Dynamic Convolutional Neural Networks for Sentence Modeling and Sentiment analysis

Jayesh Kumar Gupta[1], Arpit Shrivastava[2]
Dr. Amitabha Mukherjee

[1]Department of Electrical Engineering, [2]Department of Computer Science and Engineering, Indian Institute of Technology Kanpur

Introduction

Language understanding is the central problem in natural language processing. Critical to this understanding is accurate representation of sentences. We use a novel architecture for neural networks dubbed the Dynamic Convolutional Neural Network (DCNN) for this semantic modeling of sentences. This allows us to handle sentences of varying lengths and capture short and long-range relations. The network is language agnostic as it does not rely on any parse tree. We use this model on the classic NLP problem of sentiment analysis of sentences. We apply this technique to analyze sentiment of labeled Hindi sentences and compare our results with existing methods.

Background

Related Work

- Text categorization techniques on subjective sentences
- Distributed bag-of-words model using word vectors
- Traversing the ontology to antonyms and synonyms to identify polarity shifts in the word space
- Distributional semantics or word vector models with tf-idf techniques [1]

Convolution

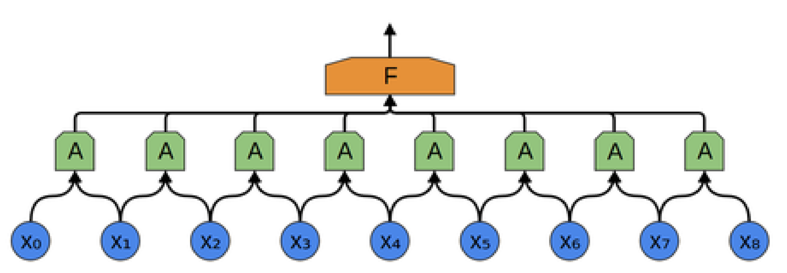


Figure 1: Convolution Layer [2]

- Types

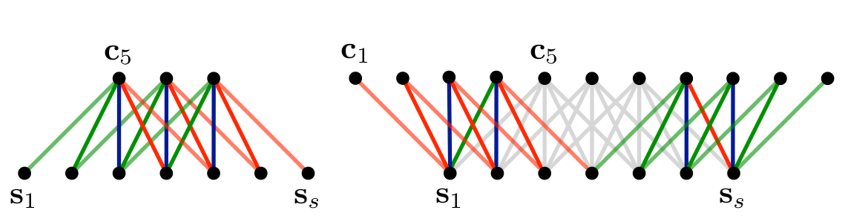


Figure 2: Narrow and wide types of convolution. The filter m has size m = 5 [3]

Pooling

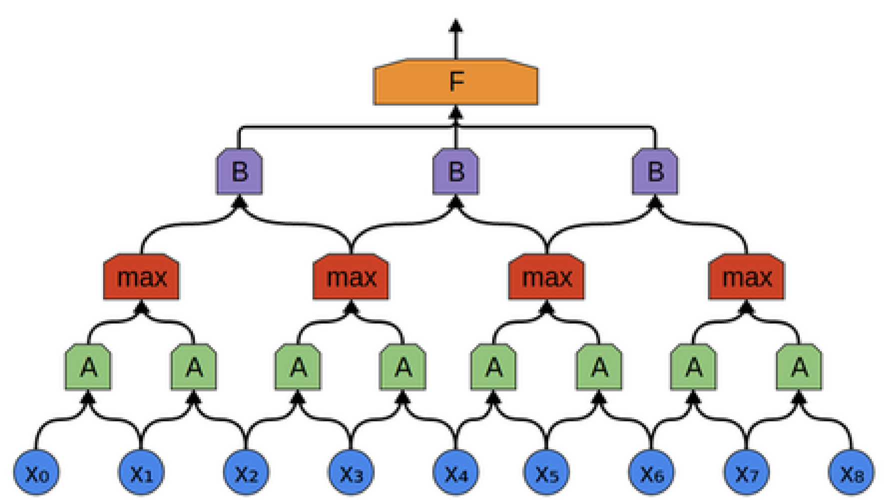
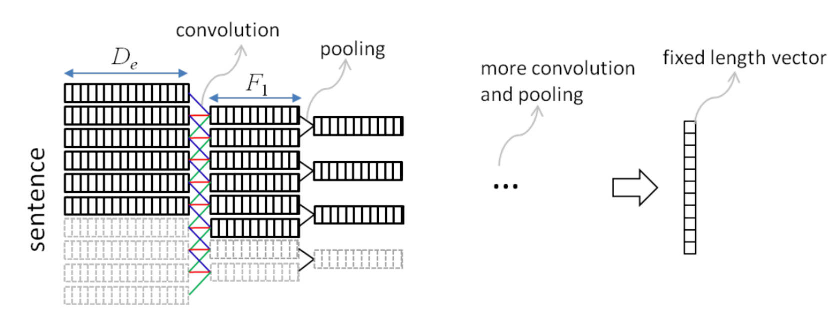


Figure 3: Pooling [2]

To make the network resilient to small transformations in the data and better generalization, we take the maximum of features over small blocks in the previous layer. This approach is termed as **max-pooling**.

Convolutional Neural Network



- Popular approach to make use of any local properties or symmetry present in the data by using convolution to look at only a segment of data to compute *features*.
- Allows expressing computationally large models with lesser number of parameters since many copies of the same neuron can be used.
- Each new layer allows us to learn more abstract features from the data.

Approach

Dynamic CNN

- Structure

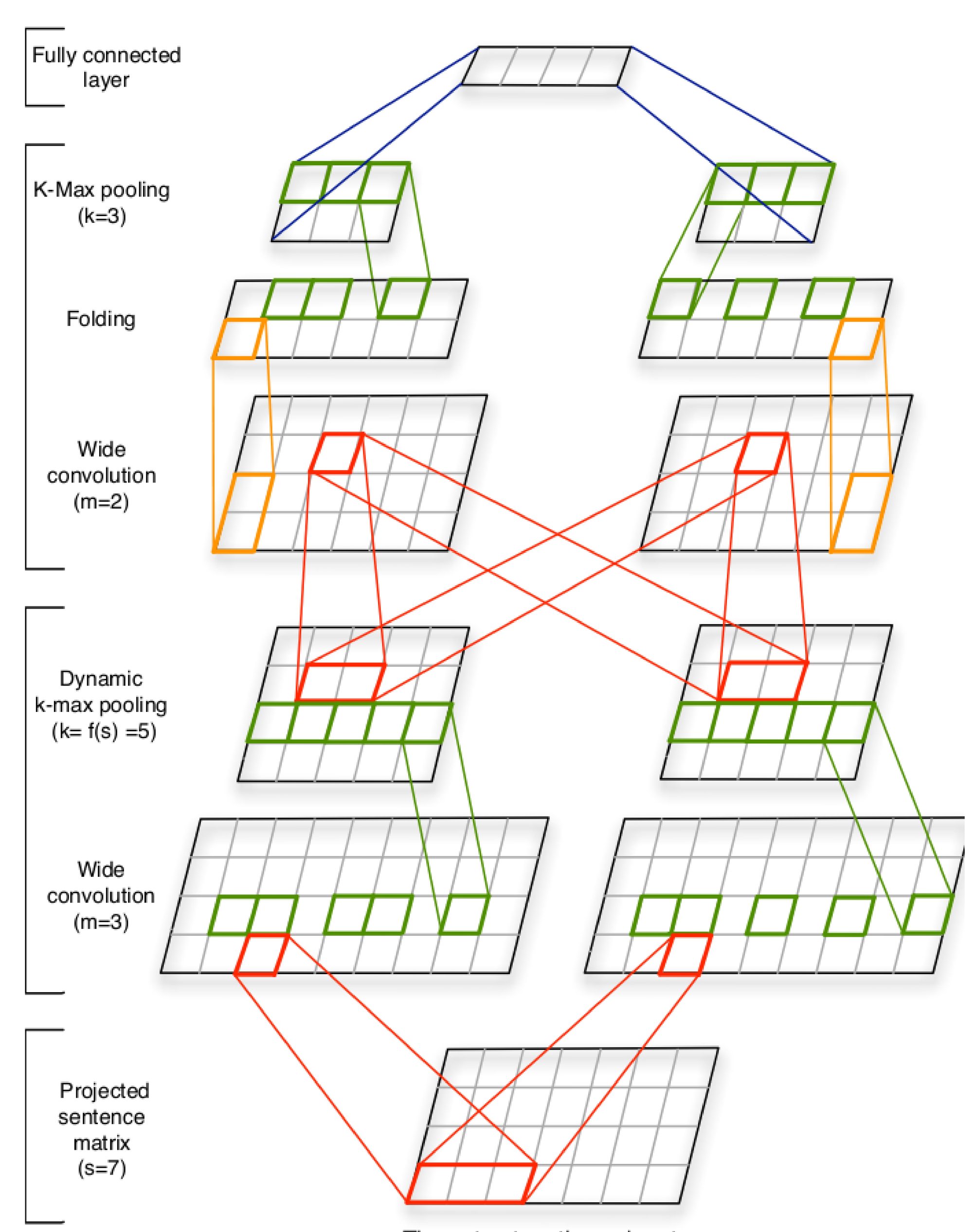


Figure 4: A DCNN for the seven word input sentence. Word embeddings have size d = 4. The network has two convolutional layers with two feature maps each. The widths of the filters at the two layers are respectively 3 and 2.

- Wide Convolution

$$c_j = m^T s_{j-m+1:j}$$

- Ensures even the words at margins are weighted.
- Ensures validity independently of sentence length, s.

Dynamic k-Max Pooling

- Instead of a selecting single feature from the previous layer, k most active features are selected. This allows us to select features that may be a number of positions apart and get finer discernment of number of times a features is highly activated.
- Input to the fully connected layer at the top is independent of the length of input sentence.
- k is a function of sentence length and network depth:

$$k_l = \max\left(k_{top}, \left\lceil \frac{L-l}{L} s \right\rceil\right)$$

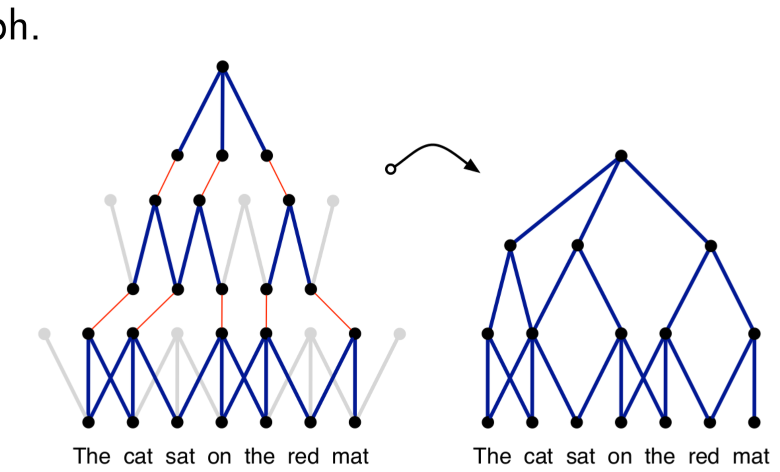
Non-linear Feature Function

We component-wise apply a bias $b \in \mathbb{R}^d$ and a non-linear function g to the result of k-max pooling and convolution.

- **Folding** - Halves the size of representation by by summing every two columns while introducing a dependence between different rows.

Sentence Model

- **Word and n-Gram Order**
  The network can **discriminate** whether a specific n-gram occurs in the input and **identifies** the relative position of most relevant n-grams.
- **Feature Graph** DCNN induces an internal feature graph over the input. Each subgraph may have a different shape that reflects the kind of relations that are detected in that subgraph.



Training

- Top layer of the network has a fully connected layer followed by a softmax non-linearity that predicts the probability distribution over classes given the input sentence.
- Network is trained to minimize the cross-entropy of the predicted and true distributions.
- We perform mini-batch back-propagation and gradient-based optimization using Adagrad [4] update rule.

Experiments

Dataset

We trained and tested our code on datasets taken from [5], [6]:
Product Review dataset (LTG, IIIT Hyderabad) containing 350 Positive reviews and 350 Negative reviews.
Movie Review dataset (CFILT, IIT Bombay) containing 127 Positive reviews and 125 Negative reviews.

Preprocessing

Preprocessing involved cleaning the reviews, extracting vocabulary, and representing these reviews as vectors of word indices. We initialize our word embeddings with random values.

Results

Experiment	Features	Accuracy
DCNN	CNN with dynamic k-max pooling	**71.5**
Word Vector with SVM [1]	tf-idf; word vector	89.97
MT Based using SVM [5]	tf-idf	65.96
In language using SVM [5]	tf-idf	78.14

References

[1] Amitabha Mukerjee Pranjal Singh.
Word vector averaging: Parserless approach to sentiment analysis.
regICON-2015: Regional Symposium on Natural Language Processing, March 2015.

[2] Christopher Olah.
Conv nets: A modular perspective.
https://colah.github.io/posts/2014-07-Conv-Nets-Modular/.
Accessed: 10-04-2015.

[3] Phil Blunsom, Edward Grefenstette, Nal Kalchbrenner, et al.
A convolutional neural network for modelling sentences.
In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.* Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014.

[4] John Duchi, Elad Hazan, and Yoram Singer.
Adaptive subgradient methods for online learning and stochastic optimization.
The Journal of Machine Learning Research, 12:2121–2159, 2011.

[5] Aditya Joshi., Balamurali A. R., and Pushpak Bhattacharyya.
A fall-back strategy for sentiment analysis in a new language: a case study for hindi.
In *International Conference on Natural Language Processing*, pages 1081–1091, 2010.

[6] A.R. Balamurali, Aditya Joshi, and Pushpak Bhattacharyya.
Cross-lingual sentiment analysis for Indian languages using linked wordnets.
In *Proc. of COLING 2012: Posters*, pages 73–82, Mumbai, India, December 2012.

[7] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen.
Convolutional neural network architectures for matching natural language sentences.
In *Advances in Neural Information Processing Systems*, pages 2042–2050, 2014.