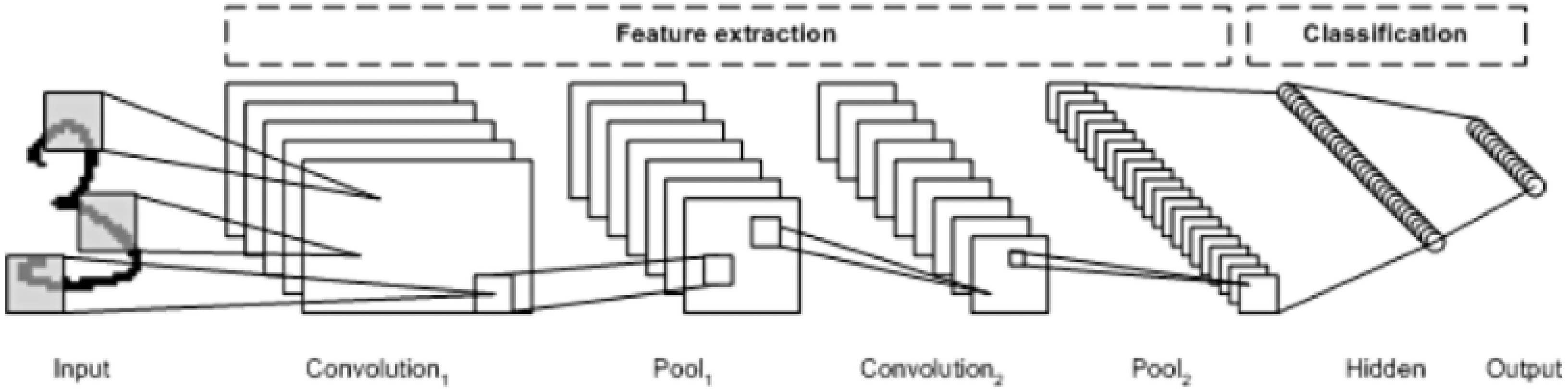
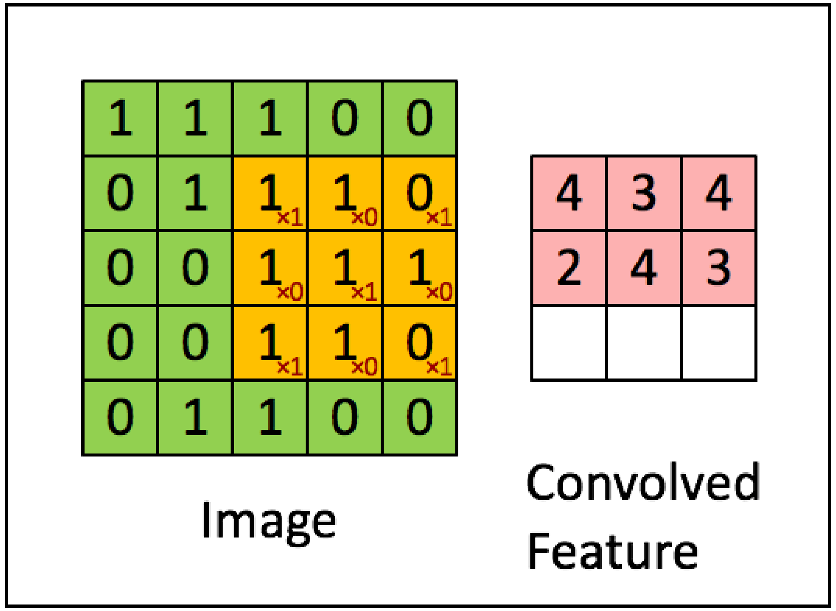
Convolutional Neural Network for Sentence Classification

2021.08.08. 김수연

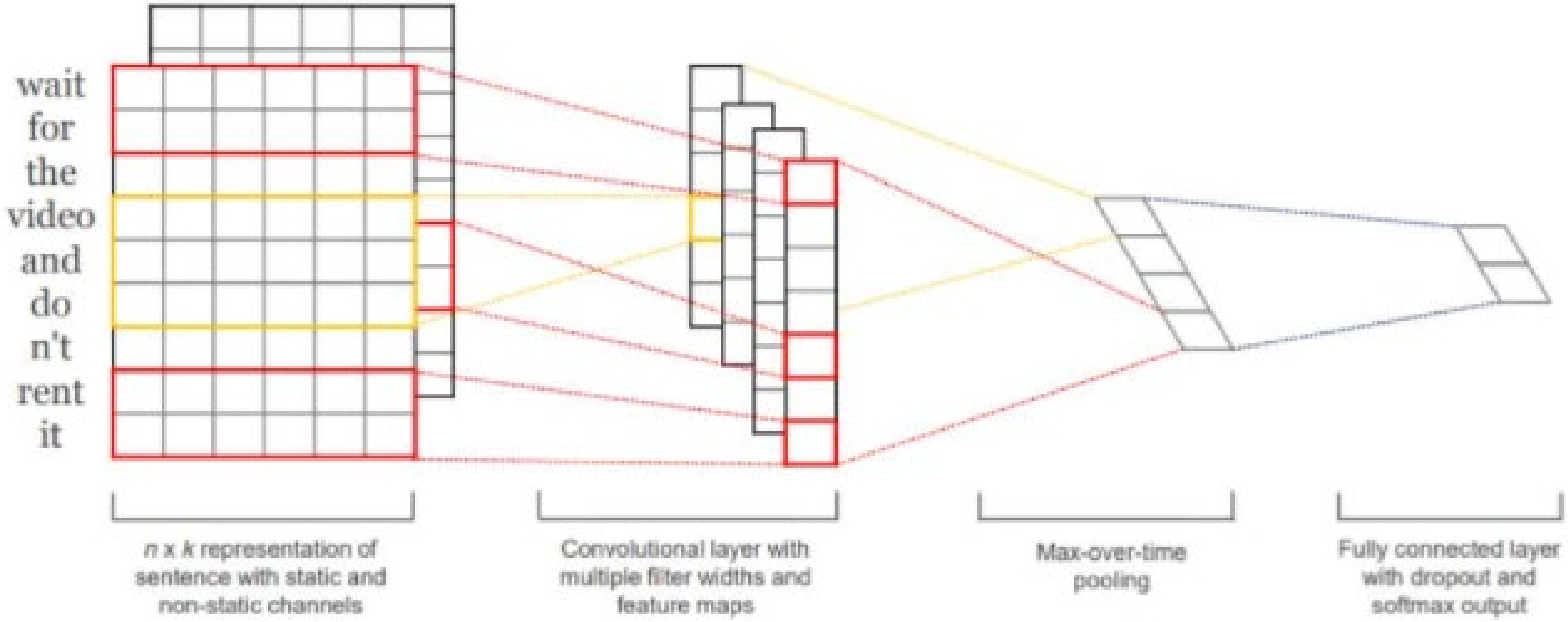
CNN



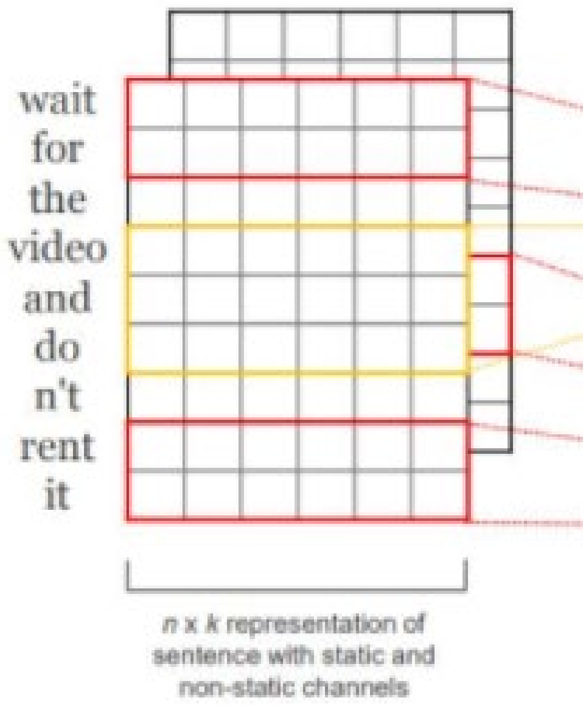


- Convolution 연산을 바탕으로 하는 net
- Convolution layer와 Fully connected layer로 이루어짐
- 이미지 분류에 주로 이용됨

Architecture



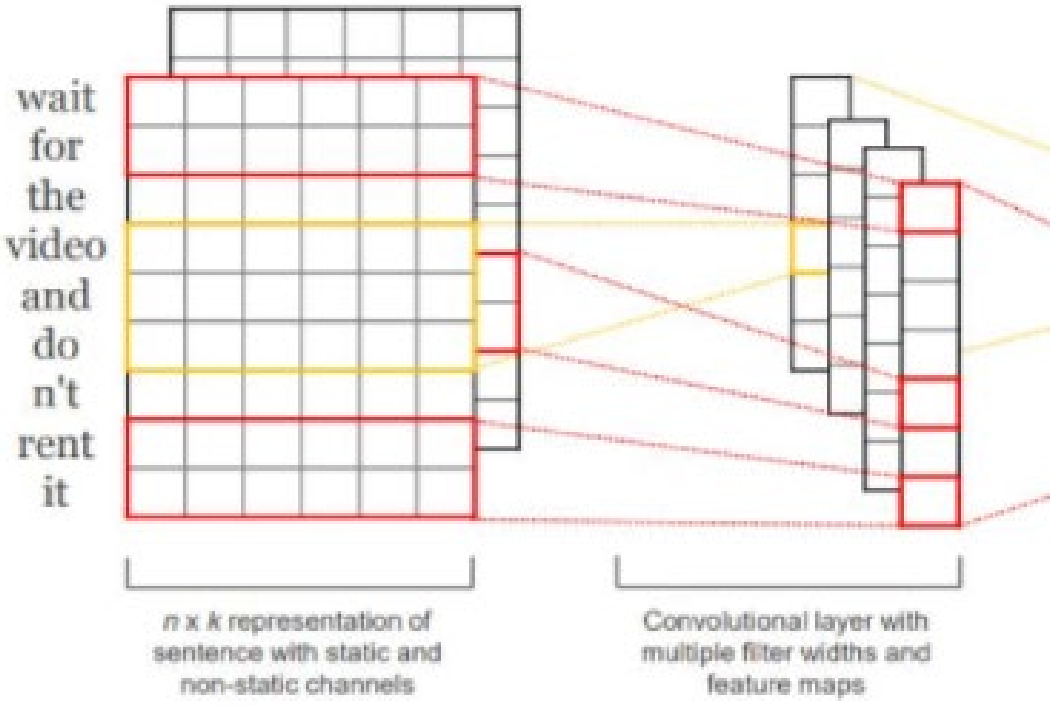
Architecture



- Input : k – dimension 단어벡터

$$\mathbf{x}_{i:i+j} = \mathbf{x}_i \oplus \mathbf{x}_{i+1} \oplus \ldots \oplus \mathbf{x}_{i+j}.$$

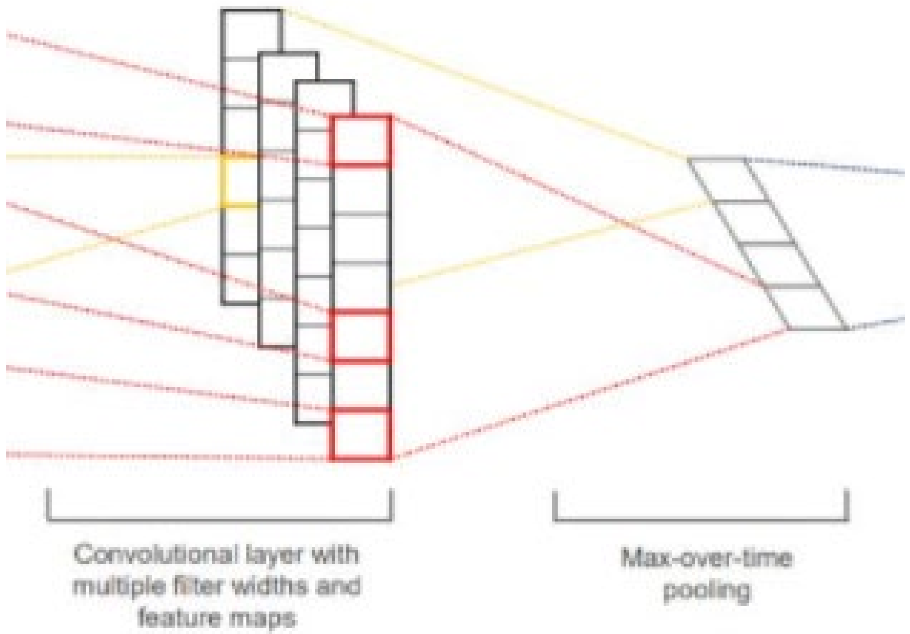
Architecture



- Convolution을 통한 feature c
- b는 bias, f 는 non-linear 함수

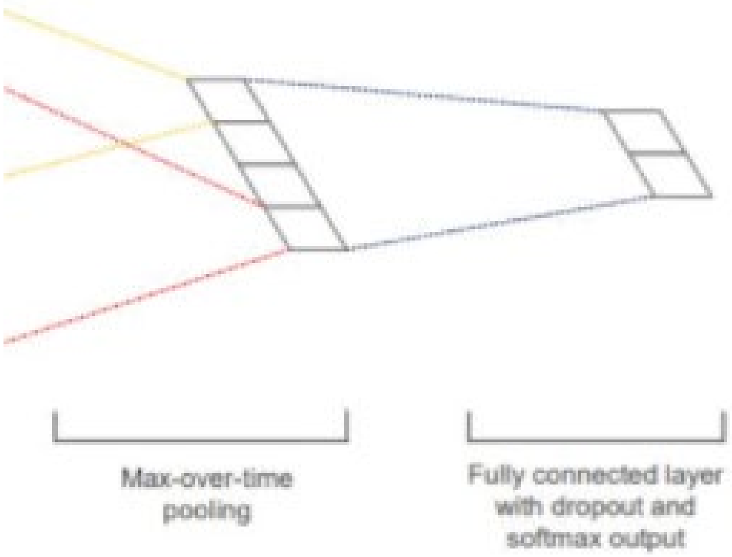
$$c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b)$$

Architecture



- Feature map에 max-pooling을 해 최대값 찾기
- 하나의 필터에서 하나의 feature
- 논문에선 penultimate layer라고 칭함

Architecture



- Fully connected softmax layer통과시켜 확률분포 만듦

- 하나의 channe은 stati한 word vector
- 나머지는 backpropagation통한 fine tuning
- 두개의 채널은 filter적용한후 더함

Regularization

- Dropout과 l2-norm 사용
- Dropout으로 hidden unit이 같이 더해지는 것 방지
- Forward와 back propagation에서 모두 사용

Regularization

$$y = \mathbf{w} \cdot \mathbf{z} + b$$

$$y = \mathbf{w} \cdot (\mathbf{z} \odot \mathbf{r}) + b$$

$$\mathbf{z} = [\hat{c}_1, \ldots, \hat{c}_m]$$

$$\hat{\mathbf{w}} = p\mathbf{w}$$

- Forward단계에서 아래의 식을 사용.
- z와 r의 원소별 곱셈
- Z는 max-pooling후의 layer
- R은 masking 벡터.
  - 확률 p의 베르누이 분포를 따르기에 0 또는 1의 값
- Test과정에서는 dropout하지않고 w대신 p를 곱해서 사용

Regularization (X)

$$\hat{\mathbf{w}} = p\mathbf{w}$$

- Test과정에서는 dropout하지않고 w대신 p를 곱해서 사용

$$\mathbf{w} = \begin{cases} w, \text{ if } \|\mathbf{w}\|_2 < s \\ s, \text{ other wise} \end{cases}$$

- L2 norm의 경우 벡터의 l2 norm이

hyperparameter

- ReLU 함수
- Feature map 100, filter size 3,4,5
- Dropout 0.5
- L2 3
- 미니배치 50
- Adadelta 이용 parameter update

결과

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	**89.6**
CNN-non-static	**81.5**	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	**88.1**	93.2	92.2	**85.0**	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	–	–	–	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	–	–	–	–
RNTN (Socher et al., 2013)	–	45.7	85.4	–	–	–	–
DCNN (Kalchbrenner et al., 2014)	–	48.5	86.8	–	93.0	–	–
Paragraph-Vec (Le and Mikolov, 2014)	–	**48.7**	87.8	–	–	–	–
CCAE (Hermann and Blunsom, 2013)	77.8	–	–	–	–	–	87.2
Sent-Parser (Dong et al., 2014)	79.5	–	–	–	–	–	86.3
NBSVM (Wang and Manning, 2012)	79.4	–	–	93.2	–	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	–	–	**93.6**	–	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	–	–	93.4	–	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	–	–	**93.6**	–	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	–	–	–	–	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	–	–	–	–	–	82.7	–
SVM_S (Silva et al., 2011)	–	–	–	–	**95.0**	–	–

- 모두 비슷한 결과를 보임
- CNN을 활용해 만든 간단한 모델도 뛰어난 성능을 보여줌