UCS 2436 PRINCIPLES OF PATTERN RECOGNITION

Unit 4

Multilayer Neural Networks

Introduction - Feed forward operation and classification - back propagation algorithm - error surfaces - back propagation as feature mapping - Improving back propagation - Stochastic methods - Stochastic search - Boltzmann Learning - Boltzmann Networks and Graphical models - evolutionary methods - genetic algorithms

* Boltzmann Machines

Origin

→ inspired by neuroscience concepts, such as Hebbian learning

→ They are a form of associative memory, meaning that memories are stored and retrieved as associations between units (neurons)

→ designed as stochastic versions of Hopfield Networks (a type of RNN)

Structure

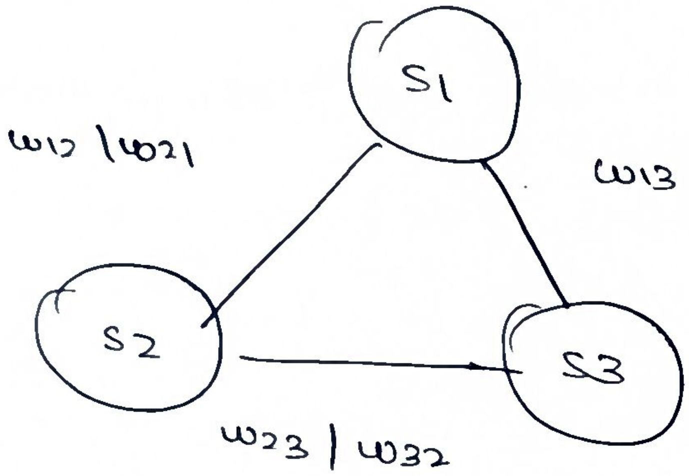
→ a fully-connected network, where every neuron is connected to every other.

→ The state of each node is binary: either 0 ('off') or 1 ('on')

→ Weights between the nodes define the connections (the weights are symmetric $w_{ij} = w_{ji}$)

(i) Positive weights: (excitary constraints) encourage nodes to be in the same state

(ii) Negative states: (inhibitory constraints) discourage nodes from the same state.



* Problems solved with Boltzmann Machines

→ Used to solve two different problems:

(i) Search Problem : weights are fixed, need to find values of the states that minimize the 'energy' of the system.

(ii) Learning Problem : given training data, learn the weights

A. Search Problem → fix a corrupted or partially hidden image

→ The energy of a state corresponds to how well it satisfies these constraints.

~~Step1~~ **Aim**: To find out which of the nodes has lower energy based on the current statements of the other nodes.

$\boxed{\text{Step1}}$ Input to node:

$$z_i = \sum_j s_j w_{ij} + b_i$$

$\boxed{\text{Step2}}$: compute total energy and minimize it

$$E(s) = -\sum s_i b_i - \sum s_i s_j w_{ij}$$

$\boxed{\text{Step3}}$: Stochastically update the state of the node.

$$P(s_i=1) = \frac{1}{1+e^{-z_i}}$$

→ The search problem can be improved with the help of simulated annealing

$$P(s_i=1) = \frac{1}{1+e^{-z_i/T}}$$

→ Temperature parameter T starts large and is reduced over time

Higher Temperature ⇒ gets the states out of local minima, can nudge towards a global optimum

Lower Temperature ⇒ favors lower energy states & converges faster

B. Learning Problem

Learning in Boltzmann machines happens as follows:

Visible and Hidden States

→ In BMs, there are two kinds of nodes or units:

(i) Visible Nodes: represent the data points that you can observe (denoted by V)

(ii) Hidden nodes: capture hidden features and dependencies that aren't directly observable (denoted by H)

Energy Function

→ BMs try to minimize the energy associated with the current configuration of visible and hidden units.

→ The lower the energy, the better the configuration.

→ The energy is given by:

$$E(H,V) = -\sum_i s_i b_i - \sum_{i<j} s_i s_j w_{ij}$$

Probability of a Configuration

→ The probability of a state configuration (H,V) is determined using the Boltzmann distribution.

$$P(H,V) = \frac{1}{Z} e^{-E(H,V)/T}$$

$E(h, v)$ = energy of the configuration

Z = normalization constant called the partition function

T = temperature

Learning Objective

→ Finds weights and biases that make the probability of the data (visible units) as high as possible

→ minimize the difference between the model's predictions and the actual data

data - dependent term = actual data (visible units)

data - independent term = from model's predictions

minimize the difference between

$P_{data}(h, v)$ → from training data and

$P_{model}(v)$ → from estimations

$$\boxed{D = \sum_{v} P_{model}(v) \log \frac{P_{data}(h, v)}{P_{model}(v)}}$$

Training Algorithm

1. Initialize weights and biases randomly.
2. Compute the data-dependent term by fixing visible units to the data.
3. Compute the data-independent term by running the network with random states

4. Update the weights based on the gradient

5. Repeat for all training vectors and update the model iteratively.

Challenges

1. Intractibility - exact computation of the gradient is intractable because it would have to be computed over all possible configurations - which is exponential in size

2. Convexity - If no hidden units are present, the problem is concave, meaning gradient descent will converge to the global minimum

- If hidden units are present, the problem becomes non-concave, and the training may get stuck in local minima.

* Stochastic Search and its Importance

→ Stochastic space refers to optimization techniques that incorporate randomness into the search process to find the solutions to problems.

→ Those search methods use probability and randomness to explore different areas of the solution space.

Characteristics of Stochastic Search

(i) Incorporation of randomness helps avoid getting stuck in local minima

(ii) Stochastic algorithms may probabilistically choose suboptimal paths to explore less obvious solutions. It takes risks to search for better solutions.

Importance of Stochastic Search

(i) avoiding local minima

(ii) exploring the solution space

(iii) handling uncertainty - handle real-world problems better than deterministic methods

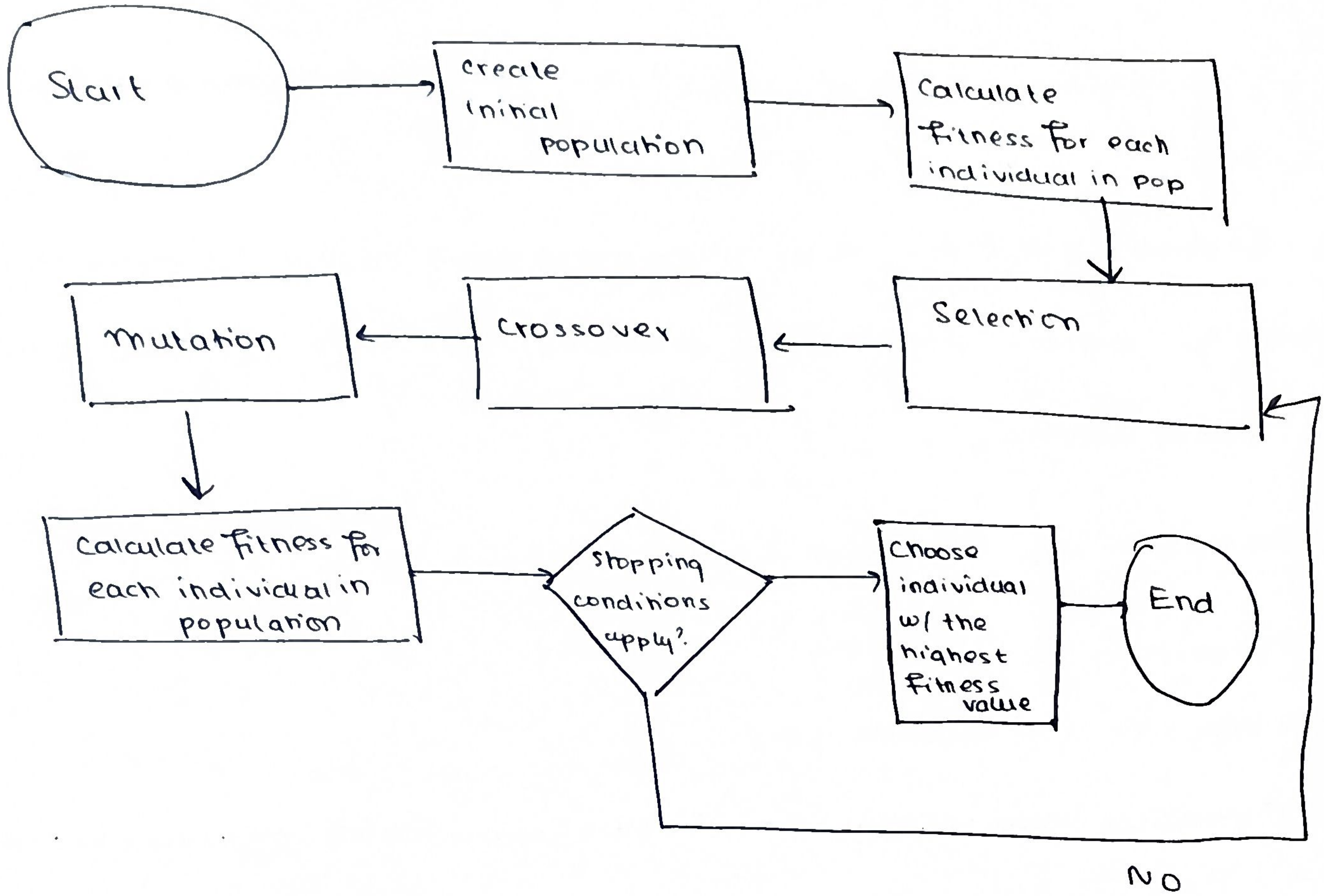
(iv) escape from suboptimal solns.

(v) helps balance exploration and exploitation -

(vi) heuristic and approximate solutions - Stochastic search methods like genetic algorithm or Monte Carlo simulations are useful for generating approximate solutions to NP-hard problems.

* Genetic Algorithms

→ An adaptive heuristic search algorithm that belongs to the larger part of evolutionary algorithm - based on the ideas of natural selection and genetics



Steps

(8) Initialization

→ A population of potential solutions is randomly generated

→ Each individual represents a point in the population space

→ Can be represented as real numbers, integer, binary or permutations etc.

Fitness Assessment

→ Each individual is evaluated using a fitness function, which measures how well it solves the problem

→ Helps determine the suitability of each solution for reproduction

SELECTION

→ Individuals are selected r reproduction based on their fitness scores.

→ Some techniques are:

(i) Roulette wheelselection

(ii) Rank Selection

(iii) Stochastic Unisal Sampling

(iv) Random Select

REPRODUCTION

→ Selected individuals ndergo operation such as:

(i) crossover - cominina traits

(ii) mutation - intoducing random changes

CYCLE

→ The best solutions frm the current generation are retained and carried over to the next generation

→ This ensures that advantageous traits persist throughout the evolution process

TERMINATION

→ Terminates when a stopping criterion is met, such as reaching a maximum number of generations or achieving a satisfactory fitness level

→ The best offspring w/ highest fitness score is selected.

Applications

(i) Optimization problems - TSP, Knapsack problem, scheduling

(ii) Feature selection

(iii) Genetic data analysis

(iv) Path planning