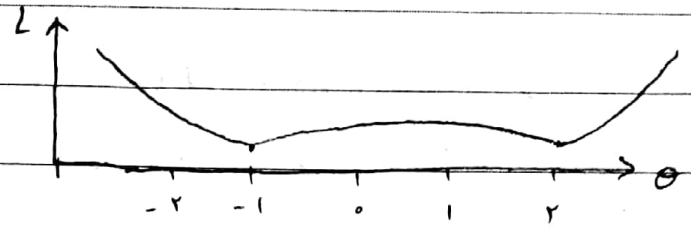
محمدرضا امیرخواه

موضوع: تمارین سری اول درس امنیت در یادگیری ماشین تاریخ: .....

① الف) $L = \frac{1}{N}\sum_{i=1}^{N} |y_i - f_\theta(x_i)|$ ، $f_\theta(x_i) = x_i(\theta^2 - \theta)$

$D = \{(3,6), (2,0), (1,4)\}$

$L = \frac{1}{3}\left[|6 - 3(\theta^2-\theta)| + |0 - 2(\theta^2-\theta)| + |4 - (\theta^2-\theta)| \right]$

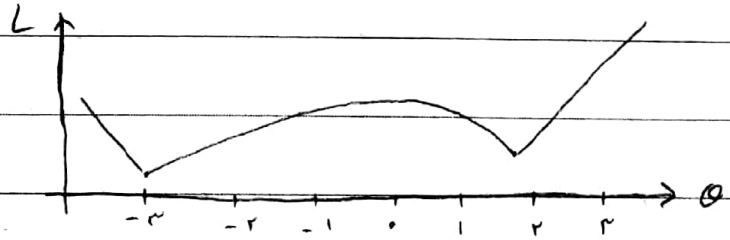


مشاهده می‌شود که تابع L دارای دو کمینه محلی که برابری هم هستند می‌باشد. / بله زیرا با استفاده از gd به یک کمینه محلی می‌رسیم که در اینجا هر دو کمینه محلی، برابری نیز هستند.

ب) $L = \frac{1}{N}\sum_{i=1}^{N} |y_i - f_\theta(x_i)|$. $f_\theta(x_i) = \ln\left(1 + e^{x_i\theta}\right)$

$D = \{(3,4), (-1,3), (1,0)\}$

$L = \frac{1}{3}\left[|4 - \ln(1+e^{3\theta})| + |3 - \ln(1+e^{-\theta})| + |-\ln(1+e^{\theta})| \right]$



مشاهده می‌شود که تابع L دارای دو کمینه محلی که یکی از آن‌ها برابری است می‌باشد / خیر زیرا با توجه به وزن‌دهی اولیه ممکن است به کمینه محلی که برابری نیست همگرا شویم.

ج) معمولاً تعداد زیادی کمینه محلی وجود دارد که برابری نیستند و ممکن است به این کمینه‌های محلی همگرا شویم. برای کم کردن احتمال رخداد این پدیده می‌توان از تکنیک‌هایی زیر بهره برد:

مقداردهی اولیه وزن‌ها به صورت تصادفی، تنظیم متغیر یادگیری Regularization، Momentum، learning rate

روش‌های Ensemble، Batch Normalization

موضوع : ..... تاریخ : .....

(۲) آ) با نرمال کردن داده های ورودی، بازه مقادیر ویژگی ها به یک بازه یکسان و منفی معمولاً [۱،۱-] نگاشت می شوند. این بازه مقادیر یکسان برای ویژگی ها باعث می شود که الگوریتم بهینه سازی بتواند راحت تر و سریعتر به بهینه محلی / سراسری همگرا شود.

در واقع با نرمال کردن داده ها، تمام مسائل الگوریتم gradient descent معادلتر می شوند و می توان از مقادیر learning rate های بزرگتر استفاده کرد که باعث همگرایی سریعتر مدل می شود. علاوه بر این نرمال کردن داده ها، مقادیر گرادیان ها را نیز کوچک نگه می دارد و به این طریق انفجار / ناپایدار شدن گرادیان نیز کاهش می یابد.

همچنین نرمال کردن داده ها می تواند باعث بهبود دقت نهایی مدل نیز شود، زیرا با این کار اهمیت / وزن های یکسان برای تمامی ویژگی ها در نظر گرفته می شود و به این صورت یک ویژگی صرفاً چون مقادیر بزرگتری دارد، نمی تواند باعث اهمیت و اثر دادن بیشتر مدل به آن شود.

ب) استفاده از منظم سازها که معمولاً به صورت اضافه کردن ترم برای وزن ها به تابع هزینه است، تلویحاً باعث می شود که مقادیر وزن ها خیلی بزرگ نشوند. از این رو به این روش ها weight decay نیز می گویند. این جلوگیری از افزایش مقادیر وزن ها باعث می شود که مدل، نسبت زیادی به یک یا چند ویژگی اهمیت ندهد و به این طریق از Overfit شدن مدل جلوگیری می کند. البته روش های دیگری مانند dropout و Batch Normalization نیز از منظم ساز دارند. به طور کلی استفاده از منظم سازها باعث افزایش بهبود عملکرد مدل روی داده هایی که در زمان آموزش آن ها را ندیده، می شود. در نتایج منظم سازها باعث می شود که وزن ویژگی هایی که خیلی برای دسته بندی مهم نیستند کاهش یابد و به این طریق generalizability مدل افزایش می یابد.

ج) هر تعداد لایه پرسپترون اما با تابع فعال ساز خطی، هیچ قدرت اضافه ای نسبت به یک لایه خطی ندارد. زیرا هیچ non linearity به مدل اضافه نشده است. شبکه دوم اما چون non linearity دارد قدرت مدل کردن روابط غیر خطی را نسبت به شبکه اول دارد. بنابراین قدرت مدل کردن شبکه دوم بیشتر است، آزمودنی شبکه اول ممکن است حتی از شبکه دوم بهتر باشد ولی هیچ برتری نسبت به مدل های خطی تک لایه ندارد.

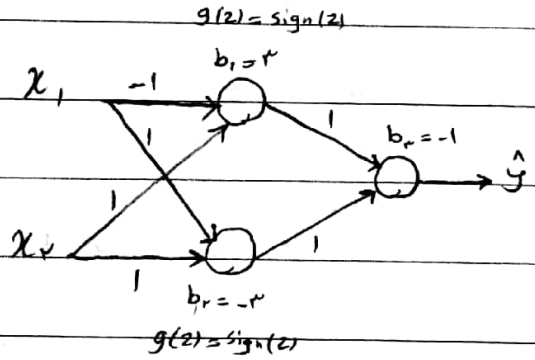
موضوع : ..... تاریخ : .....

(۳) آ) مشخص است که هیچ مرز تصمیم گیری خطی که فقط یک سلول فعال است، نمی تواند داده های موجود در شکل را به طور کامل و صحیح دسته بندی کند.

ب) مرز تصمیم گیری که به صورت $X_2 = |X_1 - ۳|$ باشد می تواند تمامی داده ها را به صحیح دسته بندی کند. بنابراین در لایه اول نیاز به دو نورون داریم که هریک از آنها وظیفه جدا سازی هریک از نیم خط های این مرز تصمیم را دارند. در لایه آخر هم نیاز به یک گیت AND داریم که مشخص کند یک نقطه داده بالای هر دو نیم خط باشد. برای اینکه در لایه آخر از گیت AND استفاده کنیم لازم است که تابع فعال ساز در لایه اول، تابع sign(z) باشد تا بتوانیم خروجی لایه اول را به صورت باینری تفسیر کنیم.

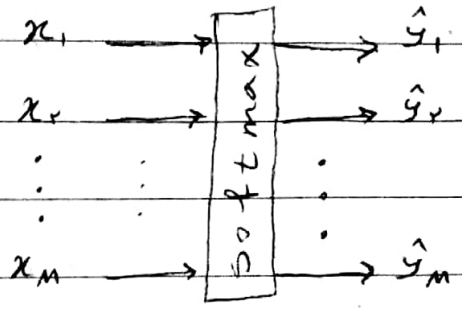
معادله نیم خط اول : $-X_1 + X_2 + ۳ = ۰$

معادله نیم خط دوم : $X_1 - X_2 - ۳ = ۰$



موضوع : ..... تاریخ : .....

④



$$L(\hat{y}, y) = -\sum_{i=1}^{M} y_i \lg \hat{y}_i \xrightarrow{\text{if true label} = t} L(\hat{y}, y) = -\lg \hat{y}_t$$

$$\hat{y}_t = \frac{e^{x_t}}{\sum_{m=1}^{M} e^{x_m}}$$

$$\frac{\partial L}{\partial x_j} = \frac{\partial L}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial x_j} \qquad L = -\lg \hat{y}_t \Rightarrow \frac{\partial L}{\partial \hat{y}_t} = -\frac{1}{\hat{y}_t}$$

$$\text{if } j \neq t \Rightarrow \frac{\partial \hat{y}_t}{\partial x_j} = \frac{-e^{x_j} e^{x_t}}{(\sum_{m=1}^{M} e^{x_m})^2} = -\hat{y}_j \hat{y}_t$$

$$\text{if } j = t \Rightarrow \frac{\partial \hat{y}_t}{\partial x_j} = \frac{e^{x_t} \sum_{m=1}^{M} e^{x_m} - e^{2x_t}}{(\sum_{m=1}^{M} e^{x_m})^2} = \hat{y}_j (1 - \hat{y}_j)$$

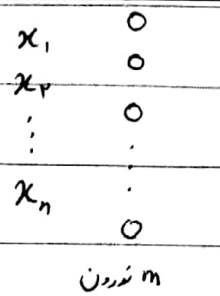
$$\Rightarrow \frac{\partial L}{\partial x_j} = \begin{cases} \hat{y}_j & ; j \neq t \\ \hat{y}_j - 1 & ; j = t \end{cases} \Rightarrow \boxed{\frac{\partial L}{\partial x_j} = \hat{y}_j - y_j}$$

موضوع : ..... تاریخ : .....

ه) آ) اگر تعداد نورون های لایه پنهان را m در نظر گیریم

بنابراین، از آنجایی که $x^{(i)} \in \mathbb{R}^{n\times 1}$ آنگاه باید

$W_1 \in \mathbb{R}^{m\times n}$



همچنین از آنجایی که مسئله دسته بندی باینری است باید در

لایه خروجی یک نورون داشته باشیم بنابراین باید $W_2 \in \mathbb{R}^{1\times m}$

و از آنجایی که هر نورون یک بایاس دارد بنابراین $b_1 \in \mathbb{R}^{m\times 1}$ و $b_2 \in \mathbb{R}$

ب) ابعاد وزن ها و بایاس ها به تعداد داده های ورودی شبکه وابسته نیست بنابراین ابعاد آن ها تغییری نمی کند.

ج)

$$L = \frac{1}{m}\left[\sum_{i=1}^{m} -y^{(i)}\log\hat{y}^{(i)} - (1-y^{(i)})\log(1-\hat{y}^{(i)})\right]$$

$$\frac{\partial L}{\partial \hat{y}^{(i)}} = \frac{1}{m}\left[\sum_{i=1}^{m} -\frac{y^{(i)}}{\hat{y}^{(i)}} + \frac{1-y^{(i)}}{1-\hat{y}^{(i)}}\right] = \frac{1}{m}\left[\sum_{i=1}^{m} \frac{\hat{y}^{(i)} - y^{(i)}}{\hat{y}^{(i)}(1-\hat{y}^{(i)})}\right]$$

$$\frac{\partial \hat{y}^{(i)}}{\partial z_2} = \sigma(z_2)\left(1-\sigma(z_2)\right) = \hat{y}^{(i)}\left(1-\hat{y}^{(i)}\right)$$

$$\frac{\partial z_2}{\partial a_1} = W_2 \qquad \frac{\partial a_1}{\partial z_1} = \begin{cases} 1 & ; z_1 > 0 \\ 0.01 & ; z_1 < 0 \end{cases} \qquad \frac{\partial z_1}{\partial w_1} = x^{(i)}$$

۶) الف) از آنجایی که منظم‌ساز L_2 باعث می‌شود که وزن‌های شبکه خیلی زیاد نشوند و در واقع آن‌ها را کاهش می‌دهد، به منظم‌ساز L_2، Weight decay نیز گفته می‌شود.

ب) $$L = \frac{1}{N}\|Xw - y\|^2 + \lambda \|w\|_2^2$$

$$\Rightarrow \frac{\partial L}{\partial w} = \frac{2}{N} X^T(Xw - y) + 2\lambda w = 0$$

$$\Rightarrow X^T X w - X^T y + \lambda N w = 0$$

$$\Rightarrow w(X^T X + \lambda N I) = X^T y \Rightarrow w = (X^T X + \lambda N I)^{-1} X^T y$$

ج) ابتدا ثابت می‌کنیم ماتریس $X^T X$ ماتریس Positive semi definite است. شرط PSD بودن ماتریس M یادآوری: M positive semi definit $\Leftrightarrow v^T M v \geq 0$ for all v

حال داریم، $v^T X^T X v = (Xv)^T (Xv) = \|Xv\|^2 \geq 0$

بنابراین $X^T X$ ماتریس PSD است یعنی تمام eigen value های آن بزرگتر مساوی صفر هستند.

حال ثابت می‌کنیم ماتریس $\lambda N I$ ماتریس Positive definite است. شرط PD بودن ماتریس M یادآوری: M positive definite $\Leftrightarrow v^T M v > 0$ for all $v \neq \vec{0}$

حال داریم، $v^T \lambda N I v = \lambda N \|v\|^2 > 0 \quad ; v \neq \vec{0}$

بنابراین ماتریس $X^T X$ PSD است و ماتریس $\lambda N I$ ماتریس PD است. می‌دانیم جمع یک ماتریس PSD با یک ماتریس PD از نوع ماتریس PD است. ($\lambda N I$ به ازای $\lambda > 0$)

بنابراین ماتریس $X^T X + \lambda N I$ یک ماتریس PD است یعنی تمام eigen value های آن بزرگتر از صفر هستند. بنابراین این ماتریس full rank و همواره معکوس‌پذیر است.