CS 4248
Natural Language Processing

Professor NG Hwee Tou

Department of Computer Science
School of Computing
National University of Singapore
nght@comp.nus.edu.sg

Materials

- NNM4NLP Chapter 13

Specialized Neural Architectures

- Specialized for dealing with language data
  - Sequence data
  - Long distance dependency
    - "**John**, who is a student at NUS which is a top university in Asia located in a region experiencing high growth which is attributed to rapidly industrializing nations which …, **is** my best friend."

Specialized Neural Architectures

- Convolutional neural networks (CNN)
  - Identify informative ngrams
  - Consider local ordering patterns
- Recurrent neural networks (RNN)
  - Capture subtle patterns and regularities in sequences
  - Model non-Markovian dependencies
  - Consider infinite window and zoom in on informative sequential patterns in the window

Specialized Neural Architectures

- CNN and RNN are used as feature extractors
- Each network produces a vector (or a sequence of vectors) that is then fed into further parts of the network that eventually leads to prediction

Specialized Neural Architectures

- The network is trained end-to-end (the predicting part and the convolutional/recurrent architectures are trained jointly)
- The vectors from the convolutional/recurrent part capture aspects of the input useful for the prediction task
- Computation graph setup allows easy mixing of MLPs, CNNs, and RNNs as components

Convolutional Neural Networks

- Word order is important in NLP
- Feed-forward NN using CBOW (continuous bag-of-words representation) ignores word order
- “It was not good, it was actually quite bad.”
- “It was not bad, it was actually quite good.”

Convolutional Neural Networks

- Naïve solution: produce word vectors for bigrams and trigrams rather than words
- Problems:
  - Huge embedding matrix, and does not scale to longer ngrams
  - Data sparsity: no sharing between different ngrams (e.g., “quite good” and “very good” are completely independent ngrams)

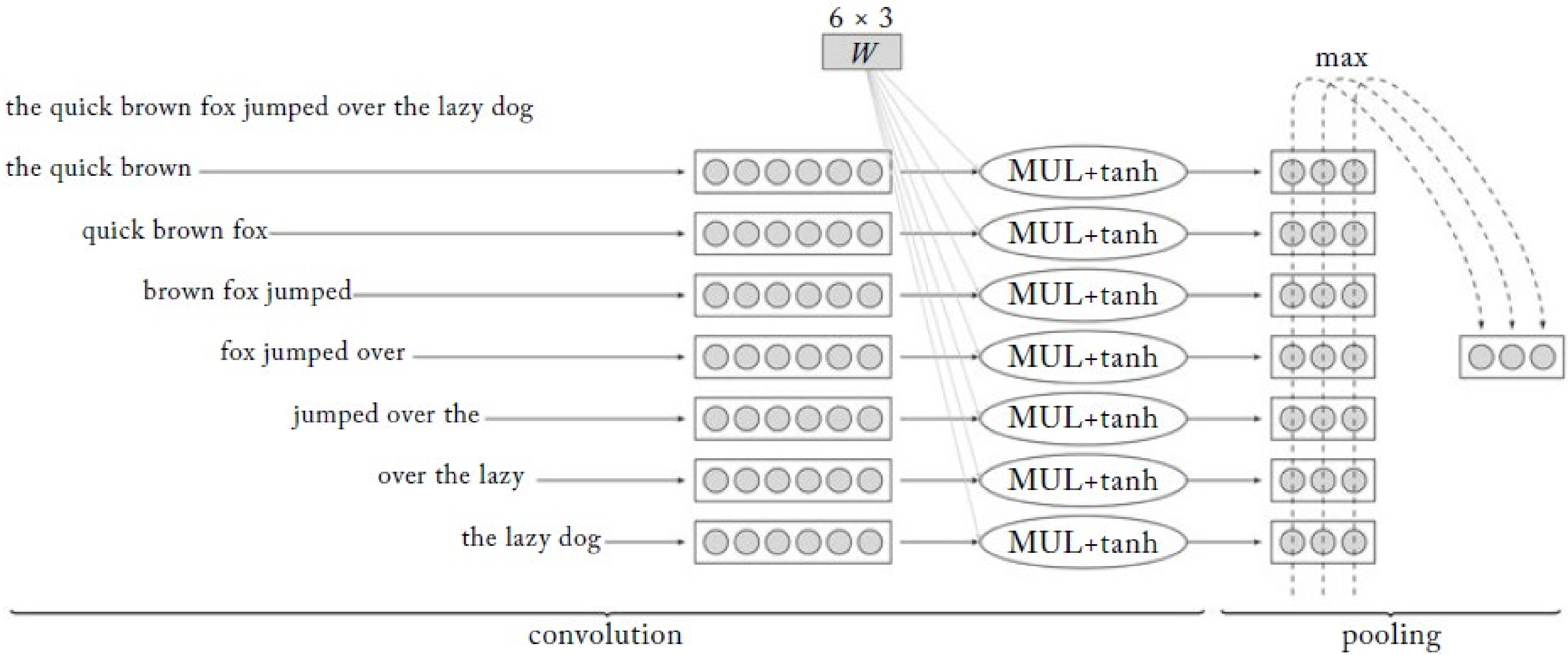
Convolutional Neural Networks

- CNN: a feature-extracting architecture
  - Extract meaningful substructures useful for the overall prediction task
- CNN was first successfully applied in computer vision as object detector, to recognize an object from a pre-defined category
- CNN terminology borrowed from computer vision community
- We focus here on 1D (sequence) convolutions

Convolutional Neural Networks

- Apply a nonlinear, learned function (a filter) over each k-word sliding window in the sentence
- l filters are applied to produce an l-dimensional vector for each sliding window of k words
- A pooling operation (e.g., max or average) then combines the vectors from the different windows into a single l-dimensional vector
- The single l-dimensional vector is fed further into a neural network for prediction

Convolutional Neural Networks



$k = 3$ $l = 3$

Convolutional Neural Networks

- Filter function learns to identify informative k-grams
- Pooling enables focusing on the most important aspects of a sentence, regardless of location

Convolutional Neural Networks

- Sentence (sequence of words) $w_{1:n} = w_1, \dots, w_n$
- Word embedding matrix $\boldsymbol{E}_{[w_i]} = \boldsymbol{w_i}$ $\quad \boldsymbol{E} \in \mathbb{R}^{|V| \times d}$
- Sliding window of size k (receptive field)
- $\boldsymbol{x_i} = \oplus\, (\boldsymbol{w_{i:i+k-1}}) = [\boldsymbol{w_i}; \boldsymbol{w_{i+1}}; \dots; \boldsymbol{w_{i+k-1}}]$ $\quad \boldsymbol{x_i} \in \mathbb{R}^{k \cdot d}$
- $\boldsymbol{u}$: a convolution filter or kernel
- g: nonlinear activation function
- $p_i = g(\boldsymbol{x_i} \cdot \boldsymbol{u})$ $\quad p_i \in \mathbb{R}$ $\quad \boldsymbol{u} \in \mathbb{R}^{k \cdot d}$

Convolutional Neural Networks

- l different filters $\boldsymbol{u_1}, \ldots, \boldsymbol{u_l}$
- $\boldsymbol{U} = \left[\boldsymbol{u_1}^{\mathrm{T}} \cdots \boldsymbol{u_l}^{\mathrm{T}}\right]$
- $\boldsymbol{p_i} = g(\boldsymbol{x_i} \cdot \boldsymbol{U} + \boldsymbol{b}) \quad \boldsymbol{p_i} \in \mathbb{R}^{l} \quad \boldsymbol{U} \in \mathbb{R}^{k \cdot d \times l} \quad \boldsymbol{b} \in \mathbb{R}^{l}$
- $\boldsymbol{p_i}$ is a collection of l values representing the ith window
- Each dimension of $\boldsymbol{p_i}$ captures a different kind of indicative information

Convolutional Neural Networks

- The convolution layer applies the same parameterized function (filters) $\boldsymbol{U}$ over all k-grams in the sentence
- Each $\boldsymbol{p_i}$ vector represents a particular k-gram in the sentence
- The representation $\boldsymbol{p_i}$ is sensitive to the identity and order of the words within a k-gram
- The same representation $\boldsymbol{p_i}$ is extracted for the same k-gram regardless of its position within the sentence

Convolutional Neural Networks

- m: number of $\boldsymbol{p_i}$ vectors ($i = 1, \ldots, m$)
- Narrow convolution (no padding)

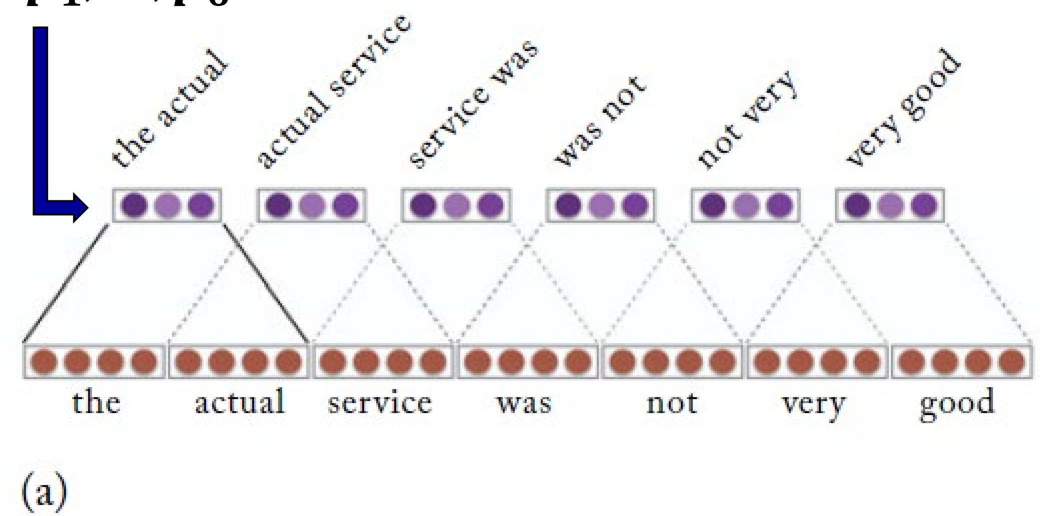
$$m = n - (k - 1) = n - k + 1$$

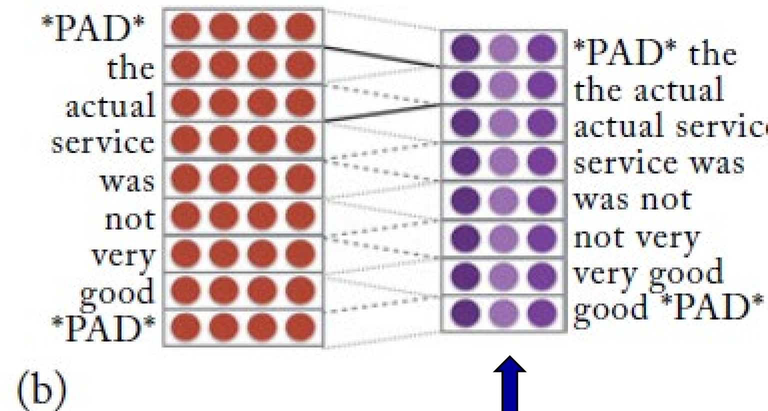
- Wide convolution (pad $k - 1$ words to each side)

$$m = n + (k - 1) = n + k - 1$$

Convolutional Neural Networks

$\boldsymbol{p_1}, \dots, \boldsymbol{p_6}$





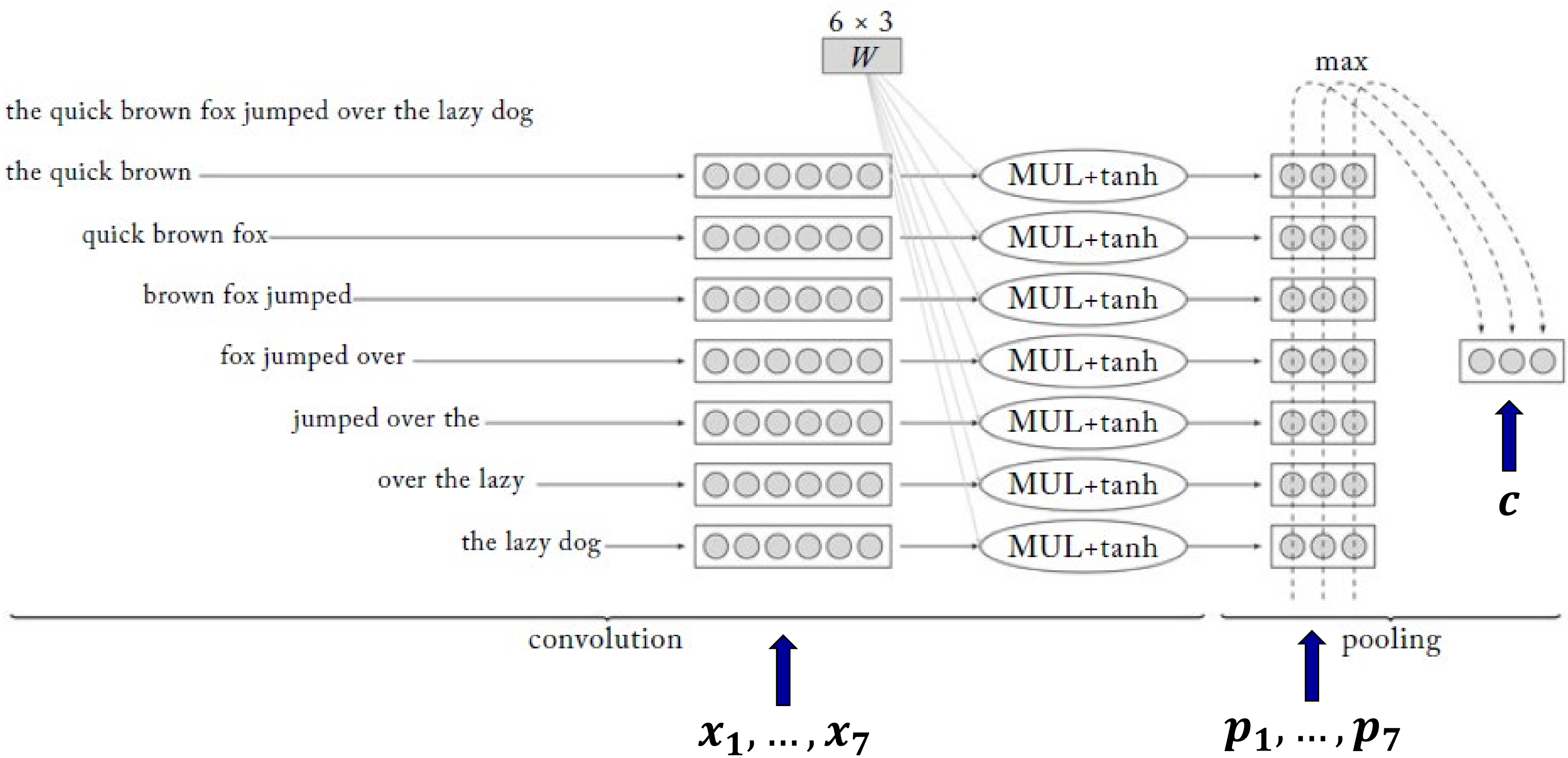
$\boldsymbol{p_1}, \dots, \boldsymbol{p_8}$

$n = 7, d = 4, k = 2, l = 3$

(a): Narrow convolution, $m = 6 = n - (k - 1)$

(b): Wide convolution, $m = 8 = n + (k - 1)$

Convolutional Neural Networks



Narrow convolution, $n = 9, d = 2, k = 3, l = 3, m = 7$

Convolutional Neural Networks

- Pooling: Combining $\boldsymbol{p_1}, \ldots, \boldsymbol{p_m}$ into a single vector $\boldsymbol{c}$
- $\boldsymbol{c}$ captures the essence of the important information in the sentence
- $\boldsymbol{c}$ is fed to downstream feed-forward NN (MLP) for prediction
- Training tunes the parameters in MLP, $\boldsymbol{U}$, $\boldsymbol{b}$, $\boldsymbol{E}$ such that $\boldsymbol{c}$ encodes information relevant to the prediction task

Convolutional Neural Networks

- Max pooling

$$\boldsymbol{c}_{[j]} = \max_{1 \le i \le m} \boldsymbol{p_i}_{[j]} \quad \forall j \in [1, l]$$

- Average pooling

$$\boldsymbol{c} = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{p_i}$$

Continuous bag of words (CBOW) of the k-gram representations