

迁移

🔗 3个文件尚未迁移

exec
load_data
base_cfgs

- Dataset 这个类的对应关系没找到
 - DataLoader与DataParallel
- 梯度这块 clip_grad_norm_
 - PyTorch中梯度是Tensor的属性，可以通过设置 requires_grad=True 使Tensor带有梯度。由于框架机制的不同，在MindSpore中，梯度和权重是互相独立的Tensor。因此在梯度裁剪时，MindSpore需要先获取梯度Tensor再进行裁剪。
 - PyTorch中能够实现原地梯度裁剪，Mindspore中梯度先提取出来，再裁剪

load_data

Dataset类正在对应Mindspore中对应类
`import torch.utils.data as Data`

```
11 import glob, json, torch, time
12 import torch.utils.data as Data
13
14
15 class DataSet(Data.Dataset):
16     def __init__(self, __C):
17         self.__C = __C
```

torch中Dataset类:

```
class Dataset(Generic[T_co]):
    """An abstract class representing a :class:`Dataset`.

    All datasets that represent a map from keys to data samples should subclass
    it. All subclasses should overwrite :meth:`__getitem__`, supporting fetching a
    data sample for a given key. Subclasses could also optionally overwrite
    :meth:`__len__`, which is expected to return the size of the dataset by many
    :class:`~torch.utils.data.Sampler` implementations and the default options
    of :class:`~torch.utils.data.DataLoader`. Subclasses could also
    optionally implement :meth:`__getitems__`, for speedup batched samples
    loading. This method accepts list of indices of samples of batch and returns
    list of samples.

    .. note::
        :class:`~torch.utils.data.DataLoader` by default constructs an index
        sampler that yields integral indices. To make it work with a map-style
        dataset with non-integral indices/keys, a custom sampler must be provided.
    """
```

base_cfgs

proc

Mindspore没有找到对应的api

```
torch.set_num_threads(2)
# ----- Seed setup
# fix pytorch seed
torch.manual_seed(self.SEED)
if self.N_GPU < 2:
    torch.cuda.manual_seed(self.SEED)
else:
    torch.cuda.manual_seed_all(self.SEED)
torch.backends.cudnn.deterministic = True
```

exec

train方法

Net网络有cuda方法和train方法吗?

```
net = Net(  
    self.__C,  
    pretrained_emb,  
    token_size,  
    ans_size  
)  
net.cuda()  
net.train()
```

DataParallel 这个方法没对应明白[mindspore.set_auto_parallel_context](#)

[比较与torch.utils.data.DataLoader的差异](#)

```
nn.utils.clip_grad_norm_(  
    net.parameters(),  
    self.__C.GRAD_NORM_CLIP  
)
```

[比较与torch.nn.utils.clipgrad norm的差异](#)