

情感分析作业报告

一、程序说明

1. 依赖

训练框架：Keras, Tensenflow

评估框架：Sklearn

语料集：V2 数据集

2. 使用说明

详见 readme 文件，注意鲁棒性较差，运行时请按照提示输入

二、模型结构图及分析

本次实验中，实现了 CNN、RNN 和 MLP baseline 三种网络。网络具体的结构如下图所示

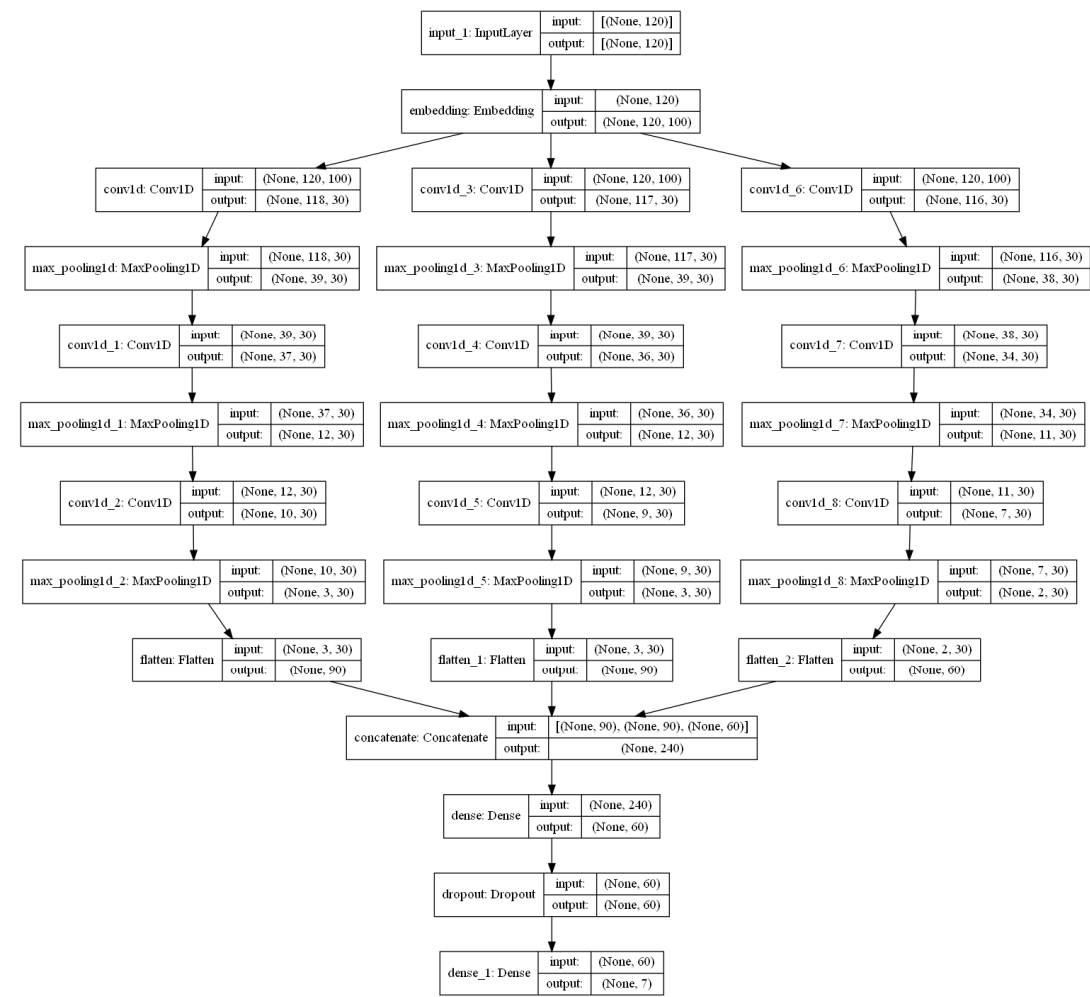


图 1 CNN 模型

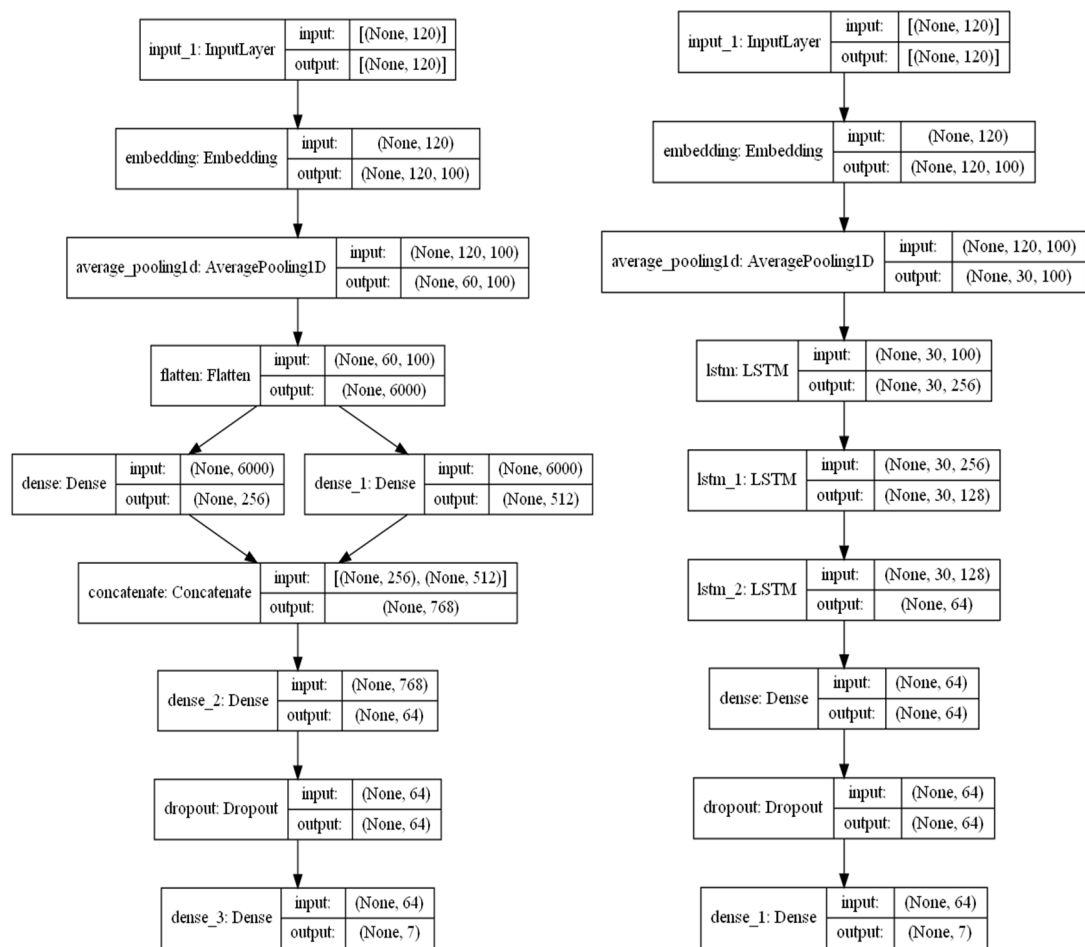


图 2 左：MLP 模型 右：RNN 模型

分析

词向量。对所提供的语料库进行预处理，去除英文标点并将所有字符转化为小写，之后借助 word2vec 模型，预先训练出词向量，保存为 model 文件夹下 wv.model 文件，词向量的维度设置为 100。

输入处理。由于所给定的语料为英文短句，共计 7666 条英文语句中所含单词超过 120 个的不足 10 条，故本次作业中将输入语句的最大次数设置为 120。之后对每句英文语料，将其中单词分别替换为对应的词向量，从而得到的 120x100 维矩阵作为网络的输入；若语料中所含单词超过 120，则超出部分自动截去，否则则自动补零至 120（注：输入网络的词向量设置为可随训练的进行调整）。

CNN 的设计。基本参照论文中的结构，整个网络一共分为 3 个通道，将 120x100 的词向量输入后，分别通过 3, 4, 5 x 30 的卷积及最大池化层，得到对应的特征值，之后将三个通道的特征值连接，经过全连接层后 softmax 输出。

RNN 的设计。采用了多层的 LSTM，而后再通过了两层全连接神经网络后 softmax 输出。但不知为何单此设置训练的结果并不好，而在输入层后添加了一平均池化层后，模型的准确率提升了近 15%（考虑添加池化层的最初目的是想抽取特征，使训练时参数少些）。

MLP 的设计。同 RNN 类似，主要用了四个全连接层，分为了两个通道，其效果也与 RNN 相当。

三、实验结果及分析

三种网络的实验指标以及部分参数如下表所示：

模型	CNN	MLP	RNN
Accuracy	0.515329	0.455969	0.472277
F1-score(macro)	0.519203	0.455821	0.444985
F1-score(micro)	0.515329	0.455969	0.472277
Optimizer	Adam	Adam	RMSprop
Learning-rate	0.0015	0.0012	0.001
Loss	CCE	CCE	CCE

其中向量维度统一为 100，最大词数为 120，batchsize 为 100，迭代次数 13 次

参数效果分析

整个实验首先要确定的参数是**向量维度**和**最大词数**，当然，理论上来讲是越大越好，但如前面已分析，给定语料为英文短句，最大词数本就有限；此外，参数设置过大反而可能会拖慢训练的速度，而且随着增长，其对准确率的影响越来越小。综合上述考虑，最终选择了 100 和 120。

训练周期数对模型的准确率影响不大，基本上 7~8 个周期后，网络就已经趋于稳定，之后会在相应值附近有一定涨落。

Optimizer 对模型的影响出乎意料的大。一开始时 MLP 的准确率只有 28%，但将 Optimizer 由 SGD 改为 Adam 后，准确率便上升到了 48%。CNN 同样如此，Adam 与 SGD 二者准确率差距也多达到 10%。而 RNN 则是选择使用 RMSprop，据称它是“特别适用于循环神经网络”的一种训练方式。总之觉得不同的网络结构，适合着不同的 optimizer，至于如何选择，更多的还需要根据经验与实际效果。

此次作业为多分类任务，因此**损失函数**可使用多分类交叉熵，实验开始之前尝试过使用均方误差 MSE 作为损失函数，预测的是情感的分布。但发现 MSE 很难收敛，并且效果很差。因此最后全部采用 CCE 来作为交叉熵损失函数

学习速率对模型的影响也很大，较大的学习速率会造成模型的过拟合，准确率会随之降低不少。

总结：比较三种模型的效果，CNN 在各方面均最佳，RNN 次之，但与 baseline 差不太多。而与 MLP 相比，RNN 的训练时间较长，所以有点得不偿失。MLP 作为一个入门模型，效果却不错，可以和 RNN 媲美，我感觉原因只可能是 RNN 网络的一些参数没有找对或者数据集的问题。此外，在作业中，RNN 网络的宏平均与微平均之间也稍有差距，令我也有点不解。

总之，综合各种结果，感觉 CNN 更适合情感分类的工作，它在卷积核与输出之间存在着比较明显的对应关系，实验效果也最好，而 RNN 则很难说清 LSTM 层的意义，甚至综合考量只能与 MLP 相当。

四、问题思考

1) 如果控制实验训练的停止时间？简要陈述你的实现方式，并试分析固定迭代次数与 early stopping 等方法的优缺点。

控制实验训练的停止一般来说有两种方式

一种为 **early stopping**，一般来讲，选定某一指标（如在验证集上的准确率、准确率不再上升的周期数等等），设置一个阈值 θ （和累计次数 λ ），当所

选定指标达到阈值（或累计达到 λ 次）时，则停止。

另一种则是**固定迭代次数**，估计过拟合前的位置，设置固定的迭代次数，当然实际中可以根据模型效果做相应的调整。

我采取的方式为第二种，先在一个较大的周期内训练，然后观察每个周期内验证集损失、准确率的变化趋势，再选择较小的周期，进行多次的验证。实际上，由于输入数据的顺序随机，变量初始化也是随机的，所以每次训练的效果并不完全一样。因此，最终我选定一个平均较为稳定的周期数。

固定迭代次数方式的**优点**是，操作简单，而且在模型不复杂的情况下，可以快速地完成多次训练，从而非常直观的比较不同模型的数据（曲线）来判断模型的优劣；但**缺点**是确定合适的迭代次数花费时间较长，需要多次训练验证。

至于 early stopping，一般来讲其选定的指标通常为验证集的准确率等，这种方式的**优点**是，在模型较复杂、时间不足或想快速选择模型时，能有效提高时间效率；**缺点**是实现起来不如前者简单，同时也不太能直观地比较出不同模型的特点。

2) 过拟合和欠拟合是深度学习常见的问题，有什么方法可以解决上述问题。

欠拟合主要是指模型的复杂度低，在训练集上表现很差，未能较好的捕捉到数据的特征，学习到数据背后的规律以拟合数据。

解决方法主要有两方面，一是网络结构太过简单，可通过增加网络复杂度、深度来解决，二是模型中特征项不够，需要在模型中增加特征项，换用更复杂的模型，如将线性模型替换为多项式模型等来解决；当然，还有可能是参数设置错误，比如训练次数过少等而导致的，这种则需要视情况做出相关调整。

过拟合可简单描述为模型复杂度高于实际问题，模型在训练集上表现很好，但在测试集上却较差，泛化能力差。

相应地，**解决方法**有

1. 获取和使用更多的数据，也即增大数据集。这个是最直接的方法，为模型提供更多的数据，使得数据能尽可能地反应整体的全貌，让模型在数据上充分地学习并不断地修正。

2. 对数据集进一步的清洗与处理，过拟合有可能是由于数据不纯，特征值过多导致的，此时可以对数据作进一步的处理，删除掉一些冗余的特征等等。

3. 采用合适的模型，降低模型的复杂度

4. 使用 dropout 函数，Dropout 指的是在训练过程中每次按一定的概率随机地隐藏一部分神经元（激活函数输出设为 0），不让其参与计算。它在一定程度上可以消除神经元节点间的联合，降低了网络对单个神经元的依赖，增强模型的泛化能力。

5. 选择训练合适的停止时间，训练次数过多，难免会产生过拟合现象，可使用控制迭代次数与 Early Stopping 的方法适时地终止训练，避免过拟合。

如何处理过拟合是深度学习中经常遇到的问题，主要的目标就是提升模型的泛化能力，通过查阅资料，解决过拟合的方法一般还有控制学习速率，采用 L1/L2 正则化，以及使用 Batch_Normalization 等，在遇到具体问题时需要具体分析，选择合适的方法以达到最优的效果。

3) 试分析梯度消失和梯度爆炸产生的原因，以及对应的解决方式。

原因：梯度消失和爆炸的根本原因在于 BP 这一训练方法，本次实验中，神经网络都是基于 BP 算法，根据由损失函数计算的误差通过梯度反向传播的方式，来指导网络中权值的更新变化，由于将误差向前传递的过程需要使用链式法则，

而链式法则为连乘的形式，当网络层数越深时，根据链式法则计算出的梯度值会接近 0 或者特别大，也即出现梯度消失或爆炸。

解决方法常用的有：

1. 选择 Relu、LeakyReLU、ELU 等激活函数。以 Relu 函数为例，其导数在正数部分是恒等于 1 的；而其他的激活函数如 sigmoid，其导函数最大值为 1/4，当网络层数越多，反向传输值难免越小，最终导致出现梯度消失的情况。因此在深层网络中常使用 relu 激活函数以缓解梯度消失与爆炸问题。

2. 梯度裁剪。主要针对梯度爆炸问题，设置一个阈值，然后更新梯度的时候，如果梯度超过这个阈值，那么就将其强制限制在这个范围之内，以防止梯度爆炸。

3. 使用 BN (Batch Normalization) 算法

此外，梯度消失与爆炸的解决方法还有对权重进行正则化、使用残差网络等，在此不再一一叙述。

4) 试分析 CNN，RNN，全连接神经网络(MLP) 三者的优缺点与各自适用的场景。

	MLP	CNN	RNN
优点	结构简单，易理解，可解释，学习全局信息	结构简单，权重共享减少了需要训练的参数，加快了训练速度，可自动提取特征，在局部特征明显时优势很大	结构简单，可解释，有时序性，可以对序列内容建模，处理序列数据
缺点	参数规模较小时学习能力弱，参数规模大时对机器配置（内存等）要求较高	网络结构比较浅时学习能力有限，不能获取全局信息	需要训练参数较多，容易出现梯度消失和梯度爆炸问题，不具有特征学习能力，且训练较慢
适用场景	由于会提取到全局特征，因此常常只放在各种网络最后，用于综合所有的信息	图像处理中的分类、目标检测等，特别是可直接将图像进行输入，避免了复杂的特征提取等	语音处理，人机对话，文本处理等模型输入为时间序列数据

五、心得体会

此次作业是我第一次接触人工神经网络，通过使用 keras 框架搭建了三个简单的网络模型，当运行起来预测程序并输入语句看到网络输出的结果时，心情还是蛮激动的，不过一些语句情感的预测并不太准，模型的准确率在测试集上也并不算不高，自己的网络结构及参数还需再改善。通过这次作业，我也更深刻地理解了人工智能的黑盒特点：针对具体的问题来讲，选择何种网络结构、参数更多的是依据经验、实际效果等，缺少强有力的理论支撑，不过这也是由于网络中所含神经元过多，难以完全描述的缘故。