Due Date: February 28th 2025, 23:00

Instructions

- *For all questions that are not graded only on the answer, show your work! Any problem without work shown will get no marks regardless of the correctness of the final answer.*
- *Please try to use a document preparation system such as LaTeX.* ***If you write your answers by hand, note that you risk losing marks if your writing is illegible without any possibility of regrade, at the discretion of the grader.***
- *Submit your answers electronically via the course GradeScope.* ***Incorrectly assigned answers can be given 0 automatically at the discretion of the grader.*** *To assign answers properly, please*
  - *Make sure that the top of the first assigned page is the question being graded.*
  - *Do not include any part of answer to any other questions within the assigned pages.*
  - *Assigned pages need to be placed in order.*
  - *For questions with multiple parts, the answers should be written in order of the parts within the question.*
- ***Questions requiring written responses should be short and concise when necessary.*** *Unnecessary wordiness will be penalized at the grader's discretion.*
- *Please sign the agreement below.*
- *It is your responsibility to follow updates to the assignment after release. All changes will be visible on Overleaf and Piazza.*
- *Any questions should be directed towards the TA for this assignment (theoretical part):* ***Jerry Huang****.*

I acknowledge I have read the above instructions and will abide by them throughout this assignment. I further acknowledge that any assignment submitted without the following form completed will result in no marks being given for this portion of the assignment.

Signature: _____

Name: Félix Wilhelmy_____

UdeM Student ID: 20333575_____

Basics (10 points)

Question 1. For each of the following sub-questions, provide an answer in the given space. You are only marked on correctness. **Please highlight your response.** Long or wordy explanations will be penalized.

For this question, you will need to make use of the following definitions

Definition 1 (Convex Function)**.** A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if for any $\alpha \in (0, 1)$ and $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$, then

$$f(\alpha \mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2) \leq \alpha f(\mathbf{x}_1) + (1 - \alpha) f(\mathbf{x}_2)$$

Definition 2 (Lipschitzness)**.** A function $f : \mathbb{R}^d \to \mathbb{R}^n$ is ρ-Lipschitz over a set $\mathbb{S} \subset \mathbb{R}^d$ if for any $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{S}$

$$\|f(\mathbf{w}_1) - f(\mathbf{w}_2)\| \leq \rho \|\mathbf{w}_1 - \mathbf{w}_2\|$$

Definition 3 (Smoothness)**.** A function f is β-smooth if it's gradient is β-Lipschitz.

1. **(3 points)** Answer **True**, **False** or **Sometimes True** for the following questions. If **Sometimes True**, provide a brief reason (no more than 1 sentence).
   (a) The L_2 norm function on the $\mathbb{R}^d$ space is not convex: **False**
   (b) If f is a convex function, then its second derivative is non-negative at any point where it is continuous (assuming that f is twice differentiable): **True**
   (c) If f and g are both convex functions, then their combination $f + g$ is also convex: **True**
2. **(2 points)** Fill in the blanks:
   (a) If $f : \mathbb{R} \to \mathbb{R}$ is convex and $a, b, x \in \text{dom}(f)$ with $a < x < b$, then

   $$f(x) \quad \leq \quad \frac{\quad}{b-a} f(a) + \frac{\quad}{b-a} f(b)$$

   (b) A continuous function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if for every $x, y \in \mathbb{R}^n$

   $$\int_0^1 f(x + \lambda(y - x)) d\lambda \leq \quad \underline{\quad\quad}$$

3. **(2 points)** Answer **True**, **False**:
   (a) $f(x) = x^2$ is 5-Lipshitz over $\mathbb{R}$: _____
   (b) $f(x) = \log(1 + \exp(x))$ is 1-Lipshitz over $\mathbb{R}$ is: _____
4. **(2 points)** The maximum β for which the following are still smooth over $\mathbb{R}$ are:
   (a) $f(x) = x^2$: _____
   (b) $f(x) = \log(1 + \exp(x))$: _____
5. **(1 point)** Answer **True**, **False**:
   (a) $\exists \beta : f$ is β-smooth $\implies \exists \alpha : f$ is α-Lipschitz:_____

Optimizers (6 points)

Question 2. Suppose we have the following objective function

$$f(x) = \frac{1}{2}\left(x_1^2 + cx_2^2\right)$$

where $c > 0$.

1. **(1 point)** What is the optimal point?
2. **(2 points)** Suppose we apply Newton's method here. Derive the following closed-form expressions for the iterates $x^{(k)}$ and the values $f\left(x^{(k)}\right)$.
3. **(1 point)** How does the value converge? Say in terms of how fast the objective function value changes.
4. **(1 point)** What can you say about convergence when $c = 1$?
5. **(1 point)** Suppose you are at a starting point of $(1, 0)$. For what learning rates will gradient descent converge?

Convolutional Neural Networks (9 points)

Question 3 (4-2-3)**.** The following questions each deal with convolutional neural networks (CNNs).

For parts (1) and (3) of this question, you will lose 0.5 marks for every 2 numbers that are incorrect.

1. **(4 points)** Given the CNN defined by the layers in the below, fill in the shape of the output volume and the number of parameters at each layer in the following table (Table 3). Write the shapes in the format (H, W, C), where H, W, C are the height, width and channel dimensions. Unless specified, assume padding 1 and stride 1 where appropriate. For MaxPooling layers, use no padding and a stride of 2. Layers are expressed as follows:
   (a) CONV(H,C): A convolution layer with filters of size $H \times H$ with C channels with appropriate weights and biases.
   (b) RELU: ReLU activation.
   (c) MAXPOOLING(N): $N \times N$ max pooling.
   (d) FC(D): A fully connected layer with D outputs with appropriate weights and biases.
   (e) BATCHNORM: Batch normalization.
   (f) RESHAPE: A flattening layer.

Layer	Output Dimensions	Number of Parameters
INPUT	$32 \times 32 \times 3$	0
CONV(3, 12)		
BATCHNORM		
RELU		
MAXPOOLING(2)		
CONV(3, 8)		
BATCHNORM		
RELU		
MAXPOOLING(2)		
RESHAPE		
FC(10)		

Table 1: Table to fill out.

2. **(2 points)** In the above setup, how many parameters can I remove and still keep the output the same? **Explain why.**

3. **(3 points)** Fill out the following table (Table 4) by finding the stride of each relevant layer. You can assume that the input size is sufficient to accommodate the provided receptive field sizes, as well as no padding.

Layer	Receptive Field
Conv(__,1) with stride **3**	6
MaxPooling(__) with stride __	9
Conv(__,1) with stride __	15
MaxPooling(__) with stride __	27
Conv(__,1) with stride __	45

Table 2: Table to fill out for part 3.

Empirical Risk Minimization (25 points)

Question 4 (8-9-8)**.** In this section, we will discuss the notion of Empirical Risk Minimization. Given a probability distribution over an input and a label space $\mathcal{X} \times \mathcal{Y}$, **risk** can be defined as

$$\mathcal{R}(g) = \mathop{\mathbb{E}}_{\boldsymbol{x},y\sim p(\mathcal{X},\mathcal{Y})} \left[\ell(g(\boldsymbol{x}), y)\right],$$

where ℓ is a loss function and g is a function from the input space to the label space. For these problems, you should assume that g outputs a probability distribution over the labels. However, as is standard in many machine learning problems, the true distribution of the data is often unknown and therefore we rely on a training set $\mathcal{D}^{\text{tr}}$. Risk on the training set is defined as the **empirical risk**

$$\widehat{\mathcal{R}}(g) = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\in\mathcal{D}^{\text{tr}}} \left[\ell(g(\boldsymbol{x}), y)\right].$$

For the rest of this question, we will look at various problems with empirical risk and how it approximates the true risk.

1. **(8 points)** In this part, we'll look at an interesting setting where the label for an example $\boldsymbol{x}$ is not given. Let us stick to a binary classification setting. Instead of being provided the true label y for any example $\boldsymbol{x}$, for each example, we are instead provided a confidence c, which we define as

   $$c(\boldsymbol{x}) = p(y = +1 \mid \boldsymbol{x}) \neq 0.$$

   Assume that $c(\boldsymbol{x}) \neq 0$ for all $\boldsymbol{x}$. Since we no longer have the label provided, we need to redefine how we estimate risk. In each of the following questions, suppose that the true label y for an example $\boldsymbol{x}$ exists, but we cannot access it. Further suppose that $y \in \{\pm 1\}$.
   Note: If you are having trouble with the first two parts, please try to do the third part. It might help you get a better idea of what to do

   (a) **(4 points)** What should our risk be in this case?

   (b) **(2 points)** Given the risk we defined, how should empirical risk be defined here?

   (c) **(2 points)** Why is the following training objective function

   $$L = \sum_{i=1}^{N} \left[c_i \cdot \ell(g(\boldsymbol{x}_i), y_i) + (1 - c_i) \cdot \ell(-g(\boldsymbol{x}_i), y_i)\right],$$

   not an appropriate loss to minimize?

2. **(9 points)** In this part, we'll consider a case where we get $\boldsymbol{x}$ without y, hence we're working with complete unlabeled data.

   (a) **(4 points)** Suppose that we have a data distribution with binary classes again. For simplicity, assume that the positive and negative examples within the data distribution do not overlap. Now suppose that we have a training set, sampled as follows:

   $$p^{\text{tr}} = \theta \cdot p_+(\boldsymbol{x}) + (1 - \theta) \cdot p_-(\boldsymbol{x}),$$

where θ is a constant in $[0, 1]$ and

$$p_+(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = +1)$$
$$p_-(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = -1).$$

In this specific setting, you can use the 01-loss.

Show that with this training set of data, it isn't always possible to estimate the true risk of a predictor g, $\mathcal{R}(g)$.

(b) **(4 points)** Suppose we're still working in the same setting. Now let's assume that we now have a second training set, sampled in the same manner as the first, but with $\theta' \neq \theta$. We can actually now estimate the true risk of g even without any labels. Show how to do so.

3. **(8 points)** Let's return to the standard binary classification problem where we are given pairs of $(\boldsymbol{x}, y)$. Suppose you are using the empirical risk as the objective to minimize again.

(a) **(3 points)** Your friend tells you that instead of minimizing $\widehat{\mathcal{R}}(g)$, he/she minimizes

$$\widetilde{\mathcal{R}}(g) = \left|\widehat{\mathcal{R}}(g) - \varepsilon\right| + \varepsilon,$$

where ε is some constant value. Can you explain what they are attempting to do and how it works?

Hint: Try to provide an answer in terms of how they are minimizing the training or testing loss and why it might be beneficial.

(b) **(4 points)** Your friend tells you their objective is better than yours. Can you justify their claim?

Hint: Try to use the mean-square error MSE between the risk estimators being used, $\widehat{\mathcal{R}}(g)$ and $\widetilde{\mathcal{R}}(g)$, and the true risk $\mathcal{R}(g)$.

(c) **(2 points)** Your friend claims their objective can be even more effective if ε is chosen more carefully. Can you explain how?

Hint: Use your previous result. Your answer here should not exceed 100 characters. **Longer answers will not be given points.**

1 Question 1

Question 1.1.a False

Question 1.1.b True

Question 1.1.c True

Question 1.2.a

$$f(x) \quad \leq \quad \frac{\boldsymbol{b}-\boldsymbol{x}}{b-a}f(a)+\frac{\boldsymbol{x}-\boldsymbol{a}}{b-a}f(b)$$

Question 1.2.b

$$\int_0^1 f(x+\lambda(y-x))d\lambda \leq \quad \underline{\frac{1}{2}(f(x)+f(y))}$$

Question 1.3.a False

Question 1.3.b True

Question 1.4.a 2

Question 1.4.b $\frac{1}{4}$

Question 1.5 False

2 Question 2

Question 2.1 TODO

Since $f(\boldsymbol{x})$ is a quadratic function (and hence convex), its unique global minimum can be found by setting its gradient equal to zero.

The gradient of $f(\boldsymbol{x})$ with respect to $\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ is

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} x_1 \\ c\, x_2 \end{pmatrix}.$$

Since $c > 0$, we get the unique optimal point (or global minimum) of $f(\boldsymbol{x})$

$$\boldsymbol{x}^* = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Question 2.2 TODO

Let's consider the objective function

$$f(\boldsymbol{x}) = \frac{1}{2}\left(x_1^2 + c\, x_2^2\right), \text{ with } c > 0$$

The gradient of this function (as shown in the previous question) is

$$\nabla f(\boldsymbol{x}) = \begin{pmatrix} x_1 \\ c\, x_2 \end{pmatrix},$$

It's **Hessian** H is

$$H = \nabla^2 f(\boldsymbol{x}) = \begin{pmatrix} 1 & 0 \\ 0 & c \end{pmatrix}.$$

Newton's method updates the iterate $\boldsymbol{x}^{(k)}$ using the formula

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - H^{-1}\nabla f(\boldsymbol{x}^{(k)}).$$

The inverse of the Hessian H is

$$H^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{c} \end{pmatrix}.$$

Thus

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{c} \end{pmatrix} \begin{pmatrix} x_1^{(k)} \\ c\, x_2^{(k)} \end{pmatrix}.$$

Performing the matrix multiplication, we obtain

$$\begin{pmatrix} 1 \cdot x_1^{(k)} + 0 \cdot (c\, x_2^{(k)}) \\ 0 \cdot x_1^{(k)} + \frac{1}{c} \cdot (c\, x_2^{(k)}) \end{pmatrix} = \begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \end{pmatrix}.$$

Therefore, the update rule simplifies to

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k)} = \mathbf{0}.$$

This shows that regardless of the starting point $\boldsymbol{x}^{(0)}$, Newton's method yields

$$\boldsymbol{x}^{(1)} = \mathbf{0},$$

Question 2.3 TODO

Recall the quadratic function

$$f(\boldsymbol{x}) = \frac{1}{2}\left(x_1^2 + c\, x_2^2\right),$$

with $c > 0$. In question 2.2, we derived that regardless of the starting point $\boldsymbol{x}^{(0)}$, the Newton's method yields

$$\boldsymbol{x}^{(1)} = \mathbf{0},$$

This means that for the given quadratic function, the Newton's method converges in a single iteration for any given point.

Question 2.4 TODO

When $c = 1$, our objective function becomes

$$f(\boldsymbol{x}) = \frac{1}{2}\left(x_1^2 + x_2^2\right).$$

Its Hessian is the identity matrix

$$H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

Since the Hessian is the identity matrix, the optimal learning rate will be 1, and the uniform curvature (condition number 1) will ensure rapid convergence.

Question 2.5 TODO

Consider the quadratic function and it's Hessian

$$f(\boldsymbol{x}) = \frac{1}{2}\left(x_1^2 + c\,x_2^2\right), \text{ with } c > 0 \text{ and } H = \begin{pmatrix} 1 & 0 \\ 0 & c \end{pmatrix}.$$

In gradient descent, to make sure we don't take too large a step (which might make us overshoot the minimum), we need the learning rate α to be smaller than a certain value. In order for the gradient descent to converge on a quadratic function, the step size must satisfy

$$0 < \alpha < \frac{2}{\lambda_{max}}.$$

Where λ_{max} is the maximum eigenvalue of the Hessian of our function. The eigenvalues of H are $\lambda_1 = 1$ and $\lambda_2 = c$. Thus we can advance that the gradient descent will converge for a learning rate that satisfy this condition

$$0 < \alpha < \frac{2}{\max\{1, c\}}.$$

This is true for any point and being on $\boldsymbol{x}^{(0)} = (1, 0)$ does not change that.

3 Question 3

Question 3.1

Layer	Output Dimensions	Number of Parameters
Input	$32 \times 32 \times 3$	0
Conv(3, 12)	$32 \times 32 \times 12$	336
BatchNorm	$32 \times 32 \times 12$	24
ReLU	$32 \times 32 \times 12$	0
MaxPooling(2)	$16 \times 16 \times 12$	0
Conv(3, 8)	$16 \times 16 \times 8$	872
BatchNorm	$16 \times 16 \times 8$	16
ReLU	$16 \times 16 \times 8$	0
MaxPooling(2)	$8 \times 8 \times 8$	0
Reshape	$512 \times 1 \times 1$	0
FC(10)	$10 \times 1 \times 1$	5130

Table 3: Table to fill out.

Question 3.2 TODO

You can remove the bias from the two convolution layers because the subsequent batch normalisation removes the bias (offsets) added by each kernel during normalisation. Therefore, the bias are redundant parameters that could be removed from the model without affecting the final output.

If we count the bias for each convolution, we end up with a total of **20 parameters** (12 for the first convolution and 8 for the second) that are redundant and could be removed from the model.

Layer	Receptive Field
Conv($\underline{3}$,1) with stride $\underline{\mathbf{3}}$	6
MaxPooling($\underline{2}$) with stride $\underline{2}$	9
Conv($\underline{2}$,1) with stride $\underline{1}$	15
MaxPooling($\underline{3}$) with stride $\underline{3}$	27
Conv($\underline{2}$,1) with stride $\underline{1}$	45

Table 4: Table to fill out for part 3.

Question 3.3

4 Question 4

Question 4.1.a TODO

In this setting, we consider a binary classification problem where a true label $y \in \{+1, -1\}$ exists for every input $\boldsymbol{x}$, but we do not have access to y. Instead, we are provided with a confidence score defined as

$$c(\boldsymbol{x}) = p(y = +1 \mid \boldsymbol{x}) \neq 0$$

Since the sum of the probabilities of all outcomes must equal 1, we have

$$1 - c(\boldsymbol{x}) = p(y = -1 \mid \boldsymbol{x})$$

By applying the *Law of Total Expectation*, we can decompose the risk into an outer expectation over inputs $\boldsymbol{x}$ drawn from $p(\boldsymbol{x})$ and an inner expectation over labels y conditioned on $\boldsymbol{x}$. In the *discrete case*, this inner expectation can further be written as a summation over all possible values of $y \in \mathcal{Y}$

$$\mathcal{R}(g) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\Big[\mathop{\mathbb{E}}_{y\sim p(y|\boldsymbol{x})}\big[\ell\big(g(\boldsymbol{x}), y\big)\big]\Big] = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\Big[\sum_{y\in\mathcal{Y}} p(y \mid \boldsymbol{x})\, \ell\big(g(\boldsymbol{x}), y\big)\Big]$$

In this setting and because y is binary, we obtain

$$\mathop{\mathbb{E}}_{y\sim p(y|\boldsymbol{x})}\big[\ell\big(g(\boldsymbol{x}), y\big)\big] = c(\boldsymbol{x})\, \ell\big(g(\boldsymbol{x}), +1\big) + \big(1 - c(\boldsymbol{x})\big)\, \ell\big(g(\boldsymbol{x}), -1\big).$$

Thus, by substituting back into the expression for $\mathcal{R}(g)$, we obtain the risk in this setting

$$\mathcal{R}(g) = \mathop{\mathbb{E}}_{\boldsymbol{x},y\sim p(\mathcal{X},\mathcal{Y})}\left[c(\boldsymbol{x})\, \ell\big(g(\boldsymbol{x}), +1\big) + \big(1 - c(\boldsymbol{x})\big)\, \ell\big(g(\boldsymbol{x}), -1\big)\right].$$

Question 4.1.b TODO

In this question, instead of the full distribution $p(\boldsymbol{x})$, we want to approximate the expectation with a finite training set

$$\mathcal{D}_{\text{tr}} = \{\boldsymbol{x}_i; c(\boldsymbol{x}_i)\}_{i=1}^{N}.$$

The *empirical risk* is defined as the average loss over the training set. Thus, the empirical risk is given by

$$\widehat{\mathcal{R}}(g) = \frac{1}{N}\sum_{i=1}^{N}\Big[c(\boldsymbol{x}_i)\, \ell\big(g(\boldsymbol{x}_i), +1\big) + \Big(1 - c(\boldsymbol{x}_i)\Big)\, \ell\big(g(\boldsymbol{x}_i), -1\big)\Big].$$

Question 4.1.c TODO

Recall that in our setting we do not have direct access to the true labels $y \in \{+1, -1\}$. Instead, for each input $\boldsymbol{x}$ we are given a confidence score

$$c(\boldsymbol{x}) = p(y = +1 \mid \boldsymbol{x}),$$

The training objective function proposed in the question is

$$L = \sum_{i=1}^{N} \Big[c(\boldsymbol{x}_i)\, \ell\big(g(\boldsymbol{x}_i), y_i\big) + \big(1 - c(\boldsymbol{x}_i)\big)\, \ell\big(-g(\boldsymbol{x}_i), y_i\big) \Big].$$

There are two issues with this proposed function

1. The expression uses the true label y_i. However, in our setting, y_i is not available; only the confidence scores $c(\boldsymbol{x}_i)$ can be observed. Therefore, the objective relies on unavailable information.
2. The second term uses the loss $\ell\big(-g(\boldsymbol{x}_i), y_i\big)$. Simply negating the output $g(\boldsymbol{x}_i)$ does not yield the correct loss for the negative class. This operation means that the objective will not accurately reflect the true risk.

Thus, it is not an appropriate loss function to minimize in this setting.

Question 4.2.a TODO

Once again, let's consider a binary classification problem, but this time the data is unlabeled. Now, let's suppose that the training set is drawn from the following mixture distribution

$$p_{\text{tr}}(\boldsymbol{x}) = \theta\, p_+(\boldsymbol{x}) + (1 - \theta)\, p_-(\boldsymbol{x}),$$

where $\theta \in [0, 1]$ is a known constant representing the proportion of positive examples in the training data.

Here, $p_+(\boldsymbol{x})$ and $p_-(\boldsymbol{x})$ are class-conditional distributions that do not overlap

$$p_+(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = +1) \quad \text{and} \quad p_-(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = -1),$$

We then define the expected loss of the classes as

$$e_+ = \mathbb{E}_{\boldsymbol{x}\sim p_+}\left[\ell\big(g(\boldsymbol{x}),+1\big)\right] \quad \text{and} \quad e_- = \mathbb{E}_{\boldsymbol{x}\sim p_-}\left[\ell\big(g(\boldsymbol{x}),-1\big)\right],$$

If the labels were available, the true risk of a predictor g under a loss function ℓ could be written as

$$R(g) = \theta\, e_+ + (1-\theta)\, e_-.$$

However, since the training data is unlabeled and comes from the mixture $p_{\text{tr}}(\boldsymbol{x})$, we can only measure the overall error on the training set

$$R_{\text{tr}}(g) = \theta\, e_+ + (1-\theta)\, e_-.$$

This expression is then a single equation with two unknowns e_+ and e_-. Without additional information, there is no way to uniquely determine e_+ and e_- from $R_{\text{tr}}(g)$ alone. Because of that, it is impossible to separately estimate the class-specific error rates based solely on the unlabeled mixture, making it impossible to accurately determine $R(g)$.

Question 4.2.b TODO

Let's now suppose we have a second unlabeled training set drawn from a different mixture distribution $\theta^{'}$ which is different from $\theta \neq \theta^{'}$. This gives us a second equation such as

1. $R_{\text{tr}}(g) = \theta\, e_+ + (1-\theta)\, e_-.$
2. $R^{'}_{\text{tr}}(g) = \theta^{'}\, e_+ + (1-\theta^{'})\, e_-.$

Since this system now contains two equations and two unknowns, we now have a simple solvable linear system of equations. Once the class-specific error rate e_+ and e_- are determined, we can compute the true risk as

$$R(g) = \theta\, e_+ + (1-\theta)\, e_-.$$

Question 4.3.a TODO

Let's consider a modified risk estimator

$$\widetilde{R}(g) = \left|\widehat{R}(g) - \epsilon\right| + \epsilon,$$

where ϵ is a constant chosen as a baseline risk level.

My initial intuition on this new risk estimator $\widetilde{\mathcal{R}}(g)$ is that it will be punishing the predictor when it is too precise (or overfitting). We can confirm that by breaking this equation into two cases:

1) **When $\widehat{\mathcal{R}}(g) > \epsilon$, then**

$$|\widehat{\mathcal{R}}(g) - \epsilon| = \widehat{\mathcal{R}}(g) - \epsilon$$

$$\widetilde{\mathcal{R}}(g) = \widehat{\mathcal{R}}(g) - \epsilon + \epsilon = \widehat{\mathcal{R}}(g)$$

Hence, when the empirical risk $\widehat{\mathcal{R}}(g)$ is above the threshold ϵ, the modified estimator equals the empirical risk.

2) **When $\widehat{\mathcal{R}}(g) < \epsilon$, then**

$$|\widehat{\mathcal{R}}(g) - \epsilon| = \epsilon - \widehat{\mathcal{R}}(g)$$

$$\widetilde{\mathcal{R}}(g) = \epsilon - \widehat{\mathcal{R}}(g) + \epsilon = 2\epsilon - \widehat{\mathcal{R}}(g)$$

This effectively "reflects" the empirical risk $\widehat{\mathcal{R}}(g)$ about the level ϵ, penalizing overly optimistic (too low) risk estimates.

When the empirical risk $\widehat{\mathcal{R}}(g)$ is below the threshold ϵ, it increases the estimated risk, while if the empirical risk is above ϵ, it remains unchanged. This strategy helps achieve a more reliable and stable estimate of the true risk $R(g)$.

Question 4.3.b TODO

The mean squared error (MSE) of any estimator E is given by

$$\mathrm{MSE}(E) = \mathbb{E}\Big[(E - R(g))^2\Big] = \big(\mathrm{Bias}(E)\big)^2 + \mathrm{Var}(E)$$

Since $\widehat{\mathcal{R}}(g)$ is unbiased, which means its average is exactly $\mathcal{R}(g)$, we can say that

$$\mathrm{MSE}\big(\widehat{\mathcal{R}}(g)\big) = \mathrm{Var}\big(\widehat{\mathcal{R}}(g)\big).$$

The modified estimator $\widetilde{\mathcal{R}}(g)$ may introduce a bias b but reduces the variance. Its MSE becomes

$$\mathrm{MSE}\big(\widetilde{\mathcal{R}}(g)\big) = b^2 + \mathrm{Var}\big(\widetilde{\mathcal{R}}(g)\big).$$

Its bias is given by

$$b = \mathbb{E}\big[\widetilde{\mathcal{R}}(g)\big] - R(g) = \mathbb{E}\Big[\big|\widehat{\mathcal{R}}(g) - \epsilon\big|\Big] + \epsilon - R(g)$$

If the reduction in variance is significant enough such that

$$b^2 + \mathrm{Var}\big(\widetilde{\mathcal{R}}(g)\big) < \mathrm{Var}\big(\widehat{\mathcal{R}}(g)\big)$$

- The term $\epsilon - R(g)$ measures how far our chosen constant ϵ is from the true risk.
- The term $\mathbb{E}\Big[\big|\widehat{\mathcal{R}}(g) - \epsilon\big|\Big]$ measures the average absolute deviation of the standard estimator from ϵ.

If ϵ is chosen very close to $R(g)$, then $\epsilon - R(g)$ is small. Additionally, if $\widehat{\mathcal{R}}(g)$ is symmetrically distributed around ϵ, the expected absolute deviation will be minimized, and thus the overall bias b will be small.

Question 4.3.c TODO

The constant ϵ should be ideally chosen close to the true risk $R(g)$. When ϵ is well-tuned (i.e. $\epsilon \approx R(g)$), the bias b in $\widetilde{\mathcal{R}}(g)$ is minimized and the reflection $|\widehat{\mathcal{R}}(g) - \epsilon| + \epsilon$ prevents extreme underestimation, reducing variance.

Thus, a more accurate choice of ϵ lowers the overall MSE, making the modified estimator more effective.