+1/1/60+

EPFL

1

Profs. Nicolas Flammarion and Martin Jaggi
Machine Learning – CS-433 - MA
16.01.2025 from 15h15 to 18h15 in STCC
Duration : 180 minutes

Student 1

SCIPER : **999000**

Wait for the start of the exam before turning to the next page. This document is printed double sided, 20 pages. Do not unstaple.

- This is a closed book exam. No electronic devices of any kind.
- Place on your desk: your student ID, writing utensils, one double-sided A4 page cheat sheet if you have one; place all other personal items below your desk or on the side.
- You each have a different exam.
- This exam has many questions. We do *not* expect you to solve all of them even for the best grade
- Only answers in this booklet count. No extra loose answer sheets. You can use the last two pages as scrap paper.
- For the **multiple choice** questions, we give +2 points if your answer is correct, and 0 points for incorrect or no answer.
- For the **true**/**false** questions, we give +1.5 points if your answer is correct, and 0 points for incorrect or no answer.
- Use a **black or dark blue ballpen** and clearly erase with **correction fluid** if necessary.
- If a question turns out to be wrong or ambiguous, we may decide to nullify it.

Respectez les consignes suivantes | Observe this guidelines | Beachten Sie bitte die unten stehenden Richtlinien

choisir une réponse | select an answer Antwort auswählen	ne PAS choisir une réponse | NOT select an answer NICHT Antwort auswählen	Corriger une réponse | Correct an answer Antwort korrigieren
		

ce qu'il ne faut **PAS** faire | what should **NOT** be done | was man **NICHT** tun sollte

For your examination, preferably print documents compiled from auto-multiple-choice.

First part: multiple choice questions

For each question, mark the box corresponding to the correct answer. Each question has **exactly one correct answer**.

Question 1 Recall that given a model $f : \mathcal{X} \to \{-1, 1\}$, an adversarial example can be found by solving the following *maximization* problem:

$$\max_{\hat{x}:\|\hat{x}-x\|_p \leq \varepsilon} 1_{f(\hat{x}) \neq y} \,,$$

where (x, y) is the original example, $1_{f(\hat{x})\neq y}$ is the zero-one loss and $\|\cdot\|_p$ is the ℓ_p norm. Often, the zero-one loss is replaced by a surrogate loss function.
Which of the following statements is **FALSE**?

- ☐ The zero-one loss is usually replaced by a surrogate loss function $\ell(yg(\hat{x}))$ where g is the output of the model before classification, i.e., $f(x) = \text{sign}(g(x))$.
- ☐ The maximization problem can be approximated by running projected gradient descent algorithm on a surrogate loss.
- ■ The Fast gradient sign method (FGSM) approximates the maximization problem by linearizing a surrogate loss for ℓ_2 perturbations.
- ☐ There exist closed-form solutions to relaxations of the maximization problem with surrogate losses when the model and loss are simple.
- ☐ The zero-one loss is not differentiable; hence, the original maximization problem is generally hard to optimize.

Solution: The Fast Gradient Sign Method (FGSM) approximates the maximization problem by linearizing the loss function for ℓ_∞ perturbations, not ℓ_2.

Question 2 Recall that the output of self-attention is given by the following formula:

$$Z = \text{softmax}\left(\frac{XW_QW_K^\top X^\top}{\sqrt{d_k}}\right)XW_V\,,$$

where $X \in \mathbb{R}^{n\times d}, W_Q \in \mathbb{R}^{d\times d_k}, W_K \in \mathbb{R}^{d\times d_k}, W_V \in \mathbb{R}^{d\times d}$ are the data, query, key, and value matrices, respectively. Here, d is the dimensionality of the tokens, d_k is the hidden dimensionality, n is the sequence length. The softmax is applied row-wise. Additionally, the dimensions satisfy $d_k < d < n$. Consider the following linear self-attention modification:

$$Z = \left(XW_QW_K^\top X^\top\right)XW_V\,.$$

Which of the following statements is **FALSE**?

- ☐ The linear self-attention modification can be used to reduce the complexity of the self-attention mechanism to be *linear* in the sequence length.
- ■ Replacing the product $W_QW_K^\top$ with a single matrix $W_{QK} \in \mathbb{R}^{d\times d}$ does not change the class of functions that can be expressed by both formulations of self-attention.
- ☐ The original self-attention mechanism is used in the Transformer model has a *quadratic* complexity in the sequence length.
- ☐ Both the original self-attention and the linear self-attention modification can be used with *causal masking*.
- ☐ The linear self-attention modification does not include normalization and thus does not benefit from a probabilistic interpretation.

Solution: 1. Transformers have quadratic complexity in sequence length. 2. The linear self-attention is one approach to reduce it to linear complexity. 3. The product $W_QW_K^\top$ is a $d \times d$ matrix, but with the rank $\leq d_k$, and replacing it with a single matrix $W_{QK} \in \mathbb{R}^{d\times d}$ can change the expressivity of the model if $d_k < d$. 4. The linear self-attention modification provided here does not include normalization and thus does not benefit from a probabilistic interpretation. 5. It is well-known that transformers can be used with causal masking, and the linear self-attention modification can be used with causal masking as well.

Question 3 Which of the following statements is **always** true for a real-valued data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{y} \in \mathbb{R}^n$ with $d > n$ where n is the number of data points and d represent the feature dimension?

- ☐ The sample points are linearly separable.
- ■ The data points lie in an at most n-dimensional subspace of $\mathbb{R}^d$, so there is at least one non-zero direction in $\mathbb{R}^d$ that is orthogonal to all these points.
- ☐ The weights $\mathbf{w}$ can be computed with least-squares linear regression as follows: $\mathbf{w} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$.
- ☐ $\mathbf{X}^\top\mathbf{X}$ has exactly $d - n$ eigenvectors with eigenvalue zero.

Solution: In the case where $d > n$ (more features than samples), $\mathbf{X}^\top\mathbf{X}$ is not invertible because it is a $d \times d$ matrix with rank at most n since $\mathbf{X}$ has only n rows. When $\mathbf{X}^\top\mathbf{X}$ is not full rank, its inverse does not exist, hence the least squares solution cannot be computed with the given formula. The matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ can have at most n linearly independent rows, so its rank is at most n. As a result, the rank of the $d \times d$ matrix $\mathbf{X}^\top\mathbf{X}$ is also limited to n, implying that there can be **at most** n non-zero eigenvalues. Consequently, there are **at least** $d - n$ zero eigenvalues and the statement cannot be considered definitively true. Linear separability of the sample points is not guaranteed simply by having $d > n$, it depends on the specific data. This remaining option is true.

Question 4 If $X \sim N(\mu, \sigma^2)$ and $Y = aX + b$, then which expression below gives the variance of Y?

- ☐ $a^2\sigma^2 + b$
- ☐ $a\sigma^2 + b$
- ☐ $a\sigma^2$
- ■ $a^2\sigma^2$

Solution: We use the following properties of variance:

- If $Y = aX$, then $\text{Var}(Y) = a^2\text{Var}(X)$.
- Adding a constant b does not affect the variance, i.e., $\text{Var}(X + b) = \text{Var}(X)$.

Hence, the variance of Y is: $a^2\sigma^2$.

Question 5 Given a dataset $\mathcal{X} = \{1, 2, 4, 9, 16\}$, which choice of b minimizes the mean average error (MAE): $\mathcal{L}(b) = \frac{1}{5}\sum_{x\in\mathcal{X}} |x - b|$?

- ☐ π
- ■ 4.0
- ☐ $\sqrt{358/5}$
- ☐ 6.4
- ☐ 2.0
- ☐ 9.0
- ☐ $\sqrt[3]{16}$
- ☐ 2π
- ☐ 8.5

Solution: The MAE in this case is uniquely minimized by the median, which is 4.

Question 6 Imagine you are designing a convolutional neural network (CNN) for image classification with 10 classes in total. Each image is of size $(32, 32, 3)$, and the layers are of the following configurations. We denote a convolutional layer by (kernel height, kernel width, number of filters).

- $(3, 3, 16)$ Convolutional layer with stride 1 and no padding
- Max-pooling layer with pooling size $(2, 2)$ and stride 2
- $(4, 4, 32)$ Convolutional layer with stride 1 and no padding
- Max-pooling layer with pooling size $(2, 2)$ and stride 2
- Flatten and connect to a fully connected layer with output dimension 128
- Fully connected layer with output dimension 10

Which of the following statements is true?

- ☐ The max pooling layers have learnable parameters.
- ☐ After the first convolutional layer, the output size is $(30, 30, 3)$.
- ■ After the second convolutional layer, the output size is $(12, 12, 32)$.
- ☐ The second convolutional layer has the most learnable parameters among all the layers.

Solution: After the 1st Conv layer: $(30, 30, 16)$, after the 1st MaxPooling: $(15, 15, 16)$, after the 2nd Conv layer: $(12, 12, 32)$, after the 2nd MaxPooling: $(6, 6, 32)$. The first Fully Connected layer has parameter count $6 * 6 * 32 * 128$, which is the most. The MaxPooling layers have no learnable parameters.

Question 7 Which of the following statements is true?

- ☐ For a batch-normalized NN, the batch statistics are calculated from the current batch during inference time.
- ■ Data augmentation can make CNNs less sensitive to image rotations.
- ☐ The computational complexity of training a fully connected neural network is quadratic in the number of layers for a single iteration.
- ☐ When normalization is applied to ensure stable training, the neural networks can be arbitrarily initialized and still achieve near-optimal performance.

Solution: The computational complexity of training a fully connected neural network is quadratic in the width for a single iteration. The batch statistics are fixed during inference time to normalize a batch, usually EMAs of training statistics are used. Even when normalization is applied, the neural networks cannot be arbitrarily initialized, in order to avoid gradient exploding/vanishing.

Question 8 Which of the following statements is TRUE regarding the application of logistic regression and linear regression on binary classification tasks?

- ■ Logistic regression models the probability of an instance belonging to a particular class, while linear regression predicts class labels directly.
- ☐ Logistic regression is able to generate non-linear decision boundaries with respect to original input features when the data are not linearly separable.
- ☐ Both logistic regression and linear regression have a closed-form solution for the optimal parameters.
- ☐ For linearly separable data, logistic regression via gradient descent converges in a finite number of steps.

Solution: The correct one is 'Logistic regression models the probability of an instance belonging to a particular class, while linear regression predicts class labels directly'.

Question 9 Suppose we are using a logistic regression model to predict the classes of y with a single feature x. Let θ_0 be the bias and θ_1 be the coefficient for x. The predicted probability that $y = 1$ given $x = x_0$ is:

$$\hat{p}(y = 1 \mid x = x_0) = \sigma(\theta_0 + \theta_1 x_0) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_0)}}.$$

Suppose we increase x_0 by one unit, resulting in:

$$\hat{p}(y = 1 \mid x = x_0 + 1) = \sigma(\theta_0 + \theta_1(x_0 + 1)).$$

Which of the following statements best describes how the predicted probability for $y = 1$ changes?

- ☐ The predicted probability is multiplied by a fixed amount of $\sigma(\theta_1)$.
- ☐ The predicted probability is multiplied by a fixed amount equal to θ_1.
- ■ The amount by which the predicted probability changes depends on its current value of x_0.
- ☐ The predicted probability is multiplied by a fixed amount of e^{θ_1}.

Solution: The correct answer is 'The amount by which the predicted probability changes depends on its current value of x_0.'. The sigmoid function $\sigma(z)$ produces a non-linear relationship between x and the predicted probability. Thus, the change in probability depends on the current value of x. At extreme values of x, the probability changes less due to the saturation effect of the sigmoid function.

Question 10 Which of the following statements is true about the K-means algorithm with the following objective function?

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \|x_n - \mu_k\|^2 \text{ subject to } z_{nk} \in \{0, 1\} \text{ and } \sum_{k=1}^{K} z_{nk} = 1 \text{ for all } n.$$

- ☐ The objective value might increase during training ($J_{t+1} > J_t$) due to non-convexity of the objective function or poor initialization.
- ☐ K-means converges to a local minimum because the objective function is convex.
- ■ Increasing the number of clusters (K) always decreases (or keeps constant) the optimal objective value.
- ☐ K-means is guaranteed to converge to the global minimum of the objective function.

Solution: The objective function in K-means is the sum of squared distances between data points and their corresponding cluster centroids. As the number of clusters (K) increases, the centroids can more finely capture the structure of the data, reducing the sum of squared distances. This will either decrease or keep constant the value of the objective function, since each point can now be assigned to a closer centroid. Reasons for incorrect options: The objective function is non-convex, the objective value will never increase during training (proof in the lecture), there is no guarantee of convergence to the global minimum.

Question 11 Suppose you are using the K-means++ algorithm to initialize the centroids for a clustering problem. The dataset consists of the following 1D points: $\{1, 2, 3, 10, 11, 12\}$. The first centroid is chosen at random as 2. Which point is more likely to be selected as the next centroid?

- ☐ All points are equally likely to be selected as the next centroid.
- ☐ 1 or 3 (equally likely)
- ☐ 11
- ☐ 10
- ■ 12

Solution: To compute the probability that 12 is chosen as the second centroid, we first calculate the squared distances (D^2) of all points from the first centroid (2): $(1-2)^2 = 1$, $(2-2)^2 = 0$, $(3-2)^2 = 1$, $(10-2)^2 = 64$, $(11-2)^2 = 81$, and $(12-2)^2 = 100$. Ignoring 2, as it is already a centroid, the remaining squared distances are $\{1, 1, 64, 81, 100\}$, summing to 247. The probability of selecting 12 is proportional to its squared distance, calculated as $\frac{100}{247}$. Thus, the correct answer is $\frac{100}{247}$.

Expectation Maximization

We consider a mixture of two exponential distributions, where a random variable X is constructed as follows: First, the class $Z \in \{0, 1\}$ is chosen according to:

$$P(Z = 0) = \pi_0, \quad P(Z = 1) = \pi_1 = 1 - \pi_0.$$

Then, the random variable X is sampled from an exponential distribution with parameter λ_0 or λ_1, depending on the class Z. The conditional densities are:

$$p(x \mid Z = 0) = \begin{cases} \lambda_0 e^{-\lambda_0 x}, & x \geq 0 \\ 0, & \text{otherwise,} \end{cases} \quad p(x \mid Z = 1) = \begin{cases} \lambda_1 e^{-\lambda_1 x}, & x \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

We are given n i.i.d. observations $x_{1:n} = (x_1, \ldots, x_n)$ from the generative model. Let $z_{1:n} = (z_1, \ldots, z_n)$ denote the hidden variables, where $z_i = 0$ if the i-th observation belongs to class 0, and $z_i = 1$ otherwise. Denote the parameter vector as $\theta = (\lambda_0, \lambda_1, \pi_0, \pi_1)$.

Question 12 Which of the following represents the complete-data likelihood $p(x_{1:n}, z_{1:n} \mid \theta)$?

- [x] $\prod_{i=1}^{n} \left[(\pi_0 \lambda_0 e^{-\lambda_0 x_i})^{1-z_i} (\pi_1 \lambda_1 e^{-\lambda_1 x_i})^{z_i}\right]$
- [ ] $\prod_{i=1}^{n} \left[(\pi_0 \lambda_0 e^{-\lambda_0 x_i}) + (\pi_1 \lambda_1 e^{-\lambda_1 x_i})\right]$
- [ ] $\prod_{i=1}^{n} \left[(\lambda_0 e^{-\lambda_0 x_i})^{1-z_i} (\lambda_1 e^{-\lambda_1 x_i})^{z_i}\right]$
- [ ] $\prod_{i=1}^{n} \left[(\lambda_0 e^{-\lambda_0 x_i}) + (\lambda_1 e^{-\lambda_1 x_i})\right]$

Solution: The complete-data likelihood $p(x_{1:n}, z_{1:n} \mid \theta)$ is given by:

$$p(x_{1:n}, z_{1:n} \mid \theta) = \prod_{i=1}^{n} p(x_i, z_i \mid \theta) = \prod_{i=1}^{n} p(z_i \mid \theta) p(x_i \mid z_i, \theta).$$

For $z_i = 0$:

$$p(z_i = 0 \mid \theta) = \pi_0 \quad \text{and} \quad p(x_i \mid z_i = 0, \theta) = \lambda_0 e^{-\lambda_0 x_i}.$$

For $z_i = 1$:

$$p(z_i = 1 \mid \theta) = \pi_1 \quad \text{and} \quad p(x_i \mid z_i = 1, \theta) = \lambda_1 e^{-\lambda_1 x_i}.$$

Combining these, we get:

$$p(x_i, z_i \mid \theta) = (\pi_0 \lambda_0 e^{-\lambda_0 x_i})^{1-z_i} (\pi_1 \lambda_1 e^{-\lambda_1 x_i})^{z_i}.$$

Therefore, the complete-data likelihood is:

$$p(x_{1:n}, z_{1:n} \mid \theta) = \prod_{i=1}^{n} \left[(\pi_0 \lambda_0 e^{-\lambda_0 x_i})^{1-z_i} (\pi_1 \lambda_1 e^{-\lambda_1 x_i})^{z_i}\right].$$

Question 13 We run the EM algorithm on this data. The vector $\theta^{(t)}$ denotes the parameters at the t-th iteration of the algorithm. At iteration $t+1$, we perform the E-step by computing:

$$Q(\theta; \theta^{(t)}) := \mathbb{E}_{z_{1:n}} \left[\log p(x_{1:n}, z_{1:n} \mid \theta) \mid x_{1:n}, \theta^{(t)}\right].$$

Denote $q_1^{(t)}(x_i) = P(z_i = 1 \mid x_i, \theta^{(t)})$, and define $q_0^{(t)}(x_i) = 1 - q_1^{(t)}(x_i)$.

What is the value of $Q(\theta; \theta^{(t)})$?

- [ ] $\sum_{i=1}^{n} \left[(\lambda_0 e^{-\lambda_0 x_i}) q_0^{(t)}(x_i) + (\lambda_1 e^{-\lambda_1 x_i}) q_1^{(t)}(x_i)\right]$
- [ ] $\sum_{i=1}^{n} \left[q_0^{(t)}(x_i)(\log(\lambda_0) - \lambda_0 x_i) + q_1^{(t)}(x_i)(\log(\lambda_1) - \lambda_1 x_i)\right]$
- [x] $\sum_{i=1}^{n} \left[q_0^{(t)}(x_i)(\log(\pi_0 \lambda_0) - \lambda_0 x_i) + q_1^{(t)}(x_i)(\log(\pi_1 \lambda_1) - \lambda_1 x_i)\right]$
- [ ] $\sum_{i=1}^{n} \left[(\pi_0 \lambda_0 e^{-\lambda_0 x_i}) q_0^{(t)}(x_i) + (\pi_1 \lambda_1 e^{-\lambda_1 x_i}) q_1^{(t)}(x_i)\right]$

Solution: Due to independence, the value of $Q(\theta; \theta^{(t)})$ is given by:

$$Q(\theta; \theta^{(t)}) = \sum_{i=1}^{n} \mathbb{E}_{z_i \mid x_i, \theta^{(t)}} \left[\log p(x_i, z_i \mid \theta)\right].$$

Using the definition of $q_1^{(t)}(x_i)$ and $q_0^{(t)}(x_i)$, we can write:

$$Q(\theta; \theta^{(t)}) = \sum_{i=1}^{n} \left[q_0^{(t)}(x_i) \log p(x_i, z_i = 0 \mid \theta) + q_1^{(t)}(x_i) \log p(x_i, z_i = 1 \mid \theta)\right].$$

Substituting the expressions for $p(x_i, z_i = 0 \mid \theta)$ and $p(x_i, z_i = 1 \mid \theta)$, we get:

$$Q(\theta; \theta^{(t)}) = \sum_{i=1}^{n} \left[q_0^{(t)}(x_i)(\log(\pi_0 \lambda_0) - \lambda_0 x_i) + q_1^{(t)}(x_i)(\log(\pi_1 \lambda_1) - \lambda_1 x_i)\right].$$

Question 14 What is the value of $q_1^{(t)}(x_i) = P(z_i = 1 \mid x_i, \theta^{(t)})$?

- ☐ $\frac{\lambda_1^{(t)} e^{-\lambda_1^{(t)} x_i}}{\lambda_0^{(t)} e^{-\lambda_0^{(t)} x_i} + \lambda_1^{(t)} e^{-\lambda_1^{(t)} x_i}}$
- ☐ $\pi_1^{(t)} \lambda_1^{(t)} e^{-\lambda_1^{(t)} x_i}$
- ■ $\frac{\pi_1^{(t)} \lambda_1^{(t)} e^{-\lambda_1^{(t)} x_i}}{\pi_0^{(t)} \lambda_0^{(t)} e^{-\lambda_0^{(t)} x_i} + \pi_1^{(t)} \lambda_1^{(t)} e^{-\lambda_1^{(t)} x_i}}$
- ☐ $\pi_0^{(t)} \lambda_0^{(t)} e^{-\lambda_0^{(t)} x_i} + \pi_1^{(t)} \lambda_1^{(t)} e^{-\lambda_1^{(t)} x_i}$

Solution: The value of $q_1^{(t)}(x_i)$ is given by the posterior probability:

$$q_1^{(t)}(x_i) = P(z_i = 1 \mid x_i, \theta^{(t)}) = \frac{P(z_i = 1, x_i \mid \theta^{(t)})}{P(x_i \mid \theta^{(t)})}.$$

Using Bayes' theorem, we can write:

$$q_1^{(t)}(x_i) = \frac{P(x_i \mid z_i = 1, \theta^{(t)}) P(z_i = 1 \mid \theta^{(t)})}{P(x_i \mid \theta^{(t)})}.$$

Substituting the expressions for $P(x_i \mid z_i = 1, \theta^{(t)})$ and $P(z_i = 1 \mid \theta^{(t)})$, we get:

$$q_1^{(t)}(x_i) = \frac{\lambda_1^{(t)} e^{-\lambda_1^{(t)} x_i} \pi_1^{(t)}}{\lambda_0^{(t)} e^{-\lambda_0^{(t)} x_i} \pi_0^{(t)} + \lambda_1^{(t)} e^{-\lambda_1^{(t)} x_i} \pi_1^{(t)}}.$$

Therefore, the value of $q_1^{(t)}(x_i)$ is:

$$q_1^{(t)}(x_i) = \frac{\pi_1^{(t)} \lambda_1^{(t)} e^{-\lambda_1^{(t)} x_i}}{\pi_0^{(t)} \lambda_0^{(t)} e^{-\lambda_0^{(t)} x_i} + \pi_1^{(t)} \lambda_1^{(t)} e^{-\lambda_1^{(t)} x_i}}.$$

Question 15 In the M-step, what is the updated value of $\lambda_0^{(t+1)}$?

- ☐ $\frac{\sum_i x_i}{\sum_i q_0^{(t)}(x_i)}$
- ■ $\frac{\sum_i q_0^{(t)}(x_i)}{\sum_i x_i q_0^{(t)}(x_i)}$
- ☐ $\frac{\sum_i q_0^{(t)}(x_i)}{\sum_i x_i}$
- ☐ $\frac{\sum_i x_i q_0^{(t)}(x_i)}{\sum_i q_0^{(t)}(x_i)}$

Solution: In the M-step of the EM algorithm, we update the parameters to maximize the expected complete-data log-likelihood. For the parameter λ_0, the update rule is derived as follows:
The expected complete-data log-likelihood is given by:

$$Q(\theta; \theta^{(t)}) = \sum_{i=1}^{n} \left[q_0^{(t)}(x_i) \log(\pi_0 \lambda_0 e^{-\lambda_0 x_i}) + q_1^{(t)}(x_i) \log(\pi_1 \lambda_1 e^{-\lambda_1 x_i}) \right].$$

To maximize $Q(\theta; \theta^{(t)})$ with respect to λ_0, we take the derivative and set it to zero:

$$\frac{\partial Q(\theta; \theta^{(t)})}{\partial \lambda_0} = \sum_{i=1}^{n} q_0^{(t)}(x_i) \left(\frac{1}{\lambda_0} - x_i \right) = 0.$$

Solving for λ_0, we get:

$$\lambda_0 = \frac{\sum_{i=1}^{n} q_0^{(t)}(x_i)}{\sum_{i=1}^{n} x_i q_0^{(t)}(x_i)}.$$

Question 16 In matrix factorization for recommender systems, the goal is to approximate a partially observed user-item rating matrix X by decomposing it as the product of two lower-dimensional matrices: W (item features) and Z (user features). The observed ratings x_{dn} represent the rating of the n^{th} user for the d^{th} item, with Ω as the set of indices for observed ratings. Which of the following is true regarding the regularized matrix factorization objective, given by:

$$\min_{W,Z} \frac{1}{2} \sum_{(d,n)\in\Omega} \left(x_{dn} - (WZ^\top)_{dn}\right)^2 + \frac{\lambda_W}{2}\|W\|_F^2 + \frac{\lambda_Z}{2}\|Z\|_F^2$$

- ☐ The optimization requires filling in all missing entries in X before training.
- ☐ The Frobenius norm regularization ensures the sparsity (the number of non-zero entries) of W and Z.
- ☐ The cost function is jointly convex in W and Z.
- ■ Regularization terms $\frac{\lambda_W}{2}\|W\|_F^2$ and $\frac{\lambda_Z}{2}\|Z\|_F^2$ are used to prevent overfitting to the observed ratings.

Solution: While the loss is convex in W when Z is fixed and vice versa, it is not jointly convex in both variables. The Frobenius norm regularization does not ensure sparsity. Instead, it penalizes the magnitude of the entries in W and Z, but it does not specifically encourage zero entries. Sparsity is typically encouraged by using L_1 regularization. The regularization terms help to prevent overfitting by penalizing large values in the matrices W and Z, thus encouraging simpler models that generalize better to unseen data. The optimization does not require filling in all missing entries in X. The objective function is defined only over the observed entries, and the optimization can proceed without imputing the missing values.

Question 17 Let $N \in \mathbb{N}$. Which of the following sequence-to-sequence functions $f : \{0,1\}^N \to \{0,1\}^N$ **cannot** be arbitrarily well approximated by a decoder-only transformer without causal masking and without positional embeddings? In the choices below, $f_i(x_1, \ldots, x_N)$ refers to the i$^{\text{th}}$ coordinate of $f(x_1, \ldots, x_N)$.

- ☐ $f(x) = \mathbf{0}$, $\forall x \in \{0,1\}^N$, where $\mathbf{0} \in \{0,1\}^N$ is the zero vector.
- ☐ $f : \{0,1\}^N \to \{0,1\}^N$ such that $f_i(x) = x_1 \oplus \cdots \oplus x_{i-1} \oplus x_{i+1} \oplus \cdots \oplus x_N$, where $\oplus$ is the addition modulo 2 operation.
- ■ $f(x) = v$, $\forall x \in \{0,1\}^N$, where $v \in \{0,1\}^N$ is the vector of alternating 0's and 1's: $v = (0,1,0,1,\ldots)$.
- ☐ $f : \{0,1\}^N \to \{0,1\}^N$ such that $f_i(x) = 1 - x_i$.

Solution: A decoder-only transformer without causal masking and without positional encodings cannot implement the function $f(x) = v$, $\forall x \in \{0,1\}^N$, where $v \in \{0,1\}^N$ is the vector of alternating 0's and 1's: $v = (0,1,0,1,\ldots)$. Indeed, such a transformer T can only implement permutation-invariant sequence-to-sequence mappings. Hence, if $T(0,1,\ldots) = (0,1,\ldots)$, then $T(1,0,\ldots) = (1,0,\ldots)$.

Question 18
Why would you use LASSO over Ridge regression?

A) It can help us identify which features are important.

B) It is faster to learn the weights for LASSO than for Ridge.

C) LASSO usually achieves a lower generalization error than Ridge.

D) If there are many features, the model learned using LASSO can make predictions more efficient.

- ☐ C and D
- ☐ A
- ☐ C
- ☐ A and B
- ☐ D
- ☐ B and C
- ■ A and D.
- ☐ B
- ☐ A and C
- ☐ B and D

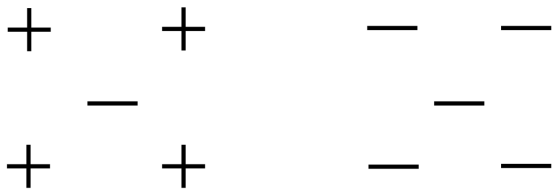
Solution: Correct choice A and D. LASSO helps identify which features are important, it also encourages sparsity which makes predictions more efficient.

Question 19 Suppose we fit "LASSO Regression" to a data set, which has 100 features $(X_1, X_2, \ldots, X_{100})$. Now, we rescale one of these features by multiplying with 10 (say that feature is X_1), and then refit LASSO regression with the same regularization parameter. Which of the following options will be **TRUE**?

- ■ It is more likely for X_1 to be included in the model.
- ☐ Can't say.
- ☐ It is more likely for X_1 to be excluded from the model.
- ☐ None of these.

Solution: if X_1 was already selected, then after multiplying by a factor greater than 1, X_1 will still be selected. If it wasn't selected, then it might be selected. In all cases, the chances of X_1 being selected are higher after multiplying by a factor greater than 1.

Question 20 We are given the following dataset as illustrated, and assume that "+" and "-" denotes the label of each data point:



The Leave-One-Out cross validation errors using 1-Nearest-Neighbor and 3-Nearest-Neighbor classifiers are:

- ☐ 0, 5/10.
- ■ 5/10, 1/10.
- ☐ 0, 1/10.
- ☐ 1/10, 5/10.

Solution: Using 1-Nearest-Neighbor, all the 5 data points on the left hand side will be wrong; while using 3-Nearest-Neighbor, only the "-" data point on the left hand side gets the wrong prediction.

Question 21 Consider a contrastive learning setup with an encoder f_θ that outputs normalized embeddings of images. During training, each image x is augmented twice to create views x_i, x_i'. The contrastive loss is then defined as

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log \frac{\exp(f_\theta(x_i)^\top f_\theta(x_i')/\tau)}{\sum_{k=1}^{N} \mathbb{1}_{k\neq i} \exp(f_\theta(x_i)^\top f_\theta(x_k)/\tau)},$$

where τ is the temperature parameter and N denotes the batch of size. Which of the following statements is FALSE?

- ■ To achieve optimal performance, all possible image augmentations should be applied with equal probability during training.
- ☐ Two different random crops of the same image should produce similar embeddings, while crops from different images should produce dissimilar embeddings.
- ☐ The quality of learned representations depends heavily on the choice of image augmentations, which should preserve semantically meaningful features while varying nuisance factors.
- ☐ The temperature parameter τ controls the sensitivity of the model to hard vs. easy negative examples.

Solution: Not all augmentations are equally useful - they must be carefully chosen based on the downstream task and data domain. Some augmentations might destroy semantic information (e.g., vertical flips for natural images). The choice and probability of augmentations should reflect invariances we want the model to learn.

Question 22 Consider a diffusion model with forward process $q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1-\alpha_t)I)$ where α_t decreases with t. Which of the following statements is FALSE?

- ☐ The forward process gradually adds Gaussian noise to the data until reaching a standard normal distribution in the limit of $t \to \infty$.
- ☐ The backward process can be implemented by training a neural network to predict the noise component that was added to the clean data.
- ■ Similar to GANs, diffusion models require training two separate models: one for the forward process and one for the backward process of diffusion.
- ☐ For training, solving the regression problem of predicting the noise at each timestep is equivalent to solving a denoising score matching problem.

Solution: The false statement is that the model must predict the exact noise at each timestep. In practice: 1. The model only needs to estimate the noise at the current timestep 2. We can skip intermediate timesteps (accelerated sampling) 3. The model learns the general denoising process, not the specific noise instances 4. Small errors in noise prediction don't significantly impact final sample quality

Question 23 Consider four points $A = (-1, 0), B = (-1, 1), C = (1, 0), D = (1, 1)$ in $\mathbb{R}^2$. Each point can be assigned a label of -1 or 1. Which of the following statements are TRUE?

P) For any assignment of labels to the points A, B, C, D, there always exists a linear classifier that perfectly separates them.

Q) Consider the cases where A, C are labelled $-1, 1$, respectively, and B and D have aribtrary labels. In such cases, no linear classifier can achieve a margin of 1.25 over these points.

- ☐ P
- ☐ Both P and Q
- ■ Q
- ☐ Neither P nor Q

Solution: Statement P is false as there exists an assignment of labels to A, B, C, D such that no linear classifier can accurately classify them. For example, if A, B, C, D are labeled as $-1, 1, 1, -1$, there is no linear classifier that can separate them because $D = 2 * C + B$. If B and C have positive labels, D must also have a positive label for a linear classifier.
For statement Q, consider any linear classifier that correctly classifies those points. It intersects the X-axis between $(-1, 0)$ and $(1, 0)$. The minimum distance of the intersection point from A and C is 1, so the margin cannot be 1.25.

Question 24 Recall that the soft-margin SVM corresponds to the following optimization problem:

$$\min_{w,\xi} \frac{\lambda}{2}\|w\|^2 + \frac{1}{N}\sum_{i=1}^{N}\xi_i, \quad \text{subject to: } y_i(w^T x_i) \geq 1 - \xi_i \text{ for all } 1 \leq i \leq N, \text{ and } \xi_i \geq 0$$

Which of the following statements is TRUE?

- ☐ For the optimal solution, it is always true that $\xi_i \leq 1$ for all i.
- ☐ The soft-margin SVM problem is equivalent to the minimization of a regularized logistic loss function.
- ■ There cannot be a solution to the soft-margin SVM where $\xi_i \geq 1$ for all i.
- ☐ The given optimization problem admits a solution if and only if the data is linearly separable.

Solution: If $\xi_i \geq 1$ for all i, it means all the points are wrongly classified. Then $-w$ classifies all the points correctly and has a lower objective value.

Question 25 Which of the following statements about the k-nearest neighbors algorithm is TRUE? Here, N denotes the number of data points.

- ☐ The variance of the k-nearest neighbors method is 0 when $k = N$.
- ☐ The bias of the k-nearest neighbors method is 0 when $k = N$.
- ■ The k-nearest neighbors method is sensitive to the choice of distance metric.
- ☐ The k-nearest neighbors approach is only applicable to classification tasks.

Solution: The k-nearest neighbors method is sensitive to the choice of distance metric. Different distance metrics can lead to different nearest neighbors. Regression can be done using k-nearest neighbors by taking the average of the neighbors. The variance of the k-nearest neighbors method is low but non-zero when $k = N$. The bias of the k-nearest neighbors method is non-zero when $k = N$.

Second part: true/false questions

For each question, mark the box (without erasing) TRUE if the statement is **always true** and the box FALSE if it is **not always true** (i.e., it is sometimes false).

Question 26 (Large Language Models) Suppose we have a decoder-only transformer T with a context window of size N. We want to use T to compute an infinite binary function $f : \{0,1\}^\infty \to \{0,1\}^\infty$, where we will use f_i to refer to the i^{th} coordinate of f. Now, for every $x = (x_1, x_2, \dots) \in \{0,1\}^\infty$, we have that $f_i(x) = 0$ for $i < N$ and $f_i(x) = g_i(x_{i-N+1}, \dots, x_i)$ for $i \geq N$, where $g_i : \{0,1\}^N \to \{0,1\}$. For every sequence $x \in \{0,1\}^\infty$, we will apply T with a sliding context window. That is, $T(x) = y$ will mean that $y_i = T(x_{\max(1,i-N+1)}, \dots, x_i)$. Suppose it is known that each g_i requires at least linear time, $\Omega(i)$, to be computed. Then, T cannot compute f.

■ TRUE ☐ FALSE

Solution: True. Every infinite function computable by a sliding-context-window transformer necessarily takes constant time per coordinate to be computed since $T(x_{\max(1,i-N+1)}, \dots, x_i)$ takes a bounded amount of time.

Question 27 (Adversarial robustness) Adversarial training with the Fast Gradient Sign Method roughly has twice the computational cost of training the model without adversarial training.

■ TRUE ☐ FALSE

Solution: True. The computation of adversarial examples requires an additional forward and backward pass through the model.

Question 28 (Masked Language Models) BERT-style masked language models cannot be used for autoregressive generation of text.

☐ TRUE ■ FALSE

Solution: False. While BERT models are not designed for autoregressive generation, they can be adapted for this purpose. Concretely, the model can predict the next token via the [MASK] token at the end of a sequence; this predicted token can then replace the [MASK] token in the sequences, which is fed back into the network and so on.

Question 29 (Text Representation Learning) Suppose we have trained a **FastText** model to learn sentence representations for the task of determining whether the subject in a given sentence is human or not. It is possible that our model could **incorrectly** classify both the sentences `"Francesco devoured the unsuspecting octopus."` and `"The unsuspecting octopus devoured Francesco."`.

☐ TRUE ■ FALSE

Solution: False. FastText uses the bag-of-words representation of a sentence for classification. Hence, both sentences will receive the same classification label, as they contain exactly the same words. Therefore, at least (and at most) one of these sentences will be correctly classified as having a human subject.

Question 30 (Overfitting) When using a train/test split, if the split is not random, this will primarily cause underfitting.

☐ TRUE ■ FALSE

Solution: False. The train data might not be representative of the entire dataset and thus overfitting should be expected.

Question 31 (Gaussian Mixture Models) We fit a GMM to a dataset utilizing the (soft) EM algorithm. Let L_t denote the log-likelihood of the data at iteration t. During this process, the log-likelihood L_t may decrease in some iterations, i.e., there may exist t such that $L_{t+1} < L_t$.

■ FALSE ☐ TRUE

Solution: It is guaranteed during the EM algorithm that the log-likelihood of the data is non-decreasing at each iteration, i.e., $L_{t+1} \geq L_t$.

Question 32 (Nearest Neighbours) In a binary classification problem with a training set of size 2, where the points have different labels, the decision boundary of the 1-nearest neighbor (1-NN) classifier with ℓ_1-distance coincides with that of the maximum margin linear classifier.

☐ TRUE ■ FALSE

Solution: Consider the case of two points $A = (0, 0)$, $B = (2, 1)$

Question 33 (Generative Adversarial Networks) In GAN training, if the discriminator achieves perfect classification accuracy on both real and fake samples, this means we have reached the training optimum and training should stop.

☐ TRUE ■ FALSE

Solution: False. A perfect discriminator actually indicates training failure – it means the generator is producing samples that are easily distinguishable from real data. In the optimal Nash equilibrium, the discriminator should achieve 50% accuracy, as it cannot distinguish between real and generated samples. Training should continue (or be restarted) until the generator improves.

Question 34 (Overfitting) Training your model until it achieves a low loss value on your test data is a good way to prevent overfitting.

☐ TRUE ■ FALSE

Solution: False. The test set would cease being a good measure of performance on unseen data.

Question 35 (Masked Language Models) In BERT-style masked language modeling, if we mask multiple tokens in a sentence during pre-training, the prediction of a masked token is independent of the predictions of other masked tokens, conditioned on the context.

■ TRUE ☐ FALSE

Solution: True. During the forward pass, BERT predicts all masked tokens in parallel and independently. The predictions for each masked position are based only on the surrounding visible context and the [MASK] tokens at other positions, but not on what the model predicts for other masked positions. This is different from autoregressive models like GPT that predict tokens sequentially.

Question 36 (Fairness) The independence criterion of fairness requires that the false positive rate (FPR) and false negative rate (FNR) must be equal across all groups.

☐ TRUE ■ FALSE

Solution: This is the separation criterion, not the independence criterion. In the independence criterion, the conditional probability of a positive outcome given the sensitive attribute should be equal across all groups.

Question 37 (Batch Normalization) Batch normalization is a technique that reduces the processing time of a single batch, and thus leads to much faster convergence.

☐ TRUE ■ FALSE

Solution: False. BN increases the processing time of a single batch, but still may speed up convergence due to its ability to stabilize and improve the training dynamics.

Question 38 (Optimization) Given a continuous, strictly convex, loss function $\mathcal{L}(w)$, a gradient descent step $w_{t+1} = w_t - \eta \nabla \mathcal{L}(w_t)$ with learning rate $\eta > 0$, results in a decrease in the loss, i.e. $\mathcal{L}(w_{t+1}) \leq \mathcal{L}(w_t)$.

☐ TRUE ■ FALSE

Solution: False. This is only true in general if we impose additional constraints on the learning rate. For example if $\mathcal{L}(w_t) = w_t^2$ with $\eta = 2, w_t = 1$ we get $w_{t+1} = -3$ with $\mathcal{L}(w_{t+1}) = 9 > 1 = \mathcal{L}(w_t)$.

Question 39 (Optimization) The $0-1$ loss is **not** a suitable loss function for training a neural network classifier with stochastic gradient descent.

■ TRUE ☐ FALSE

Solution: True. The $0-1$ loss is a piecewise-constant function (neither continuous nor differentiable). This prevents it from being minimized via gradient descent (the gradients are zero everywhere they are defined). In practice we use other functions like cross-entropy as a proxy.

Third part, open questions

Answer in the empty space below. Your answer should be carefully justified, and all the steps of your argument should be discussed in details. Leave the check-boxes empty, they are used for the grading.

Answer in the space provided! Your answer must be justified with all steps. Leave the check-boxes empty, they are used for the grading.

1 Convexity

Question 40: (*3 points*) Let $f_1, f_2 : \mathbb{R} \to \mathbb{R}$ be convex functions. Consider the following statements and determine whether each statement is true or false. If the statement is true, provide a proof. If the statement is false, either provide a counterexample or disprove it.

(a) The function $g(x) = \min(f_1(x), f_2(x))$ is convex.

(b) The function $h(x) = \max(f_1(x), f_2(x))$ is convex.

☐0 ☐1 ☐2 ■3

Solution:

(a) **False.** Consider $f_1(x) = (x-1)^2$ and $f_2(x) = (x+1)^2$. We can compute $g(-1) = 0$, $g(0) = 1$, and $g(1) = 0$. Since $g(0) > g(-1)$ and $g(0) > g(1)$, $g(x)$ is not convex.

(b) **True.** For $x, y \in \mathbb{R}$ and $\lambda \in (0,1)$, we have:

$$\begin{aligned} h(\lambda x + (1-\lambda)y) &:= \max\{f_1(\lambda x + (1-\lambda)y), f_2(\lambda x + (1-\lambda)y)\} \\ &\leq \max\{\lambda f_1(x) + (1-\lambda) f_1(y), \lambda f_2(x) + (1-\lambda) f_2(y)\} \\ &\leq \lambda \max\{f_1(x), f_2(x)\} + (1-\lambda)\max\{f_1(y), f_2(y)\} \\ &= \lambda h(x) + (1-\lambda) h(y). \end{aligned}$$

Thus, h is convex.

2 Logistic Regression Loss

Question 41: (*3 points*) Consider the logistic regression loss $L : \mathbb{R}^d \to \mathbb{R}$ for a binary classification task with data $(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \{0,1\}$ for $i \in \{1, \ldots, N\}$:

$$L(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} \big(\log(1 + e^{\mathbf{x}_i^\top \mathbf{w}}) - y_i \, \mathbf{x}_i^\top \mathbf{w}\big).$$

Questions:

(a) Compute the gradient $\nabla_{\mathbf{w}} L(\mathbf{w})$ of the loss with respect to $\mathbf{w}$.

(b) Prove that the loss function $L(\mathbf{w})$ is convex with respect to $\mathbf{w}$.

☐0 ☐1 ☐2 ■3

Solution:

(a) The gradient of $L(\mathbf{w})$ with respect to $\mathbf{w}$ is:

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} \big(\sigma(\mathbf{x}_i^\top \mathbf{w}) - y_i\big)\mathbf{x}_i,$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function.

(b) The loss function $L(\mathbf{w})$ is convex because its second derivative, the Hessian matrix, is positive semi-definite. Specifically:

$$H = \frac{1}{N}\sum_{i=1}^{N}\sigma(\mathbf{x}_i^\top \mathbf{w})(1-\sigma(\mathbf{x}_i^\top \mathbf{w}))\mathbf{x}_i\mathbf{x}_i^\top,$$

where $\sigma(\mathbf{x}_i^\top \mathbf{w})(1-\sigma(\mathbf{x}_i^\top \mathbf{w})) \geq 0$. Since the Hessian is a sum of positive semi-definite matrices, $L(\mathbf{w})$ is convex. Alternatively, the convexity could also be proved with convexity-preserving operations (as in slide 19 of lecture05b).

3 Kernels

Let $m \in \mathbb{N}^+$ be a positive integer, and let $c \in \mathbb{R}$ satisfy $c < 0$. For any positive integer $d \in \mathbb{N}^+$, consider the function $k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ defined by:

$$k(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^\top \mathbf{x}' + c\right)^d \quad \text{for all} \quad \mathbf{x}, \mathbf{x}' \in \mathbb{R}^m.$$

Question 42: (*1 points*) **Definition of a valid kernel:** State what it means for a general function $k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ to be a *valid kernel*.

☐0 ■1

Solution: A symmetric function $k : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is called a *positive semidefinite (PSD) kernel* (or Mercer kernel) if for every finite set of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^m$, the $n \times n$ *Gram matrix*

$$\mathbf{K} = \left(k(\mathbf{x}_i, \mathbf{x}_j)\right)_{i,j=1}^{n}$$

is positive semidefinite. Equivalently, for any $\mathbf{v} \in \mathbb{R}^n$,

$$\mathbf{v}^\top \mathbf{K}\, \mathbf{v} \geq 0.$$

Question 43: (*4 points*) **Proof for $m = 1$:** Show that for $c < 0$ and any $d \in \mathbb{N}^+$, the function $(xy + c)^d$ (with $x, y \in \mathbb{R}$) **fails** to be a valid kernel. (*Note: If you give a single proof covering all m, that is sufficient for full points, see next question.*)

☐0 ☐1 ☐2 ☐3 ■4

Solution: We first consider the 1D case $x, y \in \mathbb{R}$. Then our kernel is

$$k(x, y) = (xy + c)^d,$$

with $c < 0$.

Step A: Exhibit two points violating PSD. We want to show that there exists two points for which the kernel gram matrix is not PSD. We consider the two points $x_1 = 1$ and $x_2 = -1$. The corresponding 2×2 Gram matrix $\mathbf{K}$ is:

$$\mathbf{K} = \begin{pmatrix} k(1,1) & k(1,-1) \\ k(-1,1) & k(-1,-1) \end{pmatrix} = \begin{pmatrix} (1\cdot 1 + c)^d & (1\cdot(-1) + c)^d \\ ((-1)\cdot 1 + c)^d & ((-1)\cdot(-1) + c)^d \end{pmatrix} = \begin{pmatrix} (1+c)^d & (c-1)^d \\ (c-1)^d & (1+c)^d \end{pmatrix}.$$

Step B: Check Positive Semi-Definiteness For $\mathbf{K}$ to be positive semi-definite, it must satisfy the following conditions:

$$(1+c)^d \geq 0 \quad \text{and} \quad \det(\mathbf{K}) \geq 0.$$

Compute the determinant:

$$\det(\mathbf{K}) = (1+c)^{2d} - (c-1)^{2d}.$$

Step C: Evaluate the Determinant Condition Given $c < 0$, we have:

$$|1 + c| < |c - 1|.$$

Thus:

$$(1 + c)^{2d} < (c - 1)^{2d},$$

which implies:

$$\det(\mathbf{K}) = (1 + c)^{2d} - (c - 1)^{2d} < 0.$$

A strictly negative determinant implies that $\mathbf{K}$ is **not** positive semidefinite. Consequently,

$$k(x, y) = (xy + c)^d$$

is **not** a valid kernel in 1D for any $c < 0$.

Question 44: (*2 points*) **Proof for any** m**:** Show that for any arbitrary dimension m, the function $\left(\mathbf{x}^\top \mathbf{x}' + c\right)^d$ (with $c < 0$) **fails** to be a valid kernel.

Solution: To show that

$$k(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^\top \mathbf{x}' + c\right)^d$$

also fails for $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^m$, we simply *embed* the 1D example into $\mathbb{R}^m$.
Consider

$$\mathbf{e}_1 = (1, 0, \ldots, 0) \in \mathbb{R}^m \quad \text{and} \quad -\mathbf{e}_1 = (-1, 0, \ldots, 0).$$

We have

$$\mathbf{e}_1^\top \mathbf{e}_1 = 1, \quad \mathbf{e}_1^\top (-\mathbf{e}_1) = -1, \quad (-\mathbf{e}_1)^\top (-\mathbf{e}_1) = 1.$$

Hence, the 2×2 Gram matrix for the points $\{\mathbf{e}_1, -\mathbf{e}_1\}$ under k is

$$\begin{pmatrix} (1 + c)^d & (c - 1)^d \\ (c - 1)^d & (1 + c)^d \end{pmatrix},$$

the same matrix we had in the 1D case. Since we already proved it is *not* PSD for $c < -1$, it follows that k is *not* a valid kernel in $\mathbb{R}^m$.
Conclusion: For $c < 0$, the function $\left(\mathbf{x}^\top \mathbf{x}' + c\right)^d$ is not a positive semidefinite kernel in any dimension.

4 Neural networks

We consider single-hidden-layer networks with the ReLU activation function. Recall that

$$\mathrm{ReLU}(z) \;=\; \max\{0,\, z\}.$$

A *single-hidden-layer ReLU network* (with no skip connections) taking an input $\mathbf{x} \in \mathbb{R}^d$ and producing a real output can be written as

$$F(\mathbf{x}) \;=\; v^\top \mathrm{ReLU}\big(W\mathbf{x} + b\big) \;+\; c,$$

where $W \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, $v \in \mathbb{R}^m$, and $c \in \mathbb{R}$. The ReLU function is applied elementwise to its argument. The "no skip connections" constraint means $\mathbf{x}$ does not directly feed into the output neuron; all entries of $\mathbf{x}$ must pass through the hidden layer. We wish to study how certain elementary functions can be represented by such networks.

Question 45: (*4 points*) **Implementing the identity function:** consider the function $f : \mathbb{R} \to \mathbb{R}$ defined by

$$f(x) \;=\; x.$$

Show how to construct a single-hidden-layer ReLU network (with no skip connections) that exactly represents $f(x)$. In particular, specify:

- The number of hidden units m.
- The weight matrix $W \in \mathbb{R}^{m\times 1}$ and bias vector $b \in \mathbb{R}^m$ for the hidden layer.
- The vector $v \in \mathbb{R}^m$ and scalar $c \in \mathbb{R}$ for the output unit.

show that for every $x \in \mathbb{R}$, this network's output equals x.

☐0 ☐1 ☐2 ☐3 ■4

Solution: We want a single-hidden-layer ReLU network *without skip connections* (no direct input-to-output connections) that computes

$$f(x) \;=\; x \quad \text{for all real } x.$$

Recall the network form:

$$F(\mathbf{x}) \;=\; v^\top \mathrm{ReLU}\big(W\,\mathbf{x} + b\big) \;+\; c.$$

Here, we have:

$$\mathbf{x} \in \mathbb{R}^1, \quad W \in \mathbb{R}^{m\times 1}, \quad b \in \mathbb{R}^m, \quad v \in \mathbb{R}^m, \quad c \in \mathbb{R}.$$

Construction. Choose $m = 2$ hidden units. Define:

$$W \;=\; \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad b \;=\; \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad v \;=\; \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad c \;=\; 0.$$

Then for an input $x \in \mathbb{R}$, the hidden layer outputs are:

$$\mathrm{ReLU}\big(W\,x + b\big) \;=\; \mathrm{ReLU}\begin{pmatrix} 1\cdot x + 0 \\ -1\cdot x + 0 \end{pmatrix} = \begin{pmatrix} \mathrm{ReLU}(x) \\ \mathrm{ReLU}(-x) \end{pmatrix}.$$

The final output is then

$$F(x) \;=\; v^\top \begin{pmatrix} \mathrm{ReLU}(x) \\ \mathrm{ReLU}(-x) \end{pmatrix} \;+\; c \;=\; \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} \mathrm{ReLU}(x) \\ \mathrm{ReLU}(-x) \end{pmatrix} \;+\; 0.$$

Hence

$$F(x) \;=\; \mathrm{ReLU}(x) \;-\; \mathrm{ReLU}(-x) \;=\; x,$$

because $\mathrm{ReLU}(x) - \mathrm{ReLU}(-x) = x$ for all real x. Thus, this network exactly implements the identity function $f(x) = x$, as required.

Question 46: (*6 points*) **Implementing the maximum function:** consider the function $g : \mathbb{R}^2 \to \mathbb{R}$ defined by:

$$g(x_1, x_2) \;=\; \max\{x_1,\, x_2\}.$$

design a single-hidden-layer ReLU network (again, with no skip connections) whose output is $\max(x_1, x_2)$:

- Specify the number of hidden units m.
- Provide the matrix $W \in \mathbb{R}^{m\times 2}$ and the bias vector $b \in \mathbb{R}^m$.
- Provide the output-layer weight vector $v \in \mathbb{R}^m$ and scalar bias $c \in \mathbb{R}$.
- Show that for every $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$, the network's output equals $\max\{x_1,\, x_2\}$.

Hint: use ideas from the construction in the previous question.

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ■6

Solution: We want a single-hidden-layer ReLU network *without skip connections* (no direct paths from inputs to the output neuron) that computes

$$g(x_1, x_2) \;=\; \max\{x_1,\, x_2\} \quad \text{for all } (x_1, x_2) \in \mathbb{R}^2.$$

We use the fact that

$$\max\{x_1, x_2\} \;=\; x_2 \;+\; \mathrm{ReLU}(x_1 - x_2),$$

but we cannot simply "skip-connect" x_2 to the output. However, we can implement the identity using the construction in the answer to the previous question.

Construction. Let the input be $\mathbf{x} = (x_1, x_2)^\top \in \mathbb{R}^2$. We choose $m = 3$ hidden units. Define:

$$W = \begin{pmatrix} 1 & -1 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad v = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}, \quad c = 0.$$

Then, given $\mathbf{x} = (x_1, x_2)$, the hidden-layer pre-activations are

$$W\mathbf{x} + b = \begin{pmatrix} 1 & -1 \\ 0 & 1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} x_1 - x_2 \\ x_2 \\ -x_2 \end{pmatrix}.$$

Applying ReLU elementwise yields the hidden-layer outputs:

$$\mathrm{ReLU}(W\mathbf{x} + b) = \begin{pmatrix} \mathrm{ReLU}(x_1 - x_2) \\ \mathrm{ReLU}(x_2) \\ \mathrm{ReLU}(-x_2) \end{pmatrix}.$$

Next, the final output of the network is

$$F(x_1, x_2) = v^\top \begin{pmatrix} \mathrm{ReLU}(x_1 - x_2) \\ \mathrm{ReLU}(x_2) \\ \mathrm{ReLU}(-x_2) \end{pmatrix} + c = \begin{pmatrix} 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} \mathrm{ReLU}(x_1 - x_2) \\ \mathrm{ReLU}(x_2) \\ \mathrm{ReLU}(-x_2) \end{pmatrix}.$$

Because $\mathrm{ReLU}(x_2) - \mathrm{ReLU}(-x_2) = x_2$ and we keep the first hidden unit as $\mathrm{ReLU}(x_1 - x_2)$, it follows that

$$F(x_1, x_2) = \mathrm{ReLU}(x_1 - x_2) + \big[\mathrm{ReLU}(x_2) - \mathrm{ReLU}(-x_2)\big] = \mathrm{ReLU}(x_1 - x_2) + x_2.$$

This quantity is exactly

$$x_2 + \max\{0, x_1 - x_2\} = \max\{x_1, x_2\}.$$

Hence the network computes $\max(x_1, x_2)$ with no skip connections.