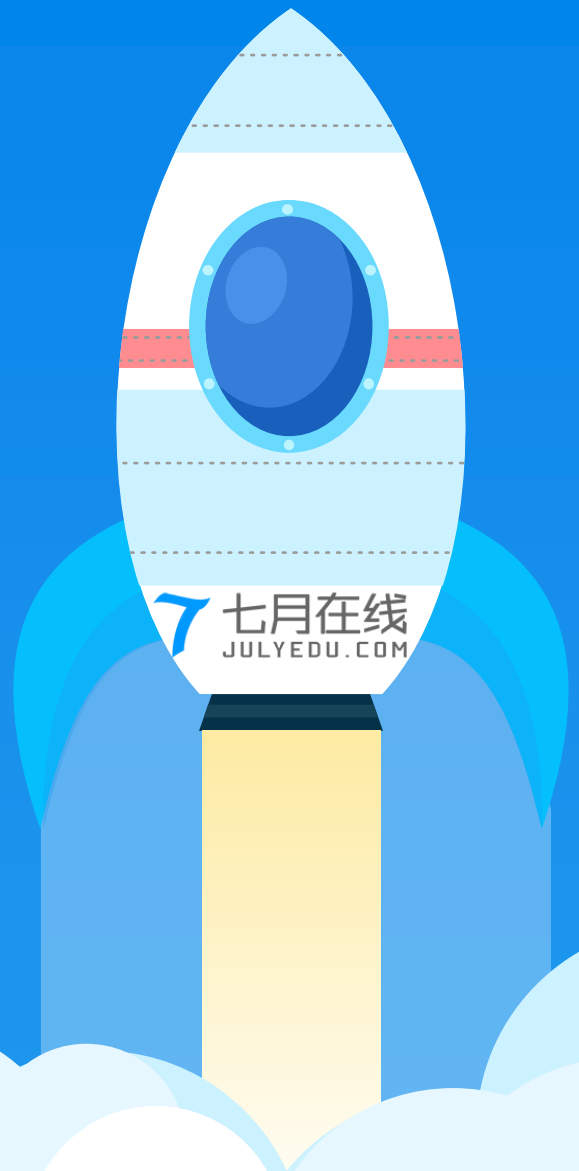
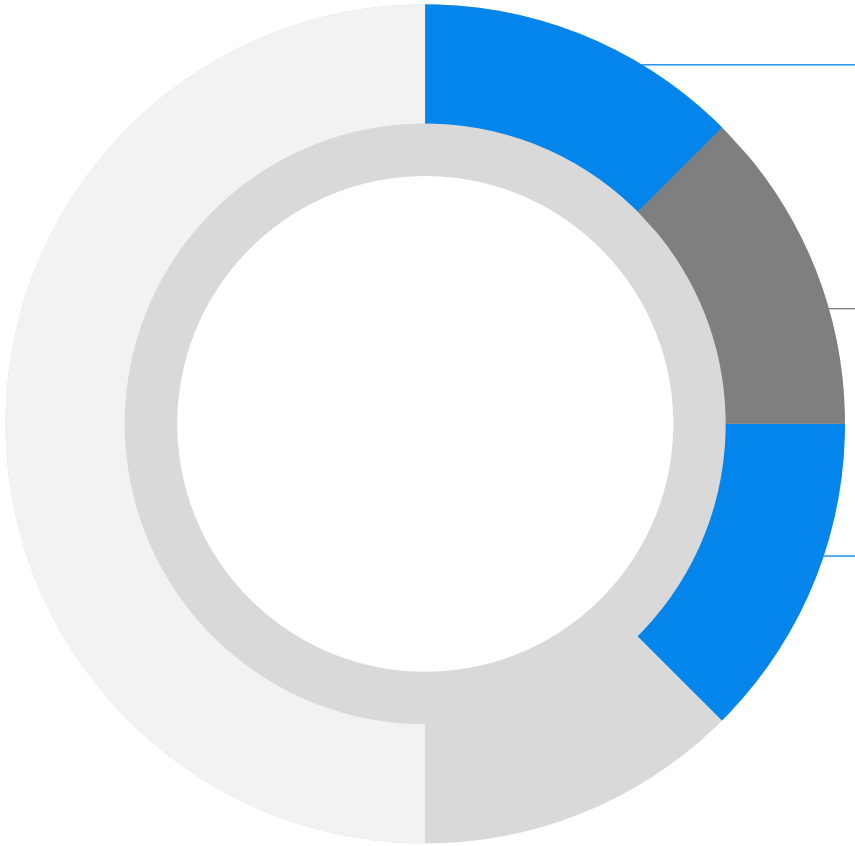
《无监督图学习》

主讲： CV 彭老师

https://www.julyedu.com/



七月在线
JULYEDU.COM



自编码器 & 变分自编码器

AE & VAE

图变分自编码器

Variational Graph Autoencoders

Project：GAE & GVAE

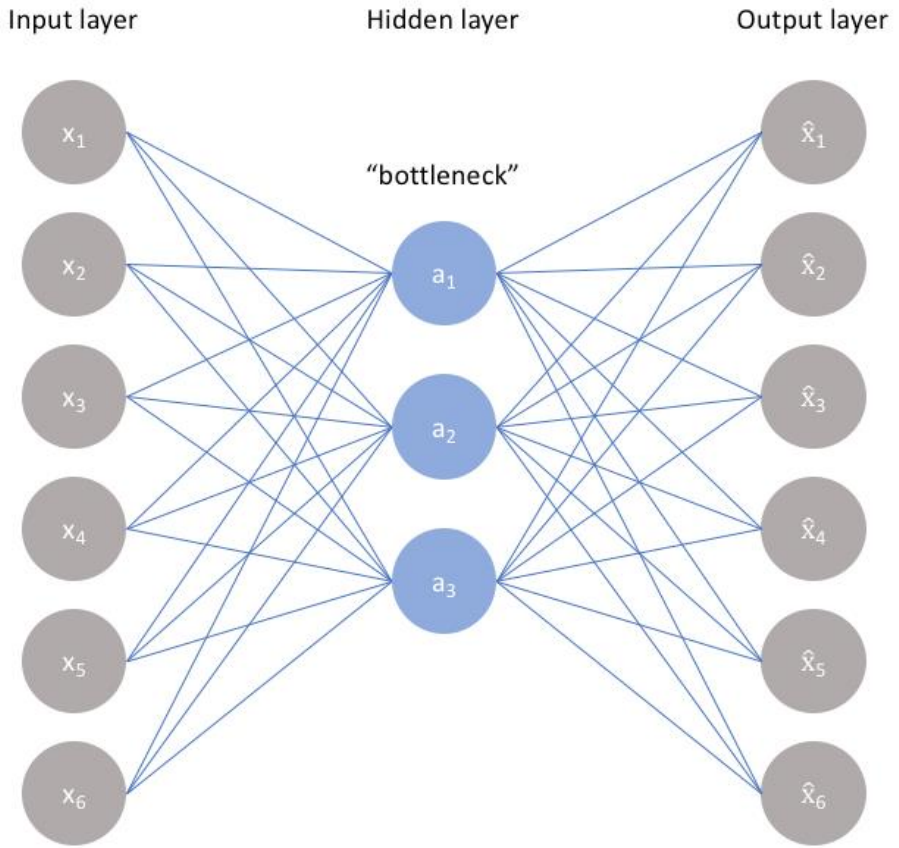
项目实战：GVAE & GVAE

/01 Auto-Encoder & Variational AE

自编码器 & 变分自编码器

Autoencoders

Let us take an unlabeled dataset and frame it as a supervised learning problem tasked with outputting $\hat{x}$, a reconstruction of the original input x.



$$\mathcal{L}(x, \hat{x})$$

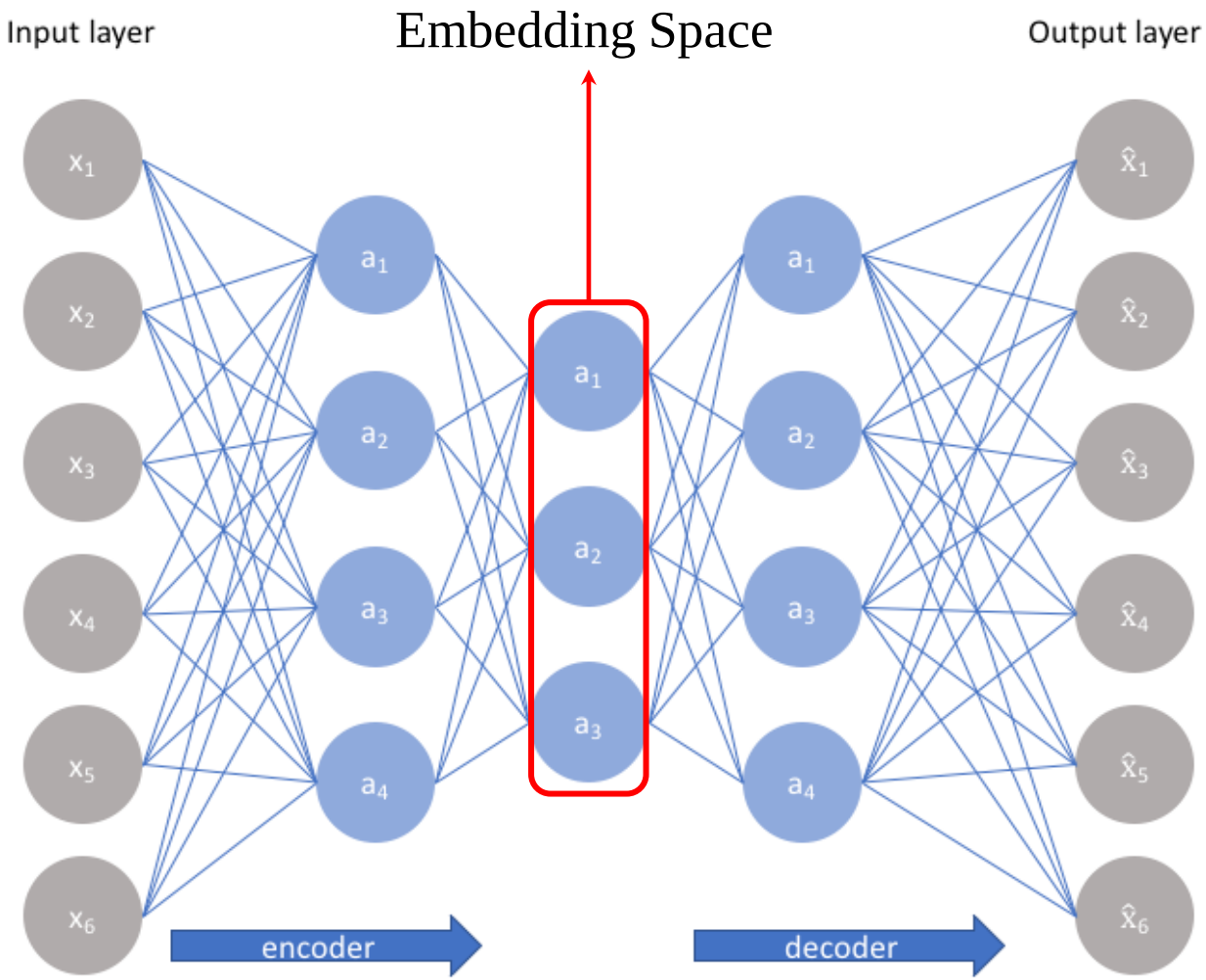
Autoencoder

Autoencoders are an unsupervised learning technique in which we leverage neural networks for the task of **representation learning**.

The ideal autoencoder model balances the following:

- Sensitive to the inputs enough to accurately build a reconstruction.
- Insensitive enough to the inputs that the model doesn't simply memorize or overfit the training data.

Autoencoder

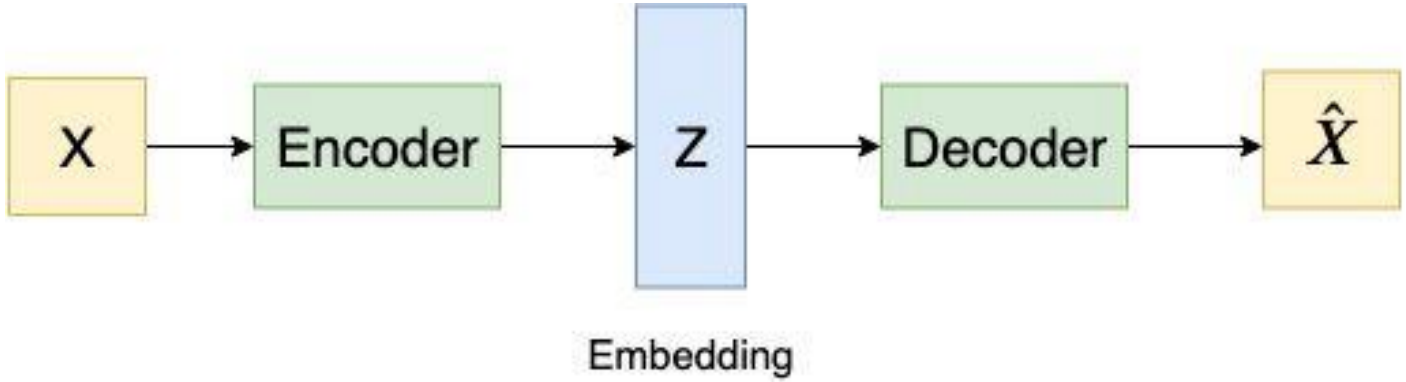


Autoencoder

Autoencoder is a neural network that contains an **encoder** and a **decoder**.

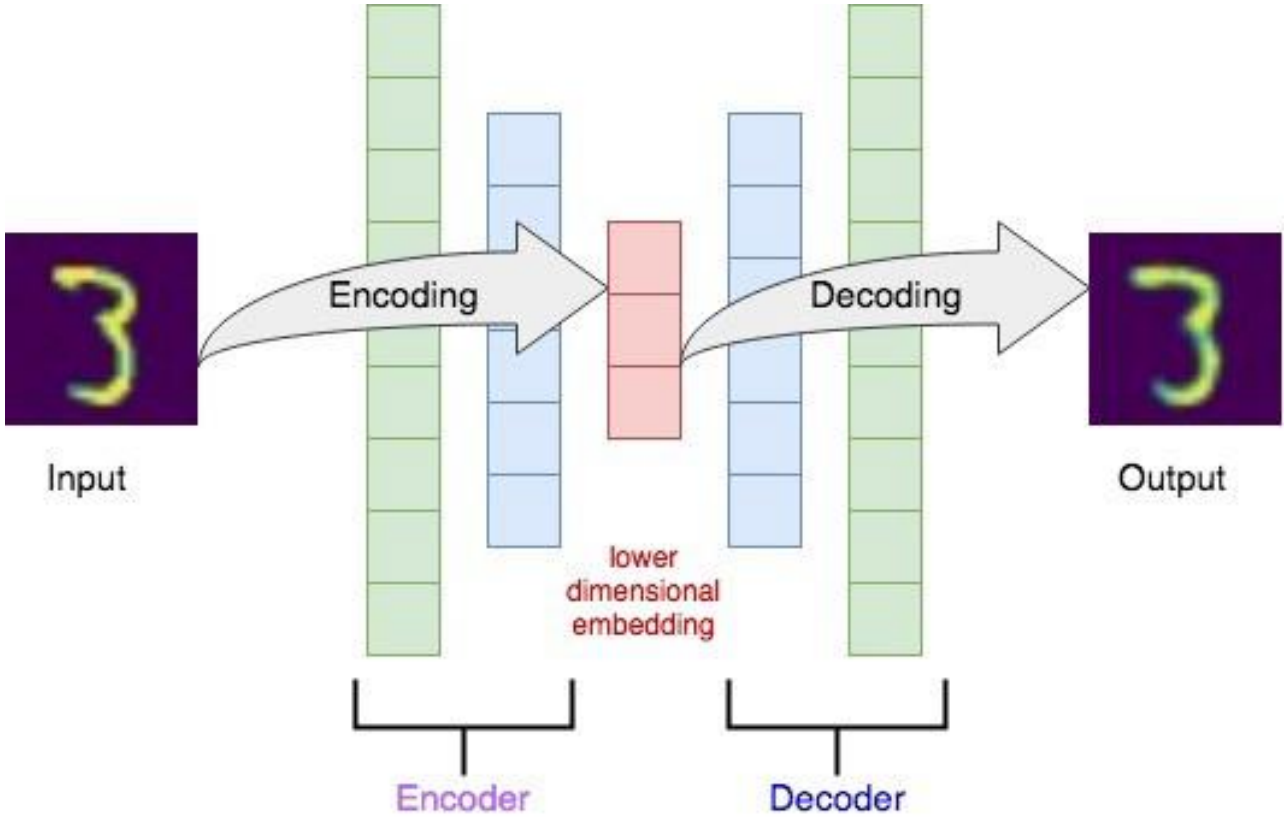
The **encoder** takes a data point X as input and converts it to a *lower-dimensional* representation Z.

The **decoder** takes the *lower-dimensional* representation Z and returns a reconstruction of the original input $\hat{x}$ that looks like the input x.



Autoencoder

The following figure shows an example of an autoencoder that takes an MNIST image as input.



Loss Function of Autoencoder

The loss function of an autoencoder measures the information lost during the reconstruction.

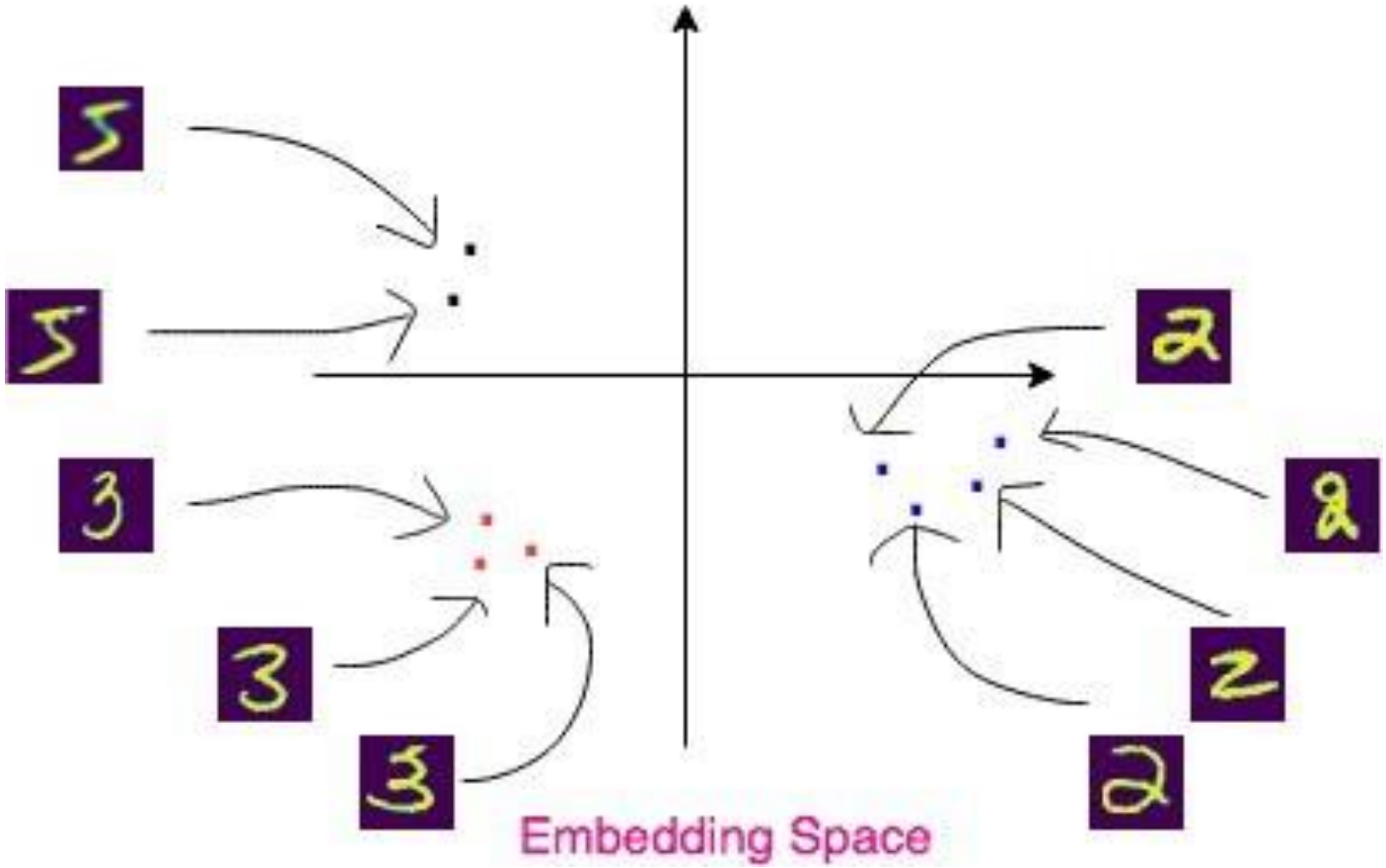
$$L(X, \hat{X}) = ||X - \hat{X}||^2$$

Embedding Space

Why do we want to convert the input to a lower-dimensional embedding?

- ✓ Storing images' low-dimensional embeddings could save storage space compared with storing their pixel intensities.
- ✓ Storing images' low-dimensional embeddings could also save computational power because the input dimensions are lower.
- ✓ We could also plot these low-dimensional embeddings to the n-d coordinate.

Embedding Space



Variational Autoencoders

Why do we need the variational autoencoders?

Traditional autoencoder could only generate images that are similar to the original inputs.

However, **VAE** could generate **new data** from the original source dataset.

Variational Autoencoders

The main idea of a VAE is that it embeds the input X to a **distribution** rather than a point.

Then a **random sample Z** is taken from the distribution rather than generated from encoder directly.

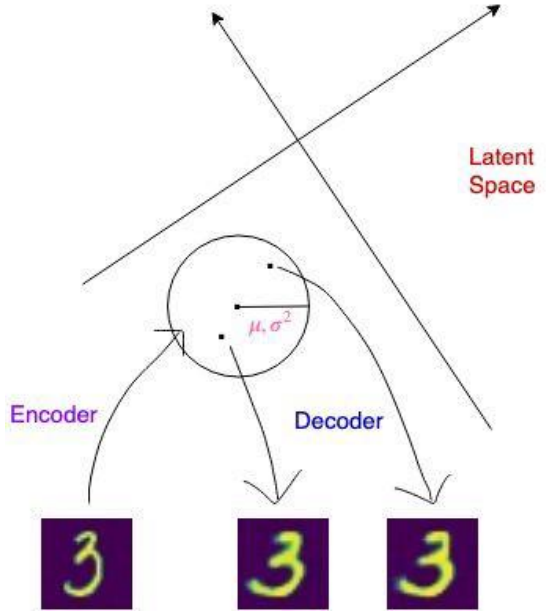
Encoder & Decoder

The encoder for a VAE is often written as $q_\varphi(z|x)$, which takes a data point X and produces a distribution (Multivariate Gaussian).

The encoder predicts the *means* and *standard deviation* of the Gaussian distribution.

The lower-dimensional embedding Z is sampled from this distribution.

The decoder is a variational approximation, $p_\theta(x|z)$, which takes an embedding Z and produces the output $\hat{x}$.



Loss Function of Variational Autoencoders

Variational lower bound: How well the network reconstructs the data;

Regularizer: How closely the output distribution $q_\varphi(z|x)$ match to $p(z)$.

$$l_i(\theta,\phi) = \underbrace{-E_{z\sim q_\phi(z|x_i)}[log_{p_\theta}(x_i|z)]}_{\text{Variational lower bound}} + \underbrace{KL(q_\phi(z|x_i)||p(z))}_{\text{Regularizer}}$$

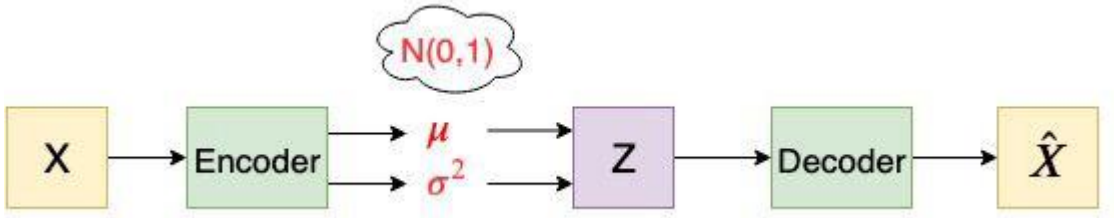
Variational Autoencoders

The encoder takes ***X*** as input and generates **μ** and **logσ²** as outputs.

Then, try to make both μ and logσ² close to 0 (σ is close to 1) → $N(0,1)$

Finally, generate the embedding **Z** from μ and σ by $z = \mu + \sigma \times \varepsilon$, where $\varepsilon \sim N(0,1)$.

Thus, with the latent variable Z, we can generate our output $\hat{X}$ through the decoder.

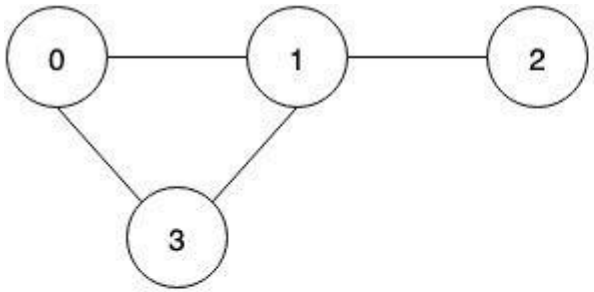


/02 Variational Graph Autoencoders

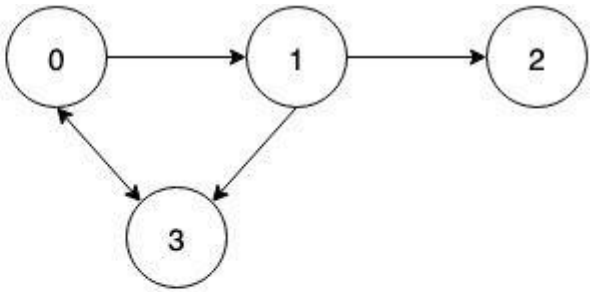
图变分自编码器

Notion: Adjacency Matrix

Adjacency matrix:
The value 1 at row i and column j means that there is an edge between vertex i and vertex j.
The value 0 at row m and column n means that there is no edge between vertex m and vertex n.



$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

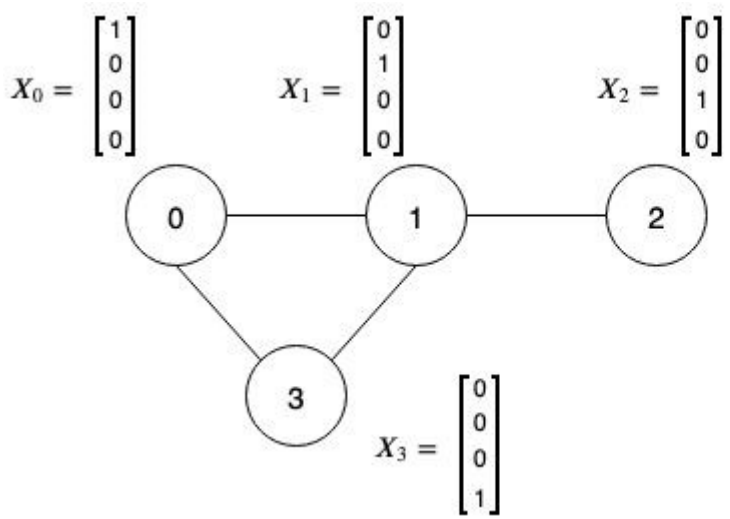


$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Notion: Feature Matrix

We use the **feature matrix X** to represent the features of each node from the input graph.

Row i of the feature matrix X represents the feature embeddings for vertex i.

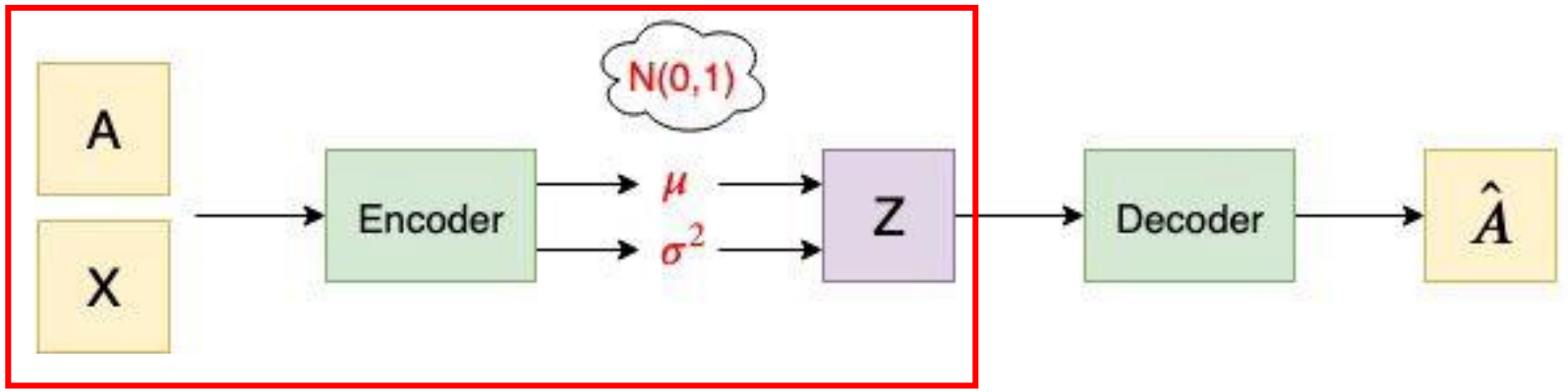


$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Encoder

The **encoder** (inference model) of VGAE consists of graph convolutional networks (**GCNs**).

It takes an adjacency matrix A and a feature matrix X as inputs and generates the latent variable Z as output.



Encoder

The first GCN layer generates a lower-dimensional **feature matrix**.

$$\bar{X} = GCN(X, A) = ReLU(\tilde{A}XW_0)$$

$$\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$$

Encoder

The second GCN layer generates **μ** and **logσ²**.

$$\mu = GCN_{\mu}(X, A) = \tilde{A}\bar{X}W_1$$

$$log\sigma^2 = GCN_{\sigma}(X, A) = \tilde{A}\bar{X}W_1$$

Now if we combine the math of two-layer GCN together, we get the follows to generates μ and logσ².

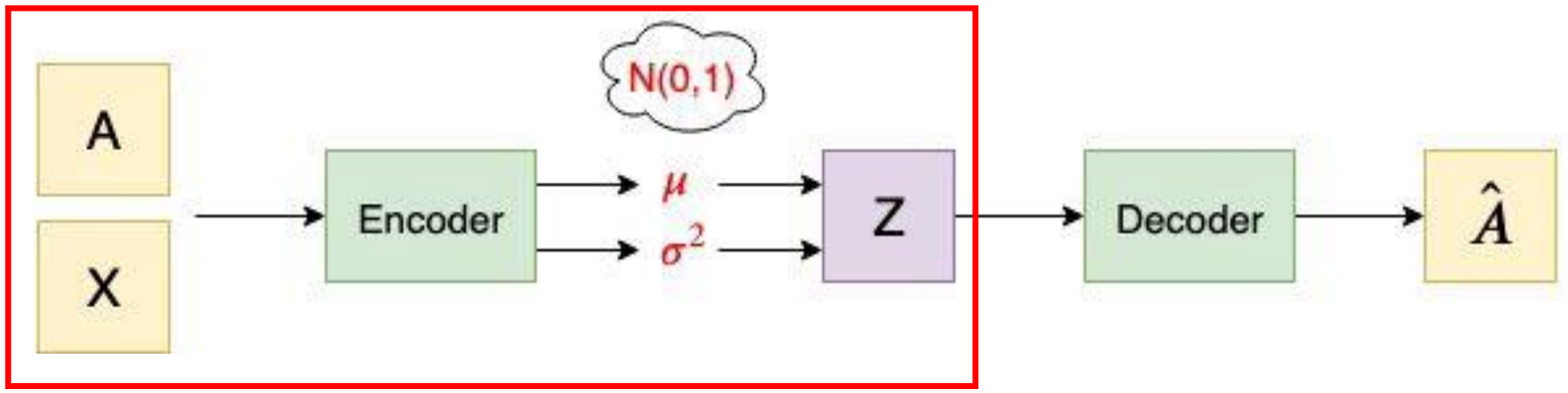
$$GCN(X, A) = \tilde{A}ReLU(\tilde{A}XW_0)W_1$$

Encoder

Then we can calculate **Z** using parameterization trick:

$$Z = \mu + \sigma * \epsilon$$

where $\varepsilon \sim N(0,1)$.



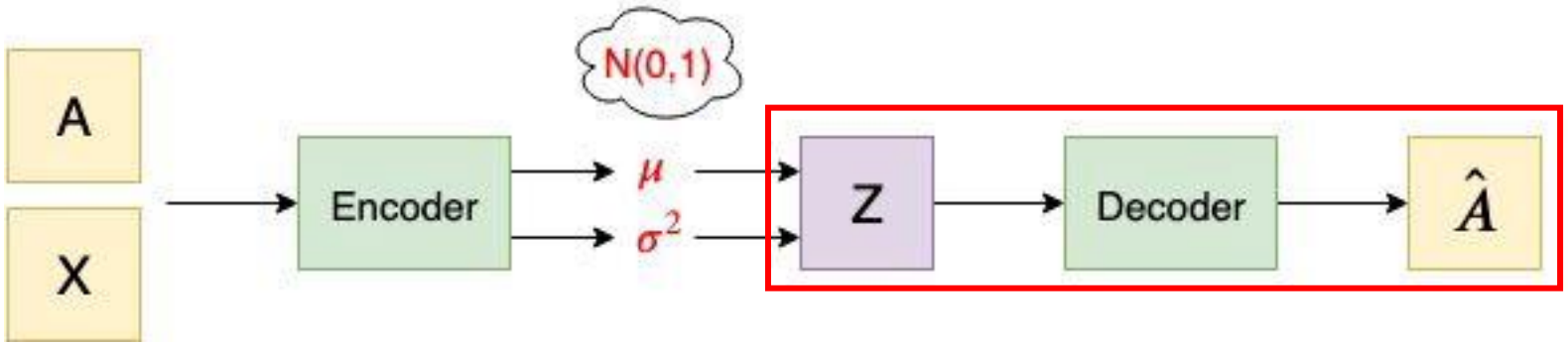
Decoder

The **decoder** (generative model) is defined by an **inner product** between latent variable Z.

The output of our decoder is a reconstructed adjacency matrix $\hat{A}$, which is defined as

$$\hat{A} = \sigma(zz^T)$$

where σ(•) is the logistic sigmoid function.



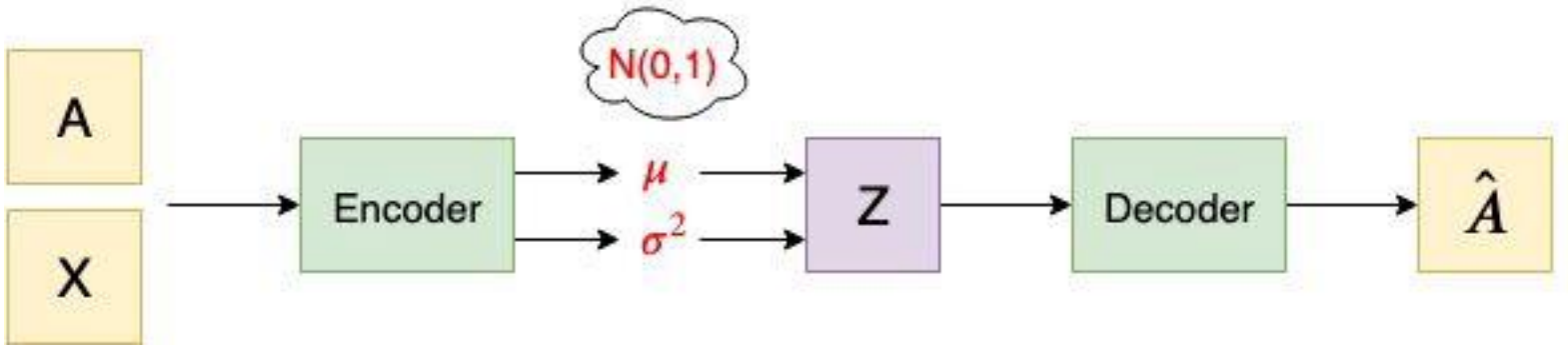
Variational Graph Autoencoders

Encoder:

$$q(z_i|X,A) = N(z_i|\mu_i, diag(\sigma_i^2)$$

Decoder:

$$p(A_{ij} = 1|z_i, z_j) = \sigma(z_i^T z_j)$$



Loss Function

The *first* part is the **reconstruction loss** between the input adjacency matrix and the reconstructed adjacency matrix.

$$L = E_{q(Z|X,A)}[logp(A|Z)] - KL[q(Z|X,A)||p(Z)]$$

The *second* part is the **KL-divergence** between $q(z|x,A)$ and $p(z)$, where $p(z) = N(0,1)$.
It measures how closely our q(Z | X, A) matches to p(Z).

Experiments on Link Prediction

Table 1: Link prediction task in citation networks. See [1] for dataset details.

Method	Cora		Citeseer		Pubmed	
	AUC	AP	AUC	AP	AUC	AP
SC [5]	84.6 ± 0.01	88.5 ± 0.00	80.5 ± 0.01	85.0 ± 0.01	84.2 ± 0.02	87.8 ± 0.01
DW [6]	83.1 ± 0.01	85.0 ± 0.00	80.5 ± 0.02	83.6 ± 0.01	84.4 ± 0.00	84.1 ± 0.00
GAE*	84.3 ± 0.02	88.1 ± 0.01	78.7 ± 0.02	84.1 ± 0.02	82.2 ± 0.01	87.4 ± 0.00
VGAE*	84.0 ± 0.02	87.7 ± 0.01	78.9 ± 0.03	84.1 ± 0.02	82.7 ± 0.01	87.5 ± 0.01
GAE	91.0 ± 0.02	92.0 ± 0.03	89.5 ± 0.04	89.9 ± 0.05	$\mathbf{96.4} \pm 0.00$	$\mathbf{96.5} \pm 0.00$
VGAE	$\mathbf{91.4} \pm 0.01$	$\mathbf{92.6} \pm 0.01$	$\mathbf{90.8} \pm 0.02$	$\mathbf{92.0} \pm 0.02$	94.4 ± 0.02	94.7 ± 0.02

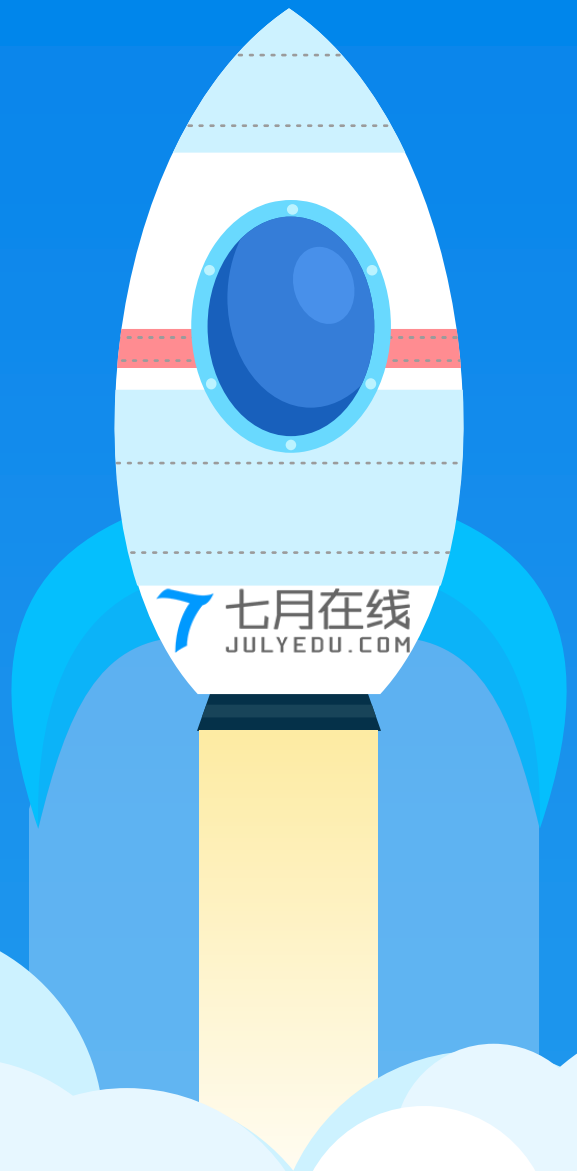
/03

Project: GAE & GVAE

项目实战：GAE & GVAE

Project: GAE & GVAE

微信扫一扫关注我们



THANKS

https://www.julyedu.com

七月在线
JULYEDU.COM