Regression: how much

Classfication: which one.

3-1 One-hot encoding (独热编码)

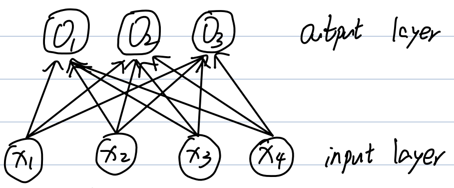
One-hot encoding 是一个 vector, 其 component 的数量就是类别数.

比如: 我们希望把一张图分入三类中: 猫, 鸡, 狗

那么 $Y \in \{\underset{\text{猫}}{\langle 1,0,0\rangle}, \underset{\text{鸡}}{\langle 0,1,0\rangle}, \underset{\text{狗}}{\langle 0,0,1\rangle}\}$

如果我们有 4个 feature 即每个 sample $\vec{x} = \langle x_1, x_2, x_3, x_4\rangle$

那么我们的 NN 是:



(即为 single layer NN)

$$\left.\begin{aligned} O_1 &= \vec{W_1}^T\vec{x} + b_1 \\ O_2 &= \vec{W_2}^T\vec{x} + b_2 \\ O_3 &= \vec{W_3}^T\vec{x} + b_3 \end{aligned}\right\} \Longrightarrow \vec{O} = W\cdot\vec{x} + \vec{b}.$$

这里用3个 affine function 得到3个 weights vector (每個4个component) 和 3个 bias.

合成一个 $\boxed{3\times4\ W\ \text{matrix}}$ 和一个 $\boxed{3\times1\ \vec{b}\ \text{vector}}$.

$$O_i = W_{i1}x_1 + W_{i2}x_2 + W_{i3}x_3 + W_{i4}x_4 + b_i$$

$$\boxed{\vec{O} = \begin{bmatrix} O_1 \\ O_2 \\ O_3 \end{bmatrix}}$$

这里由 one-hot code 得出的 $\vec{O}$ 是 3个 logit (未经规范化的预测)

我们希望得到的是: $\langle 1,0,0\rangle$, $\langle 0,1,0\rangle$, $\langle 0,0,1\rangle$ 这样的 hard labels.

但我们首先要得到由 prob 形成的 soft label.

比如 $\langle 0, 0.5, 0.5\rangle$ 表示 50%为鸡, 50%为狗

然后再把 soft labels 转化 hard labels.

然而这里 $\vec{O}$ 甚至不是一个 soft label, 因为它 components 和不为 1, 并且 components 值可以为负.

因而我们需要先把 $\vec{O}$ 转为 soft label 再转化为 hard label.

3-2 Softmax function.

softmax 可以把 $\vec{O}$ 变为总和为1的 $[0,1]$ 之间的数.

$\vec{O} = \begin{bmatrix} O_1 \\ \vdots \\ O_q \end{bmatrix}$ 的 components

$$\hat{y} = \mathrm{softmax}(\vec{O}), \quad \text{where } \hat{y}_j = \frac{\exp(O_j)}{\sum_q \exp(O_q)}$$

$$\text{即} \quad \begin{bmatrix} O_1 \\ O_2 \\ O_3 \end{bmatrix} \longmapsto \begin{bmatrix} \frac{e^{O_1}}{e^{O_1}+e^{O_2}+e^{O_3}} \\ \frac{e^{O_2}}{e^{O_1}+e^{O_2}+e^{O_3}} \\ \frac{e^{O_3}}{e^{O_1}+e^{O_2}+e^{O_3}} \end{bmatrix}$$

input: n个 sample, d个 features

$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_d^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_d^{(2)} \\ \vdots & & & \\ x_1^{(n)} & x_2^{(n)} & \dots & x_d^{(n)} \end{bmatrix} \in \mathbb{R}^{n\times d}$$

q个类别 $\left(\hat{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_q \end{bmatrix}\right)$

$$W = \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1q} \\ W_{21} & W_{22} & \cdots & W_{2q} \\ \vdots & & & \\ W_{d1} & W_{d2} & \cdots & W_{dq} \end{bmatrix} \in \mathbb{R}^{d\times q} \qquad \vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_q \end{bmatrix} \in \mathbb{R}^{1\times q}$$

(广播后为 $n\times q$, n个相同 $\vec{b}$)

$$O = XW + \vec{b} = \underbrace{\begin{bmatrix} | & | & & | \\ \vec{O_1} & \vec{O_2} & \cdots & \vec{O_q} \\ | & | & & | \end{bmatrix}}_{}\ \Big\}\ n\text{个 samples.} \quad \in \mathbb{R}^{n\times q}$$

(广播后)

$$\vec{O_1} = \begin{bmatrix} W_{11}x_1^{(1)} + W_{21}x_2^{(1)} + \dots + W_{d1}x_d^{(1)} + b \\ W_{11}x_1^{(2)} + W_{21}x_2^{(2)} + \dots + W_{d1}x_d^{(2)} + b \\ \vdots \\ W_{11}x_1^{(n)} + W_{21}x_2^{(n)} + \dots + W_{d1}x_d^{(n)} + b \end{bmatrix}$$

$\vdots$

$$\vec{O_q} = \begin{bmatrix} W_{1q}x_1^{(1)} + W_{2q}x_2^{(1)} + \dots + W_{dq}x_d^{(1)} + b \\ W_{1q}x_1^{(2)} + W_{2q}x_2^{(2)} + \dots + W_{dq}x_d^{(2)} + b \\ \vdots \\ W_{1q}x_1^{(n)} + W_{2q}x_2^{(n)} + \dots + W_{dq}x_d^{(n)} + b \end{bmatrix}$$

即: 由 design matrix X predict 得到

Consider: $\{X, Y\}$ 有 n 个 samples.

— 个 sample i 由一个 $\vec{x}^{(i)} \in \mathbb{R}^d$ 和 $\vec{y}^{(i)} \in \mathbb{R}^q$

(d个 features) (q个 class)

$[x_1\ x_2 \cdots x_d]^T \quad [y_1\ y_2 \cdots y_q]^T$

那么 $\boxed{P(Y|X) = \prod_{i=1}^{n} P(\vec{y}^{(i)}|\vec{x}^{(i)})}$

3-2 Cross-entropy loss.

注意到 $P(\vec{y}^{(i)}|\vec{x}^{(i)}) = \prod_{j=1}^{q} (\hat{y}_j)^{y_j}$

为什么呢？因为 $\vec{y}^{(i)}$ 只有一个 component 为 1，其余都是 0.

所以 $\vec{y}^{(i)}$ 中，$y_j = 0$ 的 component 上，$(\hat{y}_j)^{y_j} = 1$

$y_j = 1$ 的 component 上，$(\hat{y}_j)^{y_j} = \hat{y}_j$

因而 $\prod_{j=1}^{q} (\hat{y}_j)^{y_j} = 1 \cdot 1 \cdots \hat{y}_{correct} \cdot 1 \cdot 1 \cdots 1 = \hat{y}_{correct}$

因为 softmax 算法，所有这些 $\hat{y}_j$ 的和为 1

所以 P 会保留预测到的结果正确为 $\vec{y}^{(i)}$ 的概率

也就是在给定 $\vec{x}$，$\vec{y}$ 下模型正确从 $\vec{x}$ 推出 $\vec{y}$ 的概率即 $P(\vec{y}^{(i)}|\vec{x}^{(i)})$

那么同样，求 negative log likelihood 使其 minimize:

$$-\ln(P(Y|X)) = \sum_{i=1}^{n} -\ln(P(\vec{y}^{(i)}|\vec{x}^{(i)}))$$

我们已说到 $P(\vec{y}^{(i)}|\vec{x}^{(i)}) = \prod_{j=1}^{q} (\hat{y}_j)^{y_j}$

$$\text{因而} = \sum_{i=1}^{n} -\ln\left(\prod_{j=1}^{q} (\hat{y}_j)^{y_j}\right)$$

$$= \sum_{i=1}^{n} \left(-\sum_{j=1}^{q} y_j \ln \hat{y}_j\right)$$

$\Downarrow$

我们定义这个式子为 $\boxed{\text{cross entropy loss.}}$

$$l(\vec{y}, \hat{\vec{y}}) = -\sum_{j=1}^{q} y_j \log \hat{y}_j$$

如果进一步拆解：$l(\vec{y}, \hat{\vec{y}}) = -\sum_{j=1}^{q} y_j \ln \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)}$

$$= \ln \sum_{k=1}^{q} \exp(o_k) - \sum_{j=1}^{q} y_j o_j$$

$$\Rightarrow \boxed{\partial_{o_j} l(\vec{y}, \hat{\vec{y}})} = \frac{\exp(o_j)}{\sum_{k=1}^{q} \exp(o_k)} - y_j = \boxed{\text{softmax}(\vec{o})_j - y_j}$$