– IN5550 –

Neural Methods in Natural Language Processing

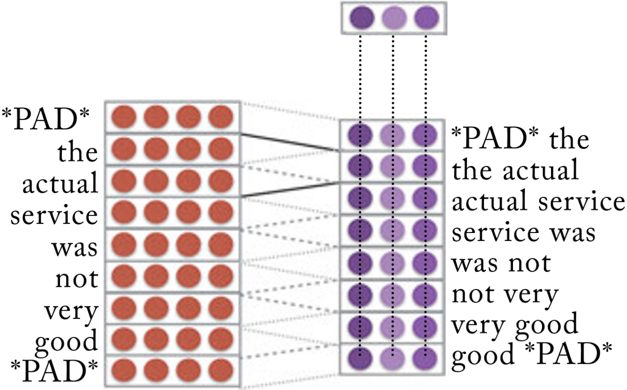
CNNs, Part 3: Pooling

Erik Velldal

Language Technology Group (LTG)
University of Oslo

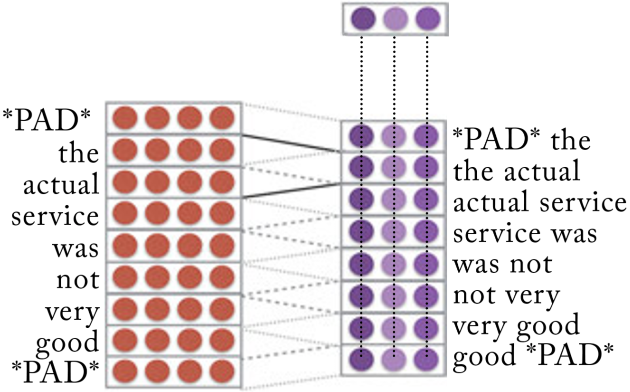
Next step: pooling (1:2)

- The convolution layer results in m vectors $\boldsymbol{p_{1:m}}$.
- Each $\boldsymbol{p_i} \in \mathbb{R}^{\ell}$ represents a particular k-gram in the input.
- m (the length of the feature maps) can vary depending on input length.
- Pooling combines these vectors into a single fixed-sized vector $\boldsymbol{c}$.



Next step: pooling (2:2)

- The fixed-sized vector $\boldsymbol{c}$ (possibly in combination with other vectors) is what gets passed to a downstream network for prediction.
- Want $\boldsymbol{c}$ to contain the most important information from $\boldsymbol{p}_{\mathbf{1}:\boldsymbol{m}}$.
- Different strategies available for ‘sampling’ features.



Pooling strategies

Max pooling

- Most common. AKA max-over-time pooling or 1-max pooling.
- $\boldsymbol{c}[j] = \underset{1<i\leq m}{\arg\max}\, \boldsymbol{p_i}_{[j]} \quad \forall j \in [1, l]$
- Picks the maximum value across each dimension (feature map).

K-max pooling

- Concatenate the k highest values for each dimension / filter.

Average pooling

- $\boldsymbol{c} = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{p_i}$
- Average of all the filtered k-gram representations.

Dynamic pooling

- Combines with any of the strategies above.
- Perform pooling separately over r different regions of the input.
- Concatenate the r resulting vectors $\boldsymbol{c_1}, \ldots \boldsymbol{c_r}$.
- Allows us to retain positional information relevant to a given task (e.g. based on document structure).

Dynamic pooling

- Combines with any of the strategies above.
- Perform pooling separately over r different regions of the input.
- Concatenate the r resulting vectors $\boldsymbol{c_1}, \ldots \boldsymbol{c_r}$.
- Allows us to retain positional information relevant to a given task (e.g. based on document structure).

- Note that pooling is not specific to CNNs: can also be used in combination with other architectures, e.g. RNNs.

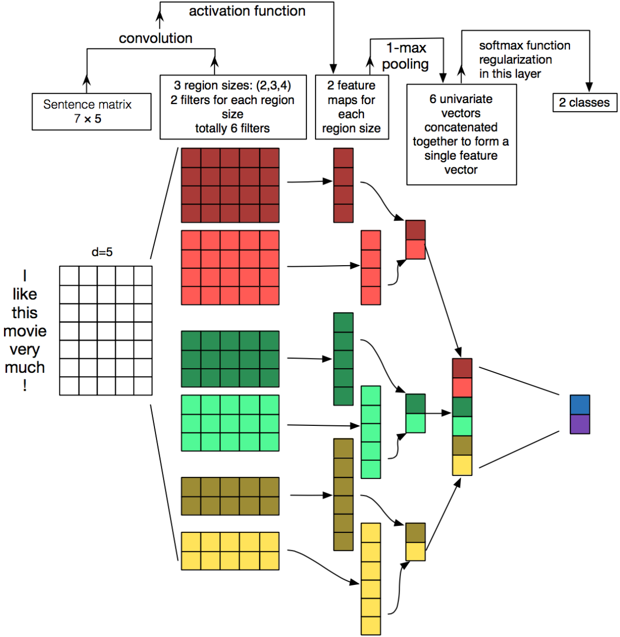
Multiple window sizes

- So far considered CNNs with ℓ filters for a single window size k.
- Typically, CNNs in NLP are applied with multiple window sizes, and multiple filters for each.
- Pooled separately, with the results concatenated.
- Rather large window sizes often used:
- 2–5 is most typical, but even $k > 20$ is not uncommon.

Multiple window sizes

- So far considered CNNs with ℓ filters for a single window size k.
- Typically, CNNs in NLP are applied with multiple window sizes, and multiple filters for each.
- Pooled separately, with the results concatenated.
- Rather large window sizes often used:
- 2–5 is most typical, but even $k > 20$ is not uncommon.
- With standard n-gram features, anything more than 3-grams quickly become infeasible.
- CNNs represent large n-grams efficiently, without blowing up the parameter space and without having to represent the whole vocabulary.
- (Related to the notion of 'neuron' in a CNN – will get back to this!)

Baseline architecture of Zhang et al. (2017)


activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

Zhang et al. (2017)