

بخش 1:

فرمول ابعاد خروجی convolutional layer :

X = input shape

K = kernel size

P = padding

S = stride

$$output\ shape = \frac{x - k + 2p}{s} + 1$$

فرمول ابعاد خروجی pooling layer

$$output\ shape = \frac{x - k}{s} + 1$$

فرمول پارامترهای RNN:

$$parameters = (inputSize + 1) * units + units^2$$

64*Conv2d 7*7

$$output\ shape = (30-7+1)=24 \Rightarrow 24*24 * 64$$

$$parameters = 64 * (7*7*3+1)=9742$$

Maxpool2d 2*2 default stride = 2

$$output\ shape = (24-2)/2 + 1 = 12*12 * 64$$

$$parameters = 0$$

128 * Conv2d 5*5


$$output\ shape = (12-5+1)=8 \Rightarrow 8*8*128$$


$$parameters = 128*(5*5*64+1)=204928$$


Maxpool2*2 default stride = 2


$$output\ shape = 4*4*128$$


$$parameters = 0$$


256 * conv2d 3*3 
output shape = $(4-3+1)=2 \Rightarrow 2*2*256$
parameters = $256*(3*3*128+1) = 295168$


Maxpool2*2 default stride = 2 
output shape = $(2-2)/2+1=1 \Rightarrow 1*1*256$
parameters = 0


Flatten 
output shape = $1*1*256 = 256$
parameters = 0


Dense with 128 units 
output shape = units = 128
parameters = $128*256 + 128 = 32896$

RNNs 
output shape = 128
parameters = $128*(256+1) + 128^2$

Dense 128 units after RNNs 
output shape = 128
parameters = $(128*128)+128$

Cocnat Dense 
output shape = $128+128=256$
parameters = 0

Dense with 128 unit after concat 
output shape = 128
parameters = $256*128 + 256$

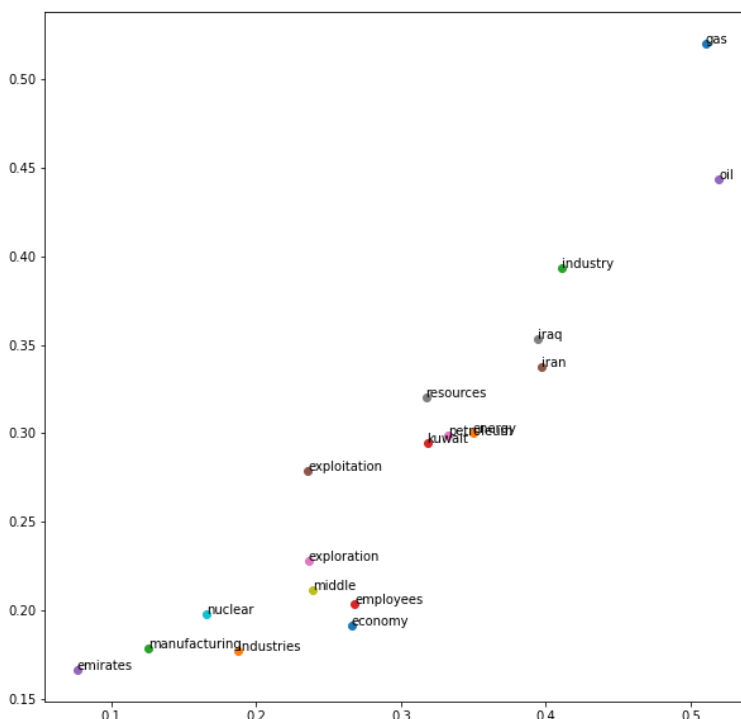
Final dense with 1000 units 
output shape = 1000
parameters = $1000*128+1000$

منبع: [androidkt](http://androidkt.com)

بخش 2)

ابتدا طبق خواسته سوال پیش می رویم تا به یک لیست 100 از متن خبر ها برسیم این لیست درواقع corpus ماست حال نیاز است که کلمات موجود در این corpus را embedded کنیم به اینصورت که برای هر کلمه یک tuple دوتایی داریم که ایندکس اول آن کلمه اصلی ماست و ایندکس دوم آن کلمه ای است که در context ما قرار داد نحوه تایین context به وسیله پارامتری به نام window انجام می شود که می شود تمام کلماتی که به فاصله window از قبل و بعد از کلمه اصلی ما وجود دارد.

اما نیاز داریم که تمام کلمات را مدل نکنیم چرا که بسیاری از کلمات طبق قواعد زبانی هر زبان نوشه شده اند و بار معنایی زیادی به جمله نمی دهند لذا ما بسیاری از کلمات (مثل کلماتی که گفته شد حتی کلماتی که ممکن است در پروژه ما اهمیت چندانی نداشته باشند) (در کل این کلمات را با نام stopwords می شناسیم)، حروف اضافه، علائم (punctuations) و اعداد را corpus خود حذف می کنیم این مرحله جزیی از بخش تمیز کردن داده ها می باشد سپس نیاز است که برای هر کلمه (unique words) context آن کلمه (تمام کلماتی که در tuple دوتایی آن کلمه وجود داشتند) را بدست آوریم. که می توانیم این لیست ها را به وسیله One-hot vector مدل کنیم و سپس به مدلی با ورودی focus words (لیست تمام کلمات) و خروجی context words (لیست context کلمات focus words) بدهیم تا embedding ما تمام شود حال فضای برداری کلمات داده شده در سوال را plot می کنیم که شکل آن را مشاهده می کنید.



سوالات تئوری

1. در این جا ما در کل چیز های که بار معنایی ای نداشتند را حذف کردیم مثل stop words ، punctuations اعداد همینطور تمام کلمات را lower case کردیم تا زمانی که می خواهیم که کلمات unique را بدست بیاوریم بخاطر بزرگ یا کوچک بودن کلمات دچار مشکل نشویم.

2. داده های ورودی و خروجی با one-hot vector مدل شدن که ورودی های focus word ها و خروجی های context word های متناظر با focus word ها هستند.
3. در اصل ما از embedding کردن کلمات می‌خواهیم به رابطه معنا داری بین کلمات یک corpus برسیم حال ما پنجره ای برای هر کلمه تعریف می‌کنیم تا بتوانیم کلمات را با توجه به context که در آن قرار دارند را متمایز کنیم حال اگر سایز پنجره بزرگ تر باشد یعنی context ما نیز بزرگ تر می‌شود به نظر این اتفاق خیلی خوب است چون می‌توانیم کلمات را با دقت بالاتری از هم متمایز کنیم اما باید در نظر داشته باشیم که با بالا بردن سایز پنجره ممکن است کلماتی در آن وجود داشته باشد که ربطی به کلمه اصلی ما ندارد و همینطور با بالا بردن سایز پنجره پیچیدگی محاسباتی ما افزایش می‌یابد. اما اگر سایز پنجره را کم کنیم یعنی context خود را کم کردیم که این معنی را میدهد که دقت ما کاهش می‌یابد و ممکن است کلمات مهمی در آن context را از دست دهیم
4. دو نورونی که در لایه پنهانی داریم در واقع word embedded هستند به اینصورت که وزن های متناظر با هر ورودی (fully connected) embedded ان کلمه را مشخص می‌کند

منبع: [towardsdatascience](https://towardsdatascience.com/word-embeddings-part-1-1d0e0e0e0e0e)

بخش 3

طبق سوال پیشرفتیم تا به بخش shuffle کردن رسیدیم همانطور که در تصویر پایین مشاهده می‌کنید عملیات shuffle شدن به درستی انجام شده است



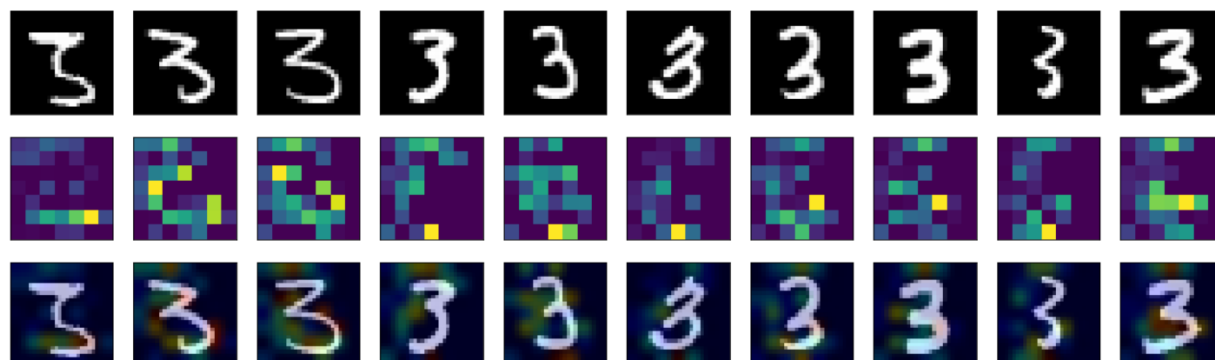
با شبکه داده شده داده ها را آموزش دادیم و به دقت 98 درصد رسیدیم

```
Epoch 14/15
844/844 [=====] - 4s 4ms/step - loss: 0.0072 - accuracy: 0.9976 - val_loss: 0.0606 - val_accuracy: 0.9888
Epoch 15/15
844/844 [=====] - 4s 5ms/step - loss: 0.0074 - accuracy: 0.9976 - val_loss: 0.0490 - val_accuracy: 0.9883
```

پس از مشاهده heatmap ها به این نتیجه می‌رسیم که مدل در تلاش است تا ویژگی های تصاویر را استخراج کند لایه آخر conv لایه ای هست که ویژگی ها کامل استخراج شده و به لایه dense میدهد تا دسته بندی انجام شود

تصاویر زیر را مشاهده کنید از مشاهده این تصاویر میتوان برداشته کرد که مدل تلاش کرده تا الگوی هر عدد را یاد بگیرد.

مثلا عدد 3:



عدد 2:

