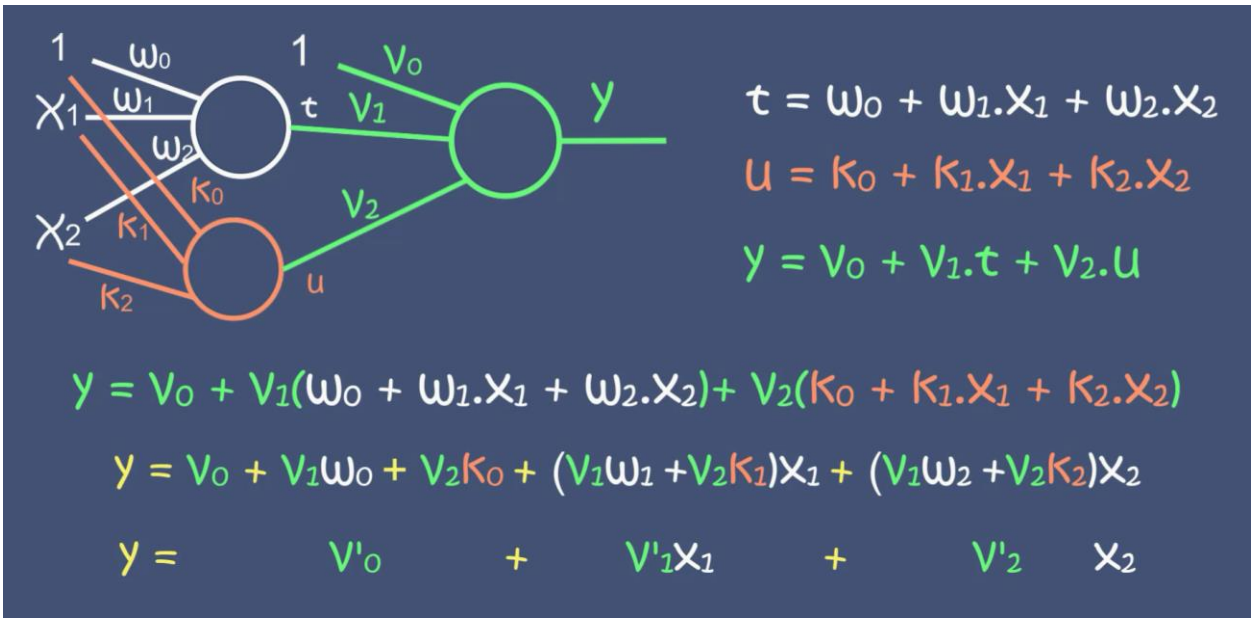
به نام خدا

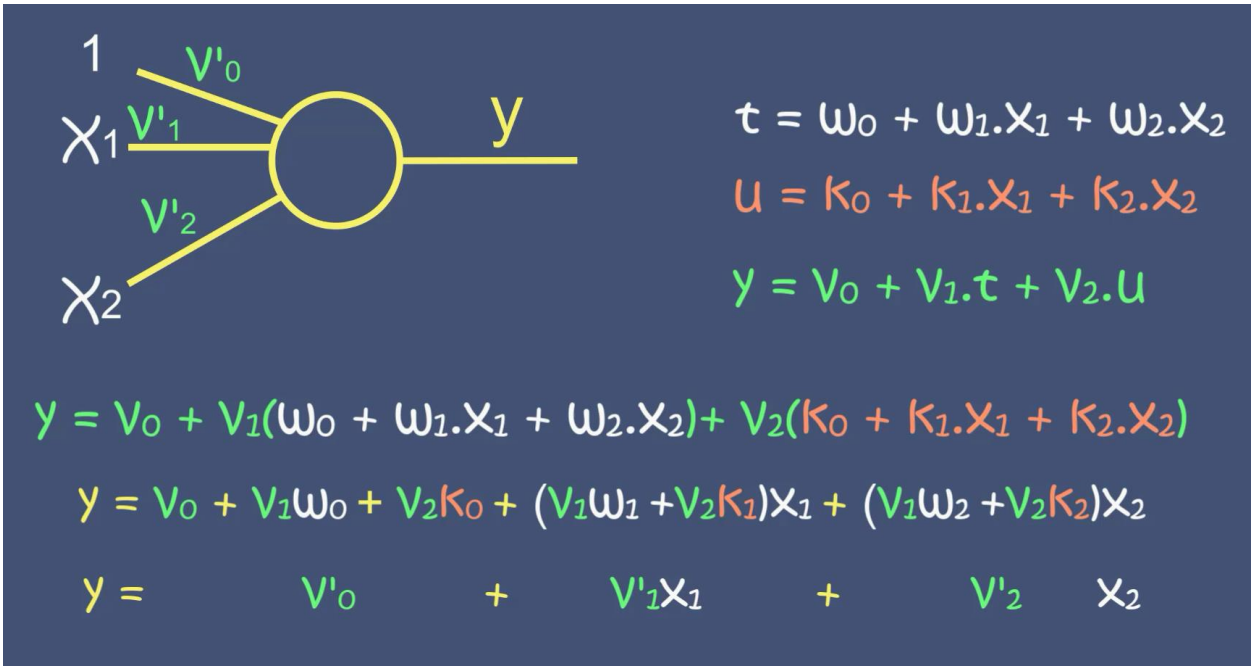
سینا کریمی

9931050

سوال 1)

1) اگر توابع فعالیت استفاده نکنیم در این صورت خروجی تمام پرسپترون ها در نهایت در قالب یک معادله خطی در خواهد آمد و عملا انگار تنها یک پرسپترون در این شبکه وجود دارد





2) این تابع میتواند مقادیر نزدیک به صفر و یا یک تولید کند، که میتوناد منجر به بروز مشکلاتی در الگوریتم های بهینه سازی شود. در واقع گرادیان این تابع در نزدیکی خروجی های 0 و ای یک بشدت کوچک میشود که باعث میشود وفق دادن وزن ها و بایاس ها برای الگوریتم های بهینه سازی سخت بشود

3) اگر صفر محور نباشد باعث میشود در که در زمان back propagation برخی از گرادیان ها نزدیک به صفر باشند و عملا در محاسبات لحاظ نشوند و در بهینه سازی ها خطا ایجاد کنند

4)ReLU یک تابع فعالیت خطی است که اگر مقدار ورودی منفی باشد صفر و اگر مثبت باشد، مقدار ورودی را برمیگرداند. با آنکه این تابع کاملا مشتق پذیر نیست میتوان از sub gradiantها برای مشتق گیری استفاده کرد

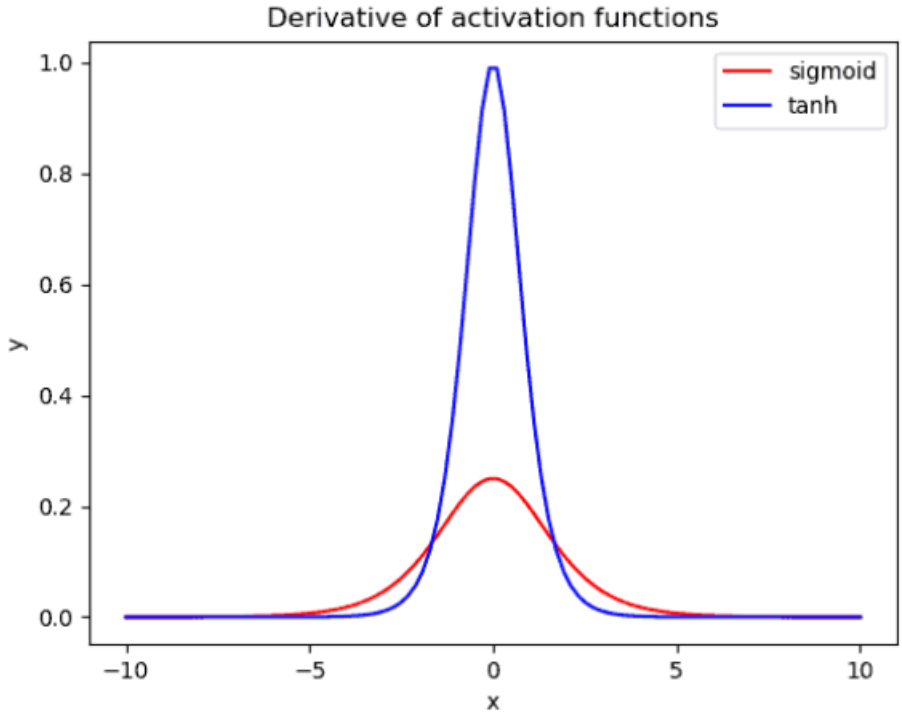
مزایا: پیاده سازی آن آسان است، مشکل vanishing Gradient را حل میکند (در سوال قبل به آن اشاره شد)، محاسبات سریع تر انجام میشود

معایب: یک تابع صفر محور نیست، نیازمند دقت در وزن دهی اولیه است، برای همه مشکلات مناسب نیست

5) این تابع تقریبا مانند تابع ReLU است با این تفاوت که برای $x < 0$ به جای مقدار صفر مقدار ax برگردانده میشود که a یک مقدار کوچکی است. در واقع از آنجایی که تابع ReLU برای مقادیر منفی گرادیان ندارد و باعص میشود نورون های این مقایدر غیر فعال شوند، leaky ReLU این مشکل را با دادن یک شیب کوچک در مقایدر منفی حل میکند

6) تابع tanh نسخه کش آمده و تغییر یافته تابع sigmoid است در نتیجه شباهت های بسیاری دارند. یکی از این شباهت ها این است که هردو خروجی را به بازه خاص محدود میکنند و این باعث میشود که وزن ها محدود شوند و مقادیر گرادیان ها بیش از حد بزرگ نشود.

از تفاوت های مهم این دو تابع رفتار گرادیان آنها است.



با توجه به عکس بالا، گرادیان تابع tanh نزدیک به 4 برابر تابع sigmoid است. این نشان میدهد مه در زمان train کردن شبکه گرادیان های بیشتری مشاهده خواهد شد و در نتیجه وزن ها بیشتر بروزرسانی میشوند. تفاوت بعدی در این است که خروجی تابع tanh در نزدیکی صفر متقارن است و این منجر به همگرایی سریع تر میشود

7) تابع softmax یک تابع است که یک بردار از kتا عدد حقیقی که فرقی نمیکند مثبت، منفی و یا صفر باشند را به برداری از kتا عدد حقیقی که جمع آنها برابر یک (مقادیری بین صفر و یک) میشود تبدیل میکند. این تابع در لایه آخر شبکه های عصبی استفاده میشود تا کلاس (نوع) یک تصویر ورودی را پیشبینی کند

سوال 2)

1) به این دلیل که گرادیان این تابع صفر خواهد بود و روش های بهینه سازی براساس گرادیان کار نخواهند کرد و چیزی یادگرفته نخواهد شد

2) استفاده از sigmoid بهتر خواهد بود چرا که خروجی در بازه 0 تا 1 خواهد بود و میتوان از این ساده سازی برای تعیین کلاس استفاده کرد و همین امر موجب میشود که بتوان خروجی را در قالب احتمال اینکه کدام کلاس است بکار برد، همچنین برای تابع sigmoid معمولا از تابع خطای binary cross entropy استفاده میشود که برای این نوع مسائل مناسب تر است

3) میتواند منجر به overfitting شود چرا که دیتا به اندازه کافی نبوده و شبکه به اصطلاح این داده ها را حفظ میکند و با آنها به خوبی عمل کرده ولی با داده های کلی تر و شرایط جدید نتیجه خوبی نخواهد داشت. همچنین این امر موجب دقت پایین میشود و همجنین میتواند باعث شود که شبکه نتواند برروی یک جواب به همگرایی برسد

4) بیشتر در مسائل binary classification استفاده میشود و فرمول آن به شرح زیر است

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1-y_i) \cdot log(1-p(y_i))$$

5) یک شبکه عصبی fully connected رای MLP مینامیم که شامل یک لایه ورودی، یک یا بیشتر لایه نهان و یک لایه خروجی است. در آموزش MLP اول داده به این شبکه وارد شده و لایه به لایه پیش میروند تا به لایه خروجی رسیده و در آنجا تابع loss بر آنها اعمال شده و error محاسبه میشود (feedforwarding). سپس error ایجاد شده back propagate میشود تا

وزن ها و bias های جدید ایجاد بشود و سپس چرخه بعدی آموزش با وزن و biasهای جدید شروع میشود.

6)

A: نرخ آموزش بیش از حد بالاس (very high). اگر داده ای دیده شود که فرق دارد سریعا به عنوان خروجی مناسب شناسایی میشود و داده های پیشین دیگر لحاظ نمیشوند.

B: نرخ آموزش کم است (low). اگر داده ای دیده بشود که در اقلیت قرار دارد به عنوان داده پرت شناسایی میشود

C: نرخ آموزش زیاد است (high)

D: نرخ آموزش خوب است.

سوال 3)

این دو در واقع به عنوان loss function استفاده میشوند و در زمان train کردن مدل ها از آنخا استفاده میشود. هرجه مقداری که این توابع نمایش میدهند کوچکتر باشد یعنی مدل ما عملکرد بهتری دارد. از Cross Entropy بیشتر در مسائل binary classification استفاده میشود و از SSE بیشتر در مسائل regression میشود. برای مثال برای پیشبینی قیمت خانه ها براساس ویژگی هایی که دارند میتوان از SSE و برای تعیین اینکه نوع حیوان موجود در یک سری عکس سگ است یا گربه میتوان از cross entropy استفاده کرد

سوال 4)

الف)

در محاسبه گرادیان میتوان به این دو روش عمل کرد:

Batch gradient descent: از کل دیتاست برای محاسبه گرادیان استفاده میکند، کند و پران قیمت است، همگرایی دیرتر رخ میدهد ولی چون از کل دیتاست استفاده میکند دقیق تر است، میتواند در مینیمم محلی گیر بیافتد، برای دیتاست های کوچک تر مناسب است

Stochastic Gradient Descent: از بخشی از دیتاست برای محاسبه گرادیان استفاده میکند، سریع تر و ارزان تر است، برای دیتاست های بزرگ مناسب تر است، سریعتر بع همگرایی میرسد ولی جواب ها کاملا بهینه نیستند

ب) نرخ آموزش که آن را اِتا نیز مینامیم درواقع نرخی که است که با استفاده از آن و براساس گرادیان، وزن ها آپدیت میشوند. هرچه نرخ آموزش بیشتر باشد شبکه بیشتر در مسیر گرادیان گام برمیدارد.

نرخ آموزش هم میتواند ثابت بماند و هم میتواند در طول آموزش از طریق step decayها و یا Adaptive learning rate algorithmها تغییر کند

سوال 5)

$$z1 = \begin{bmatrix} -0.3 & 0.2 \\ 0.6 & 0.6 \\ 0.4 & 0.8 \end{bmatrix} \begin{bmatrix} 6 \\ 10 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.2 \\ 9.6 \\ 11.4 \end{bmatrix}$$

$$S(z) = \frac{1}{1 + e^{-z}}$$

$$a1 = S(z1) = \begin{bmatrix} 0.77 \\ 0.99 \\ 0.99 \end{bmatrix}$$

$$z2 = \begin{bmatrix} 1.8 & 0.9 & -0.5 \\ 0.5 & -1.1 & 0.2 \end{bmatrix} \begin{bmatrix} 0.77 \\ 0.99 \\ 0.99 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.782 \\ -0.506 \end{bmatrix}$$

$$\hat{y} = \begin{bmatrix} 0.96 \\ 0.03 \end{bmatrix}$$

سوال 6)

۱۹

الف)

$$Tan(h) \Rightarrow \begin{cases} 1 \Rightarrow 0.76 \\ 0.5 \Rightarrow 0.46 \\ 0.5 \Rightarrow 0.46 \end{cases}$$

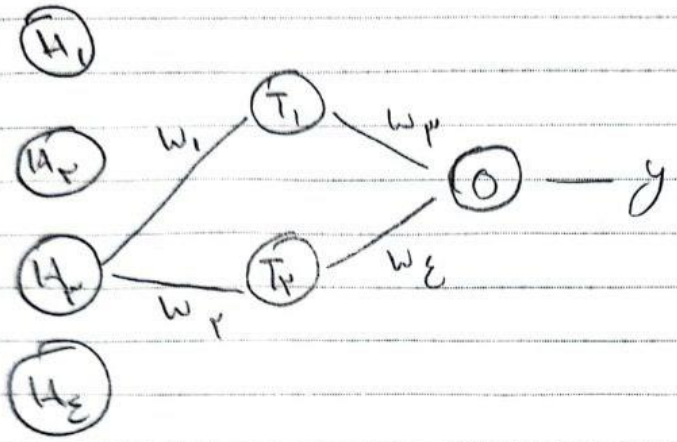
$$z_1 = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0.76 \\ 0.46 \\ 0.46 \\ 0.46 \end{bmatrix} = \begin{bmatrix} 1.34 \\ 0.74 \\ 1.34 \\ 0.74 \end{bmatrix}$$

$$S(z_1) \simeq 0.69$$

$$z_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0.69 \\ 0.69 \\ 0.69 \\ 0.69 \end{bmatrix} = \begin{bmatrix} [illegible] \\ 2.74 \\ 2.74 \\ [illegible] \end{bmatrix}, \quad S(z_2) \simeq 0.94$$

$$z_3 = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0.94 \\ 0.94 \end{bmatrix} = 1.88, \quad S(z_3) \simeq 0.86$$

ب)



$$w_{new} = w_{old} - (LR \circ \nabla C)$$

$$\frac{\partial C}{\partial w_1} = \frac{\partial C}{\partial y_p} \times \frac{\partial y_p}{\partial o} \times \frac{\partial o}{\partial T_1} \times \frac{\partial T_1}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$

$$\frac{\partial C}{\partial w_2} = \frac{\partial C}{\partial y_p} \times \frac{\partial y_p}{\partial o} \times \frac{\partial o}{\partial T_2} \times \frac{\partial T_2}{\partial z_2} \times \frac{\partial z_2}{\partial w_2}$$

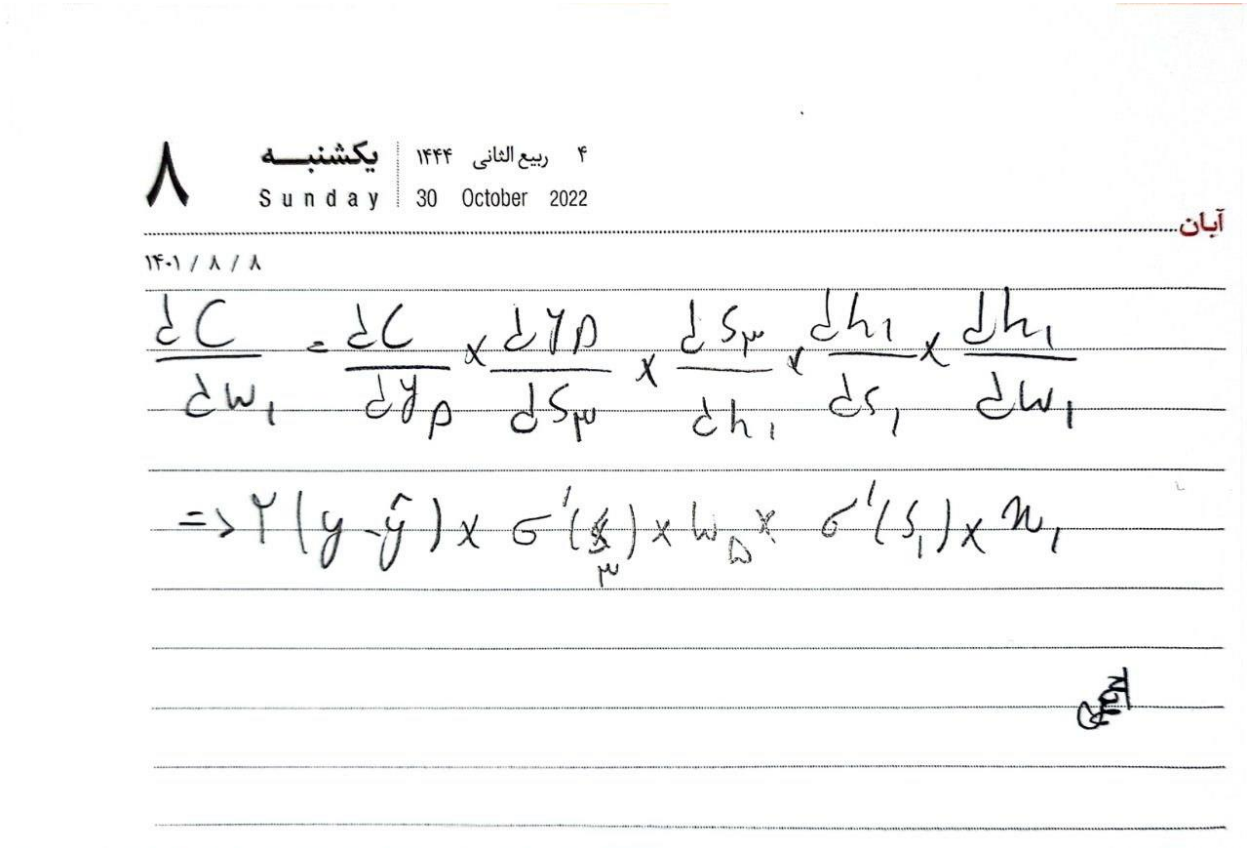
$$\Rightarrow 2(y_p - y^*) \times w_3 \times s'(o) \times s'(z_1) \times H_3$$

$$\Rightarrow 2(0,\text{۸۹}-1) \times 1 \times (0,\text{۸۹} \times (1-0,\text{۸۹})) \times (0,\text{۹۴} \times (1-0,\text{۹۴})) \times 0,\text{۴۹}$$

$$\simeq -0,001$$

$$w_1^+ , w_2^+ = 1 - (0,2 \times 0,001) = 1,0002$$

سوال امتیازی)



این مقدار درواقع gradient descent است و از این مقدار در back propagation و آپدیت کردن مقادیر وزن ها استفاده میشود