نویسنده: سارا حسینی

۱. لاجیستیک رگرشن یک تابع احتمالاتی است که عددی بین ۰ و ۱ برمیگرداند. درحالی که یک پرسپترون، یک مدل خطی است که جوابش یا از threshold بزرگتر است و ۱ برمیگرداند، یا کوچکتر است و ۰ برمی‌گرداند و فقط در صورتی به درستی داده‌ها را تشخیص می‌دهد که داده‌ها به صورت خطی تفکیک‌پذیر باشند.

Output = $\Sigma w_i x_i - b$

لاجیستیک رگرشن به صورت:

$$\log\left(\frac{p}{1-p}\right) = \Sigma w_i x_i \qquad \Rightarrow \qquad \frac{e^{\Sigma x_i w_i}}{1+e^{\Sigma w_i x_i}} = \frac{1}{1+e^{-\Sigma w_i x_i}}$$

است که p همان احتمال 1 بودن خروجی است.
از آنجایی که لاجیستیک رگرشن، مقدار دقیق احتمال را هم نشان می‌دهد، بهتر است. و اینکه لاجیستیک رگرشن غیرخطی است و برای مسائل بیشتری کاربرد دارد. برای اینکه پرسپترون به logreg تبدیل شود، باید در انتها به آن یک تابع فعالیت سیگموید اضافه کنیم.

y = $\Sigma w_i x_i - b$ and Output = $1/(1 + e^{-y})$

۲.تابع mse:

$$L_1 = \frac{1}{m}\Sigma(y - \widehat{y})^2$$

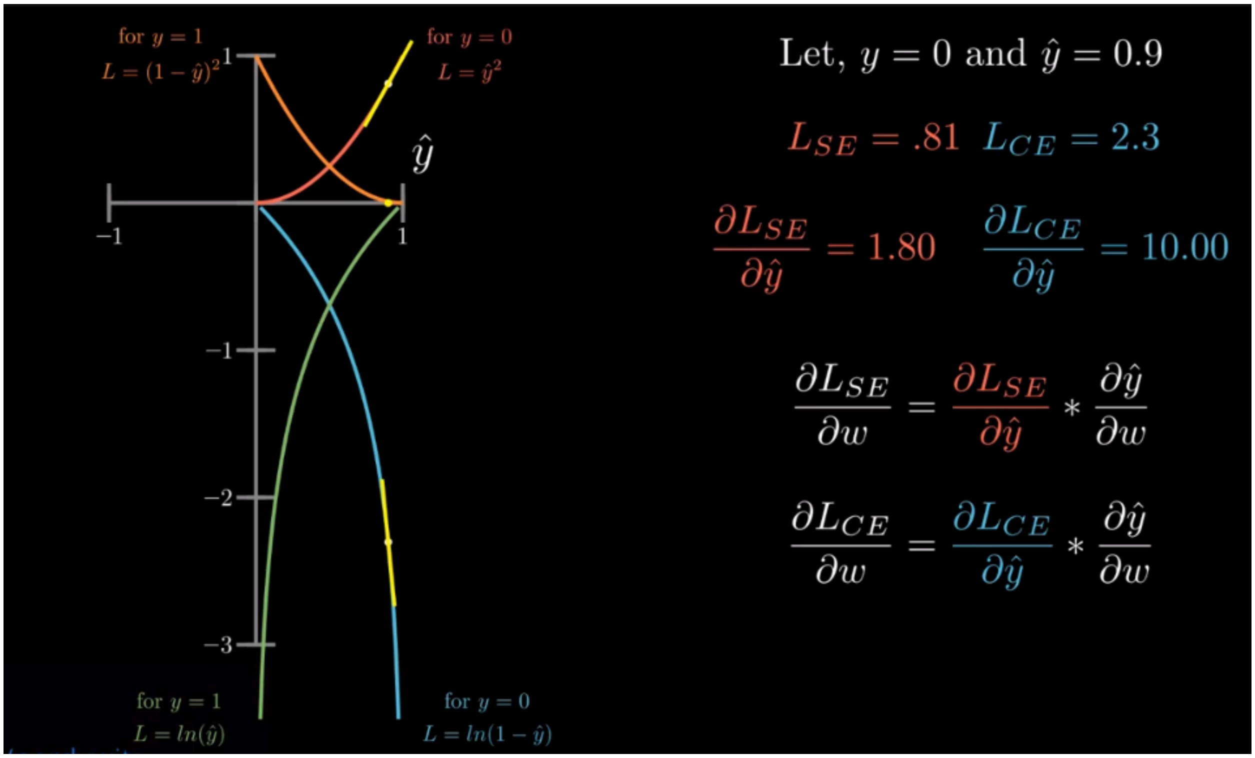
تابع کراس انتروپی:

$$L_2 = \frac{1}{m}\Sigma(y ln(\widehat{y}) + (1 - y) ln(1 - \widehat{y}))$$

برای تسک کلسیفیکیشن، کراس انتروپی مناسبتره و برای تسک رگرشن، mse.
برای تسک کلسیفیکیشن، y یعنی مقدار واقعی تارگت دو حالت دارد: صفر یا یک. اینجا مقدار لاس برای یک داده را محاسبه می‌کنیم در هر حالت:

$$if\ y = 0 \rightarrow L_1 = ln(1 - \widehat{y})) \quad and \quad L_2 =- \widehat{y}^2$$

$$if\ y = 1 \rightarrow L_1 = ln(\widehat{y})) \quad and \quad L_2 = (1 - \widehat{y})^2$$



همانطور که در شکل دیده می‌شود، مشتق تابع کراس انتروپی وقتی که یک misclassifification رخ داده بسیار بالاتر است این یعنی کراس انتروپی یک misclassification را بسیار شدیدتر penalize میکند. ما در تسک رگرشن دنبال مدلی پیوسته هستیم پس نیاز به پنالتی شدید نداریم اما در کلسیفیکیشن، این پنالتی بسیار کاربردی است.

ما بطور ریاضی یز میتوانیم اثبات کنیم که فرمول MSE از یک توزیع احتمالی گاوسی بدست می‌آید در حالی که فرمول CE از توزیع برنولی (باینری) بدست می‌آید.

همچنین تابع MSE برای این تسک convex نیست پس تضمینی نداریم که به گلوبال مینیمم برسیم

۳. اگر وزن‌ها به طورت رندوم اینیشالایز بشوند، معمولا از یک توزیع نورمال پیروی می‌کنند که میانگین ۰ و واریانس ۱ خواهند داشت. از آنجا که داریم:

$z = \Sigma w_i x_i$

خروجی نورون لایه‌ی j که نورون‌های لایه j-1 به آن وصل هستند یعنی z، قبل از اعمال تابع فعالیت، مقداری با میانگین ۰ اما واریانس n خواهد داشت. (واریانس مجموع برابر مجموع واریانس هاست) این واریانس زیاد، یعنی محتمل‌تراست که z مقداری بسیار بزرگ یا بسیار کوچک شود مثلا ریشه‌ی n (انحراف معیار).

وقتی هم روی z تابعفالیتی مثل سیگموید اعمال شود، برای مقادیر بسیار بزرگ یا کوچک، مشتق بسیار ناچیز است یعنی vanishing gradient و همین باعث میشود یادگیری کند شود.

اینیشالایز کردن با روش xavier اینگونه است که وزن‌های اولیه از توزیع نورمال انتخاب شده، سپس در $\sqrt{\frac{1}{n}}$ ضرب شوند تا میانگین صفر بماند ولی واریانس $\frac{1}{n}$ شود. در مقاله‌ای که xavier glorot ارائه داده بود، n

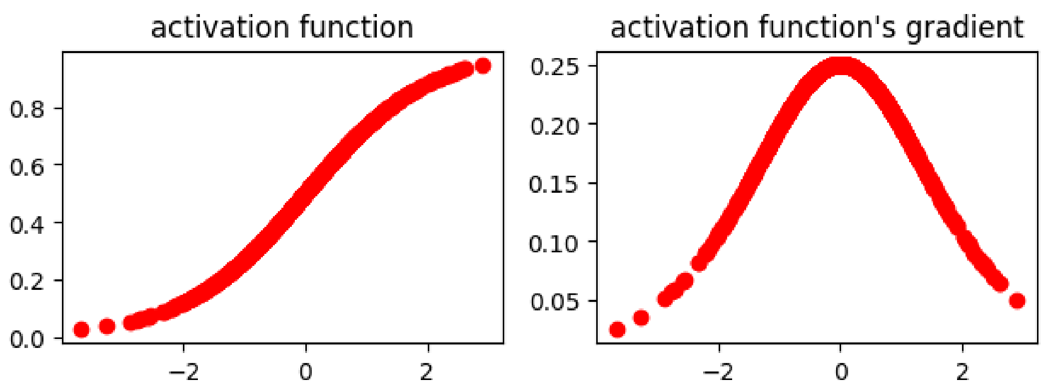
برابر با میانگین تعداد نورون‌های ورودی و تعداد نورونهای خروجی درنظر گرفته شده بود. این روش مناسب توابع فعالیت خطی، softmax، tanh و توابع لاجیستیک است.

میتوان در روش xavier، مقدار n را صرفا تعداد نورون ورودی درنظرگرفت که به آن روش lecun میگویند و مناسب selu است.

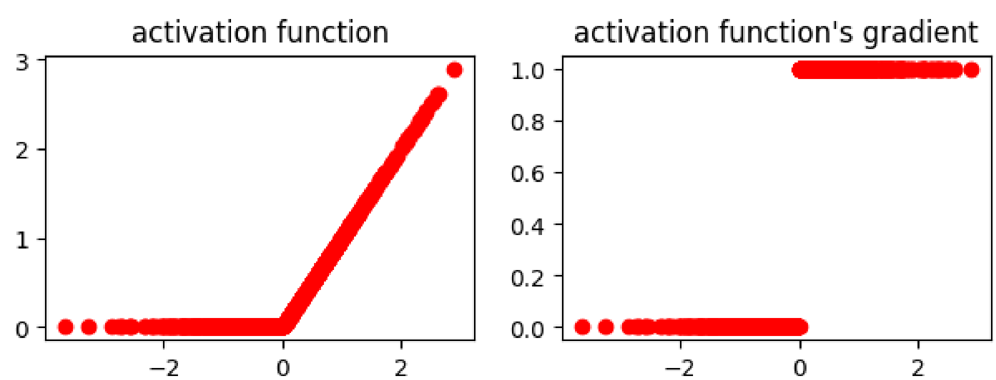
طبق روش He، برای ReLU و مشابه آن، بهتر است در $\sqrt{\frac{2}{n}}$ ضرب شوند تا واریانس وزن‌ها $\frac{2}{n}$ شود. N اینجا صرفا تعداد نورون های ورودی است.

اگر وزن‌ها را در ابتدا به طور رندوم اینیشالایز نکنیم و همه را ۰ بگذاریم، طی آپدیت‌ها وزن‌های هر لایه به مقدار یکسانی تغییر می‌کنند و عملا وزن‌ها یکسان و متقارن می‌شوند و شبکه نمی‌تواند خوب یاد بگیرد.

۴. تابع سیگموید در بازه‌ی بسیار بزرگی، مشتق بسیار کوچک و نزدیک صفر دارد. این بزرگترین نقطه ضعف سیگموید است که (وقتی pre-activation خیلی بزرگ یا خیلی کوچک باشد) سبب تغییرات بسیار کم در وزن‌ها در هر مرحله می‌شود و یادگیری را کند می‌کند. همچنین بزرگترین مشتقی که سیگموید دارد، ۰.۲۵ است که همچنان مقدار اندکی است.



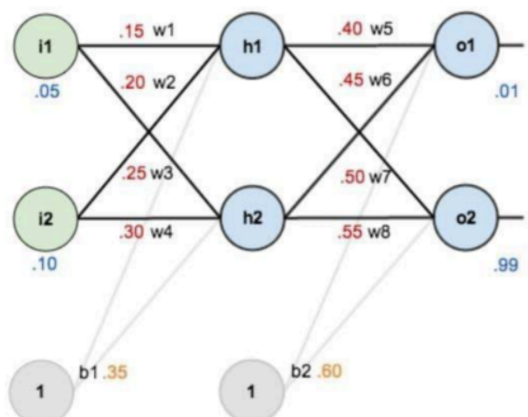
مشکل vanishing gradient در رلو کمتر پیش می‌آید. چون مشتق آن برای تمام نواحی مثبت، یک است.



همچنین، محاسبه‌ی اینکه ۰ بزرگتر است یا x برای رلو، بسیار ساده‌تر است از محاسبه‌ی e**x در سیگموید.

برای رلو، همگرایی بسیار سریع‌تر اتفاق میفتد و همین باعث می‌شود که اورفیت شدن هم بیشتر رخ دهد. و همچنین ممکن است مشکل dying relu رخ دهد یعنی ورودی یک نورون متداوما منفی باشد و آن نورون دیگر نتواند به یادگیری contribution داشته باشد و وزنی را تغییر دهد.

۵. اتصالات اسپارس در مقابل fully connected network قرار می‌گیرند. یعنی یک نورون لزوما به همه‌ی نورون‌های لایه قبلی متصل نیست (مثلا استفاده از رگولاریزیشن L1 باعث می‌شود بعضی وزن‌ها نزدیک به صفر شوند و عملا این اتصالات حذف شوند). این ایده‌ی اصلی convolution هم هست و باعث کمتر شدن و راحت‌تر شدن انجام محاسبات می‌شود. همچنین باعث می‌شود حافظه و انرژی کمتری صرف شود. احتمال اورفیت را نیز کاهش میدهد چون عملا با وزنهای کمتر همچنان داریم تابع را پیش بینی میکنیم پس جنرالایز شدن خوبی داشته‌ایم.
برای اسپارس شدن شبکه pruning انجام می‌دهیم و وزن‌های با اهمیت کمتر را از شبکه حذف می‌کنیم. معمولا بایس را حذف نمی‌کنیم چون حذف بایس‌ها طبق تجربیات، پرفورمنس را بسیار کاهش می‌دهد.
در روش Magnitude based اهمیت وزن را بر اساس magnitude اندازه آن می‌سنجیم.
اما روش sensitivity based مناسبتر است چون وزن‌هایی را حذف می‌کند که خروجی مدل به آن‌ها حساسیت بسیار کمی دارد. یعنی سهم کمی در خروجی داشته‌اند.
Optimal brain damage نیز با همین منطق کار می‌کند که وزن‌ها بر اساس میزان تاثیرشان روی تابع هزینه، حذف میشوند.
۶.



A.

$sigmoid := \sigma$

$a_{h_1} = 0.35 + 0.05 * 0.15 + 0.10 * 0.20 = 0.3775$

h1 = σ(0.3775)=0.5932

$a_{h_2} = 0.35 + 0.05 * 0.25 + 0.10 * 0.30 = 0.3925$

h2 = σ(0.3924)=0.5968

$a_{o_1} = 0.60 + 0.5932 * 0.40 + 0.5968 * 0.45 = 1.1058$

$o_1 = \sigma(a_{o_1}) = 0.7513$

$a_{o_2} = 0.60 + 0.5932 * 0.50 + 0.5968 * 0.55 = 1.2248$

$o_2 = \sigma(a_{o_2}) = 0.7729$

Error = $\frac{((0.01-0.7513)^2+(0.99-0.7729)^2)}{2} = \frac{0.5966581}{2} = 0.29832905$

B.

$\sigma'(x) = (1 - \sigma(x))\,\sigma(x)$

$\Delta_{(o_1,o_1)} = \frac{\partial L}{\partial o_1} = 1$

$\Delta_{(o_2,o_2)} = \frac{\partial L}{\partial o_2} = 1$

$\Delta w_8 = \eta\,.\,\Delta_{(o_2,o_2)}\,.\,h_2.\ \sigma'(a_{o_2}) = 0.3 * 1 * 0.5968 * (1 - 0.7729) * 0.7729 = 0.0314$

$\Delta w_7 = \eta\,.\,\Delta_{(o_2,o_2)}\,.\,h_1.\ \sigma'(a_{o_2}) = 0.3 * 1 * 0.5932 * (1 - 0.7729) * (0.7729) = 0.0312$

$\Delta w_6 = \eta\,.\,\Delta_{(o_1,o_1)}\,.\,h_2.\ \sigma'(a_{o_1}) = 0.3 * 1 * 0.5968 * (1 - 0.7513) * (0.7513) = 0.0334$

$\Delta w_5 = \eta\,.\,\Delta_{(o_1,o_1)}\,.\,h_1.\ \sigma'(a_{o_1}) = 0.3 * 1 * 0.5932 * (1 - 0.7513) * (0.7513) = 0.0332$

$$\Delta_{(h_2,\,o)} = \frac{\partial L}{\partial h_2} = (\frac{\partial L}{\partial o_1} * \frac{\partial o_1}{\partial h_2}) + (\frac{\partial L}{\partial o_2} * \frac{\partial o_2}{\partial h_2}) = (\Delta_{(o_1,o_1)} * \sigma'(a_{o_1}) * w_6) + (\Delta_{(o_2,o_2)} * \sigma'(a_{o_2}) * w_8)$$

$$= ((1 - 0.7513) * (0.7513) * 0.45\,) + ((1 - 0.7729) * (0.7729) * 0.55) = 0.1806$$

$$\Delta_{(h_1, o)} = \frac{\partial L}{\partial h_1} = (\frac{\partial L}{\partial o_1} * \frac{\partial o_1}{\partial h_1}) + (\frac{\partial L}{\partial o_2} * \frac{\partial o_2}{\partial h_1}) = (\Delta_{(o_1, o_1)} * \sigma'(a_{o_1}) * w_5) + (\Delta_{(o_2, o_2)} * \sigma'(a_{o_2}) * w_7)$$

$$= ((1 - 0.7513) * (0.7513) * 0.40) + ((1 - 0.7729) * (0.7729) * 0.50) = 0.1625$$

$$\Delta w_4 = \eta . (\Delta_{(h_2, o\ )}) . i_2 . \sigma'(a_{h_2})$$

$$= 0.3 * (0.1806) * (0.10) * (0.5968) * (1 - 0.5968) = 0.0013$$

$$\Delta w_3 = \eta . (\Delta_{(h_2, o\ )}) . i_1 . \sigma'(a_{h_2})$$

$$= (0.3) * (0.1806) * (0.05) * (0.5968) * (1 - 0.5968) = 0.0006$$

$$\Delta w_2 = \eta . (\Delta_{(h_1, o\ )}) . i_2 . \sigma'(a_{h_1})$$

$$= (0.3) * (0.1625) * (0.10) * (0.5932) * (1 - 0.5932) = 0.0011$$

$$\Delta w_1 = \eta . (\Delta_{(h_1, o\ )}) . i_1 . \sigma'(a_{h_1})$$

$$= (0.3) * (0.1625) * (0.05) * (0.5932) * (1 - 0.5932) = 0.0005$$

$$\Delta b_2 = (\eta . \Delta_{(o_1, o_1)} . \sigma'(a_{o_1})) + (\eta . \Delta_{(o_2, o_2)} . \sigma'(a_{o_2}))$$

$$= (0.3) * ((1 - 0.7513) * (0.7513) + (1 - 0.7729) * (0.7729)) = 0.1087$$

$$\Delta b_1 = (\eta . (\Delta_{(h_2, o\ )}) . \sigma'(a_{h_2})) + (\eta . (\Delta_{(h_1, o\ )}) . \sigma'(a_{h_1}))$$

$$= (0.3) * (((0.1806) * (0.5932) * (1 - 0.5932)) + ((0.1625) * (0.5968) * (1 - 0.5968)))$$

$$= 0.0248$$

$$w_1 = 0.15 - (0.0005) = 0.1495$$

$$w_2 = 0.20 - (0.0011) = 0.1989$$

$$w_3 = 0.25 - (0.0006) = 0.2494$$

$w_4 = 0.30 - (0.0013) = 0.2987$

$w_5 = 0.40 - (0.0332) = 0.3668$

$w_6 = 0.45 - (0.0334) = 0.4166$

$w_7 = 0.50 - (0.0312) = 0.4688$

$w_8 = 0.55 - (0.0314) = 0.5186$

$b_1 = 0.35 - (0.0248) = 0.3252$

$b_2 = 0.60 - (0.1087) = 0.4913$

a_h1= w1i1+w2i2+b1 =(0.1495*0.05)+(0.1989*0.10)+0.3252 =0.352565
a_h2= w3i1+w4i2+b1 =(0.2494*0.05)+(0.2987*0.10)+0.3252 =0.36754

h1=0.5872
h2=0.5908

a_o1= w5h1+w6h2+b2=(0.3668*0.5872)+(0.4166*0.5908)+0.4913
=0.95281224
a_o2= w7h1+w8h2+b2=(0.4688*0.5872)+(0.5186*0.5908)+0.4913
=1.07296824

o1=0.72168
o2=0.74516

Error = 0.28321752