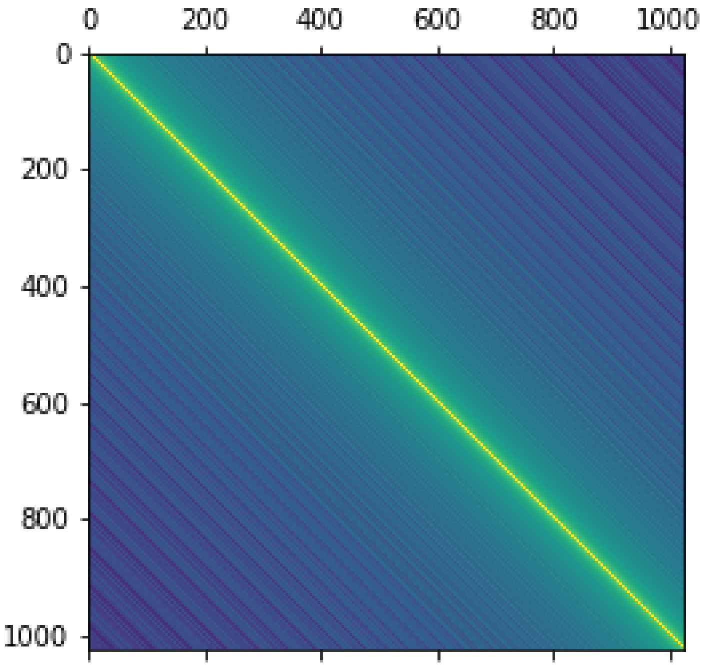
Gradescope (4 pts)

t11902210 張一凡

1. **Visualize the similarity between different pairs of positional embedding and briefly explain the result. Additionally, attach the code that you used for visualization**

Answer:



The method I use is to use the cosine similarity that comes with Pytorch to assist with the broadcast application of Torch Tensor to obtain a similarity matrix. Finally, use plt.match to draw the matrix (see the following for specific code). From the above image, it can be seen that the similarity on the diagonal is the strongest. In this image, the diagonal seeks its own similarity and is all 1. At the same time, the similarity is symmetric along the diagonal and the positional similarity at the same distance from the diagonal is the same (translation invariance). This image shows that for different embedded pairs, the farther apart they are, the lower their similarity; On the contrary, the closer the positions between different translated words and sentences are, the higher their similarity is, which is very helpful for screening and evaluating the translation results of words and sentences.

Below is the visualization code I use.

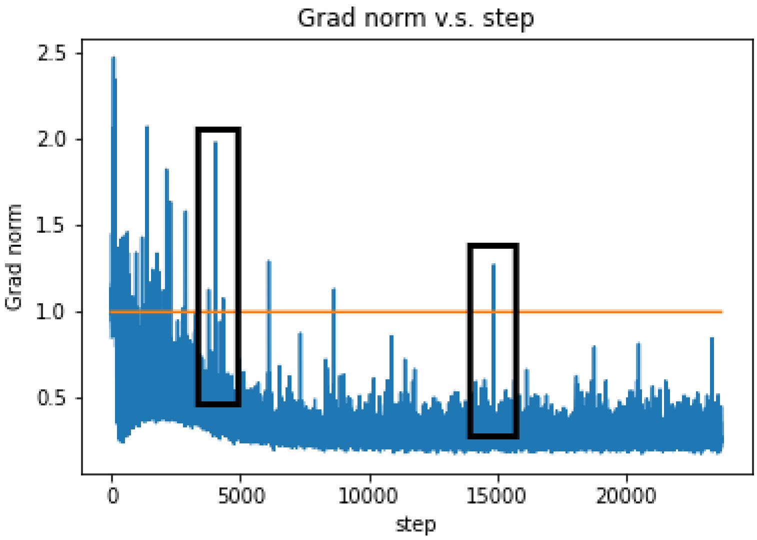
```
from torch.nn.functional import cosine_similarity as cs
pos_emb = model.decoder.embed_positions.weights.cpu().detach()
```

```
print('size of pos_emb',pos_emb.size())
ret = cs(pos_emb.unsqueeze(1),pos_emb,dim=2)
plt.figure(figsize=(8,8))
plt.matshow(ret)
plt.savefig("visualization.png")
plt.show()
```

2. Clip gradient norm and visualize the changes of gradient norm in different steps. Circle two places with gradient explosion

Answer:

Before the function train_one_epoch, I defined a list of global variables, gnorms, and placed the value of gnorm inside the function into gnorms. After the training is completed, I will visualize gnorms. The intersection between gnorms and the line y=1.0 is where clipping takes effect.



The two places I circled in the box are two typical gradient explosions. Below is the code I use.

```
plt.plot(range(1, len(gnorms)+1), gnorms)
plt.plot(range(1, len(gnorms)+1), [config.clip_norm]*len(gnorms),
 "-")
plt.title('Grad norm v.s. step')
plt.xlabel("step")
plt.ylabel("Grad norm")
plt.savefig("grad.png")
plt.show()
```

```
Reference link:
https://blog.csdn.net/weixin_42369818/article/details/124102920
```