Word embeddings

Mannu Malhotra

Agenda

- Introduction to text classification and the dataset.
- Data representation: Count vectors
- Hands-on on count vectors.
- Data representation: TFIDF
- Hands-on TFIDF.
- Data representation: “Word embeddings” the holy grail.
    - Introduction to word embeddings
    - Learning word embeddings
- Hands-on on word embeddings
- Implement skip gram

A problem on text classification!

-

Representation of text data

- **Counter vectorization**
- **TF-IDF vectorization**
- **Word embeddings**

Count vectorization / One hot encoding

```
            Rome  Paris                  word V
Rome   = [1,    0,    0, 0, 0, 0, …,     0]
Paris  = [0,    1,    0, 0, 0, 0, …,     0]
Italy  = [0,    0,    1, 0, 0, 0, …,     0]
France = [0,    0,    0, 1, 0, 0, …,     0]
```

Counter vectorization:

Hands on

Count vectors

- What is wrong?
  - Some words are useless (stopwords).
  - How relevant a term is to a document?

TF-IDF

- TF: Term frequency
- IDF: Inverse document frequency.

- Importance of a term $\propto$ frequency of the term in a document.
- Importance of a term 1 / $\propto$ frequency of the term in all the document

$$TF(i,j) = \frac{\text{Term i frequency in document j}}{\text{Total words in document j}}$$

$$IDF(i) = \log_2\left(\frac{\text{Total documents}}{\text{documents with term i}}\right)$$

TFIDF

Hands on

Word embeddings:

- What is the problem with existing representations:
  - Sparse representation.
  - No similarity betweens words.
    - Train: The movie is awesome; Test: The movie is good.
  - No featurized representation.

Word embeddings:

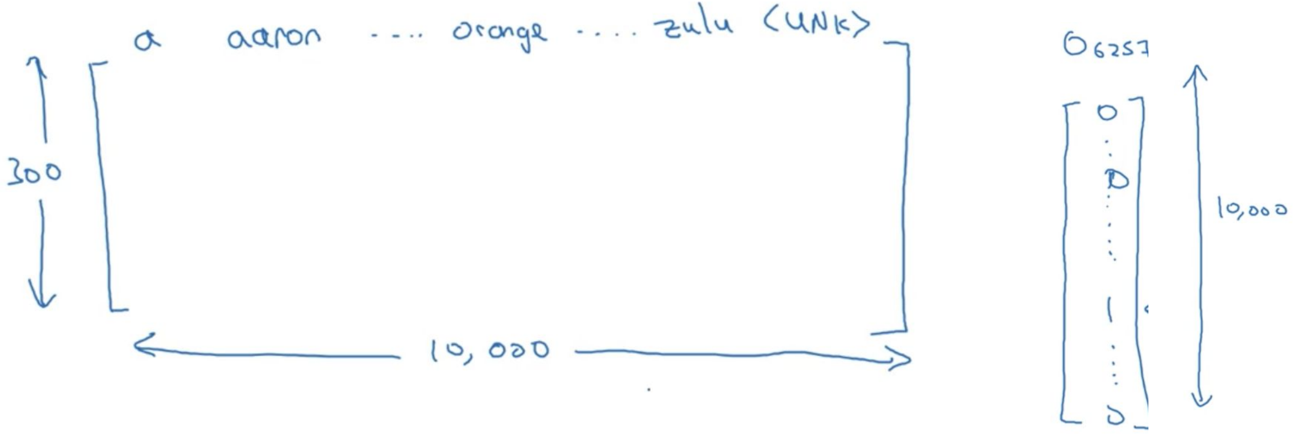
- What are word embeddings:
    - Featurized representation of words.

Featurized representation: word embedding

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97

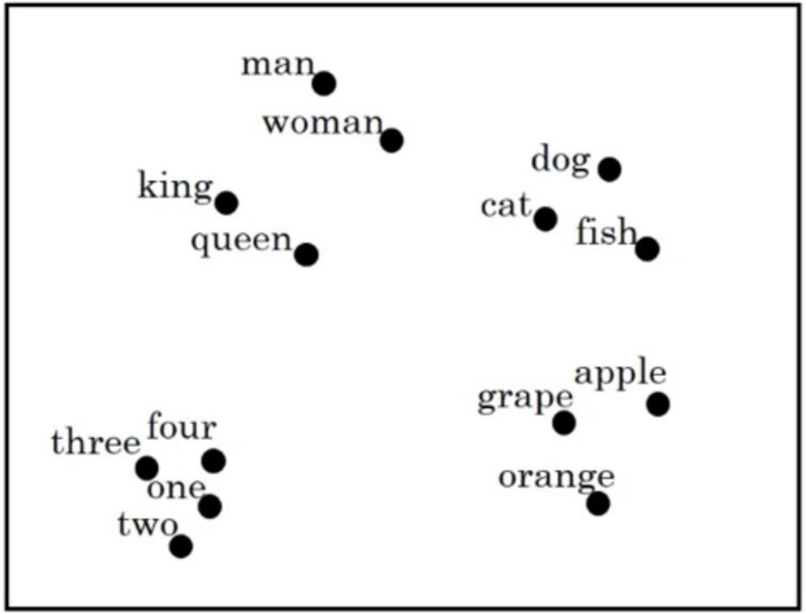
Word Embeddings

- What are word embeddings?
  - A matrix of size |V| * num_of_features
  - Embedding matrix, one-hot vector, embedding vector.
  - Embedding vector = Embedding matrix (dot product) one-hot vector



Visualizing Words Embeddings:

- If we visualize after dimension reduction:



Words Embeddings:

- Train: The movie is awesome; Test: The movie is good.
- Embeddings:
  - Awesome: [.54, .02, .93, .80, …...]
  - Good: [.50, .05, .90, .85, ….]

Word Embeddings: How do we use them

- Transfer learning:
  - Download pre trained word embeddings.
  - Transfer embeddings on the new task with your dataset.

Word Embeddings: How to learn embeddings?

- How do we find the feature representation scores?

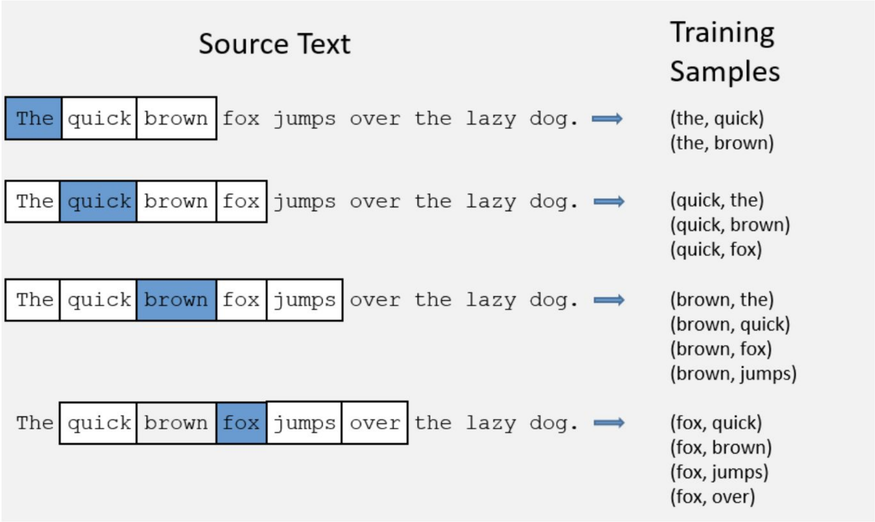
- “A word is known by the company it keeps.”

- Target/context pairs:

I want $\underbrace{\text{a glass of orange}}_{\text{context}}$ $\underset{\text{target}}{\underline{\text{juice}}}$ to go along with my cereal.

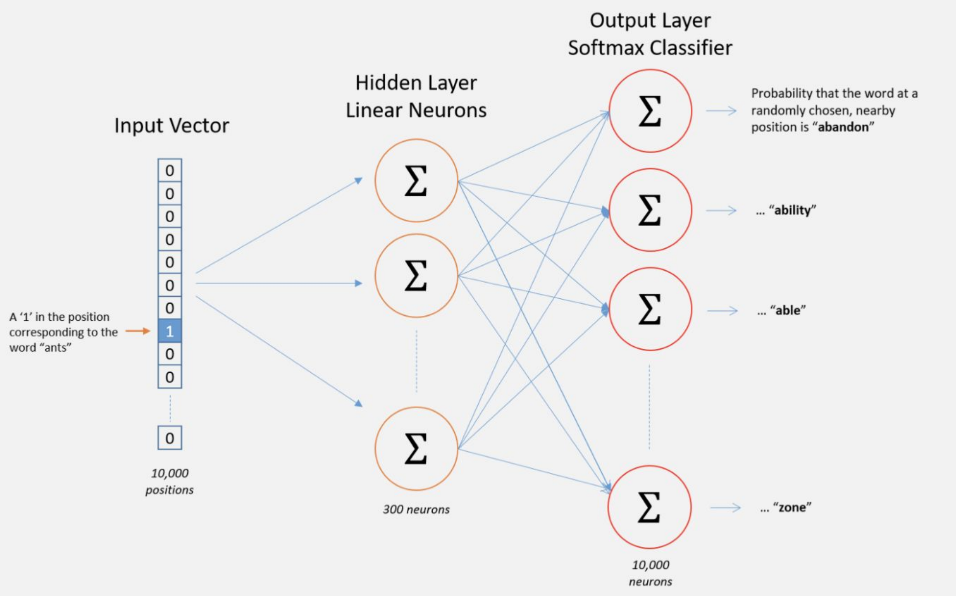
Word embeddings: skip gram

- Step 1: Create source-target pairs.



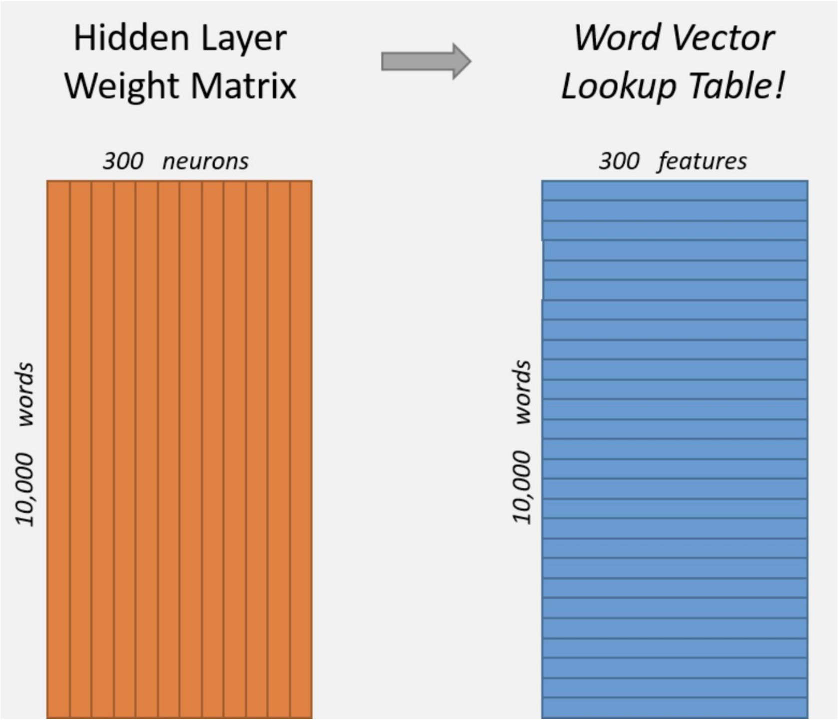
Word embeddings: skip gram

- Step 2: Convert to it one hot encoding.
- Step 3: Feed it to two layers NN: A multiclass classification problem.
  - Hidden layer: 300 neurons
  - Softmax layer: |V| neurons
- Step 4: Train the NN with source-target pairs.
  - Intitution: Ask the model to predict target given source word.
- Wait? What is 300 here?
  - Features: latent variables



Word embeddings

- Weights of the hidden layer: |v| * 300
- Voilà!! We got the embeddings matrix.



Word Embeddings: Problem with Skip Gram

- Softmax for |v| units can be really slow!
- Too many weights:
  - |V| = 10,000, latent variables = 300.
  - # of weights = 3 million weights.
- Gradient descent can be really slow.
  - # of training samples are in billions.
  - Expensive softmax operations.

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^\top v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^\top v_{w_I}\right)}$$

Word Embeddings: Negative sampling

- Idea: Divide 10000 class softmax to 10000 binary classification problem.
- Step 1: Create 1 positive + k negative samples.

I want a glass of orange juice to go along with my cereal.

Context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

- Step 2: For every pair train K+1 binary classifier.

Word embeddings

- But wait the dimension of the data representation increased.
- How do we input it to the model now?
- Two ways:
    - Mean embeddings.
    - IDF embeddings.

$$X_{mean-embeddings} = \frac{1}{|document|} \sum_{word}^{document} W2V[word]$$

$$X_{TFIDF-embeddings} = \frac{1}{|document|} \sum_{word}^{document} W2V[word] * IDF[word]$$

- Which one to use when?
    - Mean: Good for documents with high intra document similarity and low inter document similarity.
    - IDF: Good for documents with low intra document similarity and high inter document similarity.

Word embeddings

Hands on