– IN5550 –
Neural Methods in Natural Language Processing

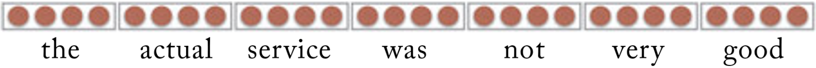
CNNs, Part 2: Convolutions

Erik Velldal

Language Technology Group (LTG)
University of Oslo

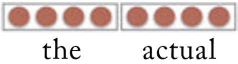
1d CNNs for NLP

- Consider a sequence of words $w_{1:n} = w_1, \ldots, w_n$.
- Each word is represented by a d dimensional embedding $\boldsymbol{E}_{[w_i]} = \boldsymbol{w_i}$.



- A convolution corresponds to 'sliding' a window of size k across the sequence and applying a filter to each.
- Let $\oplus(\boldsymbol{w_{i:i+k-1}}) = [\boldsymbol{w_i}; \boldsymbol{w_{i+1}}; \ldots; \boldsymbol{w_{i+k-1}}]$ be the concatenation of the embeddings $\boldsymbol{w_i}, \ldots, \boldsymbol{w_{i+k-1}}$.
- The vector for the ith window is $\boldsymbol{x_i} = \oplus(\boldsymbol{w_{i:i+k-1}})$, where $\boldsymbol{x_i} \in \mathbb{R}^{kd}$.

$\boldsymbol{x_1} \longrightarrow$



Convolutions on sequences

To apply a filter to a window $\boldsymbol{x_i}$:

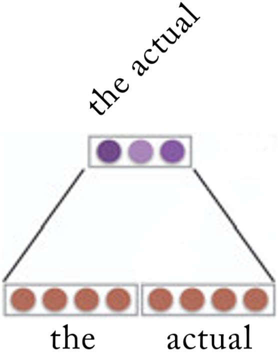
- compute its dot-product with a weight vector $\boldsymbol{u} \in \mathbb{R}^{kd}$
- and then apply a non-linear activation g,
- resulting in a scalar value $p_i = g(\boldsymbol{x_i} \cdot \boldsymbol{u})$

Convolutions on sequences

To apply a filter to a window $\boldsymbol{x}_i$:

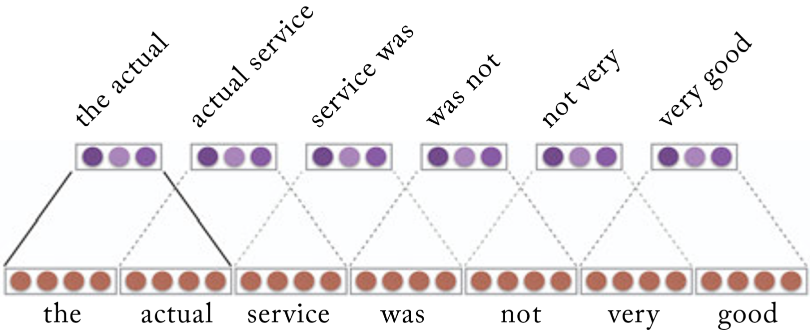
- compute its dot-product with a weight vector $\boldsymbol{u} \in \mathbb{R}^{kd}$
- and then apply a non-linear activation g,
- resulting in a scalar value $p_i = g(\boldsymbol{x_i} \cdot \boldsymbol{u})$

- Typically use ℓ different filters, $\boldsymbol{u_1}, \ldots, \boldsymbol{u_\ell}$.
- Can be arranged in a matrix $\boldsymbol{U} \in \mathbb{R}^{kd \times \ell}$.
- Also include a bias vector $\boldsymbol{b} \in \mathbb{R}^{\ell}$.
- Gives an ℓ-dimensional vector $\boldsymbol{p_i}$ summarizing the ith window: $\boldsymbol{p_i} = g(\boldsymbol{x_i} \cdot \boldsymbol{U} + \boldsymbol{b})$
- Ideally different dimensions captures different indicative information.



Convolutions on sequences

- Applying the convolutions over the text results in m vectors $\boldsymbol{p_{1:m}}$.
- Each $\boldsymbol{p_i} \in \mathbb{R}^{\ell}$ represents a particular k-gram in the input.
- Sensitive to the identity and order of tokens within the sub-sequence,
- but independent of its particular position within the sequence.



Narrow vs. wide convolutions

- What is m in $\boldsymbol{p_{1:m}}$?
- For a given window size k and a sequence $w_1, \ldots, w_n$, how many vectors $\boldsymbol{p_i}$ will be extracted?
- There are $m = n - k + 1$ possible positions for the window.
- This is called a narrow convolution.

Narrow vs. wide convolutions

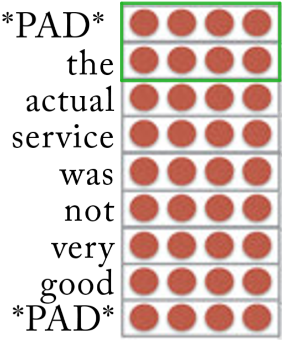
- What is m in $\boldsymbol{p_{1:m}}$?
- For a given window size k and a sequence $w_1, \ldots, w_n$, how many vectors $\boldsymbol{p_i}$ will be extracted?
- There are $m = n - k + 1$ possible positions for the window.
- This is called a narrow convolution.

- Another strategy: pad with $k - 1$ extra dummy-tokens on each side.
- Let's us slide the window beyond the boundaries of the sequence.
- We then get $m = n + k - 1$ vectors $\boldsymbol{p_i}$.
- Called a wide convolution.
- Necessary when using window-sizes that might be wider than the input.

Stacking view (1:3)

- So far we've visualized inputs, filters, and filter outputs as sequences:
- What Goldberg (2017) calls the 'concatenation notation'.

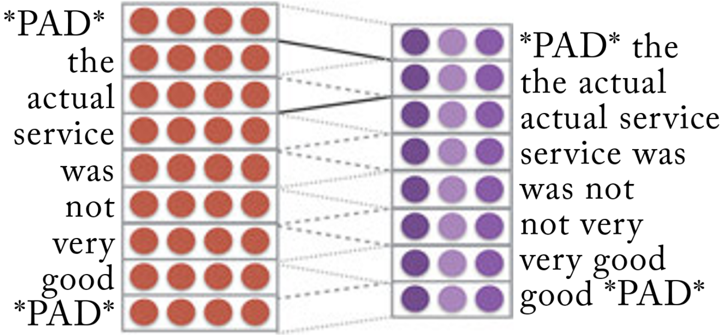
- An alternative (and perhaps more common) view: 'stacking notation'.
- Imagine the n input embeddings stacked on top of each other, resulting in an $n \times d$ sentence matrix.



- Correspondingly, imagine each column $\boldsymbol{u}$ in the matrix $\boldsymbol{U} \in \mathbb{R}^{kd \times \ell}$ be arranged as a $k \times d$ matrix.
- We can then slide ℓ different $k \times d$ filter matrices down the sentence matrix, computing matrix convolutions:
- Sum of element-wise multiplications.

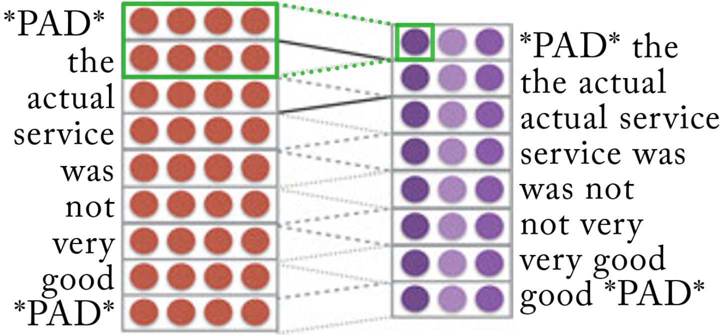
Stacking view (3:3)

- Now imagine the output vectors $\boldsymbol{p}_{1:m}$ stacked in a matrix $\boldsymbol{P} \in \mathbb{R}^{m \times \ell}$.
- Each ℓ-dimensional row of $\boldsymbol{P}$ holds the features extracted for a given k-gram by different filters.
- Each m-dimensional column of $\boldsymbol{P}$ holds the features extracted across the sequence for a given filter.
- These columns are sometimes referred to as feature maps.



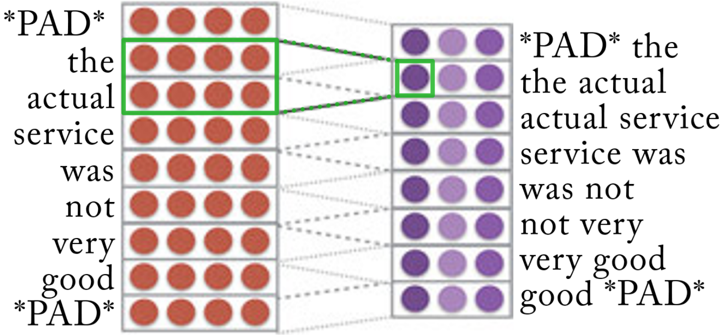
Stacking view (3:3)

- Now imagine the output vectors $\boldsymbol{p}_{1:m}$ stacked in a matrix $\boldsymbol{P} \in \mathbb{R}^{m \times \ell}$.
- Each ℓ-dimensional row of $\boldsymbol{P}$ holds the features extracted for a given k-gram by different filters.
- Each m-dimensional column of $\boldsymbol{P}$ holds the features extracted across the sequence for a given filter.
- These columns are sometimes referred to as feature maps.



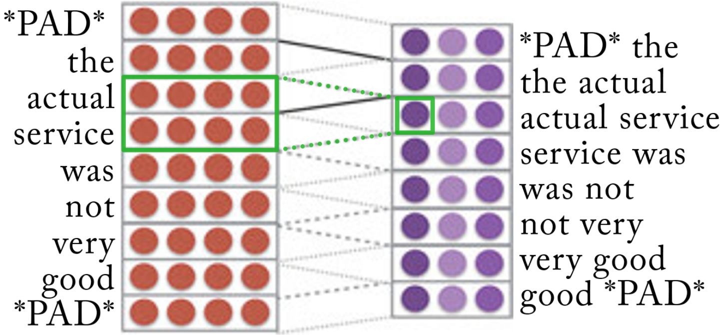
Stacking view (3:3)

- Now imagine the output vectors $\boldsymbol{p}_{1:m}$ stacked in a matrix $\boldsymbol{P} \in \mathbb{R}^{m \times \ell}$.
- Each ℓ-dimensional row of $\boldsymbol{P}$ holds the features extracted for a given k-gram by different filters.
- Each m-dimensional column of $\boldsymbol{P}$ holds the features extracted across the sequence for a given filter.
- These columns are sometimes referred to as feature maps.



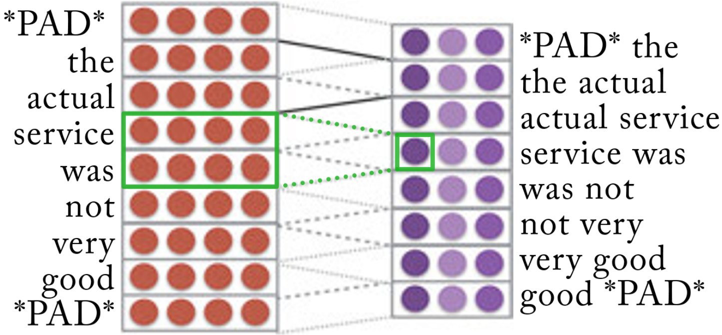
Stacking view (3:3)

- Now imagine the output vectors $\boldsymbol{p}_{1:m}$ stacked in a matrix $\boldsymbol{P} \in \mathbb{R}^{m \times \ell}$.
- Each ℓ-dimensional row of $\boldsymbol{P}$ holds the features extracted for a given k-gram by different filters.
- Each m-dimensional column of $\boldsymbol{P}$ holds the features extracted across the sequence for a given filter.
- These columns are sometimes referred to as feature maps.



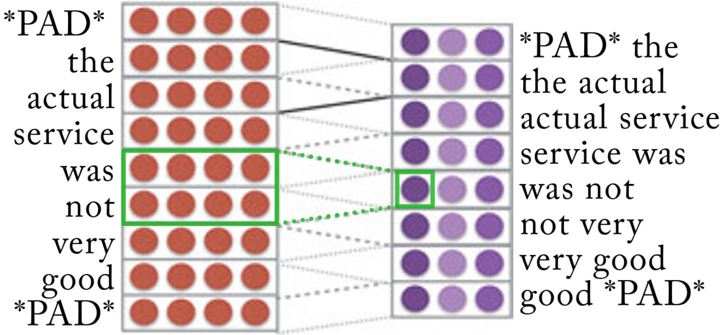
Stacking view (3:3)

- Now imagine the output vectors $\boldsymbol{p}_{1:m}$ stacked in a matrix $\boldsymbol{P} \in \mathbb{R}^{m \times \ell}$.
- Each ℓ-dimensional row of $\boldsymbol{P}$ holds the features extracted for a given k-gram by different filters.
- Each m-dimensional column of $\boldsymbol{P}$ holds the features extracted across the sequence for a given filter.
- These columns are sometimes referred to as feature maps.



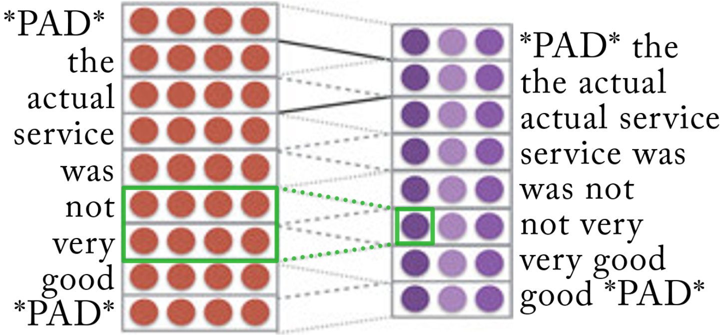
Stacking view (3:3)

- Now imagine the output vectors $\boldsymbol{p}_{1:m}$ stacked in a matrix $\boldsymbol{P} \in \mathbb{R}^{m \times \ell}$.
- Each ℓ-dimensional row of $\boldsymbol{P}$ holds the features extracted for a given k-gram by different filters.
- Each m-dimensional column of $\boldsymbol{P}$ holds the features extracted across the sequence for a given filter.
- These columns are sometimes referred to as feature maps.



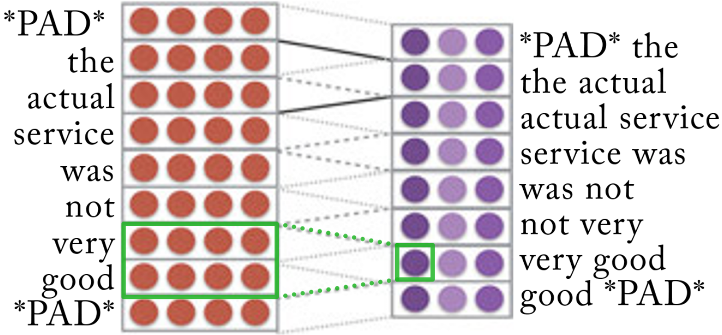
Stacking view (3:3)

- Now imagine the output vectors $\boldsymbol{p}_{1:m}$ stacked in a matrix $\boldsymbol{P} \in \mathbb{R}^{m \times \ell}$.
- Each ℓ-dimensional row of $\boldsymbol{P}$ holds the features extracted for a given k-gram by different filters.
- Each m-dimensional column of $\boldsymbol{P}$ holds the features extracted across the sequence for a given filter.
- These columns are sometimes referred to as feature maps.



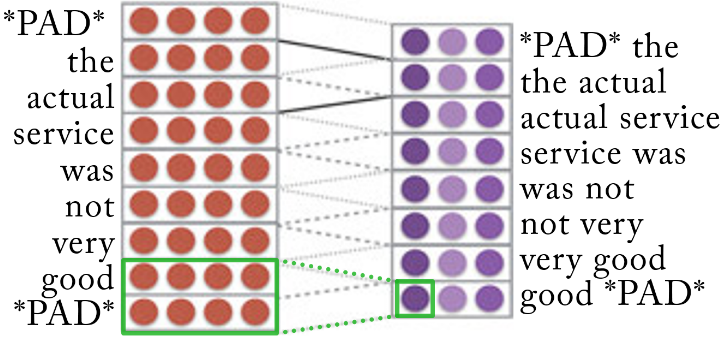
Stacking view (3:3)

- Now imagine the output vectors $\boldsymbol{p}_{1:m}$ stacked in a matrix $\boldsymbol{P} \in \mathbb{R}^{m \times \ell}$.
- Each ℓ-dimensional row of $\boldsymbol{P}$ holds the features extracted for a given k-gram by different filters.
- Each m-dimensional column of $\boldsymbol{P}$ holds the features extracted across the sequence for a given filter.
- These columns are sometimes referred to as feature maps.



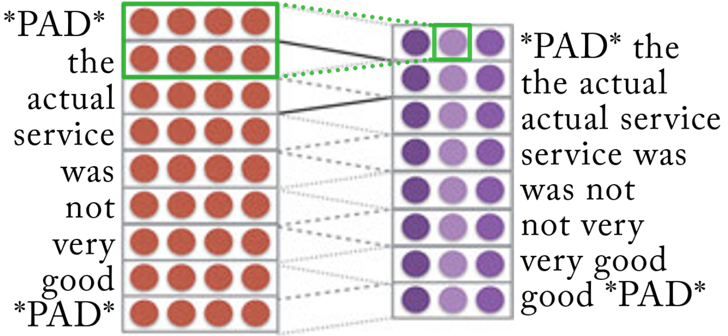
Stacking view (3:3)

- Now imagine the output vectors $\boldsymbol{p}_{1:m}$ stacked in a matrix $\boldsymbol{P} \in \mathbb{R}^{m \times \ell}$.
- Each ℓ-dimensional row of $\boldsymbol{P}$ holds the features extracted for a given k-gram by different filters.
- Each m-dimensional column of $\boldsymbol{P}$ holds the features extracted across the sequence for a given filter.
- These columns are sometimes referred to as feature maps.



Stacking view (3:3)

- Now imagine the output vectors $\boldsymbol{p}_{1:m}$ stacked in a matrix $\boldsymbol{P} \in \mathbb{R}^{m \times \ell}$.
- Each ℓ-dimensional row of $\boldsymbol{P}$ holds the features extracted for a given k-gram by different filters.
- Each m-dimensional column of $\boldsymbol{P}$ holds the features extracted across the sequence for a given filter.
- These columns are sometimes referred to as feature maps.



Next up: pooling

- Next, in Part 3 of the CNN lecture we cover *pooling*.