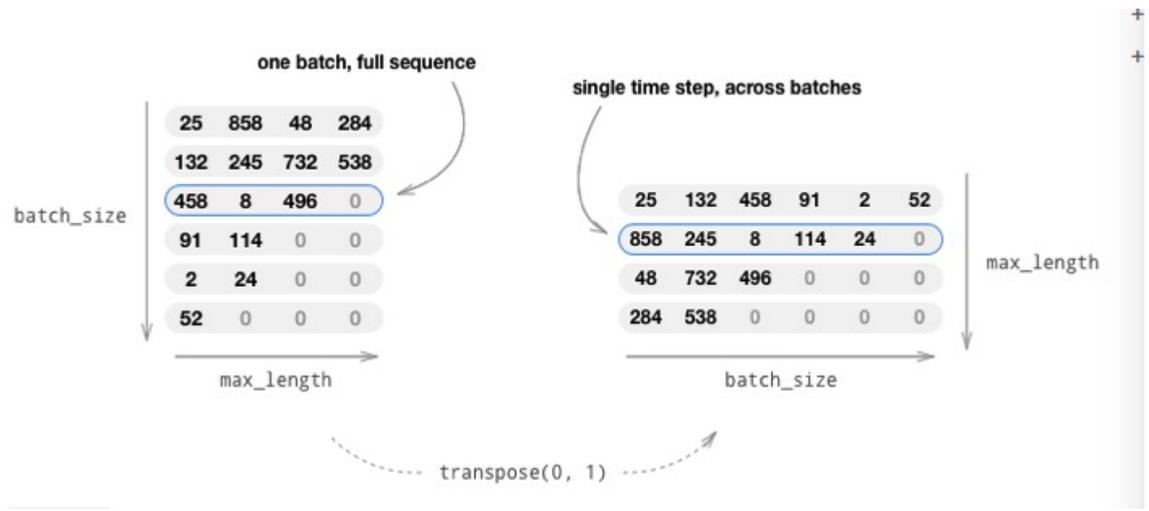
lstm pytorch梳理之 batch_first 参数 和torch.nn.utils.rnn.pack_padded_sequence

小萌新在看pytorch官网 LSTM代码时 对batch_first 参数 和torch.nn.utils.rnn.pack_padded_sequence 不太理解，

在回去苦学了一番，将自己消化过的记录在这，希望能帮到跟我有同样迷惑的伙伴

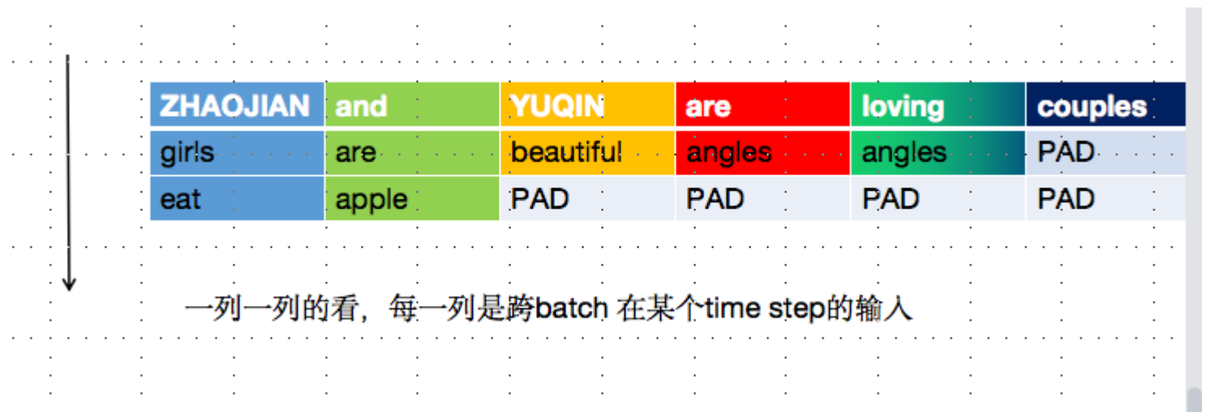
官方API：https://pytorch.org/docs/stable/nn.html?highlight=lstm#torch.nn.LSTM

- 参数
  - input_size
  - hidden_size
  - num_layers
  - bias
  - batch_first
  - dropout
  - bidirectional
- 特别说下batch_first ,参数默认为False，也就是它鼓励我们第一维不是batch，这与我们常规输入想悖，毕竟我们习惯的输入是(batch, seq_len, hidden_size),那么官方为啥会 这样子设置呢?



先不考虑hiddem_dim,左边图矩阵维度为batch_size * max_length, 6个序列，没个序列填充到最大长度4，经过转置后得到max_length *batch_size , 右图标蓝的一列 对应的就是 左图第二列，而左图第二列表示的是 每个序列里面第二个token，这样子有什么好处呢？相当于可以并行处理 每个句子在time step下时刻的计算，这样就 可以**并行过LSTM，从而一定程度上提高处理速度。**因为官网放的图例子里面数字都是句子token 索引化之后的，反而让人容易看晕，因而小萌新自己画了个好理解的图。一起看下图呀。

一共有3个句子，最大长度为6，我们之前习惯的是 按行看，我们现在按一列一列来看（就相当于转置啦）



time step 0接受的是[ZHAOJIAN girls eat];

time step1接收的是[and are apple];

time step2接受的是[YUQIN beautiful PAD];

time step3接收的是[are angles PAD];

以此类推。 现在这3个句子就可以**并行过LSTM**

pad_sequence

我们知道一个batch里的序列长度是不一致的，而LSTM是无法处理长度不同的序列的，需要pad操作用0把它们都填充成max_length长度。下图有3个句子，以最长的句子长度 6 作为max_length，其余句子都填充到max_length 。这是PAD的作用，很好理解。

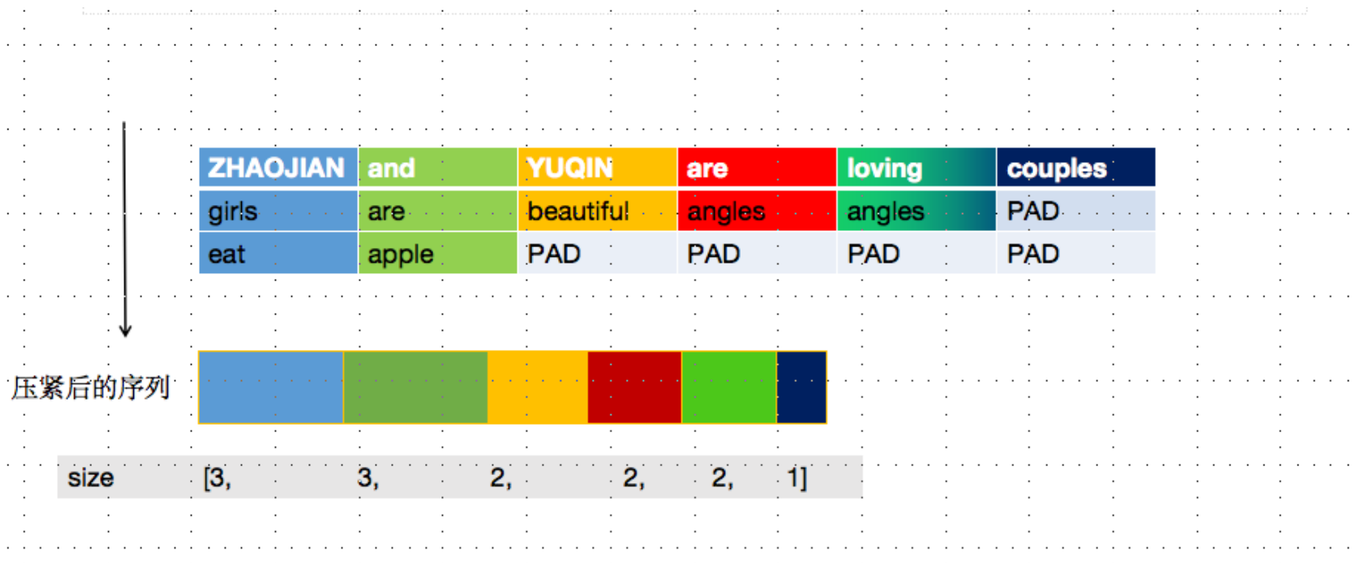
```
from torch.nn.utils.rnn import pack_padded_sequence ,pad_sequence ,pack_sequence
inputs = ["LIHUA went to The TsinghUA University",
          "Liping went to technical school ",
          "I work in the mall ",
          "we both have bright future"]
inputs.sort(key=lambda x:len(x.split()),reverse=True)
batch_size=len(inputs)
max_length=len(inputs[0].split())
lengths=[len(s.split()) for s in inputs]

word_to_idx={}
for sen in inputs:
    for word in sen.split():
        if word not in word_to_idx:
            word_to_idx[word]=len(word_to_idx)
idx=[]
for sentence in inputs:
    a=[word_to_idx[w] for w in sentence.split()]
    idx.append(a)
pprint(idx)
padded_sequence = pad_sequence([torch.FloatTensor(id) for id in idx], batch_first=True)
```

```
print(padded_sequence)
packed_sequence = pack_sequence([torch.FloatTensor(id) for id in idx])  # packed_sequence是PackedSequence的实例
```

pack_sequence

但带来一个问题，什么问题呢？ 对于长度小于MAX_LENGTH，经过PAD填充操作后的句子，会导致LSTM对它的表示多了很多无用的字符，如下图所示，我们希望的是在最后一个有用token 就输入句子的向量表示，而不是在很多PAD后才输入句子表示，这是pack就派上场了，可以理解成 将一个填充过的变长序列压紧.压缩的对象就是 padded suquence, 压缩后的输入将不含 0

看图更好理解 哦



那么，聪明如你肯定会觉得不对劲，这先 填充又 压紧, 这不是做无用功？其实不是哦，因为 pack 后可并不是一个简单的 Tensor 类型的数据，而是一个 "PackedSequence" 类型的 object，可以直接传给RNN。小萌新在苦逼地看RNN源码时，发现forward 函数 里上来就是判断 输入是否是 PackedSequence 的实例，进而采取不同的操作。如果输入是 PackedSequence，输出也是该类型。这里的输出类型都指的是 forward 函数的第一个返回值（每个time step 对应的hidden_state），第二个返回值(最后一个time step对应的hidden_state)的类型不管输入是不是 PackedSequence 类型，都是一样的。

pack_padded_sequence

pytorch里 有封装的更好的： torch.nn.utils.rnn.pack_padded_sequence(input, lengths, batch_first=False, enforce_sorted=True)

接下来说说这些参数作用

lengths： 该参数中各句子长度值的顺序要和对应的输入中的序列顺序一致

enforce_sorted： 默认值是 True，表示输入已经按句子长度降序排好序。如果输入在 pad 时没有顺序，那么此时在此处需要设置该值为 False,那么函数会再去排序

返回的对象是PackedSequence object。该类型的变量便可以直接喂给 RNN/LSTM等。

torch.nn.utils.rnn.pad_packed_sequence():之前的pack_padded_sequence 是先补齐到相同长度 再压紧，这个当然就是反过来，对压紧后的序列 进行扩充补齐操作。

注意：inputs是否排好序和 lengths参数和enforce_sorted 一定要对应起来。小萌新习惯将 inputs 按照长度先排好序，再将length 排好序enforce_sorted参数不去动它。

```
inputs.sort(key=lambda x:len(x.split()),reverse=True)
lengths=[len(s.split()) for s in inputs]
```

```
def forward(self, input_seq, input_lengths, hidden=None):
    # Convert word indexes to embeddings
    embedded = self.embedding(input_seq)
    # Pack padded batch of sequences for RNN module
    packed = nn.utils.rnn.pack_padded_sequence(embedded, input_lengths)
    # Forward pass through GRU
    outputs, hidden = self.gru(packed, hidden)
    # Unpack padding
    outputs, _ = nn.utils.rnn.pad_packed_sequence(outputs)
    # Sum bidirectional GRU outputs
    outputs = outputs[:, :, :self.hidden_size] + outputs[:, : ,self.hidden_size:]
    # Return output and final hidden state
    return outputs, hidden
```

----------恢复内容结束----------

分类: pytorch学习地图

好文要顶 关注我 收藏该文

打了鸡血的女汉子
关注 - 0
粉丝 - 9

4 0

+加关注

« 上一篇： bilstm+ctf 吐血的理论讲解
» 下一篇： pycharm 修改文件名-Refactor

posted @ 2020-12-08 11:29 打了鸡血的女汉子 阅读(2110) 评论(2) 编辑 收藏 举报