

سوال یک

(الف)

۱. **Overfitting**: استفاده از نرخ یادگیری بسیار بالا می‌تواند منجر به بیش‌برازش مدل شود. که به داد های آموزشی دقت بالایی دارد ولی در داده های تست خوب عمل نمی‌کند.

۲. عدم همگرایی: نرخ یادگیری بسیار بالا ممکن است باعث عدم همگرایی مدل شود. به عبارت دیگر، مدل ممکن است نتواند به طور کامل یاد بگیرد و هی مدام در حال تغییر باشد.

۳. پرس شدن از مینی‌مم محلی: با استفاده از نرخ یادگیری بالا، ممکن است مدل از مینی‌مم محلی خارج شود و به جای آن در جستجوی نقاط بهینه دیگر به سر ببرد. این می‌تواند باعث بهبود کیفیت مدل شود، اما همزمان ممکن است منجر به عدم پیدا کردن بهینه‌ی واقعی شود.

برای تشخیص مشکلات ناشی از استفاده از نرخ یادگیری بالا، می‌توانید به روش‌های زیر رجوع کنید:

۱. نمودار تابع هزینه: بررسی تغییرات تابع هزینه در طول زمان آموزش می‌تواند نشان دهنده وجود مشکلات باشد. اگر تابع هزینه به سرعت کاهش یابد و سپس به طور ناگهانی افزایش یابد، احتمالاً نرخ یادگیری بیش از حد بالاست.

عملکرد در داده‌های تست: اگر رو داده های آموزشی دقت خوبی داریم ولی رو داده های تست نه ممکن یکی از دلایل نرخ یادگیری باشد.

تحلیل نمودار دقت: بررسی نمودار دقت مدل در طول زمان آموزش می‌تواند راهنمایی کننده باشد. اگر دقت در داده‌های آموزشی به سرعت افزایش یابد ولی دقت در داده‌های ارزیابی یا داده‌های تست به شدت کاهش یابد، نشانه‌ای از وجود مشکل با نرخ یادگیری بالا است.

(ب)

استفاده از نرخ یادگیری بسیار پایین می‌تونه باعث **underfitting** بشه و هم چنین می‌تونه باعث مشکلات زیر نیز بشود:

۱. عدم تطبیق کافی با داده‌های آموزشی: اگر نرخ یادگیری بسیار پایین باشد، مدل نمی‌تواند به طور کامل با داده‌های آموزشی تطبیق پیدا کند. این موجب می‌شود تا مدل به طور ناکافی آموزش ببیند و نتواند دقت قابل قبولی بر روی داده‌های آموزشی داشته باشد.

۲. عملکرد ضعیف در داده‌های آموزشی و تست: استفاده از نرخ یادگیری بسیار پایین می‌تواند منجر به عملکرد ضعیف در داده‌های آموزشی و همچنین داده‌های تست شود. مدل نمی‌تواند به طور کافی یاد بگیرد و از توانایی تعمیم به داده‌های جدید کاسته می‌شود.

۳. سرعت همگرایی بسیار پایین

برای تشخیص مشکلات ناشی از استفاده از نرخ یادگیری بسیار پایین، می‌توانید به روش‌های زیر رجوع کنید:

۱. نمودار تابع هزینه: اگر تابع هزینه مقدارش در داده‌های آموزشی زیاد است و به کندی کاهش میابد یکی از دلایلش میتواند نرخ یادگیری پایین باشد.

۲. تحلیل نمودار دقت: بررسی نمودار دقت مدل در طول زمان آموزش می‌تواند مفید باشد. اگر دقت در داده‌های آموزشی به سرعت افزایش یافته و سپس به حالت ماندگاری رسید یا به حالتی کاهش یافته، ممکن است نرخ یادگیری بسیار پایین عاملی در این مشکل باشد.

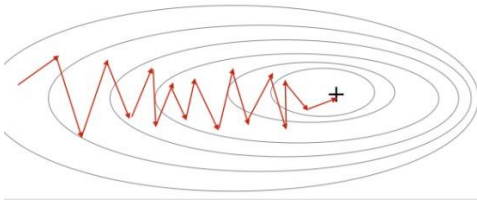
(پ)

نقطه زینی یا مینی‌مم محلی به معنای نقطه‌ای است که نسبت به نقاط اطاف خود کمینه هست. در مسائل بهینه‌سازی، یافتن نقاط زینی می‌تواند مشکلاتی را ایجاد کند زیرا ممکن است با استفاده از الگوریتم بهینه‌سازی مناسب، به جای کمینه‌ی سراسری، در نقاط زینی محلی گیر کنیم و نتوانیم به جواب بهینه دست پیدا کنیم.

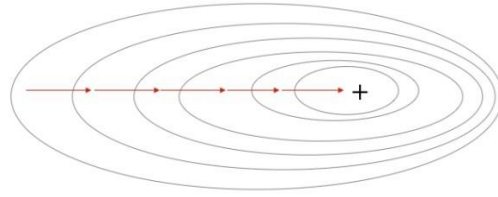
(ت)

نمودار سمت چپ یک نمودار smooth هست و نمودار سمت راست یک نمودار نویزی هست وقتی از minibatch استفاده می‌کنیم نمودار تابع ضرر ممکن هست نویزی بشه ولی در نهایت در مسیر درستی پیش میریم به این دلیل نویزی میشه که دیتاست را به batch های ک.چک تقسیم میکنیم و برای هر کدام گرادیان حساب می‌کنیم که این نسبت به حالتی برای کل دیتاست گرادیان حساب میکنیم کمی باعث نویز میشود به همین علت نمودار سمت راست متعلق به minibatch هست. شکل پایین دید خوبی از این قضیه میدهد.

Stochastic Gradient Descent



Gradient Descent



سوال دو

پاسخ این سوال در صفحه بعد نوشته شده است.

تابع هدف $S = 0_{11} + 0_{12} + 0_{21} + 0_{22}$ تابع هدف

$$\frac{\partial L}{\partial S} = 1, \quad \frac{\partial L}{\partial 0_{11}} = \frac{\partial L}{\partial S} \cdot \frac{\partial S}{\partial 0_{11}} = 1 \times \frac{1}{4}$$

در این مرحله ضرایب $\frac{\partial L}{\partial 0_{ij}}$ است در این مرحله ضرایب

$$\frac{\partial L}{\partial F_{11}} = \frac{1}{4} (3+4+2+1) = \frac{10}{4}, \quad \frac{\partial L}{\partial F_{12}} = \frac{1}{4} (4+5+1-3) = \frac{7}{4}$$

$$\frac{\partial L}{\partial F_{21}} = \frac{1}{4} (2+1+4-2) = \frac{5}{4}, \quad \frac{\partial L}{\partial F_{22}} = \frac{1}{4} (1-3-2) = \frac{-4}{4} = -1$$

$$\frac{\partial L}{\partial x_i} = \sum_{k=1}^M \frac{\partial L}{\partial 0_{ik}} \cdot \frac{\partial 0_{ik}}{\partial x_i} \quad \text{مشتق}$$

$$\frac{\partial L}{\partial x_{11}} = \frac{\partial L}{\partial 0_{11}} \cdot F_{11} = 1 \times 2 = 2 \quad \frac{\partial L}{\partial x_{12}} = \frac{\partial L}{\partial 0_{12}} \cdot F_{12} + \frac{\partial L}{\partial 0_{12}} \cdot F_{11} = \frac{1}{4} (2+0) = \frac{2}{4}$$

$$\frac{\partial L}{\partial x_{13}} = \frac{\partial L}{\partial 0_{12}} \cdot F_{12} = 0 \times \frac{1}{4} = 0 \quad \frac{\partial L}{\partial x_{21}} = \frac{\partial L}{\partial 0_{11}} \cdot -3 + \frac{\partial L}{\partial 0_{21}} \cdot 2 = \frac{1}{4} (-3+2) = \frac{-1}{4}$$

$$\frac{\partial L}{\partial x_{22}} = \frac{1}{4} (1 + -3 + 0 + 2) = 0 \quad \frac{\partial L}{\partial x_{23}} = \frac{1}{4} (1+0) = \frac{1}{4}$$

$$\frac{\partial L}{\partial x_{31}} = \frac{1}{4} \cdot -3 = \frac{-3}{4} \quad \frac{\partial L}{\partial x_{32}} = \frac{1}{4} (1 + -3) = \frac{-2}{4}$$

$$\frac{\partial L}{\partial x_{33}} = \frac{1}{4} (1) = \frac{1}{4}$$

سوال سه

(الف)

سوال (3):

① Conv1 (16, 3, activation=relu)

← تعداد باسها در این لایه:

$$16 \times (3 \times 7 + 1) = 352$$

② maxpool1D - در این لایه تعداد باسها 2 است. (16, 249)

③ Conv1D (32, 5, relu)

← تعداد باسها در این لایه:

$$(245, 32)$$

$$32 \times (5 \times 16 + 1) = 2592$$

④ maxpool1D -

تعداد باسها:

$$(122, 32)$$

سایر:

⑤ Conv1D (filter=64, 5, relu)

$$(118, 64)$$

$$64 \times (32 \times 5 + 1) = 10304$$

تعداد باسها:

⑥ MaxPool1D:

$$(59, 64)$$

⑦ Flatten:

$$(3776, 1)$$

⑧ Dense (128, relu)

$$128 \times (3776 + 1) = 483456$$

⑨ Dense (5, relu)

$$5 \times (128 + 1) = 645$$

$$352 + 2592 + 10304 + 483456 + 645 = 497349$$

(ب)

لایه همگشتی دو بعدی و لایه همگشتی سه بعدی هر دو از لایه‌های همگشتی در شبکه‌های عصبی هستند، اما تفاوت اصلی بین آنها در ابعاد ورودی و عملکرد آنها است.

لایه همگشتی دو بعدی:

- ورودی و خروجی این لایه دارای ابعاد دو بعدی است، به عنوان مثال (ارتفاع، عرض).
- این لایه از فیلترهای دو بعدی برای استخراج ویژگی‌های مکانی از تصاویر استفاده می‌کند.
- در برنامه‌های بینایی ماشین، معمولاً در شبکه‌های پیچشی (Convolutional Neural Networks) استفاده می‌شود و به خوبی برای تشخیص الگوها و ویژگی‌های مکانی در تصاویر عمل می‌کند.

لایه همگشتی سه بعدی :

- ورودی و خروجی این لایه دارای ابعاد سه بعدی است، به عنوان مثال (عمق، ارتفاع، عرض).
 - این لایه برای استخراج ویژگی‌های مکانی و زمانی در داده‌های سه بعدی مانند ویدیوها، سیگنال‌های صوتی و داده‌های مکعبی (volumetric data) استفاده می‌شود.
 - با استفاده از فیلترهای سه بعدی، داده‌های سه بعدی را در سه جهت (عمق، ارتفاع، عرض) همگشتی می‌کند و ویژگی‌های مکانی و زمانی را استخراج می‌کند.
 - در برنامه‌های مربوط به پردازش سیگنال‌های صوتی، تحلیل و پردازش ویدیوها، تشخیص حرکت در ویدیوها و پردازش داده‌های سه بعدی مانند تصاویر MRI و CT استفاده می‌شود.
- بنابراین، لایه همگشتی دو بعدی برای استخراج ویژگی‌های مکانی از تصاویر دو بعدی مناسب است، در حالی که لایه همگشتی سه بعدی برای استخراج ویژگی‌های مکانی و زمانی از داده‌های سه بعدی مناسب است.

سوال چهار

در ابتدا با اسفاده از کد زیر train, test جدا میکنیم و با استفاده از imagedataloader به صورت بچ بچ در می‌آوریم:


```

# Set the path to your dataset directory
dataset_dir = '/content/dataset'

# Create an instance of the ImageDataGenerator
datagen = ImageDataGenerator(validation_split=0.2)

# Load and preprocess the training data
train_generator = datagen.flow_from_directory(
    dataset_dir,
    subset='training',
    class_mode='categorical',
    batch_size=32,
    target_size=(224, 224)
)

# Load and preprocess the validation data
validation_generator = datagen.flow_from_directory(
    dataset_dir,
    subset='validation',
    class_mode='categorical',
    batch_size=32,
    target_size=(224, 224)
)

```

و سپس با استفاده از مدلی که به صورت زیر تعریف کردیم دیتاست را train میکنیم:

```

model = Sequential()

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(224, 224, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())

model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(2, activation='softmax'))

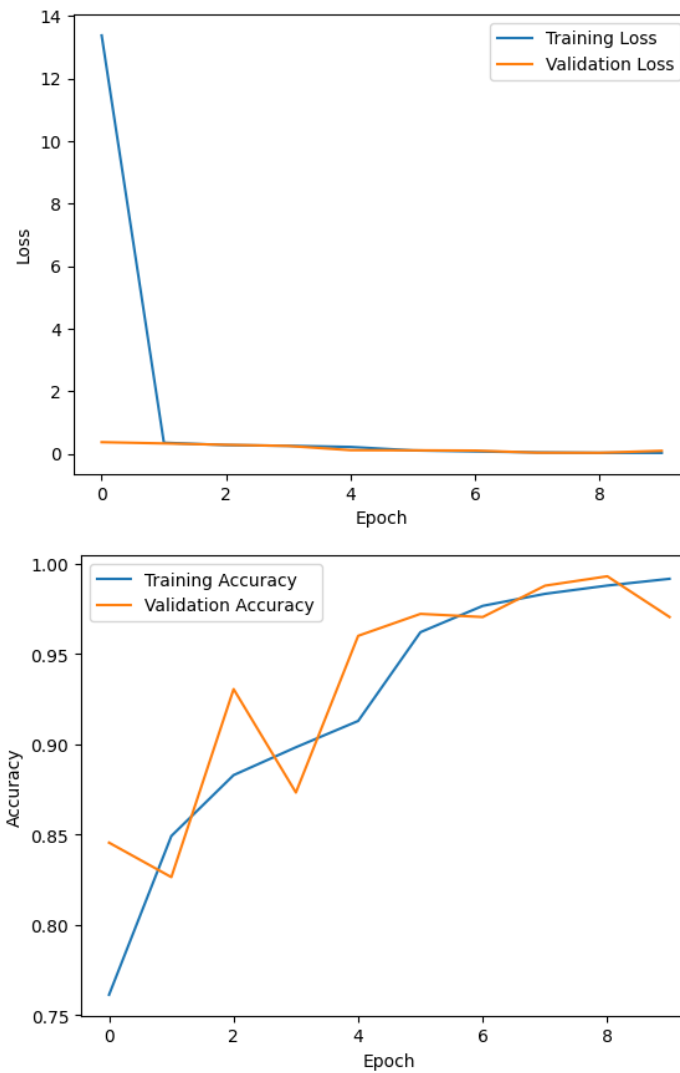
```

نتایج برای sequential بعد از پایان train رو داده های تست به صورت زیر شد:

دقت: 0.9716666936874 0.9716666936874 0.971666693687439439

0.971666693687439

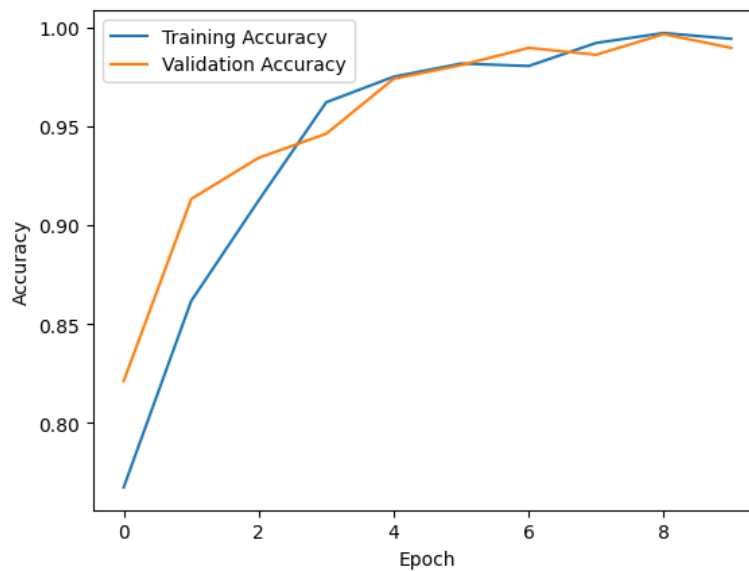
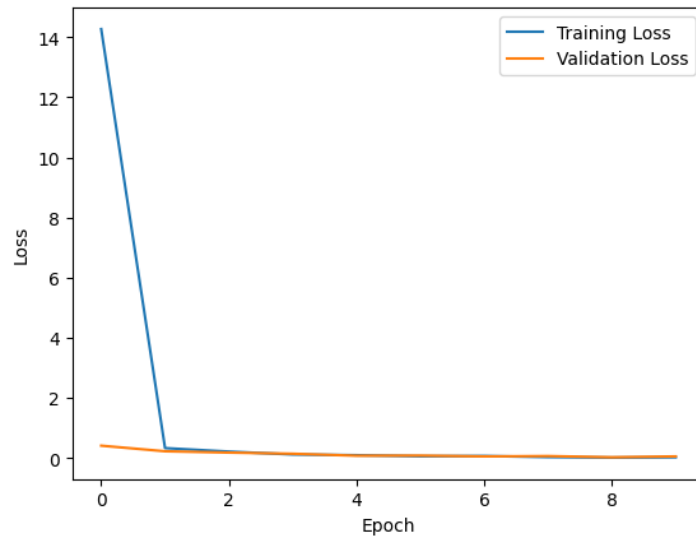
نمودار دقت و تابع ضرر:



و سپس مدل را به صورت functional پیاده سازی کردیم که نتایج آن به صورت زیر شد:

دقت: 0.9900000095367432

و نمودارها به صورت زیر شد:



تفاوت کمی که در نتایج sequential و functional هست این هست که ما داده ها را به صورت بچ میدهم و به خاطر randomness کمی نتایج متفاوت است.

سوال پنج

فرض کنید می خواهیم یک مدل دسته بندی تصاویر بسازیم که توانایی تشخیص اشیاء در تصاویر را داشته باشد. برای این کار، می توانیم از لایه های همگشتی استفاده کنیم. این لایه ها به خاطر ویژگی های خاص خود، مزایایی در دسته بندی تصاویر ایجاد می کنند.

۱. تشخیص ویژگی‌های محلی: لایه‌های همگشتی قادر به تشخیص ویژگی‌های محلی در تصاویر هستند. به عنوان مثال، لایه‌های اولیه می‌توانند لبه‌ها، خطوط و نقاط را تشخیص دهند. لایه‌های عمیق‌تر می‌توانند ویژگی‌های بیشتری را شناسایی کنند، مانند شکل‌ها، الگوها و اشیاء مخصوصاً در مسائل ناحیه بندی این تشخیص ویژگی‌های محلی خیلی اهمیت دارد.

۲. مقاومت در برابر تغییرات محلی: لایه‌های همگشتی مقاومت قابل توجهی در برابر تغییرات محلی در تصاویر دارند. به عنوان مثال، اگر یک شیفت کوچک در تصویر اتفاق بیفتد، لایه‌های همگشتی همچنان قادر به تشخیص ویژگی‌ها و الگوها خواهند بود.

۳. اشتراک پارامتر: لایه‌های همگشتی از اشتراک پارامتر بهره می‌برند. این به معنای استفاده مجدد از وزن‌ها در سراسر تصاویر است که به کاهش تعداد پارامترها و افزایش کارایی مدل منجر می‌شود.

۴. کاهش تعداد پارامترها

به طور معکوس، یک حالت تصور کنید که ویژگی‌های لایه‌های همگشتی چالش‌ها ایجاد می‌کنند:

۱. نیاز به محاسبات سنگین: لایه‌های همگشتی نیازمند محاسبات سنگین هستند، به ویژه در مواقعی که تصاویر بزرگ را پردازش می‌کنند. این موضوع می‌تواند باعث افزایش زمان آموزش و پیش‌بینی مدل شود.

۲. نیاز به داده برچسب‌دار بزرگ: لایه‌های همگشتی برای آموزش به داده برچسب‌دار بزرگ نیاز دارند.

۳. مشکل گرادیان ناپدید شونده: در شبکه‌های عصبی عمیق با لایه‌های همگشتی، مشکل گرادیان ناپدید شونده همچنان وجود دارد. این مشکل به این معنی است که گرادیان‌ها در طول زمان به صورت نادرست به لایه‌های همگشتی CNN منتقل می‌شوند. این موضوع می‌تواند باعث کاهش سرعت و کیفیت آموزش شبکه شود و ممکن است باعث بهبود ناچیز یا عدم پیشرفت در آموزش شبکه شود.

در توتبوک q5.ipynb من رو دیتاست mnist یکبار بدون CNN و بار دیگر با CNN دیتاست را train کردم نتایج بدون CNN دقت رو داده‌های تست برابر با 96.67 شد و با CNN برابر با 98.6 شد که دقت افزایش یافت البته mnist دیتاست پیچیده‌های نیست رو دیتاست‌های پیچیده‌تر این CNN خیلی زیاد هست.

سوال شش

(الف)

استفاده از فیلترهای 1×1 در شبکه‌های عصبی هم‌گشتی برای دو هدف اصلی استفاده می‌شود: کاهش تعداد نقشه‌های ویژگی در واقع همان کاهش عمق و حفظ ویژگی‌های مهم.

۱. کاهش تعداد نقشه‌های ویژگی: در شبکه‌های عصبی هم‌گشتی، تعداد نقشه‌های ویژگی مستقل افزایش می‌یابد هرچند که این موضوع می‌تواند باعث افزایش پیچیدگی محاسباتی و مصرف منابع شود. با استفاده از فیلترهای 1×1 می‌توان تعداد نقشه‌های ویژگی را کاهش داد. فیلتر 1×1 در واقع یک عملگر تک‌پیکسلی است که بر روی نقشه‌های ویژگی اعمال می‌شود. با ترکیب وزن‌های مختلف در فیلتر 1×1 با نقشه‌های ویژگی، می‌توان نقشه‌های ویژگی جدیدی تولید کرد که تعداد کانال‌ها را کاهش می‌دهد. این عمل باعث کاهش پیچیدگی محاسباتی و مصرف منابع شبکه می‌شود.

۲. حفظ ویژگی‌های مهم: استفاده از فیلترهای 1×1 به شبکه امکان می‌دهد که ویژگی‌های مهم را در نقشه‌های ویژگی حفظ کند. با ترکیب وزن‌های مختلف در فیلتر 1×1 ، شبکه قادر است ویژگی‌های مهم برای تشخیص الگوها و اشیاء را استخراج کند و در نقشه‌های ویژگی جدید حفظ کند. به عبارت دیگر، این فیلترها به شبکه امکان می‌دهند تا به طور همزمان اطلاعات مکانی و اطلاعات کانال‌های ویژگی را ترکیب کنند تا ویژگی‌های مهم را استخراج کنند و در نقشه‌های ویژگی کمتری نمایش دهند.

(ب)

فیلتر 1×1 پس از اعمال، یک نقشه ویژگی جدید تولید می‌کند. این نقشه ویژگی اطلاعاتی را ارائه می‌دهد که ترکیبی از اطلاعات مکانی و اطلاعات کانال‌های ویژگی قبلی است.

هر پیکسل در نقشه ویژگی جدید، نمایانگر یک ترکیب خطی از ویژگی‌های موجود در نقشه‌های ویژگی قبلی است. وزن‌های مختلف در فیلتر 1×1 برای هر کانال ویژگی مشخص می‌کنند که چه میزان از ویژگی‌ها در نقشه ویژگی جدید نمایان شود. با ترکیب این وزن‌ها و ویژگی‌های موجود، نقشه ویژگی جدید به صورت خطی با ترکیبی از این ویژگی‌ها تولید می‌شود.

اطلاعات موجود در نقشه ویژگی جدید شامل ویژگی‌های مهمی است که توسط فیلتر 1×1 استخراج شده‌اند. این ویژگی‌ها می‌توانند الگوها، اشیاء، و یا ویژگی‌های مهم دیگری در داده‌ها باشند، که به شبکه امکان می‌دهند الگوها را تشخیص دهد و ویژگی‌های مورد نیاز را استخراج کند.

به طور خلاصه، نقشه ویژگی جدید که با استفاده از فیلتر 1×1 تولید می‌شود، اطلاعات ترکیبی از ویژگی‌های موجود در نقشه‌های ویژگی قبلی را ارائه می‌دهد و می‌تواند ویژگی‌های مهم و الگوهای موجود در داده را نمایش دهد.

(پ)

فیلترهای $n \times n$ که n بزرگتر از ۱ است، نقشه ویژگی جدیدی با ابعاد مختلف تولید می‌شود و تفاوت‌های مهمی با فیلتر 1×1 دارد. وقتی از یک فیلتر $n \times n$ استفاده می‌کنیم، فیلتر بر روی نقشه ویژگی اصلی (ورودی شبکه) یا نقشه ویژگی قبلی اعمال می‌شود. این فیلتر با استفاده از یک پنجره مربعی به اندازه $n \times n$ ، ویژگی‌های محلی را در نقشه ویژگی تشخیص می‌دهد و باعث ایجاد تغییرات مکانی در نقشه ویژگی می‌شود. تفاوت اصلی با فیلتر 1×1 این است که فیلتر $n \times n$ قادر به تشخیص الگوهای بزرگتر و پیچیده‌تر است. با افزایش اندازه فیلتر، محدوده مکانی که توسط فیلتر در نظر گرفته می‌شود بزرگتر می‌شود و اطلاعات مکانی گسترده‌تری را در نقشه ویژگی جدید نمایش می‌دهد. به طور کلی، فیلترهای $n \times n$ می‌توانند ویژگی‌های متنوعی را در نقشه ویژگی جدید تشخیص دهند، مانند لبه‌ها، نقاط کلیدی، الگوها و ساختارهای پیچیده‌تر. این فیلترها توانایی استخراج اطلاعات مکانی و ساختاری را بهبود می‌بخشند و به شبکه عصبی امکان می‌دهند برچسب‌ها و ویژگی‌های مهم را با دقت بیشتری تشخیص دهند.

(ت)

مدلهایی که از فیلترهای 1×1 استفاده می‌کنند، عبارتند از:

۱. شبکه عصبی کانولوشنال: در شبکه‌های CNN، لایه‌های کانولوشنال 1×1 برای ترکیب ویژگی‌ها و کاهش ابعاد کانال‌ها (نیز به عنوان کانال داده می‌شود) استفاده می‌شود. این لایه‌ها می‌توانند به شبکه امکان بدهند تعداد کانال‌ها را کاهش دهد یا از ویژگی‌های غیرمهم جدا شده و فقط ویژگی‌های مهم را انتخاب کند.

۲. شبکه عصبی (ResNet) معماری ResNet از بلوک‌های اصلی استفاده می‌کند که شامل لایه‌های کانولوشنال 1×1 و 3×3 است. لایه‌های کانولوشنال 1×1 در این بلوک‌ها برای کاهش ابعاد کانال‌ها و ترکیب ویژگی‌ها استفاده می‌شوند.

۳. شبکه عصبی (Inception Network) معماری Inception نیز از فیلترهای 1×1 در لایه‌های کانولوشنال خود استفاده می‌کند. این فیلترها برای ترکیب ویژگی‌ها و کاهش ابعاد کانال‌ها در هر بلوک استفاده می‌شوند.

(ج)

در نوتبوک q6.ipynb اومدم و یک CNN ساده با استفاده از فیلترهای ۱ در ۱ مطابق شکل زیر پیاده کردم:

```
# Define the CNN model using 1x1 filters
class CNNModel(nn.Module):
    def __init__(self):
        super(CNNModel, self).__init__()
        self.conv1 = nn.Conv2d(1, 16, kernel_size=1)
        self.relu = nn.ReLU()
        self.conv2 = nn.Conv2d(16, 32, kernel_size=1)
        self.flatten = nn.Flatten()
        self.fc = nn.Linear(32 * 28 * 28, 10)
```

بر اساس خروجی چاپ شده، ابعاد تصویر ورودی و خروجی برای مدل CNN با فیلترهای یک در یک به شرح زیر است:

- Conv1: [1, 16, 28, 28] ابعاد پس از لایه

- Conv2: [1, 32, 28, 28] ابعاد پس از لایه

- FC: [1, 10] ابعاد پس از لایه

همچنین، ابعاد تصویر ورودی اصلی از مجموعه داده MNIST به شکل زیر است:

- ابعاد تصویر ورودی [1, 1, 28, 28] :

با مقایسه ابعاد تصویر ورودی و خروجی، می‌توانیم تأثیر فیلترهای ۱ در ۱ را بر روی تغییر ابعاد مشاهده کنیم. در این مدل، با استفاده از لایه Conv1 با ۱۶ فیلتر ۱ در ۱، تعداد کانال‌ها از ۱ به ۱۶ تغییر می‌کند و ابعاد تصویر همان ابعاد ورودی (28x28) باقی می‌ماند. سپس، با استفاده از لایه Conv2 با ۳۲ فیلتر ۱ در ۱، تعداد کانال‌ها از ۱۶ به ۳۲ تغییر می‌کند و ابعاد تصویر نیز بدون تغییر باقی می‌ماند. در نهایت، با استفاده از لایه FC، تصویر به یک بردار با ابعاد ۱۰ تغییر شکل می‌یابد.

با مقایسه ابعاد ورودی و خروجی، می‌توانیم ببینیم که فیلترهای ۱ در ۱ توانستند تعداد کانال‌ها را تغییر دهند ولی ابعاد تصویر را تغییر ندادند.

سوال هفت

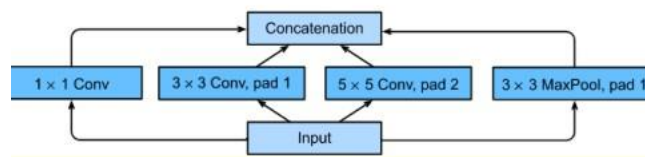
در ابتدا سوال‌ها را پاسخ می‌دهم و در ادامه پیاده‌سازی خودم را توضیح می‌دهم

(الف)

هدف اصلی معماری Inception، استفاده بهینه از اطلاعات موجود در تصاویر است. برای این منظور، از ماژول Inception استفاده می‌شود که شامل لایه‌های پیچشی با اندازه‌های مختلف است. به این ترتیب، ماژول Inception قادر است الگوهای مختلفی را با اندازه‌های مختلف در تصاویر تشخیص دهد.

ساختار Inception شامل لایه‌های پیچشی 1×1 ، 3×3 و 5×5 است. لایه‌های پیچشی 1×1 با هدف کاهش تعداد کانال‌های ورودی، اطلاعات را ترکیب می‌کنند و تعداد پارامترها را کاهش می‌دهند. لایه‌های پیچشی 3×3 و 5×5 ، الگوهای مختلف را در تصاویر تشخیص می‌دهند. در نهایت، خروجی این لایه‌ها ترکیب شده و به لایه‌های بعدی ارسال می‌شود.

که شکل ای ماژول به صورت زیر است:



(ب)

اندازه پارامتر گام (stride) در لایه‌های هم‌گشتی (convolutional layers) در شبکه‌های عصبی کانولوشنی تأثیر زیادی بر ابعاد فضایی نگاشت و ویژگی‌ها دارد. اندازه گام تعیین می‌کند که فیلترها چه فاصله‌ای از هم در طول و عرض تصویر حرکت کنند. در نتیجه، این پارامتر بر روی ابعاد فضایی نگاشت تأثیر می‌گذارد.

وقتی گام بزرگتر است، فیلترها با فاصله بیشتری حرکت می‌کنند و اندازه نگاشت کوچکتر می‌شود. به عبارت دیگر، با افزایش گام، تعداد پیکسل‌هایی که هر فیلتر را پوشش می‌دهد، کاهش می‌یابد. این می‌تواند منجر به کاهش دقت مدل در تشخیص ویژگی‌ها و اطلاعات مهم در تصویر شود. از طرف دیگر، استفاده از گام بزرگتر می‌تواند به کاهش پیچیدگی محاسباتی و افزایش سرعت آموزش و پیش‌بینی کمک کند.

در مقابل، با استفاده از گام کوچکتر، فیلترها با فاصله کمتری حرکت می‌کنند و اندازه نگاشت بزرگتر می‌شود. این باعث می‌شود که مدل بتواند ویژگی‌های جزئی‌تر را در تصویر تشخیص دهد و اطلاعات دقیق‌تری را در نگاشت حفظ کند. با این حال، استفاده از گام کوچکتر ممکن است منجر به افزایش پیچیدگی محاسباتی و کاهش سرعت آموزش و پیش‌بینی شود.

بنابراین، انتخاب مناسب برای اندازه گام بستگی به مسئله مورد نظر، اندازه تصاویر و پیچیدگی محاسباتی مورد نیاز دارد. برای مسائلی که نیاز به تشخیص ویژگی‌های دقیق و جزئی دارند، استفاده از گام کوچکتر مفید است. اما اگر سرعت محاسباتی یا سرعت آموزش و پیش‌بینی مهم باشد، ممکن است استفاده از گام بزرگتر مناسب باشد. همچنین، می‌توانید به صورت تجربی با اندازه‌های مختلف گام آزمایش کنید و اثر آن را روی دقت مدل موردنظر بررسی کنید.

(ج)

لایه‌های همگشتی در شبکه InceptionV3 از تکنیک پولینگ (pooling) استفاده می‌کنند. در کد شبکه ای که زدیم، پس از لایه GlobalAveragePooling2D، لایه Dense قرار داده شده است. این لایه با میانگین‌گیری از ویژگی‌های استخراج شده از لایه‌های قبلی، ابعاد ویژگی‌ها را کاهش می‌دهد و ویژگی‌های مهم را استخراج می‌کند.

ویژگی‌های استخراج شده توسط لایه‌های همگشتی در شبکه اغلب ویژگی‌های محلی از تصویر هستند، مانند نقاط قابل توجه، الگوها و اشکال مختلف. این ویژگی‌ها معمولاً اطلاعات مهم و تمایزدهنده‌ای در مورد تصویر و شی‌های موجود در آن را ارائه می‌دهند. این ویژگی‌ها سپس توسط لایه‌های Dense به عنوان ورودی در نظر گرفته می‌شوند و در نتیجه، شبکه قادر خواهد بود دسته‌بندی صحیح را انجام دهد.

بنابراین، لایه‌های همگشتی در شبکه InceptionV3 وظیفه استخراج ویژگی‌های تصویر را دارند و با استفاده از فیلترها و ترکیب نتایج، ویژگی‌های مختلفی را از تصویر استخراج می‌کنند. این ویژگی‌ها سپس توسط لایه‌های Dense در شبکه برای دسته‌بندی استفاده می‌شوند.

توضیح مدل:

کد شبکه‌های که طراحی کردم همانطور که در بالا اشاره کردم به صورت زیر است:

```
base_model = InceptionV3(include_top=False, weights='imagenet', input_shape=(75, 75, 3))
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
x = Dense(512, activation='relu')(x)
x = Dense(256, activation='relu')(x)
predictions = Dense(10, activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=predictions)

model = Model(inputs=base_model.input, outputs=predictions)
for layer in base_model.layers:
    layer.trainable = False
```

و تا اپک ۲۳ شبکه رو train کردم. کد ها در نوتبوک q7 موجود است.