

深度学习——NLP 已购

来自【机器学习面试题汇总与解析（蒋豆芽面试题总结）】 | 78 浏览 | 0 回复 | 2021-06-04



蒋豆芽

+关注

机器学习面试题汇总与解析——NLP

1. LSTM与transform的区别 ☆☆☆☆☆
2. 讲一下Bert原理，Bert好在哪里？ ☆☆☆☆☆
3. cbow 与 skip-gram 的区别和优缺点 ☆☆☆☆☆
4. Bert的MLM预训练任务mask的目的是什么 ☆☆☆☆☆
5. CRF原理 ☆☆☆☆☆
6. Bert 采用哪种Normalization结构，LayerNorm和BatchNorm区别，LayerNorm结构有参数吗，参数的作用？ ☆☆☆☆☆
7. 如何优化BERT效果 ☆☆☆☆☆
8. BERT self-attention相比lstm优点是什么？ ☆☆☆☆☆
9. 说说循环神经网络 ☆☆☆☆☆
10. 说说LSTM ☆☆☆☆☆
11. LSTM的结构 ☆☆☆☆☆
12. LSTM的三个门怎么运作的，写一下三个门的公式 ☆☆☆☆☆
13. LSTM为什么可以解决长期依赖，LSTM会梯度消失吗 ☆☆☆☆☆
14. LSTM相较于RNN的优势 ☆☆☆☆☆
15. 讲一下LSTM，LSTM相对于RNN有哪些改进？LSTM为什么可以解决长期问题，相对与RNN改进在哪
16. 讲一下LSTM吧，门都是怎么迭代的 ☆☆☆☆☆
17. RNN为什么难以训练，LSTM又做了什么改进 ☆☆☆☆☆
18. wide & deep 模型 wide 部分和 deep 部分分别侧重学习什么信息 ☆☆☆☆☆
19. deepfm 一定优于 wide & deep 吗 ☆☆☆☆☆
20. Bert的输入是什么？ ☆☆☆☆☆
21. Bert的词向量的embedding怎么训练得到的？ ☆☆☆☆☆

蒋豆芽

24. 翻译中Q\K\V对应的是什麼☆☆☆☆☆

25. attention和self attention的区别☆☆☆☆☆

26. 介绍transformer以及讲优势☆☆☆☆☆

27. Transformer encoder和decoder的介绍☆☆☆☆☆

28. BERT模型怎么做的?大致的网络架构是怎么样的?☆☆☆☆☆

29. transformer的position embedding和BERT的position embedding的区别☆☆☆☆☆

- =====
- 本专栏适合于Python已经入门的学生或人士，有一定的编程基础。
 - 本专栏适合于算法工程师、机器学习、图像处理求职的学生或人士。
 - 本专栏针对面试题答案进行了优化，尽量做到好记、言简意赅。这才是一份面试题总结的正确打开方式。这样才方便背诵
 - 如专栏内容有错漏，欢迎在评论区指出或私聊我更改，一起学习，共同进步。
 - 相信大家都有着高尚的灵魂，请尊重我的知识产权，未经允许严禁各类机构和个人转载、传阅本专栏的内容。
- =====

关于机器学习算法书籍，我强烈推荐一本《百面机器学习算法工程师带你面试》，这个就很类似面经，还有讲解，写得比较好。私聊我进群。

关于深度学习算法书籍，我强烈推荐一本《解析神经网络——深度学习实践手册》，简称CNN book，通俗易懂。私聊我进群。

参考资料

B站机器学习视频: <https://space.bilibili.com/10781175/channel/detail?cid=133301>

LSTM: <https://www.jianshu.com/p/95d5c461924c>

<https://zybuluo.com/hanbingtao/note/581764> (用IE浏览器打开)

transform: <https://blog.csdn.net/u013069552/article/details/108074349>

https://blog.csdn.net/qq_41664845/article/details/84969266

<http://jalamar.github.io/illustrated-transformer/> (英文transfrom详解)

Bert: <https://blog.csdn.net/jiaowoshouzi/article/details/89073944>

CRF: <https://www.zhihu.com/question/35866596/answer/139485548>

<https://zhuanlan.zhihu.com/p/148813079>

1. LSTM与transform的区别 ☆☆☆☆

参考回答

Transformers的关键优点

1. 更容易训练、更高效。Transformer中抛弃了传统的CNN和RNN，整个网络结构完全是由Attention机制组成。
2. 可以有效利用迁移训练
3. 能够应用在无监督的文本任务中

LSTM在何时有效

1. 在很长的序列中，LSTM很有效；而Transformers比较低效，时间复杂度在 N^2
2. 在需要实时的任务中有效

答案解析

无。

类似的问题还有：

2. 讲一下Bert原理，Bert好在哪里？ ☆☆☆☆

参考回答

BERT的全称是Bidirectional Encoder Representation from Transformers，即双向Transformer的Encoder，因为decoder是不能获要预测的信息的。模型的**主要创新点**都在pre-train方法上，BERT的预训练阶段包括两个任务，一个是**Masked Language Model**，还有一个是**Next Sentence Prediction**，两种方法分别捕捉词语和句子级别的特征表示。

答案解析

Masked Language Model

MLM可以理解为完形填空，作者会随机mask每一个句子中15%的词，用其上下文来做预测，例如：my dog is hairy → my dog is [MASK]

此处将hairy进行了mask处理，然后采用非监督学习的方法预测mask位置的词是什么

Next Sentence Prediction

选择一些句子对A与B，其中50%的数据B是A的下一条句子，剩余50%的数据B是语料库中随机选择的，学习其中的相关性，添加这样的预训练的的目的是目前很多NLP的任务比如QA和NLI都需要理解两个句子之间的关系，从而能让预训练的模型更好的适应这样的任务。

蒋豆芽

cbow和skip-gram都是在word2vec中用于将文本进行向量表示的实现方法

1. 在**cbow方法**中，是用周围词预测中心词，从而利用中心词的预测结果情况，使用梯度下降方法，不断的去调整周围词的向量。当训练完成之后，每个词都会作为中心词，把周围词的词向量进行了调整，这样也就获得了整个文本里面所有词的词向量；而**skip-gram**是用中心词来预测周围的词。在skip-gram中，会利用周围的词的预测结果情况，使用梯度下降来不断的调整中心词的词向量，最终所有的文本遍历完毕之后，也就得到了文本所有词的词向量。
2. 可以看到，**cbow**预测行为的次数跟整个文本的词数几乎是相等的（每次预测行为才会进行一次反向传播，而往往这也是最耗时的部分），复杂度大概是 $O(V)$ ；**skip-gram**进行预测的次数是要多于cbow的：因为每个词在作为中心词时，都要使用周围词进行预测一次。这样相当于比cbow的方法多进行了K次（假设K为窗口大小），因此时间的复杂度为 $O(KV)$ ，训练时间要比cbow要长。

优缺点

简单说，skip-gram 出来的准确率比cbow 高，但训练时间要比cbow要长；在计算时，cbow会将context word 加起来，在遇到生僻词是，预测效果将会大大降低。skip-gram则会预测生僻字的使用环境，预测效果更好。

答案解析

无。

4. Bert的MLM预训练任务mask的目的是什么☆☆☆☆☆

参考回答

个人理解是让模型学习一个句子中词与词之间的关系。

答案解析

无。

5. CRF原理☆☆☆☆☆

参考回答

设X与Y是随机变量， $P(Y|X)$ 是给定X的条件下Y的条件概率分布，若随机变量Y构成一个由无向图 $G=(V,E)$ 表示的**马尔科夫随机场**。则称条件概率分布 $P(Y|X)$ 为条件随机场。因为是在X条件下的马尔科夫随机场，所有叫**条件随机场**。

答案解析

6. Bert 采用哪种Normalization结构，LayerNorm和BatchNorm区别，LayerNorm结构有参数吗，参数的作用？☆☆☆☆☆

参考回答

采用LayerNorm结构，和BatchNorm的区别主要是做规范化的维度不同，BatchNorm针对一个batch里面的数据进行规范化，针对单个神经元进行，比如batch里面有64个样本，那么规范化输入的这64个样本各自经过这个神经元后的值（64维），LayerNorm则是针对单个样本，不依赖于其他数据，常被用于小mini-batch场景和RNN，特别是自然语言处理领域，就bert来说就是对每层输出的隐层向量（768维）做规范化，图像领域用BN比较多的原因是因为每一个卷积核的参数在不同位置的神经元当中是共享的，因此也应该被一起规范化。

有参数，引入了b再平移参数和w再放缩参数。目的是为了恢复原始数据分布

答案解析

无。

7. 如何优化BERT效果☆☆☆☆☆

参考回答

1. 感觉最有效的方式还是数据。
2. 把现有的大模型ERNIE_2.0_large, Roberta, roberta_wwm_ext_large、roberta-pair-large等进行ensemble，然后蒸馏原始的bert模型，这是能有效提高的，只是操作代价比较大。
3. BERT上面加一些网络结构，比如attention，rcnn等，个人得到的结果感觉和直接在上面加一层transformer layer的效果差不多，模型更加复杂，效果略好，计算时间略增加。
4. 文本对抗

答案解析

无。

8. BERT self-attention相比Istm优点是什么？☆☆☆☆☆

参考回答

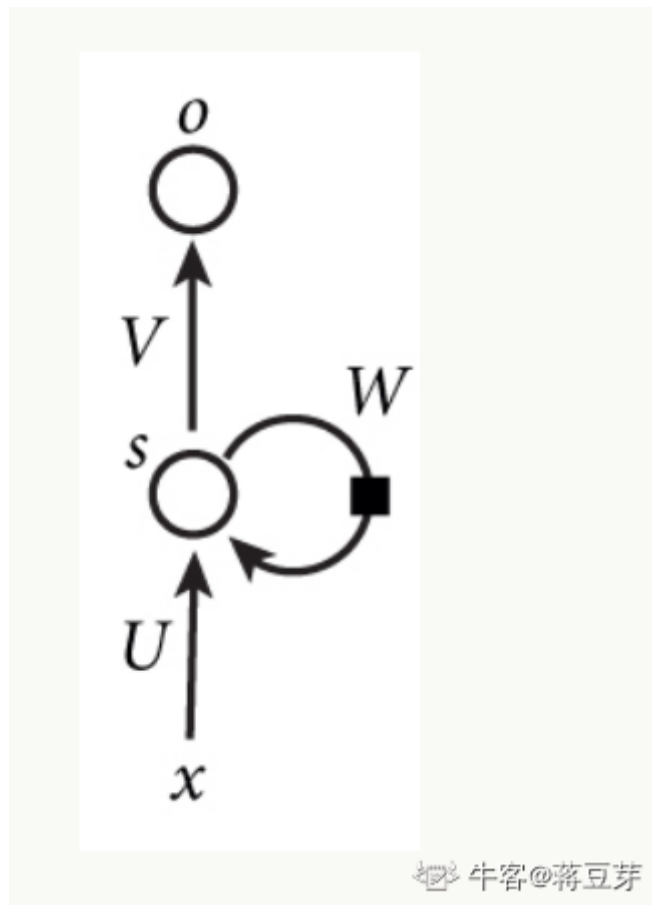
bert通过使用self-attention + position embedding对序列进行编码，Istm的计算过程是从左到右从上到下（如果是多层Istm的话），后一个时间节点的emb需要等前面的算完，而bert这种方式相当于并行计算，虽然模型复杂了很多，速度其实差不多，而且精度更高。

9. 说说循环神经网络☆☆☆☆☆

参考回答

循环神经网络(Recurrent Neural Network)用于处理**序列信息**，即前面的输入和后面的输入是有关系的。比如，当我们在理解一句话意思时，孤立的理解这句话的每个词是不够的，我们需要处理这些词连接起来的整个序列；当我们处理视频的时候，我们也不能只单独的去分析每一帧，而要分析这些帧连接起来的整个序列。

一个简单的循环神经网络如，它由**输入层**、一个**隐藏层**和一个**输出层**组成：



如果把上面有 W 的那个带箭头的圈去掉，它就变成了最普通的全连接神经网络。 x 是一个向量，它表示**输入层**的值（这里面没有画出来表示神经元节点的圆圈）； s 是一个向量，它表示**隐藏层**的值（这里隐藏层面画了一个节点，你也可以想象这一层其实是多个节点，节点数与向量 s 的维度相同）； U 是输入层到隐藏层的**权重矩阵**； o 也是一个向量，它表示**输出层**的值； V 是**隐藏层到输出层**的**权重矩阵**。**循环神经网络**的隐藏层的值 s 不仅仅取决于当前这次的输入 x ，还取决于上一次隐藏层的值 s 。**权重矩阵 W** 就是隐藏层上一次的值作为这一次的输入的权重。

答案解析

参考文章

10. 说说LSTM☆☆☆☆

参考回答

循环神经网络很难训练，导致了它在实际应用中，很难处理**长距离的依赖**。一种改进之后的循环神经网络：长短时记忆网络(Long Short Term Memory Network, **LSTM**)，它成功的解决了原始循环神经网络的缺陷，成为当前最流行的**RNN**，在语音识别、图片描述、自然语言处理等许多领域中成功应用。

与简单**RNN**结构中单一tanh循环体不同的是，LSTM使用三个“门”结构来控制不同时刻的状态和输出。所谓的“门”结构就是使用了sigmoid激活函数的全连接神经网络和一个按位做乘法的操作，sigmoid激活函数会输出一个0~1之间的数值，这个数值描述的是当前有多少信息能通过“门”，0表示任何信息都无法通过，1表示全部信息都可以通过。其中，“**遗忘门**”和“**输入门**”是LSTM单元结构的核心。

LSTM的第一步就是决定细胞状态需要丢弃哪些信息。这部分操作是通过一个称为**遗忘门**的sigmoid单元来处理的。比如“十年前，北京的天空是蓝色的”，但当看到“空气污染开始变得越来越严重”后，RNN应该忘记“北京的天空是蓝色的”这个信息。遗忘门会根据当前时刻节点的输入 x_t 、上一时刻节点的状态 C_{t-1} 和上一时刻节点的输出 h_{t-1} 来决定哪些信息将被遗忘。

下一步是决定给细胞状态添加哪些新的信息。这一步又分为**两个步骤**，**首先**，利用 h_{t-1} 和 x_t 通过一个称为**输入门**的操作来决定更新哪些信息。**然后**利用 h_{t-1} 和 x_t 通过一个tanh层得到新的候选细胞信息 \tilde{C}_t ， \tilde{C}_t 的一部分信息可能会被更新到细胞信息中。比如看到“空气污染开始变得越来越严重”后，模型需要记忆这个最新的信息。

更新完细胞状态后需要根据输入的 h_{t-1} 和 x_t 来判断输出细胞的哪些状态特征，这里需要将输入经过一个称为**输出门**的sigmoid层得到判断条件，然后将细胞状态经过tanh层得到一个-1~1之间值的向量，该向量与输出门得到的判断条件相乘就得到了最终该**RNN**单元的输出。比如当前时刻节点状态为被污染，那么“天空的颜色”后面的单词应该是“灰色”。

答案解析

无。

11. LSTM的结构☆☆☆☆

参考回答

循环神经网络很难训练，导致了它在实际应用中，很难处理**长距离的依赖**。一种改进之后的循环神经网络：长短时记忆网络(Long Short Term Memory Network, **LSTM**)，它成功的解决了原始循环神经网络的缺陷，成为当前最流行的**RNN**，在语音识别、图片描述、自然语言处理等许多领域中成功应用。

与简单**RNN**结构中单一tanh循环体不同的是，LSTM使用三个“门”结构来控制不同时刻的状态和输出。所谓的“门”结构就是使用了sigmoid激活函数的全连接神经网络和一个按位做乘法的操作。

蒋豆芽

门”是LSTM单元结构的核心。

LSTM的第一步就是决定细胞状态需要丢弃哪些信息。这部分操作是通过一个称为**遗忘门**的sigmoid单元来处理的。比如“十年前，北京的天空是蓝色的”，但当看到“空气污染开始变得越来越严重”后，RNN应该忘记“北京的天空是蓝色的”这个信息。遗忘门会根据当前时刻节点的输入 x_t 、上一时刻节点的状态 C_{t-1} 和上一时刻节点的输出 h_{t-1} 来决定哪些信息将被遗忘。

下一步是决定给细胞状态添加哪些新的信息。这一步又分为**两个步骤**，**首先**，利用 h_{t-1} 和 x_t 通过一个称为**输入门**的操作来决定更新哪些信息。**然后**利用 h_{t-1} 和 x_t 通过一个tanh层得到新的候选细胞信息 \tilde{C}_t ， \tilde{C}_t 的一部分信息可能会被更新到细胞信息中。比如看到“空气污染开始变得越来越严重”后，模型需要记忆这个最新的信息。

更新完细胞状态后需要根据输入的 h_{t-1} 和 x_t 来判断输出细胞的哪些状态特征，这里需要将输入经过一个称为**输出门**的sigmoid层得到判断条件，然后将细胞状态经过tanh层得到一个-1~1之间值的向量，该向量与输出门得到的判断条件相乘就得到了最终该**RNN**单元的输出。比如当前时刻节点状态为被污染，那么“天空的颜色”后面的单词应该是“灰色”。

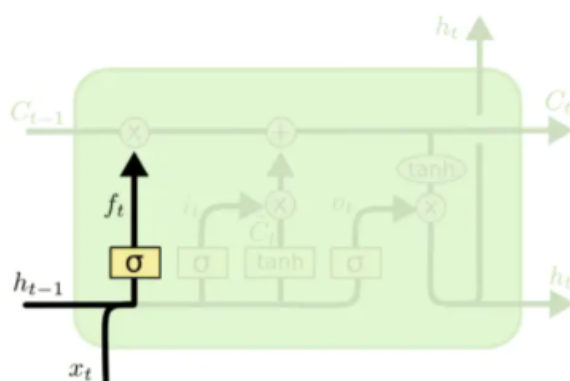
答案解析

无。

12. LSTM的三个门怎么运作的，写一下三个门的公式☆☆☆☆☆

参考回答

LSTM的第一步就是决定细胞状态需要丢弃哪些信息。这部分操作是通过一个称为**遗忘门**的sigmoid单元来处理的。它通过查看 h_{t-1} 和 x_t 信息来输出一个0-1之间的向量，该向量里面的0-1值表示细胞状态 C_{t-1} 中的哪些信息保留或丢弃多少。0表示不保留，1表示都保留。忘记门如下图所示：

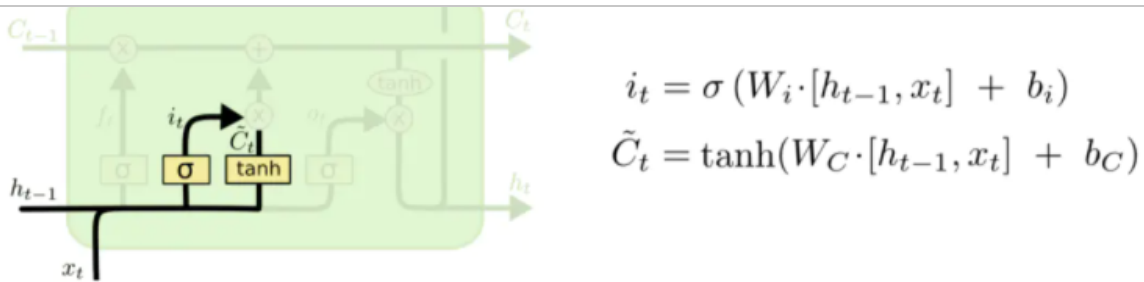


$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

牛客@蒋豆芽

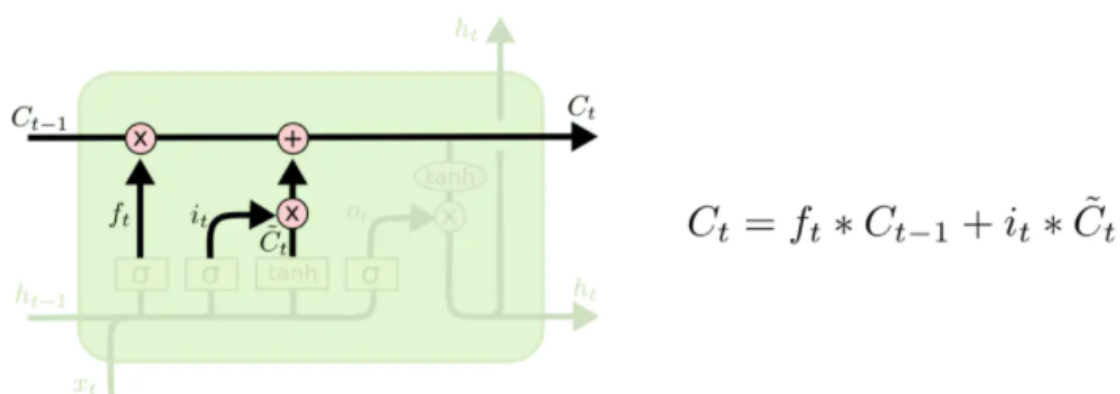
下一步是决定给细胞状态添加哪些新的信息。这一步又分为**两个步骤**，**首先**，利用 h_{t-1} 和 x_t 通过一个称为**输入门**的操作来决定更新哪些信息。**然后**利用 h_{t-1} 和 x_t 通过一个tanh层得到新的候选细胞信息 \tilde{C}_t ，这些信息可能会被更新到细胞信息中。这两步描述如下图所示。

蒋豆芽



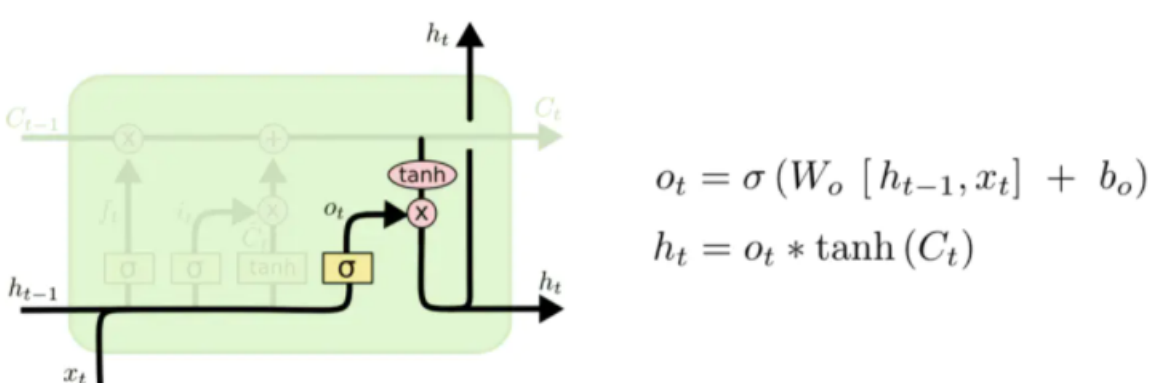
牛客@蒋豆芽

下面将更新旧的细胞信息 C_{t-1} ，变为新的细胞信息 C_t 。更新的规则就是通过忘记门选择忘记旧细胞信息的一部分，通过输入门选择添加候选细胞信息 \tilde{C}_t 的一部分得到新的细胞信息 C_t 。更新操作如下图所示



牛客@蒋豆芽

更新完细胞状态后需要根据输入的 h_{t-1} 和 x_t 来判断输出细胞的哪些状态特征，这里需要将输入经过一个称为**输出门**的sigmoid层得到判断条件，然后将细胞状态经过tanh层得到一个-1~1之间值的向量，该向量与输出门得到的判断条件相乘就得到了最终该RNN单元的输出。该步骤如下图所示



牛客@蒋豆芽

答案解析

无。

蒋豆芽

参考回答

因为LSTM中有两个通道在保持记忆：**短期记忆h**，保持非线性操作；**长期记忆C**，保持线性操作。因为线性操作是比较稳定的，所以C的变化相对稳定，保持了**长期记忆**。而对有用信息的长期记忆是通过训练获得的，也就是说在内部的几个权值矩阵中。

LSTM总可以通过选择合适的参数，在不发生梯度爆炸的情况下，找到合理的梯度方向来更新参数，而且这个方向可以充分地考虑**远距离的隐含层信息**的传播影响。

另外需要强调的是，LSTM除了在结构上**天然地克服了梯度消失的问题**，更重要的是具有更多的参数来控制模型；通过四倍于RNN的参数数量，可以更加精细地预测时间序列变量。

答案解析

无。

14. LSTM相较于RNN的优势☆☆☆☆☆

参考回答

LSTM引入三个控制门，拥有了长期记忆，更好的解决了RNN的梯度消失和梯度爆炸的问题。

答案解析

无。

15. 讲一下LSTM，LSTM相对于RNN有哪些改进？LSTM为什么可以解决长期问题，相对与RNN改进在哪

参考回答

略。

答案解析

无。

16. 讲一下LSTM吧，门都是怎么迭代的☆☆☆☆☆

参考回答

略。

答案解析

无。

蒋豆芽

略。

答案解析

无。

18. wide & deep 模型 wide 部分和 deep 部分分别侧重学习什么信息☆☆☆☆☆

参考回答

在Wide & Deep模型中包括两部分，分别为 **Wide模型**和**Deep模型**。Wide & Deep模型的思想来源是，根据人脑有不断记忆并泛化的过程，这里将 **宽线性模型（Wide Model，用于记忆）** 和 **深度神经网络模型（Deep Model，用于泛化）** 相结合，汲取各自优势形成了Wide & Deep模型，以用于**推荐排序**。

记忆（Memorization） 主要是学习特征的共性/相关性。wide 部分用来**学习记忆**

泛化（Generalization） 可以被理解为相关性的传递（Transitivity），是指算法可以学会特征背后的规律。deep 部分用来**学习泛化**

答案解析

参考文章：<https://www.jianshu.com/p/5b807e6c3801>

19. deepfm 一定优于 wide & deep 吗☆☆☆☆☆

参考回答

Wide&Deep：同时学习低阶和高阶组合特征，它混合了一个线性模型（Wide part）和Deep模型（Deep part）。这两部分模型需要不同的输入，而Wide part部分的输入，**依旧依赖人工特征工程**。

DeepFM：在Wide&Deep的基础上进行改进，不需要预训练FM得到隐向量，**不需要人工特征工程**，能同时学习低阶和高阶的组合特征；FM模块和Deep模块共享Feature Embedding部分，可以更快的训练，以及更精确的训练学习。

DeepFM在Wide&Deep的基础上进行改进，从理论上来说是优于wide & deep的，但不能说一定，还要考虑数据量、过拟合等诸多的问题。

答案解析

参考文章：<https://zhuanlan.zhihu.com/p/137894818>

20. Bert的输入是什么？☆☆☆☆☆

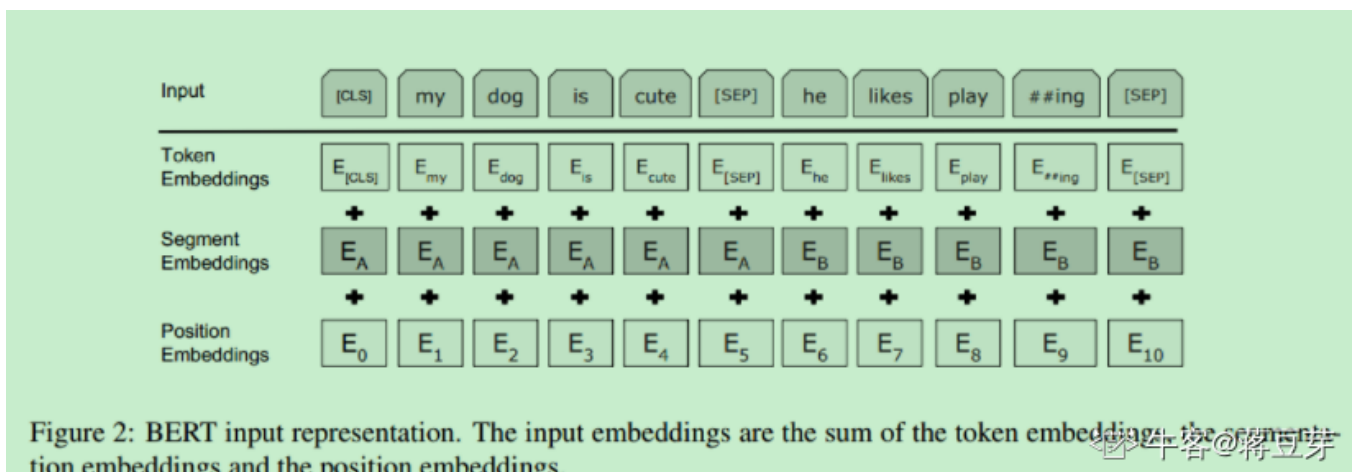
参考回答

蒋豆芽

2. segment embedding 段向量, 其中端对应的就是inputs的一句话, 句子末尾都有加[SEP]结尾符, 两句拼接开头有[CLS]符号。是因为BERT里面的下一句的预测任务, 所以会有两句拼接起来, 上句与下句, 上句有上句段向量, 下句则下有句段向量, 也就是图中A与B。
3. position embedding: 是因为 Transformer 模型不能记住时序, 所以人为加入表示位置的向量

之后这三个向量拼接起来的输入会喂入BERT模型, 输出各个位置的表示向量

答案解析



21. Bert的词向量的embedding怎么训练得到的? ☆☆☆☆☆

参考回答

Masked Language Model

MLM可以理解为完形填空, 作者会随机mask每一个句子中15%的词, 用其上下文来做预测, 例如: my dog is hairy → my dog is [MASK]

此处将hairy进行了mask处理, 然后采用非监督学习的方法预测mask位置的词是什么

答案解析

无。

22. self-attention理解和作用, 为什么要除以根号dk? ☆☆☆☆☆

参考回答

Self-Attention是Transformer最核心的内容, 其核心内容是为输入向量的每个单词学习一个权重

在self-attention中, 每个单词有3个不同的向量, 它们分别是**Query向量 (Q)**, **Key向量 (K)** 和**Value向量 (V)**, 长度均是64。它们是通过3个不同的**权值矩阵**由嵌入向量 X 乘以三个不同的权值矩阵 W_Q , W_K , W_V 得到, 其中三个矩阵的尺寸也是相同的。均是 512×64 ; Q,K,V这

蒋豆芽

作者在论文中的解释是点积后的结果大小是跟维度成正比的，所以经过softmax以后，梯度就会变得很小，除以 dk 后可以让 attention 的权重分布方差为 1，否则会由于某个输入太大的话就会使得权重太接近于 1（softmax 正向接近 1 的部分），梯度很小，造成参数更新困难。

答案解析

无。

23. BERT中并行计算体现在哪儿☆☆☆☆☆

参考回答

不同于 RNN 计算当前词的特征要依赖于前文计算，有时序这个概念，BERT 的 Transformer-encoder 中的 **self-attention** 计算当前词的特征时候，没有时序这个概念，是同时利用上下文信息来计算的，一句话的 **token** 特征是通过矩阵并行运算的，故**并行就体现在self-attention**。

答案解析

无。

24. 翻译中Q\K\V对应的是什麼☆☆☆☆☆

参考回答

分别是**Query向量（Q）**，**Key向量（K）**和**Value向量（V）**

答案解析

无。

25. attention和self attention的区别☆☆☆☆☆

参考回答

attention 和 self-attention 的计算方法是一样的，只不过是它们关注的对象不同而已。

attention主要应用在seq2seq+attention中。以seq2seq框架为例，**输入Source**和**输出Target**内容是不一样的，比如对一件商品的评价和总结来说，Source是一个对一件商品好评或差评的句子，Target是对应的评价的总结，**Attention发生在Target的元素Query和Source中的所有元素之间**。

Self Attention，指的不是Target和Source之间的Attention机制，而是**Source内部元素**之间或者**Target内部元素**之间发生的Attention机制，也可以理解为Target=Source这种特殊情况下的Attention。

答案解析

无。

蒋豆芽

参考回答

Transformer中抛弃了传统的CNN和RNN，整个网络结构完全是由**Attention机制**组成。RNN相关算法只能从左向右依次计算或者从右向左依次计算，这种机制带来了两个问题： 1、时间片 t 的计算依赖 $t-1$ 时刻的计算结果，这样限制了模型的**并行能力**； 2、顺序计算的过程中信息会丢失，尽管LSTM等门机制的结构一定程度上缓解了长期依赖的问题，但是对于特别长期的依赖现象，LSTM依旧无能为力。

Transformer的优势解决了上面两个问题。

transformer模型本质上是一个Encoder-Decoder的结构。输入序列先进行Embedding，经过Encoder之后结合上一次output再输入Decoder，最后用softmax计算序列下一个单词的概率。

transformer的输入是**Word Embedding + Position Embedding**。

答案解析

无。

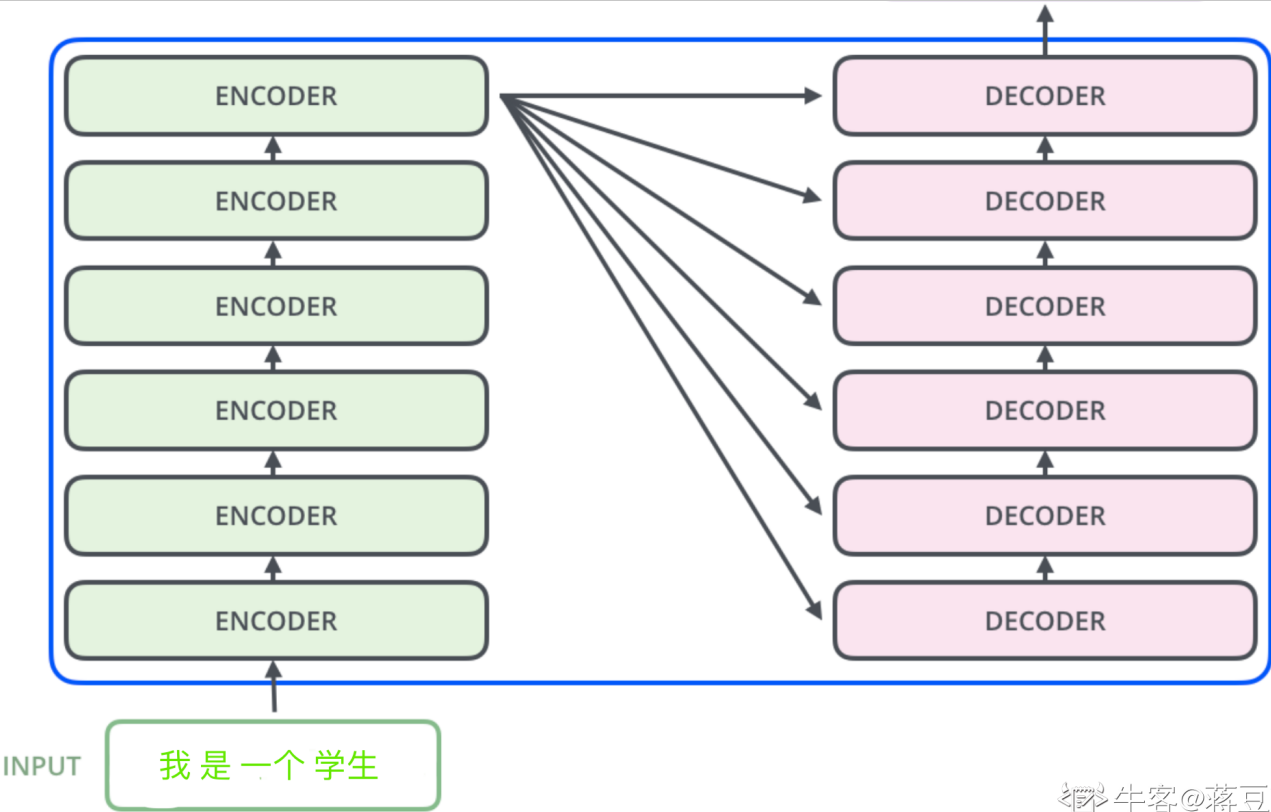
27. Transformer encoder和decoder的介绍☆☆☆☆☆

参考回答

Encoder部分是由个层相同小Encoder Layer串联而成。小Encoder Layer可以简化为两个部分：
(1) Multi-Head Self Attention (2) Feed-Forward network。

Decoder与Encoder有所不同，每一个小Encoder Layer都会输入至所有的小Decoder Layer，如下图所示：

蒋豆芽



答案解析

无。

28. BERT模型怎么做的?大致的网络架构是怎么样的?☆☆☆☆☆

参考回答

略

答案解析

无。


29. transformer的position embedding和BERT的position embedding的区别☆☆☆☆☆

参考回答

略

答案解析

无。

 蒋豆芽

相关专栏





机器学习面试题汇总与解析（蒋豆芽面试题总结）

27篇文章 | 90订阅

[已订阅](#)

0条评论

 默认排序 



没有回复

请留下你的观点吧~

[发布](#)

 **牛客博客，记录你的成长**

[关于博客](#) | [意见反馈](#) | [免责声明](#) | [牛客网首页](#)