

- 희소표현 (Sparse Representation) 과 분산표현 (Sparse Representation)
  - : 희소표현: one-hot-encoding
  - : 분산표현에 기반하여 단어 간 의미적 유사성을 백터화 == 워드 임베딩 (embedding)
- CBOW (Continuous Bag of Words) 와 Skip-Gram
  - : CBOW\_ 주변에 있는 단어들을 입력으로 중간에 있는 단어들을 예측
    - 이는 center word와 context word로 구성되고 간략히는, context word를 통해 center word를 예측합니다.
    - window size를 통해 참고하려고 하는 주변 단어의 개수를 설정합니다. **sliding window**\_ window의 위치를 연속적으로 옮겨가며 학습 데이터 셋을 생성해갑니다.

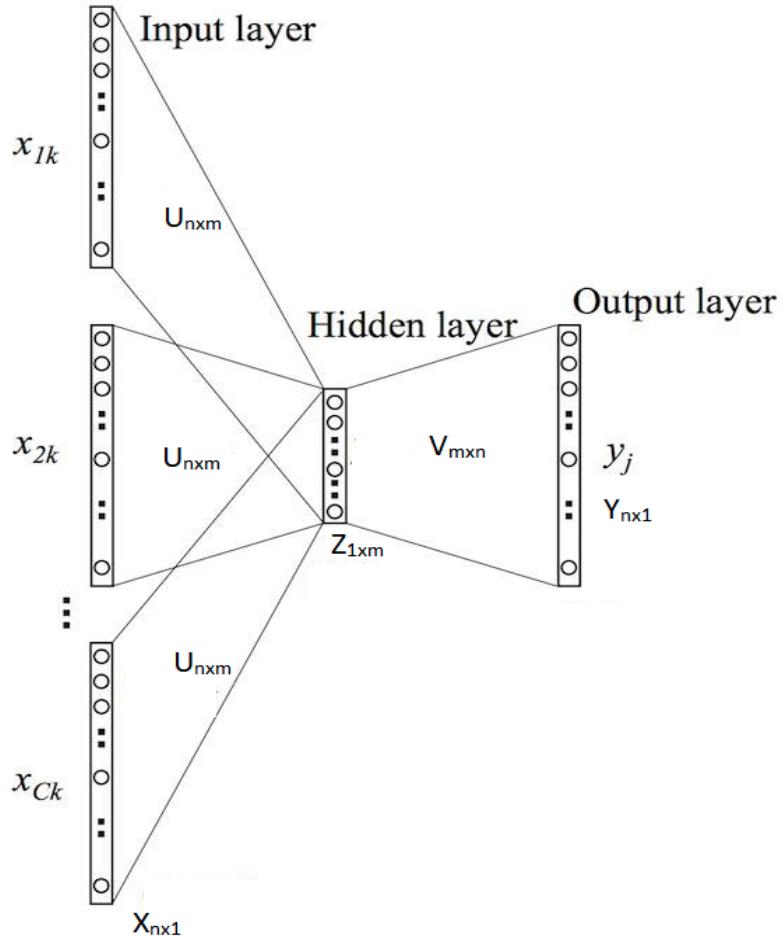


Fig1

- Fig1
- CBOW 알고리즘을 도식화 한 것입니다.  $CxV$ 는 center word를 제외한 context word의 one hot vector이며 이를 perceptron을 통과시켜 예측하고자 하는 center word에 대한  $Vx1$  (Output layer)를 얻게됩니다.
- Word2Vec은 은닉층이 1개로 Shallow neural network에 기반하고 따라서, 활성화 함수가 존재하지 않습니다.
- 활성화 함수가 존재하지 않는 대신 가중치를 따로 적용되어져야 합니다. 기본적으로 Input layer와 Hidden layer 사이의 가중치  $w$ 는  $v$  (vocab size) \*  $m$ , Projection layer와 Output layer 사이의  $w'$ 는  $m * v$ 입니다.
- $Z_{xm}$  즉 projection layer라고도 불리는 해당 알고리즘의 Hidden layer는 Embedding 후의 Vector의 차원과 같습니다.

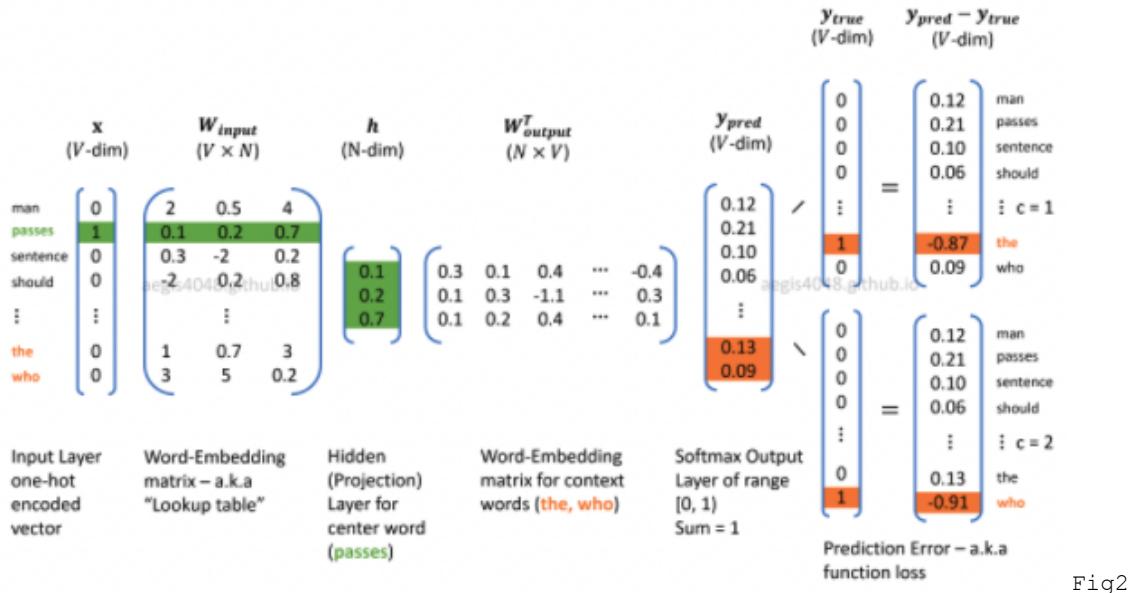


Fig2

Fig2의  $h$ 는 각 Input layer의 encoded된 vector와 Weight의 곱에 의해 결정됩니다. 여기서 input layer의 각 값이 단독의 1과 0인 나머지의 값으로 이루어진 One-Hot-Vector이기에 결과적으로는 input에 의해  $W$ 의 i번째 행이 출력되는 lookup 연산과정이 진행됨과 다름이 없습니다. 즉, i번째 i번째  $W$ 행의 값 lookup table 출력되는 것입니다.

더불어  $W$ 의 초기값은 Random한 행렬로 지정됩니다. 이는 반복학습을 통해 가장 이상적인 가중값 행렬을 갱신해갑니다.

:\_1 **lookup**해온  $W$ 의 각 행벡터가 **Word2Vec** 학습 후에는 각 단어의 **M**차원의 임베딩 벡터로 간주됩니다.

출력된 각 lookup 결과들을 projection layer에서 합하여 평균값을 도출합니다.

**평균** ( $\bar{v}$ ) =  $(v_{x_0} + \dots + v_{x_n}) / 2n$  여기서  $2n$ 은 window size만큼 input이 반복됨을 고려되어야 함으로,  $2 * \text{window\_size}$ 가 됩니다. 반대로 Skip-Gram은 입력이 center word 하나이므로 벡터의 평균을 구하지 않습니다.

:\_2  $v$ 는 2번째 가중치인  $W^T$ 와 곱해집니다. 이를 통해  $v$ 의 dimension과 같은 규격의 vector로 변환됩니다. 더불어 이는 softmax function을 지나면서 0, 1사이의 값으로 scaling됩니다.

우리는 이를 과정을 통해서, 문장 안에서 우리가 유추하고자 하는 center word가 높은 score 갖게 하는 것을 모델 학습의 목표로 합니다.

즉, center word의 One-Hot-Vector와 근사값 얻어야 합니다.

예를 들어 context word가 2개이고 center word가 3번째에 위치했다면 [0, 1, 0]과 최대로 유사한 ratio의 score vector를 얻는 것을 목표로 합니다.

더불어, 이렇게 예측된  $\hat{y}$ 와 실제 값  $y$  간의 Cross-Entropy를 통해 해당 학습의 성능을 평가할 수 있습니다.

\*\*  $n$  번의 epoch를 통해 **back propagation**을 수행함에  $W$ ,  $W^T$ 가 학습되는데 이를 통해 얻게 되는 **M** 차원의  $W$ 행렬의 각 **lookup**값을 각 단어의 **Embedding**으로 활용합니다.

#### - Skip-gram

: 이는 CBOW의 학습 아이디어와 완전히 일치하고 다만 그 해당 학습 과정을 반대로 진행하여 center word를 통해 context word를 예측한다는 것에 그 목적이 있습니다.

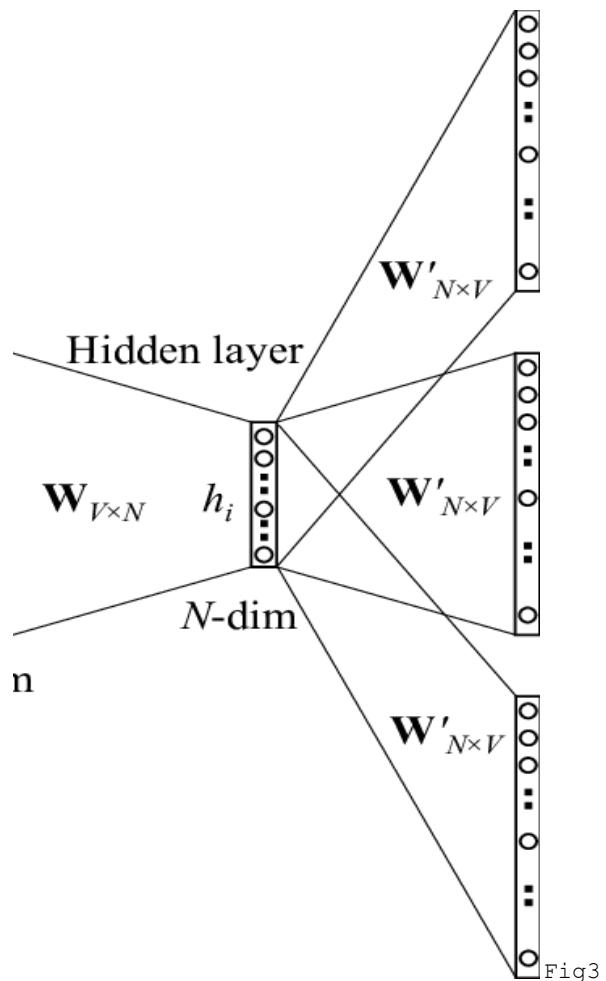


Fig3

\*\* 앞서 언급한 바와 같이 Skip-Gram은 CBOW와 달리 **center word**를 통해 **context words**를 예측하하는 것이 목표이기 때문에  $\mathbf{v}$  즉, 평균값을 구하는 과정이 생략되어 있습니다.