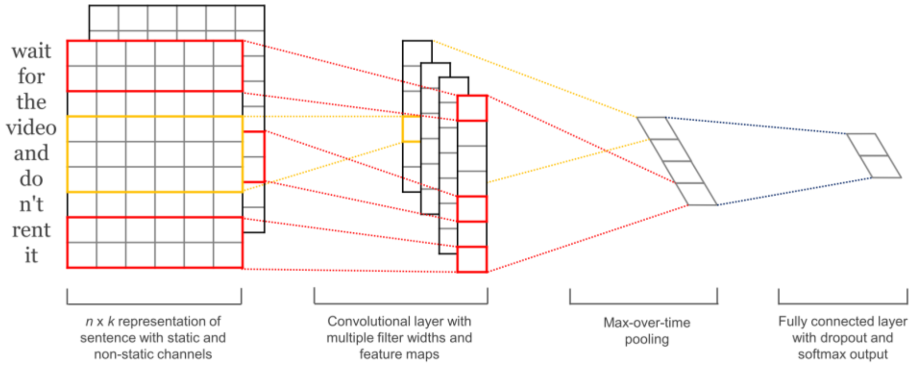
## Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.

The model presented in the paper achieves good classification performance across a range of text classification. And it has since become a standard baseline for new text classification architectures.


wait
for
the
video
and
do
n't
rent
it
*n* x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

The first layers embeds words into low-dimensional vectors. The next layer performs convolutions over the embedded word vectors using multiple filter sizes. For example, sliding over 3, 4 or 5 words at a time. Next, we max-pool the result of the convolutional layer into a long feature vector, add dropout regularization, and classify the result using a softmax layer.

Denny Britz has an implementation of the model in TensorFlow.

Here are the basic steps:

1. *Pre-processing*

- Load positive and negative sentences from the raw data files.
- Clean the text data using the same code as the original paper.
- Pad each sentence to the maximum sentence length, which turns out to be 59. We append special <PAD> tokens to all other sentences to make them 59 words. Padding sentences to the same length is useful because it allows us to efficiently batch our data since each example in a batch must be of the same length.
- Build a vocabulary index and map each word to an integer between 0 and 18,765 (the vocabulary size). Each sentence becomes a vector of integers.

2. *Generate TextCNN class*

- Input Placeholders: define the input data that we pass to our network
- Embedding Layer: map vocabulary word indices into low-dimensional vector representations
- Convolution and Max-Pooling Layers: Each convolution produces tensors of different shapes we need to iterate through them, create a layer for each of them, and then merge the results into one big feature vector.

- Dropout Layer: This prevent neurons from co-adapting and forces them to learn individually useful features.
- Scores and Predictions: generate predictions by doing a matrix multiplication and picking the class with the highest score.
- Loss and Accuracy: The loss is a measurement of the error our network makes, and our goal is to minimize it. The standard loss function for categorization problems it the cross-entropy loss.

### 3. *Instantiating the CNN and minimizing the loss*

By defining a global_step variable and passing it to the optimizer we allow TensorFlow handle the counting of training steps for us. The global step will be automatically incremented by one every time you execute train_op.

## Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences[J]. arXiv preprint arXiv:1404.2188, 2014.

This article proposed a network model called DCNN (Dynamic Convolutional Neural Network). In Kim's experimental results part also verify the effectiveness of this model. The subtlety of this model is the way Pooling, using a method called Dynamic Pooling.

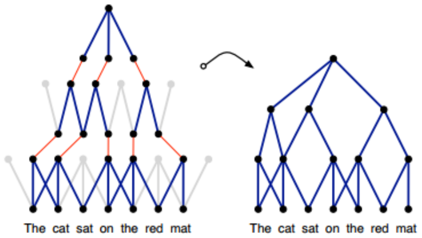


Figure 1: Subgraph of a feature graph induced over an input sentence in a Dynamic Convolutional Neural Network. The full induced graph has multiple subgraphs of this kind with a distinct set of edges; subgraphs may merge at different layers. The left diagram emphasises the pooled nodes. The width of the convolutional filters is 3 and 2 respectively. With dynamic pooling, a filter with small width at the higher layers can relate phrases far apart in the input sentence.

It can be seen that the bottom layer is passed up by combining the adjacent word information, and the upper layer is combined with the new Phrase information, so that the words in the sentence are separated Behavior (or some kind of semantic connection). From an intuitive point of view, this model can extract the important semantic information (through Pooling) in the sentence by the combination of words.

The convolution layer in the network uses a way called Wide Convolution, followed by a dynamic k-max pooling layer. The size of the output of the intermediate convolution layer, the size of the Feature Map, varies depending on the length of the input sentence. Here are some details of these operations:

1. ***Wide Convolution***

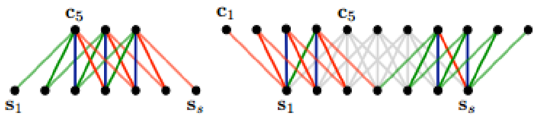


Figure 2: Narrow and wide types of convolution. The filter $\mathbf{m}$ has size $m = 5$.

Compared to the traditional convolution operation, the width of the Feature Map of the wide convolution output is wider because the convolution window does not need to cover all the input values, or it can be part of the input value. The remaining input value is 0.

2. ***k-max pooling***

The advantage of k-max pooling is that it extracts both the more important information (more than one) in the sentence, while preserving their order information (relative position).

3. ***Dynamic k-max pooling***

Dynamic k-max pooling operation, where k is a function that takes two parameters: the length of the sentence and the depth of the network, as follows:

$$K_l = \max(K_{top}, \frac{L-l}{L} * s)$$

4. ***Folding***

Folding operation is to consider the relationship between two adjacent lines. The way is to add the two lines of the vector. The operation did not increase the number of parameters, but consider some kind of association between rows and rows in the feature matrix ahead of the time.

## Zhang Y, Wallace B. A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification[J]. arXiv preprint arXiv:1510.03820, 2015.

It has done an extensive analysis of model variants (e.g. filter widths, k-max pooling, word2vec vs Glove, etc.) and their effect on performance.

***Based on the Kim Y model made a lot of tuning after the conclusion of the experiment:***

1. Due to the randomness factors in the model training process, such as random initialization weight parameters, mini-batch, stochastic gradient descent optimization algorithm, the result of

the model on the data set has some floating, such as accuracy (accuracy) can reach 1.5 % Of the float, while the AUC is 3.4% floating.

2. Word vector is the use of word2vec or GloVe, the experimental results have a certain impact, which is better depends on the specific task itself.
3. Filter size of the model performance has a greater impact, and Filter parameters should be updated.
4. The number of Feature Map also has some effect, but it needs to take into account the training efficiency of the model.
5. max pooling way is already good enough, compared to other pooling methods
6. The role of regularization is minimal.

***It provides the following guidance regarding CNN architecture and hyperparameters:***

1. Non-static word2vec or GloVe is really better than one-hot vectors in the most time.
2. Line-search over the single filter region size to find the 'best' single region size.
3. Alter the number of feature maps for each filter region size from 100 to 600.
4. ReLU and tanh are the best activation functions overall.
5. Use 1-max pooling.
6. Dropout out rate larger than 0.5.
7. Cross-fold validation procedure should be performed and variances and ranges should be considered.
8. The constraints had little effect on the end result.