Slides created by Sofia Eleftheriou

Ασκήσεις μελέτης B8
Lab 8

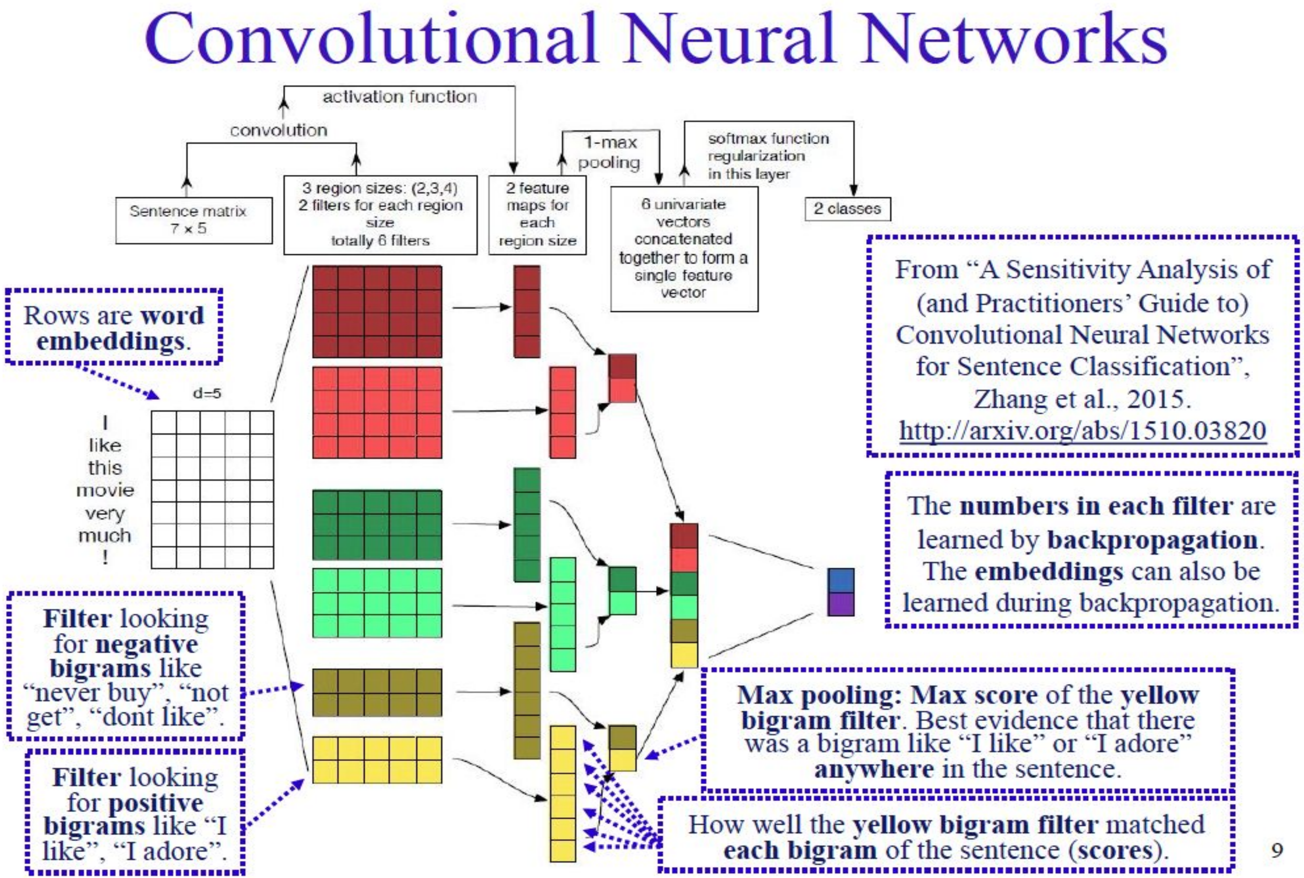
Human-Computer Interaction, AUEB
Εαρινό εξάμηνο 2022-2023

Lab Assistant: Sofia Eleftheriou

Άσκηση B8.1.

Γράψτε (όπως στις διαφάνειες 10–14) τις εξισώσεις του CNN της διαφάνειας 9.

Προσδιορίστε επίσης τις διαστάσεις όλων των εμπλεκομένων πινάκων και διανυσμάτων.



Απάντηση: The dimensionality of the word embeddings is $d = 5$. We can think of the two bigram filters as a matrix $W^{(2)} \in \mathbb{R}^{2\times 2d} = \mathbb{R}^{2\times 10}$ and a bias terms vector $b^{(2)} = \mathbb{R}^2$ (similarly to slide 12, where we have three bigram filters). Similarly, we can think of the two trigram filters as a matrix $W^{(3)} \in \mathbb{R}^{2\times 3d} = \mathbb{R}^{2\times 15}$ and a bias terms vector $b^{(3)} = \mathbb{R}^2$; and the two 4-gram filters as a matrix $W^{(4)} \in \mathbb{R}^{2\times 4d} = \mathbb{R}^{2\times 20}$ and a bias terms vector $b^{(4)} = \mathbb{R}^2$.

The embeddings of each bigram of the input text can be thought of as a vector $x^{(2)} \in \mathbb{R}^{2d}$. Applying the two bigram filters to the i-th bigram $x_i^{(2)}$ of the input text produces:

$$h_i^{(2)} = \mathrm{ReLU}\left(W^{(2)} x_i^{(2)} + b^{(2)}\right) \in \mathbb{R}^2, \qquad i = 1, \dots, 6$$

where we assumed that we use 'narrow convolutions', i.e., that the filters do not move out of the words of the input text (to partially overlap with padding tokens).

Max-pooling over $h_1^{(2)}, \dots, h_6^{(2)}$ produces a vector:

$$h^{(2)} = \langle \max_i h_{i,1}^{(2)}, \max_i h_{i,2}^{(2)} \rangle^T \in \mathbb{R}^2$$

ΔΙΑΣΤΑΣΗ $h^{(2)}$

- Input: n (number of words=7)
- Padding: p (=0)
- Stride: s (=1)
- Filter size: f (bi-gram=2)
- Output: [(n+2p-f)/s+1]

Similarly, applying the two trigram filters to the i-th trigram $x_i^{(3)} \in \mathbb{R}^{3d}$ of the input text and the two 4-gram filters to the i-th 4-gram $x_i^{(4)} \in \mathbb{R}^{4d}$ produces:

$$h_i^{(3)} = \mathrm{ReLU}\left(W^{(3)} x_i^{(3)} + b^{(3)}\right) \in \mathbb{R}^2, \qquad i = 1, \ldots, 5$$
$$h_i^{(4)} = \mathrm{ReLU}\left(W^{(4)} x_i^{(4)} + b^{(4)}\right) \in \mathbb{R}^2, \qquad i = 1, \ldots, 4$$

Max-pooling over $h_1^{(3)}, \ldots, h_5^{(3)}$ and over $h_1^{(4)}, \ldots, h_4^{(4)}$ produces:

$$h^{(3)} = \langle \max_i h_{i,1}^{(3)}, \max_i h_{i,2}^{(3)} \rangle^T \in \mathbb{R}^2$$
$$h^{(4)} = \langle \max_i h_{i,1}^{(4)}, \max_i h_{i,2}^{(4)} \rangle^T \in \mathbb{R}^2$$

The feature vector of the input text is the concatenation $h = \left[h^{(2)};\ h^{(3)};\ h^{(4)}\right]^T \in \mathbb{R}^6$.

We pass on h to a classifier, e.g., a logistic regression layer, i.e., a dense layer $W^{(P)} \in \mathbb{R}^{|C| \times 6}$ with a bias vector $b^{(P)} \in \mathbb{R}^{|C|}$ and a softmax activation function, to obtain a probability distribution $\vec{o}$ over the classes $c_1, \ldots, c_{|C|} \in C$:

$$\vec{o} = \langle P(c_1), \ldots, P\left(c_{|C|}\right) \rangle^{\mathrm{T}} = \mathrm{softmax}(W^{(P)} h + b^{(P)})$$

ΔΙΑΣΤΑΣΗ $h^{(3)}$

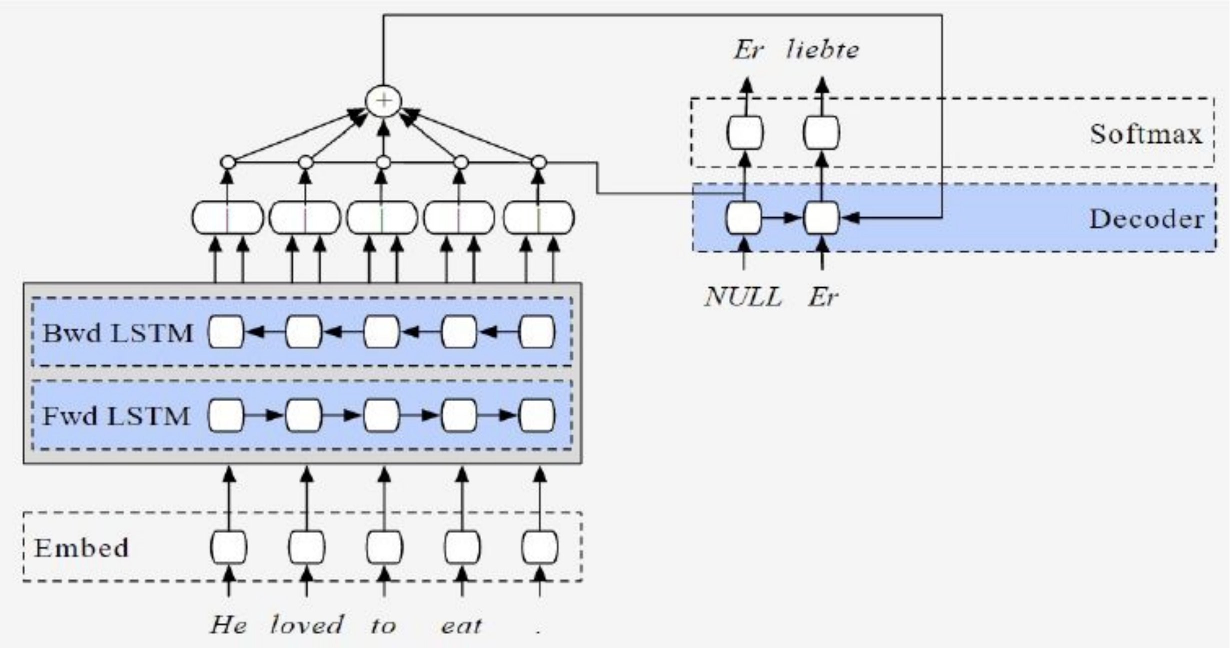
- Input: n (number of words=7)
- Padding: p (=0)
- Stride: s (=1)
- Filter size: f (tri-gram=3)
- Output: [(n+2p-f)/s+1]

ΔΙΑΣΤΑΣΗ $h^{(4)}$

- Input: n (number of words=7)
- Padding: p (=0)
- Stride: s (=1)
- Filter size: f (4-gram=4)
- Output: [(n+2p-f)/s+1]

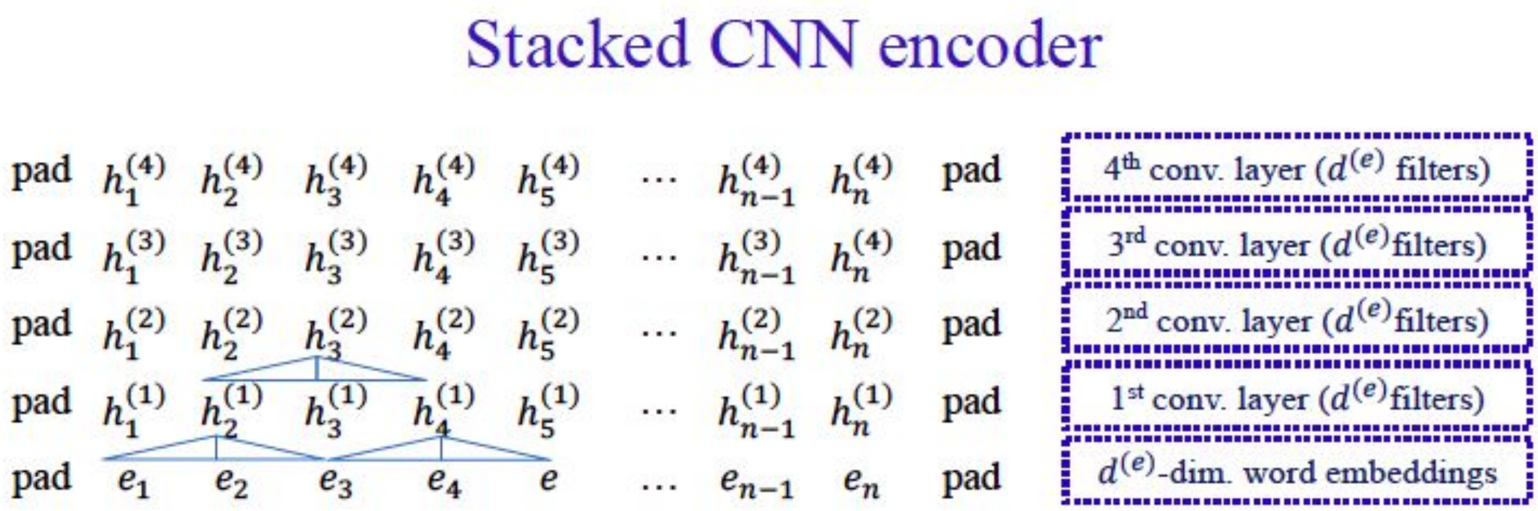
Άσκηση B8.2.

Consider the following LSTM-based machine translation model (see also exercise 4 of section B6).



Άσκηση B8.2.

We wish to replace the BiLSTM encoder of the model above by the stacked CNN-based encoder with trigram filters illustrated below, retaining the encoder-decoder attention and the LSTM decoder of the original model.



Άσκηση B8.2.

Let V, V' be the vocabularies of the source language (English) and target language (German), respectively. Each training instance is a pair consisting of (i) a sequence of one-hot vectors:

$$x_1, x_2, x_3, \dots, x_n \in \{0, 1\}^{|V|}$$

corresponding to an English sentence (each vector shows the position of the corresponding word in V) and (ii) a sequence of one-hot vectors:

$$y_1, y_2, y_3, \dots, y_m \in \{0, 1\}^{|V'|}$$

corresponding to a German sentence that is the correct (gold) translation of the English one (each vector shows the position of the corresponding word in V'). For simplicity, we assume all the English sentences are n words long, and all the German sentences are m words long.

Let $E \in \mathbb{R}^{d^{(e)} \times |V|}$ and $E' \in \mathbb{R}^{d^{(e)} \times |V'|}$ contain the word embeddings of the source and target language, respectively. Notice that word embeddings have $d^{(e)}$ dimensions in both languages, and that all the convolution layers of the CNN encoder also use $d^{(e)}$ filters.

The following formulae describe how the new model works and how the loss (L) is computed, given a training instance. **<u>Fill in the blanks</u> (they have been filled in in red in the solution).** The notation $[\dots;\dots]$ denotes concatenation and f, g denote activation functions.

Απάντηση:

Encoder: ($i \in \{1, 2, 3, \ldots, n\}$, $l \in \{2, 3, 4\}$)

$e_i = E\, x_i \in \mathbb{R}^{d^{(e)}}$ (Το embedding της σωστής αγγλικής λέξης στη θέση i.)

(Assume that $e_0 = e_{n+1}$ is always an all-zeros embedding of the padding token.)

$$h_i^{(1)} = \mathrm{ReLU}\left(W^{(1)}[e_{i-1}; e_i; e_{i+1}] + b^{(1)}\right) + e_i \in \mathbb{R}^{d^{(e)}}$$

$W^{(1)} \in \mathbb{R}^{d^{(e)} \times 3 \cdot d^{(e)}}$

$b^{(1)} \in \mathbb{R}^{d^{(e)}}$

$$h_i^{(l)} = \mathrm{ReLU}\left(W^{(j)}\left[h_{i-1}^{(l-1)}; h_i^{(l-1)}; h_{i+1}^{(l-1)}\right] + b^{(l)}\right) + h_i^{(l-1)} \in \mathbb{R}^{d^{(e)}}$$

$W^{(j)} \in \mathbb{R}^{d^{(e)} \times 3 \cdot d^{(e)}}$

$b^{(j)} \in \mathbb{R}^{d^{(e)}}$

Απάντηση:

Decoder: ($i \in \{1, 2, 3, \ldots, n\}$, $j \in \{1, 2, 3, \ldots, m\}$)

$t_j = E' y_j \in \mathbb{R}^{d^{(e)}}$ (Το embedding της σωστής γερμανικής λέξης στη θέση j.)

$z_j = \text{LSTM}(z_{j-1}, [t_{j-1}; c_j]) \in \mathbb{R}^{d^{(e)}}$ $z_0 \in \mathbb{R}^{d^{(e)}}$, $t_0 \in \mathbb{R}^{d^{(e)}}$

$\tilde{a}_{i,j} = v^T \cdot f(W^{(a)} [h_i^{(4)}; z_{j-1}] + b^{(a)}) \in \mathbb{R}$ $W^{(a)} \in \mathbb{R}^{d^{(a)} \times 2 \cdot d^{(e)}}$

$b^{(a)} \in \mathbb{R}^{d^{(a)}}$, $v \in \mathbb{R}^{d^{(a)}}$

$$a_{i,j} = \frac{\exp(\tilde{a}_{i,j})}{\sum_{i'} \exp(\tilde{a}_{i',j})}$$

$c_j = g(\sum_i a_{i,j} h_i^{(4)} + b^{(c)}) \in \mathbb{R}^{d^{(e)}}$ $b^{(c)} \in \mathbb{R}^{d^{(e)}}$

Απάντηση:

$$\tilde{o}_j = W^{(o)} z_j + b^{(o)} \in \mathbb{R}^{|V'|}$$

$$W^{(o)} \in \mathbb{R}^{|V'| \times d^{(e)}}$$
$$b^{(o)} \in \mathbb{R}^{|V'|}$$

$$o_{j,k} = \frac{\exp(\tilde{o}_{j,k})}{\sum_{k=1}^{|V'|} \exp(\tilde{o}_{j,k})}$$

(Πόσο πιθανό θεωρεί το μοντέλο η k-στή λέξη του γερμανικού λεξιλογίου να είναι η σωστή για την j-στή θέση της μετάφρασης.)

$$r_j = \text{argmax}_l \; y_{j,l}$$

(Σύμφωνα με το 1-hot y_j, η σωστή λέξη στην j-στή θέση της μετάφρασης βρίσκεται στη θέση r_j του γερμανικού λεξιλογίου.)

$$L = -\sum_j \log o_{j,r_j}$$

(Ελαχιστοποιώντας το L, μεγιστοποιούμε την πιθανότητα που δίνει το μοντέλο στις σωστές λέξεις, σε όλες τις θέσεις της μετάφρασης.)