NLP Assignment 1

SECTION 1:

Hyper-Parameters explored:

- Max_num_steps: This is basically the number of steps the training process is divided into. Decreasing the number of training steps actually degraded the performance of the model as can be expected since reducing the training steps means that the values are updated a smaller number of times as compared to when the steps were 200000. Maybe this can be the reason for the decreased accuracy.

- Batch_size: It is defined as the number of instances taken in a batch. It can be believed that having a large batch_size as it will speed up our training model but in practice it is better to have smaller batch sizes lead to good solutions as the model gets a chance to learn before seeing the entire data available to it.

- Skip_window: It is a limit on the number of words to consider in the neighborhood of our target word. Having a larger value for skip_window can help us get more information about the context in which the target word is being used as more neighbors will be considered. Having a smaller value will restrict our understanding of the word.

- Num_skips: It tells us how many times to reuse a particular word for generating labels

- Num_sampled: It is the number of negative samples that we take for sampling. It is used in calculating the nce loss as now we can perform same computations as the softmax function but we won't have to go over the entire vocabulary

SECTION 2:

Best Configurations for Cross Entropy:

1. Original Config:
   Accuracy of Least Illustrative Guesses: 30.7%
   Accuracy of Most Illustrative Guesses: 29.5%
   Overall Accuracy: 30.1%
2. Max_num_Steps=100001:
   Accuracy of Least Illustrative Guesses: 31.9%
   Accuracy of Most Illustrative Guesses: 31.7%
   Overall Accuracy: 31.8%
3. Num_sampled=32 and num_skips=4
   Accuracy of Least Illustrative Guesses: 33.7%
   Accuracy of Most Illustrative Guesses: 31.1%
   Overall Accuracy: 32.4%

4. Batch_size=64
   Accuracy of Least Illustrative Guesses: 32.7%
   Accuracy of Most Illustrative Guesses: 31.7%
   Overall Accuracy: 32.2%
5. Num_sampled=32, num_skips=4 and batch_size=64
   Accuracy of Least Illustrative Guesses: 33.5%
   Accuracy of Most Illustrative Guesses: 31.8%
   Overall Accuracy: 32.7%
6. Num_skips=4
   Accuracy of Least Illustrative Guesses: 33.8%
   Accuracy of Most Illustrative Guesses: 35.6%
   Overall Accuracy: 34.7%

What I learnt from tuning these parameters was that decreasing the batch_size, num_skips and num_sampled improves the accuracy of the model.
The best accuracy I got was 34.7% using the cross-entropy loss function. The configuration for this model was batch_size=128, skip_window=4, num_skips=4, max_num_steps=2000001.

Best Configurations for Noise Contrastive Estimation:

1. Original Config:
   Accuracy of Least Illustrative Guesses: 34.0%
   Accuracy of Most Illustrative Guesses: 32.3%
   Overall Accuracy: 33.2%
2. Max_num_steps=100001
   Accuracy of Least Illustrative Guesses: 33.8%
   Accuracy of Most Illustrative Guesses: 33.0%
   Overall Accuracy: 33.4%
3. Max_num_steps=100001 and batch_size=64
   Accuracy of Least Illustrative Guesses: 32.4%
   Accuracy of Most Illustrative Guesses: 34.5%
   Overall Accuracy: 33.4%
4. Batch_size=64
   Accuracy of Least Illustrative Guesses: 34.9%
   Accuracy of Most Illustrative Guesses: 36.7%
   Overall Accuracy: 35.8%
5. Batch_size=64 and num_skips=4
   Accuracy of Least Illustrative Guesses: 32.4%
   Accuracy of Most Illustrative Guesses: 31.4%
   Overall Accuracy: 31.9%
6. Num_skips=4
   Accuracy of Least Illustrative Guesses: 36.5%
   Accuracy of Most Illustrative Guesses: 33.4%
   Overall Accuracy: 35.0%

Even in the case of NCE, decreasing the batch_size from 128 to 64 improved the model and what I can infer from this is that having small batches helps the model to train on a small fraction of the vocabulary before working out the entire data available to it.
The best accuracy I got was 35.8% using the NCE loss function. The configuration for this model was batch_size=64, skip_window=4, num_skips=8, max_num_steps=2000001.

SECTION 3:
Top 20 similar words:
For Cross Entropy:

Nearest to first: bonde, jilted, wikitravel, shippers, hesitation, durer, chickpeas, alanson, pensacola, limousine, kralove, kearns, alyattes, garbage, mora, kurosawa, goldwater, keyboards, forged, indology
Nearest to american: waldron, furisode, brandsma, normalizing, modulus, cilician, helped, existentialism, minister, pritam, quarrying, alienware, grafting, liturgically, eidyn, diakonoff, siang, veteran, cookbook, fumaric
Nearest to would: edmonds, ether, physical, hasdrubal, softmodem, fincke, intrinsically, gundam, deque, gilliam, hurried, konerak, eavesdropper, earn, oz, imprinting, biggest, unpreparedness, cucumber, orthogonality

For Noise Contrastive Estimation:

Nearest to first: risley, roast, kb, fullspeed, rfcs, lengthwise, dumbella, tailgating, hunebedden, saru, elinor, flavors, inspirations, danelaw, astable, intoxicated, subexpression, engi, moans, downy
Nearest to american: hoceima, hoch, privatisation, dirk, schechter, milli, vitruvian, highrise, saddled, converts, effeminacy, berries, coordinating, dresses, manufactured, punting, raptors, osx, berchtesgaden, tarry
Nearest to would: metazoa, iago, hypothesized, tikar, usps, katherine, quran, dmz, airshows, geboren, nari, medarabtel, australopithecus, dawlah, serpents, galton, tuli, fouled, colony, sorely

SECTION 4:

Summary of Noise Contrastive Estimation:

The reason we want to look for other loss functions is to somehow avoid using the expensive softmax function. Noise Contrastive Estimation method is one such method which selects a sample from the true class labels and some noisy class labels. The noisy class labels are basically unigram distribution of the words.

But in doing so we get an expensive computation in the denominator of eq (1) which we want to avoid. The D=1 and D=0 parts in eq (8) in the paper basically say that only one sample is taken from the true class and k number of samples are taken from the noisy class respectively.

The idea behind this method is to train a logistic regression classifier to differentiate between the true distribution and the noise distribution. In eq (7), we replace the P(w) term with s(w,h) as we ignore the normalization and replace it with an unnormalized term which is easier to compute and is basically a sigmoid equation (eq (7)). This makes the model have the same performance as Maximum Likelihood Estimation but at a reduced cost of computation.

From eq (8) we can see that the D=1 term becomes log of the sigmoid calculated before and the D=0 term becomes 1-log of the sigmoid term which turns our model into something similar to a binary classifier as mentioned in the first few lines of part 3.1 in the paper. In this binary classifier we consider the true classes as positive samples and the noisy classes as negative samples.
The main reason why NCE is not as expensive as Softmax and is better is because it takes the sum for k noise samples and not the entire vocabulary thus making NCE linear and independent of the vocabulary size as can be seen from eq (9) in the paper.