

自编码器

概览

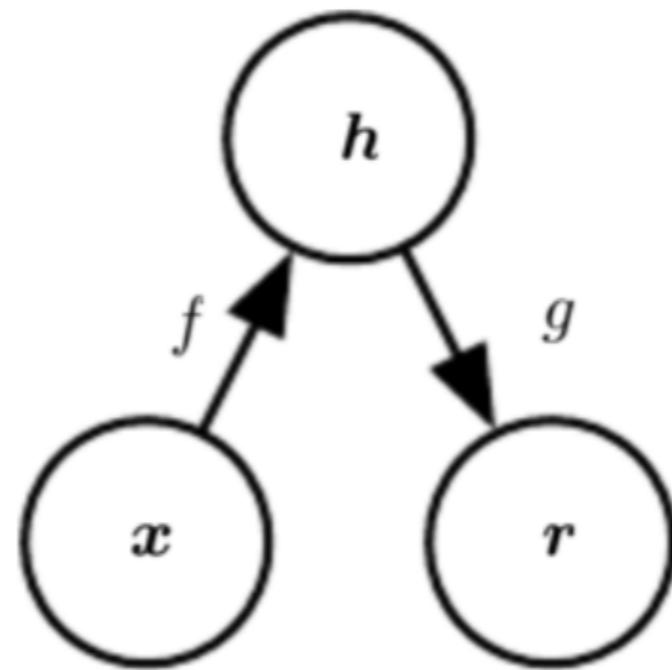
1. 简介
2. 欠完备自编码器
3. 降噪自编码器
4. 卷积自编码器
5. 栈式自编码器
6. 栈式自编码器的应用

1. 概览

自编码器

自编码器 (AutoEncoder, AE) 是神经网络的一种，经过训练后能尝试将输入复制到输出（非恒等映射）。自编码器拥有一个隐藏层，可以对输入进行编码。自编码器常用来降维、特征学习，近年来自编码器也用于生成模型建模。

自编码器



上图为自编码器的一般结构，通过内部表示或编码 h 将输入 x 映射到输出（重构） r ，自编码器具有两个组件：编码器 f （将 x 映射到 h ）和解码器 g （将 h 映射到 r ）。

2. 欠完备自编码器

欠完备自编码器

欠完备自编码器（Undercomplete AE） 即使得自编码器隐藏层维度小于输入层维度，强制自编码器学习数据中的有效特征，达到数据降维的目的。

欠完备自编码器

欠完备自编码器的学习过程可以简单地描述为最小化一个损失函数：

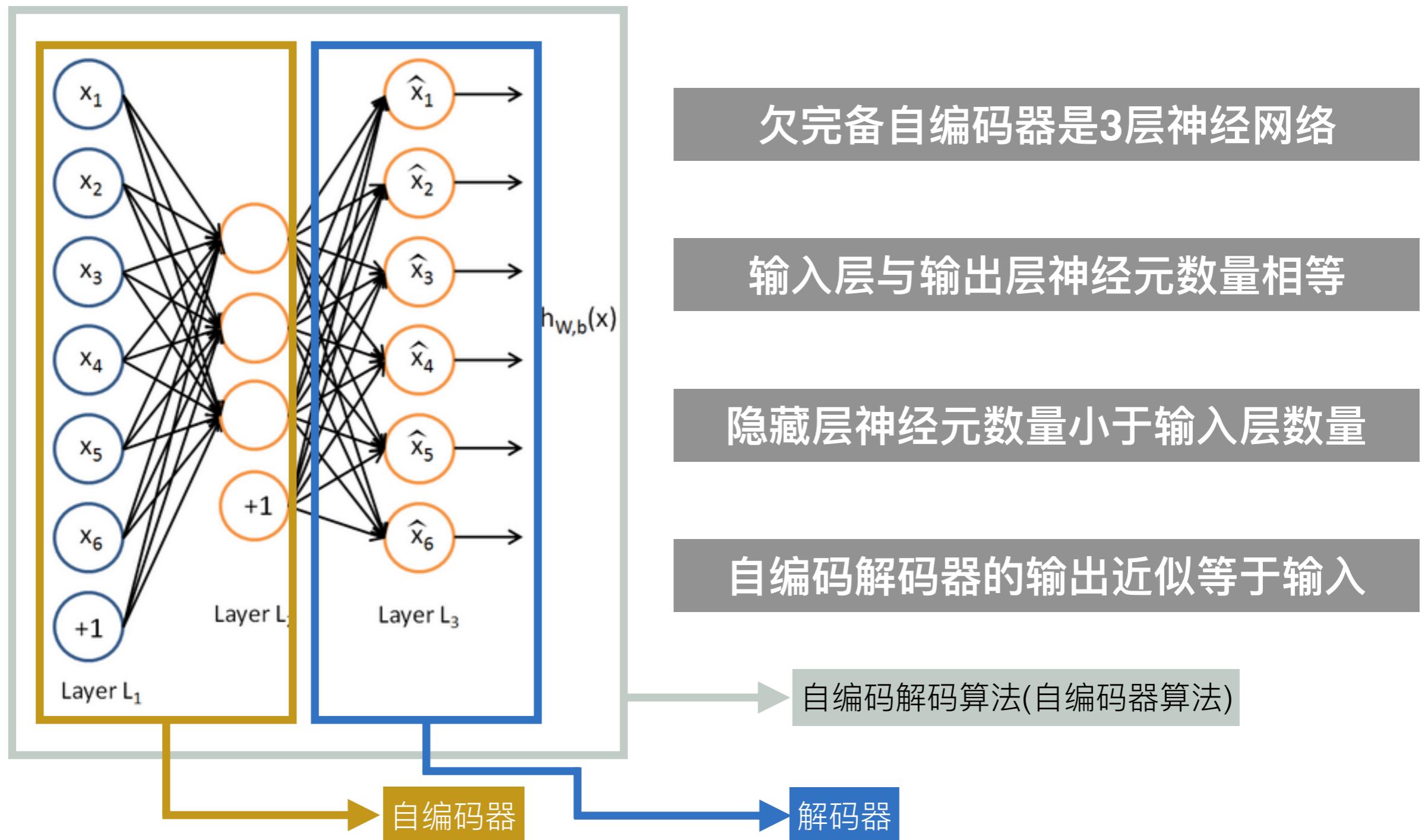
$$L(x, g(f(x))),$$

其中L表示损失函数，一般为均方误差代价函数。

当解码器是线性的且L是均方误差，欠完备自编码器等价于PCA。

当编码器与解码器用于非线性激活函数时，欠完备自编码器能够学习出更强大的PCA非线性推广。

欠完备自编码器



TensorFlow练习

1. 实现欠完备自编码器；
2. 修改不同大小的隐藏层，观察自编码器的输出表现。

思考

1. 当自编码器容量太大时会发生什么？如何解决
 - 当自编码器容量太大时，将无法学到有关数据分布的有用信息；
 - 可以加入正则惩罚。
2. 训练得到的自编码器隐藏层是否只有与之相连的输出层可以解码或使用数据？
 - 不是，自编码器的隐藏层是数据的另一种表现形式，就像PCA一样，其它模型也可以利用。

3. 降噪自编码器

降噪自编码器

降噪自编码器（Denoising AE）是一类接受损坏数据作为输入，并训练来预测原始未被损坏数据作为输出的自编码器，是一种比欠完备自编码器更具鲁棒性的自编码器。

降噪自编码器

降噪自编码器训练过程：

1. 从训练数据中采一个训练样本 \mathbf{x} 。
2. 从 $C(\tilde{\mathbf{x}} \mid \mathbf{x} = \mathbf{x})$ 采一个损坏样本 $\tilde{\mathbf{x}}$ 。
3. 将 $(\mathbf{x}, \tilde{\mathbf{x}})$ 作为训练样本来估计自编码器的重构分布 $p_{\text{reconstruct}}(\mathbf{x} \mid \tilde{\mathbf{x}}) = p_{\text{decoder}}(\mathbf{x} \mid \mathbf{h})$, 其中 \mathbf{h} 是编码器 $f(\tilde{\mathbf{x}})$ 的输出, p_{decoder} 根据解码函数 $g(\mathbf{h})$ 定义。

TensorFlow练习

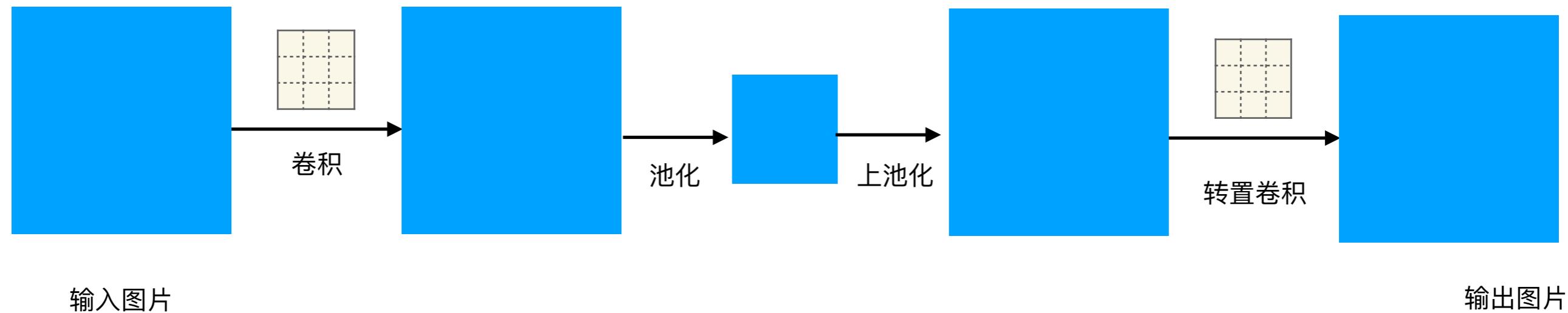
- 使用TensorFlow实现降噪自编码器。

4. 卷积自编码器

卷积自编码器

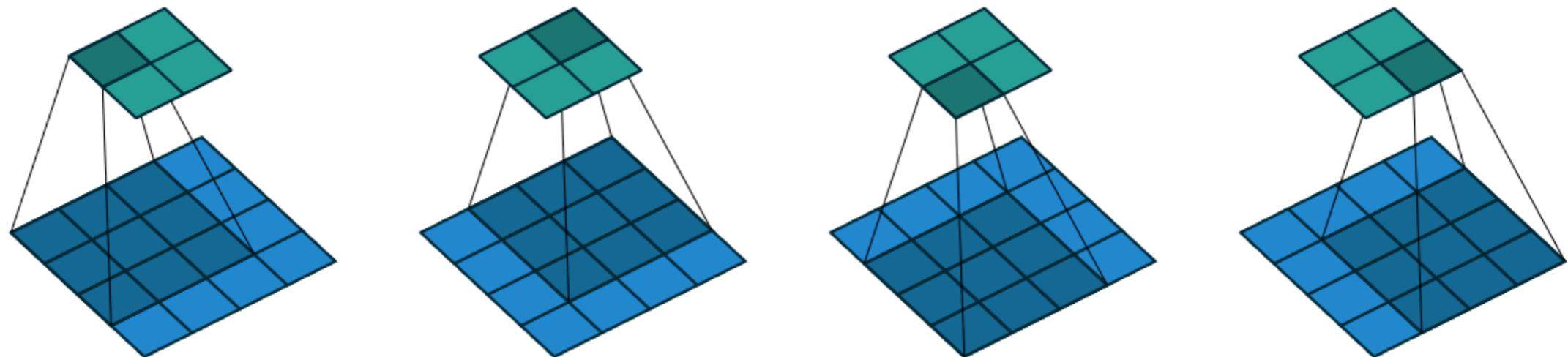
卷积自编码器（Convolutional AE） 是使用卷积层与转置卷积层实现的自编码器，相比使用全连接层，使用卷积层与可以更好的提取图像特征。

卷积自编码器



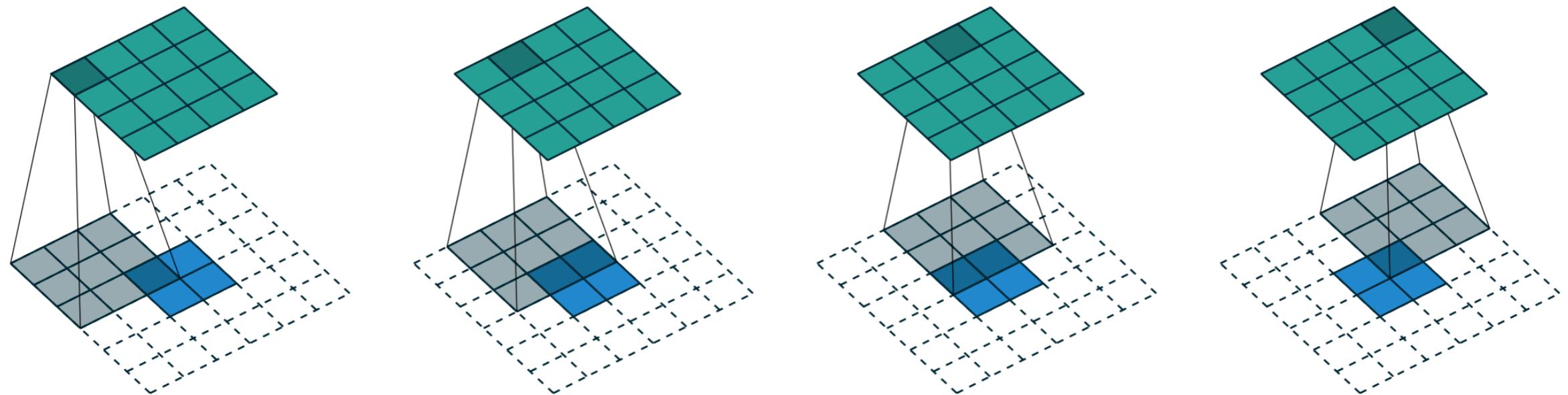
卷积使得图片大小缩小，转置卷积是一种上采样方法，可以使得图片放大。池化是下采样操作，反池化即上采样，最大池化的上池化需要记录池化位置，将对应值传递过去，其他值置0。

卷积与转置卷积



卷积：输入图片大小为 $4 * 4$ ，卷积核大小为 $3 * 3$ ，步长为1，没有填充边界，得到的特征图大小为 $2 * 2$ 。

卷积与转置卷积



转置卷积：输入图片大小为 $2 * 2$ ，卷积核大小为 $3 * 3$ ，步长为1，没有填充边界（事实上填充了），得到的特征图大小为 $4 * 4$ 。

TensorFlow实现

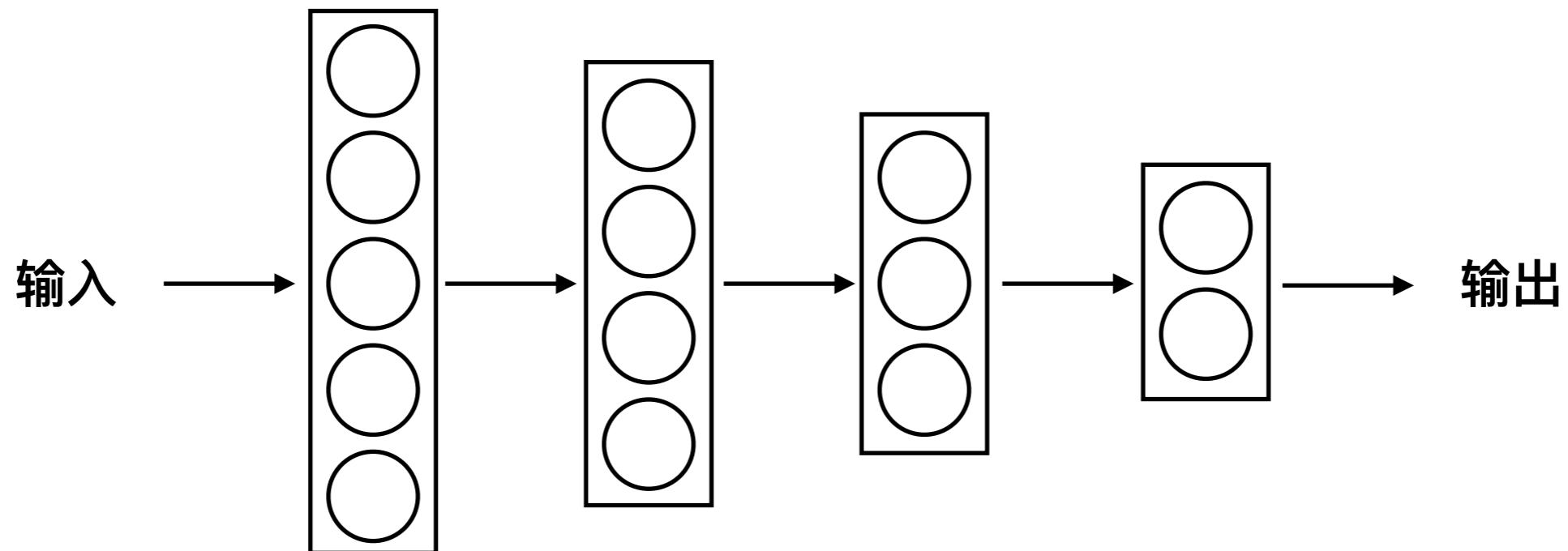
1. 使用TensorFlow实现卷积自编码器；
2. 使用卷积自编码器对图片进行降噪。

5. 栈式自编码器

栈式自编码器

栈式自编码器 (Stacked AE) 是一个由多层自编码器组成的神经网络，即前一层自编码器隐藏层的输出作为下一层自编码器的输入。栈式自编码器拥有更好的非线性表达能力。

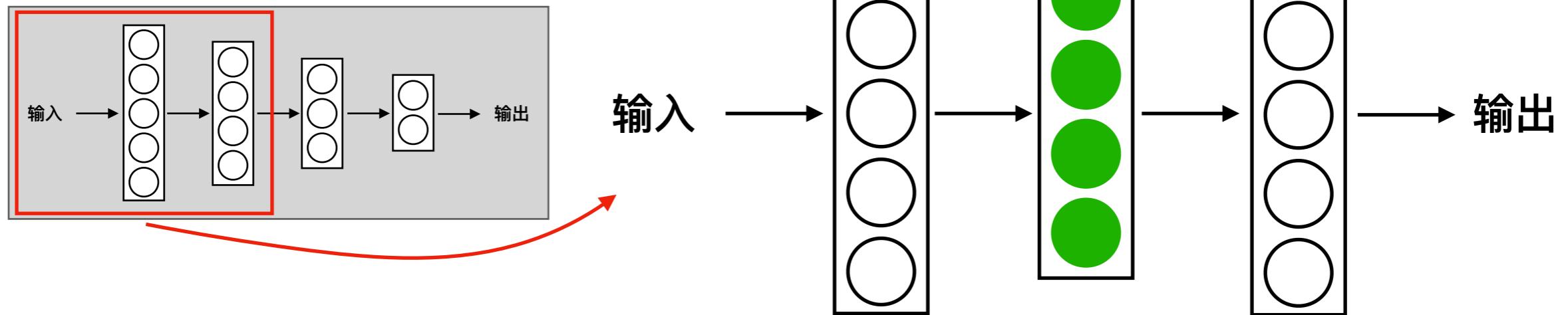
栈式自编码器



与欠完备自编码器相比，栈式自编码器拥有不止一个隐藏层。每个隐藏层均使用欠完备（或正则或降噪或卷积等）自编码器使用贪心算法训练而成。

栈式自编码器的训练

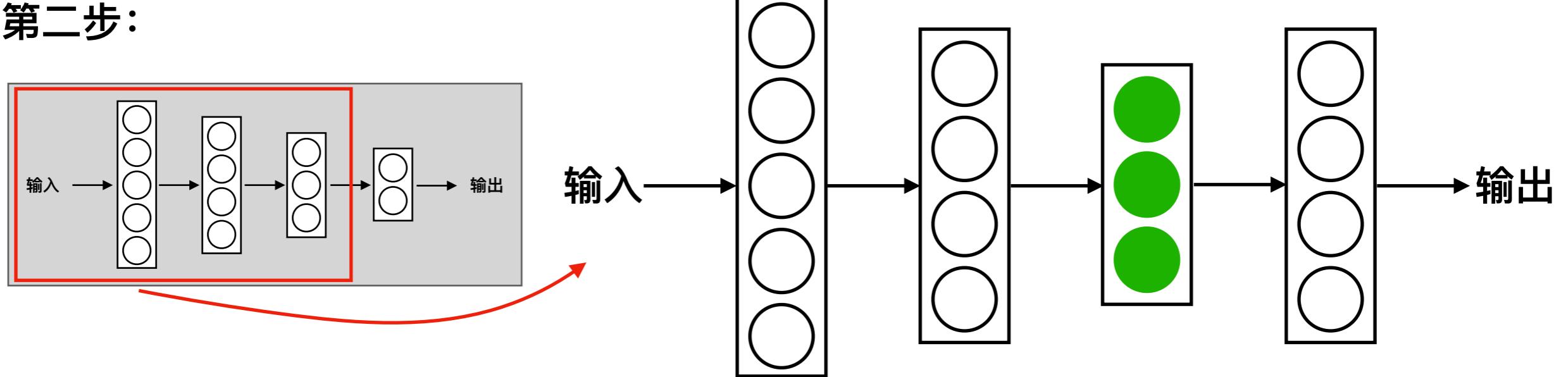
第一步：



训练第一个隐藏层：即训练第一个自编码器，其输入与标记相同。训练结束时保留输入层与此隐藏层之间的参数。

栈式自编码器的训练

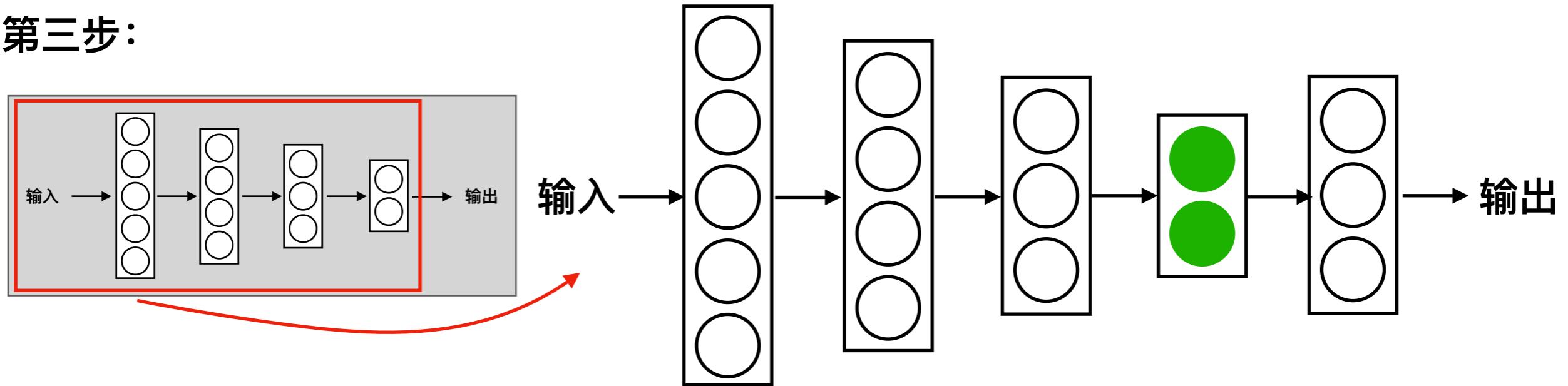
第二步：



训练第二个隐藏层：即训练第二个自编码器，其输入与标记为第一个自编码器隐藏层的输出。训练结束时保留此隐藏层与之前的隐藏层之间的参数。

栈式自编码器的训练

第三步：



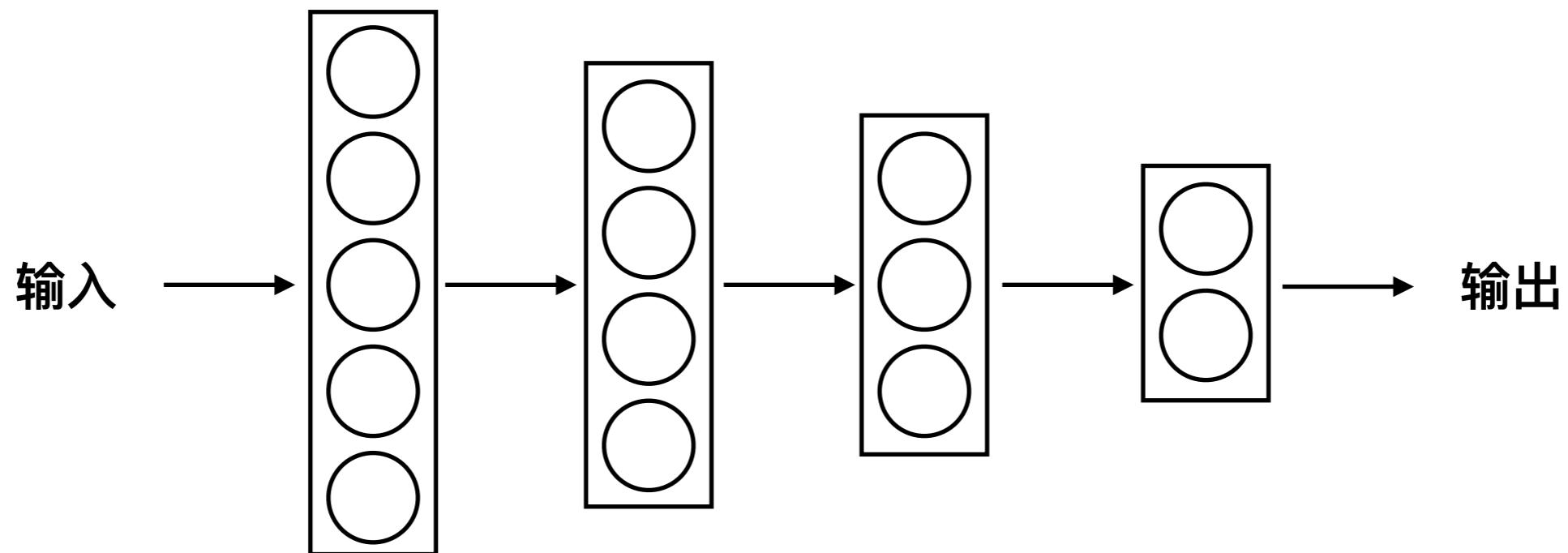
训练第三个隐藏层：即训练第三个自编码器，其输入与标记为第二个自编码器隐藏层的输出。训练结束时保留此隐藏层与之前的隐藏层之间的参数。

栈式自编码器的逐层训练

- 栈式自编码器使用多个自编码器堆叠而成；
- 训练时不存在梯度消失问题，训练速度快；
- 相比单个自编码器，其表达能力更强；
- 每个自编码器贪婪训练；
- 多次训练，操作较为繁琐。

微调

通常的，训练完毕的栈式自编码器可用于特征提取或降维，这个训练过程也被称之为**无监督预训练**。为了更好使用模型，通常会在完成预训练时，对模型进行有监督的**微调（fine-tune）**，使模型变得更有用。



微调

- 通过微调，模型的层与层之间配合更加“默契”；
- 通过微调，使得模型可以完成其他任务；
- 预训练已学习到很多知识，微调时训练模型会很快收敛；
- 微调时使用较少的监督学习样本；
- 现阶段，得益于计算机运行速度的提升与相关技术的出现，已经极少使用逐层预训练来训练模型。

TensorFlow练习

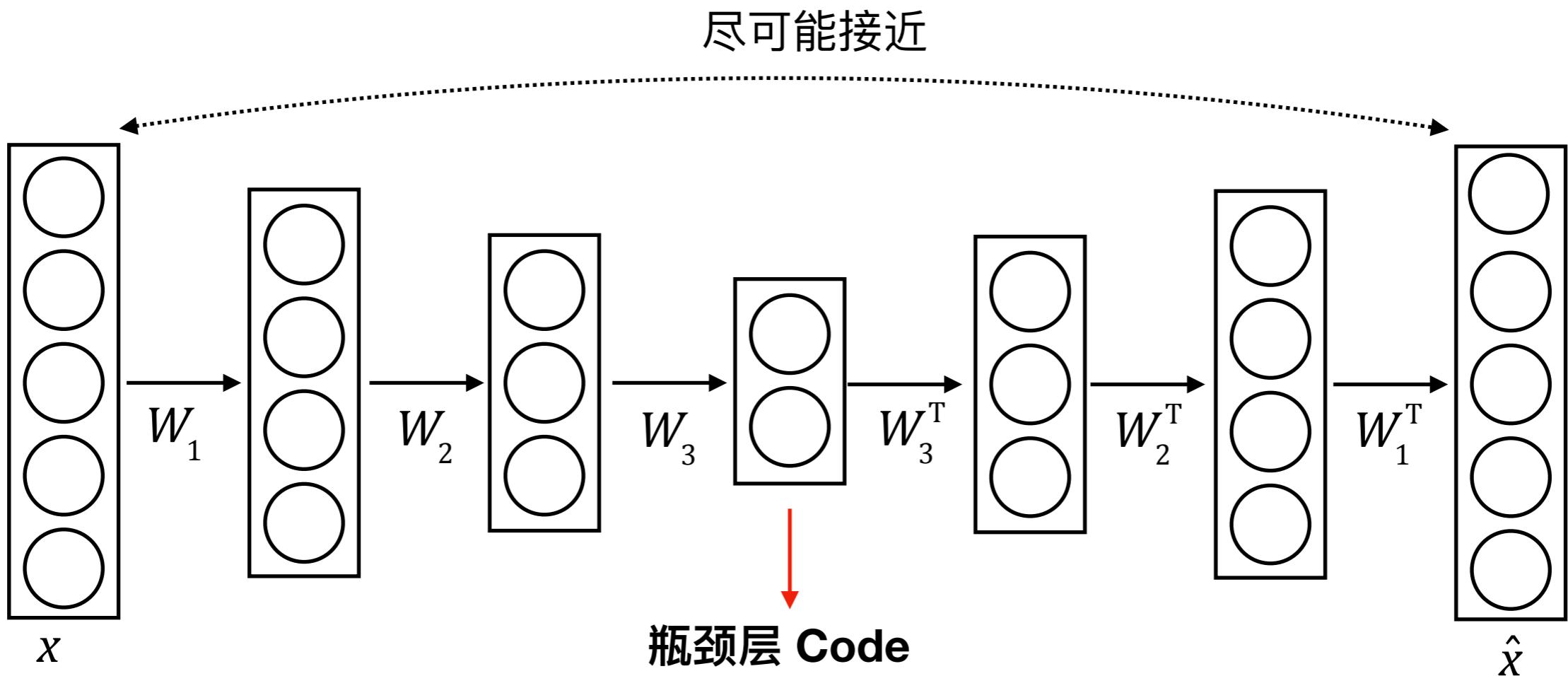
1. 构建并训练栈式自编码器；
2. 微调栈式自编码器；
3. 将栈式自编码器改进为栈式卷积自编码器。

6. 栈式自编码器应用

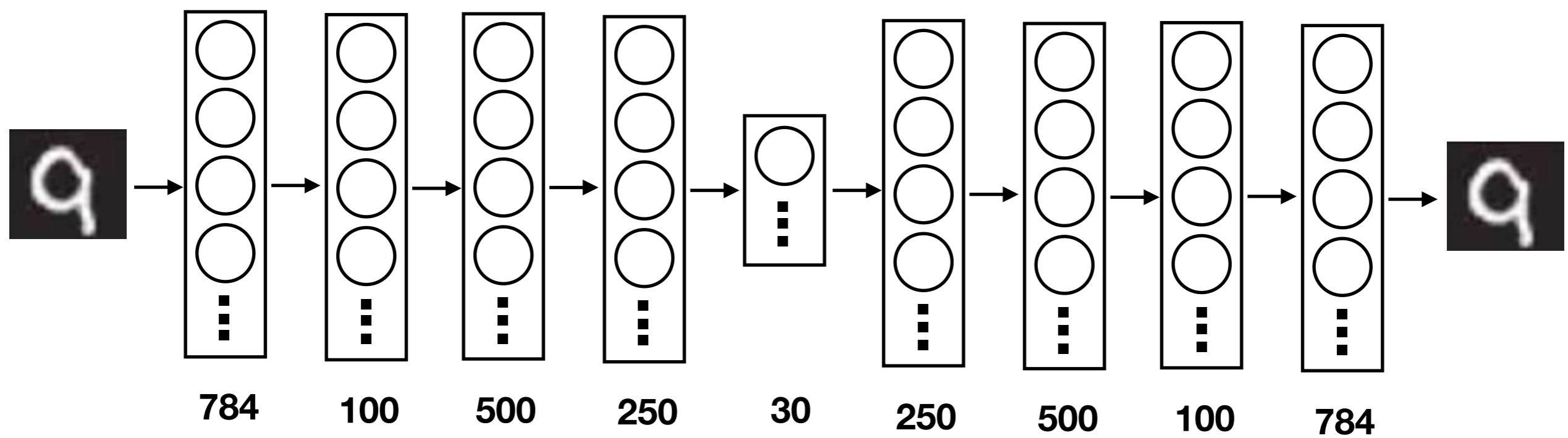
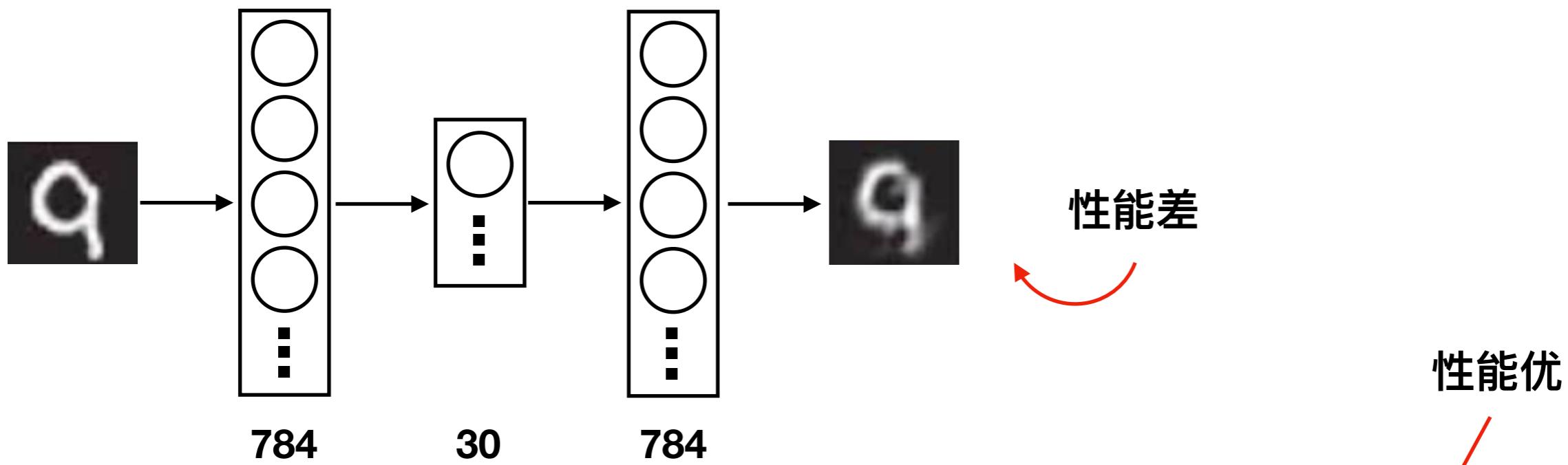
栈式自编码器的应用

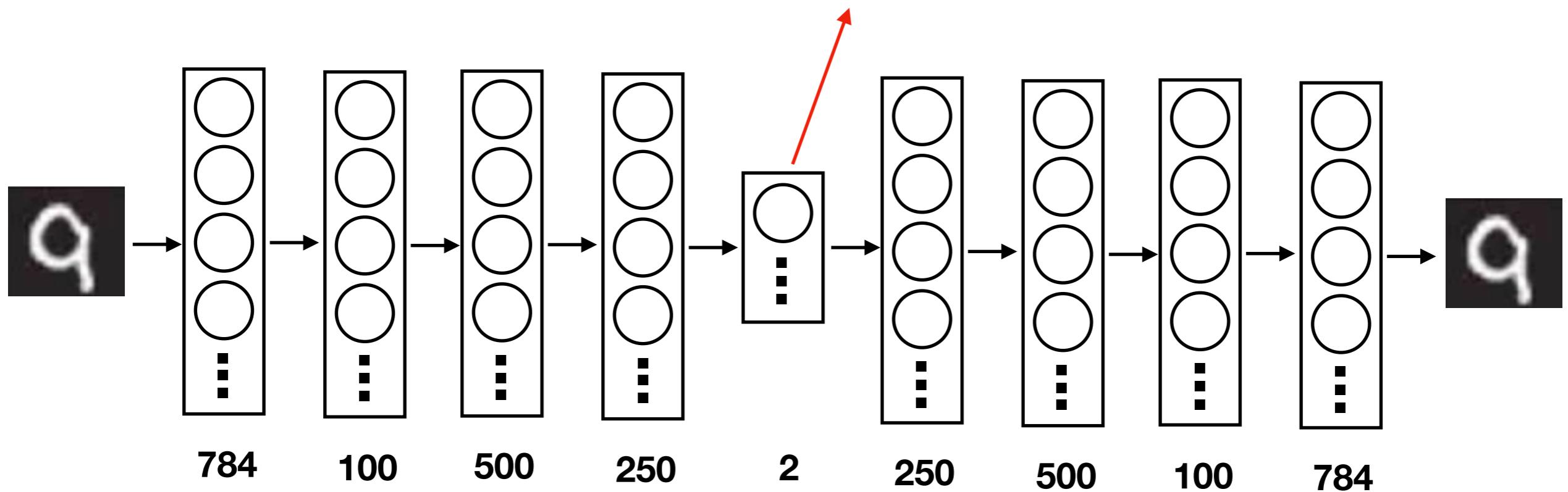
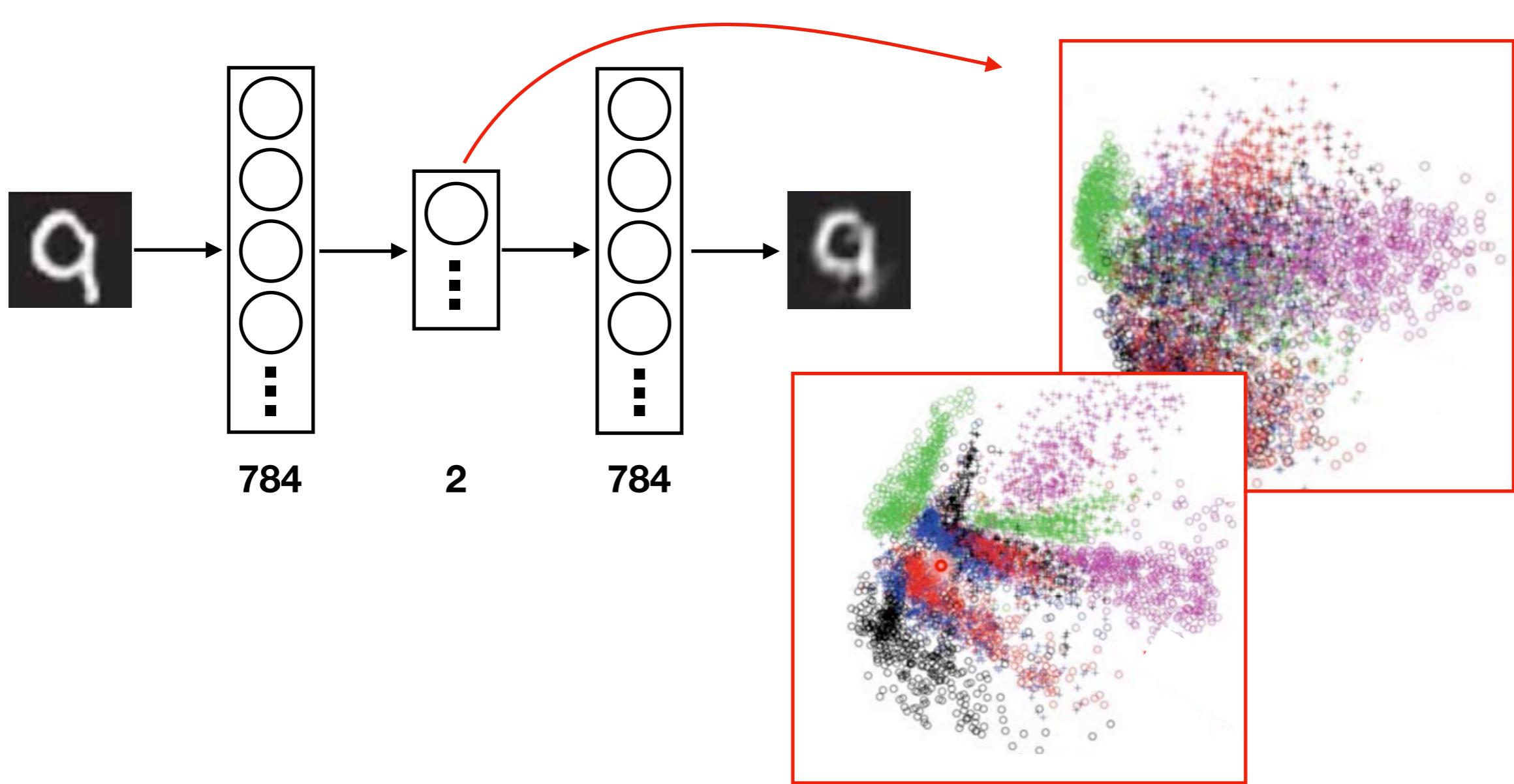
使用栈式自编码器初始化一个隐藏层逐渐减小神经网络，最后一个隐藏层将拥有非常低的维度，而低维表示可以提高很多任务的性能和运行速度。可以利用这一特性完成信息检索、语义哈希等任务。

栈式自编码器



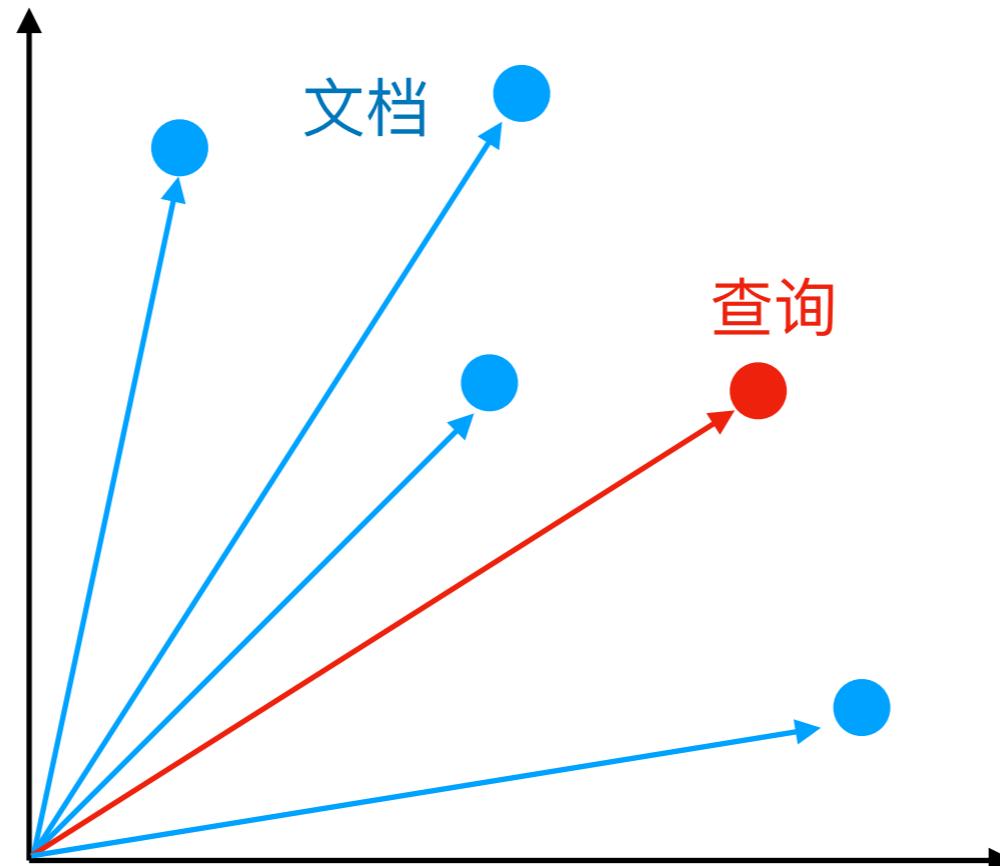
事实上，解码器并不一定需要使用“对称”的参数初始化，一般的，使用随机初始化即可。





文本检索

今	1
天	1
我	3
很	2
...	...

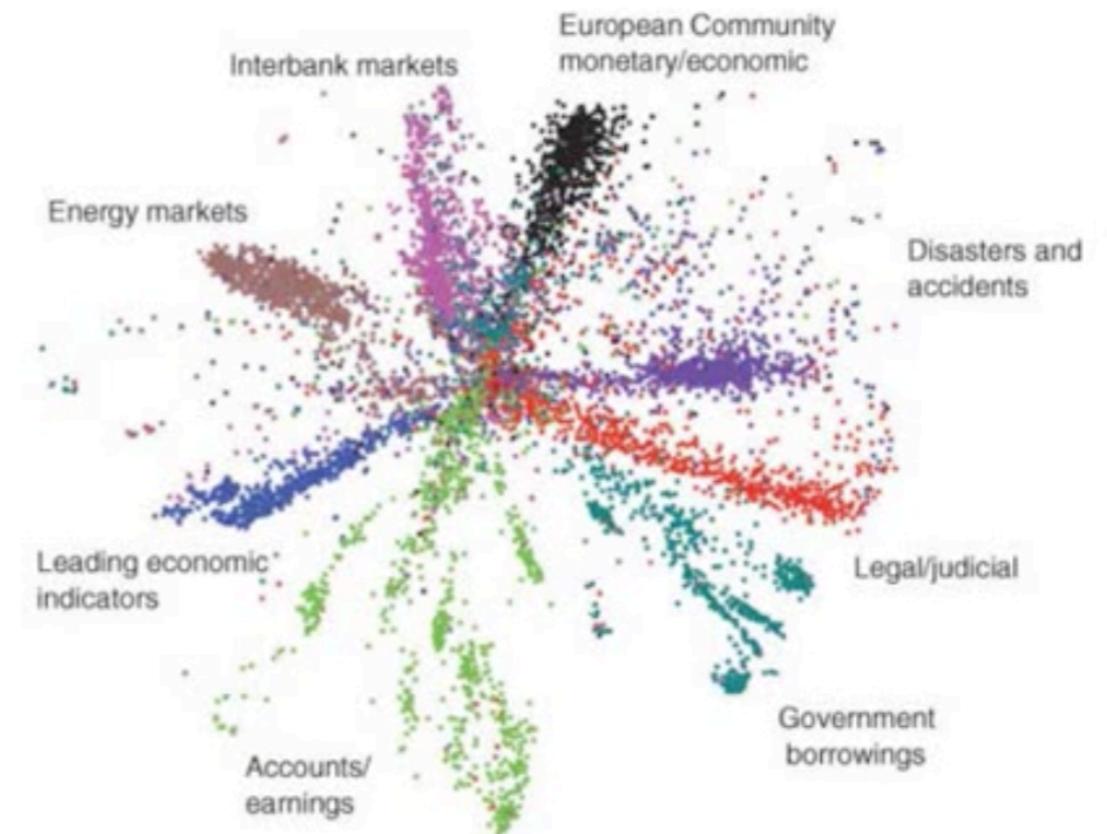
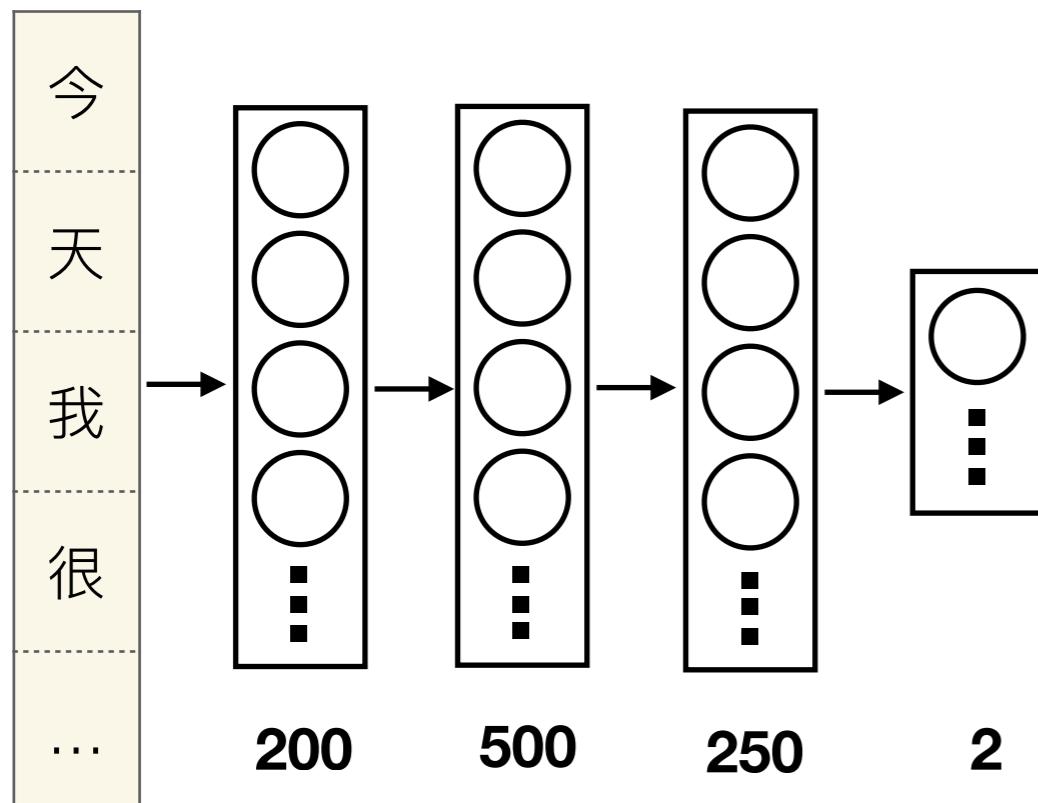


直接使用词袋模型送入向量空间中查询相近文档时，没法利用语义信息。

词袋模型
(词/词频)

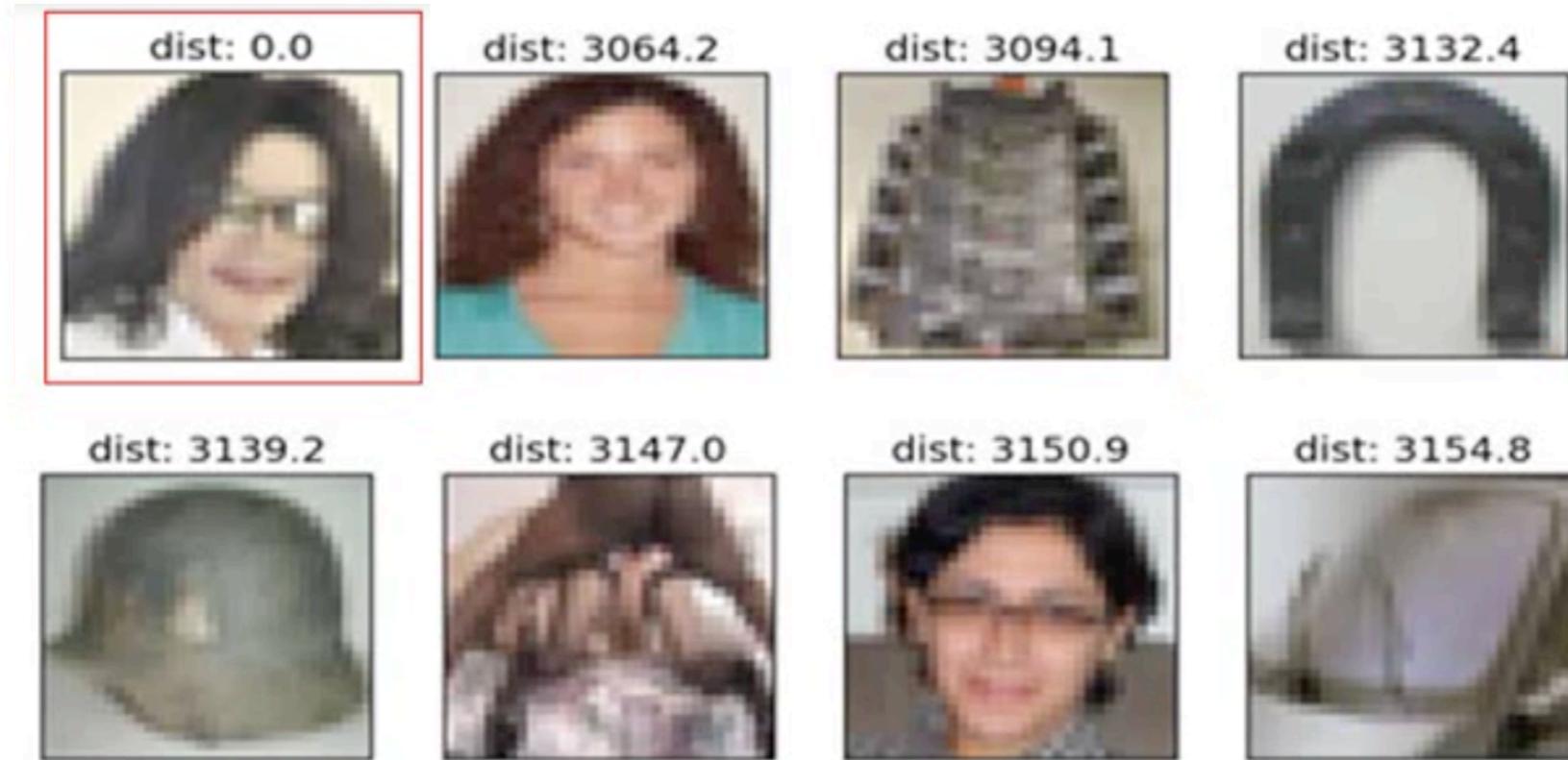
向量空间模型

使用SAE进行文本检索



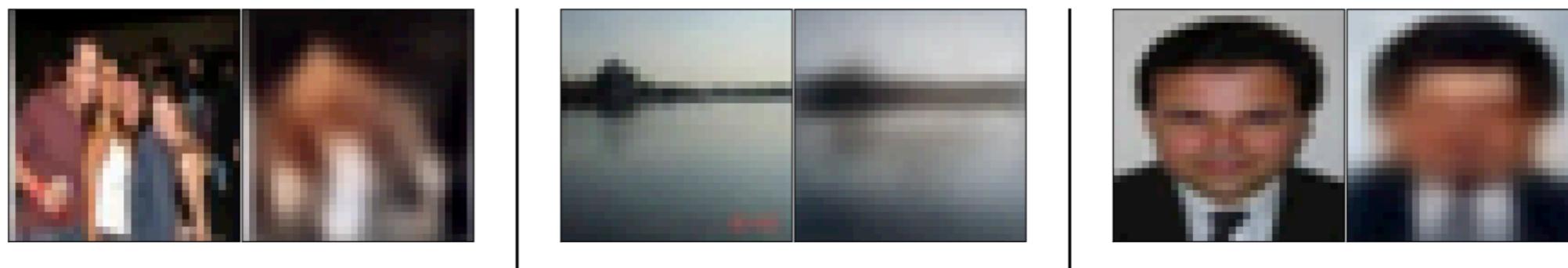
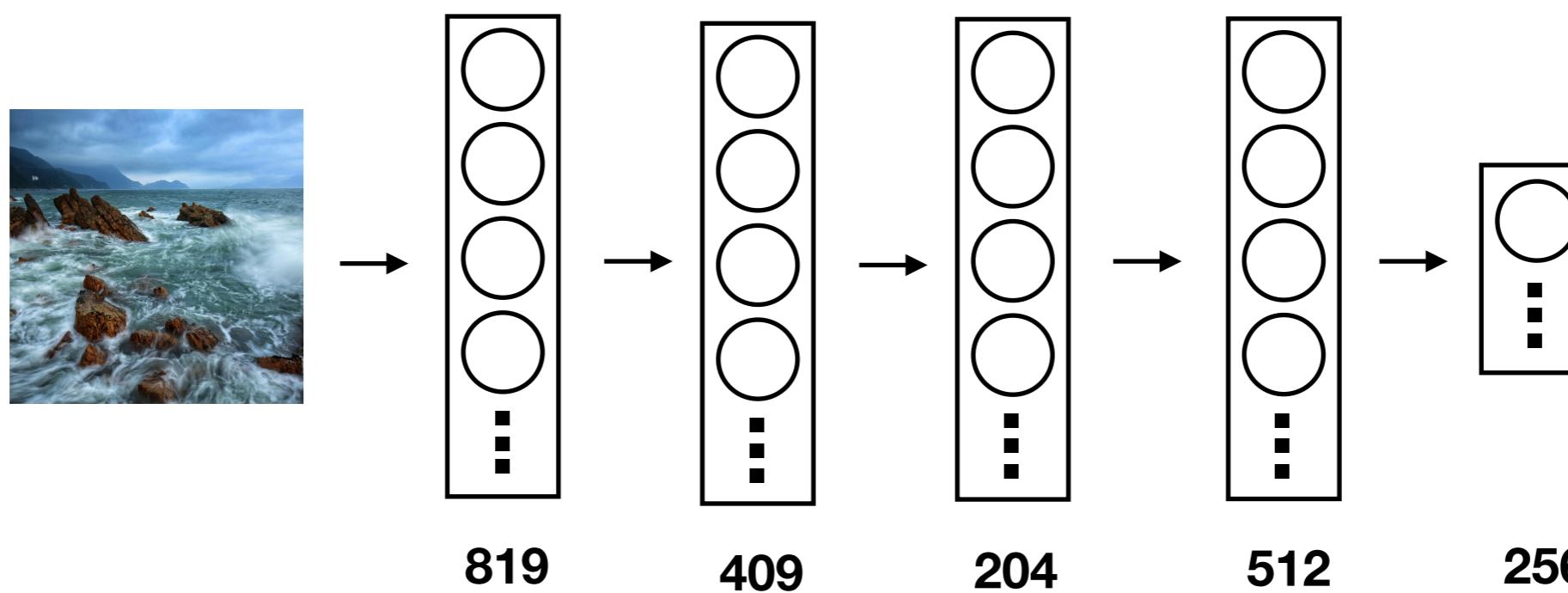
当使用SAE查询文档（类别）时，只需要将文档输入SAE，得到2个输出，找到距离最近文档（类别）即可。

图像检索



使用欧氏距离计算得到的最相近的图片，效果很差。

使用SAE进行图像检索



训练SAE完成之后输入图片与解码对照

使用SAE进行图像检索



使用SAE检索图像获得的结果中，前几个都是人脸，且很接近。

TensorFlow实现

- 使用TensorFlow构建与训练SAE模型，完成cifar10图像检索任务。

THANKS