What's the Word?

Applied Deep Learning

YUN-NUNG (VIVIAN) CHEN

WWW.CSIE.NTU.EDU.TW/~YVCHEN/F105-ADL

Slide credit from Dr. Richard Socher
Learning Target Function

Classification Task

\[ f(x) = y \quad \longrightarrow \quad f : \mathbb{R}^N \rightarrow \mathbb{R}^M \]

- \( x \): input object to be classified → a \( N \)-dim vector
- \( y \): class/label → a \( M \)-dim vector

Assume both \( x \) and \( y \) can be represented as fixed-size vectors

“這規格有誠意!” \( \longrightarrow \) +
“太爛了吧~” \( \longrightarrow \) -

How do we represent the meaning of the word?
Meaning Representations

Definition of “Meaning”

◦ the idea that is represented by a word, phrase, etc.
◦ the idea that a person wants to express by using words, signs, etc.
◦ the idea that is expressed in a work of writing, art, etc.

Goal: word representations that capture the relationships between words
Meaning Representations in Computers

Knowledge-based representation

Corpus-based representation

✓ Atomic symbol
✓ Neighbors
  ◦ High-dimensional sparse word vector
  ◦ Low-dimensional dense word vector
    ▪ Method 1 – dimension reduction
    ▪ Method 2 – direct learning
Meaning Representations in Computers

Knowledge-based representation

Corpus-based representation

✓ Atomic symbol
✓ Neighbors
  ◦ High-dimensional sparse word vector
  ◦ Low-dimensional dense word vector
    ▪ Method 1 – dimension reduction
    ▪ Method 2 – direct learning
Knowledge-based representation

Hypernyms (is-a) relationships of WordNet

```python
from nltk.corpus import wordnet as wn
panda = wn.synset('panda.n.01')
hyper = lambda s: s.hypernyms()
list(panda.closure(hyper))
```

```
['panda.n.01',
 'carnivore.n.01',
 'placental.n.01',
 'mammal.n.01',
 'vertebrate.n.01',
 'chordate.n.01',
 'animal.n.01',
 'organism.n.01',
 'living_thing.n.01',
 'whole.n.02',
 'object.n.01',
 'physical_entity.n.01',
 'entity.n.01']
```

**Issues:**
- newly-invented words
- subjective
- annotation effort
- difficult to compute word similarity
Meaning Representations in Computers

Knowledge-based representation

Corpus-based representation

✓ Atomic symbol
✓ Neighbors
  ◦ High-dimensional sparse word vector
  ◦ Low-dimensional dense word vector
    ▪ Method 1 – dimension reduction
    ▪ Method 2 – direct learning
Corpus-based representation

Atomic symbols: **one-hot** representation

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \ldots & 0
\end{bmatrix}$$

**car**

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \ldots & 0
\end{bmatrix}$$

**car**

Issues: difficult to compute the similarity (i.e. comparing “car” and “motorcycle”)

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \ldots & 0
\end{bmatrix} \text{ AND } \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0
\end{bmatrix} = 0$$

**car**

**motorcycle**

Idea: words with similar meanings often have similar neighbors
Corpus-based representation

Co-occurrence matrix

- Neighbor definition: full document v.s. windows

**full document**
- word-document co-occurrence matrix gives general topics
  - “Latent Semantic Analysis”

**windows**
- context window for each word
  - capture syntactic (e.g. POS) and semantic information
Meaning Representations in Computers

Knowledge-based representation

Corpus-based representation

✓ Atomic symbol
✓ Neighbors
  ◦ High-dimensional sparse word vector
  ◦ Low-dimensional dense word vector
    ▪ Method 1 – dimension reduction
    ▪ Method 2 – direct learning
Window-based Co-occurrence Matrix

Example

- Window length=1
- Left or right context
- Corpus:

  I love NTU.
  I love deep learning.
  I enjoy learning.

<table>
<thead>
<tr>
<th>Counts</th>
<th>I</th>
<th>love</th>
<th>enjoy</th>
<th>NTU</th>
<th>deep</th>
<th>learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>love</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>enjoy</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>NTU</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>deep</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>learning</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Issues:
- matrix size increases with vocabulary
- high dimensional
- sparsity \(\rightarrow\) poor robustness

Idea: low dimensional word vector
Low-Dimensional Dense Word Vector

Method 1: dimension reduction on the matrix

Singular Value Decomposition (SVD) of co-occurrence matrix $X$

$$X \approx \hat{X}$$
Meaning Representations in Computers

Knowledge-based representation

Corpus-based representation

✓ Atomic symbol
✓ Neighbors
  ◦ High-dimensional sparse word vector
  ◦ Low-dimensional dense word vector
    ▪ Method 1 – dimension reduction
    ▪ Method 2 – direct learning
Low-Dimensional Dense Word Vector

Method 1: dimension reduction on the matrix

Singular Value Decomposition (SVD) of co-occurrence matrix $X$

Issues:
- computationally expensive: $O(mn^2)$ when $n<m$ for $n \times m$ matrix
- difficult to add new words

Idea: directly learn low-dimensional word vectors

Meaning Representations in Computers

Knowledge-based representation

Corpus-based representation

✓ Atomic symbol
✓ Neighbors
  ◦ High-dimensional sparse word vector
  ◦ Low-dimensional dense word vector
    ▪ Method 1 – dimension reduction
    ▪ Method 2 – direct learning
Low-Dimensional Dense Word Vector

Method 2: directly learn low-dimensional word vectors

- Learning representations by back-propagation. (Rumelhart et al., 1986)
- A neural probabilistic language model (Bengio et al., 2003)
- NLP (almost) from Scratch (Collobert & Weston, 2008)
- Recent and most popular models: word2vec (Mikolov et al. 2013) and Glove (Pennington et al., 2014)
Word2Vec

**Idea:** predict surrounding words of each word

**Benefit:** faster, easily incorporate a new sentence/document or add a word to vocab

**Goal:** predict surrounding words within a window of each word

**Objective function:** maximize the log probability of any context word given the current center word

\[ C(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log p(w_{t+j} | w_t) \]

context window (size=m)
Word2Vec

Goal: predict surrounding words within a window of each word

Objective function: maximize the log probability of any context word given the current center word

\[ w_1, w_2, \cdots, w_{t-m}, \cdots, w_{t-1}, \text{\circled{w}_t}, w_{t+1}, \cdots, w_{t+m}, \cdots, w_{T-1}, w_T \]

\[
O(\theta) = \frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log p(w_{t+j} | w_t)
\]

\[
p(o | c) = \frac{\exp(u_o^T v_c)}{\sum_w \exp(u_w^T v_c)}
\]

representation learning via deep learning \(\rightarrow\) called “word embeddings”
Major Advantages of Word Embeddings

Propagate *any* information into them via neural networks

- form the basis for all language-related tasks

The networks, R and Ws, can be updated during model training
Concluding Remarks

Knowledge-based representation

Corpus-based representation

✓ Atomic symbol
✓ Neighbors
  ◦ High-dimensional sparse word vector
  ◦ Low-dimensional dense word vector
    ▪ Method 1 – dimension reduction
    ▪ Method 2 – direct learning