大模型推理系统

一、项目

基于作业阶段完成的项目，继续实现以下功能：

1. 自注意力

目标：主要完成 model.rs 中 self_attention 方法。

在得到 Q、K、V 三个张量后，Q 形状为（seq_len,q_head*dim），而 K、V 在连接 kvcache 后形状为（total_seq_len，k_head*dim），其中 seq_len 为输入序列长度，total_seq_len 为输入序列和 kvcache 总长度。我们需要的是 Q 和 K 矩阵乘后，对 seq_len 中每一个 token 独立的头各自得到一个（seq_len,total_seq_len）的权重矩阵，目前存在两个问题：

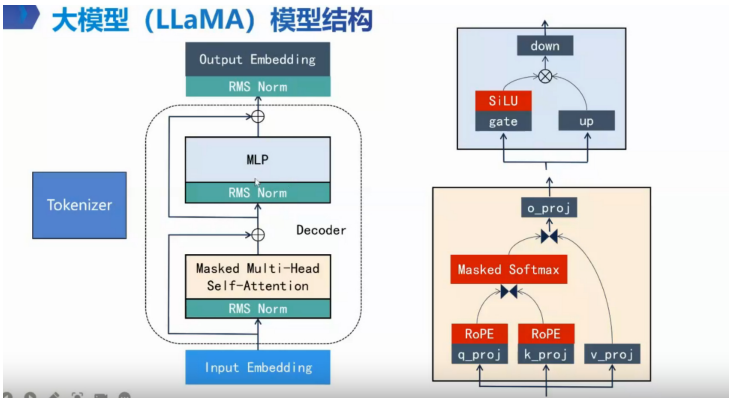
a.Q 的头数和 KV 头数不一定一一匹配，一般来说 Q 的头数是 KV 头数的整数倍；

b. 要得到（seq_len,total_seq_len）的权重矩阵，就需要将 (seq_len, dim) 和 (dim, total_seq_len) 的两个矩阵做矩阵乘才能得到，但 Q、K、V 都不是我们需要的形状。

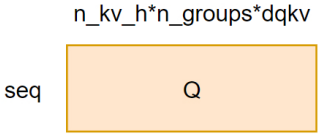
有两种解决方法：我们可以考虑对这些矩阵 reshape+ 转置 + 广播 + 矩阵乘法，或者计算对应关系索引 + 向量乘法；这里选择使用第二种方法完成，现在我们先看看对每个头的自注意力计算过程：

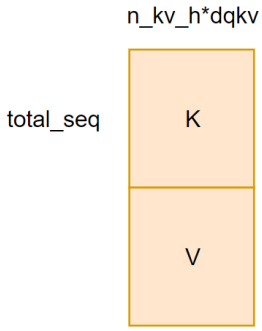
```
x = rms_norm(residual)
Q = RoPE(x @ Q_weight.T)
K = RoPE(x @ K_weight.T)
V = x @ V_weight.T
K = cat(K_cache, K)
V = cat(V_cache, V)
### 以下是你需要实现的部分
score = Q @ K.T / sqrt(dim)
attn = softmax(score)
attn_V = attn @ V
### 到这里就行
out = attn_V @ O_weight.T
residual = out + residualhost_accessor C { c, read_only };
```

其中前六步已经在 forward 中计算完毕，最后两步在 forward 中用矩阵乘法就可以实现，我们只需完成中间三步即可。回到 PPT 中的模型结构图：



我们要做的就是右边的步骤，这里可以将其中几个关键矩阵绘制出来方便理解：





为实现 Q、K 的矩阵乘，我们可以双重循环遍历头数和倍数以匹配 Q 和 KV 头数，按 group 分组计算每组 Q 的起始位置、K 的起始位置，从而计算向量点积的值 score，再计算存储中间值矩阵的索引，存储数据，最后调用 softmax 函数保存中间变量即可；同理，最后一步的中间值矩阵与 V 矩阵乘法也是计算输出矩阵对应索引，矩阵 v 对应索引，中间值对应索引，再计算对应向量乘积即可。

2. 文本生成

目标：实现 model.rs 中文本生成方法 generate。

具体来说，我们需要初始化可以复用的 kvcache，在多轮推理循环中调用 forward 方法，传入输入和 cache，得到输出，将当前的输出作为下一次的输入（这里使用了 Operator 中的采样方法，即选择输出中概率最大的那个 token）；同时，根据最大生成 token 数、结束符判断是否需要停止推理，最后返回所有推理结果：

```
Once upon a time
Once upon a time, a little girl named Lily lived in a small house with her mom, dad, and her dog, Spot, Spot, loved to play all day. One day, Lily saw a small bird on the ground. She picked it up and tried to reach the bird and see what it was.
```

3.AI 对话

目标：仿照文本生成的功能，写一个实现 AI 对话的 chat 函数。

要完成该功能，记得先去去下载model 的 chat。将 chat 放在 model 目录下，因为 chat

和 story 模型参数有区别，记得修改 params.rs 中读入模型参数的代码。

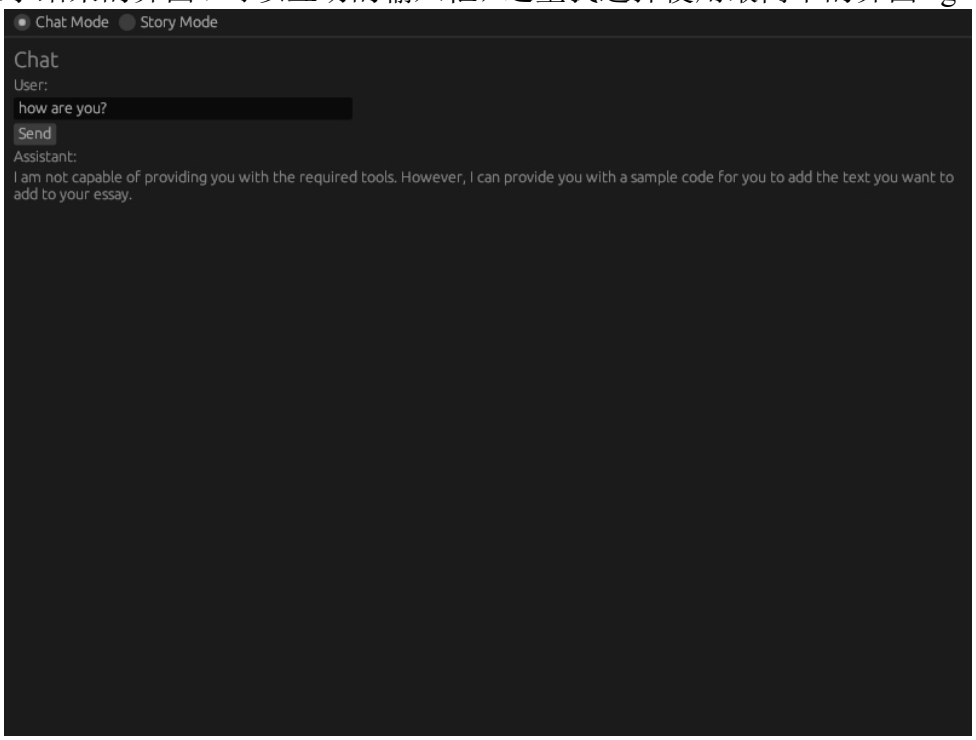
我们可以直接使用前面文本生成方法 generate，将其中 cache 替换为外部传入的参数，再用一个循环接收用户输入即可。这里使用了 jinja 模板进行对话，效果比直接输入要好得多：

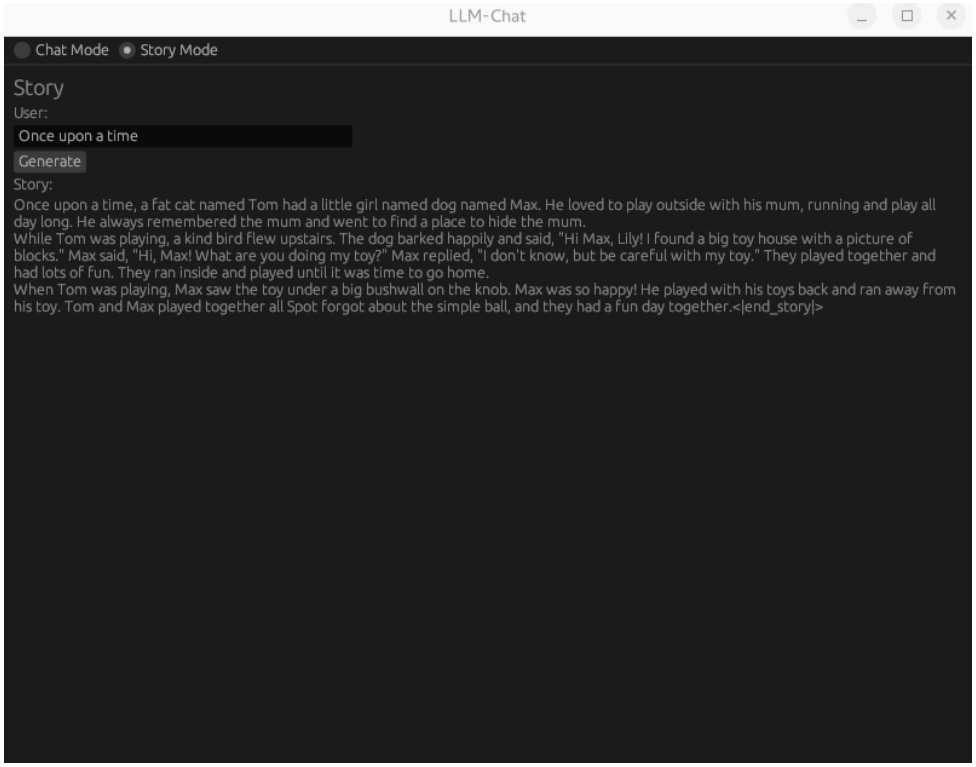
```
     Running `target/debug/learning-lm-rust`

 Choose your function:
 1: chat;
 2: story;

1
Starting chat session (type 'exit' to end):

User (turn 1):
how is the weather?

Input (<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
how is the weather?
<|im_end|>
<|im_start|>assistant
):
Assistant_result: Certainly! The weather is always going to happen and the temperature of the sun plays a crucial ro
le in the overall climate. Some factors can affect the weather and affect the overall weather.

1. Dusting: Dusting can cause heat to melt over time. It's important to ensure that the weather is always hot, but i
t can also help to make it more cold. The temperature of the sun affects the atmosphere and humidity, so it's import
ant to choose the right weather option.

2. Geothermal heat: Some types of heat can occur due to heat transfer temperatures, temperature, and heat temperatur
e. Some heat conditions can affect the heat or make the heat more cold.

3. Dust temperature: Dust temperatures can affect the temperature of the sun, so it's important to choose a temperat
```

二、进阶

1. ui 界面

目前我们已经完成了续写故事和对话两个功能，在 ui 界面需要有可以选择功能的按钮、显示结果的界面、可以互动的输入框，这里我选择使用最简单的界面 egui 来完成：





三、未来

多会话管理、历史回滚、混合精度、多线程都有思路，可以后面加入；

最近 RAG 是热门，可以考虑接入该功能，再对比看看效果好坏。

（最近突然变忙人，以后有时间再继续完善看看吧 orz