use Google Colab Notebook as your programming environment to develop and implement your task using Python programming language.

You are allowed to use PyTorch, TensorFlow or Keras to code your ANN. Standard auxiliary Python libraries such as NumPy, and Matplotlib are allowed. You can also use your code developed or given in the labs or lectures.

If you want to use any other specific libraries or code, please discuss and obtain permission from the Unit Leader first.

# Part A

## Task A1

Imagine you are given a task by your game company to create an intelligent NPC. You would like to train an artificial neural network (ANN) that acts as the brain of your NPC.

Your ANN must learn to handle your NPC's decision-making process in your game - that is, whether the NPC will attack or flee, depending on the NPC’s power and the enemy’s power, as described in the following table.

A white background with black text

Description automatically generated

First iteration p = 1   
y(2) = 0.8455   
y(3) = 0.8581   
y(4) = 0.5571   
e(4) = -0.5571   
Delta(4) = -0.1375   
w(2,4) = -1.2116   
w(3,4) = 1.0882   
Bias(4) = 0.2863   
Delta(2) = 0.0215   
w(0,2) = 0.5022   
w(1,2) = 0.4022   
Bias(2) = 0.8022   
Delta(3) = -0.0184   
w(0,3) = 0.8982   
w(1,3) = 0.9982   
Bias(3) = -0.1018   
te(4) = -0.5483   
EpocSumError = 0.3007

SOLUTIONS   
w(0,2) = 4.5742   
w(0,3) = 6.6165   
w(1,2) = 4.5669   
w(1,3) = 6.5849   
w(2,4) = -10.5474   
w(3,4) = 9.7713   
Bias(2) = -7.0188;

Bias(3) = -2.9530;

Bias(4) = -4.5013

A diagram of a diagram

Description automatically generated

### Task A1.1

Develop a computer program to implement an ANN, with the Sigmoid function as the activation function, to make the decision according to the table. Your ANN should consist of 2 input neurons, 1 hidden layer with 2 neurons and 1 output neuron. Then train your ANN using the backpropagation technique to learn the NPC's decision/action table.

### Task A1.2

Analyse and evaluate the effect of the learning rate and find the optimal learning rate in this case.

### Task A1.3

Analyse and evaluate the effects of activation functions by comparing Sigmoid function with the Sign function and Step function. Find out what is the best activation function for this application.

## Task A2

Imagine you are given a task by your game company to create an intelligent NPC. You would like to train a machine learning model that acts as the brain of your NPC. Instead of deciding according to a table, now it must decide like a human so that it can interact realistically with a human player. If your NPC (with a machine learning brain) has inputs and outputs as follows:

A blue rectangle with white text

Description automatically generated

A table with numbers and text

Description automatically generated

### Task A2.1

Conduct your own independent literature survey to identify one possible machine learning method, apart from ANN, which can be used to learn and imitate players' behaviours. Explain your justification for why you choose the method.

### Task A2.2

Systematically evaluate the method in 2.1, in comparison to ANNs. Note that in this task, you theoretically evaluate and analyse the techniques based on your literature survey. You do not need to implement the methods.

### Task A2.3

Develop a computer program to implement an ANN to learn the player's behaviour.

### Task A2.4

Systematically design and conduct experiments to determine the optimal number of neurons and layers in the ANN suitable for this task.

### Task A2.5

Evaluate the performance of your ANN.

# Part B

Generative AI is important for the game industry because it can offer many benefits and opportunities for game development and innovation. For example, it increases efficiency and scalability in game development by automating the content creation process and reducing the time and resources needed. It can be used to create endless possibilities for in-game content, such as images, 3D models, environments, etc., making games more diverse and engaging for players.

## Task B1

As a newly graduated programmer and you are new to Generative AI, your team leader gives you an opportunity to familiarise yourself with a simple generative model to generate simple images before moving to more advanced and realistic models. Your team leader asks you to start by using CLIP, which is a model built by OpenAI for generating images from text (https://openai.com/research/clip), or an equivalent model of your choice, to write a Python program to generate simple images from text and conduct some research and experiments to enhance your understanding in Generative AI.

### Task B1.1

Write a Python program to implement CLIP model, or an equivalent model, to generate an image sized 224x224 pixels from a text prompt.

### Task B1.2

Conduct a literature survey on how to improve the generated images in terms of the loss functions, optimisers and regularisation and other necessary hyperparameters.

### Task B1.3

Use the findings in Task B1.2 to improve the generated image quality.

### Task B1.4

Evaluate the performance of your program. To be consistent and for a fair comparison, you must use "A cat face" as your text prompt to generate images.

# Submission

## Report:

write a well-structured formal academic report (approx. 900 words max) in pdf format, which includes the details from the tasks. Here is an example structure and contents of the report:

**Part A**

**Task A1**

- Task A1.1: Explain your ANN, e.g. architecture, activation function, and equations involved

here. This is the description of the ANN you implement.

- Task A1.2: Describe your analysis and evaluation here.

- Task A1.3: Describe your analysis and evaluation to determine the best activation function

here.

Task A2

- Task A2.1: Describe your literature survey and your findings here.

- Task A2.2: Describe your analysis and evaluation here.

- Tasks A2.3 and A2.4: Describe your ANN architecture, your experiment design and

experiments to find the optimal number of neurons and layers here. Also, mention the

implementation.

- Task A2.5: Describe your performance evaluation here.

**Part B**

**Task B1**

- Task B1.1: Describe how you implement the model here.

- Task B1.2: Describe your literature survey and findings here.

- Task B1.3: Explain how you use your finding to improve the image quality here.

- Task B1.4: Describe your evaluation process and the results here

## Code for Tasks A1, A2 and B1

Submit 3 Google Colab Notebooks separately, one for each task.

You must ensure that assessors who do not know your program can successfully run your program on their machines. If they cannot run your program, they cannot assess your program. Use ZIP

## Demonstration and narrative video clip

A short video demonstrating step-by-step that your code works and can produce the required results for Tasks A1, A2 and B1. This should be of suitable quality. The resolution should be a maximum of HD (1920x1080), 1GB maximum file size, 5 minutes maximum duration.

## User Guide

a simple text file in your project directory to explain how to run and use your program. An in-program user guide would also be extremely helpful.

# Marking Criteria

**A white text with black text

Description automatically generated with medium confidence**