1. **目的**

材料的性能与材料微观结构密切相关，不少期刊/会议论文中也明确结构直接决定性能。利用深度学习方法分析材料微观结构图像，从而建立回归模型预测出材料的性能是一种非常优秀的方法，然而，当下的许多工作中仅仅做到了将在其他视觉任务上表现优异的方法直接迁移到材料图像领域，预测材料性能，如，将在coco，ImageNet表现优异的resnet，alexnet等直接迁移到材料图像上，将最终的分类网络换成输出值为1个类别（性能值）的网络，然后加以训练，用以预测材料性能。

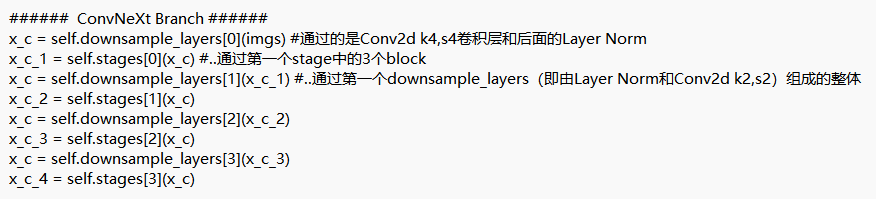
然而，在材料类的论文中，也明确了影响材料性能的不仅仅局限于材料图像的某些具体结构，事实上，全局特征和局部特征一样非常重要，局部特征细致刻画了材料的纹理，晶界密度以及孔隙率等，而全局特征则考虑了不同特征之间的长程依赖，可以发现材料中不同位置之间的相互作用关系，综合考虑更全面的信息，建立更加完整的映射关系。本程序所构建的网络同时考虑了局部网络分支和全局网络分支，分别对材料图像的局部特征和全局特征进行提取，并提出了专门的特征融合模块，对特征进行精细化融合处理，最终准确预测性能。

1. **工程包含文件即代码详解**
2. Model\_Conv\_Resnet50\_MHSA\_scse.py，这个文件是整个程序的核心文件，它构建了本文所提出的模型，具体来说，细节如下：

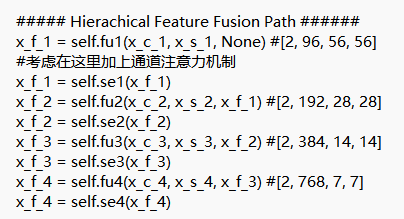
全局分支代码部分：



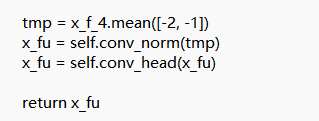
局部分支代码部分：



融合模块代码部分：

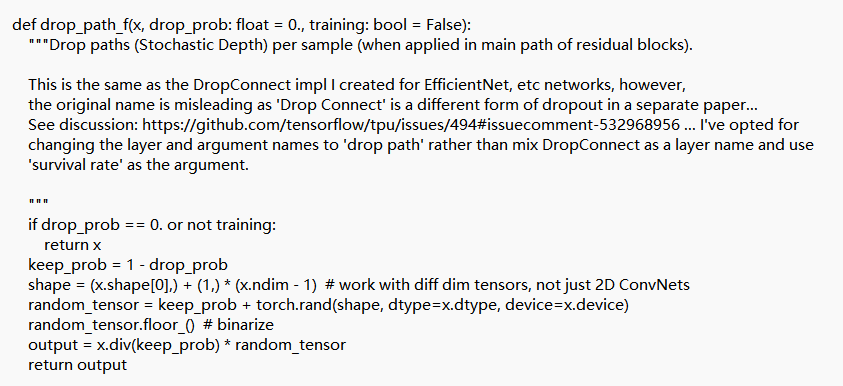


基于融合得到的结果，即x\_f\_4，对其进行展平等处理，融入回归器：

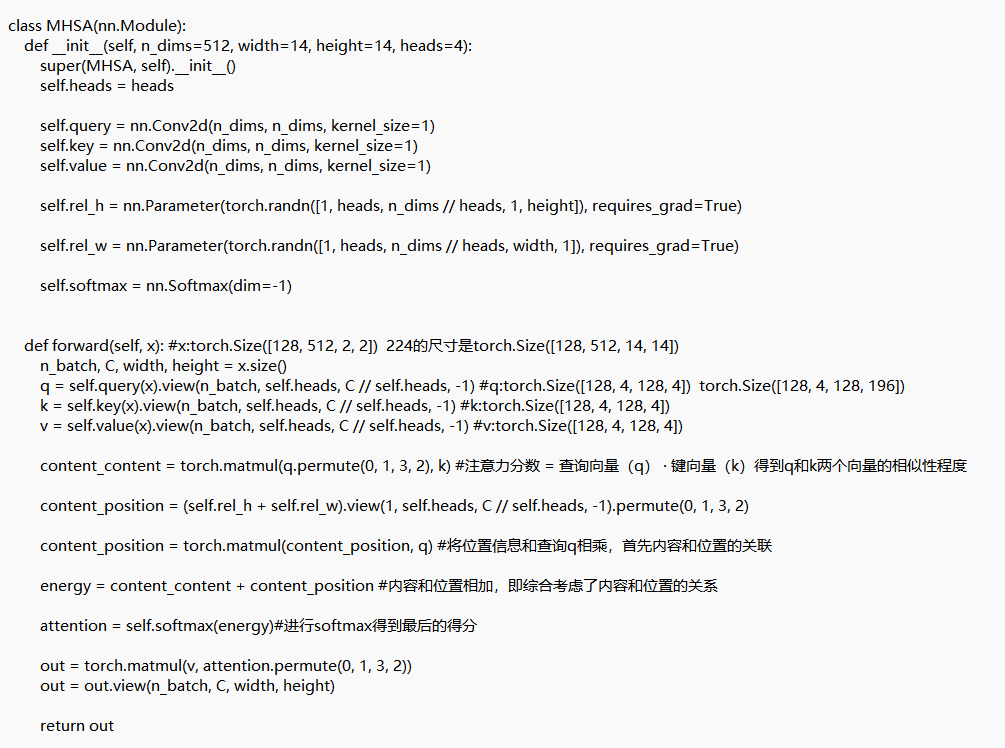


上述内容是整个程序的forward部分，即运行逻辑

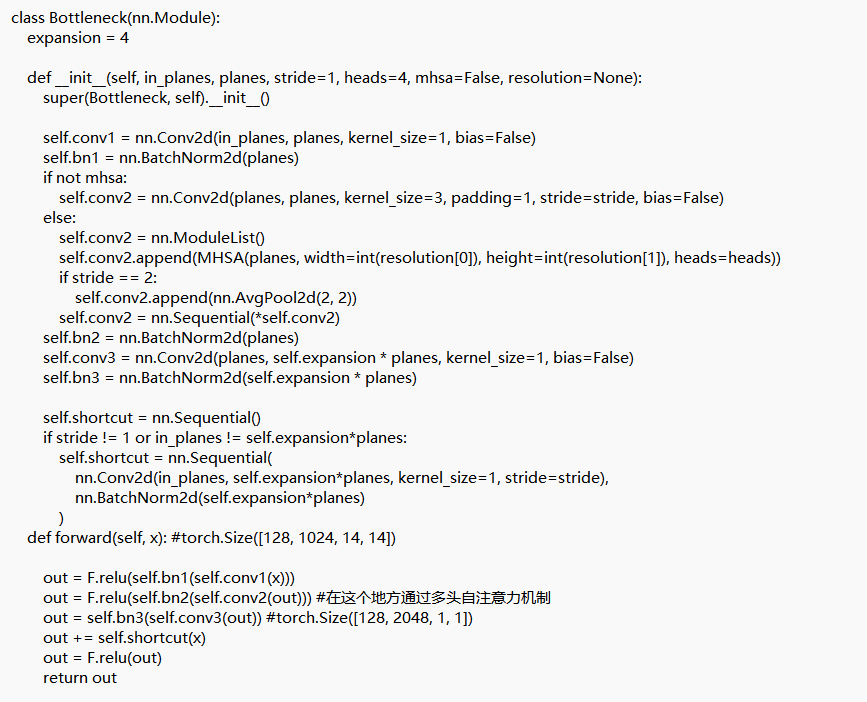
代码中的一些细节问题有：



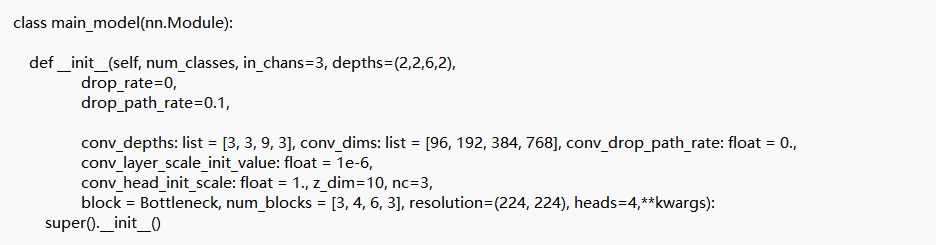
这部分代码，主要是在训练的时候，为了让一些神经元失活，但是具体做的时候，我没用上，因为进行失活后，性能并无改善。



这部分是写的多头自注意力机制，最后其整合在全局分支的最后一个阶段中。



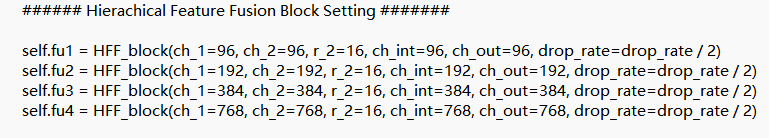
这部分代码是构建全局分支的内部阶段模块部分，其中，在第四个阶段自动整合多头自注意力机制，其余阶段直接跳过。



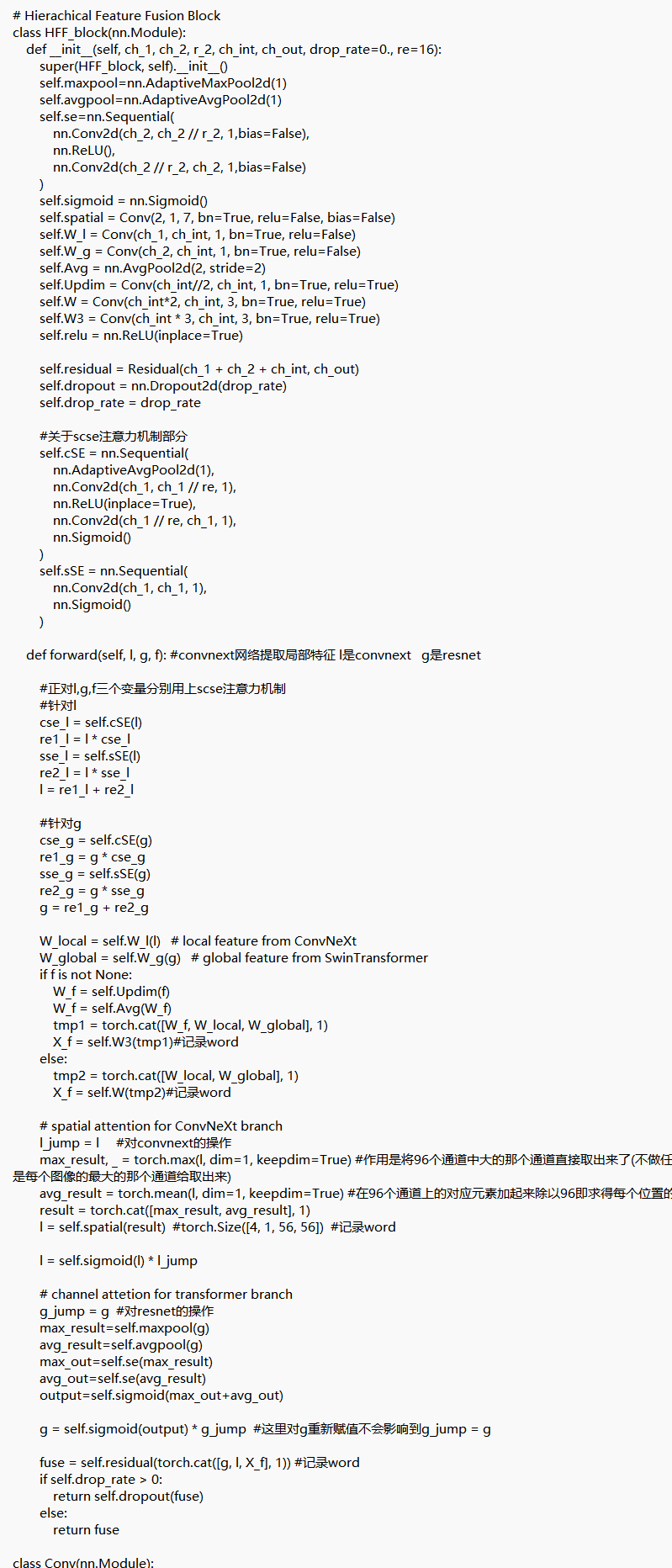
这部分代码主要是初始化变量等内容



这部分代码是构建局部分支部分，其利用了ConvNext的网络代码



这是构建的特征融合模块，一共四个阶段，具体细节如下：

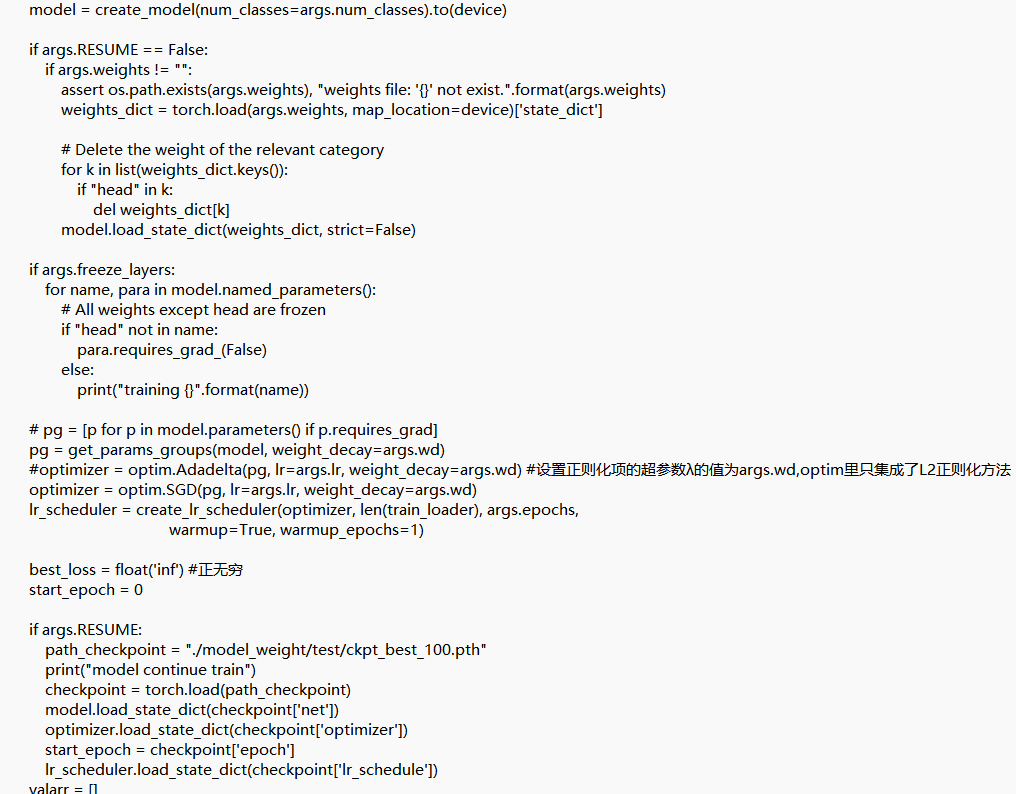


这是融合代码的细节部分，具体来说，其对来自局部分支网络和全局分支网络的不同特征分别进行了不同注意力机制处理和细节的形状变换，具体要理解的话，还是要进行断点调试。

1. Train.py 这个文件的作用是对所构建的模型进行训练，其中



固定了随机种子，传入了训练集和验证集的路径，并进行了相应的图像变换。

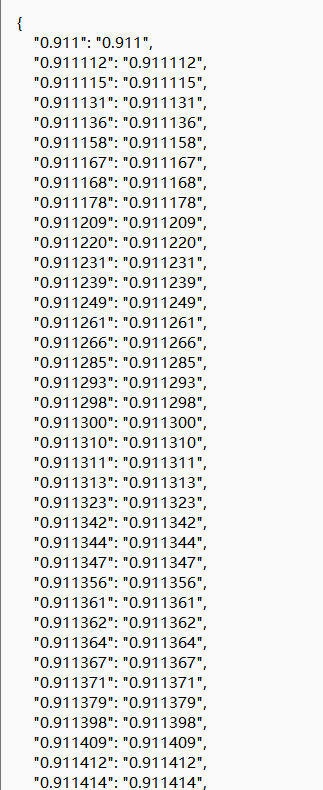


这部分代码创建了模型，冻结了必要的参数（如果有必要），并且设定了一些超参数



这部分代码正式开始了训练和验证。

1. Test.py作用是测试代码，应该传入测试集，期内部代码构造与train.py类似
2. Utils.py主要是封装了一些工具类，如对传入的数据根据文件夹名称获取其对应的标签值，返回处理后程序能使用的数据格式。
3. Class\_indices.json 作用是配合程序将标签类型由文本转化为数值类型，其内部是一个字典，格式如下：



1. Runs 存放可视化需要的文件
2. Checkpoint 存放保存的模型
3. Big\_dataset100 存放数据集，其中有训练，验证和测试三种数据，cold\_hot存放的是对材料图像中局部区域提取出来后进行预测的小块图像。
4. **环境说明**

操作系统：Windows10，Linux

编程语言：Python

运行库函数：NumPy，Pillow，PyTorch，Matplotlib，sklearn