### Thinking1 什么是AutoEncoder？能简要说明AutoEncoder的原理（10points）？

AutoEncoder是无监督学习，即学习过程不需要使用样本label，本质上是将样本的输入同时作为神经网络的输入和输出，通过最小化重构误差希望学习到样本的抽象特征表示。搭建一个AutoEncoder需要完成三个工作：搭建编码器，搭建解码器，设定一个损失函数，用以衡量由于压缩而损失掉的信息。通过AutoEncoder可以对高维数据进行高效的特征提取和表示。为什么要做自编码转换呢？转换后会具有一些优良特性，比如节省空间，方便相似度计算等；

### **Thinking2 Greedy Search与Beam Search的区别是什么？能简要说明Greedy Search和Beam Search的区别（10points）？**

Greedy Search只保留一个最优结果，Beam Search中的B代表束宽，集束搜索一次考虑B个可能结果，即Greedy Search是B=1的Beam Search。

### **Thinking3我们为什么要用Attention机制，目的是什么？能简要说明Attention机制（10points）**

为什么：Encoder和Decoder之间唯一的联系就是一个固定长度的语义向量C,即Encoder要将整个序列的信息压缩到一个固定长度的向量中，在语句较长的情况下，语义向量无法完整的表达整个序列，而且之前输入的内容信息会被后输入的信息给稀释掉，序列越长这个问题越严重，所以引入Attention机制，Attention机制跟人类翻译文章时候的思路有些类似，即将注意力关注于我们翻译部分对应的上下文。在Decoder中，使用attention模型中Encoder将所有输入信息编码进多个向量中。

机制：

利用RNN结构得到Encoder中的hidden state，

假设当前Decoder的hidden state 是 ，计算每一个输入位置j与当前输出位置的关联性，这里  表示相关性计算，比如采用点积或是余弦，写成向量形式得到，对于  进行softmax，得到概率分布，利用  进行加权得到context vector，所以下一时刻的隐藏层状态。

**Thinking4 什么是Self-Attention和 Multi-head Attention？能简要说明两者之间的区别（10points）？**

Self-Attention自注意力，用来捕获时序类的特征，传统的Attention是基于source端和target端的隐变量（hidden state）计算Attention的，得到的结果是源端的每个词与目标端每个词之间的依赖关系。但Self Attention不同，它分别在source端和target端进行，仅与source input或者target input自身相关的Self Attention，捕捉source端或target端自身的词与词之间的依赖关系；然后再把source端的得到的self Attention加入到target端得到的Attention中，捕捉source端和target端词与词之间的依赖关系。因此，self Attention比传统的Attention mechanism效果要好，主要原因之一是，传统的Attention机制忽略了源端或目标端句子中词与词之间的依赖关系，相对比，self Attention可以不仅可以得到源端与目标端词与词之间的依赖关系，同时还可以有效获取源端或目标端自身词与词之间的依赖关系。

Multi-head Attention多头注意力机制方法，Query，Key，Value首先进过一个线性变换，然后输入到放缩点积attention，这里要做h次，其实也就是所谓的多头，每一次算一个头。而且每次Q，K，V进行线性变换的参数W是不一样的。然后将h次的放缩点积attention结果进行拼接，再进行一次线性变换得到的值作为多头attention的结果。多头attention的不同之处在于进行了h次计算而不仅仅算一次，这样的好处是可以允许模型在不同的表示子空间里学习到相关的信息，从而使训练速度快（高效并行化），质量高（刷新了各大翻译任务，创造了多项新记录）。