Lecture 8: NLP and Word Embeddings

Alireza Akhavan Pour

CLASS.VISION
NLP and Word Embeddings

Word representation

\[ V = [a, aaron, ..., zulu, <UNK>] \]

\[ |V| = 10,000 \]

1-hot representation

\[
\begin{array}{c|c|c|c|c|c|c}
\hline
\text{Man} & \text{Woman} & \text{King} & \text{Queen} & \text{Apple} & \text{Orange} \\
\hline
5391 & 9853 & 4914 & 7157 & 456 & 6257 \\
\hline
\end{array}
\]

I want a glass of orange \textit{juice}.

I want a glass of apple ______.

مشکل؟

فاصله اقليدسی تمام بردارها یکسان است.

از روي كلماتي كه در آموزش ديده است، نميتواند generalize كند.
### Featurized representation: word embedding

<table>
<thead>
<tr>
<th></th>
<th>Man (5391)</th>
<th>Woman (9853)</th>
<th>King (4914)</th>
<th>Queen (7157)</th>
<th>Apple (456)</th>
<th>Orange (6257)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Šabname</td>
<td>-1</td>
<td>1</td>
<td>-0.95</td>
<td>0.97</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>۹۱ آبان</td>
<td>0.01</td>
<td>0.02</td>
<td>0.94</td>
<td>0.93</td>
<td>-0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>۷۹۳۱</td>
<td>0.03</td>
<td>0.02</td>
<td>0.71</td>
<td>0.69</td>
<td>0.03</td>
<td>-0.02</td>
</tr>
<tr>
<td>۶۲۵۷</td>
<td>0.01</td>
<td>-0.01</td>
<td>0.02</td>
<td>0.00</td>
<td>0.96</td>
<td>-0.97</td>
</tr>
</tbody>
</table>

I want a glass of orange *juice*.

I want a glass of apple *juice*.
Visualizing word embeddings

[van der Maaten and Hinton., 2008. Visualizing data using t-SNE]
Using word embeddings: Named entity recognition example

Robert Lin is an apple farmer
Sally Johnson is an orange farmer
Robert Lin is a durian cultivator
Using word embeddings: Named entity recognition example

- Now if you have tested your model with this sentence "Robert Lin is a durian cultivator“ the network should learn the name even if it hasn't seen the word durian before (during training). That's the power of word representations.

- The algorithms that are used to learn word embeddings can examine billions of words of unlabeled text - for example, 100 billion words and learn the representation from them.
Transfer learning and word embeddings

I. Learn word embeddings from large text corpus (1-100 billion of words).
   ➢ Or download pre-trained embedding online.

II. Transfer embedding to new task with the smaller training set (say, 100k words).

III. Optional: continue to finetune the word embeddings with new data.
   ➢ You bother doing this if your smaller training set (from step 2) is big enough.
Relation to face encoding (Embeddings)

- **Word embeddings have an interesting relationship to the face recognition task:**
  - In this problem, we encode each face into a vector and then check how similar are these vectors.
  - Words encoding and embeddings have a similar meaning here.
- In the word embeddings task, we are learning a representation for each word in our vocabulary (unlike in image encoding where we have to map each new image to some n-dimensional vector).

[Taigman et. al., 2014. DeepFace: Closing the gap to human level performance]
Properties of word embeddings

• Analogies

<table>
<thead>
<tr>
<th></th>
<th>Man</th>
<th>Woman</th>
<th>King</th>
<th>Queen</th>
<th>Apple</th>
<th>Orange</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td>-1</td>
<td>1</td>
<td>-0.95</td>
<td>0.97</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>Royal</td>
<td>0.01</td>
<td>0.02</td>
<td>0.93</td>
<td>0.95</td>
<td>-0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>Age</td>
<td>0.03</td>
<td>0.02</td>
<td>0.70</td>
<td>0.69</td>
<td>0.03</td>
<td>-0.02</td>
</tr>
<tr>
<td>Food</td>
<td>0.09</td>
<td>0.01</td>
<td>0.02</td>
<td>0.01</td>
<td>0.95</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Can we conclude this relation:
- Man ==> Woman
- King ==> ??

\[
e_{\text{Man}} - e_{\text{Woman}} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
e_{\text{King}} - e_{\text{Queen}} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\]

[Mikolov et. al., 2013, Linguistic regularities in continuous space word representations]
Analogies using word vectors

Find word $w$: $\arg\max_w \text{sim}(e_w, e_{\text{king}} - e_{\text{man}} + e_{\text{woman}})$
Cosine similarity

\[ \text{sim}(e_w, e_{\text{king}} - e_{\text{man}} + e_{\text{woman}}) \]

\[ \text{sim}(u, v) = \frac{U^TV}{\|u\|_2 \|v\|_2} \]

\[ \|U - V\|^2 \]

Man: Woman as boy: girl

Ottawa: Canada as Iran: Tehran

Big: bigger as tall: taller

Yen: Japan as Ruble: Russia
Embedding matrix

<table>
<thead>
<tr>
<th>here</th>
<th>...</th>
<th>orange</th>
<th>example</th>
<th>...</th>
<th>&lt;UNK&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.2</td>
<td>...</td>
<td>-0.67</td>
<td>-0.2</td>
<td>...</td>
<td>0.2</td>
</tr>
<tr>
<td>0.7</td>
<td>...</td>
<td>0.3</td>
<td>-0.5</td>
<td>...</td>
<td>0.1</td>
</tr>
<tr>
<td>0.85</td>
<td>...</td>
<td>0.25</td>
<td>0.3</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>-0.04</td>
<td>...</td>
<td>-0.18</td>
<td>0.33</td>
<td>...</td>
<td>-0.1</td>
</tr>
</tbody>
</table>

...
**Embedding matrix**

<table>
<thead>
<tr>
<th>here</th>
<th>...</th>
<th>orange</th>
<th>example</th>
<th>...</th>
<th>&lt;UNK&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.2</td>
<td>...</td>
<td>-0.67</td>
<td>-0.2</td>
<td>...</td>
<td>0.2</td>
</tr>
<tr>
<td>0.7</td>
<td>...</td>
<td>0.3</td>
<td>-0.5</td>
<td>...</td>
<td>0.1</td>
</tr>
<tr>
<td>0.85</td>
<td>...</td>
<td>0.25</td>
<td>0.3</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>-0.04</td>
<td>...</td>
<td>-0.18</td>
<td>0.33</td>
<td>...</td>
<td>-0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>...</td>
<td>1</td>
<td>0.3</td>
<td>...</td>
<td>0.2</td>
</tr>
</tbody>
</table>

\[ E \cdot O_{6257} = e_{6257} \]
Embedding matrix

<table>
<thead>
<tr>
<th>here</th>
<th>...</th>
<th>orange</th>
<th>example</th>
<th>...</th>
<th>&lt;UNK&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.2</td>
<td>...</td>
<td>-0.67</td>
<td>-0.2</td>
<td>...</td>
<td>0.2</td>
</tr>
<tr>
<td>0.7</td>
<td>...</td>
<td>0.3</td>
<td>-0.5</td>
<td>...</td>
<td>0.1</td>
</tr>
<tr>
<td>0.85</td>
<td>...</td>
<td>0.25</td>
<td>0.3</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>-0.04</td>
<td>...</td>
<td>-0.18</td>
<td>0.33</td>
<td>...</td>
<td>-0.1</td>
</tr>
<tr>
<td>0.5</td>
<td>...</td>
<td>1</td>
<td>0.3</td>
<td>...</td>
<td>0.2</td>
</tr>
</tbody>
</table>

\[ E \cdot O_{6257} = e_{6257} \]

- If \( O_{6257} \) is the one hot encoding of the word \text{orange} of shape (10000, 1), then \( \text{np.dot}(E, O_{6257}) = e_{6257} \) which shape is (300, 1).
- Generally \( \text{np.dot}(E, O_j) = e_j \) \text{(embedding for word j)}
Embedding matrix

The **Embedding layer** is best understood as a **dictionary mapping integer indices** (which stand for specific words) to dense vectors. It takes as input integers, it looks up these integers into an internal dictionary, and it returns the associated vectors. **It's effectively a dictionary lookup.**

```python
from keras.layers import Embedding

# The Embedding layer takes at least two arguments:
# the number of possible tokens, here 1000 (1 + maximum word index),
# and the dimensionality of the embeddings, here 64.
embedding_layer = Embedding(1000, 64)
```
منابع

- https://www.coursera.org/specializations/deep-learning