**LAB LOGBOOK:**

***Week 1: Uses of AI in Games:***

* **NPC behaviours:** Pathfinding (A\* algorithm) and decision trees

Advantages: Creates immersive gameplay environments and characters

Disadvantages: Every character “state” needs to be coded

* **Procedural Generation:** Terrain generation, quest generation, narratives. EG (No Man’s Sky planets, PEAK Mountains, Binding of Isaac floors)

Advantages: Low effort to implement, adds replayability

Disadvantages: Less control over game cycle.

* **Adaptive Difficulty (Reinforcement Learning):** (Learning from the players actions and adjusting the difficulty accordingly)

Advantages: Increases player satisfaction.

Disadvantages: Needs to be balanced correctly. Could result in inconsistent spikes in difficulty. Hard to transfer to other games.

* **Using AI in testing:** AI can act as players and help to identify bugs or flaws in the game. Can be automated.

Advantages: Reduces playtesting time.

* **Genetic Algorithms (GAs):** Like natural selection, AI is guided to optimal strategies by a scoring system, dropping ineffective strategies after successive generations.

Disadvantages: Suffers from premature convergence: (where an algorithm stops at a suboptimal solution rather than looking for the better option).

* **Neuroevolution of Augmenting Topologies (NEAT):** A development of GAs, which has flexible topologies allowing modification of network structure and connections. Nodes can be randomly added or remove, and this change can stay if it proves beneficial.

Advantages: Highly effective for gameplay optimisation and real-time decision making.

Issues for AI:

* **Time and Space Resources:** AI is very complex which can lead to performance issues. Multiplayer games need to have uniform experiences for players.
* **Balanced AI Difficulty:**  Need to create an opponent that is competitive enough to provide a challenge but not to feel as if they are cheating.
* **Ethics:** AI needs to not be prejudice, stereotypical or discriminative when making decisions. AI personalisation can have privacy issues.

***Week 2: ANN activation functions:***

**Sign function:**

**Step function:**

**Sigmoid function:**

**A graph of a function

AI-generated content may be incorrect.**

A graph with a line

AI-generated content may be incorrect.

***A graph of a function

AI-generated content may be incorrect.***

**ReLU (Rectified Linear Unit):**

All positive values pass through unchanged, while all negative values are set to 0.

**A number of mathematical equations

AI-generated content may be incorrect.**

A graph of a function

AI-generated content may be incorrect.

**Leaky ReLU:**

A modified version of ReLU which fixes the problem of dead neurons by allowing negative inputs to return a small negative part of the input.

A mathematical equation with numbers and symbols

AI-generated content may be incorrect.

A graph of a leaky relu function

AI-generated content may be incorrect.

**Linear Function:**

A straight line in which y = x. The output is always a linear combination of the input.

A graph of a function

AI-generated content may be incorrect.

**Tanh (Hyperbolic Tangent):**

A shifted version of the Sigmoid Function, stretching it across the y-axis.

**A math equation with a number and a symbol

AI-generated content may be incorrect.**

A graph with a line

AI-generated content may be incorrect.

***Week 3: Weights and errors:***

#Functions

def StepFunction(x, theta):

    if x > theta:

        return 1

    else:

        return 0

#Initialisation

iterations = 20

w1 = 0.3

w2 = -0.2

theta = 0.2

alpha = 0.1

x1 = [0,0,1,1]

x2 = [0,1,0,1]

Yd = [0,0,0,1]

#Activation

p = 1

while( p <= iterations):

  for i in range(4):

    X = x1[i] \* w1 + x2[i] \* w2

    Y = StepFunction(X, theta)

    error = Yd[i] - Y

    print("yD[" + str(i) + "]: error = " + str(error))

    w1 = w1 + alpha \* x1[i] \* error

    w2 = w2 + alpha \* x2[i] \* error

    p = p + 1

  print("")

print("w1: " + str(w1))

print("w2: " + str(w2))

Results (Desired outcome is an AND Gate for Y outputs):

A screenshot of a computer error

AI-generated content may be incorrect.

After 20 iterations, there are no more errors. The values required for w1 and w2 in order to get an AND gate are 0.2 and 0.1 respectively.

***Week 4: C++ Code of a Multi-Neuron Network simulating a XOR Gate***

***using System;***

***using System.Collections.Generic;***

***using System.Data.SqlTypes;***

***using System.Linq;***

***using System.Text;***

***using System.Threading.Tasks;***

***namespace AIGP***

***{***

***class Program***

***{***

***static double Sigmoid(double xValue, double bias)***

***{***

***double Y = 1 / (1 + Math.Exp(-(xValue + bias)));***

***return Y;***

***}***

***static double SigmoidDerivative(double nValue)***

***{***

***return nValue \* (1 - nValue);***

***}***

***static void Main(string[] args)***

***{***

***int numberOfInterations = 2000000;***

***double alpha = 0.2; //learning rate***

***//Starting weights***

***double w13 = 0.3;***

***double w14 = 0.2;***

***double w23 = -0.2;***

***double w24 = 0.2;***

***double w35 = 0.3;***

***double w45 = 0.1;***

***//Biases***

***double b3 = 0.1;***

***double b4 = 0.1;***

***double b5 = 0.1;***

***//Inputs and expected output***

***int[] x1 = { 0, 0, 1, 1 };***

***int[] x2 = { 0, 1, 0, 1 };***

***int[] Yd= { 0, 1, 1, 0 };***

***for(int i = 0; i < numberOfInterations; i++)***

***{***

***double totalError = 0;***

***for(int j = 0; j < 4; j++)***

***{***

***//Forward pass***

***double n3 = Sigmoid(x1[j] \* w13 + x2[j] \* w23, b3);***

***double n4 = Sigmoid(x1[j] \* w14 + x2[j] \* w24, b4);***

***double Y = Sigmoid(n3 \* w35 + n4 \* w45, b5);***

***//Compute error***

***double error = Yd[j] - Y;***

***totalError += Math.Abs(error);***

***// Console.WriteLine("yD[" + j + "]: error = " + error.ToString());***

***//Back Propagation***

***double delta5 = error \* SigmoidDerivative(Y);***

***double delta3 = delta5 \* w35 \* SigmoidDerivative(n3);***

***double delta4 = delta5 \* w45 \* SigmoidDerivative(n4);***

***//Update Weights***

***w35 += alpha \* n3 \* delta5;***

***w45 += alpha \* n4 \* delta5;***

***w13 += alpha \* x1[j] \* delta3;***

***w23 += alpha \* x2[j] \* delta3;***

***w14 += alpha \* x1[j] \* delta4;***

***w24 += alpha \* x2[j] \* delta4;***

***//Update Biases***

***b3 += alpha \* delta3;***

***b4 += alpha \* delta4;***

***b5 += alpha \* delta5;***

***}***

***if (i % 1000 == 0)***

***Console.WriteLine($"Epoch {i}, Error = {totalError}");***

***}***

***Console.ReadKey();***

***}***

***}***

***}***

**Citations:**

|  |  |
| --- | --- |
| **Date cited:** | **Citation** |
| **2/10/25** | *A. Waghale, N. Potdukhe and R. Rewatkar, "AI in Gaming: From Simple Algorithms to Complex Agents," 2024 2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI), Wardha, India, 2024, pp. 1-5, doi: 10.1109/IDICAIEI61867.2024.10842756. keywords: {Ethics;Video games;Machine learning algorithms;Reviews;Entertainment industry;Games;Virtual reality;Artificial intelligence;Protection;Testing;Game AI;Pac-Man;Video Games;3D Gaming;Virtual Reality (VR);competition},*  [*https://ieeexplore.ieee.org/document/10842756*](https://ieeexplore.ieee.org/document/10842756) |
| **2/10/25** | *V. Kumar, D. Tyagi, A. Kannaujia and I. Mittal, "Advancing Game AI: A Study on Genetic Algorithms and Neuroevolution," 2025 12th International Conference on Computing for Sustainable Global Development (INDIACom), Delhi, India, 2025, pp. 1-4, doi: 10.23919/INDIACom66777.2025.11115524. keywords: {Training;Automation;Network topology;Neural networks;Games;Switches;Machine learning;Topology;Genetic algorithms;Testing;Genetic Algorithm;NEAT;Machine Learning;Game AI},*  [*https://ieeexplore.ieee.org/document/11115524*](https://ieeexplore.ieee.org/document/11115524) |
|  |  |