

گزارش تمرین سری دوم درس یارگیری عمیق

سارا بارونی

۹۸۴۴۳۰۳۱

لودکردن داده‌ی CIFAR۱۰

این دیتاست شامل ۶۰۰۰۰ داده از تصاویر رنگی $۳۲ * ۳۲$ در ۱۰ کلاس مختلف می‌باشد که هر کلاس شامل ۶۰۰۰ تصویر می‌باشد و ۵۰۰۰۰ داده‌ی آموزش و ۱۰۰۰۰ داده‌ی تست داریم. دیتاست شامل ۶ فایل است که هر کدام ۱۰۰۰۰ نمونه را دربرمیگیرد که شامل ۵ فایل batch train و یک فایل batch test می‌باشد داده‌ی تست به طور رندوم از هر کلاس ۱۰۰۰ نمونه را دارد. فایل به صورت pickle شده موجود است و باید تبدیل به باینری شود. هر فایل شامل یک dictionary به شکل زیر می‌باشد:

یک آرایه‌ی ۱۰۰۰۰ * ۳۰۷۲ است که هر ردیف آرایه یک تصویر رنگی $۳۲ * ۳۲$ را نشان میدهد که ۱۰۲۴ تای اول مربوط به کانال red و ۱۰۲۴ تای دوم مربوط به کانال Green و ۱۰۲۴ تای سوم مربوط به کانال blue می‌باشد. میدانیم تصویر $۳۲ * ۳۲$ است بنابراین ۳۲ تای اول مربوط به مقادیر کانال red در ردیف اول تصویر هستند.

برچسب تصاویر به این شکل است که یک لیست ۱۰۰۰۰ تایی است که شامل اعداد در بازه ۰ تا ۹ می‌باشد که هر کدام مشخص میکند تصویر در کدام کلاس قرار دارد همچنین فایل نشان batches.meta می‌دهد هر کدام از اعداد ۰ تا ۹ به کدام کلاس اشاره میکند.

همچنین لازم به ذکر است که بنابر دلایلی که در تمرین قبل هم ذکر شد لازم است تا داده‌ها نرمالایز و یا استاندارد شوند که در این کد یک بار داده‌ها نرمالایز شدند و همچنین با استاندارد کردن داده‌ها عملکرد مدل مورد بررسی قرار گرفت و باتوجه به اینکه استاندارد کردن داده‌ها باعث میشد مدل عملکرد بهتری داشته باشد در نهایت از استاندارد کردن داده‌ها استفاده شد.

معماری شبکه‌ی عمیق

معماری شبکه‌های عمیق به این شکل است که از چند لایه تشکیل شده و هر لایه از تعدادی نورون تشکیل شده است لایه‌ی اول لایه‌ی مربوط به ورودی‌ها می‌باشد تعداد نورون‌های این لایه به تعداد فیچرهای ورودی‌هاست در نتیجه در این شبکه تعداد لایه‌های ورودی برابر با ۳۰۷۲ تا می‌باشد و متناسب با معماری موردنظر می‌توان تعداد لایه‌های پنهان و تعداد نورون‌های موجود در آنها را تعیین کرد لازم به ذکر است که کد مربوط به این تمرین نیز به گونه‌ای نوشته شده که میتوان لایه‌های مختلف با تعداد نورون‌های مختلف انتخاب کرد.

Fedforward

در Fedforward طبق فرمول زیر ورودی در وزن‌های بین لایه‌ی اول و لایه‌ی دوم ضرب می‌شود و با مقدار بایاسی که در نظر گرفته شده جمع میشود و درواقع خروجی این ورودی لایه‌ی دوم می‌باشد که پس از آن که از activation function لایه‌ی دوم عبور می‌کند حاصل آن خروجی لایه‌ی دوم است که سپس در وزن‌های بعدی ضرب می‌شود و به همین شکل ادامه دارد تا به نورون‌های لایه‌ی آخر برسد و درواقع خروجی لایه‌ی آخر خروجی Fedforward می‌باشد.

لازم به ذکر است که پس از محاسبه‌ی خروجی هر لایه مقادیر وزن‌ها و بایاس و ورودی آن لایه در متغیری به نام cache ذخیره شدند زیرا در روند backpropagation که در ادامه توضیح داده می‌شود مورد استفاده قرار می‌گیرند.

محاسبات مربوط به این قسمت در زیر آورده شده است که ۱ شماره‌ی لایه است:

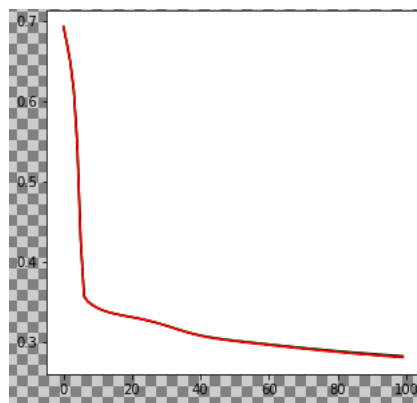
$$w(l)x + b(l) = z(l) \quad (۱)$$

$$activationfunction(z(l)) = a(l) \quad (۲)$$

$$w(l+1)a(l) + b(l+1) = z(l+1) \quad (۳)$$

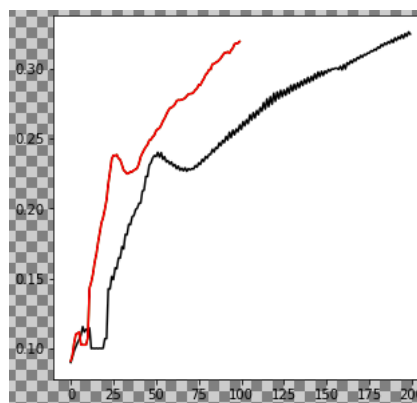
bckpropagation

الگوریتم backpropagation به دلیل روشی که برای بدست آوردن وزن‌ها در آن استفاده می‌شود این گونه نامگذاری شده است. به این شکل که خطا بین خروجی موردانتظار و خروجی بدست آمده محاسبه می‌شود تا با این روش میزان این خطا کیمنه شود به این معنا که مشتق cost function نسبت به وزن‌ها و بایاس‌ها محاسبه می‌شود و مقدار مشتق در لرنینگ ریت ضرب می‌شود و این مقدار از مقدار وزن‌های قبلی کم می‌شود و هربار به این شکل وزن‌ها و بایاس آپدیت می‌شوند. در این تمرین ابتدا با داده‌های آموزش و ارزیابی مدل آموزش داده شد و در بهترین حالت با معماری [۳۰۷۲،۵۰۰،۱۰۰،۱۰] و با learning rate: 0.01 بود که نمودار تابع cost function مربوط به داده‌های آموزش نزولی است.



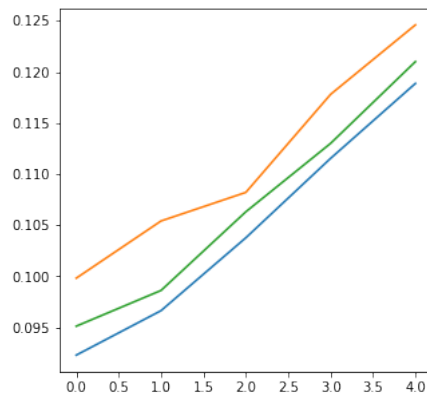
شکل ۱: نمودار تابع خطا

همچنین نمودار دقت رسم شد که تا دقت ۴۰ درصد افزایش یافت

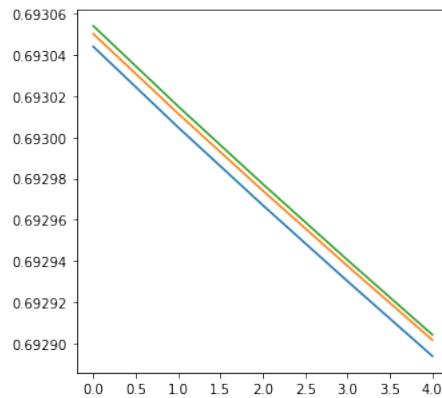


شکل ۲: نمودار مشکی مربوط به دقت داده‌های آموزش میباشد و نمودار قرمز مربوط به دقت داده‌های ارزیابی میباشد.

لازم است که نمودار دقت و هزینه‌ی مربوط به داده‌های validation مورد ارزیابی قرار بگیرد که به دو روش ایستا و پویا انجام شود که براساس نتایج بتوان تصمیم گرفت کدام روش در این دادگان مفیدتر است.



شکل ۳: نمودار دقت داده‌ها آموزش (نارنجی) و ارزیابی (سبز) تست (آبی)



شکل ۴: نمودار تابع خطای داده‌های آموزش و ارزیابی و تست

همچنین از روش k fold cross validation هم استفاده شد این روش از این جهت می‌تواند بهتر باشد که اگر در روش ایستا مثلاً ۵۰۰۰ نمونه‌ی اول داده‌ی آموزش را به عنوان داده‌ی ارزیابی در نظر می‌گیریم در روش k fold cross validation داده‌ی آموزش را به k قسمت مساوی تقسیم می‌کنیم و هر بار یکی از آن k قسمت را به عنوان داده‌ی ارزیابی در نظر می‌گیریم و این باعث می‌شود اگر داده‌ی آموزش در مثلاً ۵۰۰۰ تایی اول از یک کلاس خاص زیاد داشت که در داده‌ی آموزش از آن کم بود در قسمت‌های دیگر که به عنوان داده‌ی ارزیابی در نظر می‌گیریم این مسئله جبران شود زیرا در نهایت از تمامی جواب‌های بدست آمده میانگین می‌گیریم. در این مسئله در روش k fold cross validation نتایج مشابه روش ایستا بود. بهترین دقت بدست آمده برای داده‌های تست ۳۰ درصد بود که در مقایسه با دقت‌های اعلام شده در مراجع دقت نسبتاً خوبی است.