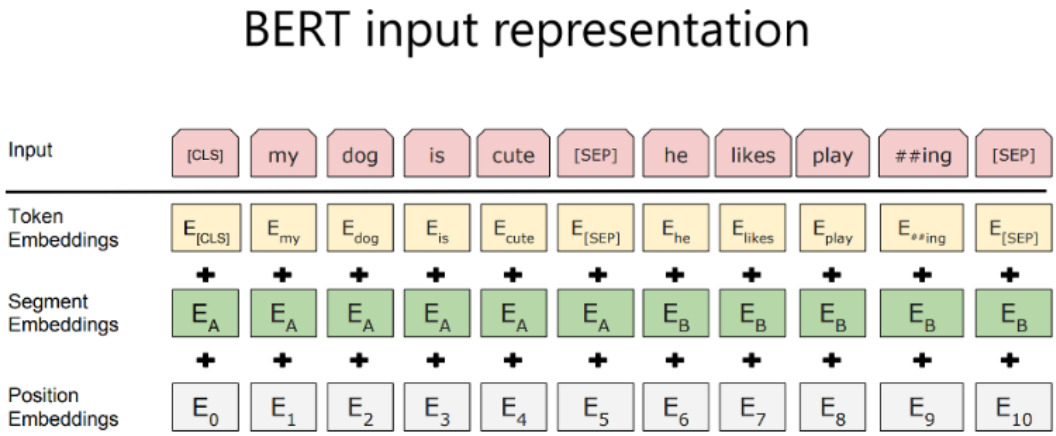
2021年10月32日总结

Deep Learning

1. 预训练模型：
    - 往往用于NLP，几乎在所有的NLP任务中都得到了最好的结果。 预训练模型就意味着把人类的语言知识，先学了一个东西，然后再代入到某个具体任务，就顺手了。 预训练模型是迁移学习的一种应用，其中最关键的三个技术分别是Transformer、自监督学习和微调。
2. 无监督学习：
    - 无监督学习广泛采用的方式是自动编码器，编码器将样本映射到隐层向量，解码器将这个隐层向量映射回样本空间，我们期待网络的输入和输出可以保持一致（理想情况，无损重构），同时隐层向量的维度远远小于输入样本的维度，以此达到了降维的目的，利用学习到的隐层向量（代替原始的输入样本）再进行聚类等任务时将更加的简单高效。
    - 如何学习隐层向量，称之为Representation Learning（表征/表示学习），但是这样简单的编码解码过程不能够学习到更多的语义特征。
3. 自监督学习：
    - 自监督学习主要是利用辅助任务（pretext）从大规模的无监督数据中挖掘自身的监督信息，通过这种构造的监督信息对网络进行训练，从而可以学习到对下游任务有价值的表征。（也就是说自监督学习的监督信息不是人工标注的，而是算法在大规模无监督数据中自动构造监督信息，来进行监督学习或训练）。
    - 自监督学习的主要方法之一就是基于上下文(Context Based)
4. BERT
    - 分为bert_base和bert_large，前者使用了12个encoder（与Transformer模型中的相同），后者使用了24个encoder。
    - 输入部分：bert中的 input = token emb + segment emb + position emb，下图是个例子：



其中的CLS、SEP都是因为NSP任务（处理两个句子的关系，是个二分类任务），SEP用于隔开句子；CLS在预训练之后的输出向量用于做NSP二分类任务，但CLS向量不能代表整个句子的语义信息。

- 如何做预训练：

  MLM-掩码语言模型

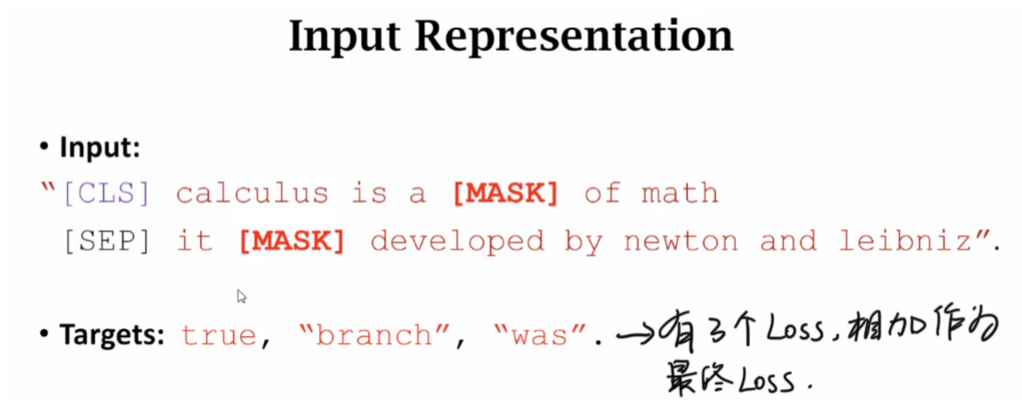
  - 对句子进行一个mask，掩盖一些地方，比如 ”我爱吃饭-> 我爱mask饭“，然后预测mask，这样打破了文本，让模型进行文本重建。
  - mask概率，bert是随机mask15%的单词，然后对mask的词，10%替换成其它，10%原封不动，80%替换成mask。

  NSP：

- 判断sentence A后面是否是sentence B。

- 从训练语料库中取出两个连续的段落作为正样本；从不同的文档中随机创建一个段落作为负样本。

将MLM任务和NSP任务结合，作为BERT的预训练模型，例子如下图：



- 微调（fine-tune）：相当于在原本模型的基础上改变后几层的参数，使得预训练的模型更适合于我们要解决的问题。

Transformer预训练模型

1. 参考链接：https://blog.csdn.net/weixin_42043940/article/details/108015797
2. transformer框架主要有三个类：model类、configuration类、tokenizer类，这三个类都有from_pretrained()和save_pretrained()方法。
    1. Tokenizer：

       负责为模型准本输入。
    2. Model：

       一般都是直接加载预训练模型，传入参数可以是本地目录也可以是模型的名字 。
    3. Configuration：

       预训练模型中的属性和方法。

CodeBERT

1. 运行逻辑：

2. 函数解读：

   run.py中：

   - read_example：加载文件，得到example(idx = idx,source=code,target = nl)，其中，id是编号，source是代码（将word和符号都分开放在一个list中），target是描述，描述代码意义（将描述word分开放list中）。
   - convert_examples_to_features()，对example进行一定的加工，得到用于输入的features。

     数据加载完features的形态：

```
idx: 4
source_tokens: ['<s>', 'public', '_function', '_login', 'Action', '_(', '_Request', '_$', '_request', '_)', '_:', '_Response', '_{', '_$', '_authentication', 'Ut', 'ils', '_=', '_$',
source_ids: 0 15110 5043 27754 36082 36 18593 68 2069 4839 4832 19121 25522 68 24790 41967 5290 5457 68 42 43839 120 36 128 15506 4 40907 14086 1215 49320 108 4839 25606 68 5849 545
source_mask: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
target_tokens: ['<s>', 'Login', '_form', '_action', '_.', '</s>']
target_ids: 0 43890 1026 814 479 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
target_mask: 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

3. bleu是一种评价方式，常用于机器翻译结果，对于模型预测序列中任意的子序列，BLEU考察这个子序列是否出现在标签序列中。

其它

1. python导入自己写的模块出错：在pycharm中，每个project都有一个sources root，这是寻找文件的默认路径，如果调用模块不再sources root 中就会报错，将模块的上层目录设定为sources root就可以了。
2. python的tqdm是一个快速，可扩展的进度条类。
3. 在使用conda创建新的虚拟环境时，要关闭网络代理，否则会报错。
4. tensorflow-gpu相当于是tensorflow的进阶版本，当没有配置好cuda和cudnn，或者没有gpu时，就会像tensorflow一样，只使用CPU运行。

问题

1. 这个是怎么运行的?

   ### Fine-Tune

   We fine-tuned the model on 4*P40 GPUs.

   ```
   cd code2nl

   lang=php #programming language
   lr=5e-5
   batch_size=64
   beam_size=10
   source_length=256
   target_length=128
   data_dir=../data/code2nl/CodeSearchNet
   output_dir=model/$lang
   train_file=$data_dir/$lang/train.jsonl
   dev_file=$data_dir/$lang/valid.jsonl
   eval_steps=1000 #400 for ruby, 600 for javascript, 1000 for others
   train_steps=50000 #20000 for ruby, 30000 for javascript, 50000 for others
   pretrained_model=microsoft/codebert-base #Roberta: roberta-base

   python run.py --do_train --do_eval --model_type roberta --model_name_or_path $pretrained_model --train_filename $train_file --dev_fil
   ```

2.