تمرین سری دوم – درس یادگیری ژرف

فاطمه براتی 4023245503

۱- برای آموزش یک شبکه GAN میتوانستیم روش گرادیان کاهشی را روی تابع خطای قسمت generator اعمال کنیم. ولی به جای آن تابع خطا را تغیر دادیم و روش گرادیان افزایشی را روی تابع تغییر یافته اعمال کردیم. این تغیر چه بود و دلیل این کار را توضیح دهید.

در آموزش شبکه‌های مولد متخاصم(GANs) ، می‌توان از روش گرادیان کاهشی برای بهینه‌سازی تابع خطای بخش مولد استفاده کرد. با این حال، بهینه‌سازی مستقیم تابع خطای مولد به دلیل مشکل vanishing gradients دشوار است. این مشکل زمانی رخ می‌دهد که گرادیان‌های مورد استفاده برای به‌روزرسانی وزن‌های شبکه مولد در طی فرآیند backpropagation بسیار کوچک می‌شوند. در نتیجه، به‌روزرسانی‌های وزن‌ها ناچیز می‌شوند و شبکه مولد در یادگیری دچار مشکل می‌شود.

برای حل مشکل vanishing gradients، GANs از تکنیکی به نام gradient ascent بر روی تابع خطای discriminator استفاده می‌کند. به جای بهینه‌سازی مستقیم تابع خطای مولد، مولد را آموزش می‌دهیم تا خطای discriminator را به حداکثر برساند. این رویکرد به این دلیل کار می‌کند که خطای discriminator به طور ذاتی با توانایی مولد در تولید داده‌های واقعی مرتبط است. هنگامی که مولد داده‌های واقعی تولید می‌کند، خطای discriminator افزایش می‌یابد. با به حداکثر رساندن خطای تشخیص‌دهنده، مولد به طور غیرمستقیم یاد می‌گیرد که داده‌های واقعی‌تری تولید کند.

دلایل این تغییر:

اجتناب از مشکل vanishing gradients که مانع از فرآیند آموزش می‌شود.

آموزش غیرمستقیم مولد برای تولید داده‌های واقعی‌تر با به حداکثر رساندن خطای discriminator.

۲- آیا استفاده از batch-normalization میتواند باعث افزایش سرعت یادگیری شبکه شود؟ توضیح دهید.

بله، استفاده از Batch Normalization می‌تواند سرعت یادگیری در شبکه را افزایش دهد. Batch Normalization تکنیکی است که به تثبیت فرآیند آموزش شبکه‌های عصبی عمیق کمک می‌کند. این تکنیک به مشکلinternal covariate shift می‌پردازد که می‌تواند در حین آموزش رخ دهد، زمانی که توزیع activation های یک لایه بین mini-batches تغییر می‌کند. این تغییر می‌تواند یادگیری را برای شبکه دشوار کند، زیرا وزن‌ها و بایاس‌ها دائماً با توجه به توزیع متغیر ورودی‌ها تنظیم می‌شوند.

Batch Normalization با نرمال‌سازی activation های هر لایه در یک mini-batch کار می‌کند. این نرمال‌سازی معمولاً شامل کم کردن میانگین activation متعلق به mini-batch از هر activation و سپس تقسیم بر انحراف معیار است. Activation های نرمال‌شده سپس با استفاده از پارامترهای قابل یادگیری مقیاس‌بندی و جابجا می‌شوند.

با نرمال‌سازی activation ها، Batch Normalization به اطمینان از ثابت ماندن توزیع ورودی‌ها به هر لایه در طول فرآیند آموزش کمک می‌کند. این ثبات می‌تواند منجر به چندین مزیت شود، از جمله:

همگرایی سریع‌تر: شبکه می‌تواند سریع‌تر یاد بگیرد زیرا گرادیان‌ها قابل اعتمادتر و آموزنده‌تر هستند.

دقت بهبود یافته: شبکه ممکن است در مجموعه تست نهایی به دقت بهتری دست یابد.

کاهش نیاز به تنظیم نرخ یادگیری: Batch Normalization می‌تواند شبکه را نسبت به نرخ یادگیری که می‌تواند یک ابرپارامتر دشوار برای تنظیم باشد، کم‌حساس‌تر کند.

همچنین Batch Normalization می‌تواند مقداری سربار محاسباتی به فرآیند آموزش اضافه کند و می‌تواند به اندازه mini-batch حساس باشد. Batch Normalization در همه شبکه‌ها به طور تضمینی باعث بهبود نمی‌شود.

به طور کلی، Batch Normalization یک تکنیک قدرتمند است که می‌تواند فرآیند آموزش شبکه‌های عصبی عمیق را به طور قابل توجهی بهبود بخشد.

۳- دیتاست MNIST در قالب فایل CSV را میتوانید از آدرس زیر دانلود کنید:

https://pjreddie.com/projects/mnist-in-csv/

داده های train و نیز test را از آدرس بالا دانلود کنید. سپس یک کلاس در پایتون با نا م myMNIST بنویسید که داده های MNIST را در قالب CSV لود کند. سپس از این کلاس در کد مربوط به CNN که در اختیارتان هست (torchTutorial-CNN.py) استفاده کنید و پس از اجرای آن و اطمینان از درستی کد، کد را به عنوان پاسخ این سوال ارسال کنید.

پاسخ در نوت بوک زیر نوشته شده است:

https://colab.research.google.com/drive/13ac0PopJz3SicwJThzzHdyE6dP2AUMHW?usp=sharing

۴- در آدرس زیر میتوانید کد Pytorch مربوط به یک AutoEncoder کانولوشنی را ببینید.

https://debuggercafe.com/machine-learning-hands-on-convolutional-autoencoders/

Encoder و Decoder این شبکه هر کدام شامل دو لایه کانولوشنی هستند. لایه ConvTranspose2d که در کد استفاده شده، همان عملیات کانولوشن را انجام میدهد ولی بر خلاف convolution عرض و ارتفاع داده ورودی را افزایش میدهد.

داده های ورودی به این کد از دیتاست CIFAR10 هستند. کد فوق را اجرا کنید و نتیجه را ببینید. سپس، تصاویر دیتاست را با نویز گوسی با میانگین صفر و واریانس σ^2 نویزی کنید و عملکرد شبکه را در کاهش نویز مشاهده کنید. برای این منظور، ورودی شبکه باید تصویر نویزی باشد و سپس در تابع train، خروجی شبکه باید با تصویر بدون نویز مقایسه شود. کد شما باید واریانس نویز را به عنوان ورودی بگیرد. در پایان، ۱۰ تصویر بدون نویز، نویزی شده آن ۱۰ تصویر، و نیز خروجی شبکه AutoEncoder به ازای این تصاویر نویزی را نشان دهید.

پاسخ در نوت بوک زیر نوشته شده است:

https://colab.research.google.com/drive/1hELo-lFTCxE6RiosryWpmd1GWl1fEnRn?usp=sharing