

درس یادگیری ماشین

گزارش تکلیف Backpropagation

استاد درس:

دکتر افتخاری

نگارش:

امیرحسین ابوالحسنی

شماره دانشجویی: ۴۰۰۴۰۵۰۰۳

پاییز ۱۴۰۳

فهرست مطالب

۱	مقدمه	۲
۲	پیاده سازی لایه خطی	۲
۳	پیاده سازی توابع فعال سازی	۲
۱.۳	تابع سیگموید	۲
۲.۳	تابع واحد یک سو شده خطی	۳
۳.۳	تابع Softmax	۳
۴	پیاده سازی توابع هزینه	۳
۵	پیاده سازی شبکه	۳
۶	آموزش شبکه	۳
۷	ارزیابی و نتایج	۴

۱ مقدمه

الگوریتم پس انتشار خطا^۱، الگوریتمی برای یادگیری با نظارت در شبکه‌های عصبی با استفاده از گرادیان کاهشی است. در این روش، برای یک شبکه عصبی مصنوعی و تابع خطای مشخص، گرادیان تابع خطا نسبت به وزن‌های شبکه عصبی محاسبه می‌شود. در این تکلیف به پیاده سازی بلوک‌های سازنده یک شبکه عصبی پراخته می‌شود، و در هر بلوک، متدهای مورد نیاز برای انجام الگوریتم پس انتشار خطا پیاده سازی می‌شود.

۲ پیاده سازی لایه خطی

هر لایه از شبکه عصبی متشکل از تعدادی نورون می باشد که تعداد بعد ورودی را به تعداد بعد خروجی نگاشت می کند. پارامترهای مهمی که باید برای هر لایه ذخیره شود وزن‌های لایه و بایاس لایه می‌باشد. همچنین گرادیان‌ها نسبت به وزن و بایاس نیز باید نگه داشته شود. برای هر لایه خطی سه متد در نظر گرفته شده است:

- forward: متدی که وظیفه انجام عمل پیشخور لایه را بر عهده دارد.
- backward: متدی که وظیفه محاسبه گرادیان نسبت به وزن های این لایه را دارد.
- step: متدی که وظیفه بروزرسانی وزن‌ها با توجه به گرادیان محاسبه شده را دارد.

۳ پیاده سازی توابع فعال سازی

برای پیاده سازی توابع فعال ساز، به آنها به شکل یک لایه نگاه کرده می‌شود. کلاس هر کدام از توابع فعال ساز متدهای زیر را پیاده‌سازی می‌کند.

- forward: وظیفه این تابع محاسبه انجام عملیات تابع روی ورودی‌ها می‌باشد.
- backward: محاسبه گرادیان خطا نسبت به ورودی

۱.۳ تابع سیگموید

تابع سیگموید^۲ به فرمول :

$$f(x) = \frac{1}{1 + e^{-x}}$$

ورودی را به بازه $[0, 1]$ نگاشت می‌کند و برای انجام دسته‌بندی دو کلاسه مورد استفاده قرار می‌گیرد. یکی دیگر از دلایل استفاده از سیگموید، سادگی در محاسبه مشتق آن است.

$$f'(x) = f(x) \cdot (1 - f(x))$$

۲.۳ تابع واحد یک سو شده خطی

تابع ReLU با فرمول

$$f(x) = \max(0, x)$$

سعی در ایجاد روابط غیر خطی در شبکه دارد. همچنین مشتق این تابع به سادگی محاسبه می گردد:

$$f'(x) = \begin{cases} 1 & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

۳.۳ Softmax تابع

تابع Softmax به فرمول:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

به طور عمده برای دسته بندی با بیش از دو کلاس مورد استفاده قرار می گیرد.

۴ پیاده سازی توابع هزینه

برای پیاده سازی هر تابع هزینه مقادیر y و \hat{y} در کلاس، نگهداری می شود. همچنین متدهای زیر در هر کلاس پیاده سازی می شود:

- forward : این متد ارور را با توجه به رابطه تعریف شده برای تابع هزینه محاسبه می کند.
- backward : این متد مشتق تابع هزینه نسبت به \hat{y} را محاسبه می کند.

۵ پیاده سازی شبکه

یک کلاس برای یک شبکه پرسپترون چند لایه^۱ نوشته می شود که در آن حلقه آموزش و متدهای مورد نیاز برای پیشخور و پس انتشار خطا پیاده سازی می گردد.

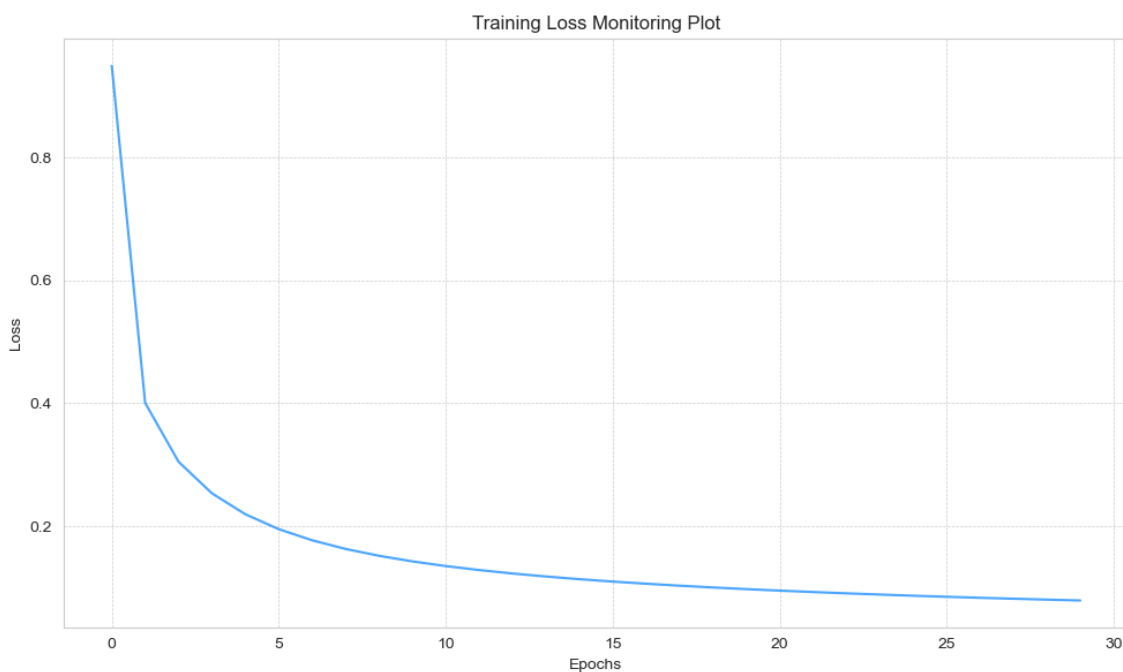
۶ آموزش شبکه

شبکه MLP پیاده سازی شده، بر روی دیتاست Fashion MNIST و کلاس های T-shirt/top و Trouser و Sneaker آموزش دیده است. کانفیگ آموزش در جدول ۱ نشان داده شده است.

نوع	مقدار
تعداد لایه ها	۳
Epoch	۳۰
نرخ یادگیری	10^{-3}
سایز Batch	۶۴
تابع هزینه	Cross Entropy

جدول ۱: کانفیگ آموزش شبکه

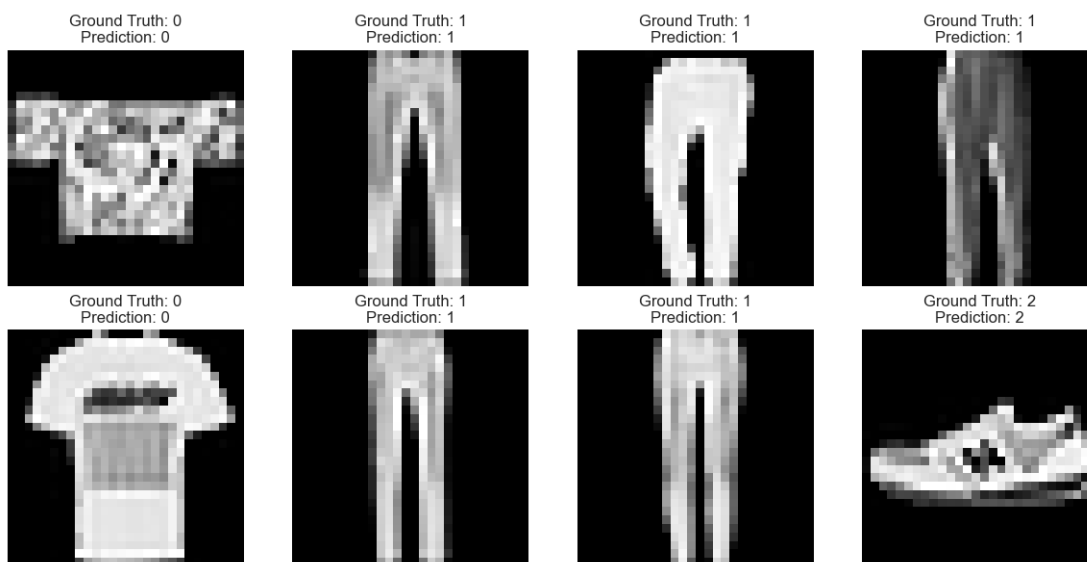
MLP^۱



شکل ۱: نمودار روند خطای آموزش در طول یادگیری

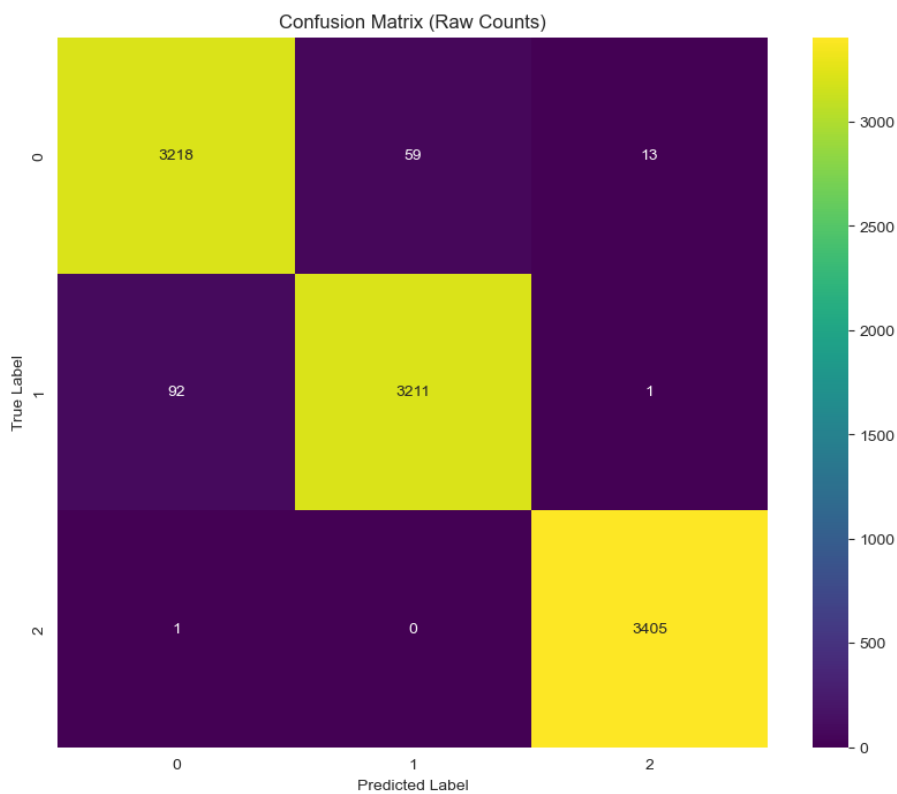
۷ ارزیابی و نتایج

مدل پس از آموزش به دقت ۹۸٪ رسید. همچنین نمونه‌ای از خروجی مدل در شکل ۲ مشاهده می‌شود.



شکل ۲: نمونه ای از دیتاست و خروجی مدل

با بررسی ماتریس سردرگمی^۱ (شکل ۳) متوجه می‌شویم که مدل به خوبی توانسته هر سه کلاس را یادگیری کرده و داده‌های تست را به خوبی دسته بندی کند. از این موضوع می‌توان نتیجه گرفت که پیاده سازی هایی که انجام شده صحیح بوده و مشکلی در آنها وجود نداشته است.



شکل ۳: ماتریس سردرگمی