

University of Toronto

ECE421: Introduction to Machine Learning, Fall 2024,

Midterm Exam: Oct. 21, 9:10 AM

Duration: 110 minutes

Aids: No aid-sheet is permitted. Calculator type 2 (*i.e.*, non-programmable electronic calculator) is allowed.

DO NOT TURN THIS PAGE UNTIL YOU ARE TOLD TO DO SO

- The University of Toronto and you, as a student, share a commitment to academic integrity. You are reminded that you may be charged with an academic offence for possessing any unauthorized aids during the writing of an exam. Clear, sealable, plastic bags have been provided for all electronic devices with storage, including but not limited to: cell phones, smart watches, SMART devices, tablets, laptops, and calculators. Please turn off all devices, seal them in the bag provided, and place the bag under your desk for the duration of the examination. You will not be able to touch the bag or its contents until the exam is over. If, during an exam, any of these items are found on your person or in the area of your desk other than in the clear, sealable, plastic bag, you may be charged with an academic offence. A typical penalty for an academic offence may cause you to fail the course.
- There are **6 questions** and **16 pages** in this exam, including this one. When you receive the signal to start, please make sure that your copy of the examination is complete.
- Answer each question directly on the examination paper, in the space provided. Please note that the provided space does not necessarily reflect the expected length of your answer.
- This exam includes two extra pages (*i.e.* page 14 and 15) for your scratch notes or in case you need extra space for any question, which must be submitted. Do not remove any page, including the scratch note page or the formula sheet page from your exam booklet.

Q1 [10 pts] Short Answer Questions

1.a [2 pts] A majority voting k-nearest neighbour classifier is applied using 100 training points belonging to two classes. Class 1 has 80 training points and class 2 has 20 training points. Every data point has a unique 2-dimensional feature vector \underline{x} .

1.a.i [1 pts] What is the classification rate on training data, when $k = 1$? The classification rate is defined as the percentage of correct predictions. (Check one.)

- ☐ 20% ☐ 80% ☐ 100% ☐ The classification rate depends on dataset.

1.a.ii [1 pts] What is the classification rate on training data when $k = 50$? (Check one.)

- ☐ 20% ☐ 80% ☐ 100% ☐ The classification rate depends on dataset.

1.b [4 pts] Circle True or False for each of the following statements.

1.b.i [1 pts] Consider a k-nearest neighbours classification with N training points. For small values of k , the model is underfitting. **True / False**

1.b.ii [1 pts] Stochastic gradient descent is better than batch gradient descent because it provides a more accurate estimate of the gradient of the loss function for the entire training dataset. **True / False**

1.b.iii [1 pts] After each iteration of batch gradient descent, we modify the weight vector in the opposite direction of the gradient. **True / False**

1.b.iv [1 pts] In neural networks, increasing the number of hidden units usually reduces the training error. **True / False**

1.c [1 pts] Which one of the following activation functions is **not** suitable for the backpropagation learning procedure? (Check one.)

- ☐ $\frac{1}{1+\exp(-z)}$.
☐ $\tanh(z)$.
☐ $\mathbb{I}(z > 0)$, where $\mathbb{I}(\cdot)$ is the indicator function.
☐ $\mathbb{I}(z > 0)z$, where $\mathbb{I}(\cdot)$ is the indicator function.

1.d [1 pts] Consider a perceptron learning algorithm applied to three training examples, \underline{x}_1 , \underline{x}_2 and \underline{x}_3 , in that order. Assume that \underline{x}_2 and \underline{x}_3 are identical. If \underline{x}_2 is misclassified and the algorithm updates the weight vector based on \underline{x}_2 , will the model correctly classify \underline{x}_3 afterward?

- ☐ Yes ☐ No ☐ Unknown

1.e [2 pts] Plot a typical graph of both the training and test error (y-axis) versus the number of iterations of stochastic gradient descent (x-axis). Your plot should have two curves.

[Write your answer to 1.e here.]

Q2 [20 pts] In this question, you will train a regularized linear regression model with an ℓ_2 regularization penalty. We are given the following training dataset:

$$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3)\},$$

where $x_1 = -1$, $x_2 = 0$, $x_3 = 0$, with associated labels $y_1 = 0$, $y_2 = -2$, and $y_3 = 4$. We wish to fit a linear model $\hat{y} = w_0 + w_1x$ by minimizing the regularized loss

$$J(\underline{w}, \lambda) = \sum_{i=1}^3 (y_i - \hat{y}_i)^2 + \lambda \|\underline{w}\|^2,$$

where $\underline{w} = (w_0, w_1)$.

2.a [15 pts] Find the optimal \underline{w} in terms of λ . To receive full credit, you must obtain simplified expressions in λ .

[**HINT:** What is the gradient of $J(\underline{w}, \lambda)$ with respect to \underline{w} ?]

[Write your answer to 2.a here.]

2.b [5 pts] Suppose we need to make a choice of λ from the set $\{0, 1, 2, \infty\}$. We only have a single validation data point $(x_4, y_4) = (2, 3)$. Which λ value should be chosen.

[**HINT:** We wish to minimize the squared error.]

[Write your answer to **2.b** here.]

Q3 [15 pts] In this problem, we will look at classification in one dimension.

3.a [10 pts] Recall that in binary classification, the goal is to learn a weight vector $\underline{w} \in \mathbb{R}^d$ in order to predict the output label $y \in \{-1, +1\}$ given input x (which is mapped to a feature vector $\phi(x) \in \mathbb{R}^d$). Define the following loss function:

$$e_n(\underline{w}) = \max\{2 - y_n \underline{w}^\top \phi(x_n), 0\}$$

Consider the following training set of (x_n, y_n) pairs:

$$(x_1, y_1) = (-4, +1), \quad (x_2, y_2) = (1, -1), \quad (x_3, y_3) = (0, +1).$$

Suppose the feature transformation $\phi(x) = (1, x)$. Assume that the stochastic gradient descent algorithm is initialized with $\underline{w} = (0, 0)$ and loops through each example (x_n, y_n) and performs an update. Assume that the learning rate is fixed and set to 1. Compute the weight vector \underline{w} after updating on example 1, example 2, and example 3 (**fill out the table below and show your process**):

| | x | $\phi(x)$ | y | weight \underline{w} |
|-----------------|-----|-----------|-----|------------------------|
| Initialization | n/a | n/a | n/a | $[0, 0]$ |
| After example 1 | -4 | $(1, -4)$ | +1 | |
| After example 2 | 1 | $(1, 1)$ | -1 | |
| After example 3 | 0 | $(1, -0)$ | +1 | |

[Write your answer to 3.a here. Show the process of updating the weights vector for each example.]

[More space for your answer to 3.a.]

3.b [5 pts] What does it mean to have $e_n(\underline{w}) = 0$ for some given $\underline{w} = (w_0, w_1)$ with $w_1 > 0$? You should give a geometric interpretation in x -space.

[What we expect: Clearly specify the region(s) in which any point with true label +1 would have $e_n(\underline{w}) = 0$, and clearly specify the region(s) in which any point with true label -1 would have $e_n(\underline{w}) = 0$.]

[Write your answer to 3.b here.]

Q4 [5 pts] Erfan is trying to do classification: $y_n \in \{-1, +1\}$ and $x_n \in \mathbb{R}$. But he slept through all of the classification lectures, so he decides to solve classification using regression. That is, he ignores the fact that y_n is binary, and fits a linear regression function via least squares. The resulting regression function is:

$$h_{\underline{w}}(x) = w_0 + xw_1$$

Erfan uses the decision rule:

$$\hat{y} = \begin{cases} +1, & \text{if } h_{\underline{w}}(x) > 1/2, \\ -1, & \text{otherwise.} \end{cases}$$

Suppose the training data is linearly separable. Is Erfan's decision rule (with associated regression function) guaranteed to classify the training data without error?

If your answer is yes, provide a short argument. If no, provide a counterexample.

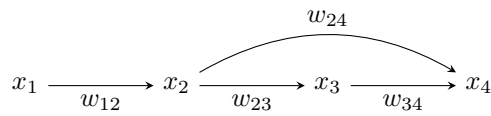
[Write your answer to **Q4** here.]

[1 pts] Your answer: ☐ Yes ☐ No

[4 pts] Justification (*i.e.*, a short argument or a counterexample):

Q5 [15 pts] In this problem, we will look at backpropagation in neural network.

5.a [5 pts] Consider the following neural network with input x_1 , output x_4 , and error $E = f(x_4)$.



The relationship between x 's are as follows:

$$x_2 = w_{12}x_1, \quad x_3 = w_{23}g(x_2), \quad x_4 = w_{24}g(x_2) + w_{34}g(x_3)$$

Show that the error derivative for w_{12} is

$$\frac{\partial E}{\partial w_{12}} = f'(x_4)[w_{24}g'(x_2) + w_{34}g'(x_3)w_{23}g'(x_2)]x_1$$

[**HINT:** It is simpler to just use the above formulas for E , x_2 , x_3 , and x_4 than to apply the general backprop formula.]

[Write your answer to 5.a here.]

5.b [10 pts] In class, for x 's that are pre-activation function values, we saw that

- the forward propagation recursion is $x_j = \sum_{i=1}^{j-1} w_{ij}g(x_i)$,
- the error derivative for an output unit is $\frac{\partial E}{\partial x_m} = -2(y_m - g(x_m))g'(x_m)$,
- the backpropagation recursion is $\frac{\partial E}{\partial x_m} = \sum_{k=m+1}^M \frac{\partial E}{\partial x_k} w_{mk}g'(x_m)$,
- and the derivative for a weight is $\frac{\partial E}{\partial w_{lm}} = \frac{\partial E}{\partial x_m} g(x_l)$.

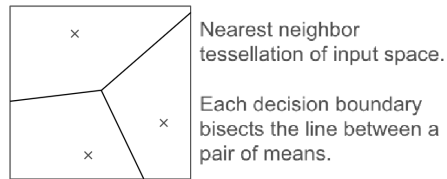
Let the post-activation function value be $\hat{x}_i = g(x_i)$ for all i . Derive expressions for \hat{x}_j , $\frac{\partial E}{\partial \hat{x}_m}$ (for m being an output unit and for m being a hidden unit), and $\frac{\partial E}{\partial w_{lm}}$ in terms of \hat{x} 's, w 's, $g(\cdot)$, and $g'(\cdot)$.

[Write your answer to 5.b here.]

Q6 [26 pts] In this problem, we will look at Mixture of Gaussians (MoG) and K-Means Clustering.

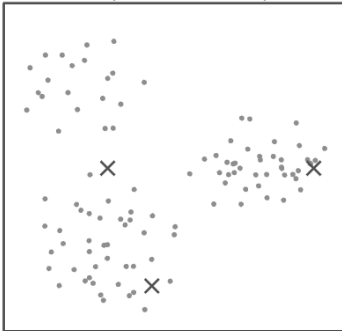
6.a [8 pts] For the dataset below and the initial means as shown, show how k-means clustering will proceed. In panel 1, draw the tessellation (non-overlapping polygons) that delineates which points belong to which cluster. In panel 2, redraw that tessellation and show the updated means. Repeat this for panels (3,4), (5,6), and (7,8).

[HINT: below is an example of a tessellation for 3 means.]



[Write your answer to 6.a here.]

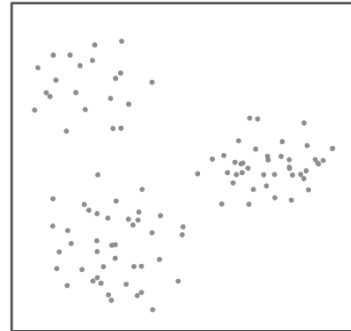
Panel 0 (initialization)



Panel 1



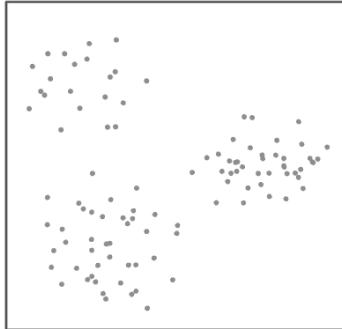
Panel 2



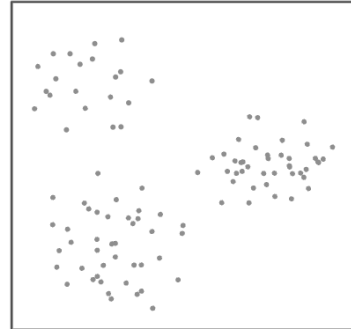
Panel 3



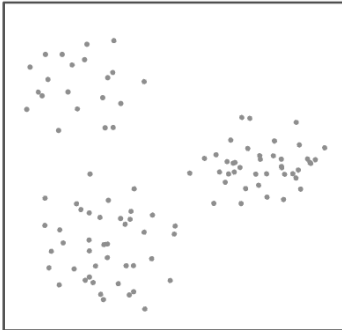
Panel 4



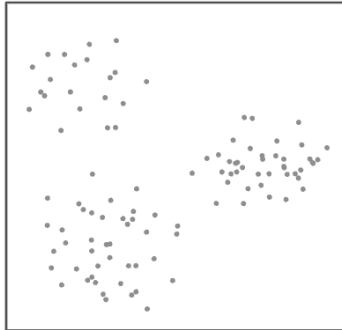
Panel 5



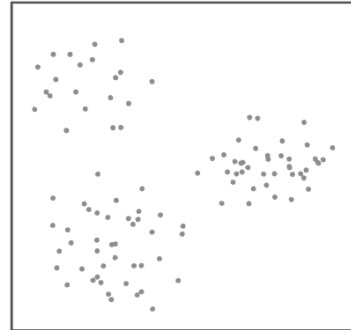
Panel 6



Panel 7



Panel 8



6.b [10 pts] Show that in an MoG, if for each datapoint x_n there is one component j such that

$$\pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j) \gg \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k), \quad \forall k \neq j,$$

then the negative log likelihood of the data can be written

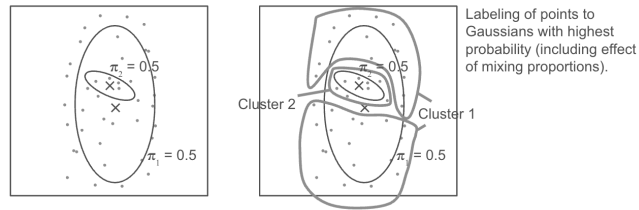
$$\sum_{n=1}^N \sum_{k=1}^K r_{nk} \left[-\log(\pi_k) + \frac{1}{2} \log(2\pi\sigma_k^2) + \frac{(x_n - \mu_k)^2}{2\sigma_k^2} \right],$$

where r_n is the one-hot encoding of the component with dominant probability (j above).

[Write your answer to **6.b** here.]

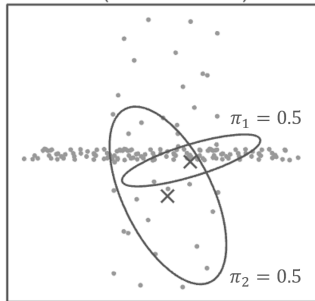
6.c [8 pts] For the dataset shown below and the initial means and covariance (drawn as ellipses) for a MoG, show how hard-assignment learning of the MoG proceeds. In panel 1, draw loops around sets of points to indicate which of the two components they're in and label each loop appropriately. In panel 2, show the updated means and covariances (as ellipses) based on the assignments in panel 1, and also provide rough estimates (e.g., one decimal place) of the mixing proportions. Repeat this (assignments and updates) in panels (4,5), (6,7) and (8,9).

[HINT: Below is an example of a MoG and labelling of points.]

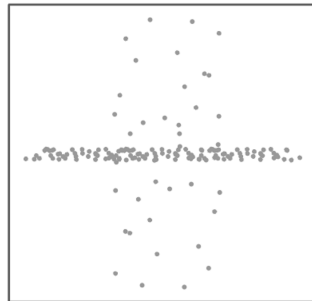


[Write your answer to 6.c here.]

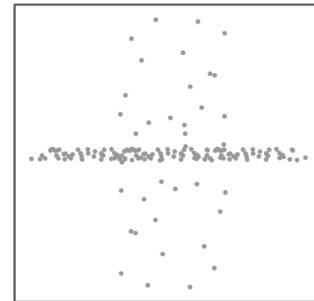
Panel 0 (initialization)



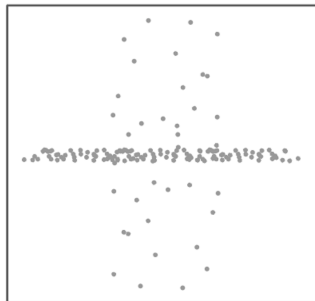
Panel 1



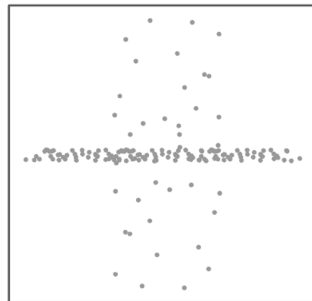
Panel 2



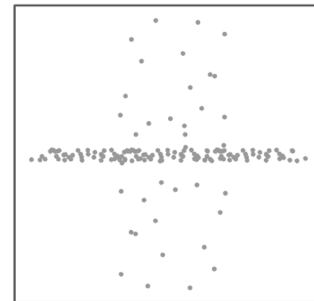
Panel 3



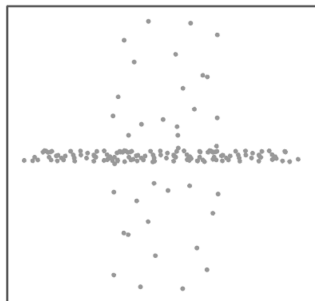
Panel 4



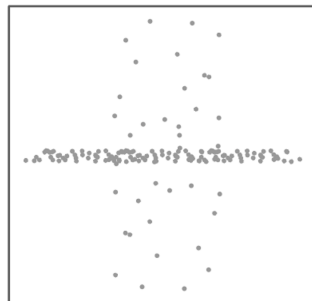
Panel 5



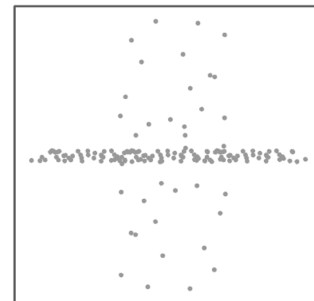
Panel 6



Panel 7



Panel 8

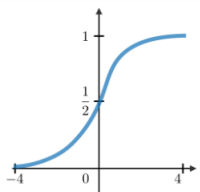
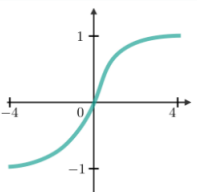
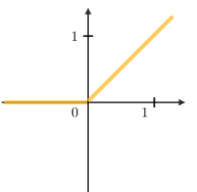
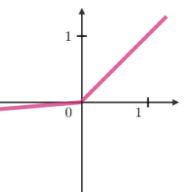


[This page is intentionally left blank. You may use it for your scratch notes or extra space for your answers.]

[This page is intentionally left blank. You may use it for your scratch notes or extra space for your answers.]

Formula Sheet

- Most common activation functions:

| Sigmoid | Tanh | ReLU | Leaky ReLU |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| $g(z) = \frac{1}{1 + e^{-z}}$ | $g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ | $g(z) = \max(0, z)$ | $g(z) = \max(\epsilon z, z)$ with $\epsilon \ll 1$ |
|  |  |  |  |

- Matrix/vector manipulation

| Rule | Comments |
|-------------------------------------------------------------------------------|-------------------------------------------------------------------|
| $(AB)^T = B^T A^T$ | order is reversed, everything is transposed |
| $(\underline{a}^T B \underline{c})^T = \underline{c}^T B^T \underline{a}$ | as above |
| $\underline{a}^T \underline{c} = \underline{c}^T \underline{a}$ | (the result is a scalar, and the transpose of a scalar is itself) |
| $(A + B)C = AC + BC$ | multiplication is distributive |
| $(\underline{a} + \underline{b})^T C = \underline{a}^T C + \underline{b}^T C$ | as above, with vectors |
| $AB \neq BA$ | multiplication is not commutative |

- Common vector derivatives

| | | | | |
|-------------------------------------------|---------------------------------|---------------------------------|---------------------------------|-----------------------------------|
| $f(\underline{x})$ | $\underline{x}^T \underline{a}$ | $\underline{a}^T \underline{x}$ | $\underline{x}^T \underline{x}$ | $\underline{x}^T A \underline{x}$ |
| $\nabla_{\underline{x}} f(\underline{x})$ | \underline{a} | \underline{a} | $2\underline{x}$ | $(A^T + A)\underline{x}$ |

- Probability density function of a normal distribution with mean $\underline{\mu}$ and covariance matrix Σ :

$$\mathcal{N}(\underline{x} | \underline{\mu}, \Sigma) = \frac{1}{|2\pi\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\underline{x} - \underline{\mu})^T \Sigma^{-1}(\underline{x} - \underline{\mu})\right)$$

Note: For scalar random variable with normal distribution with mean μ and variance σ^2 :

$$\mathcal{N}(x | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$