Machine Learning

- SCSA1601

Name: Mohnish Devaraj

Reg NO: 39110636

Assignment-II

PART-A

① a) single layer feed forward neural network with preprocessing.

② A. True

③ D. All of these

④ B. what-if questions

⑤ B. Adaptive learning

PART-B

① **Neural Network**

A neural network is a series of algorithm that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural network refer to systems of neuron, either organic or artifical in nature.

(2) a) hind brain

b) mid brain

c) thalamus

d) hypo thalamus

(3) Perceptron

A perceptron is a neural network until that does certain computations to detect features or business intelligence in the input data.

(4) Feed forward network

A feed-forward neural network is a biological inspired classification algorithm. It consists of a number of simple neuron-like processing units, organized in layers and every unit in a layer is connected with all the units in the previous layer

(5) Aritifical Neural networks (ANN) or neural networks are computational algorithms. It intended to simulate the behaviour of biological systems composed of "neurons". ANNs are computational models inspired by an animals central nervous systems. It is capable of machine learning as well as pattern recognition.

PART - C

① • Introduced by Rumelhart, Hinton & Williams in 1986.

• Multilayer layer Feed forward Network but error is back propogated, Hence the name Back Propogation Network (BPN).

• Uses Supervised Training process

• Systematic procedure for training the network is used

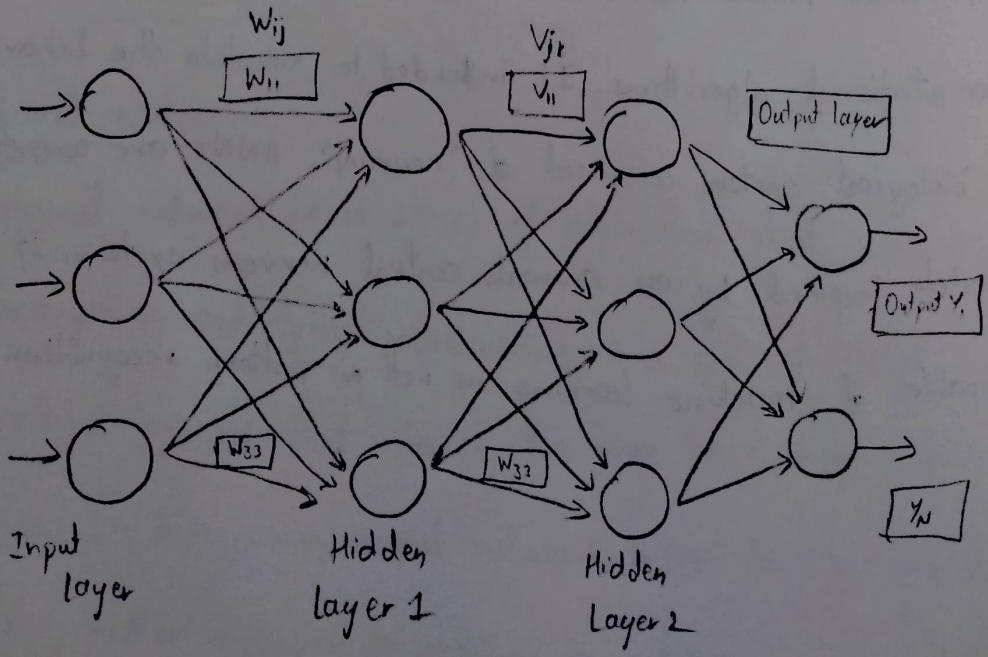
• For Error Detection and Correction Generalized Delta Law/ Continuous Perceptron law/ Gradient Descent law is used.

• Generalized Delta rule minimizes the mean squared error of the output calculated from the output.

• Training by backpropagation involves three (3) stages

1. Feed forward of input training pair
2. Calculation and back propagation of associated error
3. Adjustments of weights

• Different variants of BPN are available for increasing the training speed of the network.



Algorithm:

step 1: Initialize the weights to small random values near zero.

step 2: while stop condition is false, Do step 3 to 10

step 3: For each training pairs do steps 4 to 9

step 4: Each input x_i is received and forwarded to higher layers (next hidden)

step 5: Hidden unit sums its weighted inputs as follows

$$z_{inj} = w_{0j} + \sum x_i w_{ij}$$

Applying Activation function

$$z_j = f(z_{inj})$$

step 6: Output unit sums its weighted output

step 7: $\delta_k = (t_k - y_k) f'(y_{ink})$

step 8: $\delta_{inj} = \sum \delta_i v_{jk}$

step 9: New Weight is

$$w_{ij(new)} = w_{ij(old)} + \Delta w_{ij}$$

$$v_{jk(new)} = v_{jk(old)} + \Delta v_{jk}$$

New bias is

$$w_{0j(new)} = w_{0j(old)} + \Delta w_{0j}$$

$$v_{0k(new)} = v_{0k(old)} + \Delta v_{0k}$$

step 10: Test for stop Condition

Merits:

- Has smooth effect on weight correction
- Computing time is less if weights are small

Demerits:

- Learning phase requires intensive calculations
- Selection of number of Hidden layer neurons is an issue