

۱ شناسایی رقم با استفاده از شبکه‌های عصبی پیچشی (CNN)

در دیتاست MNIST

۱-۱ مرور کلی:

در این پروژه شما یک شبکه عصبی پیچشی (CNN) برای انجام کار شناسایی رقم روی دیتاست معروف MNIST را می‌سازید. دیتاست MNIST یک پایگاه داده بزرگ از ارقام دست‌نویس است که معمولاً برای آموزش سیستم‌های مختلف پردازش تصویر استفاده می‌شود. این دیتاست شامل ۶۰,۰۰۰ تصویر آموزشی و ۱۰,۰۰۰ تصویر تست است.

۲-۱ اهداف آموزشی

۱. فهم اصول و معماری شبکه‌های عصبی پیچشی.
۲. تجربه عملی با CNN ها با استفاده از یک دیتاست واقعی.
۳. یادگیری استفاده از ابزارهای ساخت، آموزش و ارزیابی CNN ها.
۴. فهم و کاربرد تکنیک‌هایی مانند max pooling، لایه‌های پیچشی، و لایه‌های تمام متصل.

۳-۱ وظایف

۱. دانلود و پیش‌پردازش داده‌ها: دیتاست MNIST را دانلود کنید و آن را برای CNN خود پیش‌پردازش کنید. پیش‌پردازش ممکن است شامل تغییر شکل تصاویر و نرمال‌سازی آنها، و همچنین آماده‌سازی برچسب‌ها باشد.
۲. طراحی معماری شبکه: یک CNN کم عمق برای این کار طراحی کنید. شما باید حداقل شامل یک لایه پیچشی، یک لایه max pooling، و یک لایه کاملاً متصل باشد. شما همچنین

می‌توانید از تابع فعال‌سازی مانند ReLU استفاده کنید، و همچنین باید در نظر بگیرید که چه نوع لایه خروجی برای این مسئله دسته‌بندی چند کلاسه نیاز دارید.

۳. آموزش شبکه: شبکه خود را روی دیتاست MNIST آموزش دهید. شما باید یک تابع خطا و بهینه‌ساز مناسب انتخاب کنید. همچنین باید تصمیم بگیرید که چند epoch شبکه خود را آموزش دهید.

۴. ارزیابی شبکه: عملکرد شبکه خود را روی مجموعه تست MNIST ارزیابی کنید. شما باید معیارهای مربوطه مانند دقت شبکه خود را گزارش کنید.

۵. بحث: در مورد نتایج خود بحث کنید. عملکرد شبکه شما چگونه بود؟ آیا می‌توانید بهبودی ایجاد کنید؟

۴-۱ تحویل‌ها

۱. فایل‌های کد پایتون (py یا ipynb) با کامنت‌های واضح که هر بخش از کد را توضیح می‌دهد.

۵-۱ نکات

۱. از کتابخانه‌هایی مانند TensorFlow، PyTorch، یا Keras برای ساخت و آموزش CNN خود استفاده کنید.

۲. مجموعه آموزش MNIST را به مجموعه آموزش و اعتبارسنجی تقسیم کنید تا پارامترهای مدل خود را بهینه کنید.

۳. با هایپرپارامترهای مختلفی مانند نرخ یادگیری یا اندازه دسته آزمایش کنید.

۶-۱ معیارهای ارزیابی

۱. کیفیت کد.

۲. کیفیت طراحی CNN.

۳. عملکرد CNN در مجموعه تست MNIST.

۷-۱ مراجع

- شبکه‌های عصبی پیچشی (ConvNets / CNNs)

- دیتاست MNIST

۲ فشرده‌سازی تصویر با استفاده از تبدیل فوریه

۱-۲ مرور کلی:

در این تکلیف، شما باید با استفاده از تبدیل فوریه یک تصویر را فشرده‌سازی کنید. تبدیل فوریه ابزار ریاضی‌ای است که می‌تواند یک سیگنال یا تصویر را به فرکانس‌های آن تجزیه کند، که به ما اجازه می‌دهد سیگنال را در دامنه فرکانس تجزیه و تحلیل کنیم. این تکلیف به شما کمک می‌کند تا مفاهیم مرتبط با تبدیل فوریه و کاربردهای آن در پردازش تصویر، به‌خصوص در فشرده‌سازی تصویر را فهمیده و پیاده‌سازی کنید.

۲-۲ اهداف آموزشی

۱. درک اصول و ریاضیات تبدیل فوریه.
۲. درک و پیاده‌سازی تکنیک‌های فشرده‌سازی تصویر با استفاده از تبدیل فوریه.
۳. تحلیل اثرات فشرده‌سازی بر کیفیت تصویر.

۳-۲ وظایف

۱. مطالعه و پیاده‌سازی تبدیل فوریه: درک نظریه و ریاضیات تبدیل فوریه و پیاده‌سازی آن برای تبدیل یک تصویر از دامنه مکانی به دامنه فرکانس.
۲. تحلیل مؤلفه‌های فرکانسی: تحلیل مؤلفه‌های فرکانسی تصویر و درک سهم آنها در داده تصویر.
۳. فشرده‌سازی تصویر: فشرده‌سازی تصویر با حذف برخی از مؤلفه‌های فرکانسی که کمترین سهم را در داده تصویر دارند. شما باید تعادلی بین نرخ فشرده‌سازی و کیفیت تصویر ایجاد کنید.
۴. بازسازی تصویر: پیاده‌سازی تبدیل فوریه معکوس برای بازسازی تصویر از نمایش فرکانسی فشرده‌سازی شده.

۵. ارزیابی و مقایسه: ارزیابی اثرات فشرده‌سازی شما بر کیفیت تصویر. مقایسه تصویر فشرده‌سازی شده با تصویر اصلی از نظر اندازه و کیفیت.

۴-۲ تحویل‌ها

۱. فایل‌های کد پایتون (.py) با کامنت‌های واضحی که هر بخش از کد را توضیح می‌دهد.

۲. تصاویر اصلی، فشرده‌سازی شده و بازسازی شده برای مقایسه.

۵-۲ نکات

۱. از کتابخانه‌هایی مانند scipy, numpy یا OpenCV برای پیاده‌سازی تبدیل فوریه استفاده کنید.

۲. برای سادگی، از یک تصویر خاکستری استفاده کنید، زیرا تصاویر رنگی سطح دیگری از پیچیدگی را اضافه می‌کنند.

۳. با درصد‌های مختلف حذف مؤلفه‌های فرکانسی آزمایش کنید تا اثر آن را بر کیفیت تصویر مشاهده کنید.

۶-۲ معیارهای ارزیابی

۱. صحت پیاده‌سازی تبدیل فوریه و تبدیل معکوس.

۲. کارایی تکنیک فشرده‌سازی تصویر (چقدر خوب اندازه را کاهش می‌دهد در حالی که کیفیت قابل قبول تصویر را حفظ می‌کند).

۳. وضوح و جامعیت گزارش نهایی: توضیح تبدیل فوریه، استراتژی فشرده‌سازی تصویر، و بحث درباره نتایج.

۴. کیفیت کد: ساختار، خوانایی، و کامنت‌ها.

٧-٢ مراجع

- فهم تبدیل فوریه

۳ بخش‌بندی تصویر با استفاده از الگوریتم Watershed

۱-۳ مرور کلی:

در این تکلیف، شما باید الگوریتم Watershed را برای بخش‌بندی تصویر اعمال کنید. الگوریتم Watershed یک الگوریتم کلاسیک بخش‌بندی تصویر است که برای جداسازی اشیاء مختلف در یک تصویر استفاده می‌شود.

۲-۳ اهداف آموزشی

۱. درک اصول و نظریه پشت الگوریتم Watershed.

۲. یادگیری کاربرد الگوریتم Watershed برای وظایف بخش‌بندی تصویر.

۳. تحلیل اثرات الگوریتم بر روی انواع مختلف تصاویر.

۳-۳ وظایف

۱. مطالعه الگوریتم Watershed : درک نظریه و اصل کاری پشت الگوریتم Watershed.

۲. پیش‌پردازش تصویر: پیش‌پردازش تصویر ورودی برای نتایج بهتر. این ممکن است شامل حذف نویز، تک کاناله کردن تصویر، آستانه‌گذاری و غیره باشد.

۳. اعمال الگوریتم Watershed : استفاده از الگوریتم Watershed برای وظایف بخش‌بندی تصویر. شما می‌توانید برای این کار از کتابخانه‌هایی مانند OpenCV استفاده کنید.

۴. پسا-پردازش نتیجه: پردازش تصویر خروجی برای بهتر دیدن بخش‌ها. این ممکن است شامل مارک کردن مرزهای هر بخش با رنگ متفاوت و غیره باشد.

۵. ارزیابی و مقایسه: ارزیابی نتایج به دست آمده با الگوریتم Watershed . مقایسه خروجی با تصویر اصلی.

۴-۳ تحویل‌ها

۱. گزارشی که در آن توضیح می‌دهید که الگوریتم Watershed را چگونه فهمیدید، چگونه آن را پیاده‌سازی کردید و مشاهدات خود.
۲. فایل‌های کد پایتون (.py) با نظرات واضح که عملکرد هر بخش از کد را توضیح می‌دهد.
۳. تصاویر اصلی و بخش‌بندی شده برای مقایسه.

۵-۳ نکات

۱. OpenCV یک تابع داخلی برای الگوریتم Watershed دارد که می‌توان برای این تکلیف استفاده کرد.
۲. به یاد داشته باشید که تصویر را به درستی پیش‌پردازش کنید زیرا می‌تواند بر نتایج تأثیر قابل توجهی داشته باشد.
۳. سعی کنید الگوریتم را بر روی انواع مختلف تصاویر استفاده کنید تا ببینید چگونه عمل می‌کند.

۶-۳ معیارهای ارزیابی

۱. درک الگوریتم Watershed و پیاده‌سازی آن.
۲. کیفیت بخش‌بندی تصویر - چقدر خوب بخش‌های مختلف جدا شده‌اند.
۳. وضوح و دقت گزارش نهایی.
۴. کیفیت کد - خوانایی، ساختار و نظرات.

۷-۳ مراجع

- OpenCV: بخش‌بندی تصویر با الگوریتم Watershed