Machine Learning

Embeddings

Data Science Bootcamp

The Bridge

Definición

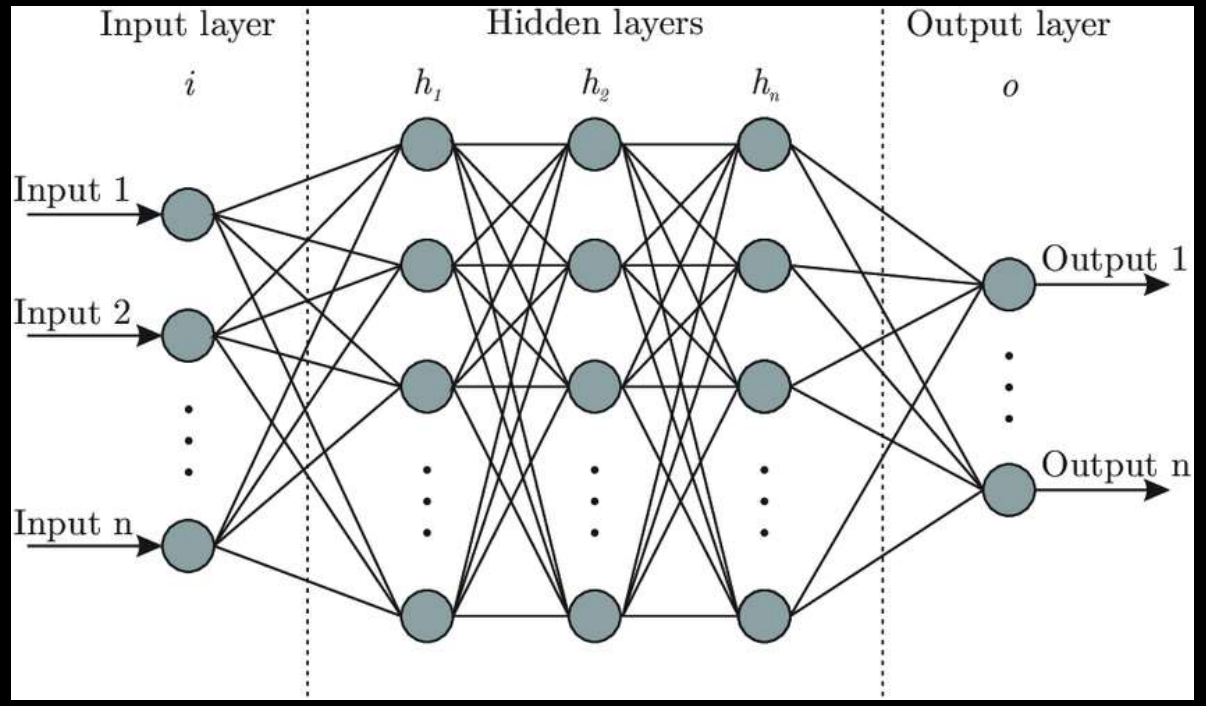
Deep Learning

Entrenamiento

"La peor película que he visto nunca"

"Muy buen argumento"

"No la recomendaría"



Output Reviews
- Buena
- Mala

El modelo no puede procesar texto, por lo que tenemos que traducir a números: CODIFICAR

Codificar

Caracteres

0	*NUL*	16	*DLE*	32	SPC	48	0	64	@	80	P	96	`	112	p
1	*SOH*	17	*DC1*	33	!	49	1	65	A	81	Q	97	a	113	q
2	*STX*	18	*DC2*	34	"	50	2	66	B	82	R	98	b	114	r
3	*ETX*	19	*DC3*	35	#	51	3	67	C	83	S	99	c	115	s
4	*EOT*	20	*DC4*	36	$	52	4	68	D	84	T	100	d	116	t
5	*ENQ*	21	*NAK*	37	%	53	5	69	E	85	U	101	e	117	u
6	*ACK*	22	*SYN*	38	&	54	6	70	F	86	V	102	f	118	v
7	*BEL*	23	*ETB*	39	'	55	7	71	G	87	W	103	g	119	w
8	*BS*	24	*CAN*	40	(	56	8	72	H	88	X	104	h	120	x
9	*HT*	25	*EM*	41	)	57	9	73	I	89	Y	105	i	121	y
10	*LF*	26	*SUB*	42	*	58	:	74	J	90	Z	106	j	122	z
11	*VT*	27	*ESC*	43	+	59	;	75	K	91	[	107	k	123	{
12	*FF*	28	*FS*	44	,	60	<	76	L	92	\	108	l	124	|
13	*CR*	29	*GS*	45	-	61	=	77	M	93	]	109	m	125	}
14	*SO*	30	*RS*	46	.	62	>	78	N	94	^	110	n	126	~
15	*SI*	31	*US*	47	/	63	?	79	O	95	_	111	o	127	DEL

Lo que tenemos que codificar son palabras, no caracteres

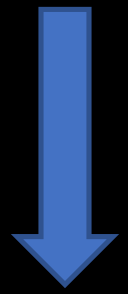
Tokenizar

Palabras

```
sentences = [
    'Today is a sunny day',
    'Today is a rainy day',
    'Is it sunny today?'
]

{'today': 1, 'is': 2, 'a': 3, 'sunny': 4, 'day': 5, 'rainy': 6, 'it': 7}
```

Comodin
OOV – Out of Vocabulary
Todas las palabras que no encuentre, serán una nueva palabra en el vocabulario: “OOV”



Ya podemos transformar frases a secuencias

```
[[1, 2, 3, 4, 5], [1, 2, 3, 6, 5], [2, 7, 4, 1]]
```

YA PODEMOS ENTRENAR UN MODELO!

Padding

Dimensiones homogeneas

Para entrenar un algoritmo de ML siempre tendremos que establecer unos inputs con el mismo número de dimensiones.
No puede ser que cada frase mida diferente.
Esto se soluciona rellenando con 0s mediante la técnica “padding”

```
sentences = [
    'Today is a sunny day',
    'Today is a rainy day',
    'Is it sunny today?',
    'I really enjoyed walking in the snow today'
]
```

```
[
  [2, 3, 4, 5, 6],
  [2, 3, 4, 7, 6],
  [3, 8, 5, 2],
  [9, 10, 11, 12, 13, 14, 15, 2]
]
```

```
[[ 0  0  0  2  3  4  5  6]
 [ 0  0  0  2  3  4  7  6]
 [ 0  0  0  0  3  8  5  2]
 [ 9 10 11 12 13 14 15  2]]
```

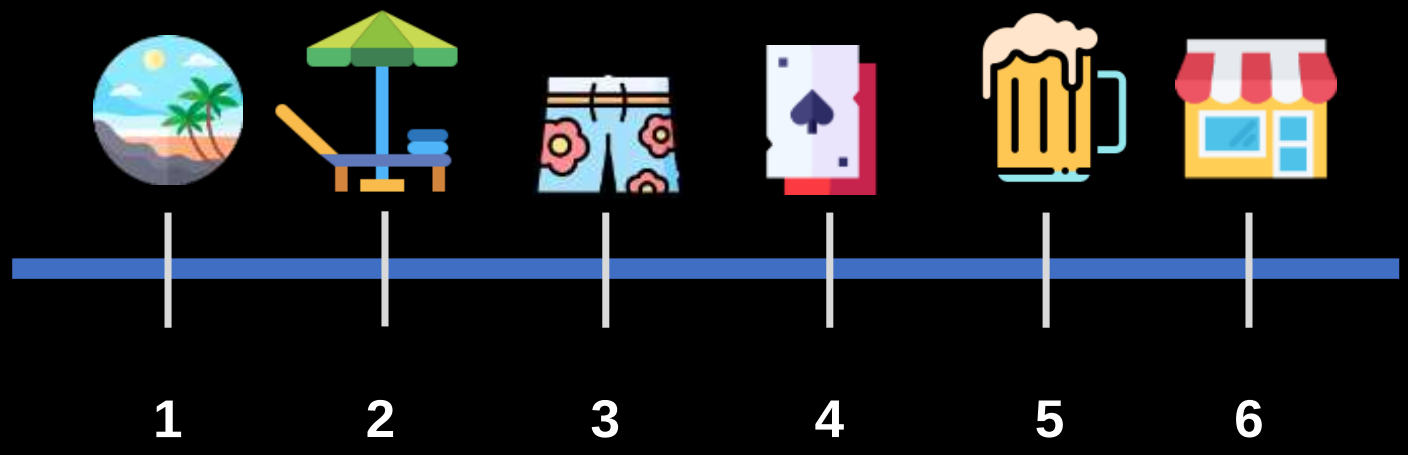
Orden vectores

Relativo

“Me llevaré a la playa una sombrilla, bañador y quizás unas cartas.
Cerveza ya compro en el chiringuito”

Eliminando stopwords…

playa: 1
sombrilla: 2
bañador: 3
cartas: 4
cerveza: 5
chiringuito: 6

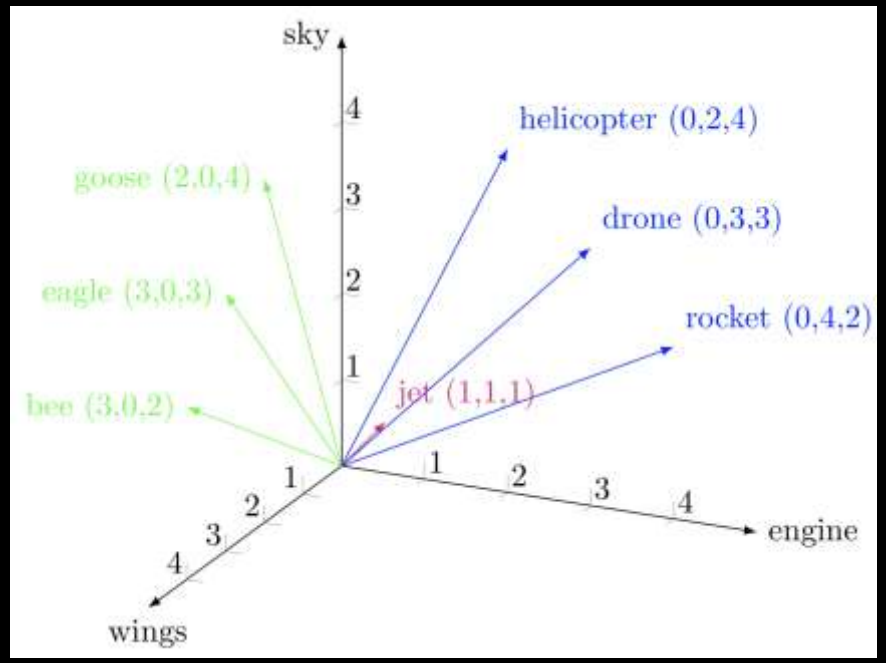


Embeddings

Vectores de alta dimensionalidad van a representar palabras

En el siguiente ejemplo vemos que cada palabra está representada en un vector en tres dimensiones. Hay algunas de las palabras que van a estar más próximas a otras.

Los modelos de Deep Learning inicializan estos vectores con pesos aleatorios y se van ajustando los pesos en cada iteración de backpropagation, por lo que cada espacio dimensional de los embeddings se adaptará a cada problema de ML.

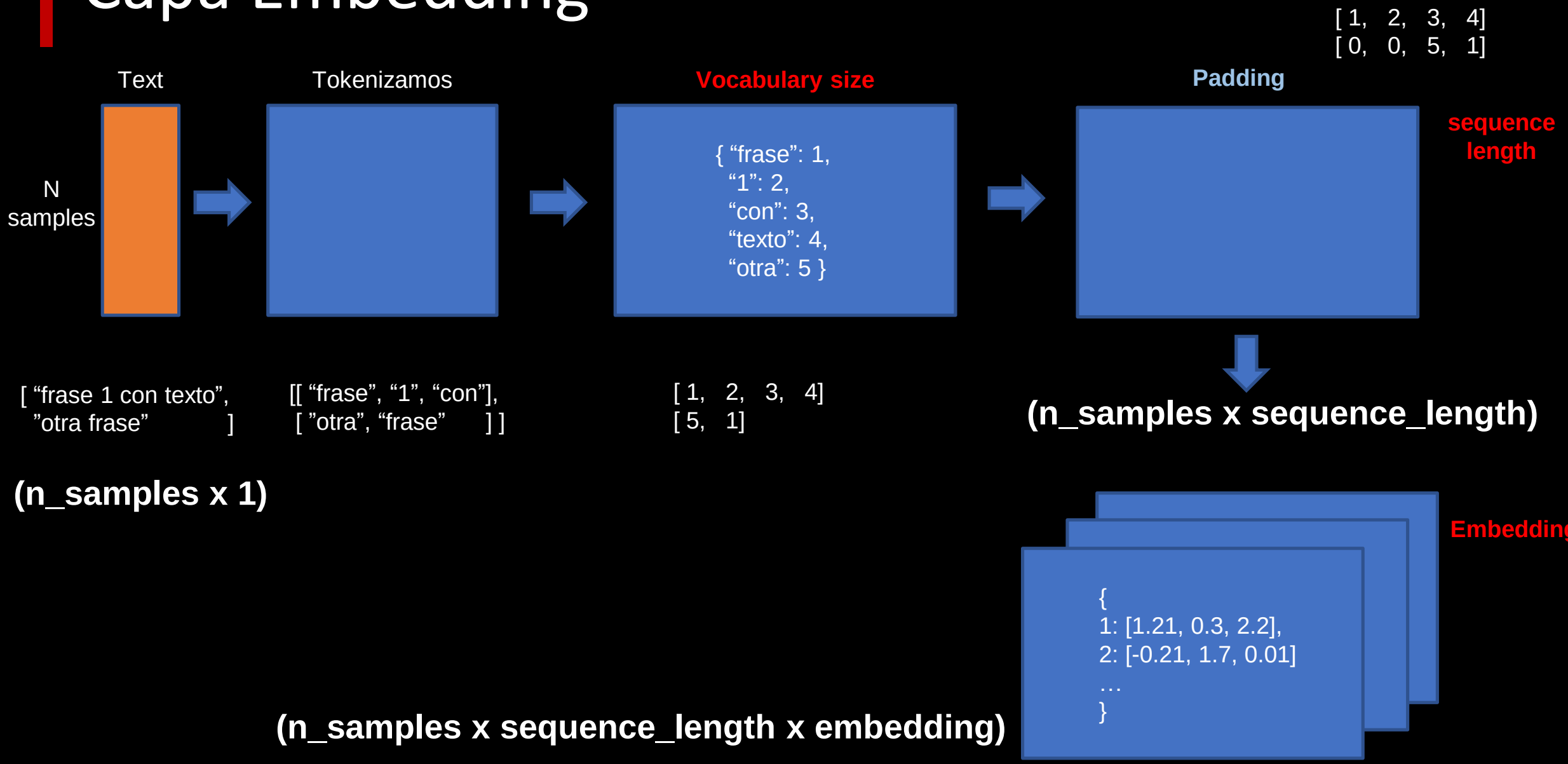


Embeddings en Tensorflow

- **Tokenizar (vocab_size)**: montar un diccionario con el vocabulario de cada observación, donde cada palabra la identificamos con un número.
- **Vectorizar**: Pasar cada frase a su secuencia de tokens numéricos
- **Padding (sequence length)**: pasamos cada frase a una secuencia de números siempre fija. Habrá que fijar la longitud fija de cada frase, mediante padding
- Establecemos la capa de **embedding**, donde ponemos la cantidad máxima del vocabulario, esto podría ser de palabras que al menos salgan 20 veces. Cuanto más vocabulario, más se ajusta el entrenamiento a todas las palabras de train. Configuramos el embedding size, que es la longitud del vector que va a representar cada palabra. Una buena práctica es usarla raíz cuarta del tamaño del vocabulario.
- Después una PoolingLayer que pasará todos los datos a una única dimensión.
- Las capas que necesitemos
- Capa de Output

```
model = Sequential([
  vectorize_layer,
  Embedding(vocab_size, embedding_dim, name="embedding"),
  GlobalAveragePooling1D(),
  Dense(16, activation='relu'),
  Dense(1)
])
```

Capa Embedding



Bibliografía

https://learning.oreilly.com/library/view/ai-and-machine/9781492078180/ch06.html

https://medium.com/@kadircanercetin/intuitive-understanding-of-word-embeddings-with-keras-6435fe92a57b