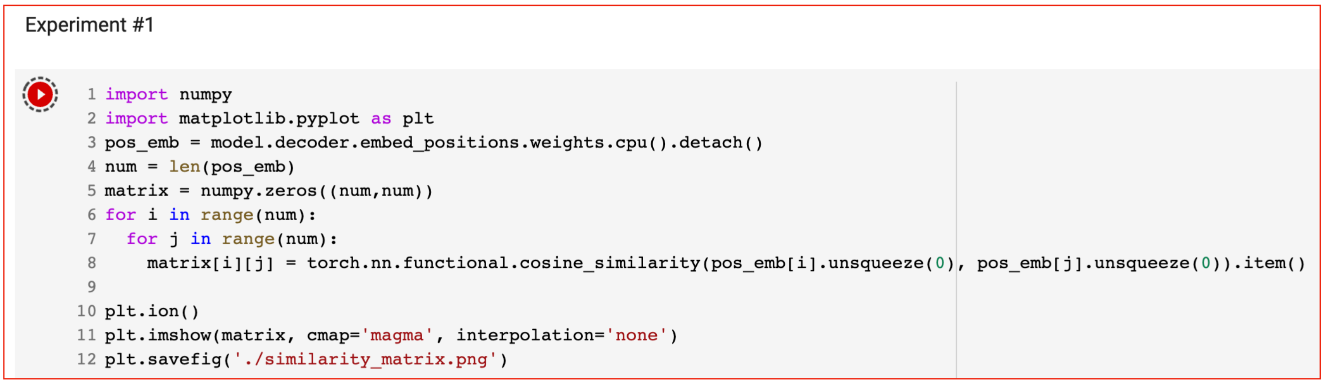
Problem 1

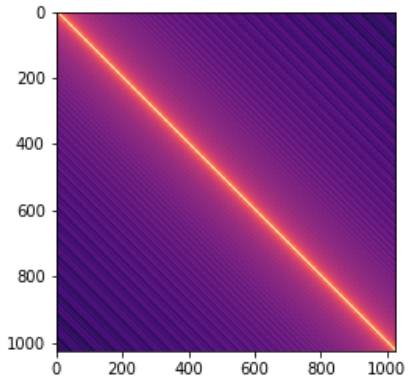
○ Visualize the similarity between different pairs of positional embedding and briefly explain the result.



Experiment #1

```
import numpy
import matplotlib.pyplot as plt
pos_emb = model.decoder.embed_positions.weights.cpu().detach()
num = len(pos_emb)
matrix = numpy.zeros((num,num))
for i in range(num):
  for j in range(num):
    matrix[i][j] = torch.nn.functional.cosine_similarity(pos_emb[i].unsqueeze(0), pos_emb[j].unsqueeze(0)).item()

plt.ion()
plt.imshow(matrix, cmap='magma', interpolation='none')
plt.savefig('./similarity_matrix.png')
```



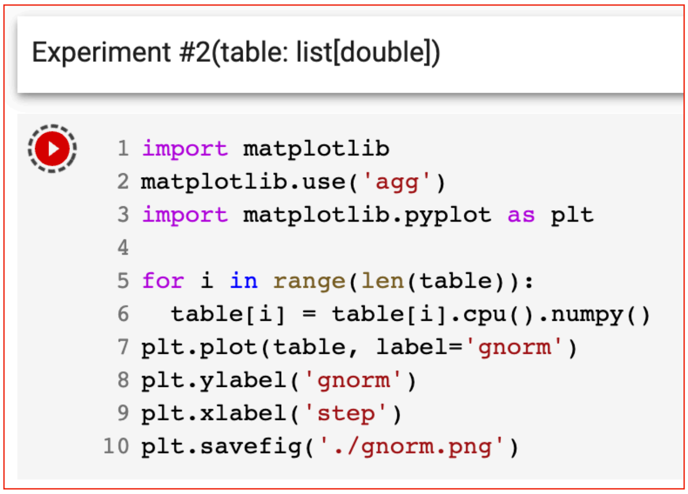
○ Briefly explain the result:
Position embedding is used to encode the information of position. Moreover, the position difference is represented by the vectors index difference.

As we know, cosine similarity can highly show the similarity of vectors; when we move away from the diagonal line (x-direction or y-direction), the vector index and position difference increase. Imply that the similarity decreases. So, we get the darker color region!

On the other hand, the diagonal line region means cosine similarity of itself or its neighbor. This fact means high similarity exists in this region. Hence, this region is a lighter color!

Problem 2

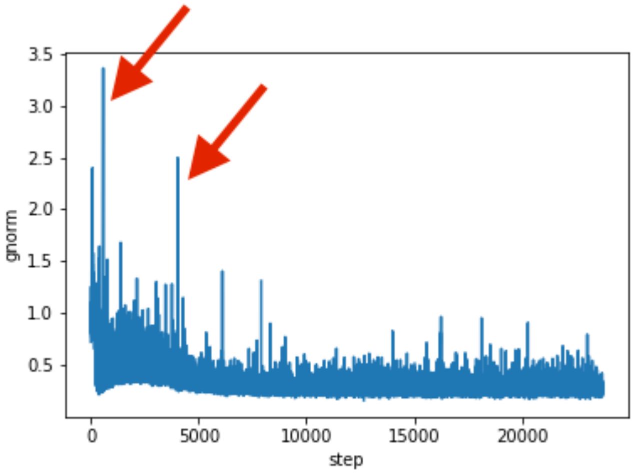
○ Clip gradient norm and visualize the changes of gradient norm in different steps. Circle two places with gradient explosion



Experiment #2(table: list[double])

```
import matplotlib
matplotlib.use('agg')
import matplotlib.pyplot as plt

for i in range(len(table)):
  table[i] = table[i].cpu().numpy()
plt.plot(table, label='gnorm')
plt.ylabel('gnorm')
plt.xlabel('step')
plt.savefig('./gnorm.png')
```



On the right-hand side, it is the gnorm-step figure. Furthermore, at least two times gradient exploration happened in-between step=0:5000, as the arrow pointing.