به نام خدا

دانشگاه صنعتی شریف
دانشکده‌ی مهندسی برق

یادگیری عمیق

پاییز ۱۴۰۱

استاد: دکتر فاطمی‌زاده

گردآورندگان:سعید رضوی، هلیا حاج‌کاظمی، ارشیا همت

تمرین سوم | شبکه‌های عمیق کانولوشنی | مهلت ارسال: جمعه ۱۷ آذر

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص‌شده است.
- در طول ترم امکان ارسال با تاخیر پاسخ همه‌ی تمارین تا سقف ۶ روز و در مجموع ۲۰ روز، وجود دارد. پس از گذشت این مدت، پاسخ‌های ارسال‌شده پذیرفته نخواهند بود. همچنین، به ازای هر روز تأخیر غیر مجاز ۱۰ درصد از نمره تمرین به صورت ساعتی کسر خواهد شد.
- هم‌کاری و هم‌فکری شما در انجام تمرین مانعی ندارد اما پاسخ ارسالی هر کس حتما باید توسط خود او نوشته شده باشد. (دقت کنید در صورت تشخیص مشابهت غیرعادی برخورد جدی صورت خواهد گرفت.)
- در صورت هم‌فکری و یا استفاده از هر منابع خارج درسی، نام هم‌فکران و آدرس منابع مورد استفاده برای حل سوال مورد نظر را ذکر کنید.
- لطفا تصویری واضح از پاسخ سوالات نظری بارگذاری کنید. در غیر این صورت پاسخ شما تصحیح نخواهد شد.
- نتایج و پاسخ های خود را در یک فایل با فرمت zip به نام HW۳-Name-StudentNumber در سایت Quera قرار دهید. برای بخش عملی تمرین نیز لینگ گیت‌هاب که تمرین و نتایج را در آن آپلود کرده‌اید قرار بدهید. دقت کنید هر سه فایل نوتبوک تکمیل شده بخش عملی را در گیت‌هاب قرار دهید .
- لطفا تمامی سوالات خود را از طریق کوئرای درس مطرح بکنید (برای اینکه تمامی دانشجویان به پاسخ‌های مطرح شده به سوالات دسترسی داشته باشند و جلوی سوالات تکراری گرفته شود، به سوالات در بسترهای دیگر پاسخ داده نخواهد شد).
- دقت کنید کدهای شما باید قابلیت اجرای دوباره داشته باشند، در صورت دادن خطا هنگام اجرای کدتان، حتی اگه خطا بدلیل اشتباه تایپی باشد، نمره صفر به آن بخش تعلق خواهد گرفت.

سوالات نظری (۳۰۰ نمره)

۱. (۲۵ نمره)

یک مسئله دسته‌بندی با ۵ دسته، که دیتاست، شامل تصاویری به اندازه ۱۰ × ۱۰ پیکسل میباشند، داریم. دو شبکه عصبی یک لایه را به صورت زیر در نظر بگیرید. توضیح دهید کدام یک انتخاب بهتری میباشد؟

- یک لایه fully connected که ورودی آن، flatten (بردار شده) تصاویر دیتاست میباشد.
- یک لایه کانولوشن که که در آن ۵ فیلتر به اندازه ۱۰ × ۱۰ داریم.

۲. (۲۵ نمره)

فرض کنید دیتاستی داریم شامل تصاویر رنگی به اندازه ۱۲۸ × ۱۲۸. میخواهیم یک شبکه عصبی کانولوشن برای آن طراحی کنیم.

- اندازه خروجی و تعداد پارامترهای لایه اول کانولوشن را محاسبه کنید اگر ۱۶ فیلتر ۵ × ۵ با $stride = ۱$ و $padding = ۲$ داشته باشیم.

- فرض کنید هر لایه، ۳ قسمت شامل : کانولوشن، max pooling و تابع فعالسازی (Relu) را دارا می‌باشد، که لایه‌های کانولوشنی، هر کدام شامل ۱۶ فیلتر ۵$\times$ ۵ با $stride = ۱$ و $padding = ۲$ و لایه‌های max pooling همگی ۲ $\times$ ۲ با $stride = ۲$ هستند. ۳ لایه با این مشخصات را پشت سر هم در نظر بگیرید. اندازه تنسور در لایه خروجی نهایی و تعداد پارامترهای این ۳ لایه را حساب کنید.

- فرض کنید هدف، حل یک مسئله classification، که شامل ۱۰ دسته هست،می‌باشد.تعداد کل پارامترهای شبکه را در این حالت حساب کنید.

- در این قسمت، با یک مفهوم مهم آشنا می‌شویم. Receptive field بیانگر این است که نورون خروجی، تحت تاثیر چه مقدار از نورون‌های ورودی می‌باشد. در حقیقت تعیین می‌کند هر نورون خروجی از چه ناحیه‌ای با چه اندازه‌ای از تنسور ورودی تاثیر می‌پذیرد. حال Receptive field را برای یک نورون خروجی لایه سوم (قبل از لایه fully connected) بررسی کنید. (برای فهم بهتر این مفهوم میتوانید به این لینک مراجعه کنید)

۳. (۲۵ نمره)

در این تمرین قصد داریم به بررسی دو شبکه‌ی معروف یعنی Densely Connected Convolutional Networks و U-Net بپردازیم. برای بررسی هرچه بهتر این دو شبکه بهتر است به لینک‌های زیر مراجعه نمایید.

- https://arxiv.org/pdf/1505.04597.pdf
- https://arxiv.org/pdf/1608.06993.pdf

در این تمرین دو نوع سوال وجود دارد، یکی از این سوالات، سوالات مفهومی‌ست که میزان تسلط شما برروی شبکه‌های موجود را بررسی می‌کند و دوم سوالات محاسبه‌ای می‌باشد.

۱ - سوالات مفهومی مربوط به U-Net:

- ویژگی اصلی شبکه‌ی U-Net که آن را از یک شبکه کانولوشنی عادی متمایز می‌دارد چه می‌باشد و دلیل اینکه ما شاهد یک ساختار U شکل هستیم چه می‌باشد؟
- می‌دانیم که Skip connection ها نقشی پررنگ در این شبکه‌ها دارند، دلیل حضور این مورد را در شبکه‌ی U-Net بیان کنید.
- **سوال اضافه (دارای نمره‌ی اضافه‌ی جزئی)**: چرا این نوع از اتصالات در تصاویر پزشکی دارای اهمیت بیشتر می‌باشند و چه کمکی به ما در دامنه‌ی تشخیص موارد پزشکی می‌کنند؟

۲ - سوالات محاسبه‌ای مربوط به U-Net:

- تصور کنید که ابعاد تصویر ورودی ما برای این شبکه ۲۵۶ $\times$ ۲۵۶ می‌باشد. حال فرض می‌شود که در این معماری هرلایه در انکدر ابعاد را به نصف کاهش می‌دهد و در دیکدر دو برابر می‌کند. در پایین‌ترین لایه (عمیق‌ترین لایه) این معماری، فضای ویژگی ما چند پیکسل خواهد داشت؟
- در U-Net، فرض کنید انکودر دارای لایه‌هایی با ۶۴، ۱۲۸، ۲۵۶ و ۵۱۲ فیلتر است. اگر هر لایه کانولوشن از کرنال‌های ۳ $\times$ ۳ استفاده بکند، تعداد پارامترهای لایه کانولوشن دوم انکدر را محاسبه کنید.

۳ - سوالات مفهومی DenseNet:

- تفاوت‌های اصلی ResNet's residual connections و DenseNet's dense connections را بیان کنید. درمورد هرکدام از موارد گفته نیز، توضیح مختصری بدهید.
- بیان کنید که DenseNet چگونه مشکل vanishing gradient را کاهش می‌دهد و مزیت محاسباتی آن چه می‌باشد؟

۴- سوالات محاسبه‌ای DenseNet:

- در یک DenseNet با سه لایه در یک Dense Block اگر لایه اول ۶۴ فیچر مپ تولید کند، لایه دوم ۱۲۸ فیچر مپ و لایه سوم ۲۵۶ فیچر مپ تولید کند، لایه سوم چند فیچر مپ ورودی را دریافت خواهد کرد؟
- با در نظر گرفتن نرخ رشد k در DenseNet، اگر هر لایه k فیچر مپ جدید تولید کند و ورودی یک dense block دارای ۳۲ کانال باشد، اگر $k = ۲۴$ باشد لایه سوم در بلوک چند کانال خروجی خواهد داشت؟

سوالات عملی (۳۰۰ نمره)

۱. (۱۰۰ نمره) همانطور که می‌دانید، در بسیاری از مدل‌های شبکه عمیق، از یک تابع ضرر (Loss function) برای آموزش مدل استفاده می‌کنند. نکته‌ای که وجود دارد این است که لزوما این تابع ضرر، مختص به لایه آخر شبکه عصبی نیست و می‌شود از آن درلایه‌های میانی نیز استفاده کنیم. همچنین، در بسیاری از موارد برای بهبود آموزش، می توانیم از جمع وزن‌دار چند تابع ضرر به صورت همزمان استفاده کنیم. در ادامه‌ی سوال، با این موارد بیشتر آشنا می‌شویم.

در این سوال، از معماری یکی از شبکه‌های معروف کانولوشنی (مانند Resnet50 ، Alexnet و...) که بر روی imagenet ترین شده به دلخواه استفاده میکنیم. هدف ما آموزش یک classifier برای دو کلاس هواپیما (airplane) و ماشین (automobile) از دیتاست cifar10 است. همانطور که میدانیم، این نوع شبکه‌ها، شامل یک لایه fully connected هستند که ورودی آن برداری به اندازه feature vector استخراج شده و خروجی آن به به اندازه تعداد کلاس‌ها است. مثلا برای Alexnet، خروجی لایه‌ی یکی مانده به آخر، یک بردار ۴۰۹۶ تایی است که از طریق یک لایه fully connected به یک بردار ۱۰۰۰ تایی برای مسئله دسته‌بندی مپ شده است. از آنجایی که تعریف‌های ما برای استخراج بردار ویژگی همیشه ممکن است دارای نقص هایی باشند و برخی موارد در نظر گرفته نشده باشند، چنانچه داده‌های بسیار زیاد و متنوعی به شبکه هایی مانند AlexNet نشان داده شود چه بسا بردار ویژگی که به دست می آورند بهتر از آنهایی باشد که متخصصین تعریف می کنند. در مسائل مختلف دیده شده است که این بردار ویژگی ها حتی برای مسائلی که دسته‌بندی نیستند و یا دسته ها متفاوت از دسته‌های ImageNet هستند هم با معنی بوده و به نتایج خوب منجر می‌شوند. در ادامه شما ملزم به انجام موارد زیر هستید :

- یک شبکه کانولوشنی معروف به دلخواه انتخاب کنید، و با تغییر دادن لایه fully connected آن، شبکه را برای دسته‌بندی یک مسئله دو کلاسه، آماده کنید. (دقت کنید این دو دسته، دو کلاس هواپیما (airplane) و ماشین (automobile) از دیتاست cifar10 هستند.) .

(آ) شبکه کانولوشنی خود را با استفاده از وزن‌های از قبل آموزش داده شده و cross entropy loss آموزش دهید(فقط وزن‌های مربوط به لایه fully connected را مقداردهی اولیه کنید). نمودار دقت و loss را به ازای epoch های مختلف رسم کنید. دقت کنید این دقت و loss بر روی دیتاست ترین باید صورت بگیرد.

(ب) در نهایت، دقت مدل ترین‌شده بالا را بر روی داده تست حساب کنید.

- این بار با جای استفاده از cross entropy loss از triplet loss استفاده کنید .(برای آشنایی بیشتر با triplet loss می‌توانید از این لینک استفاده کنید). به موارد زیر دقت کنید :

(آ) به دلیل ماهیت triplet loss، کلاس مربوط به دیتاست باید توسط خودتان نوشته شود.

(ب) هدف در این قسمت این است که با استفاده از triplet loss ابتدا یک feature extractor خوب آموزش دهیم (دقت شود در این بخش آموزش، لایه fully connected دخیل نشده). پس از آموزش دادەشدن feature extractor، حال لایه fully connected را با فریز کردن وزن لایه‌های قبلی، با cross entropy loss آموزش دهید.

(ج) نمودارهای خواسته شده در قسمت ۱ را برای این بخش نیز رسم کنید (هم برای آموزش feature extractor و هم برای آموزش لایه classifier)

(د) دقت نهایی این مدل را بر روی دیتاست تست حساب کنید.

(ه) تمام کارهای بالا را کافیست با استفاده از وزن‌های pre-trained انجام دهید و نیازی به مقداردهی اولیه نیست.

- این بار میخواهیم این دو تابع ضرر را همزمان در آموزش دخیل کنیم. برای اینکار، تابع ضرر زیر را در نظر بگیرید:

$$L_{total} = L_{triplet} + L_{cross-entropy}$$

عمل backpropagation را بر روی L_{total} انجام دهید. به موارد زیر دقت کنید :

(آ) برعکس قسمت قبل، که ابتدا مدل استخراج ویژگی آموزش داده می‌شد و سپس classifier، در اینجا کل مدل در حال ترین شدن است.

(ب) نمودارهای خواسته شده در قسمت ۱ را برای این بخش نیز رسم کنید.

(ج) دقت نهایی این مدل را بر روی دیتاست تست حساب کنید.

(د) تمام کارهای بالا را کافیست با استفاده از وزن‌های pre-trained انجام دهید و نیازی به مقداردهی اولیه نیست.

۲. (۱۰۰ نمره)

این تمرین شامل دو بخش می‌باشد؛ هدف تمرین شناخت کامل موضوع Deformable Convolution می‌باشد؛ شما در این تمرین ابتدا به چند سوال تئوری پاسخ خواهید داد و پس از آن به سراغ پیاده‌سازی این شبکه‌های کانولوشنی می‌روید. لطفا به این نکته توجه کنید که پیاده‌سازی‌های انجام شده باید از ابتدا (From Scratch) باشد. به این صورت که شما می‌توانید از فریم‌ورک‌های پیاده‌سازی مدل‌های عمیق نظیر Pytorch استفاده نمایید، اما اجازه‌ی استفاده از پیاده‌سازی آماده این مورد را ندارید.

بخش اول - سوالات تئوری مربوط به تمرین عملی:

(آ) تفاوت بین شبکه‌های کانولوشنی عادی و شبکه‌های کانولوشنی Deformable را از نظر grid sampling مقایسه کرده و نتایج مقایسه خود را بنویسید.

(ب) شبکه‌های Deformable چگونه می‌توانند انعطاف‌پذیری را در Geometric transformation در تصاویر را به وجود آوردند.

(ج) به عقیده شما چرا شبکه‌های کانولوشنی ساده در مواجهه با تصاویری که آبجکت‌های تصویر دارای تغییر یا چرخش فضایی زیادی می‌باشد دچار مشکلات جدی‌ای می‌شود؟

(د) چگونه آفست‌های موجود در Deformable Convolution محاسبه می‌شوند؟

بخش دوم - سوالات عملی:
در ابتدا به این نکته توجه کنید که دیتاست مربوط به این سوال دیتاست MS COCO می‌باشد. شما از طریق لینک‌های زیر می‌تواند اطلاعات بیشتری در مورد این دیتاست بدست آورید (نیاز نیست از همه‌ی دیتاست برای آموزش شبکه استفاده کنید)

- paperswithcode.com/dataset/coco
- cocodataset.org/#home

بخش سوم - گزاراشات مربوط به پیاده‌سازی:
موارد زیر را براساس تغییر نقاط نمونه‌گیری برای شبکه‌های کانولوشنی Deformable و عادی گزارش کنید:

- دقت داده‌های آموزشی و آزمایشی.
- خطای مدل برای داده‌های آموزشی و آزمایشی
- مدت زمان اجرا برای هرکدام از موارد خواسته شده.

همچنین می‌توانید جهت آشنایی بیشتر با ایده Deformable Convolution و کارهای مهم تحقیقاتی که بر پایه این ایده ارائه شده‌اند، به ضمیمه این تمرین مراجعه نمایید.

۳. (۱۰۰ نمره)

در این تمرین میخواهیم شبکه‌ای آموزش دهیم که بتواند تغییرات شامل **displacement (جابه جایی)**، **rotation** و **scaling** بین دو تصویر را تشخیص دهد. برای انجام این تمرین فایل detect_change.ipynb را تکمیل کنید. توجه داشته باشید که در گزارش خود نحوه‌ی آماده‌سازی و ساخت دیتاست خود را توضیح دهید.

Deformable Convolution: An Overview

Arshia Hemmat

Introduction

Deformable convolution represents a significant advancement in the field of computer vision, particularly in the context of deep learning. It extends the traditional convolution operation by enabling adaptive receptive fields, thus allowing for more effective processing of spatial transformations in images.

Key Features of Deformable Convolution

Deformable convolution, as an extension of the traditional convolution operation, brings several key advantages to the field of computer vision and deep learning:

- **Adaptive Receptive Fields**: Unlike standard convolution, deformable convolution adapts the spatial sampling locations in the input feature map based on the learned offsets. This allows for dynamic adjustment of the receptive fields to better capture irregular and complex shapes in images.
- **Enhanced Feature Learning**: By adjusting to the geometrical variations of objects, deformable convolution can learn more robust and discriminative features, improving performance in tasks like object detection and segmentation.
- **Versatility in Handling Spatial Transformations**: Deformable convolution effectively handles various spatial transformations, such as scaling, rotation, and deformation, which are common challenges in real-world images.
- **Improved Modeling of Geometric Variations**: It offers superior capabilities in modeling geometric variations within images, thus significantly enhancing the capability to understand complex visual patterns and structures.
- **Compatibility and Integration**: Deformable convolution can be seamlessly integrated into existing convolutional neural network architectures, enhancing their capability without a significant overhaul of the network structure.

These features make deformable convolution a powerful tool in the advancement of deep learning models, particularly for applications involving complex and variable visual data.

Applications

Deformable convolution has been instrumental in advancing a variety of applications, particularly in computer vision and deep learning, due to its flexibility and enhanced feature learning capabilities:

- **Object Detection and Segmentation**: It greatly improves the accuracy of detecting and segmenting objects, especially those with irregular or complex shapes. This is crucial in applications such as autonomous driving, where accurately recognizing objects in various forms and orientations is vital.
- **Medical Image Analysis**: In medical imaging, such as MRI and CT scans, deformable convolution helps in accurately identifying and segmenting tumors or other anomalies, which often vary greatly in shape and size.

- **Video Surveillance and Analysis**: It aids in the analysis of surveillance footage, particularly in tracking moving objects or understanding complex scenes with varying object orientations and deformations.
- **Image and Video Enhancement**: Deformable convolution is used in enhancing the quality of images and videos, especially in super-resolution tasks where details need to be accurately reconstructed.
- **Facial Recognition and Analysis**: The technology is effective in facial recognition systems, capable of handling varied facial expressions, angles, and occlusions.
- **Augmented and Virtual Reality**: In AR and VR, deformable convolution improves the realism and accuracy of virtual objects overlayed on real-world scenes, adapting to the dynamic environments.

These applications demonstrate the versatility and effectiveness of deformable convolution in dealing with complex, real-world visual data, making it a cornerstone technology in many cutting-edge applications.

Best References

1. Dai, Jifeng, et al. "Deformable Convolutional Networks." Proceedings of the IEEE International Conference on Computer Vision. 2017.
2. Zhu, Xizhou, et al. "Deformable ConvNets v2: More Deformable, Better Results." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.
3. Haris, Muhammad, et al. "Deep Back-Projection Networks for Super-Resolution." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
4. Thomas, Daniel E., et al. "Spatially Adaptive Computation Time for Residual Networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.
5. Sun, Ke, et al. "High-Resolution Representations for Labeling Pixels and Regions." arXiv preprint arXiv:1904.04514 (2019).
6. Chen, Kai, et al. "Hybrid Task Cascade for Instance Segmentation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.
7. Chen F, Wu F, Xu J, Gao G, Ge Q, Jing XY. "Adaptive Deformable Convolutional Network." Neurocomputing. 2021 Sep 17;453:853-64.
8. Liu N, Long Y, Zou C, Niu Q, Pan L, Wu H. "Adcrowdnet: An Attention-Injective Deformable Convolutional Network for Crowd Understanding." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019 (pp. 3225-3234).
9. Wang X, Chan KC, Yu K, Dong C, Change Loy C. "EDVR: Video Restoration with Enhanced Deformable Convolutional Networks." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2019 (pp. 0-0).
10. Zhang Q, Xiao J, Tian C, Chun-Wei Lin J, Zhang S. "A Robust Deformed Convolutional Neural Network (CNN) for Image Denoising." CAAI Transactions on Intelligence Technology. 2023 Jun;8(2):331-42.

Best Practical Resources

1. **Deformable Convolution V2 in PyTorch (by chengdazhi)**: A PyTorch implementation of Deformable Convolution V2, compatible with multiple PyTorch versions and part of the official mmdetection repository.
   URL: `https://github.com/chengdazhi/Deformable-Convolution-V2-PyTorch`

2. **Deformable Convolutional Networks in PyTorch (by 1zb)**: This implementation of Deformable Convolution in PyTorch is adapted from the author's MXNet version, with build instructions and example usage provided.
   URL: `https://github.com/1zb/deformable-convolution-pytorch`

3. **PyTorch-Deformable-Convolution-v2 (by developer0hye)**: A user-friendly PyTorch implementation of Deformable Convolution V2, designed to simplify the usage for developers.
   URL: `https://github.com/developer0hye/PyTorch-Deformable-Convolution-v2`

Conclusion

Deformable convolution represents a dynamic and evolving field, with ongoing research continually enhancing its capabilities and applications. Its adaptability and effectiveness in handling complex visual data make it a pivotal tool in advanced image processing and analysis.