

متنی خود را با کلماتی که از طریق مدل

در ابتدای کار در دایره زبانی طبعی  
خطای اضافی و ناهماهنگی می باشد.

در مرحله بعد کلمات را به اعداد تبدیل می کنیم چون کامپیوتر توانایی خواندن کلمات را ندارد.

یکی از تکنیک های تبدیل کلمات به اعداد word embedding می باشد.

همین طرز فکری دامه ها، مثال به ساخت مدل یا همان شبکه عصبی می پردازیم.

این شبکه لایه های متفاوتی دارد

لایه اول این شبکه convolution می باشد.

لایه فیلتر - که ویژگی های مهم تصویر را استخراج می کند.

لایه Pooling هم استفاده می کنیم، برای اینکه اندازه فضای مناسبی را کاهش دهیم و همچنین تعداد پارامترها را

Word embedding را به روشی دیگر می توانیم بسازیم.

رایج ترین روشی می باشد.

یکی روشی یادگیری رایج است، که کلماتی که شبیه به هم هستند شباهتی را می دانند.

مثال:

	Apple	orange
جیب	0.00	0.01
مقام	-0.01	0.00
سبز	0.03	-0.02
عذرا	0.95	0.97

=> word embedding

مجموعه ورتال

که نشان می دهد این

دولت به هم شبیه هستند.

بعد از Pooling ارسال LSTM استفاده می کنیم. بعد از این لایه، لایه آخر که شبیه به شبکه عصبی می باشد.

شبکه عصبی fully connected می باشد.

1- زمانی می‌کنیم از گنگل درامو

2- دیتای زانهای بنده را می‌خوانیم

$df.fillna = >$   $NaN$  <sup>noun</sup> داشتیم می‌چاک در دیتا فریم مقدار  
یا یک تعدادی که خودمان اعلام می‌کنیم پر می‌کنیم.

از کتاب خواند Keras متن `tokenizer` برایش می‌نویسد: `tokenizer` برای تبدیل  
بأعمال این تابع بر روی یک آرایه‌ای دیتا فریم یعنی `text` می‌توانیم از یک سری تالیفات  
ما استفاده کنیم. تالیفاتی که می‌خواهیم استفاده کنیم اینهاست:

می‌خواهیم به هم بزنیم یک عدد نسبت به صفت، از طریق `tokenizer.word_index`  
(که آن عدد `index` یا `index` است)

`padding`: به اعمال `padding` می‌توانیم به هم بزنیم  
کتابخانه از Keras برای عملیات `padding` استفاده می‌کند.

مثال:  $a = [1, 2, 3], [4, 5, 6], [7, 8]$

`array([[1, 2, 3],  
[4, 5, 6],  
[7, 8, 0]])`

`padding = 3, max_len = 3` `padding`  
تایم اندازه طولی می‌خواهیم  
داده‌ها را `padding`

چون می‌خواهیم  
این را به از آن تالیفات با مقدار  
صفتی همی کنیم.

`truncating = ?`

دستی برای `padding` است که در ادون `world` ما تبدیل به عدد کند. در مرحله بعد باید داده `train` و `test` را مشخص کنیم. در سال 80٪ داده `padding` را به `train` تخصیص می دهیم و 20٪ باقی مانده را به `test` اختصاص می دهیم.  
 راب داده `train` و `test` اضافه کنیم.  
 ما `label`

در کتاب `numpy` مانند `numpy` دسترسی وجود دارد به اسم `array` که کارایی به سرعت زیاده است.

$A = [1, 2, 3, 4]$

در واقع  $n \times p$  `array(A)`  $\Rightarrow [1, 2, 3, 4]$

ما تمیزی در می آورد.

در مثال `text` `glove-6B-token` ما یک ارائه برداری از معنی کلمه را داریم.

این مثال `text` `embedding` را به دست می آوریم.

همین را برای `vector` -

در مرحله بعد: ما `vector` و `embedding` را داریم و به کمک حله `for` کلمات داخل بردار کلمه را

`embedding`

دیتا را با کلمات داخل `embedding` ستایش می کنیم

مورد آورده و در ماتریس قرار می دهیم

فکلی که ساخته ایم، با توجه به جایگاه کلمه که می نرزد سطح ماتریسها

نتیجه میانی به `matrix` آمده انانی مرحله بینی `embedding` می شود وزن عالی لایه ورودی شبکه

`matrix`

ما.