پیش گزارش آزمایش دوم (پیاده سازی پرسپترون)

امیرحسین احمدی آشتیانی ۹۹۲۳۵۰۱

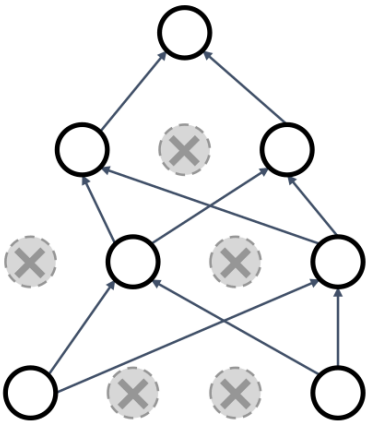
محمدرضا امیری ۹۹۲۶۰۴۰

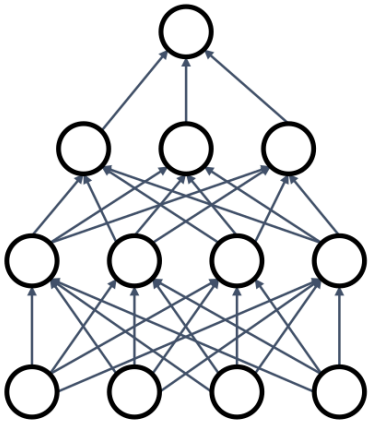
محمدمهدی نوروزی ۹۹۲۳۰۸۵

1) عملکرد dropout را توضیح دهید.

این الگوریتم را در روش های regularization قرار دارد. هدف regularization چلوگیری از overfit است. ساده ترین regularization همان اضافه کردن مجموع ضرایب به loss function است که به نام L1 و L2 شناخته می‌شود. پس dropout هم سعی میکند overfit را خنثی کند.

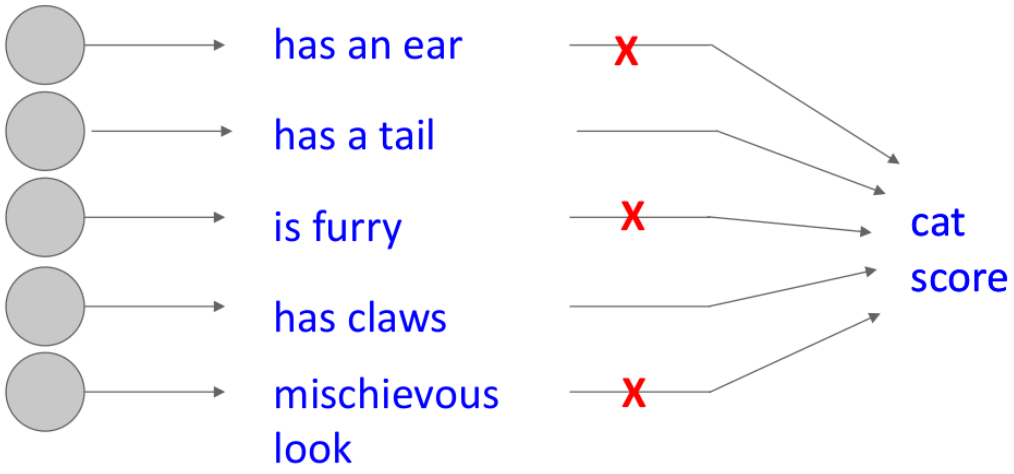
در این روش، در forward pass به صورت رندم برخی از نورون هارا صفر میکنیم. احتمال صفر کردن نورون یک هایپر پارامتر است.



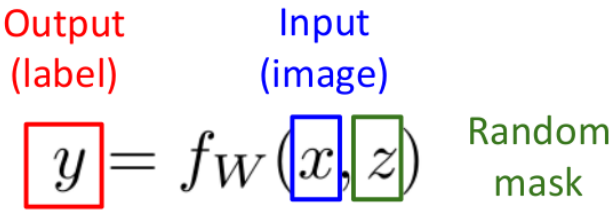


اما این روش تاثیر های دیگری نیز دارد که به برخی از آنها اشاره میکنیم:

- شبکه را مجبور می کند که یک نمایش اضافی داشته باشد. برای مثال وقتی میخواهیم یک گربه را تشخیص دهیم، حتما نباید تمام ویژگی هارا ببیند و با کنار هم قرار دادن تمام آنها، گربه بودن را تشخیص دهد. شکل زیر این مورد را نشان میدهد:



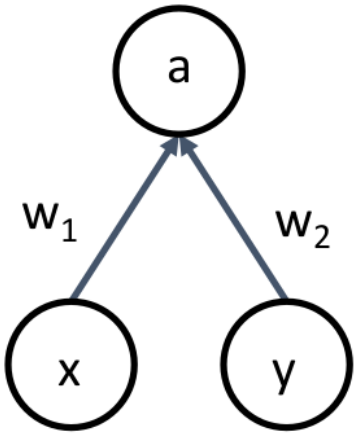
- یک تفسیر دیگر این است: در دنیای کلاسیفیکیشن، از ensemble برای تقویت مدل کلاسیفایر استفاده میشود. یعنی جواب کلاس بندی چندین مدل مختلف را باهم ترکیب کرده و بدین ترتیب مدل بسیار کارکرد بهتری خواهد داشت. Dropout نیز همین کار را میکند، چندین مدل مختلف (با نورون های حذف شده مختلف) را کنار هم قرار میدهد و بعد تصمیم گیری میکند (در زمان تست). اما سوال مهم این است که چطوری در زمان تست عمل میکند؟ پاسخ این است که dropout خروجی را random میکند چون نورون ها رندم هستند:



برای اینکه تشخیص دهیم که به طور متوسط خروجی چه مقداری خواهد بود باید expectation آنرا محاسبه کنیم.

$$y = f(x) = E_z\left[f(x,z)\right] = \int p(z) f(x,z) dz$$

برای محاسبه بیاید یک نورون را درنظر بگیریم:



در زمان تست داریم:

$$E[a] = w_1 x + w_2 y$$

اما در زمان ترین داریم:

$$\begin{aligned} E[a] =& \frac{1}{4}(w_1 x + w_2 y) + \frac{1}{4}(w_1 x + 0y) \\ &+ \frac{1}{4}(0x + 0y) + \frac{1}{4}(0x + w_2 y) \\ =& \frac{1}{2}(w_1 x + w_2 y) \end{aligned}$$

پس در زمان تست نیازی به حذف کردن نورون ها نیست، فقط خروجی را محاسبه و در احتمال dropout ضرب کنیم.

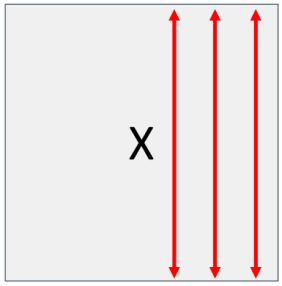
2) روش bach normalization را توضیح دهید.

ایده اصلی این است که خروجی هر لایه رو نرمال کنیم به طوری که میانگین صفر و واریانس ۱ داشته باشد. این کار به optimaz کردن کمک میکند. اما چطوری؟ در ادامه بررسی میکنیم. به روش زیر خروجی یک لایه را نرمال میکنیم:

$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

این یک تابع مشتق پذیر است است و میتوان در لایه های شبکه عصبی قرار داد و عملیات backprop را روی آنها انجام داد. از اسم این روش مشخص است که با تعدادی داده کار داریم (batch).
میانگین و انحراف معیار به صورت زیر محاسبه خواهد شد:

Input: $x : N \times D$



$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{i,j}$$ Per-channel mean, shape is D

$$\sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}(x_{i,j} - \mu_j)^2$$ Per-channel std, shape is D

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_j}{\sqrt{\sigma_j^2 + \varepsilon}}$$ Normalized x, Shape is N x D

از محاسبات بالا متوجه میشویم که این نرمال کردن روی یک داده نیست. باید فیچر ها نرمال شوند.
خب حالا سوال پیش می آید که اگر میانگین صفر و واریانس واحد شرایط مطلوبی نباشد باید چه می شود؟ پاسخ این است که برای حل کردن این مشکل دو پارامتر جدید برای لایه batch norm تعریف میشود: scale و shift.
اگر گاما که همان پارامتر scale است برابر انحراف معیار شود و بتا که همان shift است برابر میانگین شود، لاید batch norm یک لایه همانی خواهد شد.
مشکلی که وجود دارد این است که اگر بخواهیم در زمان تست این کار را انجام دهیم، خروجی بر اساس ورودی تغییر خواهد کرد، یعنی یک لایه ای درون شبکه عصبی وجود دارد که از قبل تعیین نشده و در زمان تست مشخص میشود که این مطلوب نیست. برای حل این مشکل میانگین و

واریانس را از مرحله ترین استفاده میکنند. یعنی running average میانگین و انحراف معیار در مرحله ترین، در مرحله تست استفاده میشود.

اگر از بچ نرمالیزیشن استفاده کنیم، میتوانیم لرنینگ ریت را افزایش دهیم بدون اینکه نگران شویم که واگرا خواهیم شد.

3) الگوریتم پس انتشار خطا در حالت کلی آپدیت کردن وزن ها است با توجه به خطایی که در خروجی داشتیم. پس باید مشتق خطا نسبت به تمام وزن هارا داشته باشیم. یک الگوریتم ساختارمند که از مشتق زنجیره ای حاصل شده است، گراف محاسبات است. پس اگر گراف محاسبات را محاسبه کنیم، فقط آپدیت کردن باقی می ماند. پس در زیر گراف محاسبات را توضیح می دهیم. همانند شکل زیر ابتدا باید forward انجام شود:

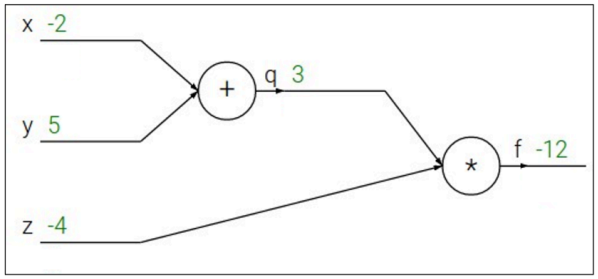
Backpropagation:
Simple Example

$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

1. Forward pass: Compute outputs

$$q = x + y \qquad f = qz$$

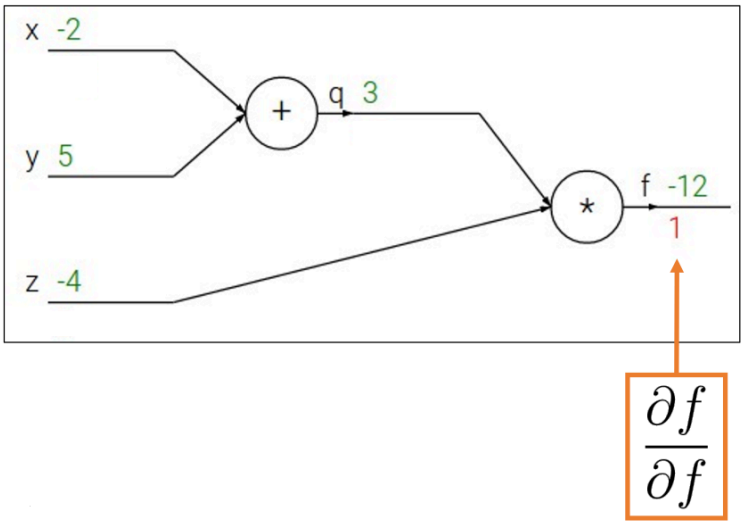


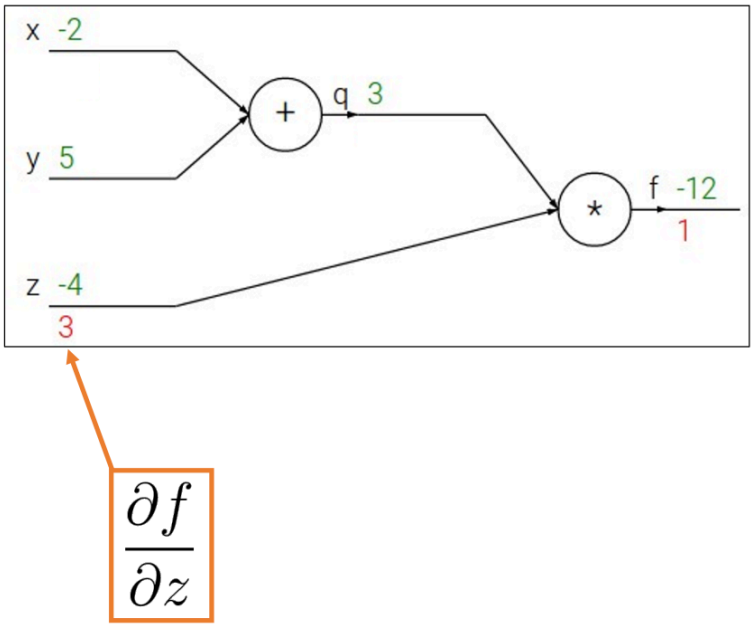
حال مقادیر زیر را می خواهیم:

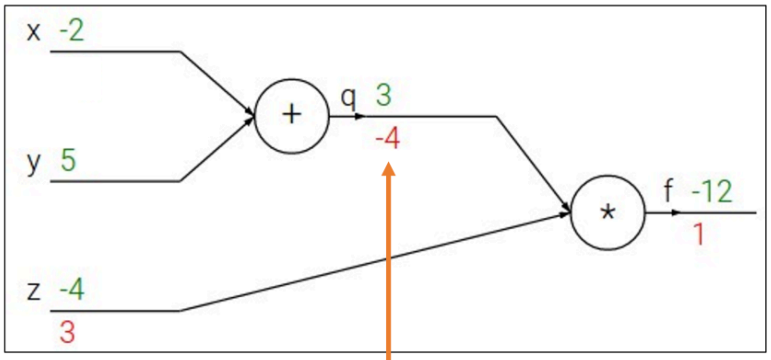
2. Backward pass: Compute derivatives

$$\text{Want: } \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$$

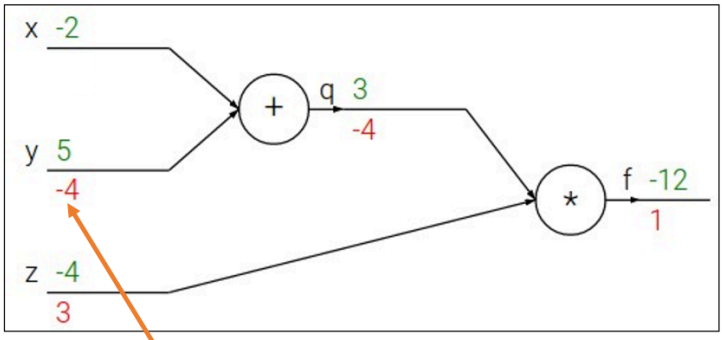
تصاویر زیر محاسبات را نشان می دهد:


x -2
y 5
z -4
+
q 3
*
f -12
1
$\frac{\partial f}{\partial f}$


x -2
y 5
z -4
3
+
q 3
*
f -12
1
$\frac{\partial f}{\partial z}$



$$\frac{\partial f}{\partial q} = z$$

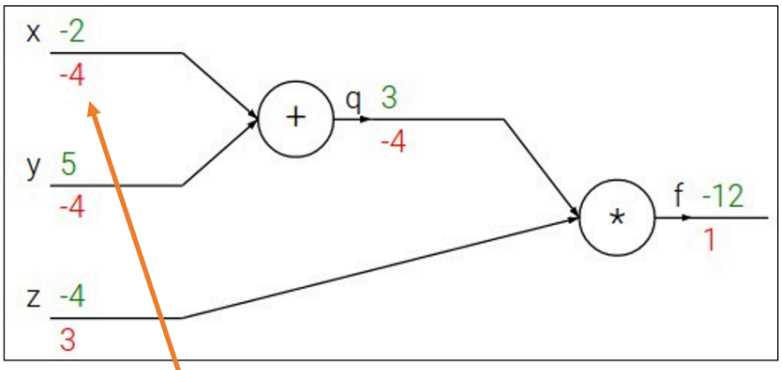


Chain Rule

$$\frac{\partial f}{\partial y} = \frac{\partial q}{\partial y}\frac{\partial f}{\partial q}$$

$$\frac{\partial q}{\partial y} = 1$$

Downstream Gradient **Local Gradient** **Upstream Gradient**



Chain Rule

$$\frac{\partial f}{\partial x} = \frac{\partial q}{\partial x}\frac{\partial f}{\partial q}$$

$$\frac{\partial q}{\partial x} = 1$$

Downstream Gradient **Local Gradient** **Upstream Gradient**