



期中PROJECT

CNN & RNN

人工智能实验

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



目录

CNN

1/ 实现思路

2/ 网络结构

3/ 结果分析

4/ 创新

RNN

5/ 实现思路

6/ 网络结构

7/ 结果分析

8/ 创新



中山大學
SUN YAT-SEN UNIVERSITY

CNN

基于CIFAR-10数据集



实现思路

➤ 数据处理

- 直接利用pytorch的接口加载数据
- 对训练集的图片进行数据增强，即对随机水平翻转，根据概率将图像转换为灰度图，防止训练出现过拟合。
- 对训练集和测试集进行标准化



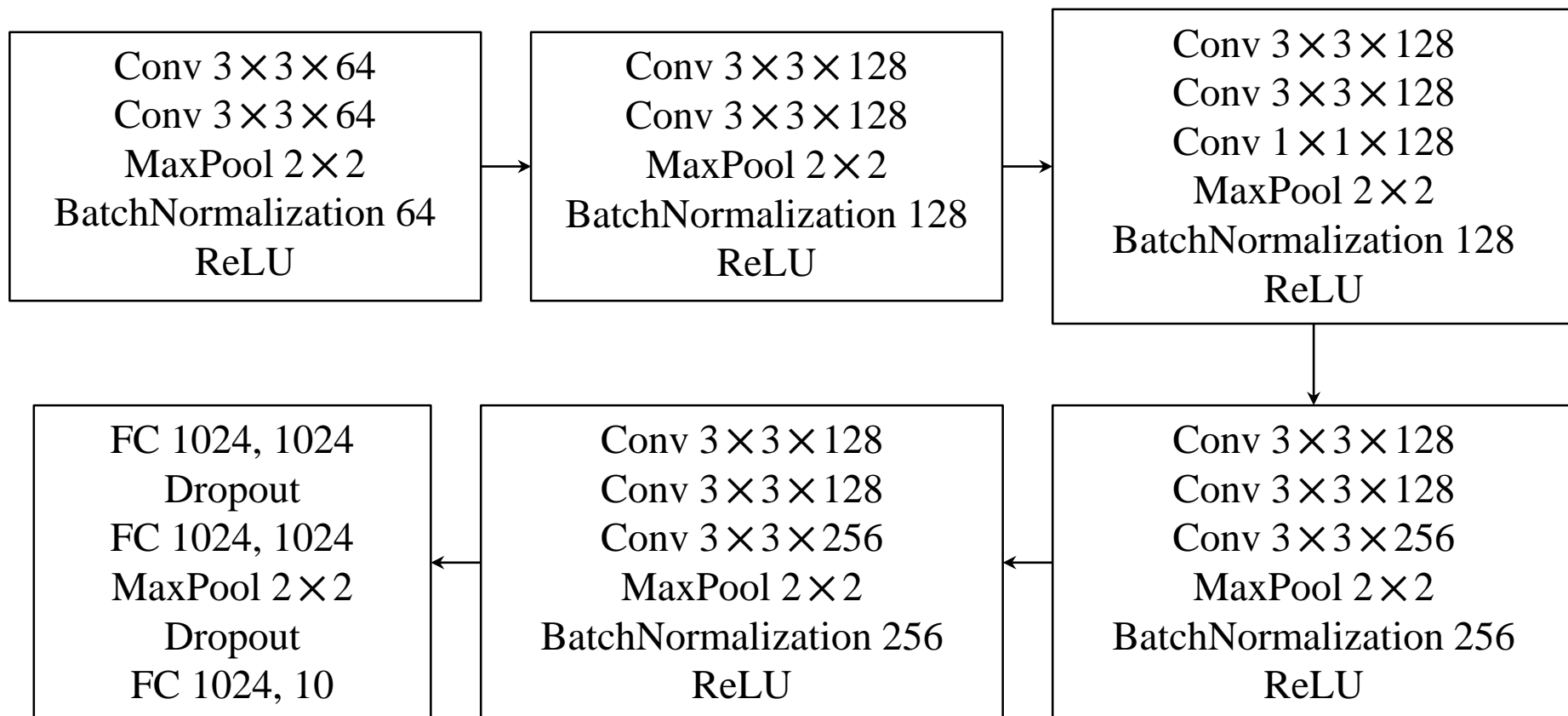
实现思路

➤ 模型构建

- 通过pytorch的Model类API进行模型搭建
- 搭建简单CNN模型，观察实验效果
- 改用VGG结构，堆叠3*3卷积核，加深神经网络
- 加上BatchNormalization层加速训练



网络结构





结果分析

➤ 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%



结果分析

➤ Adam优化算法

	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%



结果分析

➤ SGD优化算法

	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%



结果分析

➤ 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%



创新

- 使用Batch Normalization约束神经网络输入的分布，加速神经网络训练



中山大學
SUN YAT-SEN UNIVERSITY

RNN

基于STS-Benchmark数据集



实现思路

➤ 数据处理

- 从所给的数据集中提取有效信息（提取相关系数、英文句子对，筛掉停用词）
- 对出现的单词进行编码（One-Hot× 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

- 完成单词->id的映射
- 构建Embedding Matrix（第i行表示id为i的单词对应的向量）



实现思路

embedMatrix - NumPy array

	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

Format Resize ☒ Background color

Save and Close Close

Embedding Matrix



实现思路

➤ 模型构建

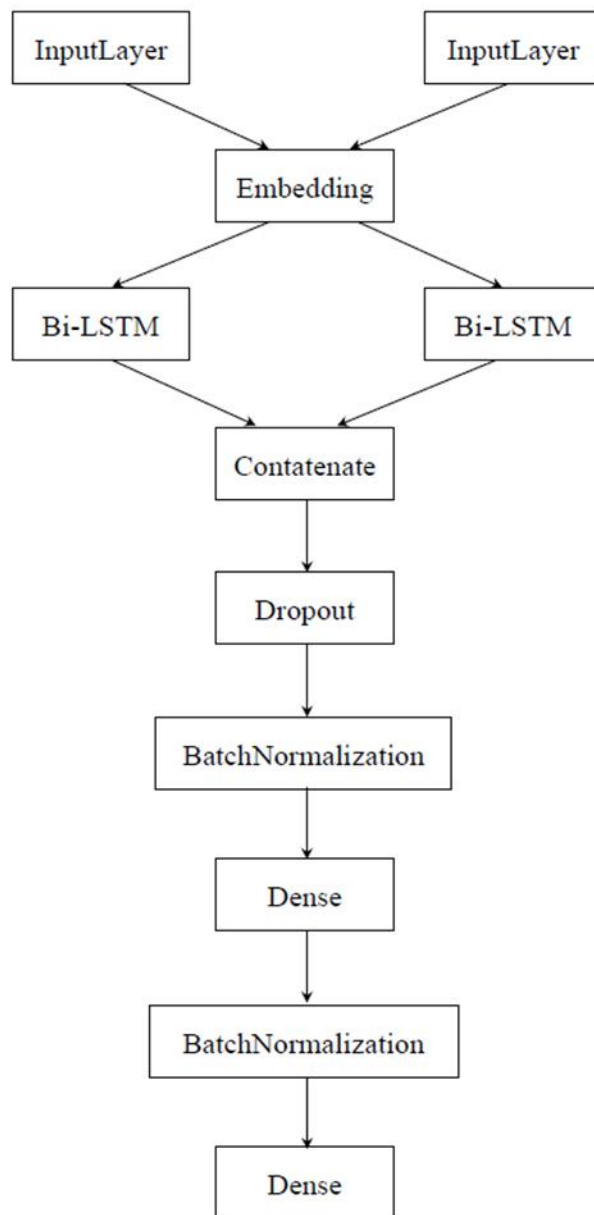
- 通过Keras的Model类API进行模型搭建
- 先搭建简单LSTM模型，在此基础上进行优化

基础结构：Input -> Embedding -> LSTM -> Dense -> Output

- 考虑加上Dropout层和BatchNormalization层进行优化
- 改用Bi-LSTM（即双向LSTM）观察预测能力是否提高
- 个别参数的调整



网络结构

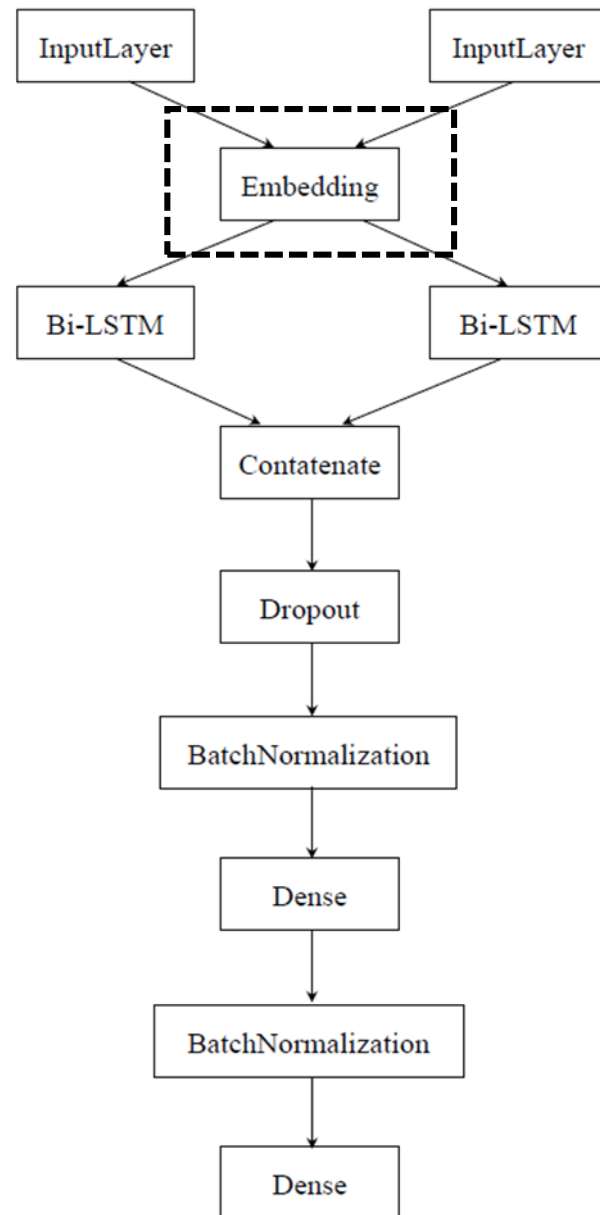




网络结构

➤ Embedding层（词嵌入层）

- 词嵌入层的主要作用是将所有的单词序列映射到一个低维的向量空间。
- 词嵌入层的参数权重是根据预先训练好的词嵌入参数来初始化的，这里我们预训练的模型语料是glove.840B.50d。

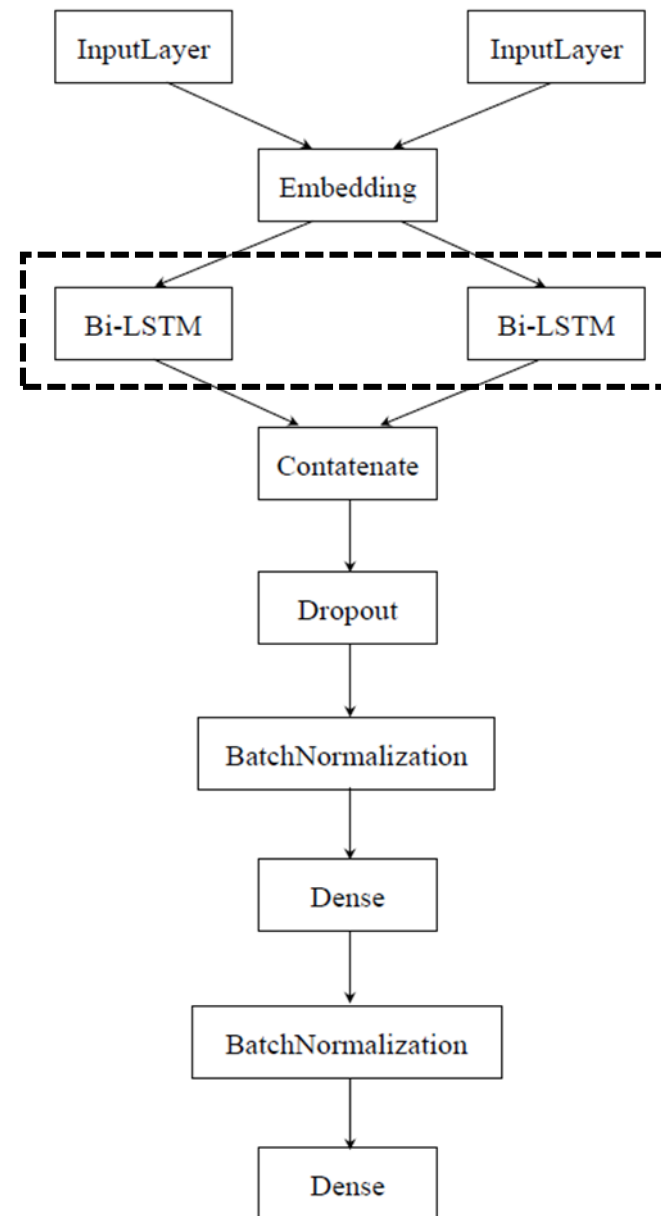




网络结构

➤ Bi-LSTM层

- LSTM网络以文本单词序列来学习句子的表示。
- 一个 Bi-LSTM 是由一个前向的 LSTM 和一个后向的 LSTM 组成。将两个方向的词向量信息进行综合，就可以得出最后的文本特征。

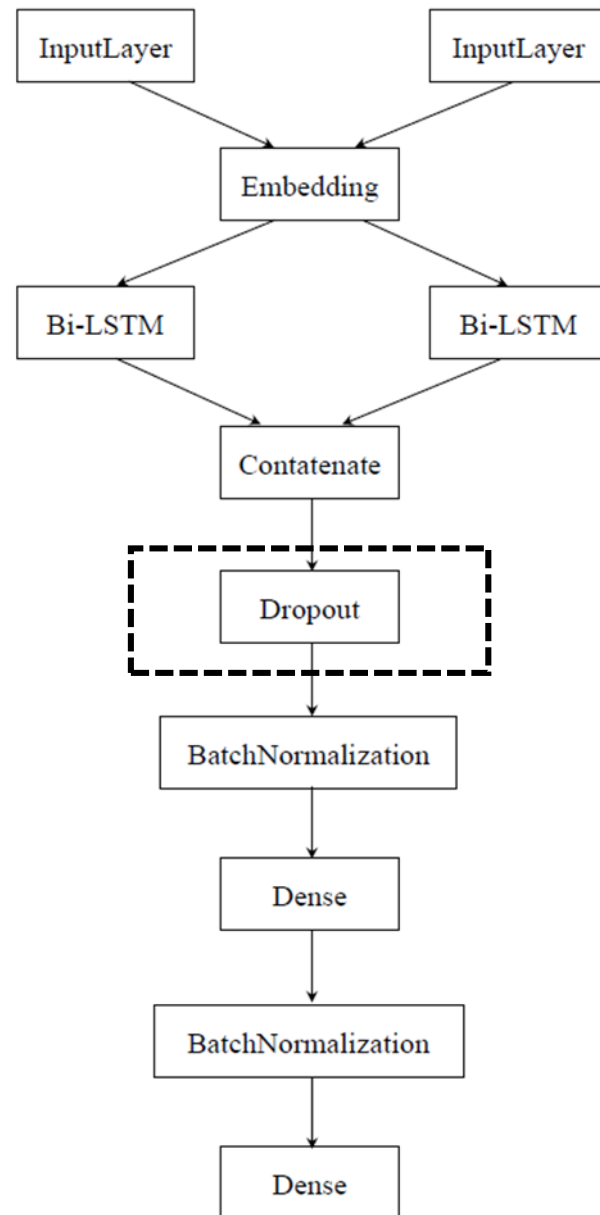




网络结构

➤ Dropout层

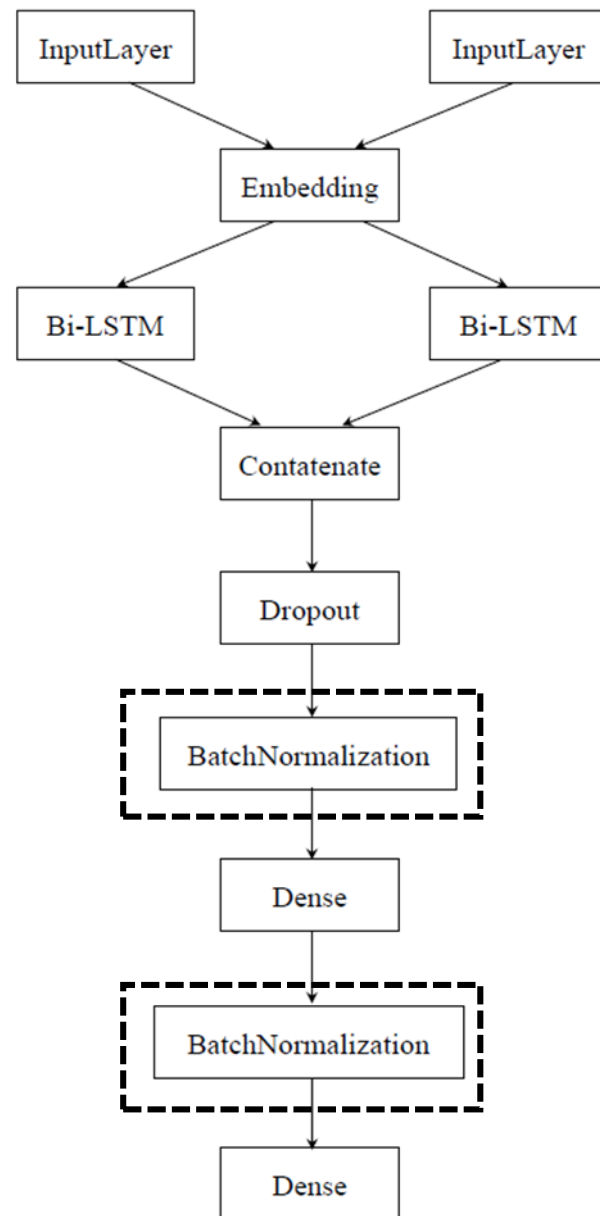
- 随机丢掉某些连接，防止网络出现过拟合情况





网络结构

- BatchNormalization层（批归一化层）
 - 使输出规范化到 $N(0,1)$ 的正态分布
 - 可以让大型网络的训练速度加快很多倍，同时收敛后的预测准确率也可以得到大幅提升

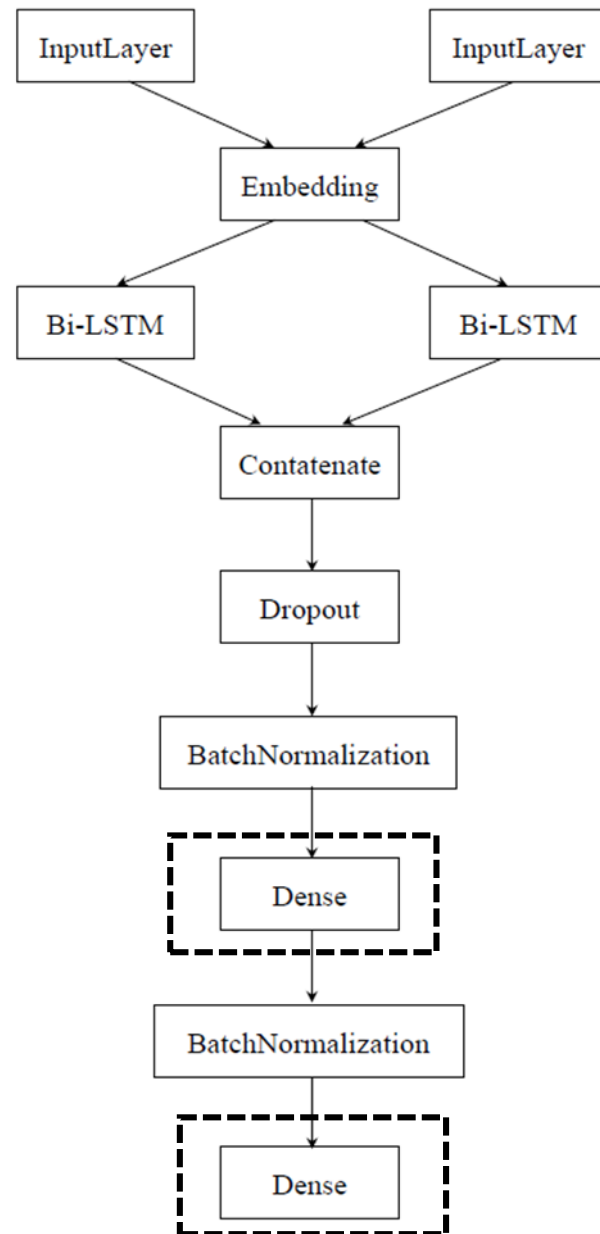




网络结构

➤ Dense层（全连接层）

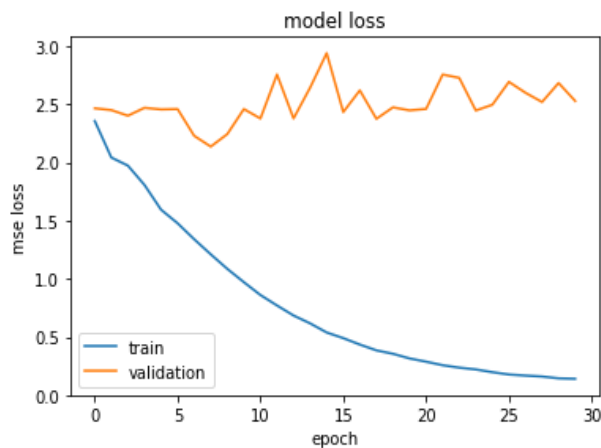
- 让之前提取的特征经过非线性变化，从而提取这些特征之间的关联，最后映射到输出空间上
- 多层Dense可以加快收敛





结果分析

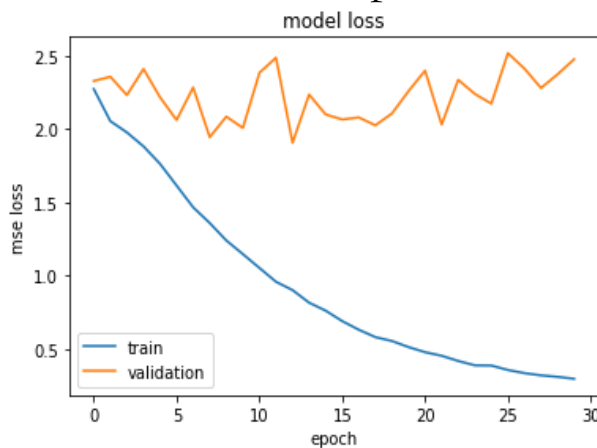
简单LSTM



测试集loss
2.4008

相关系数
0.3910

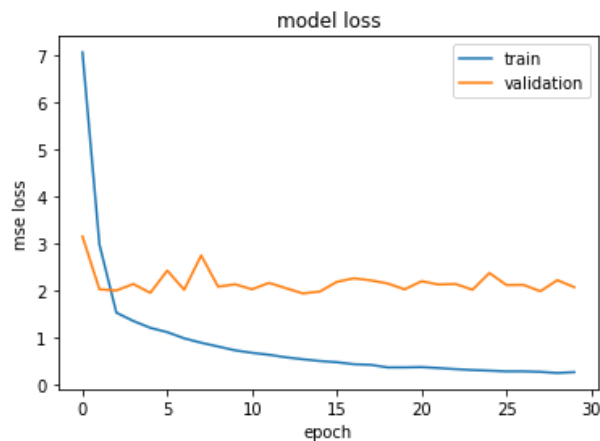
简单LSTM+Dropout层



测试集loss
2.1741

相关系数
0.4350

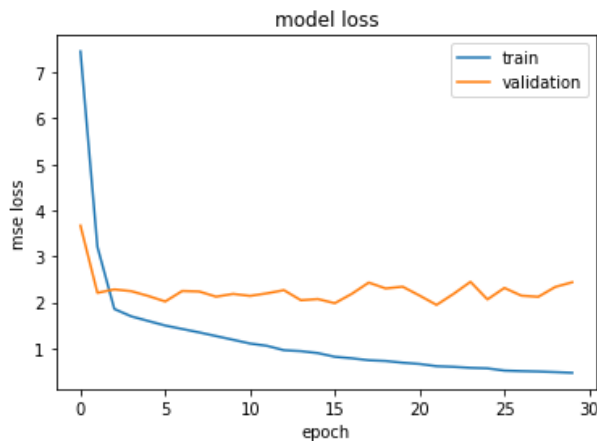
简单LSTM+批归一化层



测试集loss
1.9910

相关系数
0.4760

简单LSTM+Dropout层+批归一化层



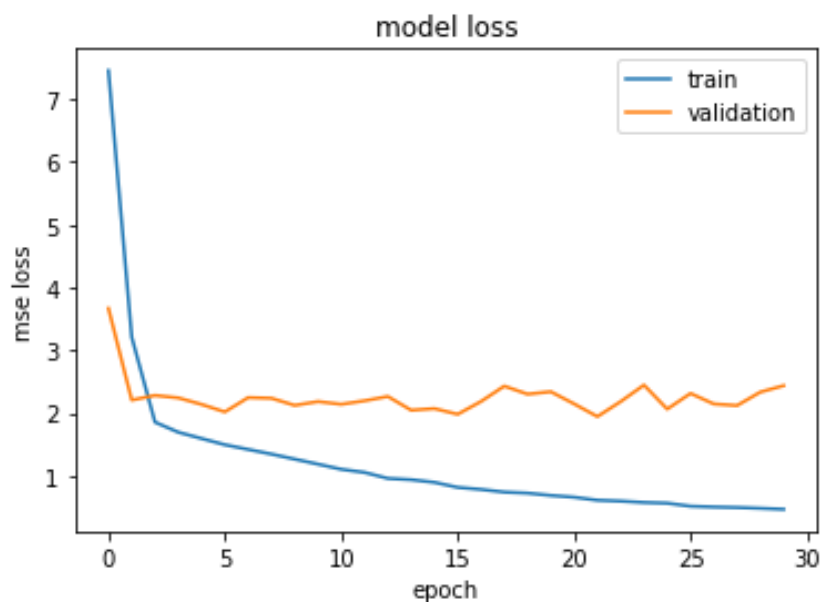
测试集loss
2.0236

相关系数
0.4706



结果分析

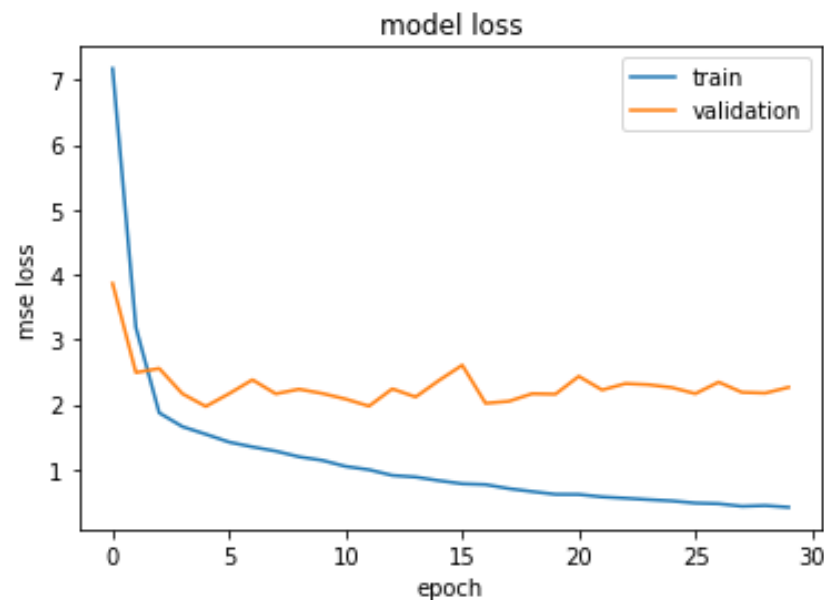
LSTM



测试集loss: 2.0236

相关系数: 0.4706

Bi-LSTM



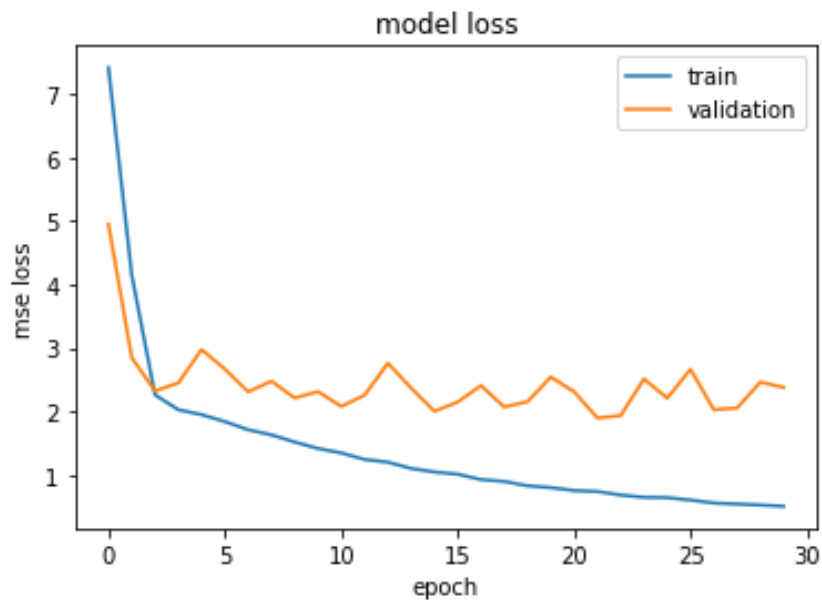
测试集loss: 1.9046

相关系数: 0.4927



结果分析

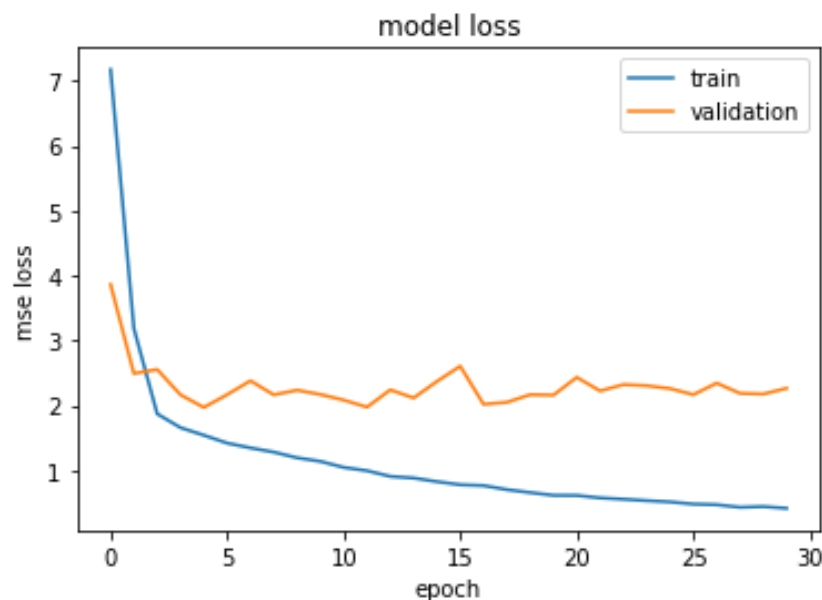
未进行停用词过滤的Bi-LSTM



测试集loss: 2.2496

相关系数: 0.4177

进行停用词过滤的Bi-LSTM



测试集loss: 1.9046

相关系数: 0.4927



创新

- 使用了BiLSTM模型，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
- 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。



THANKS

期中PROJECT
CNN & RNN



- 人工智能实验
- @17341088 梁超
- @17341178 薛伟豪
- 2019.11.08