INSTITUTE OF TECHNOLOGICAL STUDIES OF BIZERTE

AY: 2025-2026	**M2-S3: Dept. of Electrical Engineering**
EXAM **| AI-ECUE322**	**Teacher: A. Mhamdi**
Jan. 2026	**Time Limit: 1½ h**

This document contains **10** pages numbered from **1** to **10**. Upon receiving it, verify completeness. The **3** 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°3:** Correct answers earn points as indicated. There is no negative scoring.

Task N°1

⌛ 25mn | (4 points)

GANs can generate entirely new, realistic data from scratch. A GAN has two networks:

Generator: Makes fake data;

Discriminator: Guesses if data is real or fake.

```
using Flux
using Random
using Statistics

# `real_data(n)` generates random points around (2, 2) or (-2, -2)
```

The Julia implementation of `real_data` function is hidden for brevity.

(a) (1 point) Create a neural network that takes random noise (2 values) and transforms it into fake data (2 output values). Use a hidden layer with 16 units[1].

```
Generator = Chain(
    Dense(2, 16, relu),
    Dense(16, 2)
)
```

[1] Use ReLU for activation functions in hidden layers.

(b) (1 point) Create a neural network that takes data (2 values) and outputs a single probability between 0 and 1. It should output close to 1 if the data looks real, close to 0 if it looks fake.

```
Discriminator = Chain(
    Dense(2, 16, relu),
    Dense(16, 1, sigmoid)
)
```

(c) (1 point) Write two loss functions that measure how well each network is doing. The generator wants to fool the discriminator, and the discriminator wants to correctly identify real vs fake.

```
function loss_gen(disc, gen, noise)
  # TODO: Discriminator should output close to 1
end
```

Generator loss - make discriminator think fake is real:

```
function loss_gen(disc, gen, noise)
    -mean(log.(disc(gen(noise)) .+ 1e-8))
end
```

```
function loss_disc(disc, real, fake)
  # TODO: Real should output ~1, fake should output ~0
end
```

Discriminator loss - correctly classify real and fake:

```
function loss_disc(disc, real, fake)
    -mean(log.(disc(real) .+ 1e-8)) - mean(log.(1 .- disc(fake) .+ 1e-8))
end
```

(d) (1 point) Implement the competitive training: update the discriminator to get better at detecting fakes, then update the generator to create more convincing fakes. Repeat this back-and-forth for multiple epochs.

```julia
opt_g = ADAM(0.0002)
opt_d = ADAM(0.0001)

st_g = Flux.setup(opt_g, Generator)
st_d = Flux.setup(opt_d, Discriminator)

for epoch in 1:100
    # Get real data
    x_real = real_data(32)

    # Generate fake data
    noise = randn(Float32, 2, 32)
    x_fake = Generator(noise)

    # TODO: Update discriminator
    # 1. Calculate loss_disc(Discriminator, x_real, x_fake)
    # 2. Use gradient() and update!()

    # TODO: Update generator
    # 1. Generate new noise
    # 2. Calculate loss_gen(Discriminator, Generator, noise)
    # 3. Use gradient() and update!()

end
```

```julia
# Update discriminator
ld(m) = loss_disc(m, x_real, x_fake)
grad_d = Flux.gradient(ld, Discriminator)
Flux.update!(st_d, Discriminator, grad_d[1])

# Update generator
noise = randn(Float32, 2, 32)
lg(m) = loss_gen(Discriminator, m, noise)
grad_g = gradient(lg, Generator)
Flux.update!(st_g, Generator, grad_g[1])
```

Consider the following 3x3 grid world:

0 S	1	2 G
3	X	5
6	7	8

- **S** = Start state (state 0)
- **G** = Goal state (state 2), gives reward $R = +5$
- **X** = Obstacle (state 4, cannot be entered)
- Empty cells give reward $R = 0$ per step
- **Actions:** Up, Down, Left, Right
- **Discount factor:** $\gamma = 0.8$
- Attempting to move into a wall or boundary keeps the agent in place

(a) **Deterministic Setting:** Actions always succeed as intended with probability 1.

i. ($\frac{1}{2}$ point) What is $R(0, \text{Right})$?

$R(0, \text{Right}) = \boxed{0}$ (cost of moving)

ii. ($\frac{1}{2}$ point) What is $R(1, \text{Right})$ (moving from state 1 to the goal)?

$R(1, \text{Right}) = \boxed{+5}$

iii. ($\frac{1}{2}$ point) What is $V(2)$ for the goal state?

The value function for state 2 is:

$$V(2) = \boxed{0}$$

iv. ($\frac{1}{2}$ point) Assume the optimal policy is to always move toward the goal via the shortest path. Compute $V(1)$ for state 1.

The value function for state 1 is:

$$\begin{aligned} V(1) &= R(1, \text{Right}) + \gamma V(2) \\ &= 5 + 0.8 \times 0 \\ &= 5.0 \end{aligned}$$

v. (½ point) Compute $V(0)$ for the start state.

$$\begin{aligned} V(0) &= R(0, \text{Right}) + \gamma \left(R(1, \text{Right}) + \gamma V(2) \right) \\ &= 0 + 0.8(5 + 0.8 \times 0) \\ &= 4.0 \end{aligned}$$

Alternatively: $V(0) = 0 + 0.8 \times V(1) = 0 + 0.8 \times 5 = 4.0$

vi. (½ point) Compute $Q(0, \text{Right})$ from state 0.

From state 0, upon moving right:

$$\begin{aligned} Q(0, \text{Right}) &= R(0, \text{Right}) + \gamma V(1) \\ &= 0 + 0.8 \times 5.0 \\ &= 4.0 \end{aligned}$$

vii. (½ point) Compute $Q(0, \text{Down})$ from state 0.

When in state 0 and the action “Down” is taken:

$$\begin{aligned} Q(0, \text{Down}) &= R(0, \text{Down}) + \gamma V(3) \\ &= 0 + 0.8 \times (0 + 0.8 \times 4.0) \\ &= 2.56 \end{aligned}$$

viii. (½ point) Which action has the highest Q-value from state 0?

$\boxed{\text{Right}}$ has the highest Q-value $\boxed{4.0}$.

(b) **Stochastic Setting:** Actions succeed with probability 0.7, but the agent slips perpendicular to the intended direction with probability 0.15 each.

i. (½ point) Assume the following state values: $V(0) = 4.0$, $V(1) = 5.0$, $V(3) =$

2.56. When taking action Right from state 0, list all possible outcomes and their probabilities.

From state 0, action Right:

- 70%: agent successfully moves Right $\longrightarrow$ state 1
- 15%: slips Up $\longrightarrow$ hit wall, stays at state 0
- 15%: slips Down $\longrightarrow$ state 3

ii. (½ point) Compute $Q(0, \text{Right})$.

The expected cumulative reward by moving right when in state 0:

$$\begin{aligned} Q(0, \text{Right}) &= R + \gamma \sum_{s'} P(s'|s, a)V(s') \\ &= 0 + 0.8 \times [0.7 \times V(1) + 0.15 \times V(0) + 0.15 \times V(3)] \\ &= 0 + 0.8 \times [0.7 \times 5.0 + 0.15 \times 4.0 + 0.15 \times 2.56] \\ &= 3.587 \end{aligned}$$

iii. (½ point) If $Q(0, \text{Right})$ is the highest Q-value at state 0, what is $V(0)$ under the optimal policy?

Under the optimal policy:

$$V(0) = \max_{a \in \mathcal{A}} Q(0, a) = \boxed{3.587}$$

iv. (½ point) How does $V(0)$ in the stochastic setting compare to $V(0)$ in the deterministic setting? Why?

In the deterministic setting, $V(0) = 4.0$, while in the stochastic setting, $V(0) = 3.587$.

The stochastic setting has a **lower value** because:

- Uncertainty reduces expected returns
- The agent may slip into less favorable states
- There's a 15% chance of not making progress toward the goal

AY: 2025-2026 | Full Name: ....
M2-S3: Dept. of Electrical Engineering | ID: ....
Exam | AI-ECUE322 | Class: RAIA2 ....
Jan. 2026 | Room: ....
Teacher: A. Mhamdi | Time Limit: 1½ h

Answer Sheet

Task N°3

35mn | (10 points)

(a) (½ point) Which package provides reinforcement learning capabilities in `Julia`?

- ◯ `MLJ`
- ◯ `Flux`
- ◯ `Zygote`
- √ `ReinforcementLearning`

(b) (½ point) What is the primary advantage of CNNs over traditional ANNs when processing image data?

- √ CNNs use convolutional layers with shared weights for efficiency
- ◯ CNNs have more layers than ANNs, allowing them to learn deeper patterns
- ◯ CNNs eliminate the need for data preprocessing entirely
- ◯ CNNs can only work with image data while ANNs work with any data type

(c) (½ point) What is the primary purpose of the latent space in a VAE?

- ◯ To store the original input data without any transformation
- ◯ To directly classify data into predefined categories
- √ To learn probabilistic representation for generation and interpolation
- ◯ To reduce computational requirements by removing unnecessary layers

(d) (½ point) What does the ELBO loss function in a VAE consist of?

- ◯ Only the reconstruction loss between input and output
- ◯ Only the KL divergence between encoder and decoder
- √ Reconstruction loss plus KL divergence
- ◯ The sum of all layer weights and biases

Do not write anything here

✂- -

(e) ($1/2$ point) What is the primary objective of a GAN?

- ◯ To classify input data into predefined categories with maximum accuracy
- √ Train generator and discriminator through adversarial competition
- ◯ To compress input data into a lower-dimensional representation
- ◯ To directly copy and memorize the training dataset

(f) ($1/2$ point) Which of the following best describes the role of the discriminator in a GAN?

- √ The discriminator classifies real vs fake data
- ◯ The discriminator generates new synthetic data samples
- ◯ The discriminator compresses data into a latent space
- ◯ The discriminator optimizes the generator's weights directly

(g) ($1/2$ point) How does the adversarial training process differ from supervised learning?

- ◯ Supervised learning cannot handle image generation tasks
- √ In adversarial training, two networks compete against each other with opposing objectives, while supervised learning uses labeled data and a single loss function
- ◯ Adversarial training requires more labeled data than supervised learning
- ◯ GANs use SGD while supervised learning uses other optimization methods

(h) ($1/2$ point) What is the main advantage of using transfer learning in deep learning?

- ◯ Transfer learning eliminates the need for data preprocessing
- ◯ Transfer learning guarantees 100% accuracy on any task
- ◯ Transfer learning allows training without using backpropagation
- √ Transfer learning leverages knowledge from pre-trained models on large datasets, reducing training time and data requirements for new tasks

(i) ($1/2$ point) Which of the following best describes fine-tuning in transfer learning?

- ◯ Fine-tuning is the process of removing layers from a pre-trained model
- √ Fine-tuning adapts pre-trained models to new tasks by adjusting weights
- ◯ Fine-tuning means freezing all weights and only changing the input data format
- ◯ Fine-tuning is only applicable to classification tasks

Do not write anything here

(j) (½ point) Which of the following best describes the concept of word embeddings in NLP?

- √ Word embeddings map words to vectors where similar words are close
- ◯ Word embeddings are one-hot encoded representations of words
- ◯ Word embeddings store the frequency count of each word in the corpus
- ◯ Word embeddings eliminate the need for training data in NLP models

(k) (½ point) What is the primary purpose of `Ansible` in infrastructure automation?

- ◯ `Ansible` is a programming language used to write web applications
- √ `Ansible` is an agentless automation via SSH and YAML playbooks
- ◯ `Ansible` is a containerization platform similar to `Docker`
- ◯ `Ansible` requires agents to be installed on all target machines

(l) (½ point) Which of the following best describes an `Ansible` playbook?

- ◯ An `Ansible` playbook is a compiled binary that runs on remote servers
- ◯ An `Ansible` playbook is a programming language for developing applications
- ◯ An `Ansible` playbook requires manual execution on each target machine
- √ An `Ansible` playbook is a YAML file defining tasks for host configuration

ANSWERS

(m) (½ point) What is the primary purpose of a `Jenkinsfile` in a `Jenkins` pipeline?

- √ To define the pipeline as code using declarative or scripted syntax
- ◯ To store build artifacts and deployment logs
- ◯ To configure `Jenkins` server settings and plugins
- ◯ To manage user permissions and access control

(n) (½ point) What is a `Jenkins` agent?

- ◯ A backup server that stores `Jenkins` configuration
- ◯ A plugin that extends `Jenkins` functionality
- √ Machine executing `Jenkins` build jobs
- ◯ A security module for authentication

(o) (½ point) What is the smallest deployable unit in `Kubernetes`?

◯ `Container` ◯ `Node` ◯ `Deployment` √ `Pod`

(p) (½ point) Which `Kubernetes` component is responsible for maintaining the desired state of the cluster?

◯ `Kubelet` ◯ `Kube-proxy` √ `Controller Manager` ◯ `API Server`

✂- -

(q) (½ point) What is the purpose of a `Kubernetes` Service?

- ○ To store persistent data for pods
- ○ To schedule pods on available nodes
- √ To provide stable network endpoints and load balancing for pods
- ○ To monitor resource usage of containers

(r) (½ point) What data model does `Prometheus` use to store metrics?

- ○ Relational database with `SQL` queries
- ○ Document-based `NoSQL` storage
- ○ Graph database with nodes and edges
- √ Time-series database with key-value labels

(s) (½ point) What is the query language used in `Prometheus`?

○ `SQL` ○ `GraphQL` ○ `MongoDB Query Language` √ `PromQL`

(t) (½ point) How does `Prometheus` typically collect metrics from applications?

- ○ Applications push metrics to `Prometheus` server
- ○ `Prometheus` reads log files from applications
- √ `Prometheus` scrapes metrics from `HTTP` endpoints exposed by applications
- ○ Applications write metrics directly to `Prometheus` database