CS-229
PSET-2

Vasu G. Patel
06439203
Vgpatel1

Q-1)(a) Dataset A converges at (30, 372)

Dataset B does not converges.

Q-1)(b). Gradients of dataset B is very large and hence need smaller value of learning rate in order to make sure the change in weight remain within the dataset

Graph of distribution of dataset A & B are plotted herewith in next page

Q-1(c) (i) using a different learning rate i.e. smaller value of learning rate for B will be useful for dataset B to converge

(ii) Yes, decreasing the learning rate over time will be helpful for dataset B to converge

(iii) Linear scaling of dataset B would not help in it to converge.

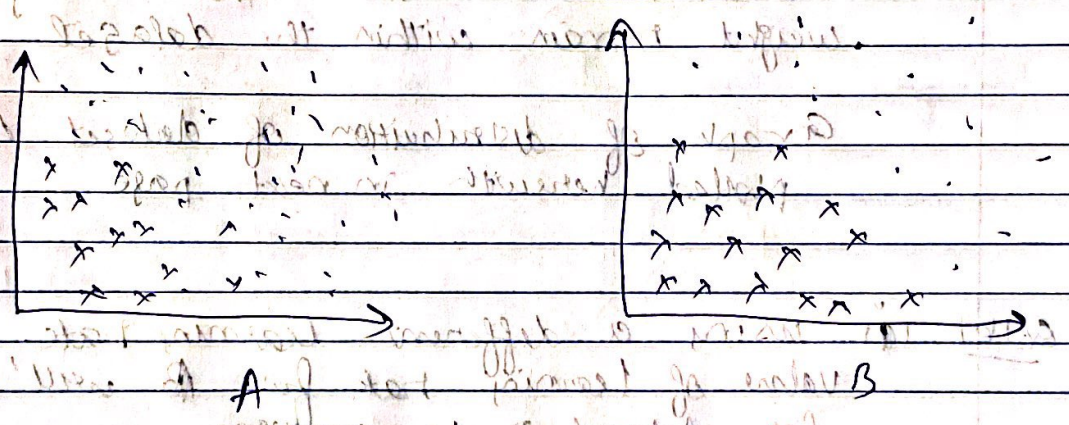
(iv). Yes, adding regularized $\|\theta\|_2^2$ to loss function will help penalize the gradient and thus will help dataset B to converge

(v) Adding zero mean gaussian noise to the training set would not help data labels.

Q-1(d) Yes i.e. SVMs are vulnerable to dataset like B.

Because if the classes are located reasonably far away then the SVM would not be effective in optimizing the max dist between decision boundary and classes.

Q-1(b)



Q-2 (a) Size of dictionary : 1758

(b) Naive Bayes has an accuracy of 97.84%.

(c) Words Top 5 indicative words for Naive Bayes are:

'claim', 'won', 'prize', 'tone', 'urgent!'

d). Best Kernel radius is 0.1

Q-3 (a) Using Bayes rule to compute posterior distribution of θ:

$$p(\theta|y,x) = \frac{p(\theta,x,y)}{p(y,x)}$$

$$= \frac{p(y|x;\theta)\cdot p(\theta,x)}{p(y|x)p(x)}$$

$$= \frac{p(y|x,\theta)\,p(\theta|x)\,p(x)}{p(y|x)\,p(x)}$$

$$= \frac{p(y|x,\theta)\,p(\theta|x)}{p(y|x)}$$

The mode of the posterior distribution is given by

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\; p(\theta|x) = \underset{\theta}{\operatorname{argmax}}\; \frac{p(y|x,\theta)\,p(\theta|x)}{p(y|x)}$$

Noting that $p(\theta|x) = p(\theta)$ and that the denominator is independent of θ gives:

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\; p(y|x,\theta)\,p(\theta)$$

Q-3 (b) Using result from 3(a)

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}}\ p(y|x,\theta)\,p(\theta)$$

$$= \underset{\theta}{\operatorname{argmax}}\ \log\left(p(y|x,\theta)\,p(\theta)\right)$$

$$= \underset{\theta}{\operatorname{argmin}}\ -\log\left(p(y|x,\theta)\,p(\theta)\right)$$

$$= \underset{\theta}{\operatorname{argmin}}\left(-\log p(y|x,\theta) - \log p(\theta)\right)$$

Since $\theta \sim N(0, \eta^2 I)$, the prior distribution is given by

$$p(\theta) = \frac{1}{(2\pi|\Sigma|)} \exp\left(-\frac{1}{2}\theta^T \Sigma^{-1} \theta\right)$$

where $\Sigma = \eta^2 I \Rightarrow |\Sigma| = \eta$ and $\Sigma^{-1} = \frac{1}{\eta} I$

$$\Rightarrow -\log p(\theta) = \log\left(\sqrt{2\pi}\,\eta\right) + \frac{1}{2\eta^2}\theta^T I \theta = \log\left(\sqrt{2\pi}\,\eta\right) + \frac{1}{2\eta^2}\|\theta\|_2^2$$

$$\Rightarrow \theta_{MAP} = \underset{\theta}{\operatorname{argmin}}\left(-\log p(y|x,\theta) + \frac{1}{2\eta^2}\|\theta\|_2^2\right)$$

Comparing the above result with MLE estimation with L2 regularization gives

$$\lambda = \frac{1}{2\eta^2}$$

Q-3(c) Given that the random noise is independent of every training example (x^i, y^i) and is distributed according to a Gaussian distribution i.e. $\epsilon^i \sim N(0, \sigma^2)$;

$$p(y^i|x^i;\theta) = p(\epsilon^i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-\epsilon^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^i - \theta^T x)^2}{2\sigma^2}\right)$$

The loss function for one training example for MAP estimation will then be given as:

$$L^i_{MAP} = -\log p(y^i|x^i;\theta) + \frac{1}{2\eta^2}\|\theta\|_2^2 = \log(\sqrt{2\pi}\sigma) + \frac{(y^i - \theta^T x)^2}{2\sigma^2} + \frac{1}{2\eta^2}\|\theta\|_2^2$$

Given the design input matrix x and output column vector $\vec{y}$

$$J_{MAP} = \log(\sqrt{2\pi}\sigma) + \frac{1}{2\sigma^2}(\vec{y} - x\theta)^T(\vec{y} - x\theta) + \frac{1}{2\eta^2}\theta^T\theta.$$

To get $\theta_{MAP} = \underset{\theta}{\operatorname{argmin}}\, J_{MAP}$, $\nabla_\theta J_{MAP} = 0$

$$\Rightarrow \nabla_\theta\left(\frac{1}{2\sigma^2}(\vec{y} - x\theta)^T(\vec{y} - x\theta) + \frac{1}{2\eta^2}\theta^T\theta\right) = 0$$

$$\Rightarrow \frac{-1}{\sigma^2}(x^T x\theta - x^t\vec{y}) + \frac{1}{\eta^2}\theta = 0$$

$$\Rightarrow \left(\frac{1}{\sigma^2}x^T x - \frac{1}{\eta^2}I_k\right)\theta = \frac{1}{\sigma^2}x^T\vec{y}$$

$$\Rightarrow \theta = \left(\frac{1}{\sigma^2}x^T x - \frac{1}{\eta^2}I_k\right)^{-1} \cdot \left(\frac{1}{\sigma^2}x^T\vec{y}\right)$$

where I_k is an Identity matrix of dimension $(k \times k)$ and k is no. of features

Q-3(d) Using the result from previous parts:

$$J_{MAP} = (-\log p(y|x,\theta) - \log p(\theta))$$

Since θ_i is independently distributed as $L(0,b)$:

$$p(\theta_i) = \frac{1}{2b} \exp\left(-\frac{|\theta|}{b}\right)$$

$$\Rightarrow -\log p(\theta_i) = \log(2b) + \frac{|\theta|}{b}$$

$$\Rightarrow J_{MAP} = \log(\sqrt{2\pi}\sigma) + \frac{1}{2\sigma^2}(\vec{y} - X\theta)^T(\vec{y} - X\theta) + \log(2b) + \frac{|\theta|}{b}$$

Comparing this with the L1 regularization of MLE:

$$J(\theta) = \|X\theta - \vec{y}\|_2^2 + \gamma\|\theta\|_1 \text{ gives } \gamma = \frac{1}{b}.$$

Q-4(c) For this section first iteration, taking the first example $\theta^0 = 0$ result in:

$$\theta^1 = \theta^0 + \alpha(y^1 - g(\theta^{0T}\phi^1))\phi^1 = \theta^1 = \alpha(y^1 - g(\theta^{0T}\phi^1))\phi^1$$

where ϕ^i is the high dimensional feature mapping of x^i.

Next,

$$\theta^2 = \theta^1 + \alpha(y^2 - g(\theta^{1T}\phi^2))\phi^2 = \alpha y^1 - g(\theta^{0T}\phi^1)\phi^1 + \alpha(y^2 - g(\theta^{1T}\phi^2))\phi^2 + \alpha(y^3 - g(\theta^{2T}\phi^3))\phi^3$$

$$\theta^2 = \theta^1 + \alpha(y^2 - g(\theta^{1T}\phi^2))\phi^2 = \alpha(y^1 - g(\theta^{0T}\phi^1))\phi^1 + \alpha(y^2 - g(\theta^{1T}\phi^2))\phi^2$$

$$\theta^3 = \theta^2 + \alpha(y^3 - g(\theta^{2T}\phi^3))\phi^3$$

$$= \alpha(y^1 - g(\theta^{0T}\phi^1))\phi^1 + \alpha(y^2 - g(\theta^{1T}\phi^2))\phi^2 + \alpha(y^3 - g(\theta^{2T}\phi^3))\phi^3$$

Let $\beta_j = \alpha(y^{j-1} - g(\theta^{j-1T}\phi^j))$ be a scalar value that represent the coefficient of ϕ^j, then θ can be written as:

$$\theta^i = \sum_{j=1}^{i} \beta_j \phi^j$$

Q. 4(a) Contd

where β_i and ϕ have the same number of rows which is the total no. of training sample used.

The prediction on the new training sample x^{i+1} is given as:

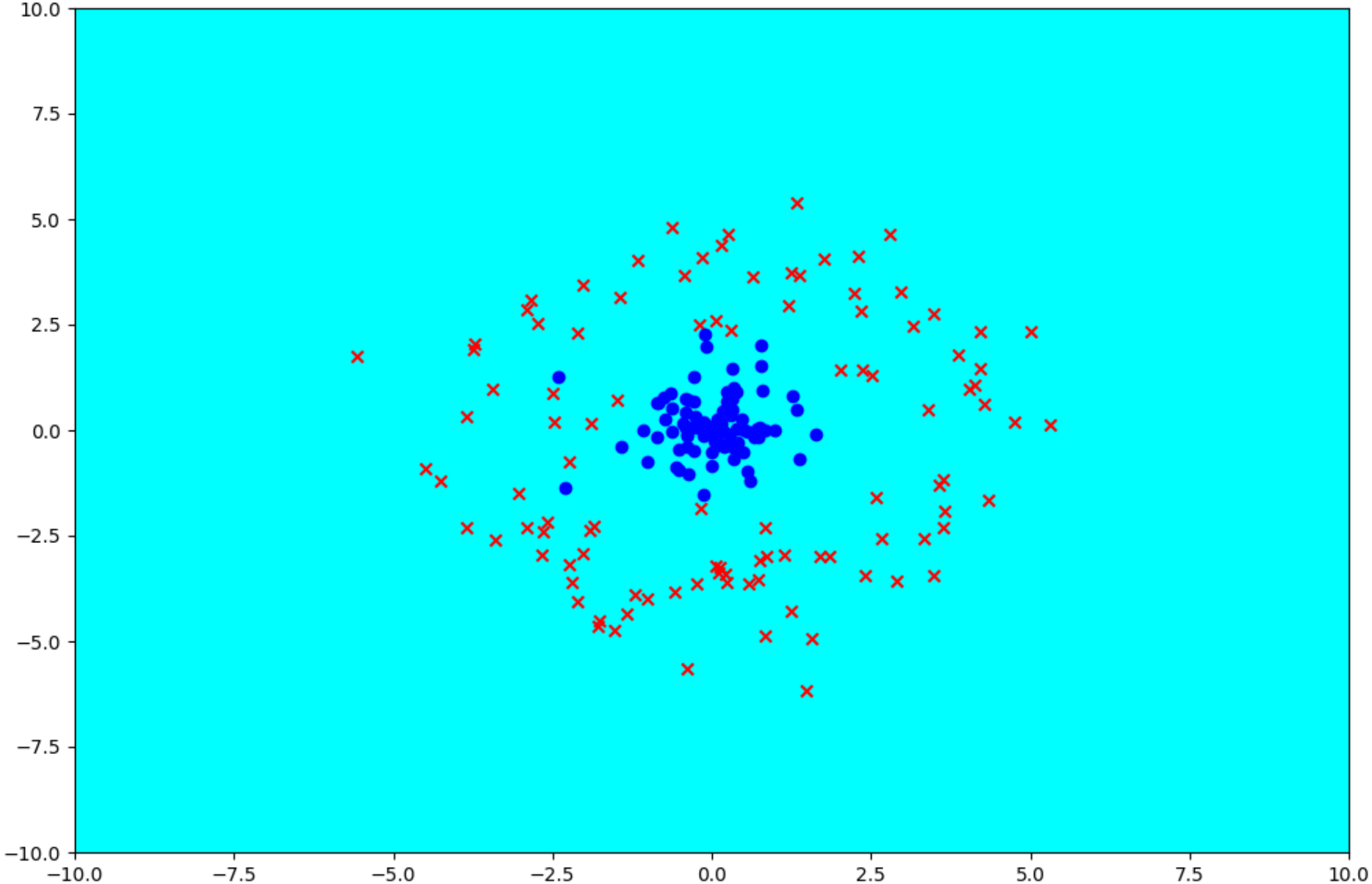
$$h_{\theta i}(x^{i+1}) = g(\theta^{iT} \phi^{(i+1)}) = g\left(\sum_{j=1}^{i} \beta_j \phi^j \phi^{i+1}\right) = g\left(\sum_{j=1}^{i} \beta_j <\phi^j, \phi^{i+1}>\right)$$

$$h_{\theta i}(x^{i+1}) = g\left(\sum_{j=1}^{i} \beta_j K(\phi^j, \phi^{i+1})\right)$$

where $K(x, z)$ is the kernel function.

The model is updated by adding a new element in β given as:

$$\beta_{i+1} = \alpha(y^{i+1} - h_\theta(x^{i+1})).$$


10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Q-5 (a) $CE(y^i, \hat{y}^i) = -\sum_{k=1}^{K} y_k^i \log\left(\frac{e^{z_k^i}}{\sum_{k=1}^{K} e^{z_k^i}}\right) = -\sum_{k=1}^{K} y_k^i \left(z_k^i - \log \sum_{k=1}^{K} e^{z_k^i}\right)$

$$\frac{\partial CE(y^i, \hat{y}^i)}{\partial z_j^i} = -\sum_{k=1}^{K} y_k^i \left(\frac{\partial z_k^i}{\partial z_j^i} - \frac{\partial}{\partial z_j^i} \log\left(\sum_{k=1}^{K} e^{z_k^i}\right)\right)$$

$$= -\sum_{k=1}^{K} y_k^i \left(\frac{\partial z_k^i}{\partial z_j^i} - \frac{1}{\sum_{k=1}^{K} e^{z_k^i}} \cdot \frac{\partial}{\partial z_j^i} \sum_{k=1}^{K} e^{z_k^i}\right)$$

$$= -y_j^i + \sum_{k=1}^{K} y_k^i \frac{e^{z_j^i}}{\sum_{k=1}^{K} e^{z_k^i}} = -y_j^i + \frac{e^{z_j^i}}{\sum_{k=1}^{K} e^{z_k^i}} \sum_{k=1}^{K} y_k^i$$

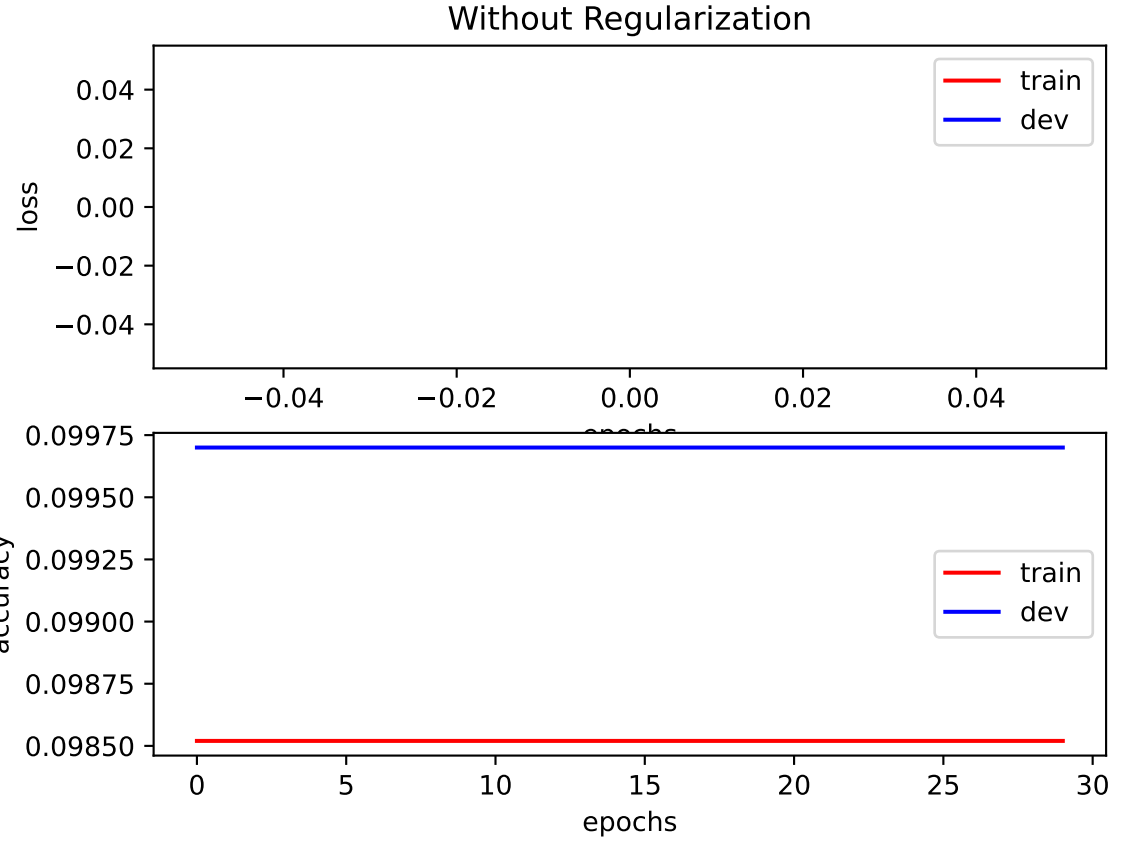
$$= -y_j^i + \frac{e^{z_j^i}}{\sum_{k=1}^{K} e^{z_k^i}} (1) = \hat{y}_j^i - y_j^i$$

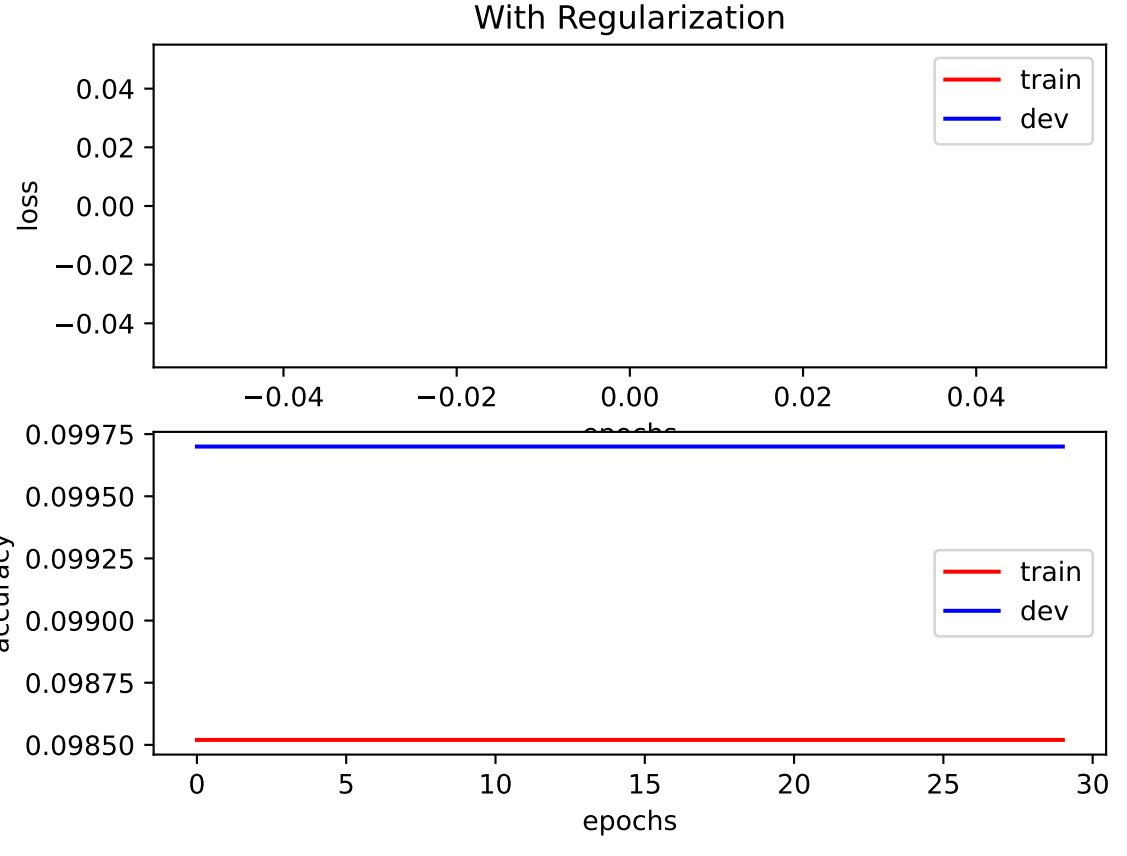
$$\Rightarrow \nabla_z CE(y^i, \hat{y}^i) = \hat{y}^i - y^i$$

(b) Model baseline accuracy = 9.8%.
Model regularized accuracy = 9.8%.

Plot attached - baseline

(c) Plot attached - regularized.


Without Regularization
train
dev
loss
0.04
0.02
0.00
−0.02
−0.04
−0.04
−0.02
0.00
0.02
0.04
epochs
0.09975
0.09950
0.09925
0.09900
0.09875
0.09850
accuracy
train
dev
0
5
10
15
20
25
30
epochs


With Regularization
train
dev
loss
0.04
0.02
0.00
−0.02
−0.04
−0.04
−0.02
0.00
0.02
0.04
epochs
0.09975
0.09950
0.09925
0.09900
0.09875
0.09850
accuracy
train
dev
0
5
10
15
20
25
30
epochs