به نام خدا

معصومه پاسبانی 9243022

تمرین سری چهارم مبانی یادگیری ماشین

سوال اول)

الف) ناپدید شدن گرادیان‌ها زمانی اتفاق می‌افتد که طی فرآیند Backpropagation، مقدار گرادیان وزن‌ها در لایه‌های اولیه (نزدیک به ورودی) بسیار کوچک می‌شود. دلایل اصلی این پدیده عبارتند از:

1. بیشتر توابع فعال‌سازی رایج مشتقات کوچکی دارند، به خصوص زمانی که ورودی‌های آن‌ها در بازه‌های اشباع (خیلی بزرگ یا خیلی کوچک) قرار بگیرند. در نتیجه، وقتی گرادیان خطا از لایه‌های انتهایی به لایه‌های اولیه منتقل می‌شود، مقدار گرادیان به تدریج کوچک‌تر شده و به صفر نزدیک می‌شود.
2. در فرآیند پس‌انتشار، گرادیان هر لایه با مشتقات لایه‌های قبلی ضرب می‌شود. در شبکه‌های عمیق، این ضرب‌های مکرر منجر به کوچک شدن سریع مقادیر گرادیان در لایه‌های اولیه می‌شود. به عبارتی، اثر زنجیره‌ای باعث کاهش نمایی گرادیان‌ها می‌شود.

در شبکه‌های عمیق، لایه‌های ابتدایی مسئول یادگیری ویژگی‌های پایه‌ای هستند. با ناپدید شدن گرادیان‌ها، وزن‌های این لایه‌ها به‌روزرسانی نمی‌شوند یا به‌روزرسانی بسیار ناچیزی دارند. این امر باعث می‌شود یادگیری ویژگی‌های اولیه مختل شود و در نتیجه، شبکه نتواند ویژگی‌های سلسله‌مراتبی و پیچیده‌تر را در لایه‌های بالاتر یاد بگیرد.وقتی لایه‌های اولیه یادگیری مناسبی نداشته باشند، دقت شبکه در داده‌های آموزش کاهش می‌یابد، زیرا این لایه‌ها اساس یادگیری شبکه را تشکیل می‌دهند.اختلال در یادگیری لایه‌های ابتدایی باعث می‌شود که شبکه نتواند ویژگی‌های تعمیم‌پذیری را بیاموزد، و این مسئله به کاهش عملکرد روی داده‌های تست منجر می‌شود.

برای مقابله با این مشکل و افزایش کارایی شبکه، روش‌های متعددی وجو دارد که عبارتند از:

توابع فعال‌سازی پیشرفته مانند ReLU و نسخه‌های اصلاح‌شده آن مثل Leaky ReLU و Parametric ReLU به دلیل مشتقات غیرصفر در دامنه گسترده‌تری از ورودی، از ناپدید شدن گرادیان جلوگیری می‌کنند.استفاده از تکنیک‌هایی مانند Batch Normalization، مقیاس‌گذاری ورودی‌ها و وزن‌ها در هر لایه را کنترل می‌کند و مانع از کوچک شدن یا بزرگ شدن بیش از حد گرادیان‌ها می‌شود.

برخی معماری‌ها از روش‌هایی مانند Gradient Clipping برای محدود کردن مقدار گرادیان‌های بزرگ یا کوچک استفاده می‌کنند.

ب) در شبکه‌های عصبی، مقداردهی اولیه وزن‌ها اهمیت زیادی دارد، زیرا وزن‌ها پایه‌ای برای یادگیری شبکه هستند. اگر وزن‌های اولیه، از جمله مقادیر فیلترهای پیچشی، صفر مقداردهی شوند؛ خروجی تمام نرون‌های شبکه یکسان خواهد بود که این یکسانی

خروجی منجر به تولید گرادیان‌های مشابه برای تمام نرون‌ها در فرآیند پس‌انتشار می‌شود، در نتیجه، همه وزن‌ها به‌طور یکسان به‌روزرسانی می‌شوند و شبکه نمی‌تواند ویژگی‌های متفاوتی از داده را یاد بگیرد.از آنجا که به‌روزرسانی وزن‌ها به یکسان انجام می‌شود، فیلترهای پیچشی قادر نخواهند بود ویژگی‌های خاص و منحصربه‌فرد را در داده شناسایی کنند. این امر مانع یادگیری شبکه شده و آموزش متوقف می‌شود.

مقداردهیXavier برای شبکه‌هایی که از توابع فعال‌سازی مانند Sigmoid یا Tanhاستفاده می‌کنند، طراحی شده است.هدف این روش، حفظ تعادل در جریان سیگنال فعال‌سازی‌ها و گرادیان‌ها در طول شبکه است.وزن‌های اولیه به‌صورت تصادفی و با استفاده از یک توزیع گاوسی یا یکنواخت مقداردهی می‌شوند .این روش تضمین می‌کند که واریانس خروجی هر لایه نزدیک به مقدار ورودی آن باشد.

Xavier Init (with Tanh as activation): $W \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_j+n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j+n_{j+1}}}\right]$

He Init (with ReLU as activation): $W \sim \mathcal{N}\left(0, \frac{2}{n^l}\right)$

مقداردهیHe برای شبکه‌هایی که از توابع فعال‌سازی ReLUیا مشتقات آن) مانند (Leaky ReLU استفاده می‌کنند، طراحی شده است.توابع ReLU مقادیر منفی را صفر می‌کنند، بنابراین تنها نیمی از خروجی‌ها در هر لایه فعال باقی می‌مانند. برای جبران این کاهش، مقداردهی He واریانس بیشتری به وزن‌ها اختصاص می‌دهداین مقداردهی باعث می‌شود جریان گرادیان در شبکه به خوبی حفظ شود، حتی در شبکه‌های بسیار عمیق.

توابع فعال‌سازی مانند ReLUمقادیر منفی را صفر می‌کنند و تنها مقادیر مثبت را عبور می‌دهند. با گذر از هر لایه، تنها بخشی از نرون‌ها فعال باقی می‌مانند. بنابراین، مقداردهی اولیه باید این کاهش را جبران کند .مقداردهی He با افزایش واریانس وزن‌ها، سیگنال‌ها را در لایه‌های مختلف تقویت می‌کند. این تقویت مانع از کوچک شدن بیش از حد گرادیان‌ها (ناپدید شدن گرادیان) در شبکه‌های عمیق می‌شود.

توابع Sigmoid و Tanh واریانس ورودی را در دو جهت (مثبت و منفی) حفظ می‌کنند، در حالی که ReLU تنها خروجی مثبت دارد. مقداردهی He این تفاوت را در نظر می‌گیرد و واریانس بیشتری اختصاص می‌دهد تا سیگنال به خوبی در شبکه منتشر شود.

مقداردهی He با تنظیم مقادیر اولیه وزن‌ها بر اساس تعداد ورودی‌های هر نرون (تعداد فیلترها در لایه‌های پیچشی)، اثر زنجیره‌ای ضرب گرادیان‌ها را بهبود می‌بخشد. این امر باعث می‌شود، جریان سیگنال در لایه‌های اولیه و انتهایی متعادل شود؛ شبکه بتواند

بدون ناپدید شدن گرادیان، ویژگی‌های سلسله‌مراتبی را یاد بگیرد؛ کارایی شبکه‌های عمیق بهبود یابد، به خصوص در ترکیب با توابع ReLU که برای داده‌های پیچیده و تصاویر مناسب‌تر هستند.

ج) در شبکه‌های بسیار عمیق، گرادیان‌ها در طول پس‌انتشار از طریق زنجیره ضربی توابع فعال‌سازی کوچک‌تر می‌شوند، که باعث ناپدید شدن گرادیان می‌شود.اتصالات میانبر، مسیر مستقیم‌تری برای انتقال گرادیان از لایه‌های انتهایی به لایه‌های ابتدایی فراهم می‌کنند.این مسیر اضافی تضمین می‌کند که حتی اگر گرادیان در لایه‌های پیچشی ضعیف شود، از طریق مسیر میانبر به لایه‌های قبلی منتقل خواهد شد.

در شبکه‌های عمیق استاندارد، افزودن لایه‌های بیشتر ممکن است باعث افزایش خطا در آموزش و تست شود. این مسئله به دلیل ناتوانی شبکه در یادگیری یک Identity Mappingرخ می‌دهد.درResNet ، اتصالات میانبر به شبکه اجازه می‌دهند به راحتی یک نگاشت هویت را پیاده‌سازی کند، زیرا اگر $F(x)=0$ باشد، بلوک Residual به سادگی $y=x$ را بازتولید می‌کند. این امر مانع از کاهش عملکرد در شبکه‌های بسیار عمیق می‌شود.

در معماری‌های استاندارد، سیگنال ورودی با گذر از لایه‌های متعدد ممکن است تغییرات زیادی پیدا کند یا تضعیف شود. اتصال میانبر تضمین می‌کند که سیگنال ورودی بدون تغییر به لایه‌های عمیق‌تر منتقل شود. این کار منجر به پایداری یادگیری می‌شود.

شبکه‌های عمیق باید یاد بگیرند که ویژگی‌های پیچیده‌ای را استخراج کنند. با این حال، در بسیاری از موارد، ویژگی‌های پایین‌دستی که توسط لایه‌های اولیه تولید می‌شوند، نیاز به تغییر ندارند.اتصال میانبر به لایه‌های بعدی اجازه می‌دهد این ویژگی‌ها را مستقیماً دریافت کنند، بدون اینکه نیاز باشد شبکه یاد بگیرد آن‌ها را بازتولید کند.

به جای تلاش برای یادگیری یک نگاشت پیچیده$H(x)$ ، شبکه تنها نیاز دارد تفاوت بین ورودی و خروجی $F(x)=H(x)-x$ را یاد بگیرد. این ساده‌سازی فرآیند یادگیری را تسریع می‌کند.

بدون اتصالات میانبر، افزایش عمق شبکه باعث کاهش عملکرد به دلیل مشکلات گرادیان و یادگیری می‌شود. اما ResNet نشان داد که حتی شبکه‌هایی با بیش از ۱۵۰ لایه می‌توانند به عملکرد بالایی دست یابند.

اتصالات میانبر باعث می‌شوند گرادیان‌ها از لایه‌های انتهایی به لایه‌های اولیه بدون تغییر قابل‌توجه منتقل شوند. این مسئله از مشکلاتی مانند ناپدید شدن یا منفجر شدن گرادیان جلوگیری می‌کند.معماری ResNet با استفاده از بلوک‌های Residual امکان افزایش عمق شبکه را بدون کاهش عملکرد فراهم می‌کند. این معماری به خوبی نشان داده است که افزایش عمق می‌تواند به یادگیری ویژگی‌های سلسله‌مراتبی پیچیده‌تر منجر شود.اتصال میانبر باعث می‌شود شبکه عمیق‌تر سریع‌تر همگرا شود، زیرا جریان گرادیان و سیگنال تثبیت شده است.اتصالات میانبر به شبکه اجازه می‌دهند تا به‌طور مؤثرتری ویژگی‌های مفید را از داده‌ها استخراج کند و این ویژگی‌ها را در سطوح مختلف سلسله‌مراتبی تجمیع کند.

Year: Month: Date: Subject:

سوال دوم:

$$X = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 1 & 2 & 1 \end{bmatrix}$$

$$W = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}$$

ابتدا باید ماتریس را تبدیل می‌کنیم.

$$\partial L/\partial Y = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$S_1 = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix} \quad S_2 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 1 & 2 \end{bmatrix} \quad S_4 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\partial L/\partial W = \partial L/\partial Y_{1,1} \cdot S_1 \Rightarrow \partial L/\partial W = 1 \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}$$

$$\partial L/\partial W = -1 \cdot S_2 \Rightarrow (-1) \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & -2 & -1 \end{bmatrix}$$

$$\partial L/\partial W \Rightarrow 1 \cdot S_3 \Rightarrow (-1) \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & -2 \\ 0 & -1 & -2 \end{bmatrix}$$

$$\partial L/\partial W = 1 \cdot S_4 \Rightarrow 1 \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 2 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 2 & 1 \end{bmatrix}$$

(((Bastan)))

$$\frac{\partial L}{\partial w} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix} + \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & -2 & -1 \end{bmatrix} + \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & -2 \\ 0 & -1 & -2 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\frac{\partial L}{\partial w} = \begin{bmatrix} -2 & 0 & 2 \\ 0 & 3 & -2 \\ 2 & -1 & 0 \end{bmatrix}$$

ب)

$$w = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \xrightarrow{\text{چرخش } 180^\circ} w' = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$$

$$\frac{\partial L}{\partial x} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Patch_{3\times3} \Rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix}$$

حاصل ضرب نقطه‌ای: 3

متناظر با درایه‌ی ورودی در (0,0) = 3

سپس ترتیب برای همه‌ی patch های متناظر از $\frac{\partial L}{\partial x}$ را انتخاب کرده و ضرب نقطه‌ای انجام میدهیم ...

(((Bastan)))

گرادیان $\partial L/\partial X$نشان‌دهنده حساسیت تابع هزینه نسبت به ورودی Xاست. این مقدار می‌تواند به صورت زیر استفاده شود:

1. انتقال گرادیان به لایه‌های قبلی :در شبکه عصبی، اگر Xخروجی یک لایه قبلی باشد، گرادیان $\partial L/\partial X$به لایه قبلی منتقل می‌شود تا وزن‌های آن لایه (از طریق گرادیان کاهشی) به‌روزرسانی شوند.
2. به‌روزرسانی وزن‌ها :وزن‌های هر لایه با استفاده از گرادیانی که از لایه‌های بعدی از طریق $\partial L/\partial X$منتقل می‌شود، به‌روزرسانی می‌شوند:

$$W_{new} = W_{old} - \eta \partial L/\partial W$$

که ηنرخ یادگیری است.

3. انتقال گرادیان به ورودی تصویر (در یادگیری ویژگی) :در برخی کاربردها مانند یادگیری ویژگی در شبکه‌های عصبی کانولوشنی(CNN) ، گرادیان $\partial L/\partial X$می‌تواند برای تحلیل و اصلاح ورودی تصویر (مانند تولید تصاویر متقابل) استفاده شود.

سوال سوم)

یک خروجی نهایی (حالت استاندارد برای تحلیل احساسات جمله):

در این حالت، جمله به‌عنوان یک واحد کلی تحلیل می‌شود. پس از پردازش کامل nnnکلمه در جمله، تنها یک خروجی نهایی از RNN گرفته می‌شود.این خروجی که معمولاً وضعیت مخفی آخرین گام زمانی h_nاست، به یک لایه Dense متصل می‌شود و تابع softmax یا هر تابع فعال‌سازی دیگر روی آن اعمال می‌شود تا پیش‌بینی نهایی تولید شود.

در این حالت :تعداد خروجی‌ها)=1(softmax

یک خروجی در هر گام زمانی

در این حالت، RNNبرای هر کلمه در جمله، یک خروجی تولید می‌کند. بنابراین، تعداد خروجی‌ها برابر با تعداد کلمات در جمله (n) است.

در این حالت تعداد خروجی‌ها (فراخوانیsoftmax)=n

.3حالت ترکیبی (خروجی نهایی یا میانی خاص):

ممکن است مدل فقط در برخی از گام‌های زمانی خروجی تولید کند. برای مثال:

فقط در انتهای جمله (خروجی نهایی، یک بار).

یا در گام‌های خاص مانند کلمات کلیدی مشخص‌شده.

در این حالت تعداد خروجی‌ها به طراحی خاص مدل بستگی دارد و به‌صورت متغیر kمی‌توان آن را بیان کرد

تعداد خروجی‌ها (فرتخوانی softmax) = k

ب) هر y^n، که از طریق تابع softmaxمحاسبه می‌شود، یک توزیع احتمالاتی چندجمله‌ای است که احتمال تعلق نمونه به هر یک از کلاس‌های ممکن را نشان می‌دهد.

اگر مسئله دسته‌بندی احساسات به بازه‌ای مثل [0, 1, 2, 3, 4] باشد:

- y^n یک بردار احتمال خواهد بود که مجموع مقادیر آن برابر با 1 است.
- هر عنصر در این بردار احتمال تعلق جمله به یکی از این کلاس‌ها را نشان می‌دهد.

اگر خروجی برای هر کلمه محاسبه شود، هر y^tتوزیع احتمالی‌ای خواهد بود که احتمال تعلق هر کلمه به دسته‌بندی‌های خاص (مثلاً دسته‌بندی احساسات محلی یا کلمات کلیدی) را نشان می‌دهد.

اگر مسئله‌ای مانند دسته‌بندی گونه سگ‌ها باشد، y ^nمی‌تواند احتمال تعلق یک تصویر به انواع سگ‌ها را نشان دهد.

y^nیک توزیع احتمالاتی چندجمله‌ای (Multinomial Distribution) است.هر مقدار در این توزیع نشان‌دهنده احتمال تعلق به یک کلاس خاص است.

مجموع تمامی مقادیر در توزیع برابر با 1است.

ج) ورودی‌های هر گام زمانی برای تولید خروجی در RNN عبارتند از :

ورودی فعلی :(x_t) بردار تعبیه‌ای مربوط به کلمه فعلی که ویژگی‌های آن را نمایش می‌دهد .

وضعیت مخفی قبلی (h_{t-1}) اطلاعات مربوط به تمام کلمات قبلی پردازش‌شده که به مدل اجازه می‌دهد وابستگی‌های زمانی را یاد بگیرد .

وزن‌های مدل (W_x, W_h) وزن‌هایی که ورودی فعلی و وضعیت مخفی قبلی را ترکیب می‌کنند .