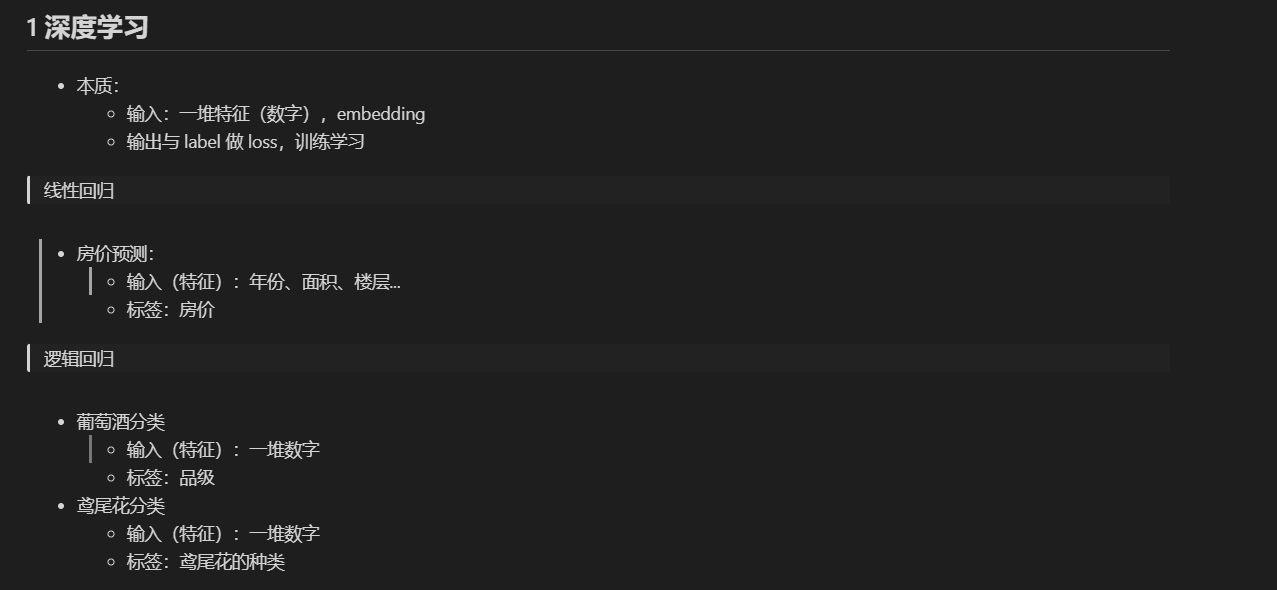
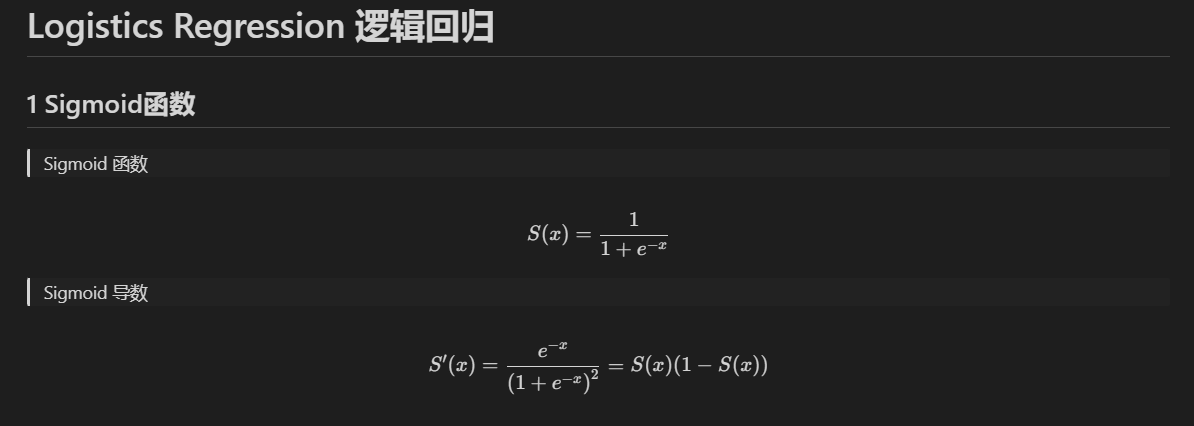
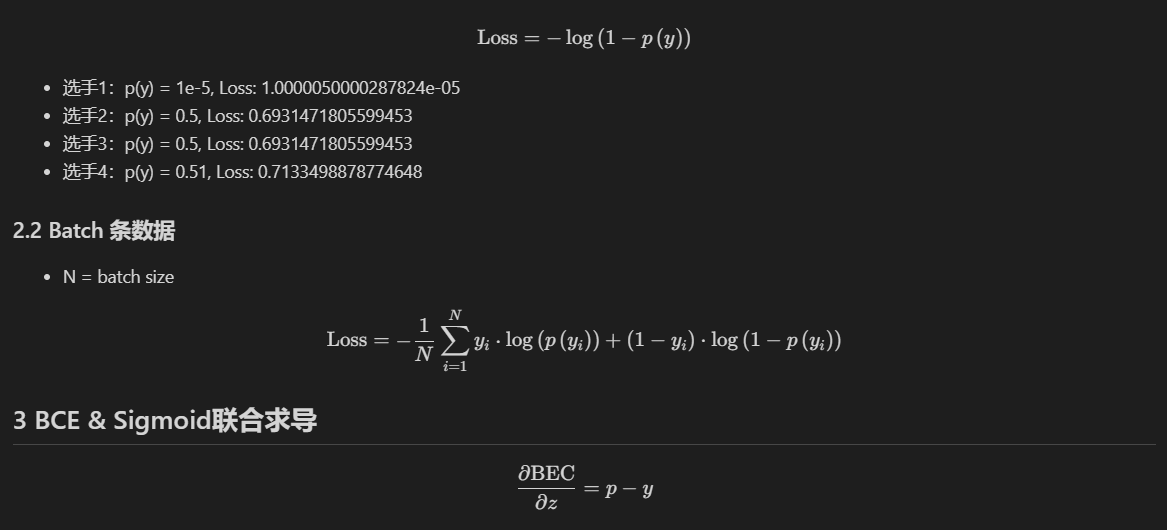
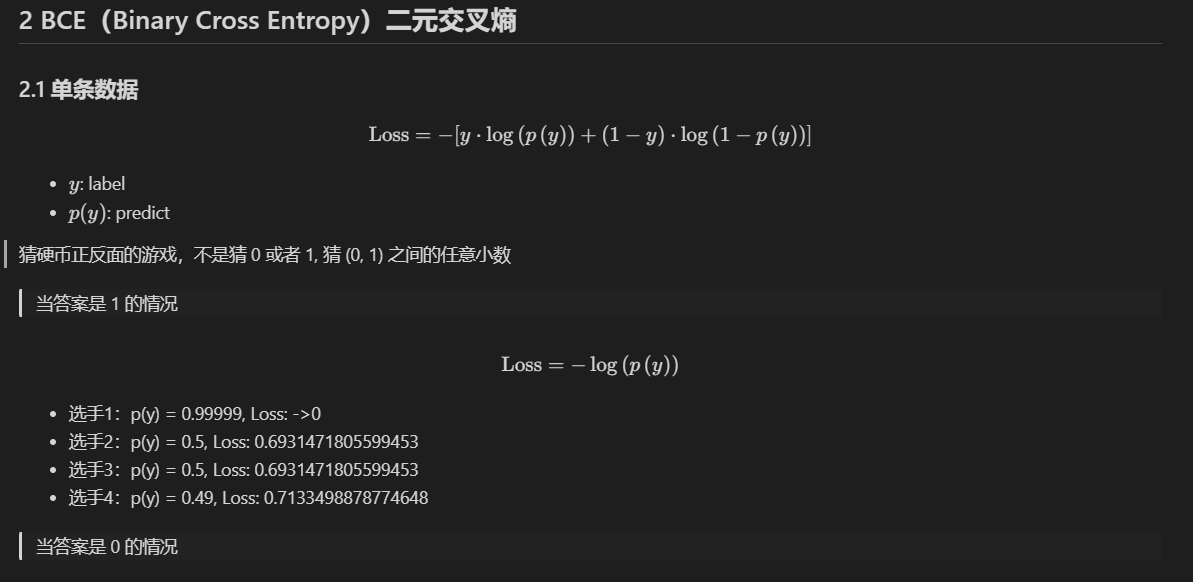
**第一阶段**：这一阶段主要是入门深度学习，了解深度学习的基本算法。这个是markdown笔记，文件存放在formula文件夹中，这里做个总结。



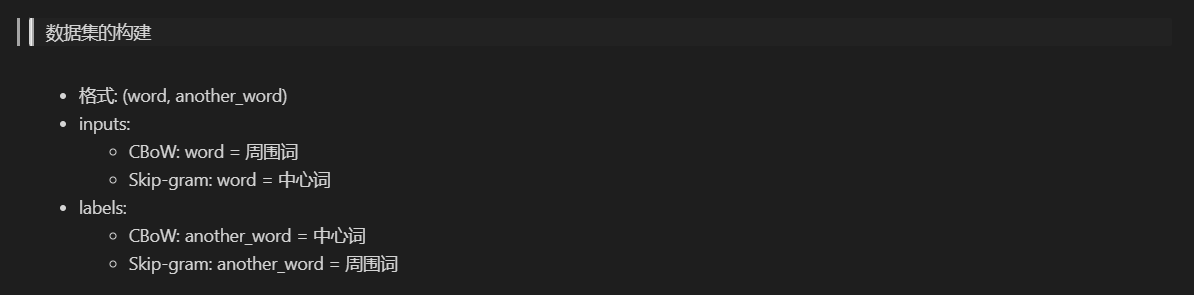


**pytorch框架组件公式**，源代码在”第一阶段学习-pytorch自主复现组件“





**Word2vec复习：**

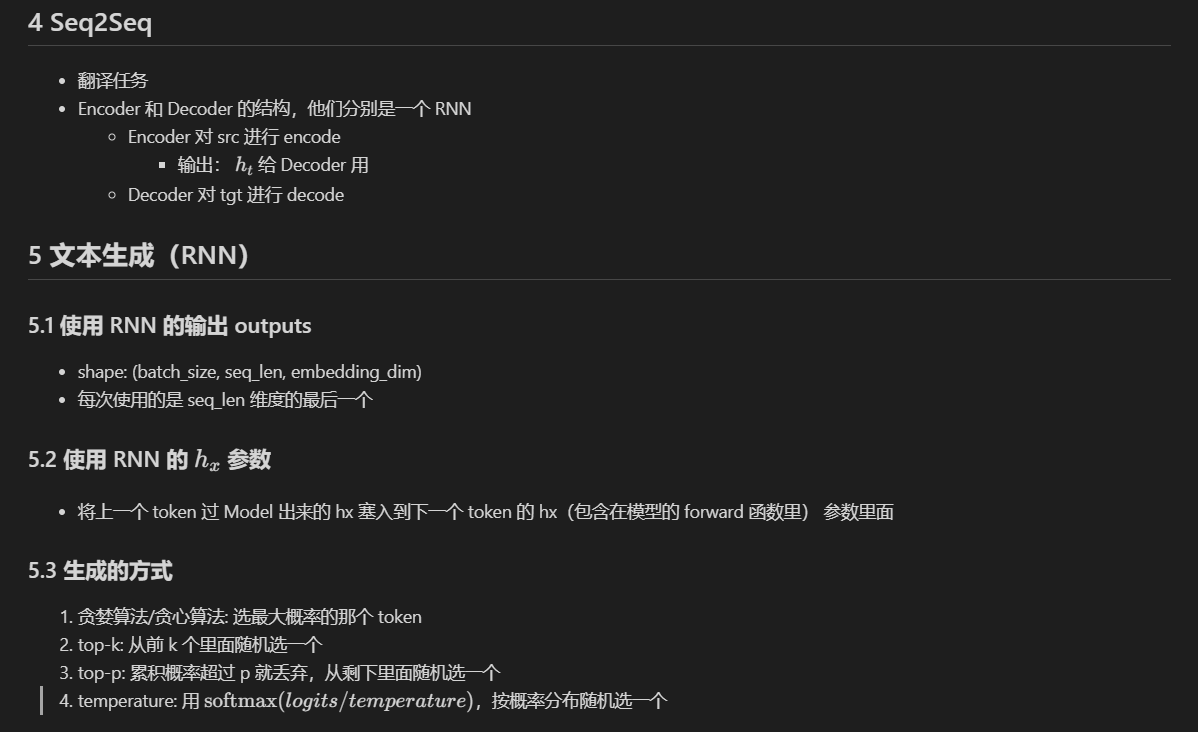
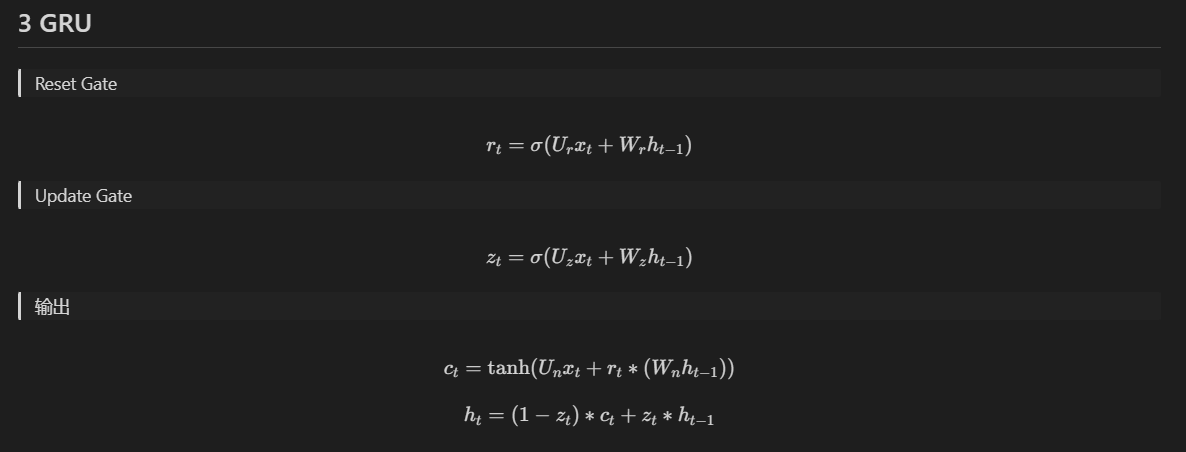
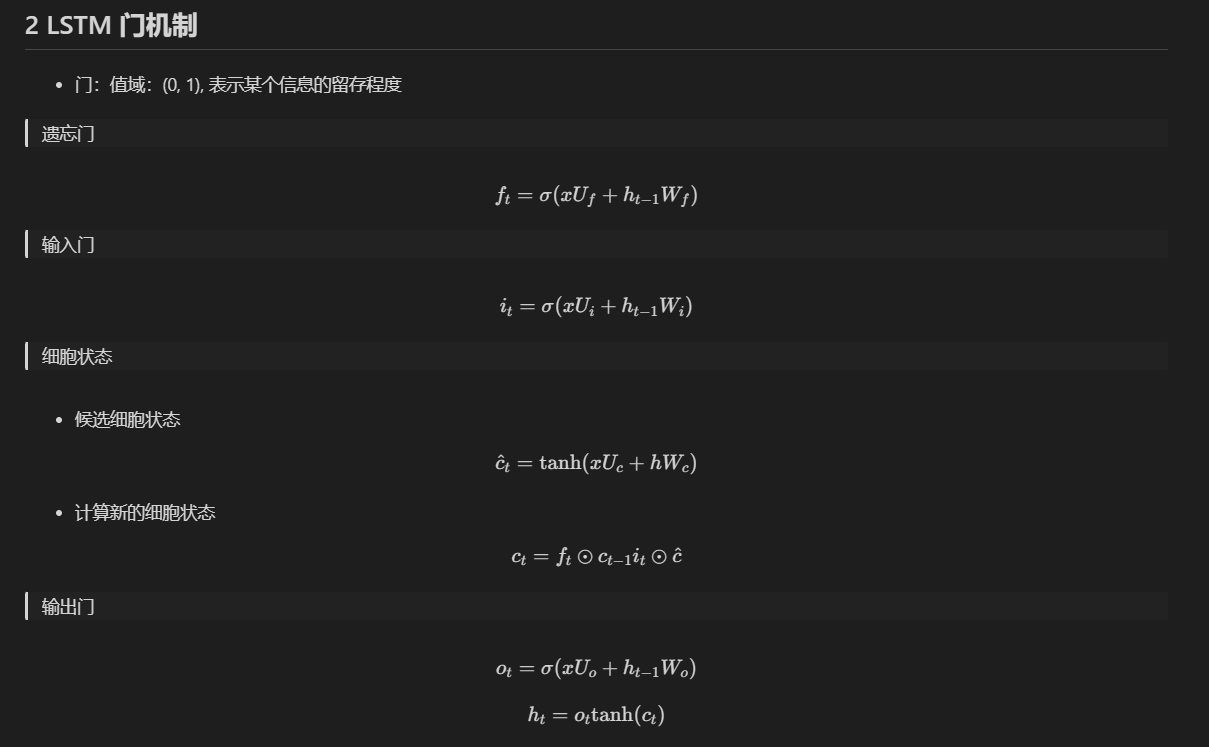
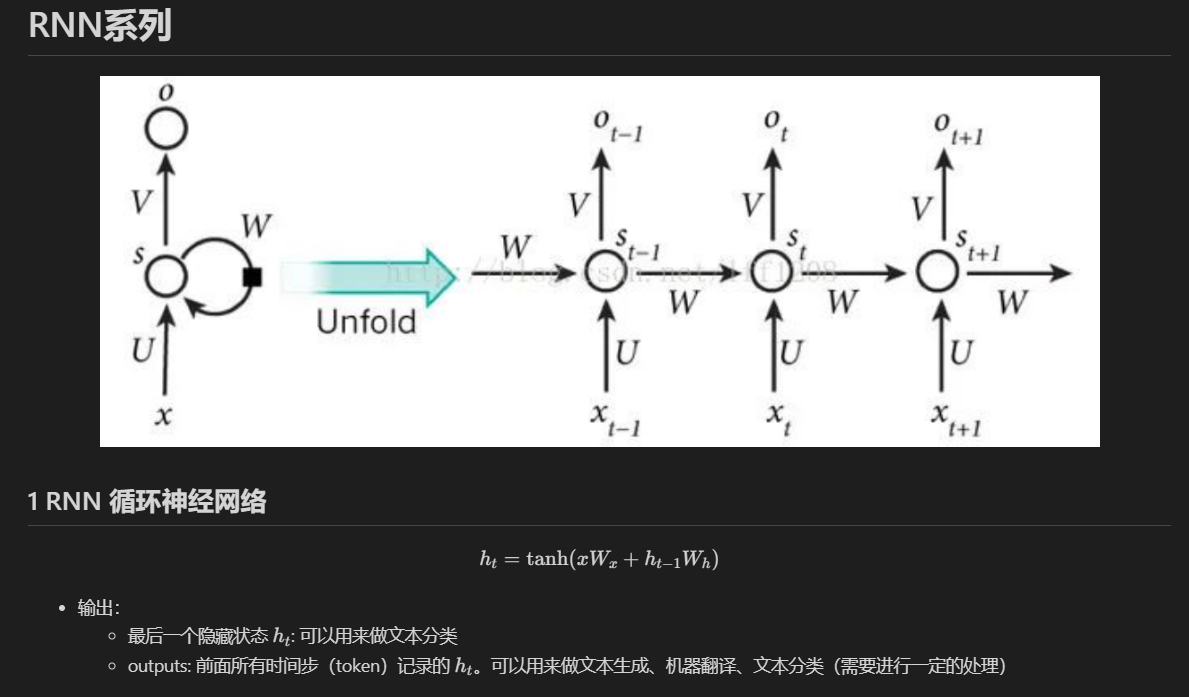


**第二阶段**：复现经典框架：RNN系列、Transformer、Bert、GPT

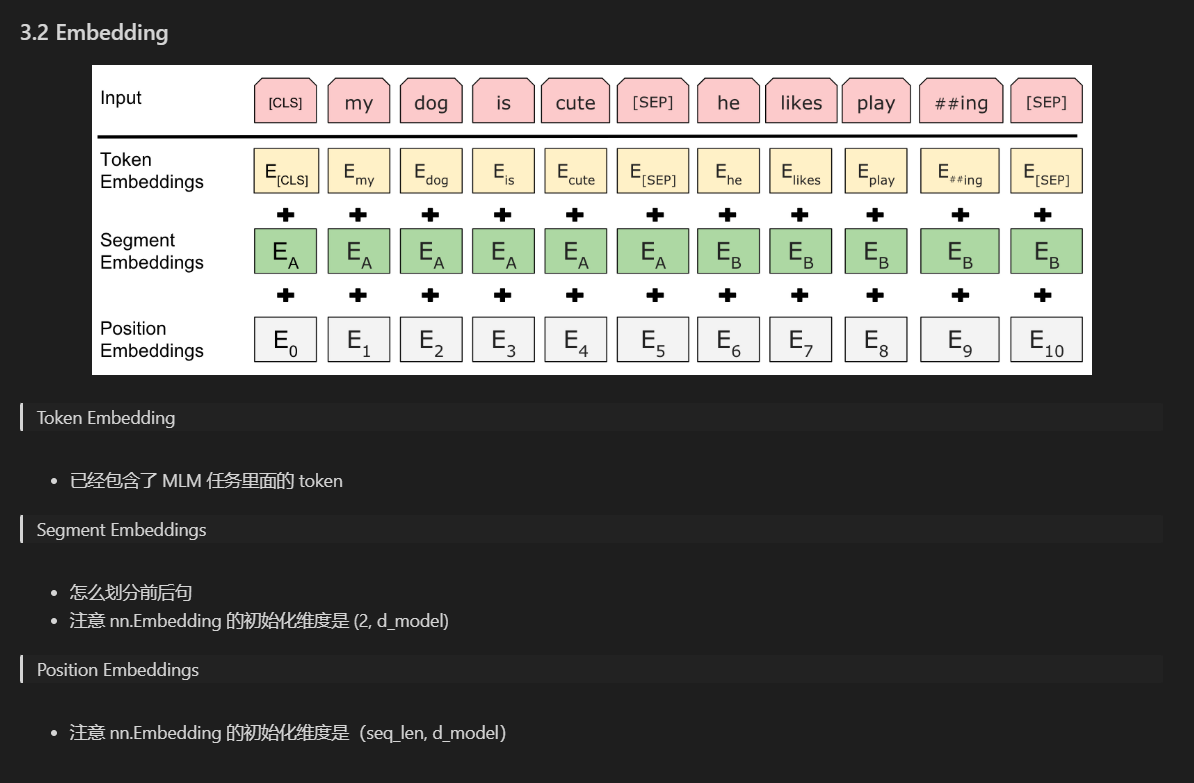
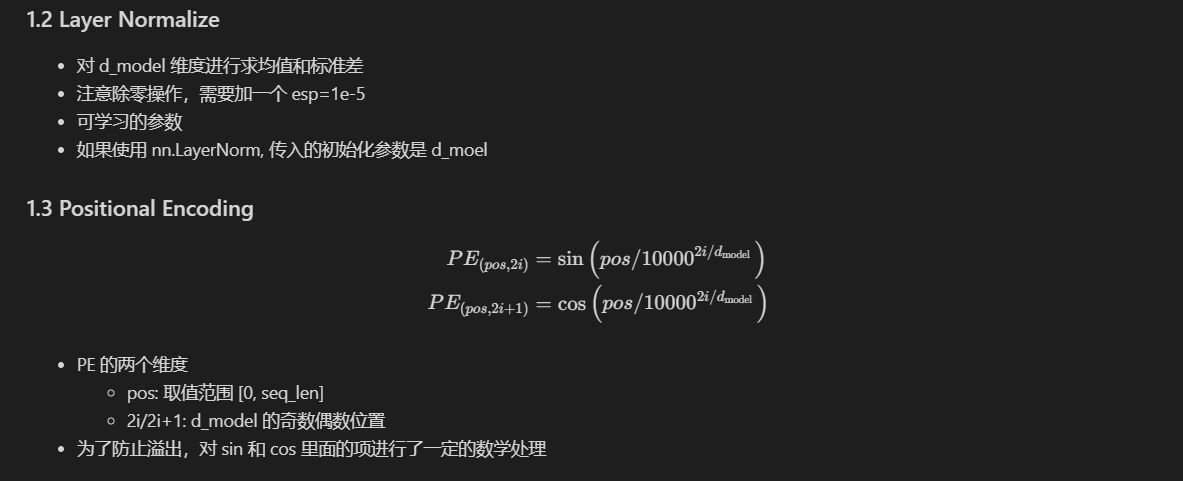
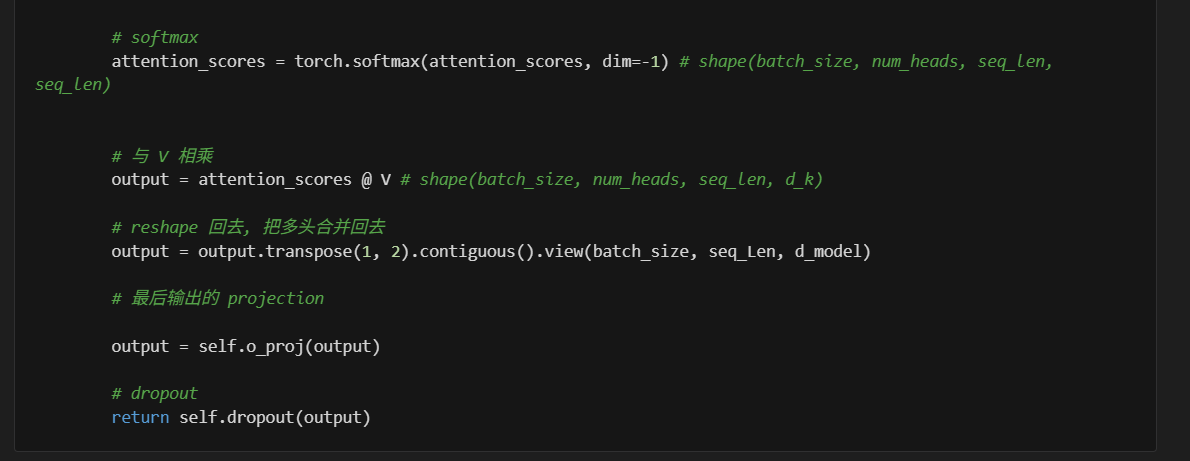
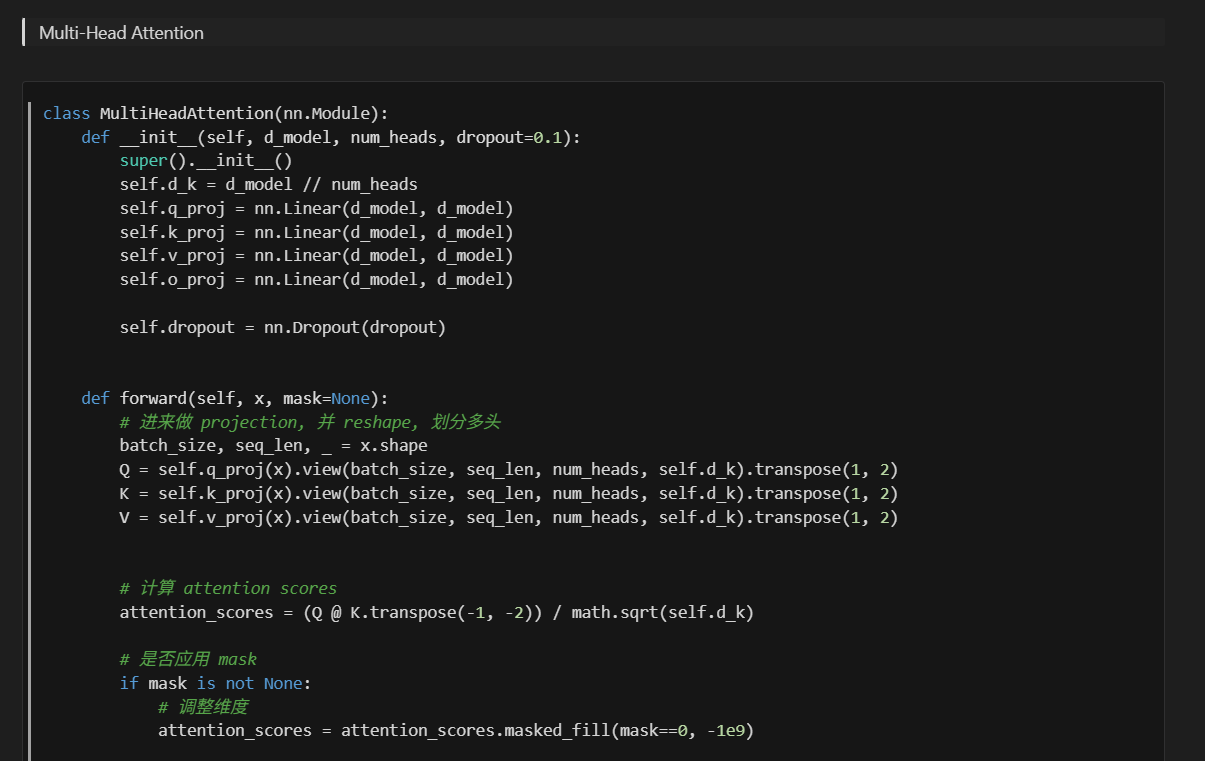
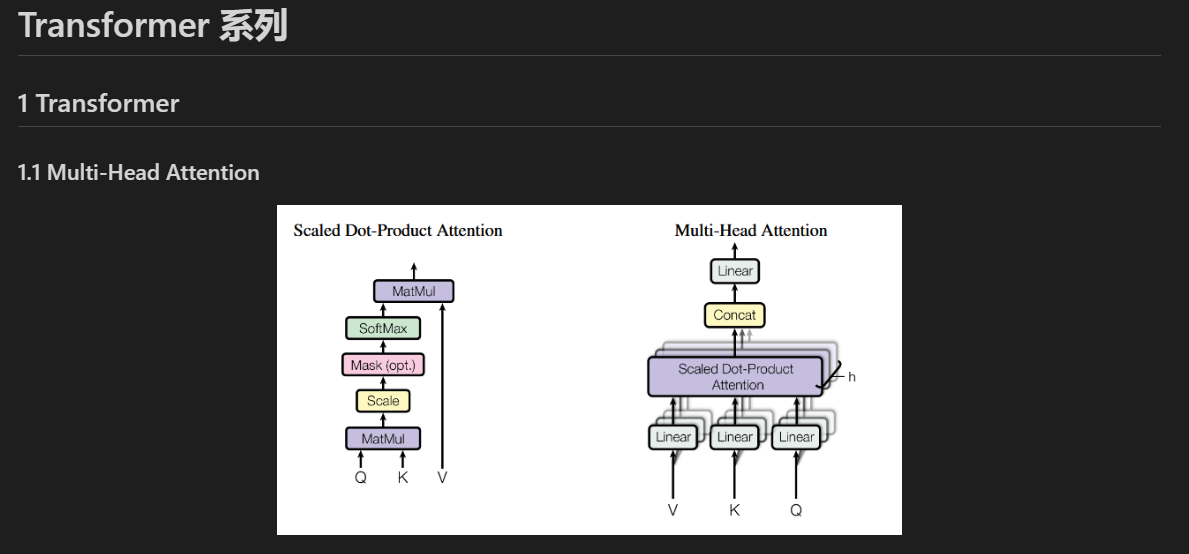
Tokenize复习：



**RNN, LSTM, GRU复习**：



**Transformer复现：**



**第三阶段：RAG美食项目**

在第三阶段学习了：大模型的微调，主要是意图识别和实体识别。然后就是整个RAG项目的搭建。源代码可查看美食RAG项目文件。

名称：基于RAG的厨房问答系统

1、数据集：data/recipe\_corpus\_full.json

一共大概150万数据

数据实例：

{"name": "西班牙金枪鱼沙拉", "dish": "金枪鱼沙拉", "description": "", "recipeIngredient": ["超市罐头装半盒金枪鱼(in spring water)", "2大片生菜", "5个圣女果", "半根黄瓜", "半个红柿椒", "半个紫洋葱", "1个七成熟水煮蛋", "适量红酒醋", "适量胡椒", "适量橄榄油"], "recipeInstructions": ["鸡蛋进水煮，七成熟捞出（依个人喜好），同时备其他菜", "生菜撕片，圣女果开半，黄瓜滚刀，红柿椒切丝，紫洋葱切丝，鸡蛋四均分", "金枪鱼去水", "撒黑胡椒，红酒醋和少许橄榄油", "拌匀，拍照，开动"], "author": "author\_67696", "keywords": ["西班牙金枪鱼沙拉的做法", "西班牙金枪鱼沙拉的家常做法", "西班牙金枪鱼沙拉的详细做法", "西班牙金枪鱼沙拉怎么做", "西班牙金枪鱼沙拉的最正宗做法", "沙拉"]}

2、milvus安装

这里用的是单节点的安装，这里租了一个服务器（Linux系统）来部署milvus数据库，阿里云服务器（轻量应用级服务器），由于阿里云的防火墙优先级高于服务器本身的system，所以要在阿里云防火墙设置可出入的id，这里我是通过部署Docker，在Docker里面拉milvus镜像，这里记得换镜像，否则拉取镜像不成功。可用镜像：Daocloud 镜像加速器地址：https://docker.m.daocloud.io

3、milvus数据保存

这里我将每个菜的name向量化存在milvus中，同时每个都有其对应的id，由于milvus是向量数据库，能够快速的检索，同时其本身具有语义理解能力，所以检索的准确率挺高的。这里也是封装了一个类来管理milvus数据库。但是其安全性和稳定性都不理想，所以这里另外使用两个数据库来共同存储数据。

4、neo4j保存数据

这里需要将数据处理一下，主要是存储了两种节点，一种是菜谱，一种是食材，由于原始数据集是没有将基本的食材提取出来的，所以这里选用微调过后的实体识别模型，将

recipeIngredient里的实体识别出来。然后再建立菜谱和食材的关系。

在neo4j使用中有个文件neo4j-oop.py里面封装了Neo4jManager类，实现了增删改查基本操作，都是cyper语言，这样速度会快一点。

5、gradio界面

这个是一个简单的问答界面，一键启动就可以了。

6、意图识别

对用户提出的问题进行意图识别，这里有数据的增广，一共是4个意图，分别是”制作方法“、”材料制作成品“、”询问配料“、”询问厨师“，标签分别对应0，1，2，3，如果不是这四种意图就返回”其他意图“。这里同样也是进行模型的微调，只是这里自己写得一个框架，使用的预训练模型是“bge-base-zh-v1.5”

7、实体识别

对用户提出的问题进行实体识别，这里其实用调用了大模型接口来进行实体识别的，也可以进行模型的微调。但是数据集没有找到，所以就没有做，如果有数据集就可以全量微调模型。

8、项目整体代码

前面的模块都是前期的准备，实际上项目运行所需的代码在”项目总结“文件夹中，只需运行这里的文件就行了，然后启动前端gradio和后端flask，即可进行问答，前提是数据库要开起来。

总之，系统的学习NLP知识之后，对于大模型的认知不仅仅是局限于copy人家的代码，我可以根据自己的需求改动模型框架，从而完成任务。未来还有很多需要学习，继续保持学习的心态，脚踏实地地努力吧！

**现有流行的工具（自己的泛化了解）**

还可以用VLLM来进行部署，但是这里最好用GPU，也可以用llamafactory来进行模型微调，都是工具，不过这些框架都进行了优化，是一种很好的工具，接受新工具。还有很多是使用工具流来运行，比如RAGflow，这个挺火的，但是这个有个缺点就是不能自定义，但是优点就是能够快速地部署好RAG，而且一些还提供了很好的可视化工具。