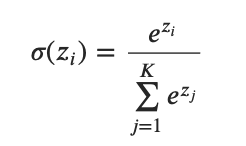
Gumbel-Softmax Trick

1. Gumbel-softmax Trick用在哪里‘

在强化学习中，如果动作空间是离散的，比如上下左右四个动作，通常的做法是网络输出一个四维的one-hot向量（不考虑空动作），分别代表四个动作。比如[1,0,0,0]代表上，[0,1,0,0]代表下等。而具体取哪个动作，就根据输出的每个维度的大小，选择值最大的作为输出动作，即arg max(v)。

例如网络输出的四维向量为v = [-20, 10, 9.6, 6.2]，第二个维度取到最大值为10，那么输出的动作就为[0,1,0,0]，也就是下，这和多分类的分类任务是一个道理。但是这种取法有个问题是不能计算梯度，也就不能更新网络。通常的做法是加softmax函数，把向量归一化，这样既能计算梯度，同时值的大小还能表示概率的含义。Softmax函数定义如下：



那么将v = [-20, 10, 9.6, 6.2]通过softmax函数后有=[0, 0.591, 0.396, 0.013]，这样做不会改变动作或者说类别的选取，同时softmax倾向于让最大值的概率显著大于其他值，比如这里10和9.6经过softmax放缩之后变成了0.591和0.396，这样有利于把网络训练成一个one-hot输出的形式，这种方式在分类问题中是常用的方法。

但是这么做还有一个问题，这个表示概率的向量=[0, 0.591, 0.396, 0.013]并没有真正显示出概率的含义，因为一旦某个值最大，就选择相应的动作或者分类。

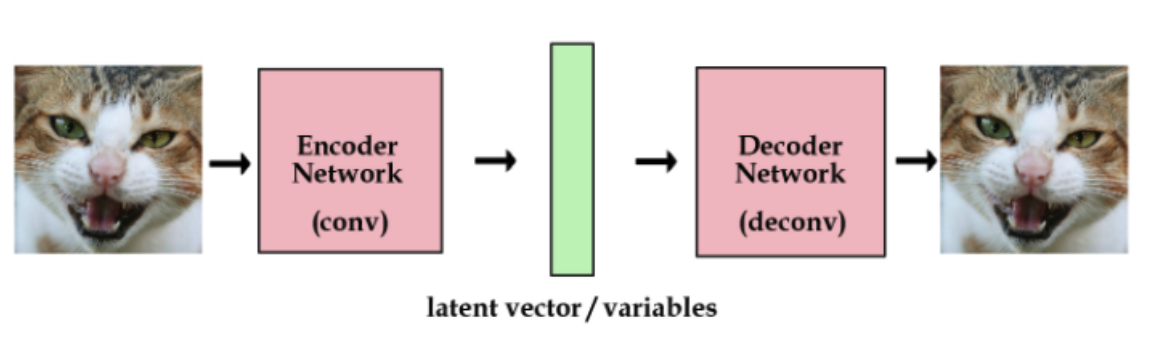
很直接的方法就是依概率采样就行了，比如直接用np.random.choice函数依照概率生成样本值，这样概率就有意义了，但是又有一个问题：这种方式怎么计算梯度？不能计算梯度怎么用BP的方式更新网络？

这时重参数（re-parameterization）技巧解决了这个问题。简单来说重参数技巧的一个用处就是把采样的步骤移出计算图，这样整个图就可以计算梯度BP更新了。

以前做分类任务直接softmax之后BP更新就可以了，为什么非要采样，后来看了VAE和GAN之后明白，还有很多需要采样训练的任务。这里举简单的VAE（变分自编码器）的例子说明需要采样训练的任务以及重参技巧。

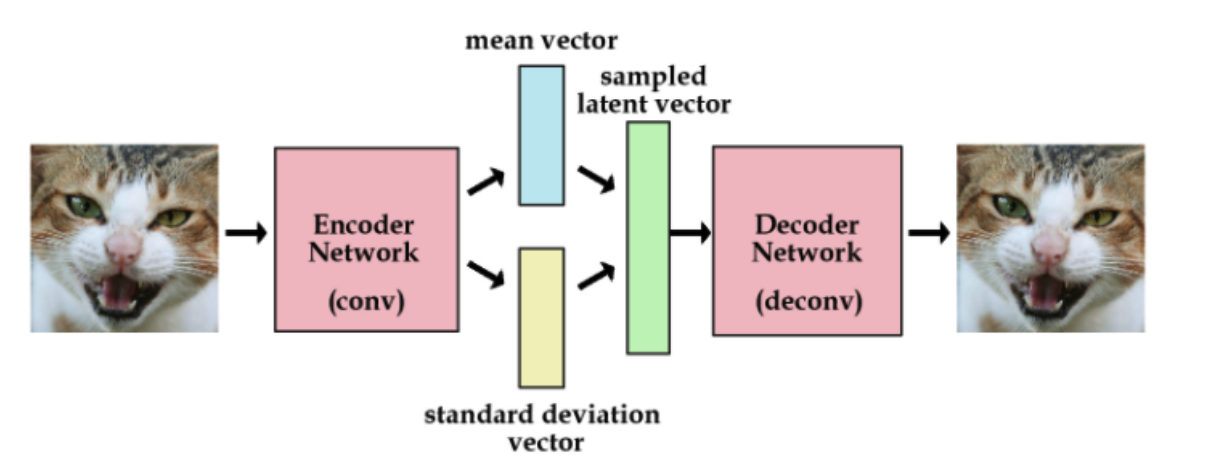
**Re-parameterization trick**

最原始的自编码器通常是这样的：



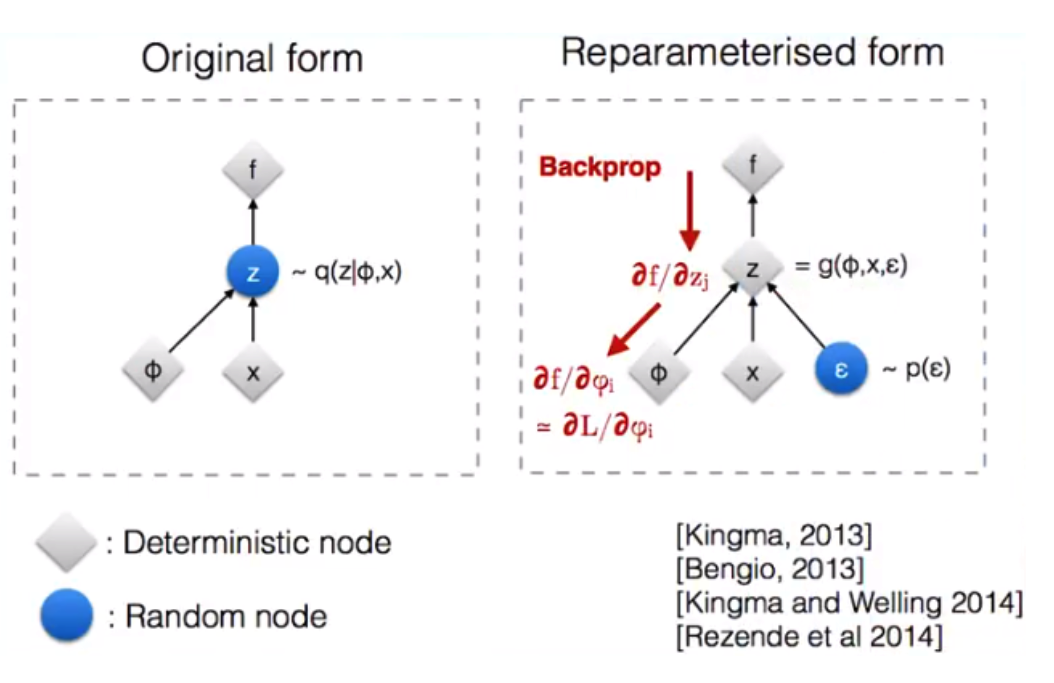
左右两边都是端到端的输入输出网络，中间的绿色是提取的特征向量，这是一种直接从图片提取特征的方式。

而VAE长这样：



VAE的想法是不直接用网络去提取特征向量，而是提取这张图像的分布特征，也就是把绿色的特征向量替换为分布的参数向量，比如说均值和标准差。然后需要decode图像的时候，就从encode出来的分布中采样得到特征向量样本，用这个样本去重建图像，这时怎么计算梯度的问题就出现了。

重参数技巧可以解决这个问题，它长下面这样：



假设图中的x和表示VAE中的均值和方差，他们是确定的节点。而需要输出的样本z是带有随机性的节点，重参数就是把带有随机性的z变成确定性的节点，同时随机性用另一个输入节点代替。例如，这里用正态分布采样，原本从均值x和标准差为的正太分布N(x, )中采样能得到z。将其转化为从标准正态分布N(0,1)中采样得到，再计算得到z=x + \*。这样一来，采样的过程就移出了计算图，整张计算图就可以计算梯度进行更新了，而新加入的的输出分值不做更新，只当成一个没有权重变化的输入。

**Gumbel-softmax Trick**

VAE的例子是一个连续分布（正态分布）的重参数，离散分布的情况也一样，首先需要采样，使得离散的概率分布有意义而不是直接取概率最大的值，其次需要可以计算梯度。那么怎么做到的呢，具体操作如下：

对于n维概率向量，对对应的离散随机变量添加Gumbel噪声，再取样：



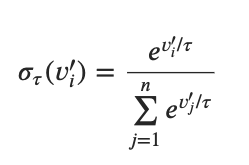
其中，是独立同分布的标准Gumbel分布的随机变量，标准Gumbel分布的表达式为。

这个是Gumbel-max Trick。可以看到由于这个中间有一个arg max操作，这是不可导的，所以用softmax函数代替它，也就是Gumbel-softmax trick，而可以通过Gumbel分布求逆从均匀分布生成，即：



Gumbel softmax具体的实践是这样操作的：

* 对于网络输出的一个n维向量v，生成n个服从均匀分布U(0,1)的独立样本
* 通过计算得到
* 对应相加得到新的值向量
* 通过softmax函数：



计算概率大小得到最终的类别，其中是温度参数。

直观上感觉，对于强化学习来说，在选择动作之前加一个扰动，相当于增加探索度，感觉上式合理的，对于深度学习的任务来说，添加随机性去模拟分布的样本生成，也是合情合理的。