HW03

Bhishan Poudel

QN1 $\boxed{\text{QN 1}}$

maximum Likelihood

(Derive MLE of event rate parameter λ)

Let $x_1, x_2, \ldots x_n$ be iid poisson random variables with prob mass function,

pmf of poisson distn
$$\boxed{p(x_i, \lambda) = \frac{e^{-\lambda} \lambda^{x_i}}{x_i!}}$$

Now, the joint prob of all variables x_i, called likelihood function, is given by

Likelihood
$$\boxed{L(\lambda) = \prod_{i=1}^{n} p(x_i, \lambda)}$$

$$= \prod_{i=1}^{n} \frac{e^{-\lambda} \lambda^{x_i}}{x_i!}$$

The log likelihood of pmf is,

Log Likelihood
$$\boxed{\ln L(\lambda) = \ln \prod_{i=1}^{n} \frac{e^{-\lambda} \lambda^{x_i}}{(x_i)!}}$$

$$= \sum_i \ln \frac{e^{-\lambda} \lambda^{x_i}}{x_i!}$$

$$= \sum_i [-\lambda + x_i \ln\lambda - \ln x_i!]$$

$$\boxed{\ln L(\lambda) = -n\lambda + \ln\lambda \sum x_i - \sum \ln (x_i!)}$$

To get the maximum likelihood estimate of the parameter λ, we maximize the Log likelihood ($\ln L(\lambda)$) w.r.t. λ.

$$0 = \frac{\partial}{\partial \lambda} \ln L(\lambda)$$

$$= \frac{\partial}{\partial \lambda} \left[-n\lambda + \ln\lambda \sum_{1} x_i - \sum_{1} \ln x_i! \right]$$

$$0 = -n + \frac{\sum_{1} x_i}{\lambda} - 0$$

$$n = \frac{\sum_{1} x_i}{\lambda}$$

MLE of λ

$$\boxed{\hat{\lambda} = \frac{\sum_{i=1}^{n} x_i}{n}}$$

number of samples

Ans

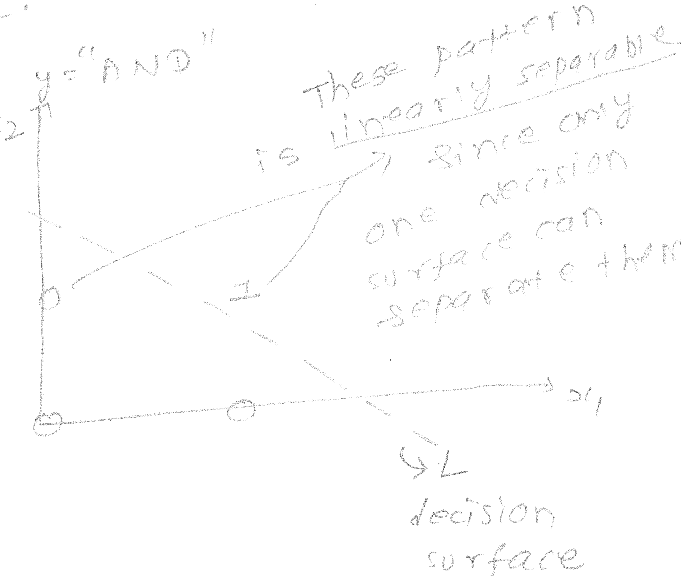
Here, the maximum likelihood estimate of the poisson distribution parameter λ is just the mean (or expectation) of the distribution.

QN 2

XOR problem in Logistic Regression

To describe XOR problem in LR, I shall start with "AND" logic.

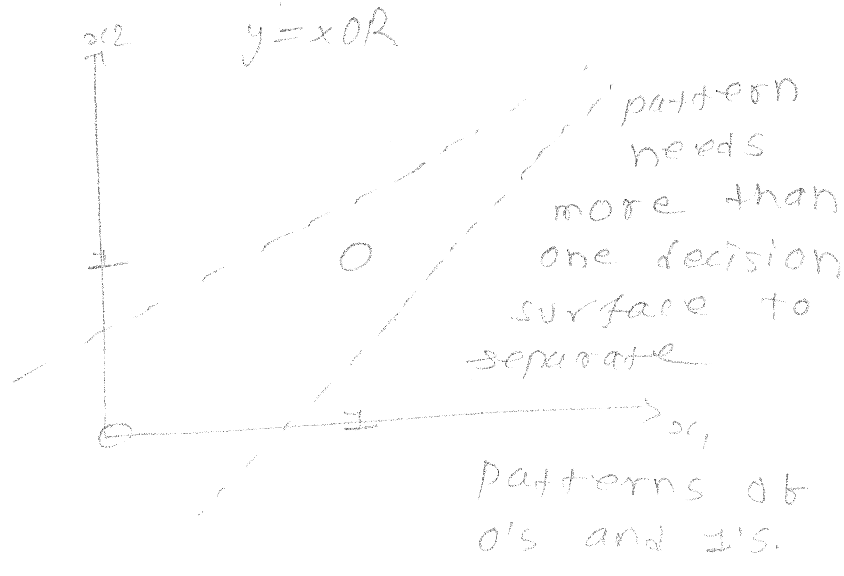
x_1	x_2	$y = x_1 \& x_2$
0	0	0
0	1	0
1	0	0
1	1	1



Now, look at XOR,

Truth table

x_1	x_2	$y = XOR$
0	0	0
0	1	1
1	6	1
1	1	0

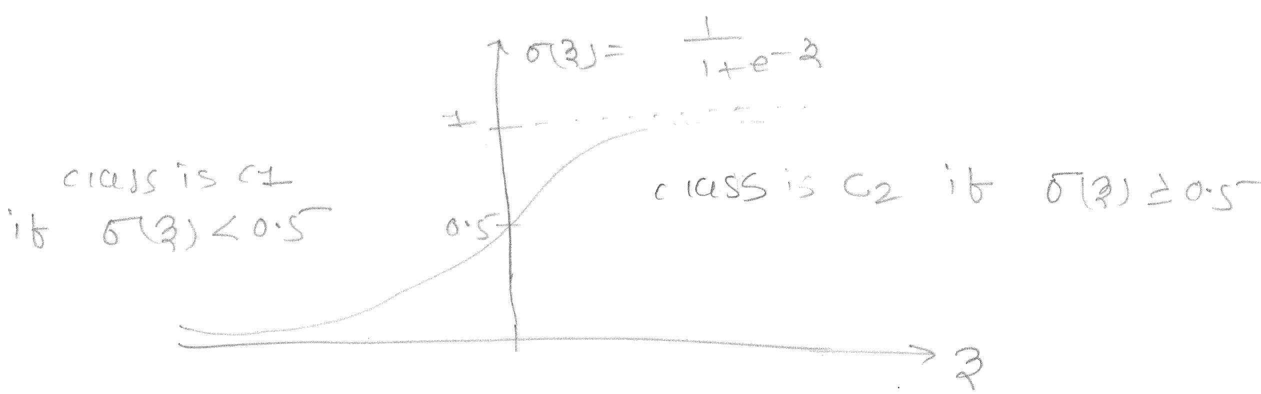


$\therefore$ XOR logic is not linearly separable.

P.T.O.

Now we know that XOR logic is not linearly separable. We also show that Logistic Regression is linear and it can only classify binary classifications that are linearly separable.

The classifier used in LR is sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$



In Logistic Regression,

$$\text{pattern is } 1 \quad \text{if} \quad \frac{1}{1+e^{-w^T x}} \geq \frac{1}{2}$$

$$0 \quad \text{if} \quad \frac{1}{1+e^{-w^T x}} < \frac{1}{2}$$

let's set the classifier equal to separator value:

$$\frac{1}{1+e^{-w^T x}} = \frac{1}{2}$$

$$2 = 1+e^{-w^T x}$$

$$\frac{1}{1} = e^{-w^T x} = \frac{1}{e^{w^T x}}$$

$$e^{w^T x} = 1$$

$$w^T x = \ln 1 = 0$$

$$\Rightarrow \boxed{\sum_i w_i x_i = 0}$$

This means LR is linear classifier.

conclusion: 1) LR is linear classifier
2) XOR problem is non-linear and linearly inseparable

$\therefore$ LR can NOT perfectly classify dataset that follows XOR logic.

Ans

QN3

~~QN3~~

Gradient of Logistic Regression

for linear regression, hypothesis $h = w^T x$

for logistic regression, hypothesis $h = \sigma = \frac{1}{1+e^{-w^T x}}$

$$\boxed{h = \sigma = \frac{1}{1+e^{-wx}}} \quad (1)$$

(I write $w^T x$ as wx since they are just matrix dot product of two matrices)

likelihood function $p(t|w) = \prod_{n=1}^{N} h_n^{t_n} (1-h_n)^{1-t_n}$

-ve log likelihood, E or $J = -\ln p(t|x)$

$$E = -\ln \prod_{n=1}^{N} h_n^{t_n} (1-h_n)^{1-t_n}$$

$$E = \sum_n E_n = \sum_{n=1}^{N} (-t_n \ln h_n - (1-h_n) \ln (1-h_n))$$

for any nth sample the cost can be written as,

LOSS
$$\boxed{E = -t \ln h - (1-h) \ln (1-h)} \quad (2)$$

here $h = \sigma = \sigma(w^T x) = \sigma(wx)$

Before proceeding further, I would derive derivative of sigmoid function.

$$\sigma(z) = \frac{1}{1+e^{-z}} \Rightarrow \frac{d\sigma}{dz} = \frac{-1 \cdot e^{-z} \cdot (-1)}{(1+e^{-z})^2}$$

$$\frac{d\sigma}{dz} = \frac{e^{-z}}{(1+e^{-z})^2} = \frac{1}{1+e^{-z}} \cdot \frac{(e^{-z}+1)-1}{1+e^{-z}}$$

$$= \frac{1}{1+e^{-z}} \cdot \left(\frac{e^{-z}+1}{1+e^{-z}} - \frac{1}{1+e^{-z}} \right)$$

$$= \frac{1}{1+e^{-z}} \cdot \left(1 - \frac{1}{1+e^{-z}} \right)$$

$$\boxed{\frac{d\sigma(z)}{dz} = \sigma(1-\sigma)}$$

similarly $\boxed{\frac{d\sigma(az)}{dz} = a\sigma(1-\sigma)}$ (here at $a(z)$)

$\therefore$ derivative of sigmoid function $\boxed{\frac{d\sigma(az)}{dz} = a\sigma(1-\sigma)}$

Now, going back to the problem, let's calculate the gradient of loss function.

$$\frac{\partial E}{\partial w} = \frac{\partial}{\partial w}\left[-t \ln h - (1-t) \ln (1-h) \right]$$

$$= -\frac{t}{h}\frac{\partial h}{\partial w} - \frac{(1-t)(-)}{1-h} \cdot \frac{\partial h}{\partial w}$$

$$= -\frac{t}{h}\frac{\partial \sigma(wx)}{\partial w} + \frac{1-t}{1-h}\frac{\partial}{\partial w}\sigma(wx)$$

Note $h = \sigma(wx)$

$$= -\frac{t}{\cancel{h}} \times \cancel{h}(1-h) + \frac{1-t}{\cancel{(1-h)}} \times h\cancel{(1-h)}$$

$$\because \frac{d\sigma(az)}{dz} = a\sigma(1-\sigma) \quad \text{and } \sigma \text{ is } h.$$

$$= -tx - \cancel{txh} + xh - \cancel{txh}$$

$$= xh - tx$$

$$\boxed{\frac{\partial E}{\partial w} = (h-t)\,x}$$

(here this is for a single sample, but for total loss we add up all the losses)

$$\Rightarrow \boxed{\nabla_w E = \sum_{n=1}^{N} (h_n - t_n)\, x_n}$$

Q.E.D.

QN4

QN 4

Logistic Regression in sklearn

part a

The cost function for L2 penalized logistic regression is given by,

$$E = \underbrace{\frac{1}{2} w^T w}_{\text{L2 regularizer}} + \underbrace{C \sum_{n=1}^{N} \ln\left(1 + e^{-t_n w^T x_n}\right)}_{\text{cost for LR}} \quad \text{——— (1)}$$

(from sklearn)

we have to show, the -ve log likelihood,

$$-\ln p = \sum_{n=1}^{N} \ln\left(1 + e^{-t_n w^T x_n}\right) \quad \text{——— (2)}$$

Now, the posterior probabilities for class 0 and 1 are,

posterior prob for LR

$$\boxed{\begin{aligned} p(C_1|x) &= \sigma(w^T x) = \frac{1}{1+e^{w^T x}} \\ p(C_2|x) &= 1 - \sigma(w^T x) = \frac{e^{-w^T x}}{1+e^{-w^T x}} \end{aligned}} \quad \text{——— (3)}$$

the likelihood function is,

Likelihood for LR

$$\boxed{p(t|w) = \prod_{n=1}^{N} h_n^{t_n} (1-h_n)^{(1-t_n)}} \quad \text{——— (4)}$$

The -ve log likelihood is,

$$-\ln p(t|w) = -\ln \prod_{n=1}^{N} h_n^{t_n} \cdot (1-h_n)^{1-t_n} = -\sum_{n=1}^{N} \ln\left(h_n^{t_n} \cdot (1-h_n)^{1-t_n}\right)$$

$$= -\sum_{n=1}^{N} \left[\ln\left(h_n^{t_n}\right) + \ln\left(1-h_n\right)^{1-t_n}\right]$$

cost for LR

$$\boxed{E = -\ln p = -\sum_{n=1}^{N} \left[\underbrace{t_n \ln h_n}_{\text{class 0}} + \underbrace{(1-t_n)\ln(1-h_n)}_{\text{class 1}}\right]} \quad (5)$$

where $t_n \in \{0,1\}$

$$\boxed{h_n = \frac{1}{1+e^{-w^T x_n}}}$$

$$1-h_n = 1 - \frac{1}{1+e^{-w^T x_n}}$$

$$= \frac{1+e^{-w^T x_n} - 1}{1+e^{-w^T x_n}}$$

$$= \frac{e^{-w^T x_n}}{1+e^{-w^T x_n}}$$

$$= \frac{1}{e^{w^T x_n} + 1}$$

$$\boxed{1-h_n = \frac{1}{1+e^{w^T x_n}}}$$

$$E = -\ln p$$

$$= -\sum_{n=1}^{N} \left[t_n \ln\left(\frac{1}{1+e^{-w^T x_n}}\right) + (1-t_n)\ln\left(\frac{1}{1+e^{w^T x_n}}\right)\right]$$

$t_n \in \{0,1\}$

$$= -\sum_{n=1}^{N} \ln \frac{1}{\begin{cases} 1+e^{-w^T x_n} & \text{for } t_n = 1 \\ 1+e^{+w^T x_n} & \text{for } t_n = 0 \end{cases}}$$

$$= -\sum_{n=1}^{N} \ln \frac{1}{1+e^{-t_n' w^T x_n}}$$ where new label $t_n' \in \{-1, 1\}$

$$\boxed{E = +\sum_{n=1}^{N} \ln\left(1+e^{-t_n w^T x_n}\right)}$$ Q.E.D.

putting back dummy variable t_n' as t_n and now $t_n \in \{-1, 1\}$

QN 4b Find 'C' parameter of LR cost f^n from sklearn.

Now, the L2 regularized cost function for LR

$$E = E_W + E_D$$

$$= \frac{\lambda}{2} w^T w + (-\ln p)$$

$$\boxed{E = \frac{\lambda}{2} w^T w - \sum_{n=1}^{N} \left[t_n \ln h_n + (1-t_n) \ln (1-h_n) \right]}$$

$t_n \in \{0, 1\}$

from Bishop

$$\boxed{E = \frac{\lambda}{2} w^T w + \sum_{n=1}^{N} \ln (1 + e^{t_n w^T x_n})} \quad (1)$$

$t_n \in \{-1, 1\}$

To get maximum likelihood estimate of parameter w,

$$\vec{w} = \underset{w}{\operatorname{argmin}} \left[\frac{\lambda}{2} w^T w + \sum_{n=1}^{N} \ln (1 + e^{t_n w^T x_n}) \right]$$

$$\vec{w} = \underset{w}{\operatorname{argmin}} \left[\frac{1}{2} w^T w + \frac{1}{\lambda} \sum_{n=1}^{N} \ln (1 + e^{t_n w^T x_n}) \right]$$

but, $\vec{w} = \underset{w}{\operatorname{argmin}} \left[\frac{1}{2} w^T w + C \sum_{n=1}^{N} \ln (1 + e^{t_n w^T x_n}) \right]$

(from sklearn)

$$\therefore \boxed{C = \frac{1}{\lambda}}$$ Ans

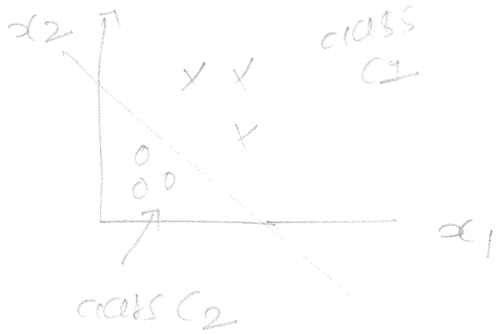
Extra note
To add bias term we use $(w^T + c)$ in E_D term in sklearn.

QNS

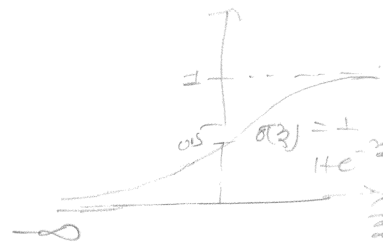
ANS

Softmax Regression as special case of Logistic Regression

FOR LR



(binary classification)



(sigmoid f^n)

the posterior prob of class C_1 is

posterior prob

$$\boxed{p(c_1 \mid x; w) = p(c_1 \mid x) = \sigma(w^T x) = \frac{1}{1+e^{-w^T x}}}$$

the posterior prob of class C_2 is,

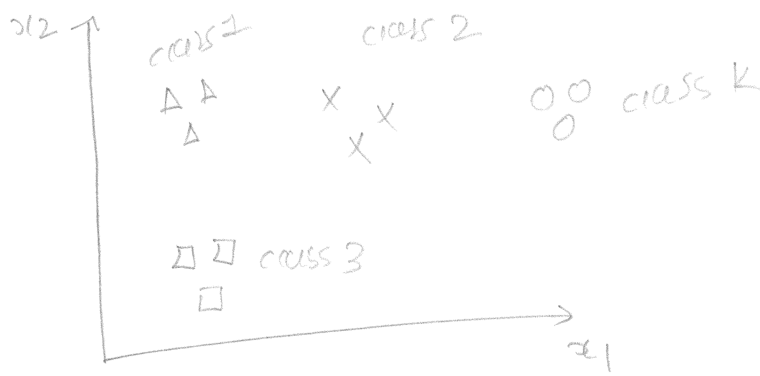
$$p(c_2 \mid x) = 1 - \sigma(w^T x) = \frac{e^{-w^T x}}{1+e^{-w^T x}}$$

hypothesis

the hypothesis for LR,

$$L(w) = \begin{bmatrix} \frac{1}{1+e^{-w^T x}} \\ \frac{e^{-w^T x}}{1+e^{-w^T x}} \end{bmatrix} \begin{array}{l} \to \text{for class 1} \\ \to \text{for class 2} \end{array}$$

Again, For SR (softmax Regression or multiclass Logistic Regression)



total number of samples $= N$

input data X		target
x_1	some values	triangle t_1
x_2		cross t_2
$\vdots$		rectangle
		----
x_N		circle t_N

In softmax Regression, the posterior probability of any class C_K is given by,

$$\boxed{p(C_K|x, w) = \frac{e^{w_K^T x}}{\sum_{K=1}^{K} e^{w_K^T x}}}$$

posterior prob.

[Bishop, 4.3.4 eq 4.104 p. 209 / pdf 227]

note $a_K = w_K^T x$ is called activation f^n for the parameter vector w_K

hyp: $$\boxed{h = \frac{1}{\sum_K e^{w_K^T x}} \begin{bmatrix} e^{w_1^T x} \\ e^{w_2^T x} \\ \vdots \\ e^{w_K^T x} \end{bmatrix}}$$

plus sign $+w_K^T x$

sigmoid is $-$
softmax is $+$

Show $h(w) = h(w-\psi)$

[SR has overparametrization property]

ψ is any fixed vector

$$h(w-\psi) = \frac{e^{(w_k-\psi)^T x}}{\sum_{k=1}^{K} e^{(w_k-\psi)^T x}}$$

$$= \frac{e^{w_k^T x} \cdot e^{-\psi^T x}}{\sum_k e^{w_k^T x} \cdot e^{-\psi^T x}}$$

$$= \frac{e^{w_k^T x} \cdot \cancel{e^{-\psi^T x}}}{\underbrace{\cancel{e^{-\psi^T x}}}_{\text{does not depend on } k} \sum_k e^{w_k^T x}}$$

$$= \frac{e^{w_k^T x}}{\sum_k e^{w_k^T x}}$$

$$\boxed{h(w-\psi) = h(w)}$$

$\therefore$ If we change any parameter vector $w_k \to w_k - \psi$ we get <u>same</u> hypothesis for softmax Regression.

for two-class SR,

$$h = \frac{1}{\sum_k e^{w_k^T x}} \begin{bmatrix} e^{w_1^T x} \\ e^{w_2^T x} \end{bmatrix}$$

$$= \frac{1}{e^{w_1^T x} + e^{w_2^T x}} \begin{bmatrix} e^{w_1^T x} \\ e^{w_2^T x} \end{bmatrix}$$

use overparametrization property of SR
$h(w-\psi) = h(w)$ where $\psi = w_2$

$$= \frac{1}{e^{(w_1-w_2)^T x} + 1} \begin{bmatrix} e^{(w_1-w_2)^T x} \\ 1 \end{bmatrix}$$

write $\vec{w_2} - \vec{w_1} = \vec{w}$

$$= \frac{1}{1+e^{-w^T x}} \begin{bmatrix} e^{-w^T x} \\ 1 \end{bmatrix}$$

$$(h)_{SR} = \begin{bmatrix} \dfrac{e^{-w^T x}}{1+e^{-w^T x}} \\ \\ \dfrac{1}{1+e^{-w^T x}} \end{bmatrix} = (h)_{LR} \ \#$$

proved

this is same as h for LR

QN6

QN6

Gradient of softmax Regression

example of softmax function

$$x = \begin{bmatrix} x_1 = 1 \\ x_2 = 2 \\ x_3 = 3 \\ x_4 = 4 \\ x_5 = 1 \\ x_6 = 2 \\ x_N = 3 \end{bmatrix} \qquad e^{x} = \begin{bmatrix} e^{x_1} = 2.72 \\ e^{x_2} = 7.39 \\ e^{x_3} = 20.09 \\ e^{x_4} = 54.6 \\ \vdots \\ e^{x_N} = 20.09 \end{bmatrix} \qquad \text{softmax}(x) = \begin{bmatrix} e^{x_1}/\text{sum} = 0.024 \\ e^{x_2}/\text{sum} = 0.054 \\ e^{x_4}/\text{sum} = 0.475 \\ \vdots \\ e^{x_N}/\text{sum} = 0.175 \end{bmatrix}$$

sum = 114.98

label 4 has maximum value.

- softmax regression is multiclass regression (classification) scheme where each data sample belongs to one of the classes.

- the posterior probability of class C_K, i.e. the data x belongs to class 'K' is,

posterior prob of class C_K

$$\boxed{p(C_K|x,w) = \frac{e^{w_k^T x}}{\sum_{K=1}^{K} e^{w_K^T x}}} \quad \text{or, } p(t_n|x_n,w) \qquad (1)$$

(Bishop 4.3.4 page. 209/227 eq 4.104)

Likelihood

likelihood is, $$\boxed{L(w) = \prod_{n=1}^{N} p(t_n|x_n,w)} \qquad (2)$$

-ve log likelihood is, $-\ln L(w) = -\ln \prod_n p(t_n|x_n,w)$

-ve log likelihood

$$\boxed{-\ln L(w) = -\sum_{n=1}^{N} \ln\left(\frac{e^{w_K^T x}}{\sum_{K=1}^{K} e^{w_K^T x}}\right)} \qquad (3)$$

Now, the error function (or cost function or objective) for softmax regression is given by

$$E(w) = \frac{1}{N} \cdot -\ln p(c_k)$$

Loss for SR

$$\boxed{E(w) = -\frac{1}{N} \sum_{n=1}^{N} \ln \left(\frac{e^{w_k^T x}}{\sum_{k=1}^{K} e^{w_k^T x}} \right)} \quad (4)$$

This is the loss for the data D, so we write,

Loss from data

$$\boxed{E_D(w) = -\frac{1}{N} \sum_{n=1}^{N} \ln \left(\frac{e^{w_k^T x}}{\sum_{k=1}^{K} e^{w_k^T x}} \right)} \quad (5)$$

Now, we add the L2 regularizer term for each of the weight vectors w_k,

Loss from L2 regularizer

$$\boxed{E_W(w) = \frac{\alpha}{2} \sum_{k=1}^{K} w_k^T w_k} \quad (6)$$

$$E = E_D(w) + E_W(w)$$

Total loss for SR

$$\boxed{E(w) = -\frac{1}{N} \sum_{n=1}^{N} \ln \left(\frac{e^{w_k^T x}}{\sum_{k=1}^{K} e^{w_k^T x}} \right) + \frac{\alpha}{2} \sum_{k=1}^{K} w_k^T w_k} \quad (7)$$

(Regularized softmax Regression)

Now calculate the gradient (derivative) of cost,

$$\frac{\partial}{\partial w} E_W(w) = \frac{\partial}{\partial w_j} \frac{\alpha}{2} \sum_{k=1}^{K} w_k^T w_k$$

$$= \frac{\alpha}{2} \frac{\partial}{\partial w_j} \Big[w_1^T w_1 + w_2^T w_2 + \cdots + \underbrace{w_{j=k}^T w_{j=k}}_{\text{only this term survieves}} + \cdots + w_K^T w_K \Big]$$

$$= \frac{\alpha}{2} \frac{\partial}{\partial w_j} w_j^T w_j \quad (\text{for } j = k\text{th term})$$

$$= \frac{\alpha}{2} \cdot \frac{\partial}{\partial w_j} \|w_j\|^2$$

$$= \frac{\alpha}{2} \, 2 \, w_j$$

$$\frac{\partial E_W}{\partial w_j} = \alpha w_j$$

$$\boxed{\frac{\partial E_W}{\partial w_K} = \alpha w_K}$$ — 6a

again, calculate derivative of E_D,

$$\frac{\partial}{\partial w_j} E_D = -\frac{1}{N} \frac{\partial}{\partial w_j} \sum_{n=1}^{N} y_n \left(\frac{e^{w_k^T x_n}}{\sum_{k=1}^{K} e^{w_k^T x_n}} \right)$$

$$= -\frac{1}{N} \sum_n \frac{\partial}{\partial w_j} \Big[w_k^T x_n - y_n \sum_K e^{w_k^T x_n} \Big]$$

$$= -\frac{1}{N} \sum_{n=1}^{N} \Big[\delta_{jK} x_n - \frac{1 \cdot e^{w_k^T x} \cdot x_n}{\sum_K e^{w_k^T x_n}} \Big]$$

$$\frac{1 \cdot e^{w_k^T x}}{\sum_K e^{w_k^T x_n}} = p(C_K)$$

$$\boxed{\frac{\partial E_D}{\partial w_j} = -\frac{1}{N} \sum_{n=1}^{N} \big(\delta_{jK} - p(C_K)\big) x_n}$$

$$\frac{\partial E_W}{\partial w_K} = \alpha w_K = \alpha w_K^T \quad (\text{write } w \text{ as } w^T)$$

$$\frac{\partial E_D}{\partial w_K} = -\frac{1}{N} \sum_{n=1}^{N} \left(\delta_{Kn} - p(c_K | x_n) \right) x_n$$

$$= -\frac{1}{N} \sum_{n=1} \left(\delta_K(t_n) - p(c_K | x_n) \right) x_n^T$$

so, $\nabla_w E = \nabla_w E_D + \nabla_w E_W$

gradient of regularized softmax Regression

$$\boxed{\nabla_w E = -\frac{1}{N} \sum_{n=1}^{N} \left(\delta_K(t_n) - p(c_K | x_n) \right) x_n^T + \alpha w_K^T}$$

Q.E.D.

Note: example of data for multiclass Regression

CIFAR-10 data

training examples 5 files
test eg 1 file

classes
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

$$X = \begin{array}{c} \\ x_1 \\ x_2 \\ \vdots \\ x_{10K} \end{array} \begin{bmatrix} \underline{1024} & \underline{1024} & \underline{1024} \\ R & G & B \\ & \text{numpy array of uint8} & \\ & & \end{bmatrix}_{10{,}000 \times 3072} \qquad t = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix}_{10{,}000,\,1}$$

$N \times 1 = (10K, 1)$

$t_i \in (c_1, c_2, \cdots, c_{20})$

$$\vec{w} = \begin{bmatrix} w_{11} & w_{12} & w_{1,10K} \\ w_{21} & w_{22} & w_{2,10K} \\ \vdots & & \\ w_{10K,1} & \cdots & w_{10K,10K} \end{bmatrix}_{10K,\,10K}$$

$w^T x = (10K, 10K)(10K, 1) = (10K, 1) = t$

each $w_{ij} \in w_K$ where $w_K = \{ w_1, w_2, \cdots, w_{10} = w_K \}$