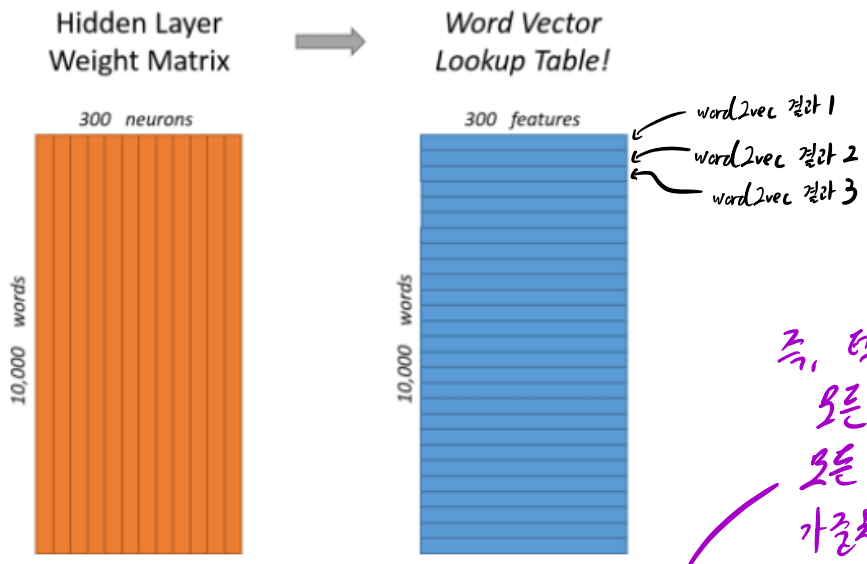


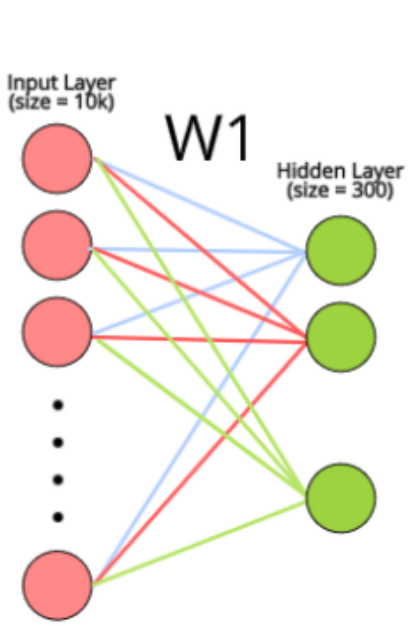
즉, 텍스트 데이터 셋 내 모든 문장들에서 사용된 모든 단어들을 훈련하여, 가중치 행렬을 학습한다.

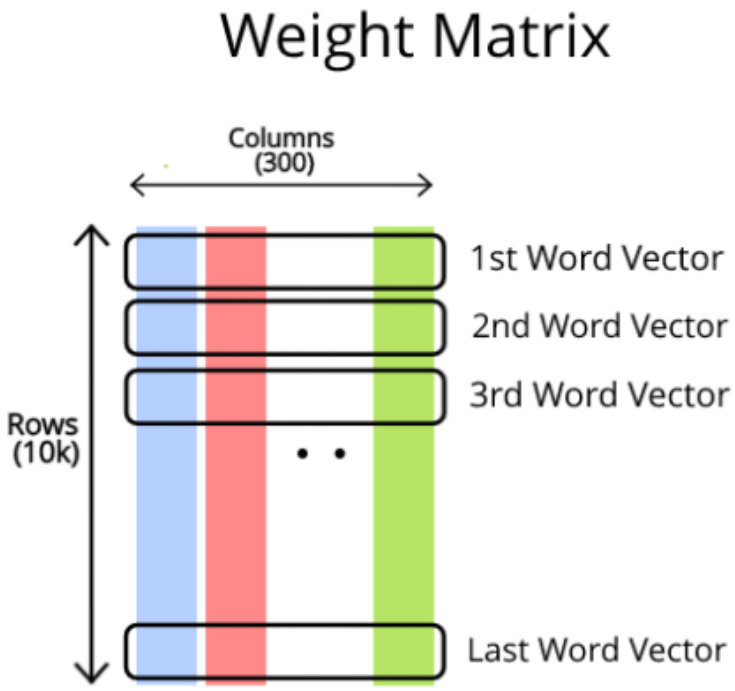
Word2Vec 아키텍처는 중심단어로 주변단어를 맞추거나, 주변단어로 중심단어를 더 잘 맞추기 위해 가중치 행렬인 W, W'을 조금씩 업데이트하면서 학습이 이뤄지는 구조입니다. 그렇다면 여기서 흥미로운 점은 W가 one-hot-encoding된 입력벡터와 은닉층을 이어주는 가중치행렬임과 동시에 Word2Vec의 최종 결과물인 **임베딩 단어벡터의 모음**이기도 하다는 사실입니다.

아래와 같이 단어가 5개뿐인 말뭉치에서 Word2Vec을 수행한다고 가정해 봅시다. 사전 등장 순서 기준으로 네번째 단어를 입력으로 하는 은닉층 값은 아래처럼 계산됩니다. 보시다시피 Word2Vec의 은닉층을 계산하는 작업은 사실상 가중치행렬 W에서 해당 단어에 해당하는 행벡터를 참조(lookup)해 오는 방식과 똑같습니다.

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

학습이 마무리되면 이 W의 행벡터들이 각 단어에 해당하는 임베딩 단어벡터가 됩니다.





Word2Vec deep learning model input and target

input (word one hot encoding)	target (neighbor one hot encoding)
[1, 0, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0]
[1, 0, 0, 0, 0, 0]	[0, 0, 1, 0, 0, 0]
[0, 1, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0]
[0, 1, 0, 0, 0, 0]	[0, 0, 1, 0, 0, 0]
[0, 0, 1, 0, 0, 0]	[1, 0, 0, 0, 0, 0]
[0, 0, 1, 0, 0, 0]	[0, 1, 0, 0, 0, 0]
[0, 0, 0, 1, 0, 0]	[0, 0, 0, 0, 1, 0]
[0, 0, 0, 1, 0, 0]	[0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 1, 0]	[0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 1, 0]	[0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 1]	[0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 0, 1]	[0, 0, 0, 0, 1, 0]

Word2Vec training