



推荐给大家！Pytorch编写代码基本步骤思想

 忆臻    
哈尔滨工业大学 计算机科学与技术博士在读

2,044 人赞同了该文章

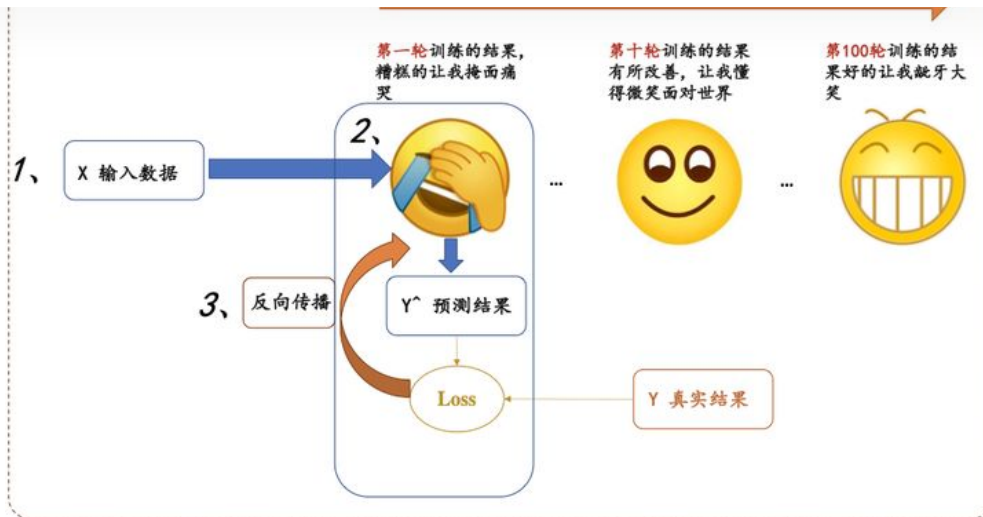
作者：忆臻 (哈工大SCIR实验室在读博士生)  
魏福煊 哈工大英才实验班本科生  
谢天宝 哈工大英才实验班本科生

一、前言

在我们要用pytorch构建自己的深度学习模型的时候，基本上都是下面这个流程步骤，写在这里让一些新手童鞋学习的时候有一个大局感觉，无论是从自己写，还是阅读他人代码，按照这个步骤思想（默念4大步骤，找数据定义、找model定义、(找损失函数、优化器定义)，主循环代码逻辑），直接去找对应的代码块，会简单很多。

二、基本步骤思想

所有的深度学习模型过程都可以形式化如下图：



分为四大步骤：

- 1、**输入处理模块** (X 输入数据，变成网络能够处理的Tensor类型)
- 2、**模型构建模块** (主要负责从输入的数据，得到预测的 $y^{\wedge}$ ，这就是我们经常说的前向过程)
- 3、**定义代价函数和优化器模块** (注意，前向过程只会得到模型预测的结果，并不会自动求导和更新，是由这个模块进行处理)
- 4、**构建训练过程** (迭代训练过程，就是上图表情包的训练迭代过程)

这几个模块分别与上图的数字标号1，2，3，4进行一一对应！

### 三、实例讲解

知道了上面的宏观思想之后，后面给出每个模块稍微具体一点的解释和具体一个例子，再帮助大家熟悉对应的代码！

#### 1. 数据处理

对于数据处理，最为简单的方式就是将数据组织成为一个。但许多训练需要用到mini-batch，直接组织成Tensor不便于我们操作。pytorch为我们提供了Dataset和Dataloader两个类来方便的构建。

##### torch.utils.data.Dataset

继承Dataset 类需要override 以下方法：

```
from torch.utils.data import Dataset
class trainDataset(Dataset):
    def __init__(self):
        ## constructor

    def __getitem__(self, index):
        ## 获得第index号的数据和标签

    def __len__(self):
        ## 获得数据量
```

##### torch.utils.data.Dataloader

DataLoader Batch。如果选择shuffle = True，每一个epoch 后， mini-Batch batch\_size 常见的使用方法如下：

```
dataLoader = DataLoader(dataset, shuffle=True, batch_size=32)
for i, data in enumerate(dataLoader, 0):
    x, y = data ## 同时获得数据和标签
```

## 2. 模型构建

所有的模型都需要继承torch.nn.Module， 需要实现以下方法：

```
class MyModel(torch.nn.Module):
    def __init__(self):
        super(MyModel, self).__init__()
        .....

    def forward(self, x):
        .....
        return ..

model = MyModel()
```

其中forward() 方法是前向传播的过程。在实现模型时，我们不需要考虑反向传播。

## 3. 定义代价函数和优化器

```
criterion = torch.nn.BCELoss(reduction='sum') ##代价函数
optimizer = torch.optim.Adam(model.parameters(), lr=0.0001, betas=(0.9,
0.999), eps=1e-08, weight_decay=0, amsgrad=False) ## 优化器
```

这部分根据自己的需求去参照doc

## 4、构建训练过程

pytorch的训练循环大致如下：

```
def train(epoch):## 一个epoc的训练
    for i, data in enumerate(dataLoader, 0):
        x, y = data ##取出minibatch数据和标签
        y_pred = model(x) ## 前向传播
        loss = criterion(y_pred, y) ##计算代价函数
        optimizer.zero_grad() ##清零梯度准备计算
        loss.backward() ##反向传播
        optimizer.step() ##更新训练参数
```

下面再用一个简单例子，来巩固一下：

```
class LogisticRegressionModel(torch.nn.Module):
    def __init__(self):
        super(LogisticRegressionModel, self).__init__()
        self.linear = torch.nn.Linear(1, 1)

    def forward(self, x):
        y_pred = F.sigmoid(self.linear(x))
        return y_pred

model = LogisticRegressionModel()

#-----#
criterion = torch.nn.BCELoss(size_average=False)
optimizer = torch.optim.SGD(model.parameters(), lr=0.01)
#-----#

for epoch in range(1000):
    y_pred = model(x_data)
    loss = criterion(y_pred, y_data)
    print(epoch, loss.item())

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

we shall talk about this later

```
x_data = torch.Tensor([[1.0], [2.0], [3.0]])
y_data = torch.Tensor([[0], [0], [1]])

#-----#
class LogisticRegressionModel(torch.nn.Module):
    def __init__(self):
        super(LogisticRegressionModel, self).__init__()
        self.linear = torch.nn.Linear(1, 1)

    def forward(self, x):
        y_pred = F.sigmoid(self.linear(x))
        return y_pred

model = LogisticRegressionModel()

#-----#
criterion = torch.nn.BCELoss(size_average=False)
optimizer = torch.optim.SGD(model.parameters(), lr=0.01)
#-----#

for epoch in range(1000):
    y_pred = model(x_data)
    loss = criterion(y_pred, y_data)
    print(epoch, loss.item())

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

2

Design model using Class  
inherit from nn.Moduleslides来自[https://www.bilibili.com/video/BV1Y7411d7Ys?](https://www.bilibili.com/video/BV1Y7411d7Ys?from=search&am&am&am&am&seid=3765076366663992699)

from=search&amp;am&amp;am&amp;am&amp;am&amp;seid=3765076366663992699

```
x_data = torch.Tensor([[1.0], [2.0], [3.0]])
y_data = torch.Tensor([[0], [0], [1]])

#-----#
class LogisticRegressionModel(torch.nn.Module):
    def __init__(self):
        super(LogisticRegressionModel, self).__init__()
        self.linear = torch.nn.Linear(1, 1)

    def forward(self, x):
        y_pred = F.sigmoid(self.linear(x))
        return y_pred

model = LogisticRegressionModel()

#-----#
criterion = torch.nn.BCELoss(size_average=False)
optimizer = torch.optim.SGD(model.parameters(), lr=0.01)
#-----#

for epoch in range(1000):
    y_pred = model(x_data)
    loss = criterion(y_pred, y_data)
    print(epoch, loss.item())

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()
```

3

Construct loss and optimizer  
using PyTorch APIslides来自[https://www.bilibili.com/video/BV1Y7411d7Ys?](https://www.bilibili.com/video/BV1Y7411d7Ys?from=search&am&am&am&am&seid=3765076366663992699)

from=search&amp;am&amp;am&amp;am&amp;am&amp;seid=3765076366663992699





机器学习算法与自然语言处理  
公号[机器学习算法与自然语言处理] 微信号yizhenotes

推荐阅读

推荐给大家！Pytorch编写代码基本步骤思想

一、前言 在我们要用pytorch构建自己的深度学习模型的时候，基本上都是下面这个流程步骤，写在这里让一些新手童鞋学习的时候有一个大局感觉，无论是从自己写，还是阅读他人代码，按照这个步...  
新智元 发表于新智元

有了这个工具，不执行代码就可以找PyTorch模型错误

几秒钟扫完代码，比训练一遍再找快多了。机器之心报道，编辑：陈萍、泽南。张量形状不匹配是深度神经网络机器学习过程中会出现的重要错误之一。由于神经网络训练成本较高且耗时，在执行代...  
机器之心 发表于机器之心



【开源框架】Pytorch图像分类从模型自定义到测试  
龙鹏-笔名... 发表于有三AI学...

23 条评论

切换为时间排序

评论由作者筛选后显示

刘洪普

2020-06-20

谢谢抬爱😘

33

Timothyxxx 回复 刘洪普

2020-06-20

刘老师是最最棒的！

2

Darcy 回复 刘洪普

2020-06-30

刘老师讲课很棒😎

赞

展开其他 3 条回复

Reinhold

2020-06-20

自己写的代码体量太小，总是看别人的写的很大的模型，也很困惑

17

忆臻 (作者) 回复 Reinhold

2020-06-20

多看

赞

忆臻 (作者) 回复 Reinhold

2020-06-20

熟能生巧

4

展开其他 2 条回复





8

知乎用户

2020-06-20

我看有蛮多人喜欢把训练的代码塞在model的定义里面，还有人喜欢单独定义一个trainer类来做，感觉都不如这样子分开写的更加清晰。

6

辉哥天天上班 回复 知乎用户

2021-08-09

我觉得模型就是模型，训练就是训练 肯定要分开。解耦合很重要

赞

Yubin Wang

2020-06-20

try pytorch\_lighting

3

吴京京

2020-06-22

It's a practical post for us to read pytorch code. If there is a group to sharing reading skills or knowledge, It will accelerate the learning rate.

2

知乎用户

2020-06-22

实用！简洁！明了！

2

JacobyLee

2020-06-20

刘老师学生过来打卡👉

2

丹尼尔小博士

2021-04-28

确实很棒

1

寒青

2020-06-21

谢谢，非常感谢，有帮助

1

知乎用户

2021-05-26

优秀

赞

Fang Suk

2020-06-21

我在入门框架时咋没看到这篇文章，感觉要复现顶会论文相近的结果还是很难的，毕竟调参是个玄学

赞

忆臻 (作者) 回复 Fang Suk

2020-06-22

调参就是玄学，没有办法~

3

