HYPER-PARAMETERS EXPLORED AND OBSERVATIONS FOLLOWING IT:

1. The accuracy that differs by few decimal percentage have not be considered as big changes for this analysis.

2. **Context window**: In both cases, we observe that window size greater than 8 gives the best results (marked by row numbers 5 and 7 respectively). I couldn't run the code for larger context windows like 64 as it was running forever, however, I hypothesise that the accuracy would start decreasing as we try to create a mapping/relation between two words which can be separated by more than 20 words! In all practical senses the possibility that a word is contextual to another word that occurs after 20 positions is very highly unlikely to be close to each other. **Smaller context windows (rows 6 and 2 resp.) perform consistently in both cases however, I wouldn't want to use it since it might not capture the relation of the words that always occur at relative positions that is slightly greater than the window size.**

3. **max_num_steps:** The number of iterations for training shows visible increase in overall accuracy in table 2 (rows 3 and & 9 and 10). However, there is not a visible 'pattern' observed in cross entropy. I had however expected it to increase with increasing number of steps and then flatten out at a threshold (since training on the same data beyond a point would only overfit).

4. **Batch Size:** The increasing batch size initially increases the performance and then starts slacking off. *From table 2,* for batch_size that increases from 32 -> 64 -> 128, the accuracy increases and then decreased for batch_size of 256. But *Table 1,* doesn't show a visible pattern on this front.

5. **Sparsity of num_skips wrt to context windows:** To my surprise, when the num_skips (number of words taken out of the window) is smaller than the window size, it produced better results overall. (rows 4, 5 in *Table 1* and rows 7, 11 in *Table 2*). My implementation first gathers the window of words for each centre word and then randomly samples 'num_skips' words from that to be its context. Hence this was a case when random sampling worked for the best.

6. **Num_samples:** The num_samples is the number of negative samples generated. I hypothesised that increasing num_samples will increase the accuracy initially and then it would start decreasing it, since,

    1. More words that sampled in random to be negative words, might start picking more than a few 'contextual' ones as well.

    2. the model would start tying together the (wc,wo) pairs more closely, since it tries to distance all the other negative words far away. This would in one way result to overfitting!

This observation can be seen by comparing rows 8, 9 and 10 in *table 2.*

RESULTS ON ANALOGY TASK:

Cross Entropy:

Batch_size	num_skips	skip_windows	Max num steps	Least Illustrative	Most Illustrative	Overall Accuracy	Final Loss
64	2	2	50001	32.7	35.1	33.9	4.0801
64	16	8	100001	32.3	34.9	33.6	4.1555
128	2	2	50001	32.9	35	34	4.7301
128	2	4	50001	32.7	35.6	34.1	4.8384
128	4	4	50001	32.4	35.1	33.8	4.8441
128	8	4	50001	32.2	34.9	33.5	4.83809
128	8	4	100001	32.4	34.2	33.3	4.8321
128	8	4	200001	32.2	35.2	33.7	4.82686
256	8	4	200001	32.2	36.1	34.1	5.512
256	2	2	50001	33.0	35.6	34.3	5.3901

Table 1: Cross Entropy

NCE:

Batch_size	num_skips	skip_windows	Max num steps	Least Illustrative	Most Illustrative	Overall Accuracy	Num sample
64	4	8	50001	33.3	31.0	32.1	64
64	8	4	100001	32.7	30.6	31.7	64
64	8	4	200001	29.2	35.6	32.4	64
64	16	8	200001	32.2	33.4	32.8	128
128	2	2	50001	32.4	34.4	33.4	64
128	4	8	100001	33.5	35.7	34.6	64
128	8	4	100001	31.4	30.6	31.0	16
128	8	4	100001	34.9	32.8	33.9	32
128	8	4	100001	31.9	30.3	31.1	128
128	8	4	200001	32.7	36.2	34.5	64
256	4	8	200001	29.5	35.7	32.6	64

Table 2: NCE

TOP 20 SIMILAR WORDS:

Cross Entropy:

American:

```
[‘american', 'german', 'british', 'french', 'english',
'italian', 'russian', 'irish', 'canadian', 'december',
'player', 'in', 'its', 'terminal', 'european', 'edinburghers',
'hyundai', 'contributors', 'composer', 'war', ‘borges’]
```

First:

```
['first', 'last', 'following', 'same', 'name', 'most',
'original', 'second', 'end', 'during', 'government', 'city',
'best', 'largest', 'before', 'area', 'main', 'after',
'entire', 'title', 'rest']
```

Would:

```
['would', 'will', 'could', 'must', 'did', 'can', 'india',
'should', 'do', 'we', 'may', 'to', 'they', 'does', 'said',
'but', 'so', 'made', 'even', 'only', ‘families']
```

NCE:

American:

```
['american', 'eritrea', 'caused', 'orville', 'damage',
'permeability', 'approximately', 'champion', 'farmer',
'shalmaneser', 'redenbacher', 'livestock', 'eca', 'sherman',
'suzerainty', 'ethiopian', 'million', 'serious', 'stanwyck',
'fz', 'eight']
```

First:

```
['first', 'strongly', 'caesarian', 'niemeyer', 'harvey',
'alle', 'is', 'danville', 'geologist', 'incarnate', 'pitcher',
'haiti', 'comic', 'storage', 'titusville', 'lizards', 'gage',
'surroundings', 'homosexuality', 'buckaroo', 'dist']
```

Would:

```
['would', 'teletype', 'asynchronously', 'telephony',
'cromarty', 'redirects', 'that', 'outbreaks', 'petn',
'blasting', 'race', 'eschaton', 'depletion', 'firstperson',
'fountain', 'adaptor', 'forfar', 'longer', 'bute',
'paradigms', 'gateway']
```

EXPLANATION FOR NCE:

The the Noise contrastive estimation loss function was introduced to overcome the limitation posed by the cross entropy function that uses a compute intensive normalisation factor in the denominator of its softmax equation.

To do this, the objective of this function was slightly changed. Instead of going about predicting the context words for a given centre word, now the aim was to tell the true context word apart from the 'num_sampled' (or 'k') negative words. Hence it now becomes a classification problem on some levels.

My understanding from the equation:

With respect to embeddings, while Cross entropy was just trying to make sure that contextual words are closer to each other in the embedding space, it does not bother to keep the non-contextual words far apart from each other.
To break it down, Cross entropy might make sure that the words 'cap', 'hat' and 'umbrella' are close to each other, but doesn't explicitly ensure 'cap' and 'orange' are farther apart. (*Unless cross entropy is also negative sampled*)

And statistically, a word that is a part of a vocabulary has more non-contextual words than contextual words. Hence to make sure that most of the non-contextual/negative words are far at the same time keeping the related words close to each other are the two parts of the NCE equation.

The first part tries to increase the probability of two (closely related) words to occur together, while the second part tries to decrease the probability of all the other negative words to occur together with the centre word.

What was beyond my reach:

I. I wasn't able to understand why they would multiply the 'k' with unigram probabilities. (math beyond my reach)

II. I wanted to implement a different 'k' samples for each word pair in a batch. (complexity in the vector preparation)

III. I wanted to understand how a small noise added to the unigram probabilities affect its performance. (Just didn't have the time).

IV. I wanted to see how the model performed with RMSprop and Adam as optimisers. (Expensive to start from a random weight over pertained model)