常见面试问题

- 为什么BERT在第一句前会加一个[CLS]标志?

论文中说：

> The first token of every sequence is always a special classification token [CLS]. The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks.

因为这个无明显语义信息的符号会更“**公平**”地融合文本中各个词的语义信息，从而更好的表示整句话的语义。

- BERT的三个Embedding直接相加会对语义有影响吗?

> Embedding的数学本质，就是以one-hot为输入的单层全连接。也就是说，世界上本没什么Embedding，有的只是one-hot。（苏剑林）

在这里用一个例子再解释一下：

假设 token Embedding 矩阵维度是 [200000,768]；position Embedding 矩阵维度是 [512,768]；segment Embedding 矩阵维度是 [2,768]。

对于一个token，假设它的 token one-hot 是[1,0,0,0,...0]；它的 position one-hot 是[1,0,0,...0]；它的 segment one-hot 是[1,0]。那这个token最后的 word Embedding，就是上面三种 Embedding 的加和。

如此得到的 word Embedding，和concat后的特征：[1,0,0,0...0,1,0,0...0,1,0]，再过维度为 [200000+512+2,768] 的全连接层，得到的向量其实就是一样的。

- 在BERT中，token分3种情况做mask，分别的作用是什么?

15%token做mask；其中80%用[MASK]替换，10%用random token替换，10%不变。其实这个就是典型的 Denosing Autoencoder的思路，那些被Mask掉的单词就是在输入侧加入的所谓噪音

这么做的主要原因是：① 在后续finetune任务中语句中并不会出现 [MASK] 标记；②预测一个词汇时，模型并不知道输入对应位置的词汇是否为正确的词汇（ 10% 概率），这就迫使模型更多地依赖于上下文信息去预测词汇，并且赋予了模型一定的*纠错* 能力。

原文：

> Although this (15% mask) allows us to obtain a bidirectional pre-trained model, a downside is that we are creating a mismatch between pre-training and fine-tuning, since the [MASK] token does not appear during fine-tuning. To mitigate this, we do not always replace "masked" word with the actual [MASK] token.

- BERT的输入是什么，哪些是必须的，为什么position id不用给，type_id 和 attention_mask没有给定的时候，默认会是什么

```
class BertEmbeddings(nn.Module):
    """Construct the embeddings from word, position and token_type embeddings."""

    def __init__(self, config):
        super().__init__()
```

```
        self.word_embeddings = nn.Embedding(config.vocab_size, config.hidden_size,
padding_idx=config.pad_token_id)
        self.position_embeddings = nn.Embedding(config.max_position_embeddings,
config.hidden_size)
        self.token_type_embeddings = nn.Embedding(config.type_vocab_size,
config.hidden_size)

        # self.LayerNorm is not snake-cased to stick with TensorFlow model variable
name and be able to load
        # any TensorFlow checkpoint file
        self.LayerNorm = nn.LayerNorm(config.hidden_size, eps=config.layer_norm_eps)
        self.dropout = nn.Dropout(config.hidden_dropout_prob)
        # position_ids (1, len position emb) is contiguous in memory and exported when
serialized
        self.position_embedding_type = getattr(config, "position_embedding_type",
"absolute")
        self.register_buffer("position_ids",
torch.arange(config.max_position_embeddings).expand((1, -1)))
        if version.parse(torch.__version__) > version.parse("1.6.0"):
            self.register_buffer(
                "token_type_ids",
                torch.zeros(self.position_ids.size(), dtype=torch.long),
                persistent=False,
            )

    def forward(
        self, input_ids=None, token_type_ids=None, position_ids=None,
inputs_embeds=None, past_key_values_length=0
    ):
        if input_ids is not None:
            input_shape = input_ids.size()
        else:
            input_shape = inputs_embeds.size()[:-1]

        seq_length = input_shape[1]

        if position_ids is None:
            position_ids = self.position_ids[:, past_key_values_length : seq_length +
past_key_values_length]

        if token_type_ids is None:
            if hasattr(self, "token_type_ids"):
                buffered_token_type_ids = self.token_type_ids[:, :seq_length]
                buffered_token_type_ids_expanded =
buffered_token_type_ids.expand(input_shape[0], seq_length)
                token_type_ids = buffered_token_type_ids_expanded
            else:
                token_type_ids = torch.zeros(input_shape, dtype=torch.long,
device=self.position_ids.device)

        if inputs_embeds is None:
            inputs_embeds = self.word_embeddings(input_ids)
        token_type_embeddings = self.token_type_embeddings(token_type_ids)
```

```
        embeddings = inputs_embeds + token_type_embeddings
        if self.position_embedding_type == "absolute":
            position_embeddings = self.position_embeddings(position_ids)
            embeddings += position_embeddings
        embeddings = self.LayerNorm(embeddings)
        embeddings = self.dropout(embeddings)
        return embeddings
```

BERT和transformer encoder的区别是，在输入端增加segment embedding；position embedding不再使用三角函数的方法生成，而是采用可学习的embedding table。

- BERT训练时使用的学习率 warm-up 策略是怎样的？为什么要这么做？

warmup 需要在训练最初使用较小的学习率来启动，并很快切换到大学习率而后进行常见的 decay。

这是因为，刚开始模型对数据的“分布”理解为零，或者是说“均匀分布”（当然这取决于你的初始化）；在第一轮训练的时候，每个数据点对模型来说都是新的，模型会很快地进行数据分布修正，如果这时候学习率就很大，极有可能导致开始的时候就对该数据“过拟合”，后面要通过多轮训练才能拉回来，浪费时间。当训练了一段时间（比如两轮、三轮）后，模型已经对每个数据点看过几遍了，或者说对当前的batch而言有了一些**正确的先验**，较大的学习率就不那么容易会使模型学偏，所以可以适当调大学习率。这个过程就可以看做是warmup。那么为什么之后还要decay呢？当模型训到一定阶段后（比如十个epoch），模型的分布就已经比较固定了，或者说能学到的新东西就比较少了。如果还沿用较大的学习率，就会破坏这种稳定性，用我们通常的话说，就是已经接近loss的local optimal了，为了靠近这个point，我们就要慢慢来。

https://www.zhihu.com/question/338066667

- BERT训练过程中的损失函数是什么？

BERT的损失函数由两部分组成，第一部分是来自 MLM 的单词级别分类任务，另一部分是来自NSP的句子级别的分类任务。通过这两个任务的联合学习（是同时训练，而不是前后训练），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：

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

其中 θ是 BERT 中 Encoder 部分的参数，θ_1是 MLM 任务中在 Encoder 上所接的输出层中的参数，θ_2 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：

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

在句子预测任务中，也是一个分类问题的损失函数：

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

因此，两个任务联合学习的损失函数是：

$$L\left(\theta, \theta_1, \theta_2\right)=-\sum_{i=1}^{M} \log p\left(m=m_i|\theta, \theta_1\right)-\sum_{j=1}^{N} \log p\left(n=n_i|\theta, \theta_2\right)$$

- 为什么BERT比ELMo效果好？ELMo和BERT的区别是什么？

因为LSTM抽取特征的能力远弱于Transformer，即使是拼接双向特征，其融合能力也偏弱；BERT的训练数据以及模型参数均多于ELMo。

ELMO给下游提供的是每个单词的embedding，所以这一类预训练的方法被称为**“Feature-based Pre-Training”**。而BERT模型是“**基于Fine-tuning的模式**”，这种做法的下游任务需要将模型改造成BERT模型，才可利用BERT模型预训练好的参数。

- bert的mask为何不学习transformer在attention处进行屏蔽score的技巧？

https://www.zhihu.com/question/318355038
