Report - HW1

Name: Faizaan Charania
SBU ID: 111463646

1. Hyperparameters explored

a. **Embedding size:** The embedding size decides the size of the vector we use to store information about the word. The higher the vector size, more is the information that we can store about the word. However, larger embeddings are more difficult to train and also increase the memory footprint of our model.

b. **Skip window:** Skip window denotes the number of words we consider to the left and right of our center word, when trying to generate an embedding for the center word. As the surrounding words define how a word is used in a context and what its actual meaning can be, skip window is an important parameter to tune.

c. Num skips: It is not necessary to consider all words from the skip window when we are trying to generate data samples for training. We vary num_skips to consider only a sub-sample of 'context words' when trying to generate training data.

d. Learning rate: This is the learning rate for the SGD optimizer. While the initial learning rate (1) is too high, decreasing the learning rate also reduces the pace at which the model converges.

e. Number of Epochs: Defines the number of times the model is trained over the training dataset. Generally speaking, higher the epochs, lower is the training loss of the model. However, this can also lead to overfitting, so that needs to be taken into account.

f. Negative samples for NCE: This defines *'k'* negative samples that we consider when calculating NCE loss.

2. Five Experiment configurations

a. NCE

Loss	Batch size	Neg samples	Emb size	Skip window	Learning rate	Epochs	Accuracy
nce	128	64	128	5	1e-2	200001	34.6%
nce	128	64	256	8	1e-2	200001	31.9%
nce	128	128	128	5	1e-2	200001	29.8%
nce	128	32	128	5	1e-2	200001	31.9%
nce	128	64	128	5	1e-3	1000000	33.2%
cross_entropy	128	-	128	5	1e-2	200001	33.4%
cross_entropy	128	-	128	8	1e-2	400001	32.1%
cross_entropy	256	-	256	5	1e-2	400001	29.9%
cross_entropy	128	-	256	8	1e-2	200001	31.7%
cross_entropy	128	-	128	8	5e-3	400000	33.6%

I have listed (approx. top) 5 of the many experiments performed on the hyperparameters.

Important Observations:

i. Increasing or decreasing the number of negative samples resulted in decreased accuracy (which means the current size is ideal)
ii. Increasing the embedding size didn't help in achieving a higher accuracy
iii. Increasing the size of the skip window has shown contradictory results based on other parameters. It considers more context words, hence getting more samples, but it also pairs together relatively unrelated words in training
iv. Increasing the number of epochs generally didn't lead to a higher accuracy model as the learning gets saturated, but sometimes achieved same accuracy with lower loss
v. Training models from scratch tend to give higher accuracy, but when most similar words are evaluated, the pairings don't make a linguistic sense.

The best models are chosen based on the tradeoff on getting sensible 'similar words' as well as a good accuracy on the dev set.

3. ## Words similar to [first,american, would]

   a. **Cross Entropy**

      i. **first :** ['last', 'name', 'following', 'during', 'most', 'original', 'second', 'same', 'until', 'end', 'after', 'best', 'city', 'book', 'before', 'united', 'next', 'main', 'beginning', 'title']
      ii. **American :** ['german', 'british', 'french', 'english', 'italian', 'its', 'russian', 'war', 'european', 'understood', 'international', 'borges', 'irish', 'canadian', 'united', 'trade', 'of', 'd', 'writer', 'player']
      iii. **would :** ['not', 'could', 'will', 'been', 'that', 'we', 'said', 'must', 'india', 'they', 'do', 'does', 'who', 'did', 'you', 'families', 'to', 'if', 'should', 'may']

   b. NCE

      i. **first :** ['most', 'during', 'name', 'was', 'at', 'after', 'and', 'one', 'of', 'in', 'on', 's', 'is', 'to', 'last', 'he', 'following', 'nine', 'which', 'from']
      ii. **american :** ['british', 'german', 'war', 'english', 'french', 'its', 'united', 'european', 'states', 'sale', 'borges', 'understood', 'century', 'international', 'enrollment', 'of', 'italian', 'alesia', 'hawthorne', 'sholay']
      iii. **would :** ['not', 'been', 'they', 'will', 'could', 'who', 'that', 'we', 'said', 'to', 'india', 'must', 'but', 'did', 'from', 'only', 'with', 'which', 'these', 'do']

It is noteworthy that similar words are words that generally appear in the same context. A lot of words similar to ‘american’ are other nationalities. These ‘similar words’ are words that we would expect to see around our target words, therefore we conclude that the word embedding makes at least some sense.

4. Explanation of NCE loss

The main advantage of NCE loss is that we can calculate the loss for the data, independent of the vocabulary size. This gives us the advantage of using billion word vocabularies without worrying about the computation cost of the NCE loss.

The basic idea is to train a logistic regression classifier to discriminate between samples from the data distribution and samples from some “noise” distribution, based on the ratio of probabilities of the sample under the model and the noise distribution. The trained ‘model’ should give high probabilities for words from the same context (same ‘distribution’) and low probabilities for words from the ‘noise’ distribution.

Suppose we would like to learn the distribution of words for some specific context *h*, denoted by $P^h(w)$. To do that, we create an auxiliary binary classification problem, treating the training data as positive examples and samples from a noise distribution $P^n(w)$ as negative examples. We are free to choose any noise distribution that is easy to sample from and and that does not assign zero probability to any word.

We fit the model by maximizing the log-posterior probability of the correct labels D averaged over the data and noise samples, where we assume noise samples are k times more likely to occur compared to our sample distribution

$$J(\theta, Batch) = \sum_{(w_o, w_c) \in Batch} - \left[\log Pr(D = 1, w_o | w_c) + \sum_{x \in V^k} \log(1 - Pr(D = 1, w_x | w_c)) \right]$$

where,

$$Pr(D = 1, w_o | w_c) = \sigma\left(s(w_o, w_c) - \log\left[kPr(w_o)\right]\right)$$
$$Pr(D = 1, w_x | w_c) = \sigma\left(s(w_x, w_c) - \log\left[kPr(w_x)\right]\right)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

and

$$s(w_o, w_c) = (u_c^T u_o) + b_o$$

where u_c, and u_o are the context and the target word vectors, and b_o is a bias vector specific to w_o

The perplexity of Neural probabilistic language models (NPLMs) trained using NCE loss have been on par with those trained with maximum likelihood learning, but at a fraction of the computational cost.