

به نام خدا



دانشکده مهندسی کامپیوتر

تمرینات درس یادگیری عمیق

پاسخنامه تمرین سری ششم

دکتر محمدرضا محمدی

دانشجو: مجتبی نافذ

96431335

نیمسال دوم ۹۹-۱۰۰

۱. استفاده از dropout در چه شرایطی پیشنهاد می‌شود؟ بر چه اساس پارامتر نگهداری نورون‌ها مقداردهی می‌شود؟

زمانی که دچار **overfitting** می‌شویم استفاده از آن پیشنهاد می‌شود و در کنار این هدف **robustness** مدل ما را افزایش

میدهد و باعث میشود با از دست دادن چند نورون شبکه از کار نیفتد.

اما پارامتر dropout بر اساس میزان **overfitting** باید مقداردهی شود اگر **overfitting** ما زیاد باشد یعنی اختلاف دقت داده

های **train** و **test** ما زیاد باشد باید مقدار این پارامتر را بیشتر در نظر بگیریم تا دچار **overfitting** نشویم.

معمولا در لایه های **hidden** این پارامتر را حدود ۰.۵ مقداردهی می‌کنیم و در بقیه ی لایه ها بیشتر هم امکان دارد قرار دهیم

چیزی حدود ۰.۸

در تعیین این پارامتر تست کردن نکته ی مهمی است.

۲. الف) تفاوت میان لایه‌های Fully Connected، Locally Connected و Convolutional را به طور دقیق

توضیح دهید. هر کدام از این لایه‌ها برای چه شرایطی کاربردی‌تر هستند؟

ب) استفاده از Stride در شبکه‌های کانولوشنی دارای چه مزایا و معایبی است؟ توضیح دهید.

لایه ی Fully Connected :

همان لایه ی **dense** است در که در آن هر لایه به طور کامل به لایه بعد متصل است (شبکه های **MLP** خیلی از لایه های

dense استفاده میکنند)

در شرایطی که همسایگی ها معنای خاصی ندارد. و ورودی ها با هم دیگر ارتباط معناداری ندارند. مثلا مسئله خرید آپارتمان که

به ما گروهی ویژگی های ساختمان مانند متراژ و محل و .. را میدهند و هدف تخمین قیمت آن است.

لایه ی Locally Connected :

در این نوع لایه ها به جای اینکه هر نورون مانند لایه های **fully connected** به تمام نورون های لایه ی قبل متصل باشد فقط به

تعدادی از نورون های کنار هم و همسایه متصل میشود که یه صورت **pattern** های محلی را می آموزد. وزن های هر نورون ویژه

ی آن نورون بود و وزن ها اشتراکی نیستند.

در شرایطی که همسایگی ها معنی دار باشند و علاوه بر این خروجی با مکان و مختصات ورودی رابطه داشته باشد.

به طور نمونه فرض کنید یک دیتاست از تصاویر برش خورده ی هم سائز از چهره ی افراد داشته باشیم که بخواهیم روی آن

شبکه دسته بندی رنگ چشم بین چند چهره را انجام دهیم در این دیتاست چشم افراد همواره در یک منطقه از تصویر است و

نیازی اشتراک وزن ها نداریم و اشتراک وزن سودی برای ما ندارد و اتفاقا معایبی هم دارد.

لایه ی Convolutional :

در این نوع لایه ها به مانند لایه های **locally connected** اتصالات محلی است و کاملاً مشابه **locally connected** می باشد

با یک تفاوت بزرگ و آن هم در مورد اشتراک وزن ها،

در این مدل وزن بین اتصالات محلی مشترک است.

مناسب برای شرایطی است که مکان برای ما مهم نباشد به طور مثال دنبال تصاویر گربه هستیم حال گربه می تواند بالای تصویر

سمت چپ باشد یا پایین تصویر سمت راست. در هر حالت گربه در تصویر وجود دارد. و برای ما فرقی نمیکند.

مزایای **Stride** در شبکه ها کانولوشنی : کاهش هزینه ی محاسبات، **respective field** را افزایش داده ایم یعنی الان هر نورون ما

از تعداد بیشتری ورودی تاثیر پذیرفته در نتیجه پترن های بهتری آموزش خواهد دید.

معایب **Stride** در شبکه های کانولوشنی: همه ی ویژگی ها به خوبی **extract** نمی شوند. چون بعضی داده ها حذف شده و

downsampling صورت گرفته.

۳. تعداد پارامترهای قابل آموزش شبکه زیر را محاسبه کنید (جزئیات محاسبات را به صورت دقیق و کامل بنویسید، اما نیازی به ساده‌سازی پاسخ نیست).

```
model = keras.Sequential()
model.add(keras.layers.Input(shape=(28, 28, 1)))
model.add(keras.layers.Conv2D(filters=20, kernel_size=(5, 5), activation='relu'))
model.add(keras.layers.MaxPool2D(pool_size=(2, 2), strides=(2, 2)))
model.add(keras.layers.Conv2D(filters=10, kernel_size=(5, 5), activation='relu'))
model.add(keras.layers.LocallyConnected2D(filters=2, kernel_size=(5, 5), activation='relu'))
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(units=10, activation='softmax'))
model.summary()
```

layers	parameters
Conv2D	$(5 \times 5 + 1) \times 20 = 520$ ۲۰ نورون داریم که اتصالات هر کدام یک کرنل را تشکیل میدهد و هر نورون یک بایاس هم دارد. هر نورون یک خروجی 28×28 هم دارد که با حساب ۲۰ نورون خروجی $28 \times 28 \times 20$ بعد دارد.
MaxPooling2D	این لایه پارامتری نداشته و صرفاً یک کار آماری است. خروجی در نهایت از بعد $28 \times 28 \times 20$ می باشد.
Conv2D	$(5 \times 5 \times 20 + 1) \times 10 = 5010$ در واقع ما ۱۰ نورون داریم که هر نورون یک کرنل 5×5 دارد که برای اینکه بتواند ورودی را کاور کند باید با بعد آن هم هم گام شود و در نتیجه باید به یک قسمت 5×5 از تمام ۲۰ نورون لایه ی قبلی اتصال پیدا کند و همچنین هر نورون یک بایاس هم دارد. در نتیجه خروجی باید از ابعاد $10 \times 8 \times 8$ باشد.
LocallyConnected2D	$(5 \times 5 \times 10 + 1) \times 2 \times 16 = 8032$ دو نورون داریم که هر نورون خروجی 4×4 دارد که در نتیجه ۱۶ خانه خروجی دارد و به ازای همه ی خروجی هایش پارامترها مشترک نبوده و متفاوت هستند و هر نورون یک کرنل 5×5 دارد که باید ۱۰ مرتبه تکرار شود تا با ورودی قابل ضرب شدن باشد و هر نورون یک بایاس دارد.
Flatten	این لایه فقط بعد را تغییر میدهد و پارامتر جدیدی تولید نمی کند.
Dense	در لایه ی آخر یک لایه ی کاملاً متصل با تعداد ۱۰ نورون می باشد و چون لایه ی قبلی ۳۲ نورون داشت و هر نورون لایه آخر هم یک بایاس دارد : $32 \times 10 + 10 = 330$

در مجموع 13892 پارامتر قابل یادگیری در مدل وجود دارد.

۴. در این سوال می‌خواهیم یک مسئله دسته‌بندی را بر روی مجموعه داده سگ و گربه و با استفاده از تکنیک‌های داده‌افزایی و dropout انجام دهیم.

معمولاً برای آموزش مدل‌های یادگیری عمیق، از مجموعه داده‌های نسبتاً بزرگی استفاده می‌شود که حتی پیشرفته‌ترین سخت‌افزارها، حافظه کافی برای پردازش داده‌ها به صورت یکجا و یکپارچه را ندارند. به همین دلیل است که ما باید راه‌های دیگری برای انجام کارآمد آن پیدا کنیم. در ادامه قصد داریم به شما نشان دهیم که چگونه مجموعه داده را در چندین هسته و در زمان اجرا تولید کرده و بلافاصله آن را به مدل یادگیری عمیق خود بدهیم. بدین منظور کلاسی به نام ImageDataGenerator در keras پیاده‌سازی شده است که همزمان با تولید یک batch از تصاویر در زمان اجرا، می‌تواند اعمال مختلف داده‌افزایی مانند rotation، flipping و ... را انجام دهد.

برای آشنایی بیشتر با ImageDataGenerator و نحوه استفاده از آن برای داده‌افزایی، از مثال موجود در این [لینک](#) استفاده نمایید.

الف) در ابتدا یک شبکه کانولوشنی شامل سه لایه کانولوشن و دو لایه Dense را به عنوان مدل تعریف کرده و سپس با داده‌های موجود بدون داده‌افزایی مدل را ارزیابی کنید.

ب) در این قسمت شما باید ابتدا داده‌افزایی انجام داده و سپس مدل را آموزش داده و آن را ارزیابی کنید.

پ) dropout با نرخ مناسب را به شبکه اضافه کرده و نتایج را با بخش‌های قبل مقایسه کنید.