

CS 446 MJT — Homework 3

junzhew3
haoruan8

Version 1

Instructions.

- Homework is due **Tuesday, March 12, at 11:59pm**; no late homework accepted.
- Everyone must submit individually at gradescope under **hw3** and **hw3code**.
- The “written” submission at **hw3 must be typed**, and submitted in any format gradescope accepts (to be safe, submit a PDF). You may use L^AT_EX, markdown, google docs, MS word, whatever you like; but it must be typed!
- When submitting at **hw3**, gradescope will ask you to mark out boxes around each of your answers; please do this precisely!
- Please make sure your NetID is clear and large on the first page of the homework.
- Your solution **must** be written in your own words. Please see the course webpage for full academic integrity information. Briefly, you may have high-level discussions with at most 3 classmates, whose NetIDs you should place on the first page of your solutions, and you should cite any external reference you use; despite all this, your solution must be written in your own words.
- We reserve the right to reduce the auto-graded score for **hw3code** if we detect funny business (e.g., rather than implementing an algorithm, you keep re-submitting the assignment to the auto-grader, eventually completing a binary search for the answers).
- There are **no regrade requests** on **hw3code**, which is the code auto-grader; however, you can re-submit and re-grade as many times as you like before the deadline! Start early and report any issues on piazza!
- Methods and functions in the template and utility code include docstrings describing the inputs and outputs. The autograder relies on correct implementations of these methods. Follow the docstrings to avoid failing tests.

1. The ln-sum-exp and cross entropy.

- (a) Given $\mathbf{z} \in \mathbb{R}^k$, prove that $g(\mathbf{z}) = \ln \sum_{j=1}^k \exp(z_j)$ is convex.

Hint: prove that the Hessian matrix is positive semi-definite, meaning $\nabla^2 g(\mathbf{z}) \succeq 0$.

- (b) Recall that given a data example (\mathbf{x}, y) where $\mathbf{x} \in \mathbb{R}^d$ and $y \in \{1, 2, \dots, k\}$, and a classifier $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$, the cross entropy loss is defined as

$$\ell_{\text{ce}}(f(\mathbf{x}), y) = -\ln \left(\frac{\exp(f(\mathbf{x})_y)}{\sum_{j=1}^k \exp(f(\mathbf{x})_j)} \right) = -f(\mathbf{x})_y + \ln \sum_{j=1}^k \exp(f(\mathbf{x})_j).$$

Let data examples $((\mathbf{x}_i, y_i))_{i=1}^n$ be given, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, 2, \dots, k\}$. Consider the linear predictor $\mathbf{x} \mapsto \mathbf{W}\mathbf{x}$, where $\mathbf{W} \in \mathbb{R}^{k \times d}$. Prove that the empirical risk

$$\hat{\mathcal{R}}(\mathbf{W}) = \frac{1}{n} \sum_{i=1}^n \ell_{\text{ce}}(\mathbf{W}\mathbf{x}_i, y_i)$$

is convex.

Hint: use part (a) and the fact that convexity is preserved under affine composition and nonnegative combination. (It doesn't matter that this part uses matrix variables; this affine composition property holds for convex functions over matrices, and then when applying the previous part, its input is a vector after the affine transformation.)

- (c) For $\mathbf{z} \in \mathbb{R}^k$ and $r > 0$, let $g_r(\mathbf{z}) = \frac{1}{r} \ln \sum_{j=1}^k \exp(rz_j)$. Prove that

$$\lim_{r \rightarrow \infty} g_r(\mathbf{z}) = \max_{1 \leq j \leq k} z_j.$$

- (d) As a corollary, for $z \in \mathbb{R}$ and $r > 0$, the logistic loss $\ell(z) = \ln(1 + \exp(-z))$, and $\ell_r(z) = \frac{1}{r} \ln(1 + \exp(-rz))$, prove that

$$\lim_{r \rightarrow \infty} \ell_r(z) = \max\{0, -z\} = \text{ReLU}(-z).$$

Solution. (Your solution here.)

- (a) Hessian matrix of $g(\mathbf{z})$:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 g}{\partial z_1^2} & \frac{\partial^2 g}{\partial z_1 \partial z_2} & \cdots & \frac{\partial^2 g}{\partial z_1 \partial z_k} \\ \frac{\partial^2 g}{\partial z_2 \partial z_1} & \frac{\partial^2 g}{\partial z_2^2} & \cdots & \frac{\partial^2 g}{\partial z_2 \partial z_k} \\ \cdots & \cdots & \ddots & \cdots \\ \frac{\partial^2 g}{\partial z_k \partial z_1} & \frac{\partial^2 g}{\partial z_k \partial z_2} & \cdots & \frac{\partial^2 g}{\partial z_k^2} \end{bmatrix} \quad (1)$$

$$\frac{\partial^2 g}{\partial z_1^2} = \frac{\exp(z_1)(\sum_{j=1}^k \exp(z_j)) - \exp(z_1)\exp(z_1)}{(\sum_{j=1}^k \exp(z_j))^2}, \quad \frac{\partial^2 g}{\partial z_1 \partial z_2} = \frac{0(\sum_{j=1}^k \exp(z_j)) - \exp(z_1)\exp(z_2)}{(\sum_{j=1}^k \exp(z_j))^2} \quad (2)$$

$$\cdots, \quad \frac{\partial^2 g}{\partial z_1 \partial z_k} = \frac{0(\sum_{j=1}^k \exp(z_j)) - \exp(z_1)\exp(z_k)}{(\sum_{j=1}^k \exp(z_j))^2} \quad (3)$$

$$\frac{\partial^2 g}{\partial z_2 \partial z_1} = \frac{0(\sum_{j=1}^k \exp(z_j)) - \exp(z_2)\exp(z_1)}{(\sum_{j=1}^k \exp(z_j))^2}, \quad \frac{\partial^2 g}{\partial z_2^2} = \frac{\exp(z_2)(\sum_{j=1}^k \exp(z_j)) - \exp(z_2)\exp(z_2)}{(\sum_{j=1}^k \exp(z_j))^2} \quad (4)$$

$$\dots, \frac{\partial^2 g}{\partial z_2 \partial z_k} = \frac{0(\sum_{j=1}^k \exp(z_j)) - \exp(z_2) \exp(z_k)}{(\sum_{j=1}^k \exp(z_j))^2} \quad (5)$$

$$\dots \quad (6)$$

$$\frac{\partial^2 g}{\partial z_k \partial z_1} = \frac{0(\sum_{j=1}^k \exp(z_j)) - \exp(z_k) \exp(z_1)}{(\sum_{j=1}^k \exp(z_j))^2}, \frac{\partial^2 g}{\partial z_k \partial z_2} = \frac{0(\sum_{j=1}^k \exp(z_j)) - \exp(z_k) \exp(z_2)}{(\sum_{j=1}^k \exp(z_j))^2} \quad (7)$$

$$\dots, \frac{\partial^2 g}{\partial z_k^2} = \frac{\exp(z_k)(\sum_{j=1}^k \exp(z_j)) - \exp(z_k) \exp(z_k)}{(\sum_{j=1}^k \exp(z_j))^2} \quad (8)$$

So:

$$\mathbf{H} = \frac{1}{(\sum_{j=1}^k \exp(z_j))} \text{diag}(\exp(z_1), \exp(z_2), \dots, \exp(z_k)) - \frac{1}{(\sum_{j=1}^k \exp(z_j))^2} \mathbf{Z}^T \mathbf{Z} \quad (9)$$

$$\mathbf{Z}^T = [\exp(z_1), \exp(z_2), \dots, \exp(z_k)] \in \mathbb{R}^{k \times 1} \quad (10)$$

So for any non-zero $\mathbf{v} \in \mathbb{R}^k$:

$$\mathbf{v}^T \mathbf{H} \mathbf{v} = \frac{(\sum_{j=1}^k \exp(z_j) \mathbf{v}_j^2)(\sum_{j=1}^k \exp(z_j)) - (\sum_{j=1}^k \exp(z_j) \mathbf{v}_j)^2}{(\sum_{j=1}^k \exp(z_j))^2} \quad (11)$$

Using Cauchy-Schwarz inequality:

$$\left(\sum_{j=1}^k \left(\sqrt{\exp(z_j) \mathbf{v}_j^2} \right)^2 \right) \left(\sum_{j=1}^k \left(\sqrt{\exp(z_j)} \right)^2 \right) \geq \left(\sum_{j=1}^k \exp(z_j) \mathbf{v}_j \right)^2 \quad (12)$$

so $\mathbf{v}^T \mathbf{H} \mathbf{v} \geq 0$, \mathbf{H} is a positive semi-definite matrix, so $g(\mathbf{z})$ is convex.

(b) assume $\mathbf{z} = [f(\mathbf{x})_1, f(\mathbf{x})_2, \dots, f(\mathbf{x})_k]$. using the conclusion in part(a):

$\ln \sum_{j=1}^k \exp(z_j) = \ln \sum_{j=1}^k \exp(f(\mathbf{x})_j)$ is convex.

$$\ell_{\text{ce}}(f(\mathbf{x}), y) = -\ln \left(\frac{\exp(f(\mathbf{x})_y)}{\sum_{j=1}^k \exp(f(\mathbf{x})_j)} \right) = \ln \left(\frac{\sum_{j=1}^k \exp(f(\mathbf{x})_j)}{\exp(f(\mathbf{x})_y)} \right), \mathbf{A} = \frac{1}{\exp(f(\mathbf{x})_y)}$$

Since convexity is preserved under affine composition, $\ell_{\text{ce}}(f(\mathbf{x}), y)$ is also convex.

$$\widehat{\mathcal{R}}(\mathbf{W}) = \frac{1}{n} \sum_{i=1}^n \ell_{\text{ce}}(\mathbf{W} \mathbf{x}, y), \alpha_1 = \dots = \alpha_n = \frac{1}{n} > 0$$

Since convexity is preserved under nonnegative combination of convex functions, $\widehat{\mathcal{R}}(\mathbf{W})$ is also convex.

(c) assume $m = \max_{1 \leq j \leq k} z_j$

$$\lim_{r \rightarrow \infty} \frac{1}{r} \ln \exp(rm) \leq \lim_{r \rightarrow \infty} g_r(\mathbf{z}) \leq \lim_{r \rightarrow \infty} \frac{1}{r} \ln \sum_{i=1}^k \exp(rm) \quad (13)$$

$$\lim_{r \rightarrow \infty} \frac{1}{r} \ln \exp(rm) = m, \lim_{r \rightarrow \infty} \frac{1}{r} \ln \sum_{i=1}^k \exp(rm) = \lim_{r \rightarrow \infty} \frac{1}{r} (\ln k + \ln \exp(rm)) = 0 + m = m \quad (14)$$

so:

$$\begin{aligned} m &\leq \lim_{r \rightarrow \infty} g_r(\mathbf{z}) \leq m \\ \lim_{r \rightarrow \infty} g_r(\mathbf{z}) &= m = \max_{1 \leq j \leq k} z_j \end{aligned} \quad (15)$$

(d) if $z \geq 0$, $r \rightarrow \infty$, $-rz \rightarrow -\infty$, $\exp(-rz) \rightarrow 0$, so:

$$\lim_{r \rightarrow \infty} \ell_r(z) = \lim_{r \rightarrow \infty} \frac{1}{r} \ln(1) = 0$$

if $z < 0$, $r \rightarrow \infty$, $-rz \rightarrow \infty$,

$$\lim_{r \rightarrow \infty} \ell_r(z) = \lim_{r \rightarrow \infty} \frac{1}{r} \ln(1 + \exp(-rz)) = \lim_{r \rightarrow \infty} \frac{1}{r} \ln(\exp(-rz)) = -z$$

So:

$$\lim_{r \rightarrow \infty} \ell_r(z) = \max\{0, -z\} = \text{ReLU}(-z).$$

2. On initialization.

Consider a 2-layer network

$$f(\mathbf{x}; \mathbf{W}, \mathbf{v}) = \sum_{j=1}^m v_j \sigma(\langle \mathbf{w}_j, \mathbf{x} \rangle),$$

where $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{W} \in \mathbb{R}^{m \times d}$ with rows \mathbf{w}_j^\top , and $\mathbf{v} \in \mathbb{R}^m$. For simplicity, the network has a single output, and bias terms are omitted.

Given a data example (\mathbf{x}, y) and a loss function ℓ , consider the empirical risk

$$\widehat{\mathcal{R}}(\mathbf{W}, \mathbf{v}) = \ell(f(\mathbf{x}; \mathbf{W}, \mathbf{v}), y).$$

Only a single data example will be considered in this problem; the same analysis extends to multiple examples by taking averages.

- (a) For each $1 \leq j \leq m$, derive $\partial \widehat{\mathcal{R}} / \partial v_j$ and $\partial \widehat{\mathcal{R}} / \partial \mathbf{w}_j$.
- (b) Consider gradient descent which starts from some $\mathbf{W}^{(0)}$ and $\mathbf{v}^{(0)}$, and at step $t \geq 0$, updates the weights for each $1 \leq j \leq m$ as follows:

$$\mathbf{w}_j^{(t+1)} = \mathbf{w}_j^{(t)} - \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{w}_j^{(t)}}, \quad \text{and} \quad v_j^{(t+1)} = v_j^{(t)} - \eta \frac{\partial \widehat{\mathcal{R}}}{\partial v_j^{(t)}}.$$

Suppose there exists two hidden units $p, q \in \{1, 2, \dots, m\}$ such that $\mathbf{w}_p^{(0)} = \mathbf{w}_q^{(0)}$ and $v_p^{(0)} = v_q^{(0)}$. Prove by induction that for any step $t \geq 0$, it holds that $\mathbf{w}_p^{(t)} = \mathbf{w}_q^{(t)}$ and $v_p^{(t)} = v_q^{(t)}$.

Remark: as a result, if the neural network is initialized symmetrically, then such a symmetry may persist during gradient descent, and thus the representation power of the network will be limited.

- (c) Random initialization is a good way to break symmetry. Moreover, proper random initialization also preserves the squared norm of the input, as formalized below.

First consider the identity activation $\sigma(z) = z$. For each $1 \leq j \leq m$ and $1 \leq k \leq d$, initialize $w_{j,k}^{(0)} \sim \mathcal{N}(0, 1/m)$ (i.e., normal distribution with mean $\mu = 0$ and variance $\sigma^2 = 1/m$). Prove that

$$\mathbb{E} \left[\left\| \mathbf{W}^{(0)} \mathbf{x} \right\|_2^2 \right] = \|\mathbf{x}\|_2^2.$$

Next consider the ReLU activation $\sigma_r(z) = \max\{0, z\}$. For each $1 \leq j \leq m$ and $1 \leq k \leq d$, initialize $w_{j,k}^{(0)} \sim \mathcal{N}(0, 2/m)$. Prove that

$$\mathbb{E} \left[\left\| \sigma_r(\mathbf{W}^{(0)} \mathbf{x}) \right\|_2^2 \right] = \|\mathbf{x}\|_2^2.$$

Hint: linear combinations of Gaussians are again Gaussian! For the second part (with ReLU), consider the symmetry of a Gaussian around 0.

Solution. (Your solution here.)

(a)

$$\frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{v}_j} = \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial \mathbf{v}_j} = \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \sigma(\langle \mathbf{w}_j, \mathbf{x} \rangle) \quad (16)$$

$$\frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{w}_j} = \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \frac{\partial \mathbf{f}}{\partial \sigma} \frac{\partial \sigma}{\partial \langle \mathbf{w}_j, \mathbf{x} \rangle} \frac{\partial \langle \mathbf{w}_j, \mathbf{x} \rangle}{\partial \mathbf{w}_j} = \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \frac{\partial \sigma}{\partial \langle \mathbf{w}_j, \mathbf{x} \rangle} \mathbf{v}_j \mathbf{x} \quad (17)$$

(b) using mathematical induction:
when $t = 0$:

$$\mathbf{w}_p^{(1)} = \mathbf{w}_p^{(0)} - \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{w}_p^{(0)}} = \mathbf{w}_p^{(0)} - \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \frac{\partial \sigma}{\partial \langle \mathbf{w}_p, \mathbf{x} \rangle} \mathbf{v}_p^{(0)} \mathbf{x} \quad (18)$$

$$\mathbf{w}_q^{(1)} = \mathbf{w}_q^{(0)} - \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{w}_q^{(0)}} = \mathbf{w}_q^{(0)} - \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \frac{\partial \sigma}{\partial \langle \mathbf{w}_q, \mathbf{x} \rangle} \mathbf{v}_q^{(0)} \mathbf{x} \quad (19)$$

$$so : \mathbf{w}_p^{(1)} - \mathbf{w}_q^{(1)} = \mathbf{w}_p^{(0)} - \mathbf{w}_q^{(0)} - \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \frac{\partial \sigma}{\partial \langle \mathbf{w}_p, \mathbf{x} \rangle} \mathbf{v}_p^{(0)} \mathbf{x} + \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \frac{\partial \sigma}{\partial \langle \mathbf{w}_q, \mathbf{x} \rangle} \mathbf{v}_q^{(0)} \mathbf{x} \quad (20)$$

$$since : \mathbf{w}_p^{(0)} = \mathbf{w}_q^{(0)} \quad \text{and} \quad \mathbf{v}_p^{(0)} = \mathbf{v}_q^{(0)} \quad (21)$$

$$so : \mathbf{w}_p^{(1)} - \mathbf{w}_q^{(1)} = 0 \quad \text{and} \quad \mathbf{w}_p^{(1)} = \mathbf{w}_q^{(1)} \quad (22)$$

if the conditions are true at $t = m$ which means $\mathbf{w}_p^{(m)} = \mathbf{w}_q^{(m)}$ and $\mathbf{v}_p^{(m)} = \mathbf{v}_q^{(m)}$

$$so : \mathbf{w}_p^{(m+1)} - \mathbf{w}_q^{(m+1)} = \mathbf{w}_p^{(m)} - \mathbf{w}_q^{(m)} - \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \frac{\partial \sigma}{\partial \langle \mathbf{w}_p, \mathbf{x} \rangle} \mathbf{v}_p^{(m)} \mathbf{x} + \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \frac{\partial \sigma}{\partial \langle \mathbf{w}_q, \mathbf{x} \rangle} \mathbf{v}_q^{(m)} \mathbf{x} = 0 \quad (23)$$

$$so : \mathbf{w}_p^{(m+1)} = \mathbf{w}_q^{(m+1)} \quad (24)$$

so for any step $t \geq 0$, $\mathbf{w}_p^{(t)} = \mathbf{w}_q^{(t)}$

when $t = 0$:

$$\mathbf{v}_p^{(1)} - \mathbf{v}_q^{(1)} = \mathbf{v}_p^{(0)} - \mathbf{v}_q^{(0)} - \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \sigma(\langle \mathbf{w}_p^{(0)}, \mathbf{x} \rangle) + \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \sigma(\langle \mathbf{w}_q^{(0)}, \mathbf{x} \rangle) \quad (25)$$

$$since : \mathbf{w}_p^{(0)} = \mathbf{w}_q^{(0)} \quad \text{and} \quad \mathbf{v}_p^{(0)} = \mathbf{v}_q^{(0)} \quad (26)$$

$$so : \mathbf{v}_p^{(1)} - \mathbf{v}_q^{(1)} = 0 \quad \text{and} \quad \mathbf{v}_p^{(1)} = \mathbf{v}_q^{(1)} \quad (27)$$

if the conditions are true at $t = m$ which means $\mathbf{w}_p^{(m)} = \mathbf{w}_q^{(m)}$ and $\mathbf{v}_p^{(m)} = \mathbf{v}_q^{(m)}$

$$so : \mathbf{v}_p^{(m+1)} - \mathbf{v}_q^{(m+1)} = \mathbf{v}_p^{(m)} - \mathbf{v}_q^{(m)} - \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \sigma(\langle \mathbf{w}_p^{(m)}, \mathbf{x} \rangle) + \eta \frac{\partial \widehat{\mathcal{R}}}{\partial \mathbf{f}} \sigma(\langle \mathbf{w}_q^{(m)}, \mathbf{x} \rangle) \quad (28)$$

$$so : \mathbf{v}_p^{(m+1)} = \mathbf{v}_q^{(m+1)} \quad (29)$$

so for any step $t \geq 0$, $\mathbf{v}_p^{(t)} = \mathbf{v}_q^{(t)}$

(c) assume $A = \mathbf{W}^{(0)}\mathbf{x}$, so $\mathbf{A} \in \mathbb{R}^m$, so $\mathbf{A}_1 = w_{1,1}^{(0)}\mathbf{x}_1 + w_{1,2}^{(0)}\mathbf{x}_2 + \cdots + w_{1,d}^{(0)}\mathbf{x}_d$
since $w_{j,k}^{(0)} \sim \mathcal{N}(0, 1/m)$ and linear combinations of Gaussians are again Gaussian
so: $\mathbf{A}_1 \sim \mathcal{N}(0, (\mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2)/m)$
 $\mathbf{A}_2 = w_{2,1}^{(0)}\mathbf{x}_1 + w_{2,2}^{(0)}\mathbf{x}_2 + \cdots + w_{2,d}^{(0)}\mathbf{x}_d$, so: $\mathbf{A}_2 \sim \mathcal{N}(0, (\mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2)/m)$
so: For each $1 \leq i \leq m$, $\mathbf{A}_i \sim \mathcal{N}(0, (\mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2)/m)$
 $\mathbb{E} \left[\left\| \mathbf{W}^{(0)}\mathbf{x} \right\|_2^2 \right] = \mathbb{E}(\mathbf{A}_1^2) + \mathbb{E}(\mathbf{A}_2^2) + \cdots + \mathbb{E}(\mathbf{A}_m^2)$
since $\mathbb{E}(\mathbf{A}_1^2) = \mathbb{E}(\mathbf{A}_1)^2 + D(\mathbf{A}_1) = 0 + (\mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2)/m$
so $\mathbb{E}(\mathbf{A}_1^2) = \mathbb{E}(\mathbf{A}_2^2) = \cdots = \mathbb{E}(\mathbf{A}_m^2) = (\mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2)/m$
so $\mathbb{E} \left[\left\| \mathbf{W}^{(0)}\mathbf{x} \right\|_2^2 \right] = (\mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2) = \|\mathbf{x}\|_2^2$

assume $A = \sigma_r(\mathbf{W}^{(0)}\mathbf{x})$, so $\mathbf{A}_1 = \sigma_r(w_{1,1}^{(0)}\mathbf{x}_1 + w_{1,2}^{(0)}\mathbf{x}_2 + \cdots + w_{1,d}^{(0)}\mathbf{x}_d)$
assume $B_1 = (w_{1,1}^{(0)}\mathbf{x}_1 + w_{1,2}^{(0)}\mathbf{x}_2 + \cdots + w_{1,d}^{(0)}\mathbf{x}_d)$
since $w_{j,k}^{(0)} \sim \mathcal{N}(0, 2/m)$, $\mathbf{B}_1 \sim \mathcal{N}(0, 2(\mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2)/m)$
so: For each $1 \leq i \leq m$, $\mathbf{B}_i \sim \mathcal{N}(0, 2(\mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2)/m)$, $A_i = \sigma_r(\mathbf{B}_i)$
 $\mathbb{E} \left[\left\| \sigma_r(\mathbf{W}^{(0)}\mathbf{x}) \right\|_2^2 \right] = \mathbb{E}(\mathbf{A}_1^2) + \mathbb{E}(\mathbf{A}_2^2) + \cdots + \mathbb{E}(\mathbf{A}_m^2)$
since $\mathbb{E}(\mathbf{B}_i) = 0, P(\mathbf{B}_i \geq 0) = P(\mathbf{B}_i \leq 0) = 1/2$
if $\mathbf{B}_i \geq 0$, $\mathbf{A}_i = \mathbf{B}_i$. if $\mathbf{B}_i \leq 0$, $\mathbf{A}_i = 0$
so $\mathbb{E}(\mathbf{A}_1^2) = \mathbb{E}(P(\mathbf{B}_1 \leq 0)\mathbf{A}_1^2(\mathbf{B}_1 \geq 0) + P(\mathbf{B}_1 \leq 0)\mathbf{A}_1^2(\mathbf{B}_1 \leq 0)) = \mathbb{E}(0 + \frac{1}{2}\mathbf{B}_1^2)$
 $= \frac{1}{2}(\mathbb{E}(\mathbf{B}_1)^2 + D(\mathbf{B}_1)) = \frac{1}{2}(0 + 2(\mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2)/m) = \mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2/m$
so $\mathbb{E}(\mathbf{A}_1^2) = \mathbb{E}(\mathbf{A}_2^2) = \cdots = \mathbb{E}(\mathbf{A}_m^2) = (\mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2)/m$
so $\mathbb{E} \left[\left\| \sigma_r(\mathbf{W}^{(0)}\mathbf{x}) \right\|_2^2 \right] = (\mathbf{x}_1^2 + \mathbf{x}_2^2 + \cdots + \mathbf{x}_d^2) = \|\mathbf{x}\|_2^2$

3. ResNet.

In this problem, you will implement a simplified ResNet. You do not need to change arguments which are not mentioned here (but you of course could try and see what happens).

- (a) Implement a class `Block`, which is a building block of ResNet. It is described in (?) Figure 2. The input to `Block` is of shape (N, C, H, W) , where N denotes the batch size, C denotes the number of channels, and H and W are the height and width of each channel. For each data example \mathbf{x} with shape (C, H, W) , the output of `block` is

$$\text{Block}(\mathbf{x}) = \sigma_r(\mathbf{x} + f(\mathbf{x})),$$

where σ_r denotes the ReLU activation, and $f(\mathbf{x})$ also has shape (C, H, W) and thus can be added to \mathbf{x} . In detail, f contains the following layers.

- i. A `Conv2d` with C input channels, C output channels, kernel size 3, stride 1, padding 1, and no bias term.
- ii. A `BatchNorm2d` with C features.
- iii. A ReLU layer.
- iv. Another `Conv2d` with the same arguments as i above.
- v. Another `BatchNorm2d` with C features.

Because 3×3 kernels and padding 1 are used, the convolutional layers do not change the shape of each channel. Moreover, the number of channels are also kept unchanged. Therefore $f(\mathbf{x})$ does have the same shape as \mathbf{x} .

Additional instructions are given in docstrings in `hw3.py`.

- (b) Explain why a `Conv2d` layer does not need a bias term if it is followed by a `BatchNorm2d` layer.
- (c) Implement a (shallow) `ResNet` consists of the following parts:
- i. A `Conv2d` with 1 input channel, C output channels, kernel size 3, stride 2, padding 1, and no bias term.
 - ii. A `BatchNorm2d` with C features.
 - iii. A ReLU layer.
 - iv. A `MaxPool2d` with kernel size 2.
 - v. A `Block` with C channels.
 - vi. An `AdaptiveAvgPool2d` which for each channel takes the average of all elements.
 - vii. A `Linear` with C inputs and 10 outputs.

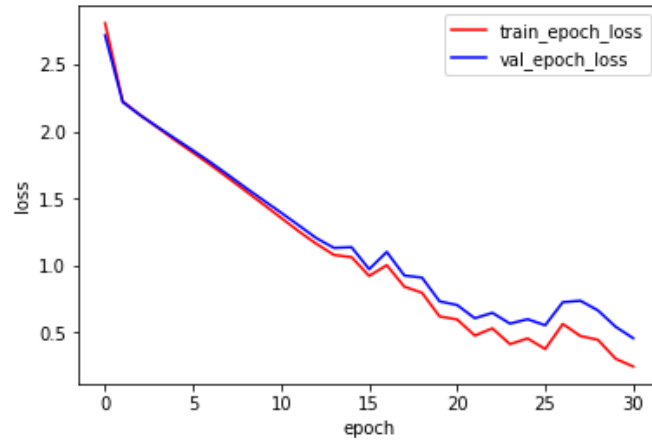
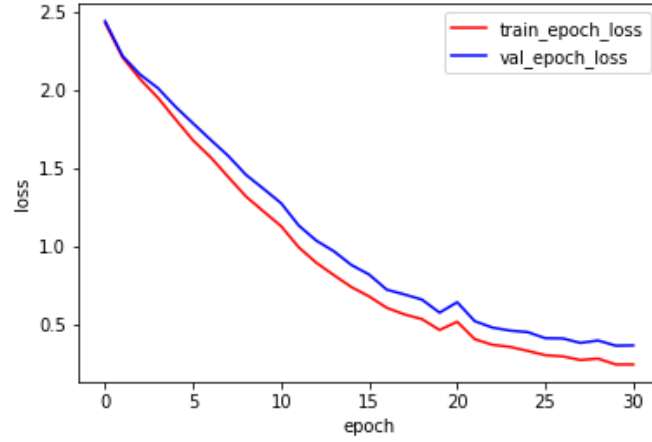
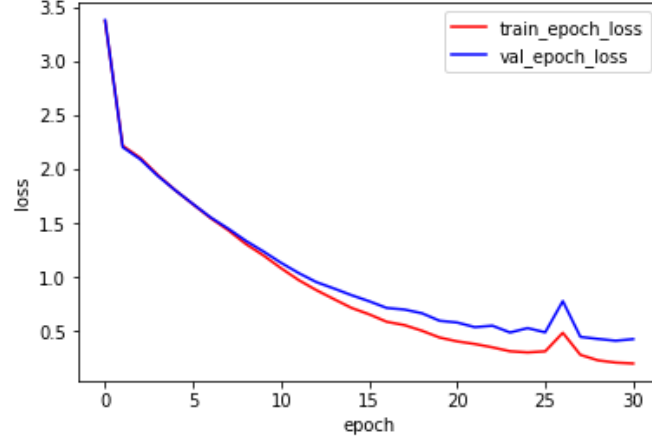
Additional instructions are given in docstrings in `hw3.py`.

- (d) Train a `ResNet` with 16 channels on the data given by `hw3_utils.torch_digits()`, using the cross entropy loss and SGD with learning rate 0.005 and batch size 16, for 30 epochs. You can just use your `fit_and_validate()` in `hw2`. Plot the epochs vs training and validation cross entropy losses. Since there is some inconsistency due to random initialization, try 3 runs and have 3 plots.

Solution. (*Your solution here.*)

- (b) Because the normalize step of Batch normalization ($\hat{x}_i = \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \epsilon}}$) will make the mean and covariance of data in convolution layer be 0 and 1, so the results are same whether we add bias or not. And the last step of Batch normalization is scale and shift ($y_i = \gamma \hat{x}_i + \beta$) which already includes the addition of the bias term.

(d)



4. Kernel properties.

- (a) Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ denote a symmetric positive semi-definite matrix of rank r . Prove that there exists n vectors $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n \in \mathbb{R}^r$, such that $A_{i,j} = \langle \mathbf{z}_i, \mathbf{z}_j \rangle$.

Hint: use the eigendecomposition $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$.

- (b) On the other hand, given n vectors $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n \in \mathbb{R}^m$, prove that the matrix \mathbf{A} defined by $A_{i,j} = \langle \mathbf{z}_i, \mathbf{z}_j \rangle$ is positive semi-definite.

Remark: note that the rank of \mathbf{A} is at most m , and it could be strictly less than m . In particular, m could be larger than n .

- (c) Using (a) and (b), prove that if $K_1(\mathbf{x}, \mathbf{y})$ and $K_2(\mathbf{x}, \mathbf{y})$ are kernels, then $K(\mathbf{x}, \mathbf{y}) = K_1(\mathbf{x}, \mathbf{y})K_2(\mathbf{x}, \mathbf{y})$ is also a kernel.
- (d) Using (c), prove that $K(\mathbf{x}, \mathbf{y}) = (1 + \langle \mathbf{x}, \mathbf{y} \rangle)^r$ is a kernel for any positive integer r . (It is the polynomial kernel with degree r .)
- (e) Assume $K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|_2^2 / 2\sigma^2)$ is a kernel in the 1-dimensional case (i.e., $x, y \in \mathbb{R}$). Using (c), prove that $K(\mathbf{x}, \mathbf{y})$ is indeed a kernel for any dimension. (It is the Gaussian / RBF kernel.)

Solution. (*Your solution here.*)

- (a) since $\mathbf{A} \in \mathbb{R}^{n \times n}$ denote a symmetric positive semi-definite matrix of rank r .

so \mathbf{A} has r eigenvalues which are > 0 and $n-r$ eigenvalues which are $= 0$

so I assume the eigenvalues are ranked by the size of their numbers.

so $\mathbf{\Lambda} = \text{diag}(\Lambda_1, \Lambda_2, \dots, \Lambda_r)$, where $\Lambda_1 \geq \Lambda_2 \geq \dots \geq \Lambda_r > 0$

and $\mathbf{U} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_r]$ where $\mathbf{x}_i (1 \leq i \leq r) \in \mathbb{R}^n$

$$\text{so } \mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top = [\Lambda_1 \mathbf{x}_1 \quad \Lambda_2 \mathbf{x}_2 \quad \dots \quad \Lambda_r \mathbf{x}_r] \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \dots \\ \mathbf{x}_r^\top \end{bmatrix}$$

$$\text{so } A_{i,j} = [\Lambda_1 \mathbf{x}_1(i) \quad \Lambda_2 \mathbf{x}_2(i) \quad \dots \quad \Lambda_r \mathbf{x}_r(i)] \bullet [\mathbf{x}_1(j) \quad \mathbf{x}_2(j) \quad \dots \quad \mathbf{x}_r(j)]$$

$$= [\sqrt{\Lambda_1} \mathbf{x}_1(i) \quad \sqrt{\Lambda_2} \mathbf{x}_2(i) \quad \dots \quad \sqrt{\Lambda_r} \mathbf{x}_r(i)] \bullet [\sqrt{\Lambda_1} \mathbf{x}_1(j) \quad \sqrt{\Lambda_2} \mathbf{x}_2(j) \quad \dots \quad \sqrt{\Lambda_r} \mathbf{x}_r(j)]$$

$$= \langle \mathbf{z}_i, \mathbf{z}_j \rangle$$

$$\text{so for } 1 \leq i \leq n, \mathbf{z}_i = [\sqrt{\Lambda_1} \mathbf{x}_1(i) \quad \sqrt{\Lambda_2} \mathbf{x}_2(i) \quad \dots \quad \sqrt{\Lambda_r} \mathbf{x}_r(i)]$$

(b) since $A_{i,j} = \langle \mathbf{z}_i, \mathbf{z}_j \rangle = A_{j,i}$, so A is a symmetric matrix

$$\text{so } \mathbf{A} = \mathbf{Z}^\top \mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1^\top \\ \mathbf{Z}_2^\top \\ \dots \\ \mathbf{Z}_n^\top \end{bmatrix} \begin{bmatrix} \mathbf{Z}_1 & \mathbf{Z}_2 & \dots & \mathbf{Z}_n \end{bmatrix}, \mathbf{Z} \in \mathbb{R}^{m \times n}$$

so for any non-zero vector $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^\top \mathbf{A} \mathbf{x} = \mathbf{x}^\top \mathbf{Z}^\top \mathbf{Z} \mathbf{x} = (\mathbf{Z} \mathbf{x})^\top \mathbf{Z} \mathbf{x} = (\mathbf{Z} \mathbf{x}) \bullet (\mathbf{Z} \mathbf{x}) \geq 0$

so A is positive semi-definite

(c) since $K_1(\mathbf{x}, \mathbf{y})$ is kernel function, so the Gram matrix A of $K_1(\mathbf{x}, \mathbf{y})$ is a symmetric positive semi-definite matrix

so using the conclusion in part (a), there exists n vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^{r_1}$, such that

$$A_{i,j} = K_1(\mathbf{i}, \mathbf{j}) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle.$$

so for the Gram matrix B of $K_2(\mathbf{x}, \mathbf{y})$, there also exists n vectors $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n \in \mathbb{R}^{r_2}$, such

$$\text{that } B_{i,j} = K_2(\mathbf{i}, \mathbf{j}) = \langle \mathbf{y}_i, \mathbf{y}_j \rangle.$$

so for the Gram matrix C of $K(\mathbf{x}, \mathbf{y})$, $C_{i,j} = K(\mathbf{i}, \mathbf{j}) = K_1(\mathbf{i}, \mathbf{j})K_2(\mathbf{i}, \mathbf{j}) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle \langle \mathbf{y}_i, \mathbf{y}_j \rangle$

$$= [\mathbf{x}_i(1)\mathbf{x}_j(1) + \mathbf{x}_i(2)\mathbf{x}_j(2) + \dots + \mathbf{x}_i(r_1)\mathbf{x}_j(r_1)][\mathbf{y}_i(1)\mathbf{y}_j(1) + \mathbf{y}_i(2)\mathbf{y}_j(2) + \dots + \mathbf{y}_i(r_2)\mathbf{y}_j(r_2)]$$

$$= \mathbf{x}_i(1)\mathbf{y}_i(1)\mathbf{x}_j(1)\mathbf{y}_j(1) + \mathbf{x}_i(2)\mathbf{y}_i(1)\mathbf{x}_j(2)\mathbf{y}_j(1) + \dots + \mathbf{x}_i(r_1)\mathbf{y}_i(1)\mathbf{x}_j(r_1)\mathbf{y}_j(1)$$

$$+ \mathbf{x}_i(1)\mathbf{y}_i(2)\mathbf{x}_j(1)\mathbf{y}_j(2) + \mathbf{x}_i(2)\mathbf{y}_i(2)\mathbf{x}_j(2)\mathbf{y}_j(2) + \dots + \mathbf{x}_i(r_1)\mathbf{y}_i(2)\mathbf{x}_j(r_1)\mathbf{y}_j(2)$$

$$+ \dots$$

$$+ \mathbf{x}_i(1)\mathbf{y}_i(r_2)\mathbf{x}_j(1)\mathbf{y}_j(r_2) + \mathbf{x}_i(2)\mathbf{y}_i(r_2)\mathbf{x}_j(2)\mathbf{y}_j(r_2) + \dots + \mathbf{x}_i(r_1)\mathbf{y}_i(r_2)\mathbf{x}_j(r_1)\mathbf{y}_j(r_2)$$

so there exists n vectors $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n \in \mathbb{R}^{r_1 \times r_2}$

$$\text{where } \mathbf{z}_i(1) = \mathbf{x}_i(1)\mathbf{y}_i(1), \mathbf{z}_i(2) = \mathbf{x}_i(2)\mathbf{y}_i(1), \mathbf{z}_i(3) = \mathbf{x}_i(3)\mathbf{y}_i(1), \dots, \mathbf{z}_i(r_1) = \mathbf{x}_i(r_1)\mathbf{y}_i(1)$$

$$\mathbf{z}_i(r_1+1) = \mathbf{x}_i(1)\mathbf{y}_i(2), \mathbf{z}_i(r_1+2) = \mathbf{x}_i(2)\mathbf{y}_i(2), \mathbf{z}_i(r_1+3) = \mathbf{x}_i(3)\mathbf{y}_i(2), \dots, \mathbf{z}_i(2r_1) = \mathbf{x}_i(r_1)\mathbf{y}_i(2)$$

$$\dots$$

$$\mathbf{z}_i((r_2-1)r_1+1) = \mathbf{x}_i(1)\mathbf{y}_i(r_2), \mathbf{z}_i((r_2-1)r_1+2) = \mathbf{x}_i(2)\mathbf{y}_i(r_2), \mathbf{z}_i((r_2-1)r_1+3) = \mathbf{x}_i(3)\mathbf{y}_i(r_2), \dots, \mathbf{z}_i(r_1r_2) = \mathbf{x}_i(r_1)\mathbf{y}_i(r_2)$$

(d) the Gram matrix A of $K_1(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$ is a symmetric positive semi-definite matrix since $K_1(\mathbf{x}, \mathbf{y})$

is kernel function using (b).

so for any non-zero vector \mathbf{x} , $\mathbf{x}^\top \mathbf{A} \mathbf{x} \geq 0$.

the Gram matrix B of $K_2(\mathbf{x}, \mathbf{y}) = 1$ is obviously a symmetric positive semi-definite matrix.

so for any non-zero vector \mathbf{x} , $\mathbf{x}^\top \mathbf{B} \mathbf{x} \geq 0$.

so the Gram matrix C of $K_3(\mathbf{x}, \mathbf{y}) = 1 + \langle \mathbf{x}, \mathbf{y} \rangle$ is equal to $A + B$

since for any non-zero vector \mathbf{x} , $\mathbf{x}^\top \mathbf{C} \mathbf{x} = \mathbf{x}^\top (\mathbf{A} + \mathbf{B}) \mathbf{x} \geq 0$

so C is a symmetric positive semi-definite matrix

so $K_3(\mathbf{x}, \mathbf{y}) = 1 + \langle \mathbf{x}, \mathbf{y} \rangle$ is kernel function

for $K(\mathbf{x}, \mathbf{y}) = (1 + \langle \mathbf{x}, \mathbf{y} \rangle)^r$, using mathematical induction:

for $r = 1$, $K(\mathbf{x}, \mathbf{y}) = 1 + \langle \mathbf{x}, \mathbf{y} \rangle$ is a kernel function

if $K(\mathbf{x}, \mathbf{y}) = (1 + \langle \mathbf{x}, \mathbf{y} \rangle)^n$ is a kernel function:

$K(\mathbf{x}, \mathbf{y}) = (1 + \langle \mathbf{x}, \mathbf{y} \rangle)^{n+1} = (1 + \langle \mathbf{x}, \mathbf{y} \rangle)(1 + \langle \mathbf{x}, \mathbf{y} \rangle)^n$ is also a kernel function

so $K(\mathbf{x}, \mathbf{y}) = (1 + \langle \mathbf{x}, \mathbf{y} \rangle)^r$ is a kernel for any positive integer r

(e) for any positive integer r , $x, y \in \mathbb{R}^r$, using mathematical induction:

for $r = 1$, $K_1(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|_2^2 / 2\sigma^2)$ is a kernel function

if $x, y \in \mathbb{R}^n$, $K_n(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|_2^2 / 2\sigma^2) = \exp([-(x(1) - y(1))^2 - (x(2) - y(2))^2 - \dots - (x(n) - y(n))^2] / 2\sigma^2)$ is a kernel function:

so for $x, y \in \mathbb{R}^{n+1}$:

$$\begin{aligned} K_{n+1}(\mathbf{x}, \mathbf{y}) &= \exp(-\|\mathbf{x} - \mathbf{y}\|_2^2 / 2\sigma^2) \\ &= \exp([-(x(1) - y(1))^2 - (x(2) - y(2))^2 - \dots - (x(n) - y(n))^2 - (x(n+1) - y(n+1))^2] / 2\sigma^2) \\ &= \exp([-(x(1) - y(1))^2 - (x(2) - y(2))^2 - \dots - (x(n) - y(n))^2] / 2\sigma^2) \times \exp(-(x(n+1) - y(n+1))^2 / 2\sigma^2) \\ &= K_n(\mathbf{x}, \mathbf{y}) K_1(\mathbf{x}, \mathbf{y}) \end{aligned}$$

so $K_{n+1}(\mathbf{x}, \mathbf{y})$ is also a kernel function

so for any positive integer r , $x, y \in \mathbb{R}^r$, $K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|_2^2 / 2\sigma^2)$ is a kernel function

5. RBF kernel and nearest neighbors.

- (a) Recall that given data examples $((\mathbf{x}_i, y_i))_{i=1}^n$ and an optimal dual solution $(\hat{\alpha}_i)_{i=1}^n$, the RBF kernel SVM makes a prediction as follows:

$$f_\sigma(\mathbf{x}) = \sum_{i=1}^n \hat{\alpha}_i y_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|_2^2}{2\sigma^2}\right) = \sum_{i \in S} \hat{\alpha}_i y_i \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|_2^2}{2\sigma^2}\right),$$

where $S \subset \{1, 2, \dots, n\}$ is the set of indices of support vectors.

Given an input \mathbf{x} , let $T := \arg \min_{i \in S} \|\mathbf{x} - \mathbf{x}_i\|_2$ denote the set of closest support vectors to \mathbf{x} , and let $\rho := \min_{i \in S} \|\mathbf{x} - \mathbf{x}_i\|_2$ denote this smallest distance. (In other words, $T := \{i \in S : \|\mathbf{x} - \mathbf{x}_i\| = \rho\}$.) Prove that

$$\lim_{\sigma \rightarrow 0} \frac{f_\sigma(\mathbf{x})}{\exp(-\rho^2/2\sigma^2)} = \sum_{i \in T} \hat{\alpha}_i y_i.$$

Remark: in other words, when the bandwidth σ becomes small enough, RBF kernel SVM is almost the 1-nearest neighbor predictor with the set of support vectors as the training set.

- (b) Consider the XOR dataset:

$$\begin{aligned} \mathbf{x}_1 &= (+1, +1), & y_1 &= +1, \\ \mathbf{x}_2 &= (-1, +1), & y_2 &= -1, \\ \mathbf{x}_3 &= (-1, -1), & y_3 &= +1, \\ \mathbf{x}_4 &= (+1, -1), & y_4 &= -1. \end{aligned}$$

Verify that $\hat{\alpha} = (1/\alpha, 1/\alpha, 1/\alpha, 1/\alpha)$ is an optimal dual solution to the RBF kernel SVM, where

$$\alpha = \left(1 - \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2}{2\sigma^2}\right)\right)^2 = \left(1 - \exp\left(-\frac{2}{\sigma^2}\right)\right)^2 > 0.$$

Hint: prove that the gradient of the dual function is $\mathbf{0}$ at $\hat{\alpha}$. Since the dual function is concave, and $\hat{\alpha} > \mathbf{0}$, it follows that $\hat{\alpha}$ is an optimal dual solution.

Remark: in other words, all four data examples are mapped to support vectors in the reproducing kernel Hilbert space. In light of (a), when σ is small enough, $f_\sigma(\mathbf{x})$ is almost the 1-nearest neighbor predictor on the XOR dataset. In fact, it is also true for large σ , due to the symmetry of the XOR data.

Solution. (Your solution here.)

- (a)

$$\lim_{\sigma \rightarrow 0} \frac{f_\sigma(\mathbf{x})}{\exp(-\rho^2/2\sigma^2)} = \lim_{\sigma \rightarrow 0} \sum_{i \in S} \hat{\alpha}_i y_i \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}_i\|_2^2 + \rho^2}{2\sigma^2}\right) \quad (30)$$

If $i \notin T$, $\|\mathbf{x} - \mathbf{x}_i\|_2 > \rho$, so $(-\|\mathbf{x} - \mathbf{x}_i\|_2^2 + \rho^2) < 0$

So when $\sigma \rightarrow 0$,

$$\frac{-\|\mathbf{x} - \mathbf{x}_i\|_2^2 + \rho^2}{2\sigma^2} \rightarrow -\infty, \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}_i\|_2^2 + \rho^2}{2\sigma^2}\right) \rightarrow 0 \quad (31)$$

if $i \in \mathbf{T}$, $\|\mathbf{x} - \mathbf{x}_i\|_2 = \rho$, so $(-\|\mathbf{x} - \mathbf{x}_i\|_2^2 + \rho^2) = 0$

$$\text{So } \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}_i\|_2^2 + \rho^2}{2\sigma^2}\right) = 1 \quad (32)$$

$$\text{So } \lim_{\sigma \rightarrow 0} \frac{f_\sigma(\mathbf{x})}{\exp(-\rho^2/2\sigma^2)} = \sum_{i \in T} \hat{\alpha}_i y_i.$$

(b)

$$f(\alpha) = \sum_{i=1}^4 \alpha_i - \frac{1}{2} \sum_{i,j=1}^4 \alpha_i \alpha_j \mathbf{y}_i \mathbf{y}_j \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) \quad \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right) \quad (33)$$

$$\frac{\partial f(\alpha)}{\partial \alpha_1} = 1 - \frac{1}{2} (2\alpha_1 \mathbf{y}_1^2 \mathbf{K}(\mathbf{x}_1, \mathbf{x}_1) + \alpha_2 \mathbf{y}_1 \mathbf{y}_2 \mathbf{K}(\mathbf{x}_1, \mathbf{x}_2) + \alpha_3 \mathbf{y}_1 \mathbf{y}_3 \mathbf{K}(\mathbf{x}_1, \mathbf{x}_3) + \alpha_4 \mathbf{y}_1 \mathbf{y}_4 \mathbf{K}(\mathbf{x}_1, \mathbf{x}_4) + \quad (34)$$

$$\alpha_2 \mathbf{y}_2 \mathbf{y}_1 \mathbf{K}(\mathbf{x}_2, \mathbf{x}_1) + \alpha_3 \mathbf{y}_3 \mathbf{y}_1 \mathbf{K}(\mathbf{x}_3, \mathbf{x}_1) + \alpha_4 \mathbf{y}_4 \mathbf{y}_1 \mathbf{K}(\mathbf{x}_4, \mathbf{x}_1)) \quad (35)$$

$$= 1 - \frac{1}{2} (2\alpha_1 \mathbf{y}_1^2 \mathbf{K}(\mathbf{x}_1, \mathbf{x}_1) + 2\alpha_2 \mathbf{y}_1 \mathbf{y}_2 \mathbf{K}(\mathbf{x}_1, \mathbf{x}_2) + 2\alpha_3 \mathbf{y}_1 \mathbf{y}_3 \mathbf{K}(\mathbf{x}_1, \mathbf{x}_3) + 2\alpha_4 \mathbf{y}_1 \mathbf{y}_4 \mathbf{K}(\mathbf{x}_1, \mathbf{x}_4)) \quad (36)$$

$$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 1/\alpha, \mathbf{K}(\mathbf{x}_1, \mathbf{x}_1) = 1, \mathbf{K}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{K}(\mathbf{x}_1, \mathbf{x}_4) = \exp\left(-\frac{2}{\sigma^2}\right), \mathbf{K}(\mathbf{x}_1, \mathbf{x}_3) = \exp\left(-\frac{4}{\sigma^2}\right) \quad (37)$$

$$\text{so } \frac{\partial f(\alpha)}{\partial \alpha_1} = 1 - \left(\frac{1 - 2 \times \exp\left(-\frac{2}{\sigma^2}\right) + \exp\left(-\frac{4}{\sigma^2}\right)}{\left(1 - \exp\left(-\frac{2}{\sigma^2}\right)\right)^2} \right) = 1 - 1 = 0 \quad (38)$$

So in the same way:

$$\frac{\partial f(\alpha)}{\partial \alpha_2} = \frac{\partial f(\alpha)}{\partial \alpha_3} = \frac{\partial f(\alpha)}{\partial \alpha_4} = 0 \quad (39)$$

So the gradient of the dual function is $\mathbf{0}$ at $\hat{\alpha}$.

Since the dual function is concave, and $\hat{\alpha} > \mathbf{0}$, so $\hat{\alpha} = (1/\alpha, 1/\alpha, 1/\alpha, 1/\alpha)$ is an optimal

dual solution to the RBF kernel SVM

6. SVM implementation.

Recall that the dual problem of SVM is

$$\max_{\boldsymbol{\alpha} \in \mathcal{C}} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j).$$

where the domain $\mathcal{C} = [0, \infty)^n = \{\boldsymbol{\alpha} : \alpha_i \geq 0\}$ for hard-margin SVM, and $\mathcal{C} = [0, C]^n = \{\boldsymbol{\alpha} : 0 \leq \alpha_i \leq C\}$ for soft-margin SVM.

Equivalently, it can be formulated as a minimization problem

$$\min_{\boldsymbol{\alpha} \in \mathcal{C}} f(\boldsymbol{\alpha}) := \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^n \alpha_i.$$

It can be solved by projected gradient descent, which starts from some $\boldsymbol{\alpha}_0 \in \mathcal{C}$ (e.g., $\mathbf{0}$) and updates as follows

$$\boldsymbol{\alpha}_{t+1} = \Pi_{\mathcal{C}} [\boldsymbol{\alpha}_t - \eta \nabla f(\boldsymbol{\alpha}_t)].$$

Here $\Pi_{\mathcal{C}}[\boldsymbol{\alpha}]$ is the *projection* of $\boldsymbol{\alpha}$ onto \mathcal{C} , defined as the closet point to $\boldsymbol{\alpha}$ in \mathcal{C} :

$$\Pi_{\mathcal{C}}[\boldsymbol{\alpha}] := \arg \min_{\boldsymbol{\alpha}' \in \mathcal{C}} \|\boldsymbol{\alpha}' - \boldsymbol{\alpha}\|_2.$$

If \mathcal{C} is convex, the projection is uniquely defined.

(a) Prove that

$$\left(\Pi_{[0, \infty)^n}[\boldsymbol{\alpha}] \right)_i = \max\{\alpha_i, 0\},$$

and

$$\left(\Pi_{[0, C]^n}[\boldsymbol{\alpha}] \right)_i = \min\{\max\{0, \alpha_i\}, C\}.$$

- (b) Implement an `svm_solver()`, using projected gradient descent formulated as above. See the docstrings in `hw3.py` for details.
- (c) Implement an `svm_predictor()`, using an optimal dual solution, the training set, and an input. See the docstrings in `hw3.py` for details.
- (d) On the area $[-5, 5] \times [-5, 5]$, plot the contour lines of the following kernel SVMs, trained on the XOR data. Different kernels and the XOR data are provided in `hw3_utils.py`. Learning rate 0.1 and 10000 steps should be enough. To draw the contour lines, you can use `hw3.svm_contour()`.
- The polynomial kernel with degree 2.
 - The RBF kernel with $\sigma = 1$.
 - The RBF kernel with $\sigma = 2$.
 - The RBF kernel with $\sigma = 4$.

Solution. (*Your solution here.*)

(a)

$$\Pi_{[0,\infty)^n}[\alpha] := \arg \min_{\alpha' \in [0,\infty)^n} \|\alpha' - \alpha\|_2 = \arg \min_{\alpha' \in [0,\infty)^n} \sqrt{(\alpha'_1 - \alpha_1)^2 + (\alpha'_2 - \alpha_2)^2 + \cdots + (\alpha'_n - \alpha_n)^2}$$

Since $\alpha' \in [0, \infty)^n$, so for $1 \leq i \leq n$:

If $\alpha_i \geq 0$, $\min((\alpha'_i - \alpha_i)^2) = 0$, when $\alpha'_i = \alpha_i$

If $\alpha_i < 0$, $\min((\alpha'_i - \alpha_i)^2) = \alpha_i^2$, when $\alpha'_i = 0$

$$\text{So } \left(\Pi_{[0,\infty)^n}[\alpha] \right)_i = \max\{\alpha_i, 0\},$$

$$\Pi_{[0,C]^n}[\alpha] := \arg \min_{\alpha' \in [0,C]^n} \|\alpha' - \alpha\|_2 = \arg \min_{\alpha' \in [0,C]^n} \sqrt{(\alpha'_1 - \alpha_1)^2 + (\alpha'_2 - \alpha_2)^2 + \cdots + (\alpha'_n - \alpha_n)^2}$$

Since $\alpha' \in [0, C]^n$, so for $1 \leq i \leq n$:

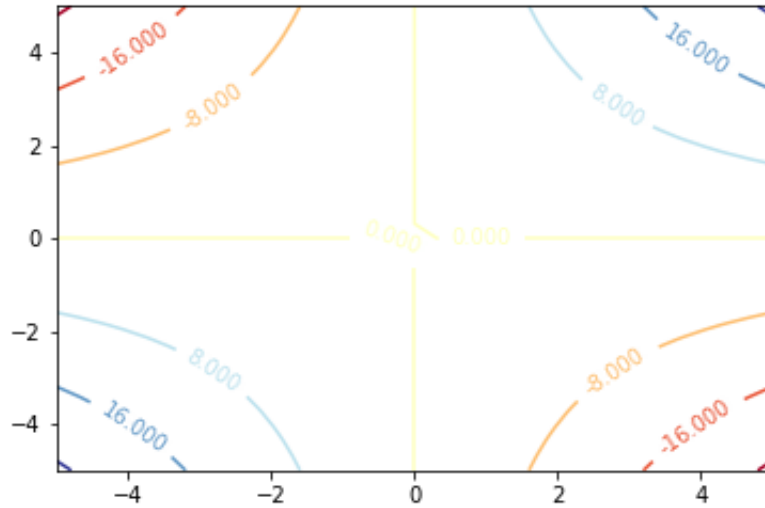
If $0 \leq \alpha_i \leq C$, $\min((\alpha'_i - \alpha_i)^2) = 0$, when $\alpha'_i = \alpha_i$

If $\alpha_i < 0$, $\min((\alpha'_i - \alpha_i)^2) = \alpha_i^2$, when $\alpha'_i = 0$

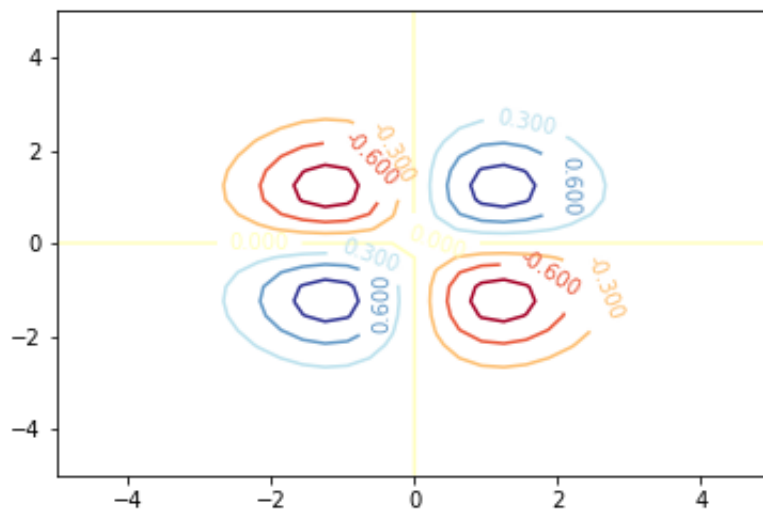
If $\alpha_i > C$, $\min((\alpha'_i - \alpha_i)^2) = \min((\alpha_i - \alpha'_i)^2) = (\alpha_i - C)^2$, when $\alpha'_i = C$

$$\text{So } \left(\Pi_{[0,C]^n}[\alpha] \right)_i = \min\{\max\{0, \alpha_i\}, C\}.$$

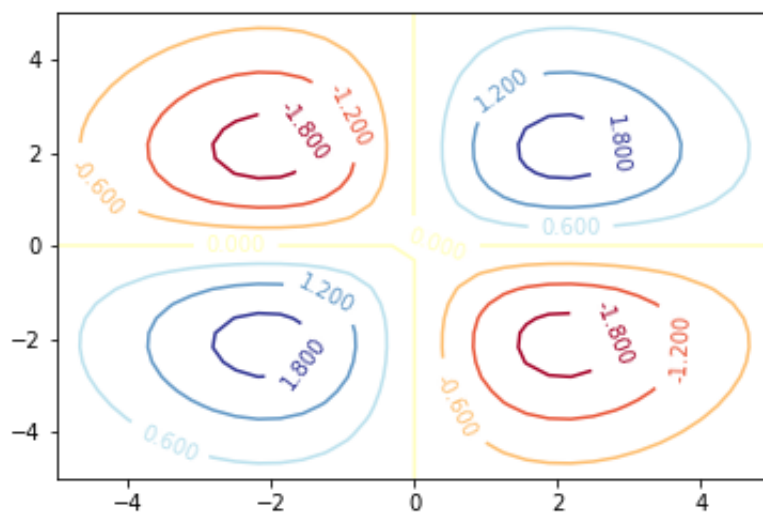
(d) The polynomial kernel with degree 2:



The RBF kernel with $\sigma = 1$:



The RBF kernel with $\sigma = 2$:



The RBF kernel with $\sigma = 4$:

