Date / /
Page No.

Machine learning Techniques:
Assignment-2

Sumant Kumar
18290101157
3CSC

Q.1 Genetic algorithm uses principle of natural evolution.
There are 5 important features of Genetic algorithm.

1. Encoding - possible solutions of a problem are considered as individuals in a population. If the solution can be divided into series of small steps, than these steps are represented by genes and a series of genes will encode the whole solution. This way different solutions of a problem are represented in GA as chromosomes of individuals.
2. Fitness function - represents the main requirement of the desired solution of problem. This function calculates and returns the fitness of an individual
3. Selection - operator defines the way individuals in the current population are selected for responsible. There are many strategies for that but the usually the individuals that are more fit are selected.
4. Crossover - operator defines how chromosomes of parents are mixed in order to maintain and obtain genetic codes of their offspring. This operator implements the inheritance property.
5. Mutation - operator creates random changes in genetic codes of the offspring. This operator is needed to bring some random diversity into the genetic code. In some cases GA cannot find the optimal solution without mutation operator.

Q.2(a) Each chromosomes will consists of 10 genes. Each genes representing the path between a pair of cities in a tour.

The alphabet will consists of 45 genes. Indeed each of 10 cities each be

Connected with 9 remaining.

Thus

10 x 9 = 90 is the no. of ways in which 10 cities can be ~~generated~~ grouped into pairs.

However because the direction is not important the no. must be divided by 2. So, we shall need 90/2 = 45 genes in order to encode all pairs.

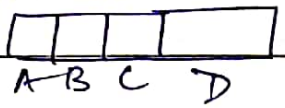
The formulae of n cities = $\frac{n(n-1)}{2} = \frac{10(10-1)}{2} = \frac{10 \times 9}{2} = \frac{90}{2} = \boxed{45}$

Q.3

Item	Weight	Value
A	5 kg	$12
B	3 kg	$5
C	7kg	$10
D	2kg	$7

Here are 4 items, Each is associated with some weight (W) and value at item (N). There is a knapsack (K) with limited capacity that can hold atmost 12 kg. The problem is that which item should be kept in the knapsack so it will be maximise knapsack value without breaking knapsack.

Step-1 chromose encoding



Irene 0 represent absence of item in the knapsack.

1 represent presents of item in the knapsack.

Let 4 bit be used to represent chromose encoding

. Setspace $2^4 = 16$

Initial population is created and chromosomes randomly created.

Date / /
Page No.

generation.

C_1 | 0 | 1 | 1 | 0 |

C_2 | 0 | 1 | 0 | 1 |

C_3 | 1 | 1 | 0 | 1 |

C_4 | 1 | 1 | 1 | 1 |

Step-2 In this we will find out how good particular solution is.

for C_1 | 0 | 1 | 1 | 0 |
A B C D

Value of knapsack = value of B + value of C
= 5 + 10 = 15

Weight of knapsack = weight of item B + weight of item C
= 3 + 7 = 10kg.

12 kg > 10kg, so C_1 is accepted.

for C_2 | 0 | 1 | 0 | 1 |
A B C D

Value of knapsack = value of C + value of D
= 5 + 7 = 12

Weight of knapsack = Weight of B + Weight of D
= 3 + 2 = 5 kg.

12kg > 5 kg, so C_2 is accepted.

for C_3 | 1 | 1 | 0 | 1 |
A B C D

Value of knapsack = 12 + 5 + 7 = 24

Weight of knapsack = 5 + 3 + 7 = 10 kg.

12 kg > 10kg, so C_3 is accepted.

for C_4 | 1 | 1 | 1 | 1 |
A B C D

Value of knapsack = 12 + 5 + 10 + 7 = 34.

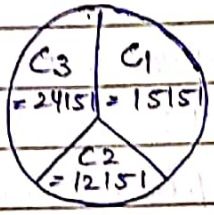
Weight of Knapsack = 5+3+7+2 = 17 Kg.
12 Kg < 17 Kg, So C4 not accepted.

Step-4 selection

Total fitness value = 15+12+24+0 = 51

fitness value of C3 = 24, largest fitness.

C4 has zero chance of winning. C3 has highest probability of getting selected in next generation.



Step-4 crossover

generation 2 →

C3	1	1	0	1	→	1	1	0	0
C1	0	1	1	0	→	0	1	1	1
C3	1	1	0	1	→	1	1	0	0
C2	0	1	1	0	→	0	1	1	1

Step-5 Mutation

1	1	0	0

↑

1	0	0	0

Q.4 Reinforcement learning has 2 main components

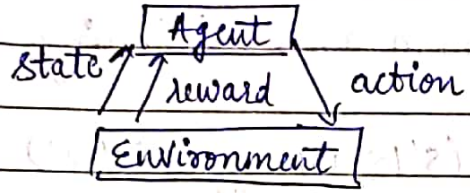
1. Agent
2. Environment

Agent – the reinforcement learning algorithm that learns from trial and error.

Date / /
Page No.

Environment - The world through which the agent interacts.

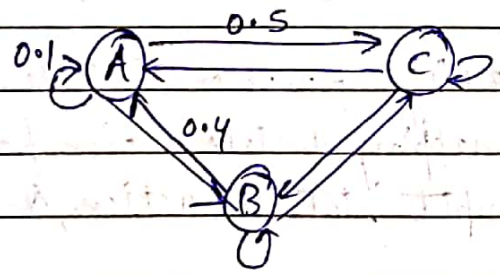
- action - all possible steps that agents can take.
- state - current condition returned by the environment.



Markov decision problem - the agent can perceive a set S of distinct states of its environment and has a set A of actions that it can perform. At each discrete time step t, the agent senses the current state st, chooses a current action ai and perform it.

The environment responds by giving the agent a reward $ri = r(st, ai)$ and providing the succeeding state $st+1 = \delta(st, ai)$. Here function δ and r are part of environment and are not neccessarily known to the great.

Markov decision process says that the probability of making decision from the current node to all possible node (i.e sum of probability of all should be 1).



$0.1 + 0.5 + 0.4 = 1$

Q.5 Q learning - Q learning finds a mapping from state / action pairs to values (called Q-values) it is a model-free reinforcement learning algorithm to learn quality of actions telling an agent what action to take under what circumstances.

It does not require a model of the environment, and it can handle

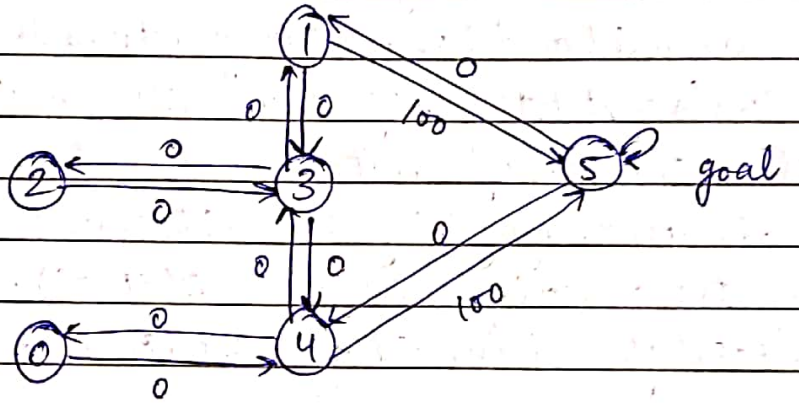
problems with stochastic transitions and rewards without adaptations.

~~It does not require a model of the environment~~

$Q(s,a)$ is the expected discounted future reward for starting in state s taking actions a and continuing optimally that after.

$$Q(s,a) = R(s) + \gamma \sum_{s'} P_{s'}(s'|s,a) \max_{a'} Q(s',a')$$

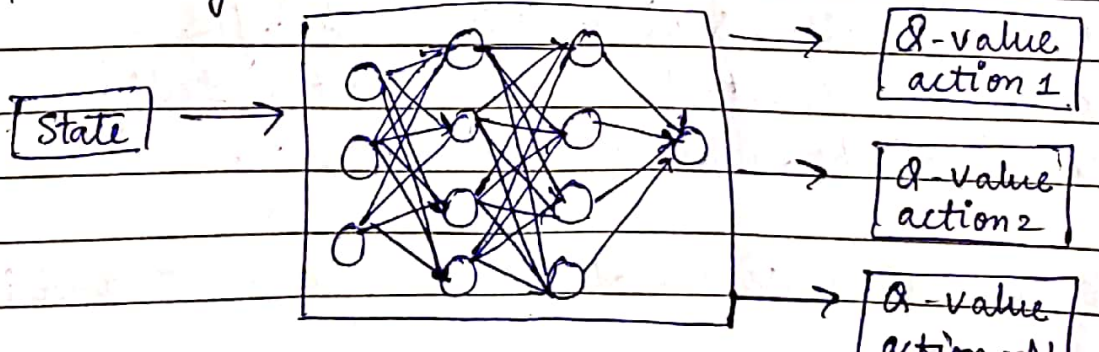
eg -



assign 0 to edges which does not directly lead to goal state.
assign 100 to edges which directly lead to goal state.

Deep Q learning - Deep Q-learning replaces the Q-table with a neural network. Rather than mapping a state action pair to q-value, a neural network maps input state to (action, Q-value) pairs.

Deep Q-learning uses a neural network.



Deep Q-learning

Date / /
Page No.

The basic working step of deep-Q learning is that the initial state is fed into the neural network and it returns the Q-value of all possible actions as an output.