



《人工智能实验》 实验报告

期中 PROJECT CNN & RNN

学院名称：数据科学与计算机学院

专业（班级）：17 级计算机科学与技术

组员 A 学号：17341088

组员 A 姓名：梁超

组员 B 学号：17341178

组员 B 姓名：薛伟豪

期中PROJECT: CNN & RNN

1. 卷积神经网络 (CNN)

1.1. 算法原理

对于全连接神经网络而言,由于相邻两层之间的节点都有边相连,当相互连接的节点个数很多时,节点之间的边也会随之增加,这便会导致权重矩阵的参数数量非常大。神经网络参数过多不仅会耗费大量运算时间,也容易出现过拟合的情况。此外,全连接神经网络很难提取局部不变性特征,需要进行数据增强来提高性能。由于全连接神经网络的这些缺点,卷积神经网络 (CNN) 应运而生。

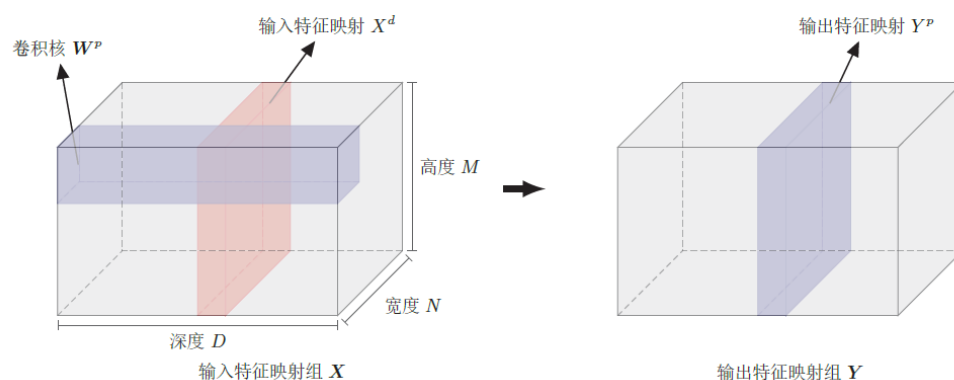
卷积神经网络一般是由卷积层、汇聚层和全连接层交叉堆叠而成,使用反向传播算法进行训练。卷积神经网络有三个结构上的特性:局部连接、权重共享以及空间上的下采样。这些特性使得卷积神经网络具有一定程度上的局部不变性。同时,与全连接神经网络相比,卷积神经网络的模型参数大大减少。

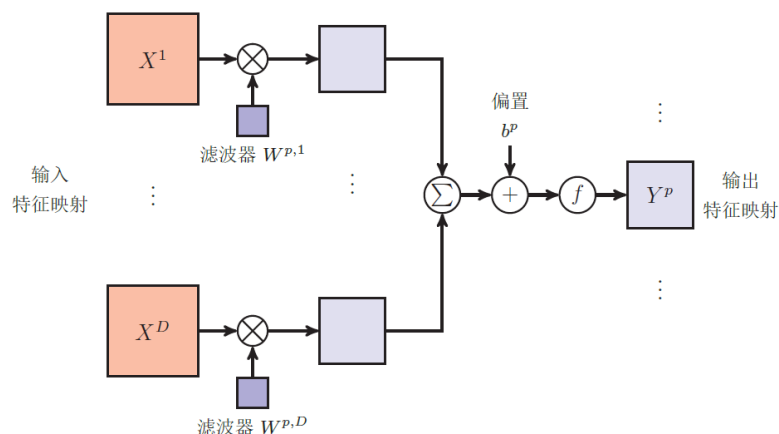
➤ 卷积层

卷积层的作用是提取一个局部区域的特征,不同的卷积核相当于不同的特征提取器。因为卷积神经网络主要应用在图像处理上,所以我们通常将神经元组织为三维结构的神经层,其大小为高度 M ×宽度 N ×深度 D 。如果是灰度图像,那么深度 $D=1$;如果是RGB三个颜色通道的彩色图像,则深度 $D=3$ 。

具体地,我们不妨假设一个卷积层的结构如下:

- 输入: $X \in \mathbb{R}^{M \times N \times D}$, 即 D 个 $M \times N$ 大小的特征输入
- 输出: $Y \in \mathbb{R}^{M' \times N' \times P}$, 即 P 个 $M' \times N'$ 大小的特征映射
- 卷积核: $W \in \mathbb{R}^{m \times n \times D \times P}$, 即 $D \times P$ 个 $m \times n$ 大小的二维卷积核





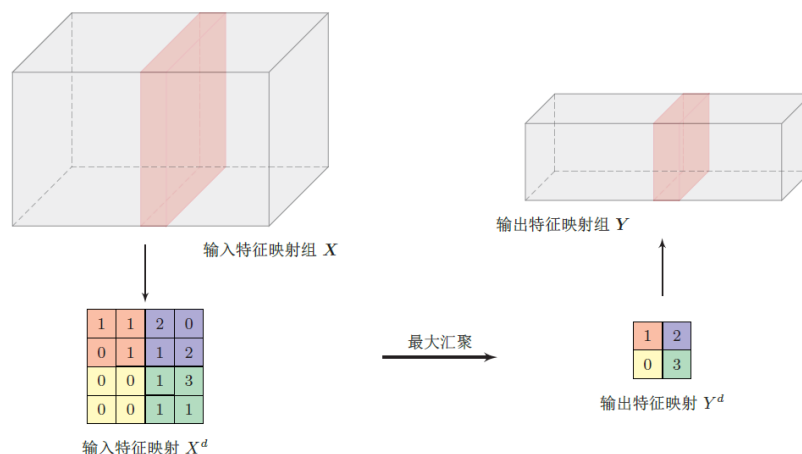
如图所示，我们需要计算输出特征映射 Y^p ，那么我们便需要计算满足 $1 \leq p \leq P$ 的每一个 Y^p ，具体步骤如下：首先用卷积核 $W^{p,1}, W^{p,2}, \dots, W^{p,D}$ 分别对输入特征 X^1, X^2, \dots, X^D 进行卷积，然后将卷积结果相加，并加上一个标量偏置 b^p 得到 Z^p ，最后通过某个激活函数后得到输出特征映射 Y^p ，公式如下：

$$Y^p = f(Z^p) = f\left(\sum_{d=1}^D W^{p,d} \otimes X^d + b^p\right)$$

➤ 池化层

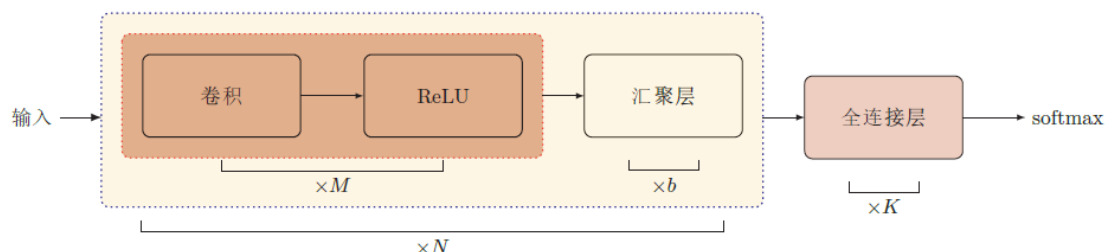
卷积层虽然可以显著减少网络中参数的数量，但特征映射输出的维度依然很高，在此基础上进行分类还是很容易出现过拟合的情况。为此，我们可以考虑在卷积层之后加上一个池化层对输入特征映射组进行下采样，从而降低特征数量，达到减少参数的目的。

同样地，假设池化层的输入为 $X \in \mathbb{R}^{M \times N \times D}$ ，对于其中的每一个 X^d ，我们将其划分为多个（可重叠）局部区域 $R_{m,n}^d$ ，满足 $1 \leq m \leq M'$ ， $1 \leq n \leq N'$ 。所谓的池化，就是从这些局部区域中进行下采样得到一个值，作为这个区域的输出。最常使用的池化类型为最大池化，即取一个区域内的最大值，同时大小与步长均设置为2，效果相当于特征高度和宽度缩减为原来的一半，如图所示：



➤ 卷积神经网络结构

卷积神经网络一般由卷积层、池化层、全连接层交叉堆叠而成。一个典型结构如下所示：



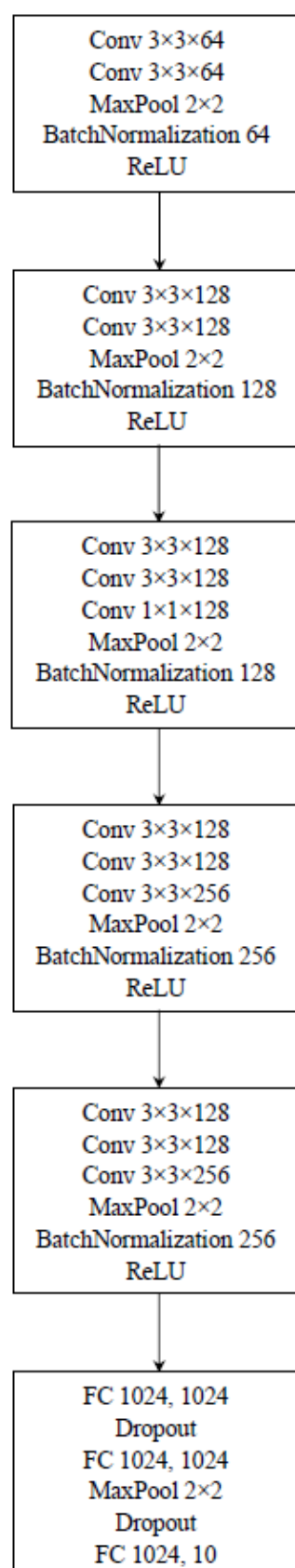
- 定义一个卷积块为连续 M 个卷积层和 b 个池化层（ M 通常设置为2~5, b 为0或1）。
- 一个卷积神经网络中首先堆叠 N 个连续的卷积块（ N 的取值区间比较大，比如1~100或者更大）
- 最后再连接 K 个全连接层（ K 一般位0~2，相当于最后接一个分类器）

➤ VGG神经网络

本次实验采用的CNN结构参考VGG-16。VGG是Visual Geometry Group的缩写。VGG中采用的卷积核主要是 3×3 大小，步长为1，padding为1，池化层为max-pooling的滑动窗口，大小是 2×2 ，步长是2，最后有3层全连接层。VGG最大的特点是使用大量的 3×3 尺寸的卷积核。 3×3 卷积核是能感受上下、左右、中点最小的感受野尺寸。多个 3×3 卷积核叠加的感受野等同于其他更大尺寸的卷积核，在感受野大小一样的情况下， 3×3 的卷积核的优势在于，减少参数，加深网络。此外，VGG中还出现了大量的 1×1 的卷积层，它出现在 3×3 的卷积层后，目的是加深网络， 1×1 卷积核的优势是在不改变感受野的情况下，进行升维和降维，同时可以加深网络而且引入的参数少。

在神经网络中，加入Batch Normalization进行优化。Batch Normalization的好处是加速神经网络的训练过程，减少对Droupout的依赖，可以用较大的学习率去训练网络，并且不用在意权重参数的初始化问题。随着卷积核数量的增加，神经网络层数的加深，神经网络中的参数也会大幅度增加。在这种情况下，随着前向传播的进行，每一层的输入值的分布区间变化很大，为了适应新输入值的分布，模型的参数就很难稳定。因此，引入Batch Normalization的目的就是约束输入值的分布，减缓输入变化的剧烈程度，以此加速模型参数稳定，从而达到加速训练的效果。

1.2. 网络结构



1.3. 结果分析

下面出现的表格的行索引为图像的原标签，列索引为图像的预测标签。

➤ 简单CNN

	plane	car	bird	cat	deer	dog	frog	horse	ship	truck	Accuracy
plane	748	41	65	26	38	10	0	14	58	0	74.80%
car	44	836	16	32	20	11	0	6	35	0	83.60%
bird	85	15	508	81	165	87	0	46	13	0	50.80%
cat	32	25	89	471	98	229	0	44	12	0	47.10%
deer	31	8	78	59	645	62	0	103	14	0	64.50%
dog	14	7	76	166	61	606	0	66	4	0	60.60%
frog	44	99	129	311	277	92	0	29	19	0	0.00%
horse	18	9	42	38	79	91	0	720	3	0	72.00%
ship	131	65	23	15	14	23	0	8	721	0	72.10%
truck	737	449	27	131	31	45	0	100	80	0	0.00%

测试集准确率：52.55%

从上表可以看出，简单CNN的效果很差，其中frog和truck两种标签的图片的准确率是0。frog标签的图片主要错误识别为cat和deer，truck标签的图片主要识别为deer。由于卷积层过少，图片特征提取不充分，因此图片容易错误识别为其他标签。

➤ VGG

➤ 优化方法采用Adam算法

	plane	car	bird	cat	deer	dog	frog	horse	ship	truck	Accuracy
plane	854	7	24	8	12	5	8	13	48	21	85.4%
car	6	934	2	2	1	0	2	0	8	45	93.40%
bird	35	2	788	31	44	35	33	23	5	4	78.80%
cat	13	6	41	616	53	159	54	26	15	17	61.60%
deer	8	2	28	31	837	25	26	39	2	2	83.70%
dog	4	6	28	90	28	791	19	29	1	4	79.10%
frog	6	2	19	25	11	17	908	4	5	3	90.80%
horse	5	1	11	17	27	22	5	906	0	6	90.60%
ship	27	15	7	4	0	3	4	1	921	18	92.10%
truck	7	41	1	4	2	2	2	4	13	924	92.40%

测试集准确率：84.790%

➤ 优化方法采用SGD算法

	plane	car	bird	cat	deer	dog	frog	horse	ship	truck	Accuracy
plane	861	10	38	15	11	1	3	5	34	22	86.1%
car	5	941	5	2	1	1	0	0	12	33	94.10%
bird	35	1	822	27	41	31	22	9	7	5	82.20%
cat	9	3	54	708	32	112	37	25	8	12	70.80%
deer	5	1	56	48	834	18	14	22	0	2	83.40%
dog	4	7	38	106	25	770	16	25	4	5	77.00%
frog	2	5	55	34	14	6	876	2	3	3	87.60%
horse	7	0	16	26	32	29	3	879	1	7	87.90%
ship	37	11	5	3	4	4	2	2	916	16	91.60%
truck	10	45	5	7	0	0	1	2	10	920	92.00%

测试集准确率：85.270%

改用VGG结构的CNN后,测试集准确率大幅度提高,错误识别为其他标签的图像减少。

从实验结果来看,两种优化方法对测试集准确度的影响不大,但是使用SGD算法的CNN在最后测试阶段,识别cat标签的图像准确率比使用Adam算法的CNN准确率提高了接近10%。从表中可以看出,采用Adam算法的CNN,错误率较高的两种图片是cat标签和dog标签的图片,cat标签错误识别为dog标签的图片约占错误识别图片的50%,dog标签错误识别为cat标签约占错误识别图片的50%。优化方法改为SGD后,cat标签图片的识别错误率有所下降,cat标签错误识别为dog标签的占有所有错误识别图片的30%。修改优化算法前后,除上述两种标签的其他标签的图像的准确率变化不大。

➤ 增加卷积核

	plane	car	bird	cat	deer	dog	frog	horse	ship	truck	Accuracy
plane	862	10	22	16	10	0	10	5	35	30	86.2%
car	4	933	1	6	2	0	2	0	11	41	93.30%
bird	42	3	757	70	42	23	42	16	3	2	75.70%
cat	15	9	33	772	42	59	40	16	4	10	77.20%
deer	10	2	38	53	840	14	16	22	2	3	84.00%
dog	5	4	19	206	28	685	20	26	1	4	68.50%
frog	5	2	27	49	16	8	881	5	5	3	88.10%
horse	8	2	15	47	47	24	6	843	0	8	84.30%
ship	37	9	8	13	3	2	8	1	889	30	88.90%
truck	9	41	1	7	1	0	7	2	10	922	92.20%

测试集准确率：83.840%

本次实验还尝试通过增加卷积核的方法,来提高识别的准确率,但是实验效果并不明显。标签为dog和cat两种标签的图片错误率依然很高,而且dog标签图片错误识别为cat标签的图片数量占到了该类标签的总错误识别图像数量的60%。由此可以看出,增加卷积核并不一定能提高模型的预测准确率,由于两种标签的图片互为错误识别的概率较高,可以通过加深神经网络,或者扩充数据集的方法进行优化。

1.4. 创新

在神经网络中,加入 Batch Normalization 进行优化。随着卷积核数量的增加,神经网络层数的加深,神经网络中的参数也会大幅度增加。在这种情况下,随着前向传播的进行,每一层的输入值的分布区间变化很大,为了适应新输入值的分布,模型的参数就很难稳定。因此,引入 Batch Normalization 的目的就是约束输入值的分布,减缓输入变化的剧烈程度,以此加速模型参数稳定,从而达到加速训练的效果。

Batch Normalization(简称 BN)中的 batch 就是批量数据,即每一次优化时的样本数目,通常用在卷积层后,用于重新调整数据分布。假设神经网络某层一个 batch 的输入为 $X = [x_1, x_2, \dots, x_n]$, 其中 x_i 代表一个样本, n 为 batch size。

首先计算 mini-batch 里元素的均值:

$$\mu_B = \frac{1}{n} \sum_{i=1}^n x_i$$

记下来计算 mini-batch 的方差:

$$\sigma_B^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_B)^2$$

这样就可以对每个元素进行标准化:

$$x'_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2}}$$

最后进行尺度缩放和偏移操作,这样可以变换回原始的分布,实现恒等变换,这样的目的是为了补偿网络的非线性表达能力,因为经过标准化之后,偏移量丢失。最终输出为:

$$y_i = \gamma_i \cdot x'_i + \beta_i$$

若 γ_i 表示方差, β_i 表示均值,则实现恒等变换。

对于 CNN, Batch Normalization 实在各个通道分别进行的,假如输入的图像大小是 (N, C, H, W) , 则每层 Normalization 就是基于 $N \cdot H \cdot W$ 个数值进行平均以及方差操作。

Batch Normalization 的好处是加速神经网络的训练过程,减少对 Dropout 的依赖,可以用较大的学习率去训练网络,并且不用在意权重参数的初始化问题。

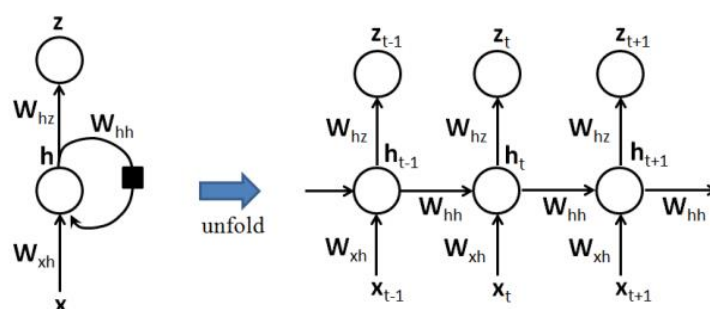
2. 循环神经网络 (RNN)

2.1. 算法原理

在前馈神经网络中,信息的传递是单向的,这种限制虽然使得网络变得更容易学习,但在一定程度上也减弱了神经网络模型的能力。

循环神经网络 (RNN) 是一种具有某种记忆能力的神经网络。在循环神经网络中,神经元不但可以接受来自其它神经元的信息,也可以接受来自自身的信息,形成具有环路的网络结构。与其他类型的神经网络相比,循环神经网络能够对处理过的信息留存一定的记忆,更加符合生物神经网络的结构。循环神经网络的核心思想是:核心思想:样本间存在顺序关系,每个样本和它之前的样本存在关联。通过神经网络在时序上的展开,我们能够找到样本之间的序列相关性。

具体地,一个典型的循环神经网络包含一个输入 x ,一个输出 h 和一个神经网络单元,再经过 $softmax$ 函数后得到 z 。如图的左半部分所示,我们可以很清楚地看到,RNN网络的神经网络单元A不仅仅与输入和输出存在联系,其与自身也存在一个回路。这也说明了RNN的实质,即上一个时刻的网络状态信息将会作用于下一个时刻的网络状态。更具体地,我们将RNN网络以时间序列展开成如下形式:



图的右半部分是RNN的展开形式。展开形式的循环神经网络中,最初时刻0的输入是 x_0 ,经过神经元后得到 h_0 ,在其之后连接有两个支路:一个经过 $softmax$ 函数后得到输出 z_0 ,另一个则作为下一个时刻的输入的一部分。这也就意味着,当下一个时刻1到来时,此时网络神经元的状态不仅仅由时刻1的输入 x_1 决定,也由时刻0的神经元输出 h_0 决定。

所以我们可以作出以下归纳:时刻 t 时,网络神经元的输出不仅仅由时刻 t 的输入 x_t 决定,也由时刻 $t-1$ 时的神经元输出 h_{t-1} 决定。如此下去直到时间序列的末尾时刻。在此基础上我们可以得到以下公式(RNN常用的激活函数为 \tanh 和 sigmoid ,这里我们使用 \tanh 函数),需要说明的是,RNN模型的 W_{xh}, W_{hh}, W_{hz} 三个参数是全局共享的,也就是说不同时刻的模型参数是完全一致的:

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

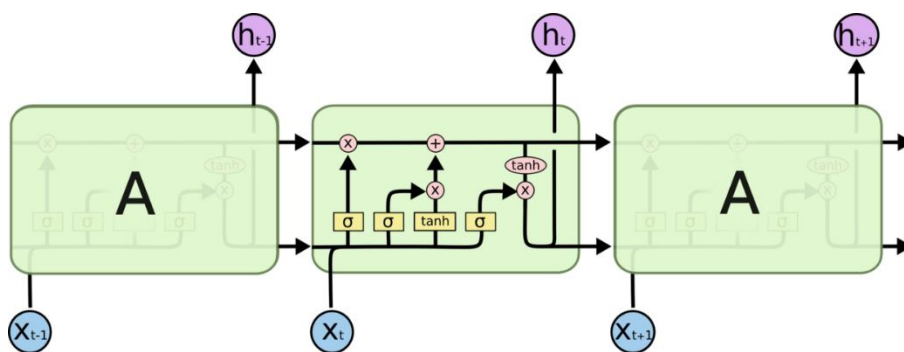
$$z_t = \text{softmax}(W_{hz}h_t + b_z)$$

其中，*softmax*函数可以看作是*sigmoid*函数的一个变种，通常我们将其用在多分类任务的输出层，将输入转化成标签的概率：

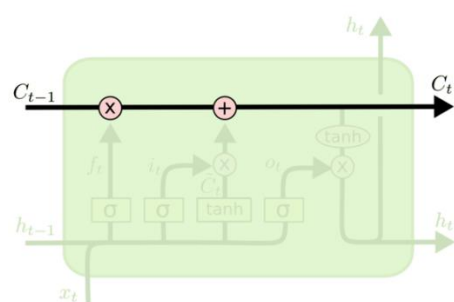
$$h_{\theta}(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 | x^{(i)}; \theta) \\ p(y^{(i)} = 2 | x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k | x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

➤ 长短期记忆网络 (LSTM)

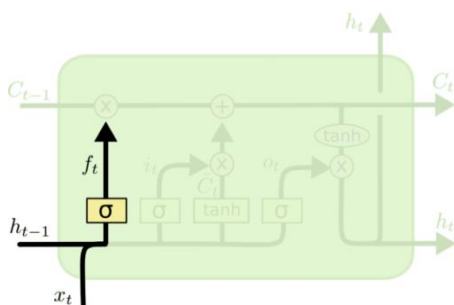
LSTM是一种特殊类型的RNN，它可以学习长期依赖信息。通过引入了自循环，以产生梯度长时间持续流动的路径，解决RNN梯度消失的问题。它在RNN的基础上添加了输入门、遗忘门、输出门和细胞状态。



上图为一个LSTM的结构图，我们可以很明显地看出其比原始RNN复杂了不少。为了说明LSTM模型的工作原理，将每个状态下的神经元拆分成若干个部分进行说明。

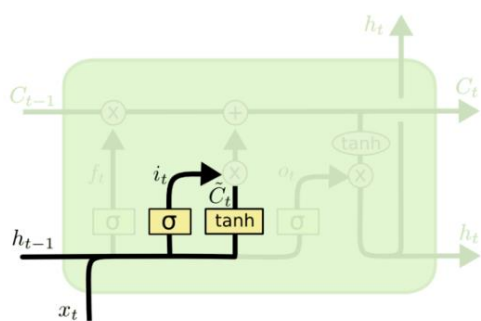


细胞状态类似于传送带。直接在整个链上运行，只有一些少量的线性交互。信息在上面流传保持不变会很容易。



此部分为遗忘门，决定会从细胞状态中丢弃什么信息。遗忘门的输入是 h_{t-1} 和 x_t ，输出一个在区间 $[0, 1]$ 的数值给每个在细胞状态 C_{t-1} 中的数字。1表示完全保留，0表示完全舍弃。

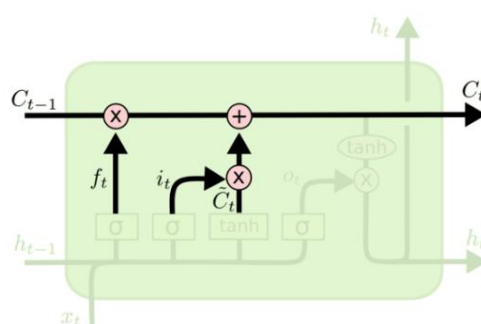
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$



此部分为输入门，决定什么样的新信息可以存入到细胞状态中。输入门由2个部分组成，一部分使用sigmoid激活函数，决定哪些信息需要更新，最后输出为 i_t 。另一部分使用了tanh激活函数，输出为备用的用来更新的内容 \tilde{C}_t 。

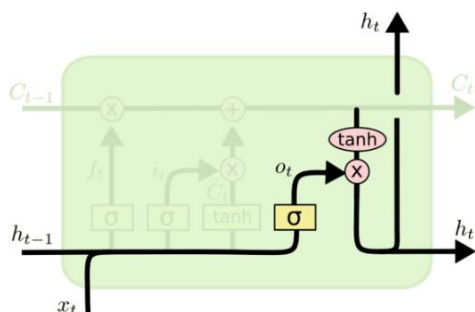
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$



在确定了丢失哪些信息和更新哪些信息之后，就可以对细胞状态进行更新了。首先把旧状态 C_{t-1} 和遗忘门输出 f_t 相乘，丢弃一些需要丢弃的信息。然后把 \tilde{C}_t 和 i_t 相乘，选出新的信息。最后把这2个结果相加就是新的细胞状态。

$$C_t = C_{t-1} \cdot f_t + \tilde{C}_t \cdot i_t$$



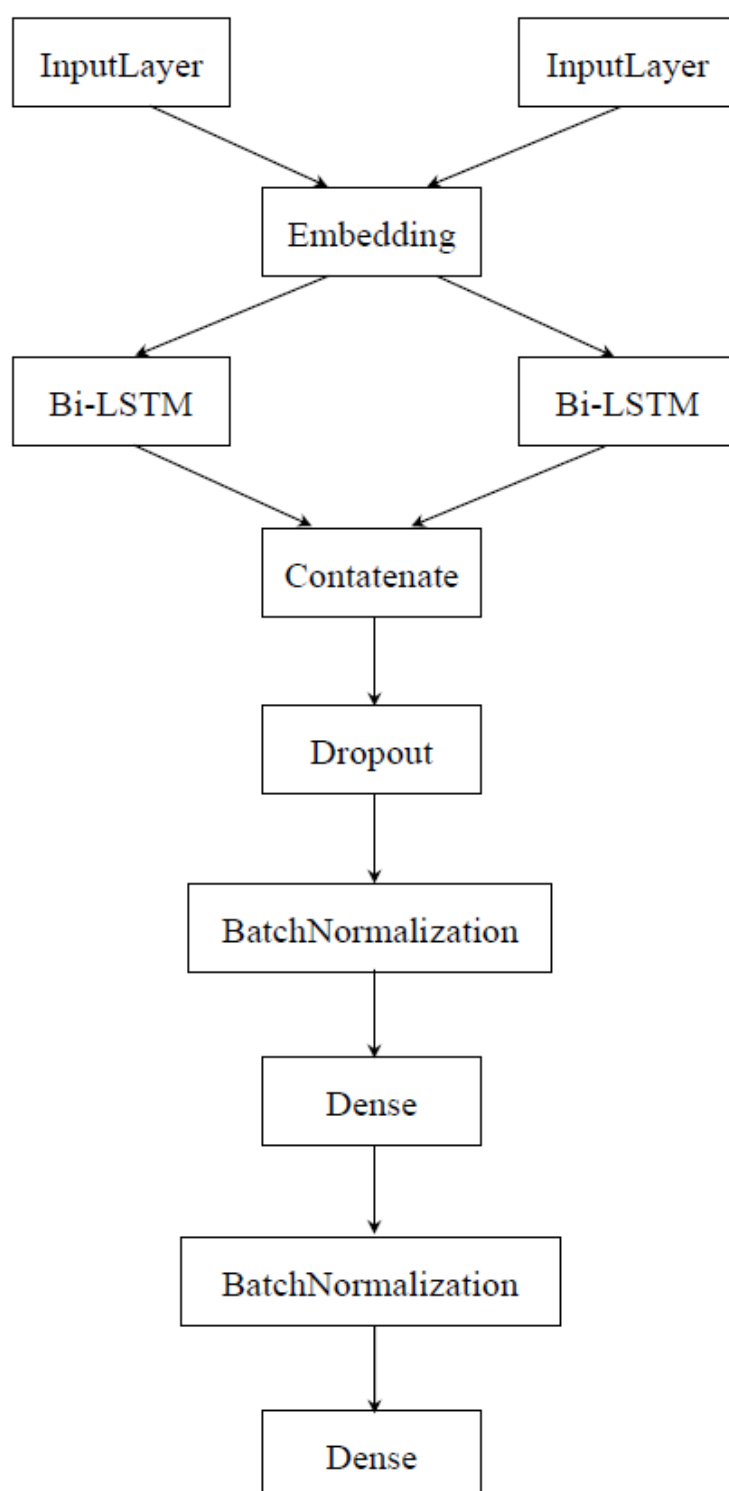
更新完细胞状态之后就可以计算输出。输出也包括2部分：一部分使用sigmoid激活函数，输出为 o_t 。另一部分使用了tanh激活函数激活刚才求得的细胞状态 C_t 。最后将它们相乘，便可求得隐藏层的输出 h_t 。

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

从以上过程分析我们可以看到，LSTM与RNN最大的不同就是LSTM在结构上的改变，LSTM增加了门控结构。LSTM通过门控单元从根本上改变了循环网络的前向传播方式，从叠乘变成了叠加，这样就解决了RNN叠乘所导致的梯度消失问题。

2.2. 网络结构



2.3. 结果分析

➤ 数据预处理结果

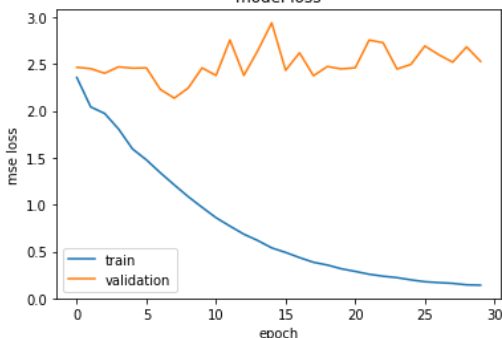
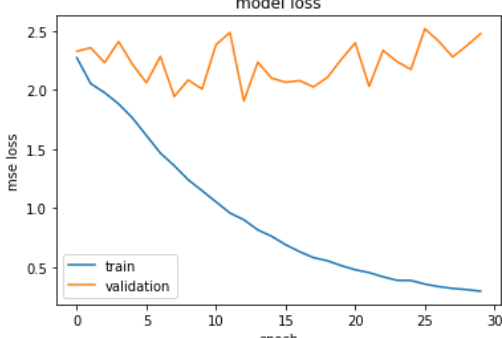
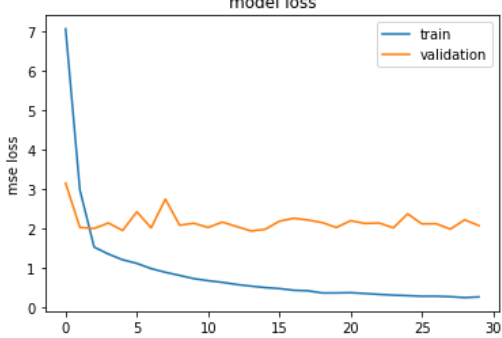
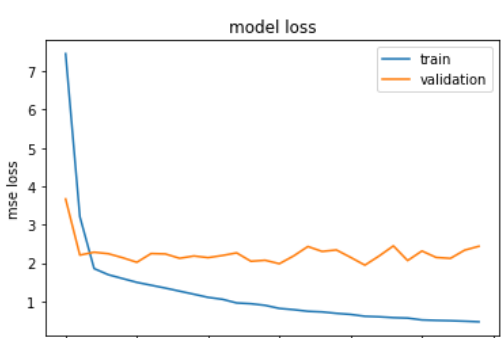
如图所示为实验提供的原始的数据集（包含训练集、验证集和测试集）。简单来讲，实验提供了一系列的英文句子对，每个句子对的两个句子，在语义上具有一定的相似性；每个句子对，获得一个在0-5之间的分值来衡量两个句子的语义相似性，打分越高说明两者的语义越相近。在对数据进行预处理提取有用信息后便可开始正式工作。

Line	main-captions	MSRvid	2012test	ID	Score	Sentence 1	Sentence 2
1	main-captions	MSRvid	2012test	0001	5.000	A plane is taking off.	An air plane is taking off.
2	main-captions	MSRvid	2012test	0004	3.800	A man is playing a large flute.	A man is playing a flute.
3	main-captions	MSRvid	2012test	0005	3.800	A man is spreading shredded cheese on a pizza.	A man is spreading shredded cheese.
4	main-captions	MSRvid	2012test	0006	2.600	Three men are playing chess.	Two men are playing chess.
5	main-captions	MSRvid	2012test	0009	4.250	A man is playing the cello.	A man seated is playing the cello.
6	main-captions	MSRvid	2012test	0011	4.250	Some men are fighting.	Two men are fighting.
7	main-captions	MSRvid	2012test	0012	0.500	A man is smoking.	A man is skating.
8	main-captions	MSRvid	2012test	0013	1.600	The man is playing the piano.	The man is playing the guitar.
9	main-captions	MSRvid	2012test	0014	2.200	A man is playing on a guitar and singing.	A woman is playing an acoustic guitar.
10	main-captions	MSRvid	2012test	0016	5.000	A person is throwing a cat on to the ceiling.	A person throws a cat on the ceiling.
11	main-captions	MSRvid	2012test	0017	4.200	The man hit the other man with a stick.	The man spanked the other man with a stick.
12	main-captions	MSRvid	2012test	0018	4.600	A woman picks up and holds a baby kangaroo.	A woman picks up and holds a baby kangaroo.
13	main-captions	MSRvid	2012test	0019	3.867	A man is playing a flute.	A man is playing a bamboo flute.
14	main-captions	MSRvid	2012test	0020	4.667	A person is folding a piece of paper.	Someone is folding a piece of paper.
15	main-captions	MSRvid	2012test	0021	1.667	A man is running on the road.	A panda dog is running on the road.
16	main-captions	MSRvid	2012test	0022	3.750	A dog is trying to get bacon off his back.	A dog is trying to eat the bacon off his back.
17	main-captions	MSRvid	2012test	0025	5.000	The polar bear is sliding on the snow.	A polar bear is sliding across the snow.
18	main-captions	MSRvid	2012test	0026	0.500	A woman is writing.	A woman is swimming.
19	main-captions	MSRvid	2012test	0028	3.800	A cat is rubbing against baby's face.	A cat is rubbing against a baby.
20	main-captions	MSRvid	2012test	0029	5.000	The man is riding a horse.	A man is riding on a horse.
21	main-captions	MSRvid	2012test	0030	3.200	A man pours oil into a pot.	A man pours wine in a pot.
22	main-captions	MSRvid	2012test	0031	2.800	A man is playing a guitar.	A girl is playing a guitar.
23	main-captions	MSRvid	2012test	0032	4.600	A panda is sliding down a slide.	A panda slides down a slide.
24	main-captions	MSRvid	2012test	0034	3.000	A woman is eating something.	A woman is eating meat.
25	main-captions	MSRvid	2012test	0035	5.000	A woman peels a potato.	A woman is peeling a potato.
26	main-captions	MSRvid	2012test	0038	4.800	The boy fell off his bike.	A boy falls off his bike.
27	main-captions	MSRvid	2012test	0040	5.000	The woman is playing the flute.	A woman is playing a flute.
28	main-captions	MSRvid	2012test	0042	4.200	A rabbit is running from an eagle.	A hare is running from an eagle.
29	main-captions	MSRvid	2012test	0044	4.200	The woman is frying a breaded pork chop.	A woman is cooking a breaded pork chop.
30	main-captions	MSRvid	2012test	0046	4.000	A girl is flying a kite.	A girl running is flying a kite.
31	main-captions	MSRvid	2012test	0047	4.000	A man is riding a mechanical bull.	A man rode a mechanical bull.
32	main-captions	MSRvid	2012test	0048	4.909	The man is playing the guitar.	A man is playing a guitar.

实验开始，我采用one-hot对句子进行编码，但是发现效果不是特别好，同时由于矩阵过于庞大，需要花费大量时间进行训练，因而我转而采用了Word Embedding，这里使用了glove的预训练模型，预训练的模型语料是glove.840B.50d。最后构造出的EmbedMatrix如下所示，其中横坐标表示单词映射的Id，该行表示其对应的向量：

	0	1	2	3	4	5	6
1	0.53567	-0.46164	0.3376	1.5365	0.66307	-0.83601	-0.70347
2	-0.17587	1.3508	-0.18159	0.45197	0.37554	-0.20926	0.014956
3	0.36718	-0.4415	0.29724	0.59774	0.35744	1.0793	0.77628
4	0.81386	-0.13653	-0.078142	-0.18285	0.32254	0.25944	-0.39545
5	0.65017	-1.3155	-0.24966	0.10845	-0.61042	0.45391	0.42854
6	0.013441	0.23682	-0.16899	0.40951	0.63812	0.47709	-0.42852
7	0.0045095	-0.0088647	-0.13722	-0.873	-0.021597	-0.23392	0.64114
8	0.62526	1.3771	-1.1196	1.1705	0.2491	0.30182	0.63379
9	-0.055265	1.0692	-0.65828	0.77279	0.31682	0.46432	-0.036506
10	0.45903	0.17633	0.3159	-0.14818	0.19047	-1.0299	-0.8795

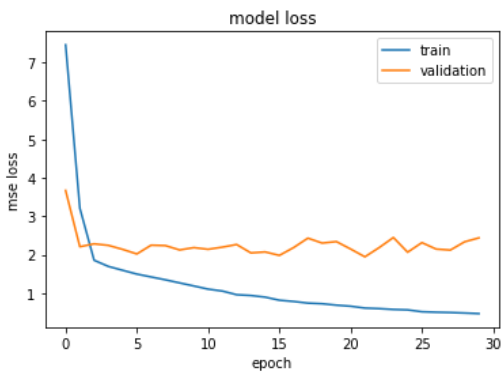
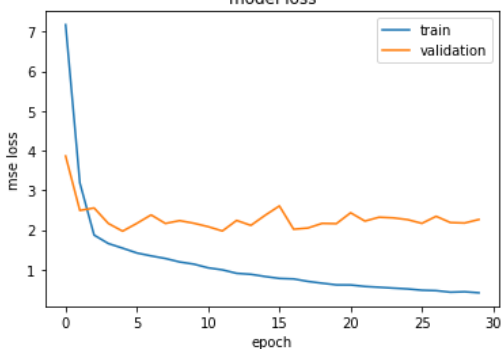
➤ Dropout层和BatchNormalization层对结果的影响

网络结构	训练过程loss变化	测试集loss	预测与实际 相关系数
简单LSTM		2.4008	0.3910
简单LSTM + Dropout层		2.1741	0.4350
简单LSTM + 批归一化		1.9910	0.4760
简单LSTM + Dropout层 + 批归一化		2.0236	0.4706

在神经网络的搭建上,我进行了多种尝试。最开始我采用了最为简单的LSTM网络结构,即[Input -> Embedding -> LSTM -> Dense -> Output],然后再加上减少特征数量的Dropout层和批归一化的BatchNormalization层,训练过程及结果如表格所示(损失函数采用mse)。

不难发现,在简单LSTM网络中加入Dropout层和BatchNormalization层均能够有效提高模型的性能。从图中我们可以观察到,添加BatchNormalization层会让初始的损失函数值变大,但是之后对模型性能的提高相较于添加Dropout层更加明显。训练完成后,测试集的预测结果与实际结果的相关系数达到了0.4760, mse也降到了2以下。而同时加上Dropout层和BatchNormalization层相较于只添加BatchNormalization层并没有能够提高模型性能,相反还有略微的下降。

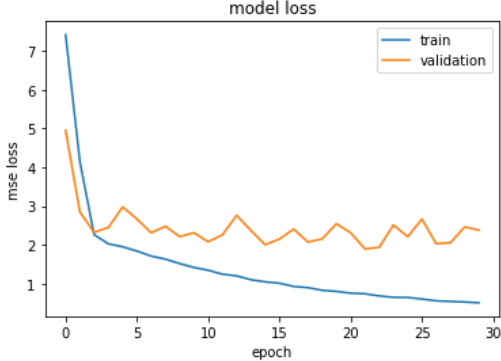
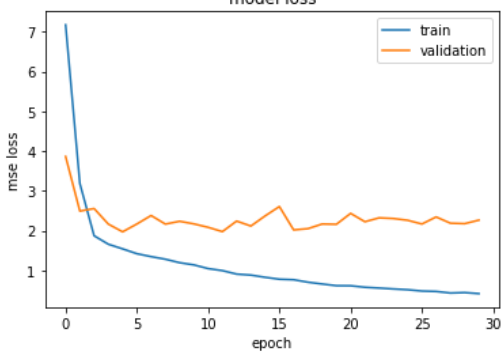
➤ LSTM与Bi-LSTM模型性能比较

网络结构	训练过程loss变化	测试集loss	预测与实际 相关系数
LSTM		2.0236	0.4706
Bi-LSTM		1.9046	0.4927

在LSTM模型的基础上,我们进行更多的尝试,这里主要是将LSTM模型修改为Bi-LSTM(即双向LSTM)。模型的效果如表格所示。

我们可以看到,在使用Bi-LSTM模型后,在测试集上的预测结果相较于LSTM有了显著的提高。训练完成后,后测试集的预测结果与实际结果的相关系数达到了0.4927, mse损失函数值降到了1.9046。

➤ 去除停用词与否对结果的影响

网络结构	训练过程loss变化	测试集loss	预测与实际 相关系数
未进行 停用词过滤的 Bi-LSTM		2.2496	0.4177
进行 停用词过滤的 Bi-LSTM		1.9046	0.4927

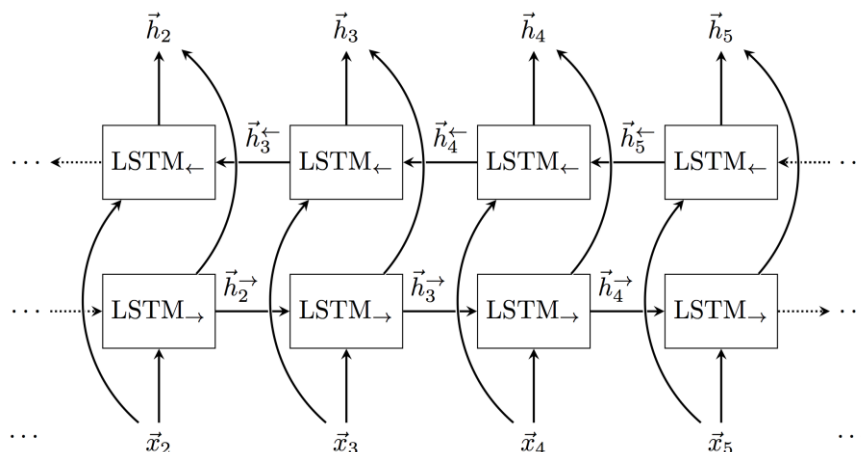
为了提高文本语义相似度评判的准确性,在处理原始文本时,我过滤了其中的停用词(此前模型的训练均使用了停用词过滤后的文本)。去除停用词后,模型效果与原来的对比如下所示。可以发现,去掉停用词能大大提高模型的效果。

2.4. 创新

➤ Bi-LSTM模型

实验伊始,我采用LSTM并在此基础上不断优化。为了能让模型效果进一步提高,我将LSTM改用Bi-LSTM(双向LSTM)进行了尝试,使得模型在测试集上的损失函数降到了1.9左右,预测结果和实际结果的相关系数超过了0.49,可以说有了明显的提升。

如图所示是一个Bi-LSTM层的结构,Bi-LSTM可以看作是两层的LSTM叠加在一起。第一层从左边作为序列的起始输入,在文本处理上可以理解成从句子的开头开始输入,而第二层则是从右边作为序列的起始输入,在文本处理上可以理解成从句子的最后一个词语作为输入,反向做与第一层一样的处理。最后对得到的两个结果进行处理。



➤ 停用词过滤

停用词过滤可用于文本预处理，它的功能是过滤分词结果中的噪声。

在本次实验中，我对实验文本进行了停用词过滤。对于文本语义相关性的评价，对停用词进行过滤可以有效地帮助我们提高关键词密度，使得信息更为集中、突出，更好地对文本的相关性进行评价。从实验结果来看，停用词过滤对提高模型性能有着不小的帮助。

3. 组员分工

小组成员	负责内容
17341088 梁超	CNN模型构建与优化
	实验报告撰写
	PPT制作
17341178 薛伟豪	RNN模型构建与优化
	实验报告撰写与整合
	PPT制作与整合

4. 参考资料

- [1] <https://frank909.blog.csdn.net/article/details/84325722>
- [2] <https://blog.csdn.net/briblue/article/details/83151475>
- [3] <https://blog.csdn.net/briblue/article/details/84201447>
- [4] https://blog.csdn.net/qq_35639867/article/details/79952590
- [5] <https://www.leiphone.com/news/201807/ymZ2a4OI9iBQIZDz.html>