

1、以序列到序列模型（seq2seq）为框架的神经机器翻译的原理。

seq2seq 是一个 Encoder-Decoder 结构的网络，它的输入是一个序列，输出也是一个序列。编码器处理输入序列中的每个元素(在这里可能是1个词)，将捕获的信息编译成向量（称为上下文内容向量）。在处理整个输入序列之后，编码器将上下文发送到解码器，解码器逐项开始产生输出序列。

2、解释编码器和解码器各自的作用。

编码器接收source sentence，然后将其转换为包含具体语义信息的中间向量。

解码器接收编码器传递过来的中间向量，将其输出为最终的目标语句（翻译结果）。

3、如何训练神经机器翻译。

1. 首先要得到一个平行语料库，平行语料库中有需要翻译的句子和对应的翻译后的目标语言句子。
2. 将输入序列传送至编码器，编码器返回编码器输出和编码器隐藏层状态。
3. 将编码器输出、编码器隐藏层状态和解码器输入（即 开始标记）传送至解码器，解码器进行解码，当模型预测结束标记时停止预测。
4. 解码器返回预测和解码器隐藏层状态。
5. 解码器隐藏层状态被传送回模型，预测被用于计算损失。
6. 最后是计算梯度，并将其应用于优化器和反向传播，更新模型参数。

4、怎样在训练阶段计算模型的损失（误差）？

在解码阶段，计算每个输出词语与平行语料中的目标语言句子对应位置的词语的交叉熵，最后将所有词语的交叉熵求平均作为模型的损失函数。

5、什么是 free-running 模式和 teacher-forcing 模式？

模型	Training	Testing	优点	缺点
Free-running	将预测的结果传给下一阶段	将预测的结果传给下一阶段	收敛慢，不稳定	有识别错误的能力
teacher-forcing	将真实结果传给下一阶段	将预测结果传给下一阶段	收敛快，稳定	没有识别错误的能力

6、介绍三种不同的神经机器翻译解码策略（decoding strategies）。

策略	思想	优点	缺点
Greedy decoding	每个时间步都选取概率最大的词语进行输出	实现简单，快	无法保证解码出来的序列全局最佳
穷举法	穷举所有的解码情况，找到最优的序列进行输出	结果全局最优	时间成本高，计算成本高，效率低下
Beam search	介于贪婪和穷举法之间的方法，每个时间步保留 beam size 个概率最大的序列，在最后的 beam size 序列中选择概率最大的序列进行输出	一定程度上增加生成的多样性。相比贪婪采样，也可以防止在某一步出错后，后面的全部出错	随着beam_size的增大，对内存的占用率也会增大，生成的速度也会慢下来。

7、为什么要在神经机器翻译中引入注意力机制（attention）？

在引入注意力转移机制前，如果需要翻译的句子很长，编码器无法记住完整的句子信息，编码器最后输出的状态可能会漏掉一些信息。假如输入的句子有一些信息被忘记，那么解码器将无从得知完整的句子，也不可能产生正确的翻译。

引入注意力机制，解码器每次更新状态的时候，都会再看一遍编码器每个时刻的状态信息，这样解码器就可以知道句子的完整信息，解决了遗忘问题。另外，注意力机制还能告诉解码器当前时刻应该关注哪个状态的信息。

8、神经机器翻译（NMT）有什么优缺点？

优点：

1. 表现好。翻译出来的句子更通顺。由于使用了深度学习的方法，可以引入语义特征，能够很好地利用文本相似度、文本内隐含的多层次特征等统计学习方法捕捉不到的信息。
2. 可以把整个模型看作是一个大的神经网络，只需要端到端地优化神经网络，没有要单独优化的组件。
3. 训练过程不需要人工的特征工程，同一模型对所有语言都适用。

缺点：

1. 解释性差，很难调试。
2. 难以控制，给定一个文本，不能保证翻译的结果。在重要场合使用存在风险。

9、解释机器翻译的评价指标BLEU。

BLEU 的主要思想是基于N-gram等特征来比较译文和参考译文之间n组词的相似的一个占比。BLEU 取值范围在0-1, 取值越靠近1表示机器翻译结果越好。

N-gram 的匹配度可能会随着句子长度的变短而变好，因此会存在这样一个问题：一个翻译引擎只翻译出了句子中部分句子且翻译的比较准确，那么它的匹配度依然会很高。为了避免这种评分的偏向性，BLEU在最后的评分结果中引入了**长度惩罚因子**。当翻译译文长度大于参考译文的长度时，惩罚系数为1，不惩罚，只有机器翻译译文长度小于参考答案才会计算惩罚因子。

由于各N-gram统计量的精度随着阶数的升高而呈指数形式递减，所以为了平衡各阶统计量的作用，对其采用几何平均形式求平均值然后加权，再乘以长度惩罚因子。最后的评价公式为：

$$BLEU = BP \times \exp\left(\sum_{n=1}^N W_n \log P_n\right)$$

其中BP是惩罚因子：

$$BP = \begin{cases} 1 & \text{if } l_c > l_s \\ e^{1 - \frac{l_s}{l_c}} & \text{if } l_c \leq l_s \end{cases}$$