

باسمه تعالی



یادگیری عمیق

دکتر فاطمی زاده

تمرین دوم

گزارش تمرین عملی

سوال دوم

پوریا دادخواه

401201381

1- Supervised:

در این گزارش کمی توضیح اضافه برای پیاده سازی الگوریتم forward-forward supervised که بر مبنای روش ارائه شده در مقاله پیاده شده می دهیم:

ابتدا توابع و قسمت های ساده را توضیح می دهیم سپس دو کلاس اصلی را با جزئیات بیشتری تشریح می کنیم:

- مجموعه داده و بارگیری کننده ها (لودرها): MNIST

کد مجموعه داده MNIST را بارگیری کرده و بارگیرهای داده برای آموزش و تست ایجاد می کند.

تبدیل هایی اعمال می کند که شامل نرمال سازی و تخت کردن (flattening) است.

- فرآیند آموزش:

نمونه های مثبت (x_{pos}) و منفی (x_{neg}) بر اساس ورودی اصلی (X) و برچسب ها (Y) تولید می شوند.

شبکه با استفاده از الگوریتم آموزشی سفارشی در متد train کلاس Net آموزش داده می شود.

- تصویر سازی:

کد شامل یک تابع (`visualize_sample`) برای تصویر سازی نمونه ها از مجموعه داده است.

- تست و ارزیابی:

پس از آموزش، کد مدل را بر روی مجموعه آموزش و تست ارزیابی می کند و خطای دسته بندی را محاسبه می کند.

همچنین از کتابخانه `torchmetrics` برای محاسبه دقت استفاده می شود.

اکنون به کلاس های اصلی می رسم:

- کلاس (Layer)

کلاس `Layer` از `nn.Linear` ارث‌بری می‌کند و قابلیت‌های اضافی را دربرمی‌گیرد. در ادامه تجزیه و تحلیل شده است:

- `Initialization`:

از `nn.Linear` ارث‌بری می‌کند و یک تابع فعال‌سازی `ReLU` را تنظیم می‌کند.
یک بهینه‌ساز `(self.opt)`` `Adam` با نرخ یادگیری `0.03` را مقداردهی اولیه می‌کند.
یک آستانه `(self.threshold)` و تعداد دوره‌های آموزش `(self.num_epochs)` را تعیین می‌کند.

- `forward method`

گذر به جلو را با یک تابع فعال‌سازی `ReLU` پیاده‌سازی می‌کند.
قبل از محاسبه‌ی ضرب داخلی با ماتریس وزن ترانپاده ، ورودی‌ها را به طول یک نرمال می‌کند.

- `train method`

از یک الگوریتم آموزشی سفارشی برای تعداد مشخصی از دوره‌ها `(self.num_epochs)` استفاده می‌کند.
مقادیر «goodness» `(g_pos)` و `(g_neg)` بر اساس میانگین مربع فعال‌سازی `ReLU` برای نمونه‌های مثبت و منفی محاسبه می‌شود.
یک تابع هزینه سفارشی تعریف می‌کند که نمونه‌هایی را که از یک آستانه انحراف می‌کنند، مجازات می‌دهد.
پارامترهای مدل را با استفاده از بهینه‌ساز `Adam` بهینه‌سازی می‌کند.

- کلاس `Net`

کلاس `Net` کل شبکه عصبی را نمایان می‌کند. اساساً یک مجموعه از نمونه‌های `Layer` است.

`Initialization`

یک لیست از ابعاد `dims` که معماری شبکه را مشخص می‌کند را قبول می‌کند.
یک لیست از نمونه‌های `(self.layers)` `Layer` بر اساس ابعاد ارائه شده را مقداردهی اولیه می‌کند.

Predict method

یک تانسور ورودی x را می‌گیرد و هر برچسب ('0' تا '9') را بر روی 10 پیکسل اول ورودی نقش می‌زند. «goodness» هر برچسب را با گذر از هر لایه و جمع کردن میانگین مربع فعال‌سازی ReLU محاسبه می‌کند. برچسب با بیشترین مقدار «خوبی» برای هر ورودی را باز می‌گرداند.

train method

نمونه‌های مثبت x_pos و منفی x_neg را برای آموزش قبول می‌کند. از طریق هر لایه حرکت کرده و آن را با استفاده از متد train در کلاس Layer آموزش می‌دهد. فرآیند آموزش به ترتیب بر روی هر لایه اعمال می‌شود.

فرآیند آموزش:

1. نمونه‌های مثبت و منفی:

- نمونه‌های مثبت x_pos با نقش آستانه واقعی بر روی ورودی ایجاد می‌شوند.
- نمونه‌های منفی x_neg با تصادفی کردن برچسب‌های واقعی و روی ورودی ایجاد می‌شوند.

2. آموزش ترتیبی لایه:

- شبکه لایه به لایه آموزش داده می‌شود. هر لایه با استفاده از متد train خود آموزش داده می‌شود.

3. تابع هزینه:

- تابع هزینه نمونه‌های مثبت را تشویق می‌کند که مقادیر «goodness» آنها از یک آستانه بیشتر باشد و نمونه‌های منفی را که مقادیر «خوبی» آنها از آستانه کمتر باشد.

4. گذر به عقب:

- گذر به عقب برای هر لایه انجام می‌شود تا پارامترهای مدل را به‌روز کند.

پاسخ سوال مطرح شده راجع به چرایی انتخاب این تابع هزینه:

انتخاب تابع هزینه، همانطور که در کد تعریف شده است، به طور خاص به مفهوم "goodness" که در این الگوریتم مد نظر است، سفارشی شده است. تحلیل می‌کنیم که چرا این تابع هزینه برای شکل‌دهی به فضای ویژگی بر اساس مقادیر "goodness" مناسب است:

1. آستانه‌گذاری

- تابع هزینه یک آستانه معرفی می‌کند که یک مقدار "goodness" هدف را نشان می‌دهد. این آستانه بخش مهمی از الگوریتم است و فرآیند یادگیری را برای شکل دادن فضای ویژگی هدایت می‌کند.

2. هزینه لجستیک:

- هزینه لجستیک $\log(1 + e^x)$ به طور معمول در مسائل طبقه‌بندی دودویی استفاده می‌شود. به طور موثر از انحرافات از آستانه جریمه می‌کند به یک شیوه‌ای که به رگرسیون لجستیک شباهت دارد.

3. مدیریت نمونه‌های مثبت و منفی:

- تابع هزینه به منظور جداگانه اداره نمونه‌های مثبت و نمونه‌های منفی طراحی شده است. این ترغیب به داشتن مقادیر "goodness" بیشتر از آستانه برای نمونه‌های مثبت و مقادیر "goodness" کمتر از آستانه برای نمونه‌های منفی است.

4. تأثیر حاشیه نرم:

- هزینه لجستیک تأثیر حاشیه نرم را معرفی می‌کند، به این معنی که جریمه برای نمونه‌هایی که از آستانه دور هستند به تدریج افزایش می‌یابد. این می‌تواند به عنوان یک مزیت در گرفتن مفهوم "goodness" به یک شیوه انعطاف‌پذیرتر نسبت به یک حاشیه سخت تر مقایسه شود.

5. محاسبه میانگین

- میانگین گیری از هزینه لجستیک بر روی تمام نمونه‌ها مقدار اسکالر یکتا را برای خوبی کل فضای ویژگی فراهم می‌کند. این مقدار اسکالر هدف است که در طول آموزش به حداقل رسانده شود.

به طور کلی، تابع هزینه انتخاب شده هدف الگوریتم را منعکس می‌کند که به شکلی است که نمونه‌های مثبت تحت تأثیر آستانه را دارای مقادیر "goodness" بیشتری کند، در حالی که نمونه‌های منفی تحت تأثیر آستانه را دارای مقادیر "goodness" کمتری کند. این با مفهوم "goodness" که در سیاق الگوریتم تعریف شده است همخوانی دارد.

****توجه:****

- این کد برای محاسبات از GPU (`cuda`) استفاده می‌کند و پارامتریک بر اساس device در دسترس تعریف نشده (متأسفانه وقت اصلاح این اشتباه نشد!)

2- Unsupervised:

- پاسخ سوال مطرح شده راجع به خروجی شبکه و ارتباط با لیبل‌ها:

- خروجی شبکه نتیجه گذر از لایه‌های شبکه عصبی خواهد بود.

- از آنجایی که شما از مفهوم "goodness" استفاده می‌کنیم، خروجی می‌تواند به عنوان ارزیابی شبکه از اینکه چقدر یک نمونه "goodness" یا مناسب است، تفسیر شود.

- شبکه آموزش دیده است که تا حد امکان خوبی نمونه‌های مثبت (که با برچسب‌های صحیح هماهنگ هستند) را بیشینه کند و خوبی نمونه‌های منفی (که با ماسک تولید شده‌اند) را به حداقل برساند.

- در طول آموزش، شبکه یاد می‌گیرد که بر اساس این اندازه‌گیری خوبی بین نمونه‌های مثبت و منفی تفاوت قائل شود.

- پس از آموزش، می‌توانیم از شبکه برای پیش‌بینی خوبی نمونه‌های جدید استفاده کنید. مقادیر خوبی بالا به معنای یک ارزیابی مثبت است و برچسب مرتبط با کلاسی است که خوبی را بیشینه می‌کند.

به طور خلاصه، شبکه یک نمایش یاد می‌گیرد که نمونه‌های مثبت مقادیر خوبی بالا دریافت کنند و نمونه‌های منفی مقادیر خوبی پایین دریافت کنند. این نمایش یادگرفته شده سپس برای پیش‌بینی بر اساس مفهوم خوبی استفاده می‌شود.

- پاسخ سوال راجع به لزوم استفاده از طبقه‌بند خطی و آموزش آن:

در این فرآیند، معرفی یک طبقه‌بند خطی یک هدف خاص در چارچوب گسترده‌تر رویکرد یادگیری بدون نظارتی دارد. دلایلی که چرا تعریف و آموزش یک طبقه‌بند خطی در این زمینه لازم است:

1. ارزیابی نمایش‌های یادگرفته‌شده به صورت نظارت‌شده:

- طبقه‌بند خطی ابزاری فراهم می‌کند تا کیفیت نمایش‌های یادگرفته‌شده را با ارزیابی کارایی آن‌ها در یک وظیفه طبقه‌بندی نظارت‌شده ارزیابی کنید.

- در حالی که شبکه FF به صورت یادگیری بدون نظارت برای differentiating بین نمونه‌های مثبت و منفی بر اساس "goodness" آموزش می‌بیند، طبقه‌بند خطی یک مرحله ارزیابی نظارت‌شده را معرفی می‌کند.

- با آموزش یک طبقه‌بند خطی بر روی ویژگی‌های یادگرفته‌شده، ارزیابی می‌کنید که نمایش‌های یادگرفته‌شده چقدر اطلاعات مرتبط با برچسب‌های کلاس واقعی را گرفته‌اند.

2. ارتباط با برچسب‌های واقعی

- در یک سناریوی یادگیری بدون نظارت معمولی، مدل نمایش‌ها را بدون دانسته‌شدن صریح برچسب کلاس یاد می‌گیرد.

- اما طبقه‌بند خطی فاصله را بین یادگیری بدون نظارت و برچسب‌های واقعی می‌پل کند. این امکان را فراهم می‌کند تا ویژگی‌های یادگرفته‌شده را به برچسب‌های کلاس واقعی متصل کنید.

- این مرحله اطمینان حاصل می‌کند که نمایش‌های یادگرفته‌شده اطلاعات مفیدی را برای تمایز بین کلاس‌های مختلف به خوبی ضبط کرده‌اند و این اطلاعات قابل تفسیر و احتمالاً قابل انتقال به وظایف پسین است.

3. تقویت عملکرد عمومی:

- طبقه‌بند خطی تبدیلی خطی بر روی ویژگی‌های یادگرفته‌شده اعمال می‌کند تا برچسب‌های کلاس را پیش‌بینی کند. این تبدیل خطی با یادگیری یک مرز تصمیم ساده در فضای ویژگی، تشویق به عملکرد عمومی می‌کند.

- یک طبقه‌بند خطی یک مدل ساده و قابل تفسیر است که به ارزیابی قابلیت‌های عمومی ویژگی‌های یادگرفته‌شده کمک می‌کند.

4. تنظیم و بهبود پس‌زمینه:

- آموزش یک طبقه‌بند خطی امکان تنظیم نمایش‌های یادگرفته‌شده را با توجه به برچسب‌های خاص فراهم می‌کند.

- این امکان را فراهم می‌کند تا ویژگی‌ها را بهبود بخشیده و آن‌ها را بهتر با ساختار اساسی داده هماهنگ کنید و عملکرد طبقه‌بندی را بهبود ببخشید.

به طور خلاصه، طبقه‌بند خطی به عنوان یک جزء حیاتی برای ارزیابی فرآیند یادگیری بدون نظارت عمل می‌کند. این ارتباط را بین نمایش‌های بدون نظارت و برچسب‌های واقعی برقرار می‌کند، یک اندازه روشن و قابل تفسیر از کیفیت ویژگی‌های یادگرفته شده و قابلیت آن‌ها برای گرفتن اطلاعات مرتبط با وظایف طبقه‌بندی ارائه می‌دهد.

همچنین نحوه پیاده‌سازی و سایر الگوریتم‌های دیگر مورد استفاده مشابه supervised بوده و نیاز به توضیح مجدد ندارد (برخی تفاوت‌ها هم در کد کامنت گذاری شده است)