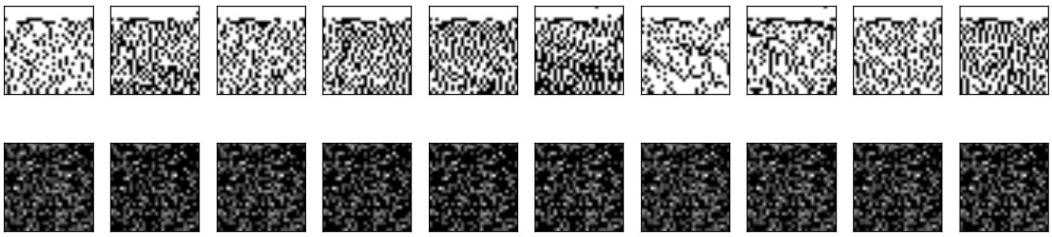
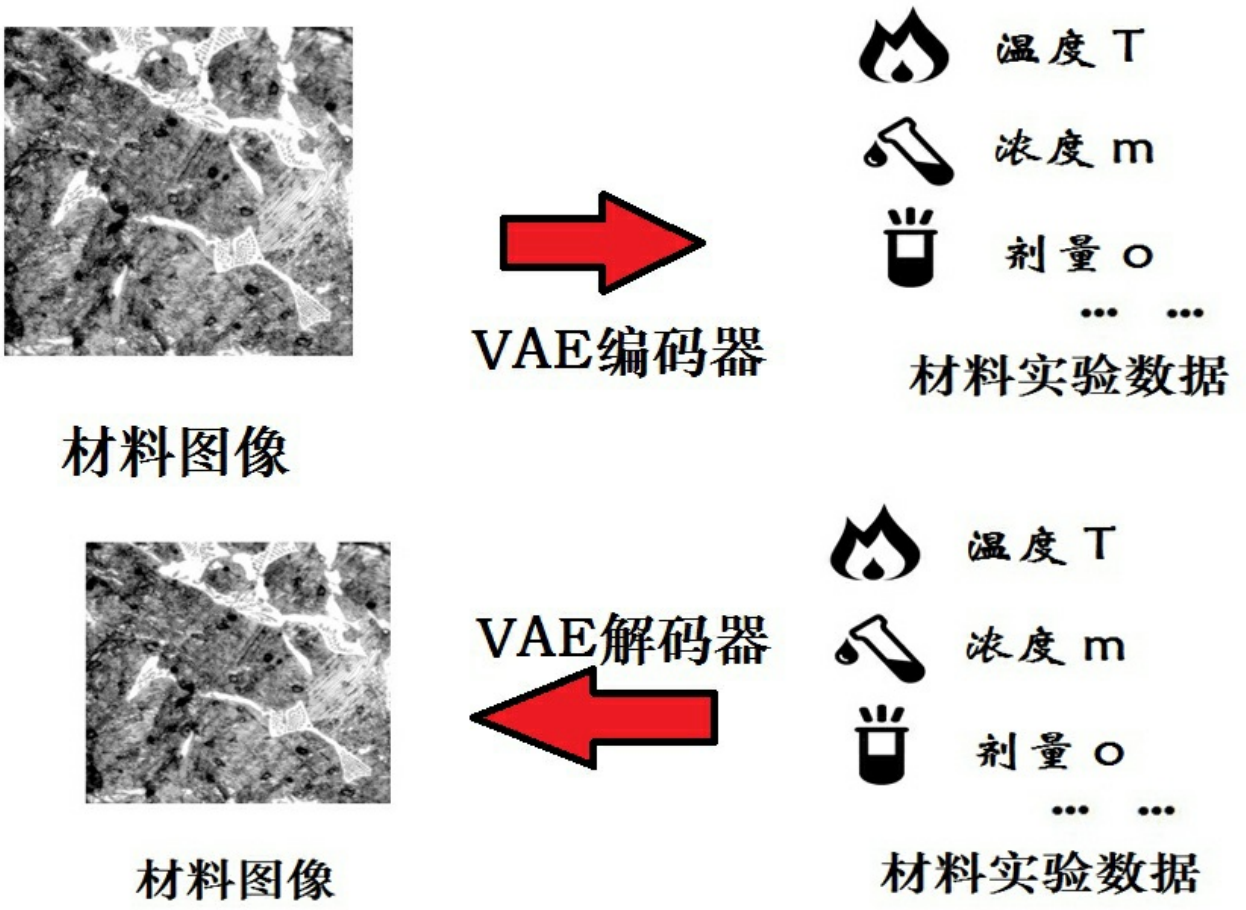
从辐照实验条件到材料辐照肿胀图像：运用变分自编码器VAE



变分自编码器VAE对材料辐照肿胀实验意味着什么？



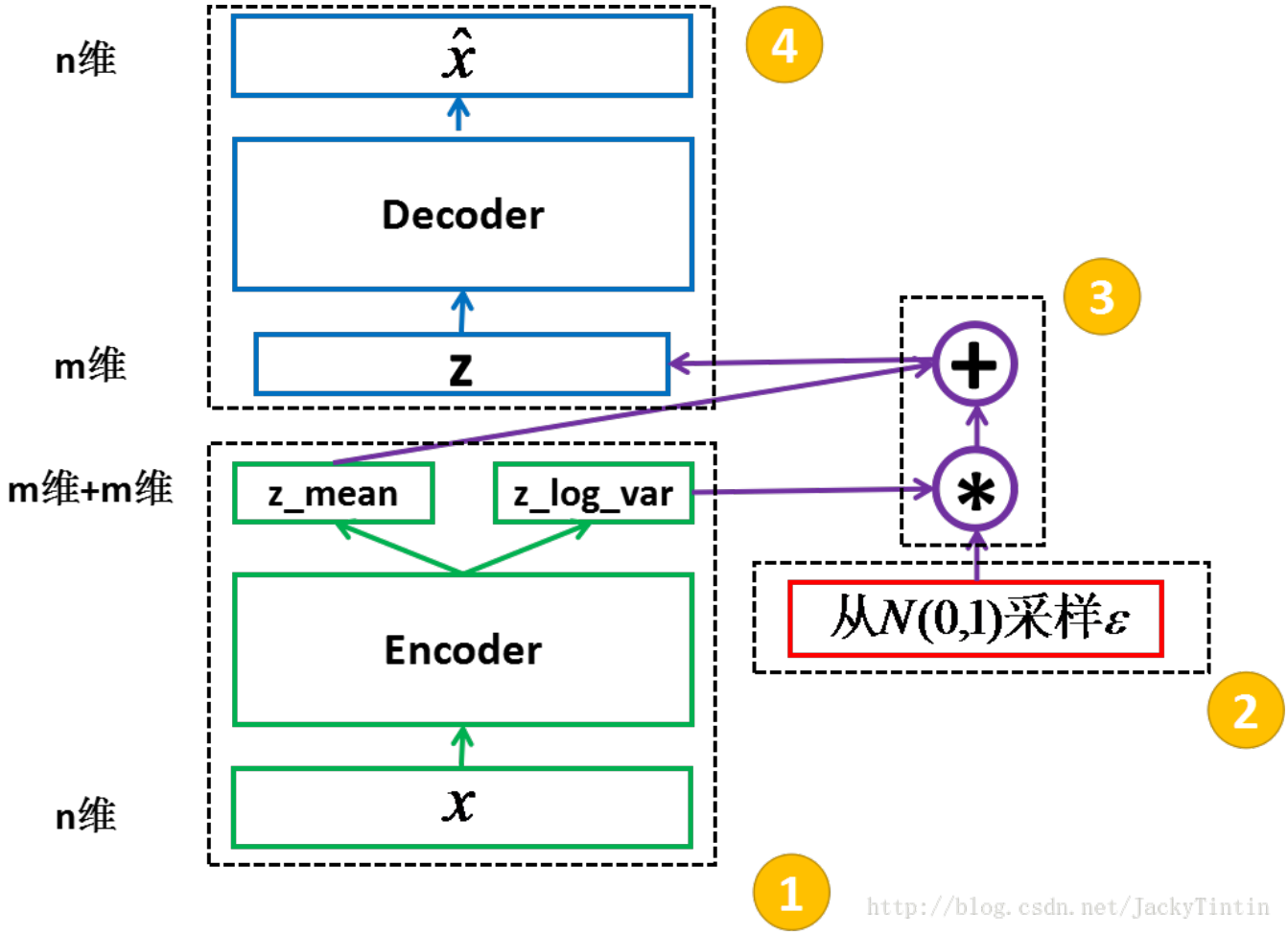
简单来说，其实就两个作用：

- 当**我们只知道实验参数**(辐照温度、剂量等)的时候，我们可以利用训练好的VAE模型直接由参数生成对应材料辐照图像；
- 当**我们只有实验图像**的时候，我们可以利用训练好的VAE模型直接由材料辐照图像计算对应的实验参数(辐照温度、剂量等)；

这样一来我们相当于解构了实验的因果，我们利用VAE，不光可以拟合一个形如$p(X_s|image)$的解释器(其中X_s是实验参数)，还可以构造一个形如$q(image|X_s)$的生成器；

什么是VAE?

变分自编码器(VAE)

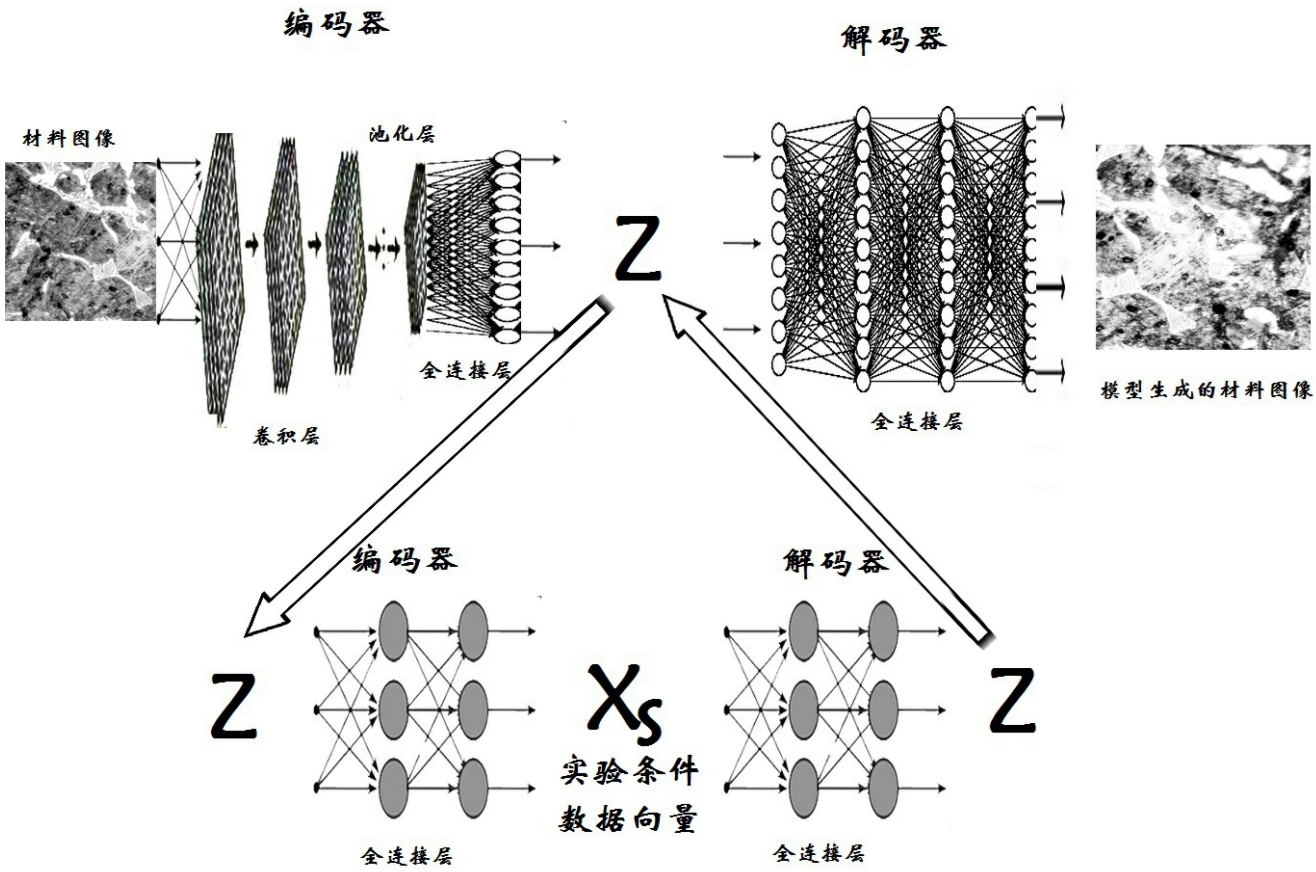


- **自编码器(autoencoder)** :是一种非监督式学习的神经网络模型，采用原始数据作为输入和输出，含有一个数量小于输入输出的隐藏层。
- **从输入到隐藏层**，这一段神经元数量下降，被称为”encoder”，也可以称为模式识别模型或判别模型；而从隐藏层到输出，这一段神经元数量上升，被称为”decoder”，也称生成模型。因而编码器在一定程度上是类似于GAN的
- **由于隐藏层数量小于输入**，所以会对数据进行压缩，之后输出神经元数量大于隐藏层，压缩后的隐藏层相互组合重现原始输出
- 它尝试使用一个函数，**将输入逼近输出**，经过训练之后，撤去输入层，仅凭隐藏层也可以实现重现输出
- 这样隐藏层就可以做到**提取主要成分的任务**。这样的功能实现了隐藏层对于数据特征的提取，类似于主成分分析(PCA)

也就是说，VAE可以看做编码器$encoder$和解码器$decoder$这两个模块的组合； 事实上

> “
>
> - 编码器$encoder$对应的就是将材料辐照图像编码为实验参数(辐照温度、剂量等)的过程;
> - 解码器$decoder$对应的就是将实验参数(辐照温度、剂量等)编码为材料辐照图像的过程;

构造嵌套的VAE来"解构"材料辐照实验



先来看看我们运用了哪些组件：

- **材料图像编码器encoder**:可以写作$P(Z|image)$，Z是一个维度很高的特征向量，也是材料图像编码器encoder编码出来的特征编码code；
- **实验数据编码器encoder**:可以写作$P(X_S|Z)$，X_S是实验参数(辐照温度、剂量等)组成的向量；
- **实验数据解码器decoder**:可以写作$Q(Z|X_S)$，将实验参数(辐照温度、剂量等)组成的向量映射到特征编码Z；
- **材料图像解码器decoder**:可以写作$Q(image'|Z)$，Z是特征编码，$image'$是我们的模型根据实验参数(辐照温度、剂量等)自己生成的材料图像；

程序全部采用keras编写；

材料图像编码器encoder

由卷积层、池化层加全连接层组成：

```
x = Conv2D(16, (3, 3), activation='relu', padding='same')(input_img)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(8, (3, 3), activation='relu', padding='same')(x)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(8, (3, 3), activation='relu', padding='same')(x)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Flatten()(x)
x = Dense(64, activation='relu')(x)
encoded = Dense(32, activation='relu')(x)
```

材料图像解码器decoder

由全连接层和卷积层、上采样层组成；

```
input_con = Input(shape=(32,))
x = Dense(64, activation='relu')(input_con)
x = Dense(128, activation='relu')(x)
x = Reshape((4, 4, 8))(x)
x = Conv2D(8, (3, 3), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
x = Conv2D(8, (3, 3), activation='relu', padding='same')(x)
x = UpSampling2D((2, 2))(x)
x = Conv2D(16, (3, 3), activation='relu')(x)
x = UpSampling2D((2, 2))(x)
decoded = Conv2D(1, (3, 3), activation='sigmoid', padding='same')(x)
```

实验数据编码器encoder

由全连接层组成：

```
input = Input(shape=(32,))
encoded = Dense(128, activation='relu')(input)
encoded = Dense(64, activation='relu')(encoded)
encoded = Dense(32, activation='relu')(encoded)
```

实验数据解码器decoder

由全连接层组成：

```
decoded = Dense(32, activation='relu')(encoded)
decoded = Dense(64, activation='relu')(decoded)
decoded = Dense(32, activation='relu')(decoded)
```

模型训练

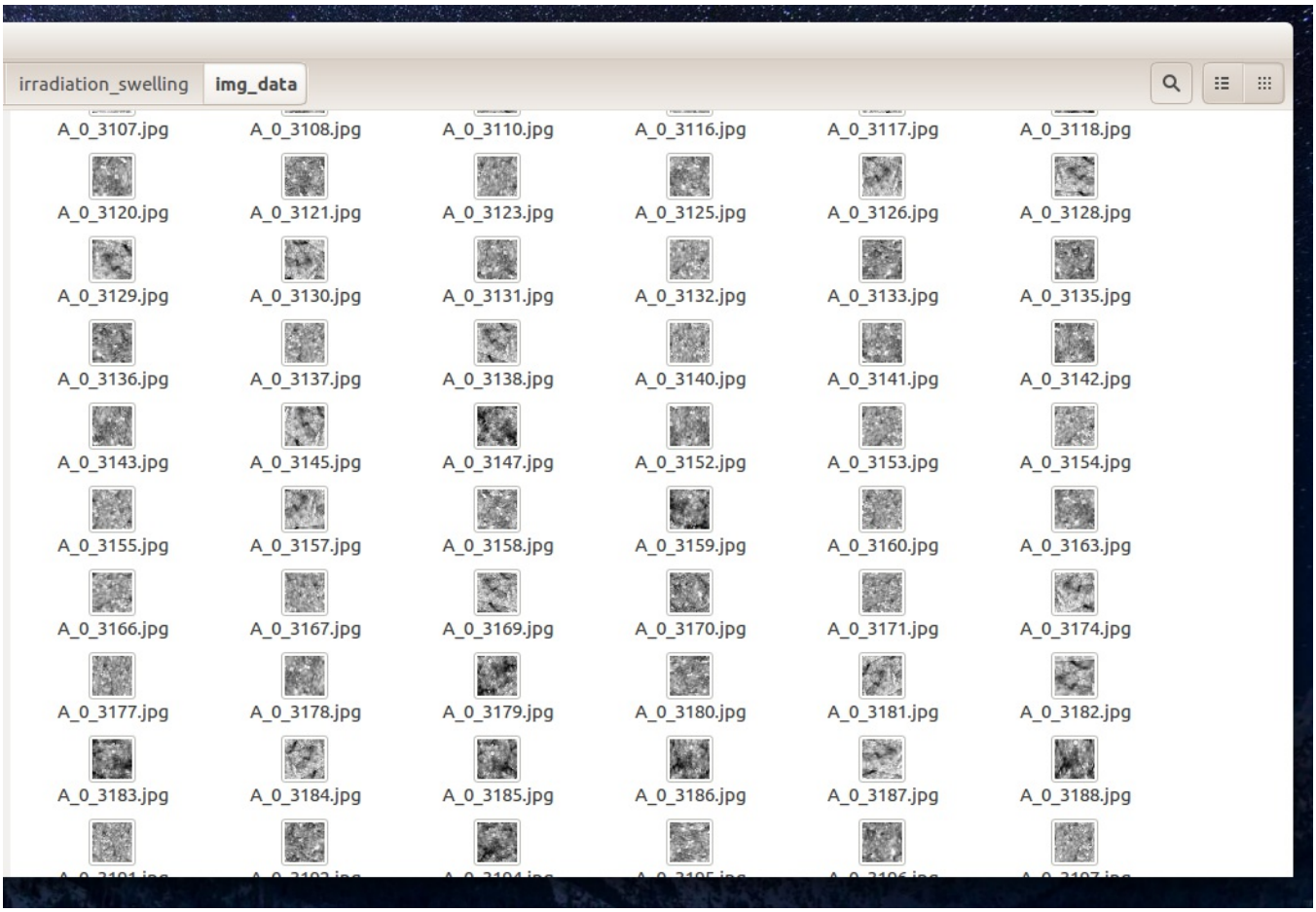
训练并存储模型：

```
encoder = Model(input_img, encoded)
encoder.compile(optimizer='adadelta', loss='binary_crossentropy')

encoder.fit(x_train, y_train,
                epochs=5,
                batch_size=256,
                shuffle=True,
                validation_data=(x_train[0:100], y_train[0:100]))
encoder.save('./models/model_ir.h5')
```

训练详情

首先我采用了图像变换函数，将为数不多的材料实验图像，生成了28000张规格为32*32的用于训练的材料辐照图像； 图像生成器语句：

```
from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_
import PIL

datagen = ImageDataGenerator(
          rotation_range=40,
          width_shift_range=0.2,
          height_shift_range=0.2,
          shear_range=0.2,
          zoom_range=0.2,
          horizontal_flip=True,
          fill_mode='wrap');
```



最总我们得到了拟合效果不错的材料辐照图像(这里是二值化的)：