

¿Para qué sirve la capa Emb

Formulada hace 2 años y 11 meses Modificada hace



Inicia sesión en Stack Exchange con Google



Jairo Angel

sgepruebas2000@gmail.com



Empresa EnLinea SGE

empresaenlineasge@gmail.com

1 cuenta más



7



He estado leyendo sobre LSTM y hay alg
capa de tipo **Embedding** a la entrada de

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Embedding(big
model.add(tf.keras.layers.LSTM(200))
model.add(tf.keras.layers.Dense(biggest
```

Busqué en la documentación de tensorflow y leí al respecto, sé lo que hace pero no acabo de entender la ventaja o funcionalidad que tiene agregarla.

Saludos y Gracias...

python

python-3.x

tensorflow

redes-neuronales

lstm

Compartir Mejora esta pregunta

Seguir

editada el 13 ago. 2021 a las 11:19



Rubiales Alberto

7,793 9 21 40

formulada el 12 ago. 2021 a las 23:01



Omar Puentes

165 8

2 respuestas

Ordenado por: Mayor puntuación (por defecto)



¿Qué es?

3



La capa `tf.keras.layers.Embedding` es un método de [clustering](#), y como todo método de clustering matemático lo que pretende es, **agrupar palabras similares en grupos homogéneos y que dichos grupo sean lo más heterogéneos entre si, unos de otros**. Explicándolo más fácil, que las palabras similares van juntas y están lo más separadas posible de las palabras no similares.



La explicación de @Alfredo Maussa es muy buena.



¿Qué sentido tiene?

Antes de que existiera este método matemático, cuando se quería analizar texto, se usaba **One Hot Encoding**.

Como bien sabes, una red neuronal, o cualquier otro proceso matemático, no entiende el texto, ni las palabras, por lo que **debes transformar las palabras en números..** [One Hot Encoding](#) consiste simplemente en una matriz de ceros y unos, que transforma las palabras a números. Aquí un ejemplo:

One Hot Encoding			
Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Este método era algo rudimentario, por lo que se crearon distintos métodos para mejorarlo, entre ellos el que implementa Tensorflow con `tf.keras.layers.Embedding`. Las mejoras además de representar el texto como números al igual que hace One Hot Encoding, son las siguientes:

1. [Reducción de la dimensionalidad](#): este método reduce la dimensionalidad del texto haciendo posible de entrenar conjuntos de datos más grandes.
2. Creación de un contexto: al aplicar un método de clustering que diferencia unas palabras de otras, es capaz de proporcionar a la red neuronal información valiosa

Tu pregunta

Una vez sabemos esto, hemos prácticamente respondido a tu pregunta:

- Funcionalidad y Ventajas: con respecto a métodos anteriores permite entrenar mayor cantidad de datos gracias a la reducción de la dimensionalidad, y además codifica los datos dándole información extra a la red neuronal gracias al método de clustering.

Bonus extra

- La capa `tf.keras.layers.Embedding` no solo se puede usar en texto, se puede usar en cualquier problema que tenga alta dimensionalidad. Por ejemplo, **sistemas de recomendación**
- Actualmente existen mejores procesos matemáticos, como por ejemplo **Transformers** basado en capas de codificación y decodificación, con los que se han creado modelos de texto muy avanzados como BERT o GPT-3

Compartir Mejora esta respuesta

Seguir

editada el 13 ago. 2021 a las 14:27

respondida el 13 ago. 2021 a las 11:19



Rubiales Alberto

7,793 9 21 40

Me aportó mucho tu respuesta. Gracias!!! Pero en el tutorial que vi utilizan la capa embedding en otro contexto, o sea, se utiliza primero la clase Tokenizer de keras para convertir las oraciones en secuencias de

enteros y estas secuencias son las que se les pasan al modelo para entrenarlo. Bajo estas condiciones ¿qué podría aportar la capa embeddings? – [Omar Puentes](#) el 13 ago. 2021 a las 14:22

Hola @OmarPuentes. Lo que pone en la respuesta, clustering y reducción de dimensionalidad. Lo he editado para que se entienda mejor, ya que esa parte inducía a confusión. – [Rubiales Alberto](#) el 13 ago. 2021 a las 14:26 ✎

Si, luego lo vi, estaba editando el comentario pero me andaba dando error. En el tuto al que hago referencia la oración más larga es de 40 caracteres y la salida de la capa embeddings es de un tensor de 200 posiciones, o sea, no logro ver la reducción de dimensiones ahí – [Omar Puentes](#) el 13 ago. 2021 a las 14:33

Ok, lo que reduce no son los caracteres. Es todo tu dataset, es decir, tu le has pasado un conjunto de textos de tamaño "X" y cada texto tiene una longitud "y". Y toda esa gran cantidad de textos, con todas sus palabras se han quedado reducido a un vector de 200 posiciones. Puedes reducirlo más cambiando los parámetros de la capa. – [Rubiales Alberto](#) el 13 ago. 2021 a las 16:06 ✎

- 1 Si, claro que lo merecen. De hecho, me disculpo, es que soy nuevo en la plataforma y aún no la domino. Saludos – [Omar Puentes](#) el 17 ago. 2021 a las 14:59

**2**

La explicación personal que tengo es; una forma de reducción del tamaño de las entradas y al mismo tiempo establece una relación semántica entre distintas palabras aprovechando propiedades de vectores.

De tal forma que si ingresas por ejemplo "Toro" el embedding va a transformar ese token de la palabra Toro a uno distinto en terminos adjetivos concretos. por ejemplo, "macho", "animal", "quadrupe", "tiene cuernos".

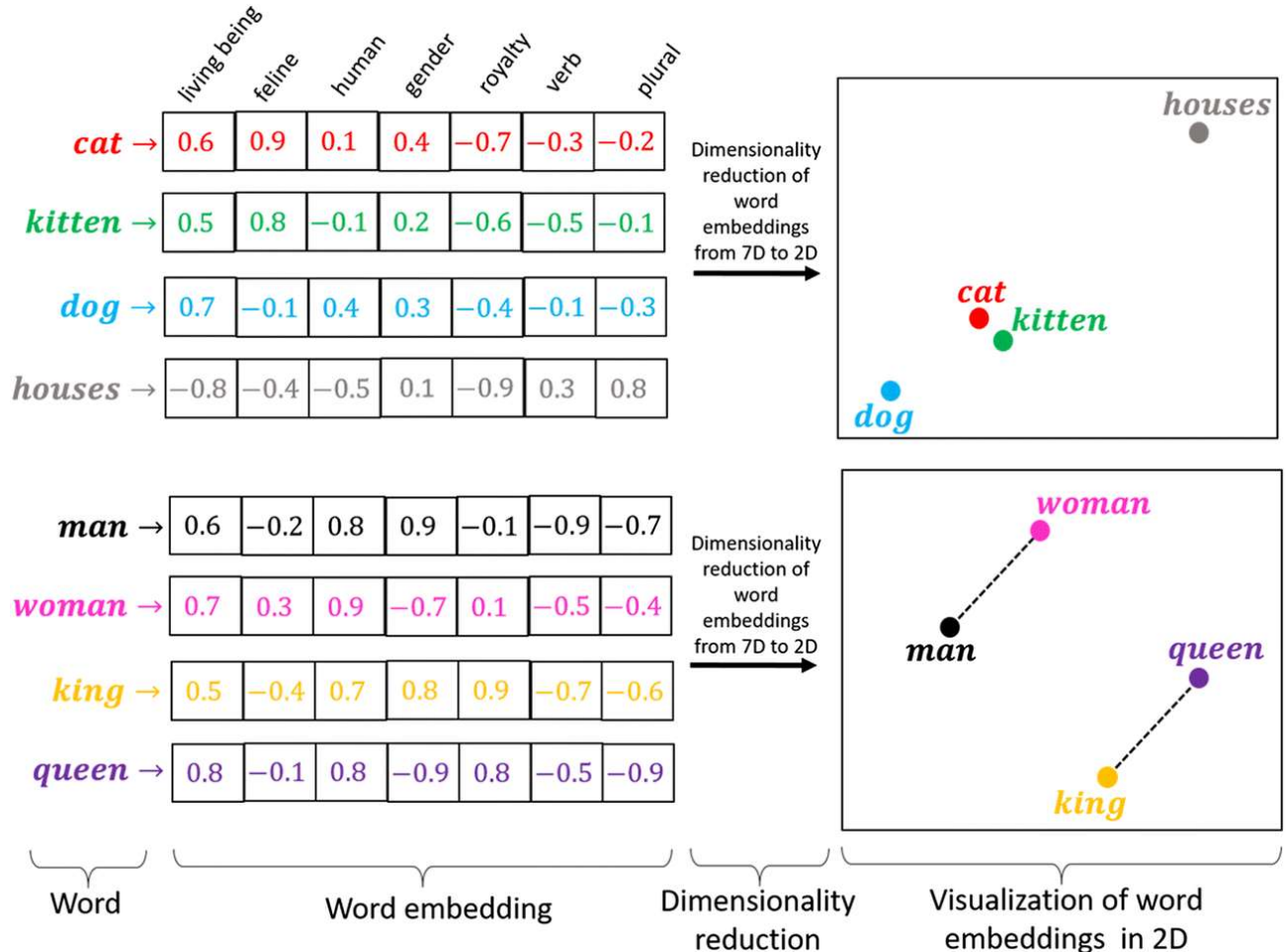
Así la red neuronal es ciega e invariante ante entradas similares, por ejemplo si introducimos ahora "Búfalo", sin embedding la red neuronal tendría que inferir desde cero las características del token (es un animal, si es macho o hembra, si tiene cuatro patas o dos, etc...). En cambio **con embedding** la red neuronal no trabajaría con el token de "Búfalo" lo haría mas bien con la proyección sobre los adjetivos... osea la entrada equivalente sería: tiene cuatro patas (la red neural dice: ah siiii?), tiene cuernos, no se sabe si es macho o hembra solo con el nombre entonces asignemosle 60% ya que la mayoría de mis datos son machos (bias de la red). Para la red neuronal entonces sería muy similar trabajar con "Toro" que con "Búfalo", y como resultado lo que aprenda con una clase lo inferirá para la otra lo que facilita el aprendizaje y la asociación.

Ahora, el ejemplo más usado es el de "Rey" y "Reina", que están asociados directamente a si es hombre o mujer (adjetivos), probablemente tengan otros como "ser gobernante", "bestimenta ostentosa". Esta asociación que causa el embedding permite que la red neuronal trabaje no con los tokens individuales "Rey" y "Reina" que aunque para un humano se vean muy similares en términos de token un computador no lo sabría, sin embargo la capa de embedding se encargaría de darle una "interpretación" semántica asociandolo con adjetivos.

La forma en que se asocian es con una proyección sobre un espacio algebraicamente distinto (que es lo que hace una matriz), entonces cada token escrito como vector (en la que la dirección

y la posición del vector no tiene significado alguno sin embedding) sería puesta en un sistema de referencia de menor dimensión pero con correlación entre vectores (tendría significado tanto las posiciones como las direcciones)... por ejemplo todo lo que categorice como "ser vivo" apuntaría en la misma dirección bajo esa dimensión ya que tendrían la misma componente.

Tomaré la siguiente imagen de [word-embedding](#) recomendando el artículo al mismo tiempo.



Así mismo la distancia en el espacio de "embedding" te diría que tan similares o distintos semanticamente hablando son dos vectores. por ejemplo: un rey tiene una relación similar a una reina en lo que un hombre sería a una mujer. (en este caso la dirección sería la misma)

En la imagen puedes ver como entran las palabras [cat,kitten,dog,houses,man,woman,king,queen], las cuales son llevadas a un espacio de menor dimensión pero que tienen correlación semántica [ser vivo,felino,humanogenero,realiza,verbo,plural], encontrarás que rey y hombre tienen una correlación fuerte (sus filas tienen valores similares, significa vectores casi paralelos), lo mismo para Reina y mujer. podrás establecer una relación de analogías haciendo una resta: vector (rey - reina)=(hombre-mujer). o una relación de similitudes haciendo el producto punto (rey,hombre)

Por favor comenta si un parrafo es poco claro para intentar corregirlo.

Compartir Mejora esta respuesta

editada el 13 ago. 2021 a las 0:52

respondida el 13 ago. 2021 a las 0:23

Seguir



Alfredo Maussa

1,562 6 20

Muy buena tu respuesta!!! Pero en el tutorial que vi utilizan la capa embbeding en otro contexto, o sea, se utiliza primero la clase Tokenizer de keras para convertir las oraciones en secuencias de enteros y estas secuencias son las que se les pasan al modelo para entrenarlo. Bajo estas condiciones ¿qué podría aportar la capa embbedings? – [Omar Puentes](#) el 13 ago. 2021 a las 14:21
