L8 Word Vectors

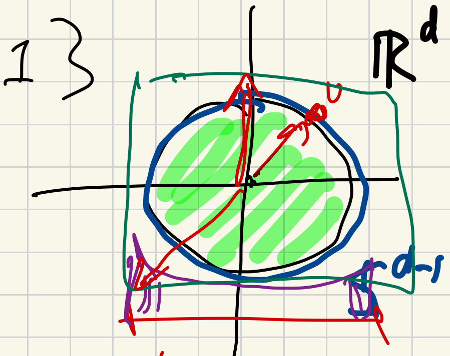
+ Self-Supervised Learning

Feb 3, 2025

Data Mining:
Jeff M. Phillips

Generate set iid $u_1, u_2, \dots u_t \sim \mathrm{Unif}(S^{d-1})$

$$S^{d-1} = \{ x \in \mathbb{R}^d \text{ s.t. } \|x\| = 1 \}$$



Step 1: generate $v \in [-1,1]$

$v \in \mathrm{Unif}(0,1)$

$= 0.b_1 \, b_2 \, b_3 \dots b_{\log N}$

$\{0,1\}$

$u = (u(1), u(2) \dots, u(d))$

Generate angle $\theta \sim \mathrm{Unif}(0, 2\pi)$

works $d=2$, hard for $d > 3$

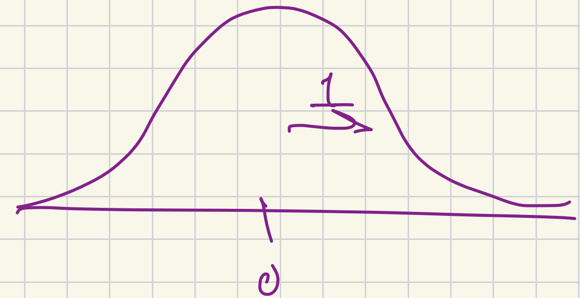
Generate $v_1, v_2 \dots v_d \sim [\mathrm{Unif}(0,1)]^d \Rightarrow$ <u>normalize</u>

Generate points $g_1, g_2, \dots g_d$ $\boxed{\underset{iid}{\sim} N(0,1)}$

normal

vector $g = (g_1 \dots g_d)$

return $u = g / \|g\|$



Box-Mueller Transform

$$N(0,1) = g(x) = \frac{1}{\sqrt{2\pi}} e^{-\|x\|^2/2}$$

$u_1, u_2 \sim \text{Unif}(0,1)$

$$g_1 = \sqrt{-2\ln(u_1)}\,\cos(2\pi u_2) \sim N(0,1)$$

$$g_2 = \sqrt{-2\ln(u_1)}\,\sin(2\pi u_2) \sim N(0,1)$$

Word Embeddings

subject

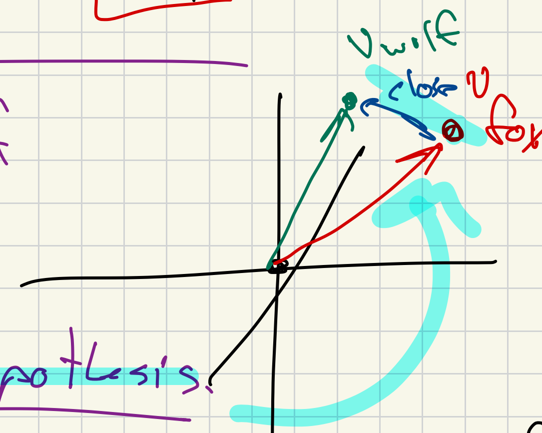
A quick brown $\boxed{\text{fox}}$ jumped over the lazy dog.

verb

Newer map each word $\boxed{\text{fox}}$

Premise: Words are similar if they appear in similar context.

→ Distribution Hypothesis.



$\mathbb{R}^d$

$d=300$

First Gen   PPMI vectors.   CBOW

A quick brown $\boxed{\text{fox}}$ jumped over a lazy dog

- context window
  - k words on each side
  - k words before $\boxed{\text{word}}$

aggregate bag of words : # instances of
j = jumped in any context window of i = $\boxed{\text{fox}}$

b_{ij} store $B = m \times m$ $B_{ij} = b_{ij}$ → $m = 100,000$

possible word in context. jumped

word in question $\boxed{\text{fox}}$

Is frequency-context vector

$$b_i = (b_{i1}, b_{i2}, \dots b_{im}) \in \mathbb{R}^m$$

a good representation?

$$p(i,j) = \frac{b_{ij}}{N} \qquad N = \#\text{ words in corpus}$$

$$p(i) = \frac{n_i}{N} \qquad n_i = \#\text{ times word } i \text{ occurs}$$

$$v_{ij} = \max \left\{ \log \left(\frac{p(i,j)}{p(i) \cdot p(j)} \right), 0 \right\}$$

PPMI vec

$$V_i = (v_{i,1}, v_{i,2}, \dots v_{i,m}) \in \mathbb{R}^m$$

useful vector rep word i.

Second Generation

want vectors $v_i \in \mathbb{R}^d$

$d = 300$.

work like PPMI vectors

1. Get PPMI vectors $\mathbb{R}^m$

2. Dim Reduction $\mathbb{R}^m \to \mathbb{R}^d$.

Better idea: learn mapping to $\mathbb{R}^d$

using Self-supervised learning

→ Word2vec, GloVe 2013!

Self-Supervised Learning

Supervised Learning

Input $X_{xy} = \{(x_i, y_i)\}_i$

$x_i \in \mathcal{X} (\subseteq \mathbb{R}^d)$

$y_i \in \{-1, +1\}$ classific.

$\in \mathbb{R}$ regression

Labels.

Set $X = \{x_1 \dots x_n\}$

independent variables

↳ make prediction

x_i on y_i

$f(x_i) = y_i$

learns

$y = (y_1, y_2 \dots y_n)$

$X = \{x_1, x_2, \dots x_n)$

$\in \mathbb{R}^{n \times d}$

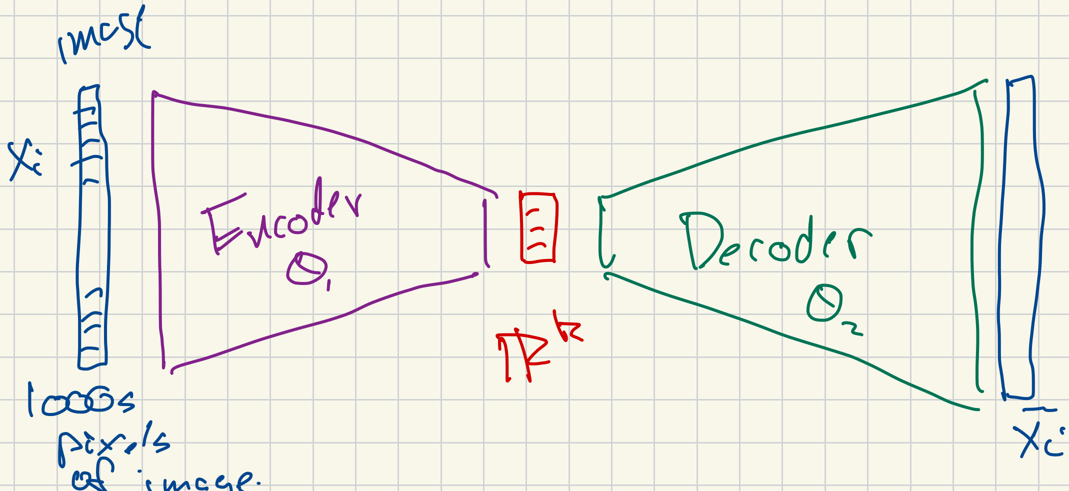
Problem

labels hard to get.

Self-Supervised learning

Train on data x_i to predict itself, or part of itself.

Ex. Auto-encoder



Goal

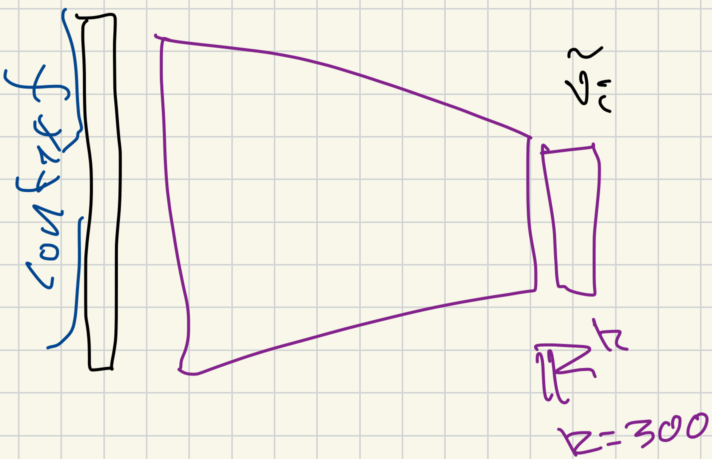
$\sum_i \|x_i - \tilde{x}_i\|^2$ small

→ often blurry.

For text: masked model

A quick brown $\boxed{\text{fox}}$ jumped over the lazy dog

$\underbrace{\text{A quick brown}}$ $\boxed{}$ $\underbrace{\text{jumped over the}}$ ...



Data set

$v_1, v_2, \ldots, v_m \in \mathbb{R}^{300}$

Guess $\quad i = \underset{i}{NN}(\tilde{v}_i)$

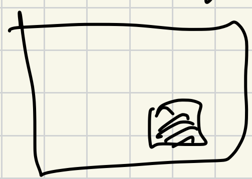
context window fox

next word prediction.

↳ applies other data

graph embedding

images



spatial data

Third generation

ELMo (OWash, AI), BERT → RoBERTa (google)

① Contextual Embeddings

homonyms: apple → fruit
→ company.

ELMo: learn function f_θ[context + word] → $\mathbb{R}^d$

② Transformer architecture, w/ attention

much larger context.

selected most influential words in context.