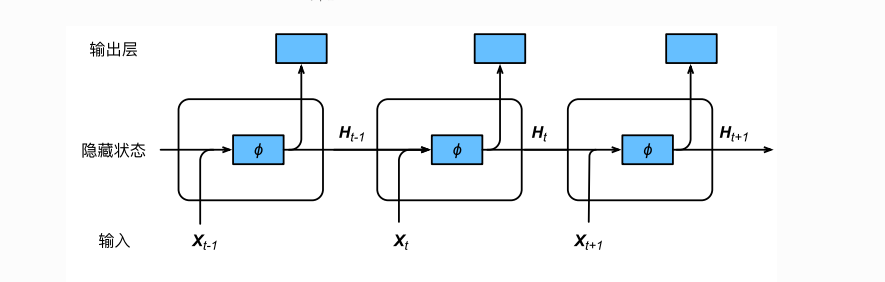
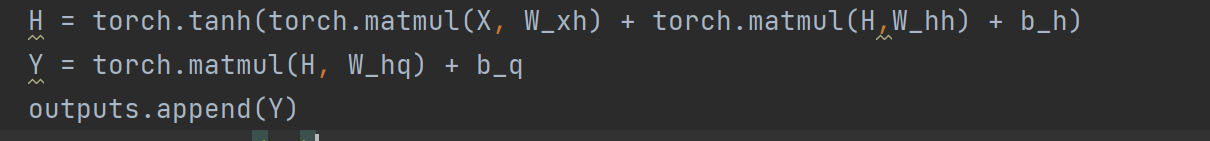
对RNN和LSTM的理解

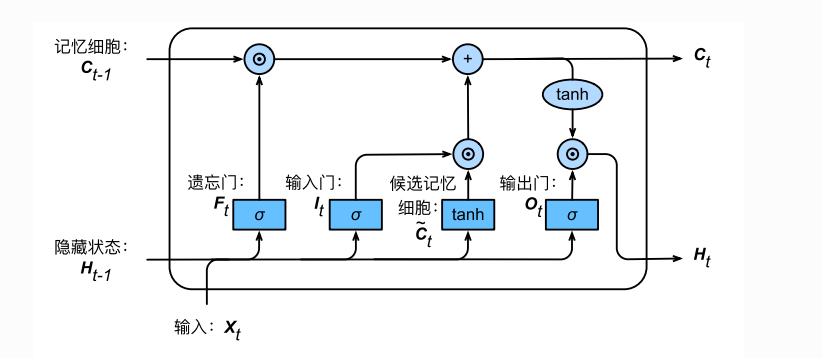
RNN是在一个含有隐藏状态的循环神经网络，这里需要保存上⼀时间步的隐藏变量并引⼊⼀个新的权重参数 ，该参数⽤来描述在当前时间步如何使⽤上⼀时间步的隐藏变量。原理图如下所示



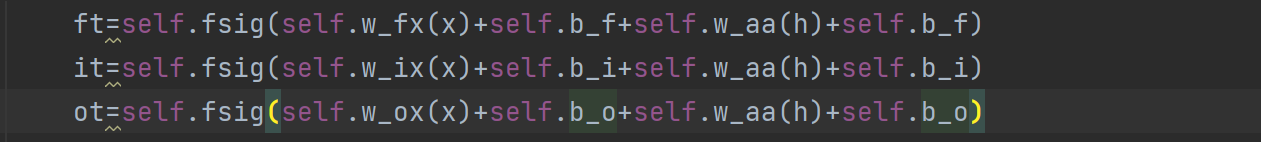


如上所示，其中X是输入，W\_xh是输入层到隐藏臣之间的权重参数，W\_hh是，H是隐藏状态，W\_hh是隐藏状态间的权重参数，b\_h是隐藏状态的一个偏差。

LSTM又引⼊了3个⻔，即输⼊⻔、遗忘⻔和输出⻔，以及与隐藏状态形状相同的记忆细胞，从⽽记录额外的信息。原理图如下所示



各个门的计算公式如下：其中it是输入门，ft是遗忘门，ot是输出门,h是上一个隐藏状态，b\_i、b\_f、b\_o分别是三个门各自的偏差

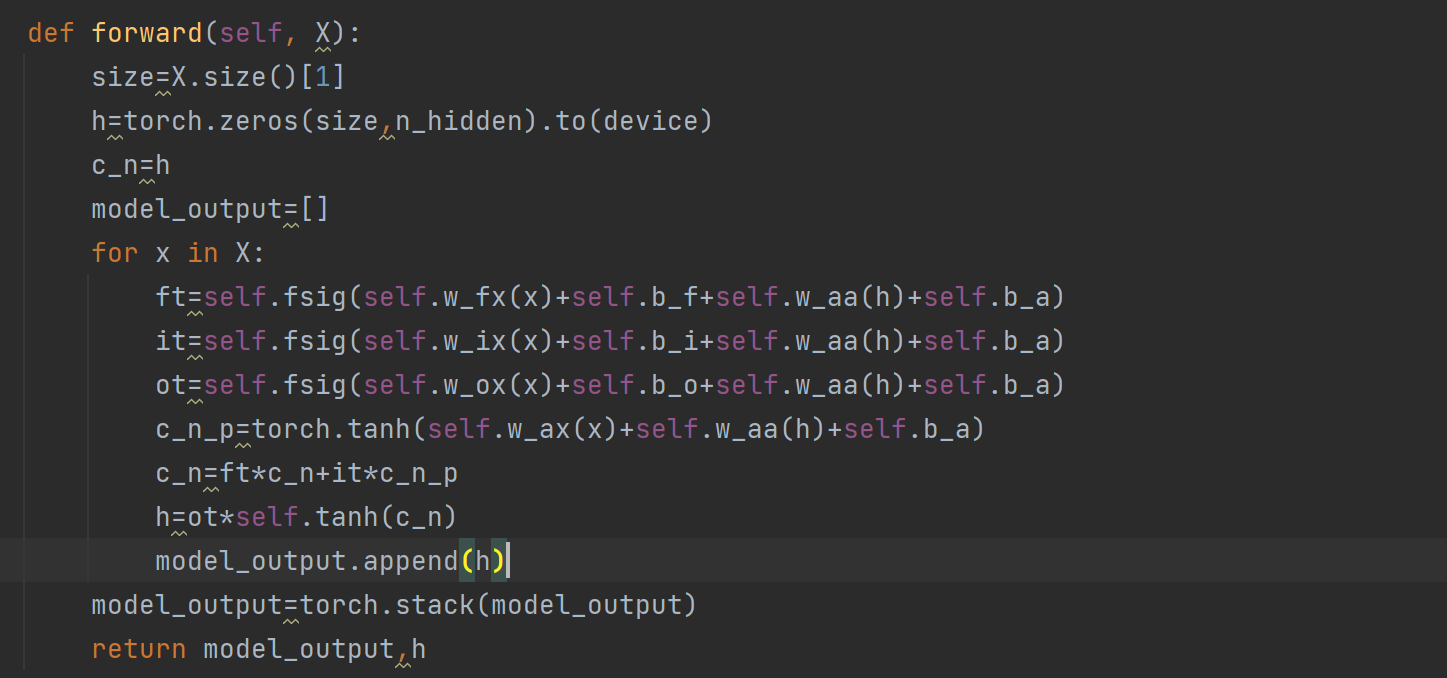
候选记忆细胞的计算公式如下：



记忆细胞计算公式：

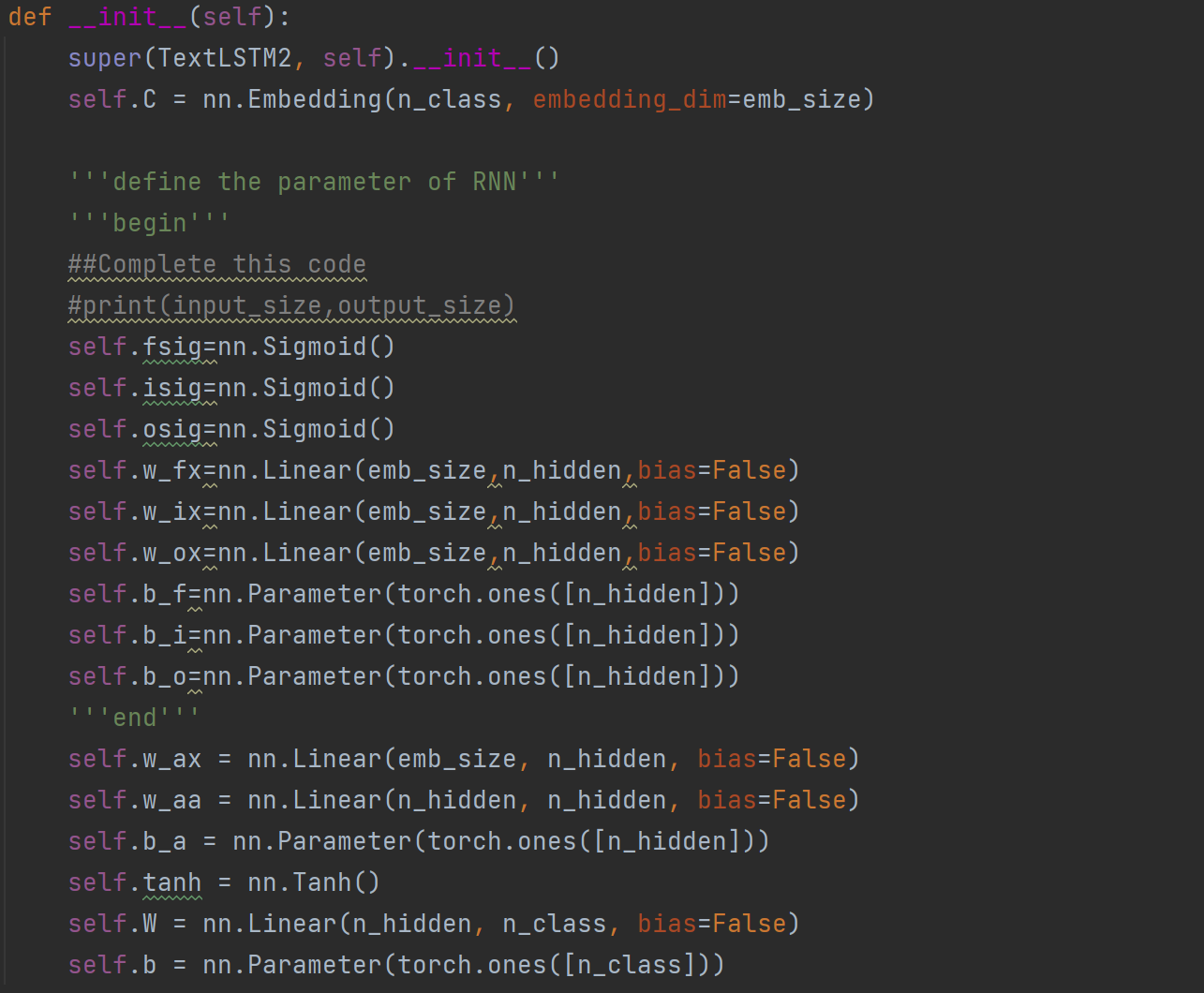
搭建LSTM模型的过程：

直接继承自nn.Module即可，再重写其中的构造函数和前向传递函数即可，如图分别重写了forward和\_\_init\_\_函数



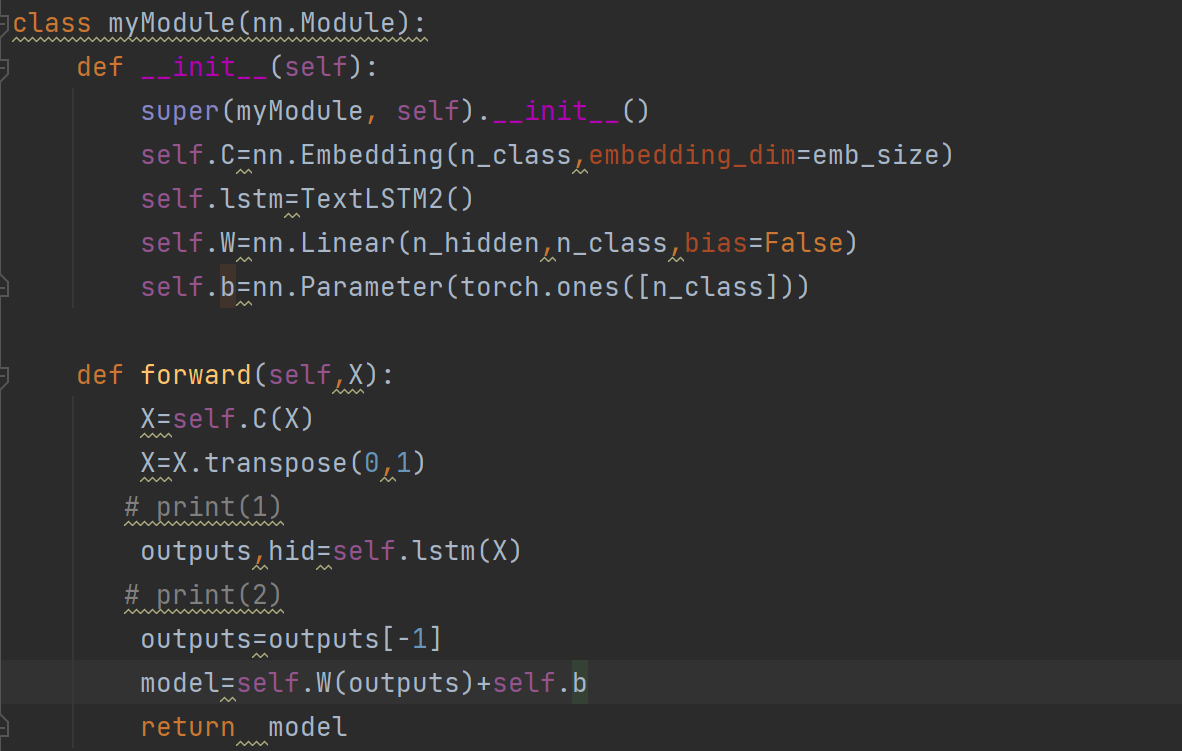
初始化一个全0的隐藏层和第一个记忆细胞。

返回值是所有的隐藏层model\_output和最后一个隐藏层h

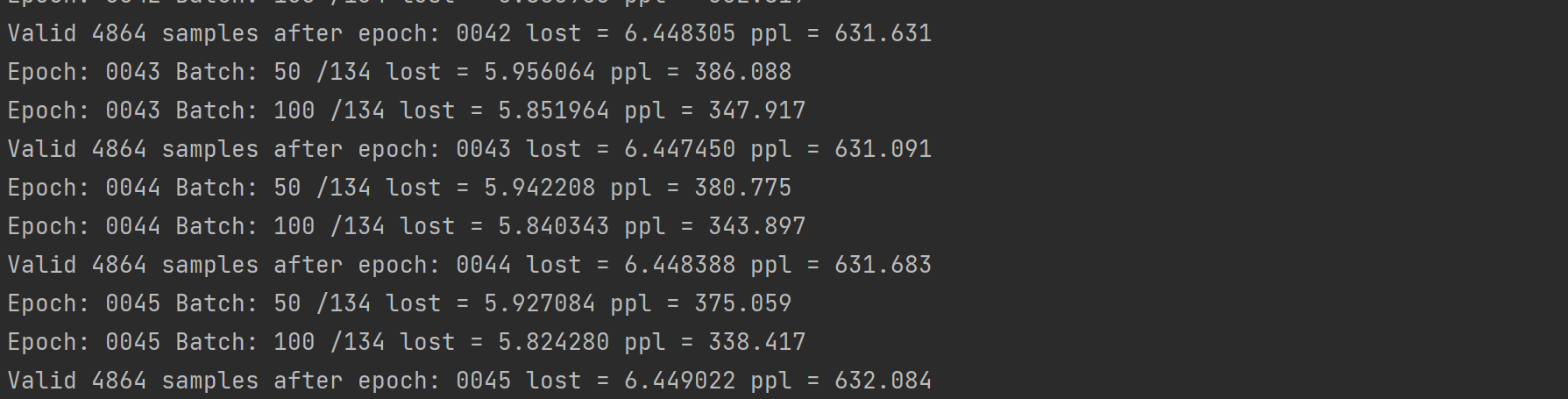


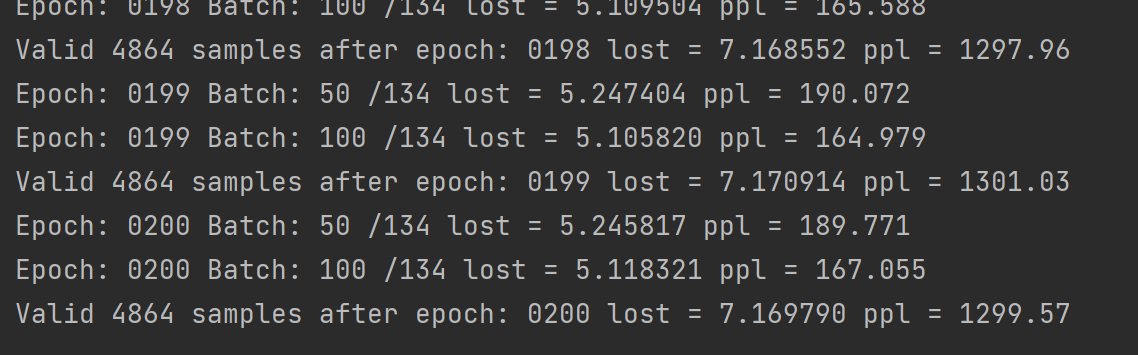
W\_fx,w\_ix,w\_ox分别是遗忘门、输入门和输出门的权重参数，b\_f,b\_i,b\_o是其对应的偏差；w\_ax是输入到隐藏层之间的全连接层，w\_aa是隐藏状态间的全连接成，b\_a是对应的偏差;W就是最后的连接输出的全连接层，b是其对应偏差。

再写一个整体训练的模型myMoudel:如图分别重写构造函数和前向传播函数，自定义一个nn.Embedding层这里只需要定义一个连接到输出层的全连接层即可，得到最后一个隐藏状态从而得到输出



最后的结果：





明显看出存在过拟合问题

收获：通过手写对RNN和LSTM的理解加深了，对ptorch框架的使用有了更深的了解，对自己的编程能力有了提升，对自然语言处理技术也有了一定认知。