Encoding Words

One-Hot Encoding

Label
0
4
4
3
0
9

0	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1

Bag of Words

Input
Text

"Bag of Words"

	a	…	hate	I	iPhone	love	my	…	zoo
I love my iPhone →	0	…	0	1	1	1	1	…	0
I hate my iPhone →	0	…	1	1	1	0	1	…	0

Word Embedding

I love this movie

I	love	this	movie
0.4	9.7	-0.7	0.4
23.4	5.4	-5.2	-3.2
-0.6	-9.4	-2.1	-9.9
-2.4	-1.2	-0.9	-1.9

Embeddings

	Val 1	Val 2	Val 3	Val 4
a	0.1	-0.3	1.7	2.4
aardvark	-2.3	4.1	-5.2	3.1
…	…	…	…	…
<unknown>	0.3	0.9	0.8	0.2

Pre-computed Encodings

GloVe: Global Vectors for Word Representation

Jeffrey Pennington, Richard Socher, Christopher D. Manning

Introduction

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

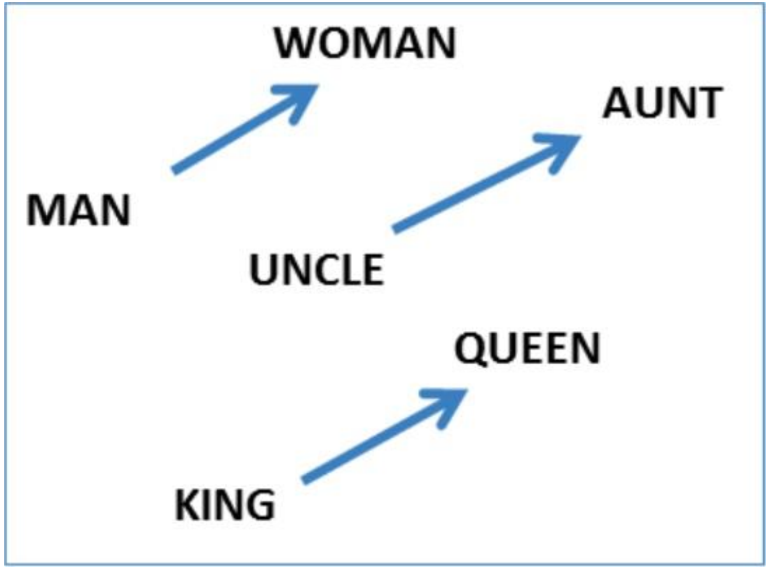
Getting started (Code download)

- Download the code (licensed under the Apache License, Version 2.0)
- Unpack the files: unzip GloVe-1.2.zip
- Compile the source: cd GloVe-1.2 && make
- Run the demo script: ./demo.sh
- Consult the included README for further usage details, or ask a question
- The code is also available on GitHub

Download pre-trained word vectors

- Pre-trained word vectors. This data is made available under the Public Domain Dedication and License v1.0 whose full text can be found at: http://www.opendatacommons.org/licenses/pddl/1.0/.
  - Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab, uncased, 50d, 100d, 200d, & 300d vectors, 822 MB download): glove.6B.zip
  - Common Crawl (42B tokens, 1.9M vocab, uncased, 300d vectors, 1.75 GB download): glove.42B.300d.zip
  - Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download): glove.840B.300d.zip
  - Twitter (2B tweets, 27B tokens, 1.2M vocab, uncased, 25d, 50d, 100d, & 200d vectors, 1.42 GB download): glove.twitter.27B.zip
- Ruby script for preprocessing Twitter data

GloVe + word2vec



Word Analogy Task

man is to *woman* as *king* is to ___ ?

good is to *best* as *smart* is to ___ ?

china is to *beijing* as *russia* is to ___ ?

Turns out the word-context based vector model we just learnt is good for such analogy tasks,

[king] – [man] + [woman] ≈ [queen]

Microsoft Levy, Goldberg, and Israel, *Linguistic Regularities in Sparse and Explicit Word Representations*, CoNLL 2014.

Convolutions

2D Convolution Review

52	34	14	5
45 1	12 0	17 0.2	11
29 0.4	20 0.7	19 0.1	27
99 0.2	85 0	60 0.8	55
120	112	88	29

Input * Kernel

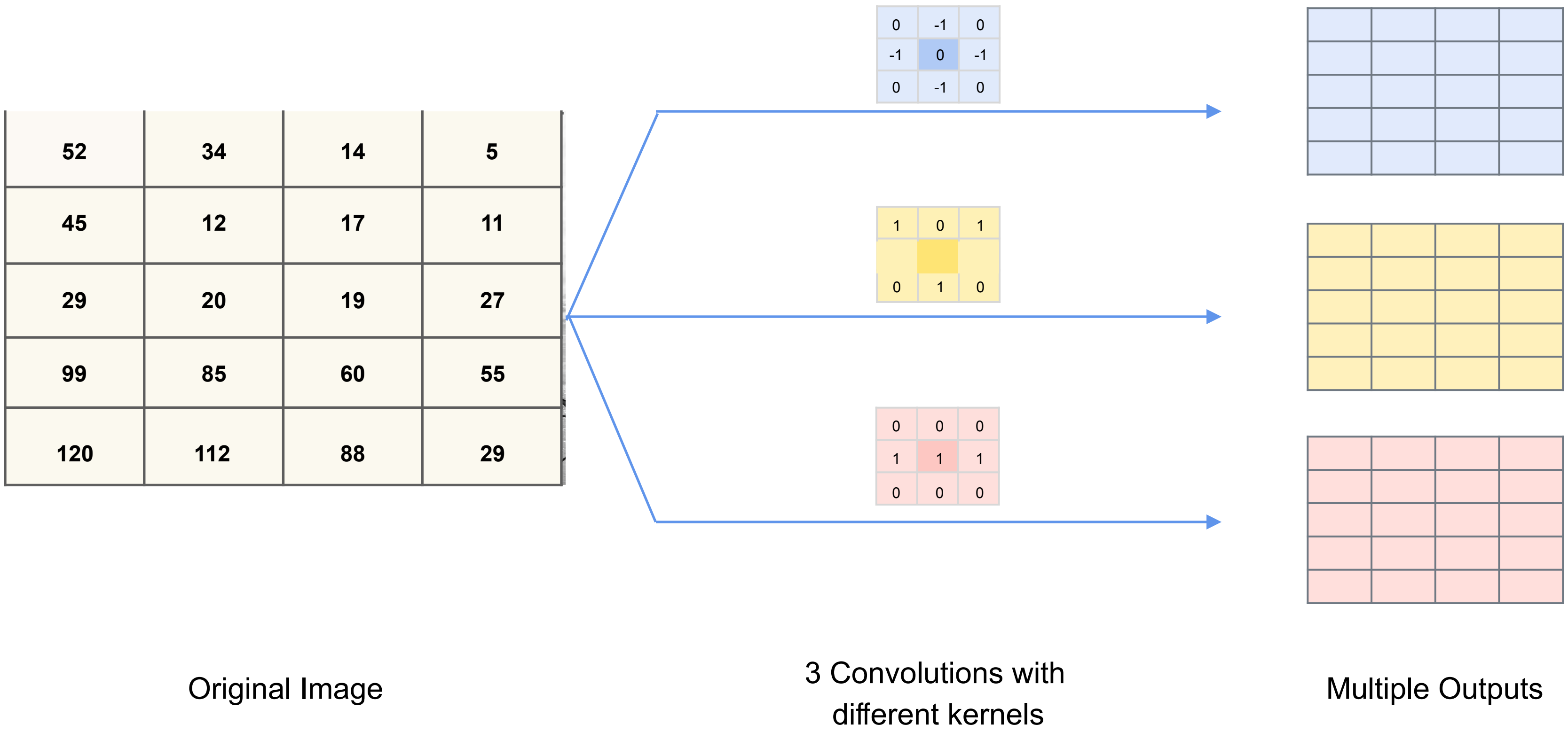
(45 x 1)
(12 x 0)
(17 x 0.2)
(29 x 0.4)
(20 x 0.7)
(19 x 0.1)
(99 x 0.2)
(85 x 0)
+ (60 x 0.8)

143.7

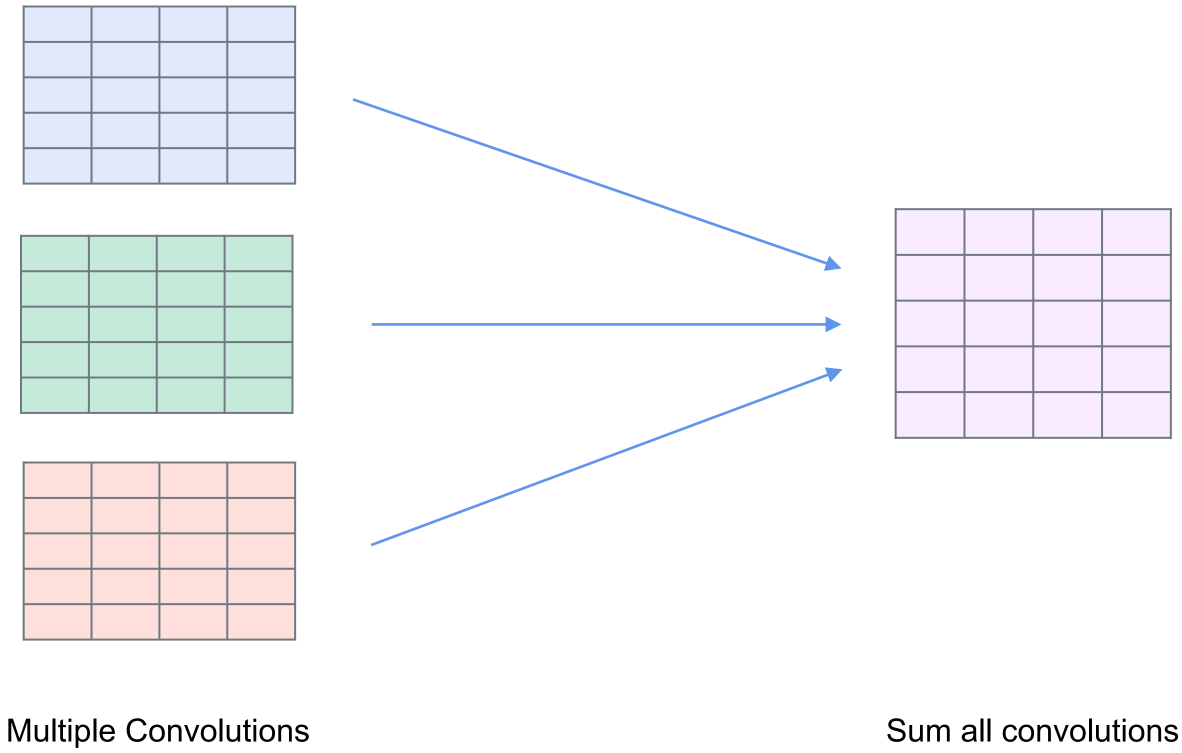
131	122	97	17
114	104	78	88
	144		

Output

2D Convolution Review - Multiple Outputs



2D Convolution Review - Multiple Inputs



1D Convolution

	I	love	this	movie
Channel 1	0.4 0.2	9.7 0.3	-0.7 0.2	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

0.4*0.2 + 9.7*0.3 + (-0.7)*0.2
= 2.85

Conv 1
2.85

1D Convolution

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4 0.2	5.4 0.3	-5.2 0.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

23.4*0.2 + 5.4*0.3 + (-5.2)*0.2
= 5.26

Conv 1
2.85
5.26

1D Convolution

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6 0.2	-9.4 0.3	-2.1 0.2	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

(-0.6)*0.2 + (-9.4)*0.3 + (-2.1)*0.2
= -3.36

Conv 1
2.85
5.26
-3.36

1D Convolution

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4 0.2	-1.2 0.3	-0.9 0.2	-1.9

$(-2.4)*0.2 + (-1.2)*0.3 + (-0.9)*0.2$
$= -1.02$

Conv 1
2.85
5.26
-3.36
-1.02

1D Convolution

	I	love	this	movie
Channel 1	0.4	9.7 0.3	-0.7 0.3	0.4 0.1
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

Conv 1	Conv 2
2.85	2.74
5.26	
-3.36	
-1.02	

9.7*0.3 + (-0.7)*0.3 + 0.4*0.1
= 2.74

1D Convolution

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4 0.3	-5.2 0.3	-3.2 0.1
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

Conv 1	Conv 2
2.85	2.74
5.26	-0.26
-3.36	
-1.02	

$$5.4*0.3 + (-5.2)*0.3 + (-3.2)*0.1 = -0.26$$

1D Convolution

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4 0.3	-2.1 0.3	-9.9 0.1
Channel 4	-2.4	-1.2	-0.9	-1.9

Conv 1	Conv 2
2.85	2.74
5.26	-0.26
-3.36	-4.44
-1.02	

$$(-9.4)*0.3 + (-2.1)*0.3 + (-9.9)*0.1 = -4.44$$

1D Convolution

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2 0.3	-0.9 0.3	-1.9 0.1

Conv 1	Conv 2
2.85	2.74
5.26	-0.26
-3.36	-4.44
-1.02	-0.82

$(-1.2)*0.3 + (-0.9)*0.3 + (-1.9)*0.1$
$= -0.82$

Max Pooling

2D Max Pooling Review

52	34	14	5
45	12	17	11
29	20	19	27
99	85	60	55
118	103	180	192
120	112	88	29

Input 6x4

Output 3x2

52	17		

2D Max Pooling Review

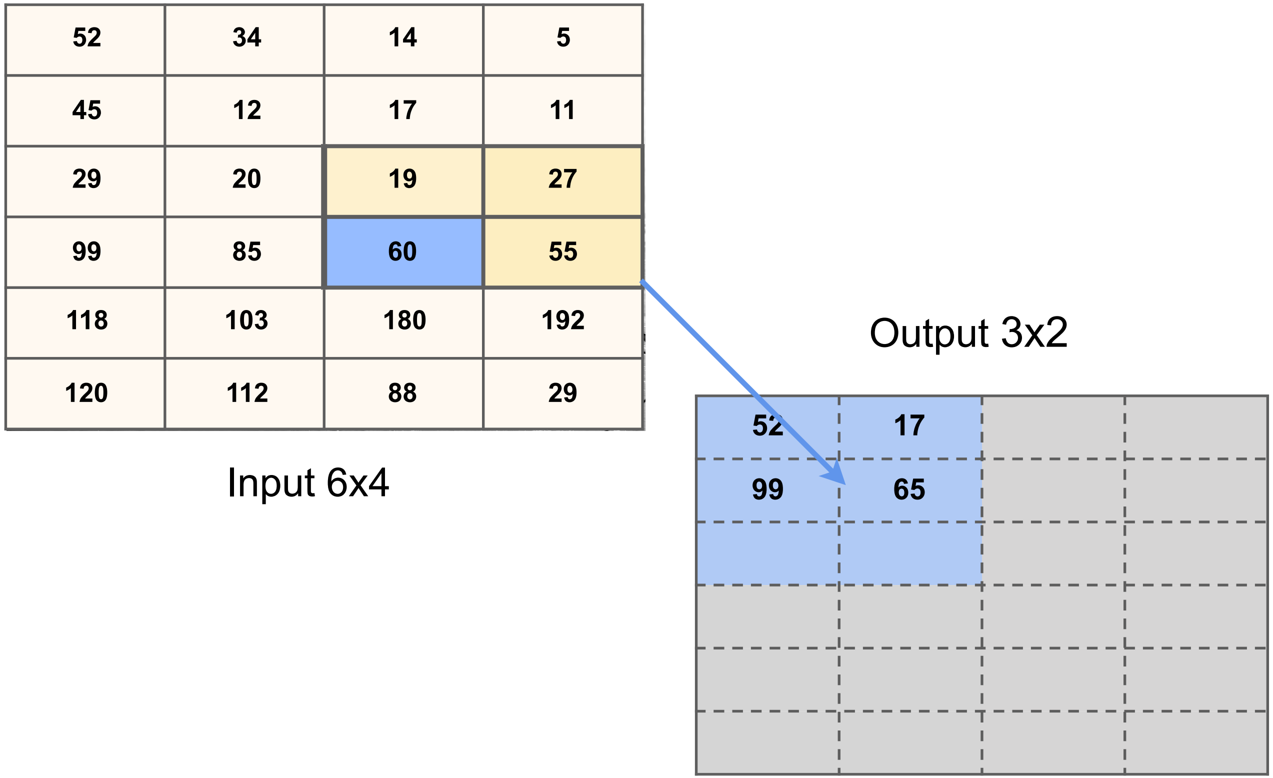
52	34	14	5
45	12	17	11
29	20	19	27
99	85	60	55
118	103	180	192
120	112	88	29

Input 6x4

Output 3x2

52	17		
99			

2D Max Pooling Review



1D Max Pooling

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

Max Pooling 1	Max Pooling 2
9.7	

1D Max Pooling

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

Max Pooling 1
9.7
23.4

Max Pooling 2

1D Max Pooling

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

Max Pooling 1	Max Pooling 2
9.7	
23.4	
-0.6	

1D Max Pooling

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

Max Pooling 1	Max Pooling 2
9.7	
23.4	
-0.6	
-1.2	

1D Max Pooling

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

Max Pooling 1	Max Pooling 2
9.7	0.4
23.4	
-0.6	
-1.2	

1D Max Pooling

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

Max Pooling 1	Max Pooling 2
9.7	0.4
23.4	-3.2
-0.6	
-1.2	

1D Max Pooling

	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

Max Pooling 1	Max Pooling 2
9.7	0.4
23.4	-3.2
-0.6	-2.1
-1.2	

1D Max Pooling

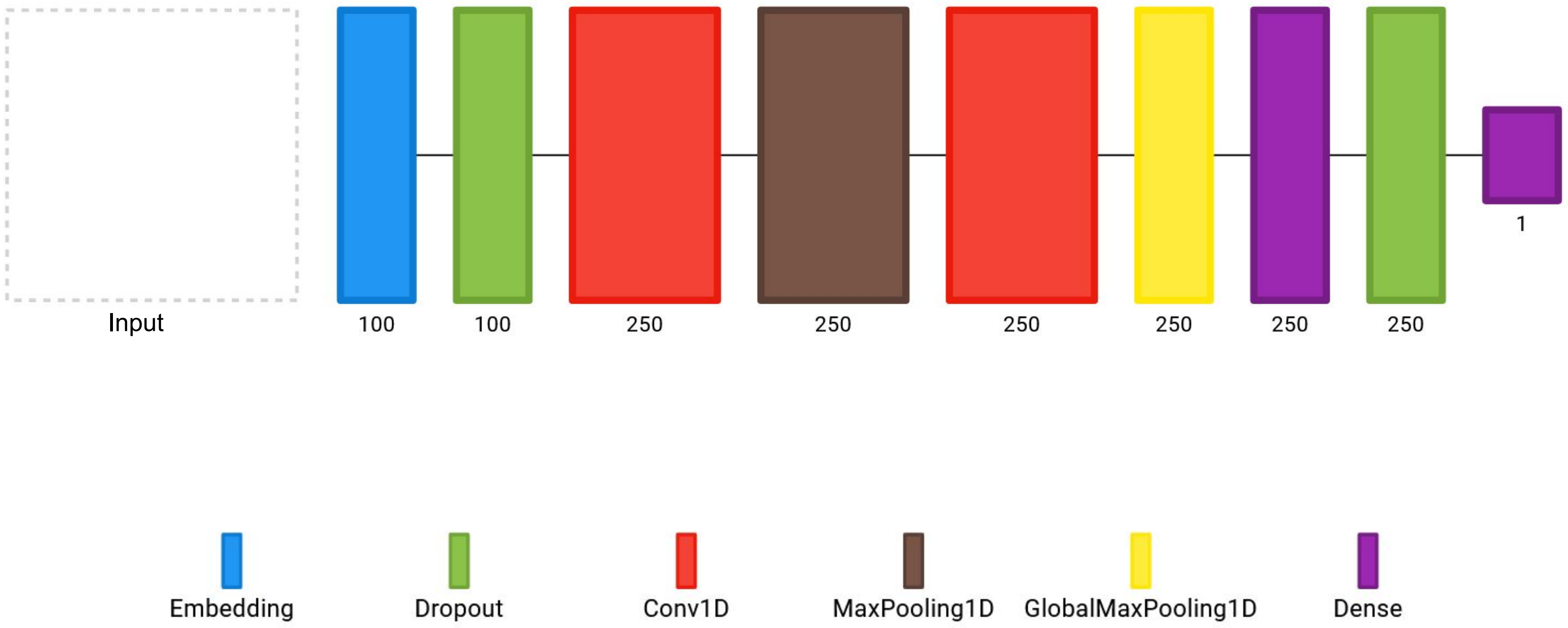
	I	love	this	movie
Channel 1	0.4	9.7	-0.7	0.4
Channel 2	23.4	5.4	-5.2	-3.2
Channel 3	-0.6	-9.4	-2.1	-9.9
Channel 4	-2.4	-1.2	-0.9	-1.9

Max Pooling 1
9.7
23.4
-0.6
-1.2

Max Pooling 2
0.4
-3.2
-2.1
-0.9

Convolutional Neural Network

Network Architecture



Network Layers

Embedding — Map input text to word vectors

Dropout — Randomly drop nodes to reduce overfitting

Conv1D — Extract features from input

Network Layers

MaxPooling1D

Downsample feature vector to the maximum value in each patch of the feature vector

GlobalMaxPooling1D

Downsample feature vector to the maximum value in the entire feature vector

Dense

Fully-connected, output layer