

ECTA Homework 5
Neuroevolution with the
Enforced Subpopulations (ESP) algorithm

Dharmin Bakaraniya, dharmin.bakaraniya@smail.inf.h-brs.de
Arun Prabhu, arun.prabhu@smail.inf.h-brs.de

July 11, 2018

WOW - RESEARCHERS TAUGHT A COMPUTER
TO BEAT THE WORLD'S BEST HUMANS AT
YET *ANOTHER* TASK. DOES OUR SPECIES
HAVE *ANYTHING* LEFT TO BE PROUD OF?

WELL, IT SOUNDS LIKE WE'RE
PRETTY AWESOME AT TEACHING.

HUH? WHAT
GOOD IS *THAT*?



1 Assignment Description

1. Solve the following Reinforcement Learning tasks with Neuroevolution.
 - (a) Single Pole Balancing with Velocities
 - (b) Double Pole Balancing with Velocities
 - (c) Double Pole Balancing without Velocities
2. Compare a standard approach to the ESP algorithm in (c).
 - The task is “solved” if the poles does not fall for 1000 time steps.
 - A 1 and 2 pole balancing simulator function is attached. Given a state and a command it computes the state at the next time step. Examples are included (help: onePole, twoPole).
Note: there is only one 2 pole balancing function, in the case without velocities it is up to you to “blind” you ANN of these states, but they are required by the simulator to compute what happens in the next time step.
 - You are responsible for all other code, including the ANN.
 - Additional resources for ESP are available on the LEA

2 Submission Instructions

Follow along with the instructions in this PDF, filling in your own code, data, and observations as noted. Your own data should be inserted into the latex code of the PDF and recompiled. All code must be done in MATLAB.

To be perfectly clear we expect two submissions to LEA:

1. 1 PDF (report) – a modified version of your submission PDF, with your own code snippets, figures, and responses inserted
2. 1 ZIP (code) – a .zip file containing all code use to run experiments (.m files) *and* resulting data as a .mat file
3. 4 MAT (solution) – Mat files with the weight matrices of the best found solutions: *spb.mat*, *dpb.mat*, *dpbnv.mat*, and *dpbnv_esp.mat*

3 The Assignment

3.1 Single Pole Balancing (25pts)

Implement a feed forward network which solves the single pole balancing problem using a fixed topology and a GA or ES (your choice).

- (15 pts) Describe your approach. Be sure to mention: the topology of your network, the form of the genotype, the form of the phenotype, the optimization approach used and its details including hyperparameters. Your results should be replicable from your description. Pictures are worth 1000 words.

1. Testing with GA :

- **Feed Forward Neural Network** was used for this task, to learn the action-reaction behaviour of the environment, using Reinforcement Learning. [Maximum number of steps per episode = 1000]
- **Genotype:** The Genotype represents the weight vector. Each Gene in the Genotype represents the weight of a particular connection of the nodes in the network. Size of Genotype depends on the number of inputs and the neural network topology. The Size of genotype used for the FFN topology mentioned below is given by,

Genotype Size =

$$(nInput \cdot nHidden) + (nHidden \cdot nOutput)$$

where

nInput: Number of Inputs Nodes

nHidden: Number of Hidden nodes

nOutput: Number of Output Nodes

Genotype Size = $4 \cdot 1 + 1 \cdot 1 = 5$

- **Phenotype:** Phenotype in a way defines the neural network, given that the topology of the network being used is fixed. Phenotype breathes life into a lifeless fixed topology neural network.
- **Optimization Technique:** The tuning of the weights of the Neural Network was done using the Genetic Algorithm. The hyperparameter details are as shown below.
 - (a) **Population Size:** 10
 - (b) **Maximum Generations:** 100
 - (c) **Selection Pressure:** 2

- (d) **Crossover Probability:** 0.9
- (e) **Mutation Probability:** 0.4
- **Feed Forward Neural Network Topology :**
 - (a) Number of Inputs Nodes = 4 = [Cart Position, Cart Velocity, Pole1 Position, Pole1 Angular Velocity]
 - (b) Number of Hidden Nodes = 1 [**Only 1 hidden Layer**]
 - (c) Number of Output Nodes = 1 [**There is only one output, which is the force to be applied to the cart.**]

2. Testing with ES :

- **Feed Forward Neural Network** was used for this task, to learn the action-reaction behaviour of the environment, using Reinforcement Learning. [Maximum number of steps per episode = 1000]
- **Genotype:** The Genotype represents the weight vector. Each Gene in the Genotype represents the weight of a particular connection of the nodes in the network. Size of Genotype depends on the number of inputs and the neural network topology. The Size of genotype used for the FFN topology mentioned below is given by,
Genotype Size =

$$(nInput \cdot nHidden) + (nHidden \cdot nOutput)$$

where

nInput: Number of Inputs Nodes

nHidden: Number of Hidden nodes

nOutput: Number of Output Nodes

Genotype Size = $4 \cdot 1 + 1 \cdot 1 = 5$

- **Phenotype:** Phenotype in a way defines the neural network, given that the topology of the network being used is fixed. Phenotype breathes life into a lifeless fixed topology neural network.
- **Optimization Technique:** The tuning of the weights of the Neural Network was done using the basic Evolutionary Strategy. The hyperparameter details are as shown below.
 - (a) **Population Size:** 1
 - (b) **Maximum Generations:** 100
 - (c) **Lambda:** 1
 - (d) **Sigma:** 1.0
 - (e) **Change in Sigma:** 0.5
- **Feed Forward Neural Network Topology :**
 - (a) Number of Inputs Nodes = 4 = [Cart Position, Cart Velocity, Pole1 Position, Pole1 Angular Velocity]
 - (b) Number of Hidden Nodes = 1 [**Only 1 hidden Layer**]

(c) Number of Output Nodes = 1 [There is only one output, which is the force to be applied to the cart.]

- (5 pts) Plot the fitness progress over 10 runs.

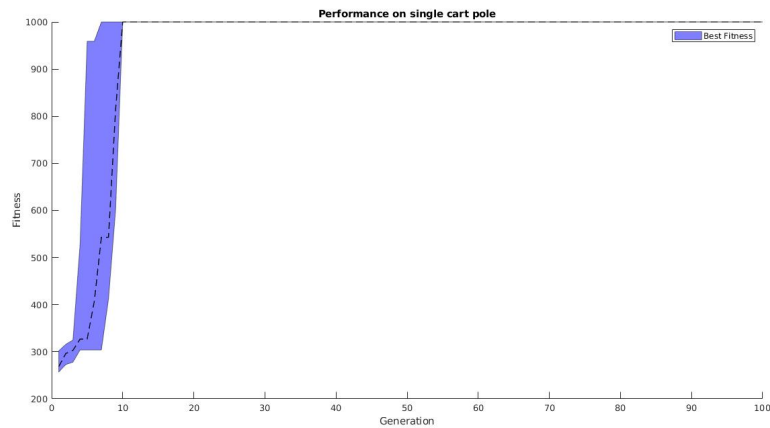


Figure 1: GA Feed forward network for single cart pole balancing problem with above mentioned hyperparameters for 10 experiments

- (5 pts) Upload the weight matrix of your most successful solution in a file called: *spb.mat*
 - Out of our solutions using ES and GA, the GA solution performed better and hence the **spb.mat** represents the GA solution.

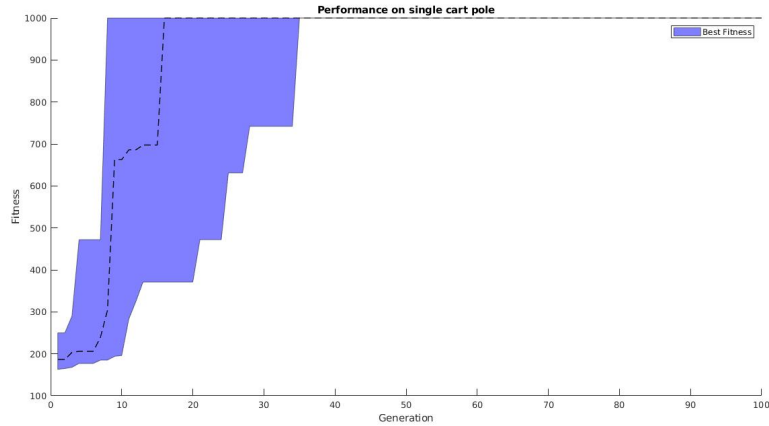


Figure 2: ES Feed forward network for single cart pole balancing problem with above mentioned hyperparameters for 10 experiments

3.2 Double Pole Balancing with Velocities (25pts)

Adapt your feed forward network to solve the double pole balancing problem.

- (15 pts) Describe how this approach differs from above. Your results should be replicable from the this description and the one above.
 1. From the above description(ES and GA) almost everything remains the same except the following, irrespective of the optimization technique used (ES or GA).
 2. **Feed Forward Neural Network Topology :**
 - (a) Number of Inputs Nodes = 6 = [Cart Position, Cart Velocity, Pole1 Position, Pole1 Angular Velocity, Pole2 Position, Pole2 Angular Velocity]
 - (b) Number of Hidden Nodes = 8 [Only 1 hidden Layer]
 - (c) Number of Output Nodes = 1 [There is only one output, which is the force to be applied to the cart.]
 3. **Genotype :** Size of Genotype depends on the number of inputs and the neural network topology. The Size of genotype used for the FFN topology mentioned below is given by,
Genotype Size =

$$(nInput \cdot nHidden) + (nHidden \cdot nOutput)$$

where

n_{Input} : Number of Inputs Nodes

n_{Hidden} : Number of Hidden nodes

n_{Output} : Number of Output Nodes

Genotype Size = $6*8 + 8*1 = 56$

- (5 pts) Plot the fitness progress over 10 runs.

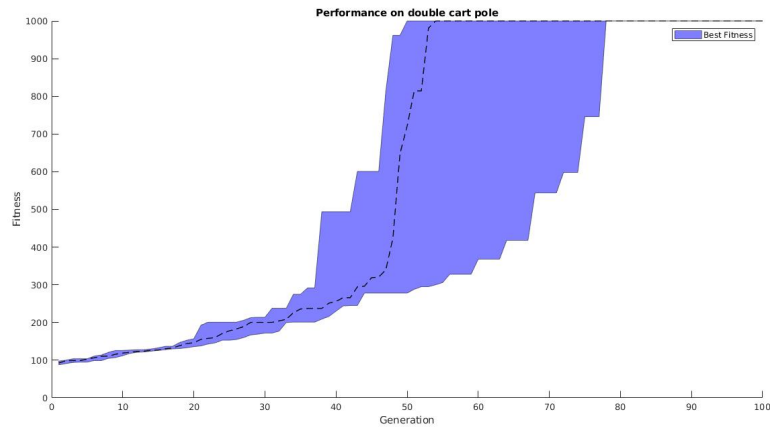


Figure 3: GA Feed forward network for 2 cart pole balancing problem for 10 experiments with above mentioned hyperparameters

- (5 pts) Upload the weight matrix of your most successful solution in a file called: *dpb.mat*
 - Out of our solutions using ES and GA, the GA solution performed better and hence the **dpb.mat** represents the GA solution.

3.3 Double Pole Balancing without Velocities (50pts)

Alter your network topology to solve the double pole balancing problem without velocities using a recurrent neural network. Implement the ESP algorithm to optimize the same network topology. Compare the performance of the two approaches.

- (10 pts) Describe how the first approach differs from above. Your results should be replicable from the this description and the one above.

1. Recurrent Neural Network is used instead of the Feed Forward Neural Network.
2. All the entities marked in red are differing from the ones mentioned previously.
3. **Recurrent Neural Network Topology :**
 - (a) Number of Inputs Nodes = 3 = [Cart Position, Pole1 Position, Pole2 Position]
 - (b) Number of Hidden Nodes = 5 [Only 1 hidden Layer]
 - (c) Number of Output Nodes = 1 [There is only one output, which is the force to be applied to the cart.]
4. **Genotype :** Size of Genotype depends on the number of inputs and the neural network topology also it depends on the type of neural network being used (RNN or FFN). The size of genotype used for the RNN topology mentioned below is given by,
Genotype Size =

$$((nInput + 1) \cdot nHidden) + (nHidden \cdot (nHidden - 1))$$

where

nInput: Number of Inputs Nodes

nHidden: Number of Hidden node

$$Genotype\ Size = (3+1)*5 + 5*4 = 40$$

- **GA related changes:**
 1. **Population Size:** 10
 2. **Maximum Generations:** 500
 3. **Selection Pressure:** 2
 4. **Crossover Probability:** 0.9
 5. **Mutation Probability:** 0.5

- (25 pts) Describe the ESP approach, how it differs from above, and other details, including hyperparameters.
 - **ESP Approach:** Repeat the below steps for a fixed number of Generations.
 - * **Initialization:**
 1. Number of nodes in the topology is decided.
 2. A sub-population is created for each of the nodes.
 3. Each individual in a sub-population represents a vector of weights, corresponding to the node(to which the sub-population belongs).
 - * **Evaluation:**
 1. Select an individual randomly from each subpopulation and create a network from the chosen individuals.
 2. Test the generated network activation on the task and get the fitness of the network.
 3. This fitness value is added to each individuals cumulative fitness.
 4. Repeat this till each individual in each subpopulation has participated in k=5 trials.
 5. Get average fitness of each individual by dividing the cumulative fitness of each individual by number of times that individual participated in the network creation.
 - * **Recombination:**
 1. Top quartile of each subpopulation is recombined with another individual from same top quartile of that sub population.
 2. The recombined individuals replace the lowest half of the subpopulation.
 - **ESP vs Basic RNN(with GA):**
 1. In ESP, the individuals(single node in the network) in each sub-population are optimized in such a way that when they form the network, the combined network performs well.
 2. In RNN, the the network itself is evolved as a whole, over time rather than evolving each node in the network.
 - **ESP hyperparameters:**
 1. **Sub Population Size:** 10
 2. **Minimum number of trials per individual :** 5
 3. **Maximum Generations:** 100
 4. **Selection Pressure:** 2
 5. **Crossover Probability:** 1

6. Mutation Probability: $3/n\text{Genes}$

where $n\text{Genes} = (\text{number of inputs} + \text{number of hidden nodes})$
 $= (3 + 5) = 8$ in our case.

- Basic RNN hyperparameters(with GA) are as described above.
 - The network topology for the ESP is almost same as the one used in the Basic RNN(with GA) described above. Only difference being there is no bias used in ESP.
- (10 pts) Plot the fitness progress of both approaches over 10 runs, including significance

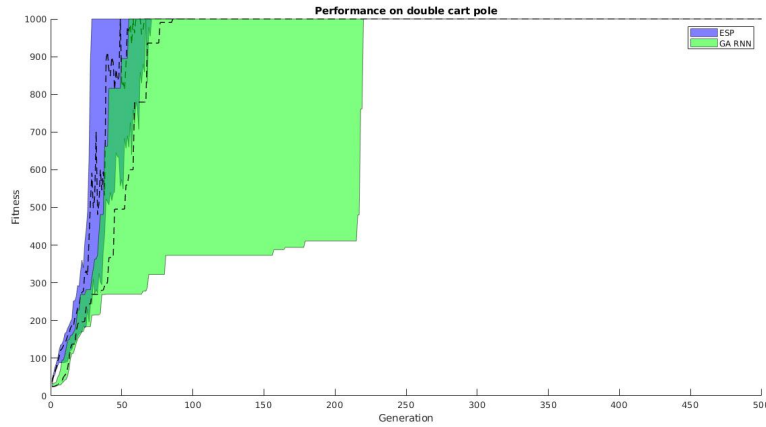


Figure 4: GA RNN vs. ESP for double cart pole balancing problem with above mentioned hyperparameters for 10 experiments. The dotted line represent the median of the best fitness per generation. The area covers by the colored plots are 25 percentile and 75 percentile. The state does not contain velocity information of cart or any of the 2 poles.

- (5 pts) Upload the weight matrix of your most successful solution in 2 files called: *dpbnv.mat* and *dpbnv.esp.mat* The mat file for RNN with GA is provided with *dpbnv.mat* and ESP weight matrix file is named *dpbnv_{esp}.mat*.

4 Coding an ANN in MATLAB