



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
CAMPUS DI FORLÌ

91258 / B0385 Natural Language Processing

Lesson 15. Convolutions in Text

Alberto Barrón-Cedeño
a.barron@unibo.it

17/11/2025

Table of Contents

1. Quick Keras Reminder
2. Prologue to CNN and RNN
3. CNN
4. CNNs for NLP

Chapter 7 of Lane et al. (2019)

Quick Keras Reminder

Keras

Sequential()

- Python class
- Neural network abstraction
- Grants access to the basic Keras API

Sequential.compile()

- Builds the underlying weights
- Builds the interconnected relationships

Sequential.fit()

- Computes the training errors (loss)
- Applies backpropagation (weight adjustment)

Some “cooking” hyperparameters

`epochs` number of iterations over the data

`batch_size` number of instances before adjusting

`optimizer` function

Prologue to CNN and RNN

Prologue

- We have learned to build embedding spaces for words and texts
- We are considering the neighborhood of the words ($\sim$ the bag)
- We are not considering *actual* connections yet
- The downstream application is usually classification or regression

We will start heading towards text generation

Words have relations and influence each other

Word order

s_1 = The dog chased the cat.

s_2 = The cat chased the dog.

$$\text{sim}(\text{tfidf}(s_1), \text{tfidf}(s_2)) = 1$$

$$\text{sim}(\text{wv}(s_1), \text{wv}(s_2)) = 1$$

(1)

But s_1 and s_2 are not the same!

Word proximity

s = His mother, besides her son's willingness to amend the issue,
decided to punish him

mother...decided | son...him

(Lane et al., 2019, p. 220)

Words have relations and influence each other

Spatial relation

Consider the position of words
($\sim$ written)

→ fixed-width window
convolutional neural networks

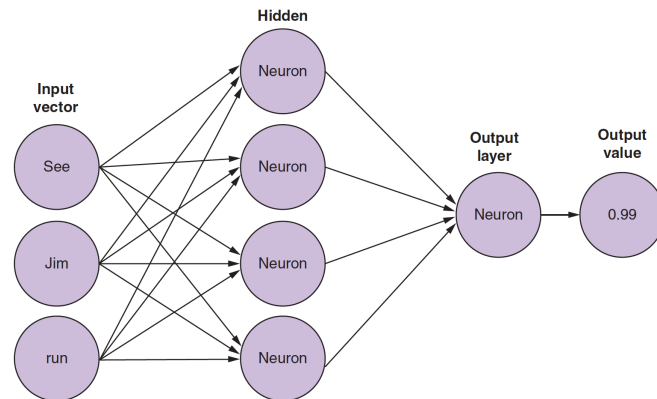
Temporal relation

Consider words as time series
($\sim$ spoken)

→ ongoing (unk) amount of time
recurrent neural networks

(Lane et al., 2019, p. 220)

Multiple Input Words



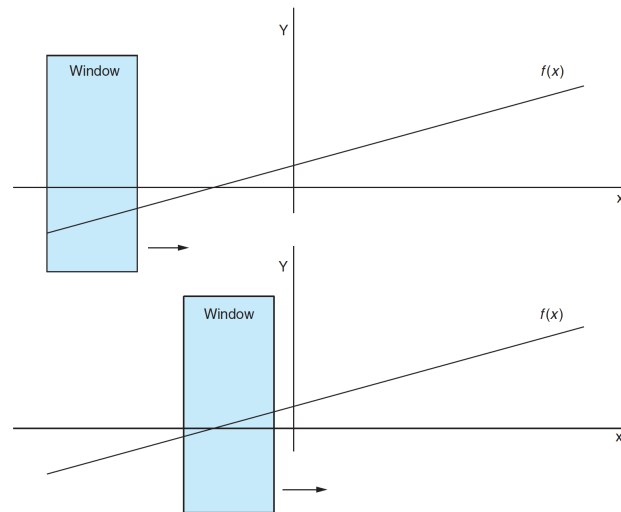
- Three tokens are passed at a time
- Two input alternatives
 - one-hot vector
 - pre-trained word vector

CNN

(Lane et al., 2019, p. 221)

Convolutional Neural Networks

Sliding —or convolving¹— a window over the sample

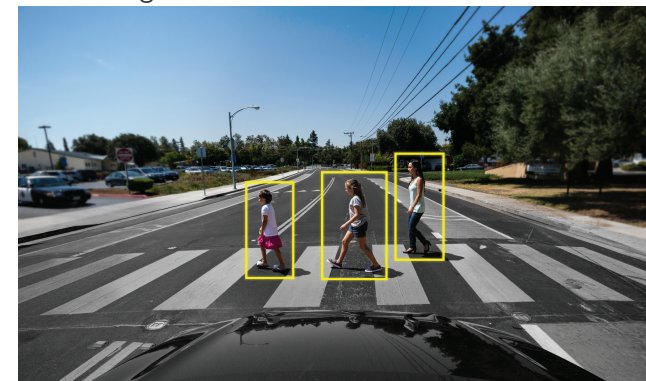


¹To roll or wind together (Webster's)

Convolutional Neural Networks

Back to the roots: image recognition

- Input: pixels of an image
- Output: the image contains x

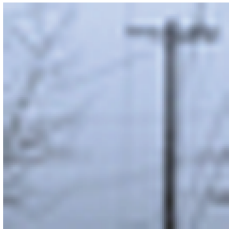


https://blogs.nvidia.com/wp-content/uploads/2019/04/ADAS-IMG_0052.jpg

Convolutional Neural Networks

When the input is an image

- B&W: [0,1] (with a smooth binariser)
- Grayscale: [0, 255]
- Colour: R: [0, 255] G: [0, 255] B: [0, 255]



0	120	119	118	108	103	111	115	115	111	117	120	120	120	120	111	114	114	116	118	120	120	120	120	121	121	21	22	23	24	25	26	27
1	115	114	113	103	93	104	119	115	108	110	116	115	116	117	115	107	108	108	111	118	49	87	109	105	115	110	115	115	115	115	115	115
2	109	114	121	116	108	112	118	104	111	118	112	119	119	119	119	114	114	119	118	117	118	114	119	118	117	117	117	117	117	117	117	117
3	109	114	121	116	108	112	118	104	111	118	112	119	119	119	119	114	114	119	118	117	118	114	119	118	117	117	117	117	117	117	117	117
4	108	110	116	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111	111
5	103	112	109	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110	110
6	111	111	112	104	93	104	119	115	108	110	116	115	116	117	115	107	108	108	111	118	49	87	109	105	115	110	115	115	115	115	115	115
7	111	111	110	105	103	110	103	103	106	111	115	115	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116	116
8	99	112	106	108	86	85	95	100	94	103	106	106	108	109	116	114	113	108	112	113	113	114	115	115	45	83	104	108	111	112	114	114
9	112	106	108	86	85	109	104	103	106	106	108	109	116	114	113	108	112	113	113	114	115	115	115	45	83	104	108	111	112	114	114	114
10	110	106	91	86	108	107	110	109	111	112	108	107	111	112	113	111	107	112	113	112	112	110	38	78	109	108	111	114	103	103	103	103
11	101	91	76	101	103	107	107	108	110	107	103	111	111	110	109	106	112	111	111	111	109	37	82	108	107	111	113	111	113	111	113	113
12	98	92	89	115	108	108	111	106	100	98	106	108	110	110	107	106	108	109	107	37	78	106	103	106	108	103	94	94	94	94	94	94
13	100	72	97	102	92	93	89	89	97	103	103	105	106	106	102	101	106	106	106	23	75	105	103	108	108	98	107	107	107	107	107	107
14	81	80	87	87	85	85	85	85	85	85	85	85	85	85	85	85	85	85	85	21	72	104	106	108	100	102	109	102	109	102	109	102
15	71	82	87	85	78	81	88	84	89	98	94	99	104	104	104	105	103	101	105	103	106	102	76	103	103	106	98	101	101	101	101	101
16	67	87	64	88	84	89	98	94	99	104	104	104	104	105	103	101	105	103	106	102	23	76	103	103	106	98	101	101	101	101	101	101
17	62	82	87	82	81	84	84	84	84	84	84	84	84	84	84	84	84	84	84	16	74	84	84	84	84	84	84	84	84	84	84	84
18	60	71	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	13	64	93	89	81	89	81	79	79	79	79	79	79
19	64	88	87	84	101	100	101	101	101	101	101	101	101	101	101	101	101	101	101	98	89	105	94	97	87	13	100	87	93	105	81	101
20	52	80	80	82	94	97	94	95	93	92	93	92	92	91	84	94	95	98	89	12	79	103	91	100	89	98	104	104	104	104	104	104
21	77	80	80	82	94	97	94	95	93	92	93	95	94	94	94	94	94	94	94	84	7	74	89	84	90	70	81	73	73	73	73	73
22	81	87	83	84	88	89	91	87	90	92	93	95	94	94	94	94	94	94	94	81	0	76	90	92	81	77	65	58	58	58	58	58
23	60	66	82	92	90	87	90	94	94	94	94	94	94	94	94	94	94	94	94	81	7	74	89	84	90	70	81	73	73	73	73	73
24	87	81	83	86	87	84	90	92	92	92	92	92	92	92	92	92	92	92	92	77	4	73	91	92	81	95	96	95	95	95	95	95
25	77	88	83	85	91	91	89	90	91	92	91	91	91	89	90	92	90	89	72	8	66	92	83	84	92	91	91	91	91	91	91	91
26	81	86	88	91	89	89	89	89	89	88	88	88	89	89	89	84	83	79	90	87	74	0	60	89	77	90	91	90	89	89	89	89

(Lane et al., 2019, p. 223)

Convolutional Neural Networks

When the input is an image

An image is just a bunch of numbers

- Appropriate as input for a NN
- But one single pixel has no real meaning

→ Sliding over fragments of the image

The convolution defines a set of filters (aka kernels) to do just that

- Take “snapshots” of different areas of the image
- Process them, one at a time

Convolutional Neural Networks

Strides and filters

Stride

- The distance “traveled” when sliding
- Yet another parameter
- Never bigger than the size of the filter → overlapping areas

Sounds familiar? *n*-grams

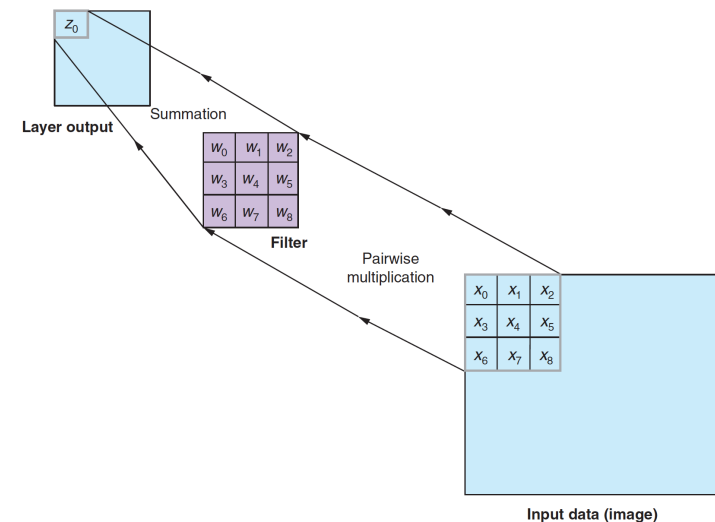
Filter

- $n \times m$ surfaces
- Typically $n = m = 3$ (but $n \neq m$ is possible)
- Includes a set of weights (fix for the whole image)
- Includes an activation function: usually ReLU

$$z = \max(\text{sum}(x * w), 0)$$

Convolutional Neural Networks

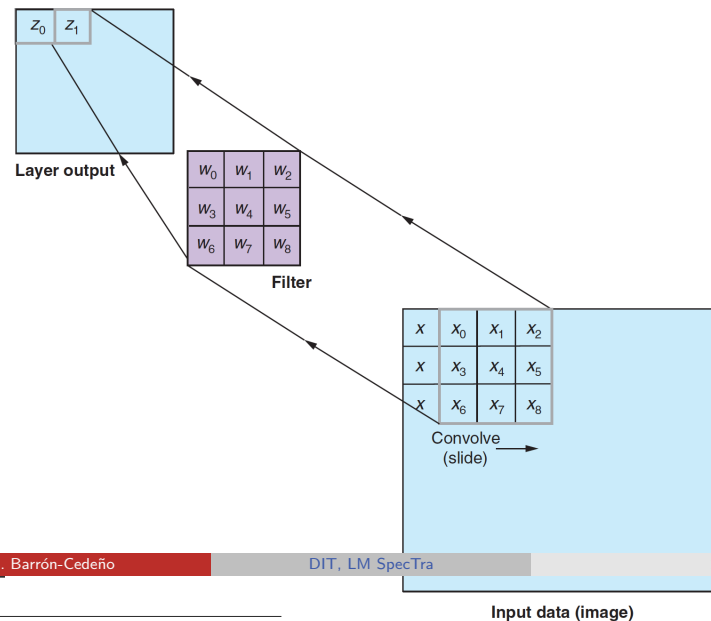
Convolutional step



(Lane et al., 2019, p. 225)

Convolutional Neural Networks

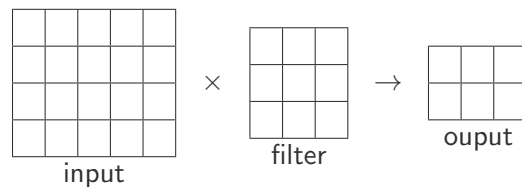
Convolution



(Lane et al., 2019, p. 226)

Convolutional Neural Networks

Padding



We are producing smaller images

"I don't care": Keras' argument `padding='valid'`

The edges of the image are undersampled

"I do care": Keras' padding argument `padding='same'`

In NLP we care

Convolutional Neural Networks

Producing multiple images

- k filters exist which carry out different operations
- Every filter will produce a new image, combination of source and filter

Convolutional Neural Networks

Pipeline

Input: an image, text

Output: a class, a real number

- Produce k new images through k filters
- Wire the filtered images to a feed-forward network
- Proceed as usual

We can add multiple convolution layers

A full path of learning layers and abstractions

- Edges
- Shapes
- Colours
- Concepts

What is learned

- Good filters
- "Standard" weights

Convolutional Neural Networks

Keras premier

```
from keras.models import Sequential
from keras.layers import Conv1D

model = Sequential()

model.add(Conv1D(filters=16,
                  kernel_size=3,
                  padding='same',
                  activation='relu',
                  strides=1,
                  input_shape=(100, 300))
)
```

CNN Wrap up

- Sliding —or convolving— a window over the sample
- Filters (kernels; matrices) slide over fragments of the image
- “Snapshots” of different areas of the image are taken and processed
- Multiple filters produce multiple images
- Multiple convolution layers can be added
- At the end, we can plug a “standard” fully-connected NN

CNNs for NLP

Back to Text

- In images both vertical and horizontal relationships are relevant
- In text only horizontal ones do²
- We need “1D” filters

1 × 3 Filter
The cat and dog went to the bodega together.

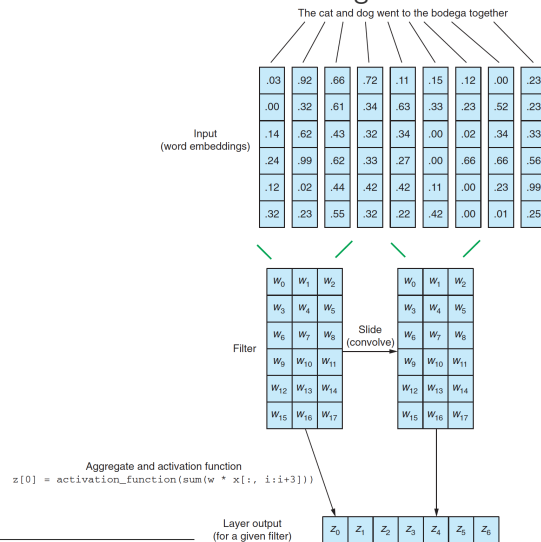
1 × 3 Filter
The cat and dog went to the bodega together.

1 × 3 Filter
The cat and dog went to the bodega together.

²l2r or r2l; for some languages it's the vertical direction that matters (e.g., Japanese) (Lane et al., 2019, p. 229)

But we do have 2D “filters”

Words are represented with word embeddings: vectors



The convolution is (practically) the same as for images

- We now *convolve* in one dimension (not two)
- The computation order is irrelevant, but the outputs have to be fed in the same order
- The filters' weights are fixed for a full sample (parallel computation)
- Their outputs become the features for the classifier

Let us see

Padding

- (In general) in image processing the inputs are of fixed size, regardless of the instance (same source!)
- Texts are not fixed length (regardless of their source)
- Instances longer than `maxlen` will be truncated
- Instances shorter than `maxlen` will be **padded**

$x_0, x_1, x_2, x_3, \dots, x_{398} x_{399} \quad x_{400} x_{401}$

$x_0, x_1, x_2, x_3, \dots, x_{397} \quad \text{PAD PAD}$

Let us see

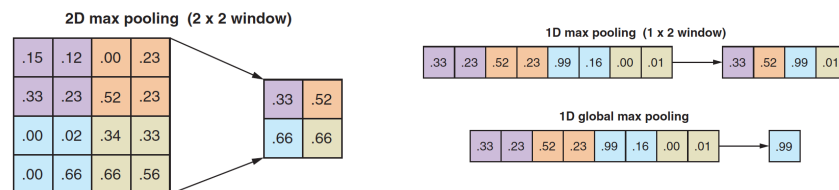
Pooling

- For each filter one new version of the instance is produced (250 in the example)
- Pooling evenly divides the output of each filter into subsections
- It selects (or computes) a representative value for each subsection

Pooling

Pooling is “the CNN path to dimensionality reduction [...] by learning higher-order representations of the source data” (Lane et al., 2019, p. 236)

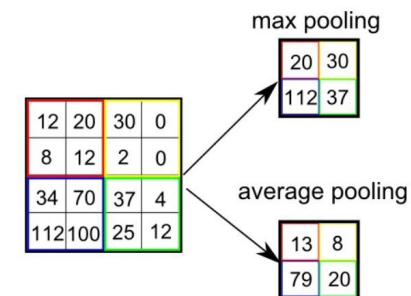
- The filters job is finding patterns → relationships between words and their neighbours
- Pooling in text: a 1D window (e.g., 1×2 or 1×3)



(Lane et al., 2019, p. 237)

Pooling

Max vs Average Pooling



- Average is more intuitive: retaining most of the info
- Max is better: the NN keeps the most prominent feature

Let us see

Image borrowed from
www.quora.com/What-is-max-pooling-in-convolutional-neural-networks

Recap

- Each filter will produce a 1×398 vector
- For each of the 250 filter outputs, we take the single maximum value for each 1D vector
- Output: one 1×250 vector

This is a crude semantic representation of the text

Dropout: Preventing Overfitting

On each training pass **turn off** a percentage of the input of a layer; it will become 0

- Chosen randomly on each pass
- It will not rely heavily on any feature
- It will generalise better
- Dropout is applied during training only



Let us see

Photogram from the film “The Platform” (2019)

Workhorse Loss Functions

Out of the 13+ available loss functions:

`binary_crossentropy`: the output neuron is either on or off

`categorical_crossentropy`: the output is one out of many classes

 Let us see

Closing Remarks

- Your input is a series of max 400 words; 300 elements each
- Nothing prevents you from stacking other embeddings (think of RGB)
- The output of the convolution layer is tied to the task (in this case, sentiment analysis)
- A CNN is more efficient, thanks to the pooling process and the filters
- You can add many convolution layers

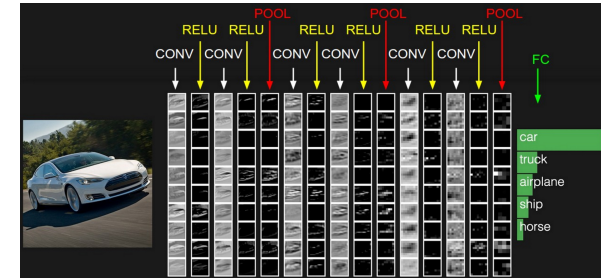


Image borrowed from <https://blog.mapillary.com>

Next time

- Recurrent Neural Networks

References

Lane, H., C. Howard, and H. Hapkem
2019. *Natural Language Processing in Action*. Shelter Island, NY: Manning Publication Co.