

به نام خدا

دانشکده‌ی مهندسی برق و کامپیوتر

درس هوش مصنوعی

تمرین کامپیوتری شماره ۵

استاد: دکتر فدایی

علی پاکدل صمدی

۸۱۰۱۹۸۳۶۸

هدف پروژه:

هدف از انجام این پروژه آشنایی با `neural network` می باشد، در این پروژه با استفاده از این روش، عکس اعداد ۰ تا ۹ حروف فارسی را می خواهیم طبقه بندی کنیم.

توضیح کلی پروژه:

در این پروژه ما یک فایل برای عکس های حروف ۰ تا ۹ و یک فایل از کلاس آن ها در اختیار داریم. حال ابتدا پیش پردازش کرده و سپس با استفاده از `neural network`، شبکه را آموزش می دهیم و سپس به طبقه بندی داده تست می پردازیم.

بخش اول: بررسی و پیش پردازش داده

۵. به دلیل اینکه وجود یک ویژگی با مقدار بالا، می تواند باعث غالب شدن این ویژگی شده و در شبکه های عصبی پیش بینی ما دقیق نخواهد بود. همچنین از آنجایی که دامنه مقادیر ممکن است زیاد باشد، توابع شاید بدون نرمال سازی به درستی کار نکنند و ممکن است در قسمتی گیر کرده و آموزش صورت نگیرد.

بخش دوم: تکمیل بخش های ناقص شبکه عصبی

در این قسمت ما بخش های `#TODO` را تکمیل می کنیم و `neural network` را پیاده سازی می کنیم.

بخش سوم: طبقه بندی داده ها

قسمت دوم) وزن دهی شبکه: اگر وزن ها صفر دهیم، `derivative` همیشه یکسان باقی می ماند، به همین دلیل باعث می شود شبکه عصبی همیشه یک چیز را آموزش ببیند. مقدار وزن اولیه صفر باعث می شود که نوروں تقریباً در هر تکرار همان توابع را به خاطر بسپارد. این مشکل به عنوان شکست شبکه در شکستن تقارن یاد می شود. این مشکل همچنین هنگامی که وزن ها ثابت باشند نیز رخ می دهد.

قسمت سوم) تاثیر `learning rate`:

`Learning rate` سرعت انطباق مدل با مشکل را کنترل می کند. با توجه به تغییرات کوچک تر در وزن های هر به روز رسانی، نرخ های یادگیری کوچک تر به دوره های آموزشی بیشتری نیاز دارند، در حالی که نرخ های یادگیری بزرگ تر منجر به تغییرات سریع و نیاز به دوره های آموزشی کمتری می شود.

نرخ یادگیری خیلی زیاد می تواند باعث شود که مدل خیلی سریع به یک راه حل غیربهینه همگرا شود. در حالی که نرخ یادگیری بسیار کم می تواند باعث گیر افتادن فرآیند شود. چالش آموزش شبکه های عصبی یادگیری عمیق شامل انتخاب دقیق نرخ یادگیری است. نرخ یادگیری کمتر، خطر $overfitting$ را افزایش می دهد.

همانطور که مشاهده می شود، با نرخ یادگیری زیاد گیر کرده و تغییرات نداریم، در واقع به یک راه حل غیر بهینه رسیده است، و در نرخ یادگیری کم، تغییرات در مراحل آخر کم است و تقریباً در آن قسمت گیر می کند.

قسمت چهارم) تاثیر $activation\ function$:

• Sigmoid

وقتی تابع فعال سازی یک نورون یک تابع $sigmoid$ باشد، تضمینی است که خروجی این واحد همیشه بین 0 و 1 خواهد بود. همچنین، از آنجایی که $sigmoid$ یک تابع غیر خطی است، خروجی این واحد یک تابع غیر خطی خواهد بود. تابع خطی مجموع وزنی ورودی ها. چنین نورونی که از یک تابع $sigmoid$ به عنوان یک تابع فعال سازی استفاده می کند، واحد $sigmoid$ نامیده می شود.

• Hyperbolic Tangent

\tanh نیز مانند $sigmoid$ لاجستیک است اما بهتر است. محدوده تابع \tanh از (-1 تا 1) است. تابع \tanh عمدتاً از طبقه بندی بین دو کلاس استفاده می شود.

• Leaky ReLU

Leaky ReLU (یا LReLU یا LReLU) تابع را طوری تغییر می دهد تا زمانی که ورودی کمتر از صفر است، مقادیر منفی کوچکی را مجاز کند. هنگامی که واحد اشباع شده و فعال نیست، یکسو کننده $leaky$ اجازه می دهد تا یک گرادیان کوچک و غیر صفر ایجاد کند.

دلیل مناسب نبودن $Sigmoid$ و $Tanh$:

توابع فعال سازی $Sigmoid$ و $Tanh$ دارای مشکل $Vanishing\ Gradient$ هستند. بنابراین، مشتقات این توابع بین 0، 0.25 و 1 است. بنابراین، مقادیر وزن به روز شده کوچک هستند، و مقادیر وزن جدید بسیار شبیه به مقادیر وزن قدیمی هستند. در $sigmoid$ دقت خیلی بالا نیست و در آخر تغییرات بسیار اندک هستند.

در \tanh همانطور که مشاهده می شود تغییرات کم است و دقت شیب خیلی کمی دارد، در آخر افزایش دقت با شیب بالا داریم ولی همچنان دقت مناسب نیست.

دلیل برتری Leaky ReLU:

Leaky ReLU دو مزیت دارد: مشکل Dying ReLU را برطرف می کند، زیرا قطعات شیب صفر ندارد.

باعث افزایش سرعت تمرین می شود.

Dying ReLU: وقتی مقدار منفی است، هیچ یادگیری اتفاق نمی افتد زیرا وزن جدید برابر با وزن قبلی باقی می ماند زیرا مقدار مشتق 0 است. زیرا وقتی مقدار مثبت داریم Relu مشتق = 1 را می دهد. همچنین وقتی مقدار منفی داریم Relu مشتق = 0 را در حین انتشار پس زمینه می دهد.

قسمت پنجم) تاثیر batch size:

Batch size تعداد نمونه هایی است که در یک زمان به شبکه ارسال می شود.

Batch size دقت برآورد گرادیان خطا را هنگام آموزش شبکه های عصبی کنترل می کند. Batch، Stochastic، و Minibatch gradient descent سه مزیت اصلی الگوریتم یادگیری هستند. تنش بین اندازه دسته و سرعت و ثبات فرآیند یادگیری وجود دارد.

دلیل دیگری برای اینکه چرا باید استفاده از دسته ای را در نظر بگیرید این است که وقتی مدل یادگیری عمیق خود را بدون تقسیم به دسته ای آموزش می دهید، الگوریتم یادگیری عمیق شما (ممکن است یک شبکه عصبی باشد) باید مقادیر خطا را برای تمام آن 100000 تصویر در حافظه ذخیره کند. باعث کاهش شدید سرعت تمرین خواهد شد. در واقع دلیل استفاده از دسته این است که حافظه بی نهایت نداریم، با استفاده از دسته ها می توانیم حافظه برنامه را به قسمت های مختلف تقسیم کنیم.

مزایای batch size بسیار کوچک:

دسته های کوچک با سرعت بیشتری و با تنوع کمتری از سیستم عبور می کنند که یادگیری سریع تر را تقویت می کند. دلیل سرعت بیشتر واضح است. کاهش تنوع ناشی از تعداد کمتر اقلام در دسته است. دقت معمولاً بالاتر و همگرایی سریعتر می باشد.

معایب batch size بسیار کوچک:

استفاده از batch size کوچک باعث می شود نمونه های ما سوگیری داشته باشند. به همین دلیل دقت می تواند پایین بیاید. ممکن است مدل واگرا شود یا در نقطه ای غیربهبوده همگرا شود.

نتیجه گیری کلی:

در این پروژه با استفاده از neural network توانستیم عکس ها را طبقه بندی کنیم و در این روش دقت بالایی نیز داشتیم. همچنین تاثیر انواع متغیرها را بر روی دقت و دیگر ویژگی ها بررسی کردیم. با انواع توابع activation نیز آشنا شدیم و مزایا و معایب آن ها را نیز نسبت به یکدیگر دیدم.

منابع استفاده شده:

<https://stackoverflow.com>

<https://www.geeksforgeeks.org>

<https://scikit-learn.org/>

<https://vidyasheela.com/>

<https://medium.com/>

<https://www.deeplearning.ai/>

<https://machinelearningmastery.com/>