
بسم الله الرحمن الرحیم

یادگیری عمیق

مدرس: محمدرضا محمدی

زمستان ۱۴۰۱

شبکه‌های عصبی خطی

Linear Neural Networks

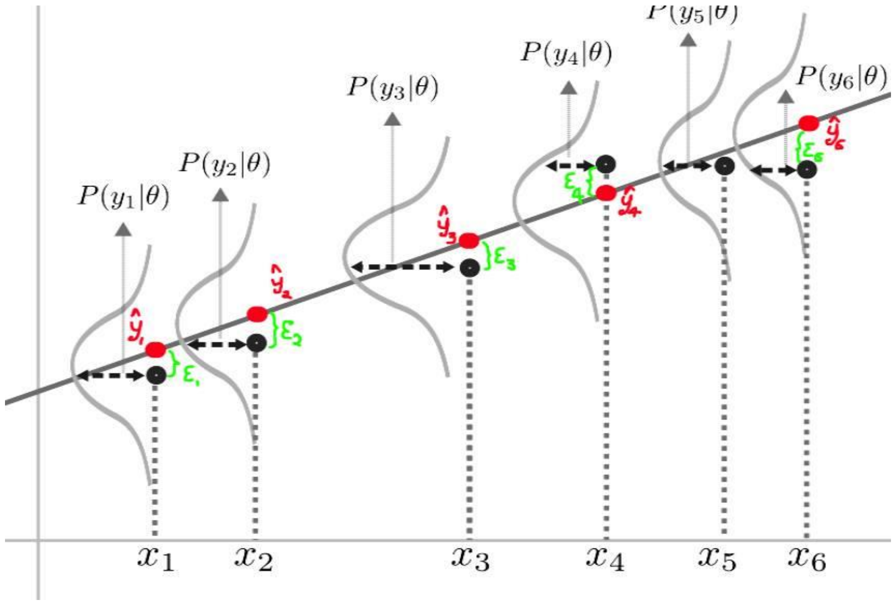
رگرسیون خطی با ML

- فرض کنید مشاهدات از یک مدل خطی همراه با نویز نرمال (گاوسی) جمع‌آوری شده‌اند

$$\hat{y} = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b + \epsilon \ , \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

$$P(y \mid \mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(y - \mathbf{w}^{\mathrm{T}}\mathbf{x} - b\right)^2\right)$$

$$P(\mathbf{y} \mid \mathbf{X}) = \prod_{i=1}^{n} P\left(y^{(i)} \mid \mathbf{x}^{(i)}\right)$$



$$-\log P(\mathbf{y} \mid \mathbf{X}) = \sum_{i=1}^{n} \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\left(y^{(i)} - \mathbf{w}^{\mathrm{T}}\mathbf{x}^{(i)} - b\right)^2 \Rightarrow \min \sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

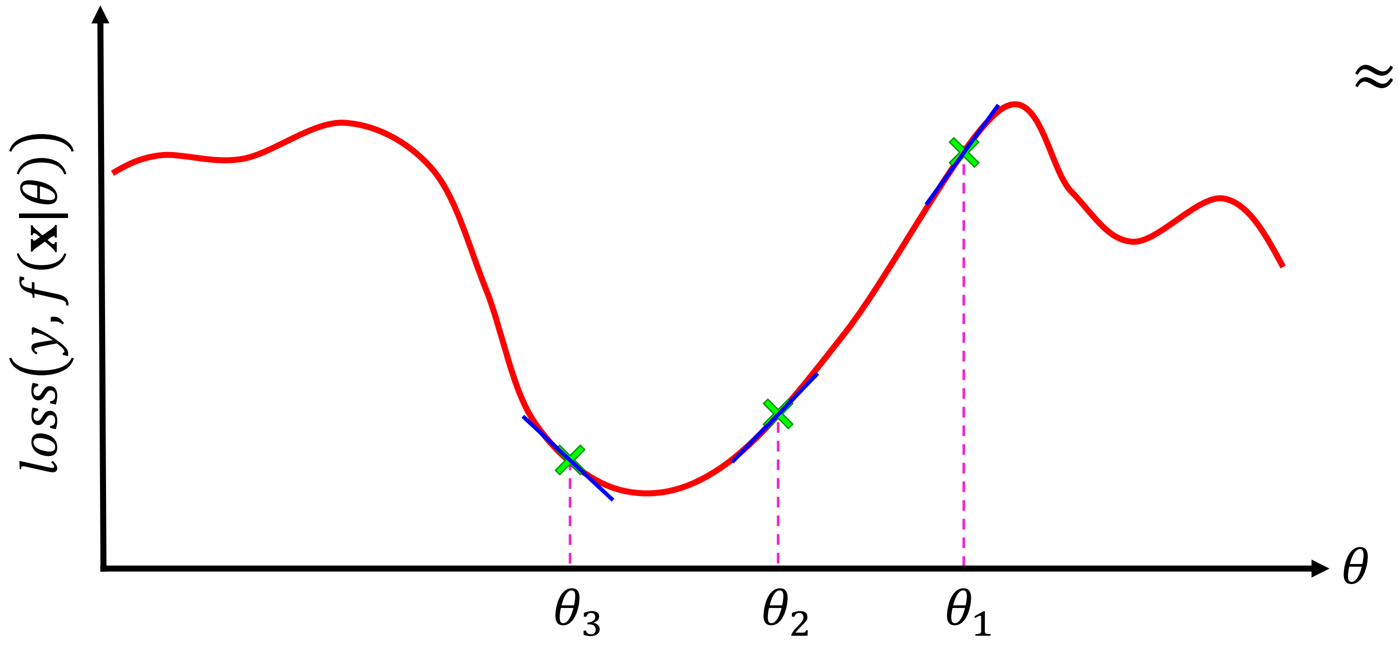
بهینه‌ساز گرادیان کاهشی

- با یک نقطه اولیه شروع می‌کنیم و در هر گام در جهتی حرکت می‌کنیم که منجر به کاهش تابع شود

$$L(\theta + \Delta\theta) = L(\theta) + \Delta\theta \frac{\partial L(\theta)}{\partial \theta} + \frac{(\Delta\theta)^2}{2!} \frac{\partial^2 L(\theta)}{\partial \theta^2} + \cdots$$

$$\approx L(\theta) + \Delta\theta \frac{\partial L(\theta)}{\partial \theta}$$

- در خلاف جهت گرادیان حرکت می‌کنیم



دسته‌بندی باینری

- در بسیاری از مسائل یادگیری ماشین نیاز به پیش‌بینی مقدار یک متغیر باینری است
- شبکه‌های عصبی باید تنها یک مقدار را پیش‌بینی کنند $P(y = 1 \mid \mathbf{x}) \in [0,1]$
- بهتر است خروجی محدود به بازه $[0,1]$ باشد

	Two Class Classification	
$y \in \{0,1\}$	**1 or Positive Class**	**0 or Negative Class**
Email	Spam	Not Spam
Tumor	Malignant	Benign
Transaction	Fraudulent	Not Fraudulent

$$\text{e.g.}\ \ P(y = 1|\mathbf{x}) = \max\{0, \min\{1, \mathbf{w}^{\mathrm{T}}\mathbf{x} + b\}\}$$

- بهینه‌سازهای مبتنی بر گرادیان نمی‌توانند پارامترهای مطلوب چنین شبکه‌ای را بیابند
  - مشتق تابع ضرر برای داده‌هایی که کاملا اشتباه پیش‌بینی شوند ۰ است

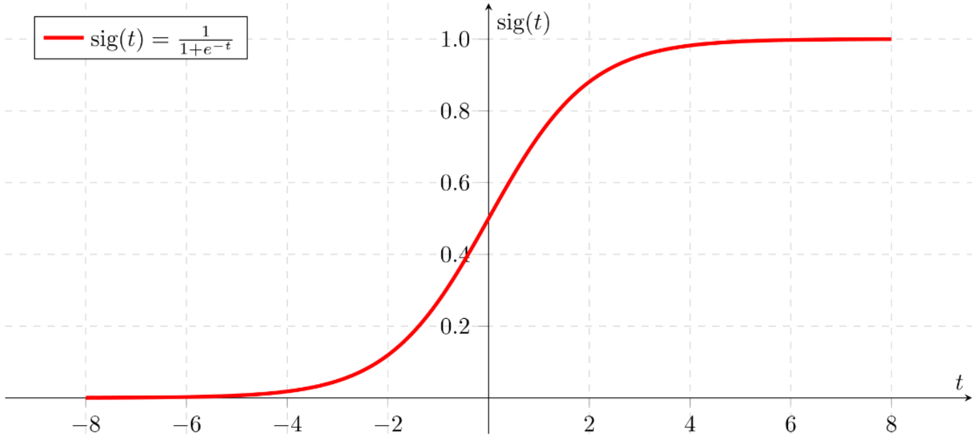
دسته‌بندی باینری

- بهتر است از تابعی برای محدود کردن خروجی استفاده کنیم که مشتق آن صفر نشود
- تابع سیگموئید برای این منظور پرکاربرد است $P(y=1 \mid \mathbf{x}) = \sigma\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}+b\right)$
- آیا تابع ضرر MSE در این حالت مناسب است؟

$$l = \left(y - \sigma(o)\right)^2, \qquad o = \mathbf{w}^T\mathbf{x} + b$$

- مثال عددی: اگر $o = 5.6$ و $y = 0$

$$\frac{dl}{do} = -2\left(y-\sigma(o)\right)\sigma(o)\left(1-\sigma(o)\right) = 0.0073$$
$$\approx -2(0-1)1(1-1)$$



Maximum Likelihood

$$-\log P(\mathbf{y} \mid \mathbf{X}) = -\sum_{i=1}^{n} \log P\left(y^{(i)} \mid \mathbf{x}^{(i)}\right)$$

- y یک متغیر باینری است و احتمال $P(y \mid \mathbf{x})$ به صورت زیر قابل بیان است

$$P(y \mid \mathbf{x}) = \begin{cases} \hat{y} & \text{if } y = 1 \\ 1 - \hat{y} & \text{if } y = 0 \end{cases} = \hat{y}^{y}(1 - \hat{y})^{1-y}$$

Bernoulli distribution

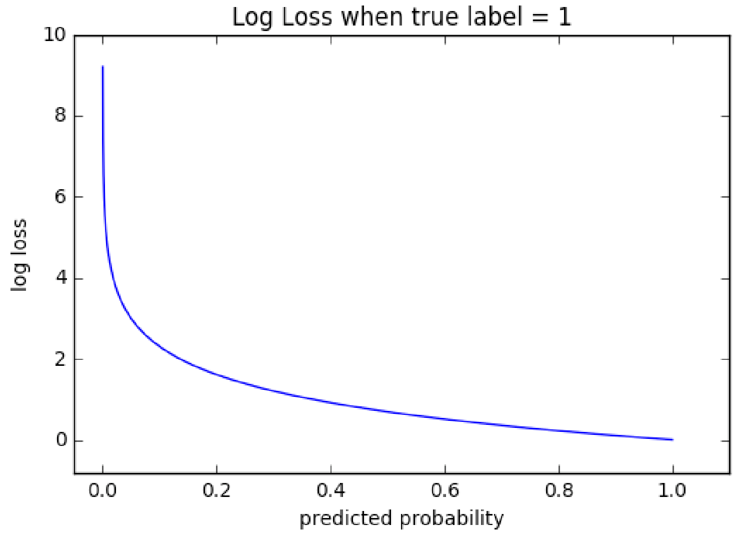
$$-\log P(\mathbf{y} \mid \mathbf{X}) = -\sum_{i=1}^{n} y^{(i)} \log \hat{y}^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right)$$

- تابع ضرر binary cross-entropy نامیده می‌شود که برای مسائل دسته‌بندی باینری پرکاربرد است

دسته‌بندی باینری

- بهتر است از تابعی برای محدود کردن خروجی استفاده کنیم که مشتق آن صفر نشود
- تابع سیگموئید برای این منظور پرکاربرد است $P(y=1\mid\mathbf{x})=\sigma\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}+b\right)$
- با استفاده از ML تابع ضرر binary cross-entropy بدست می‌آید

$$l=-y\log\sigma(o)-(1-y)\log\left(1-\sigma(o)\right)$$



- مثال عددی: اگر $o=5.6$ و $y=0$

$$\frac{dl}{do}=-\frac{-\sigma(o)\left(1-\sigma(o)\right)}{1-\sigma(o)}=\sigma(o)=0.9963$$

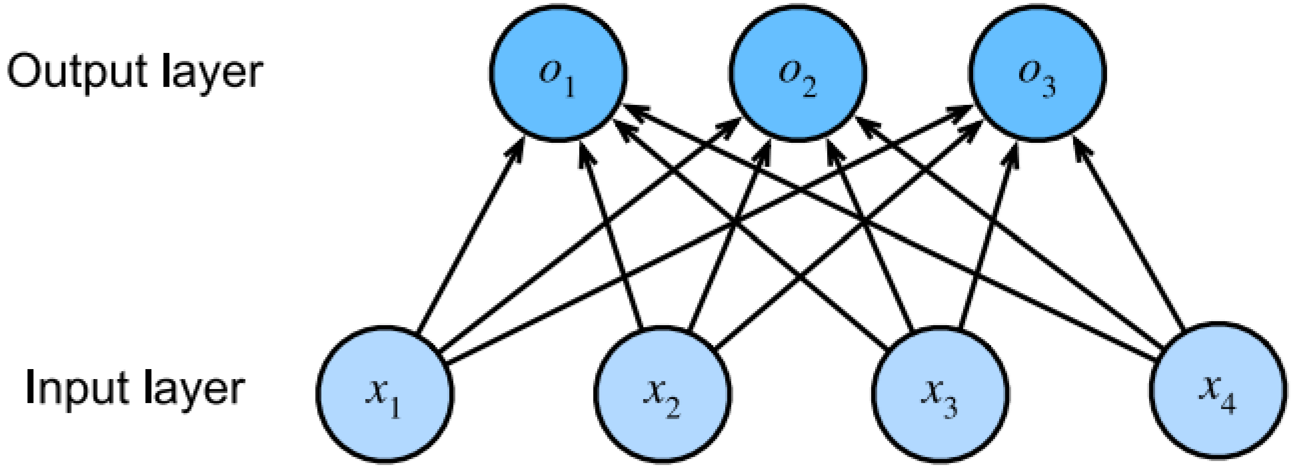
دسته‌بندی چندکلاسه

- تعیین دسته یک نمونه جدید از میان چندین دسته دارای کاربردهای بسیار فراوانی است
- می‌توان برای هر کلاس یک نورون قرار داد تا احتمال تعلق نمونه به آن کلاس را تخمین بزند
- با استفاده از یک لایه خطی احتمال غیرنرمالیزه **o** را پیش‌بینی می‌کنیم که logit نامیده می‌شوند
- چطور نرمالیزه کنیم؟

$$o_1 = w_{11}x_1 + w_{12}x_2 + w_{13}x_3 + w_{14}x_4 + b_1$$

$$o_2 = w_{21}x_1 + w_{22}x_2 + w_{23}x_3 + w_{24}x_4 + b_2$$

$$o_3 = w_{31}x_1 + w_{32}x_2 + w_{33}x_3 + w_{34}x_4 + b_3$$



دسته‌بندی چندکلاسه

- می‌خواهیم احتمال پسین مربوط به هر کلاس را تخمین می‌زنیم $P(y = i \mid \mathbf{x}) \in [0,1]$
- برای آنکه مقادیر خروجی از جنس احتمال باشند (هر کدام نامنفی و مجموع برابر با ۱) می‌توانیم از تابع فعال‌سازی Softmax استفاده کنیم که تعمیم تابع Sigmoid است

$$\mathbf{o} = \mathbf{W}^{\mathrm{T}}\mathbf{x} + \mathbf{b}$$

$$\hat{y}_i = \mathrm{softmax}(\mathbf{o})_i = \frac{\exp(o_i)}{\sum_{k=1}^{q} \exp(o_k)}$$

- تابع ضرر متناسب با این تابع فعال‌سازی، categorical cross-entropy است

$$l(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{i=1}^{q} y_i \log \hat{y}_i$$

https://drive.google.com/file/d/11FmT5RE3ujFax4ggIMtq62dYEIOOx8B4/view?usp=sharing

مشتق softmax

$$l(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{q} y_i \log \frac{\exp(o_i)}{\sum_{k=1}^{q} \exp(o_k)} = \sum_{i=1}^{q} y_i \log \sum_{k=1}^{q} \exp(o_k) - \sum_{i=1}^{q} y_i o_i$$

$$= \log \sum_{k=1}^{q} \exp(o_k) \sum_{i=1}^{q} y_i - \sum_{i=1}^{q} y_i o_i = \log \sum_{k=1}^{q} \exp(o_k) - \sum_{i=1}^{q} y_i o_i$$

$$\partial_{o_i} l(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\exp(o_i)}{\sum_{k=1}^{q} \exp(o_k)} - y_i = \mathrm{softmax}(\mathbf{o})_i - y_i$$

- مشابه با رگرسیون خطی، گرادیان تابع ضرر نسبت به خروجی بخش خطی برابر با میزان اختلاف پیش‌بینی با مقدار واقعی است