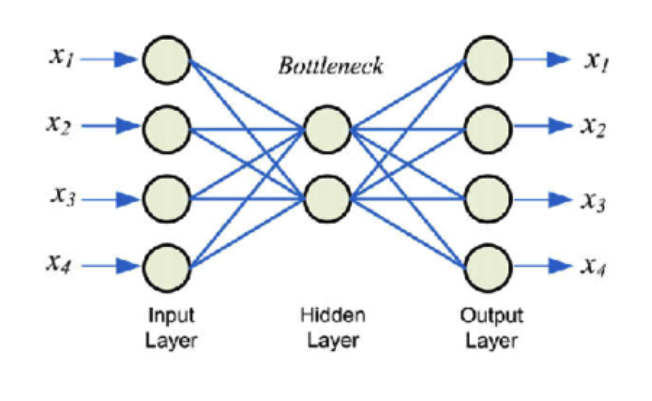
VAE

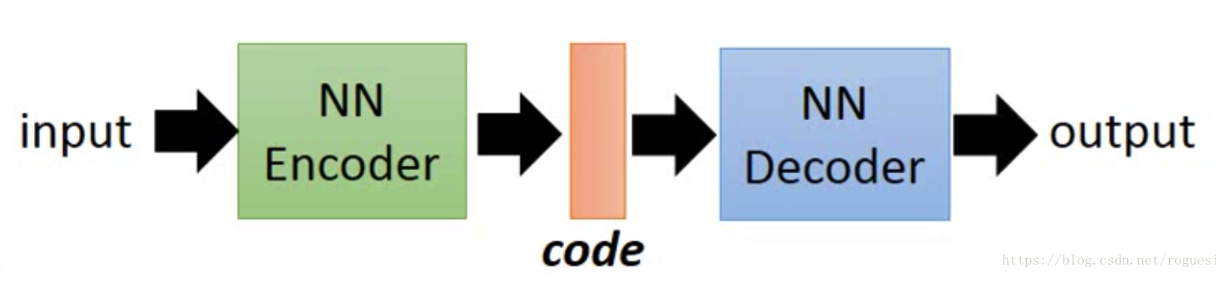
VAE(variational autoencoder)就是指Auto Encoder的编码输出服从正态分布。

**Auto Encoder**

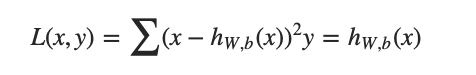
Auto Encoder(后面用AE简称)是一个自动编码器，它是通过重建输入的神经网络训练过程，它的隐藏层的向量具有降维的作用。它的特点是编码器会创建一个隐藏层（或者多个隐藏层）包含了输入数据含义的低维向量，然后有一个解码器，会通过隐藏层的低维向量重建输入数据。通过神经网络的训练最后AE会在隐藏层中得到一个代表输入数据的低维向量。它可以帮助数据分类、可视化、存储。AE是一个自动编码器是一个非监督的学习模式，只需要输入数据，不需要label或者输入输出对的数据。



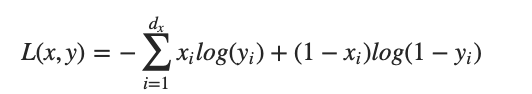
AutoEncoder的基本原理是通过一个encoder和一个decoder实现的，在卷积的Autoencoder中，encoder通过卷积池化进行降维，decoder通过反卷积反池化进行升维，目标是使输入和输出图片的差异性最小，这也使得自编码器生成的图片多余输入相似，无法生成新的图片，下图是Autoencoder的结构示意图：



虽然AE是一个非监督学习算法，但如果它的解码器是线性重建数据，可以用MSE来表示它的损失函数：



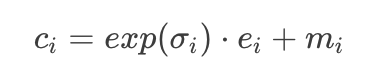
如果解码器用sigmoid的激活函数，那主要用交叉熵损失函数：



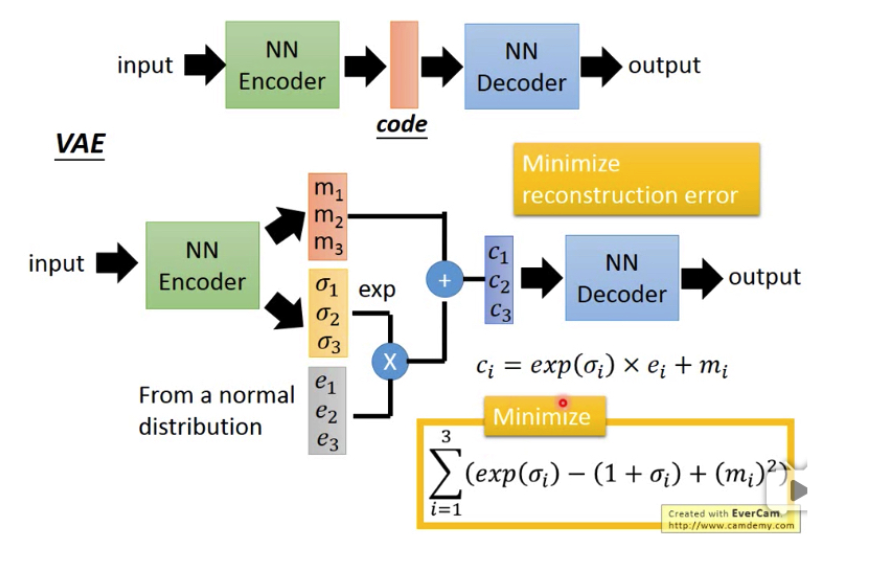
VAE

VAE(variational autoencoder)和普通的auto encoding相比，VAE的encoder会输出两个向量，可以把其中的一个看做mean,另一个看做variance.同时还要加入一个误差error【也可以视为噪声noise】，这个error是从一个高斯分布sample出来的，最终把这3个向量合成一个code【编码】。

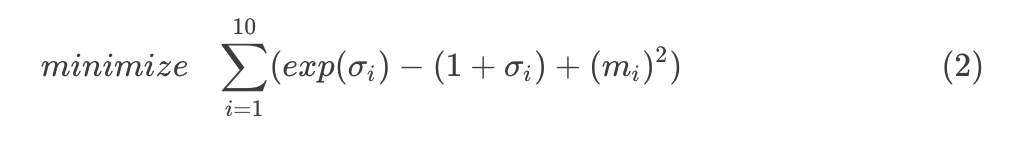
VAE的具体实现过程如下：与AE相比，encoder过程和decoder过程都不变，在中间加入了一些trick, encoder输出两个vector，假设原本AE中生成的coder是十维，生成的两个即(,)和(),之后再从正太分布(normalize distribution)中生成一个vector(),执行计算如下：



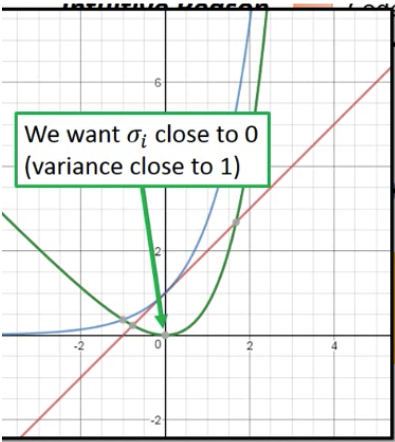
从而生成(),再把()丢进decoder中得到输出，通过“使input和output越接近越好”这个目标来优化模型。



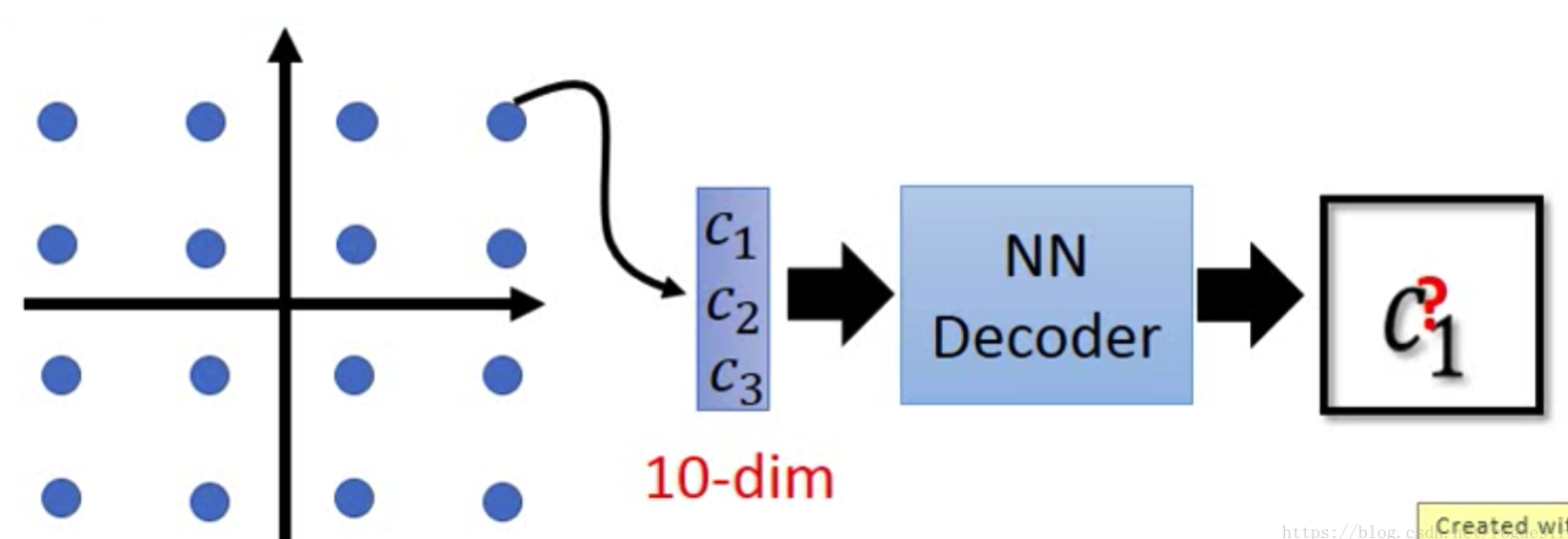
上面说的noise，就是上面示意图中的乘积部分：m对应于原来AE中的code, 是从输入图片中生成的，但他表示noise的variance， e是从正态分布中抽样得到的，二者相乘即得到了影响code的数值区间的noise, variance影响了noise的大小，而且是有encoder训练得到的，如果只考虑“使Input和output越接近越好”这个目标来优化模型，那训练得到的variance为0时候模型理论最佳，也就变成了原来的AE。因此需要对这个计算过程做一个限制，如下：



如下图所示，其中exp()的图像如蓝色所示，1+的图像如下图红色部分所示，他们的差值是绿色图像，即当为0时，二者的差值会取得最小值，因此需要加上一个L2正则项，强迫这个variance不能太小。



VAE过程的一个优势就是：可以通过调整code中的某维来确定code的维度所代表的东西，假设得到的code是十维的向量，我们控制其中的八维，来讨论剩下的两维，如下图所示，现在坐标系中均匀采点，然后把取样的点连同固定的八维一起丢到decoder中，这样就可以观察这两维对于VAE生成图片的作用效果。



如下图所示，分别为人脸的生成映射和MNIST数据集上固定其中两维进行图像生成，从左侧的人脸图片中，可以猜测两维特征可能分别控制着人脸的角度和表情；右侧MNIST数据集中，两维特征可能控制着数字中笔画的弯曲程度和数字的倾斜程度。

