**使用PyTorch训练多标签文本分类模型**

  好久未更新，近来上海疫情严重，自己工作繁忙，才发现之前能有时间和心情写文章是一件多么奢侈的事情~

  本文将介绍如何使用[PyTorch](https://so.csdn.net/so/search?q=PyTorch&spm=1001.2101.3001.7020)训练多标签文本分类模型。

  所谓多标签[文本分类](https://so.csdn.net/so/search?q=文本分类&spm=1001.2101.3001.7020)，指的是文本可能会属于多个类别，而不是单个类别。与文本多分类的区别在于，文本多分类模型往往有多个类别，但文本至属于其中一个类别；而多标签文本分类也会有多个类别，但文本会属于其中多个类别。

**数据集**

  本文演示的数据集为英语论文数据集，参考网址为：<https://datahack.analyticsvidhya.com/contest/janatahack-independence-day-2020-ml-hackathon>，数据下载需翻墙，读者也可参看后续给出的项目Github。该论文数据集实际上是比赛数据，供选手尝试模型。本文所采用的数据集为英语，至于中文，其原理是一致的，稍微做调整即可。

  该数据集给出论文的标题（TITLE）和摘要（ABSTRACT），来预测论文属于哪个主题。该数据集共有20972个训练样本，有六个主题，分别为：Computer Science, Physics, Mathematics, Statistics, Quantitative Biology, Quantitative Finance。在此给出一个样例数据：

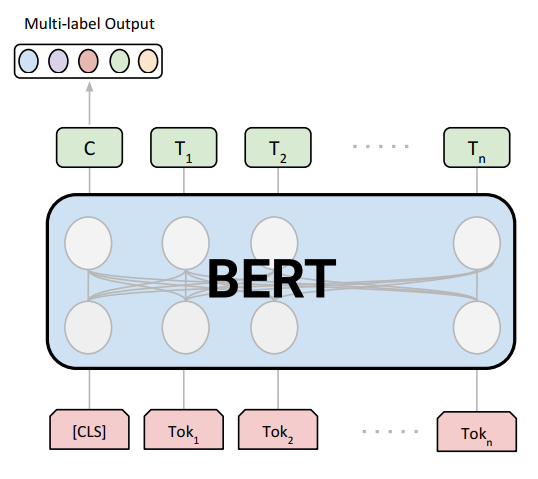
TITLE : Many-Body Localization: Stability and Instability

ABSTRACT: Rare regions with weak disorder (Griffiths regions) have the potential to spoil localization. We describe a non-perturbative construction of local integrals of motion (LIOMs) for a weakly interacting spin chain in one dimension, under a physically reasonable assumption on the statistics of eigenvalues. We discuss ideas about the situation in higher dimensions, where one can no longer ensure that interactions involving the Griffiths regions are much smaller than the typical energy-level spacing for such regions. We argue that ergodicity is restored in dimension d > 1, although equilibration should be extremely slow, similar to the dynamics of glasses.

TOPICS: Physics, Mathematics

**模型结构**

  本文给出的多标签文本分类模型使用预训练模型（BERT），下游网络结构较为简单，算是比较中庸但简单好用的模型方案，模型结构图如下：



该模型使用PyTorch的transformers模块来实现，代码如下:

|  |
| --- |
| Python  # 模型类  class BERTClass(torch.nn.Module):  def \_\_init\_\_(self):  super(BERTClass, self).\_\_init\_\_()  self.l1 = transformers.BertModel.from\_pretrained(MODEL\_NAME\_OR\_PATH)  self.l2 = torch.nn.Dropout(0.2)  self.l3 = torch.nn.Linear(HIDDEN\_LAYER\_SIZE, 6)    def forward(self, ids, mask, token\_type\_ids):  \_, output\_1 = self.l1(ids, attention\_mask=mask, token\_type\_ids=token\_type\_ids)  output\_2 = self.l2(output\_1)  output = self.l3(output\_2)  return output |

使用损失函数为torch.nn.BCEWithLogitsLoss，因而不需要在output层后加上sigmoid激活函数。

  模型训练过程中，将训练数据随机分为训练集和测试集，两部分比例为8:2，同时模型参数设置如下：

|  |
| --- |
| Python  # 模型参数  MAX\_LEN = 128 # 文本最大长度  TRAIN\_BATCH\_SIZE = 32 # 训练批次数量  VALID\_BATCH\_SIZE = 32 # 测试批次数量  EPOCHS = 10 # 训练轮数  LEARNING\_RATE = 1e-05 # 学习率  # 模型  MODEL\_NAME\_OR\_PATH = './bert-base-uncased' # 预训练模型  HIDDEN\_LAYER\_SIZE = 768 # 隐藏层维数 |

**模型效果**

  笔者分别尝试使用bert-base-uncased和bert-large-uncased训练模型，并在测试数据上进行预测，在比赛官网上进行提交，结果如下表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模型** | **max length** | **batch size** | **private score** | **rank** |
| bert-base-uncased | 128 | 32 | 0.8320 | 107 |
| bert-large-uncased | 128 | 16 | 0.8355 | 79 |

  看过一个rank为17的方案，其采用的是多个预训练模型训练后的集成，后接网络与笔者一致。

**总结**

  本项目已经开源，其Github网址为:<https://github.com/percent4/pytorch_english_mltc>。后续将尝试该模型在中文多标签文本分类数据集上的效果，感谢大家阅读~

**参考网址**

1. <https://jovian.ai/kyawkhaung/1-titles-only-for-medium>
2. <https://datahack.analyticsvidhya.com/contest/janatahack-independence-day-2020-ml-hackathon>
3. Fine-tuned BERT Model for Multi-Label Tweets Classification: https://trec.nist.gov/pubs/trec28/papers/DICE\_UPB.IS.pdf