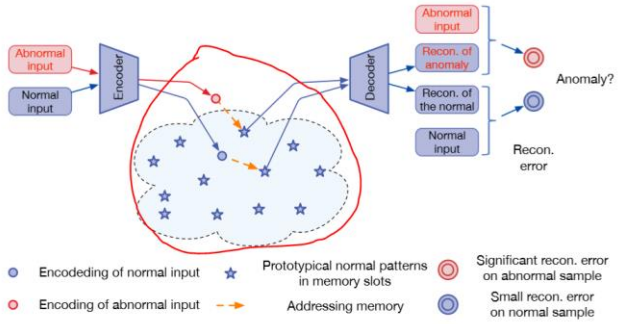
论文小结 MemAE 异常检测

《Memorizing Normality to Detect Anomaly: Memory-augmented Deep Auto encoder for Unsupervised Anomaly Detection》

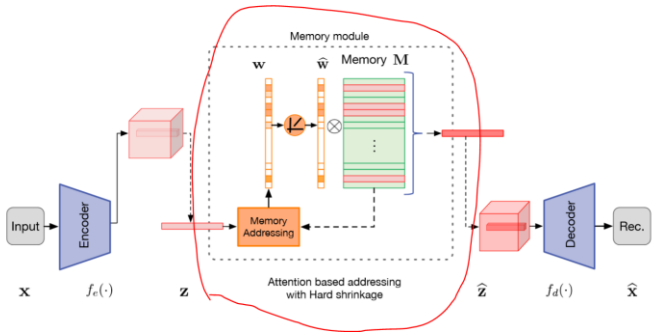
这篇文章解决的主要是异常检测半监督自编码器(unsupervised AE) 方法中的一个问题，即由于训练过程中没有异常样本，从而可能会出现编码器捕捉到的其实是异常和正常数据的共有模式，或是解码器过于强大，从而导致了异常的数据也能够很好地解码，无法从解码错误率中识别出异常的情况。

MemAE 提出的方法本质在于让解码的结果尽可能地接近正常的数据。
从而使得正常数据解码出来重构错误率低，而异常数据重构错误率高。

方法:



在编码后的序列并不直接放入解码器中，而是增加了一个索引的步骤(利用了 Memory)，将其和训练得到的正常数据编码的结果进行匹配。再将匹配的结果放入解码器中，因为匹配的结果是从正常数据中提取的，所以解码后的数据就自然而然地向正常数据中靠近了，对于异常数据而言便是较大地重构误码率。



Memory 是有限的，因此为了模型会有更好地效果，训练过程中这些放入 memory 的样例自然就是能够表现正常数据中典型特征的那一部分。

如何检索匹配?

$$\hat{\mathbf{z}} = \mathbf{w}\mathbf{M} = \sum_{i=1}^{N} w_i \mathbf{m}_i,$$

说匹配其实不太得当，它的本质是综合这些典型的正常数据编码。

w 是权值矩阵。

$$w_i = \frac{\exp(d(\mathbf{z}, \mathbf{m}_i))}{\sum_{j=1}^{N} \exp(d(\mathbf{z}, \mathbf{m}_j))}, \qquad d(\mathbf{z}, \mathbf{m}_i) = \frac{\mathbf{z}\mathbf{m}_i^{\mathsf{T}}}{\|\mathbf{z}\|\|\mathbf{m}_i\|}.$$

所以$\hat{z}$其实就是 z 和 m_i 的越接近，那么 m_i 对最后$\hat{z}$贡献度也就越大。

多个 m_i 的综合可能会得到异于正常数据的模式，为了避免这种情况，要让 w 尽量稀疏，即由少量的比较相似的 m_i 综合即可。
设置一个阈值

$$\widehat{w}_i = h(w_i; \lambda) = \begin{cases} w_i, & \text{if } w_i > \lambda, \\ 0, & \text{otherwise}, \end{cases}$$

该值不可导，增添一个 ReLu 激活

$$\widehat{w}_i = \frac{\max(w_i - \lambda, 0) \cdot w_i}{|w_i - \lambda| + \epsilon},$$

进一步增加稀疏性，要求最小化 wi 的熵

$$E(\widehat{\mathbf{w}}^t) = \sum_{i=1}^{T} -\widehat{w}_i \cdot \log(\widehat{w}_i).$$

最终的 loss function 就是

$$L(\theta_e, \theta_d, \mathbf{M}) = \frac{1}{T} \sum_{t=1}^{T} \left(R(\mathbf{x}^t, \widehat{\mathbf{x}}^t) + \alpha E(\widehat{\mathbf{w}}^t) \right)$$

其中 $R(\mathbf{x}^t, \widehat{\mathbf{x}}^t) = \|\mathbf{x}^t - \widehat{\mathbf{x}}^t\|_2^2,$