《人工智能导论》课程项目

多模态的电影分类建模

小组成员：

张凯旋 517021910156　　周稚宜 517021910665

刘浩文 517021911065　　董弈伯 517021910668

2020 年 10 月 28 日

摘要

大量的电影、多种多样的电影类别让电影的分类成了一大难题，需要采用综合多种信息模式的深度学习方法进行精确分类，即**多模态深度学习**。我们基于 $PyTorch$ 深度学习平台与 $Kaggle$ 电影数据集，训练了一个主体为 CNN 和 $LSTM$ 的电影分类多模态深度学习模型。训练过程中，我们尝试了多种经典深度学习模型，如 $ResNet$ 和 $FastText$，多种训练方法，如微调和冻结，和多种多模态融合方法，期望达到理想的测试集分类准确率。此外，我们还测试了单模态分类的准确率作为对比，观察多模态深度学习对分类效果的提升。

关键词：　**多模态学习**　**深度学习**　**CNN**　**LSTM**

1 Introduction

计算机、智能手机普及的现在，电视剧、电影等娱乐方式已经成为了人们生活的一部分，也理所当然成为许多视频软件的发展契机。然而，大量的电影、多种多样的电影类别让电影的分类成了一大难题。依靠人力分类视频软件中海量的电影是不现实的，而复杂的样本与类别使传统的机器学习方法也难以派上用场，因此现如今的技术都是基于深度学习模型进行电影分类。电影包含图像、台词、配音、配乐等多种信息，如果仅仅使用图像或台词等单一信息来对电影进行分类，势必会损失电影中许多重要的信息，造成分类准确率低下，因此需要采用综合多种信息模式的深度学习方法，即**多模态深度学习**。

模态是指人接受信息的特定方式。单模态的表示学习负责将信息表示为计算机可以处理的数值向量或者进一步抽象为更高层的特征向量，而多模态（Multimodal Deep Learning）表示学习是指通过利用多模态之间的互补性，剔除模态间的冗余性，从而学习到更好的特征表示。由于多媒体数据往往是多种信息的传递媒介(例如一段视频中往往会同时使得文本信息、视觉信息和听觉信息得到传播)，多模态学习已逐渐发展为多媒体内容分析与理解的主要手段。

我们基于 $PyTorch$ 深度学习平台与 $Kaggle$ 电影数据集，训练了一个主体为 CNN 和 $LSTM$ 的电影分类多模态深度学习模型。我们首先将电影数据集分为海报与文本两部分进行预处理，之后对多模态深度学习模型进行训练，CNN 对海报进行特征提取，$LSTM$ 对文本进行特征提取，融合层对图像特征与文字特征进行多模态融合。训练过程中，我们尝试了多种经典深度学习模型，如 $ResNet$ 和 $FastText$，多种训练方法，如微调和冻结，和多种多模态融合方法，期望达到理想的测试集分类准确率。此外，我们还测试了单模态分类的准确率（分别图像或文本）作为对比，观察多模态深度学习对分类效果的提升。

我们的工作有：

- 成功复现代码。

- 尝试了多种经典深度学习模型，如 $ResNet$ 和 $FastText$。
- 尝试了多种训练方法，如微调和冻结。
- 尝试了多种多模态融合方法，如最大值融合、$sigmoid$ 激活融合。
- 测试了单模态分类的准确率（分别图像或文字）作为对比，观察多模态深度学习对分类效果的提升。

2 Idea & Rationale

2.1 数据预处理

在进行图像预处理时，发现下载的图片有很多空文件，图片的格式也有灰度图和 RGB 图，所以为了能够利用这些图片，我们删除了空文件，并将灰度图和 RGB 图统一格式。在进行文本预处理时，为了提升训练效率，我们删除标点符号，将全部字符改为小写，去掉所有的非字母字符和 NLTK 中的停止符号，进行了词形还原，将多个连续空格简化为一个。我们使用了 $Keras\ Tokenizer\ API$ 对文本进行向量化，并且将每一个文本（影评）pad 成相同的长度。

2.2 预训练

原作分别在 $GLoVe$ 模型（$Global\ Vector$ 模型）、$fasttext$ 和 $Word2Vec$ 模型上进行了测试，发现 $Word2Vec$ 效果更好。$Word2Vec$ 本质上是一个神经网络语言模型，基于该语言模型可以进行分布式词向量的训练。特点为提出了新的方法进行训练：$CBOW$ 模型（$Continuous\ Bag-of-Words\ Model$，在已知当前词的上下文的情况下预测当前词）和 $Skip-gram$ 模型（$Continuous\ Skip-gram\ Model$，在已知当前词的情况下，预测其上下文）。同时为了加快训练速度，使用 $Hierarchical\ Softmax$ 和 $Negative\ Sampling$ 这两种 $tricks$。

2.3 神经网络

卷积神经网络（Convolutional Neural Network, CNN）是一种前馈神经网络，它的人工神经元可以响应一部分覆盖范围内的周围单元，对于大型图像处理有出色表现。卷积神经网络由一个或多个卷积层和顶端的全连通层（对应经典的神经网络）组成，同时也包括关联权重和池化层（pooling layer）。这一结构使得卷积神经网络能够利用输入数据的二维结构。与其他深度学习结构相比，卷积神经网络在图像和语音识别方面能够给出更好的结果。这一模型也可以使用反向传播算法进行训练。相比较其他深度、前馈神经网络，卷积神经网络需要考量的参数更少，使之成为一种颇具吸引力的深度学习结构。

长短期记忆网络（LSTM, Long Short-Term Memory）是一种时间循环神经网络，是为了解决一般的 RNN（循环神经网络）存在的长期依赖问题而专门设计出来的，所有的 RNN 都具有一种重复神经网络模块的链式形式。在标准 RNN 中，这个重复的结构模块只有一个非常简单的结构，例如一个 tanh 层。

2.4 多模态融合

除了参考代码提供的拼接融合方式外，我们希望尝试更多的多模态融合方法，如分量加法融合、分量最大值融合等。

多模态融合旨在将多个模态信息整合以得到一致、公共的模型输出，是多模态领域的一个基本问题。多模态信息的融合能获得更全面的特征，提高模型鲁棒性，并且保证模型在某些模态缺失时仍能有效工作。

目前，多模态数据融合主要有三种融合方式：前端融合 (early-fusion) 或数据水平融合 (data-level fusion)、后端融合 (late-fusion) 或决策水平融合 (decision-level fusion) 和中间融合 (intermediate-fusion)。

前端融合将多个独立的数据集融合成一个单一的特征向量，然后输入到机器学习分类器中。由于多模态数据的前端融合往往无法充分利用多个模态数据间的互补性，且前端融合的原始数据通常包含大量的冗余信息。因此，多模态前端融合方法常常与特征提取方法相结合以剔除冗余信息，如主成分分析（PCA）、最大相关最小冗余算法（mRMR）、自动解码器（Autoencoders）等。

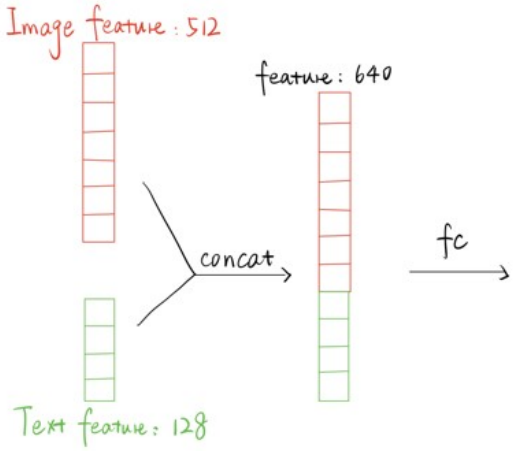
后端融合则是将不同模态数据分别训练好的分类器输出打分 (决策) 进行融合。这样做的好处是，融合模型的错误来自不同的分类器，而来自不同分类器的错误往往互不相关、互不影响，不会造成错误的进一步累加。常见的后端融合方式包括最大值融合 (max-fusion)、平均值融合 (averaged-fusion)、贝叶斯规则融合 (Bayes' rule based) 以及集成学习 (ensemble learning) 等。其中集成学习作为后端融合方式的典型代表，被广泛应用于通信、计算机识别、语音识别等研究领域。

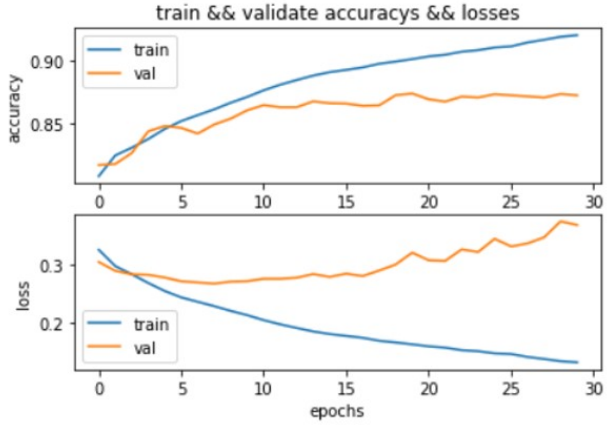
中间融合是指将不同的模态数据先转化为高维特征表达，再于模型的中间层进行融合。以神经网络为例，中间融合首先利用神经网络将原始数据转化成高维特征表达，然后获取不同模态数据在高维空间上的共性。中间融合方法的一大优势是可以灵活的选择融合的位置。

3 Experiments & Discussion

3.1 复现参考代码

我们先依据参考代码复现了使用海报图片和概述文本的多模态信息预测电影类型的模型。CNN 网络提取出图像特征，$LSTM$ 网络提取出文本特征。参考代码使用两特征拼接的方式融合。



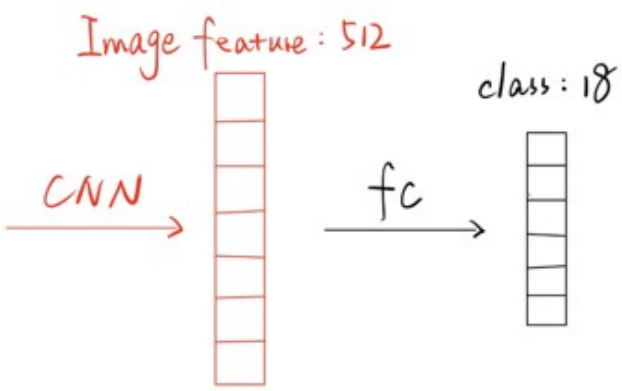


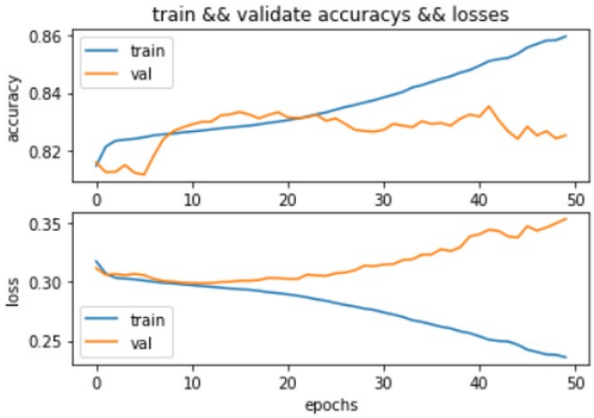
训练时的准确率和损失如下图所示，在测试集上准确率 0.87，AUC 值 0.8。

Model	Test acc	Test loss	AUC score
CNN_LSTM	0.8728	0.3234	0.8062

3.2 基于图像分类

为了更好的分析模型，我们将图像和文本单独作为输入训练两个分类器，观察两个 baseline。这是只基于图像的分类结果，我们将 512 维的图像特征直接连接全连接层进行分类，但发现验证集和测试集上准确率只有 83% 左右。



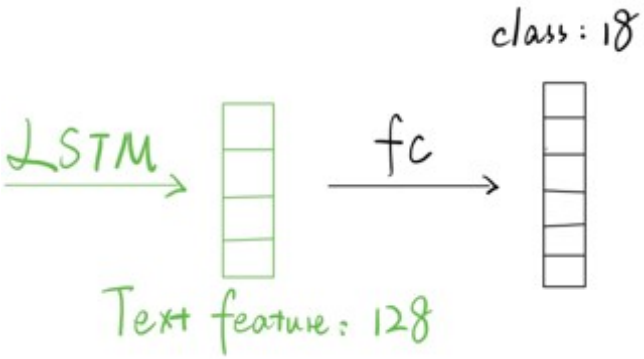


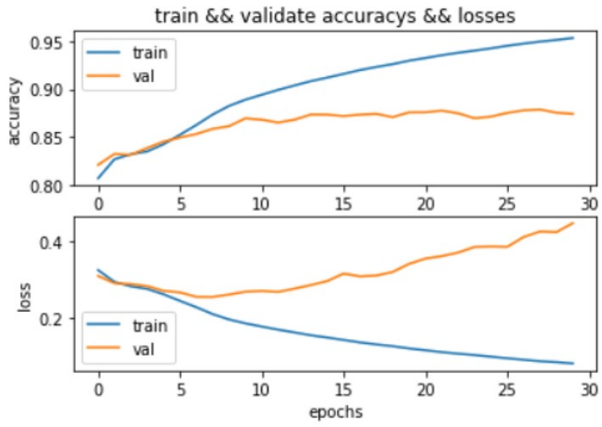
Model	Test acc	Test loss	AUC score
CNN	0.8328	0.3048	0.6341

我们猜测是 CNN 过于精简，模型表达能力不足，于是用在 ImageNet 上预训练好的 $Resnet18$ 模型替换掉我们简单的 CNN，发现验证集上结果持平再 83% 左右。我们猜测是因为单从海报中获取电影类型这个多标签分类任务有些难度，因为我们人类也比较难分类准确。

3.3 基于文本分类

我们也测了单用文本信息分类的结果：使用 LSTM 网络提取 128 维特征直接连接全连接层进行分类，发现与多模态的结果基本持平。可能多模态分类时从文本得到的信息占了大部分比重。





Model	Test acc	Test loss	AUC score
LSTM	0.8715	0.3199	0.8173

3.4 对于融合方法的探索

- 分量加法融合：$Val\_acc = 0.8791$

$$o(x_n) = W(Ux_n^t + Vx_n^v)$$

- 分量乘法融合：$Val\_acc = 0.8803$

$$o(x_n) = W(Ux_n^t * Vx_n^v)$$

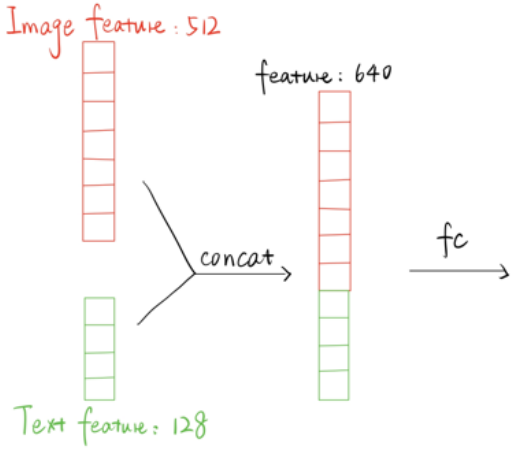
- 分量最大值融合：$Val\_acc = 0.8788$

$$o(x_n) = Wmax(Ux_n^t, Vx_n^v)$$

- 分量 $sigmoid$ 激活融合：$Val\_acc = 0.8295$

$$o(x_n) = W(\sigma(Ux_n^t) * Vx_n^v)$$

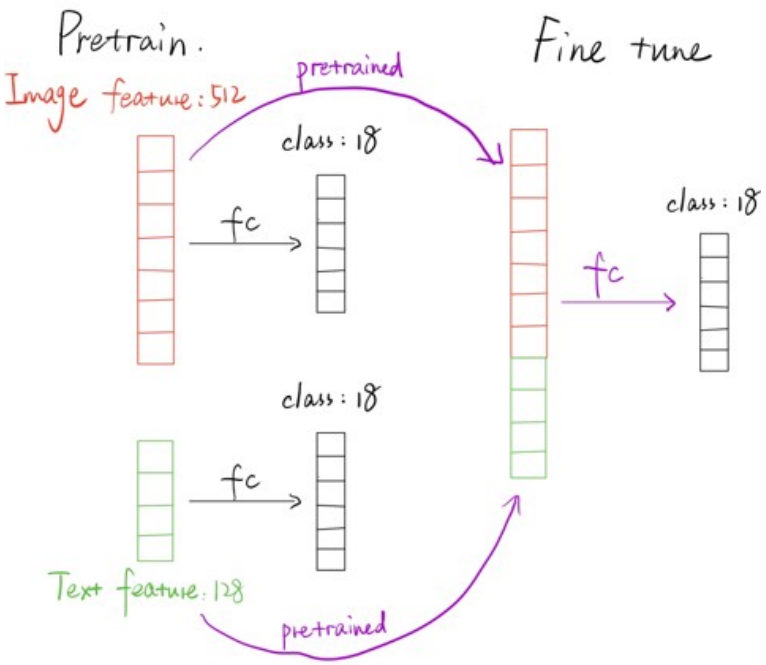
我们将两种模态特征大小修改为 256 之后进行融合，结果与原先拼接的模型持平，相加融合的方式略好于 $baseline$，但训练成本略高于拼接融合。

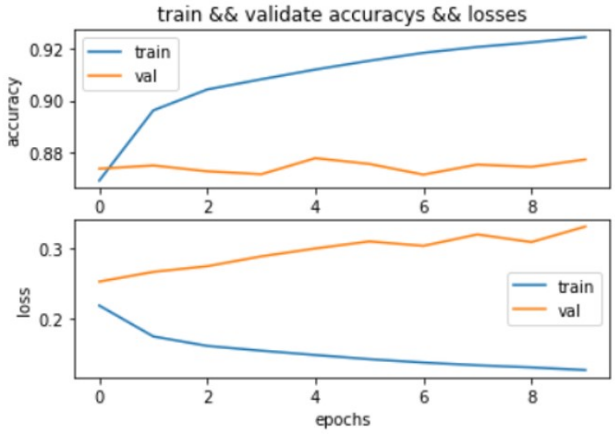


Model	Test acc	Test loss	AUC score
add_fusion	0.8762	0.3470	0.8122
dot_fusion	0.8765	0.4150	0.7921
max_fusion	0.8767	0.3366	0.8039
sigmoid_fusion	0.8292	0.3050	0.611

3.5 预训练-微调流程

预训练-微调（pretrain-fine tune）的流程在分类任务中十分常见，我们先单独训练图像分类器与文本分类器，取各自预训练出验证集上表现最好的模型，最后进行融合、分类。我们认为这样能保证图像和文本的特征提取分别是最优的状态，而且可以缩减训练时间，因为 CNN 的训练速度远快于 $LSTM$，而最后 $finetune$ 时很快就可收敛。我们发现相对于对网络所有参数 $fine\ tune$，固定特征提取器的参数，$fine\ tune$ 时训练更快，而且准确率几乎持平。





Model	Test acc	Test loss	AUC score
finetuned	0.8751	0.3122	0.8165
finetuned frozen	0.8755	0.311	0.822

4 Conclusion

在这次课程设计中我们学习了多模态深度学习的理论与思想,将课堂上学习的深度学习的知识和 $PyTorch$ 的使用应用于实践，并得到了较好的结果，受益匪浅。

实验过程中，我们发现：

- 使用最大池化的融合方法在测试集上准确率高,使用固定前端网络参数的 finetune 方法在测试集上 AUC 值最高，但与参考代码的 baseline 相比提升并不明显。
- 从训练时的 loss 曲线也可以看出 epoch 增加时过拟合现象严重，我们尝试过提高网络中 dropout 概率，但没有显著效果。
- 我们猜测若要进一步显著提升分类效果，应该更换更有效的文本分类器，但囿于时间和资源，我们未作尝试。