NOTES (LEC-1)

WordNet

- Word meaning and word relationships
- a thesaurus.
- Problems
  - Impossible to update with possible new words.
  - Can't compute accuracy for word similarity

Representing words as discrete symbols

- Until 2012, localist representation of words were used for Neural Nets

  motel = [0 0 0 0 1 0 0 0]

  hotel = [0 0 0 1 0 0 0 0]

- Bad → The above two vector doesn't represent a relationship
- Sol^n → Make vectors in a way that they express similarity
  - Problem - LOTS OF WORDS (xx o) !!!

Representing words by their context

- Distributional semantics: A word's meaning is given by neighboring words
- Window size - No. of words around the focused words..

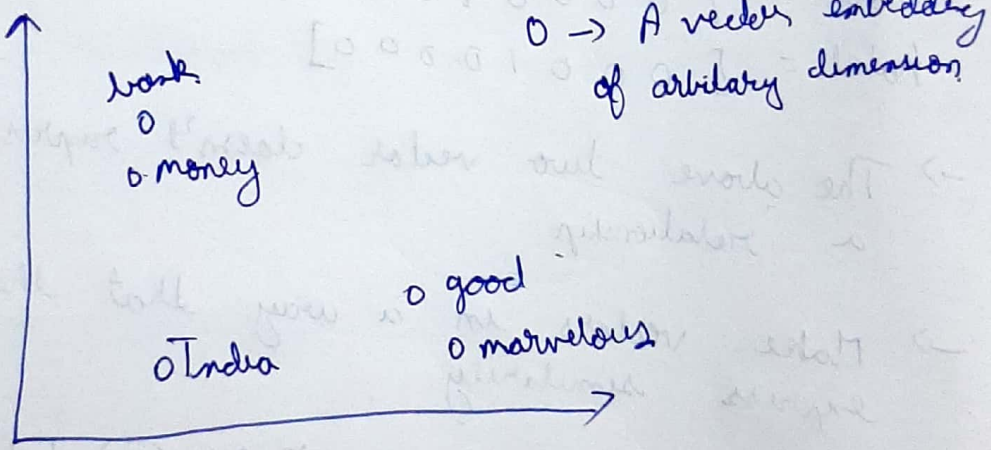
Distributed representation of words in vector

→ It is based on context words.

banking = $\begin{bmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -1.07 \\ \vdots \end{bmatrix}$

- Word vectors also called word embedding

Word Vector Space



O → A vector embedding of arbitary dimension

. We can see similarities bet" words.

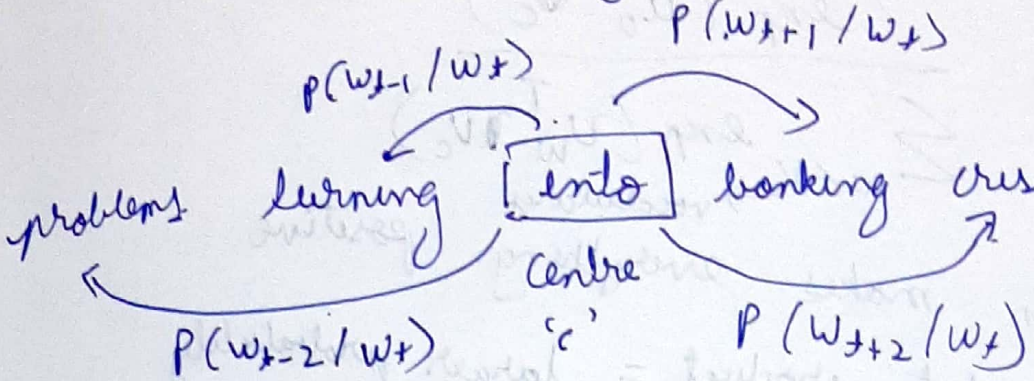
WORD2VEC

→ Large corpa of text

→ words are represented by vectors

→ Go through each position 't' in the text, which has a centre word 'c' and context words (other) 'o'

Using the similarity between 'c' and 'o' figuring out 'o' given 'c' probability



Likelihood $L(\theta) = \prod_{t=1}^{T} \prod_{-m \le j \le m, j \neq 0} P(w_{t+j} / w_t; \theta)$

θ : parameters.
T : text (list of words)

Objective function $J(\theta) = -\frac{1}{T} \log L(\theta)$

$$= -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \neq 0}} \log P(w_{t+j} / w_t; \theta$$

Minimizing objective function
⇓
Maximizing predictive accuracy

3. How to calc $P(w_{t+j} / w_t; \theta)$

We use two vectors per word w_i:

- v_w : ~~centre word~~ when w is centre word
- u_w : when w is context word

Prediction function

$$P(O/C) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(v_w^T v_c)}$$

$V \rightarrow$ vocabulary

- Exponential makes everything positive
- Larger dot product = larger probability
- It's a softmax function

Optimization of parameters

→ θ : represents all model parameters

→ Every word has two vectors

→ Optimize the params by walking down the hill

* → Carrying down the hill is walking down the gradient till we reach destination.

→ First random initialisation of vector for every word is done.

→ They are updated by training

To minimize the J^n

$$\frac{\partial}{\partial v_c} \log\left(\frac{\exp(u_o^T v_c)}{\sum_{w=1}^{V} \exp(u_o^T v_c)}\right)$$

$$\Rightarrow \frac{\partial}{\partial v_c} \log \exp(u_o^T v_c) - \frac{\partial}{\partial v_c} \log \sum_{w=1}^{V} \exp(u_o^T v_c)$$

$$\Rightarrow \frac{\partial}{\partial v_c} u_o^T v_c - \frac{\partial}{\partial v_c} \log \sum_{w=1}^{V} \exp(u_o^T v_c)$$

$$\Rightarrow u_o - \frac{1}{\sum_{w=1}^{V} \exp(u_o^T v_c)} \cdot \frac{\partial}{\partial v_c} \sum_{x=1}^{V} \exp(u_x^T v_c)$$

$$\Rightarrow u_o - \frac{1}{\sum_{w=1}^{V} \exp(u_o^T v_c)} \sum_{x=1}^{V} u_x \exp(u_x^T v_c)$$

$$\Rightarrow u_o - \boxed{\sum_{x=1}^{V} \frac{\exp(u_x^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)} \cdot u_x}$$

$p(x|c)$

$$\Rightarrow u_o - \sum_{x=1}^{V} p(x|c) \cdot u_x \quad \rightarrow \text{Slope}$$

u_o — actual word

— of expected context word.