1) Direct construction of Valid Kernels

(a) $K(x,z) = K_1(x,z) + K_2(x,z)$

⇒ Proof:

Using Mercer's Theorem,

K_1, K_2 are valid Kernels, then their K_1, K_2 (Gram matrixs) are positive-semidefinite for all possible choices of the data set $\{x^{(n)}\}_{n=1}^{N}$

$$a^T K_1 a = \sum_{i=1}^{N}\sum_{j=1}^{N} a_i K_{1,i,j} a_j \geq 0 \quad \forall a \in \mathbb{R}^N \quad (1)$$

$$a^T K_2 a = \sum_{i=1}^{N}\sum_{j=1}^{N} a_i K_{2,i,j} a_j \geq 0 \quad \forall a \in \mathbb{R}^N \quad (2)$$

adding (1) and (2)

$$a^T (K_1 + K_2) a = \sum_{i=1}^{N}\sum_{j=1}^{N} a_i (K_{1,i,j} + K_{2,i,j}) a_j \geq 0 \quad \forall a \in \mathbb{R}^N$$

$$a^T (K) a = \sum_{i=1}^{N}\sum_{j=1}^{N} a_i (K_{i,j}) a_j \geq 0 \quad \forall a \in \mathbb{R}^N$$

Hence $K(x,z) = K_1(x,z) + K_2(x,z)$ is a valid Kernel satisfy the Mercer's Theorem

Hence Proved

(b) $k(x,z) = K_1(x,z) - K_2(x,z)$

Using (1) and (2) from part (a), subtracting (2) from (1),

$a^T(K_1 - K_2)a = \sum_{i=1}^{N}\sum_{j=1}^{N} a_i (K_{1,i,j} - K_{2,i,j}) a_j$, $\forall a \in \mathbb{R}^N$

we cannot guarantee that $K_{1,i,j} - K_{2,i,j} \geq 0$ $\forall$ $1 \leq i \leq N$ and $1 \leq j \leq N$. Counter example, take $K_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $K_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$, hence for these K_1 and K_2 $a^T(K_1 - K_2)a \leq 0$, semi-negative, Hence, it is not a valid Kernel

(c) $K(x,x') = -a K_1(x,z)$

Using eq (1),

$b^T K_1 b = \sum_{i=1}^{N}\sum_{j=1}^{N} b_i K_{1,i,j} b_j \geq 0 \quad \forall b \in \mathbb{R}^N$

multiply the eq. by $-a$, where $a \in \mathbb{R}^+$ be a positive real number.

$b^T - aK_1 b = \sum_{i=1}^{N}\sum_{j=1}^{N} b_i - aK_{1,i,j} b_j \leq 0 \quad \forall b \in \mathbb{R}^N$

Hence, it is semi-negative,
Hence it is not a valid Kernel function, because of $(-)$, but if $(-a)$ was a positive real number, then it would be a Kernel.

(d) $K(x,z) = f(x)\, f(z)$

$\Rightarrow$ counter example,

$f(x) = \sum_{i=1}^{N} x_i \quad , \forall\, x \in \mathbb{R}^N$

let $x = \{1, -2, -1\}$, $N = 3$, $z = \{1, 1, 1\}$

using Mercer's theorem, the matrix K,

$K_{ij} = K(x^{(i)}, z^{(j)})$

$$K = \begin{bmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ -1 & -1 & -1 \end{bmatrix}$$ } Not symmetric and not semi-positive definite

Hence, Kernel is not a valid kernel.

(e) $K(x,z) = K_3(\phi(x), \phi(z))$

$\phi : \mathbb{R}^D \rightarrow \mathbb{R}^M$

$\Rightarrow$ K_3 is a valid Kernel;

$K(x,z)$ is symmetric, i.e $K_3(\phi(x), \phi(z)) = K_3(\phi(z), \phi(x))$

This is true as K_3 is Kernel so it is symmetric.

Now, Positive-semi difinite from Mercer's Theorem

$a^T K a \geq 0 \quad \forall\, a \in \mathbb{R}^N$, where

$K_{ij} = k_3(x^{(i)}, x^{(j)})$, K is symmetric and positive semi definite.

For, $k(x,z) = k_3(\phi(x), \phi(z))$,

we construct K', where $K'_{ij} = k_3(\phi(x^{(i)}), \phi(z^{(i)}))$

since k_3 is valid kernel, then K' will be a semi-positive definite matrix, hence

$a^T K' a \geq 0$, $\forall\ a \in \mathbb{R}^N$.

(g) $k(x,z) = k_1(x,z)\ k_2(x,z)$

Let $k_1(x,z) = \langle \phi^1(x), \phi^1(z) \rangle$ and

$k_2(x,z) = \langle \phi^2(x), \phi^2(z) \rangle$, $x \in \mathbb{R}^{D}$, $z \in \mathbb{R}^{D}$

$$k_1(x,z)\ k_2(x,z) = \langle \phi^1(x), \phi^1(z) \rangle \cdot \langle \phi^2(x), \phi^2(z) \rangle$$

$$= \left(\sum_{i=1}^{M} \phi^1_i(x)\, \phi^1_i(z) \right) \left(\sum_{j=1}^{M} \phi^2_j(x)\, \phi^2_j(z) \right)$$

$$= \sum_{i=1}^{M} \sum_{j=1}^{M} \left(\phi^1_i(x)\, \phi^2_j(x) \right) \cdot \left(\phi^1_i(z)\, \phi^2_j(z) \right)$$

$$= \langle \phi(x), \phi(z) \rangle$$

where, $\phi(x) = \begin{bmatrix} \phi_1^1(x)\ \phi_1^2(x) \\ \phi_1^1(x)\ \phi_2^2(x) \\ \phi_2^1(x)\ \phi_1^2(x) \\ \phi_1^1(x)\ \phi_3^2(x) \\ \phi_2^1(x)\ \phi_2^2(x) \\ \phi_3^1(x)\ \phi_1^2(x) \\ \vdots \end{bmatrix}$

ϕ is a feature map of $\Delta^2 \times 1$, where

$\phi_{i,j}(x) = \phi_i^1(x)\ \phi_j^2(x)$, $1 \le i, j \le \Delta$

Hence, $K(x,z)$ is a Kernel for a feature map $\phi(\cdot)$.

(5) $K(x,z) = p(K_1(x,z))$

$p: \mathbb{R} \to \mathbb{R}$ with positive coefficients.

$\Rightarrow$

since, polynomial term is a product of kernels with a positive kernel and then sum, i.e

$p(K_1(x,z)) = a_n(K_1(x,z))^n + a_{n-1}(K_1(x,z))^{n-1} + \cdots + a_0$, where

$a_n, a_{n-1}, \cdots a_0 > 0$ are positive coefficient

So, using property of (g) i.e
$k(x,z) = k_1(x,z)\, k_2(x,z)$ is a Kernel
and property of (c) $k(x,z) = a\, k_1(x,z)$
is a Kernel, where a is positive real number
and property of (a) $k(x,z) = k_1(x,z) + k_2(x,z)$
, we can say
$f(k_1(x,z))$ is also a Kernel
Hence Proved

(h) $k(x,z) = (x^T z + 1)^2$ is a Kernel,
feature map ϕ associated with $k(x,z)$ s.t
$k(x,z) = \phi(x)^T \phi(z)$. Let $D=2$
$\Rightarrow$ $x = \{x_1, x_2\}$, $z = \{z_1, z_2\}$

$$k(x,z) = (x_1 z_1 + x_2 z_2 + 1)^2$$
$$= x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 x_2 z_1 z_2 + 2x_1 z_1 + 2x_2 z_2 + 1$$
$$= \phi(x)^T \phi(z)$$

where $\phi(x) = (x_1^2, x_2^2, \sqrt{2}\, x_1 x_2, \sqrt{2} x_1, \sqrt{2} x_2, 1)$ is a feature map.

(i) Extra Credit

$$K(x,z) = \exp\left(-\|x-z\|^2/2\sigma^2\right)$$
$$= \phi(x)^T \phi(z)$$, find the feature vector $\phi(x)$

$\Rightarrow$ using Taylor series, $\exp(x) = \sum_{n=0}^{\infty} \frac{1}{n!} x^n$

$$K(x,z) = \exp\left(\frac{-x^Tx - z^Tz + 2x^Tz}{2\sigma^2}\right)$$

$$= \exp\left(\frac{-x^Tx}{2\sigma^2}\right) \exp\left(\frac{-z^Tz}{2\sigma^2}\right) \exp\left(\frac{x^Tz}{\sigma^2}\right)$$

$$= \exp\left(\frac{-x^Tx}{2\sigma^2}\right) \exp\left(\frac{-z^Tz}{2\sigma^2}\right) \left(\sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{x^Tz}{\sigma^2}\right)^n\right)$$

So

$$\phi(x) = \exp\left(-\frac{x^Tx}{2\sigma^2}\right) A$$, where A is

$$A = \left[\cdots \sqrt{\frac{1}{p!\,\sigma^{2p}}} \sqrt{\binom{p}{j_1, j_2, \cdots j_M}} x_1^{j_1} x_2^{j_2} \cdots x_M^{j_M} \cdots\right]$$

2) Implementing soft margin SVM by optimizing Primary Objective

(a) Derivatives of loss function $E(w,b)$ w.r.t w, b.

$$E(w,b) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \max\left(0, 1 - y^{(i)}\left(w^T\phi(x^{(i)}) + b\right)\right)$$

(i) Taking derivative w.r.t "w"

$$\Rightarrow \nabla_w E(w,b) = \nabla_w \frac{1}{2}\|w\|^2 + \frac{\partial}{\partial w}\left(C\sum_{i=1}^{N} \max\left(0, 1 - y^{(i)}\left(w^T\phi(x^{(i)}) + b\right)\right)\right),$$

$$\Rightarrow \nabla_w E(w,b) = w + C\sum_{i=1}^{N} \frac{\partial}{\partial w}\left(\max\left(0, 1 - y^{(i)}\left(w^T\phi(x^{(i)}) + b\right)\right)\right)$$

$$\nabla_w E(w,b) = w - C\sum_{i=1}^{N} \mathbb{I}\left[0 < \left(1 - y^{(i)}\left(w^T\phi(x^{(i)}) + b\right)\right)\right] y^{(i)}\phi(x^{(i)})$$

$$\nabla_w E(w,b) = w - C\sum_{i=1}^{N} \mathbb{I}\left[1 < \left(y^{(i)}\left(w^T\phi(x^{(i)}) + b\right)\right)\right] y^{(i)}\phi(x^{(i)})$$

(ii) Taking derivative w.r.t "b"

$$\frac{d}{db}E(w,b) = \frac{d}{db}\left(\frac{1}{2}\|w\|^2\right) + C\sum_{i=1}^{N}\frac{d}{db}\left(\max\left(0, 1-y^{(i)}\left(w^T\phi(x^{(i)})+b\right)\right)\right)$$

$$= -C\sum_{i=1}^{N}\mathbb{I}\left[1-y^{(i)}\left(w^T\phi(x^{(i)})+b\right)>0\right]y^{(i)}$$

$$\frac{d}{db}E(w,b) = -C\sum_{i=1}^{N}\mathbb{I}\left[y^{(i)}\left(w^T\phi(x^{(i)})+b\right)<1\right]y^{(i)}$$

(b) Autograder

(c)

[NumEpochs: 1] Accuracy: 54.17%
b: [0.01], W: [[0.224 -0.0855 0.545 0.206]]

[NumEpochs: 3] Accuracy: 54.17%
b: [0.02], W: [[0.44848122 -0.17019759 1.08918105 0.41163421]]

[NumEpochs: 10] Accuracy: 95.83%
b: [-0.14], W: [[-0.1648026 -0.80606447 1.37816462 0.57445096]]

[NumEpochs: 30] Accuracy: 95.83%
b: [-0.175], W: [[-0.20885861 -0.69979483 1.30489009 0.56605151]]

[NumEpochs: 100] Accuracy: 95.83%
b: [-0.315], W: [[-0.28240917 -0.77529188 1.75856715 0.82441652]]

3) Asymmetric SVM

$$\min_{w,b,\xi} \frac{1}{2} w^T w + c_0 \sum_{i:y^{(i)}=-1} \xi^{(i)} + c_1 \sum_{i:y^{(i)}=1} \xi^{(i)}$$

s.t $y^{(i)}(w^T \phi(x^{(i)}) + b) \geq 1 - \xi^{(i)}, \quad \forall i = 1, \ldots, N$

$$\xi^{(i)} \geq 0 \quad \forall i = 1, \ldots, N$$

(a) Defining

$$g_i(x) = 1 - \xi^{(i)} - y^{(i)}(w^T \phi(x^{(i)}) + b)$$
$$h_i(x) = -\xi^{(i)}$$

so,

$$\mathcal{L}(w, b, \xi, \alpha, \mu) = \frac{1}{2} w^T w + c_0 \sum_{i:y^{(i)}=-1} \xi^{(i)} + c_1 \sum_{i:y^{(i)}=1} \xi^{(i)} + \sum_{i=1}^{N} \alpha^{(i)} g_i(x) + \sum_{i=1}^{N} \mu^{(i)} h_i(x)$$

here $\alpha^{(i)}, \mu^{(i)} \geq 0 \quad \forall i = \{1, \ldots, N\}$

(b)

(i) $\nabla_w \mathcal{L}(w, b, \xi, \alpha, \mu)$

$$= w + \sum_{i=1}^{N} \alpha^{(i)} \frac{d}{dw} g_i(x)$$

$$\nabla_w \mathcal{L}(w, b, \xi, \alpha, \mu) = w - \sum_{i=1}^{N} \alpha^{(i)} \left(y^{(i)} \phi(x^{(i)}) \right)$$

(ii)

$$\frac{\partial \mathcal{L}(w,b,\xi,\alpha,\mu)}{\partial b} = \sum_{i=1}^{2} \alpha^{(i)} \frac{dg_i(x)}{dp}$$

$$\frac{\partial \mathcal{L}(w,b,\xi,\alpha,\mu)}{\partial b} = -\sum_{i=1}^{2} \alpha^{(i)} y^{(i)}$$

(iii) $\nabla_{\xi^{(i)}} \mathcal{L}(w,b,\xi,\alpha,\mu)$

$$\nabla_{\xi^{(i)}} \mathcal{L}(w,b,\xi,\alpha,\mu) = \sum_{i=1}^{2} c_0 \mathbb{I}\left[y^{(i)}=-1\right] + c_1 \mathbb{I}\left[y^{(i)}=1\right] - \sum_{i=1}^{2} \alpha^{(i)} - \sum_{i=1}^{2} \mu^{(i)}$$

(c) Dual optimization problem.

$$p^* = \max_{\substack{\alpha,\mu \\ \alpha \geq 0 \\ \mu^{(i)} \geq 0 \\ \forall i}} \min_{w,b} \left(\frac{1}{2} w^T w + c_0 \sum_{i:y^{(i)}=-1} \xi^{(i)} + c_1 \sum_{i:y^{(i)}=1} \xi^{(i)} + \sum_{i=1}^{2} \alpha^{(i)} g_i(x) + \sum_{i=1}^{2} \mu^{(i)} h_i(x) \right)$$

$\Rightarrow$ solving min first, $\nabla_w \mathcal{L}(w,b,\xi,\alpha,\mu) = 0$ and $\frac{\partial \mathcal{L}}{\partial b}(w,b,\xi,\alpha,\mu) = 0$

$$w = \sum_{i=1}^{N} \alpha^{(i)} y^{(i)} \phi(x^{(i)}) \quad \text{and} \quad 0 = \sum_{i=1}^{N} \alpha^{(i)} y^{(i)}$$

$$\max_{\alpha, \xi, \mu} \tilde{\mathcal{L}}(\alpha) = \sum_{i=1}^{N} \alpha^{(i)} \left(1 - \xi^{(i)}\right) + \sum_{i=1}^{N} \mu^{(i)} \left(\xi^{(i)} - 1\right) + \sum_{i=1}^{N} \left(C - \left[1 - \xi^{(i)}\right] \Pi^{(i)} + \left[1 - \xi^{(i)}\right] \Pi^{(i)}\right) \xi^{(i)}$$

$$- \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha^{(i)} \alpha^{(j)} y^{(i)} y^{(j)} \phi\left(x^{(i)}\right)^{T} \phi\left(x^{(j)}\right)$$

$$\text{s.t. } \alpha^{(i)} \geq 0, \quad \mu^{(i)} \geq 0, \quad \forall i$$

This can be rewritten as below by removing redundant variable μ.

$$\max_{\alpha, \xi} \tilde{\mathcal{L}}(\alpha) = \sum_{i=1}^{N} \alpha^{(i)} \left(1 - \xi^{(i)}\right) + \sum_{i=1}^{N} \left(C - \left[1 - \xi^{(i)}\right] \Pi^{(i)} + \left[1 - \xi^{(i)}\right] \Pi^{(i)}\right) \xi^{(i)}$$

$$- \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha^{(i)} \alpha^{(j)} y^{(i)} y^{(j)} \phi\left(x^{(i)}\right)^{T} \phi\left(x^{(j)}\right)$$

$$\text{s.t. } \alpha^{(i)} \geq 0, \quad \forall i$$

4) SVMs with Convex optimization

(a)

$$\underset{x}{\text{minimize}} \quad \frac{1}{2} x^T P x + q^T x$$

$$\text{subject to} \quad Gx \le h$$

$$Ax = b$$

$$\text{Minimize} \quad \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha^{(n)} \alpha^{(m)} y^{(n)} y^{(m)} k(x^{(n)}, x^{(m)}) - \sum_{n=1}^{N} \alpha^{(n)}$$

$$\text{subject to} \quad 0 \le \alpha^{(n)} \le C$$

$$\sum_{n=1}^{N} \alpha^{(n)} y^{(n)} = 0$$

Here, $b = 0$, $x = [\alpha^1, \alpha^2, \cdots \alpha^N]^T$ $(N \times 1)$ matrix

$A = [y^1, y^2, \cdots y^N]$ $(1 \times N)$ matrix

$q = [-1, -1, \cdots, -1]$ $(N \times 1)$ matrix

$h = [0, 0, \cdots, 0, C, C, \cdots C]$ $((N+N) \times 1)$ matrix

$G = \begin{bmatrix} -I_{N \times N} \\ I_{N \times N} \end{bmatrix}$ $((N+N) \times N)$ matrix; I is a identity matrix

$P = \begin{bmatrix} p_{1,1} & \cdots & p_{1,N} \\ \vdots & & \vdots \\ p_{N,1} & & p_{N,N} \end{bmatrix}$, $(N \times N$ matrix$)$

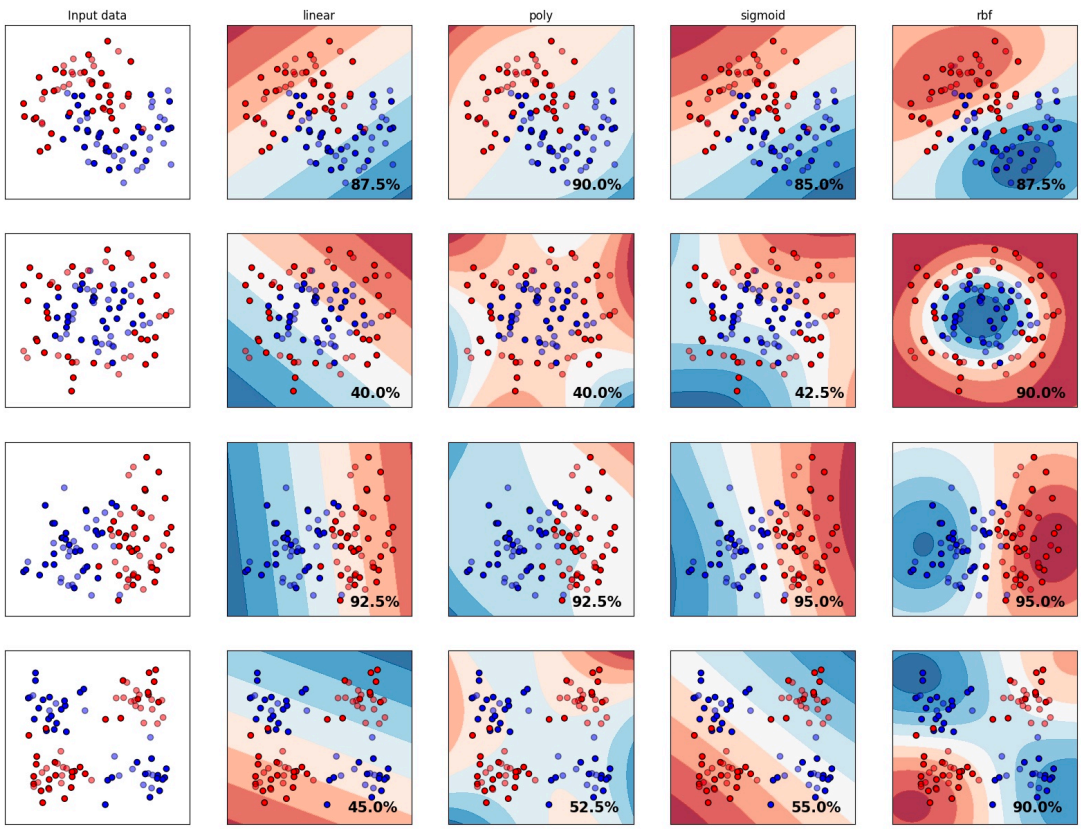
where $p_{i,j} = y^{(i)} y^{(j)} k(x^{(i)} x^{(j)})$

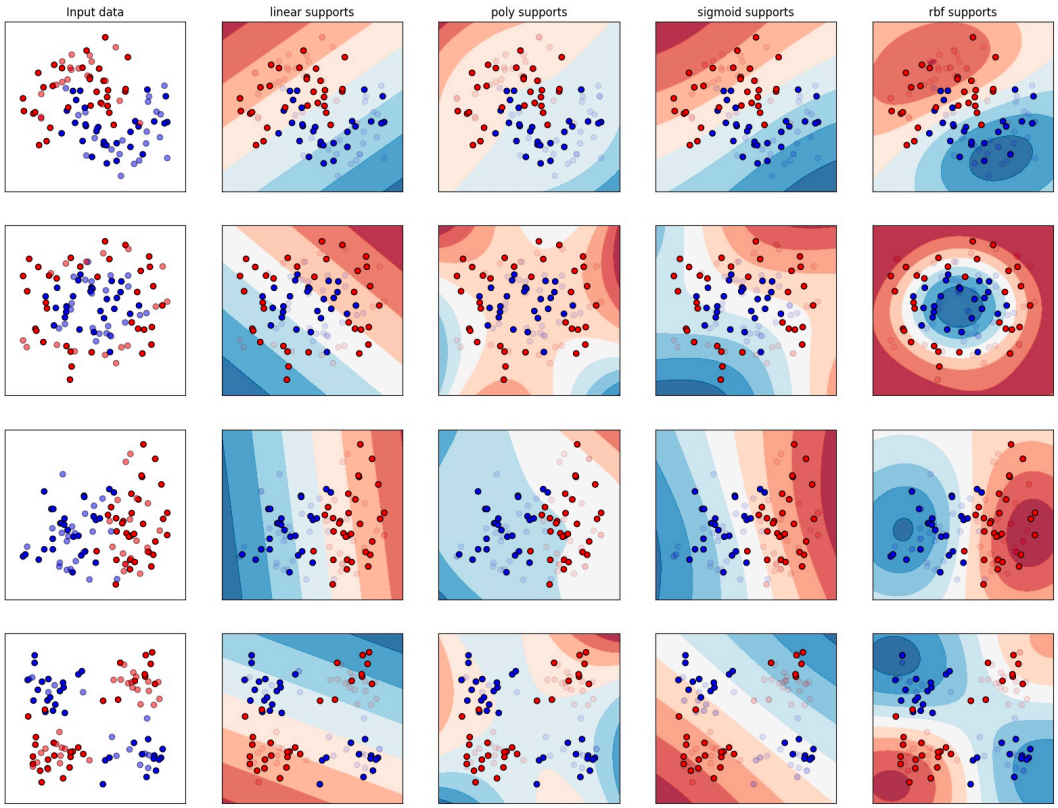
b Autograder

c

CVXOPTSVM performance

	linear	poly	sigmoid	rbf
dataset 0	87.50	90.00	85.00	87.50
dataset 1	40.00	40.00	42.50	90.00
dataset 2	92.50	92.50	95.00	95.00
dataset 3	45.00	52.50	55.00	90.00




Input data
linear supports
poly supports
sigmoid supports
rbf supports

5) Neural Network layer Implementation

(a) $Y = XW + B$, $Y \in \mathbb{R}^{N \times D_{out}}$

$B \in \mathbb{R}^{N \times D_{out}}$

$X \in \mathbb{R}^{N \times D_{in}}$

$W \in \mathbb{R}^{D_{in} \times D_{out}}$

$\frac{\partial \mathcal{L}}{\partial W}$

$\mathcal{L} \in \mathbb{R}$ } Loss function

Let $\frac{\partial \mathcal{L}}{\partial W_{i,j}}$ is the i^{th} row, j^{th} column element of $\frac{\partial \mathcal{L}}{\partial W} \in \mathbb{R}^{D_{in} \times D_{out}}$

$1 \le n \le N$, $1 \le i \le D_{in}$, $1 \le j \le D_{out}$

$$\Rightarrow \frac{\partial \mathcal{L}}{\partial W_{i,j}} = \sum_{n=1}^{N} \frac{\partial \mathcal{L}}{\partial Y_j^{(n)}} \frac{\partial Y_j^{(n)}}{\partial W_{i,j}}$$

where $Y_j^{(n)} = X^{(n)} W_j + B_j^{(n)}$

here $Y_j^{(n)}$ is the output of j^{th} neuron for the n^{th} samples in a batch.

$$\frac{\partial \mathcal{L}}{\partial W_{i,j}} = \sum_{n=1}^{N} \frac{\partial \mathcal{L}}{\partial Y_j^{(n)}} X_i^{(n)} = X_i^T \frac{\partial \mathcal{L}}{\partial Y_j}$$

Hence,

$$\boxed{\frac{\partial \mathcal{L}}{\partial W} = X^T \frac{\partial \mathcal{L}}{\partial Y}}$$

$\frac{\partial f}{\partial b}$

$$\frac{\partial f}{\partial b_j} = \sum_{n=1}^{N} \sum_{m=1}^{D_{out}} \frac{\partial f}{\partial Y_m^{(n)}} \frac{\partial Y_m^{(n)}}{\partial b_j}$$

where $Y_m^{(n)} = X^{(n)} W_m + B_m^{(n)}$ $\quad \Big| \quad \frac{\partial Y_m^{(n)}}{\partial b_j} = 1$

$$\frac{\partial f}{\partial b_j} = \sum_{n=1}^{N} \sum_{m=1}^{D_{out}} \frac{\partial f}{\partial Y_m^{(n)}} \times 1$$

$$\boxed{\frac{\partial f}{\partial B} = \sum_{n=1}^{N} \frac{\partial f}{\partial Y^{(n)}}}$$

$\frac{\partial f}{\partial X}$

$$\frac{\partial f}{\partial X_i^{(n)}} = \sum_{n'=1}^{N} \sum_{m=1}^{D_{out}} \frac{\partial f}{\partial Y_m^{(n')}} \frac{\partial Y_m^{(n')}}{\partial X_i^{(n)}}$$

where $Y_m^{(n)} = X^{(n)} W_m + B_m^{(n)}$

as the n^{th}-sample in X is only related with the n^{th}-sample in Y and $\frac{\partial Y_m^{(n')}}{\partial X_i^{(n)}} = 0 \;\forall\; n' \neq n$ so,

$$\frac{\partial L}{\partial X_{i,m}^{(n)}} = \sum_{m=1}^{D_{out}} \frac{\partial L}{\partial Y_m^{(n)}} \frac{\partial Y_m^{(n)}}{\partial X_{i,m}^{(n)}}$$

$$\frac{\partial L}{\partial X_{i,m}^{(n)}} = \sum_{m=1}^{D_{out}} \frac{\partial L}{\partial Y_m^{(n)}} \left(W_{i,m} \right)$$

$$\frac{\partial L}{\partial X_{i,m}^{(n)}} = \frac{\partial L}{\partial Y^{(n)}} \left(W_{i,:} \right)^T$$

$$\boxed{\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y} W^T}$$

(b) $X \in \mathbb{R}^{N \times D}$, $Y = ReLU(X)$

$x \in X$, $y = ReLU(x) = \max(0, x)$

$\frac{\partial L}{\partial X} \in \mathbb{R}^{N \times D}$, $1 \le i \le N$, $1 \le j \le D$

Let $\frac{\partial L}{\partial X_{i,j}} = \left(\frac{\partial L}{\partial Y_{i,j}} \right) \frac{\partial Y_{i,j}}{\partial X_{i,j}}$ } True as ReLU is applied element wise

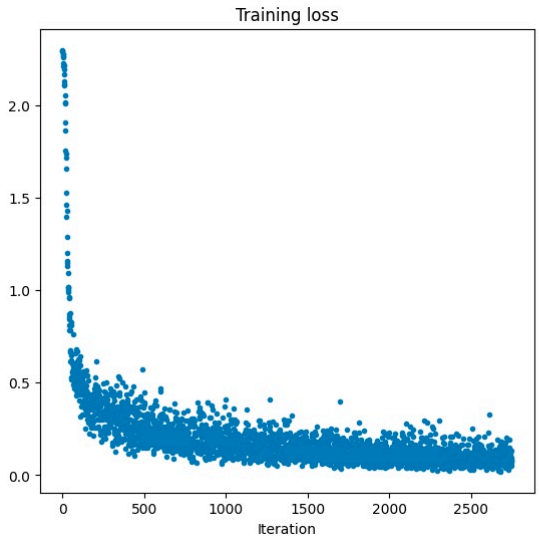
$$\frac{\partial L}{\partial X_{i,j}} = \frac{\partial L}{\partial Y_{i,j}} \cdot \mathbb{I}(X_{i,j} > 0)$$

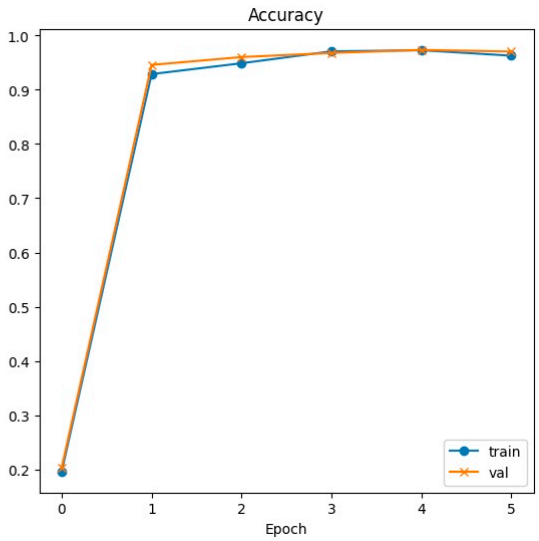
$\odot$ is Element wise multiplication

$$\boxed{\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y} \odot \mathbb{I}(X > 0)}$$

c Autograder

d





e

(Epoch 0 / 5) train acc: 19.60% val_acc: 20.36%

(Epoch 1 / 5) train acc: 92.90% val_acc: 94.60%

(Epoch 2 / 5) train acc: 94.90% val_acc: 96.04%

(Epoch 3 / 5) train acc: 97.10% val_acc: 96.80%

(Epoch 4 / 5) train acc: 97.30% val_acc: 97.36%

(Epoch 5 / 5) train acc: 96.30% val_acc: 97.06%

Accuracy on training set

Test accuracy: 96.31%

Accuracy on test set.