Assignment-1 Discussion (Word Vectors and Analogy Test)

sanjna mohan, 20305R006
joshi meet anilkumar, 21319R001
mahesh ashok abnave, 203059010
akshay eknath mali, 213079006
date-13 feb 2023

Problem Statement (1/2)

- In this assignment, you will have to implement the backpropagation algorithm from scratch. After implementing backpropagation ab-initio, train CBow and Skip-gram with backpropagation. (This link might help as a quick refresher for Skip-gram and CBoW.)
- The task is to compare the performance of CBoW and Skip-gram embeddings on the word analogy task.
  - Analogy task: Given an analogy, find a word by correctly determining its relationship with another word. For example,

Problem Statement (2/2)

- man:woman :: king:_____
- (man is to woman, what king is to ____)
- The blank should be filled with “queen”.
- **Input**: An analogy pair with one blank. For e.g.,
  - Delhi:India :: Paris:_____
- **Output**: The correct word to satisfy the analogy given in the input.
- You will have to report on the validation data:
  - Accuracy
  - Compare the performance of CBoW and Skip-gram models.
  - Perform detailed error analysis

Working with Data

- Pre-processing:
  - stop words,number, punctuations removed
  - no lemmatization as it removes morphological words

- Data Scraping to facilitate analogy:
  - 200+ sentences for each analogy word from wikipedia pages.

- Vocabulary size:
  - 49245

Experimental Setup

- Embedding dimension = 130-200
- k(no of sentences extracted from wikipedia for each word)=200
- window =3
- no. of iterations=35
- learning rate= 0.025 - 0.008
- input output pairs:659420

```
['emma', 'woodhouse', 'handsome', 'clever', 'comfortable', 'home', 'happy', 'disposition']rich
['woodhouse', 'handsome', 'clever', 'rich', 'home', 'happy', 'disposition', 'seemed']comfortable
['handsome', 'clever', 'rich', 'comfortable', 'happy', 'disposition', 'seemed', 'unite']home
['clever', 'rich', 'comfortable', 'home', 'disposition', 'seemed', 'unite', 'best']happy
```

Backpropogation

- input vector (V,batchsize), hidden layer(dim,1),output layer(softmax)(V,batchsize)
- h= output of hidden layer
- y_pred=softmax layer
- calculate the loss vector =loss= (y_true-y_pred)
- l1 = np.dot(W1 ,loss)
- dW1 = np.dot(h,loss.T)
- dW0 = np.dot(x,loss.T)
- W1 -= alpha * dW1
- W0 -= alpha * dW0

Results

- accuracy:
- skipgram & cbow: vocabulary = 49245

top k	skipgram	cbow
1	0.03935	0.02421
10	0.10797	0.06962
50	0.19778	0.13824
100	0.24318	0.16851
500	0.33198	0.24924

- observation:top-50 accuracy on geographical pair(65)
- 0.42 for cbow , 0.61 for skipgram

Analysis

- need to vary context size, epochs
- tried with certain embedding dimesion and found that with decresing size accuracy starts increasing.need to explore this further.
- in backprop , cbow loss reduces satisfactory but skipgram loss stuck and does not decrease after certain epochs till satisfactary low value.
- accuracy of skipgram is slightly higher than cbow
- skip-gram to better capture the nuances of the context in which a word appears.
- CBOW outperforms skip-gram, such as when the goal is to generate embeddings that capture the overall meaning of a sentence rather than just the relationships between individual words.

Demo

- <on notebook, show>
- <remain ready; no hurry scurry during the evaluation>
- <should take evaluators examples>

Marking (max 100)

- Data pre-processing: 10 (no pre-processing 0)
- Scraping: 20 (respectable size and good scraping strategy full marks)
- Good implementation of BP: 20
- Accuracy: >90: 20 marks; >70-90: 10 marks; >50-70: 5 marks; else 0
- Analysis: 10 marks
- Comparison of CBOW and Skip Gram: 10 marks
- Demo: 10 marks
- Topper of leader board: 5 marks bonus