1 a

$$E[L(y,t)|y=\text{keep}] = 0.9 \times 0 + 0.1 \times 1 = 0.1$$

$$E[L(y,t)|y=\text{remove}] = 0.9 \times 100 + 0.1 \times 0 = 90$$

1 b

if $y = \text{keep}$, $E[L(y,t)] = Pr(t=\text{spam}|x) \times 1$

if $y = \text{remove}$, $E[L(y,t)] = [1 - Pr(t=\text{spam}|x)] \times 100$

So:

if $Pr(t=\text{spam}|x) \times 1 > [1 - Pr(t=\text{spam}|x)] \times 100 \Rightarrow$

$\Rightarrow Pr(t=\text{spam}|x) > \frac{100}{101}$ $\quad y_* = \text{remove}$

if $Pr(t=\text{spam}|x) \times 1 \leq [1 - Pr(t=\text{spam}|x)] \times 100 \Rightarrow$

$\Rightarrow Pr(t=\text{spam}|x) \leq \frac{100}{101}$ $\quad y_* = \text{keep}$

1c

follow the rule in 2b

$$Pr(t=s \mid X_1=0, X_2=0) = \frac{0.1 \times 0.4}{0.1 \times 0.4 + 0.9 \times 0.998} \leq \frac{100}{101} \quad y_* = \text{keep}$$

$$Pr(t=s \mid X_1=0, X_2=1) = \frac{0.1 \times 0.3}{0.1 \times 0.3 + 0.9 \times 0.001} \leq \frac{100}{101} \quad y_* = \text{keep}$$

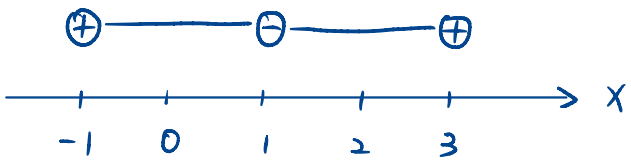
$$Pr(t=s \mid X_1=1, X_2=0) = \frac{0.1 \times 0.2}{0.1 \times 0.2 + 0.9 \times 0.001} \leq \frac{100}{101} \quad y_* = \text{keep}$$

$$Pr(t=s \mid X_1=1, X_2=1) = \frac{0.1 \times 0.1}{0.1 \times 0.1 + 0.9 \times 0} > \frac{100}{101} \quad y_* = \text{remove}$$

1d

$$E[L(y_*, t)] = 1 \times 0.9901 + 100 \times (1 - 0.9901) = 1.9801$$

2a



if it's linear separable than we can find a decision bounday line,

with $X=-1$ $X=3$ in one side $X=1$ in other side,

and not intersect with the line from $X=-1$ to $X=3$

Because $X=-1$ on the line from $X=-1$ to $X=3$

So, if decision boundary line not intersect with line from $X=-1$ to $X=3$ then $X=-1$, $X=1$, $X=3$ all in same side

if decision boundary line intersect with line from $X=-1$ to $X=3$, then $X=-1$ and $X=3$ in different side

Contradiction

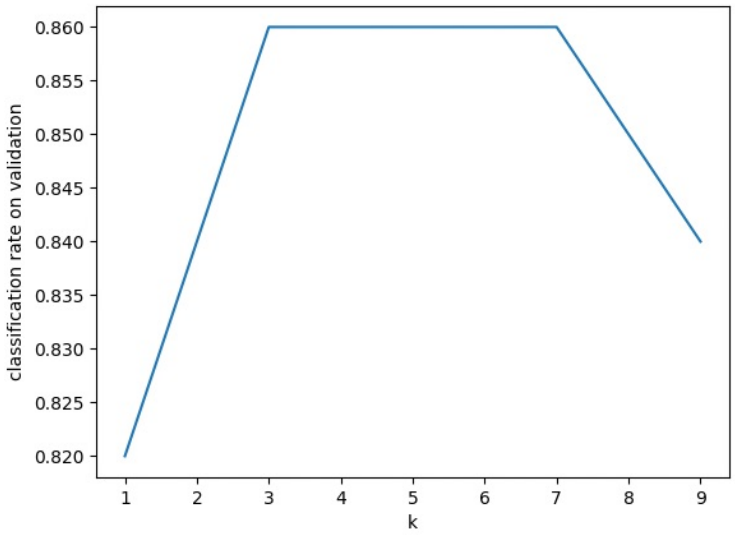
So it's not linear separable

2b

X		t
-1	1	1
1	1	0
3	9	1

$$\begin{cases} -W_1 + W_2 > 0 \\ W_1 + W_2 < 0 \\ 3W_1 + 9W_2 > 0 \end{cases} \Rightarrow \begin{cases} W_1 = -2 \\ W_2 = 1 \end{cases}$$

3.1a



3.1b

k = 1 over fitting training data

k = 9 underfitting training data

k = 3, 5, 7 has best performance

$k^* = 3$, perform good, not too complex

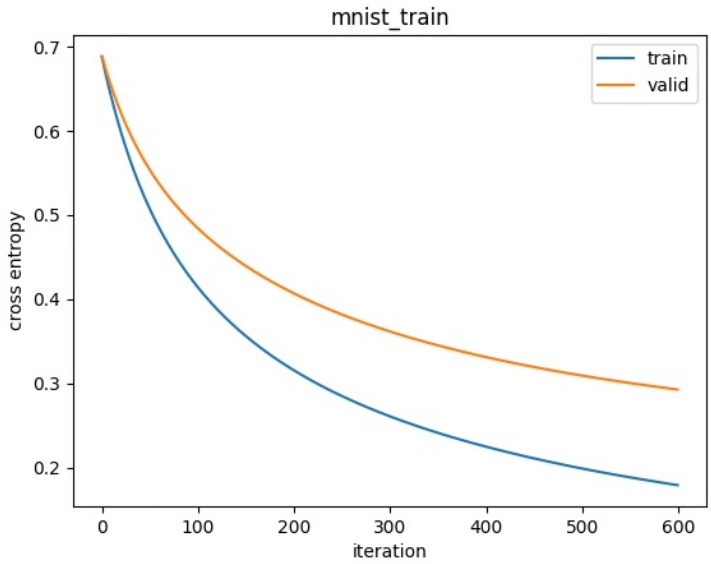
classification rate of k^*-2, k^*, k^*+2 are

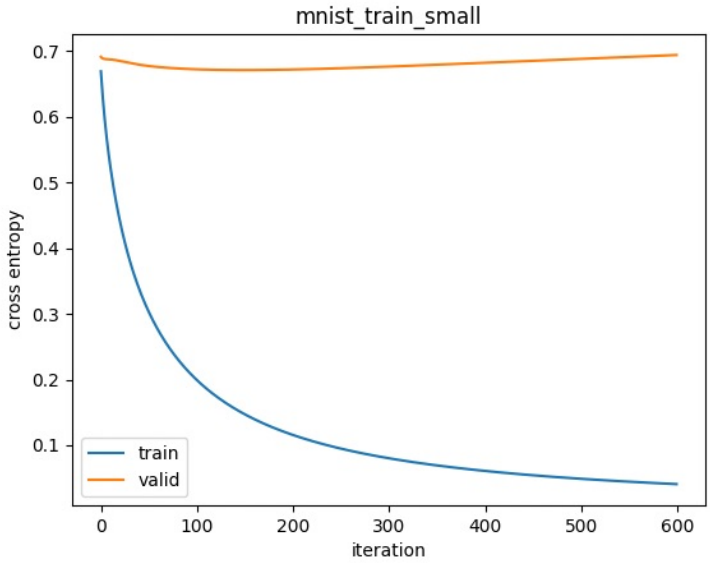
0.82 0.86 0.86

3.2b

```
best hyperparameters on mnist_train:

learning rate:  0.01 num_iterations:  600  weights: all zeros

train final cross entropy:  0.1791036205340543  train classification error:  0.012499999999999956

valid final cross entropy:  0.29271651738589105 valid classification error:  0.09999999999999998

test final cross entropy:  0.2643556917798134   test classification error:  0.07999999999999996


best hyperparameters on mnist_train_samll:

learning rate:  0.01 num_iterations:  600  weights: all zeros

train_fce:  0.04092173692963222 train_class_error:  0.0

valid_fce:  0.6937200510582616 valid_class_error:  0.3400000000000001

test_fce:  0.5800816326928813 test_class_error:  0.21999999999999997
```

3.2c





If training data can well represent validation data and hyperparameter fits, then cross entropy reduce as training process for both train and validation data. Just like mnist_train

In minist_train_small, train size too small, cannot represent validation well, so weight adjustment base on train data can't help reduce cross entropy loss of validation data

4a

function is convex, to minimiz it, we need to find critical point, where derivatives WRT w is 0

$$\frac{d}{dw}\left(\frac{1}{2}\sum_{i=1}^{N} a^{(i)}(y^{(i)}-w^T x^{(i)})^2 + \frac{\lambda}{2}\|w\|^2\right) = 0$$

$$= \frac{d}{dw}\left(\frac{1}{2}\sum_{i=1}^{N} a^{(i)}(y^{(i)}-w^T x^{(i)})^2\right) + \frac{d}{dw}\frac{\lambda}{2}\|w\|^2$$

$$= -\sum_{i=1}^{N} a^{(i)} x^{(i)}(y^{(i)}-w^T x^{(i)}) + \lambda w$$

$$= -\sum_{i=1}^{N} a^{(i)} x^{(i)} y^{(i)} + \sum_{i=1}^{N} a^{(i)} x^{(i)} w^T x^{(i)} + \lambda w$$

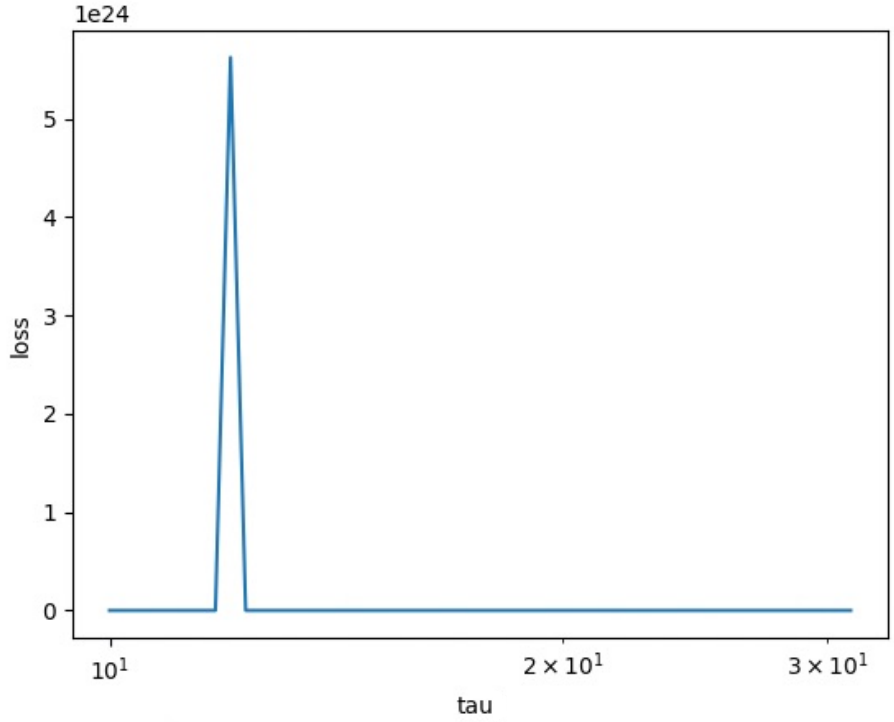
$$= -\sum_{i=1}^{N} a^{(i)} x^{(i)} y^{(i)} + \sum_{i=1}^{N} x^{(i)} a^{(i)} x^{(i)T} w + \lambda w$$

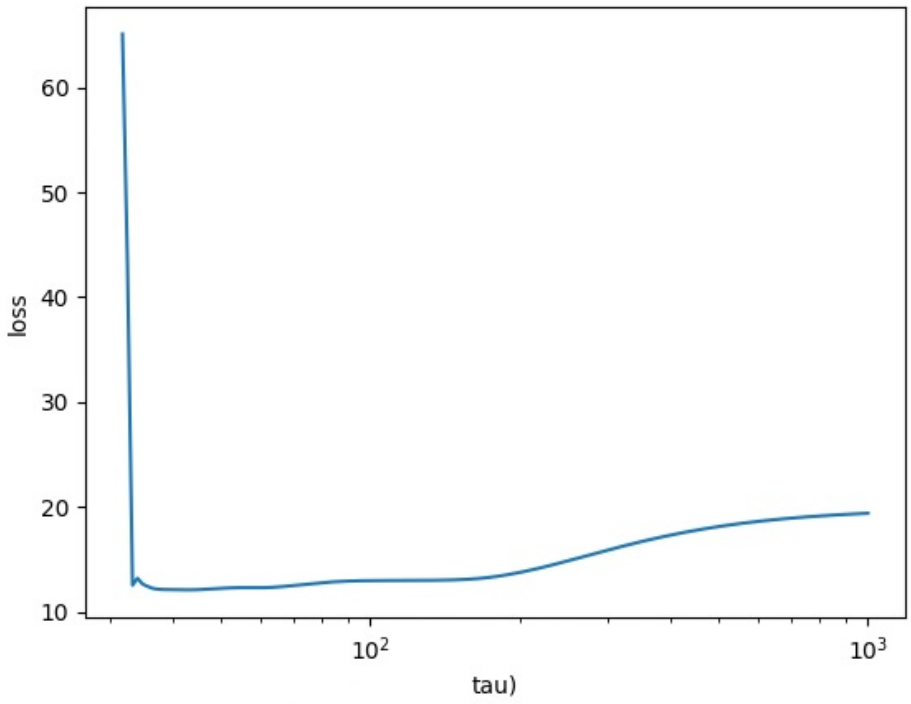
$$= -\sum_{i=1}^{N} a^{(i)} x^{(i)} y^{(i)} + \left(\sum_{i=1}^{N} x^{(i)} a^{(i)} x^{(i)T} + \lambda\right) w$$

$$= -X^T A y + (X^T A X + \lambda I) w = 0$$

So: $w = (X^T A X + \lambda I)^{-1} X^T A y$

4 c





Since Loss behave weired when Tau is small so I split data to 2 parts

4 d

tau measure how far away points from test point do we care

if tau is large, we care about far point

- when tau $\to \infty \Rightarrow a_i$ is similar for all training points, so no distance panelty, just like normal linear regression

if tau is small, we care only near points

when tau $\to 0 \Rightarrow$ we only care about nearest point

It's what actually happened