

《人工智能导论》大作业

任务名称： 人工智能导论大作业

完成组号： 12

小组人员： 汪健新 栗雪柔 王晓璐

完成时间： 2024.6.21

1. 任务目标

基于暴力图像检测数据集，构建一个 2 分类检测模型。通过暴力检测数据集和 AIGC 生成图像数据集训练该模型，使得该模型可以对数据集的图像进行不良内容检测与识别。同时也要模型具有一定的泛化能力：不仅能够识别与训练集分布类似的图像，对于 AIGC 风格变化、图像噪声、对抗样本等具有一定的鲁棒性；优化模型使其有合理的运行时间。

2. 具体内容

(1) 实施方案

数据读入：包括提供的训练集、验证集以及作为测试集的 AIGC 生成图像；

创建模型：使用 ResNet18 预训练模型获取输入特征数，再配置优化器和定义训练、验证、测试步骤；

训练模型：进行训练迭代保存最佳训练模型并开始训练；

测试模型：使用测试集通过测试准确率和测试损失评估模型；

(2) 核心代码分析

dataset.py 数据读入

图像的预处理：

```
class CustomDataset(Dataset):
    def __init__(self, split):
        #将图片转化为Tensor
        assert split in ["train", "val", "test"]
        data_root = "C:/Users/lenovo/Desktop/violence_224/" #修改为数据集文件路径
        self.data = [os.path.join(data_root, split, i) for i in os.listdir(data_root +
split)] #将不同数据集中的图像构建列表
        if split == "train":
            self.transforms = transforms.Compose([
                transforms.RandomHorizontalFlip(), # 随机翻转
                transforms.ToTensor(), # 将图像转换为Tensor
            ])
        else:
            self.transforms = transforms.Compose([
                transforms.ToTensor(), # 将图像转换为Tensor
            ])
```

将训练集、验证集、测试集的内容储存到数据加载器中

```
def train_dataloader(self):
    return DataLoader(self.train_dataset, batch_size=self.batch_size, shuffle=True,
num_workers=self.num_workers)

def val_dataloader(self):
    return DataLoader(self.val_dataset, batch_size=self.batch_size, shuffle=False,
num_workers=self.num_workers)

def test_dataloader(self):
    return DataLoader(self.test_dataset, batch_size=self.batch_size, shuffle=False,
num_workers=self.num_workers)
```

model.py 创建模型

使用 ResNet18 预训练模型

```
class ViolenceClassifier(LightningModule):
    def __init__(self, num_classes=2, learning_rate=1e-3):
        super().__init__()
        self.model = models.resnet18(pretrained=True)
        num_fts = self.model.fc.in_features
        self.model.fc = nn.Linear(num_fts, num_classes)
        # self.model = models.resnet18(pretrained=False, num_classes=2)

        self.learning_rate = learning_rate
        self.loss_fn = nn.CrossEntropyLoss() # 交叉熵损失
        self.accuracy = Accuracy(task="multiclass", num_classes=2)
```

配置优化器：

```
def configure_optimizers(self):
    optimizer = torch.optim.Adam(self.parameters(), lr=self.learning_rate) # 定义优
    化器

    return optimizer #返回优化器供LightningModule使用
```

分别定义训练步骤、验证步骤和测试步骤：

```
def training_step(self, batch, batch_idx):
    x, y = batch
    logits = self(x)
    loss = self.loss_fn(logits, y)
    self.log('train_loss', loss)
    return loss

def validation_step(self, batch, batch_idx):
    x, y = batch
    logits = self(x)
    loss = self.loss_fn(logits, y)
    acc = self.accuracy(logits, y)
    self.log('val_loss', loss)
    self.log('val_acc', acc)
    return loss

def test_step(self, batch, batch_idx):
    x, y = batch
    logits = self(x)
    loss = self.loss_fn(logits, y)
    acc = self.accuracy(logits, y)
    self.log('test_acc', acc)
    self.log('test_loss', loss)
    return loss
```

与训练步骤相比验证集和测试集需要计算准确率。

train.py 训练模型

规定迭代次数和使用设备，不断进行迭代将最佳模型保存到模型检查点中。

```
# 实例化训练器
trainer = Trainer(
    max_epochs=10,
    accelerator='cpu',
    devices=cpu_id,
    logger=logger,
    callbacks=[checkpoint_callback]
)

# 实例化模型
model = ViolenceClassifier(learning_rate=lr)
# 开始训练
trainer.fit(model, data_module)
```

test.py 测试模型

修改最佳模型检查点路径，使用测试集对模型进行评估，评估参数有测试准确率 test_acc 和测试损失 test_loss

```
data_module = CustomDataModule(batch_size=batch_size)
ckpt_root = "C:/Users/lenovo/Desktop/violenceDetection/train_logs/"
ckpt_path = ckpt_root +
"resnet18_pretrain_test/version_6/checkpoints/resnet18_pretrain_test-epoch=01-
val_loss=0.06.ckpt"
logger = TensorBoardLogger("test_logs", name=log_name)

model = ViolenceClassifier.load_from_checkpoint(ckpt_path)
trainer = Trainer(accelerator='cpu', devices=cpu_id)
trainer.test(model, data_module)
```

3. 测试结果:

与训练集同源的数据测试集结果:

```
Testing DataLoader 0: 100%|██████████| 9/9 [00:11<00:00, 0.76it/s]

+-----+-----+
| Test metric | DataLoader 0 |
+-----+-----+
| test_acc   | 0.9855465292930603 |
| test_loss  | 0.04176286980509758 |
+-----+-----+

Process finished with exit code 0
```

AIGC 生成图像测试集结果:

```
Testing DataLoader 0: 100%|██████████| 1/1 [00:06<00:00, 0.16it/s]

+-----+-----+
| Test metric | DataLoader 0 |
+-----+-----+
| test_acc   | 0.09900990128517151 |
| test_loss  | 3.8489720821380615 |
+-----+-----+

Process finished with exit code 0
```

4. 工作总结

(1) 收获、心得

完成不良内容识别检测模型项目后，我们深感 AI 的强大并且收获丰富，下面是我在完成该项目后的一些心得收获：

通过本项目，我深入理解了卷积神经网络及其变种在图像分类任务中的应用。我学习了如何调整模型结构、优化参数以及使用迁移学习来提高模型的性能。这些技术不仅限于图像分类，对其他计算机视觉任务也有借鉴意义。

在项目中，我学会了如何清洗数据集、划分训练集与测试集、进行图像增强以及如何处理不平衡数据。这些技能对于提高模型的泛化能力和鲁棒性至关重要。在项目实施过程中，我学会了如何有效地管理时间和资源，如何与团队成员进行沟通和协作。这些技能不仅对于学术研究有益，对于未来的职业生涯也具有重要意义。

处理不良内容图像时，我深刻认识到技术背后的伦理和责任。我始终遵循相关法律法规和道德准则，确保项目的合法性和道德性。同时，我也意识到作为技术人员，我们应该积极承担社会责任，利用技术为社会带来正面影响。

最后，通过完成这个不良内容图像检测项目，我不仅提高了自己的技术能力，还增强了对现实问题的认识和思考能力。这些经验和收获将对我的未来发展产生深远的影响。

(2) 遇到问题及解决思路

在执行文件时涉及到多线程问题，在 windows 系统下报错，在文件开始加上 `if __name__ == 'main'`，同时注意缩进问题；

获取图像的标签时系统自动补全的路径分割符号为 `\\`，所以将 `\\` 作为路径划分标准，并确定标签值的位置；

迭代次数过少时，对模型评估的结果较差，测试集评估的准确率较低，测试损失度相对较高，为获得较好的评估结果，可适当增加迭代次数。

5. 课程建议

希望课程能多安排一些实践项目，在平时积累一定的经验，期末大作业做起来也会更顺畅。同时也希望老师可以对实践内容进行一定地讲解，有利于学生学习了解工具的使用。

通过本课程的学习，我关于 ai 的理解和应用都更加深刻，再次感谢老师的教导与同学的陪伴。