华东师范大学数据科学与工程学院实验报告

课程名称：当代人工智能	年级：大三	上机实践成绩：
指导教师：李翔	姓名：甄逸飞	学号：10195501422
上机实践名称：实验五	上机实践日期：2022/7	

github 仓库： kdai-910hr/Contemporary-AI-Lab5 (github.com)

本次实验我并未完成全部内容，因为实在是不太明白该如何融合模型，网上仅有的一些代码参考不明白，CoLab 的 GPU 资源额度也用完了。所以我就只单独对文本以及在图片上做了模型训练，最后提交的预测结果也是纯文本 BERT 的预测结果，因为图片的模型在我划分的验证集上的精确度并没有纯文本的模型在验证集上的精确度高。

文本训练

数据集加载

根据 train.txt 获取作为训练集的所有序号，然后根据序号到 data 文件夹中，利用 Python open() 方法获取对应的文本数据。这里遇到了一个问题，无论使用 gbk 或者 utf-8 作为 txt 解码方式，都会有文本无法正常读取。我一开始的想法是使用 try except 语句，将报错的语句跳过掉，但是这样训练集就从四千多条一下子锐减到三千条左右，我觉得训练数据过少会导致模型过拟合，在查阅后，在 open() 方法中添加参数 errors='ignore' 来忽略掉无法解码的地方，最终得到了 4000 条数据:

```
    text_data.describe
```

```
<bound method NDFrame.describe of        id  label                                               text
0     4597      0  RT @AmitSwami77: The conspirators have an evil...
1       26      1  Waxwing trills, Chickadees calling "here sweet...
2     4383      0  @NYSE is looking a little despondent today...?...
3      212      2  FERVENT | S,M,L | 140k free PLASTIC CLIP, keyc...
4     2626      2  Nice day chilling in the park yesterday reliev...
...    ...    ...                                                ...
3995  3944      0  RT @IraqiSecurity: Car bomb in Aden Sq, Kadhim...
3996   945      2  #Remember no one can make you #feel #inferior ...
3997   695      2  RT @RapFavorites: Ye turned Valentine's Day in...
3998  4874      0  RT @cybercuichi: Miracles and Faith have a hol...
3999   246      0  RT @stoned_satan: https://t.co/y8SXRiJrP8 #cig...

[4000 rows x 3 columns]>
```

同理获取测试集，得到 511 条数据：

```
print(test_data)
```

```
      id                                               text
0      8  Energetic training today with our San Antonio ...
1   1576  Let your voice be heard! 18+ #endsuicide #blit...
2   2320  RT @Austin_Powers__: Shark Week would be so mu...
3   4912             #TheTruthCaster http://t.co/S8jvqpKq5h
4   3821  RT @jarpad: Hey #WBSDCC look what we're up to!...
..   ...                                               ...
506 1048        ??? #stunned #sunglasses #gafas #gafasdesol
507 1059  Seeing @nbdbnb in a few days #excited #or #ter...
508 1485  And the #dragon guarding our #heart is called ...
509 3195  RT @NWSAlbuquerque: Isolated strong to severe ...
510 2029  Cliff dwelling above Peacocks Gallop (recently...

[511 rows x 2 columns]
```

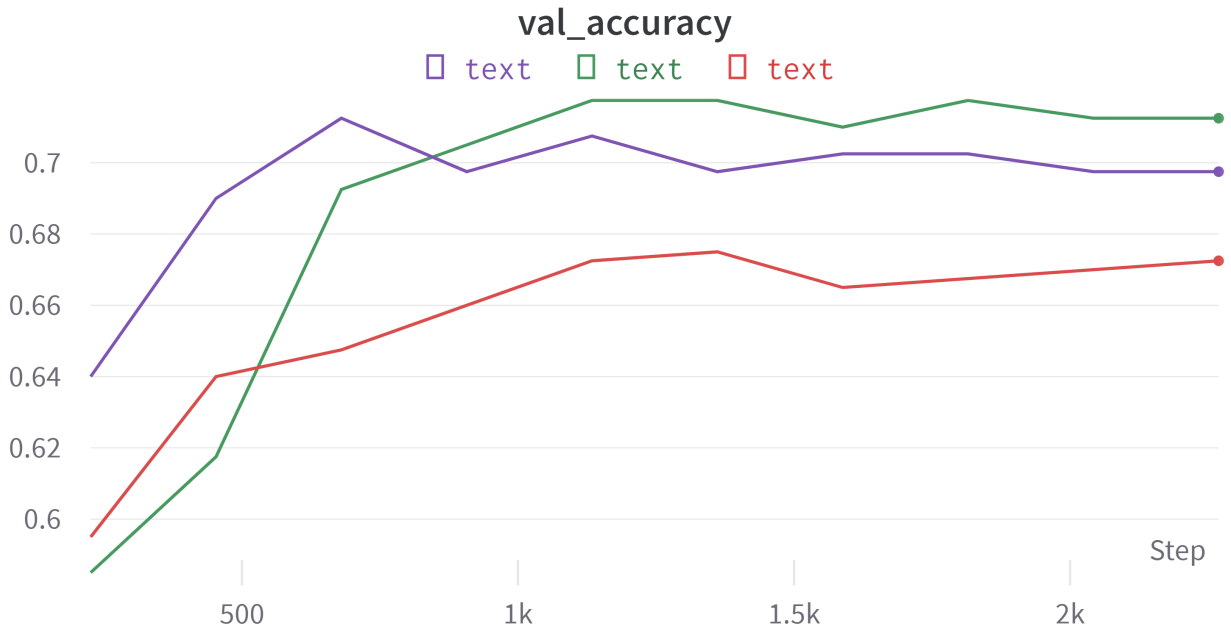
然后使用 BertTokenizer 将训练集数据以及测试集数据都转换为 ids

```
sentences = text_data.text.values
input_ids, attention_masks = [], []
for sent in sentences:
    encoded_dict = tokenizer.encode_plus(
                        sent,
                        add_special_tokens = True, # 添加 cls 与 sep
                        max_length = 130,
                        padding= "max_length",           # 使用 0-padding 统一长度
                        return_attention_mask = True,    # 构建注意力以及掩码
                        return_tensors = 'pt',     # 以 pytorch 的 tensor 类型返回
                   )
    input_ids.append(encoded_dict['input_ids'])
    attention_masks.append(encoded_dict['attention_mask'])
```

然后将数据转为 dataloader，用于模型训练。

学习使用的参数：

```
model = BertForSequenceClassification.from_pretrained( # 使用预训练的 Bert 模型
        "bert-base-uncased",
        num_labels = 3,
        output_attentions = True,
        output_hidden_states = True,
    )
optimizer = AdamW(model.parameters(), # 优化器使用 transformers 包的 AdamW
                  lr = wandb.config.learning_rate,
                  eps = wandb.config.epsilon
                )
wandb.init(project="multimodel_emotion_recognize",
      name="text",
      config={"epochs":10, "batch_size":16, "learning_rate":  2e-6, "epsilon": 1e-7})
# 学习率选用 2e-6, 训练 10 个 epochs, 是下图中验证集上正确率最高的那一条的参数
```



最终训练结果，模型实际在第九个 epoch 就训练不下去了：

```
======== Epoch 10 / 10 ========
Training...
...
Train  Accuracy: 0.77

Average training loss: 0.61
Training epcoh took: 0:01:22

Running Validation...
Accuracy: 0.70
Validation Loss: 0.71
Validation took: 0:00:03
```

模型预测

```
for batch in test_dataloader:
  b_input_ids = batch[0].to(device)
  b_input_mask = batch[1].to(device)

  with torch.no_grad():
      output = model1(b_input_ids,
              token_type_ids=None,
              attention_mask=b_input_mask)
      output_ = output[0].cpu().numpy() # 同样需要先转成 cpu 才能继续计算
      for i in output_:
        preds.append(i.argmax())
```

这里和上面一样，从 model 获得到 output 以后需要使用 cpu() 方法，这样数据才能在 cpu 下做计算，否则会报错。

```
def label2tag(label): # 将 label 转换回实际的 tag
  label = int(label)
  if label == 0:  return "negative"
  elif label == 1:  return "neutral"
  else: return "positive"

with open('test_with_label.txt', 'w') as f:  # 输出预测结果
  f.write("guid,tag\n")
  for i in range(len(test_list)):
     f.write(str(test_list[i])+","+label2tag(preds[i])+"\n")
```

图片训练

数据集加载

直接加载图片数据，然后打标签有点麻烦，torchvision 提供了 ImageFolder() 方法，可以将根文件夹内的子文件夹名称作为子文件夹内图片的标签，因此考虑将图片保存为如下路径结构：

- pic
  - train
    - 0
    - 1
    - 2
  - validate
    - 0
    - 1
    - 2

首先想读取文本数据那样，获取训练集的序号，然后使用 shutil 将图片复制到指定路径的文件夹内。

```
def copy_img(dataFrame, suffix, option):
   for i in range(len(dataFrame)):
        target_path = pic_path+"/"+suffix+"/"+str(dataFrame.iloc[i].label)
        target_path = pic_path+"/"+suffix+"/"
        source_path = dataFrame.iloc[i].img
        shutil.copy(source_path, target_path)
```

然后分别使用 ImageFolder("pic/train") 以及 ImageFolder("pic/validate") 即可获取两类有标签的数据。而且，ImageFolder() 允许提供对输入图片做转换的参数 transform，继承自 torchvision.transforms:

```
train_transform = transforms.Compose([
      transforms.Resize((224,224)),      # 将大小不一的图片固定为 224*224
      transforms.RandomHorizontalFlip(p=0.5), # 将图片随机水平翻转
      transforms.RandomVerticalFlip(p=0.5), # 将图片随机上下翻转
      transforms.ToTensor(),  # 转为张量
      transforms.Normalize(  # 标准化
          mean=[0.485,0.456,0.406],
          std=[0.229,0.224,0.225]
  )
])
```

将图片随机翻转可视为 Data Arugumentation，一般用来增加特征，减少过拟合。然后使用 torch.utils.data 的 DataLoader() 方法读取 ImageFolder() 后返回的数据：

```
img_train_loader = DataLoader(img_train_datasets, batch_size=16, shuffle=True,
num_workers=0)
img_validate_loader = DataLoader(img_validate_datasets, batch_size=16, shuffle=True,
num_workers=0)
```

这里的 num_workers 默认设为 0，否则会读取数据的线程与实际处理的线程不匹配，我有遇到报错：

```
RuntimeError: Trying to resize storage that is not resizable
```

模型训练

本次实验使用的图片训练模型使用的是 resnet32 模型，最后一层全连接层将输出 out_feature 设为 label 种类数 3：

```
num_classes = 3
model = models.resnet34(pretrained=pretrained)
model.fc = nn.Linear(512, num_classes)
```

参数选择:

```
lr = 1e-4
epochs = 10
optimizer = optim.Adam(model.parameters(), lr=lr)
criterion = nn.CrossEntropyLoss() # 使用交叉熵
```

在验证集上的结果，在第十个 epoch 趋于稳定：

```
Epoch 9 of 10
Training:
Train Loss: 0.7921
Train Accuracy: 0.6489
Validating:
Validate Loss: 0.8034
Validate Accuracy: 0.6425
```

```
-------------------------------------------------
Epoch 10 of 10
Training:
Train Loss: 0.789
Train Accuracy: 0.6497
Validating:
Validate Loss: 0.8022
Validate Accuracy: 0.6425
```

实验总结

本实验我没能实现文本与图片模型的融合，只是各在一种上使用模型训练了一下，在九比一划分训练集与验证集后，两种模型分别在在验证集上得到了 71% 与 64.25% 的精确度。感觉比起图片，模型更容易捕捉到文本中携带的情感信息，因此我在最后输出预测标签的时候使用了文本模型做预测。

对于融合模型，我想可能是需要将文本的输入向量与图片的输入向量做归一化后，然后可以按权重加在一起，文本的权重应该可以设大一些(因为我经过试验，感觉单一模型下文本比图片更能捕捉到情感信息)，最简单的话可以直接经过一个隐藏层以后，就用全连接输出到 3 个 out_features 上分别代表 negative，neutral 和 positive，但我实际学不太明白该怎么用代码实现，GPU也不够用了，就没能做完，实在是不好意思。