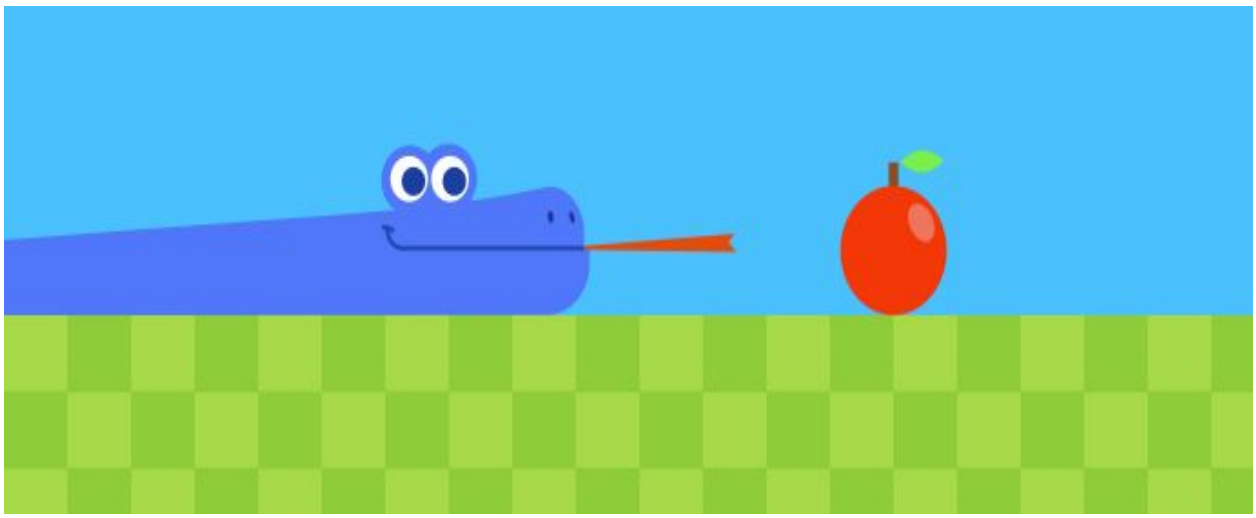


# Deep Neural Network In Snake Game

*Deep Learning Final Project Proposal*

Ahmed Magedy (07)
Mohamed Kamal El-Din Ali El-Shazly (59)
Hesham Medhat (71)

# Introduction and Context

This is the proposal for the final team project of the elective Deep Learning course **CS435** for Spring 2020 in the final year of Computer and Systems Engineering class '20 in Faculty of Engineering, Alexandria University.

## Staff

**Lecturer:** Dr. Moustafa Youssef

**Teaching Assistant:** Eng. Khaled El-Tahan

# Problem Statement

The classical game **Snake** was a world's amusement since 1977. The game is about challenging a snake on the screen to be able to manage moving around a 2-dimensional grid to get rewards (apples/food).

## Game Rules

- Upon eating food, the snake gets its body grown.
- **The snake has the choices of either:**
    1. Continuing to move forward in the given direction.
    2. Take a left turn.
    3. Take a right turn.
- The snake dies upon contact with its own body or a wall.

## Deep Learning Challenge

The challenge is essentially creating using deep learning to build a neural network that would learn playing the game maximizing the score in a form of intersection between modern Deep Learning and Artificial Intelligence.

## Motivation

Applying what we have learnt in the Artificial Intelligence course last semester, in addition to what we learn in this semester in Deep Learning exploring the capabilities of deep neural networks in solving what used to be a childhood challenge for us.

---

## State of the Art

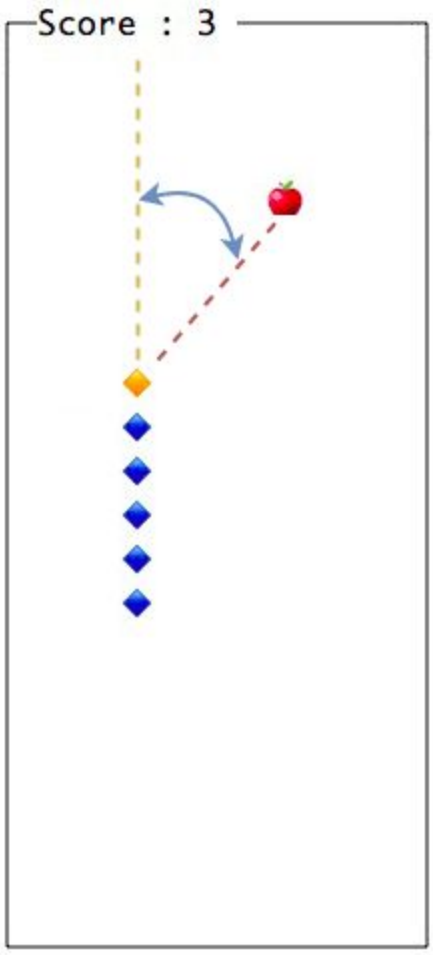
**"Autonomous Agents in Snake Game via Deep Reinforcement Learning"** paper

In this paper, The agent learns how to play the Snake Game based on the screen snapshots. Specifically, it relies on a deep Q-learning network (DQN) to choose the best action based on both the observations from the environment and prior learned knowledge. It uses a series of four screenshots of the Snake Game as the network input. Therefore, the network is able to capture the game information including direction and position, and then output the estimated Q-value of each action.

## Survey of Available Models

1. "Autonomous Agents in Snake Game via Deep Reinforcement Learning" paper is taking screenshots for the play grid and making analysis to fetch the location of snake and apple using CNN . The reward mechanism depends on many factors:
    a. Length of snake.
    b. Distance to apple.
    c. Timeout.
    d. training gap by preventing the agent from learning for M time steps after it ate an apple.
2. In this article they use a forward neural network to learn the snake which direction it should take to max its score. The input parameter to the neural network is:
    - Is there an obstacle to the left of the snake (1 — yes, 0 — no)
    - Is there an obstacle in front of the snake (1 — yes, 0 — no)
    - Is there an obstacle to the right of the snake (1 — yes, 0 — no)

- Angle between snake's movement direction and direction to an apple (from -1 to 1)




---

## Description of the Model to be Used

***Our goal is to merge both neural networks and a genetic algorithm to get more accuracy but why?***

In December 2017, Uber AI Labs released five papers, related to the topic of neuroevolution, a practice where deep neural networks are optimised by evolutionary algorithms. One of them is "Deep Neuroevolution: Genetic Algorithms are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning" which deduces that:

*"Using a simple Genetic Algorithm (GA) it is possible to optimise DNNs and that GAs are a competitive alternative to gradient based methods when applied to Reinforcement Learning (RL) tasks such as learning to play Atari games, simulated humanoid locomotion or deceptive maze problems."*

However, what is the idea behind the **Genetic Algorithm**?

**Genetic Algorithm**

Genetic Algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.

The process of natural selection starts with the selection of fittest individuals from a population. They produce offspring which inherit the characteristics of the parents and will be added to the next generation. If parents have better fitness, their offspring will be better than parents and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals will be found.

This notion can be applied for a search problem. We consider a set of solutions for a problem and select the set of best ones out of them.

## Our Implementation and Approach

For the sake of challenging our feature engineering skills and exploring the genetic learning capabilities of the deep neural networks, we will let the genetic algorithm (GA) and neural network(NN) play the snake game, and Instead of backpropagation, we will update weights using GA to find the best parameters.

We think our Implementation will go through some steps:

1. Determine neural network architecture.
2. Creating an initial population and fitness function (can use some factors from similar models which attack the same problem)
3. Play a game for each individual in the population and sort each individual in the population based on the fitness function score.

4. Select a few top individuals from the population and create the remaining population from these top selected individuals using Crossover and mutation.
5. The new population is created (meaning the next generation).

And so on..

We will simulate the game for the network, and make use of that facility of observing results and how the mind of the neural network works as it makes decisions attempting to win the game with top score. As much as charts and graphs are hard in terms of extracting information out of them, observing the simulation of a game that has been a great part of our childhood would be most educational and beneficial to us as we explore deep neural networks and how changing hyper-parameters reflects on its decision-making process.

## How to Evaluate Results

We will compute the average score the agent will gain after playing snake many times and compare our result to “Autonomous Agents in Snake Game via Deep Reinforcement Learning” paper.

We will try to make this test for different sizes of play grid if it is possible.

## Graduation Project Problem Statement

Our graduation projects are mostly Machine Learning-Free.

- Ahmed and Hesham are teammates in a large-scale system project for a product that serves sharing knowledge with everyone in a large social network of learners and experts.
- Mohamed’s graduation project is aiming to increase the scalability of voting systems using public block chain.

## Resources

https://www.researchgate.net/publication/327638529_Autonomous_Agents_in_Snake_Game_via_Deep_Reinforcement_Learning?fbclid=IwAR2tYSFsQtAeB9rH1evuCkyi7Gk4gsXoEEOZGbiCo4WpKsOU1zbrRFxjT0I

https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html

https://en.wikipedia.org/wiki/Q-learning#Deep_Q-learning

https://towardsdatascience.com/today-im-going-to-talk-about-a-small-practical-example-of-using-neural-networks-training-one-to-6b2cbd6efdb3

https://towardsdatascience.com/deep-neuroevolution-genetic-algorithms-are-a-competitive-alternative-for-training-deep-neural-822bfe3291f5

https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3