

NJU_NLP_SummerCamp_2019_report_week3

2019.07.15-2019.07.21

@author Eric ZHU

本周学习内容

1. 完成了院内[课程作业](#)（基于DQN的Atari游戏AI，使用Tensorflow）（所以TF终于可以放一放了）
2. Pytorch 入门：阅读了课程材料中提供的[教程](#) + Pytorch 官方的 [60 MINUTE BLITZ](#) + 部分[官方example](#)与 [doc](#)（只有少数chapter是完整看完的，其他的这几天继续抽空看，因为以后不需要写TF了所以时间多了一点）
3. 论文阅读：[Show and Tell: A Neural Image Caption Generator](#)

论文部分内容

1. 核心思路：对于给定的图像输入 I ，寻找文本序列 $S = \{S_1, S_2, \dots\}$ 使得对应概率 $P(S|I)$ 最大。
2. 启发：MT任务中RNN的Encoder-Decoder模型取得良好效果——对于图像数据，使用在图像分类任务上预训练过的CNN作为Encoder，再用RNN（LSTM）作为Decoder输出Caption文本（Neural Image Caption, NIC）
3. 将在大规模数据上预训练过的 Vision Model 和 Language Model 进行组合，完成图像——文本间的转换过程，减少对训练数据的依赖，提高模型表现。
4. 与先前研究的重要区别：直接把经过处理的图像数据（CNN结果）输入到RNN中，使RNN能够追踪图片中的哪些物体（语义）已经在先前的文字中被提到了。
5. Caption结果的评价：论文中使用了Amazon Mechanical Turk（众包平台）。其他标准：BLEU（生成结果与参考结果的n-grams准确率，效果较好，用作评估标准），perplexity（困惑度，简单，调参时用）
6. 训练过程中遇到过拟合严重的问题，部分是因为训练样本量不足。
7. 过拟合的改进方法：
 1. 使用在ImageNet上预训练的CNN（Help quite a lot in terms of generalization）
 2. 使用过在大语料库上得出的Word Embeddings（No significant gains were observed）
 3. Dropout（a few BLEU points improvement）
 4. 集成学习（a few BLEU points improvement）
8. 相似的数据集（如Flickr30k与Flickr8k）间迁移学习表现较好（标注方式统一），而不相似的数据集（如MSCOCO与Flickr30k）间的迁移学习则导致了BLEU分数的降低（但Caption结果的语义依旧良好，应该是因为不同数据集标注数据的习惯与词汇库不同）
9. 模型结果的Diversity良好，top15 结果的 BLEU 分数达到了59，接近人类水平，说明模型得出结果较为合理且丰富。
10. 训练得出的 Word embedding vectors 很好地捕捉了单词的语义信息。