



به نام خدا
درس مبانی یادگیری عمیق
پاسخنامه تمرین سری سوم
استاد درس : دکتر مرضیه داوودآبادی
دستیاران : سحر سرکار، فائزه صادقی، پریسا ظفیری،
شایان موسوی نیا
دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر
نیمسال اول تحصیلی ۱۴۰۲ - ۱۴۰۳

پاسخ های مناسب برای هر سوال، لزوماً یکتا نیستند

۱. سوالات زیر را با توجه به مبحث بهینه‌سازها پاسخ دهید (۱۵ نمره).

الف) استفاده از نرخ یادگیری بسیار بالا چه مشکلاتی را ایجاد می‌کند؟ چگونه می‌توان این مشکلات را تشخیص داد؟

مشکلات نرخ یادگیری بالا: تابع هزینه به یک راه‌حل بهینه همگرا نمی‌شود و حتی می‌تواند باعث واگرایی شود. - تشخیص: پس از هر تکرار به هزینه‌ها نگاه کنید (تابع هزینه در مقابل تعداد تکرارها را رسم کنید)، اگر هزینه به شدت نوسان کند، نرخ یادگیری بسیار بالاست. برای شیب نزولی دسته‌ای، اگر هزینه افزایش یابد، نرخ یادگیری بسیار بالاست.

ب) استفاده از نرخ یادگیری بسیار پایین چه مشکلاتی را ایجاد می‌کند؟ چگونه می‌توان این مشکلات را تشخیص داد؟

مشکلات نرخ یادگیری پایین: تابع هزینه ممکن است به یک راه‌حل بهینه همگرا نشود یا بعد از مدت زمان بسیار طولانی همگرا شود (گام‌های لازم برای رسیدن به نقطه بهینه را افزایش دهد). - تشخیص: پس از هر تکرار به هزینه‌ها نگاه کنید (تابع هزینه در مقابل تعداد تکرارها را رسم کنید). تابع هزینه بسیار آهسته (تقریباً خطی) کاهش می‌یابد. همچنین می‌توانید نرخ‌های یادگیری بالاتر را امتحان کنید تا ببینید آیا عملکرد بهبود می‌یابد.

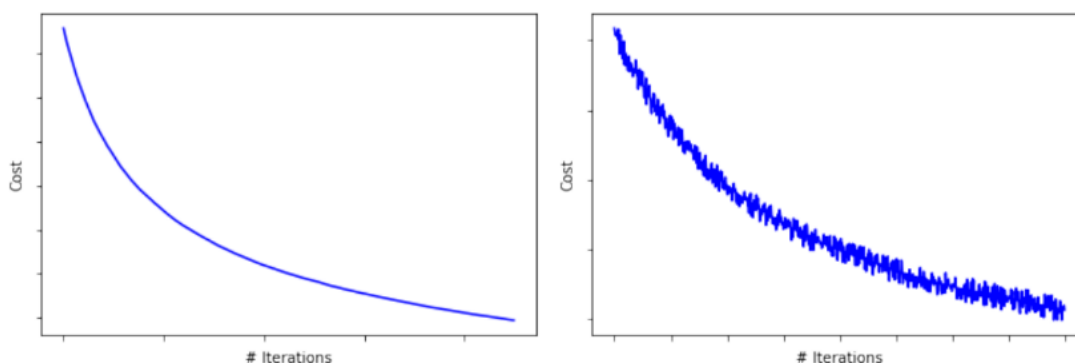
پ) نقطه زینی چیست؟ دو الگوریتم $Adam$ و SGD را در برخورد با این نقاط مقایسه کنید. مزایا و معایب هر کدام را بنویسید.

نقطه زینی: در این نقطه گرادیان صفر است، اما نه حداقل محلی است و نه حداکثر محلی. همچنین در تعریف دیگری از نقطه زینی آمده است که نقطه‌ای که در آن گرادیان صفر است و تابع در یک

جهت دارای حداکثر محلی است، اما در جهت دیگر یک حداقل محلی دارد. SGD: این بهینه‌ساز به روزرسانی‌های نویزی دارد که می‌تواند به فرار از یک نقطه زینی کمک کند. گاهی اوقات اگر اندازه گام به اندازه کافی بزرگ باشد (نرخ یادگیری بالا) می‌تواند از نقاط زینی ساده خارج شود، ولی این نوع بهینه‌ساز در برخورد با نقاط زینی نسبتاً پیچیده ممکن است در دام بیفتد.

Adam: این نوع بهینه‌ساز در برخورد با نقاط زینی بهتر عمل می‌کند. بهینه‌ساز Adam با داشتن اطلاعات مفیدتری مانند تکانه، همچنین استفاده از نرخ آموزش تطبیقی می‌تواند از نقاط زینی عبور کند.

ت) شکل زیر کاهش هزینه را (با افزایش تکرارها) زمانی که از دو الگوریتم بهینه‌سازی مختلف برای آموزش استفاده می‌شود، نشان می‌دهد. کدام یک از این نمودارها با استفاده از شیب نزولی دسته‌ای به عنوان الگوریتم بهینه‌سازی و کدام یک مربوط به استفاده از شیب نزولی دسته‌ای کوچک^۱ است؟ توضیح دهید.



شیب نزولی دسته‌ای: نمودار A

شیب نزولی دسته‌ای کوچک: نمودار B

در شیب نزولی دسته‌ای هزینه در هر تکرار کاهش می‌یابد (منحنی صاف) ولی در شیب نزولی دسته‌ای کوچک، هزینه در هر تکرار کاهش نمی‌یابد زیرا مدل فقط در یک دسته کوچک آموزش می‌بیند؛ پس تابع هزینه نویزی است

۲. ورودی یک لایه هم‌گشتی^۲ (X) با ابعاد سه در سه را در نظر بگیرید. فیلتر F با ابعاد دو در دو روی ورودی X اعمال شده است. روی خروجی این لایه هم‌گشتی، یک لایه ادغام میانگین سراسری^۳ اعمال می‌شود که خروجی نهایی یک عدد خواهد شد. باتوجه به اینکه گرادیان تابع ضرر نسبت به

^۱Mini-batch

^۲Convolutional layer

^۳Global average pooling (GAP)

این خروجی نهایی که یک عدد است، ۱ می‌شود، با استفاده از الگوریتم پس‌انتشار^۴ خطا، گرادیان‌های این لایه‌ی هم‌گشتی را به دست آورید (برای محاسبه می‌توانید از فرمول‌های این مرجع استفاده کنید)(۱۵ نمره).

۳	۴	۵
۲	۱	-۳
۴	-۲	۰

X

۲	۰
-۳	۱

F

ورودی X سه در سه است که یک فیلتر F دو در دو روی آن اعمال می‌شود. خروجی آن یک ماتریس دو در دو O است. از آن جا که بعد از O یک لایه GAP اعمال می‌شود که خروجی آن یک عدد خواهد شد و عملیات GAP یک میانگین‌گیری ساده از این چهار عدد به دست آمده است که حاصل یک شده است؛ بنابراین، مشتق عملیات GAP نسبت به هر یک از این چهار مقدار ماتریس O عدد ثابت ۰/۲۵ است. گرادیان نهایی هم یک فرض شده که ضرب آن در ۰/۲۵ تاثیری ندارد.

$$\frac{\partial L}{\partial F} = \text{Conv} \left(X, \frac{\partial L}{\partial O} \right) = \text{Conv} \left(\begin{bmatrix} 3 & 4 & 5 \\ 2 & 1 & -3 \\ 4 & -2 & 0 \end{bmatrix}, \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix} \right) = \begin{bmatrix} 2.5 & 1.75 \\ 1.25 & -1 \end{bmatrix}$$

$$\begin{aligned} \frac{\partial L}{\partial X} &= \text{FullConv} \left(180^\circ \text{rotated} F, \frac{\partial L}{\partial O} \right) = \text{FullConv} \left(\begin{bmatrix} 1 & -3 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix} \right) \\ &= \begin{bmatrix} 0.5 & 0.5 & 0 \\ -0.25 & 0 & 0.25 \\ -0.75 & -0.5 & 0.25 \end{bmatrix} \end{aligned}$$

^۴Backpropagation

۳. لطفاً به سوالات زیر پاسخ دهید.

الف) تعداد پارامترهای شبکه زیر را محاسبه کنید (محاسبات خود را برای هر لایه به‌طور کامل یادداشت کنید) (۱۰ نمره).

```
model = Sequential()  
model.add(Input(shape=(500, 7)))  
model.add(Conv1D(filters=16, kernel_size=3, activation="relu"))  
model.add(MaxPool1D())  
model.add(Conv1D(filters=32, kernel_size=5, activation="relu"))  
model.add(MaxPool1D())  
model.add(Conv1D(filters=64, kernel_size=5, activation="relu"))  
model.add(MaxPool1D())  
model.add(Flatten())  
model.add(Dense(units=128, activation="relu"))  
model.add(Dense(units=5, activation="softmax"))
```

Input :0

(500,7)

Conv1D: 352

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

(498,16)

MaxPool1D: 0

(249,16)

Conv1D: 2592

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

(245,32)

MaxPool1D:0

(122,32)

Conv1D:10304

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

(245,32)

MaxPool1D: 0

(3776)

Dense: 483456

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

(128)

Dense: 645

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

ب) لایه هم‌گشتی دو بعدی ($Conv2D$) و سه‌بعدی ($Conv3D$) را مقایسه کرده و کاربرد لایه $Conv3D$ را ذکر کنید (۱۰ نمره امتیازی).

$Conv2D$ برای طبقه‌بندی تصاویر استفاده می‌شود. این شبکه عصبی کانولوشن برای اولین بار در معماری $Lenet-5$ معرفی شد در $Conv3D$ ، هسته در ۳ بعد است. $Conv3D$ بیشتر برای داده‌های تصویربرداری MRI و CT که به طور گسترده برای بررسی مغز، نخاع اندام‌های داخلی هستند در زمینه طبقه‌بندی‌های پزشکی استفاده می‌شود. یکی دیگر از نمونه‌های داده‌های سه بعدی، ویدیو است. ویدیو چیزی نیست جز توالی فریم‌های تصویر در کنارهم. $Conv3D$ قابلیت بررسی ویژگی‌های مکانی و زمانی را دارد که برای داده‌هایی مثل سری زمانی بسیار مناسب است. به دلیل تعداد پارامترهای بیشتر، زمان آموزش بیشتری دارد. (این پاسخ یک نمونه است و نیازی نیست که پاسخ شما عیناً اینگونه باشد).

۴. در این سوال قصد داریم دادگان تومور مغزی را دسته‌بندی کنیم. لطفاً موارد خواسته شده را در نوت‌بوک $DL - Q4.ipynb$ را پیاده‌سازی کنید. اهداف این سوال به ترتیب موارد زیر می‌باشد (۲۰ نمره):

- خواندن تصاویر دادگان که در یک مسیر مشخص قرار داده شده‌اند و استفاده از آن‌ها برای آموزش و ارزیابی شبکه.

• ساختن یک شبکه عصبی هم‌گشتی با معماری یکسان با *Sequential API* و *Functional API*

با توجه به آنکه پیاده‌سازی‌های مختلف برای این سوال می‌تواند درست باشد، پاسخی برای این سوال نمی‌آوریم

۵. با در نظر گرفتن یک کاربرد واقعی مانند دسته‌بندی تصاویر، یک مثال عملی ارائه دهید که ویژگی‌های منحصر به فرد لایه‌های هم‌گشتی (مزایا) منجر به عملکرد مفیدی می‌شود؟ به طور عکس، آیا می‌توانید یک حالت تصور کنید که ویژگی‌های لایه‌های هم‌گشتی چالش‌هایی (معایب) را ایجاد می‌کنند؟ توضیح دهید که چرا و تأثیر آن بر کارایی مدل چیست (۱۰ نمره).

مزایا و معایب لایه‌های شبکه عصبی کانوالوشنی (*CNN*):

مزایا لایه‌های *CNN*:

استخراج ویژگی:

لایه‌های *CNN* به خوبی از تصاویر ویژگی‌های سلسله‌مراتبی استخراج می‌کنند. لایه‌های کانوالوشنی قادر به یادگیری الگوهای محلی، لبه‌ها و قلمروها هستند، در حالی که لایه‌های عمیق‌تر می‌توانند ویژگی‌های انتزاعی و پیچیده‌تری را یاد بگیرند. این قابلیت انتخاب خودکار ویژگی‌های مرتبط، *CNN* را در وظایف تحلیل تصاویر بسیار موثر می‌کند.

بی‌تغییری نسبت به ترجمه:

لایه‌های *CNN* به طور ویژه برای تشخیص الگوها بدون در نظر گرفتن محل آن‌ها طراحی شده‌اند. این ویژگی با استفاده از فیلترهای کانوالوشنی که تمام تصویر ورودی را پوشش می‌دهند، به دست می‌آید و باعث می‌شود *CNN* قادر به شناسایی الگوها با هر موقعیتی در تصویر شود.

به اشتراک گذاری پارامتر:

لایه‌های *CNN* از به اشتراک گذاری پارامتر استفاده می‌کنند که تعداد پارامترهای قابل یادگیری را نسبت به لایه‌های کاملاً متصل کاهش می‌دهد. از طریق به اشتراک گذاری وزن‌ها در مکان‌های مختلف، *CNN* ها از نظر استفاده از حافظه و نیازمندی‌های محاسباتی کارآمدتر می‌شوند. این به اشتراک گذاری پارامترها همچنین باعث می‌شود که *CNN* ها به خوبی برای داده‌های جدید و نامعلوم عمل کنند.

نمایش سلسله‌مراتبی:

لایه‌های *CNN* معمولاً شامل چندین لایه کانوالوشنی و استخراجی هستند که یک نمایش سلسله‌مراتبی از تصویر ورودی ایجاد می‌کنند. هر لایه بعدی ویژگی‌های انتزاعی پیچیده‌تری را یاد می‌گیرد و به شبکه اجازه می‌دهد روابط پیچیده را در داده‌ها یاد بگیرد. این نمایش سلسله‌مراتبی برای موفقیت *CNN* ها در وظایف درک تصویر بسیار حائز اهمیت است.

معایب لایه‌های *CNN*:

نیاز به حافظه بالا و نیازهای محاسباتی:

با عمق شدن *CNN*، تعداد پارامترها و محاسبات مورد نیاز به طور قابل توجهی افزایش می‌یابد. این ممکن است باعث افزایش مصرف حافظه و طولانی شدن زمان آموزش و پیش‌بینی شود. آموزش مدل‌های بزرگ *CNN* ممکن است نیاز به منابع محاسباتی قابل توجهی داشته باشد و این موضوع مناسب برای دستگاه‌های با توان محدود یا منابع محدود نیست.

عدم تغییرپذیری مکانی:

اگرچه لایه‌های *CNN* از نظر ترجمه بی‌تغییر هستند، ممکن است در قبال تغییرات مکانی دیگر مانند چرخش، مقیاس‌بندی یا تغییر شکل، تغییرپذیر نباشند. *CNN* هایی که بر روی تغییرات مشخص آموزش دیده‌اند، ممکن است به خوبی به تصاویری با تغییرات مکانی متفاوت عمل نکنند. تکنیک‌های مختلفی مانند افزایش داده و اضافه کردن لایه‌های اضافی می‌تواند به کاهش این محدودیت کمک کند.

محدودیت در درک متناسب:

لایه‌های *CNN* دارای میدان اطلاعاتی محدود هستند، به این معنی که تنها می‌توانند اطلاعات را از نزدیکی محل مورد نظر استخراج کنند. این محدودیت می‌تواند بر قابلیت شبکه در درک زمینه کلی یا وابستگی‌های دوربرد روی تصاویر تأثیر بگذارد. برای حل این مسئله، می‌توان از میدان اطلاعاتی بزرگتر یا استفاده از ساختارهای شبکه اضافی (مانند مکانیزم‌های توجه) استفاده کرد.

چالش‌های تفسیرپذیری:

لایه‌های *CNN* معمولاً به عنوان مدل‌های جعبه سیاه در نظر گرفته می‌شوند، زیرا تفسیر تصمیمات آن‌ها می‌تواند مشکل باشد. درک اینکه چرا یک *CNN* تصمیم خاصی را گرفته است ممکن است دشوار باشد، به ویژه زمانی که شبکه عمیق‌تر و پیچیده‌تر می‌شود. تکنیک‌هایی مانند تصویرسازی و روش‌های اختصاص می‌توانند به درک عملکرد داخلی *CNN* کمک کنند، اما تفسیر کامل همچنان چالش برانگیز است.

۶. فیلترهای 1×1 در شبکه‌های عصبی هم‌گشتی ویژگی‌های منحصربه‌فردی دارند که عملکرد شبکه را تحت تأثیر قرار می‌دهند. در این سوال قصد داریم تأثیر فیلترهای 1×1 را بررسی کنیم (۱۵ نمره).

الف) هدف استفاده از فیلترهای 1×1 در شبکه‌های عصبی هم‌گشتی چیست و چگونه به کاهش تعداد نقشه‌های ویژگی در حین حفظ ویژگی‌های مهم کمک می‌کنند؟
هدف از استفاده از فیلترهای 1×1 در شبکه‌های عصبی هم‌گشتی عبارت است از:

(آ) کاهش ابعاد فضایی: فیلترهای 1×1 به شبکه اجازه می‌دهند که تعداد نقشه‌های ویژگی را کاهش دهند، این امر باعث می‌شود که محاسبات کلی شبکه کاهش یابد.

(ب) ادغام ویژگی: این فیلترها کمک می‌کنند تا ویژگی‌ها از لایه‌های قبلی ادغام شوند و ویژگی‌های مهم تر و ترکیبی از ویژگی‌های کمتر اهمیت به دست آید.

ب) پس از اعمال فیلتر 1×1 ، نقشه ویژگی چه اطلاعاتی ارائه می‌دهد؟
پس از اعمال فیلتر 1×1 ، نقشه ویژگی اطلاعاتی مربوط به ترکیب خطی و وزن‌های مختلف از ویژگی‌های ورودی را ارائه می‌دهد.

پ) این نقشه ویژگی چگونه از تصویر اصلی یا فیلترهای دیگر با اندازه‌های مختلف متفاوت است؟
این نقشه ویژگی از تصویر اصلی با اندازه‌ها و ترکیب‌های مختلف متفاوت است. به عبارت دیگر، این نقشه ویژگی با استفاده از فیلترهای 1×1 ، یک ترکیب خطی از ویژگی‌های مختلف ایجاد می‌کند که می‌تواند ویژگی‌های خاص و مهمی را برجسته کند.

ت) در چه مدل‌هایی از این نوع فیلتر استفاده شده است؟
این نوع فیلتر در مدل‌های مختلف مورد استفاده قرار گرفته است. به عنوان مثال، *ResNet* از این فیلترها در برخی از لایه‌ها برای افزایش عمق و ادغام ویژگی‌ها استفاده کرده است. همچنین، در *Inception module* ها نیز این نوع فیلتر به منظور ادغام ویژگی‌های مختلف با اندازه‌های مختلف استفاده شده است.

ث) آیا حالت‌هایی وجود دارد که استفاده از فیلترهای 1×1 ممکن است مفید نباشد؟ اگر بله، دلیل چیست؟

بله، ممکن است در برخی حالات استفاده از فیلترهای 1×1 مفید نباشد. اگر ابعاد مکانی ویژگی‌ها خیلی کوچک باشند، استفاده از فیلتر 1×1 ممکن است بهبود محاسباتی معنی‌داری نداشته باشد. همچنین، اگر ترکیب خطی از ویژگی‌ها به نفع مسئله مورد نظر نباشد، استفاده از این فیلترها ممکن است موثر نباشد.

ج) یک مدل هم‌گشتی ساده با فیلتر 1×1 پیاده‌سازی کنید و تاثیر آن را روی یک ورودی دلخواه با مقایسه اندازه ورودی و خروجی مدل نشان دهید.


```

import tensorflow as tf

# Define the input tensor
input_tensor = tf.random.normal([batch_size, height, width,
    num_channels])

# Define the 1x1 filter
num_filters = 32
filter_size = 1
conv_layer = tf.keras.layers.Conv2D(filters=num_filters, kernel_size
    =filter_size, activation='relu')

# Apply the 1x1 filter
output_tensor = conv_layer(input_tensor)

# Print the shape of the output tensor
print(output_tensor.shape)

```

۷. یک شبکه عصبی هم‌گشتی که شامل یک ماژول *Inception* برای طبقه بندی تصویر با استفاده از مجموعه داده ۱۰ - *CIFAR* است، پیاده سازی کنید. در اجرای خود، حتما به موارد زیر توجه کنید (۱۵ نمره):

- ساختار و هدف ماژول *Inception* را در این شبکه توضیح دهید.
- چگونه اندازه پارامتر گام^۵ در لایه‌های هم‌گشتی بر ابعاد فضایی نگاشت^۶ ویژگی‌ها تأثیر می‌گذارد؟
- ویژگی‌های کلیدی و عملیات لایه‌های هم‌گشتی مورد استفاده در شبکه خود و اهمیت آنها در استخراج ویژگی را شرح دهید.

^۵Stride

^۶Mapping

- انجام پیاده سازی صرفاً با استفاده از مدل های آماده در کتابخانه های *keras* و نظیر آنها بدون مشکل است.

- هدف این تمرین رسیدن به دقت بالای ۸۰ درصد در آموزش است. برای رسیدن به این هدف از همه روش هایی که در درس آموخته اید استفاده کنید (۱۰ نمره امتیازی).

به نوتبوک [Q7.ipynb](#) مراجعه کنید.