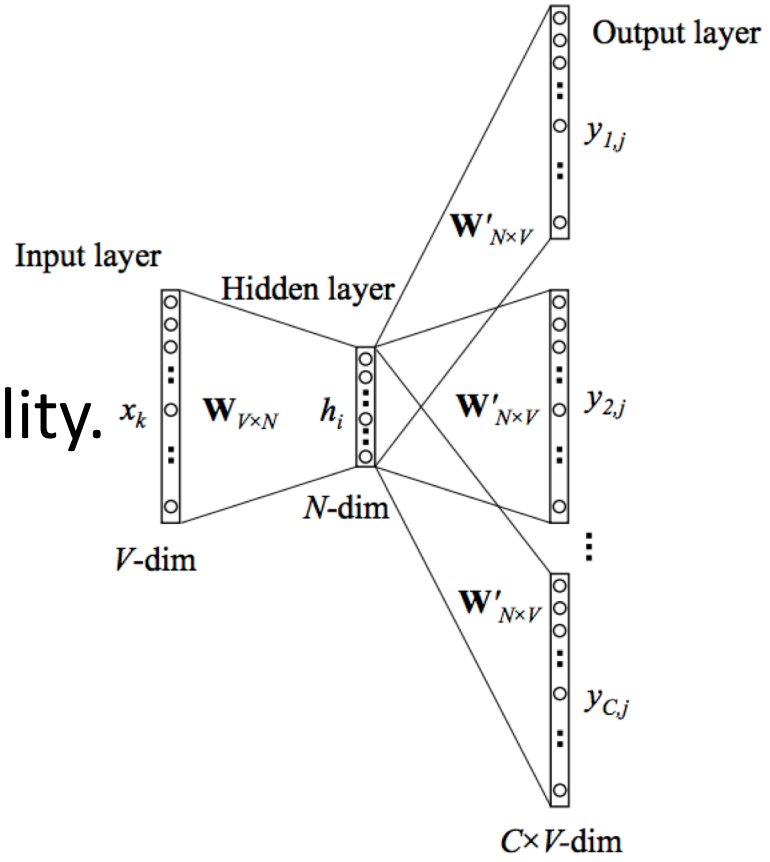
Word2Vec

W2V Structure

- CBOW & Skip-gram
- Assume target word based on context words

- More comparisons enhance the embedding quality.

→ Skip-gram is better than CBOW



Negative Sampling

- Each step, classify the pair into positive or negative (Binary classification)

  \+ : pair of target and context (unique)

  \- : pair of target and non-context (k cases)

- Reduce the burden of calculation (Whole corpus -> only k+1 sample)

Probability of negative sample

$$P_{negative}(w_i) = \frac{U(w_i)^{3/4}}{\sum_{j=0}^{n} U(w_j)^{3/4}}$$

$U(w_i)$: Uni-gram Probability of the word(frequency of the word/the number of whole words)

- Lower the high frequency word, raise the lower frequency words.

Subsampling

$$P_{subsampling}(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

When training, practically

- Exclude high frequency word
- Include low frequency word

The probability of target and context words are positive sample

$$P(+|t,c) = \frac{1}{1+\exp(-u_t \cdot v_c)}$$

The probability of target and context words are negative sample

$$P(-|t,c) = 1 - P(+|t,c)$$

Log-likelihood function

$$\mathcal{L}(\theta) = \log P\left(+\middle|t_p, c_p\right) + \sum_{i=1}^{k} \log P(-|t_i, c_i)$$

When model(θ) is updated, k+1 samples are trained