Bert

知道了 transformer 之后再理解 bert 就简单多了。

bert 思想：
1. 将下游的完形填空和下一个句子预测放在了训练词向量的任务中
bert 损失函数：
预测 mask 值的损失函数和预测下一个句子的二分类的损失函数之和

> BERT的损失函数由两部分组成，第一部分是来自 Mask-LM 的单词级别分类任务，另一部分是句子级别的分类任务。通过这两个任务的联合学习，可以使得 BERT 学习到的表征既有 token 级别信息，同时也包含了句子级别的语义信息。具体损失函数如下：

$$L(\theta, \theta_1, \theta_2) = L_1(\theta, \theta_1) + L_2(\theta, \theta_2)$$

> 其中 θ 是 BERT 中 Encoder 部分的参数，θ1 是 Mask-LM 任务中在 Encoder 上所接的输出层中的参数，θ2 则是句子预测任务中在 Encoder 接上的分类器参数。因此，在第一部分的损失函数中，如果被 mask 的词集合为 M，因为它是一个词典大小 |V| 上的多分类问题，那么具体说来有：

$$L_1(\theta, \theta_1) = -\sum_{i=1}^{M} \log p(m = m_i|\theta, \theta_1), m_i \in [1, 2, \ldots, |V|]$$

> 在句子预测任务中，也是一个分类问题的损失函数：

$$L_2(\theta, \theta_2) = -\sum_{j=1}^{N} \log p(n = n_i|\theta, \theta_2), n_i \in [IsNext, NotNext]$$

bert 怎样做预训练
这个博客上内容非常多了。不写了。

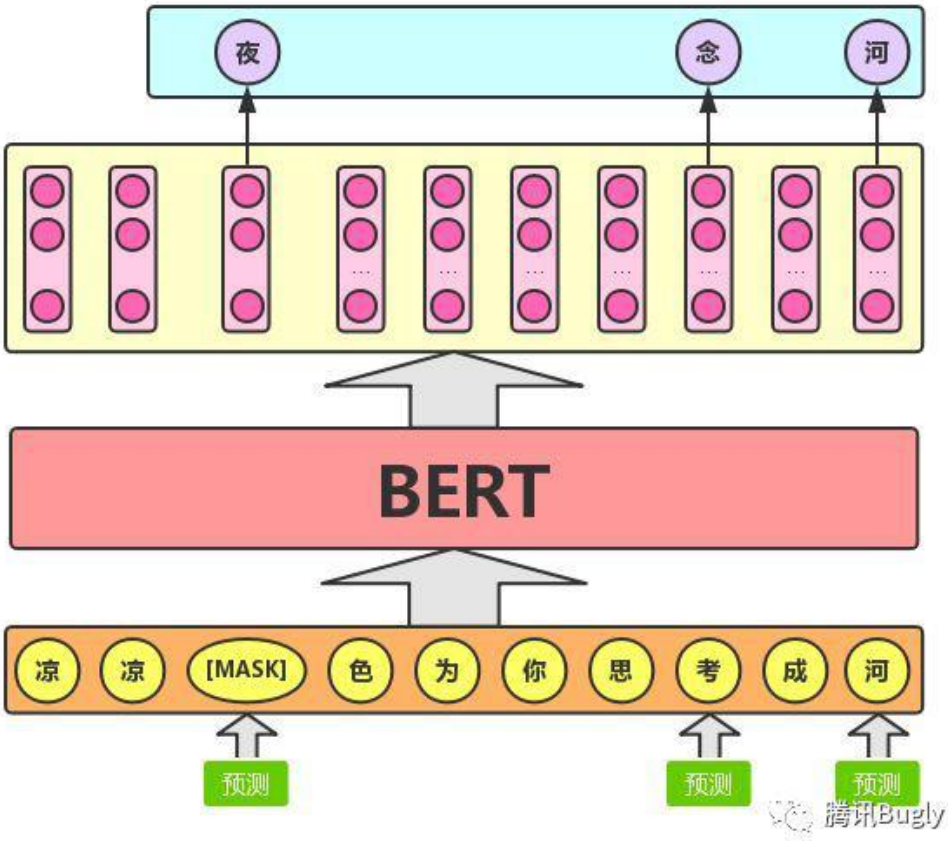
bert 怎样做微调
这个问题比较关键。
首先，看你的任务是什么？
1）文本分类
　　文本分类任务我们需要的是文本的向量表示，这个时候并没有下一个句子预测的思想，因此，在微调的时候必须 mask 词，当成训练一样去调整词向量和文档向量
2）标注
　　譬如分词等任务。这个时候语料往往是多篇文档之类的东西。这个时候便有了当前这句话和下一句话的概念。考虑损失函数，因此可以不 mask，直接得到词向量。

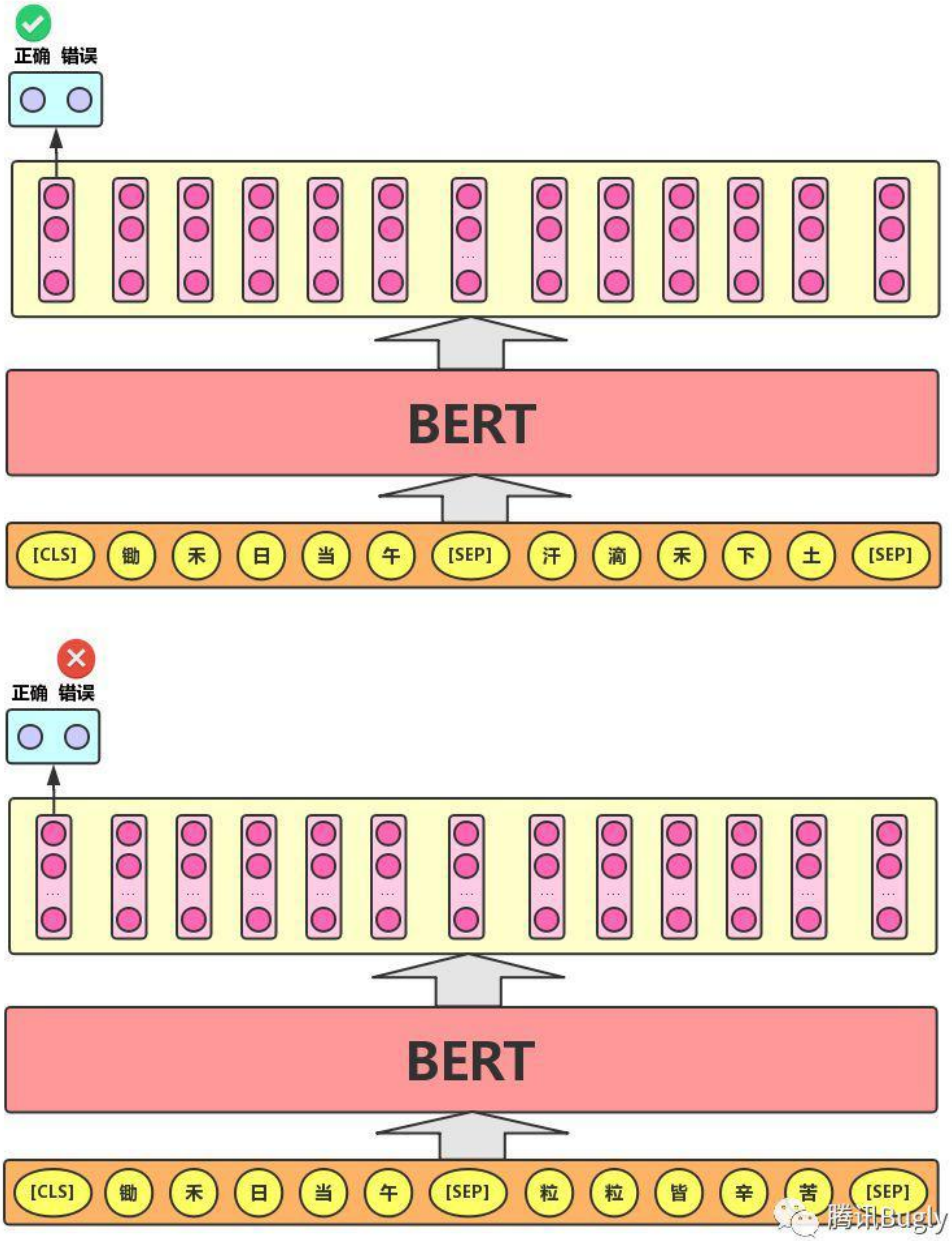
3）问答匹配

问答匹配，是有下一个句子的概念的。同上。

bert 怎样做实时预测

字作为输入，不需要 mask，代入模型，模型的输出就是词的向量表示和句子的向量表示，不需要更新模型。




正确 错误
BERT
[CLS] 锄 禾 日 当 午 [SEP] 汗 滴 禾 下 土 [SEP]
正确 错误
BERT
[CLS] 锄 禾 日 当 午 [SEP] 粒 粒 皆 辛 苦 [SEP]