**研究生学习周报**

作者：蔡明炜

**目录**

第一周 9月16日~9月22日 2

第二周 9月23日~9月29日 7

第三周 10月7日~10月13日 11

第四周 9月14日~9月20日 15

### 第一周 9月16日~9月22日

**李宏毅老师的深度学习视频部分：**

***part1：线性回归***

线性回归的主要思路据视频中描述如下：

第一步，找一个函数集合（模型）；第二步，定义一个损失函数L（f）来判断函数（模型）的好坏（寻找最小值）；第三步，运用梯度下降优化模型。

首先建立一个模型，在初始时可以设为y=bi+wi\*xi (i为下标),损失函数定义为：

L(f)=∑ (yⁿ- (b+w\*xiⁿ) ) ²

(n为上标，i为下标，其中yⁿ与xiⁿ为给定训练数据集中的n个样本）

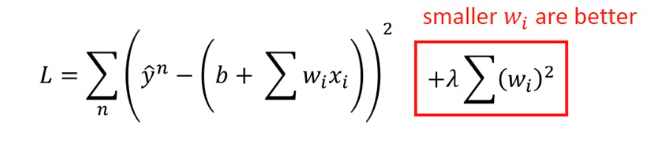
任取L（f）中的一点（w0，b0），在该点分别取偏导，根据偏导的值来移动w与b的值。若w0、b0的偏导为正，说明函数正字上升，为了取最小值，则应该减少w与b；否则就要增加。

ŋ是学习率（learning rate），需要我们自己调整。有关更多梯度下降的内容在下一部分会有阐释。

若是想要模型更加符合实际，我们可以提升模型的容量，加入跟高的X次方，x的次数越多，模型的表示越广。但是这种情况可能会出现过拟合的问题，这一种情况会在第三部分阐释。所以若是单单提升容量无法找到合适的模型，可以考虑往模型中添加更多的参数来设立模型。

还有一种方法是正则化，重新设计损失参数。



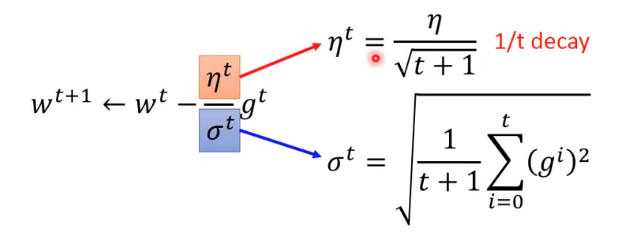
当wi越小，输入的变化对结果输出的影响越小，损失图像越平滑。所以要找到一个大小适合的，使图像尽量平滑。

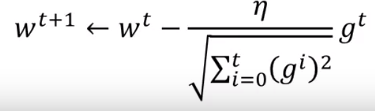
***part2：梯度下降***

Tip1:小心调整learning rate

在进行梯度下降时，如果learning rate较大，可能会发生直接跳过局部最小值，导致错误判断最小值的位置产生误差，所以倾向于随着参数更新减小learning rate。

一种办法是Adagrad算法：

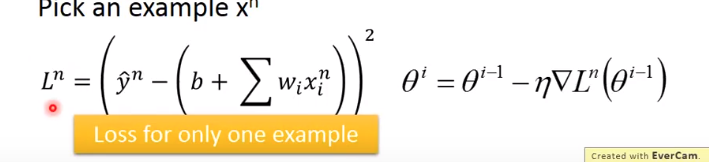




其中σt是所有微分值的均方根。

Tip2：随机梯度下降（Stochastic grradient descent，SGD）

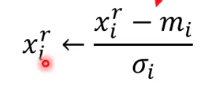
能让训练速度更快。每次只取1个样本，在计算损失时只计算该样本的损失，并且更新参数时只根据该损失来更新参数。



Tip3：特征缩放(Feature Scaling)

让不同的特征的分布相同，更新参数会更加容易。

常见的做法：对所有的样本的第i个特征计算平均值mi与标准差σi，更新所有样本的第i个特征，公式为：



其中r是样本的序号，i是该样本的第i个特征。更新完后，所有样本的第i个特征的平均值是0，方差是1。

Tip4：梯度下降的缺点

在极小值或者微分值是0的地方就会停止，不一定会到达全局最小值。

***part3：误差的主要来源***

我们在做机器学习时，误差主要来源于偏差（bias）与方差（varience）。偏差描述的是算法的预测的平均值和真实值的关系，而方差描述的是同一个算法在不同数据集上的预测值和所有数据集上的平均预测值之间的关系。方差和偏差是有冲突的，称之为变差方差窘境（bias-variance dilemma）。一般来说，当模型容量小的时候，bias较大而varience较小；当容量较大时，bias较小而varience较大。所以我们在寻找目标模型时，要寻找一个平衡点，使偏差方差的总体的泛化误差最小（最理想状态是低bias与低varience，但是一般不存在）。

一般来说，当你建立的模型与训练样本不能拟合时，bias较大；当模型能与训练样本拟合但是不与测试样本拟合时，varience较大。也就是说当欠拟合（underfitting）时，bias较大；过拟合（overfitting）时，varience较大。在判断完当前误差的主要因素后，就可以使用一系列操作来降低误差。

当bias较大时，我们可以重新设计模型，加入更多的特征并增大容量。当varience过大时，我们可以通过增加采样数量和正则化来减小varience。

需要注意的是，我们往往使用公共的测试样本来测试模型的误差，但是我们在根据公共测试样本来调整我们的模型时，其实是在对公共的测试样本来进行拟合，这有可能导致当我们的模型应用于其他的测试样本时出现误差增大的现象。为了防止这种情况出现，我们可以使用交叉验证(Cross-validation)的方法，将训练集分成两部分，一部分用于训练模型，另一部分用于验证误差。这样得出来的模型才有可能通过公共测试样本得出对于这类问题所有可能样本的误差。

一种交叉验证方法时K-foldcross-validation：我们将训练集分为K份，每次只取1份用于验证误差，另外k-1份作为训练样本。对每一个模型都重复k次，直到所有子样本都成为了一次验证样本，得出平均k次的结果。最后选择平均误差最小的那一个模型。

**论文学习部分：**

***《RGCN:用于目标相关情绪分析的循环图卷积网络》***

本文首先阐释了情感分析（意见挖掘）的重要价值，并举了一个例子来解释了一个用户在同一段评论中可能既有正面的情绪，也有负面的情绪，即使整体情绪可能是正面或是负面中的一个。

依赖于目标的情绪分析，也称为基于情景的情绪分析在情绪分析领域是一项细粒度的任务，传统的方法主要是使用基于特征的支持向量机（SVM），然而这些方法需要手动设计特征。而神经网络具有学习连续文本特征的能力，能够通过多个隐藏层生成新的表示，近年来，越来越多的深度学习方法在这项任务中被探索出来，这些方法比基于特征的方法具有更多的可拓展的方式。

基于递归的神经网络的框架使用了发散合成函数，使他们能够处理情感分析中的复杂合成。尽管这些方法利用了句法结构，但他们关注的是与目标词相关联的结构，而忽略了整个文本的结构，而且仅考虑与目标词相关的词，很难处理长句，虽然目标词在任务中起着重要作用，但对整个句子的表征也是必不可少的。

基于递归的神经网络（RNN）、记忆网络和卷积神经网络（CNN）的流行方法不不使用句法信息，而是通过句子序列或注意机制隐式的保持文本结构。但是必须考虑到整个文本结构以及与目标词相关的特定区域结构的重要性。作者举了一个例子：“我吃了一顿丰盛的饭，我们的服务员引导我们去买一种价格范围内非常好的葡萄酒，而不是让我们去买一种价格相近但质量不好的葡萄酒。”目标词“葡萄酒”出现了两次，在句子的不同部分有不同的情感极性。总的来说，用户对“葡萄酒”持有积极的极性。然而，如果我们在句子中交换两个方面，情感的极性就会改变。结果表明，目标词不仅与句子中的词有关，而且与整个文本结构有关。

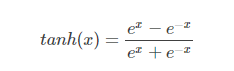
为了解决这个问题，对文本的图形结构执行操作是至关重要的。图卷积网络可以在图结构上运行，其中节点状态通过合并相邻节点的信息来表示。因此随着层数的增加，同一子图中的节点的特征难以区分

（到目前为止论文已经提到了三种神经网络，所以我决定先了解这几种神经网络的大致结构再回来继续学习该篇论文）

***《从多层感知器到卷积网络（一）》***

在多层感知器（MLP）中，曾经科学家使用阶跃函数作为神经元来模拟神经元传递信号，但是直接输出-1或1的方式对于以后的优化工作及其困难，所以有人想到了使用正切函数tanh。

正切函数：

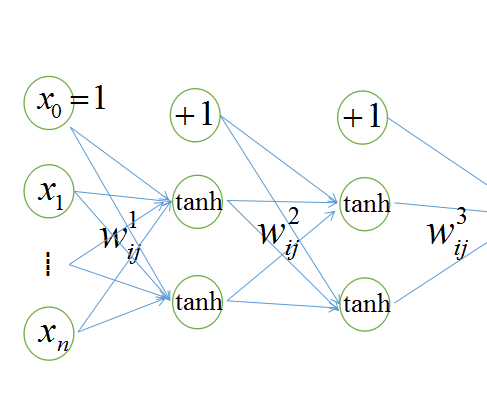


其导数形式为：



tanh函数具有明显的S曲线，在x远大于0时输出1，x远小于0时输出-1，而在其他时候输出一个-1到1之间的数，这样一个平滑的输出范围更利于后期的优化工作。

MLP共3层，分为输入层，隐藏层与输出层。对于输入向量x，在输入第1层前，进行转换变成一个加权分数：。W1指的是第0层与第1层的权重矩阵，b0就是bias项。添加bias项的原因是为了让我们的神经网络表达能力更强。算出加权分数后，就用tanh函数处理：。处理的结果就作为下一层的输入，经过层层处理，最后一层输出我们想要的结果。



为了控制网络的意图，使用一个cost函数（用以衡量神经网络的输出好坏）。运用梯度下降等方法来寻找最优的权重，这个部分在上一部分有所总结，不多加赘述。

**下周学习计划：**

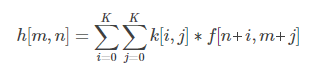
继续跟随深度学习视频课程学习深度学习理论方法，并大致了解完卷积神经网络以及图神经网络的大致结构，推进《RGCN》论文的理解程度。

### 第二周 9月23日~9月29日

***《RGCN:用于目标相关情绪分析的循环图卷积网络》***

为了更好的理解论文，我简要的了解了一下卷积神经网络的概念，具体如下：

卷积是两个变量在某范围内相乘后求和的结果。在wiki中卷积定义为：



f是原始图片，k是核函数，h是卷积后的特征。我们的目的是利用神经网络，学习得出类似k这样的核函数来获取有利于图像识别的特征。

卷积网络是一种特殊的前馈网络，只是将前馈网络中的某些层换成了卷积层。而卷积网络中的卷积层就是一个多核卷积的过程的抽象。

接续上周，本文提出的框架RGCN像递归神经网络一样引入了一个隐状态，节点状态是隐状态和相邻节点信息的表示。使用依存句法分析的结果构造一个文本图，而且目标被视为整个图中的一个特殊区域。用RGCN对文本和目标进行编码并通过这些表示得到情绪分类。

递归神经网络简要概念如下：

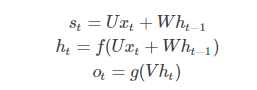
如果一系列输入有明显的上下文特征，也就是说这样的输出必须依赖以前的输入，那么网络需要有一定的“记忆能力”，而递归神经网络（Recurrent Neural Network）便能够赋予网络这样的记忆力。



左侧是递归神经网络的原始结构。与普通三层结构稍有不同的是，隐藏层添加了一个闭环。上图右侧是展开之后的结构。

xt是t时刻的输入，是输入序列的一部分，对于语言模型，每一个xt将代表一个词向量，一整个序列就代表一句话。ht是时刻t的隐藏状态，ot代表时刻t的输出。输入层到隐藏层直接的权重由U表示，它将原始输入进行抽象作为隐藏层的输入。隐藏层到隐藏层的权重W是网络记忆的控制者，负责调度记忆。隐藏层到输出层的权重V，从隐藏层学习到的表示将通过它再一次抽象，并作为最终输出。

在t=0时刻，UVW都被随机初始化好，h0通常初始化为0。然后随着时间推进进行如下计算：



每一次上一轮的隐藏状态作为ht-1将作为时刻t-1的记忆状态参与下一次预测活动，f是激活函数，g通常为softmax函数。所以可以这样理解隐藏状态：

h=f（现在的输入+过去记忆总结）

论文中对图卷积神经网络（GCN）进行了介绍。

在GCN中，图形结构由相邻矩阵表示，对图形结构进行卷积运算。其描述为：



在中，A是图的邻接矩阵，定义为=A+I,其中I是单位矩阵，是其中元素可以被表示为的对角矩阵。表示训练期间要学习的加权矩阵，f（·）是一个激活函数。

将作为隐层向量，单层GCN的隐表示只能捕获直接邻域的信息。因此，图中节点的表示是根据其相邻节点的特征向量递归计算出来的。此外，对于k层gcn，节点的表示可以捕获其k-hop网络邻居中的结构信息。

将图结构与卷积中的节点特征相结合的GCN模型使得同一集群中节点的输出特征更加相似。虽然GCN模型在特征提取方面很强大，但并不完全适合目标依赖情绪分析任务。

作者认为，当前场景的情感极性不仅与目标词有关，还与上下文（全文）有关。他们提出的模型充分考虑了给定文本中词与词的关系，基于以文本为图的图形，而目标词被认为是图的特定区域。框架如下图所示：

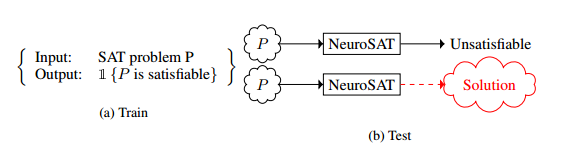


框架分为2块，一部分是文本图形的表示，另一部分是目标词的表示。通过RGCN获取节点的特征，然后通过所有的平均池层获得文本的表示，通过情景词的最大池层获得目标词的表示。两个隐藏状态被发送到完全连接层（fc）后，将完全连接层的输出传递到SoftMax层以获取类似标签。

***《从单位监督学习SAT解算器》***

作者提出的NeuroSAT是一种信息传递神经网络，它只经过训练成为一个分类器来预测可满足性，然后学习解决SAT问题。NeuroSAT通过简单的多次迭代来解决更复杂困难的问题。

命题可满足性问题（SATISFIABLITY problem，简称SAT）是计算机科学中最基本的问题之一。为了让神经网络学会解决SAT问题，作者开发了一个新型消息传递神经网络（novel message passing neroMPNN）NeuroSAT然后训练它作为分类器以预测随机SAT问题集的可满足性。对于每个SAT问题，我们只向NeuroSAT提供单位监督，以表明该问题是否可以得到满足。当对一个新的SAT问题进行预测时，NeuroSAT在找到一个解之前，都是用低置信度进行不满意的猜测，然后收敛到非常高的置信度进行猜测。解决方案本身几乎总是可以从网络的激活中自动解码，从而使NeuroSAT成为一个端到端的SAT求解器。有关训练和测试状态的说明参考下图：



也可以用同样的神经网络结构来帮助构造不满意问题的证明。当作者在一个每个不满意的问题都包含一个小的矛盾的不同的数据集上训练它时（NeuroUNSAT），它不会去寻找令人满意的任务，而是学习检测这些矛盾。正如我们可以从NeuroSAT的激活中提取解一样，我们也可以从神经SAT的激活中提取矛盾的变量。当矛盾中涉及的变量数量相对于变量总数较小时，了解矛盾中涉及的变量可以更有效地构造一个解的证明。

问题背景：命题逻辑的公式是一个使用常量true（1）和false（0）、变量、否定、连词和析取构建的布尔表达式。若将布尔值赋给公示的变量使公式的计算结果为1，则公式是可满足的。例如，公式(*x*1 ∨*x*2 ∨*x*3) ∧┐(*x*1 ∧ *x*2 ∧ *x*3) 是可满足的，因为它将在每个未将x1、x2和x3映射到相同值的赋值下计算为1。对于每一个公式，都存在一个合取范式（cnf）的等价公式，表示为变量（可能为负）析取的合取。在cnf中，公式的每一个连词都称为子句，而子句中的每一个（可能是负数）变量都称为literal（字？）。上面的公式相当于cnf公式（x1∨x2∨x3）∧（┐x1∨┐x2∨┐x3），我们可以更简洁地表示为{1|2|3,‾1|‾2|‾3}。cnf中的公式有一个可满足的赋值条件是且仅当它有一个赋值时，每个子句至少有一个literal映射到1。SAT问题是CNF中的一个公式，其目标是确定公式是否可满足，如果可以，则生成一个令人满意的变量真值分配。我们用n表示SAT问题中变量的数目，用m表示从句的数目。

分类任务。对于SAT问题p，当且仅当P为可满足时我们定义φ（P）为真。第一个目标是学习一个近似于φ的分类器。给定SAT问题的分布ψ，我们可以用形式（P，φ（P））的例子来构造数据集D*train*和D*test*，方法是用现有的SAT解算器对问题p～ψ进行抽样并计算φ（p）。在测试时，我们只得到问题p，目标是预测φ（p），即确定p是否满足要求。最后，我们关心的是解算的方法，其中也包括找到解决问题的方法。

**下周学习计划：**

### 第三周 10月7日~10月13日

***《RGCN:用于目标相关情绪分析的循环图卷积网络》***



框架分为2块，一部分是文本图形的表示，另一部分是目标词的表示。通过RGCN获取节点的特征，然后通过所有的平均池层获得文本的表示，通过情景词的最大池层获得目标词的表示。两个隐藏状态被发送到完全连接层（fc）后，将完全连接层的输出传递到SoftMax层以获取类似标签。

首先，单词序列进入嵌入层。一般来说，嵌入层的作用是将文本映射到一个向量空间中。这样做的目的时方便机器进行计算，将单词转变为词向量，机器便可以通过计算不同词向量之间夹角余弦值而得出单词之间的相似性。再Embedding Layer中，一般方法是对单词进行one hot编码，向量空间的大小或维度被指定为模型的一部分，向量以小的随机数进行初始化。被编码过的词映射成词向量，如果使用循环神经网络则每个单词都可以作为序列中的一个输入。

**图形构造**：设W = {w 1 ，w 2 ，... w N }（| W | = N）表示在语料库中的单词集合。 W s是给定文本的单词集合，W s = {w s1 ，w s2 ，... w sM}（| W s | = M，W s ∈W）和W t ={w t1 ，w t2 ，... w tK}（W ti∈W s ），如图一所示，给定这样一个句子”the food is so good and popular that waiting can really be a nightmare”,W t={food}且Ws = {the; food; is; so; good; and; popular; that;waiting; can; really; be; a; nightmare*}。*

更正式地说，无向文本图定义为G=（V：E），其中V和E分别是节点集和边集。文本图的构造如下：首先，每个顶点对应于Ws中的一个词，然后建立了从属于一组特定对话的两个词的句法依赖关系的连接。。最后，将自循环添加到文本图中。从无向文本图中得到对称邻接矩阵A（Aij ∈ {0，1}），L代表A的归一化拉普拉斯矩阵。（拉普拉斯矩阵定义为：D-A。D是该图G度的矩阵。归一化算式在上一周已经给出）

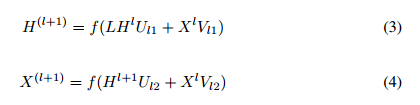
**RGCN层：**

在图形构造后给定的文本将转换为图形。图中的节点由嵌入层处理后的向量xi表示。一般来说语料库中的集合W的长度N大于文本集合Ws的长度M。由于不需要为每个给定的文本建立邻接矩阵A∈，因此将我们将节点序列转换成当前文本的单词序列，然后邻接矩阵A变成

嵌入层的输出表示为Xembed。引入了节点的隐藏状态，将节点的初始隐藏表示定义为Xembed∈，F为特征尺寸。图中节点的初始状态定义为：



其中U 0是权重矩阵，并且∈ 。 对于RGCN层，传播规则定义为：



其中Ul1，Ul2，Vl1和Vl2是可训练参数。由方程可以看出，在RGCN中节点的隐藏状态由节点的上一个隐藏状态和相邻节点的状态更新。它不仅像GCN一样包含了图形结构的信息，还像RNN一样使用了上一次的隐藏状态来更新当前状态。

聚合函数设计为池化操作，这是对特征向量的元素操作。在池化后，用来表示文本图的节点的隐藏特征向量表示为。在这个项目中，作者对给定的文本应用平均池化层，对目标词应用最大池化层。

池化层：使用某一位置的相邻输出的总体统计特征来代替网络在该位置的输出。对输入的特征图进行压缩，一方面使特征图变小，简化网络计算复杂度，一方面进行特征压缩，提取主要特征。平均池化层对特征图的某一个区域内的所有特征值取一个平均值作为该区域的特征，而最大池化层寻找区域内的最大值作为该区域的特征。使用池化可以看作增加了一个无限强的先验。

该模型经过训练，以监督的方式最小化交叉熵误差，损失函数定义为：



其中，pti是模型给出的第i个训练示例的概率。我们使用反向传播来计算所有参数的梯度，使用学习率为0.001的Adam优化算法来训练模型。dropout（丢弃法）和早期停止技术是用来缓解过度拟合。使用GloVe预训练的300维文字嵌入，训练期间不调整。此外，批大小设置为128，最大历元为50。

Adam优化算法：Adam是一种可以替代传统随机梯度下降过程的一阶优化算法，它能基于训练数据迭代的更新神经网络权重。

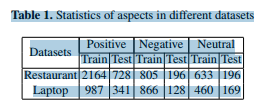
dropout：dropout是指在深度学习网络的训练过程中，对于神经网络单元，按照一定的概率将其暂时从网络中丢弃。

Early stop：早期停止法是指在验证误差达到最小时停止训练。

G1oVe：GloVe是通过整个语料库的所有句子取分析词与词之间的关系的模型。

**实验和结果**

**实验设置：**作者在Semeval2014的公共数据集上测试模型，包括用户生成的有关笔记本电脑和餐厅域的评论。他们删除了一些带有冲突标签的样本。数据集的统计数据如下



作者只对数据集进行了基本的文本预处理，包括字母转换为小写、删除空白、使用python nltk工具进行标记化以及使用斯坦福解析器解析语法依赖项。

在这个的模型中，操作是在图结构上执行的，我们要求基于语法依赖关系的边连接到属于特定词性集的词上。词性集的一部分格式以[“jj”、“nn”、“r”、“cc”、“in”]开头，对于给定的文本，如果句法相关性中成对词的词性部分都是词性集的一部分，则它们之间建立了链接。在实验中，我们开发了具有RELU激活功能的五层RGCN。模型的隐藏尺寸设置为200。

RELU函数：在CNN卷积神经网络中，习惯用RELU函数代替sigmoid, tahh等目标激活函数，这应该是因为，RELU函数在大于0的时候，导数为恒定值，而sigmoid和tahh函数则不是恒定值，相反，sigmoid和tahh的导数，类似于高斯函数的曲线形状，在两端接近目标时，导数变小。

导数小的话，在训练神经网络的时候，会BP反向传播误差，导致收敛减慢；而relu函数则避免了这点。

作者将模型与以下基线方法进行比较：

AdaRNN用于基于递归神经网络的目标相关依赖树上的特征学习，根据输入自适应地选择合成函数。

TD-LSTM是基于lstm网络的模型，其中两个lstm模型用于模拟围绕情绪分类目标字符串的前后上下文。

ATAE-LSTM在lstm网络中使用注意机制，其中不同的方面参与句子的不同部分。

MemNet是一个外部记忆的神经注意模型，由多个计算层组成。

RAM是一个在递归神经网络上采用多注意机制的框架。

IAN是一个交互式注意力网络模型。它使用两个注意力网络以交互方式对目标和上下文进行建模。

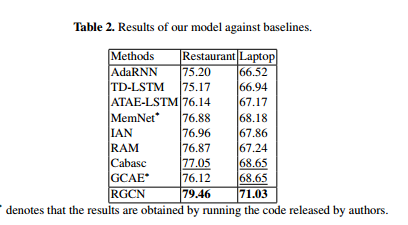
Cabasc基于记忆模型，通过句子级内容注意机制和上下文注意机制两种注意机制，解决语义失配问题。

GCAE基于卷积神经网络和门控机制。

我们注意到，在不同的文献中，对同一数据集上执行的同一模型报告了不同的结果。我们认为基线方法的结果受文本预处理和单词嵌入的影响，如[3]所述。

为了揭示模型的能力，所有模型都使用了相同的词向量，在pytorch中重新实现了作者没有提供实现的模型。按照原始文件中报告的超参数设置，将未报告的参数设置为与此相同。

所有方法均以精度评价为度量，结果如表2所示。最好的分数用粗体突出显示，下划线表示第二好的表现。



结果表明，RGCN在这些数据集上始终优于所有比较方法。这可能是由于句法依赖关系图在两个长距离词之间建立了联系，缩短了情景（方面）和相关词之间的距离。

在基于LSTM的模型中，TD-LSTM、ATAE-LSTM和IAN比其他模型有更好的效果，因为IAN使用上下文和目标注意机制，更好地将句子的重要部分用于情景（方面）词。

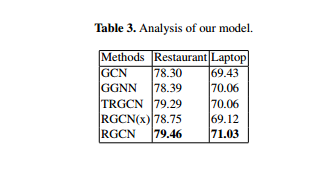
MemNet基于存储网络，包含多个关注层，优于基于LSTM的笔记本电脑模型。与RAM和MemNet相比，Cabasc通过句子级内容注意机制和上下文注意机制增强了从全局角度捕捉某一方面重要信息的能力，从而在笔记本电脑和餐厅的所有基线中都具有最佳的性能。GCAE利用卷积神经网络和门控机制，在笔记本电脑上获得最佳的CABASC效果。

尽管AdaRNN利用了了依赖性分析的结果，但它的性能比其他人差，特别是在笔记本电脑上。与AdaRNN相比，我们的方法将整个文本信息保存在RGCN中，目标集中在特定的方面区域，以避免其他图形部分的噪声。

### 第四周 10月14日~10月20日

***《RGCN:用于目标相关情绪分析的循环图卷积网络》***

为了说明该模型的有效性，作者设计了更多的相关的模型来比较结果。其中包括将RGCN层换成GCN层和门控图神经网络层（GGNN）的模型，模型分别表示为GCN和GGNN。与RNN不同的是，RGCN根据最后节点状态和隐藏状态来更新节点状态。TRGCN指的是节点的状态实现由隐藏状态更新的RGCN图。RGCN（x）是一个将节点状态传递到池层而不是隐藏状态的模型。这些模型的最佳性能如下表所示：



结果表明随着层数的增加，GCN比其他模型的性能更差，这是由于GCN很难区分不同的节点特征。RGCN 的精度高于所有模型，表明该模型在图形结构中具有更好的节点表示能力。实验结果表明，将隐状态引入GCN中可以提高系统性能。

**层的影响：**

如下图2所示，单层的隐藏表示只能捕获直接领域的信息，而具有K层的模型可以包含最多为k帧信息的节点。图2显示，当层设置为2时，GCN优于RGCN。随着层数增加，GCN的精度低于RGCN，且GCN线比RGCN线更平，这与之前的分析结果一致。同时还可以从图中观察到，精度不会随着层数增加而提高。

**嵌入调优的效果：**

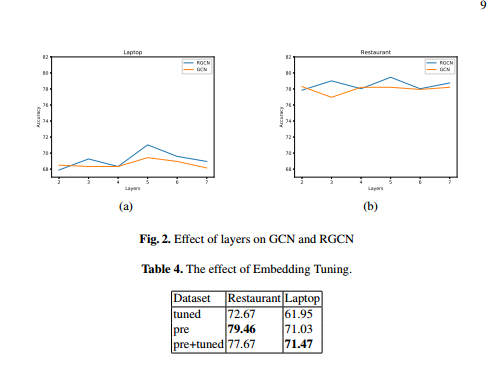
作者推测该神经网络可以捕获没有在预先训练的单词嵌入中表示的机构信息。为此，他比较了以下嵌入调优策略：

—预处理：使用预先培训过的单词嵌入，并不会在培训期间进行调优。

—调优：单词向量由预先训练的单词嵌入初始化，这些参数在监督训练中更新。

—预处理+调优：单词嵌入由两部分组成，一部分是预先训练的单词向量，另一部分是随机初始化的单词嵌入。前者不进行调整，后者在培训中更新。该实验使用50维随机初始化字嵌入并在RGCN模型上进行。

下表4中的结果表明,预处理+调优能在“笔记本电脑”上达到最佳性能，而“预处理”在“餐厅”上得到最佳效果。作者不确定结果的原因。



**结论与个人感受：**

本文提出了一种新的目标依赖情绪分析任务框架。通过与各种模型进行数据对比，可以发现该模型优于大多数模型。同时分析了层数以及嵌入调优的影响，为整合句法信息提供了新的方法。

我认为，将循环神经网络的隐状态引入图形神经网络从而让神经网络拥有记忆能力这一点是有效的，这种方法能够考虑到目标的情感与整个文本结构的关系，相比于传统的模型应该能更容易获取目标节点的特征。

***《从单位监督学习SAT解算器》***

**模型**

SAT问题具有简单的句法结构，因此可以使用标准方法（如RNN）将其编码到向量空间中。然而，命题逻辑的语义学却产生了这种句法方法所忽略的丰富的不变性，如排列不变性和否定不变性。具体地说，一个公式的可满足性不受 重新排序变量的影响（如在整个公式中交换x1和x2），不受重新排列子句的影响(例如，用第二个子句替换第一个子句)，不受重新排列子句中字的影响(例如，将替换为。)。同时公式的可满足性也不受一个给定变量对应的每一个字的求反影响（如在中对所有的x1求反得到）。

NeuroSAT强调排列不变性和否定不变性。作者将SAT问题编码为一个无向图，每一个字、子句都有一个节点，每个文字和子句之间有一条边，而每一对互补文字之间有不同类型的边（如xi与xi求反）。NeuroSAT通过沿着图的边缘来回传递“信息”，迭代地细化每个节点的向量空间嵌入。在每个时间步骤，作者都对每个文字和每个子句进行了嵌入。迭代包括两个阶段。首先，每个子句从其邻近的字接受消息并相应的更新其嵌入。然后每个字从其相邻的子句及其补语（即本身取反）接受消息，然后相应的更新其嵌入。下图2提供了架构的高级说明。



这个图代表了的神经系统高级图解。两个图的顶部分别是4个文字的节点，底部是两个子句节点。

更正式地说，该模型由两个向量、三个多层感知器(**L**msg, **C**msg, **L**vote)和两层layer-norm（层标准化）的 LSTMs参数化。在每一个时间步骤t，我们有一个矩阵，它的第i行包含对字的嵌入，和一个矩阵它的第j行包含对子句的嵌入，我们分别通过（平铺）（什么是平铺）来初始化它。

层标准化：Layer Normalization和普通的标准化类似，是将网络中的层进行归一化的操作。标准化是将数据按比例缩放，使之落入一个小的特定区间。其中最典型的就是数据的归一化，即将数据统一映射到[0,1]区间上。

**LSTMs**：Long Short Term Memory，即长短期记忆。这是一种为了避免常规RNN的梯度消失而被广泛应用的RNN的特例。（为什么？）

关于为什么LSTM可以解决这个问题我看不懂，只能直接附上讲解人的原文：

LSTM的结构可以如下表示：RNN总是以“覆写”的方式计算状态，根据求导的链式法则，这种形式直接导致梯度被表示成连乘积的形式，以致于造成梯度消失—粗略地说，很多个小于1的项的连乘很快的逼近0。现代的RNN（包括但不限于使用LSTM单元的RNN）使用“累加”的形式计算状态，这种累加形式导致导数也是累加形式，因此避免了梯度消失。



RNN在t时刻的隐藏状态h(t)是由h(t-1)和当前输入Xt得到的，而LSTM可以从图中看出与原始的RNN结构有些许不同，也更为复杂。从上图中可以看出，在每个序列索引位置t时刻向前传播的除了和RNN一样的隐藏状态h(t)，还多了另一个隐藏状态，如图中上面的长横线。这个隐藏状态我们一般称为**细胞状态**(Cell State)，记为C(t)。细胞状态就是长记忆，而隐藏状态就是短期记忆。如下图所示:



除了细胞状态，LSTM图中还多了一种结构，称为门控结构（Gate）。一般包括遗忘门，输入门和输出门三种。

**遗忘门（forget gate）**在LSTM中以一定的概率控制是否遗忘上一层的隐藏细胞状态。其结构如下图所示：



使用上一序列的隐藏状态h(t-1)和本序列数据x(t)通过一个激活函数（一般是sigmoid）得到遗忘门的输出f(t)。sigmoid的输出在[0,1]之间，因此这里的输出代表了遗忘上一层隐藏细胞状态的概率。

**输入门（input gate）：**从下图可以看出第一部分使用了sigmoid激活函数得到结果i(t)，第二部分使用了tanh激活函数得到a(t)，两个函数的结果相乘再去更新细胞状态。

****

细胞状态C(t)由两部分组成，如下图，第一部分是C(t-1)和遗忘门输出ft的乘积，第二部分是输入门的i(t)和a(t)的乘积，C(t)=C(t−1)⊙f(t)+i(t)⊙a(t)，⊙为Hadamard积。



**输出门（output gate）**：h(t)的更新由h(t-1)和x(t)通过sigmoid激活函数得到的结果o(t)与细胞状态C(t)和tanh激活函数组成。

h(t)=o(t)⊙tanh(C(t))

⊙为Hadamard积。