1. logistic regression

liner regression function : $z = w^T x$

use sigmoid function to classify $y = \sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-w^T x}}$

The cross entropy loss for logistic function

$$L_{CE}(y,t) = \begin{cases} -\log y & , t=1 \\ -\log(1-y) & , t=0 \end{cases}$$

and the function can write as $L_{CE}(y,t) = -t\cdot\log y - (1-t)\cdot\log(1-y)$

when $t=1$, $1-t=0$, the loss for target 0 (the part $-(1-t)\log(1-y)$ is 0 , when $t=0$, $1-t=1$, the loss for target 1 (the part $-t\log y$ is 0 , so it is equivalent with the function before.

so, the cost function : $J(w) = \frac{1}{N}\cdot\sum_{i=1}^{N}\left[-t_i\log y_i - (1-t_i)\log(1-y_i)\right]$

the gradient for the cost function

$$\frac{\partial}{\partial w_j} = \frac{1}{N}\sum_{i=1}^{N}\left[-t_i\cdot\frac{1}{y_i}\cdot\frac{\partial y}{\partial w}(y_i) + (1-t_i)\cdot\frac{1}{1-y_i}\cdot\frac{\partial y}{\partial w}(y_i)\right]$$

$$\frac{\partial y}{\partial w} = \left(\frac{1}{1+e^{-w^T x}}\right)'$$ is difficult to directly solve

let $f(x) = y_i$

let $f(x)\cdot g(x) = 1$, $g(x) = 1+e^{-x}$

$f'(x)g(x) + g'(x)f(x) = 0$

$f'(x) = -\frac{1}{g(x)}\cdot f(x)\cdot g'(x) = f(x)\cdot(1-f(x))\cdot x$

so $\frac{\partial y}{\partial w}(y) = y_i\cdot(1-y_i)\cdot x_j^i$

so the gradient of cost function: $\frac{\partial J}{\partial w_j} = \frac{1}{N}\sum_{i=1}^{N}\left[-t_i(1-y_i)\cdot x_j^i + (1-t_i)\,y_i\cdot x_j^i\right]$

$$\frac{\partial J}{\partial w_j} = -\frac{1}{N}\sum_{i=1}^{N}\left[(t_i - y_i)\cdot x_j^i\right]$$

so gradient descent: $w_j \leftarrow w_j - \alpha\frac{\partial J}{\partial w_j} = w_j - \frac{\alpha}{N}\sum_{i=1}^{N}(y_i - t_i)\cdot x_j^i$

2. gradient for softmax

softmax used to classify k classes

$$Y(x_i) = \begin{bmatrix} P(t_i=1 \mid x^i; w) \\ P(t_i=2 \mid x_i; w) \\ \vdots \\ P(t_i=k \mid x_i; w) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{w_j^T x_i}} \begin{bmatrix} e^{w_1^T x^i} \\ e^{w_2^T x^i} \\ \vdots \\ e^{w_k^T x^i} \end{bmatrix}$$

cost function: $L(W) = -\frac{1}{N}\left[\sum_{i=1}^{N}\sum_{m=1}^{k} I(t_i=j)\cdot \log \frac{e^{w_j^T x_i}}{\sum_{p=1}^{k} e^{w_p^T x_i}}\right]$

The $I(\ )$ represent the value true or false (0 or 1)

$$\frac{\partial L(w)}{\partial w_j} = -\frac{1}{N}\cdot\frac{\partial}{\partial w_j}\left[\sum_{i=1}^{N}\sum_{m=1}^{k} I(t_i=j)\cdot \log\frac{e^{w_j^T\cdot x^i}}{\sum_{p=1}^{k} e^{w_p^T x_i}}\right]$$

$$= -\frac{1}{N}\frac{\partial}{\partial w_j}\left[\sum_{i=1}^{N} I(t_i=j)\cdot\sum_{m=1}^{k}\left(w_j^T\cdot x_i - \log\sum_{p=1}^{k} e^{w_p^T\cdot x_j}\right)\right]$$

$$\cancel{= -\frac{1}{N}\cdot\sum_{i=1}^{N} I(t_i=j)\cdot\sum_{m=1}^{k}\left(x_i - \frac{e^{w_j^T x_i}}{\sum_{p=1}^{k} e^{w_p^T\cdot x_i}}\right)}$$

$$\cancel{= -\frac{1}{N}}$$

$$= -\frac{1}{N}\sum_{i=1}^{N} I(t_i=j)\cdot\left(x_i - \sum_{m=1}^{k}\cdot\frac{e^{w_j^T\cdot x_i}}{\sum_{p=1}^{k} e^{w_p^T x_i}}\right)$$

$$= -\frac{1}{N}\cdot x_i\cdot\sum_{i=1}^{N}\left(I(t_i=j) - \sum_{m=1}^{k} I(t_i=j)\cdot\frac{e^{w_j^T\cdot x_i}}{\sum_{p=1}^{k} e^{w_p^T\cdot x_i}}\right)$$

Because of the sum of $t_i=j$ only have the term $t_i=j$

so the formulas:

$$\frac{\partial L(w)}{\partial w_j} = -\frac{1}{N}\cdot\sum_{i=1}^{N} x_i\cdot\left(I(t_i=j) - \frac{e^{w_j^T\cdot x_i}}{\sum_{p=1}^{k} e^{w_p^T\cdot x_i}}\right)$$

$$= -\frac{1}{N}\left[\sum_{i=1}^{N} x_i\cdot\left(I(t_i=j) - p(t_i=j \mid x_i; w)\right)\right]$$

gradient

gradient descent

$$w_j \leftarrow w_j - \alpha\frac{\alpha}{N}\cdot\sum_{i=1}^{N}\left(p(t_i=j \mid x_i; w) - I(t_i=j)\right)\cdot x_i$$

3. Compare SVM and softmax

The loss function for SVM

$$L(w,b,\alpha)=\frac{|w|^2}{2}+\sum_{i=1}^{N}\alpha_i\cdot(1-t_i(w^Tx_i+b))$$

equivalent to

$$L(w,b,\alpha)=\underbrace{\sum_{i=1}^{N}\cdot[1-t_i(w^Tx_i+b)]_+}_{\text{hinge loss}}+\lambda|w|^2$$

$$\sum_{i=1}^{N}[1-t_i(w^Tx_i+b)]_+=\sum_{i=1}^{N}\max(0,1-t_i(w^Tx_i+b))$$

Note that: $[z]_+=\begin{cases} z, & z>0 \\ 0, & z\le 0\end{cases}$

so the loss function of SVM, can write as

$$L_i=\sum_{j\neq y_i}\max(0,s_j-s_{y_i}+1)$$

where s_j is the score for class j, s_{y_i} is the score for correct class y_i

~~the~~

The single loss for softmax

$$L_i=-\log P(Y=y_i|X=x_i)=-\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

use a example, to classify the animals, there are 3 scores.
cat: 2.2 dog: 1.8 bird: 1.5, the true class is cat

Loss SVM $=\max(0,1.8-2.2+1)+\max(0,1.5-2.2+1)=0.9$

Loss softmax: $P(cat)=\dfrac{e^{2.2}}{e^{2.2}+e^{1.8}+e^{1.5}}=0.4614$ ~~P(dog)~~

$$L=-\log(0.4614)=0.7718$$

SVM consider whether the score of correct class exceeds by a margin the score of other class. But softmax the scores for all classes, convert them to probabilities

SVM is more satisfied with a local objective as long as the score not go beyond the margin, (only determine the closet to the result)

Softmax never satisfy, a wrong classification must result in a lower score (probabilistic result)

In conclusion, svm suitable to classify one-to-many tasks, softmax suitable for mutually exclusive tasks.