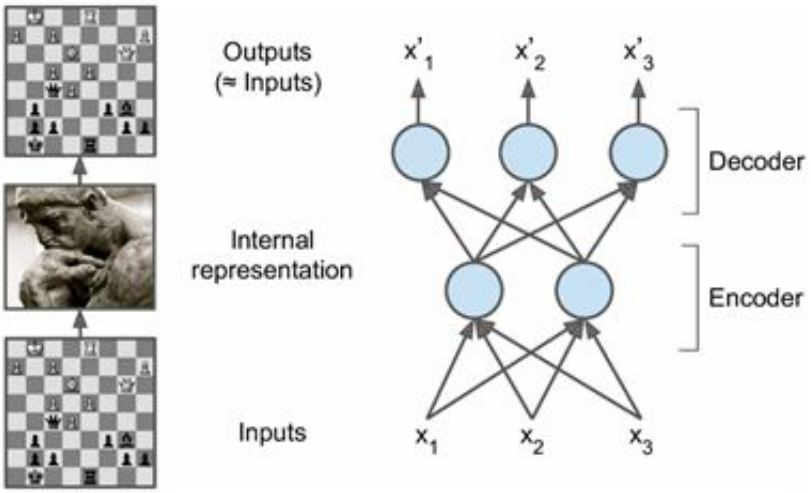
Report 4

自动编码器

自动编码器（Auto-Encoder，简称AE）是一种特殊的（无监督）神经网络模型，用于特征提取和数据降维。最简单的自动编码器由一个输入层，一个隐含层，一个输出层组成。隐含层的映射充当编码器，输出层的映射充当解码器。



编码器：数据x => 隐变量z 学习到输入数据的隐含特征

解码器：隐变量z => 数据x 用学习到的新特征可以重构出原始输入数据

用途：1. 数据降维。类似PCA，但基于神经网络的自动编码器比PCA性能更强。

2. 特征提取。把自动编码器的输出的新特征输入到有监督学习模型中。

自动编码器是将输入 $\boldsymbol{x}$ 进行编码，得到新的特征 $\boldsymbol{y}$ ，并且希望原始的输入 $\boldsymbol{x}$ 能够从新的特征 $\boldsymbol{y}$ 重构出来。编码过程如下：

$$y = f(Wx + b)$$

可以看到，和神经网络结构一样，其编码就是线性组合之后加上非线性的激活函数。如果没有非线性的包装，那么自动编码器就和普通的PCA没有本质区别了。利用新的特征 $\boldsymbol{y}$ ，可以对输入 $\boldsymbol{x}$ 重构，即解码过程：

$$x' = f\left(W'x + b'\right)$$

编码器和解码器同时训练，训练的目标是最小化重构误差。我们希望重构出的 x' 和**尽可能一致**，可以采用最小化负对数似然的损失函数来训练这个模型：

$$L = -\log P\left(x|x'\right)$$

对于高斯分布的数据，采用均方误差就好，而对于伯努利分布可以采用交叉熵，这个是可以根据似然函数推导出来的。有时候，我们还会给自动编码器加上更多的约束条件，去噪自动编码器以及稀疏自动编码器就属于这种情况，因为大部分时候单纯地重构原始输入并没有什么意义，我们希望自动编码器在近似重构原始输入的情况下能够捕捉到原始输入更有价值的信息。

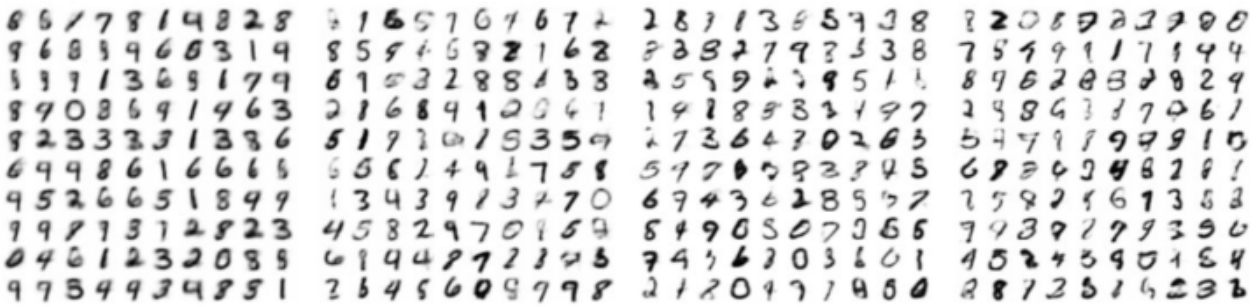
> 另一种训练时优化的目标函数：
>
> $$\min \frac{1}{2l} \sum_{i=1}^{l} \left\| x_i - g_{\theta'}\left(h_\theta\left(x_i\right)\right) \right\|_2^2$$
>
> 这里采用了欧氏距离损失。其中为训练样本数， θ 和 θ' 是分别是编码器和解码器要确定的参数。

训练完成之后，在预测时只使用编码器而不再需要解码器，编码器的输出结果被进一步使用，用于分类，回归等任务。

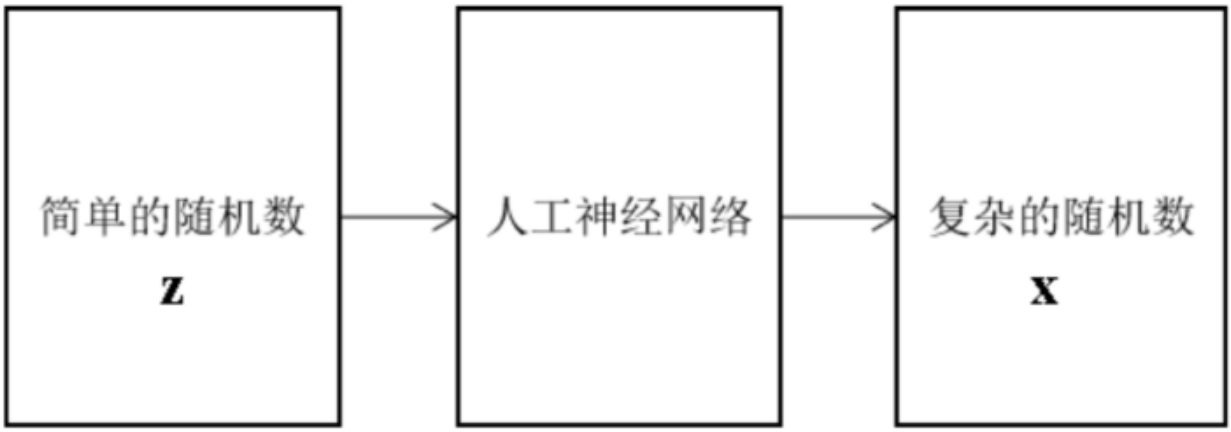
变分自动编码器

变分自动编码器（Variational Auto-Encoder，简称VAE）是一种深度生成模型，用于生成图像，声音之类的数据，类似于生成对抗网络（GAN）。**虽然也叫自动编码器，但和标准的自动编码器有很大的不同，二者用于完全不同的目的。**

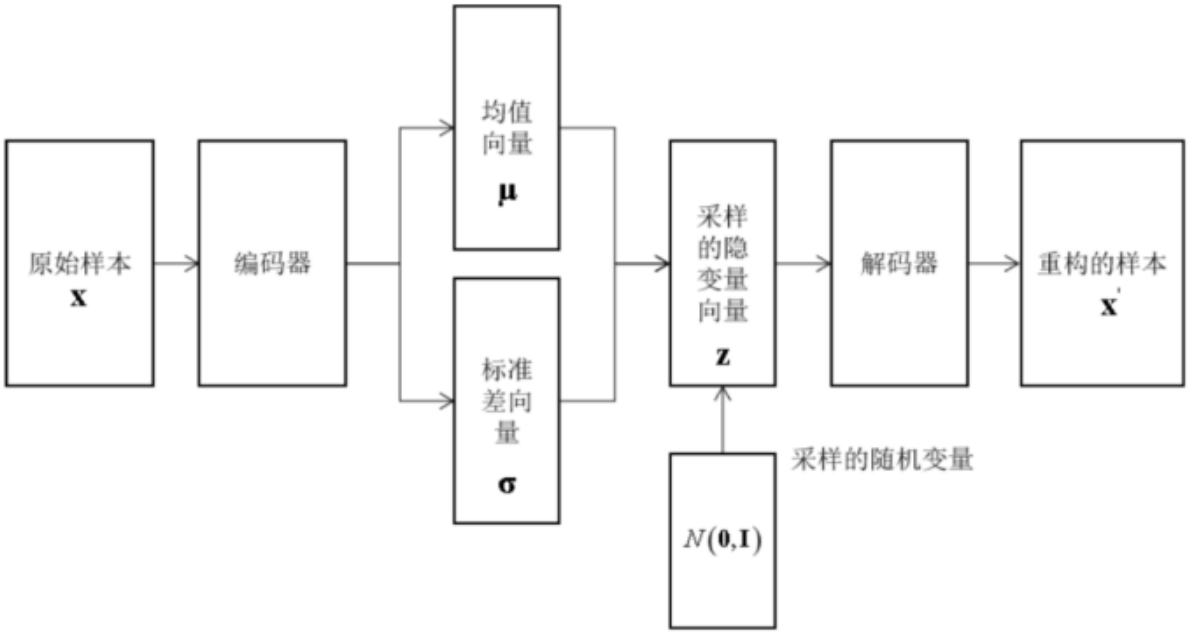
考虑数据生成问题，如写字，最简单的是写出MNIST数据集这样的手写数字



如果我先收集一些训练样本，然后让算法原样输出它们，当然也可以完成写字，但这样生成的样本完全就没用多样性了。因此一般的解决思路是先生成一些随机数，然后对其进行变换，生成我们想要的复杂的样本数据。这一过程如下图所示



问题是，这个简单的随机数该怎么生成？应该服从何种概率分布？它们可以代表笔画特征，字体宽度，清晰角度，字体类型等信息，但如果这样人工设计，一是费时费力，二是具有局限性。因此我们会想到：能不能从文字图像中先学习这些特征，然后对这些特征进行随机扰动，生成新的样本？变分自动编码器就采用了这种思路。其结构如下图所示



这里的隐变量可以看做是从图像中学习得到的特征。同样的，编码器和解码器同时训练，训练时的目标是下面方程的右端

$$\log p(x) - D_{KL}\left[q(z|x)\right| |p(z|x)] = E_{z\sim q}\left[\log p(x|z)\right] - D_{KL}\left[q(z|x)||p(z)\right]$$

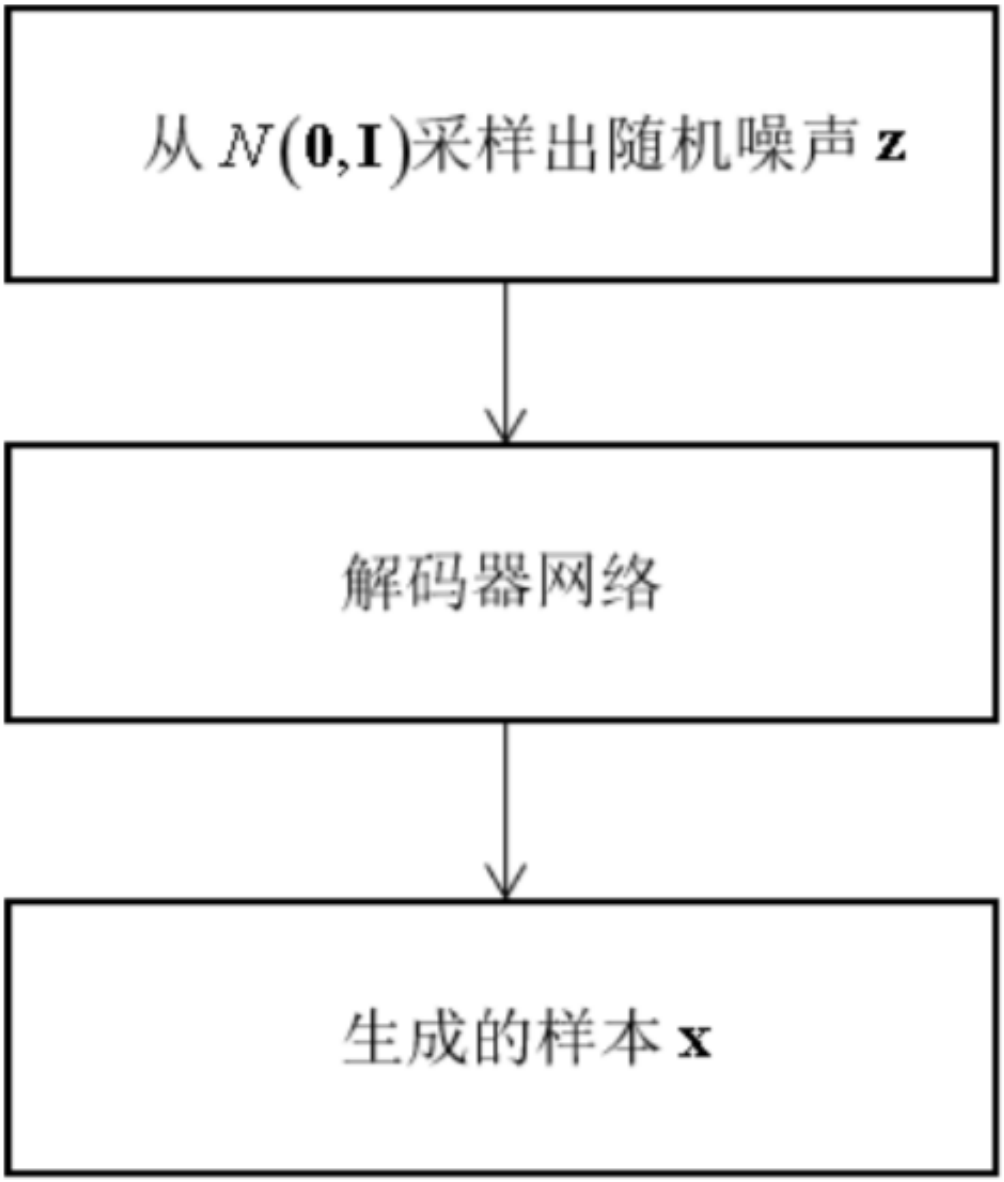
> VAE原理推导前瞻：

> 为了求解真实的后验p(z|x)概率分布，VAE引入一个识别模型q(z|x)去近似p(z|x)，那么衡量这两个分布之间的差异自然就是相对墒了，也就是KL散度，VAE的目的就是要让这个相对墒越小

由于原理较为复杂，在这里不做深究。训练完成之后，预测阶段可以直接生成样本。具体做法是，从正态分布

$$Z \sim N(0, I)$$

产生一个随机数，然后送入解码器中，得到预测结果，即为生成的样本。此时已经不再需要编码器。



总结：

AE用于提取特征和数据降维，最后保留的是编码器。

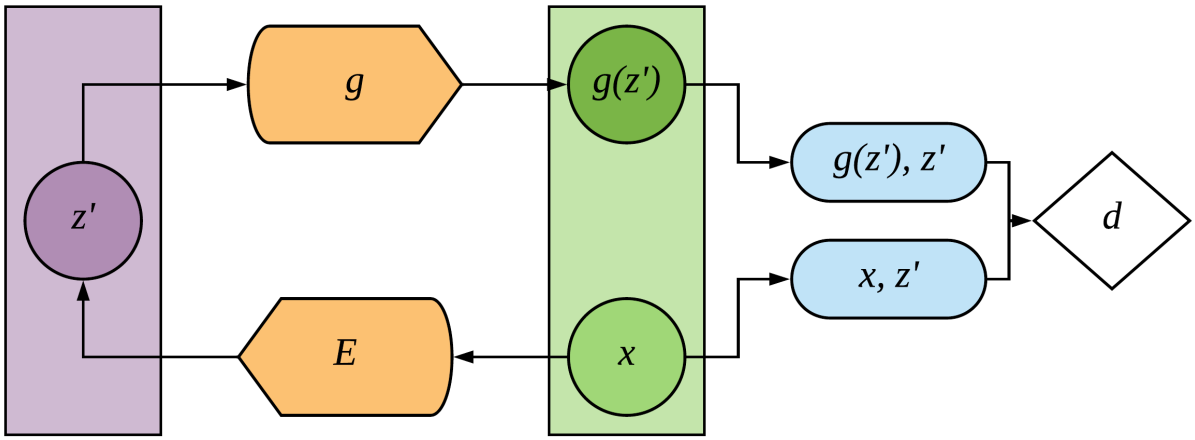
VAE属于生成模型，用于生成数据，最后保留的是解码器。

VAE也可以按照AE的使用方法进行特征提取和数据降维，但是AE不能像VAE一样用于生成数据。理由如下：

变分自动编码机的目的是想知道观测数据x背后的潜在变量z分布。VAE的解决方案是把真实数据x对应的潜在分布映射到一个先验分布上。所以只需要对这个先验分布进行采样，即可生成新的样本。

而AE不行，因为AE中每个x被固定编码为对应的z，无法知道z的分布（若此时我们知道了z的分布，就等于知道了真实数据x的分布，这显然是不可能的）。若AE硬要获得新样本，此时只能随机采样z，很显然我们无法验证：根据这个z是否能正确地还原出一个符合真实样本x的新样本。

回顾Efficient GAN-based AD：



隐式空间左侧实际上就是一个AE的结构。

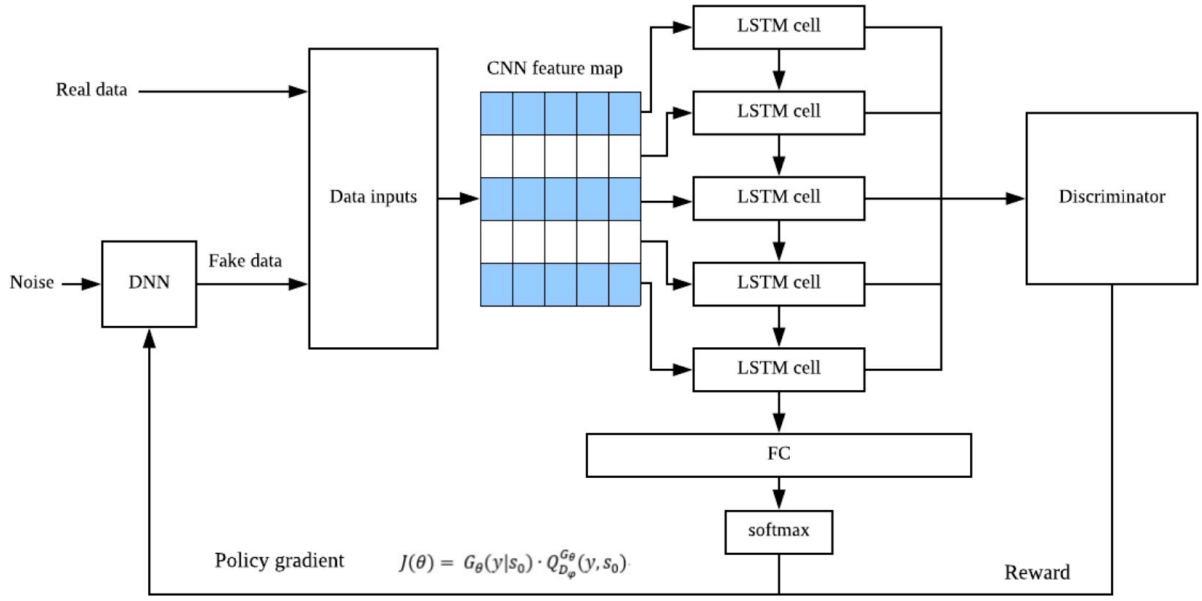
网络结构源码：

```
    gen = network.decoder
    enc = network.encoder
    dis = network.discriminator
```

Encoder源码：

```python
def encoder(x_inp, is_training=False, getter=None, reuse=False):
    """ Encoder architecture in tensorflow

    Maps the data into the latent space

    Args:
        x_inp (tensor): input data for the encoder.
        reuse (bool): sharing variables or not

    Returns:
        (tensor): last activation layer of the encoder

    """

    with tf.variable_scope('encoder', reuse=reuse, custom_getter=getter):

        name_net = 'layer_1'
        with tf.variable_scope(name_net):
            net = tf.layers.dense(x_inp,
                                  units=64,
                                  kernel_initializer=init_kernel,
                                  name='fc')
            net = leakyReLu(net)

        name_net = 'layer_2'
        with tf.variable_scope(name_net):
            net = tf.layers.dense(net,
                                  units=latent_dim,
                                  kernel_initializer=init_kernel,
                                  name='fc')

    return net
```

模型思考



1. C-LSTM用于特征提取，删除FC和softmax以及返回DNN的通路。并删除强化学习相关的内容。
2. ~~删除DNN Discriminator，直接以C-LSTM作为判别器。并删除强化学习相关的内容。~~

   方案二无法对滑窗进行检测，只具备对整个序列检测的能力。