

如何优雅的使用pytorch内置torch.nn.CTCLoss的方法



人工智能...

132 人赞同了该文章

一、开篇简述

CTC 的全称是Connectionist Temporal Classification，中文名称是“连接时序分类”，这个方法主要是解决神经网络label 和output 不对齐的问题（Alignment problem），其优点是不用强制对齐标签且标签可变量，仅需输入序列和监督标签序列即可进行训练，目前，该方法主要应用于场景文本识别（scene text recognition）、语音识别（speech recognition）及手写字识别

（handwriting recognition）等工程场景。以往我们在百度上搜索pytorch + ctc loss得到的结果基本上warp-ctc的使用方法，warp-ctc是百度开源的一个可以应用在CPU和GPU上高效并行的CTC代码库，但是为了在pytorch上使用warp-ctc我们不仅需要编译其源代码还需要进行安装配置，使用起来着实麻烦。而在Pytorch 1.0.x版本内早就有内置ctc loss接口了，我们完全可以直接使用，只是很少有资料介绍如何使用该API。因此，本篇文章结合我个人工程实践中的经验介绍我在pytorch中使用其内置torch.nn.CTCLoss的方法，但不会对ctc loss原理进行展开，期望能给大家在工程实践中使用torch.nn.CTCLoss带来帮助！

二、CTCLoss接口使用说明

第一步，获取CTCLoss()对象

```
ctc_loss = nn.CTCLoss(blank=len(CHARS)-1, reduction='mean')
```

类初始化参数说明：

blank：空白标签所在的label值，默认为0，需要根据实际的标签定义进行设定；

reduction：处理output losses的方式，string类型，可选'none'、'mean'及'sum'，'none'表示对output losses不做任何处理，'mean'则对output losses取平均值处理，'sum'则是对output losses求和处理，默认为'mean'。

第二步，在迭代中调用CTCLoss()对象计算损失值

```
loss = ctc_loss(log_probs, targets, input_lengths, target_lengths)
```

CTCLoss()对象调用形参说明：

log_probs：shape为(T, N, C)的模型输出张量，其中，T表示CTCLoss的输入长度也即输出序列长度，N表示训练的batch size长度，C则表示包含有空白标签的所有要预测的字符集总长度，log_probs一般需要经过torch.nn.functional.log_softmax处理后再送入到CTCLoss中；

targets：shape为(N, S) 或(sum(target_lengths))的张量，其中第一种类型，N表示训练的batch size长度，S则为标签长度，第二种类型，则为所有标签长度之和，但是需要注意的是targets不能包含有空白标签；

input_lengths：shape为(N)的张量或元组，但每一个元素的长度必须等于T即输出序列长度，一般来说模型输出序列固定后则该张量或元组的元素值均相同；

target_lengths：shape为(N)的张量或元组，其每一个元素指示每个训练输入序列的标签长度，但标签长度是可以变化的；

举个具体例子说明如何使用CTCLoss()，如下为CTCLoss在车牌识别里面的应用：

比如我们需要预测的字符集如下，其中'-'表示空白标签；



```
CHARS = ['京', '沪', '津', '渝', '冀', '晋', '蒙', '辽', '吉', '黑',
        '苏', '浙', '皖', '闽', '赣', '鲁', '豫', '鄂', '湘', '粤',
        '桂', '琼', '川', '贵', '云', '藏', '陕', '甘', '青', '宁',
        '新',
        '0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
        'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'J', 'K',
        'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'U', 'V',
        'W', 'X', 'Y', 'Z', 'I', 'O', '-']
```

因为空白标签所在的位置为len(CHARS)-1，而我们需要处理CTCLoss output losses的方式为‘mean’，则需要按照如下方式初始化CTCLoss类：

```
ctc_loss = nn.CTCLoss(blank=len(CHARS)-1, reduction='mean')
```

我们设定输出序列长度T为18，训练批大小N为4且训练数据集仅有4张车牌（为了方便说明）如下，总的字符集长度C如上面CHARS所示为68：



那么我们在训练一次迭代中打印各个输入形参得出如下结果：

1) log_probs由于数值比较多且为神经网络前向输出结果，我们仅打印其shape出来，如下：

```
torch.Size([18, 4, 68])
```

2) 打印targets如下，表示这四张车牌的训练标签，根据target_lengths划分标签后可分别表示这四张车牌：

```
tensor([18, 45, 33, 37, 40, 49, 63, 4, 54, 51, 34, 53, 37, 38, 22, 56, 37, 38, 33, 39, 34, 46, 2, 41, 44,
        37, 39, 35, 33, 40])
```

3) 打印target_lengths如下，每个元素分别指定了按序取targets多少个元素来表示一个车牌即标签：

```
(7, 7, 8, 8)
```

我们划分targets后得到如下标签：

```
18, 45, 33, 37, 40, 49, 63 -->> 车牌 “湘E269JY”
4, 54, 51, 34, 53, 37, 38 -->> 车牌 “冀PL3N67”
22, 56, 37, 38, 33, 39, 34, 46 -->> 车牌 “川R67283F”
2, 41, 44, 37, 39, 35, 33, 40 -->> 车牌 “津AD68429”
```

target_lengths元素数量的不同则表示了标签可变长。

4) 打印input_lengths如下，由于输出序列长度T已经设定为18，因此其元素均是固定相同的：

```
(18, 18, 18, 18)
```

其中，只要模型配置固定了后，log_probs不需要我们组装再传送到CTCLoss，但是其余三个输入形参均需要我们根据实际数据集及C、T、N的情况进行设定！

三、需要注意的地方

3.1 官方所给的例程如下，但在实际应用中需要将log_probs.detach()去掉，否则无法反向传播进行训练；



```
>>> ctc_loss = nn.CTCLoss()
>>> log_probs = torch.randn(50, 16, 20).log_softmax(2).detach().requires_grad_(
>>> targets = torch.randint(1, 20, (16, 30), dtype=torch.long)
>>> input_lengths = torch.full((16,), 50, dtype=torch.long)
>>> target_lengths = torch.randint(10,30,(16,), dtype=torch.long)
>>> loss = ctc_loss(log_probs, targets, input_lengths, target_lengths)
>>> loss.backward()
```

3.2 blank空白标签一定要依据空白符在预测总字符集中的位置来设定，否则就会出错；

3.3 targets建议将其shape设为(sum(target_lengths))，然后再由target_lengths进行输入序列长度指定就好了，这是因为如果设定为(N, S)，则因为S的标签长度如果是可变的，那么我们组装出来的二维张量的第一维度的长度仅为min(S)将损失一部分标签值（多维数组每行的长度必须一致），这就导致模型无法预测较长长度的标签；

3.4 输出序列长度T尽量在模型设计时就要考虑到模型需要预测的最长序列，如需要预测的最长序列其长度为l，则理论上T应大于等于2l+1，这是因为CTCLoss假设在最坏情况下每个真实标签前后都至少有一个空白标签进行隔开以区分重复项；

3.5 输出的log_probs除了进行log_softmax()处理再送入CTCLoss外，还必须要调整其维度顺序，确保其shape为(T, N, C)！

编辑于 2019-06-03 23:12

PyTorch OCR（光学字符识别） 损失函数

推荐阅读

PytorchOCR：一个基于pytorch的OCR算法库

为什么做这个项目从事OCR相关工作近两年，这期间也复现了几篇论文，每次复现一篇论文我都需要重新开个仓库，这让我有了一个想法：不同仓库之间的代码重复度这么高，为什么不把代码都放在……
周军 发表于深度学习在…

【Pytorch】CNN实现手写汉字识别（数据集制作，网络搭建…

之前毕业设计用TensorFlow做了手写汉字识别，使用的中科院的数据集。参考了一篇博客：
<https://zhuanlan.zhihu.com/p/246>
现在用Pytorch复现一下。 下载链接在文末环境：Pytorch：1…
陈小白233



如何用 TensorFlow 实现 OCR

绿萝123 发表于慢慢学Te...

44 条评论

切换为时间排序

写下你的评论...



freeneuro

2020-06-04

请问有具体的演示代码吗？我target.shape为[10]，假如是[1,2,3,4,5,6,7,8,9,0]，target_length.shape为[2, 1],分别是[4, 6]。也就是我第一个样本的target是1234,第二个样本的target是567890。这样的tensor直接输入ctcloss，还是需要将target转换成[[1,2,3,4],[5,6,7,8,9,0]]？

赞



斐波那契洗澡水


2020-04-02


请问多卡情况下要怎么使用呢？我会出现size不匹配的问题

赞同 132 44 条评论 分享 喜欢 收藏 申请转载 ...


 DuckJ 回复 斐波那契洗澡水 2020-06-09


请问你解决了么？是不是将preds_size获取值时候*batchsize/ngpu

 赞


 一名无知的程序员 2020-03-12


感谢您的分享，我想问一下，probs在进入ctc损失前，是不是需要softmax？第二，input_lengths里面的值是不是可以不同？

 赞


 嘟嘟 回复 一名无知的程序员 2020-08-13


是需要的，文中3.11计算log_probs的公式已经.logsoftmax（2）了，就是已经softmax了。另外inpute_length的值，我感觉每个batch应该是一样的，不同batch之间是不一定一样的

 赞


 王尼玛的辅助 2019-12-30


请问blank标签怎么理解的呢？比如说一句话里面有空格，就把空格的索引加上吗？但有些好像是在句子里插空加空格，实在不怎么理解。还有就是必须在字符集里面加空格吗？

 赞


 何虎 2019-11-28


请问blank标签怎么理解的呢？比如说一句话里面有空格，就把空格的索引加上吗？但有些好像是在句子里插空加空格，实在不怎么理解

 1


 Genome 回复 何虎 2020-03-18


原文是：The activation of the extra unit is the probability of observing a ‘blank’，or no label. 所以blank并非指空格而是指无标签，或者可以叫占位符，空格可以自己定义，不过我不太确定ctc会不会混淆无标签和空格

 赞


 小灰灰超 2019-11-05


楼上+1，解码方法我现在只能求最大概率的一条通路

 1


 Supine 回复 小灰灰超 2019-11-14

我求最大概率的输出都是blank，不知道是什么情况

 赞

 小灰灰超 回复 Supine 2019-11-15

我按最大概率路径输出没问题，我是用 $p, preds_i, dx = preds.max(2)$ ，preds是网络输出，p和preds_idx分别是最大概率和对应的类别索引


 赞

查看全部 17 条回复


 Z N 2019-10-12

有解码函数吗

 1

 wowo 2019-06-19

1.1的ctcclose有一个zero inf的选项，会把不存在路径的情况忽略掉

 赞

 jiarenfy 2019-05-30

3.4不对吧

 赞

 人工智能技术干货 (作者) 回复 jiarenfy 2019-05-31



 **jiarenyf** 回复 人工智能技术干货 (作者) 2019-05-31

T ≥ 2U-1。

 赞

查看全部 7 条回复

 **Even** 2021-10-15


T在哪里设置？

 赞


 **不拉的彼得** 回复 Even 2021-11-23


文中：理论上T应大于等于2l+1

 1


 **偶尼酱** 2021-08-27


对于不定常的字符文本检测这里的ctcloss里面的四个参数应该怎么设置？我训练时会遇到RuntimeError: Expected tensor to have size at least 1 at dimension 1, but got size 0 for argument #2 'targets' (while checking arguments for ctc_loss_gpu),

 赞


 **吴筱澈** 2021-05-18


请问log_probs是怎么学习分成68个类的，后面的loss的计算也没有CHARS

 赞


 **我不是罐头** 2021-05-18

请问Dataloader读取数据的时候，然后每张图片的label长度都不一样，在pytorch用dataloader处理的时候会要求一个batch的数据都是一样的size，请问这种情况如何处理呢

 赞

 **cool** 回复 我不是罐头 2021-09-27


在label起始段和末尾段补0，偶数补相同的，奇数就前少后多

 赞


 **代码王子** 2020-12-22

可以的，有收获！

 赞

 **熊掌猪肉** 2020-10-28

3.3中， targets的shape如果设定为(N,S) 难道不可以通过打padding解决第一维长度不一致的问题吗？

 赞