CSC413Coding_A1

haoyanjiang 1003324716

February 3, 2020

1 Part 1: Linear Embedding (GLoVE)

Answer the following questions:

- 1. **Given the vocabulary size V and embedding dimensionality d, how many parameters does the GLoVE model have?**

  The models is $L(\{\mathbf{w}_i, b_i\}_{i=1}^{V}) = \sum_{i,j=1}^{V}(\mathbf{w}_i^\top \mathbf{w}_j + b_i + b_j - \log X_{ij})^2$, where w_i and w_j comes from matrix W with dimension $V \times d$, b_i and b_j comes from matrix dimension d.
  The therefore, in total, there are **V× d + d** variables to be trained

- 2. **Write the gradient of the loss function with respect to one parameter vector $\mathbf{w}_i$.**

$$\begin{aligned}\frac{\partial L}{\partial w_i} &= 2\sum_{j=1,j\neq i}^{V}(w_i^T w_j + b_i + b_j - \log X_{ij})w_j \\ &= 4\sum_{j=1}^{V}(w_i^T w_j + b_i + b_j - \log X_{ij})w_j\end{aligned}$$

Also, for b

$$\begin{aligned}\frac{\partial L}{\partial b_i} &= 2\sum_{j=1,j\neq i}^{V} w_i^T w_j + b_i + b_j - \log X_{ij} \\ &= 4\sum_{j=1}^{V} w_i^T w_j + b_i + b_j - \log X_{ij}\end{aligned}$$

- **3. Implement the gradient update of GLoVE.**

  $$grad\_w = 4(W * W^T + b * [1, 1, 1, \ldots, 1]_V + [1, 1, 1, \ldots, 1]_V^T * b^T + \log(X)) * W$$
  $$grad\_b = 4(W * W^T + b * [1, 1, 1, \ldots, 1]_V + [1, 1, 1, \ldots, 1]_V^T * b^T + \log(X)$$

- **4. Train the model with varying dimensionality d. Which d leads to optimal validation performance? Why does / doesn't larger d always lead to better validation error?**

  when d = 11, d leads to optimal validation performance with the smallest validation error. Model may be over fit if dimension is too large, while underfit if dimension is too small.

2 Part 2: Network architecture

- 1. Word embedding weight: 250×16
  Embedded to hidden weight: $3 \times 16 \times 128$
  Hidden to output weight: 128×250
  Hidden bias: 128×1
  Output bias: 250×1
  Total: 42522 parameters. The hidden to output weight has the most parameters

- 2. A 4-grams model is a model predict the 4th word from 3 previous words. That is, we have 250 vocabularies in total, there is a combination of 250^4=3906250000 possible non-repeated outcomes.

3 Part 3: Training the Neural Network

Output for *print_gradients*():
loss_derivative[2, 5] 0.001112231773782498
loss_derivative[2, 121] -0.9991004720395987
loss_derivative[5, 33] 0.0001903237803173703
loss_derivative[5, 31] -0.7999757709589483

param_gradient.word_embedding_weights[27, 2] -0.27199539981936866
param_gradient.word_embedding_weights[43, 3] 0.8641722267354154
param_gradient.word_embedding_weights[22, 4] -0.2546730202374648
param_gradient.word_embedding_weights[2, 5] 0.0

param_gradient.embed_to_hid_weights[10, 2] -0.6526990313918257
param_gradient.embed_to_hid_weights[15, 3] -0.13106433000472612
param_gradient.embed_to_hid_weights[30, 9] 0.11846774618169399
param_gradient.embed_to_hid_weights[35, 21] -0.10004526104604386

param_gradient.hid_bias[10] 0.25376638738156426
param_gradient.hid_bias[20] -0.03326739163635369

param_gradient.output_bias[0] -2.062759603217304
param_gradient.output_bias[1] 0.03902008573921689
param_gradient.output_bias[2] -0.7561537928318482
param_gradient.output_bias[3] 0.21235172051123635

4 Part 4: Analysis

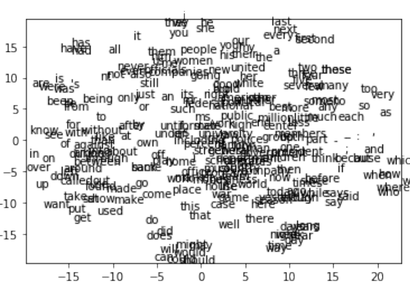
- 1. Test on "she is a", with "part", "family" being very plausible 4th word in that context, while not appearing in the dataset

  ```
  trained_model.predict_next_word("she", "is", "a")
  find_occurrences("she", "is", "a")
  ```

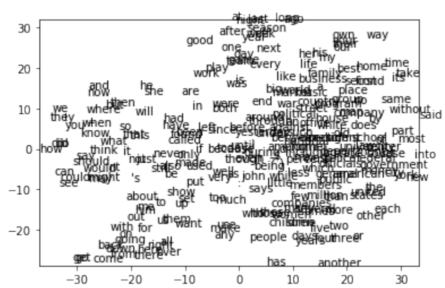
  output:
  she is a good Prob: 0.24169
  she is a part Prob: 0.07294
  she is a very Prob: 0.06827
  she is a family Prob: 0.06452
  she is a man Prob: 0.05457
  she is a big Prob: 0.05185
  she is a home Prob: 0.03534
  she is a team Prob: 0.03325
  she is a long Prob: 0.02967
  she is a new Prob: 0.02846
  The tri-gram "she is a" was followed by the following words in the training set:
  year (1 time)
  big (1 time)
  good (1 time)

- 2. Compared with the second tsne GLoVE plot which has embedded dimension of 256, the first tsne plot has 250 embedded dimension, converges with similar 110 iterations, and has smaller error of about 0.633. The first model trains much slower than GLoVE model, while this tsne graph has more distinguishable clusters, which classifies better than GLoVE model. Some example cluster of words in the first model are: ("has", "have", "had"), ("might", "will", "would", "should", "can", "may", "could"), while the second model using GLoVE groups "see" and "say" with "would", "have" with "called". The first model with neural net has more precise and clear classification result than the second model.

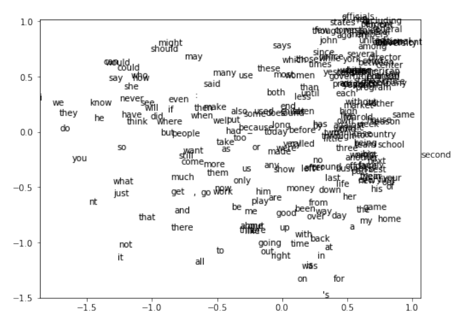
  For the third plot, it plots the 2d GLoVE model without tsne, while the fourth plot represents the same model with tsne, converges around 110 iteration with about 0.304 error. Compared with the third plot, the fourth plot with tsne has better clustering, while the third plot is more dispersed and distracted.



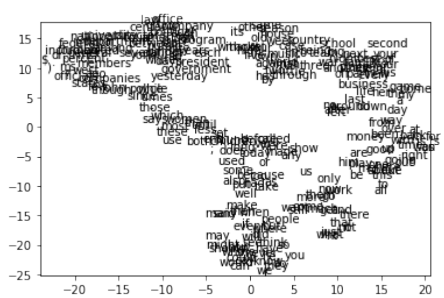
(a) Figure 1. tsne plot for model



(b) Figure 2. tsne plot for GLoVE model over 256 dimensions



(c) Figure 3. 2d plot for GLoVE model, dimension 2



(d) Figure 4. tsne 2d plot for GLoVE model, dimension 2

Figure 1: 2d visualization of models

- 3. As we can see, "new" and "york" are not close together, which is also as expected, because if two words close together, this means that the similarity between two words are very large. While "new" and "york" have very different meanings and nature in language, and should not be categorized into the same group.

```
print(trained_model.word_distance("new", "york"))
print(trained_model.display_nearest_words("new"))
print(trained_model.display_nearest_words("york"))
```

  output:
  3.548393385127713
  old: 2.3600505246264323
  white: 2.4383030498943365
  back: 2.5759303604425643
  american: 2.6466202075516416
  such: 2.6599246057735786
  own: 2.6878098904671783
  political: 2.701124031526305
  national: 2.7494668874744193
  several: 2.7643670072967392
  federal: 2.8155523660546993
  None
  public: 0.9048826050086719
  music: 0.9105096376369168
  university: 0.9309985183745197
  city: 0.9640999583181336
  department: 0.9810341593819769
  ms.: 0.985606172603071
  john: 1.0174907854088202
  school: 1.035702631398839
  general: 1.0443388378683818
  team: 1.0641153998658832
  None

- 4. Comparing ("goverment", "political") and ("government", "university"), "government" is more close to "university" rather than "political". This is plausible, because both of the government and university are noun in a sentance, while political is subjective.

```
print(trained_model.word_distance("government", "political"))
print(trained_model.word_distance("government", "university"))
```

  output:
  1.4021420047033426
  0.9350013818643479