1D CONVOLUTIONAL - EXAMPLE

Feature Detector

filter

Height

Start Position

- I
- Love
- One
- Dimensional
- Convolutional
- Neural
- Networks
- Very

Final Position

- Much

한 단어에 대한
밀집벡터.

Encoded Representation Of Word
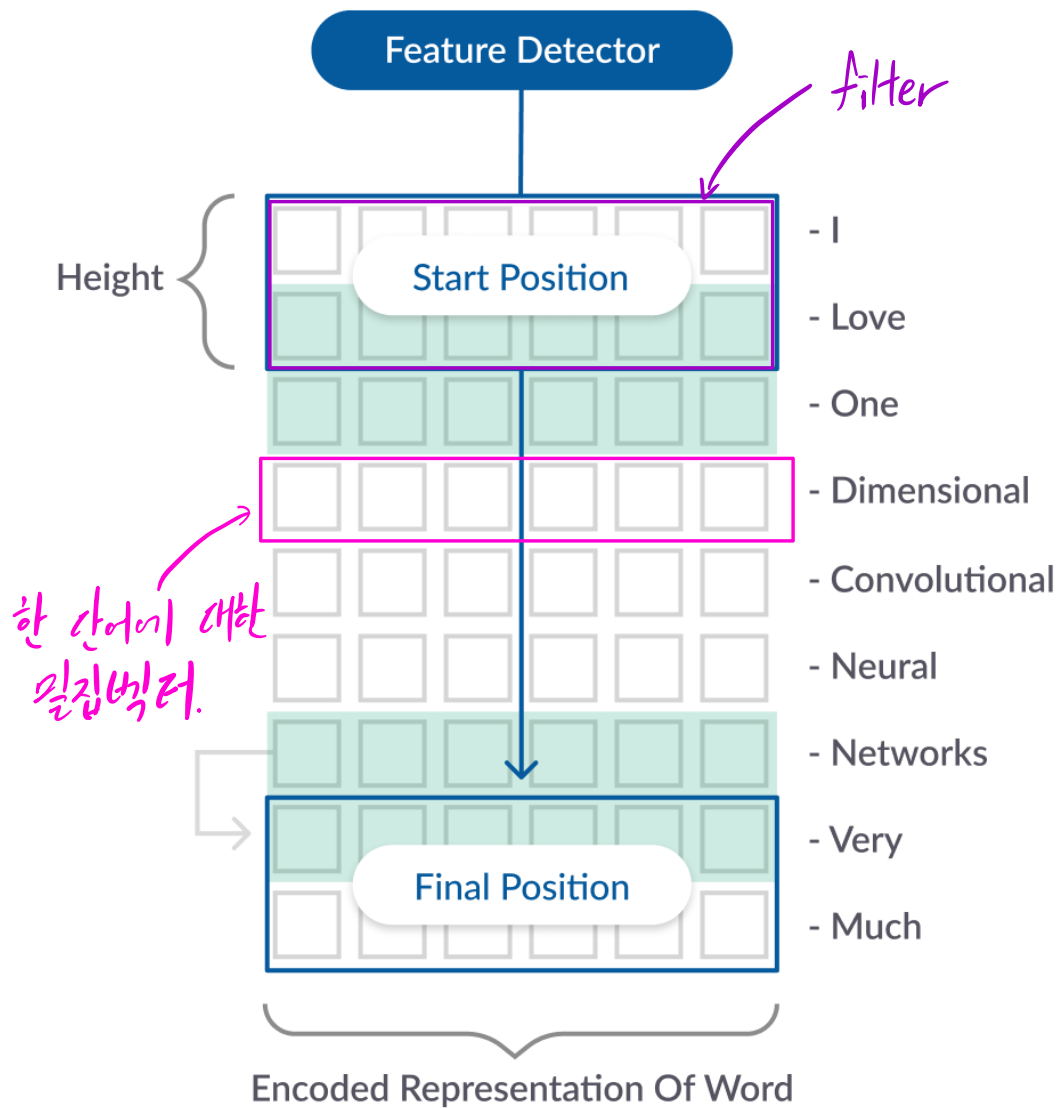
한 단어에 대한
밀집벡터.

sample sentence	Features (i.e. word embeddings)				
word_1					
word_2					
word_3					
word_4					
word_5					
word_6					
. . .	. . .	. . .	. . .	...	. . .
word_m					

Window of length k (=3)

Filters with kernel size k (=3)

Filters response
(i.e. convolution result)

word_i				...	
word_(i+1)					
word_(i+2)					

$*$
convolution

filter_1

...

r_1

filter_2

...

r_2

.
.
.

.
.
.

filter_n

...

r_n

Filters response	filter_1	filter_2	filter_3		filter_n
window_1					
window_2					
window_3					
window_4					
window_5					
window_6				...	
. . .	. . .	. . .	. . .		. . .
window_(m-k+1)					

As you can see in the figure above, the response of each filter is equivalent to the result of its convolution (i.e. element-wise multiplication and then summing all the results) with the extracted window of length k (i.e. i-th to $(i+k-1)$-th words in the given sentence). Further, note that each filter has the same number of channels as the number of features (i.e. word-embeddings dimension) of the training sample (hence performing convolution, i.e. element-wise multiplication, is possible). Essentially, each filter is detecting the presence of a particular feature of pattern in a **local** window of training data (e.g. whether a couple of specific words exist in this window or not). After all the filters have been applied on all the windows of length k we would have an output of like this which is the result of convolution: