

دانشکده مهندسی کامپیوتر

Faculty of Computer Engineering

دانشگاه اصفهان



مجموعه تمرینات درس هوش محاسباتی

مبحث شبکه های عصبی مصنوعی:

تمرین پیاده سازی شبکه عصبی از صفر با پایتون (مبتدی تا پیشرفته)

استاد: دکتر حسین کارشناس

دستیاران آموزشی:

رضا برزگر

علی شاه زمانی

آرمان خلیلی

نیمسال دوم ۱۴۰۳-۱۴۰۴

تمرین پیاده‌سازی شبکه عصبی از صفر با پایتون (مبتدی تا پیشرفته)

هدف این تمرین ساخت و آموزش یک شبکه عصبی از پایه (بدون استفاده از کتابخانه های آماده) برای دسته‌بندی تصاویر رنگی از دیتاست CIFAR-10 می‌باشد. این تمرین شامل سه سطح از پیاده‌سازی است که از یک مسئله ساده طبقه‌بندی دودویی شروع شده و به یک شبکه عصبی سه‌لایه با softmax برای ۱۰ کلاس ختم می‌شود. در انتها، در بخش چهارم، چالش‌های پیاده‌سازی پیشرفته‌تری برای کاوش و بهبود مدل ارائه شده است.

الزامات تحویل:




یک فایل zip یا لینک GitHub حاوی:

- ipynb برای هر بخش.
- یک فایل pdf. که رویکرد، چالش‌ها و نتایج شما را توضیح می‌دهد.


بخش ۱: رگرسیون لجستیک - آیا تصویر یک هواپیماست؟

در این بخش می‌بایست یک طبقه‌بندی‌کننده تک نورونی بسازید که پیش‌بینی کند آیا یک تصویر داده شده متعلق به کلاس "هواپیما" (کلاس 0) است یا خیر (همه کلاس‌های دیگر - طبقه‌بندی دودویی).

وظایف:

۱. بارگذاری و نرمال‌سازی دیتاست CIFAR-10.
۲. بازبرچسب‌گذاری داده‌ها: اگر تصویر متعلق به کلاس "هواپیما" (برچسب 0) باشد، $y = 1$ ، در غیر این صورت $y = 0$.
۳. مسطح کردن (flatten) هر تصویر به شکل (۳۰۷۲،).
۴. پیاده‌سازی توابع زیر:
 $\text{sigmoid}(z)$: تابع فعال‌سازی سیگموئید. 
 $\text{binary_cross_entropy_loss}(y_true, y_pred)$: تابع هزینه ی کراس-انترپی دودویی. 
۵. پیاده‌سازی آموزش مدل با استفاده از گرادیان کاهشی (Gradient Descent):
مقداردهی اولیه وزن‌ها (W) و بایاس (b). 

$$W = W - lr * dW$$

قانون بروزرسانی: 

۶. مدل خود را بر روی مجموعه داده آزمون با استفاده از confusion matrix و F1-score ارزیابی کنید.

بخش ۲: شبکه دودویی با یک لایه پنهان

مدل خود را گسترش دهید تا شامل یک لایه پنهان با ۶۴ نورون با استفاده از تابع فعال‌سازی سیگموئید شود. وظیفه دودویی ("آیا هواپیما است یا خیر") را حفظ کنید.

وظایف:

۱. معماری شبکه خود را برای شامل شدن یک لایه پنهان با ۶۴ نورون تغییر دهید.

۲. "forward pass" را پیاده‌سازی کنید:

$$\begin{aligned} Z1 &= W1 \cdot X + b1 & A1 &= \text{sigmoid}(Z1) \\ Z2 &= W2 \cdot A1 + b2 & A2 &= \text{sigmoid}(Z2) \end{aligned}$$

۳. "backpropagation" را برای محاسبه گرادیان‌های زیان نسبت به تمام وزن‌ها و بایاس‌ها پیاده‌سازی کنید. وزن‌ها و بایاس‌ها را با استفاده از گرادیان کاهشی به‌روزرسانی کنید.

۴. مدل خود را آموزش دهید و همانند قبل ارزیابی کنید.

بخش ۳: طبقه‌بندی‌کننده چند کلاسه

شبکه خود را ارتقا دهید تا طبقه‌بندی ۱۰ کلاسه را برای تمام دسته‌های موجود در مجموعه داده CIFAR-10 با استفاده از فعال‌سازی خروجی softmax و تابع زیان آنتروپی متقاطع انجام دهد.

وظایف:

۱. وان-هات کردن (One-hot encode) برچسب‌ها برای تمام ۱۰ کلاس در دیتاست.

۲. اصلاح لایه خروجی از ۱ نورون به ۱۰ نورون، یکی برای هر کلاس. همچنین تابع فعال‌سازی سیگموئید در لایه خروجی را با تابع $\text{softmax}(Z)$ جایگزین کنید.

۳. تابع زیان آنتروپی متقاطع چند کلاسه را پیاده‌سازی کنید.

۴. الگوریتم "backpropagation" را برای محاسبه گرادیان‌ها برای سناریوی چند کلاسه با خروجی softmax و زیان آنروپی متقاطع تطبیق دهید.

۵. مدل خود را با استفاده از گرادیان کاهشی آموزش دهید و با استفاده از گزارش طبقه‌بندی (Classification Report) شامل recall, precision و امتیاز F1 برای هر کلاس ارزیابی کنید.

بخش چهارم: چالش پیاده‌سازی نیمه‌پیشرفته

این بخش شما را به چالش می‌کشد تا با پیاده‌سازی تکنیک‌های پیشرفته‌تر، عملکرد و درک خود از شبکه‌های عصبی را عمیق‌تر کنید.

وظایف:

۱. الگوریتم بهینه‌ساز مومنتوم (momentum optimizer) را پیاده‌سازی کنید و سرعت همگرایی آن را با گرادیان کاهشی استاندارد برای بخش ۳ مقایسه کنید.

۲. نمودارهای زیان آموزش و دقت را در طول دوره‌ها (epochs) برای بخش ۳ رسم کنید. در صورت پیاده‌سازی تقسیم اعتبارسنجی (cross-validation split)، نمودارهای زیان و دقت اعتبارسنجی را نیز در نظر بگیرید.

۳. از تابع فعال‌سازی ReLU در لایه(های) پنهان شبکه خود در بخش ۳ استفاده کنید و عملکرد آن را با استفاده از سیگموئید مقایسه کنید. توجه داشته باشید که باید مقادیر اولیه وزن‌ها را متناسب با آن (مثلاً با He Initialization) تنظیم کنید.

۴. (اختیاری) یک کلاس مدولار NeuralNetwork() در بخش ۳ بسازید که امکانات زیر را فراهم کند:
تعیین تعداد دلخواه لایه با اندازه‌های آن‌ها.

انتخاب توابع فعال‌سازی مختلف برای هر لایه (به عنوان مثال، tanh, ReLU, sigmoid).

پیاده‌سازی بهینه‌سازهای مختلف (به عنوان مثال، SGD, momentum, Adam).

۵. (اختیاری) مدل طبقه‌بندی خود با انتخاب موارد مختلف بالا آموزش دهید و نتایج آن‌ها را با یکدیگر مقایسه کنید.

بخش پنجم: پیاده‌سازی CNN با استفاده از کتابخانه TensorFlow/Torch

با تکیه بر تمرین‌های قبلی که به تدریج یک شبکه عصبی را برای دسته‌بندی CIFAR-10 توسعه داده‌اید، این تمرین شما را به چالش می‌کشد تا یک شبکه عصبی کانولوشنال (CNN) را برای بهبود عملکرد پیاده‌سازی کنید. CNN ها به دلیل توانایی خود در بهره‌برداری از سلسله مراتب فضایی، به ویژه برای داده‌های تصویر مناسب هستند.

در این تمرین، هدف اصلی، پیاده‌سازی یک CNN برای دسته بندی تصاویر CIFAR-10 با استفاده از TensorFlow یا PyTorch است. این شامل درک و پیاده‌سازی اجزای اساسی CNN مانند لایه‌های کانولوشن و لایه‌های Pooling، طراحی یک معماری CNN مناسب برای مجموعه داده CIFAR-10، و مقایسه عملکرد CNN با یک پرسپترون چند لایه است که قبلاً پیاده‌سازی شده است. انتظار می‌رود که دانشجو بتواند با موفقیت یک مدل CNN بسازد، آن را آموزش دهد و عملکرد آن را به طور موثر ارزیابی کند.

تجزیه و تحلیل و گزارش:

- یک گزارش مختصر در مورد معماری CNN، جزئیات پیاده‌سازی، فرآیند آموزش و نتایج ارزیابی بنویسید.
- مزایا و معایب استفاده از CNN برای این کار را در مقایسه با یک پرسپترون چند لایه مورد بحث قرار دهید.
- (اختیاری) با معماری‌های مختلف CNN، ابرپارامترها یا تکنیک‌های منظم‌سازی (regularization) برای بهبود عملکرد آزمایش کنید.

یک معماری CNN پیشنهادی برای شروع:

- لایه کانولوشن اول: ۳۲ فیلتر، اندازه کرنل 3x3، فعال سازی ReLU.
- Max Pooling لایه اول: اندازه pool 2x2.
- لایه کانولوشن دوم: ۶۴ فیلتر، اندازه کرنل 3x3، فعال سازی ReLU.
- Max Pooling لایه دوم: اندازه pool 2x2.
- Flatten لایه
- لایه Fully Connected اول: ۱۲۸ نورون، فعال سازی ReLU.
- لایه Fully Connected دوم (لایه خروجی): ۱۰ نورون، فعال سازی Softmax.