Theory Questions

(1) As suggested we will choose $m = d = \lfloor \frac{1}{\gamma^2} \rfloor$ and $\{x_i\}_i$ to be the standard basis of $\mathbb{R}^d$, namely $\{\hat{e}_i\}_i$ (e.g. $e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$) Then:

(a) by definition $|x_i| = |\hat{e}| = 1 \quad \forall i$

(b) we would like to find w^* s.t $y_i \frac{x_i \cdot w^*}{|w^*|} \geq \gamma$ and $|w^*| \leq 1$

$* \quad 1 \geq |w^*|^2 = d w_i^{*2} \Rightarrow w_i^{*2} \leq \frac{1}{d} = \frac{1}{\lfloor \frac{1}{\gamma^2} \rfloor}$

assume for simplicity $w_i = w_j \ \forall i,j$ see next page

$**$ also we know from (c) $(x_i, y_i) = (\hat{e}_i, -1)$

$\Rightarrow y_i \frac{x_i \cdot w^*}{|w^*|} = -\frac{w_i^*}{|w^*|} \geq \gamma \Rightarrow w_i^* \leq -\frac{\gamma}{|w^*|}$

$|w^*| \geq 0, \ 0 < \gamma < 1 \Rightarrow w_i^* \leq 0$

$* + ** \Rightarrow w_i^* \leq -\frac{1}{\sqrt{\lfloor \frac{1}{\gamma^2} \rfloor}}$

we choose

$\Rightarrow w^* = \frac{1}{\sqrt{\lfloor \frac{1}{\gamma^2} \rfloor}} (-1, -1, \ldots, -1)$

(c) we have to choose $\{y_i\}_i$ so that the perceptron will make $\frac{1}{[r^2]}$ mistakes

$\frac{1}{[r^2]} = m = d \Rightarrow$ mistakes for all X_i !

we start with $w_1 = (0, \dots 0)$.

$\text{sign}(w_1 x_1) = 0 \Rightarrow \hat{y}_1 = 1$

So we can choose $y_1 = -1$ for the perceptron to make a mistake.

$\Rightarrow$ non-zero update:

$w_2 = w_1 + y_1 x_1 = (-1, 0 \dots)$.

we can repeat that logic and then we expect:

$w_t = (-1, -1, \dots, -1, 0 \dots 0)$

t index

We can prove it by induction. we already have shown for $t = 2$. now assuming our logic. we have w_t as described.

(and $y_t = -1$)

$\text{sign}(w_t \cdot x_t) = \text{sign}[(-1, -1 \dots -1, 0 \dots) \cdot (0, \dots 0, 1, 0 \dots 0)] = 0$

t entry $\quad$ t entry

$\Rightarrow \hat{y}_t = 1$, again we choose $y_t = -1$ for making another mistake and then

$w_{t+1} = w_t + y_i x_i = (-1 \dots -1, 0 \dots) + (-1)(0 \dots 0, 1, 0 \dots)$

$= (-1, \dots -1, 0 \dots)$

t $\quad$ $t+1$

$\Rightarrow$ our point are $\{(\hat{e}_i, -1)\}_{i=1}^{m}$ and we have $m = \frac{1}{[r^2]}$ mistakes

(2) $d \geq 6 \quad \mathcal{X} \in \mathcal{X} = \{1, 2 \dots d\} \quad H = \{h_{i,j} \mid 1 \leq i < j \leq d\}$

$$h_{i,j}(x) = \begin{cases} 1 & (x = i) \cup (x = j) \\ 0 & \text{otherwise} \end{cases}$$

define $V_1 = |h \in H_1 : h(x) = 1|$

$V_0 = |h \in H_1 : h(x) = 0|$

we start with $|H_1| = \binom{d}{2} = \frac{d(d-1)}{2}$

assuming we got a point x_1.

Then $|V_1| = |\{h_{x_1, x>x_1}\}_x \cup \{h_{x<x_1, x_1}\}_x| = d-1$

and $|V_0| = |H_1| - |V_1| = d\frac{(d-1)}{2} - (d-1)$

$$= \left(\frac{d}{2} - 1\right)(d-1) \underset{d \geq 6}{\geq} 2(d-1) > |V_1|$$

$\Rightarrow$ first choise of label is $\hat{y}_1 = 0$

assuming we made a mistake and $y_0 = 1$

$\Rightarrow |H_2| = |H_1| - |V_0| = |V_1| = d-1$

$H_2 = \{\{h_{x_1, x>x_1}\}_x \cup \{h_{x<x_1, x_1}\}_x\}$

now assuming we have a new point $x_2 \neq x_1$

there is only one hypothesis for which $\hat{y}_2 = 1$

$\Rightarrow |V_1| = 1 \quad , |V_0| = d-1 - |V_1| = d-2$

$\underset{d \geq 6}{\Rightarrow} |V_0| > |V_1| \Rightarrow \hat{y}_2 = 0$.

but if we made a mistake and $y_2 = 1$

$\Rightarrow |H_3| = |H_2| - |V_0| = 1 \Rightarrow$ we are done!

$\Rightarrow$ we should that at most, we will have 2 mistakes until we are left with only one hypothesis. if at any stag we wouldn't make a mistake we will be left with even smaller hypothesis class size. So the case of 2 mistakes in a raw is the most stricted.
Also, we can think of an example for such a sequence. (its labels shell be $=1$) so $(3,1),(4,1)$ will do.
after the first we are left with h_{13}, h_{23} and $\{h_{3x}\}_{x>3}$ and for the second we only tilter h_{34}.

$\Rightarrow$ $\sup_S M_A(S) = 2$ //

(3) $f_1 \dots f_m$ are convex $f_i : \mathbb{R}^d \to \mathbb{R}$

$g(x) = \max_i f_i(x)$

(a) define $j = \arg\max_{i \in [1,m]} f_i(\alpha x_1 + (1-\alpha)x_2)$ $\quad 0 < \alpha < 1$

$\Rightarrow$

$$g(\alpha x_1 + (1-\alpha)x_2) = f_j(\alpha x_1 + (1-\alpha)x_2)$$

$$\leq \alpha f_j(x_1) + (1-\alpha) f_j(x_2) \leq \alpha g(x_1) + (1-\alpha) g(x_2)$$

f_j is convex

$\downarrow$

$f_i(x) \leq g(x) \quad \forall x$

$\Rightarrow$ $g(x)$ is convex. //

(b) sub gradient set $\{v\}$ at point $\tilde{x}$

$\forall x \quad f(x) \geq f(\tilde{x}) + v \cdot (x - \tilde{x})$

$$g(x) \geq f_i(x) \geq f_i(\tilde{x}) + \nabla f_i(\tilde{x})(x - \tilde{x}) =$$

$\downarrow$

f_i is convex

$$= g(\tilde{x}) + \nabla f_i(\tilde{x})(x - \tilde{x})$$

$i = \arg\max_{i \in [1,m]} (\{f_i(\tilde{x})\}_{i=1}^m)$

(4) $\min_{w,b,\xi} \frac{1}{2} w^T w + \frac{C}{2} \sum_{i=1}^{m} \xi_i^2$

s.t. $y_i(w^T \cdot x_i + b) \geq 1 - \xi_i \quad \forall i = 1 \dots m$

(a) we have to show that some other problem, similar to the original but with the additional set of constraints $\xi_i \geq 0$ is equivalent.

Assum (in contradiction) that there is an optimal solution to the original problem in which there exist some k for which $\xi_k < 0$.

we can think of some other set ξ_i' such that $\xi_i' = \xi_i \ \forall i \neq k$ and $\xi_k' = 0$.

this set of variables also satisfies the constraint:

$y_j(w^T \cdot x_j + b) \geq 1 - \xi_j$ —— $= 1 - \xi_j' \quad j \neq k$

\ $\geq 1 = 1 - \xi_j' \quad j = k$

But it has a smaller optimum because the the element $\frac{C}{2} \xi_k'^2$ now vanish and for the original set $\frac{C}{2} \xi_k^2 > 0$

$\Longrightarrow$ contradiction to the assumption that $\{\xi_i\}$ are of the optimal solution

$\Longrightarrow$ $\xi_i \geq 0 \ \forall i$ must hold anyway!

(b) The Lagrangian:

$$\mathcal{L}(x_i, y_i, w, b, \xi_i, \lambda_i) =$$

$$= \frac{1}{2} w^T w + \frac{c}{2} \sum_i \xi_i^2 + \sum_i \lambda_i (1 - \xi_i - y_i (w^T \cdot x_i + b))$$

(c)

$$\nabla_w \mathcal{L} = w - \sum_i \lambda_i y_i x_i \stackrel{!}{=} 0$$

$$\Rightarrow w = \sum_i \lambda_i y_i x_i$$

$$\nabla_{\xi_i} \mathcal{L} = c\xi_i - \lambda_i = 0$$

$$\Rightarrow \xi_i = \lambda_i / c$$

$$\nabla_b \mathcal{L} = -\sum_i \lambda_i y_i \stackrel{!}{=} 0$$

$$\Rightarrow \sum_i \lambda_i y_i = 0$$

minimizing $\mathcal{L}$ w.r.t the parameters w, b, ξ_i

for optimal value ↖

$$\Rightarrow \mathcal{L} = \frac{1}{2} \overbrace{\left(\sum_i \lambda_i y_i x_i\right)^T}^{w^T} \overbrace{\left(\sum_j \lambda_j y_j x_j\right)}^{w} + \frac{c}{2} \sum_i \overbrace{\frac{\lambda_i^2}{c^2}}^{\xi_i^2}$$

$$- b \underbrace{\sum_i \lambda_i y_i}_{=0} + \sum_i \lambda_i - \underbrace{\sum_i \frac{\lambda_i^2}{c}}_{\lambda_i \cdot \xi_i} - \sum_i \lambda_i y_i x_i \underbrace{\sum_j \lambda_j y_j x_j}_{w^T}$$

$$= -\frac{1}{2} \sum_i \lambda_i y_i x_i \sum_j \lambda_j y_j x_j - \frac{1}{2c} \sum_i \lambda_i^2 + \sum_i \lambda_i$$

(d) Our primal problem is

$g(\lambda) = \min_{w,b,\xi} \mathcal{L}(w,b,\xi,\lambda)$ (as seen in class)

where $g(\lambda)$ is the dual function.

The dual problem then is

$\max_{\lambda \geq 0} g(\lambda) = \min_{\lambda \geq 0} -g(\lambda)$ *

(This is because the problem is

$\min_{w,b,\xi} \max_{\lambda} \mathcal{L}(w,b,\xi,\lambda)$)

* with the constraint from minimization of $\mathcal{L}$

$\sum \lambda_i y_i = 0$ ($\lambda \geq 0$ to satisfy $\xi_i \geq 0$ by $\lambda_i = C\xi_i$)

$\Longrightarrow$

$$\min_{\lambda \geq 0} \frac{1}{2} \sum_{ij} \lambda_i y_i x_i \lambda_j y_j x_j + \frac{1}{2C} \sum_i \lambda_i^2 - \sum_i \lambda_i$$

$$\text{S.T. } \sum \lambda_i y_i = 0$$

(of cours $\mathcal{L}$, and g also depend on x_i, y_i. The inputs of the original function)

(5) $H = \{h_{a,b} : a < b\}$ $\mathcal{X} = \mathbb{R}$

$$h(x) = \begin{cases} 1 & x \in [a,b] \\ 0 & \text{else} \end{cases}$$

$\Pi_H(m) = \max_{C:|C|=m} |H_C|$

for convenience, we can of a sorted sample of m points. $(x_1 < x_2 \dots < x_m)$

now, we can think of a possible labeling

* $(0, 0 \dots 0) \rightarrow x_i \notin [a,b] \ \forall i.$

** other possibilities:

we can start from $x_1 \in [a,b]$, There are m possibilities to label like that. $[a,b]$ can include points from x_1 up to some x_k $(k = [1,m])$. for $x_2 \in [a,b]$ and $x_1 \notin [a,b]$ we have $m-1$ possibilities. points up to x_k $k = [2,m]$. can be included. and so on. This is an algebric sereis. $\Longrightarrow$

$$\Pi_H(m) = \underset{*}{1} + \sum_{k=1}^{m} (m+1-k) = \underset{**}{1 + \frac{m}{2}(m+1)}$$

(6) we start with

$$n \geq \frac{256d}{\varepsilon^2}\ln\left(\frac{128d}{\varepsilon^2}\right) + \frac{256}{\varepsilon^2}\ln\left(\frac{1}{\delta}\right)$$

using $x \geq 4a\ln(2a) + 2b$

$$\Rightarrow x > a\ln(x) + b$$

we get $\left(a \equiv \frac{64}{\varepsilon^2}d \quad b \equiv \frac{128}{\varepsilon^2}\ln\left(\frac{1}{\delta}\right)\right)$

$$n \geq \frac{64d}{\varepsilon^2}\ln(n) + \frac{128}{\varepsilon^2}\ln\left(\frac{1}{\delta}\right)$$

$$(\delta<1) \quad \geq \frac{64d}{\varepsilon^2}\ln(n) + \frac{32}{\varepsilon^2}\ln\frac{1}{\delta} \geq$$

$$\left(\begin{matrix} d\geq 1 \\ \varepsilon<1 \end{matrix}\right) \geq \frac{64d}{\varepsilon^2}\ln(n) + \frac{32}{\varepsilon^2}\ln\left(\frac{1}{\delta}\right)$$

$$* = \frac{32d}{\varepsilon^2}\ln(n) + \frac{32}{\varepsilon^2}\ln\left(\frac{1}{\delta}\right) + \frac{32d}{\varepsilon^2}\ln(n)$$

now, we can find a numeric bound on n

$$n \geq \frac{256}{\varepsilon^2}\ln\left(\frac{128d}{\varepsilon^2}\right) + \frac{256}{\varepsilon^2}\ln\left(\frac{1}{\delta}\right) \geq \frac{256}{\varepsilon^2}\ln(128d)$$

$$\left(\begin{matrix} \varepsilon<1 \\ d\geq 1 \end{matrix}\right) \geq 256\ln(128) \geq \frac{20.4}{d}$$

$$* \Rightarrow n \geq \frac{32d}{\varepsilon^2}\ln(n) + \frac{32}{\varepsilon^2}\ln\frac{1}{\delta}) + \frac{32d}{\varepsilon^2}\left(\ln 2e + \ln(n) - \ln(d)\right)$$

$$\geq \frac{32}{\varepsilon^2}\left(d\ln(n) + \ln\left(\frac{1}{\delta}\right) + d\ln(2e) + \ln n - d\ln d\right)$$

$$\Rightarrow \frac{n\varepsilon^2}{32} \geq \ln\left(\frac{2en}{d}\right)^d + \ln n + \ln\frac{1}{\delta}$$

$$e^{n\varepsilon^2/32} \geq n\left(\frac{2en}{d}\right)^d \frac{1}{\delta}$$

$$\delta \geq n\left(\frac{2en}{d}\right)^d e^{-n\varepsilon^2/32} \quad //$$

Also we know from PAC learnable for the Agnostic case ($VCdim = d < \infty$)

$$P\left(e_p(ERM(S)) - \min_{h\in H} e_p(h) > \varepsilon\right) \leq n\left(\frac{2en}{d}\right)^d e^{-n\varepsilon^2/32}$$

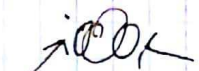
$$\Rightarrow P\left(e_p(ERM(S)) - \min_{h\in H} e_p(h) > \varepsilon\right) \leq \delta \quad //$$

Programming Assignment

(a) see Stdout when running script

(b) we can consider the classifier image of perceptron as an heat maps in which pixels with high and positive values are strong characterizers of "8" and not of "0" this pixels are those of the "cross" in the middle of "8" these pixels

and pixels that are sagnificant in the "0" digit but not in "8" will have negative valuse



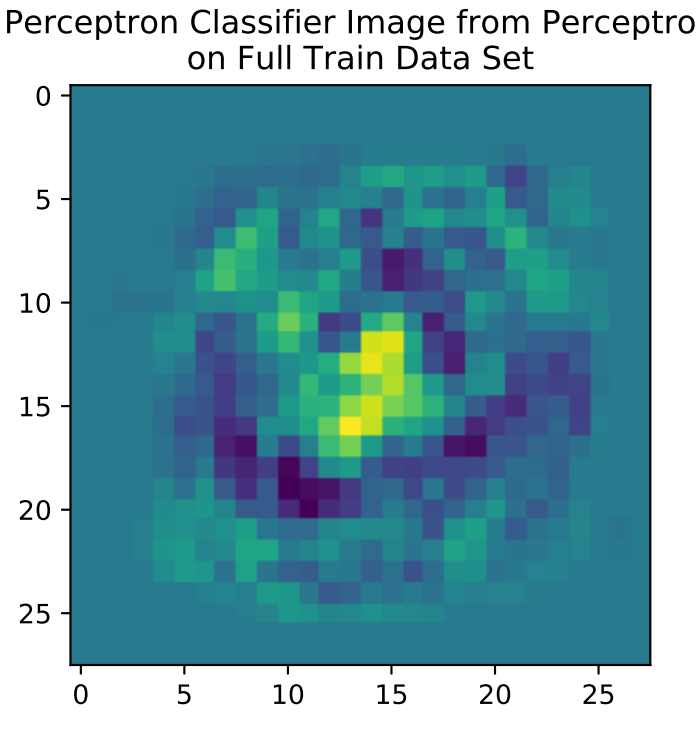
those pixels

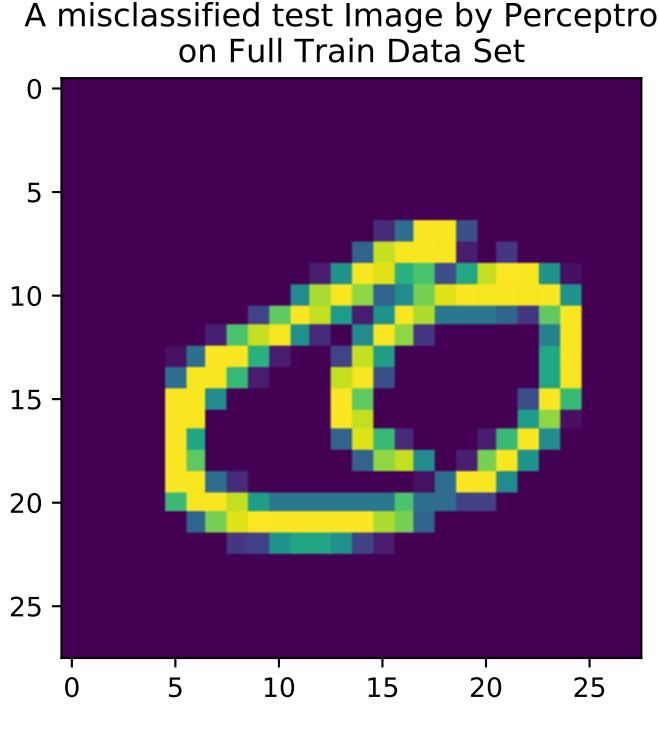
in that way, a scalar product of the classifier with "8" picturs tend to yield a posibive value and with "0" picturs - a negative value. as expected from the classifier

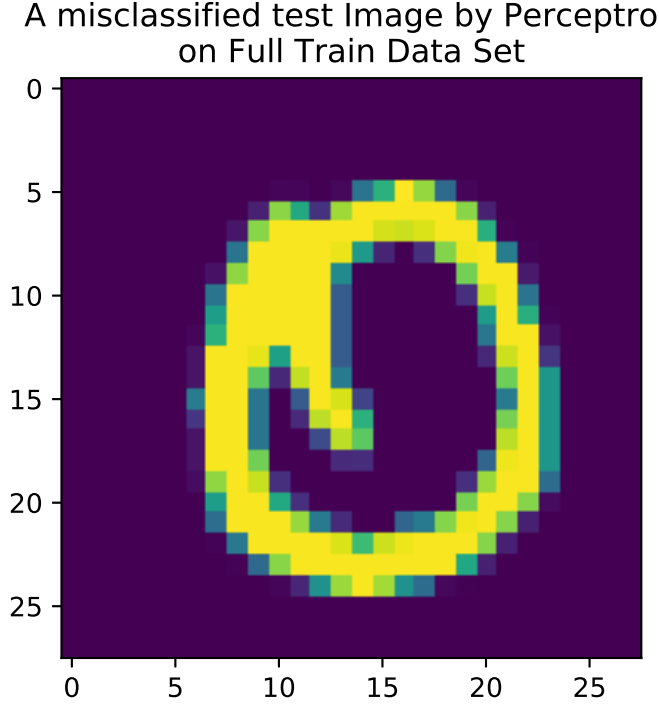
(c) Printed as stdout. see script or better, run it yourself

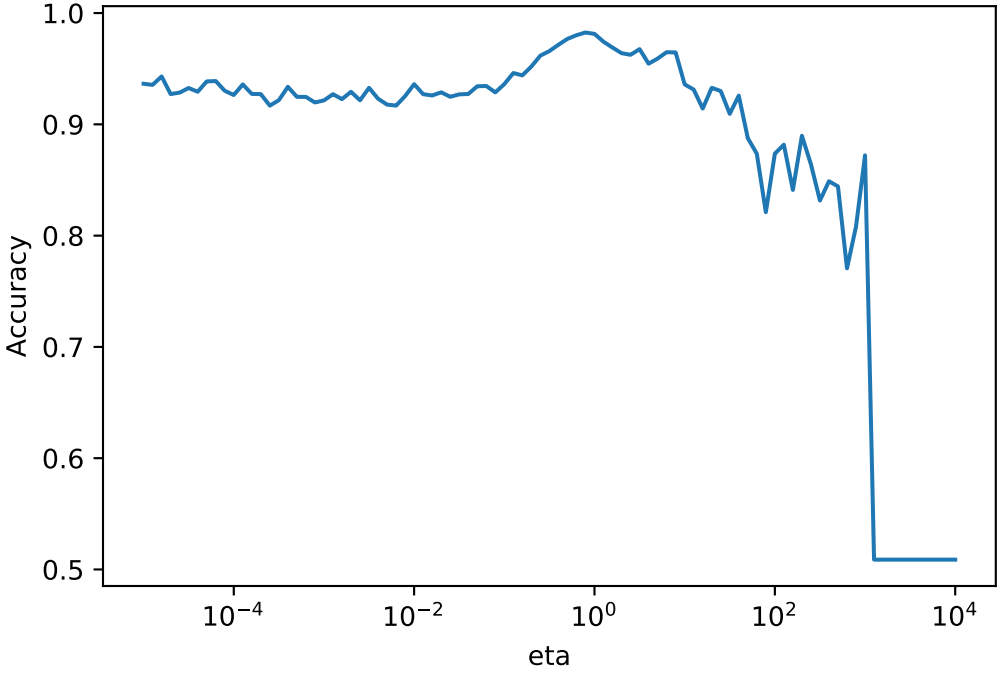
(d) The two misclassified picturs are "0" picturs that were classified as "8". not surprising, we can see that in both there is a smeare towards the middle of the picture, why the classifier has high positive values for classifying positively

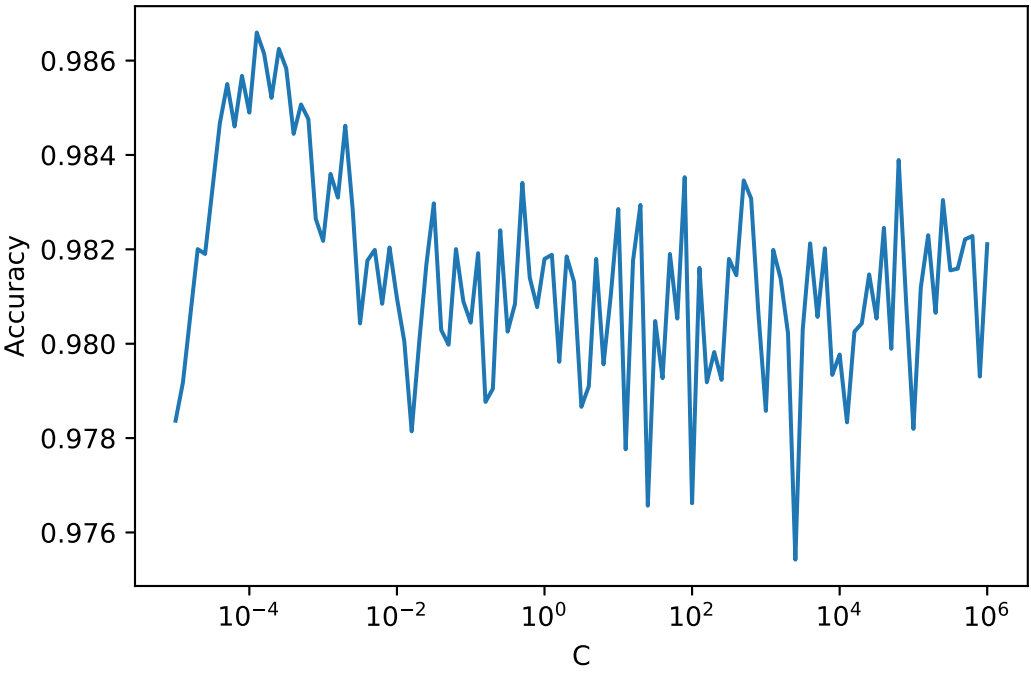
(2) see all results in stdout and attached plots.

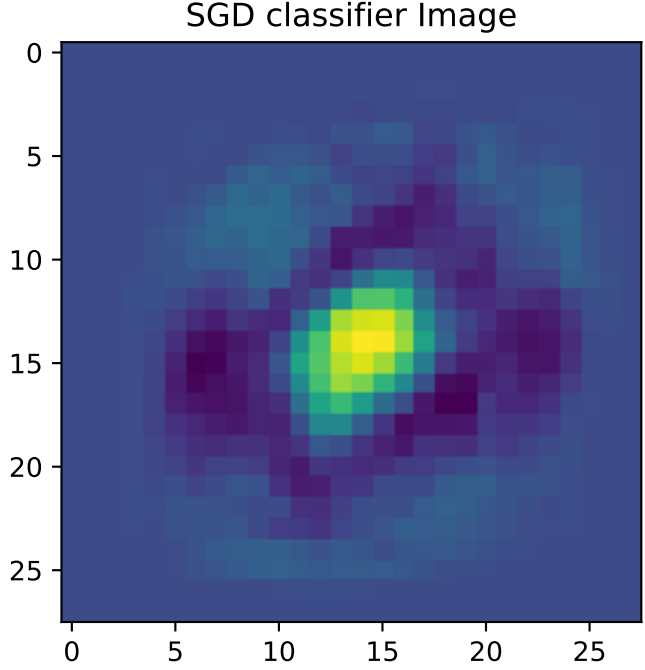

Perceptron Classifier Image from Perceptron
on Full Train Data Set
0
5
10
15
20
25
0
5
10
15
20
25


A misclassified test Image by Perceptron
on Full Train Data Set
0
5
10
15
20
25
0
5
10
15
20
25


A misclassified test Image by Perceptron
on Full Train Data Set
0
5
10
15
20
25
0
5
10
15
20
25


1.0
0.9
0.8
0.7
0.6
0.5
Accuracy
10^{-4}
10^{-2}
10^{0}
10^{2}
10^{4}
eta


0.986
0.984
0.982
0.980
0.978
0.976
Accuracy
10^{-4}
10^{-2}
10^{0}
10^{2}
10^{4}
10^{6}
C


SGD classifier Image
0
5
10
15
20
25
0
5
10
15
20
25

```
C:\Users\shosh\Google_Drive\Studies\Intro_to_ML\ex\ex3>python perceptron.py

Running perceptrion for train data sizes = [   5   10   50  100  500 1000 5000], 100 times per train data size

Accuracy results on test data:
N       Accuracy        5 percentile    95 percentile
5       0.9028          0.8332          0.9335
10      0.9096          0.8677          0.9483
50      0.9591          0.9295          0.9791
100     0.9684          0.9471          0.9826
500     0.9812          0.9611          0.9898
1000    0.9859          0.9769          0.9913
5000    0.9891          0.9805          0.9933

Running perceptrion for the full train data set
Accuracy of the result classifier on the test data set is 0.9877
```

```
C:\Users\shosh\Google_Drive\Studies\Intro_to_ML\ex\ex3>python sgd.py

Running SGD for deifferent values of eta_0, T = 1000, C = 1, iterations per eta = 10

sgd.py:56: RuntimeWarning: overflow encountered in multiply
  w = (1-eta_t)*w

Best eta is 0.794


Running SGD for different values of C, T = 1000, eta = 0.794, iterations per C = 10

Best C is 0.00025


Running SGD for C = 0.00025, T = 20000, eta = 0.794

Accuracy of best classifier is 0.993
```