SI630 HW2 Report

Chongdan Pan

February 18, 2022

1 Batch Size

In my computer, it turns out that larger batch size will lead to less training time. However, it also will uses a lot of memory, and I need to turn on the swap space in case the kernel crash due to the memory limit. On the other hand, the use of swap space may also lead to a longer training time.

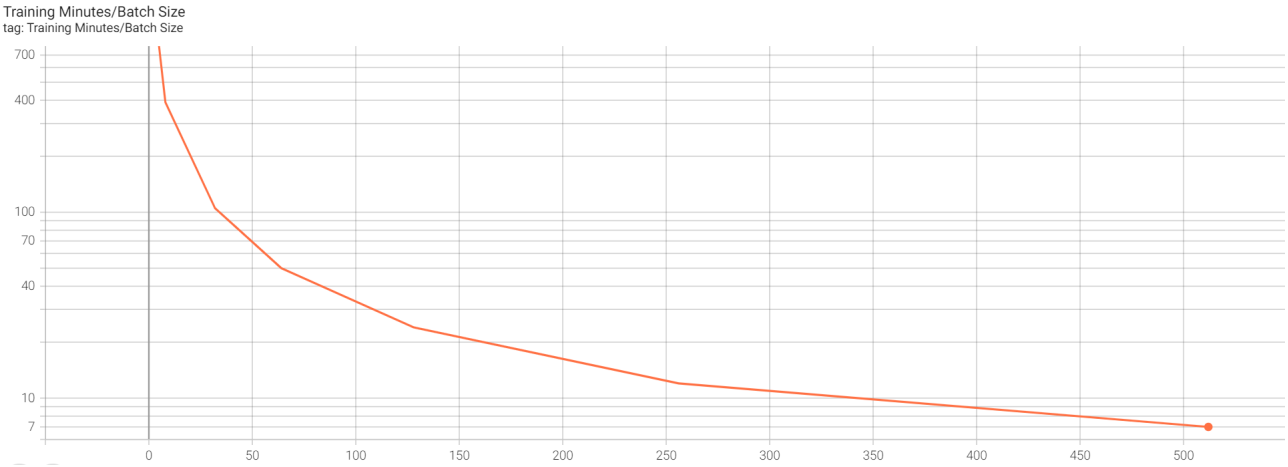

Figure 1: Batch size for training time

2 Loss behavior

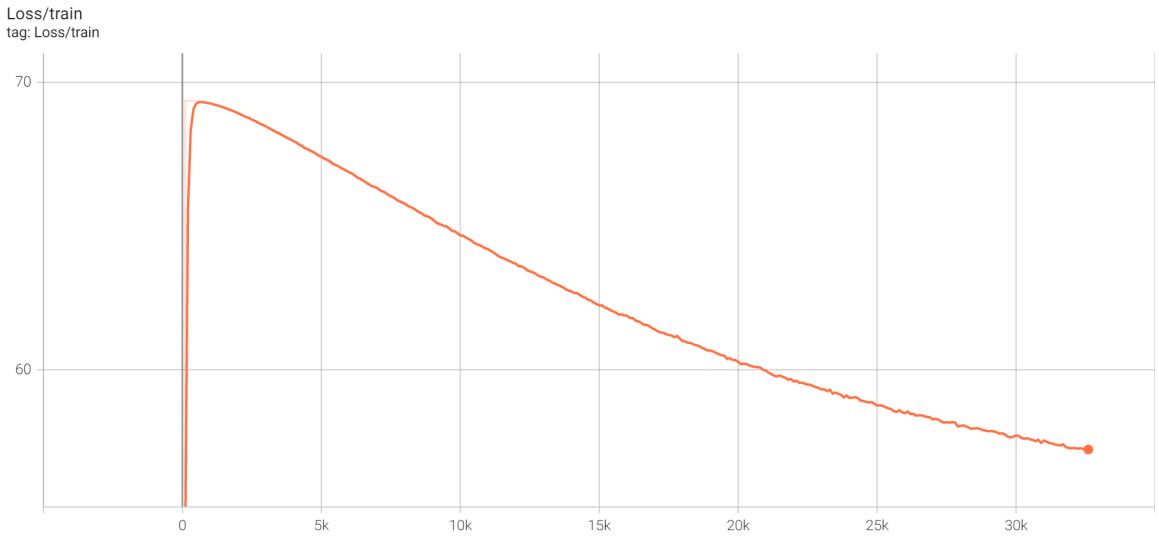

Figure 2: Loss Function for training

3 Neighbours

	calculator	president	vehicle	language	carefully	january	various	walk	medicine	girl
0	computer	chairwoman	solution	nomenclature	accessible	july	versatile	shoot	medication	daughter
1	comparative	chairperson	generous	terminology	conflicting	february	programs	paseo	harvard	son
2	clinical	secretary	potency	feminist	structured	september	companies	putt	mathematics	boy
3	mathematical	chairman	problem	culture	fragments	august	numerous	inject	physics	elder
4	chemical	vice	adequate	folklore	hardships	june	legion	pitch	economics	sister
5	astronomy	chief	evaluate	educational	appreciating	october	projects	clause	philosophy	younger
6	mechanical	deputy	requisite	manufacture	sensory	november	newspapers	bite	maths	firstborn
7	electrical	frailty	factual	ancient	cutting	december	magazines	tar	chemistry	sis
8	communicating	minister	hire	psychological	rigorous	march	groups	walking	yale	widow

Figure 3: Close neighbours for target words

It turns out that the word embedding is working well, especially for the frequent nouns, such as "January", "girl", or "president". It's very interesting that "president" has a higher cosine similarity to "chairwoman" than "chairman", perhaps it's due to the bias term in the loss function. However, the result for verb like "walk" is not so intuitive.

4 Analogies

```
In [22]: vec_queen = eva.get_vec("king") - eva.get_vec("man") + eva.get_vec("woman")
         eva.compute_cosine(vec_queen, eva.get_vec("queen"))
Out[22]: 0.8872133493423462

In [32]: vec_artist = eva.get_vec("scientist") - eva.get_vec("science") + eva.get_vec("art")
         eva.compute_cosine(vec_artist, eva.get_vec("artist"))
Out[32]: 0.7452678680419922

In [33]: vec_walk = eva.get_vec("run") - eva.get_vec("fast") + eva.get_vec("slow")
         eva.compute_cosine(vec_walk, eva.get_vec("walk"))
Out[33]: 0.8028973340988159
```

Figure 4: Some interesting analogies

It looks like the analogies also exists for professions, such as scientist - science + art, can lead to an artist. The similar thing also happen with verb, such as we can remove the fast from run, and get walk.