Natural Language Processing:

Assignment 1 - Report

Name: Payal Mehta

SBU id: 112715351

==========

Section1: Hyper-Parameters exploration:

The following hyper parameters were explored during this assignment: -

1. Learning Rate:
   In layman terms, the value of learning rate decides how quickly a model forgets. If the value is too high, a model abandons all its previous learning and quickly adapts to the new ones. In case of a smaller learning rate, it takes a while to distinguish between new and old learnings and the model takes longer to adapt to the newer changes.
   Values explored: 0.5, 1.0

2. Batch Size:
   The value of batch size is an indicator of the number of negative samples used, to normalize the value of positive samples. The overall distribution of a dataset, rather a training dataset can be better represented by a larger batch size. However, a larger batch size also means larger computation time. A model with a smaller batch size can be computed in comparatively lesser amount of time but the distribution of the dataset won't be as accurate as the one with a larger batch size. Generally, the smaller the batch size, the less accurate a model tends to be. A trade-off has to be made between accuracy and computation, thereby also avoiding over-fitting (a model learning everything and negatively impacting its performance) or under-fitting (a simple basic model) your model.
   Values explored: 64,128,192,256

3. Maximum number of steps:
   The higher the number of steps, the better a model understands the overall distribution of the training data.
   Values explored: 100001,200001

4. Skip window:
   The value of a skip window decides how many words to consider left and right from a context word. Logically, having a higher skip window means the model can better learn the context of a word; it can better understand what role a word plays in a sentence.
   Example: for a skip window = 3,
   I love apple as a company
   (apple,love), (apple,I), (apple,as), (apple,a), (apple,company)

The last pair helps me decide that the context here actually means apple as a company and not as a fruit. Skip windows help decide how big the context window would be. Window_size = skip_window * 2 + 1. Therefore, for a skip window of 2. The context window size is 5.
Values explored: 4,8

5. Number of skips:
   The value of "number of skips" tells us how many samples we should draw in a window. Skip window tells us how far we see, and number of skips how much we see with respect to a model. Too high a value would spoil the context, the model is trying to learn.
   Values explored: 8,16

==========

Section 2 – Results of analogy tasks.

Cross entropy – six best configurations of the hyper parameters.

Sr no.	Batch size	Skip window	No. of skips	Max num steps	Overall accuracy(%)
Default config-1	128	4	8	200001	33.6
2	64	4	8	200001	34.2
3	192	4	8	200001	34.2
4	256	4	8	200001	33.4
5	192	4	8	100001	34.2
6	64	4	8	100001	33.8
7	192	8	8	200001	34.3

Batch size: - Four different values were tried – 64,128,192,256 keeping max number of steps, skip window and number of skips constant at 200001,4 and 8. The overall accuracy of the model lay between 33.6-34.2. The highest being at a batch size of 64 and 192. To find the most optimal one, I ran both of these at max steps of 100001 and the batch size with 192 came out to be the better performing model. Specialization can be achieved with a smaller batch size and a larger batch size helps in generalizing the model, with 192 being an optimal one.

Skip windows: - Two different configurations were tried for fixed batch size of 192 – 4, 8. The cross entropy model performed better at a batch size of 8. The accuracies were 34.2 and 34.3. The batch size of 8 turned out to be better with max_steps constant at 200001.

Max steps: - Two different values were explored for maximum number of steps 100001, 200001 – keeping batch size and number of skips constant at 192 and skip window at 4. My model was more accurate with 100001 being the optimal value. As more max steps might lead to overfitting issues.

NCE loss function – five best configurations of the hyper parameters.

Sr no.	Batch size	Skip window	No. of skips	Max num steps	Avg loss at the last training step	Overall accuracy(%)
Default config-1	128	4	8	200001	1.6	33.4
2	64	4	8	200001	1.33	34.3
3	192	4	8	200001	1.18	34.7
4	256	4	8	200001	1.23	34.1
5	192	4	8	100001	1.3	35.0
6	192	8	8	100001	1.4	34.8

Batch size: - Five different values were tried – 64,128,192,256 keeping max number of steps, skip window and number of skips constant at 200001,4 and 8. The overall accuracy of the model lay between 34-35. The highest being at a batch size of 192. Specialization can be achieved with a smaller batch size and a larger batch size helps in generalizing the model, with 192 being an optimal one.

Skip window: - Two different configurations were tried for fixed batch size of 192 – 4, 8 and max steps constant at 100001. The accuracies were 34.8, 35. The NCE model performed better at a batch size of 4. Skip windows with size less than 4 captured lesser information and more than 4 more information leading to the poor performance of the model. Here 4 is an optimal skip window size for NCE.

Max steps: - Two different values were explored for maximum number of steps 100001, 200001 – keeping batch size and number of skips constant at 192 and skip window at 4. My model was more accurate with 100001 being the optimal value. As more max steps might have led to overfitting issues.

==========

Section3 – Top 20 similar words

Below is the list of top 20 similar words to “first”,” american” and “would”, as per both the models; in decreasing similarity, with the most similar listed first.

For cross entropy loss model

first	american	would
most	german	could
last	british	will
best	english	must
same	italian	can

name	french	may
following	russian	is
original	canadian	was
largest	european	might
only	barzani	did
latter	united	had
next	freedoms	should
second	roman	does
main	hastened	has
continued	espionsa	came
a	states	are
left	tentatively	we
paghman	eu	began
entire	judeo	do
word	heckman	were
lenape	west	wanted

For noise contrastive loss model

first	**american**	**would**
th	war	only
during	during	will
all	century	so
time	english	him
century	other	called
war	time	more
which	th	t
some	all	no
since	first	such
before	its	i
early	more	where
he	so	many
other	later	later
american	since	since
at	many	their
however	have	did
history	where	until
where	some	zeno
about	how	could
many	no	might

Section 4 – Justification of NCE loss.

The main issues that NCE addresses, which other models like NPLM face are targeted more towards scalability. NPLM suffers from scalability issues because of hidden layers and the expensive probability computation spanning over entire vocabulary. Cross entropy targets to predict outer words, given the context word. Since the vocabulary is huge, it becomes a huge problem as we need to normalize the probabilities over a large vocabulary. NCE works on the concept of binary classification. Given a word, NCE tells me whether the word is from a positive sample or from a negative sample.

NCE does so by creating a dataset of positive samples drawn from the vocabulary which is a noisy distribution. It also draws for every positive sample, k negative samples. The probabilities of these samples are calculated by looking up the unigram vector.

To distinguish between positive and negative samples, the score of the outer word is calculated under both original and negative distribution.

The score is calculated as the dot product of the word **uc** and **uo**. **uc** is the context word embedding and **uo** is the embedding of the outer word. Bias is calculated by looking up the biases vector with the positive label ids. This bias is then added to the dot product.

$$s(wo, wc) = (T(uc).\ uo) + bo$$

$$s(wx, wc) = (T(ux).\ uo) + bo1$$

The score for the negative samples is calculated as k multiplied by the dot product of **ux** and **uc**, where **ux** is the negative word embedding, k is the length of the negative samples and **uc** is again our context word embedding.

The log of the probabilities is then taken, where the probabilities are calculated by performing a lookup on the unigram probability distribution. The final positive probability is then achieved by taking a sigmoid function of the difference of the scores under the positive and negative distributions.

$$Pr(D = 1, wo|wc) = \sigma\ (s(wo, wc) - \log\ [kP\ r(wo)])$$

$$Pr(D = 1, wx|wc) = \sigma\ (s(wx, wc) - \log\ [kP\ r(wx)])$$