Neural networks

Natural language processing - word representations

NATURAL LANGUAGE PROCESSING

Topics: one-hot encoding

- The major problem with the one-hot representation is that it is very high-dimensional
  - the dimensionality of $\mathrm{e}(w)$ is the size of the vocabulary
  - a typical vocabulary size is $\approx 100\ 000$
  - a window of 10 words would correspond to an input vector of at least $1\ 000\ 000$ units!
- This has 2 consequences:
  - vulnerability to overfitting
    - millions of inputs means millions of parameters to train in a regular neural network
  - computationally expensive
    - not all computations can be sparsified (ex.: reconstruction in autoencoder)

WORD REPRESENTATIONS

Topics: continuous word representation

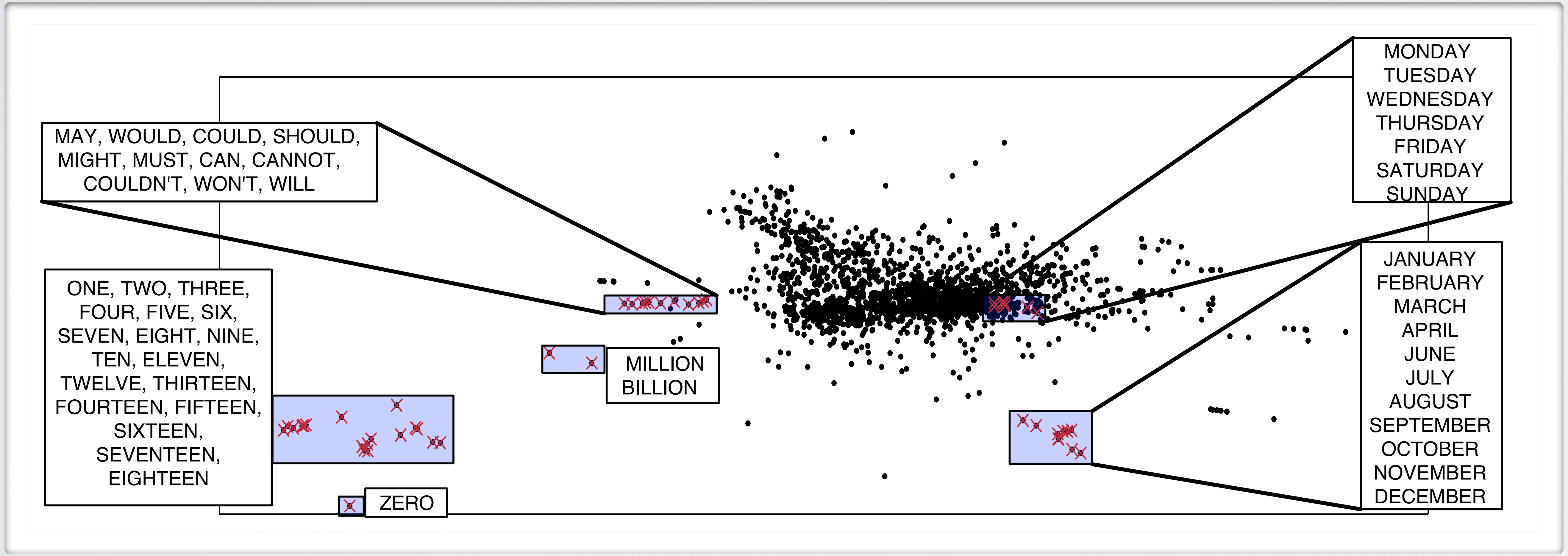
- Idea: learn a continuous representation of words
  - each word w is associated with a real-valued vector $C(w)$

Word	w	$C(w)$
" the "	1	[0.6762, -0.9607, 0.3626, -0.2410, 0.6636]
" a "	2	[0.6859, -0.9266, 0.3777, -0.2140, 0.6711]
" have "	3	[0.1656, -0.1530, 0.0310, -0.3321, -0.1342]
" be "	4	[0.1760, -0.1340, 0.0702, -0.2981, -0.1111]
" cat "	5	[0.5896, 0.9137, 0.0452, 0.7603, -0.6541]
" dog "	6	[0.5965, 0.9143, 0.0899, 0.7702, -0.6392]
" car "	7	[-0.0069, 0.7995, 0.6433, 0.2898, 0.6359]
...	...	...

WORD REPRESENTATIONS

Topics: continuous word representation

- Idea: learn a continuous representation of words
  - we would like the distance $||C(w)-C(w')||$ to reflect meaningful similarities between words



(from Blitzer et al. 2004)

WORD REPRESENTATIONS

Topics: continuous word representation

- Idea: learn a continuous representation of words
  - we could then use these representations as input to a neural network
  - to represent a window of 10 words $[w_1, \ldots, w_{10}]$, we concatenate the representations of each word

$$\mathbf{x} = [C(w_1)^\top, \ldots, C(w_{10})^\top]^\top$$

- We learn these representations by gradient descent
  - we don't only update the neural network parameters
  - we also update each representation $C(w)$ in the input $\mathbf{x}$ with a gradient step

$$C(w) \Longleftarrow C(w) - \alpha \nabla_{C(w)} l$$

where l is the loss function optimized by the neural network

WORD REPRESENTATIONS

Topics: word representations as a lookup table

- Let $\mathbf{C}$ be a matrix whose rows are the representations $C(w)$
  - obtaining $C(w)$ corresponds to the multiplication $\mathrm{e}(w)^\top\ \mathbf{C}$
  - view differently, we are projecting $\mathrm{e}(w)$ onto the columns of C
    - this is a reduction of the dimensionality of the one-hot representations $\mathrm{e}(w)$
  - this is a continuous transformation, through which we can propagate gradients

- In practice, we implement $C(w)$ with a lookup table, not with a multiplication
  - $C(w)$ returns an array pointing to the w^{th} row of $\mathbf{C}$