

Procesamiento de Datos

José Irribarra- Mauricio Ramírez

¹ Universidad de Concepción- Facultad de Ciencias Físicas y Matemáticas
Depto. Ingeniería Matemática

18 Julio 2019

Contenidos

1 Referencias

2 Procesamiento de Imágenes

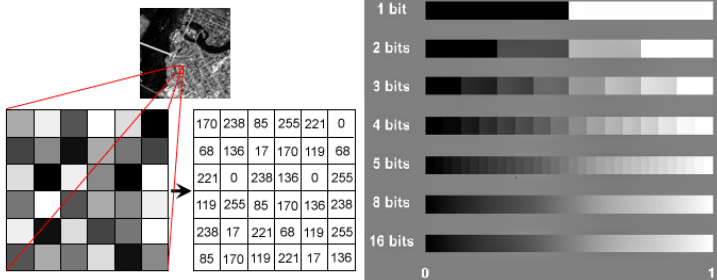
3 Procesamiento del Lenguaje

Referencias

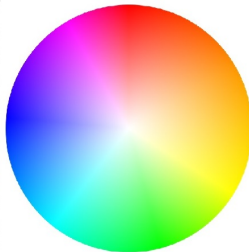
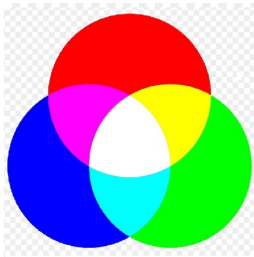
- "CS231n Convolutional Neural Networks for Visual Recognition". cs231n.github.io. Retrieved 2018-12-13.
- "Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation". DeepLearning 0.1. LISA Lab. Retrieved 31 August 2013.
- Mikolov, Tomas ; et al. (2013). "Efficient Estimation of Word Representations in Vector Space". arXiv :1301.3781.
- Goldberg, Yoav ; Levy, Omer (2014). "word2vec Explained : Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method". arXiv :1402.3722.

Imágenes

- Qué es una imagen para un computador?
No es más que una matriz de pixeles con una profundidad de color, usualmente 1 o 3.



Espectro de colores



RGB Output from 0-255 Grayscale Values

255 192 128 64 0



Cómo preprocesar la imagen ?

- Cambiar la profundidad de color.
- Normalizar los valores.
- Reducir la resolución

Qué tipos de redes me sirven para trabajar con imágenes ?

Las Redes neuronales *convolucionales*

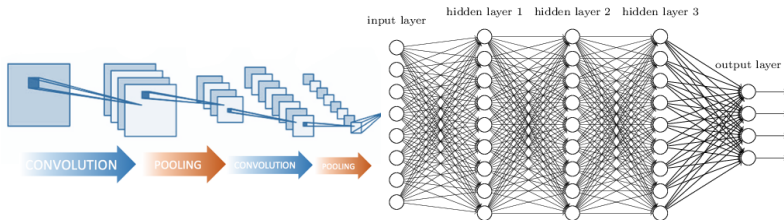


FIGURE – Convolutional neural networks

Capa de convolución

consisten en tomar grupos de pixeles cercanos de la imagen de entrada e ir operando matemáticamente (producto escalar) contra una pequeña matriz que se llama kernel

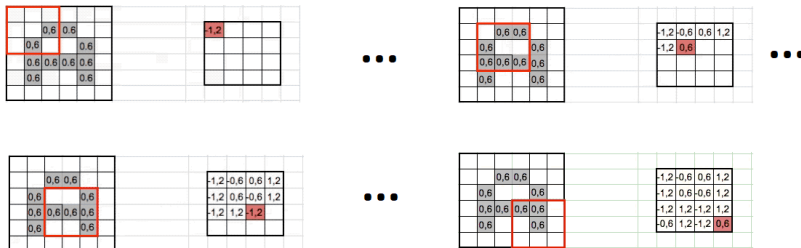
		0.6	0.6		
	0.6			0.6	
	0.6	0.6	0.6	0.6	
	0.6			0.6	

Imagen de
entrada

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

kernel

Capa de convolución



IMAGEN

		0,6	0,6		
	0,6			0,6	
	0,6	0,6	0,6	0,6	
	0,6			0,6	

KERNEL

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

CONVOLUCIÓN
DEL KERNEL

-1,2	-0,6	0,6	1,2
-1,2	0,6	-0,6	1,2
-1,2	1,2	-1,2	1,2
-0,6	1,2	-1,2	0,6

RELU

0	0	0,6	1,2
0	0,6	0	1,2
0	1,2	0	1,2
0	1,2	0	0,6

Capa de pooling



Procesado del Lenguaje (NLP)

Área de las ciencias de la computación que estudia el manejo del lenguaje en un máquina, en particular, requisitos o métodos para procesar y analizar grandes cantidades de datos en lenguaje natural.

Existe una colección en rápido crecimiento de aplicaciones útiles derivadas de este campo de estudio, algunas como :

- Corrección ortográfica, búsqueda de palabras, sinónimos.
- Extracción de información de sitios web.
- Clasificación : nivel de lectura de textos escolares, análisis de sentimientos.
- Traducción.
- Sistemas de diálogo y respuesta a preguntas complejas.

Text embeddings

Representaciones de palabras en vectores de valores reales, tales que guardan características de la palabra.

Text embeddings permite que el aprendizaje profundo sea efectivo en conjuntos de datos más pequeños. A menudo son las primeras entradas a una arquitectura de aprendizaje profundo y la forma más popular de transferir el aprendizaje a una red. Algunas de las técnicas mas conocidas son :

- Word2vec
- Glove
- FastText
- Poincare Embeddings
- sense2vec
- Skip-Thought
- Adaptive Skip-Gram

Word2Vec

Word2vec toma un texto y produce vectores únicos para cada palabra. Los vectores de palabras se colocan en un espacio vectorial de manera que palabras con contextos comunes, en el cuerpo, se ubican muy cerca unas de otras en el espacio.

Tipos de arquitectura :

- CBOW (continuous bag-of-words).
- Skip-gram.

Word2Vec : skip-gram

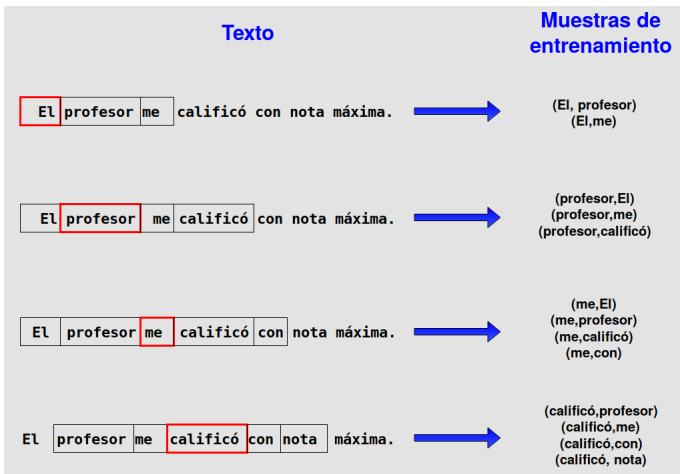
Dado un texto (**Corpus**), se genera un diccionario, para luego entrenar un modelo que analiza las palabras de cada sentencia (del Corpus), tal que, para una palabra dada, nos diga la probabilidad de que cada palabra del vocabulario sea vecina de la primera.

Por ejemplo, a la palabra “Caperucita” le seguirá “Roja” con mayor probabilidad que cualquier otra palabra. En resumen, el procedimiento a seguir :

- 1 Crear el diccionario.
- 2 Para cada palabra, transformarla en vector de 0's y 1's.
- 3 Entrenar una red neuronal con pares de palabras.
- 4 Evaluar la red.
- 5 Obtener la matriz de pesos (aprendizaje no supervisado).

Word2Vec : el modelo, entrenamiento

Para entrenar se usan pares de palabras encontradas en los datos de entrenamiento. La selección de los pares de palabras dependerá del tamaño de la ventana (**window size**). Por ej, con un tamaño de dos :

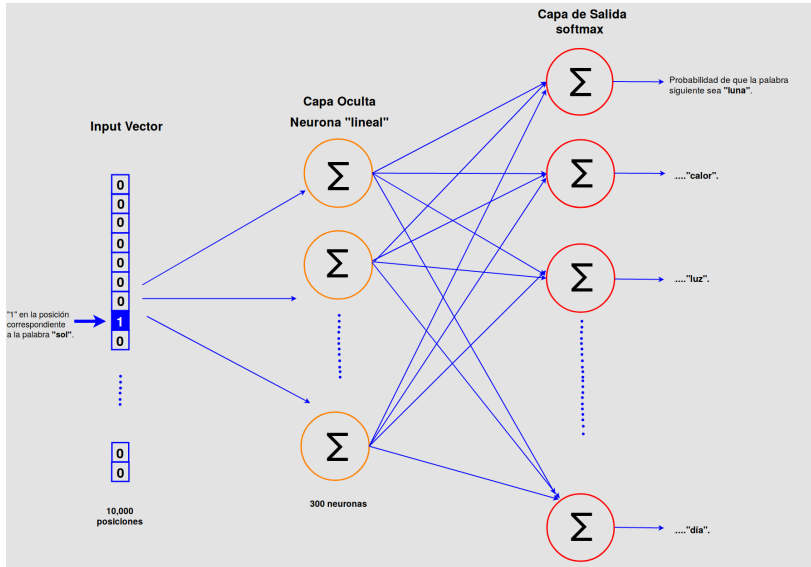


Word2Vec : el modelo, entrenamiento

Entrenamiento- Evaluación

Cuando se entrena la red en pares de palabras, la entrada es un vector de 1's que representa la palabra de entrada. A su vez, la salida de entrenamiento es también un vector de 1's que representa la palabra de salida. Pero cuando se evalúa la red entrenada en una palabra de entrada, el vector de salida será en realidad una distribución de probabilidad (es decir, un grupo de valores entre 0 y 1, no un vector de 0's y 1's).

Word2Vec : el modelo, arquitectura de la red



Word2Vec : el modelo, matriz de pesos

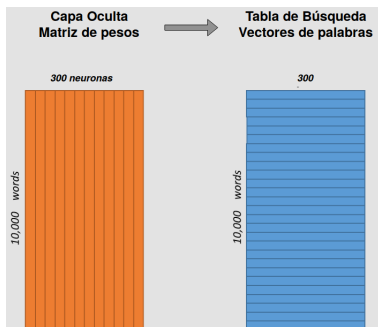
Para el ejemplo, se está aprendiendo vectores de palabras con 300 características. Por lo tanto, la capa oculta será representada por una matriz de peso con 10,000 filas (una para cada palabra en nuestro vocabulario) y 300 columnas (una para cada neurona oculta).

$$[0, 0, \dots, 0, 1, \dots, 0]_{1 \times 10000} \begin{bmatrix} w_{1,1} & \dots & w_{1,300} \\ \vdots & \ddots & \vdots \\ \vdots & & \vdots \\ w_{10000,1} & \dots & w_{10000,300} \end{bmatrix} = [w_{i,1}, w_{i,2}, \dots, w_{i,300}]$$

Observación

300 neuronas es lo que Google usó en su modelo publicado, entrenado con el Google news dataset (abarca subtítulos de películas, información de wikipedia, preguntas, etc).

Word2Vec : el modelo, arquitectura de la red



Las palabras con un significado similar estarán cerca unas de otras en el espacio, esto es porque están rodeadas por un contexto similar y esto provoca un ajuste similar de pesos.

Finalmente, una vez entrenada la red, nos quedaremos con los pesos de la capa oculta, ya que estos son los “word vectors” que se quieren aprender (de ahí el nombre “word 2 vector”).