Effective Use of Word Order for Text Categorization with Convolutional Neural Network

Presenter: Yi-Hsin Chen

Text Categorization

- Automatically assign pre-defined categories to documents written in natural language
  - Sentiment Classification
  - Topic Categorization
  - Spam Detection

Previous Works

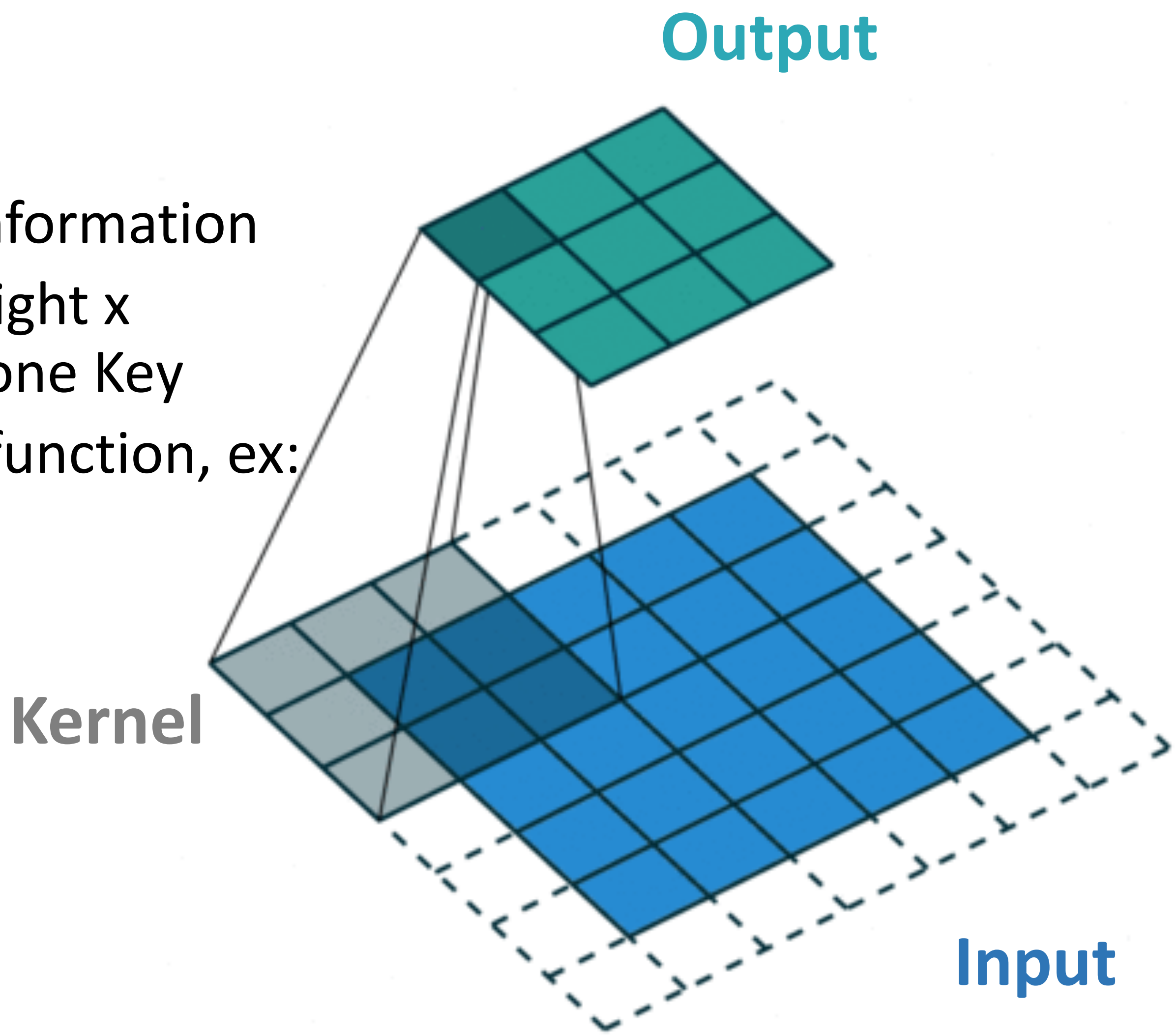
- First representing a document using a bag-of-n-gram vector and then using SVM for classification
  - Lose information of word order
- First converting words to vectors as the input, then using Convolutional Neural Network (CNN) for classification
  - CNN output will retain the word order information
  - The word embedding might need separate training and additional resources

N-Gram

- A set of co-occuring words within a given window
- For example, given a sentence “How are you doing”
  - For N=2, there are three 2-gram: “How are”, “are you”, “you doing”
  - For N=3, there are two 3-gram: “How are you”, “are you doing”

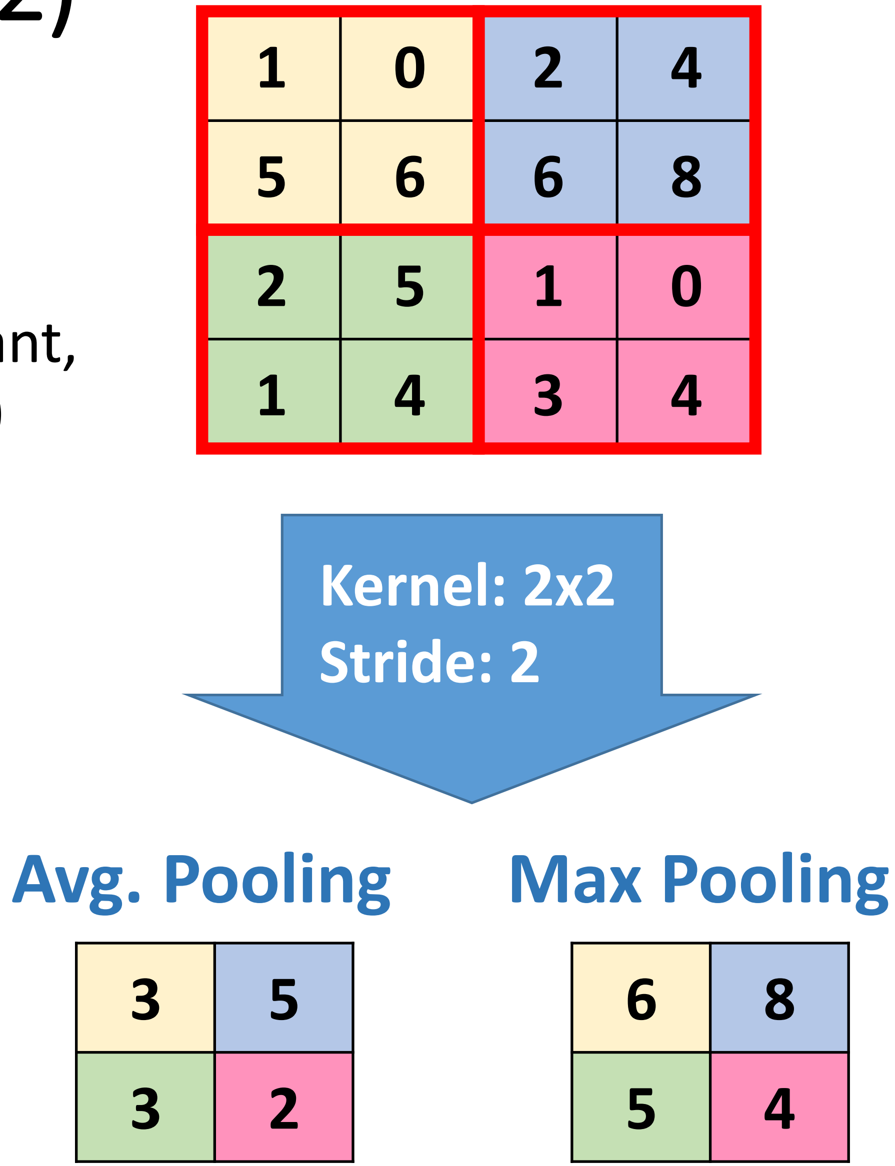
Convolutional Neural Network (1/2)

- Convolution Layer
  - The output will retain the location information
  - Usually the input is a 3-D matrix (Height x Width x Channel) rather than a 2-D one Key
  - Followed by a non-linear activation function, ex: reLU = max(0, x)
  - Key Parameters:
    - Kernel size
    - Stride / Padding
    - # of Kernel



Convolutional Neural Network (2/2)

- Pooling Layer
  - Pooling down-samples the input spatially
  - The pooling function could be any function you want, the two most common ones are: 1) Max Pooling 2) Average Pooling
  - Key Parameters:
    - Kernel Size
    - Stride / Padding



View Sentences as Images

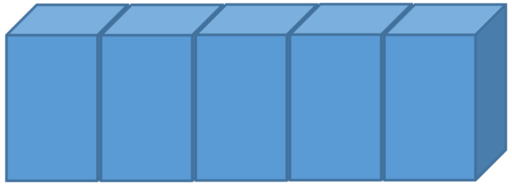
- View each word as a "pixel" of an image

Words Hi, how are you doing?

One-Hot Vectors

$$\underbrace{\left.\begin{pmatrix}0\\0\\1\\0\\0\end{pmatrix}\begin{pmatrix}0\\1\\0\\0\\0\end{pmatrix}\begin{pmatrix}0\\0\\0\\1\\0\end{pmatrix}\begin{pmatrix}0\\0\\0\\0\\1\end{pmatrix}\begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix}\right\}\text{V: \# of words in vocabulary}}_{\text{N: \# of words in the sentence}}$$

Stack Vectors to an "image"



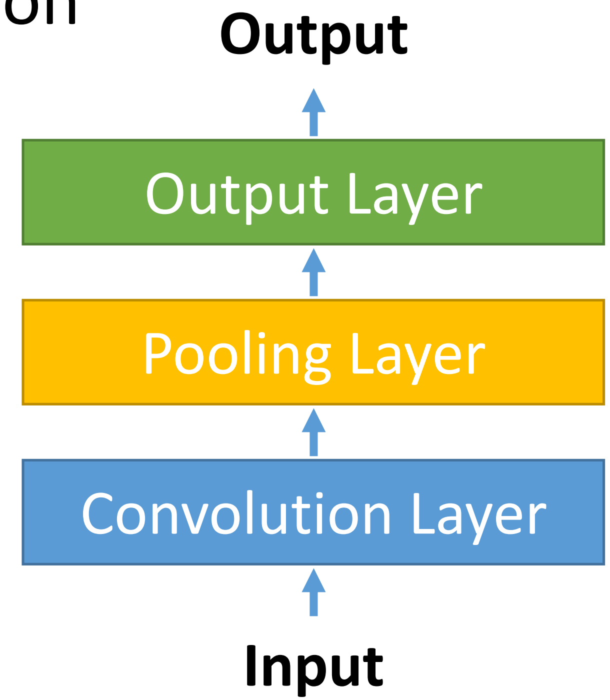
1 x N x V "Image"

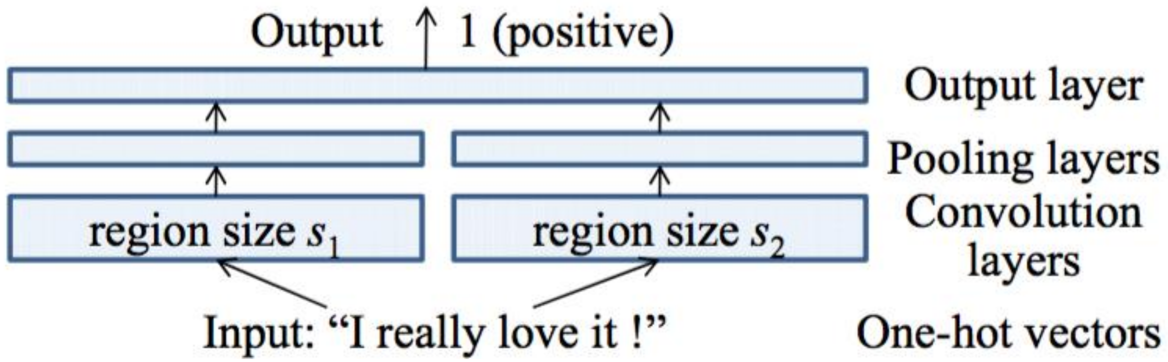
Apply CNN Hi, how are you doing?

1 x p kernel

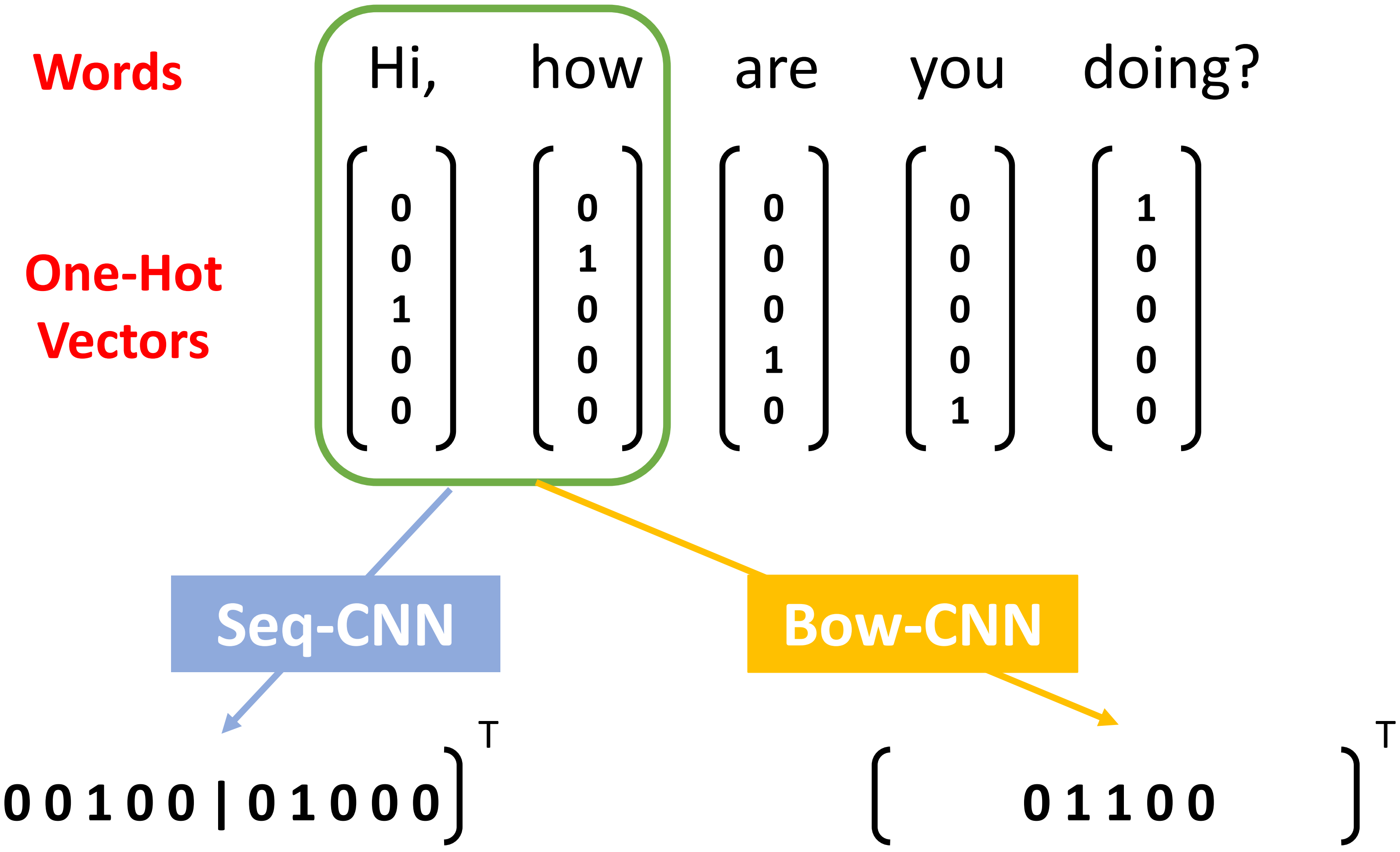
Proposed Models

- Directly apply CNN to learn the embedding of a text region
- Seq-CNN: treat each word as an entity
  - For a 1 x p kernel, there will be p x V parameters
  - Harder to train, easier to overfit
- Bow-CNN: treat p words as an entity
  - Reduce # of parameter from p x V to V
  - Lose the order information for these p words
- Parallel-CNN: use multiple CNNs in parallel to learn multiple types of embedding to improve performance





Seq-CNN v.s. Bow-CNN



Experiment

- Dataset
  - IMDB: movie review (Sentiment Classification)
  - Elec: electronics product reviews (Sentiment Classification)
  - RCV1 (topic categorization)
- Performance Benchmark (Error Rate)
  - The proposed models outperform B/L
  - **The model configuration for sentiment classification and topic categorization is quite different**

methods	IMDB	Elec	RCV1
SVM bow3 (30K)	10.14	9.16	10.68
SVM bow1 (all)	11.36	11.71	10.76
SVM bow2 (all)	9.74	9.05	10.59
SVM bow3 (all)	9.42	8.71	10.69
NN bow3 (all)	9.17	8.48	10.67
NB-LM bow3 (all)	8.13	8.11	13.97
bow-CNN	8.66	8.39	**9.33**
seq-CNN	8.39	7.64	9.96
seq2-CNN	8.04	7.48	–
seq2-bow$n$-CNN	**7.67**	**7.14**	–

Model Configuration for Different Tasks

- Sentiment Classification: a short phrase that conveys strong sentiment will dominate the results
  - Kernel size is small: 2~4
  - Using global max pooling
- Topic Categorization: need more context to provide information, the entire document matters, the location of text also matters
  - Kernel size is large : (20 for RCV1)
  - Using average pooling with 10 pooling units

CNN v.s. Bag-of-n-gram SVM (1/2)

- By directly learning the embedding of n-gram (n is decided by the kernel size), CNN is more able to utilize higher order n-gram for prediction

Model	CNN	SVM
Positive	Works perfectly! ,love this product Very pleased! I am pleased	Great, excellent, perfect, love, easy, amazing…
Negative	Completely useless., return policy It won’t even, but doesn’t work	Poor, useless, returned, not worth, return…

Predictive text region in the training set of Elec. dataset

CNN v.s. Bag-of-n-gram SVM (2/2)

- With the bag-of-n-gram representation, only the n-grams that appear in training data could help prediction
- For CNN, even a n-gram doesn’t appear in the training data, once its constituent words does, it could still be helpful for prediction

Model	CNN
Positive	Best concept ever, best idea ever, best hub ever, am wholly satisfied…
Negative	Were unacceptably bad, is abysmally bad, were universally poor…

Predictive text regions in the testing set which don’t appear in the training set

Thank You For Your Attention!!!