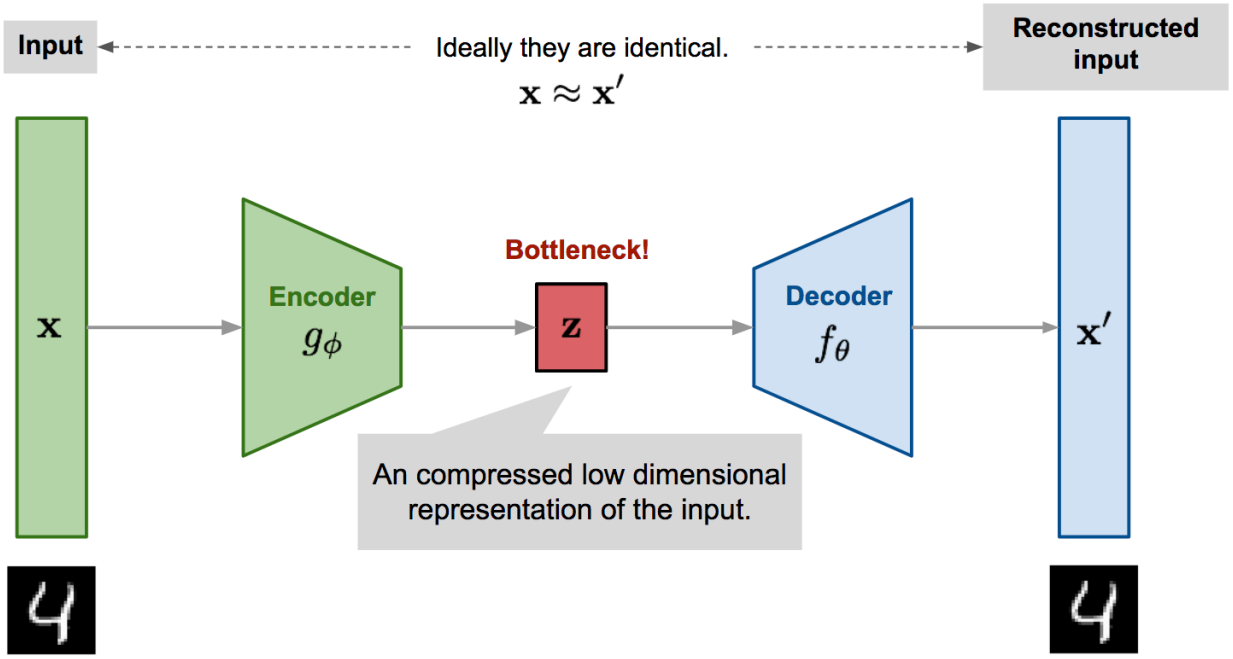
黄家曦 2022141460084

1.自编码器的原理是什么?

自编码器是一种无监督表示学习的方式，可以仅仅依靠数据来训练。通过设计一个非监督式的神经网络，其中的中间层表示可以看作是对输入资料做压缩（类私于PCA降维）或者加入一些其他信息到输入数据中。



通过$\min_{g,f}(x-x')$，可以或者效果好的压缩和解压方式(g,f)，以及降维后的特征空间(z)
对**编码器（Encoder）**：将输入数据 $x \in \mathbb{R}^D$ 映射到低维潜在空间表示 $z \in \mathbb{R}^K$（$K < D$）：

$$z = f_\theta(x) = \sigma(W_e x + b_e)$$

其中 $\theta = \{W_e, b_e\}$ 为编码器参数，σ 为激活函数（如ReLU）。
对**解码器（Decoder）**：从潜在表示 z 重建输入数据 $\hat{x}$：

$$\hat{x} = g_\phi(z) = \sigma(W_d z + b_d)$$

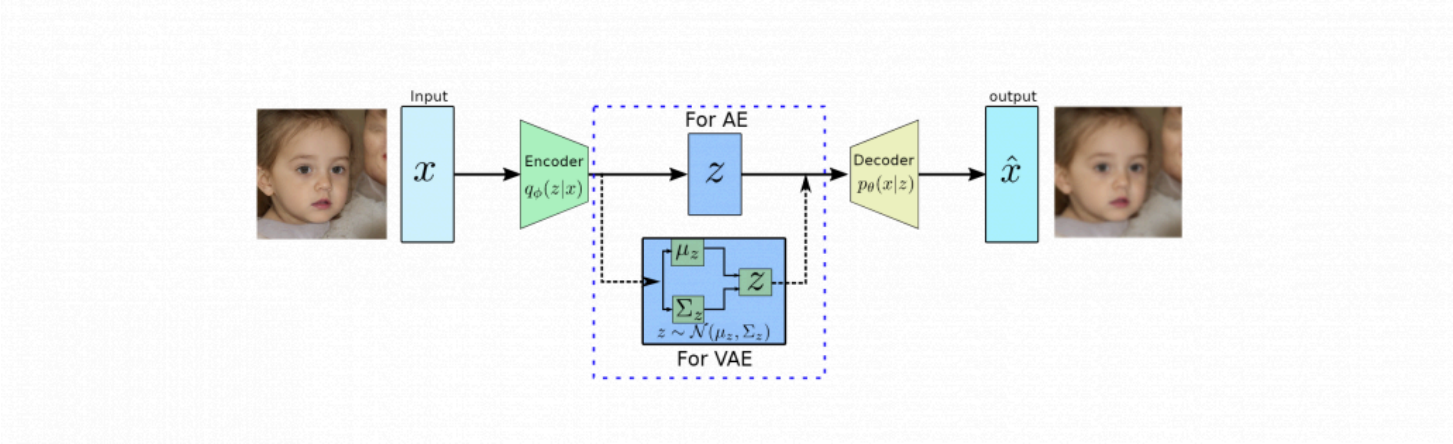
其中 $\phi = \{W_d, b_d\}$ 为解码器参数。

- **优化目标**：最小化重构误差（如L2范数）：

$$\min_{\theta,\phi} \mathcal{L}(x, \hat{x}) = \|x - g_\phi(f_\theta(x))\|_2^2$$

2.简述自编码器的工作机制

AutoEncoder有几种主要类型，包括AE,VAE,DAE,SAE，可以适用于不同场景。AE如上面所说，可以有效的重建输入数据，帮助资料分类，视觉化，储存。
VAE是AE的进阶版，结构上也是Encoder-Decoder



相比于AE，VAE在Encoder过程中增加了一些限制，使得通过Encoder生成的向量服从高斯分布，所以理论上VAE可以控制Encoder过程中的一些细节，进而影响一些任务比如可控图像生成。
引入**概率建模**，强制潜在空间服从先验分布（如高斯分布）。

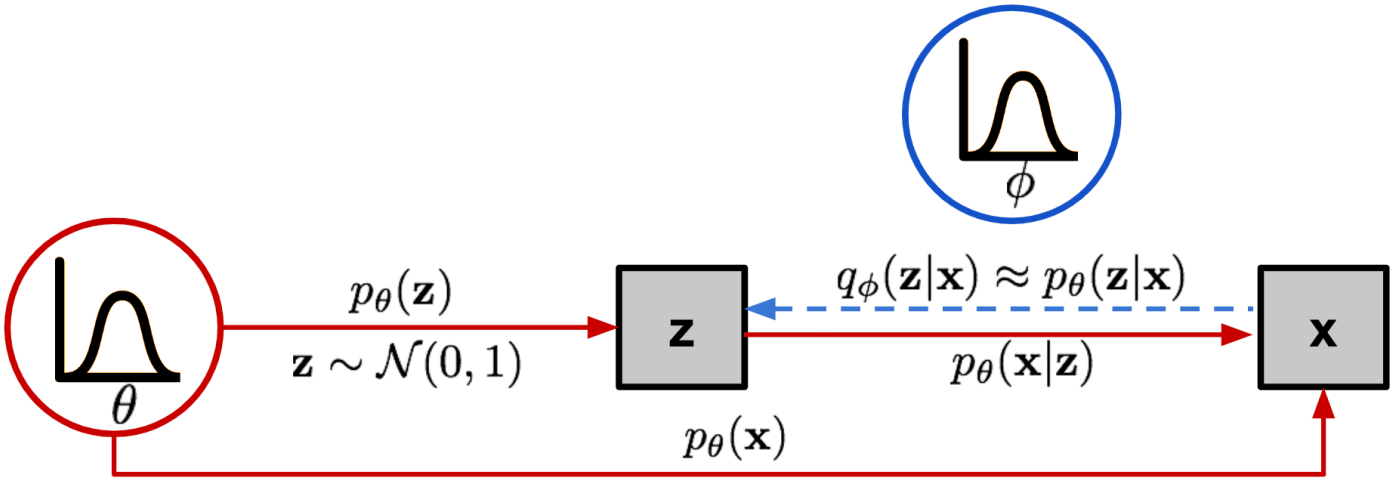
- **编码器输出**：生成潜在变量的均值 $\mu(x)$ 和方差 $\sigma(x)$：

$$z \sim \mathcal{N}(\mu(x), \sigma(x)^2 I)$$

- **损失函数**：包含重构误差与KL散度约束：

$$\mathcal{L}_{\text{VAE}} = \underbrace{\mathbb{E}_{z\sim q(z|x)}[\log p(x|z)]}_{\text{重构误差}} + \underbrace{\beta \cdot D_{\text{KL}}(q(z|x)\|p(z))}_{\text{分布约束}}$$

其中 β 控制潜在空间约束强度。



DAE是一种用于学习对图片去噪的神经网络，相比于AE，在模型输入阶段对输入添加随机噪声使得$x \to x'$，目标是$\min_{g,f}(g(f(x')) - x)$，还有种SAE(Sparse AutoEncoder)是在AE基础上加上L1 Regularzation,迫使AE将每个输入表示为少量节点的组合，这种特征稀疏的过程可以过滤掉一些无用的信息

3.简述波尔兹曼机

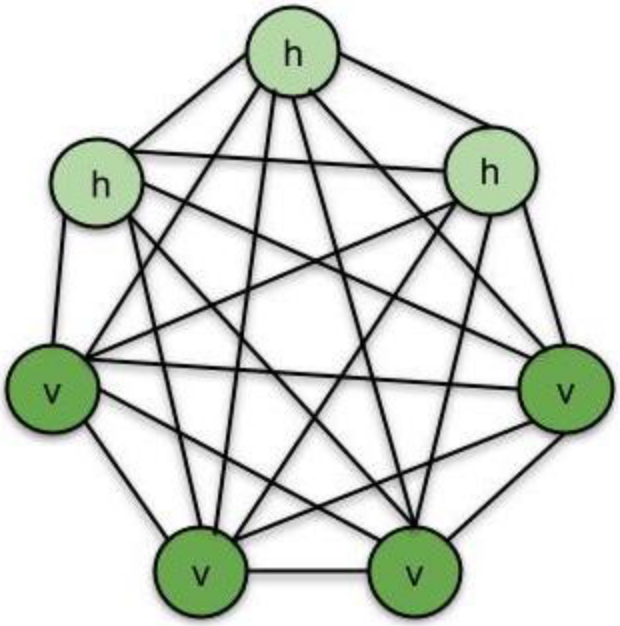
波尔兹曼机是随机神经网络和循环神经网络的一种，由Hinton和Terry在1985年提出。
玻尔兹曼机可被视作随机过程的，可生成的相Hopfiled Network。它是最早能够学习内部表达，并能表达和（给定充足的时间）解决复杂的组合优化问题的神经网络。但是，没有特定限制连接方式的玻尔兹曼机目前为止并未被证明对机器学习的实际问题有什么用。所以它目前只在理论上显得有趣。然而，由于局部性和训练算法Hebb Learning的一些性质，以及它们和简单物理过程相似的并行性，如果连接方式是**受约束**的（即受限玻尔兹曼机），在解决实际问题上将会足够高效有用。
玻尔兹曼机是一种**基于能量的模型** （Energy-Based Model），其核心思想是通过一个能量函数来定义网络的状态分布。每个神经元状态（0或1）的组合对应一个能量值，整个系统的状态服从**玻尔兹曼分布**。能量分布为：

$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i w_{ij} h_j$$

其中：

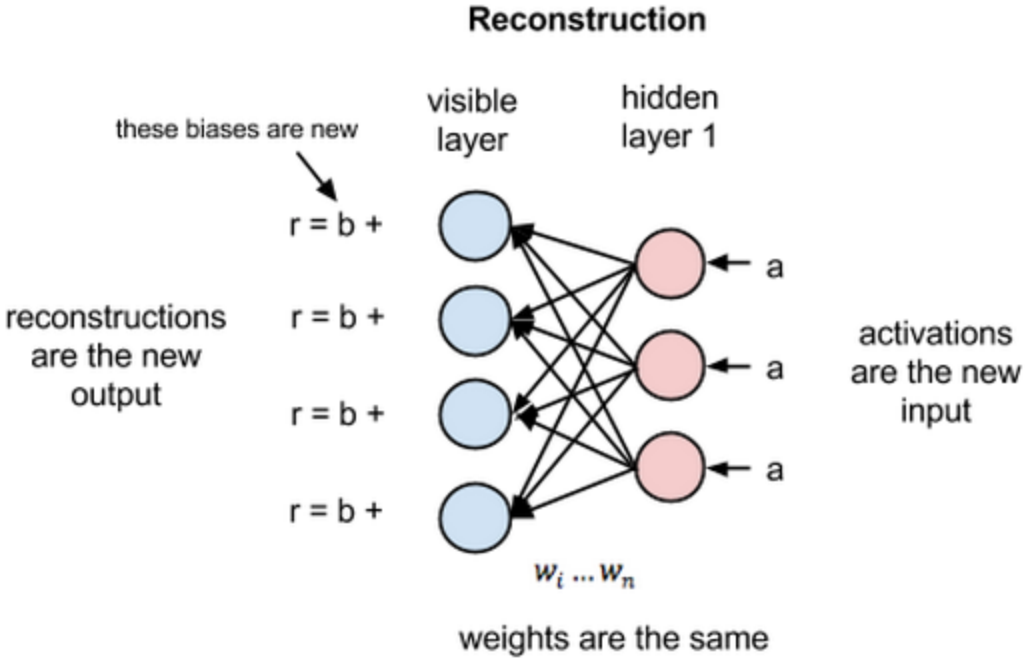
- v：可见层（输入数据）
- h：隐藏层
- a_i,b_j：偏置项
- w_{ij}：连接权重

系统的目标是最小化数据的能量，从而最大化观测数据的概率。



v - visible nodes, h - hidden nodes

受限波尔兹曼机（RBM）是一种可通过输入数据集学习概率分布的随即生成神经网络。RBM限定模型必须为二分图，模型包含输入单元和隐藏单元，每条边必须连接一个可见单元和隐藏单元，不允许隐藏单元之间的连接。



波尔兹曼机，在预训练时一般都是逐层进行训练，冻结其他隐藏层。因为当时GPU和训练框架等原因，

全参数梯度下降很难以训练实现。

4.怎么将单模态波尔兹曼机迁移到多模态？

将单模态 RBM 扩展到多模态，核心思想是为每个模态引入独立的可见层，并共享一个统一的隐藏层，从而建模多个模态之间的联合分布，即对两个输入x_{v_1}和x_{v_2}，共享一个h隐藏层。
对于双模态 RBM，其能量函数可表示为：

$$
E(v^{(1)},v^{(2)},h)=-a^{(1)}v^{(1)}-a^{(2)}v^{(2)}-b^{T}h-v^{(1)T}W^{(1)}h-v^{(2)T}W^{(2)}h
$$

其中：

- $v^{(1)},v^{(2)}$：分别代表两个模态的可见单元；
- $W^{(1)},W^{(2)}$：各自模态与隐藏层之间的连接权重；
- $a^{(1)},a^{(2)},b$：各层的偏置项；
- h：共享的隐藏单元。
  对应的联合概率分布为：

$$P(v^{(1)},v^{(2)},h)=\frac{1}{Z}e^{E(v^{(1)},v^{(2)},h)}$$

> 通过设计多模态波尔兹曼机，使得模型能够捕捉不同模态间的潜在语义关联。能支持模型万岁跨模态生成与检索（如图像 → 文本 或文本 → 图像），无监督/弱监督条件下学习多模态对齐等任务，是早期深度学习中实现统一潜在语义空间的重要尝试。

5.简述多模态自编码器的过程

多模态自编码器（Multimodal Autoencoder）是一种用于学习多个模态数据共享表示的深度学习模型。与传统的单模态自编码器不同，它处理的是来自不同来源或形式的数据，例如图像、文本、音频等。其核心目标是通过编码器将多种模态的信息压缩到一个共享的潜在空间中，并通过解码器尽可能地重建原始输入。

相比于单模态自编码器，多模态自编码器面临的一个关键问题是：**如何处理多个模态的低维表示？**

常见的策略包括以下三种：

1. **模态融合（Early Fusion or Late Fusion）**
   - 在编码阶段就将不同模态的数据进行拼接或加权组合，形成统一的输入，送入共享的编码器网络。

- 优点是可以利用模态之间的互补信息，学习更丰富的联合表示；
- 缺点是可能导致模态间的信息干扰，尤其是当模态之间差异较大时。

2. **分别编码（Separate Encoding）**
   - 每个模态都有独立的编码器，分别提取各自模态的特征表示；
   - 然后再通过某种方式（如拼接、注意力机制、门控机制等）对这些表示进行整合；
   - 这种方式保留了模态的特异性，同时也能在高层进行跨模态交互，更适合模态之间存在较大差异的情况。
   - 常见例子有Raw-CCA，RBM-CCA，DCCAE等方法
3. **联合训练统一表示（Joint Representation Learning）**
   - 不仅使用多个编码器处理不同模态，还设计共享的中间层来学习一个统一的语义空间；
   - 解码器部分也可能采用共享结构或模态专用结构，以重建各模态的输入；
   - 目标是让不同模态在潜在空间中具有可比性，便于后续任务如跨模态检索、生成等；
   - 常见方法包括变分多模态自编码器（Multimodal VAE）、CLIP（主要通过对比学习来训练）、AE-2 Net（AutoEncoder in AutoEncoder）等。