**OSLO METROPOLITAN UNIVERSITY**
STORBYUNIVERSITETET

Pre Project report

# Create a machine learning algorithm that combines cellular automata and learning automata.

# Presentation

Group members:

Hedda Marie Westlin
Information technology
Contact: s333721@oslomet.no

Rosy Oo
Information technology
Contact: s333726@oslomet.no

Supervisor: Stefano Nichele.
Institute: Faculty of Technology, Art and Design, Department of Computer Science, Artificial intelligence
Contact: Associate Professor Stefano Nichele, stenic@oslomet.no

Co-supervisor: Professor Anis Yazidi.
Institute: Faculty of Technology, Art and Design, Department of Computer Science, Artificial intelligence
Contact: Anis.Yazidi@oslomet.no

Co-supervisor: PhD Candidate Sidney Pontes-Filho.
Institute: Faculty of Technology, Art and Design, Department of Computer Science, Artificial intelligence
Contact: Sidney.Pontes-Filho@oslomet.no

## Summary

Our project is a basic research type project where we are going to develop our own Cellular Learning automaton algorithm. The work is based on research literature and developing our own code with the goal of combining Cellular automata, and Learning automata. This project is part of the OsloMet Living technology Lab (part of OsloMet AI Lab) on neuro-inspired unconventional computing machines. To prove or deny that the algorithm is able to function as described in research literature, we want to run experiments on the algorithm to verify viability of the algorithm.

## Goals and framework conditions

The base for our thesis is a basic research project where our primary goal of the project is to create a new CA-based machine learning algorithm by combining cellular automata (CA) and learning automata (LA) to make a cellular learning automaton and confirm or deny that the algorithms are able to produce the expected results described in scientific literature. We set out to develop an algorithm that combines cellular automata, and learning automata by using research literature and other implementations of the algorithms to support us in programming the algorithm from scratch using python as our main programming language.

## Status quo

This thesis is part of ongoing research projects at the OsloMet Living Technology Lab (part of OsloMet AI Lab) on neuro-inspired unconventional computing machines. The long term goal of this project is to build computing machines that go beyond the current von Neumann paradigm of computing, by taking inspiration from how the brain works. Cellular Automata (CA) are interesting models of cellular computing, where the actual information processing,

transmission of information and storage are massively distributed and parallelized, and each component of the system interacts only locally with the closest neighbors. One such example of cellular automata is the Game of Life, which is proven to be computationally universal. While CA can produce very complex computations, they lack one key aspect of (biological and artificial) neural networks, i.e., plasticity. Neuroplasticity is the ability of the brain to change (learn) over time. One class of automata that can learn over time is Learning Automata (LA), a special type of reinforcement learning automata. This project aims at creating a new CA-based machine learning paradigm by combining CA and LA, i.e., Cellular Learning Automata (CLA). In this way, each cell in a CA may change over time its function (transition rule) based on the actual local activity of the system, providing a mechanism of plasticity in cellular automata (a kind of Hebbian learning for CA).

## Solutions

While no clear solution is readily available, there are algorithms that achieve the goals we are trying to achieve, but none that we are aware of at this time that combines the algorithms into cellular learning automata (CLA) that is available to us, however they are described in literature. Our main goal is to write our own cellular learning automata algorithm. We want to achieve this by going through relevant literature on the topic, as well as analyzing the code from CA and LA algorithms. Benefits and drawbacks with writing our own algorithm; analyzing and writing our own code may be time consuming, and our timeframe is limited to approximately 4 months total. Benefits may be that by writing our own algorithm we have greater control of what the end product is going to be, and a greater understanding of how the code works.

## Analyses

Given a functioning CLA algorithm, we have a preliminary criticality experiment. The experiment will test the LA part of the CLA algorithm when the CA has a faulty cell. The goal of the experiment is to see if the LA is adapted to handle a faulty cell in the CA, and provide a learned input that will compensate for this, allowing the CA to function even while a cell is faulty.