

□ **نرمال سازی دسته‌ای (batch normalization)** – یک مرحله از فراسنج‌های γ, β که دسته‌ی $\{x_i\}$ را نرمال می‌کند. نما μ_B, σ_B^2 به میانگین و واریانس دسته‌ای که می‌خواهیم آن را اصلاح کنیم اشاره دارد که به صورت زیر است:

$$x_i \leftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

معمولا بعد از یک لایه‌ی تمام‌متصل یا لایه‌ی کانولوشنی و قبل از یک لایه‌ی غیرخطی اعمال می‌شود و امکان استفاده از نرخ یادگیری بالاتر را می‌دهد و همچنین باعث می‌شود که وابستگی شدید مدل به مقاداردهی اولیه کاهش یابد.

آموزش یک شبکه‌ی عصبی

□ **تکرار (epoch)** – در مضمون آموزش یک مدل، تکرار اصطلاحی است که مدل در یک دوره تکرار تمامی نمونه‌های آموزشی را برای به‌روزرسانی وزن‌ها می‌بیند.

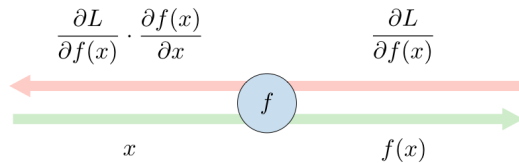
□ **گرادیان نزولی دسته‌ی کوچک (mini-batch gradient descent)** – در فاز آموزش، به‌روزرسانی وزن‌ها معمولا بر مبنای تمامی مجموعه آموزش به علت پیچیدگی‌های محاسباتی، یا یک نمونه داده به علت مشکل نویز، نیست. در عوض، گام به‌روزرسانی بر روی دسته‌های کوچک انجام می‌شود، که تعداد نمونه‌های داده در یک دسته یک ابرفراسنج است که می‌توان آن را تنظیم کرد.

□ **(loss function)** – به منظور سنجش کارایی یک مدل مفروض، معمولا از تابع خطای L برای ارزیابی اینکه تا چه حد خروجی حقیقی y به شکل صحیح توسط خروجی z مدل پیش‌بینی شده‌اند، استفاده می‌شود.

□ **خطای آنتروپی متقاطع (cross-entropy loss)** – در مضمون دسته‌بندی دودویی در شبکه‌های عصبی، عموما از تابع خطای آنتروپی متقاطع $L(z, y)$ استفاده و به صورت زیر تعریف می‌شود:

$$L(z, y) = - \left[y \log(z) + (1 - y) \log(1 - z) \right]$$

□ **انتشار معکوس (backpropagation)** – انتشار معکوس روشی برای به‌روزرسانی وزن‌ها با توجه به خروجی واقعی و خروجی مورد انتظار در شبکه‌ی عصبی است. مشتق نسبت به هر وزن w توسط قاعده‌ی زنجیری محاسبه می‌شود.

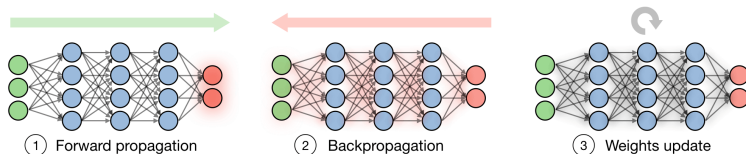


با استفاده از این روش، هر وزن با قانون زیر به‌روزرسانی می‌شود:

$$w \leftarrow w - \alpha \frac{\partial L(z, y)}{\partial w}$$

□ **به‌روزرسانی وزن‌ها** – در یک شبکه‌ی عصبی، وزن‌ها به شکل زیر به‌روزرسانی می‌شوند:

- گام ۱: یک دسته از داده‌های آموزشی گرفته شده و با استفاده از انتشار مستقیم خطا محاسبه می‌شود
- گام ۲: با استفاده از انتشار معکوس مشتق خطا نسبت به هر وزن محاسبه می‌شود
- گام ۳: با استفاده از مشتقات، وزن‌های شبکه به‌روزرسانی می‌شوند



راهنمای کوتاه نکات و ترفندهای یادگیری عمیق

اشرین عمیدی و شروین عمیدی

۱۵ شهریور ۱۳۹۸

ترجمه به فارسی توسط الیستر. بازبینی توسط عرفان نوری.

پردازش داده

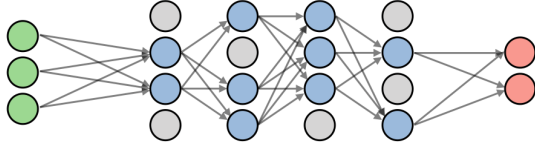
□ **داده‌افزایی (data augmentation)** – مدل‌های یادگیری عمیق معمولا به داده‌های زیادی نیاز دارند تا بتوانند به خوبی آموزش ببینند. اغلب، استفاده از روش‌های داده‌افزایی برای گرفتن داده‌ی بیشتر از داده‌های موجود، مفید است. اصلی‌ترین آنها در جدول زیر به اختصار آمده‌اند. به عبارت دقیق‌تر، با در نظر گرفتن تصویر ورودی زیر، روش‌هایی که می‌توان اعمال کرد بدین شرح هستند:

تصویر اصلی	قرینه	چرخش	برش تصادفی
– تصویر (آغازین) بدون هیچ‌گونه تغییری	– قرینه‌شده نسبت به محوری که معنای (محتوای) تصویر را حفظ می‌کند	– چرخش با زاویه‌ی اندک – خط افق نادرست را شبیه‌سازی می‌کند	– روی ناحیه‌ای تصادفی از تصویر متمرکز می‌شود – چندین برش تصادفی را می‌توان پشت سرهم انجام داد

تغییر رنگ	اضافه کردن نویز	هدررفت اطلاعات	تغییر تایین
– عناصر RGB کمی تغییر کرده است – نویزی که در هنگام مواجه شدن با نور رخ می‌دهد را شبیه‌سازی می‌کند	– افزودگی نویز – مقاومت بیشتر نسبت به تغییر کیفیت تصاویر ورودی	– بخش‌هایی از تصویر نادیده گرفته می‌شوند – تقلید (شبیه سازی) هدررفت بالقوه بخش‌هایی از تصویر	– تغییر درخشندگی – با توجه به زمان روز تفاوت نمایش (تصویر) را کنترل می‌کند

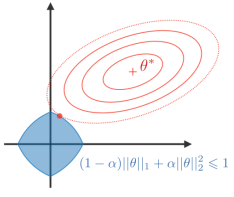
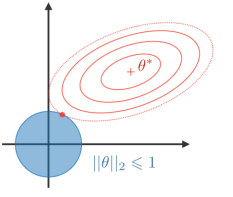
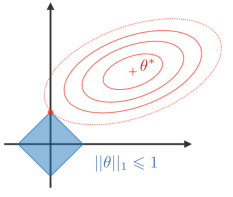
نظام‌بخشی

❑ **برون‌اندازی (dropout)** – برون‌اندازی روشی است که در شبکه‌های عصبی برای جلوگیری از بیش‌برازش بر روی داده‌های آموزشی با حذف تصادفی نورون‌ها با احتمال $p > 0$ استفاده می‌شود. این روش مدل را مجبور می‌کند تا از تکیه کردن بیش‌ازحد بر روی مجموعه خاصی از ویژگی‌ها خودداری کند.

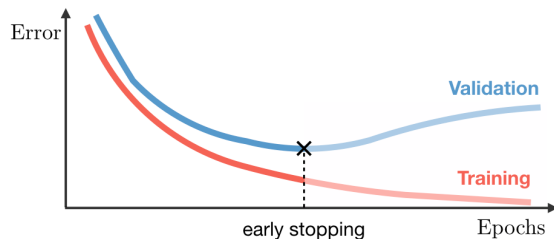


نکته: بیشتر کتابخانه‌های یادگیری عمیق برون‌اندازی را با استفاده از فراسنج 'نگهداشتن' $p - 1$ کنترل می‌کنند.

❑ **نظام‌بخشی وزن** – برای اطمینان از اینکه (مقادیر) وزن‌ها بیش‌ازحد بزرگ نیستند و مدل به مجموعه‌ای آموزش بیش‌برازش نمی‌کند، روش‌های نظام‌بخشی معمولاً بر روی وزن‌های مدل اجرا می‌شوند. اصلی‌ترین آنها در جدول زیر به اختصار آمده‌اند:

Elastic Net	Ridge	LASSO
بین انتخاب متغیر و ضرایب کوچک مصالحه می‌کند	ضرایب را کوچکتر می‌کند	– ضرایب را تا صفر کاهش می‌دهد – برای انتخاب متغیر مناسب است
		
$\dots + \lambda \left[(1 - \alpha) \ \theta\ _1 + \alpha \ \theta\ _2^2 \right]$ $\lambda \in \mathbb{R}, \alpha \in [0, 1]$	$\dots + \lambda \ \theta\ _2^2$ $\lambda \in \mathbb{R}$	$\dots + \lambda \ \theta\ _1$ $\lambda \in \mathbb{R}$

❑ **توقف زودهنگام (early stopping)** – این روش نظام‌بخشی، فرآیند آموزش را به محض اینکه خطای اعتبارسنجی ثابت می‌شود یا شروع به افزایش پیدا کند، متوقف می‌کند.



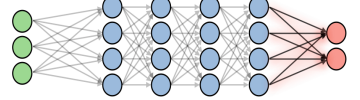

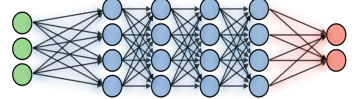
عادت‌های خوب

❑ **بیش‌برازش روی دسته‌ی کوچک** – هنگام اشکال‌زدایی یک مدل، اغلب مفید است که یک سری آزمایش‌های سریع برای اطمینان از اینکه هیچ مشکل عمده‌ای در معماری مدل وجود ندارد، انجام شود. به طور خاص، برای اطمینان از اینکه مدل می‌تواند

تنظیم فراسنج

❑ **مقداردهی اولیه Xavier** – به‌جای مقداردهی اولیه‌ی وزن‌ها به شیوه‌ی کاملاً تصادفی، مقداردهی اولیه Xavier این امکان را فراهم می‌سازد تا وزن‌های اولیه‌ی داشته باشیم که ویژگی‌های منحصربه‌فرد معماری را به حساب می‌آورند.

❑ **یادگیری انتقالی (transfer learning)** – آموزش یک مدل یادگیری عمیق به داده‌های زیاد و مهم‌تر از آن به زمان زیادی احتیاج دارد. اغلب بهتر است که از وزن‌های پیش‌آموزش روی پایگاه داده‌های عظیم که آموزش بر روی آن‌ها روزها یا هفته‌ها طول می‌کشد استفاده کرد، و آن‌ها را برای موارد استفاده‌ی خود به کار برد. بسته به میزان داده‌هایی که در اختیار داریم، در زیر روش‌های مختلفی که می‌توان از آنها بهره‌برداری کرده شده‌اند:

تعداد داده‌های آموزش	نگاره	توضیح
کوچک		منجمد کردن تمامی لایه‌ها، آموزش وزن‌ها در بیشینه‌ی هموار
متوسط		منجمد کردن اکثر لایه‌ها، آموزش وزن‌ها در لایه‌های آخر و بیشینه‌ی هموار
بزرگ		آموزش وزن‌ها در (تمامی) لایه‌ها و بیشینه‌ی هموار با مقداردهی اولیه‌ی وزن‌ها بر طبق مقادیر پیش‌آموزش

❑ **نرخ یادگیری (learning rate)** – نرخ یادگیری اغلب با نماد α و گاهی اوقات با نماد η نمایش داده می‌شود و بیانگر سرعت (گام) به‌روزرسانی وزن‌ها است که می‌تواند مقداری ثابت داشته باشد یا به صورت سازگار شونده تغییر کند. محبوب‌ترین روش حال حاضر Adam نام دارد، روشی است که نرخ یادگیری را در حین فرآیند آموزش تنظیم می‌کند.

❑ **نرخ‌های یادگیری سازگار شونده** – داشتن نرخ یادگیری متغیر در فرآیند آموزش یک مدل، می‌تواند زمان آموزش را کاهش دهد و راه‌حل بهینه عددی را بهبود ببخشد. با آنکه بهینه‌ساز Adam محبوب‌ترین روش مورد استفاده است، دیگر روش‌ها نیز می‌توانند مفید باشند. این روش‌ها در جدول زیر به اختصار آمده‌اند:

روش	توضیح	به‌روزرسانی w	به‌روزرسانی b
Momentum	– نوسانات را تعدیل می‌دهد – بهبود SGD – دو فراسنج که نیاز به تنظیم دارند	$w - \alpha v_{dw}$	$b - \alpha v_{db}$
RMSprop	– انتشار جذر میانگین مربعات – سرعت بخشیدن به الگوریتم یادگیری با کنترل نوسانات	$w - \alpha \frac{dw}{\sqrt{s_{dw}}}$	$b \leftarrow b - \alpha \frac{db}{\sqrt{s_{db}}}$
Adam	– تخمین سازگار شونده ممان – محبوب‌ترین روش – چهار فراسنج که نیاز به تنظیم دارند	$w - \alpha \frac{v_{dw}}{\sqrt{s_{dw}} + \epsilon}$	$b \leftarrow b - \alpha \frac{v_{db}}{\sqrt{s_{db}} + \epsilon}$

نکته: سایر متدها شامل *Adagrad*، *Adadelata* و *SGD* هستند.

به شکل صحیح آموزش ببیند، یک دسته‌ی کوچک (از داده‌ها) به شبکه داده می‌شود تا دریابیم که مدل می‌تواند به آنها بیش‌برازش کند. اگر نتواند، بدین معناست که مدل از پیچیدگی بالایی برخوردار است یا پیچیدگی کافی برای بیش‌برازش شدن روی دسته‌ی کوچک ندارد، چه برسد به یک مجموعه آموزشی با اندازه عادی.

❑ **وارسی گرادیان (gradient checking)** – وارسی گرادیان روشی است که در طول پیاده‌سازی گذر روبه‌عقب یک شبکه‌ی عصبی استفاده می‌شود. این روش مقدار گرادیان تحلیلی را با گرادیان عددی در نقطه‌های مفروض مقایسه می‌کند و نقش بررسی درستی را ایفا می‌کند.

گرادیان عددی	گرادیان تحلیلی	
$\frac{df}{dx}(x) \approx \frac{f(x+h) - f(x-h)}{2h}$	$\frac{df}{dx}(x) = f'(x)$	فرمول
<p>– پرهزینه (از نظر محاسباتی)، خطا باید دو بار برای هر بُعد محاسبه شود</p> <p>– برای تأیید صحت پیاده‌سازی تحلیلی استفاده می‌شود</p> <p>– مصالحه در انتخاب h :</p> <p>نه بسیار کوچک (ناپایداری عددی) و نه خیلی بزرگ (تخمین گرادیان ضعیف) باشد</p>	<p>– نتیجه 'عینی'</p> <p>– محاسبه مستقیم</p> <p>– در پیاده‌سازی نهایی استفاده می‌شود</p>	توضیحات

* * *