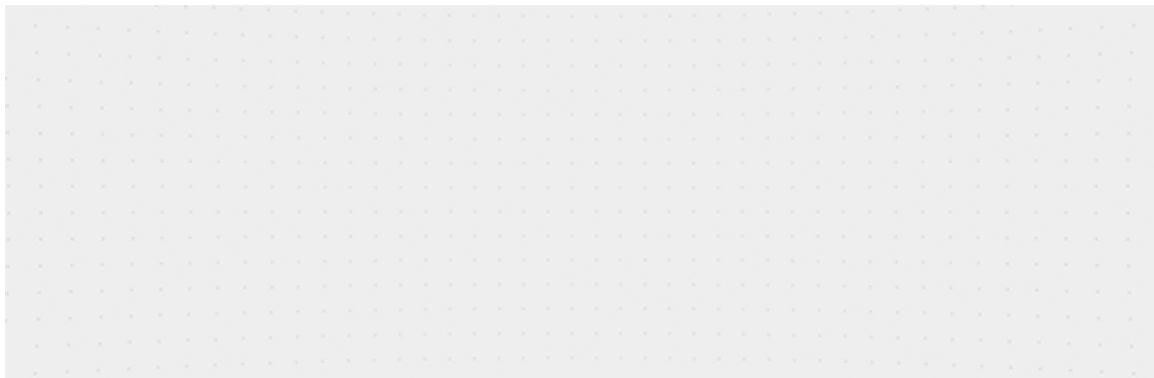


# 实操教程 | 深度学习pytorch训练代码模板(个人习惯)

CV开发者都爱看的 极市平台 2023-03-21 22:00:27 发表于广东 手机阅读 𠄎

↑ 点击蓝字 关注极市平台



作者 | wfnian@知乎 (已授权)

来源 | <https://zhuanlan.zhihu.com/p/396666255>

编辑 | 极市平台

## 极市导读

本文从参数定义，到网络模型定义，再到训练步骤，验证步骤，测试步骤，总结了一套较为直观的模板。 >>加入极市CV技术交流群，走在计算机视觉的最前沿

目录如下：

1. 导入包以及设置随机种子
2. 以类的方式定义超参数
3. 定义自己的模型
4. 定义早停类(此步骤可以省略)
5. 定义自己的数据集Dataset, DataLoader
6. 实例化模型，设置loss，优化器等
7. 开始训练以及调整lr
8. 绘图
9. 预测

## 一、导入包以及设置随机种子

```
import numpy as np
import torch
import torch.nn as nn
import numpy as np
import pandas as pd
from torch.utils.data import DataLoader, Dataset
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

import random
seed = 42
torch.manual_seed(seed)
np.random.seed(seed)
random.seed(seed)
```

## 二、以类的方式定义超参数

```
class argparse():
    pass

args = argparse()
args.epochs, args.learning_rate, args.patience = [30, 0.001, 4]
args.hidden_size, args.input_size= [40, 30]
args.device, = [torch.device("cuda:0"if torch.cuda.is_available() else"cpu"),]
```

## 三、定义自己的模型

```
class Your_model(nn.Module):
    def __init__(self):
        super(Your_model, self).__init__()
        pass

    def forward(self,x):
        pass
        return x
```

## 四、定义早停类(此步骤可以省略)

```
class EarlyStopping():
    def __init__(self,patience=7,verbose=False,delta=0):
        self.patience = patience
        self.verbose = verbose
        self.counter = 0
        self.best_score = None
        self.early_stop = False
        self.val_loss_min = np.Inf
        self.delta = delta
    def __call__(self,val_loss,model,path):
        print("val_loss={}".format(val_loss))
        score = -val_loss
        if self.best_score is None:
            self.best_score = score
            self.save_checkpoint(val_loss,model,path)
        elif score < self.best_score+self.delta:
            self.counter+=1
            print(f'EarlyStopping counter: {self.counter} out of {self.patience}')
            if self.counter>=self.patience:
                self.early_stop = True
        else:
            self.best_score = score
            self.save_checkpoint(val_loss,model,path)
            self.counter = 0
    def save_checkpoint(self,val_loss,model,path):
        if self.verbose:
            print(
                f'Validation loss decreased ({self.val_loss_min:.6f} --> {val_loss:.6f}).
            torch.save(model.state_dict(), path+'/'+ 'model_checkpoint.pth')
        self.val_loss_min = val_loss
```

## 五、定义自己的数据集Dataset,DataLoader

```
class Dataset_name(Dataset):
    def __init__(self, flag='train'):
        assert flag in ['train', 'test', 'valid']
        self.flag = flag
        self.__load_data__()
```

```

def __getitem__(self, index):
    pass

def __len__(self):
    pass

def __load_data__(self, csv_paths: list):
    pass
    print(
        "train_X.shape:{}\ntrain_Y.shape:{}\nvalid_X.shape:{}\nvalid_Y.shape:{}\n"
        .format(self.train_X.shape, self.train_Y.shape, self.valid_X.shape, self.valid_Y.shape)
    )

train_dataset = Dataset_name(flag='train')
train_dataloader = DataLoader(dataset=train_dataset, batch_size=64, shuffle=True)
valid_dataset = Dataset_name(flag='valid')
valid_dataloader = DataLoader(dataset=valid_dataset, batch_size=64, shuffle=True)

```

## 六、实例化模型，设置loss，优化器等

```

model = Your_model().to(args.device)
criterion = torch.nn.MSELoss()
optimizer = torch.optim.Adam(Your_model.parameters(),lr=args.learning_rate)

train_loss = []
valid_loss = []
train_epochs_loss = []
valid_epochs_loss = []

early_stopping = EarlyStopping(patience=args.patience,verbose=True)

```

## 七、开始训练以及调整lr

```

for epoch in range(args.epochs):
    Your_model.train()
    train_epoch_loss = []
    for idx,(data_x,data_y) in enumerate(train_dataloader,0):
        data_x = data_x.to(torch.float32).to(args.device)
        data_y = data_y.to(torch.float32).to(args.device)
        outputs = Your_model(data_x)
        optimizer.zero_grad()
        loss = criterion(data_y,outputs)
        loss.backward()
        optimizer.step()
        train_epoch_loss.append(loss.item())

```

```

train_loss.append(loss.item())
if idx%(len(train_dataloader)//2)==0:
    print("epoch={}/{}, {}/{} of train, loss={}".format(
        epoch, args.epochs, idx, len(train_dataloader), loss.item()))
train_epochs_loss.append(np.average(train_epoch_loss))

#=====valid=====
Your_model.eval()
valid_epoch_loss = []
for idx,(data_x,data_y) in enumerate(valid_dataloader,0):
    data_x = data_x.to(torch.float32).to(args.device)
    data_y = data_y.to(torch.float32).to(args.device)
    outputs = Your_model(data_x)
    loss = criterion(outputs,data_y)
    valid_epoch_loss.append(loss.item())
    valid_loss.append(loss.item())
valid_epochs_loss.append(np.average(valid_epoch_loss))

#=====early stopping=====
early_stopping(valid_epochs_loss[-1],model=Your_model,path=r'c:\\your_model_to_save')
if early_stopping.early_stop:
    print("Early stopping")
    break

#=====adjust lr=====
lr_adjust = {
    2: 5e-5, 4: 1e-5, 6: 5e-6, 8: 1e-6,
    10: 5e-7, 15: 1e-7, 20: 5e-8
}
if epoch in lr_adjust.keys():
    lr = lr_adjust[epoch]
    for param_group in optimizer.param_groups:
        param_group['lr'] = lr
    print('Updating learning rate to {}'.format(lr))

```

## 八、绘图

```

plt.figure(figsize=(12,4))
plt.subplot(121)
plt.plot(train_loss[:])
plt.title("train_loss")
plt.subplot(122)
plt.plot(train_epochs_loss[1:], '-o', label="train_loss")
plt.plot(valid_epochs_loss[1:], '-o', label="valid_loss")
plt.title("epochs_loss")

```

```
plt.legend()  
plt.show()
```

## 九、预测

```
# 此处可定义一个预测集的Dataloader。也可以直接将你的预测数据reshape,添加batch_size=1  
Your_model.eval()  
predict = Your_model(data)
```

公众号后台回复“**CVPR2023**”获取最新论文分类整理资源



极市平台

为计算机视觉开发者提供全流程算法开发训练平台，以及大咖技术分享、社区交流、竞...  
848篇原创内容

公众号

## 极市干货

**极视角动态：**「无人机+AI」光伏智能巡检，硬核实力遇见智慧大脑！ | 「AI 警卫员」上线，极视角守护龙大食品厂区安全！ | 点亮海运指明灯，极视角为海上运输船员安全管理保驾护航！

**CVPR2023：**浙大提出全归一化流模型PyramidFlow：高分辨率缺陷异常定位新范式 | 大模型流行之下，SN-Net给出一份独特的答卷 | 大脑视觉信号被Stable Diffusion复现图像！“人类的谋略和谎言不存在了”

**数据集：**自动驾驶方向开源数据集资源汇总 | 医学影像方向开源数据集资源汇总 | 卫星图像公开数据集资源汇总

## ● 获取真实CV项目经验 ●

**极市打榜**是极市平台推出的一种算法项目合作模式，至今已上线 100+ 产业端落地算法项目，已对接智慧城市、智慧工地、明厨亮灶等多个行业真实需求，算法方向涵盖目标检测、行为识别、图像分割、视频理解、目标跟踪、OCR等。

开发者可用平台上**已标注真实场景数据集+免费算力**，单个算法榜单完成算法开发后成绩达到指定标准便可获得**定额奖励**，成绩优异者可与极市平台签约合作获得**长期的算法分成收益**！

对于想丰富项目开发经验的小伙伴们，极市每个月还有**免费的CV实训周活动**，实战型的导师手把手教学，帮助大家学习从模型开发到部署落地全流程的AI算法开发！



扫码了解更多

点击阅读原文进入CV社区

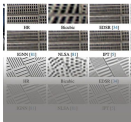
收获更多技术干货

阅读原文

喜欢此内容的人还喜欢

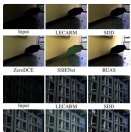
ICCV 2023 | 南开程明明团队提出适用于SR任务的新颖注意力机制（已开源）

极市平台



ICCV23 | 将隐式神经表征用于低光增强，北大张健团队提出NeRCo

极市平台



实践教程 | 从零开始用pytorch搭建Transformer模型

极市平台

