AI Course: Home assignment

II. Smart grid control using Neuro Evolution/genetic programming **(Group work. max:10 students)**

We consider the problem of control of a micro electricity grid that contains residential electric loads, thermostatically controlled loads TCLs (mainly air conditioning systems), a battery, an electricity generator as well as a connection to the main electricity grid on which it can sell or buy electricity. The micro grid manager must balance between the electricity produced/purchased and the electricity consumed/stored/sold in a cost optimal way. This problem is formulated as a Markov-decision process where the Agent (The microgrid manager) interacts with the environment (the microgrid) and tries to maximize the reward. This is a reinforcement learning problem that can be solved in many ways using DQN, Policy gradient or proximal policy optimization. However, we want to use a **neuro-evolution** method to evolve a neural network that takes optimal actions given a state of the environment. For this project we provide a simulation of the environment using **Openai gym** implemented in Python. Therefore, the focus should be on the neuro evolution method.

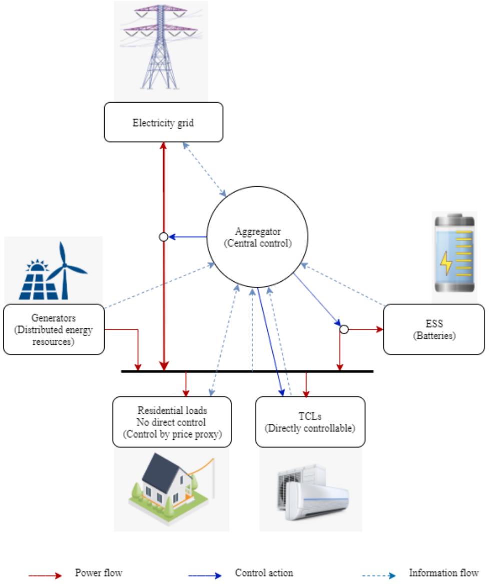

Figure 1: Control framwork and micro grid components

The action constitutes of 4 sub actions: TCL action, Price action, Energy deficiency action and Energy excess action. The possible values of each sub action are positive integers and are as follows:
TCL action: [0:3]
Price action: [0:4]
Energy deficiency action: [0:1]
Energy excess action [0:1]
Each action is a combination of sub-actions, therefore we have 4*5*2*2= 80 possible actions.
Therefore, the NN we want to design has to provide 80 outputs that define the probability distribution of the actions. (This can be seen as a classification problem: choose one action from 80 possible actions)
The fitness function should be the total reward for one day (24 timesteps)

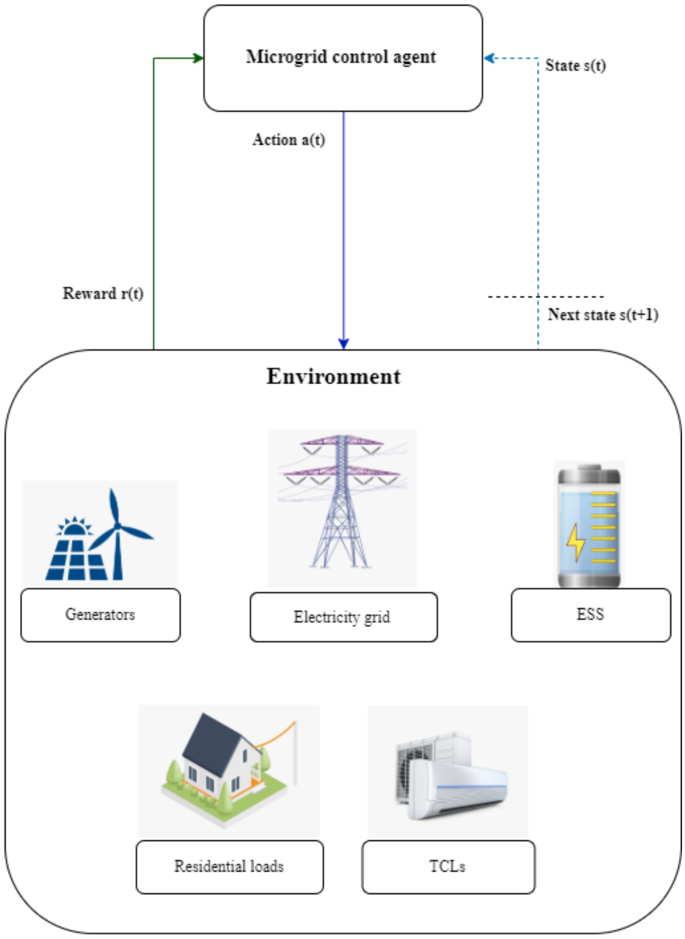

Figure 2 Agent/ Environment modelling

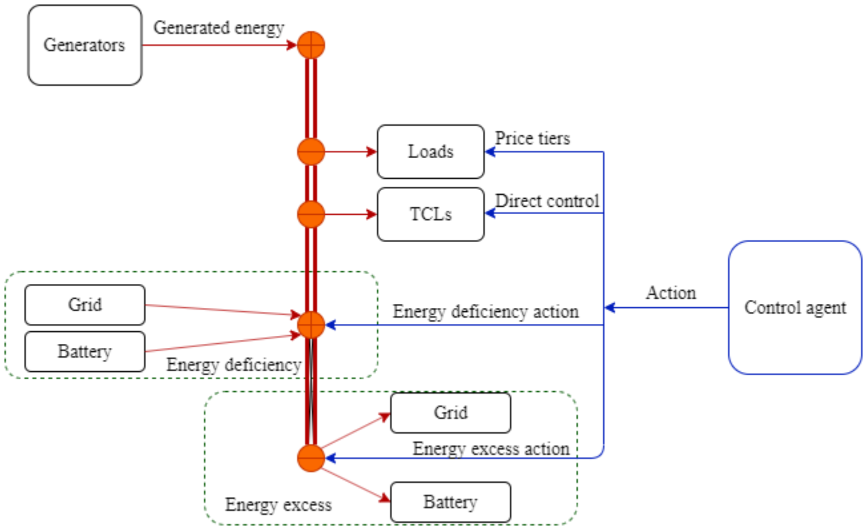

Figure 3: Control scenario

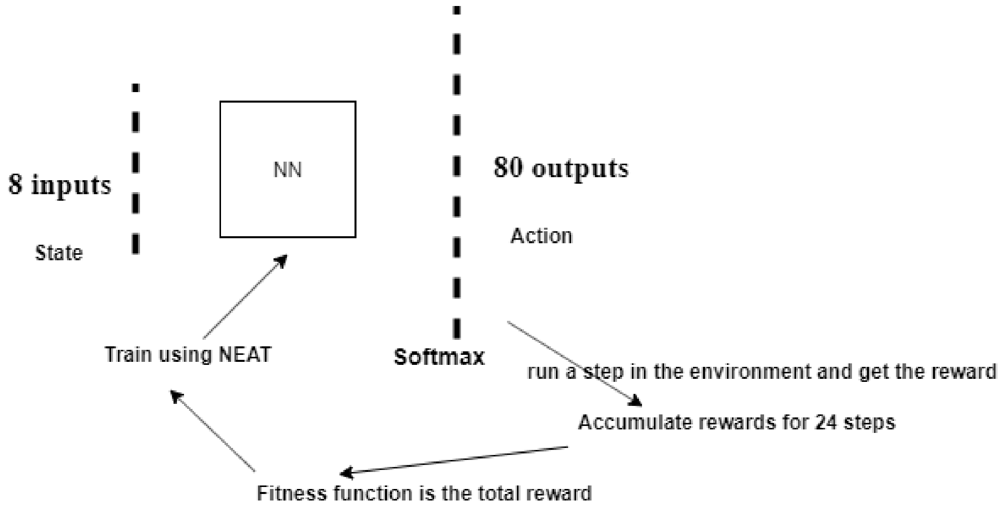
8 inputs
State
NN
80 outputs
Action
Train using NEAT
Softmax
run a step in the environment and get the reward
Accumulate rewards for 24 steps
Fitness function is the total reward

Figure 4:NN prototype

Task:

- Read the paper about Evolving ***Neural Networks through Augmenting Topologies (NEAT)***
- Download the code of the simulation and the related data
- Implement a **NEAT** algorithm suited for this simulation (Find some tutorials on how to implement **NEAT** algorithm for a Gym environment)
- Visualize and analyze the results: Reward and environment states for every 24 hours (control time steps)
- Return code+report

Paper: http://nn.cs.utexas.edu/downloads/papers/stanley.ec02.pdf