To implement the Back Propagation algorithm for Feed Forward Neural Networks to learn down gestures from training images and use the trained network to predict the labels for the gestures in the test images.

PROJECT DESCRIPTION

- Programming language used: Python
- Data Structure: Lists, Arrays
- File Name: NeuralNetworks.py
- Inputs: gestures, downgesture_train.list, downgesture_test.list
- Output: Prediction of gestures for test images with accuracy(%)

PART 1 - IMPLEMENTATION

- Modules created:
  - read_pgm(filename, byteorder='>') : Function to convert each PGM image file into readable pixel values which are normalized to be 0 or 1.
  - calMain() – The main task of this module is to read the gesture file from the training list as a list of pixel values in the range 0 through 1 and store in the image as a list variable. This image is used for training the network and initializing the layers. The control flows image by image to the training network from this module.

  - Init_layer(image1, label1) – Performs the base case calculation of setting up the Neural network by initializing weights for hidden and input layers for the given image1 and its predicted target value – label1. The implementation involves both feed forward and back propagation algorithms. During the feed forward step, the sigmoids for each neuron are calculated in the hidden layer of 100 neurons and further for output layer of 1 neuron. The back propagation step involves feeding the output layer sigmoid and updating the weights backwards from output to hidden layer and from hidden to input layer, accordingly for every single image. The termination condition is either when 1000 epochs are completed or the error term of the output perceptron approaches zero. The weights are updated using the formula
    $wijL = wijL + (\eta * \delta jL * XjL\text{-}1)$
    until all the images are passed for training the neural network.

  - predictTest(image1) – It tests the set of gesture images from the set of test images, using the final updated weights and displays the prediction of each gesture along with the accuracy % which signifies the correctness of program.

  - sigmoid(z) – Calculates $\Theta(s)=1/(1+e^{-z})$ for each perceptron.

  - errorCal (z, label) - It returns the error value i.e. the label minus the predicted sigmoid value for each perceptron.

  - Delta1(e,s) - Returns delta value i.e. the derivative of error for the perceptrons from output layer to the hidden layer & hidden layer to input layer Its formula

is:

$$\delta = \text{error} * (\text{Sigmoid}) * (1-(\text{Sigmoid}^{(L)})^2)$$

- Termination Condition:
  For training each image 1000 epochs have been used and apart from that the loop terminates before 1000 epochs if the error between the label and predicted sigmoid value approaches 0. As soon as 184 images are trained the predictions are automatically generated.

- Result Interpretation:
  The first set of output being generated is the training output from the gesturetrain_list images, with its respective sigmoid prediction which approaches 0 for up, hold & stop gestures and approaches 1 for down gestures. Training set is 100% accurate in its prediction and computation. The second set of results is the prediction on test images and found to be **78%accurate** (can be more as well depending on the initial random weights with which the program starts to train) .