Assignment NO.7 Solutions

Deep learning | spring 1401 | Dr.Mohammadi

Teacher Assistant:

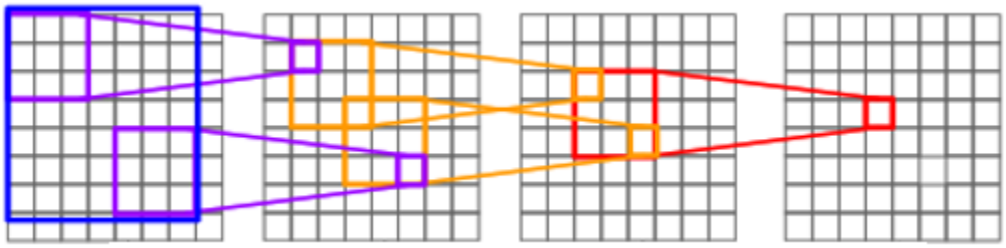
Amirmehdi Nikokaran

Student name : **Amin Fathi**

Student id : **400722102**

Problem 1

Cnn ها اطلاعات زیادی را در لایه های pooling از دست می دهند که این کار موجوب کاهش وضوح فضایی شده و در نتیجه خروجی شبکه نسبت به تغییرات کوچک در ورودی تغییر نمی کند ، این مشکل در کاربرد هایی مانند semantic segmentation به وضوح به چشم می آید چرا که pooling هم موقعیت دقیق پدید ه ای که در تصویر داریم را ازدست میدهد و ه م اطلاعات زیادی را از دست میدهد چرا که بخش خوبی از اطلاعات را ایگنور کرده و به طور مثال در یک کرنل ۳*۳ فقط یک المان را در نظر میگیرد



اطلاعات زیادی در pooling ها از بین می رود

معمولا برای حل این مشکل در CNN ، دست به ساخت معماری های پیچیده در CNN برای بازیابی برخی اطلاعات از دست رفته می زنند .

همانطور که در معماری CNN آموخته ایم هر نورون خروجی اسکالری را تحویل می داد ، در کپسول نت اما هر کپسول مجموعه ای از نورون هاست و این بار خروجی هر کپسول برداری با N داریه است (N تعداد نورون های کپسول) ، کپسول ها در واقع کوچکترین واحد در شبکه کپسول نت هستند و توانایی ذخیره اطلاعات بیشتری دارند ؛ هر کپسول دو نوع اطلاعات را خروجی می دهد : ۱ – اینکه آیای object را دیده است (presence) و ۲- در مورد pose یا موقعیت عناصر آن object صحبت کند یا انواع ویژگی های محلی دیگر که در cnn به دست نمی آیند مثل deformation, scale, velocity , orientation و ...

و این ویژگی دوم کپسول نت کمک میکند که شبکه مقاوم شود نسبت به اینکه تصویر دقیقا از روبه رو گرفته شده باشد ، یا مایل باشد یا سر و ته باشد و ... و اگر هم جابه جایی درobject رخ داده تمام اجزای آن object با یک منطق ریاضی کاملا دقیق جا به جا شوند (به مثال پاندا در مرجع بیاندیشید که در آن اگر المان های صورت پاندا به صورت رندم جا به جا میشدند کپسول نت آن را به عنوان no panda طبقه میکرد ولی چنانچه اجزای صورت آن به صورت منطق ریاضی یکسان تغییر می کردند به صورت panda طبقه بندی می کرد

Image_TrainingDataSetType

Actual Result: Panda;

CNN Result: Panda;

Capsule Net Result: Panda

Image_RotatedPanda

Actual Result: Panda;

CNN Result: Not Panda;

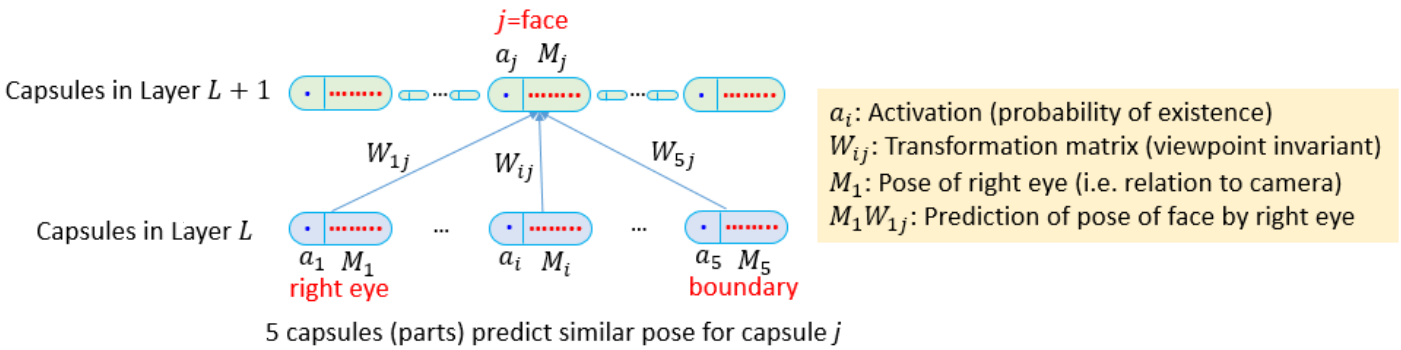
Capsule Net Result: Panda

Image_Deformed

تصویر deformed شده را کپسول نت هم در not panda طبقه بندی میکند چرا که رابطه بین چشم ها و دماغ به درستی تغییر نیافته اند بر خلاف Rotatedpanda که در آن هر دو ۲۷۰ درجه چرخیده اند فلذا کپسول نت به درستی آن ها را panda تشخیص داده است

در مورد سایر تفاوت های با CNN می توان این موارد را ذکر کرد که برای train کردن CNN تعداد زیادی تصویر نیاز هست اما در capsule network با trainin data کمتری هم میتوان عملکرد مناسبی را انتظار داشت و این به معنای کمتر بودن پیچیدگی capsule network می باشد چرا که به طور مثال کاری که یک شبکه CNN در 20 الی 30 لایه میکند را capsule network در 3 یا 4 لایه می تواند انجام دهد . البته این نکته را هم باید ذکر کرد که train کردن شبکه کپسول نت زمان بیشتری را لازم دارد و این مسئله باید به مرور زمام بهبود یابد .

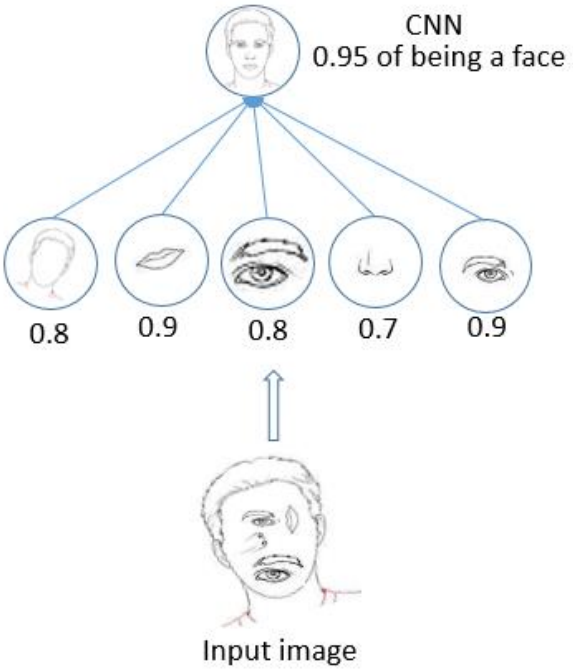
توضیح مختصر در باره معماری capsule network:

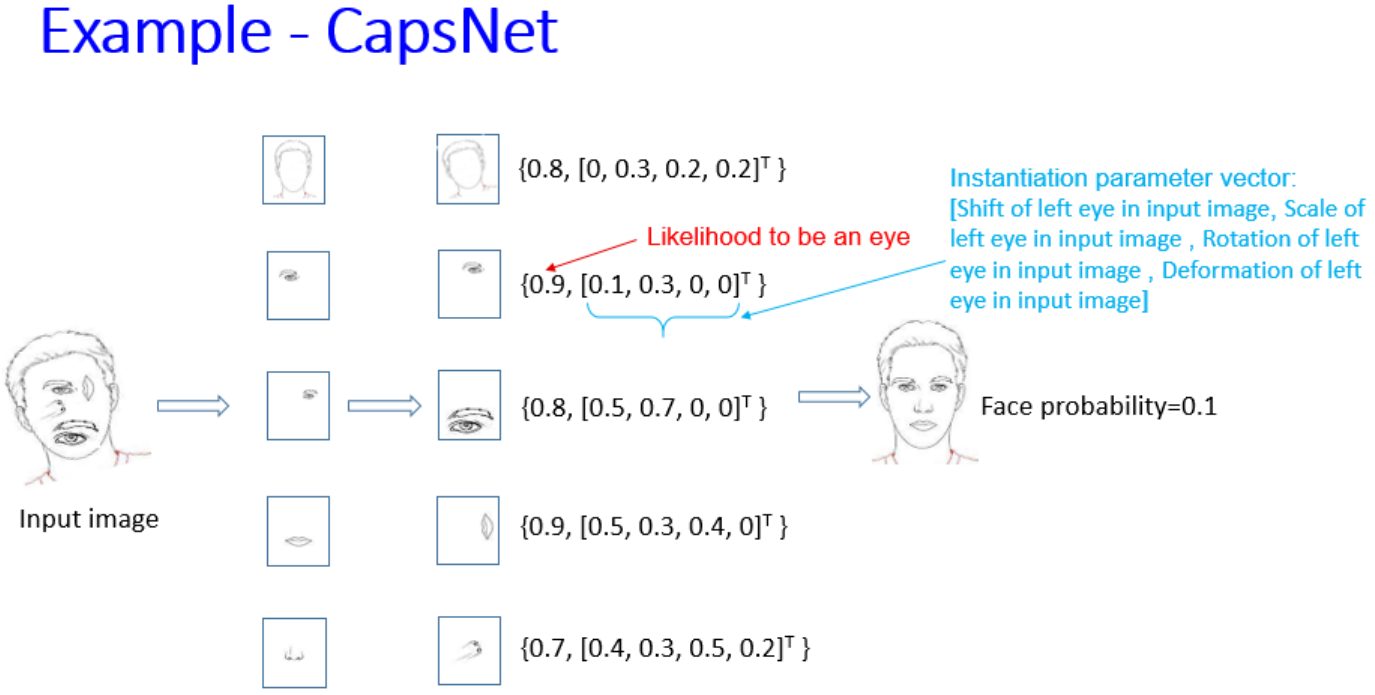


در شکل بالا یک لایه L و یک لایه L+1 را مشاهده میکنیم و هر کدام از بیضی های کپسول شکل هم به طور سمبلیک بیانگر یک capsule است. هر کپسول یک کمیت دارد که آن را با a نشان میدهند (activation) ، به طور مثال فرض کنید کپسول j در لایه L+1 مسئول تشخیص صورت است ، این کپسول ورودی هایی از ۵ کپسول لایه قبل دارد (با وزن هایی مختص هر کپسول ، و مشاهده میشود هر کپسول لایه پایین تر مشخصاتی از صورت است که اگر همه آن ها باشند در آن صورت کپسول jمی تواند تشخیص بدهد که صورت انسان در تصویر ورودی شبکه هست یا نه) چنانچه مقدار a j مقدار بزرگی شود در این صورت این کپسول تشخیص می دهد که صورت در تصویر وجود دارد و بعد از آن ۱۶ درایه دیگر بردار MJ مشخص کننده pose مربوط به صورت است و اطلاعات دیگری از صورت می دهد .

$$\text{pose of } j \text{ is } M_j = Average\ (M_1W_{1j}, M_2W_{2j}, \dots, M_5W_{5j})$$

به طور مثال ضعف cnn را در تشخیص صورت در شکل زیر مشاهده می کنید :





منابع :

مقاله صورت سوال ، لینک مرجع و مباحث درسی درس طراحی الگورتیم دکتر آنالویی

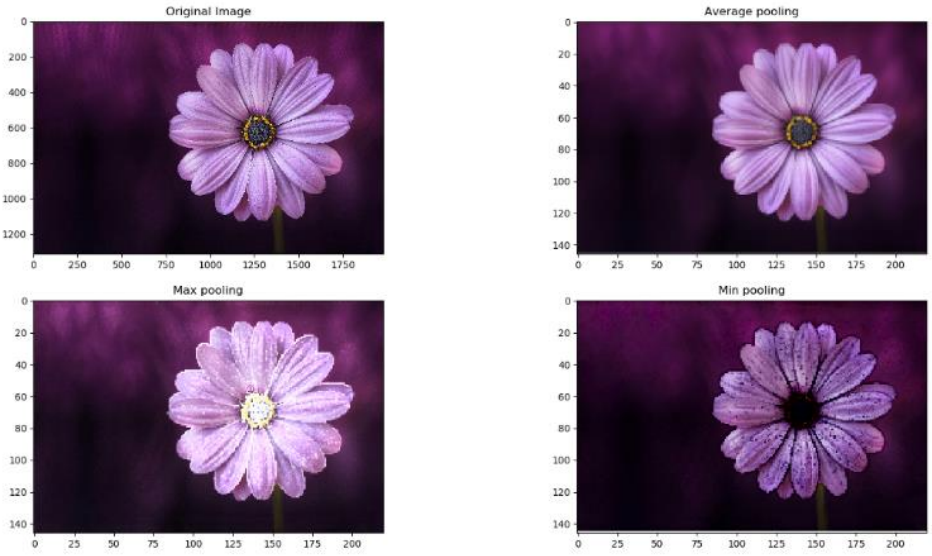
Problem2

مزایا:

اصلی ترین کاربرد لایه های ادغام (pooling layer) برای کاهش ابعاد نقشه ویژگی (feature map) است ، چرا که این لایه ها از چند درایه و خانه همسایه طبق الگوی مشخصی (... , max , average) یکی را به نمایندگی انتخاب کرده و با دیگر درایه ها کاری ندارند و به اصطلاح آن ها را دور ریخته و به این ترتیب ابعاد نقشه ویژگی کاهش میابد و این کار سبب کاهش تعداد پارامتر ها و همچنین محاسبات سریع تر و حجم محاسبات کمتر در شبکه همگشتی (CNN) می شود این اقدام همچنین به مقاوم سازی شبکه نسبت به نویز و overfit شدن می انجامد . از دیگر مزایای pooling می توان به مقاوم شدن شبکه نسبت به شیفت مکانی در تصاویر ورودی هم اشاره کرد ، چرا که هر درایه در نقشه ویژگی نماینده همسایگانش هست و شیفت اندکی مکانی تاثیر چندانی بر نقشه ویژگی نخواهد داشت .

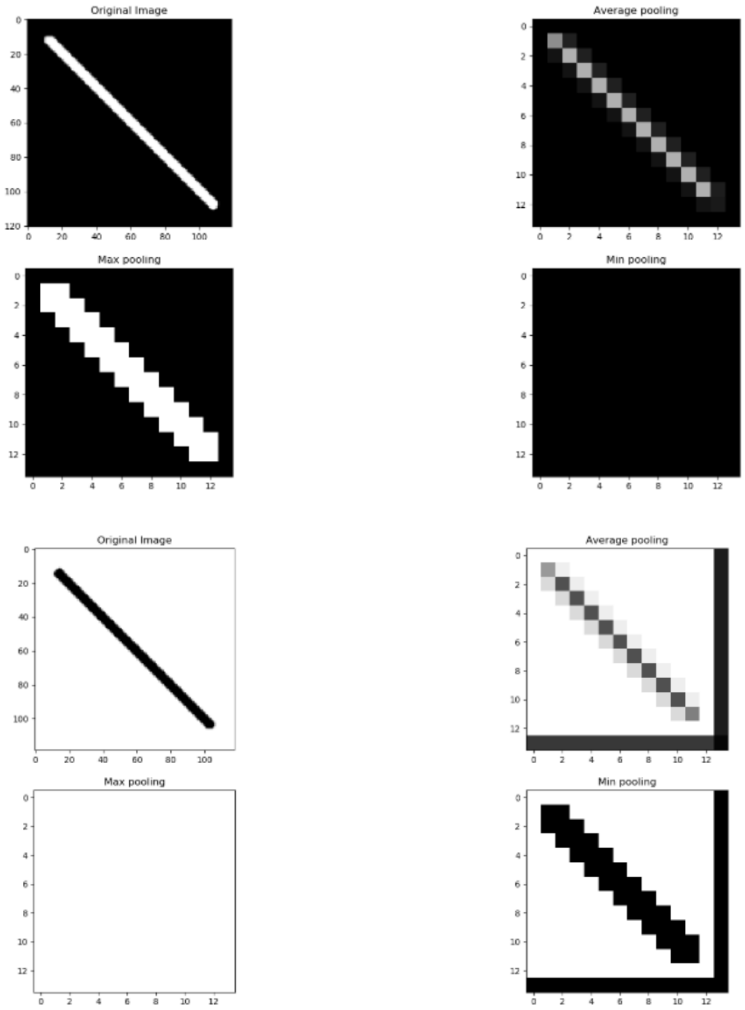
معایب :

اگر اندازه کرنل pooling بزرگ انتخاب شود می تواند منجر به از بین رفتن برخی اطلاعات مکانی شود (در قسمت قبل و سوال اول به آن اشاره شد) . استفاده از max pooling و min pooling در مواقعی که پیکسل های سیاه و سفید (0 و 255) همسایه هستند می تواند مشکل ساز شود چرا که به طور مثال نماینده چند پیکسل سیاه در نقشه ویژگی ها می تواند پیکسل نسبتا روشنی شود (این پدیده در شکل زیر به خوبی نمایش داده می شود)



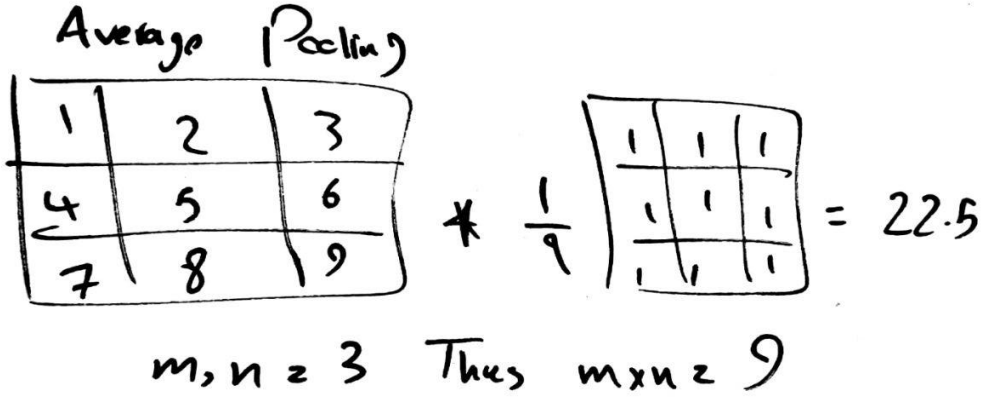
تفاوت average pooling & max pooling :

مشخصا چنانچه تفاوت بین پیکسل های زیاد نباشد هر دو pooling نتیجه تقریبا یکسانی در خروجی خواهند داد ولی در غیر این صورت خروجی دو نوع pooling با هم متفاوت است چرا که ادغام میانگین از آنجا که به نوعی تصویر را هموار (smooth) می کند ممکن است باعث از بین رفتن برخی ویژگی های برجسته یا به اصطلاح sharp شود اما ادغام حداکثری روشن ترین پیکسل را انتخاب می کند و این برای زمانی که بک گراند عکس تاریک است و ما اهمیت بیشتری برای پیکسل های روشن قایلیم مناسب تر است (به طور فرض در مجموعه داده MNIST) همانطور که در شکل زیر مشاهده می شود ادغام حداکثری عمکرد بهتری در هنگامی که بک گراند تیره است دارد و عیضا ادغام حداقلی نیز عملکرد بهتری در هنگامی که بک گراند سفید است دارد ؛ اما در کل می توان گفت که ادغام حداکثری برای مواقعی که نیاز به ویژگی های برجسته (مثل لبه یابی) داریم بسیار عملکرد بهتری دارد



از طرفی دیگر اما می توان گفت چنانچه مکان اشیا در تصویر مهم باشد نه لبه ها ، average pooling عملکرد بهتری خواهد داشت چرا که در این حالت همه پیکسل ها در نقشه ویژگی تاثیری بر نماینده خود دارند و نماینده شان میانگین و به اصطلاح ترکیب آن هاست .

در مورد ادغام میانگین می توان گفت از آنجا که خروجی ترکیب خطی ورودی هاست می توان با استفاده از کانولوشن (شبکه هم گشتی) آن را پیاده سازی کرد و کافی است مقدار درایه های کرنل کانولوشن برابر با 1 / n * m (n و m ابعاد کرنل کانولوشن هستند) قرار دهیم .



در مورد ادغام حداکثری از انجا که با ترکیب خطی ورودی ها روبه رو نیستیم ، نمی توانیم آن را با استفاده از کانولوشن پیاده سازی کنیم .

منابع :

(2) Why do we use the average-pooling layers instead of max- pooling layers in the densenet (machine learning, neural networks, statistics)? - Quora

machine-learning-articles/what-are-max-pooling-average-pooling-global-max-pooling-and-global-average-pooling.md at main · christianversloot/machine-learning-articles · GitHub

(2) What is the benefit of using average pooling rather than max pooling? - Quora

Maxpooling vs minpooling vs average pooling | by Madhushree Basavarajaiah | Medium

Problem 3

محسابه شکل خورجی در لایه هم گشتی به صورت زیر انجام می شود :

$$\text{Floor}\left(\frac{(W-F+2P)}{S}+1\right)$$

ابعاد ورودی = W

سایز کرنل کانولوشن = F

سایز پدینگ = P

سایز قدم یا stride = S

همچنین عمق لایه خروجی برابر تعداد فیلتر های کانولوشن می باشد .

بنابر این خروجی لایه اول کانلوشن به شکل زیر خواهد بود :

$$\text{Floor}\left(\frac{128-5+0}{2}+1\right)=62$$

Thus output shape = (62 ,62 ,64)

برای لایه دوم :

$$Floor\left(\frac{62-3+0}{2}+1\right)=30$$

Thus output shape = (30 ,30 ,64)

خروجی لایه pooling هم از فرمول زیر به دست می آید :

$$Floor\left(\frac{(W-F)}{S}+1\right)$$

ابعداد ورودی = w

سایز کرنل pooling = F

سایز قدم یا stride = S

بنابراین خروجی لایه pooling به شکل زیر خواهد بود ، لازم به ذکر است عمق ورودی بدون تغییر خواهد بود

Stirde در این سوال اشاره نشده است که چند است ولی به صورت دیفالت بنده آن را برابر 3 در نظر گرفتم.

$$Floor\left(\frac{30-3+0}{3}+1\right)=10$$

Thus output shape = (10 ,10 ,64)

نحوه محاسبه پارامتر ها در لایه کانولوشن :

```
param_number = output_channel_number * (input_channel_number * kernel_height * kernel_width + 1)
```

در این صورت تعداد پارامتر های لایه اول برابر خواهد بود با :

$$Parms = 64 * (1 * 5 * 5 + 1) = 65 * 26 = 1664$$

پارامتر های لایه دوم برابر خواهد بود با :

$$Params = 64 *(64 * 3 * 3 + 1) = 64 * 577 = 36928$$

لازم به ذکر است لایه pooling فاقد پارامتر است .

بنابراین جمع کل پارامتر ها برابر است با :

$$36928 + 1664 = 38592$$

به طور خلاصه حاصل به شکل زیر خواهد بود

layer	output	params
Conv1	(62, 62 , 64)	1664
Conv2	(30 , 30 , 64)	36928
Max-Pool	(10 , 10 , 64)	0

کد مربوط به این سوال در محیط کولب هم پیاده سازی شد جهت اطمینان که نتیجه به شکل زیر است :

```
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Conv2D , MaxPooling2D , Dense
model = Sequential()
model.add(Conv2D(filters = 64 , kernel_size =(5,5),strides =2  , input_shape = ( 128 , 128 , 1)))
model.add(Conv2D(filters = 64 , kernel_size =( 3,3),strides =2 ))
model.add(MaxPooling2D( pool_size=(3, 3), strides=3, padding="valid"))
model.summary()
```

```
Model: "sequential_45"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_83 (Conv2D)          (None, 62, 62, 64)        1664

 conv2d_84 (Conv2D)          (None, 30, 30, 64)        36928

 max_pooling2d_36 (MaxPoolin  (None, 10, 10, 64)       0
 g2D)

=================================================================
Total params: 38,592
Trainable params: 38,592
Non-trainable params: 0
_________________________________________________________________
```

منابع :

[How to Calculate the Number of Parameters in Keras Models | by Yong Cui | Towards Data Science]

https://androidkt.com/calculate-output-size-convolutional-pooling-layers-cnn/

Problem 4

خروجی کانولوشن از طریق فرمول زیر محاسبه می شود :

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$
$$O_{12} = X_{12}F_{11} + X_{13}F_{12} + X_{22}F_{21} + X_{23}F_{22}$$
$$O_{21} = X_{21}F_{11} + X_{22}F_{12} + X_{31}F_{21} + X_{32}F_{22}$$
$$O_{22} = X_{22}F_{11} + X_{23}F_{12} + X_{32}F_{21} + X_{33}F_{22}$$

O11 = (3 * 2) + (4 * 0) + (2 * -3) + (1 * 1) = 1

O12 =(4 * 2)+ (5 * 0) + (1 * -3)+ (1 * -3)= 2

O21 = (2 * 2)+ (1 * 0) + (4 * -3) + (-2 * 1)= -10

O22 = (1 * 2) + (0 * -3) + (-3 * -2) +(0 *1) = 8

1	2
-10	8

حال GAP را اعمال میکنیم که حاصل به شکل زیر می شود :

OUTPUT = (1 + 2 + 8 -10) /4 = 0.25

حال دوباره به این فرمول نگاهی بیاندازیم

$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22}$$

مشتق خروجی نسبت به کانولوشن به شرح زیر است :

$$\frac{\partial O_{11}}{\partial F_{11}} = X_{11} \quad \frac{\partial O_{11}}{\partial F_{12}} = X_{12} \quad \frac{\partial O_{11}}{\partial F_{21}} = X_{21} \quad \frac{\partial O_{11}}{\partial F_{22}} = X_{22}$$

رابطه مشتق زنجیری تابع ضرر نسبت به آپدیت کردن فیلتر F به شکل زیر است :

$$\frac{\partial L}{\partial F} = \frac{\partial L}{\partial O} * \frac{\partial O}{\partial F}$$

Gradient to update Filter F

Loss Gradient from previous layer

Local Gradients

که برای درایه های مختلف کرنل به شکل زیر میتوان بسط داد :

$$\frac{\partial L}{\partial F_{11}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{11}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{11}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{11}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{11}}$$

$$\frac{\partial L}{\partial F_{12}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{12}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{12}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{12}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{12}}$$

$$\frac{\partial L}{\partial F_{21}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{21}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{21}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{21}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{21}}$$

$$\frac{\partial L}{\partial F_{22}} = \frac{\partial L}{\partial O_{11}} * \frac{\partial O_{11}}{\partial F_{22}} + \frac{\partial L}{\partial O_{12}} * \frac{\partial O_{12}}{\partial F_{22}} + \frac{\partial L}{\partial O_{21}} * \frac{\partial O_{21}}{\partial F_{22}} + \frac{\partial L}{\partial O_{22}} * \frac{\partial O_{22}}{\partial F_{22}}$$

مشتق خروجی نسبت به کانولوشن را که در ابتدای این سوال حساب کردیم در فرمول بالا جایگزین می کنیم .

$$\frac{\partial L}{\partial F_{11}} = \frac{\partial L}{\partial O_{11}} * X_{11} + \frac{\partial L}{\partial O_{12}} * X_{12} + \frac{\partial L}{\partial O_{21}} * X_{21} + \frac{\partial L}{\partial O_{22}} * X_{22}$$

$$\frac{\partial L}{\partial F_{12}} = \frac{\partial L}{\partial O_{11}} * X_{12} + \frac{\partial L}{\partial O_{12}} * X_{13} + \frac{\partial L}{\partial O_{21}} * X_{22} + \frac{\partial L}{\partial O_{22}} * X_{23}$$

$$\frac{\partial L}{\partial F_{21}} = \frac{\partial L}{\partial O_{11}} * X_{21} + \frac{\partial L}{\partial O_{12}} * X_{22} + \frac{\partial L}{\partial O_{21}} * X_{31} + \frac{\partial L}{\partial O_{22}} * X_{32}$$

$$\frac{\partial L}{\partial F_{22}} = \frac{\partial L}{\partial O_{11}} * X_{22} + \frac{\partial L}{\partial O_{12}} * X_{23} + \frac{\partial L}{\partial O_{21}} * X_{32} + \frac{\partial L}{\partial O_{22}} * X_{33}$$

که فرمول بالا را به شکل زیر می توان باز نویسی کرد

$$\begin{bmatrix} \frac{\partial L}{\partial F_{11}} & \frac{\partial L}{\partial F_{12}} \\ \frac{\partial L}{\partial F_{21}} & \frac{\partial L}{\partial F_{22}} \end{bmatrix} = \text{Convolution} \left(\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix} , \begin{bmatrix} \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} \\ \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} \end{bmatrix} \right)$$

where

$$\begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix} = \text{Input X} \qquad \begin{bmatrix} \frac{\partial L}{\partial O_{11}} & \frac{\partial L}{\partial O_{12}} \\ \frac{\partial L}{\partial O_{21}} & \frac{\partial L}{\partial O_{22}} \end{bmatrix} = \frac{\partial L}{\partial O} \text{ Loss gradient from previous layer}$$

اما در مورد محاسبه مشتق تابع ضرر نسبت به خروجی می توان گفت از آنجا که از GAP استفاده کرده ایم و مشتق تابع اتلاف نسبت به خروجی نهایی برابر ۱ است ، خروجی کل به شکل زیر به دست آمده است .

$$O_{total} = (O_{11} + O_{12} + O_{21} + O_{22})/4$$

$$\Rightarrow \frac{\partial L}{\partial O_{11}} = \frac{1}{4}, \frac{\partial L}{\partial O_{12}} = \frac{1}{4}, \frac{\partial L}{\partial O_{21}} = \frac{1}{4}, \frac{\partial L}{\partial O_{22}} = \frac{1}{4}$$

که به صورت ماتریسی به شکل زیر می شود :

0.25	0.25
0.25	0.25

و در نهایت برای به دست آوردن گرادیان لایه هم گشتی باید ماتریس پایین را در ماتریس بالا کانوالو کنیم .

۳	۴	۵
۲	۱	-۳
۴	-۲	۰

که حاصل برابر است با :

$$\frac{\partial L}{\partial F_{11}} = 3\times\frac{1}{4} + 4\times\frac{1}{4} + 2\times\frac{1}{4} + 1\times\frac{1}{4} = 2.5$$

$$\frac{\partial L}{\partial F_{12}} = 4\times\frac{1}{4} + 5\times\frac{1}{4} + 1\times\frac{1}{4} - 3\times\frac{1}{4} = 1.75$$

$$\frac{\partial L}{\partial F_{21}} = 2\times\frac{1}{4} + 1\times\frac{1}{4} + 3\times\frac{1}{4} - 2\times\frac{1}{4} = 1.25$$

$$\frac{\partial L}{\partial F_{22}} = 1\times\frac{1}{4} + 0\times\frac{1}{4} - 3\times\frac{1}{4} - 2\times\frac{1}{4} = -1$$

که به صورت ماتریسی به شکل زیر است :

2.5	1.75
1.25	-1

منبع :

لینک موجود در صورت سوال