Word Embedding

Word Representation

① Words

② One-hot $\vec{\text{Word 1}} \cdot \vec{\text{Word 2}} = 0$ No Relationship

But Meaning Underlying

③ Featurized Representation: Word Embeddings

Features \ Words	Word 1	Word 2	...	Word N
Gender				
Age				
Fruit				
Food				
Size				
	e_1	e_2	...	e_N

n

e- embedding

e_i $1 \times n$ Vector

Small Dataset

Training $\xrightarrow[\text{Word Embeddings}]{\text{Generalize Well}}$ Test

Rare Words

Transfer Learning & Word Embeddings

Transfer Learning

① Learn word embeddings from Large Corpus

② Transfer embedding to new task with smaller training set.

(Optional: Continue to finetune word embeddings if dataset sufficiently large

A smaller feature space than one-hot encoding.

Properties of Word Embeddings

Analogies in Original High-D Space (t-SNE might break it)

Man is to king as Woman is to <u>Queen</u>.

Find e_{Queen} to maximize similarity

between e_{Queen} and $e_{king} - e_{man} + e_{woman}$

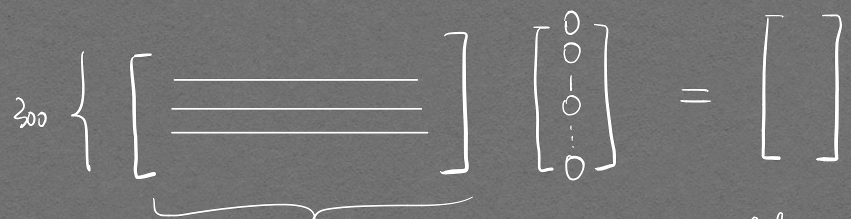
$$\arg\max_{e_{Queen}} \ sim(e_{Queen}, e_{king} - e_{man} + e_{woman})$$

$sim(u, v)$

1. Cosine similarity $\quad \dfrac{u^T v}{\|u\|_2 \|v\|}$
2. euclidean distance $\quad \|u - v\|^2$

Learn Embeddings

Embedding Matrix



10,000 Words

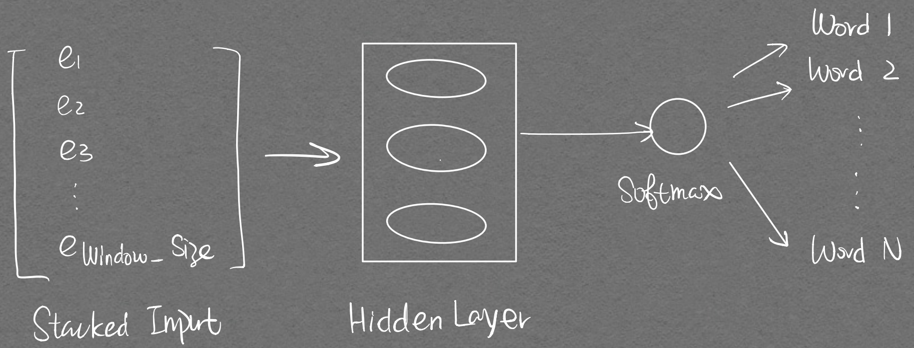
One_Hot Encoder

Select out one embedding

$E \cdot O_j = e_j$ (embedding for word j)

In practice, use specialized functions rather than matrix multiplication to get e_j.

Neural Probabilistic Model



Other Context / Target Pairs

Context

① last 4 words

② 4 words on left & right

③ last 1 word

④ nearby word

Skip-grams Model

Given Context

Predict Target

Word2Vec

Model Vocab Size = 10k

Context	Mapping	Target
(Orange	$\longrightarrow$	Juice)
(Orange	$\longrightarrow$	Glass)
(Orange	$\longrightarrow$	my)

o

$$c \rightarrow E \rightarrow e_c \rightarrow \bigcirc \rightarrow \hat{y}$$

$e_c = E \cdot O_c$

Softmax

$\downarrow$

$$P(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10k} e^{\theta_j^T e_c}}$$

θ_t Parameter associated with output t

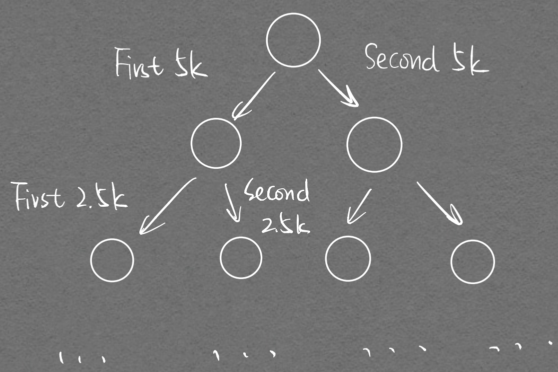
Loss Function

$$\mathcal{L}(\hat{y}, y) = -\sum_{i=1}^{10{,}000k} y_i \log \hat{y}_i$$

Computationally Slow

Due to the sum term

Hierarchical Softmax



Use Common Words groups first

Uncommon Words groups later.

Define a new learning problem

Pick A Context Word	Pick A Target Word		Target?
Orange	Juice		1
k times Orange	$\boxed{\text{Book}}$	Negative	0
Orange	$\boxed{\text{king}}$	Sampling	0
c	t		y

$$P(w_i) = \frac{f(w_i)^{\frac{3}{4}}}{\sum_{j=1}^{10{,}000k} f(w_j)^{\frac{3}{4}}}$$

$f \rightarrow$ frequency

$k = [5, 20]$ smaller datasets

$k = [2, 5]$ large datasets

$$P(y=1 \mid c, t) = \sigma(\theta_t^T e_c)$$

10k Binary Logistic Classification

Pick $(k+1)$ Logistic Classifier to train on every iteration

Glove Word Vectors (Global Vectors for word representation)

$$\min \sum_{i=1}^{10k} \sum_{j=1}^{10k} \underbrace{f(X_{ij})}_{\text{Weighting Factor}} \left(\theta_i^T e_j + b_i + b_j' - \log \underline{X_{ij}} \right)^2$$

" $\theta_t^T e_c$ "

Weighting Factor

Cooccurance Matrix

$f(X_{ij}) = 0$ for $X_{ij} = 0$

θ_i, e_j are symmetric

$$e_w^{(find)} = \frac{e_w + \theta_w^T}{2}$$

Pennington et al. 2014 Glove : Global Vectors for word representation

Word Embeddings are not guaranteed to unique

each dimension is not necessarily interpretable.

$$\theta_i^T e_j = (A\theta_i)^T (A^{-T} e_j)$$

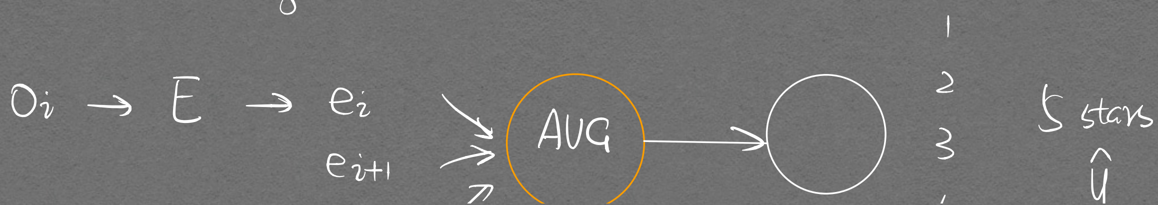
$$= \theta_i^T A^T A^{-T} e_j$$

Sentiment Analysis

Problem: no large training set with labels

10k, 100k words

Use Embedding Matrix E trained on 100B words



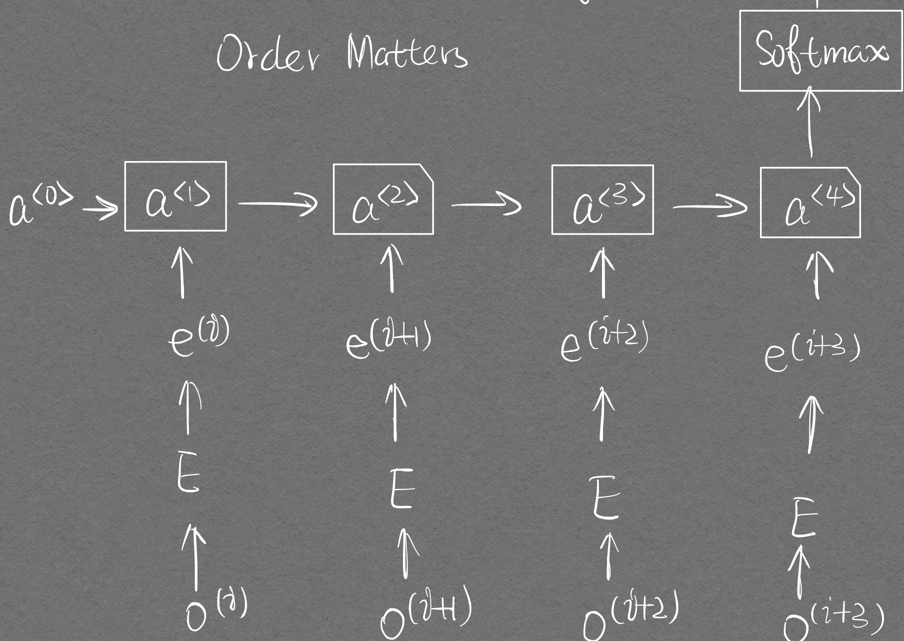
e_{i+2} / Problematic SoftMax 4 5

Ignores word order

RNN Sentiment Classifier

Order Matters

$\hat{y}$



Debiasing Word Embeddings

Man : Computer Programmer

Women : Homamaker

Gender Bias

Father : Doctor

Mother : Nurse

① Identify Bias Direction

$$\text{AVG}\left(e_{he} - e_{she} \right.$$

$$\begin{pmatrix} e_{male} - e_{female} \\ e_{boy} - e_{girl} \end{pmatrix}$$

train a classifier

② Neutralize :

For every word that is not definitional

Project to get rid of bias

③ Equalization

Equalize Pairs