

AY: 2025-2026  
MIDTERM | AI-ECUE322  
Nov. 2025

M2-S3: Dept. of Electrical Engineering  
Teacher: A. Mhamdi  
Time Limit: 1h

This document contains 7 pages numbered from 1 to 7. Upon receiving it, verify completeness. The 2 tasks are independent and can be solved in any order you prefer. The following rules apply:

- ❶ A handwritten double-sided A4 sheet is permitted.
- ❷ Any electronic material, except basic calculator, is prohibited.
- ❸ Mysterious or unsupported answers will not receive full credit.
- ❹ Round results to the nearest thousandth (i.e., the third digit after the decimal point).
- ❺ Task N<sup>o</sup>2: Correct answers earn points as indicated. There is no negative scoring.

### Task N<sup>o</sup>1

⌚ 35mn | (8 points)

Complete the following VAE code skeleton by implementing all the missing parts. Use identity activation functions throughout.

SETUP: DATA AND HYPERPARAMETERS

```
1 X = [0.5 0.3; 0.6 0.4; 0.1 0.2; 0.7 0.8]' # 2x4 matrix
2 batch_size = size(X, 2)
3 input_dim = size(X, 1)
4 latent_dim = 2
```



- (a) (1 point) Initialize the encoder weights  $\mathbf{W}_e^1$  and bias  $\mathbf{b}_e$  to map input to latent distribution parameters.

- $\mathbf{X} \in \mathbb{R}^{2 \times 4}$  (2 features, 4 samples)
- $\text{batch\_size} = 4$
- $\text{input\_dim} = 2$

<sup>1</sup>Initialize with small random values:  $\text{randn}(\dots) \times 0.01$ .

- $\mathbf{W}_e \in \mathbb{R}^{4 \times 2}$  maps input dimension 2 to output dimension 4
- Output contains:  $[\mu_1, \mu_2, \log(\sigma_1^2), \log(\sigma_2^2)]$
- $\mathbf{b}_e \in \mathbb{R}^4$  is the bias vector

```
1 We = randn(2*latent_dim, input_dim) .* 0.01
2 be = zeros(2*latent_dim)
```

- (b) (1 point) Compute encoder output `z_params` and extract mean `mu` and log-variance `log_var` parameters.

The code in the latent space is:

$$\mathbf{z}_{\text{params}} = \mathbf{W}_e \mathbf{X} + \mathbf{b}_e$$

$$\boldsymbol{\mu} = \mathbf{z}_{\text{params}}[1 : \text{latent\_dim}, :]$$

$$\log(\boldsymbol{\sigma}^2) = \mathbf{z}_{\text{params}}[\text{latent\_dim} + 1 : \text{end}, :]$$

$$\boldsymbol{\sigma} = \exp\left(\frac{\log(\boldsymbol{\sigma}^2)}{2}\right)$$

```
1 z_params = We * X .+ be
2 mu = z_params[1:latent_dim, :]
3 log_var = z_params[latent_dim+1:end, :]
```

- (c) (1 point) Sample from the latent distribution using the reparameterization trick to enable gradient flow through stochastic nodes.

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon} \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \quad \boldsymbol{\epsilon} \in \mathbb{R}^{\text{latent\_dim} \times \text{batch\_size}}$$

$\odot$  denotes element-wise multiplication.

```
1 epsilon = randn(latent_dim, batch_size)
2 z = @. mu + exp(0.5 * log_var) * epsilon
```

- (d) (1 point) Initialize decoder weights and bias to map latent space back to input space.

- $\mathbf{W}_d \in \mathbb{R}^{2 \times 2}$  (latent to input)
- $\mathbf{b}_d \in \mathbb{R}^2$

```
1 Wd = randn(input_dim, latent_dim) .* 0.01
2 bd = zeros(input_dim)
```

- (e) (1 point) Reconstruct the input  $\mathbf{x}_{\text{recon}}$  from latent codes.

$$\mathbf{x}_{\text{recon}} = \mathbf{W}_d \mathbf{z} + \mathbf{b}_d$$

```
1 X_recon = Wd * z .+ bd
```

- (f) (1 point) Compute the Mean Squared Error between input and reconstruction.

The **MSE** is given by:

$$\mathcal{L}_{\text{recon}} = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{x}^{(i)} - \mathbf{x}_{\text{recon}}^{(i)} \right\|^2$$

```
1 using Statistics
2 reconstruction_loss = mean(@. (X - X_recon)^2 )
```

- (g) (1 point) Compute the **KL** divergence between the learned posterior and the standard normal prior.

The Kullback-Leibler divergence between  $q(\mathbf{z}|\mathbf{x})$  (posterior) and  $p(\mathbf{z})$  (prior) is:

$$\begin{aligned} \mathcal{L}_{\text{KL}} &= \mathcal{D}_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})) \\ &= \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{q(\mathbf{z}|\mathbf{x}^{(i)})} \left[ \log q(\mathbf{z}|\mathbf{x}^{(i)}) - \log p(\mathbf{z}) \right] \\ &= \frac{1}{N} \sum_{i=1}^N \frac{1}{2} \left( \|\boldsymbol{\mu}^{(i)}\|^2 + \|\boldsymbol{\sigma}^{(i)}\|^2 - \log \left( \left( \boldsymbol{\sigma}^{(i)} \right)^2 \right) - 1 \right) \end{aligned}$$

```
1 kl_loss = .5 * mean(@. (mu^2+exp(0.5*log_var)^2-log_var-1))
```

- (h) (1 point) Combine reconstruction and **KL** losses to form the Evidence Lower Bound (**ELBO**).

The loss is given by:

$$\mathcal{L}_{\text{VAE}} = -\text{ELBO} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{KL}}$$

This objective balances reconstruction accuracy with latent space regularization.

```
1 vae_loss = reconstruction_loss + kl_loss
```

AY: 2025-2026

Full Name: .....

M2-S3: Dept. of Electrical Engineering

ID: .....

MIDTERM | AI-ECUE322

Class: RAIA2.....

Nov. 2025

Room: .....

Teacher: A. Mhamdi

Time Limit: 1h

✂

**ANSWER SHEET**

**Task N°2**

⌚ 25mn | (12 points)

- (a) ( $\frac{1}{2}$  point) What is the correct way to define a parametric type in Julia?
- ☐ struct Point{T>    ✓ struct Point{T}    ☐ struct Point[T]
- (b) ( $\frac{1}{2}$  point) What happens when you call a function with arguments that don't match any defined method?
- ☐ Julia automatically converts the types
- ☐ The function returns nothing
- ✓ Julia throws a MethodError
- ☐ Julia uses the most general method available
- (c) ( $\frac{1}{2}$  point) How do you define a method that accepts any number of arguments in Julia?
- ☐ function f(\*args)    ☐ function f(...args)    ✓ function f(args...)
- (d) ( $\frac{1}{2}$  point) Which package provides GPU support for Julia machine learning?
- ✓ CUDA.jl    ☐ CuArrays.jl    ☐ GPUArrays.jl
- (e) ( $\frac{1}{2}$  point) What is the correct syntax for moving a Flux model to GPU?
- ✓ model |> gpu    ✓ gpu(model)    ☐ model.to\_gpu()
- (f) ( $\frac{1}{2}$  point) How do you save a trained Flux model?
- ✓ BSON.@save "model.bson" model
- ✓ JLD2.@save "model.jld2" model
- ☐ Serialization.@save "model.jls" model
- ☐ None of the above
- (g) ( $\frac{1}{2}$  point) In Flux.jl, what does the trainable() function return?
- ☐ Model architecture
- ☐ Hyperparameters
- ☐ Model performance metrics

✂

✓ Trainable parameters of the model

(h) ( $\frac{1}{2}$  point) In information theory, what does the self-information of an event represent?

- ☐ The average information content across all events in the sample space
- ☐ The total number of possible outcomes in a sample space
- ✓ The amount of information revealed when the event occurs
- ☐ The correlation between two independent events

(i) ( $\frac{1}{2}$  point) Shannon entropy measures what property of a probability distribution?

- ☐ The maximum likelihood estimate of a parameter
- ☐ The distance between two probability distributions
- ✓ The expected number of bits needed to encode outcomes from a distribution
- ☐ The covariance between two random variables

(j) ( $\frac{1}{2}$  point) Given a true distribution  $P$  and an estimated distribution  $Q$ , the KL divergence can be related to cross-entropy by which relationship?

- ✓  $D_{KL}(P||Q) = H(P, Q) - H(P)$
- ☐  $D_{KL}(P||Q) = H(P) - H(Q)$
- ☐  $D_{KL}(P||Q) = H(P, Q) + H(P)$
- ☐  $D_{KL}(P||Q) = \log(H(P, Q))$

(k) ( $\frac{1}{2}$  point) The loss function of a **VAE** typically combines two terms. What do these two terms represent?

- ☐ Classification accuracy and reconstruction error
- ☐ Cross-entropy and squared error
- ✓ Reconstruction loss and KL divergence between the learned latent distribution and prior distribution
- ☐ Mutual information and conditional entropy

(l) ( $\frac{1}{2}$  point) In a **VAE**, why is the KL divergence term crucial for learning a meaningful latent representation?

- ☐ It ensures the encoder network converges faster during training
- ☐ It measures the accuracy of the reconstruction loss
- ☐ It computes the mutual information between input and latent variables
- ✓ It forces the posterior to match the prior distribution



- (m) ( $\frac{1}{2}$  point) What does the command 'git init' do?
- ☒ Initializes a new Git repository in the current directory
  - ☐ Clones an existing repository
  - ☐ Adds files to the staging area
  - ☐ Commits changes to the remote repository
- (n) ( $\frac{1}{2}$  point) Which command is used to view the commit history in Git?
- ☐ git status   ☐ git diff   ☒ git log   ☐ git branch
- (o) ( $\frac{1}{2}$  point) What is the purpose of 'git add'?
- ☐ To create a new branch
  - ☒ To stage changes for commit
  - ☐ To push changes to remote
  - ☐ To merge branches
- (p) ( $\frac{1}{2}$  point) In Git, what does a "branch" represent?
- ☒ A pointer to a commit
  - ☐ A separate copy of the entire repository
  - ☐ A file in the working directory
  - ☐ A remote server connection
- (q) ( $\frac{1}{2}$  point) Which command creates a new branch?
- ☒ git checkout -b newbranch
  - ☐ git merge newbranch
  - ☐ git pull newbranch
  - ☐ git rebase newbranch
- (r) ( $\frac{1}{2}$  point) What does 'git clone' do?
- ☐ Deletes a local repository
  - ☐ Updates the remote repository
  - ☐ Resets the current branch
  - ☒ Copies a remote repository to your local machine
- (s) ( $\frac{1}{2}$  point) What does 'git pull' do?
- ☐ Only fetches changes without merging

DO NOT WRITE ANYTHING HERE

✂

- ✓ Fetches and merges changes from a remote repository
  - ☐ Pushes local changes to remote
  - ☐ Stashes uncommitted changes
- (t) ( $\frac{1}{2}$  point) What is **Docker**?
- ✓ A platform for developing, shipping, and running applications inside containers
  - ☐ A version control system for code
  - ☐ A cloud storage service
  - ☐ A programming language for web development
- (u) ( $\frac{1}{2}$  point) What is a **Docker** image?
- ☐ A running instance of a container
  - ☐ A configuration file for **Docker Compose**
  - ✓ A lightweight, executable package that includes everything needed to run a piece of software
  - ☐ A command-line tool for managing containers
- (v) ( $\frac{1}{2}$  point) Which command is used to build a **Docker** image from a Dockerfile?
- ✓ **docker build**   ☐ docker run   ☐ docker push   ☐ docker pull
- (w) ( $\frac{1}{2}$  point) What does 'docker run' do?
- ✓ Starts a new container from an image
  - ☐ Stops a running container
  - ☐ Lists all images on the system
  - ☐ Removes unused images
- (x) ( $\frac{1}{2}$  point) In **Docker**, what is a container?
- ✓ A runnable instance of an image
  - ☐ A template for building images
  - ☐ A network bridge between containers
  - ☐ A storage volume for data persistence