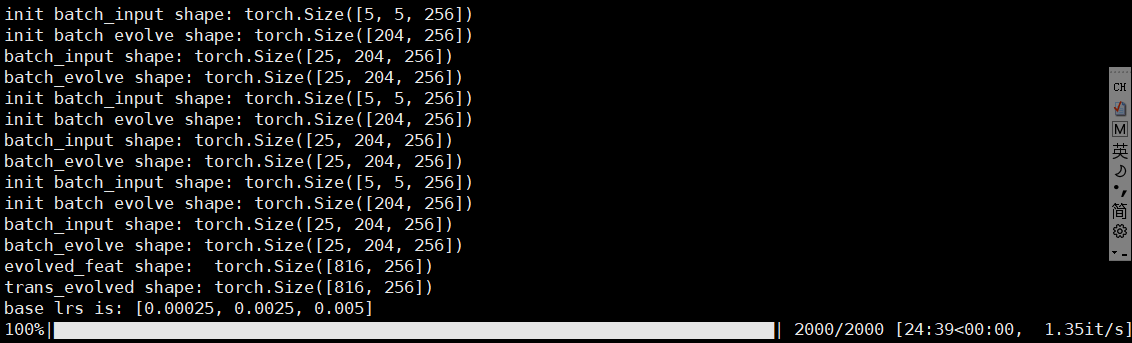
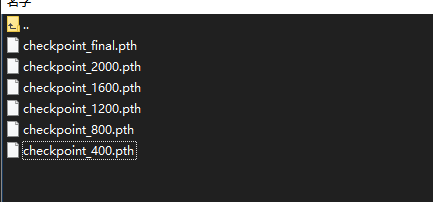
学习周报

1. 学习内容与进度
2. 继续编写论文代码
3. 完成情况：完成了图像和分割任务上的模型训练和测试代码部分，虽然有代码可以用，但还是自己手动重写了一遍；但是知识图上的邻接矩阵和符号节点的语言学嵌入只有cocostuff这个数据集，也不确定其数据可靠
4. 实验结果：

模型训练：





1. 代码经验总结：

【图片读取的维度问题】：

|  |  |
| --- | --- |
| a=Image.open('xxx.png').convert('RGB')  [H，W，C] | 不进行转化读出的就是灰度图像，读出的结果还是PIL.Image, 需要使用np.asarray(image)进行转化 |
| b=scipy.misc.imread('xxx.png',mode='RGB')  [H，W，C] | 加不加上mode=“RGB”都是读出的三通道RGB图片,且读取的直接是numpy矩阵 |
| c=cv2.imread（image\_path)  cv2.imwrite("xxx.png", c) | 也是默认读取的三通道，可以不加上cv2.IMREAD\_COLOR,且本身读取的就是numpy数组;  若想要读取灰度值图像，则可以直接使用  cv2.IMREAD\_GRAYSCALE |

【图片更改维度】：

|  |  |
| --- | --- |
| image = cv2.resize(image, (w, h), interpolation=cv2.INTER\_LINEAR) | image原本是三维的[h,w,c], resize之后任然是三维的 |
| image = F.interpolate(image, size=(H,W),  mode='bilinear', align\_corners=False)  F.interpolate(x,scale\_factor=scale\_f, mode="bilinear", align\_corners=False) | 既可以直接指定缩放后的维度大小，也可以直接使用缩放因子；    插值模式为双线性时，输入必须是[?,C,H,W]这种4D格式，通道数在1维处 |

【pytorch中矩阵相乘的技巧】：

|  |  |
| --- | --- |
| 三维矩阵乘以二维向量  [？，M， Dc] @ [Dc，Dl] =>[？，M，Dl] | 如果使用torch.mm()，则必须进行维度转换  [?,M,Dc] => [?\*M,Dc] @ [Dc,Dl] =>[?\*M,Dl]=>[?,M,Dl];  可以直接使用torch.matmul(); |
| 三维批量矩阵相乘  [?, M, N] @ [?, N, Q] => [?,M,Q] | 如果使用torch.mm()，则只能使用index单独对每个矩阵相乘，然后stack；  可以直接使用torch.matmul()或者torch.bmm()完成 |

【常见错误】：

|  |
| --- |
| 记得区分torch.stack和torch.cat，前面一个会增加一个新的维度，后面一个只会增加一个维度方向上的长度 |
| tensor的形状可以用torch.shape[0]，而不能用torch.shape(0),因为torch.shape本身也是torch.Size类型；也可以用torch.size（0），而不能使用torch.size[0] |
| 两个做运算的tensor必须在同一个设备上 （+ - / \* ），即同在cpu或者同在gpu |
| float和double类型的tensor不能一起运算 |
| 自定义模型中的参数如不是self.conv等这种结构自带的模块参数，都需要注意将参数放在相应设备上, 并进行模块参数的注册 |
| 尽量使得函数的输入和输出都是tensor，其他情况才把tensor换成numpy数组 |
| 使用nn.DataParallel时注意必须使用to(device)这种语法，而不能直接使用cuda（），  且模型的方法都要变成model.module.method(),包括在模型预测时应改为model.module(image) |

1. 继续开展毕设

将原来的代码重新看了一下，但是还没修改

1. 学习问题
2. 找到了fasttest工具来生成词向量，但是没找到使用wordTree获取层次概念图的方法和代码，导致数据有限
3. 毕设进度较慢， 主要原论文理解方面可能存在偏差
4. 学习计划和安排
5. 研究毕设论文，争取能修改好论文的代码
6. 补充学习tensorflow框架的知识，对这个框架用的也不多