

一、论文一： Denoising Auto-Encoder

1.Auto-Encoder

1.自编码器通过输入 d 维向量 x ，经过函数 $y = f(x) = s(Wx + b)$ 映射为 d' 维向量，然后再通过函数 $g(y) = s(W'y + b')$ 映射为与 x 维度一样的向量，然后通过损失函数计算训练两者误差，从而达到数据特征提取、压缩、还原的效果。

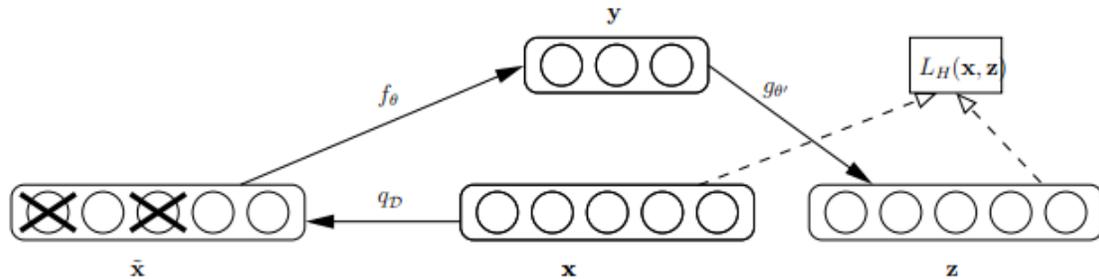
- 损失函数如下：

$$\begin{aligned}\theta^*, \theta'^* &= \arg \min_{\theta, \theta'} \frac{1}{n} \sum_{i=1}^n L(x^{(i)}, z^{(i)}) \\ &= \arg \min_{\theta, \theta'} \frac{1}{n} \sum_{i=1}^n L(x^{(i)}, g_{\theta'}(f_{\theta}(x^{(i)})))\end{aligned}$$

LossF

2.Denoising Auto-Encoder

Pascal Vincent 论文中关于 dAE 的示意图，如下：



数据1

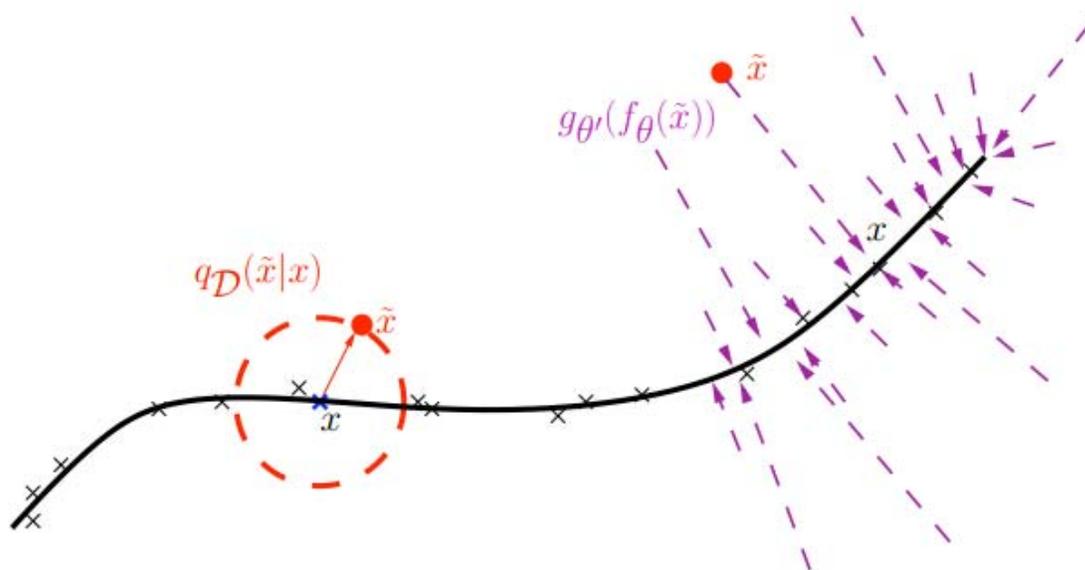
由上图可知，样本 x 按照 q_D 分布加入随机噪声后变为 \tilde{x} ，按照文章的意思，这里并不是加入高斯噪声，而是以一定概率使输入层节点的值清为 0 。此时输入到可视层的数据变为 \tilde{x} ，隐含层输出为 y ，然后由 y 重构 x 的输出 z ，注意此时这里不是重构 \tilde{x} ，而是 x 。

作者对 dAE 的直观解释为:

- dAE 有点类似人体的感官系统，比如人眼看物体时，如果物体某一小部分被遮住了，人依然能够将其识别出来，
- 多模态信息输入人体时（比如声音，图像等），少了其中某些模态的信息有时影响也不大。
- 普通的 autoencoder 的本质是学习一个相等函数，即输入和重构后的输出相等，这种相等函数的表示有个缺点就是当测试样本和训练样本不合同一分布，即相差较大时，效果不好，明显，dAE 在这方面的处理有所进步。

作者也从数学上给出了一定的解释。

1. 流形学习的观点。一般情况下，高维的数据都处于一个较低维的流形曲面上，而使用 dAE 得到的特征就基本处于这个曲面上，如下图所示。而普通的 autoencoder，即使是加入了稀疏约束，其提取出的特征也不是都在这个低维曲面上（虽然这样也能提取出原始数据的主要信息）。



数据 1

2. 自顶向下的生成模型观点的解释。

3. 信息论观点的解释。

4. 随机法观点的解释。

与 Dropout 的区别

当在训练深度网络时，且采用了无监督方法预训练权值，通常，Dropout 和 Denoise Autoencoder 在使用时有一个小地方不同：Dropout 在分层预训练权值的过程中是不参与的，只是后面的微调部分引入；而 Denoise Autoencoder 是在每层预训练的过程中作为输入层被引入，在进行微调时不参与。另外，一般的重构误差可以采用均方差的形式，但是如果输入和输出的向量元素都是位变量，则一般采用交叉熵来表示两者的差异。

3. 实验结果

对比不同方法在不同数据上的实验结果

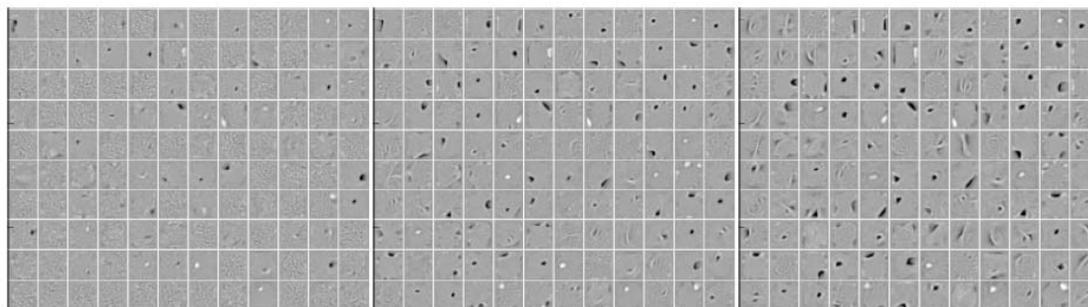
Table 1. Comparison of stacked denoising autoencoders (SdA-3) with other models.

Test error rate on all considered classification problems is reported together with a 95% confidence interval. Best performer is in bold, as well as those for which confidence intervals overlap. SdA-3 appears to achieve performance superior or equivalent to the best other model on all problems except *bg-rand*. For SdA-3, we also indicate the fraction ν of destroyed input components, as chosen by proper model selection. Note that SAA-3 is equivalent to SdA-3 with $\nu = 0\%$.

Dataset	SVM _{rbf}	SVM _{poly}	DBN-1	SAA-3	DBN-3	SdA-3 (ν)
<i>basic</i>	3.03±0.15	3.69±0.17	3.94±0.17	3.46±0.16	3.11±0.15	2.80±0.14 (10%)
<i>rot</i>	11.11±0.28	15.42±0.32	14.69±0.31	10.30±0.27	10.30±0.27	10.29±0.27 (10%)
<i>bg-rand</i>	14.58±0.31	16.62±0.33	9.80±0.26	11.28±0.28	6.73±0.22	10.38±0.27 (40%)
<i>bg-img</i>	22.61±0.37	24.01±0.37	16.15±0.32	23.00±0.37	16.31±0.32	16.68±0.33 (25%)
<i>rot-bg-img</i>	55.18±0.44	56.41±0.43	52.21±0.44	51.93±0.44	47.39±0.44	44.49±0.44 (25%)
<i>rect</i>	2.15±0.13	2.15±0.13	4.71±0.19	2.41±0.13	2.60±0.14	1.99±0.12 (10%)
<i>rect-img</i>	24.04±0.37	24.05±0.37	23.69±0.37	24.05±0.37	22.50±0.37	21.59±0.36 (25%)
<i>convex</i>	19.13±0.34	19.82±0.35	19.92±0.35	18.41±0.34	18.63±0.34	19.06±0.34 (10%)

results1

随着噪声比例的增加，网络提取特征的能力不断增强



(a) No destroyed inputs

(b) 25% destruction

(c) 50% destruction

results2

二、论文二：用 CDAE 对 ECG 信号去噪

1.网络结构

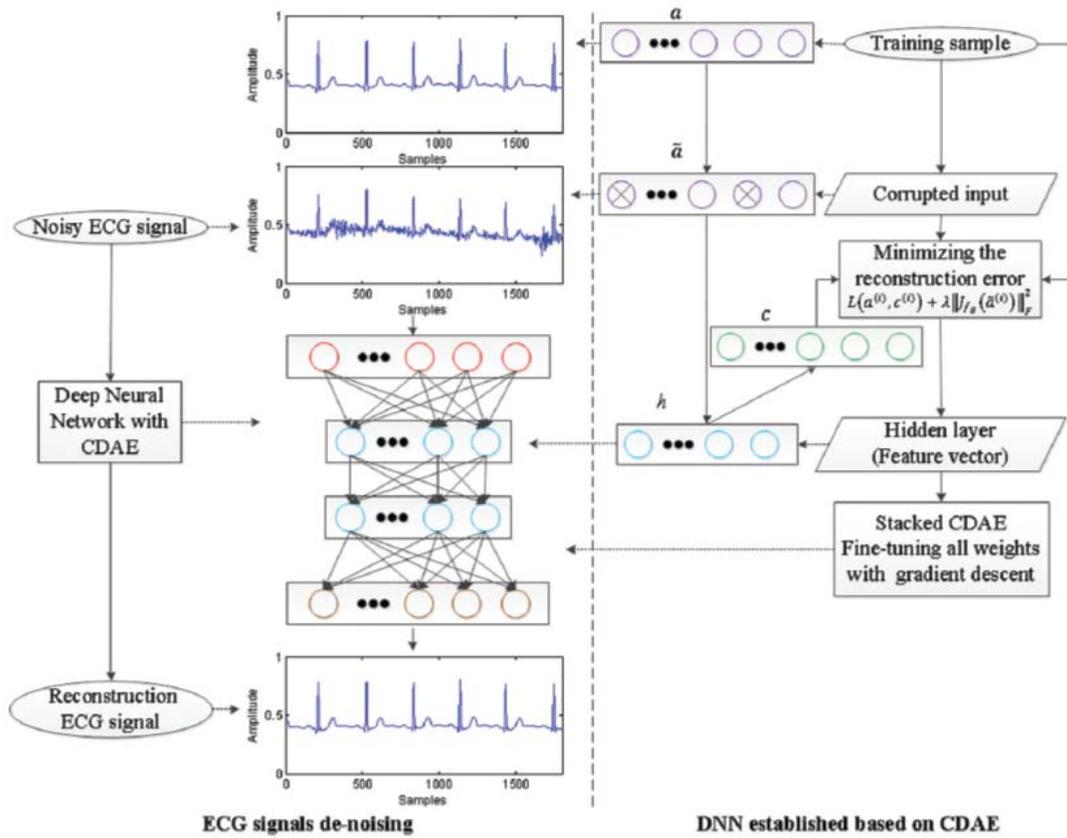


Figure 1. Framework of the proposed method.

structure2

- Stacked 初始化

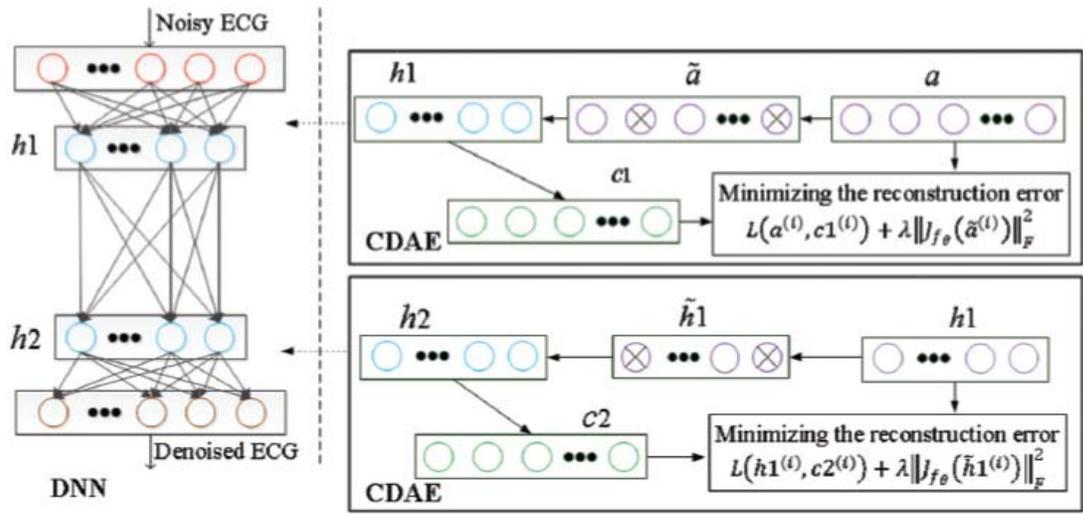


Figure 3. The DNN based on the stacked CDAE.

structure1

2.此篇论文基本采用去噪编码器的方法对心电信号进行去噪，唯一一点不同的是损失函数中加入了一项惩罚：**(01f01fj.. is a term to penalize its sensitivity to the training input a~)**

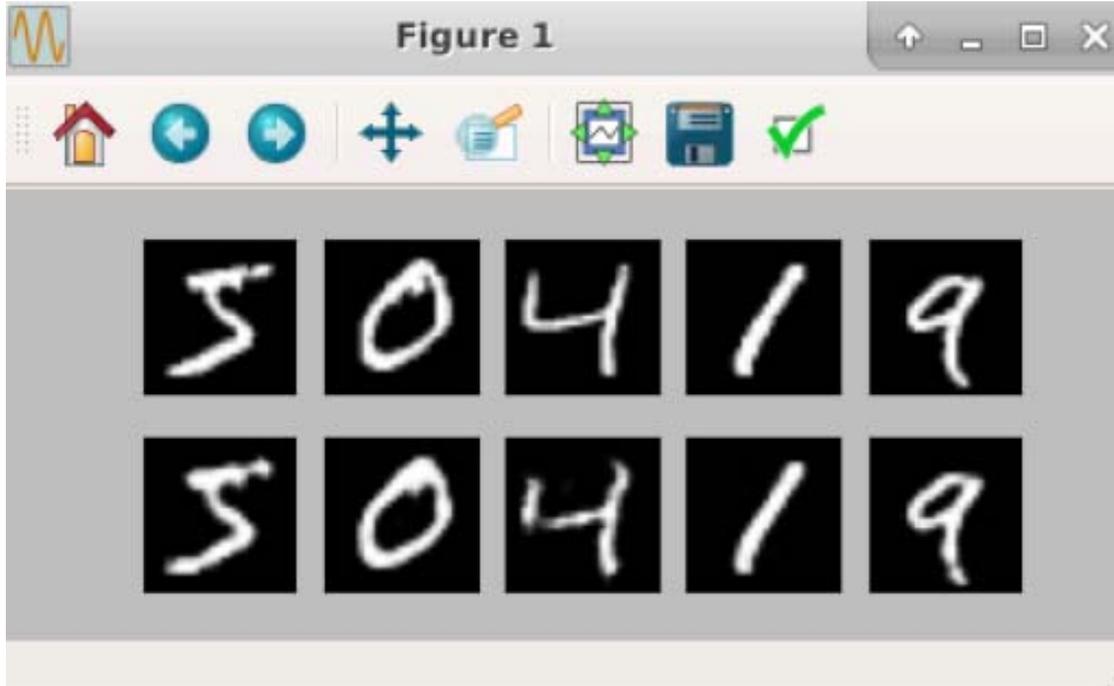
$$\theta^*, \theta'^* = \operatorname{argmin}_{\theta, \theta'} \frac{1}{n} \sum_{i=1}^n \{L(a^{(i)}, c^{(i)}) + \lambda \|J_{f_\theta}(\tilde{a}^{(i)})\|_F^2\}$$

results1

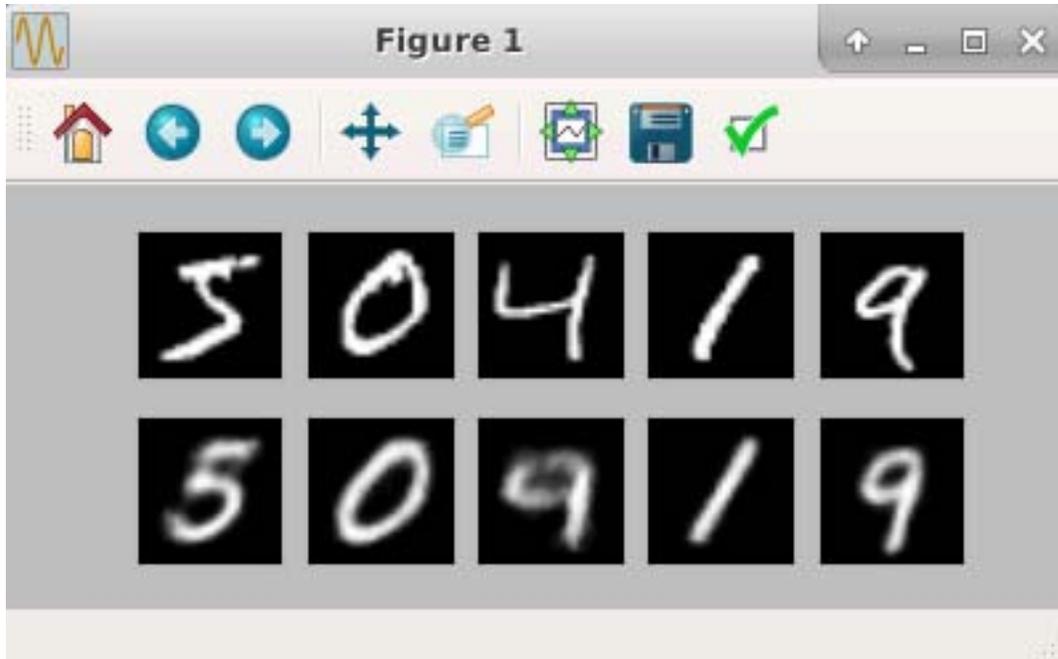
三、AE 与 DAE 代码实验

- 以手写数字体 MNIST 作为实验数据，参考其他代码，完成了代码主要框架，实验结果如下：

1.AutoEncoder (Pytorch), 网络结构 784-128-64-128-784 和 784-128-64-12-3-12-64-128-784



results1



results1

**2.Denoising AutoEncoder, 网络结构 784-256-784, 可自定义
添加噪声和网络层数**

