Cascaded EnLSTM代码逻辑

目的：是否cascaded以及是否使用ENN结构可以独立调节，即可以LSTM，Cascaded LSTM，EnLSTM，Cascaded EnLSTM

方法：主要通过修改main函数架构

Main：

判断是否cascaded，若采用cascaded，修改Data\_Loader和config

判断是否有pretrained model，若有，继续训练，相当于对epoch扩充

Train（），EnLSTM和LSTM不同

Predict（）

Draw

EnLSTM的Train（）进行调整：

目前text\_loader数据载入过程和训练后的save\_txt在train（）外，可以放入train（）内部，以便和LSTM的train（）功能上等价。也就是把当前代码中243-260的代码都包在train（）里面。

Main中的Cascaded部分修改为可调节的：

例如config中添加flag，确定是否采用cascaded；

若flag = True，提供一个list。List的元素数目与cascaded层数相同，每个元素本身也为一个list，包含对应层中的物理量名称。比如，[[a1, a2], [a3, a4], [a5]]，对应3层cascaded，第一层预测2个（a1, a2），第二层将第一层结果添加到输入中并预测另外2个（a3, a4），第三层将第一层和第二层的结果都添加到输入中，并预测另外1个（a5）。

Cascaded迭代逻辑修改：

目前epoch是在cascaded之外的，也就是每次会把所有cascaded都训练一遍，整体记为epoch +=1. 实际上每个cascaded的网络的epoch数目应该是可以独立调节的。所以可以考虑对每个cascaded循环对应的epoch数目。可以通过list给出。

这一改变有利于提升灵活性，比如某一层的epoch不足，可以只增加该层的epoch数目，而不是整体增加。

疑问：

目前EnLSTM中draw\_result只能画一条曲线？如果一口井out\_dim>1，则无法使用？当前draw的代码只可用于每层输出一维的cascaded？