CNN for Sentence Classification

Group Members

IEC2016012 (Nikhil Mundra)

IEC2016026 (Agam Dwivedi)

IEC2016072 (Ruchin Agrawal)

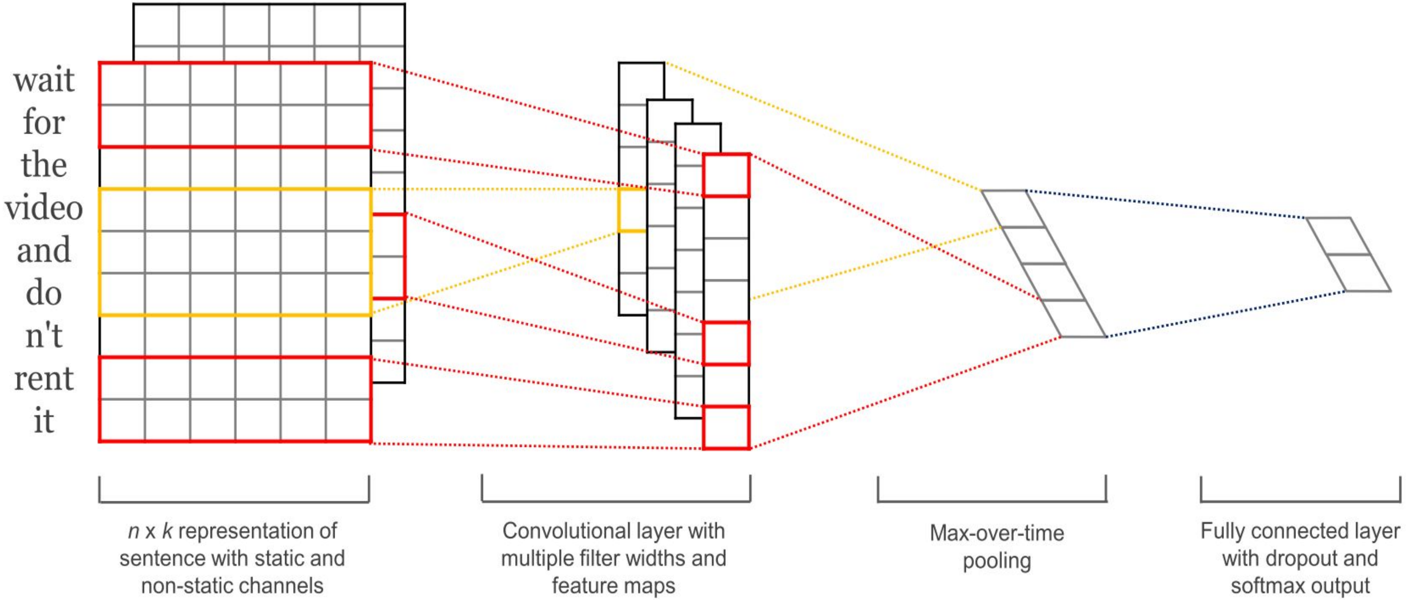
IIT2016509 (Mohammad Aquib)

Project Aim

- To train a Convolutional Neural Network on top of word vectors obtained from an unsupervised neural language model.
- The classification involves classifying sentences in multiple classes, such as positive or negative, subjective or objective, etc.
- The target is to improve upon the state-of-the-art techniques by using simple CNN with little hyperparameter tuning and static vectors.

- The vectors were obtained from the Google News dataset trained on 100B words using a word2vec model.
- Initially, we tune the parameters of the model, as well as initialize the word embeddings as random.
- For the sentence classification part, we have used Pang and Lee's movie review dataset.

Architecture



- A convolution operation involves a filter w = hxk, applied to a window of h words to produce a new feature c_i.
- For example ci is generated from a window of words $x_{i:i+h}$ by
- ci = f(w.$x_{i:i+h}$ + b), where b is a bias term, f is a non-linear function such as hyperbolic tangent.
- The model uses multiple filters (with varying window sizes) to obtain multiple features. These features form the penultimate layer and are passed to a fully connected softmax layer whose output is the probability distribution over labels.

Procedure

- We have used only one Convolutional layer with varying filter widths of size 3x3, 4x4 and 5x5.
- We used 100 filters with a stride of 1. Max pooling was performed after convolution followed by flattening with the dropout of 0.5.
- Finally, we have used the softmax activation to categorize the sentiment of the reviews accordingly

Regularization

- Dropout prevents co-adaptation of hidden units by randomly dropping out or setting to zero a proportion of the hidden units during forward backpropagation.
- Instead of using y -> w.z + b for output unit y in the forward propagation, the dropout uses y-> w.(z * r) + b, where * is the element wise multiplication. And
- r is masking vector of bernoulli random variables with probability p of being 1.

Results

Static vector : Variable
Task-specific vector: Random

```
loading data... data loaded!
model architecture: CNN-non-static
using: random vectors
[('image shape', 64, 300), ('filter shape', [(100, 1, 3, 300), (100, 1, 4, 300), (100, 1, 5, 300)]), ('hidden_units', [100, 2])
WARNING (theano.tensor.blas): We did not find a dynamic library in the library_dir of the library we use for blas. If you use A
... training
epoch: 1, training time: 346.71 secs, train perf: 62.68 %, val perf: 60.21 %
epoch: 2, training time: 347.31 secs, train perf: 71.75 %, val perf: 64.11 %
epoch: 3, training time: 342.70 secs, train perf: 84.27 %, val perf: 70.53 %
epoch: 4, training time: 343.03 secs, train perf: 91.73 %, val perf: 73.68 %
epoch: 5, training time: 342.86 secs, train perf: 95.78 %, val perf: 73.79 %
epoch: 6, training time: 341.80 secs, train perf: 99.19 %, val perf: 75.26 %
epoch: 7, training time: 344.79 secs, train perf: 99.75 %, val perf: 76.00 %
cv: 0, perf: 0.7458256029684601
0.7458256029684601
```

Results

Static vector : Constant
Task-specific vector: word2vec

```
loading data... data loaded!
model architecture: CNN-non-static
using: word2vec vectors
[('image shape', 64, 300), ('filter shape', [(100, 1, 3, 300), (100, 1, 4, 300), (100, 1, 5, 300)]), ('hidden_units', [100, 2]), ('dropout', [0.5]), ('batch_size', 50
WARNING (theano.tensor.blas): We did not find a dynamic library in the library_dir of the library we use for blas. If you use ATLAS, make sure to compile it with dyna
... training
epoch: 1, training time: 284.91 secs, train perf: 81.12 %, val perf: 79.47 %
epoch: 2, training time: 277.31 secs, train perf: 86.67 %, val perf: 80.21 %
epoch: 3, training time: 272.31 secs, train perf: 91.92 %, val perf: 80.32 %
epoch: 4, training time: 272.83 secs, train perf: 94.65 %, val perf: 79.58 %
epoch: 5, training time: 279.27 secs, train perf: 98.02 %, val perf: 82.11 %
epoch: 6, training time: 273.18 secs, train perf: 99.27 %, val perf: 80.84 %
epoch: 7, training time: 273.50 secs, train perf: 99.61 %, val perf: 81.68 %
cv: 0, perf: 0.8157407407407408
0.8157407407407408
```

Results

Static vector : Variable
Task-specific vector: word2vec

```
loading data... data loaded!
model architecture: CNN-static
using: word2vec vectors
[('image shape', 64, 300), ('filter shape', [(100, 1, 3, 300), (100, 1, 4, 300), (100, 1, 5, 300)]), ('hidden_units', [100, 2]), ('dropout', [0.5]), ('batch_size', 50
WARNING (theano.tensor.blas): We did not find a dynamic library in the library_dir of the library we use for blas. If you use ATLAS, make sure to compile it with dyna
... training
epoch: 1, training time: 234.72 secs, train perf: 80.36 %, val perf: 79.37 %
epoch: 2, training time: 234.79 secs, train perf: 84.66 %, val perf: 81.16 %
epoch: 3, training time: 234.57 secs, train perf: 83.97 %, val perf: 76.00 %
epoch: 4, training time: 234.67 secs, train perf: 91.51 %, val perf: 81.47 %
epoch: 5, training time: 234.74 secs, train perf: 93.06 %, val perf: 79.37 %
epoch: 6, training time: 234.43 secs, train perf: 96.49 %, val perf: 81.05 %
epoch: 7, training time: 234.69 secs, train perf: 95.44 %, val perf: 79.47 %
cv: 0, perf: 0.819811320754717
0.819811320754717
```

Results

With varying filter sizes, accuracy obtained after 10 epochs:

Filter Size	**Accuracy**
3x3	76.2%
4x4	79.8%
5x5	77.5%

Results

Varying the activation function used after the single convolutional layer

Activation Function	Hidden Units	Accuracy
TanH	100	82.74%
ReLu	100	79.8%
Sigmoid	100	78%
ReLu	50	80.2%

References

- Yoon Kim. 2014. CNN for Sentence Classification.
- Le, T. Mikolov. 2014. Distributed Represenations of Sentences and Documents.
- Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality.