人工智能实验5

庄欣熠 10225501451

github地址 https://github.com/zxyyxb/Multimodal_Sentiment.git

模型的设计

文本处理模型（基于深度学习） 对于文本数据，模型首先通过 Tokenizer 将文本转换为数字序列，并使用 Embedding 层进行词嵌入。通过 GlobalAveragePooling1D 层对文本序列进行全局平均池化，将文本数据的维度压缩，从而捕捉到文本中的全局语义特征。接下来通过全连接层进行分类。该文本模型结构简单且有效，适合情感分析任务，尤其适合处理中短文本的情感分析。

图像处理模型（卷积神经网络） 对于图像数据，模型使用卷积神经网络（CNN）进行特征提取，Conv2D 层能够自动识别图像中的局部特征（如边缘、纹理等），MaxPooling2D 层则用于减少计算量并提取最重要的特征。通过 Flatten 层将图像特征转化为一维，再通过全连接层进行分类。这种结构使得模型能够有效地从图像中提取情感相关特征，尤其是在情感分析任务中，图像可能通过面部表情、颜色或其他视觉信息传达情感。

模型设计考虑到了扩展性和灵活性，在文本和图像处理的部分，可以根据需要调整文本的最大长度、词汇量或图像的大小。此外，使用深度学习框架（如TensorFlow/Keras）使得模型可以更容易地进行调试、优化和迁移到更复杂的网络结构上。

遇到的bug

1、内存的不足
在加载大量图像和文本数据时，尤其是当数据集较大时，可能会出现内存不足的问题，导致程序崩溃或无法正常运行。

解决方案：
对图像和文本数据进行批处理，逐步加载和训练。
使用数据生成器（ImageDataGenerator）逐批加载图像，而不是将所有图像加载到内存中。
减少批处理大小（batch_size）或降低图像分辨率。

2、图像尺寸不一致
load_img 函数在加载图像时，可能会遇到图像尺寸不一致的问题。
使用 target_size=(img_height, img_width) 进行调整。
但某些图像可能因过度压缩、损坏或非标准格式而导致加载失败。可以添加异常处理机制。如果图像无法加载，可以跳过该样本或替换为默认图像。

```
try:
    img = load_img(img_path, target_size=(img_height, img_width))
    img_array = img_to_array(img)
except Exception as e:
    print(f"[Warning] 图像加载失败: {img_path}, 错误信息: {e}")
    continue
```

3、融合结果不合理 在模型预测时，融合了文本和图像模型的结果，但简单的平均方法可能并不总是有效。文本或图像模型可能有更高的权重。
可以通过权重融合，调整文本和图像模型在最终预测中的贡献比例。

```
text_weight = 0.7  # 假设文本模型表现较好
image_weight = 0.3  # 假设图像模型表现较差
ensemble_probs_val = text_weight * text_probs_val + image_weight * image_probs_val
```

结果分析

将学习率和epoch提高后，准确率略有上升（学习率从0.001到0.05）

```
Epoch 2/5
225/225 [==============================] - 43s 193ms/step - loss: 0.9028 - accuracy: 0.6061 - val_loss: 0.9093 - val_accuracy: 0.6275
Epoch 3/5
225/225 [==============================] - 44s 195ms/step - loss: 0.8114 - accuracy: 0.6414 - val_loss: 0.8976 - val_accuracy: 0.6225
Epoch 4/5
225/225 [==============================] - 44s 194ms/step - loss: 0.6670 - accuracy: 0.7178 - val_loss: 0.9109 - val_accuracy: 0.6250
Epoch 5/5
225/225 [==============================] - 43s 190ms/step - loss: 0.5025 - accuracy: 0.7944 - val_loss: 0.9688 - val_accuracy: 0.6100
13/13 [==============================] - 0s 1ms/step
13/13 [==============================] - 1s 36ms/step

融合后在验证集上的准确率: 0.6450
16/16 [==============================] - 0s 1ms/step
16/16 [==============================] - 1s 36ms/step

 Epoch 2/5
 225/225 [==============================] - 43s 192ms/step - loss: 0.9118 - accuracy: 0.5936 - val_loss: 0.9121 - val_accuracy
 Epoch 3/5
 225/225 [==============================] - 44s 195ms/step - loss: 0.9119 - accuracy: 0.5936 - val_loss: 0.8836 - val_accuracy
 Epoch 4/5
 225/225 [==============================] - 45s 198ms/step - loss: 0.9132 - accuracy: 0.5936 - val_loss: 0.8746 - val_accuracy
 Epoch 5/5
 225/225 [==============================] - 45s 200ms/step - loss: 0.9124 - accuracy: 0.5936 - val_loss: 0.8804 - val_accuracy
 13/13 [==============================] - 0s 2ms/step
 13/13 [==============================] - 1s 36ms/step

 融合后在验证集上的准确率: 0.6700
 16/16 [==============================] - 0s 1ms/step
 16/16 [==============================] - 1s 38ms/step
```

通过调整txt和img的权重大小，可以得到更高的准确率

```
222
223     # 设置文本和图像的权重
224     w_text = 0.8
225     w_image = 0.2
226
227     # 融合模型预测的结果时，使用加权平均
228     ensemble_probs_val = (w_text * text_probs_val + w_image * image_probs_val) / (w_text + w_image)
```

问题 25 输出 调试控制台 终端 端口 Pyt

```
225/225 [==============================] - 55s 244ms/step - loss: 0.9123 - accuracy: 0.5936 - val_loss: 0.8812 - val_accuracy: 0.6275
13/13 [==============================] - 0s 2ms/step
13/13 [==============================] - 1s 41ms/step

融合后在验证集上的准确率: 0.6575
16/16 [==============================] - 0s 1ms/step
16/16 [==============================] - 1s 40ms/step
```

```
222
223     # 设置文本和图像的权重
224     w_text = 0.2
225     w_image = 0.8
226
```

问题 25 输出 调试控制台 终端 端口

```
225/225 [==============================] - 42s 185ms/step - lo
Epoch 4/5
225/225 [==============================] - 42s 189ms/step - lo
Epoch 5/5
225/225 [==============================] - 47s 208ms/step - lo
13/13 [==============================] - 0s 1ms/step
13/13 [==============================] - 1s 39ms/step

融合后在验证集上的准确率: 0.6275
16/16 [==============================] - 0s 830us/step
16/16 [==============================] - 1s 36ms/step
```

可以看到通过增大txt的权重比例，准确率有了提高

消融实验结果

```
113/113 [==============================] - 1s 4ms/step - loss: 0.9537 - acc
Epoch 2/5
113/113 [==============================] - 0s 3ms/step - loss: 0.8870 - acc
Epoch 3/5
113/113 [==============================] - 0s 3ms/step - loss: 0.8381 - acc
Epoch 4/5
113/113 [==============================] - 0s 3ms/step - loss: 0.7372 - acc
Epoch 5/5
113/113 [==============================] - 0s 3ms/step - loss: 0.6114 - acc

==== 训练图像模型 ====
Epoch 1/5
225/225 [==============================] - 44s 195ms/step - loss: 21.7401 -
Epoch 2/5
225/225 [==============================] - 41s 184ms/step - loss: 0.9122 -
Epoch 3/5
225/225 [==============================] - 42s 185ms/step - loss: 0.9134 -
Epoch 4/5
225/225 [==============================] - 42s 189ms/step - loss: 0.9119 -
Epoch 5/5
```

只训练txt的训练速度要显著快于训练图像

这是因为图像数据通常较为复杂，需要更大的计算资源。图像模型能够提取视觉上的情感信息，但对于纯文本

内容（如情感词汇或文本上下文）缺乏理解。而文本模型对于处理语言数据非常高效，能够更精确地分析情感。文本模型通常训练速度较快，因为它处理的是结构化的文本数据，而图像处理需要较复杂的计算。

只训练txt的结果

```
ated. Please use tf.compat.v1.executing_eagerly_outside_functions instead.

113/113 [==============================] - 1s 3ms/step - loss: 0.9698 - accuracy: 0.5803 - val_loss: 0.868
Epoch 2/3
113/113 [==============================] - 0s 2ms/step - loss: 0.8966 - accuracy: 0.5936 - val_loss: 0.858
Epoch 3/3
113/113 [==============================] - 0s 2ms/step - loss: 0.8680 - accuracy: 0.5950 - val_loss: 0.840
13/13 [==============================] - 0s 1ms/step

文本模型在验证集上的准确率: 0.6450
16/16 [==============================] - 0s 933us/step
```

只训练img的结果

```
WARNING:tensorflow:From E:\python\lib\site-packages\keras\src\engine\base_layer_utils
ated. Please use tf.compat.v1.executing_eagerly_outside_functions instead.

225/225 [==============================] - 42s 184ms/step - loss: 47.4466 - accuracy:
Epoch 2/3
225/225 [==============================] - 41s 183ms/step - loss: 0.9164 - accuracy:
Epoch 3/3
225/225 [==============================] - 41s 183ms/step - loss: 0.9118 - accuracy:
13/13 [==============================] - 1s 33ms/step

图像模型在验证集上的准确率: 0.6275
16/16 [==============================] - 1s 33ms/step
```

可以发现单独训练两种数据的结果都比一起训练两种数据的结果要差

总结

本实验展示了文本和图像两种不同模态数据在情感分析中的应用，并通过多模态融合方法取得了较好的效果。通过调整模型结构、优化训练策略和改进融合方式，我成功提高了模型的准确性。虽然文本模型在处理情感分析任务时更为高效，但结合图像模型的多模态方法可以充分发挥两种模态的优势，进一步提高情感分析的效果。

未来的工作可以进一步探索更复杂的多模态融合方法，优化模型的计算效率和准确度，并在更大规模的数据集上进行验证和优化。