

Redes Neuronales Convolucionales (Texto)

Orlando Ramos Flores

Contenido

- Introducción
- Convolución en 1D
 - Ejemplo
- Convolución 1D para texto
 - Intuición Convolución 1D
 - Padding
 - Pooling
 - Stride
 - Pooling local (con stride)
- Arquitectura general CNN para texto
- Ejercicio
- Redes Convolucionales a nivel de caracteres

Introducción: N-gramas

- Un modelo de n-grama, es el modelo más simple que asigna probabilidades a secuencias de palabras sin considerar el orden de las palabras.
- Formulación intuitiva: $P(\mathbf{w}|\mathbf{h})$
- La probabilidad de la palabra w , dada alguna historia h .

Dada la oración: *Un acuerdo para mantener el gobierno abierto*

$w = el$

$h = un\ acuerdo\ para\ mantener$

Introducción: N-gramas

$w = el$

$h = un\ acuerdo\ para\ mantener$

De las veces que viste la historia h , ¿cuántas veces la siguió la palabra w ?

$$P(el|un\ acuerdo\ para\ mantener) = \frac{un\ acuerdo\ para\ mantener\ el}{un\ acuerdo\ para\ mantener}$$

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

Introducción: N-gramas

unigramas = [un], [acuerdo], [para], [mantener], ...

bigramas = [un acuerdo], [acuerdo, para], [para mantener], ...

trigramas = [un acuerdo para], [acuerdo, para, mantener], ...

Convolución 1D

Convolución en 1D

Para la **convolución** entre dos funciones, se tiene $\mathbf{f}, \mathbf{g}: \mathbb{R}^D \rightarrow \mathbb{R}$, es definida como:

$$s(t) = \int x(a)w(t - a)da$$

En la terminología de red convolucional, el primer argumento (en este ejemplo, la función \mathbf{x}) de la convolución se suele denominar **entrada** y el segundo argumento (en este ejemplo, la función \mathbf{w}) como **kernel**. La salida a veces se denomina **mapa de características**. Esta operación es llamada convolución. Típicamente es denotada con un (*): $s(t) = (x * w)(t)$

Referencias:

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Murphy, K. P. (2022). Probabilistic machine learning: an introduction. MIT press.

Ketkar, N., & Santana, E. (2017). Deep learning with Python (Vol. 1). Berkeley: Apress.

Convolución 1D

Por lo general, cuando trabajamos con datos en una computadora, el tiempo se discretiza y nuestro sensor proporcionará datos a intervalos regulares. Entonces:

$$s(t) = (x * w)(t) = \sum_a x(a) \cdot w(t - a)$$

Una forma equivalente de esta operación dada la conmutatividad de la operación de convolución es la siguiente:

$$(x * w)(t) = \sum_a x(t - a) \cdot w(a)$$

Referencias:

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Murphy, K. P. (2022). Probabilistic machine learning: an introduction. MIT press.

Ketkar, N., & Santana, E. (2017). Deep learning with Python (Vol. 1). Berkeley: Apress.

Convolución 1D

En la literatura de aprendizaje profundo, el término "convolución" generalmente se usa para significar correlación cruzada; seguiremos esta convención:

$$(x * w)(t) = \sum_a x(t + a) \cdot w(a)$$

En general la operación de convolución se puede ver de la siguiente forma:

$$(w * x)(i) = \sum_{a=0} w_a \cdot x_{(i+a)}$$

Referencias:

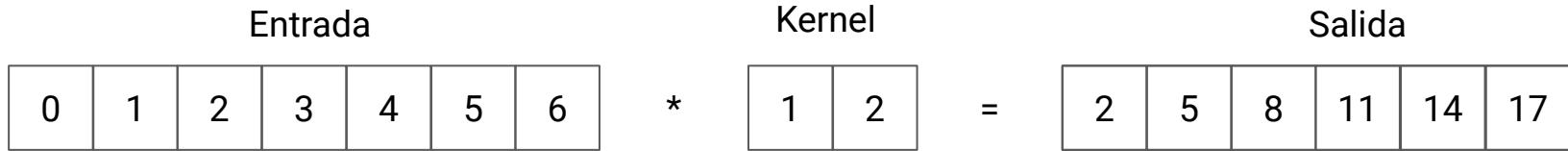
Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Murphy, K. P. (2022). Probabilistic machine learning: an introduction. MIT press.

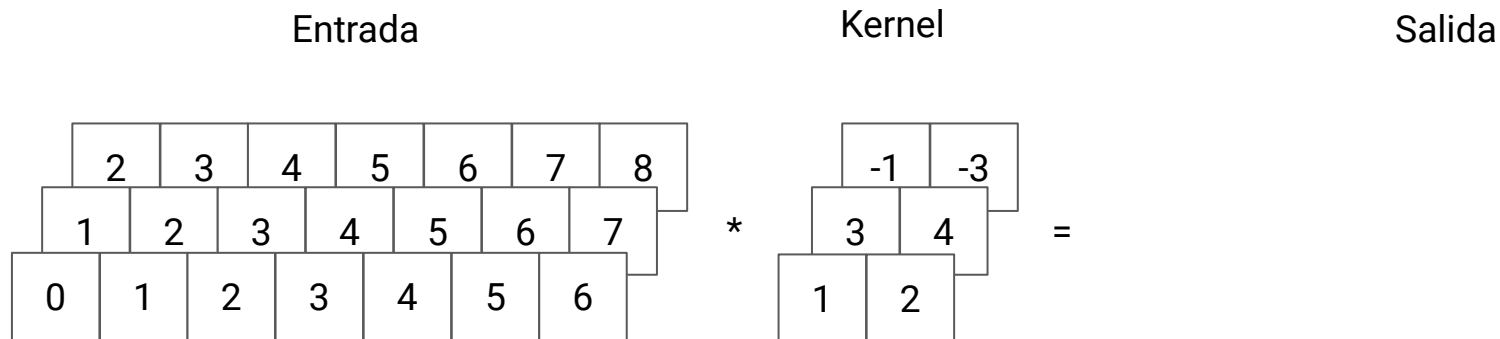
Ketkar, N., & Santana, E. (2017). Deep learning with Python (Vol. 1). Berkeley: Apress.

Convolución 1D: Ejemplo 1

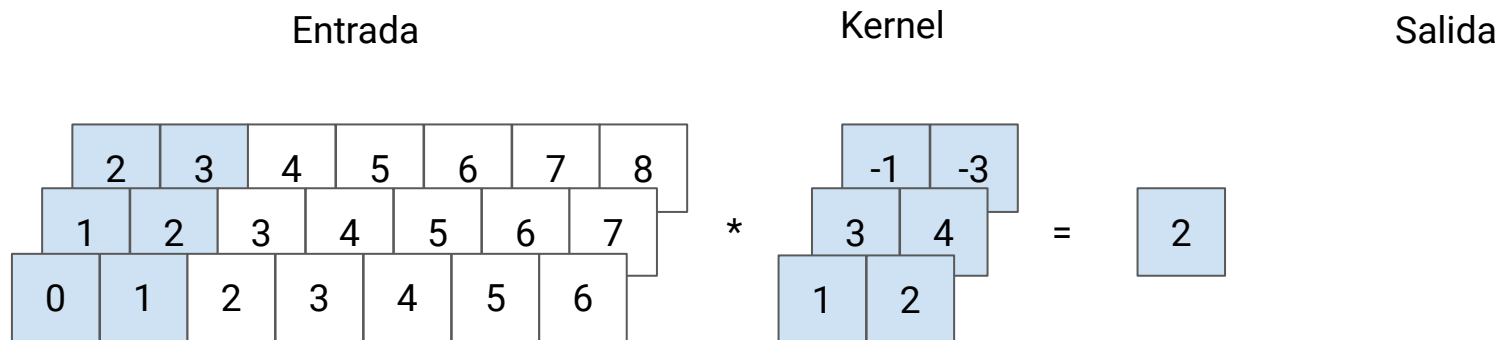
$$\mathbf{x} = [0, 1, 2, 3, 4, 5, 6] \quad \mathbf{w} = [1, 2]$$



Convolución 1D: Ejemplo 2



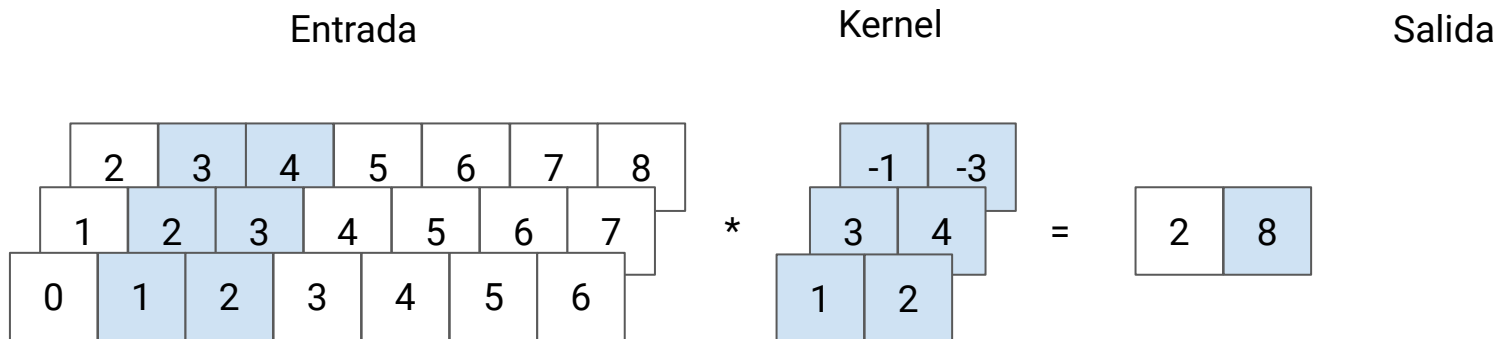
Convolución 1D: Ejemplo 2



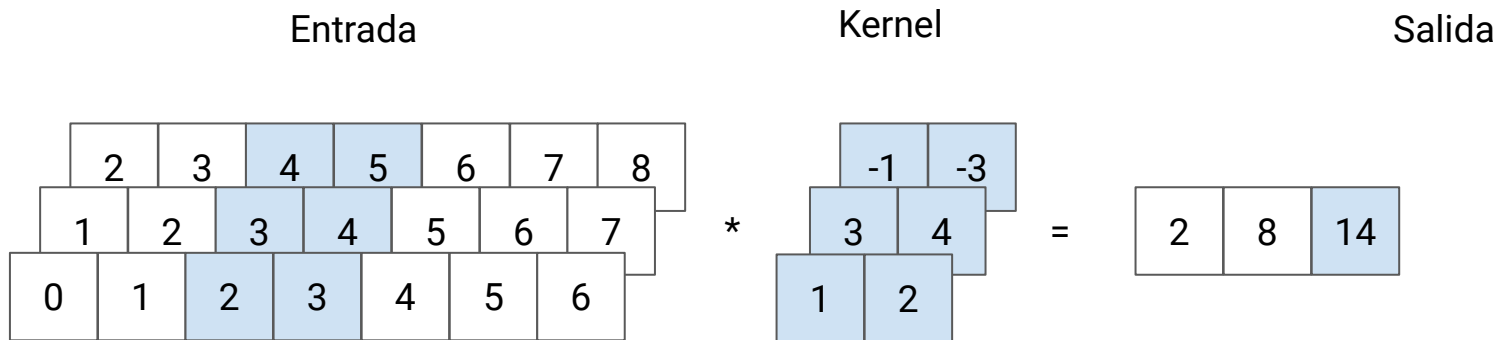
$$(0 * 1) + (1 * 2) + (1 * 3) + (2 * 4) + (2 * -1) + (3 * -3) = 0 + 2 + 3 + 8 - 2 - 9 = 13 - 11 = 2$$

Implementar la operación convolución 1D

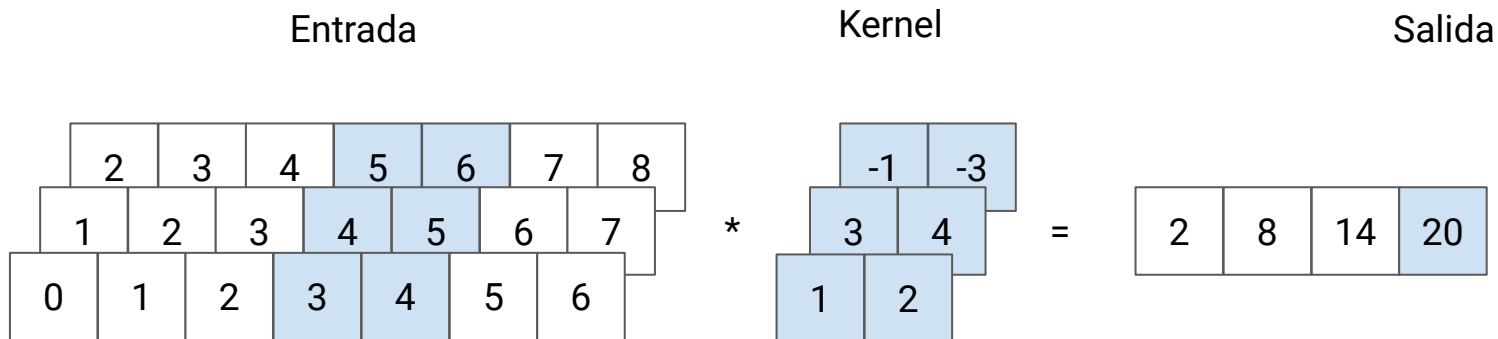
Convolución 1D: Ejemplo 2



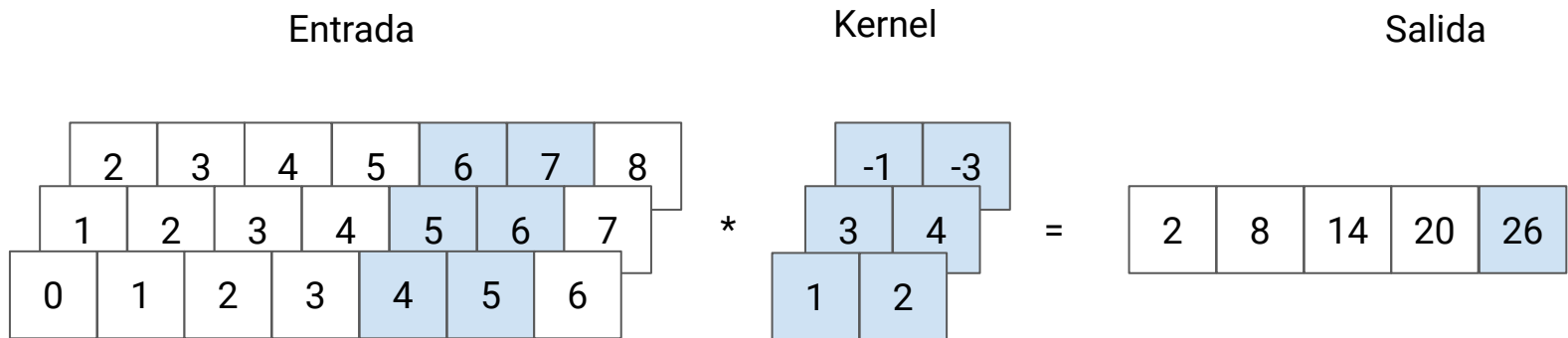
Convolución 1D: Ejemplo 2



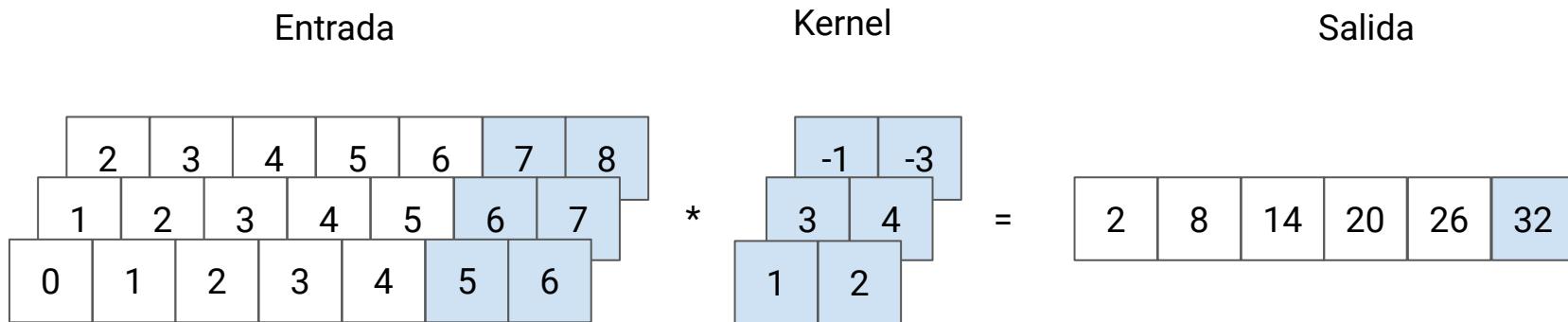
Convolución 1D: Ejemplo 2



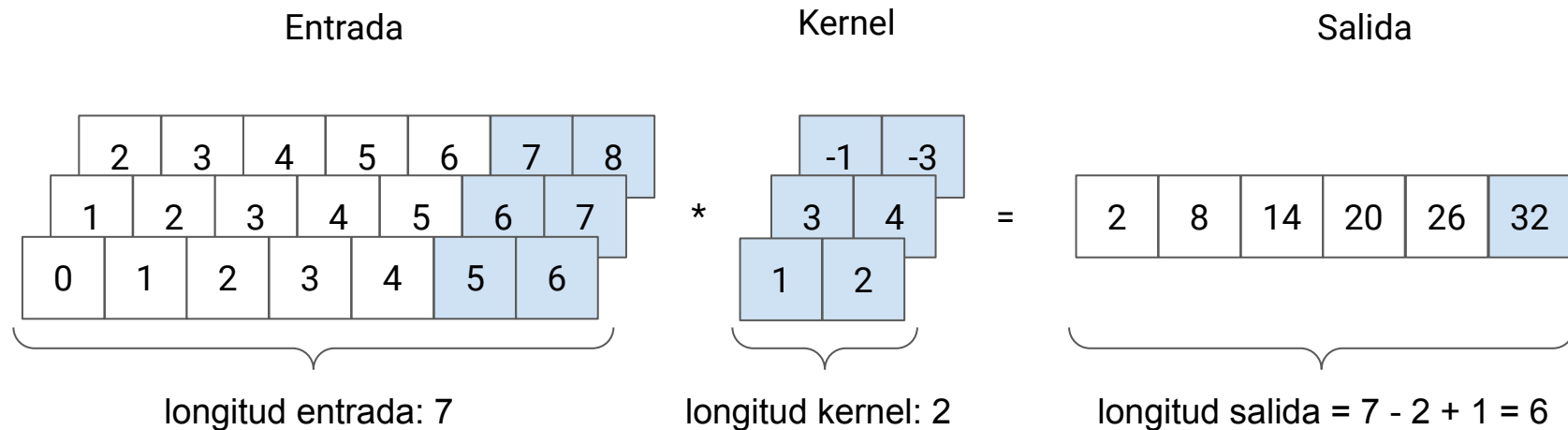
Convolución 1D: Ejemplo 2



Convolución 1D: Ejemplo 2



Convolución 1D: Ejemplo 2



$$\text{longitud salida} = \text{longitud entrada} - \text{longitud kernel} + 1$$

Convolución 1D: Correlación cruzada unidimensional de múltiples canales de entrada

Entrada

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

Kernel

-1	-3
3	4
1	2

*

=

Salida

Convolución 1D: Correlación cruzada unidimensional de múltiples canales de entrada

Entrada

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

Kernel

-1	-3
3	4
1	2

*

=

Salida

2	8	14	20	26	32
---	---	----	----	----	----

$$(2 * -1) + (3 * -3) + (1 * 3) + (2 * 4) + (0 * 1) + (1 * 2) = -2 + -9 + 3 + 8 + 0 + 2 \\ = -11 + 11 + 0 + 2 = 2$$

Modificar Implementación para la operación convolución 1D

Convolución 1D para texto

Convolución en 1D para texto

- Un acuerdo para mantener el gobierno abierto

Un	acuerdo	para	mantener	el	gobierno	abierto
0	1	2	3	4	5	6

Convolución en 1D para texto

(canales) vector de palabras denso (4D)

Oración de entrada	Un	0.2	0.1	-0.3	0.4
	acuerdo	0.5	0.2	-0.3	-0.1
	para	-0.1	-0.3	-0.2	0.4
	mantener	0.3	-0.3	0.1	0.1
	el	0.2	-0.3	0.4	0.2
	gobierno	0.1	0.2	-0.1	-0.1
	abierto	-0.4	-0.4	0.2	0.3

Convolución en 1D para texto

La dimensión de salida que obtendremos después de la convolución, será:

$$out = \left\lfloor \frac{sentence\ length + 2 \cdot padding - kernel}{stride} \right\rfloor + 1$$

Convolución en 1D para texto

Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3

Calcular el producto punto

Kernel de tamaño 3

3	1	2	-3
-1	2	1	-3
1	1	-1	1

Convolución en 1D para texto

Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3

$$0.2*3 + 0.1*1 + -0.3*2 + 0.4*-3$$

$$0.6 + 0.1 + (-0.6) + (-1.2) = -1.1$$

$$0.5*-1 + 0.2*2 + -0.3*1 + -0.1*-3$$

$$-1.1 + -0.5 + 0.4 + (-0.3) + 0.3 = -1.2$$

$$0.1*1 + -0.3*1 + -0.2*-1 + 0.4*1$$

$$-1.2 + -0.1 + -0.3 + 0.2 + 0.4 = -1.0$$

u,a,p	-1.0
-------	------

3	1	2	-3
-1	2	1	-3
1	1	-1	1

Convolución en 1D para texto

Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3

Calcular el resto

3	1	2	-3
-1	2	1	-3
1	1	-1	1

u,a,p	-1.0
a,p,m	
p,m,e	
m,e,g	
e,g,a	

Convolución en 1D para texto

Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3

¿Qué se puede observar?

- La oración se redujo a un solo vector
- La oración se hizo más pequeña
 - cinco palabras

3	1	2	-3
-1	2	1	-3
1	1	-1	1

u,a,p	-1.0
a,p,m	-0.5
p,m,e	-3.6
m,e,g	-0.2
e,g,a	0.3

¿Cómo resolver esto?

Convolución en 1D para texto con padding=1

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

Calcular:

- ∅,u,a
- g,a,∅

3	1	2	-3
-1	2	1	-3
1	1	-1	1

∅,u,a	-0.6
u,a,p	-1.0
a,p,m	-0.5
p,m,e	-3.6
m,e,g	-0.2
e,g,a	0.3
g,a,∅	-0.5

Convolución en 1D para texto con 3 kernels

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

Calcular convolución para los filtros 2 y 3

∅,u,a	-0.6		
u,a,p	-1.0		
a,p,m	-0.5		
p,m,e	-3.6		
m,e,g	-0.2		
e,g,a	0.3		
g,a,∅	-0.5		

3	1	2	-3
-1	2	1	-3
1	1	-1	1

1	0	0	1
1	0	-1	-1
0	1	0	1

1	-1	2	-1
1	0	-1	3
0	2	2	1

Convolución en 1D para texto

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

¿Como resumimos el texto respecto a esas características?

Obtener diferentes características latentes en el texto

∅,u,a	-0.6	0.2	1.4
u,a,p	-1.0	1.6	-1.0
a,p,m	-0.5	-0.1	0.8
p,m,e	-3.6	0.3	0.3
m,e,g	-0.2	0.1	1.2
e,g,a	0.3	0.6	0.9
g,a,∅	-0.5	-0.9	0.1

palabras sobre comida

palabras amables

3	1	2	-3
-1	2	1	-3
1	1	-1	1

1	0	0	1
1	0	-1	-1
0	1	0	1

1	-1	2	-1
1	0	-1	3
0	2	2	1

palabras negativas

Convolución en 1D para texto: max pooling

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

Al parecer existe un
marcador en el texto
que captura en cierto
sentido “comida”

resumir la salida de una red
convolucional

∅,u,a	-0.6	0.2	1.4
u,a,p	-1.0	1.6	-1.0
a,p,m	-0.5	-0.1	0.8
p,m,e	-3.6	0.3	0.3
m,e,g	-0.2	0.1	1.2
e,g,a	0.3	0.6	0.9
g,a,∅	-0.5	-0.9	0.1

3	1	2	-3
-1	2	1	-3
1	1	-1	1

1	0	0	1
1	0	-1	-1
0	1	0	1

1	-1	2	-1
1	0	-1	3
0	2	2	1

max pool	0.3	1.6	1.4
----------	-----	-----	-----

Convolución en 1D para texto: average pooling

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

El resultado es una semántica diferente.

¿Que porcentaje es sobre amabilidad, comida o negatividad?

∅,u,a	-0.6	0.2	1.4
u,a,p	-1.0	1.6	-1.0
a,p,m	-0.5	-0.1	0.8
p,m,e	-3.6	0.3	0.3
m,e,g	-0.2	0.1	1.2
e,g,a	0.3	0.6	0.9
g,a,∅	-0.5	-0.9	0.1

3	1	2	-3
-1	2	1	-3
1	1	-1	1

1	0	0	1
1	0	-1	-1
0	1	0	1

1	-1	2	-1
1	0	-1	3
0	2	2	1

avg pool	-0.87	0.26	0.53
----------	-------	------	------

Convolución en 1D para texto: stride=2

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

Mover la convolución
con stride = 2

Calcular la convolución

∅,u,a	-0.6	0.2	1.4
a,p,m	-0.5	-0.1	0.8
m,e,g	-0.2	0.1	1.2
g,a,∅	-0.5	-0.9	0.1

3	1	2	-3
-1	2	1	-3
1	1	-1	1

1	0	0	1
1	0	-1	-1
0	1	0	1

1	-1	2	-1
1	0	-1	3
0	2	2	1

Convolución en 1D para texto: stride=2

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

Mover la convolución
con stride = 2

Calcular la convolución

∅,u,a	-0.6	0.2	1.4
a,p,m	-0.5	-0.1	0.8
m,e,g	-0.2	0.1	1.2
g,a,∅	-0.5	-0.9	0.1

3	1	2	-3
-1	2	1	-3
1	1	-1	1

1	0	0	1
1	0	-1	-1
0	1	0	1

1	-1	2	-1
1	0	-1	3
0	2	2	1

Convolución en 1D para texto: stride=2

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

Mover la convolución
con stride = 2

Calcular la convolución

∅,u,a	-0.6	0.2	1.4
a,p,m	-0.5	-0.1	0.8
m,e,g	-0.2	0.1	1.2
g,a,∅	-0.5	-0.9	0.1

3	1	2	-3
-1	2	1	-3
1	1	-1	1

1	0	0	1
1	0	-1	-1
0	1	0	1

1	-1	2	-1
1	0	-1	3
0	2	2	1

Convolución en 1D para texto: stride=2

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

Mover la convolución
con stride = 2

Calcular la convolución

∅,u,a	-0.6	0.2	1.4
a,p,m	-0.5	-0.1	0.8
m,e,g	-0.2	0.1	1.2
g,a,∅	-0.5	-0.9	0.1

3	1	2	-3
-1	2	1	-3
1	1	-1	1

1	0	0	1
1	0	-1	-1
0	1	0	1

1	-1	2	-1
1	0	-1	3
0	2	2	1

Convolución en 1D para texto: local max pool, stride=2

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

Se toman las dos
(stride) primeras filas
y se aplica el max
pooling

∅,u,a	-0.6	0.2	1.4
u,a,p	-1.0	1.6	-1.0
a,p,m	-0.5	-0.1	0.8
p,m,e	-3.6	0.3	0.3
m,e,g	-0.2	0.1	1.2
e,g,a	0.3	0.6	0.9
g,a,∅	-0.5	-0.9	0.1

3	1	2	-3
-1	2	1	-3
1	1	-1	1

1	0	0	1
1	0	-1	-1
0	1	0	1

1	-1	2	-1
1	0	-1	3
0	2	2	1

∅,u,a,p	-1.0	1.6	1.4
---------	------	-----	-----

Convolución en 1D para texto: local max pool, stride=2

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

Se aplica nuevamente el stride=2 y se aplica el max pooling

∅,u,a	-0.6	0.2	1.4
u,a,p	-1.0	1.6	-1.0
a,p,m	-0.5	-0.1	0.8
p,m,e	-3.6	0.3	0.3
m,e,g	-0.2	0.1	1.2
e,g,a	0.3	0.6	0.9
g,a,∅	-0.5	-0.9	0.1

3	1	2	-3
-1	2	1	-3
1	1	-1	1

1	0	0	1
1	0	-1	-1
0	1	0	1

1	-1	2	-1
1	0	-1	3
0	2	2	1

∅,u,a,p	-1.0	1.6	1.4
a,p,m,e	-0.5	0.3	0.8

Convolución en 1D para texto: local max pool, stride=2

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

∅,u,a	-0.6	0.2	1.4
u,a,p	-1.0	1.6	-1.0
a,p,m	-0.5	-0.1	0.8
p,m,e	-3.6	0.3	0.3
m,e,g	-0.2	0.1	1.2
e,g,a	0.3	0.6	0.9
g,a,∅	-0.5	-0.9	0.1

3	1	2	-3
-1	2	1	-3
1	1	-1	1

1	0	0	1
1	0	-1	-1
0	1	0	1

1	-1	2	-1
1	0	-1	3
0	2	2	1

∅,u,a,p	-1.0	1.6	1.4
a,p,m,e	-0.5	0.3	0.8
m,e,g,a	0.3	0.6	1.2

Convolución en 1D para texto: local max pool, stride=2

∅	0.0	0.0	0.0	0.0
Un	0.2	0.1	-0.3	0.4
acuerdo	0.5	0.2	-0.3	-0.1
para	-0.1	-0.3	-0.2	0.4
mantener	0.3	-0.3	0.1	0.1
el	0.2	-0.3	0.4	0.2
gobierno	0.1	0.2	-0.1	-0.1
abierto	-0.4	-0.4	0.2	0.3
∅	0.0	0.0	0.0	0.0

Se toman las últimas dos primeras filas y se aplica el max pooling

∅,u,a	-0.6	0.2	1.4
u,a,p	-1.0	1.6	-1.0
a,p,m	-0.5	-0.1	0.8
p,m,e	-3.6	0.3	0.3
m,e,g	-0.2	0.1	1.2
e,g,a	0.3	0.6	0.9
g,a,∅	-0.5	-0.9	0.1
∅	-Inf	-Inf	-Inf

3	1	2	-3
-1	2	1	-3
1	1	-1	1

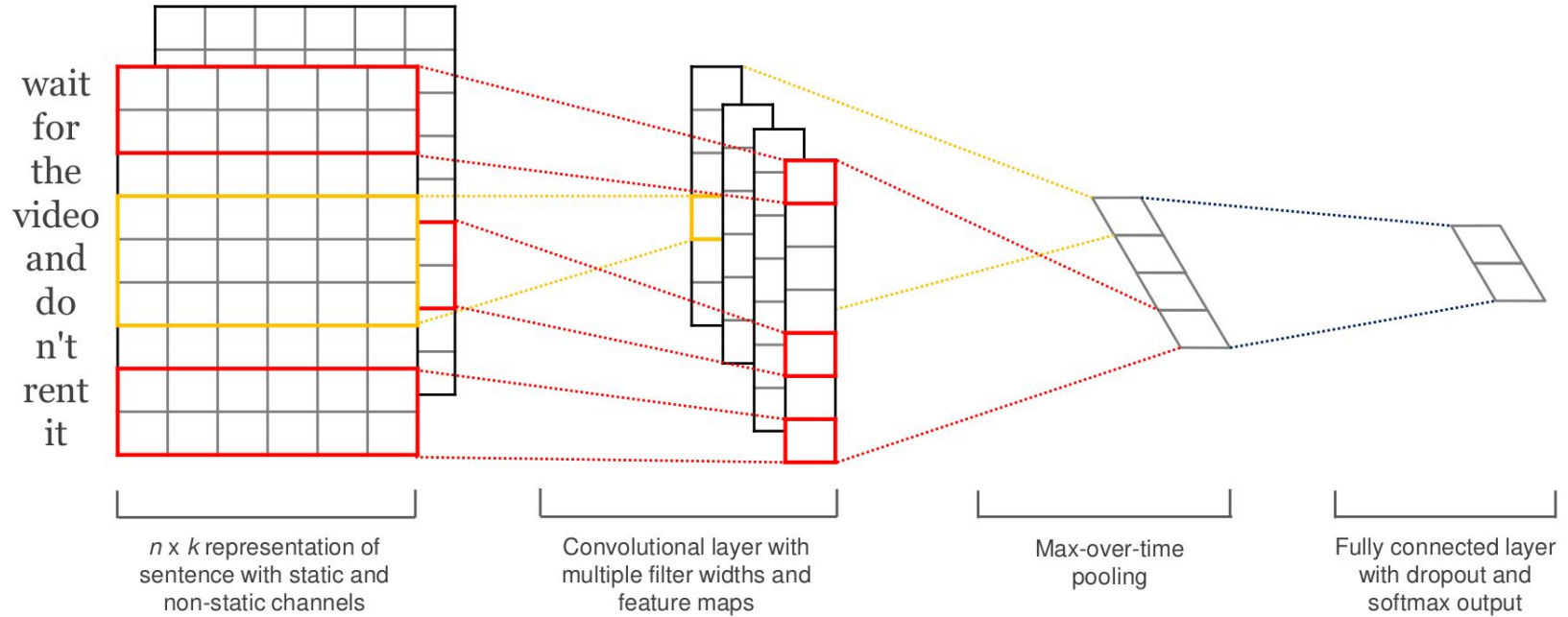
1	0	0	1
1	0	-1	-1
0	1	0	1

1	-1	2	-1
1	0	-1	3
0	2	2	1

∅,u,a,p	-1.0	1.6	1.4
a,p,m,e	-0.5	0.3	0.8
m,e,g,a	0.3	0.6	1.2
g,a,∅,∅	-0.5	-0.9	0.1

Arquitectura general CNN para texto

Arquitectura General

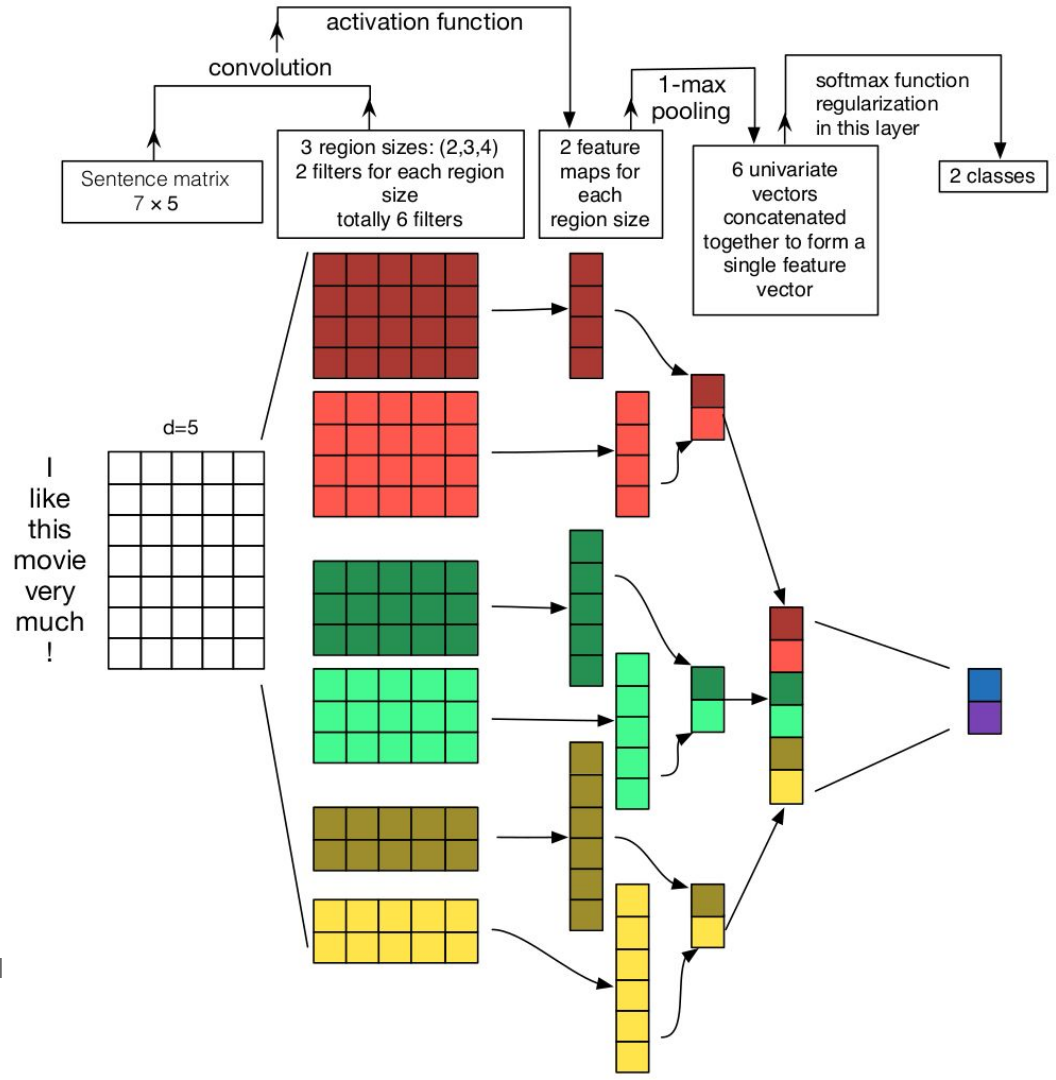


Chen, Y. (2015). *Convolutional neural network for sentence classification* (Master's thesis, University of Waterloo).

Arquitectura General

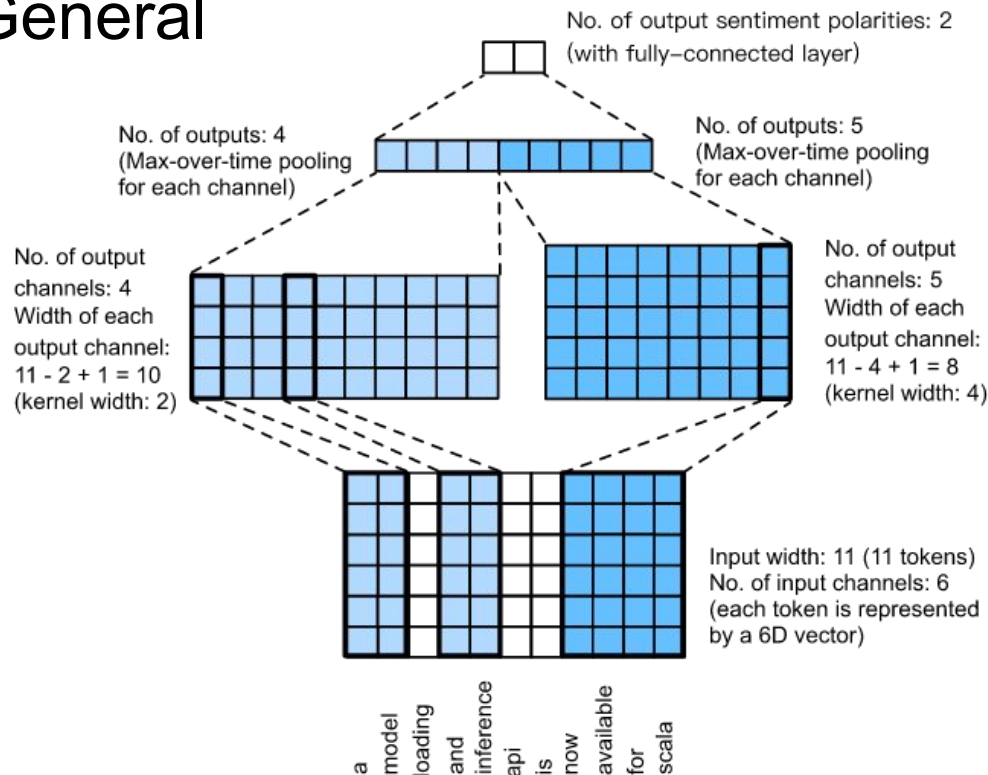
- Función de activación: ReLU
- kernels: 3, 4 y 5
- Filtros: 100 (mapas de características)
- Dropout: 0.5
- Tamaño de mini-batch: 50
- Embeddings: word2vec de tamaño 300

Arquitectura General



Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820.

Arquitectura General



Ejercicio

Ejercicio

misma longitud que la entrada

```
0.0, 0.0, 0.0, 0.0, 0.0
0.4, 0.1, 0.5, 0.8, -0.1
0.0, 0.2, -0.8, 0.3, -0.2
0.1, 0.7, 0.6, 0.5, -0.4
-0.2, -0.3, 0.1, -0.6, 0.7
-0.8, -0.9, 0.7, 0.9, -0.2
0.3, 0.9, 0.2, 0.9, 0.5
-0.1, 0.6, -0.1, -0.7, 0.3
0.5, 0.1, 0.2, 0.7, 0.4
0.0, 0.0, 0.0, 0.0, 0.0
```

```
1.6, 2.9, 2.4
0.2, 3.6, -1.8
3.1, 3.3, -4.4
-5.6, 4.5, 4.8
1.3, 2.1, 3.5
0.1, 7.9, -7.2
6.2, 0.8, 4.8
2.6, 2.22045E-16, 9.2
```

```
2, 1, -3, 1, 3
0, 2, 0, 1, 2
3, 1, -1, 0, 1
```

```
1, 0, 1, 3, -2
-2, 1, 1, 4, 2
-2, 0, 0, 1, 1
```

```
0, 3, -1, -4, 3
4, 1, 3, -1, 4
1, 1, 2, 4, -3
```


Ejercicio

1.6, 2.9, 2.4
0.2, 3.6, -1.8
3.1, 3.3, -4.4
-5.6, 4.5, 4.8
1.3, 2.1, 3.5
0.1, 7.9, -7.2
6.2, 0.8, 4.8
2.6, 2.22045E-16, 9.2

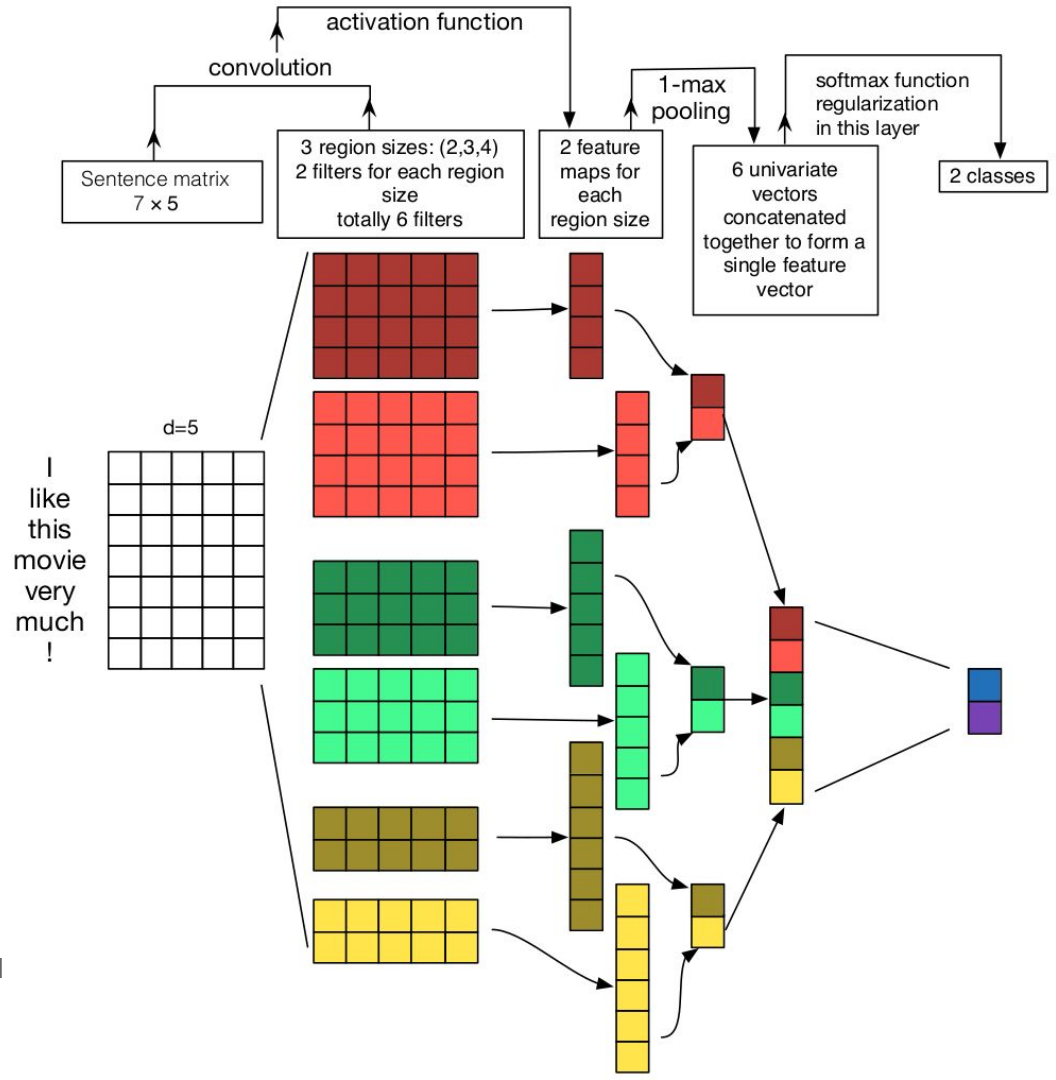
-13.3, -1.5
-2.6, -33.2
-9.0, -10.6
-22.6, 8.6
-7.4, -17.0
-9.2, -37.1

1, -1, 1
0, -1, 0
-1, -1, 1

-2, -1, 0
-2, -1, 0
1, -1, -2

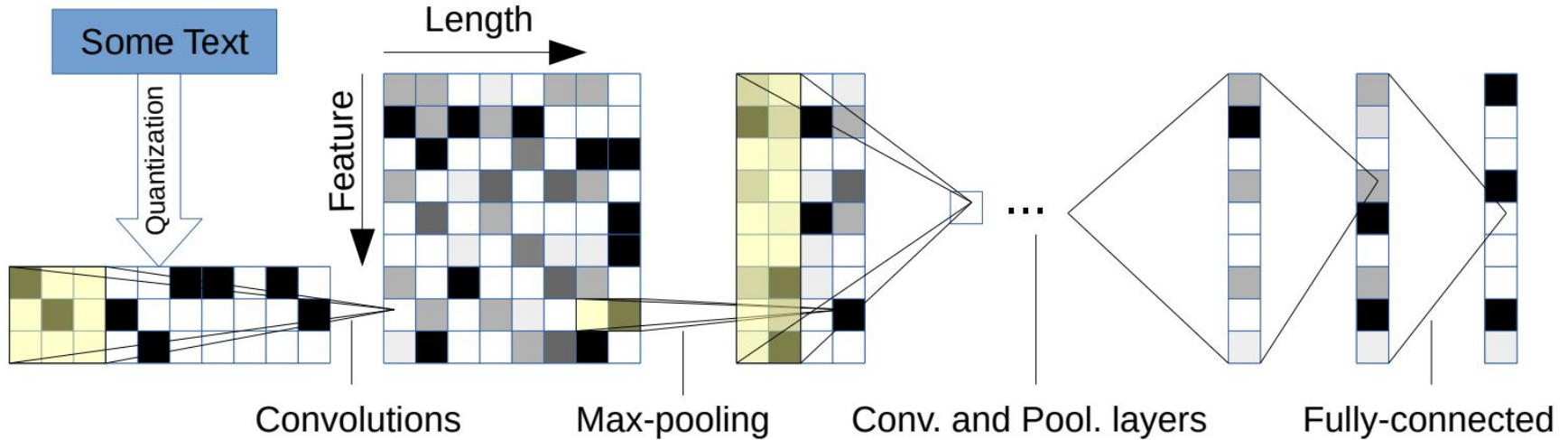
Redes Convolucionales a nivel de caracteres

Arquitectura General con palabras



Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820.

Arquitetura General a nível de caracteres



CNN - a nivel de caracteres

1. Definieron un alfabeto utilizado en todos sus modelos, consta de 70 caracteres, incluidas 26 letras en inglés, 10 dígitos, otros 33 caracteres y el carácter de nueva línea: `abcdefghijklmnopqrstuvwxyz0123456789 -,:;!?'\"^&*~'+-=<>()[]{}`
2. Realizaron la cuantización de los caracteres.
 - a. La codificación se realiza prescribiendo un alfabeto de tamaño m para el idioma de entrada y luego cuantificando cada carácter usando la codificación I de m (o codificación "one-hot")
 - b. La secuencia de caracteres se transforma en una secuencia de tales vectores de tamaño m con longitud fija l_θ .
 - c. Cualquier carácter que exceda la longitud l_θ se ignora, y cualquier carácter que no esté en el alfabeto, incluidos los caracteres en blanco, se cuantifica como vectores todos ceros.
 - d. El orden de cuantificación de caracteres es inverso, de modo que la última lectura de los caracteres siempre se coloca cerca del comienzo de la salida, lo que facilita que las capas totalmente conectadas asocien pesos con la última lectura.

CNN - a nivel de caracteres: cuantificación de caracteres

Example sequence: “**abc bc bbd ca**”

Ejemplo

		h	o	l	a	" "	m	u	n	d	o
1	a	0	0	0	1	0	0	0	0	0	0
2	b	0	0	0	0	0	0	0	0	0	0
3	c	0	0	0	0	0	0	0	0	0	0
4	d	0	0	0	0	0	0	0	0	1	0
5	e	0	0	0	0	0	0	0	0	0	0
6	f	0	0	0	0	0	0	0	0	0	0
7	g	0	0	0	0	0	0	0	0	0	0
8	h	1	0	0	0	0	0	0	0	0	0
9	i	0	0	0	0	0	0	0	0	0	0
10	j	0	0	0	0	0	0	0	0	0	0
11	k	0	0	0	0	0	0	0	0	0	0
12	l	0	0	1	0	0	0	0	0	0	0
13	m	0	0	0	0	0	1	0	0	0	0
14	n	0	0	0	0	0	0	0	n	0	0
15	o	0	1	0	0	0	0	0	0	0	1
16	p	0	0	0	0	0	0	0	0	0	0
17	q	0	0	0	0	0	0	0	0	0	0
18	r	0	0	0	0	0	0	0	0	0	0
19	s	0	0	0	0	0	0	0	0	0	0
20	t	0	0	0	0	0	0	0	0	0	0
21	u	0	0	0	0	0	0	1	0	0	0
22	v	0	0	0	0	0	0	0	0	0	0
23	w	0	0	0	0	0	0	0	0	0	0
24	x	0	0	0	0	0	0	0	0	0	0
25	y	0	0	0	0	0	0	0	0	0	0
26	z	0	0	0	0	0	0	0	0	0	0
27	" "	0	0	0	0	1	0	0	0	0	0
28	<pad>	0	0	0	0	0	0	0	0	0	0



alfabeto

hola mundo



matriz codificada
"one hot"

longitud del tweet

CNN - a nivel de caracteres: cuantificación de caracteres

- Implementar una función en python que realice la cuantificación de caracteres.
- Recomendaciones
 - Definir la longitud máxima del tweet más largo
 - Rellenar tweets con longitud menor
 - El token de relleno y tokens desconocidos se identifican con cero (0)

CNN - a nivel de caracteres

Layer	Large Feature	Small Feature	Kernel	Pool
1	1024	256	7	3
2	1024	256	7	3
3	1024	256	3	N/A
4	1024	256	3	N/A
5	1024	256	3	N/A
6	1024	256	3	3

Model	Error rate	Dictionary
ConvNets	4.88%	abcdefghijklmnopqrstuvwxyz 0123456789,-,.,!?:'"/\ _@#\$\$%`~&* ~ '+=<>{}[]
ConvRec	4.83%	abcdefghijklmnopqrstuvwxyz 0123456789,-,.,!?:'"/\ _@#\$\$%`~&* ~ '+=<>{}[]
Proposed model	4.66%	abcdefghijklmnopqrstuvwxyz 0123456789,.,!?

CNN - a nivel de caracteres: es útil cuando

- Los datos contienen ruido (noisy data), es decir datos sin sentido: tweets, palabras mal escritas, argot, etc.
- El conjunto de datos es grande, mayor a 1 millón de documentos
- Además, eliminan la necesidad de realizar preprocesamiento
- Son adecuadas cuando existen palabras mal escritas o están fuera del vocabulario (alfabeto)

CNN - a nivel de caracteres

- Implementar una CNN a nivel caracteres con el conjunto de datos de tweets con 13 clases