Mathematical structures for word embeddings

Siddharth Bhat

IIIT Hyderabad

October 23th, 2021

What is a word embedding?

What is a word embedding?

- Map words to *mathematical objects*.

What is a word embedding?

- Map words to *mathematical objects*.
- Semantic ideas on words $\simeq$ mathematical operations on these objects.

What is a word embedding?

- Map words to *mathematical objects*.
- Semantic ideas on words $\simeq$ mathematical operations on these objects.
- Most common: *vector embeddings* (`word2vec`)

What is a word embedding?

- Map words to *mathematical objects*.
- Semantic ideas on words $\simeq$ mathematical operations on these objects.
- Most common: *vector embeddings* (`word2vec`)

What's word2vec?

```python
def train(corpus: list, DIMSIZE: int):
    """
    train word2vec of dimension DIMSIZE on the given corpus (list of words).
    Eg:train(["the", "man", "was" "tall", "the", "quick", "brown", "fox"], 20)
    """
    vocab = set(corpus); VOCABSIZE = len(vocab)
    # map each unique word to an index for array indexing.
    vocab2ix = dict([(word, ix) for (ix, word) in enumerate(corpus)])
    # +ve and -ve sample vectors.
    # +ve vectors are random initialized, -ve vectors are zero initialized
    poss = np.rand((VOCABSIZE, DIMSIZE)); negs = np.zeros((VOCABSIZE, DIMSIZE))

    for wix in range(len(corpus)): # for every location in the corpus
      w = vocab2ix[corpus[wix]]  # find word at location,
      l = max(wix-WINDOWSIZE, 0); r = min(wix+WINDOWSIZE, len(corpus)-1) # take a window

      for w2ix in range(l, r+1): # word in window
          w2 = vocab2ix[corpus[w2ix]] # prallel.
          learn(l=poss[w], r=negs[w2], target=1.0)

      for _ in range(NNEGSAMPLES): # random words outside window.
          w2ix = random.randint(0, len(corpus)-1) # random word.
          w2 = vocab2ix[corpus[w2ix]]
        learn(l=poss[w], r=negs[w2], target=0.0) # perpendicular
    return { v: poss[vocab2ix[v]] for v in vocab }
```

What's word2vec?

```
def learn(l: np.array, r:np.array, target: float):
  """
  gradient descent on
  loss = (target - dot(l, r))^2 where l = larr[lix]; r = rarr[rix]
  """
  dot = np.dot(l, r); grad_loss = 2 * (target - out)
  #dloss/dl = 2 * (target - dot(l, r)) r
  #dloss/dr = 2 * (target - dot(l, r)) l
  lgrad =  EPSILON * grad_loss * r; rgrad =  EPSILON * grad_loss * l
  # l -= eps * dloss/dl; r -= eps * dloss/dr
  l += EPSILON * grad_loss * r;
  r += EPSILON * grad_loss * l

def train(corpus: list, DIMSIZE: int):
    for w2ix in range(l, r+1): # positive samples, parallell
        w2 = vocab2ix[corpus[w2ix]] # word in window
        learn(l=poss[w], r=negs[w2], target=1.0)
    for _ in range(NNEGSAMPLES): # negative samples: perpendicular.
        w2ix = random.randint(0, len(corpus)-1) # random word outside window.
        learn(l=poss[w], r=negs[w2], target=0.0) # perpendicular
```

Using `word2vec`

- Dot products capture similarity.

Using word2vec

- Dot products capture similarity.
- nope! *cosine similarity* captures similarity: $v \cdot w / |v||w|$.
- Vector space structure captures analogy: $\mathtt{king} - \mathtt{man} + \mathtt{woman} = \mathtt{queen}$. [Analogy]

Using `word2vec`

- Dot products capture similarity.
- nope! *cosine similarity* captures similarity: $v \cdot w / |v||w|$.
- Vector space structure captures analogy: $\mathtt{king} - \mathtt{man} + \mathtt{woman} = \mathtt{queen}$. [Analogy]
- nope! $\mathit{normalize}(\hat{\mathtt{king}} - \hat{\mathtt{man}} + \hat{\mathtt{woman}}) = \hat{\mathtt{queen}}$
- `word2vec`"vectors" are always normalized!

Using `word2vec`

- Dot products capture similarity.
- nope! *cosine similarity* captures similarity: $v \cdot w / |v||w|$.
- Vector space structure captures analogy: $\mathtt{king} - \mathtt{man} + \mathtt{woman} = \mathtt{queen}$. [Analogy]
- nope! *normalize*$(\hat{\mathtt{king}} - \hat{\mathtt{man}} + \hat{\mathtt{woman}}) = \hat{\mathtt{queen}}$
- `word2vec`"vectors" are always normalized!
- Cannot add, substract, scale them. So in what sense is the embedding "vectorial"?

Using `word2vec`

- Dot products capture similarity.
- nope! *cosine similarity* captures similarity: $v \cdot w / |v||w|$.
- Vector space structure captures analogy: $\texttt{king} - \texttt{man} + \texttt{woman} = \texttt{queen}$. [Analogy]
- nope! $\textit{normalize}(\hat{\texttt{king}} - \hat{\texttt{man}} + \hat{\texttt{woman}}) = \hat{\texttt{queen}}$
- `word2vec`"vectors" are always normalized!
- Cannot add, substract, scale them. So in what sense is the embedding "vectorial"?
- In the sense that we have "vectors" — elements of the space $[-1, 1]^N$ with a normalization condition $(\sum_i x_i^2 = 1)$.

Using word2vec

- Dot products capture similarity.
- nope! *cosine similarity* captures similarity: $v \cdot w / |v||w|$.
- Vector space structure captures analogy: $\texttt{king} - \texttt{man} + \texttt{woman} = \texttt{queen}$. [Analogy]
- nope! $\textit{normalize}(\hat{\texttt{king}} - \hat{\texttt{man}} + \hat{\texttt{woman}}) = \hat{\texttt{queen}}$
- word2vec"vectors" are always normalized!
- Cannot add, substract, scale them. So in what sense is the embedding "vectorial"?
- In the sense that we have "vectors" — elements of the space $[-1, 1]^N$ with a normalization condition $(\sum_i x_i^2 = 1)$.
- Can we ascribe a *different* meaning to these "vectors"?

Part I: What's a philosopher to do?

- Montague semantics: The *meaning* of a word is the *set* of possible worlds where the meaning holds true.

Part I: What's a philosopher to do?

- Montague semantics: The *meaning* of a word is the *set* of possible worlds where the meaning holds true.
- A mathematical analogy: The *meaning* of an expression $\forall x \in \mathbb{Z}, x \leqslant 2$ is the *set* of possible values where the meaning holds true: $(-\infty, 2] = \{x \in \mathbb{Z} : x \leqslant 2\}$.

Part I: What's a philosopher to do?

- Montague semantics: The *meaning* of a word is the *set* of possible worlds where the meaning holds true.
- A mathematical analogy: The *meaning* of an expression $\forall x \in \mathbb{Z}, x \leqslant 2$ is the *set* of possible values where the meaning holds true: $(-\infty, 2] = \{x \in \mathbb{Z} : x \leqslant 2\}$.
- Meaning $\simeq$ subsets. Is `word2vec` subsets?

Part I: What's a philosopher to do?

- Montague semantics: The *meaning* of a word is the *set* of possible worlds where the meaning holds true.
- A mathematical analogy: The *meaning* of an expression $\forall x \in \mathbb{Z}, x \leqslant 2$ is the *set* of possible values where the meaning holds true: $(-\infty, 2] = \{x \in \mathbb{Z} : x \leqslant 2\}$.
- Meaning $\simeq$ subsets. Is `word2vec` subsets? Yes, *fuzzy sets*.

Part I: What's a philosopher to do?

- Montague semantics: The *meaning* of a word is the *set* of possible worlds where the meaning holds true.
- A mathematical analogy: The *meaning* of an expression $\forall x \in \mathbb{Z}, x \leqslant 2$ is the *set* of possible values where the meaning holds true: $(-\infty, 2] = \{x \in \mathbb{Z} : x \leqslant 2\}$.
- Meaning $\simeq$ subsets. Is `word2vec` subsets? Yes, *fuzzy sets*.
- Set: binary membership. $(1 \in_? \{1, 2\} = T$, $3 \notin_? \{1, 2\} = F)$.

Part I: What's a philosopher to do?

- Montague semantics: The *meaning* of a word is the *set* of possible worlds where the meaning holds true.
- A mathematical analogy: The *meaning* of an expression $\forall x \in \mathbb{Z}, x \leqslant 2$ is the *set* of possible values where the meaning holds true: $(-\infty, 2] = \{x \in \mathbb{Z} : x \leqslant 2\}$.
- Meaning $\simeq$ subsets. Is `word2vec` subsets? Yes, *fuzzy sets*.
- Set: binary membership. ($1 \in_? \{1,2\} = T$, $3 \notin_? \{1,2\} = F$).
- Fuzzy set: probabilistic membership. ($1 \in_{fuz} F = 0.1$, $2 \in_{fuz} F = 0.5$).

The hidden sets in `word2vec`

- Given the set of vectors, normalize the ith component of the vector across *all vectors*.

The hidden sets in `word2vec`

- Given the set of vectors, normalize the ith component of the vector across *all vectors*.
- $\texttt{fuzembed}_{\texttt{word}}[i] \equiv \texttt{vecembed}_{\texttt{word}}[i] / \sum_{w \in \texttt{CORPUS}} \texttt{vecembed}_{w}[i]$.

The hidden sets in `word2vec`

- Given the set of vectors, normalize the ith component of the vector across *all vectors*.
- $\texttt{fuzembed}_{\texttt{word}}[i] \equiv \texttt{vecembed}_{\texttt{word}}[i] / \sum_{w \in \texttt{CORPUS}} \texttt{vecembed}_{w}[i]$.
- Fuzzy set embedding from `word2vec` embeddings.

The hidden sets in `word2vec`

- Given the set of vectors, normalize the ith component of the vector across *all vectors*.
- $\texttt{fuzembed}_{\texttt{word}}[i] \equiv \texttt{vecembed}_{\texttt{word}}[i] / \sum_{w \in \texttt{CORPUS}} \texttt{vecembed}_{w}[i]$.
- Fuzzy set embedding from `word2vec` embeddings.

What does this buy us anyway? (Set operations)

$$(A \cap B)[i] \equiv A[i] \times B[i] \quad \text{(set intersection)}$$

$$(A \cup B)[i] \equiv A[i] + B[i] - A[i] \times B[i] \, \text{(set union)}$$

$$(A \sqcup B)[i] \equiv \max(1, \min(0, A[i] + B[i])) \, \text{(disjoint union)}$$

$$(\neg A)[i] \equiv 1 - A[i] \quad \text{(complement)}$$

$$(A \setminus B)[i] \equiv A[i] - \min(A[i], B[i]) \quad \text{(set difference)}$$

$$(A \subseteq B) \equiv \forall x \in \Omega : \mu_A(x) \leqslant \mu_B(x) \, \text{(set inclusion)}$$

$$|A| \equiv \sum_{i \in \Omega} \mu_A(i) \quad \text{(cardinality)}$$