# Continuous bag of words model

- The methods that we have seen so far are called **count based models** because they use the co-occurrence counts of words
- We will now see methods which directly **learn** word representations (these are called **(direct) prediction based models**)

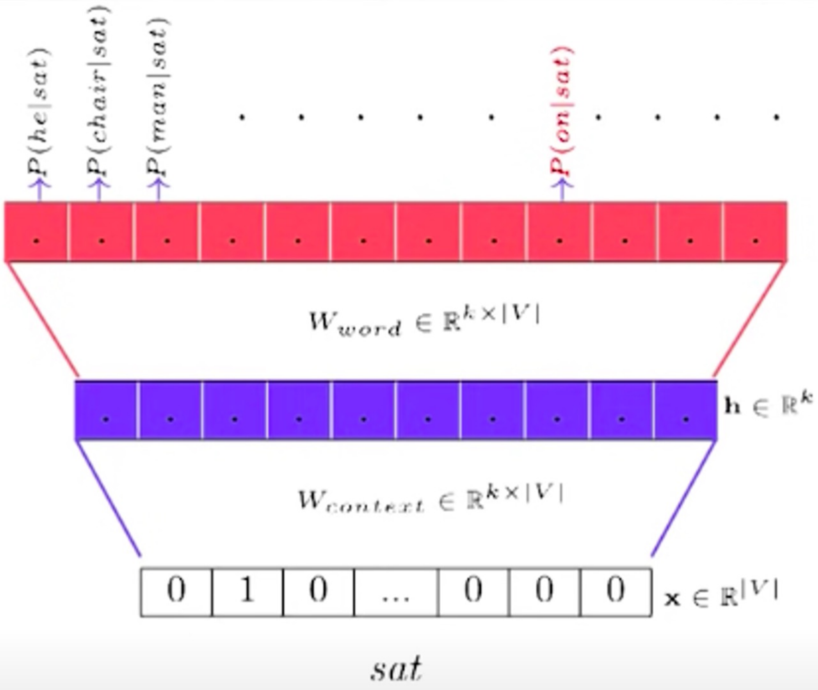
*Sometime in the 21st century, Joseph Cooper, a widowed former engineer and former NASA pilot, runs a farm with his father-in-law Donald, son Tom, and daughter Murphy. It is post-truth society (Cooper is reprimanded for telling Murphy that the Apollo missions did indeed happen) and a series of crop blights threatens humanity's survival. Murphy believes her bedroom is haunted by a poltergeist. When a pattern is created out of dust on the floor, Cooper realizes that gravity is behind its formation, not a "ghost". He interprets the pattern as a set of geographic coordinates formed into binary code. Cooper and Murphy follow the coordinates to a secret NASA facility, where they are met by Cooper's former professor, Dr. Brand.*

**Some sample 4 word windows from a corpus**

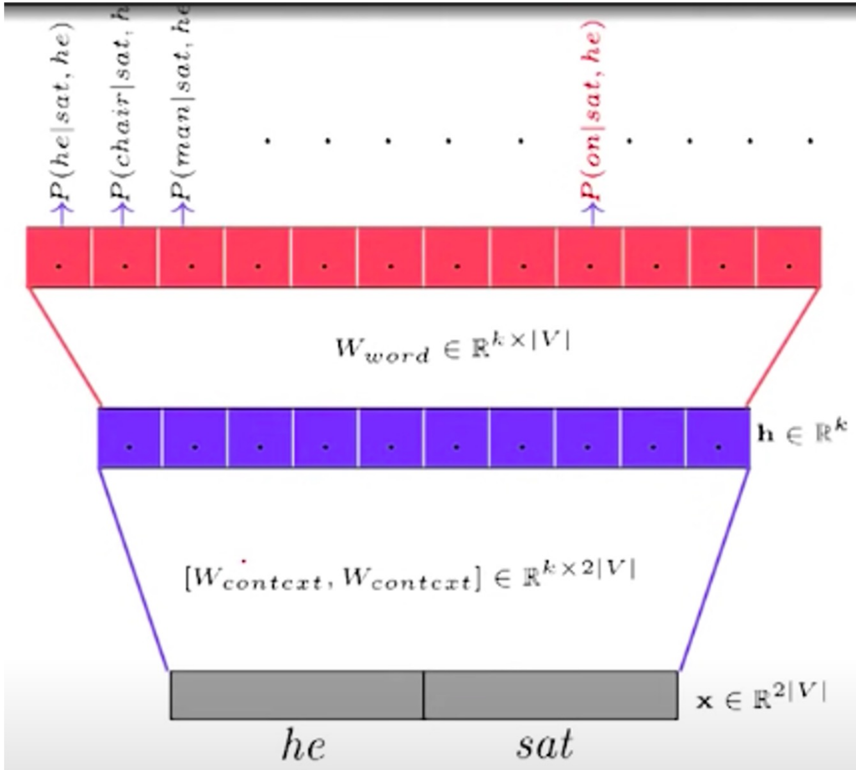
- **Consider this Task:** Predict $n$-th word given previous $n$-1 words
- **Example:** he sat on a chair
- **Training data:** All $n$-word windows in your corpus
- Training data for this task is easily available (take all $n$ word windows from the whole of wikipedia)
- For ease of illustration, we will first focus on the case when $n = 2$ (*i.e.*, predict second word based on first word)

We will now try to answer these two questions:

- How do you model this task?
- What is the connection between this task and learning word representations?




- We will model this problem using a feedforward neural network
- **Input:** One-hot representation of the **context word**
- **Output:** There are $|V|$ words (classes) possible and we want to predict a probability distribution over these $|V|$ classes (multi-class classification problem)
- **Parameters:** $\mathbf{W}_{context} \in \mathbb{R}^{k \times |V|}$ and $\mathbf{W}_{word} \in \mathbb{R}^{k \times |V|}$
  (we are assuming that the set of **words** and **context** words is the same: each of size $|V|$)




**Some problems:**

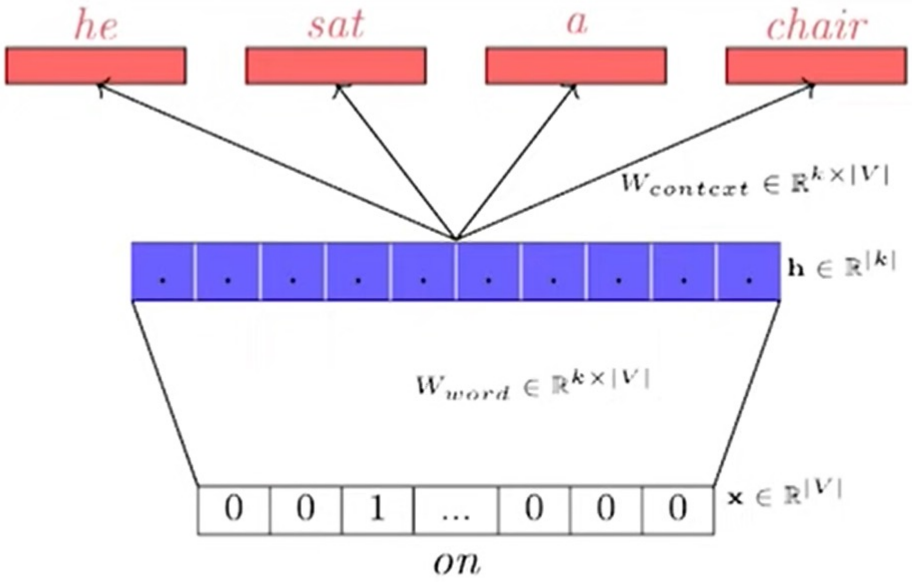
- Notice that the softmax function at the output is computationally very expensive

$$\hat{y}_w = \frac{exp(u_c \cdot v_w)}{\sum_{w' \in V} exp(u_c \cdot v_{w'})}$$

- The denominator requires a summation over all words in the vocabulary

# Skip gram method

- The model that we just saw is called the continuous bag of words model (it predicts an output word give a bag of context words)




- Notice that the role of *context* and *word* has changed now

# Glove representation

- **Count** based methods (SVD) rely on global co-occurrence counts from the corpus for computing word representations
- Predict based methods **learn** word representations using co-occurrence information

# Corpus:

- Human machine interface for computer applications
- User opinion of computer system response time
- User interface management system
- System engineering for improved response time

$$X =$$

| | human | machine | system | for | ... | user |
|---|---|---|---|---|---|---|
| human | 2.01 | 2.01 | 0.23 | 2.14 | ... | 0.43 |
| machine | 2.01 | 2.01 | 0.23 | 2.14 | ... | 0.43 |
| system | 0.23 | 0.23 | 1.17 | 0.96 | ... | 1.29 |
| for | 2.14 | 2.14 | 0.96 | 1.87 | ... | -0.13 |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| . | . | . | . | . | . | . |
| user | 0.43 | 0.43 | 1.29 | -0.13 | ... | 1.71 |

$$P(j|i) = \frac{X_{ij}}{\sum X_{ij}} = \frac{X_{ij}}{X_i}$$

$$X_{ij} = X_{ji}$$

- $X_{ij}$ encodes important global information about the co-occurrence between $i$ and $j$ (global: because it is computed from the entire corpus)