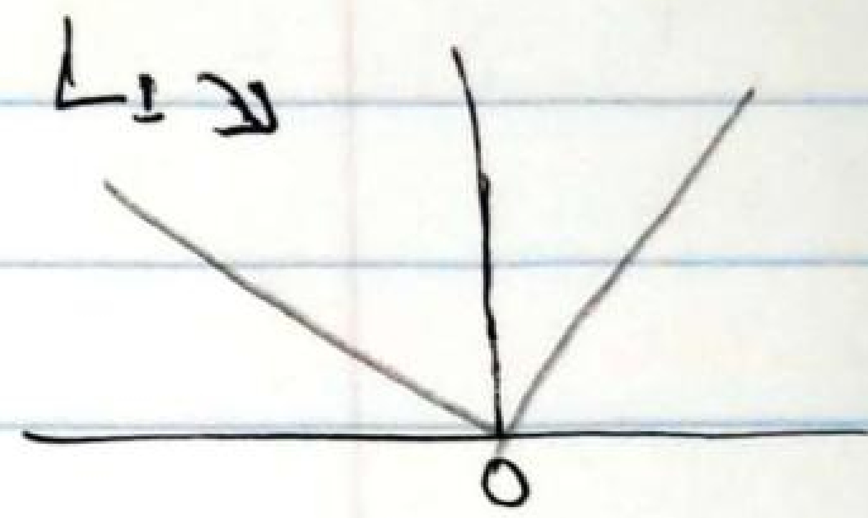
Theoretical Questions.
LOSS

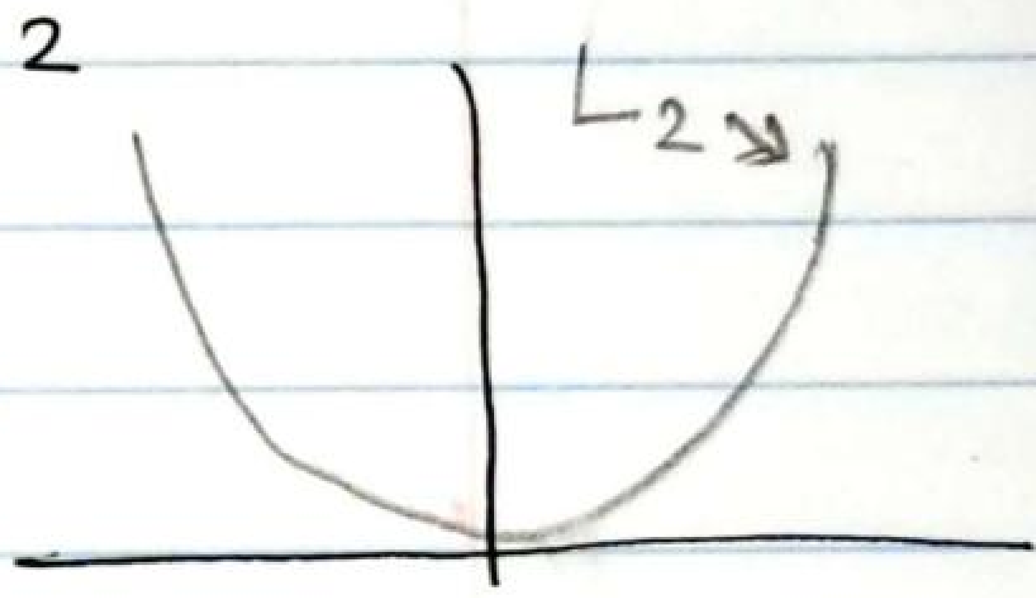
Name → Arinjay Jain
A.no. → A20447307
Email → ajain80@hawk.iit.edu.

1. Mean Absolute Error, L1 loss → sum of absolute differences between our target and predicted variables. $L_1 : L_i(\theta) = \frac{\sum_{j=1}^{K} |\hat{y}_j^{(i)} - y_j^{(i)}|}{n}$



2 Mean Square error, Quadratic Loss, L2 loss.
Commonly used regression loss function.

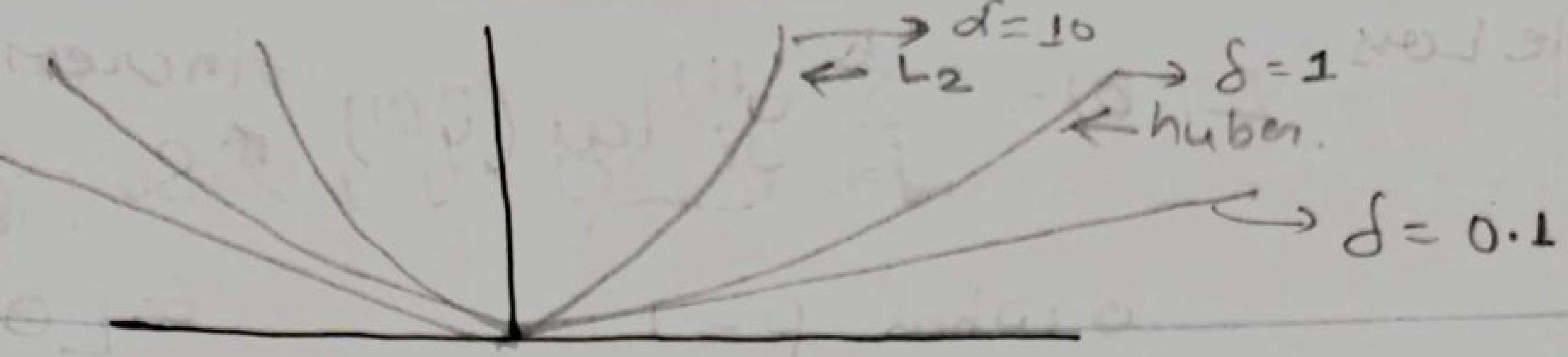
$$L_2 = L_i\theta = \frac{\sum_{j=1}^{K} \left(\hat{y}_j^{(i)} - y_j^{(i)}\right)^2}{n}$$



L_2 loss vs L_1 loss.
Using the squared error is easier to solve when we have small difference in predictions value ($\hat{y}$) and observed value (y). But using (L_1) the absolute error is more robust to outliers.

3. Huber Loss, Smooth mean absolute error.
Huber loss is less sensitive to ~~ot~~ outliers in data than the squared error loss. It's also differentiable at 0. It's basically absolute error which become quadratic when error is small. How small that error has to be to make it quadratic depends on a hyperparameter, δ (delta), which can be tuned. Huber loss approaches L_1 when $\delta \to 0$ and L_2 when $\delta \to \infty$ (large no).

$$L_\delta(y, \underset{\hat{y}}{f(x)}) = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & \text{for } |y-\hat{y}| \le \delta \\ \delta|y-\hat{y}| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

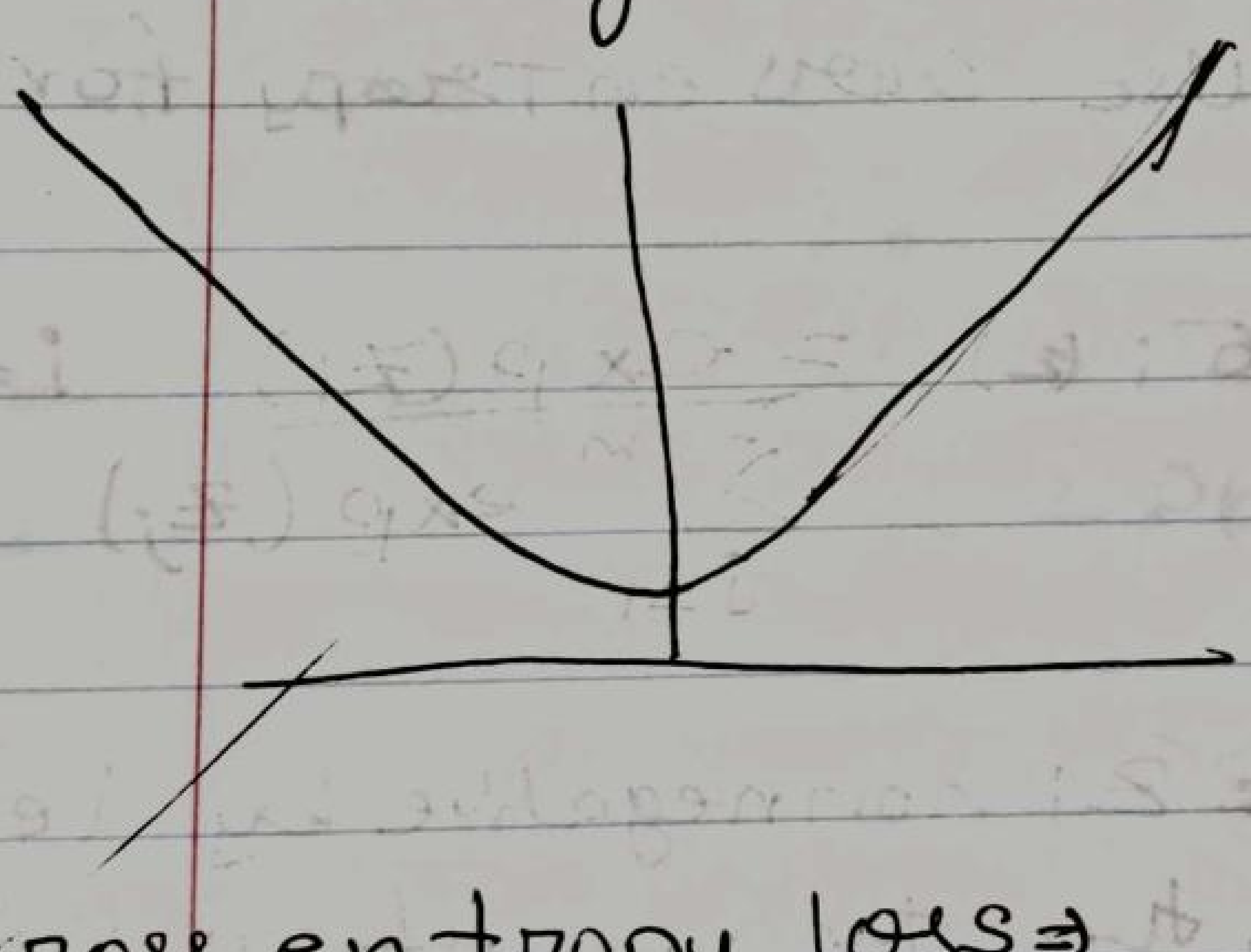


(4) Log-Cosh Loss ⇒

$$L_i(\theta) = \sum_{i=1}^{k} \log(\cosh(\hat{y}_i - y_i))$$

Advantage:- $\log(\cosh(x))$ is approximately equal to $\frac{x^2}{2}$ for small x and to $abs(x) - \log(2)$ for large x. It has all the advantages of huber loss and its twice differentiable every where unlike Huber loss.

$$\log(\cosh(x)) \approx \begin{cases} x^2/2 & \text{if } x \text{ is small} \\ |x| - \log(2) & \text{otherwise.} \end{cases}$$



Q2 Cross entropy loss ⇒

$$\hat{y}_j^{(i)} = P(y=j \mid x^{(i)}) \quad j \in [1, k]$$

we know the likelihood : $L(\theta) = \prod_{i=1}^{m} \prod_{j=1}^{k} \left(P(y=j \mid x^{(i)})\right)^{y_j^{(i)}}$

to derived Cross entrop loss taking the "log" of $L(\theta)$, → $l(\theta)$ ← log-likelihood

$$l(\theta) = -\log L(\theta) = -\sum_{i=1}^{m} \sum_{j=1}^{k} y_j^{(i)} \log(P(y=j / x^{(i)}))$$

$$= -\sum_{i=1}^{m} \sum_{j=1}^{k} y_j^{(i)} \log(\hat{y}_j^{(i)})$$

Simple Loss $L_i(\theta) = -\sum_{j=1}^{k} \underbrace{y_i^{(i)} \log(\hat{y}_j^{(i)})}$ increasing function So $L_i(\theta) \in [0, \infty]$

0 when $p \to 1$

∞ when $p \to 0$.

for the worst case :→ Random class assignment

Probability $P(y=j|x^{(i)}) = 1/k$ → for equals classes.

$$L(\theta) = -\log(P(y=j|x^{(i)}))$$
$$= -\log(1/k) = \log(k)$$

$$\boxed{L(\theta) = \log(k)} \leftarrow \text{bad logs value.}$$

Q 3 Softmax loss.

is a combination of softmax as activation in the output layer and use cross entropy for loss.

equation ⇒ $$\sigma_i(z) = \frac{\exp(z_i)}{\sum_{j=1}^{m} \exp(z_j)} \quad i = 1, \dots, m$$

$$z_i = W_i * x + B_i$$

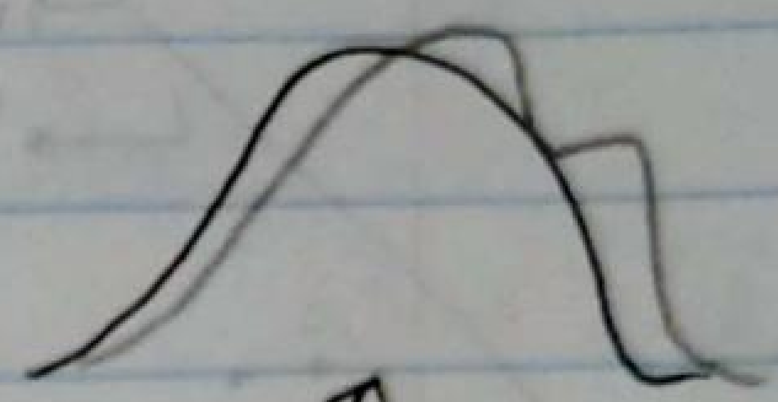
Softmax can make z_i nonnegative by letting them become exponential, then the sum of all items is normalized, now each $O_i = \sigma_i(z)$ can be interpreted as the probability of data x belong to the category i or the likehood.

* In sigmoid we have probabilityes of every output like every k-class outputs. ~~but~~ So in this case will get greater the 1 sum of of all probabilityes, but if we use softmax will get the probability in $(0 \leftrightarrow 1)$.

Q4 Kullback-Liebler loss ⇒

- Kullback-liebler loss divergen measures similarity between distribution.

$$L(\theta) = -\sum \sum y^{(i)} \log\left(\frac{y^{(i)}}{\hat{y}^{(i)}}\right)$$



↑ find the similarity.

for single loss

$$L_i(\theta) = -\sum_{j=1}^{k} y_j^{(i)} \log\left(\frac{y_j^{(i)}}{\hat{y}_j^{(i)}}\right)$$

→ Determin the likehood ratio for entire data set.

$$\frac{p(x_1, \ldots, x_m)}{q(x_1, \ldots, x_m)} = \prod_{i=1}^{m} \frac{p(x_i)}{q(x_i)}$$

$> 1 \rightarrow p(x)$ better model

$< 1 \rightarrow q(x)$ better model.

taking log to change $\pi \rightarrow \Sigma$

$$\sum \log\left(\frac{p(x_i)}{q(x_i)}\right)$$

$> 0 \rightarrow p(x)$ better.

$< 0 \rightarrow q(x)$ better.

Expected value to compute

$$\underset{x \sim p}{E}\left[\log \frac{p(x_i)}{q(x_i)}\right] = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

$$\simeq \sum_i p(x_i) \log\left(\frac{p(x_i)}{q(x_i)}\right) = KL(p \| q)$$

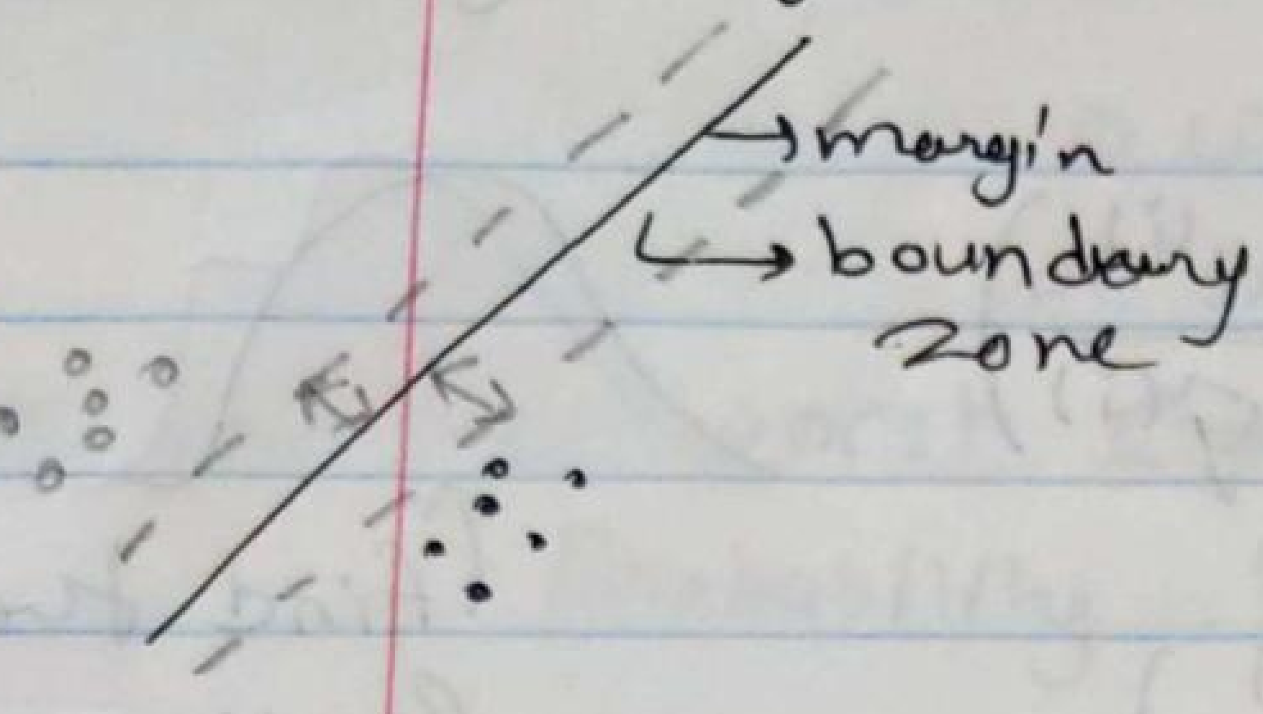
$$= \underbrace{\sum p(x_i) \log p(x_i)}_{-\text{entropy}} - \sum \underbrace{p(x_i) \log q(x_i)}_{\text{cross entropy}}$$

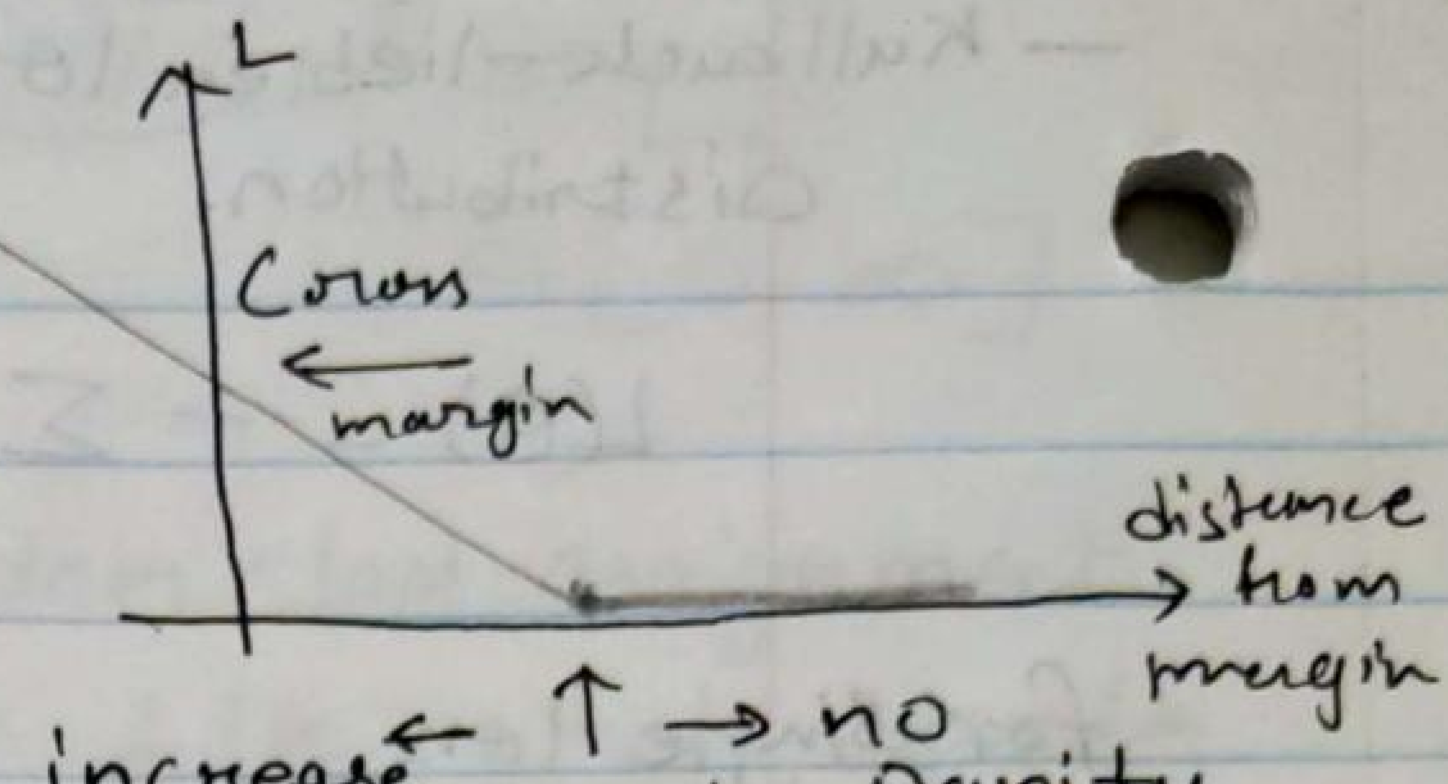
in this case $p(x_i) = y^{(i)}$ and $q(x_i) = \hat{y}^{(i)}$

if $y^{(i)}$ is fixed then mean (− entropy) is not changing

In this case Kullback-Liebler Loss is same as cross entropy loss.

Q5 ⇒ Hinge loss ⇒ SVM (support vector machine) take margin of SVM.





Hing Loss

$$L_i(\theta) = \max\left(0,\ \underbrace{1 - y^{(i)}\hat{y}^{(i)}}\right)$$

positive if prediction agree in sign with margin < 1 or disagree in sign

negative if predictions agree in sign with margin > 1

in distance form $\hat{y}$ - distance from discriminat function

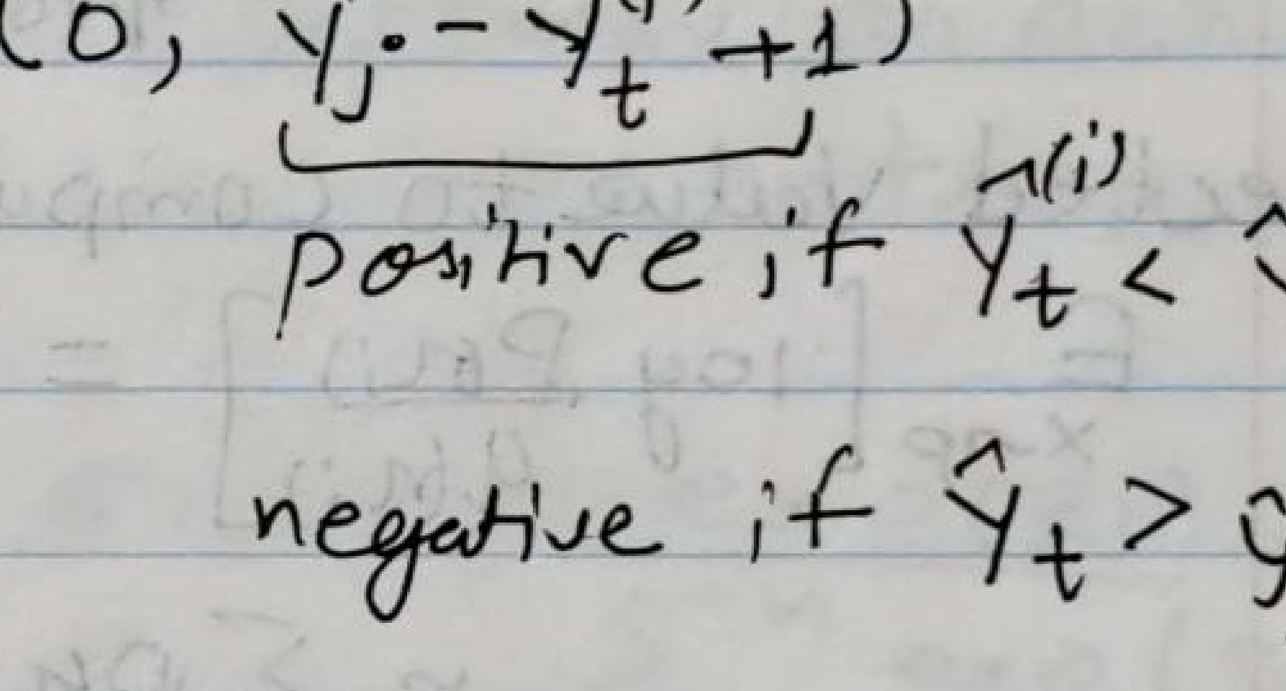
$$L_i = \sum_{j \neq t} \max\left(0,\ \underbrace{\hat{y}_j - \hat{y}_t^{(i)} + 1}\right)$$

Sum over incorrect class labels

positive if $\hat{y}_t^{(i)} < \hat{y}_j^{(i)} + 1$ (incorrect)

negative if $\hat{y}_t > \hat{y}_j^{(i)} + 1$ (correct)

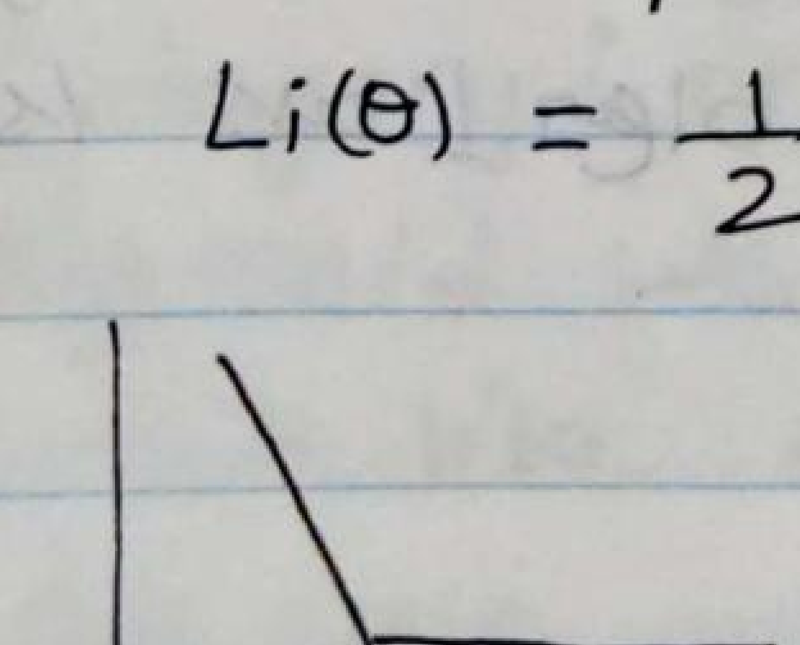
$$L_i(\theta) = \sum_{j \neq y^{(i)}}$$



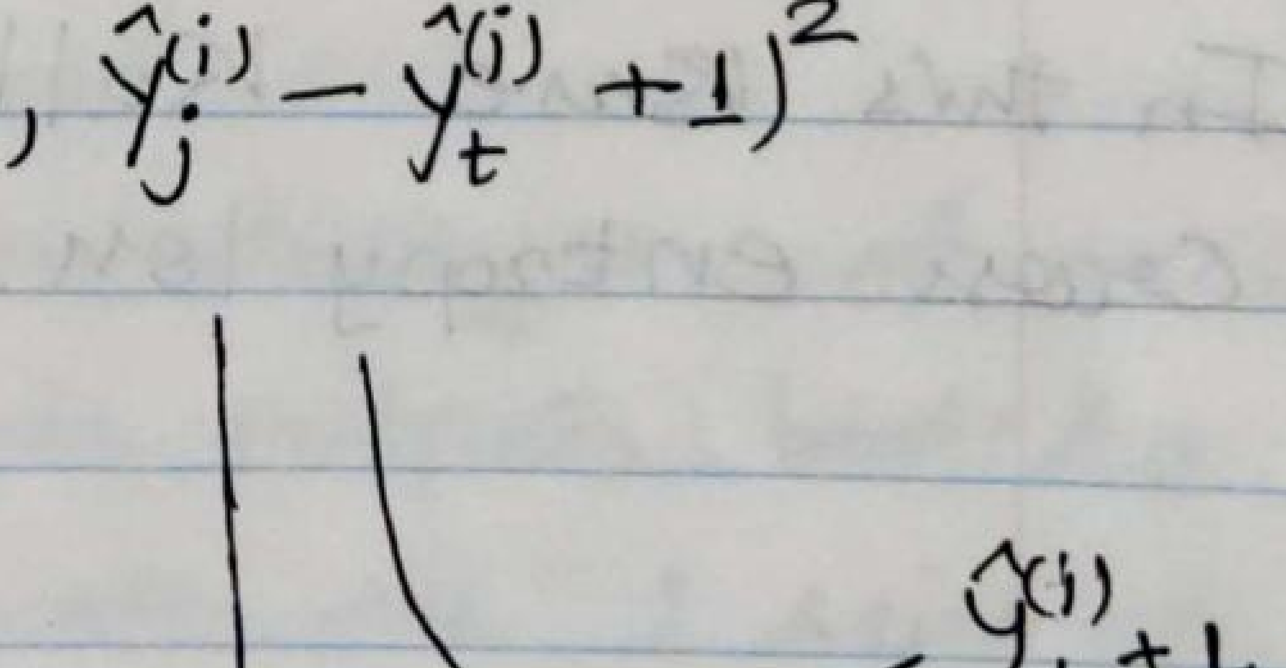
Squared hig hinge loss

↳ Just a square of the elements.

$$L_i(\theta) = \frac{1}{2} \sum_{j=1}^{K} \max\left(0,\ \hat{y}_j^{(i)} - \hat{y}_t^{(i)} + 1\right)^2$$



hing loss



Square hing loss.

the worst value or low random score before learning.

when $\boxed{\hat{y}_i \simeq 0}$ so all $L_i = \max(0, 0-0+1) \dots \max(0, 0-0+1)$

#K → no of classes. $= \underline{K-1}$ ← bad value.

Q6 ⇒ ⇒ $y_1 = 1$, $y_2 = 2$, $y_3 = 3$

$\hat{y}^1 = (0.5, 0.4, 0.3)$ $\hat{y}^{(2)} = (1.3, 0.8, -0.6)$

$\hat{y}^3 = (1.4, -0.4, 2.7)$.

	x_1	x_2	x_3
$\hat{y}_1$	0.5	0.4	0.3
$\hat{y}_2$	1.3	0.8	-0.6
$\hat{y}_3$	1.4	-0.4	2.7
y	1	2	3

we assuming that x_1 belong to class 1, x_2 belong to class 2, x_3 belong to class 3.

x_3 will be the best because it has large distance not only from decision boundary but also from other values.

$$L_i = \sum_{j \neq t} \max(0, \hat{y}_j - \hat{y}_t + 1)$$

$L_1 = \max(0, 0.5 - 1.4 + 1) + \max(0, 1.3 - 1.4 + 1) = 0.1 + 0.9 = 1$

$L_2 = \max(0, 0.4 - 0.8 + 1) + \max(0, -0.4 - 0.8 + 1) = \underline{0.6}$

$L_3 = \max(0, 0.3 - 2.7 + 1) + \max(0, -0.6 - 2.7 + 1)$

$= 0 + 0 = \underline{0}$

for x_3 we found best hinge loss value.

Q7 By adding regularization term in loss it become low weight means. or its called weight decay.

Smaller cofficients meas more stable solution that will generalize better our model.

$$L(\theta) = \frac{1}{m} \underbrace{\sum_{i=1}^{m} L_i(\overline{f(x_i, \omega)}^{\hat{y}}, y_i)}_{\text{data loss}} + \lambda \underline{R(\theta)}$$

λ: weight of regularization (hyper parameter)

$R(\theta)$: Regularization term.

L_1 Regularization : $R(\theta) = \sum_{ij} |\theta_{ij}|$ absolute.

L_2 Regularization : $R(\theta) = \sum_{ij} (\theta_{ij})^2$

* L_1 makes weights sparse (concentrate weights). more likely to get some units are 1 and some 0 basically its shutdown some cofficient.

$$eg \Rightarrow |0.5| + |0.5| = |1| + |0|$$

* L_2 ~~Can be drive it using~~ try to make them smaller and spreading them equally e.g. $0.5^2 + 0.5^2 < 1^2 + 0^2$. Can be drive it using MAP inference using a Gaussion prior for ω.

So will choose the L_2 because it not shurt the cofficients. it divide the weights equally over the model/network. better way to minimus the loss value.

Q8 ⇒ $L_1 : R(\theta) = \sum_{ij} |\theta_{ij}|$ Gradient descent $\frac{\partial L}{\partial \theta} = \frac{\partial R(\theta)}{\partial \theta}$

$$= \text{sign}(\theta)$$

during gradient descent subtract $\lambda(\text{sign}(\theta))$
Just add or subtruct "λ"

$L_2 : R(\theta) = \sum \theta_i^2 = \theta^T . \theta \Rightarrow \frac{\partial}{\partial \theta} R(\theta) = 2\theta$

during gradient descent subtract '$\lambda\theta$"

$$\frac{\partial L}{\partial \theta} = \frac{\partial}{\partial \theta} \frac{1}{m} \sum_{i=1}^{m} L_i(\underbrace{f(x^{(i)};\theta)}_{\hat{y}}, y^{(i)}) + \lambda \frac{\partial}{\partial \theta} R(\theta)$$

$\{L_2 = \lambda\theta\}$ $\{L_1 = \lambda \,\text{sign}(\theta)\}$

Q9 Kernel, bias and activity regularization ⇒
Kernel regularization : regularize ω → reduce ω
Bias regularization : regularize b → reduce b
activity regularization : regularize $\hat{y}$ → reduce $\hat{y}$ (including ω and b)

① * majorli we use kernel regularization because its se multiply the input. and not depends on input.
② activity ($\hat{y}$) is also effective but its depends on of input. (i.e. $\hat{y} = \omega \underline{x} + b$.) Implicitly we cover kernel (ω) and bias (b) with this.
③ Some time bias regularization if function expected to output small values.

Scanned by TapScanner